Evaluation properties of invariant polynomials

Xavier Dahan [a] Éric Schost [b] Jie Wu [b,c]

[a] *College of Science, Department of Mathematics, Rikkyô university, Tôkyô, Japan*

[b] *Computer Science Department, The University of Western Ontario, London ON*

[c] *The Graduate University of the Chinese Academy of Sciences, Beijing, China*

Abstract

A polynomial invariant under the action of a finite group can be rewritten using generators of the invariant ring. We investigate the complexity aspects of this rewriting process; we show that evaluation techniques enable one to reach a polynomial cost.

1 Introduction

Let $\mathbf{X} = X_1, \ldots, X_n$ be indeterminates over a field $\mathbb{K}$ and let $\mathscr{G}$ be a finite matrix group acting on $\mathbb{K}[\mathbf{X}]$; we denote the ring of invariant polynomials for this action by $\mathbb{K}[\mathbf{X}]^{\mathscr{G}}$. For simplicity, the base field $\mathbb{K}$ has characteristic zero; most results can be extended to finite characteristic p, as long as p does not divide the cardinality of the group $\mathscr{G}$.

Our question. The computational aspects of invariant theory have already been extensively studied; many algorithms are presented in e.g. [22,3]. However, several questions remain open, especially under the complexity viewpoint.

We investigate one such question. We suppose that primary and minimal secondary invariants $\boldsymbol{\pi} = \pi_1, \ldots, \pi_n$ and $\boldsymbol{\sigma} = \sigma_1, \ldots, \sigma_e$ are known for the action of $\mathscr{G}$. Then, any $F \in \mathbb{K}[\mathbf{X}]^{\mathscr{G}}$ can be uniquely written as

$$F = \sum_{\sigma \in \boldsymbol{\sigma}} F_\sigma(\pi_1, \ldots, \pi_n)\sigma, \tag{1}$$

for some F_σ in $\mathbb{K}[\mathbf{P}] = \mathbb{K}[P_1, \ldots, P_n]$. Our question is the cost of this rewriting process; our main result says that working in the *straight-line program* model, the coefficients F_σ can be computed efficiently.

Email addresses: dahan@lix.polytechnique.fr (Xavier Dahan), eschost@uwo.ca (Éric Schost), michael122_wj@hotmail.com (Jie Wu).

A straight-line program is a sequence of instructions $(+, -, \times)$ that computes a (sequence of) polynomial(s); the cost measure is the *size*, *i.e.*, the number of instructions. It has long been known that this representation is well-adapted to obtain complexity results for questions such as multivariate factorization [17], GCD computation [16] and polynomial system solving [12,11,10,14,13,19]. Our work goes in the same direction; a first result along these lines in invariant theory was [7], which dealt with the invariants of the symmetric group.

Hence, we assume that $\boldsymbol{\pi}$, $\boldsymbol{\sigma}$ and F are given in the straight-line representation, and output the coefficients F_σ as straight-line programs as well. We let $\delta = \deg(\pi_1) \cdots \deg(\pi_n) = e|\mathscr{G}|$, where e is the number of secondary invariants.

Theorem 1 *Let $F \in \mathbb{K}[\mathbf{X}]$ (resp. $\boldsymbol{\pi}, \boldsymbol{\sigma}$) be given by a straight-line program Γ of size L (resp. Γ' of size $L_{\boldsymbol{\pi},\boldsymbol{\sigma}}$). Given Γ and Γ', one can construct a straight-line program Γ'' of size*

$$O(n^4\delta^4 + n\delta^6 + L_{\boldsymbol{\pi},\boldsymbol{\sigma}} n\delta^4 \ + \ L\delta^3) \in (L + L_{\boldsymbol{\pi},\boldsymbol{\sigma}})(n\delta)^{O(1)}$$

that computes all polynomials $(F_\sigma)_{\sigma \in \boldsymbol{\sigma}}$.

The construction of Γ'' takes time $O(n^5\delta^4 + n\delta^6 + L_{\boldsymbol{\pi},\boldsymbol{\sigma}} n^2\delta^4 + L\delta^3)$. The construction algorithm is Las Vegas; it chooses $k = O(n^2)$ points in $\mathbb{K}$; choices that lead to failure are contained in a hypersurface of $\overline{\mathbb{K}}^k$.

Comments. Our statement is twofold: the first part is an *existence result*, of a short straight-line representation; the second part expresses the cost of *constructing* it. These aspects are described further in the next section, where we make our computational model more precise.

Our main contribution is a complexity *polynomial* in $n, \delta, L_{\boldsymbol{\pi},\boldsymbol{\sigma}}, L$ (*i.e.*, in $n, e, |\mathscr{G}|, L_{\boldsymbol{\pi},\boldsymbol{\sigma}}, L$). Of course, the questions we discuss can readily be solved using classical Gröbner bases techniques [5]. However, without using the straight-line representation, one can probably not hope for a cost better than $\binom{n+\delta}{n}$, due to the size of the intermediate objects.

Still, our cost is high: our result is first of all a *feasibility* result. Before any serious implementation, it should be refined, at the very least using fast polynomial and matrix arithmetic. The algorithm is probabilistic to ensure its polynomial running time; it can be made deterministic whenever $\boldsymbol{\pi}$ generate the invariant ring of a group $\mathscr{H}$ containing $\mathscr{G}$ (see Proposition 2).

Finally, note that the cost estimates involve two distinct components: one depends only on the group and its invariants $\boldsymbol{\pi}, \boldsymbol{\sigma}$ through the quantities $n, \delta, L_{\boldsymbol{\pi},\boldsymbol{\sigma}}$; the other depends on F through the quantity L. If $\boldsymbol{\pi}, \boldsymbol{\sigma}$ are fixed, a large part of the algorithm becomes a precomputation, and the cost becomes

linear in L. Surprisingly, compared to similar algorithms using the straight-line representation, the degree of F does not appear.

Applications. Colin and Giusti [9] discuss further questions along our lines, with a view towards effective Galois theory. This needs in particular bounds on the complexity of evaluation of the polynomials $A_{i,j,k}$ in the relations

$$\sigma_i \sigma_j = \sum_{k \leq e} A_{i,j,k}(\boldsymbol{\pi}) \sigma_k.$$

Applying Theorem 1 with $F = \sigma_i \sigma_j$ yields an estimate in $L_{\boldsymbol{\pi},\boldsymbol{\sigma}}(n\delta)^{O(1)}$.

Our question is also motivated by applications to polynomial system solving, using algorithms of the *geometric resolution* family [12,11,10,14,13,19]. Such algorithms have a well-understood complexity, that depends on *(i)* geometric quantities and *(ii)* the complexity of evaluation of the input system. If this system is invariant under a group $\mathscr{G}$, a standard approach is to rewrite it using the primary and secondary invariants of $\mathscr{G}$, and solve the system in these new variables [5,2]. However, it is not obvious to quantify the gain of this approach: the output will be more structured, but could be more costly to compute. We bring a partial answer to this question, regarding point *(ii)* above: in the new variables $\boldsymbol{\pi}, \boldsymbol{\sigma}$, the complexity of evaluation of the system, which partially controls the cost of the resolution algorithm, grows moderately. The detailed analysis of this approach is the subject of future work.

Outline of the paper. The first sections are preliminaries: Section 2 introduces our computational model and gives a few basic properties; Section 3 recalls results about zero-dimensional ideals. A key property of Gröbner bases associated to homogeneous systems of parameters is given in Section 4 and is at the basis of our algorithm. The algorithm itself proceeds in two steps: a preparation step, that involves only the group and its invariants (Section 5), and the rewriting process (Section 6). We conclude with preliminary experimental results.

2 Computational model

Two models are used in this paper: *algebraic RAMs* and *straight-line programs*; in that, we follow Kaltofen [16,17].

Straight-line programs. Straight-line programs are a basic model: a sequence of additions and multiplications, without test or branching; this is for instance enough to describe polynomial or matrix multiplication. Precisely, a straight-line program Γ over $\mathbb{K}[\mathbf{X}] = \mathbb{K}[X_1, \ldots, X_n]$ is a list of triples $(\circ_i, \ell_i, r_i)_{1 \leq i \leq L}$, where:

- $\circ_i$ is one of the strings $\{\mathsf{add}, \mathsf{sub}, \mathsf{mul}\}$;
- ℓ_i (resp. r_i) takes one of the forms $(\mathsf{const}, \lambda)$, (var, ℓ) or (g, j), with $\lambda \in \mathbb{K}$, $\ell \in \{1, \ldots, n\}$ or $j \in \{1, \ldots, i-1\}$.

We assign to Γ a list of polynomials $G_1, \ldots, G_L \in \mathbb{K}[\mathbf{X}]$ in a straightforward manner. Assuming that $G_1, \ldots, G_{i-1}$ are known, define $R_i \in \mathbb{K}[\mathbf{X}]$ by

$$R_i = \lambda \text{ if } r_i = (\mathsf{const}, \lambda), \quad R_i = X_\ell \text{ if } r_i = (\mathsf{var}, \ell), \quad R_i = G_j \text{ if } r_i = (\mathsf{g}, j).$$

One defines L_i in a similar manner, and finally obtains G_i by

$$G_i = L_i + R_i \text{ if } \circ_i = \mathsf{add}, \quad G_i = L_i - R_i \text{ if } \circ_i = \mathsf{sub}, \quad G_i = L_i R_i \text{ if } \circ_i = \mathsf{mul}.$$

The polynomials G_i are *given* (or *computed*) by Γ. The complexity measure for straight-line programs is their *size*, *i.e.*, the integer L of the definition.

Obviously, some polynomials such as $(X_1 + 1)^k$ have short straight-line representation (here, logarithmic in k), even though they may have many monomials (here, $k + 1$). The insight of Kaltofen, Heintz, Giusti, Pardo, ... is that this phenomenon arises in many situations, from multivariate factorization and GCD to polynomial system solving. Our work follows their approach.

Algebraic RAMs. Straight-line programs are simple syntactic objects, that can be represented using character strings, integers and field elements and can be manipulated algorithmically in a higher-level computational model. For us, this higher-level model will be *algebraic RAMs*, where usual constructs (testing, branchings, etc) on integers, character strings, or elements of the base field, are allowed. The precise definition being complex, we refer the reader to [16] for details. The cost of an algorithm in this model is simply the number of steps the RAM performs (to make things easier, we do not use the logarithmic cost criterion of [16]).

Hence, our algorithms are written in the RAM model; most take straight-line programs as input and output straight-line programs as well. One should then distinguish between the *size* of the straight-line program we construct, and the *time* it takes to construct it. In many cases, they will be similar (in which case we will be brief on the time analysis), but this is not necessarily so.

Basic results. All the results in this paragraph are well-known. One of our basic operations is linear algebra with matrices whose entries are polynomials given by straight-line programs. Most operations are straightforward, as long as no zero-test or division is involved.

Lemma 1 *Let $\mathscr{M}$ and $\mathscr{M}'$ be matrices of sizes (a, b) and (b, c), with polynomial entries given by a straight-line program Γ of size L. Then one can construct in time $L + O(abc)$ a straight-line program of size $L + O(abc)$ that com-*

putes the same polynomials as Γ*, plus the entries of* $\mathscr{M}\mathscr{M}'$*. If* $(a,b)=(b,c)$*, one can construct in time* $L+O(ab)$ *a straight-line program of size* $L+O(ab)$ *that computes the same polynomials as* Γ*, plus the entries of* $\mathscr{M}+\mathscr{M}'$*.*

Proof. For multiplication, we extend Γ by the $O(abc)$ operations that encode matrix product in size $(a,b)\times(b,c)$. The case of addition is similar. □

Corollary 1 *Let* $\mathscr{M}_1,\ldots,\mathscr{M}_n$ *be matrices of size* δ*, with polynomial entries given by a straight-line program of size* L*. Let further* $F\in\mathbb{K}[X_1,\ldots,X_n]$ *be given by a straight-line program of size* L'*. Then one can construct in time* $L+O(L'\delta^3)$ *a straight-line program of size* $L+O(L'\delta^3)$ *that computes the entries of* $F(\mathscr{M}_1,\ldots,\mathscr{M}_n)$*.*

Solving linear systems is more delicate, since it involves zero-tests and divisions. Berkowitz' algorithm [1] provides the following result.

Lemma 2 *Let* $\mathscr{M}$ *be a matrix of size* δ*, with polynomial entries in given by a straight-line program* Γ *of size* L*. One can construct in time* $L+O(\delta^4)$ *a straight-line program of size* $L+O(\delta^4)$ *that computes the same polynomials as* Γ*, plus the determinant of* $\mathscr{M}$ *and the entries of its adjoint matrix. If the determinant of* $\mathscr{M}$ *is known to be in* $\mathbb{K}-\{0\}$*, then the same result holds for computing the entries of* $\mathscr{M}^{-1}$*.*

Finally, we show how to deal with divisions using Strassen's *Vermeidung von Divisionen* [21]: divisions are replaced by truncated power series computation. Consider some rational functions $\mathbf{F}=F_1,\ldots,F_r$ in $\mathbb{K}(\mathbf{X})$, with no denominator vanishing at $\mathbf{0}$, so that we can write $F_i=\sum_{j\geq 0}F_{i,j}$ in $\mathbb{K}[[\mathbf{X}]]$, with $F_{i,j}$ homogeneous of degree j. Let also $\mathbf{G}=G_0,\ldots,G_s$ be in $\mathbb{K}[Y_1,\ldots,Y_r]$; then, assuming that $G_0(\mathbf{F})$ does not vanish at $\mathbf{0}$, the rational functions $H_i=G_i(\mathbf{F})/G_0(\mathbf{F})$ can be expanded in power series as well: $H_i=\sum_{j\geq 0}H_{i,j}$ in $\mathbb{K}[[\mathbf{X}]]$, with $H_{i,j}$ homogeneous of degree j. What can be computed here are only truncations of the series H_i.

Lemma 3 *Notation being as above, suppose that* $G_0,\ldots,G_s$ *are given by a straight-line program of size* L*. Suppose also that all* $F_{i,j}$*, for* $j\leq\kappa$*, can be computed by a straight-line program of size* L'*. Then one can construct in time* $L'+O(L\kappa^2)$ *a straight-line program of size* $L'+O(L\kappa^2)$ *that computes all* $H_{i,j}$*, for* $j\leq\kappa$*.*

3 Preliminaries on zero-dimensional systems

We start this section with a general discussion on zero-dimensional ideals. Let thus I be a zero-dimensional radical ideal in the ring $\mathbb{K}[\mathbf{X}]$, where $\mathbb{K}$ is a

perfect field and $\mathbf{X} = X_1, \ldots, X_n$; let further $\Delta = \dim_{\mathbb{K}} \mathbb{K}[\mathbf{X}]/I$.

A *primitive element* for I is a linear form $u = \sum_{i \leq n} u_i X_i$, with $u_1, \ldots, u_n$ in $\mathbb{K}$, such that the powers $1, u, \ldots, u^{\Delta-1}$ are a $\mathbb{K}$-basis of $\mathbb{K}[\mathbf{X}]/I$; hence $\mathbb{K}[\mathbf{X}]/I$ is isomorphic to $\mathbb{K}[U]/\langle T \rangle$, where T is the monic minimal polynomial of u in $\mathbb{K}[\mathbf{X}]/I$. In particular, there exist polynomials $S_i \in \mathbb{K}[U]$, with $\deg(S_i) < \Delta$, such that $X_i = S_i(u)$ in $\mathbb{K}[\mathbf{X}]/I$.

Primitive elements always exist, assuming that $|\mathbb{K}|$ is large enough; we call the data of the linear form u and of the polynomials $T, S_1, \ldots, S_n$ a *shape lemma representation* of I [8]. Once $u_1, \ldots, u_n$ are fixed, these polynomials are uniquely defined.

Suppose now that we are given a basis $\mathbf{m}$ of $\mathbb{K}[\mathbf{X}]/I$, which is the set of standard monomials for a term order $>$ on $\mathbb{K}[\mathbf{X}]$. Given a shape lemma representation of I, we will be interested in Section 5 in finding the multiplication matrices $\mathscr{M}_{X_i}$ by $X_1, \ldots, X_n$ in the basis $\mathbf{m}$. The entries of these matrices are *rational functions* of the coefficients of $T, S_1, \ldots, S_n$. Since we are interested in the straight-line complexity, we compute numerators and denominators separately.

Lemma 4 *Given* $\mathbf{m}$*, one can construct in time* $O(n\Delta^4)$ *a straight-line program of size* $O(n\Delta^4)$ *that takes as input the coefficients of* $T, S_1, \ldots, S_n$ *and outputs a polynomial* $\mathscr{D}$ *and the entries of polynomial matrices* $\mathscr{M}'_{X_1}, \ldots, \mathscr{M}'_{X_n}$*, such that* $\mathscr{M}_{X_i} = \mathscr{M}'_{X_i}/\mathscr{D}$*.*

Proof. Let $\mathscr{C}$ be the companion matrix of T; its entries can be computed by a straight-line program of size $O(n)$ that changes the sign of the coefficients of T. In the basis $1, u, \ldots, u^{\Delta-1}$, the multiplication matrix by X_i is $\mathscr{N}_{X_i} = S_i(\mathscr{C})$. Since each S_i has degree less than Δ, using Horner's evaluation scheme and Lemma 1, we can thus construct a straight-line program of size $O(n\Delta^4)$ that computes all entries of all these matrices.

Next, we compute the coefficient vectors of the elements of $\mathbf{m}$ in the basis $1, u, \ldots, u^{\Delta-1}$. By assumption, for any $m \neq 1$ in $\mathbf{m}$, there exists $i \leq n$ and m' in $\mathbf{m}$ such that $m = X_i m'$. Starting from the vector $[1\ 0\ \cdots\ 0]^t$ corresponding to the monomial $1 \in \mathbf{m}$, we obtain all other ones inductively, by multiplication by the appropriate matrix $\mathscr{N}_{X_i}$. Using Lemma 1, this can be done by extending our previous straight-line program with $O(\Delta^3)$ operations.

Let finally $\mathscr{B}$ be the matrix obtained as the concatenation of all these vectors. It follows that the multiplication matrix $\mathscr{M}_{X_i}$ in $\mathbf{m}$ is $\mathscr{B}^{-1}\mathscr{N}_{X_i}\mathscr{B}$. Computing the adjoint and determinant of $\mathscr{B}$ by Lemma 2 and multiplying the adjoint by $\mathscr{N}_{X_i}\mathscr{B}$ by Lemma 1, we conclude the proof. □

4 A property of the graph ideal

Starting from the set of primary invariants $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$, we study in this section the ideal of the graph of $\mathbf{x} \mapsto \boldsymbol{\pi}(\mathbf{x})$. Let thus $\mathbf{P} = P_1, \ldots, P_n$ be new variables and consider the ideal

$$J = \langle \pi_1 - P_1, \ldots, \pi_n - P_n \rangle \subset \mathbb{K}[\mathbf{P}, \mathbf{X}].$$

We give here properties of some of the Gröbner bases of J. In our context, $\boldsymbol{\pi}$ are primary invariants for the action of $\mathscr{G}$, but all results in this section hold whenever $\boldsymbol{\pi}$ is a homogeneous system of parameters in $\mathbb{K}[\mathbf{X}]$.

In what follows, we let $>$ be a degree-compatible monomial order on $\mathbb{K}[\mathbf{X}]$; we put on $\mathbb{K}[\mathbf{P}, \mathbf{X}]$ a block order $>'$ with $X_i > P_j$ for all i, j and which extends the order $>$ defined on $\mathbb{K}[\mathbf{X}]$.

Theorem 2 *Let $\mathbf{G} \subset \mathbb{K}[\mathbf{P}, \mathbf{X}]$ be the reduced Gröbner basis of J for the order $>'$. Then all leading terms of the polynomials in $\mathbf{G}$ are in $\mathbb{K}[\mathbf{X}]$.*

This result requires $>$ to be a degree order: for a lexicographic order, it fails with $n = 2$, for $\pi_1 = X_1^2 + X_2^2$ and $\pi_2 = X_1 X_2$. A similar result is given in [23] for the degree reverse lexicographic order on $\mathbb{K}[\mathbf{P}, \mathbf{X}]$.

Before the proof, we discuss a useful consequence. For $\mathbf{p} = (p_1, \ldots, p_n)$ in $\mathbb{K}^n$, let $J_{\mathbf{p}}$ be the ideal $\langle \pi_1 - p_1, \ldots, \pi_n - p_n \rangle \subset \mathbb{K}[\mathbf{X}]$. In all that follows, we let $\mathbf{m}$ be the standard monomials for the zero-dimensional ideal $J_{\mathbf{0}} = \langle \pi_1, \ldots, \pi_n \rangle$, for the order $>$. Our main application of the previous theorem will be the following corollary, where we recall that $\delta = \deg(\pi_1) \cdots \deg(\pi_n)$.

Corollary 2 *The set $\mathbf{m}$ is simultaneously the set of standard monomials for all ideals $J_{\mathbf{p}}$, and for the ideal $J \cdot \mathbb{K}(\mathbf{P})[\mathbf{X}]$, for the order $>$. Besides, $\mathbf{m}$ is a $\mathbb{K}[\boldsymbol{\pi}]$-module basis of $\mathbb{K}[\mathbf{X}]$, so that $|\mathbf{m}| = \delta$.*

Proof. By the choice of our term order, $\mathbf{G}$ remains a Gröbner basis in $\mathbb{K}[\mathbf{P}][\mathbf{X}]$. Theorem 3.1 in [15] then shows that for any field $\mathbb{L}$ containing $\mathbb{K}$ and any $\mathbf{p}' = (p'_1, \ldots, p'_n)$ in $\mathbb{L}$, the specialization $\{G(\mathbf{p}, \mathbf{X}) \mid G \in \mathbf{G}\}$ is the reduced Gröbner basis of $\langle \pi_1 - p'_1, \ldots, \pi_n - p'_n \rangle \subset \mathbb{L}[\mathbf{X}]$ for the order $>$. This proves the first part of the corollary. The second part follows from the graded Nakayama Lemma [3, Lemma 3.5.1]. □

The rest of this section is devoted to prove Theorem 2.

Lemma 5 *The set*

$$\mathbf{m}' = \{m P_1^{\alpha_1} \cdots P_n^{\alpha_n} \mid m \in \mathbf{m},\ \alpha_1, \ldots, \alpha_n \geq 0\}$$

is a basis of $\mathbb{K}[\mathbf{P}, \mathbf{X}]/J$.

PROOF. First we prove that $\mathbf{m}'$ is linearly independent. A linear relation

$$\sum_{m \in \mathbf{m}} m \sum_j a_{m,j} P_1^{\alpha_{1,m,j}} \cdots P_n^{\alpha_{n,m,j}} = 0 \mod J,$$

with $a_{m,j} \in \mathbb{K}$, implies the equality

$$\sum_{m \in \mathbf{m}} m \sum_j a_{m,j} \pi_1^{\alpha_{1,m,j}} \cdots \pi_n^{\alpha_{n,m,j}} = 0.$$

By the graded Nakayama Lemma, $\mathbf{m}$ is a module basis of $\mathbb{K}[\mathbf{X}]$ over $\mathbb{K}[\boldsymbol{\pi}]$, so all coefficients

$$\sum_j a_{m,j} \pi_1^{\alpha_{1,m,j}} \cdots \pi_n^{\alpha_{n,m,j}}$$

are zero. The algebraic independence of $\boldsymbol{\pi}$ implies that all $a_{m,j}$ are zero, as requested. We conclude by proving that $\mathbf{m}'$ generates $\mathbb{K}[\mathbf{P}, \mathbf{X}]/J$. Since $\mathbf{m}$ is a $\mathbb{K}[\boldsymbol{\pi}]$-basis of $\mathbb{K}[\mathbf{X}]$, any H in $\mathbb{K}[\mathbf{X}]$ can be written in the form

$$H = \sum_{m \in \mathbf{m}} a_m(\boldsymbol{\pi}) m,$$

with $a_m \in \mathbb{K}[\mathbf{P}]$, so that in $\mathbb{K}[\mathbf{P}, \mathbf{X}]$, we have the equality

$$H = \sum_{m \in \mathbf{m}} a_m m \mod J.$$

This in turn implies the same statement for arbitrary H in $\mathbb{K}[\mathbf{P}, \mathbf{X}]$. □

To prove our claim on the leading terms of the polynomials in $\mathbf{G}$, we actually prove that $\mathbf{G}$ specializes well at $\mathbf{P} = \mathbf{0}$. As before, we thus let $J_{\mathbf{0}}$ be the ideal $\langle \pi_1, \ldots, \pi_n \rangle \subset \mathbb{K}[\mathbf{X}]$, and let $\mathbf{H}$ be its reduced Gröbner basis for the order $>$ on $\mathbb{K}[\mathbf{X}]$. Then, we actually show the following: *all leading terms of the polynomials in* $\mathbf{G}$ *belong to* $\mathbb{K}[\mathbf{X}]$, *and* $\mathbf{H} = \{G(\mathbf{0}, \mathbf{X}) \mid G \in \mathbf{G}\}$. This contains in particular the statement of Theorem 2.

The generators of J are weighted homogeneous, with $w(X_i) = 1$ and $w(P_j) = \deg(\pi_j)$. For such a weighted homogeneous polynomial G in $\mathbb{K}[\mathbf{P}, \mathbf{X}]$, $G(\mathbf{0}, \mathbf{X}) \neq 0$ is equivalent to $\mathrm{lt}(G) \in \mathbb{K}[\mathbf{X}]$, where $\mathrm{lt}(.)$ denotes the leading term. It is also straightforward to see that $G(\mathbf{0}, \mathbf{X})$ is in $J_{\mathbf{0}}$ for all G in J.

The following statement will be crucial to the proof: *for* H *homogeneous in* $J_{\mathbf{0}}$, *there exists* G *in* $\mathbf{G}$ *such that* $\mathrm{lt}(G)$ *divides* $\mathrm{lt}(H)$. Indeed, write $H = h_1\pi_1 + \cdots + h_n\pi_n$. Defining $H' = h_1(\pi_1 - P_1) + \cdots + h_n(\pi_n - P_n)$, the choice of our monomial order in $\mathbb{K}[\mathbf{P}, \mathbf{X}]$ implies that H and H' have the same leading term. In particular, since H' is in J, there exists $G \in \mathbf{G}$ such that $\mathrm{lt}(G)$ divides

$\mathrm{lt}(H') = \mathrm{lt}(H)$, as claimed. Using this point, we prove the equality

$$\{\mathrm{lt}(H) \mid H \in \mathbf{H}\} \;=\; \{\mathrm{lt}(G) \mid G \in \mathbf{G} \text{ and } G(\mathbf{0}, \mathbf{X}) \neq 0\}. \tag{2}$$

- Let G be in $\mathbf{G}$. If $G_{\mathbf{0}} = G(\mathbf{0}, \mathbf{X}) \neq 0$, then its leading term can be divided by the leading term of some polynomial H in $\mathbf{H}$. Since $J_{\mathbf{0}}$ is homogeneous, H is homogeneous, so by the preliminary remark, there exists $G' \in \mathbf{G}$ such that $\mathrm{lt}(G')$ divides $\mathrm{lt}(H)$, and thus $\mathrm{lt}(G_{\mathbf{0}})$, which equals $\mathrm{lt}(G)$. Because $\mathbf{G}$ is a reduced basis, we deduce that $G = G'$, and that in particular G and H have the same leading term.
- Conversely, let H be in $\mathbf{H}$. As before, there must exist $G \in \mathbf{G}$ such that $\mathrm{lt}(G)$ divides $\mathrm{lt}(H)$. In particular, $\mathrm{lt}(G)$ is in $\mathbb{K}[\mathbf{X}]$, which implies by the previous point that $\mathrm{lt}(G)$ is in $\mathrm{lt}(\mathbf{H})$. Since $\mathbf{H}$ is a reduced Gröbner basis, $\mathrm{lt}(G) = \mathrm{lt}(H)$, as claimed.

Next, we prove that (2) implies the further equality

$$\mathbf{H} = \{G(\mathbf{0}, \mathbf{X}) \mid G \in \mathbf{G} \text{ and } G(\mathbf{0}, \mathbf{X}) \neq 0\}.$$

Indeed, let G be in $\mathbf{G}$, with $G_{\mathbf{0}} = G(\mathbf{0}, \mathbf{X}) \neq 0$, and let H be in $\mathbf{H}$, with $\mathrm{lt}(H) = \mathrm{lt}(G) = \mathrm{lt}(G_{\mathbf{0}}) = \ell$. In view of Equality (2), since $G - \ell$ is reduced with respect to $\mathbf{G}$, $G_{\mathbf{0}} - \ell$ is reduced with respect to $\mathbf{H}$. Similarly, $H - \ell$ is reduced with respect to $\mathbf{H}$; hence $G_{\mathbf{0}} - H = (G_{\mathbf{0}} - \ell) - (H - \ell)$ is in $J_{\mathbf{0}}$, but reduced with respect to $\mathbf{H}$. Hence, $G_{\mathbf{0}} = H$, proving our claim.

The last thing to show is that for all $G \in \mathbf{G}$, $G(\mathbf{0}, \mathbf{X}) \neq 0$, or equivalently that $\mathrm{lt}(G)$ is in $\mathbb{K}[\mathbf{X}]$. As in Lemma 5, define

$$\mathbf{m}' = \{m P_1^{\alpha_1} \cdots P_n^{\alpha_n} \mid m \in \mathbf{m},\ \alpha_1, \ldots, \alpha_n \geq 0\}.$$

Let next $\mathbf{n} \subset \mathbb{K}[\mathbf{P}, \mathbf{X}]$ be the standard monomials modulo $\mathbf{G}$. Since $\mathrm{lt}(\mathbf{H}) \subset \mathrm{lt}(\mathbf{G})$, we have the inclusion $\mathbf{n} \subset \mathbf{m}'$. Besides, Lemma 5 proves that $\mathbf{m}'$ is a basis of $\mathbb{K}[\mathbf{P}, \mathbf{X}]/J$; hence, $\mathbf{n} = \mathbf{m}'$. This is enough to conclude: suppose that there is G in $\mathbf{G}$ with a leading term $\ell = \ell_{\mathbf{X}} \ell_{\mathbf{P}}$, with $\ell_{\mathbf{X}}$ and $\ell_{\mathbf{P}}$ in respectively $\mathbb{K}[\mathbf{X}]$ and $\mathbb{K}[\mathbf{P}]$, and $\ell_{\mathbf{P}} \neq 1$. Since $\mathbf{G}$ is reduced, $\ell_{\mathbf{X}}$ is reduced with respect to $\mathrm{lt}(\mathbf{H})$, so $\ell_{\mathbf{X}}$ is in $\mathbf{m}$ and ℓ is in $\mathbf{m}'$. Since $\mathbf{m}' = \mathbf{n}$, we have a contradiction.

5 Computing the multiplication matrices

The first component of our algorithm is introduced now: the computation of the $\mathbb{K}[\boldsymbol{\pi}]$-module basis $\mathbf{m}$ of $\mathbb{K}[\mathbf{X}]$, together with the multiplication matrices in this basis. Precisely, we let $\mathscr{M}_{X_1}, \ldots, \mathscr{M}_{X_n}$ be matrices in $\mathbb{K}[\mathbf{P}]$ such that $\mathscr{M}_{X_i}(\boldsymbol{\pi})$ is the multiplication matrix by X_i in the $\mathbb{K}[\boldsymbol{\pi}]$-module $\mathbb{K}[\mathbf{X}]$. Since we are interested in the cost of this process, we have to pay attention to the

algorithms and data structures. Indeed, the entries of these multiplication matrices are multivariate polynomials of (weighed) degree $O(\delta)$ in n variables $P_1, \ldots, P_n$, so they may involve up to $\binom{n+\delta}{n}$ monomials.

Hence, there is no hope to obtain a cost in $(n\delta)^{O(1)}$ using the dense polynomial representation: there are too many monomials. The straight-line representation becomes useful here, as we will use straight-line programs to represent the coefficients of the multiplication matrices. The main result of this section shows that one can construct such straight-line programs of size $(n\delta)^{O(1)}$.

Theorem 3 *Suppose that the polynomials $\boldsymbol{\pi}$ are given by a straight-line program of size $L_{\boldsymbol{\pi}}$. There exists a Las Vegas probabilistic algorithm that performs the following tasks in time $O(n^5\delta^4 + n\delta^6 + L_{\boldsymbol{\pi}} n^2\delta^4)$:*

- *determine the basis $\mathbf{m}$;*
- *construct a straight-line program of size $O(n^4\delta^4 + n\delta^6 + L_{\boldsymbol{\pi}} n\delta^4)$ that computes all entries of all matrices $\mathscr{M}_{X_i}$.*

The algorithm chooses $k = O(n^2)$ points in $\mathbb{K}$; choices that lead to failure are contained in a hypersurface of $\overline{\mathbb{K}}^k$.

As before, J is the ideal $\langle \pi_1 - P_1, \ldots, \pi_n - P_n \rangle$ of $\mathbb{K}[\mathbf{P}, \mathbf{X}]$ and for $\mathbf{p} \in \mathbb{K}^n$, $J_{\mathbf{p}}$ is the ideal $\langle \pi_1 - p_1, \ldots, \pi_n - p_n \rangle$ of $\mathbb{K}[\mathbf{X}]$. To obtain the matrices $\mathscr{M}_{X_i}$, we use *lifting techniques*: starting from a description of $V(J_{\mathbf{p}})$, for a generic enough $\mathbf{p}$, we obtain an approximation of a description of $V(J \cdot \mathbb{K}(\mathbf{P})[\mathbf{X}])$. These descriptions are *shape lemma* representations, which are convenient for computations. Using the results of the previous section, we conclude with a change of basis as in [4]. The same techniques could give the Gröbner basis $\mathbf{G}$ of the last section for a similar cost, but $\mathbf{G}$ is not required below.

5.1 Lifting fibers

We say that a point $\mathbf{p} \in \mathbb{K}^n$ is a *lifting point* if $V(J_{\mathbf{p}})$ has cardinality δ. For $\mathbf{p}$ a lifting point, we call *lifting fiber* the data of $\mathbf{p}$, together with a shape lemma representation of $V(J_{\mathbf{p}})$ [13]. In this subsection, we discuss the cost of computing a lifting fiber. For such zero-dimensional situations, there is no need for us to use a straight-line program representation, since a lifting fiber only involves $O(n\delta)$ monomials.

First, let D be the Jacobian determinant of $(\pi_1 - P_1, \ldots, \pi_n - P_n)$ with respect to $\mathbf{X}$. The Jacobian criterion gives us a criterion for $\mathbf{p}$ to be a lifting point.

Lemma 6 *The point $\mathbf{p}$ is a lifting point if and only if $D(\mathbf{p}, \mathbf{X})$ vanishes nowhere on $V(J_{\mathbf{p}})$.*

PROOF. By Corollary 2, the dimension of the quotient $\mathbb{K}[\mathbf{X}]/J_{\mathbf{p}}$ is δ for all $\mathbf{p}$. Hence, $V(J_{\mathbf{p}})$ has cardinality δ if and only if $J_{\mathbf{p}}$ is radical, and the result follows from the Jacobian criterion. □

As a consequence, the points $\mathbf{p}$ that are *not* lifting points are contained in the projection of $V(D) \cap V(J) \subset \overline{\mathbb{K}}^{2n}$ on the $\mathbf{P}$-space; by Bézout's theorem, this projection is contained in a hypersurface of $\overline{\mathbb{K}}^n$ of degree at most $\delta\eta$, with $\eta = \deg(\pi_1) + \cdots + \deg(\pi_n)$.

Even in this dimension zero case, we know no deterministic algorithm with a cost in $\delta^{O(1)}$ for computing a lifting fiber. The following proposition reaches the required complexity, at the cost however of becoming probabilistic.

Proposition 1 *Suppose that the polynomials $\boldsymbol{\pi}$ are given by a straight-line program of size $L_{\boldsymbol{\pi}}$. There exists a Las Vegas probabilistic algorithm that computes a lifting fiber in time $O(n^5\delta^4 + L_{\boldsymbol{\pi}} n^2 \delta^4)$. The algorithm chooses $k = O(n^2)$ points in $\mathbb{K}$; choices that lead to failure are contained in a hypersurface of $\overline{\mathbb{K}}^k$.*

PROOF. We simply pick $\mathbf{p}$ at random (the previous remark shows that a generic $\mathbf{p}$ is a lifting point). If $\mathbf{p}$ is a lifting point, the system $(\pi_1 - p_1, \ldots, \pi_n - p_n)$ defines a regular reduced sequence; hence, it satisfies the assumptions of Theorem 1 in [13], which gives our complexity estimate (and yields another source of probabilistic behavior). □

To conclude this subsection, we discuss the special case where the polynomials $\boldsymbol{\pi}$ generate the invariant ring of a group $\mathscr{H}$ containing $\mathscr{G}$. In this case, a straightforward deterministic algorithm is available (as a consequence, our main algorithm becomes deterministic as well). We start with a lemma on the deterministic avoidance of hyperplanes.

Lemma 7 *Let $\ell_1, \ldots, \ell_k$ be non-zero linear forms $\mathbb{K}^n \to \mathbb{K}$, and let $\alpha_1, \ldots, \alpha_k$ be in $\mathbb{K}^k$. One can find $\mathbf{x} \in \mathbb{K}^n$ with $\ell_i(\mathbf{x}) \neq \alpha_i$ for all i in time $O(nk)$.*

PROOF. We determine one coordinate of $\mathbf{x}$ at a time. To find x_n, we first inspect whether there are some linear forms ℓ_i of the form $\ell_i = \lambda_i x_n$. In this case, we need to choose a value x_n different from all corresponding α_i/λ_i. This is done in time $O(k)$: we scan the sequence of values α_i/λ_i and mark all integers in $0, \ldots, k$ that appear in it; one such integer i_0 will be left unmarked, and we let $x_n = i_0$. We update all remaining linear forms and continue recursively. □

Proposition 2 *Suppose that $\mathbb{K}[\boldsymbol{\pi}] = \mathbb{K}[\mathbf{X}]^{\mathscr{H}}$, for some finite matrix group $\mathscr{H}$. Then one can find a lifting fiber for J in time $O(n\delta^2 + n^2\delta)$.*

PROOF. For $\mathbf{x}_0 \in \mathbb{K}^n$ and $\mathbf{p} = \boldsymbol{\pi}(\mathbf{x}_0)$, the variety $V(J_{\mathbf{p}})$ is precisely the orbit of $\mathbf{x}_0$ under the action of $\mathscr{H}$. Hence, by Lemma 6, $\mathbf{p}$ is a lifting point if and

only $\mathbf{x}_0$ is fixed under none of the elements $h \in \mathscr{H}$. For any h in $\mathscr{H}$, the fixed points of h (*i.e.*, the eigenspace for the eigenvalue 1) are contained in a hyperplane whose equation can be determined in time $O(n^2)$. Since $\delta = |\mathscr{H}|$, the total cost is $O(n^2\delta)$. Then, by Lemma 7, one can find a point $\mathbf{x}_0$ such that the associated $\mathbf{p}$ is a lifting point in time $O(n\delta)$.

Knowing $\mathbf{x}_0$, one can then determine its orbit using the matrices in $\mathscr{H}$, in time $O(n^2\delta)$. Next, we determine a linear form u such that $u(\alpha) \neq u(\alpha')$ for $\alpha \neq \alpha'$ in $V(J_\mathbf{p})$; such a linear form is then a primitive element for $J_\mathbf{p}$. To do so, remark that the inequalities $u(\alpha) \neq u(\alpha')$ impose $O(\delta^2)$ constraints on the coefficients of u; by Lemma 7, we can find a suitable u in time $O(n\delta^2)$. Once u is known, $T, S_1, \ldots, S_n$ are obtained by Lagrange interpolation. □

5.2 Finding the matrices by lifting techniques

The algorithm now follows the ideas initiated in [10,11,14], even though our output (multiplication matrices in the basis $\mathbf{m}$) is different; proofs of the next few statements can be found there as well.

Let $\mathbf{p}$, $u = \sum_{i \leq n} u_i X_i$ and $T_\mathbf{p}, S_{1,\mathbf{p}}, \ldots, S_{n,\mathbf{p}} \in \mathbb{K}[U]$ be the lifting fiber obtained in the previous subsection. Then, u is also a primitive element for the maximal ideal $J \cdot \mathbb{K}(\mathbf{P})[\mathbf{X}]$, and we let $T, S_1, \ldots, S_n \in \mathbb{K}(\mathbf{P})[U]$ be the corresponding shape lemma representation. No denominator in the coefficients of these polynomials vanishes at $\mathbf{p}$, and they satisfy the specialization property

$$T(\mathbf{p}, U) = T_\mathbf{p}(U),\ S_1(\mathbf{p}, U) = S_{1,\mathbf{p}}(U),\ \ldots,\ S_n(\mathbf{p}, U) = S_{n,\mathbf{p}}(U).$$

Hence, all coefficients in $T, S_1, \ldots, S_n$ admit power series expansions at $\mathbf{p}$. Decomposing these power series in their homogeneous components, we have

$$T = U^\delta + \sum_{j=0}^{\delta-1} \sum_{\kappa \geq 0} T_{j,\kappa} U^j, \quad S_i = \sum_{j=0}^{\delta-1} \sum_{\kappa \geq 0} S_{i,j,\kappa} U^j,$$

with $T_{j,\kappa}$ and $S_{i,j,\kappa}$ in $\mathbb{K}[P_1 - p_1, \ldots, P_n - p_n]$, homogeneous of degree κ. The following proposition gives a cost estimate for computing truncations of these expansions modulo arbitrary powers of $\langle P_1 - p_1, \ldots, P_n - p_n \rangle$.

Proposition 3 *Suppose that the polynomials $\boldsymbol{\pi}$ are given by a straight-line program of size $L_{\boldsymbol{\pi}}$. Given a lifting fiber for J and an integer $k \geq 0$, one can construct in time $O(n^4k^2\delta^2 + L_{\boldsymbol{\pi}}nk^2\delta^2)$ a straight-line program of size $O(n^4k^2\delta^2 + L_{\boldsymbol{\pi}}nk^2\delta^2)$ that evaluates all coefficients $T_{j,\kappa}$ and $S_{i,j,\kappa}$, for $\kappa < k$.*

Proof. This is a classical application of lifting techniques as in [10,11,14]. However, these references give worse complexity estimates, so we briefly in-

dicate here how to obtain the requested cost. We recall first the results of Section 4.3 in [13]; this describes a situation similar to ours, but with only one free variable Z to lift.

Let $\mathbb{F}$ be a field and let $\mathbf{F} = (F_1, \ldots, F_n)$ be in $\mathbb{F}[Z, X_1, \ldots, X_n]$. As above, let $u = \sum u_i X_i$ and let $\tau, \zeta_1, \ldots, \zeta_n$ be polynomials in $\mathbb{F}(Z)[U]$, with τ monic of degree d, such that $F_i(Z, \zeta_1, \ldots, \zeta_n) = 0 \bmod \tau$ for all i. Suppose also that $\iota(0, \zeta_1, \ldots, \zeta_n)$ is invertible modulo τ, where ι is the Jacobian determinant of $\mathbf{F}$ in $\mathbf{X}$. Finally, assume that no denominator in $\tau, \zeta_1, \ldots, \zeta_n$ vanishes at $Z = 0$, so all coefficients of these polynomials admit series expansions at $Z = 0$:

$$\tau = U^d + \sum_{j=0}^{d-1} \sum_{\kappa \geq 0} \tau_{j,\kappa} Z^\kappa U^j, \quad \zeta_i = \sum_{j=0}^{d-1} \sum_{\kappa \geq 0} \zeta_{i,j,\kappa} Z^\kappa U^j.$$

Then, if the polynomials $\mathbf{F}$ are given by a straight-line program $\Gamma_{\mathbf{F}}$ of size $L_{\mathbf{F}}$, Lemma 2 in [13] shows that all coefficients $\tau_{j,\kappa}$ and $\zeta_{i,j,\kappa}$, for $\kappa < k$, can be computed by a straight-line program of size $O(n^4 k^2 \delta^2 + L_{\mathbf{F}} n k^2 \delta^2)$, taking as input $\Gamma_{\mathbf{F}}$, u and $\tau(0, U), \zeta_1(0, U), \ldots, \zeta_n(0, U)$. The construction of this straight-line program can be done in the same cost.

Let us apply this result to our problem. Let Z be a new variable, let $\mathbb{F}$ be $\mathbb{K}(\mathbf{P})$ and let $P_i' = p_i + Z(P_i - p_i)$, for $i \leq n$. The polynomials F_i given by $F_i(Z, \mathbf{X}) = \pi_i - P_i'$ satisfy the assumptions of the previous paragraph, with $\tau(Z, U) = T(P_1', \ldots, P_n', U)$ and $\zeta_i(Z, U) = S_i(P_1', \ldots, P_n', U)$. For $j \leq \delta$ and $i \leq n$, we have by construction $\tau_{j,\kappa} = T_{j,\kappa}$ and $\zeta_{i,j,\kappa} = S_{i,j,\kappa}$. Hence, the statement of the previous paragraph concludes the proof. □

We finally show how to compute the multiplication matrices $\mathscr{M}_{X_i}$, starting from the power series expansions obtained in the previous proposition.

Proposition 4 *Given a straight-line program Γ of size L that computes the coefficients $T_{j,\kappa}$ and $S_{i,j,\kappa}$ for $\kappa \leq 2\delta$, one can perform the following tasks in time $L + O(n\delta^6)$:*

- *determine the basis $\mathbf{m}$;*
- *construct a straight-line program of size $L + O(n\delta^6)$ that evaluates all entries of all matrices $\mathscr{M}_{X_i}$.*

Proof. We first determine the basis $\mathbf{m}$. We use a slight variant of the FGLM algorithm given in [18]: the shape lemma representation $T_{\mathbf{p}}, S_{1,\mathbf{p}}, \ldots, S_{n,\mathbf{p}} \in \mathbb{K}[U]$ of $V(J_{\mathbf{p}})$ is sufficient to recover the Gröbner basis of $J_{\mathbf{p}}$ for the order $>$. By Corollary 2, this gives us the monomial basis $\mathbf{m}$. The cost is $O(n\delta^3)$.

In view of Corollary 2 again, one sees that the matrices $\mathscr{M}_{X_i}$ are also the multiplication matrices in $\mathbb{K}(\mathbf{P})[\mathbf{X}]/J$ in the basis $\mathbf{m}$. Now, knowing $\mathbf{m}$, Lemma 4

shows how to construct in time $O(n\delta^4)$ a straight-line program of size $O(n\delta^4)$ that takes as input the coefficients of $T, S_1, \ldots, S_n$ and outputs a denominator $\mathscr{D}$ and the entries of matrices $\mathscr{M}'_{X_i}$, with $\mathscr{M}_{X_i} = \mathscr{M}'_{X_i}/\mathscr{D}$.

Remark now that the entries of $\mathscr{M}_{X_i}$ are polynomials of degree at most 2δ. Besides, the coefficients of $T, S_1, \ldots, S_n$ are rational functions, that are given through truncated power series expansions. Applying Lemma 3 (to power series expanded at $\mathbf{p}$), we can construct in time $L + O(n\delta^6)$ a new straight-line program of size $L + O(n\delta^6)$, that computes all homogeneous components of all entries of the matrices $\mathscr{M}_{X_i}$ up to degree 2δ. Since the entries of these matrices are polynomials of that degree, it suffices to add their homogeneous components to conclude. □

The proof of Theorem 3 follows by putting together the results of Propositions 1, 3 and 4.

6 Rewriting in the invariant basis

We finally conclude the proof of our main theorem, proceeding in two steps:

- A polynomial F in $\mathbb{K}[\mathbf{X}]$ can be uniquely written as

$$F = \sum_{m \in \mathbf{m}} \varphi_m(\boldsymbol{\pi}) m,$$

  where all φ_m are in $\mathbb{K}[\mathbf{P}]$. Using the multiplication matrices computed before, one can readily obtain the polynomials φ_m from F.

- If F is invariant under $\mathscr{G}$, the vector of its coefficients $(\varphi_m)_{m \in \mathbf{m}}$ is a linear combination of the coefficient vectors giving the secondary invariants; linear system solving will conclude the proof.

This process is of course quite natural; what requires care is the control of its cost. We let $\Gamma_{\boldsymbol{\sigma}}$ be a straight-line program of size $L_{\boldsymbol{\sigma}}$ that computes the secondary invariants $\boldsymbol{\sigma}$ (this is part of our input) and we write, for $\sigma \in \boldsymbol{\sigma}$,

$$\sigma = \sum_{m \in \mathbf{m}} a_{m,\sigma}(\boldsymbol{\pi})\, m, \tag{3}$$

with $a_{m,\sigma} \in \mathbb{K}[\mathbf{P}]$. We let Γ_0 be the straight-line program of Theorem 3 that computes the entries of $\mathscr{M}_{X_1}, \ldots, \mathscr{M}_{X_n}$, and let L_0 be its size. Finally, in all that follows, $F \in \mathbb{K}[\mathbf{X}]$ is given by a straight-line program Γ of size L.

Proposition 5 *With notation as above, given Γ_0, $\Gamma_{\boldsymbol{\sigma}}$ and Γ, one can construct in time $L_0 + O(L\delta^3 + L_{\boldsymbol{\sigma}}\delta^3)$ a straight-line program of size $L_0 + O(L\delta^3 + L_{\boldsymbol{\sigma}}\delta^3)$ that computes all φ_m and all $a_{m,\sigma}$.*

PROOF. The coefficients φ_m form the column indexed by the monomial $1 \in \mathbf{m}$ in the matrix $F(\mathscr{M}_{X_1}, \ldots, \mathscr{M}_{X_n})$. The reasoning is the same for the polynomials $a_{m,\sigma}$, and the result is a direct consequence of Corollary 1. □

Suppose now that F is in $\mathbb{K}[\mathbf{X}]^{\mathscr{G}}$, and let $(F_\sigma)_{\sigma\in\boldsymbol{\sigma}}$ be the unique polynomials in $\mathbb{K}[\mathbf{P}]$ such that $F = \sum_{\sigma\in\boldsymbol{\sigma}} F_\sigma(\boldsymbol{\pi})\sigma$. The following proposition will conclude the proof of our main theorem.

Proposition 6 *With notation as above, given Γ_0, $\Gamma_{\boldsymbol{\sigma}}$ and Γ, one can construct in time $L_0 + O(L\delta^3 + L_{\boldsymbol{\sigma}}\delta^3 + \delta^4)$ a straight-line program of size $L_0 + O(L\delta^3 + L_{\boldsymbol{\sigma}}\delta^3 + \delta^4)$ that computes all F_σ.*

PROOF. For definiteness, let us order $\mathbf{m}$ and $\boldsymbol{\sigma}$ by increasing degree and let $\mathscr{M}$ be the $\delta \times e$ matrix with entries $a_{m,\sigma}$. This matrix represents the map $\mathbb{K}[\mathbf{P}]^e \to \mathbb{K}[\mathbf{P}]^\delta$ given by $(G_\sigma)_{\sigma\in\boldsymbol{\sigma}} \mapsto (\sum_{\sigma\in\boldsymbol{\sigma}} a_{m,\sigma}G_\sigma)_{m\in\mathbf{m}}$. Since F is in $\mathbb{K}[\mathbf{X}]^{\mathscr{G}}$, the coefficient vector $(\varphi_m)_{m\in\mathbf{m}}$ is in the image of $\mathscr{M}$; the coefficients $(F_\sigma)_{\sigma\in\boldsymbol{\sigma}}$ are its (unique, by the following lemma) preimage.

Lemma 8 *One can determine in time $O(\delta e^2)$ an invertible $e \times e$ submatrix $\mathscr{M}'$ of $\mathscr{M}$ with determinant in $\mathbb{K}$.*

PROOF. For σ in $\boldsymbol{\sigma}$ and m in $\mathbf{m}$, the coefficient $a_{m,\sigma}$ is either 0 or weighted homogeneous of degree $\deg(\sigma) - \deg(m)$, with P_i of weight $\deg(\pi_i)$. For $d \geq 0$, we let $\mathbf{m}_d$ be the subset of $\mathbf{m}$ consisting of elements of degree d; similarly $\boldsymbol{\sigma}_d$ is the subset of all elements of degree d in $\boldsymbol{\sigma}$. For σ in $\boldsymbol{\sigma}_d$, we can then rewrite Equation (3) as

$$\sigma = \sum_{m\in\mathbf{m}_{d'},\ d'<d} a_{m,\sigma}(\boldsymbol{\pi})\, m \ + \sum_{m\in\mathbf{m}_d} a_{m,\sigma}\, m, \tag{4}$$

where in the second sum, the coefficients $a_{m,\sigma}$ are in $\mathbb{K}$. Remark that for a fixed $d \geq 0$, the homogeneous polynomials

$$\left\{ \sum_{m\in\mathbf{m}_d} a_{m,\sigma}\, m \mid \sigma \in \boldsymbol{\sigma}_d \right\}$$

are linearly independent. Indeed, reducing Equation (4) modulo $\langle\boldsymbol{\pi}\rangle$, we find the relations, for $\sigma \in \boldsymbol{\sigma}_d$:

$$\sigma = \sum_{m\in\mathbf{m}_d} a_{m,\sigma}\, m \mod \langle\boldsymbol{\pi}\rangle.$$

Since the family σ is linearly independent modulo $\langle\boldsymbol{\pi}\rangle$, the claim follows. Thus, there exists $\mathbf{m}'_d \subset \mathbf{m}_d$ of cardinality $|\boldsymbol{\sigma}_d|$ such that the scalar matrix $\mathscr{M}'_d = [a_{m,\sigma}]$, for $m \in \mathbf{m}'_d$ and $\sigma \in \boldsymbol{\sigma}_d$, is invertible. For a given d, $\mathbf{m}'_d$ can be found by Gaussian elimination in time $O(|\mathbf{m}_d||\boldsymbol{\sigma}_d|^2)$; the total cost is in $O(\delta e^2)$.

Let $\mathbf{m}'$ be the union of all $\mathbf{m}'_d$, and let $\mathscr{M}'$ be the $e \times e$ square submatrix of $\mathscr{M}$ consisting of rows indexed by $\mathbf{m}'$. By construction, this matrix is block lower-triangular; the blocks on the diagonal are precisely the scalar matrices $\mathscr{M}'_d$ introduced previously. Hence, the determinant of $\mathscr{M}'$ is in $\mathbb{K} - \{0\}$. $\square$

We conclude by Lemma 2: since $\mathscr{M}'$ has constant determinant, the overhead over the cost of Proposition 5 induced by computing the inverse of $\mathscr{M}'$ and deducing the coefficients $(F_\sigma)_{\sigma \in \boldsymbol{\sigma}}$ is $O(\delta^4)$. Adjoining to this result the estimate on L_0 given in Theorem 3, this finishes the proof of our main theorem.

7 Experiments

We describe here the potential applications of our approach on a toy example of polynomial system solving. Consider the group $\mathscr{G} = \{\mathfrak{i}, \mathfrak{j}\}$, with

$$\mathfrak{i} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathfrak{j} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Hence, $\mathfrak{j}$ acts on the polynomial ring $\mathbb{K}[X_1, X_2]$ through $(X_1, X_2) \mapsto (-X_1, -X_2)$ and we can take $\boldsymbol{\pi} = (X_1^2, X_2^2)$ and $\boldsymbol{\sigma} = (1, X_1X_2)$.

Steps of the algorithm. Recall that our algorithm has two main steps: a computation depending only on $\boldsymbol{\pi}, \boldsymbol{\sigma}$ (Section 5) followed by the rewriting of an invariant polynomial F (Section 6). In this case, the first computation can be done by hand (and the straight-line representation is not really required, since the result is so simple): the basis $\mathbf{m}$ is $\{1, X_1, X_2, X_1X_2\}$ and the multiplication matrices by X_1 and X_2 are respectively

$$\mathscr{M}_{X_1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ P_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & P_1 & 0 \end{bmatrix}, \quad \mathscr{M}_{X_2} = \begin{bmatrix} 0 & 0 & P_2 & 0 \\ 0 & 0 & 0 & P_2 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Any F in $\mathbb{K}[X_1, X_2]$ can be uniquely written $F = \varphi_1(\boldsymbol{\pi}) + \varphi_{X_1}(\boldsymbol{\pi})X_1 + \varphi_{X_2}(\boldsymbol{\pi})X_2 + \varphi_{X_1X_2}(\boldsymbol{\pi})X_1X_2$. Then, the coordinates $[\varphi_1, \varphi_{X_1}, \varphi_{X_2}, \varphi_{X_1X_2}]^t$ are given by the first column of the matrix $F(\mathscr{M}_1, \mathscr{M}_2)$.

If F is given through a sequence of additions and multiplications of length L, we can construct a sequence of kL additions and multiplications that computes the entries of $F(\mathscr{M}_{X_1}, \mathscr{M}_{X_2})$, and thus $\varphi_1, \varphi_{X_1}, \varphi_{X_2}, \varphi_{X_1X_2}$; here, k is a constant such that 4×4 matrices can be added or multiplied in k operations (actually, in this simple case, the constant k can even be reduced by exploiting the structure of the matrices).

Finally, the secondary invariants are respectively the first and last elements of the monomial basis $1, X_1, X_2, X_1X_2$. Then, if F is in $\mathbb{K}[X_1, X_2]^{\mathscr{G}}$, $\varphi_{X_1} = \varphi_{X_2} = 0$, and φ_1 and $\varphi_{X_1X_2}$ are the output we are looking for.

Solving symmetric systems. As said in the introduction, evaluation properties partially control the cost of polynomial system solving, for the algorithms of the geometric resolution family [12,11,10,14,13,19]. We conclude with a preliminary study of the applications of these techniques to systems with symmetries. For varying d, we consider as an example the invariant system

$$F_1 = (x_1 + x_2 - 1)^4 + (x_1 + x_2 + 1)^4 + 2, \quad F_2 = (x_1 + x_2 + x_2^3)^{2d} + 1.$$

This is a very favorable situation for us: these polynomials can be evaluated fast, in $L = O(\log(d))$ operations. Remark that such situations are *not* entirely artificial: some symmetric systems (for the symmetric group $\mathfrak{S}_2$) with a similar low complexity of evaluation arose in hyperelliptic point-counting problems [6].

The system (F_1, F_2) has $12d$ solutions. To exploit the symmetries, we follow [5,2]: we rewrite (F_1, F_2) in the variables P_1, P_2, S, obtaining equations (F_1', F_2') in $\mathbb{K}[P_1, P_2, S]$, and adjoin the relation $F_3' = S^2 - P_1P_2$. The system (F_1', F_2', F_3') has $6d$ solutions: the change of variables makes for a better output, but it remains to examine the impact on the computation time.

We compare two approaches. As said above, we focus on the geometric resolution algorithm, whose running time depends linearly on the number of operations it takes to evaluate the system. The rewriting process we described in the previous sections gives an evaluation scheme for (F_1', F_2', F_3') using $O(\log(d))$ operations. On the contrary, a plain rewriting process will expand the equations: the system loses its structure, and it takes $O(d^2)$ operations to evaluate (F_1', F_2', F_3') in their expanded form. We stress the fact that the only difference between these approaches is the way the system (F_1', F_2', F_3') is represented.

In the following table, we give the timings in seconds for solving the system (F_1, F_2) in its original variables, and for the system in the new variables (F_1', F_2', F_3') using the two approaches above; our code is based on Lecerf's `Kronecker` package [20]. All results are obtained using Magma 2.14-8 on a 2.80 Ghz Pentium 4 (for completeness, we mention that for $d \geq 32$, the timings in the first two columns outperform Gröbner basis computation).

d	Original system	Our approach	Plain rewriting
8	2.1	3.0	4.9
16	6.3	8.4	53.2
32	27.4	36.8	874
64	137	191	19867

The results are promising: our approach allows for much better computation time than the naive rewriting strategy; the timings are very close to those for the initial system. However, it is clear that this is still a very first experiment on a very favorable example. More work is required to estimate precisely the running time for solving invariant systems following this strategy.

References

[1] S. J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Inform. Process. Lett.*, 18(3):147–150, 1984.

[2] A. Colin. Solving a system of algebraic equations with symmetries. *Journal of Pure and Applied Algebra*, 117/118:195–215, 1997.

[3] H. Derksen and G. Kemper. *Computational Invariant Theory*, volume 130 of *Encyclopaedia of Mathematical Sciences*. Springer Verlag, 2002.

[4] J.-C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *J. Symbolic Comput.*, 16(4):329–344, 1993.

[5] K. Gatermann. Semi-invariants, equivariants and algorithms. *Appl. Algebra Engrg. Comm. Comput.*, 7(2):105–124, 1996.

[6] P. Gaudry and É. Schost. Construction of secure random curves of genus 2 over prime fields. In *Eurocrypt'04*, pages 239–256. Springer, 2004.

[7] P. Gaudry, É. Schost, and N. Thiéry. Evaluation properties of symmetric polynomials. *Internat. J. Algebra Comput.*, 16(3):505–523, 2006.

[8] P. M. Gianni and T. Mora. Algebraic solution of systems of polynomial equations using groebner bases. In *AAECC'5*, volume 356 of *Lecture Notes in Computer Science*, pages 247–257. Springer, 1987.

[9] M. Giusti and A. Colin. Evaluation techniques as an efficient tool for geometric invariant theory. in preparation.

[10] M. Giusti, K. Hägele, J. Heintz, J.-E. Morais, J.-L. Montaña, and L.-M. Pardo. Lower bounds for diophantine approximation. *J. Pure and Applied Algebra*, 117–118:277–317, 1997.

[11] M. Giusti, J. Heintz, J.-E. Morais, J. Morgenstern, and L.-M. Pardo. Straight-line programs in geometric elimination theory. *J. Pure and Applied Algebra*, 124(1–3):101–146, 1998.

[12] M. Giusti, J. Heintz, J.-E. Morais, and L.-M. Pardo. When polynomial equation systems can be solved fast? In *AAECC'11*, volume 948 of *Lecture Notes in Computer Science*, pages 205–231. Springer, 1995.

[13] M. Giusti, G. Lecerf, and B. Salvy. A Gröbner free alternative for polynomial system solving. *J. Complexity*, 17(2):154–211, 2001.

[14] J. Heintz, T. Krick, S. Puddu, J. Sabia, and A. Waissbein. Deformation techniques for efficient polynomial equation solving. *J. Complexity*, 16(1):70–109, 2000.

[15] M. Kalkbrener. On the stability of Gröbner bases under specializations. *Journal of Symbolic Computation*, 24(1):51–58, 1997.

[16] E. Kaltofen. Greatest common divisors of polynomials given by straight-line programs. *J. ACM*, 35(1):231–264, 1988.

[17] E. Kaltofen. Factorization of polynomials given by straight-line programs. In *Randomness and Computation*, volume 5 of *Advances in Computing Research*, pages 375–412. JAI Press, 1989.

[18] Y. Lakshman. On the complexity of computing a Gröbner basis for the radical of a zero dimensional ideal. In *STOC*, pages 555–563. ACM, 1990.

[19] G. Lecerf. Computing the equidimensional decomposition of an algebraic closed set by means of lifting fibers. *J. Complexity*, 19(4):564–596, 2003.

[20] G. Lecerf. Kronecker v.0.166-9. `http://www.math.uvsq.fr/~lecerf/`, 2008.

[21] V. Strassen. Vermeidung von Divisionen. *J. Reine Angew. Math.*, 264:184–202, 1973.

[22] B. Sturmfels. *Algorithms in invariant theory.* Texts and Monographs in Symbolic Computation. Springer-Verlag, 1993.

[23] B. Sturmfels and N. White. Computing combinatorial decompositions of rings. *Combinatorica*, 11(3):275–293, 1991.