Newton iteration for lexicographic Gröbner bases in two variables

Éric Schost[a], Catherine St-Pierre[a]

[a]*School of Computer science, University of Waterloo, 200 University Ave W, Waterloo, N2L 3G1, Ontario, Canada*

Abstract

We present an $\mathfrak{m}$-adic Newton iteration with quadratic convergence for lexicographic Gröbner basis of zero dimensional ideals in two variables. We rely on a structural result about the syzygies in such a basis due to Conca and Valla, that allowed them to explicitly describe these Gröbner bases by affine parameters; our Newton iteration works directly with these parameters.

Keywords: Primary components, $\mathfrak{m}$-adic algorithm, Gröbner bases

1. Introduction

Solving bivariate polynomial equations plays an important role in algorithms for computational topology or computer graphics. As a result, there exists a large body of work dedicated to this question, using symbolic, numeric or mixed symbolic-numeric techniques. To wit, the literature included algorithms based on subresultant techniques [23, 13, 18], subdivision [1], numerical or p-adic Newton iteration [55, 38, 47], root isolation using two projections [3, 17, 35, 36], as well as refined root separation bounds [12] or the relations between bivariate Gröbner bases and the subresultant algorithm [10].

In many instances, these algorithms find a set-theoretic description of the solutions of a given system $f_1, \ldots, f_t$ in $\mathbb{K}[x, y]$ (here, $\mathbb{K}$ is a field). This can notably be done through the *shape lemma*: in generic coordinates, the output is a pair of polynomials u, v in $\mathbb{K}[x]$, with u squarefree, such that $V(\langle f_1, \ldots, f_t\rangle)$ is described by $u(x) = 0$ and $y = v(x)/u'(x)$ (this rational form for y allows for a sharp control of the bit size of v, if $\mathbb{K} = \mathbb{Q}$). One could slightly enrich this set-theoretic description by lifting the requirement that u be squarefree, and instead assign to a root ξ of u, corresponding to a point (ξ, ν), the multiplicity of $J = \langle f_1, \ldots, f_t\rangle$ at (ξ, ν) (adapting the definition of v accordingly). This is notably done in Rouillier's Rational Univariate Parametrization [54], but this still only gives partial information: for instance, it is not sufficient to detect local isomorphisms.

In order to describe the solutions of J, but also the local structure of J at these roots (that is, the localizations of the algebra $\mathbb{K}[x, y]/J$ at these points), it is natural to turn to Gröbner bases. This is what we address in this paper, our focus being an $\mathfrak{m}$-adic approximation procedure, in a sense we define below.

Our problem and our main result. Let us assume that our base field $\mathbb{K}$ is the field of fractions of a domain $\mathbb{A}$, and take $f_1, \ldots, f_t$ in $\mathbb{A}[x, y]$.

Consider further the ideal $J = \langle f_1, \ldots, f_t \rangle$ in $\mathbb{K}[x, y]$. We are interested in finding a Gröbner basis of J itself, or possibly of some specific primary components of it. We will thus let I be an ideal in $\mathbb{K}[x, y]$, which we assume to be the intersection of some of the zero-dimensional primary components of J: typical cases of interest are $I = J$, if it has dimension zero, or I being the $\langle x, y \rangle$-primary component of J, if the origin is isolated in $V(J)$. For general primary components, we recall that arbitrary isolated components may be translated to the origin via a change of basis, at the expense possibly of a change of base field [30].

We let $\mathcal{G} = (g_0, \ldots, g_s)$ be the minimal, reduced Gröbner basis of I for the lexicographic order induced by $y \succ x$; this is the object we are interested in.

Example 1.1. *Let $\mathbb{A} = \mathbb{Z}$, and thus $\mathbb{K} = \mathbb{Q}$, $t = 2$ and input polynomials*

$$f_1 = -12xy^5 - 20x^2y^4 - 14y^4 - 7x^3y^3 - 3x^2y^2 + 13x^3y - 17xy + 34x^2$$
$$f_2 = -x^2y^4 - 19x^3y^3 + 18xy^3 + 22x^3y^2 + 2x^2y^2 - 10x^2y.$$

We let I be the $\langle x, y \rangle$-primary component of $\langle f_1, f_2 \rangle$; its Gröbner basis $\mathcal{G}$ is

$$\left| \begin{array}{l} y^4 + \frac{17}{14}xy - \frac{17}{7}x^2, \\ xy^3 - \frac{10}{9}x^3, \\ x^2y - 2x^3, \\ x^4. \end{array} \right. \tag{1}$$

Let now $\mathfrak{m}$ be a maximal ideal in $\mathbb{A}$, with residual field $\Bbbk = \mathbb{A}/\mathfrak{m}$. Starting from the reduction of $\mathcal{G}$ modulo $\mathfrak{m}$ (assuming it is well-defined), the goal of this paper is to show how to recover $\mathcal{G}$ modulo powers of $\mathfrak{m}$. The case $\mathbb{A} = \mathbb{Z}$ seen above is the fundamental kind of example; another important situation is the "parametric" case, with $\mathbb{A} = \Bbbk[t_1, \ldots, t_m]$ and $\mathfrak{m}$ a maximal ideal of the form $\langle t_1 - \tau_1, \ldots, t_m - \tau_m \rangle$.

Let $\mathbb{A}_\mathfrak{m}$ ($\mathbb{A}_\mathfrak{m} \subseteq \mathbb{K}$) be the localization of $\mathbb{A}$ at $\mathfrak{m}$. For $K \geq 0$, there exists a well defined reduction operator $\mathbb{A}_\mathfrak{m} \to \mathbb{A}/\mathfrak{m}^K$, which we write $c \mapsto c \text{ rem } \mathfrak{m}^K$; we extend it coefficient-wise to a reduction mapping $\mathbb{A}_\mathfrak{m}[x, y] \to \mathbb{A}/\mathfrak{m}^K[x, y]$, and further to vectors of polynomials.

Definition 1.2. *We say that $\mathfrak{m}$ is* **good** *with respect to $f_1, \ldots, f_t$ and $\mathcal{G}$ if the following holds:*

- *all elements in $\mathcal{G}$ are in $\mathbb{A}_\mathfrak{m}[x, y]$,*
- *the ideal generated by the reduction $(\mathcal{G} \text{ rem } \mathfrak{m})$ of $\mathcal{G}$ modulo $\mathfrak{m}$ in $\Bbbk[x, y]$ is the intersection of some of the primary components of the ideal $\langle f_1 \text{ rem } \mathfrak{m}, \ldots, f_t \text{ rem } \mathfrak{m} \rangle$.*

In particular, if $\mathfrak{m}$ is good, we will write $\mathcal{G}_\mathfrak{m}$ for the reduction $\mathcal{G}$ rem $\mathfrak{m}$. These are polynomials in $\Bbbk[x, y]$, and they still form a minimal, reduced Gröbner basis for the lexicographic order $y \succ x$.

Example 1.3. *In Example 1.1,* $\mathfrak{m} = \langle 11 \rangle$ *is good with respect to* $f_1, \ldots, f_t$ *and* $\mathcal{G}_{\mathfrak{m}}$ *is*

$$\left| \begin{array}{l} y^4 + 2xy + 7x^2, \\ xy^3 + 5x^3, \\ x^2y + 9x^3, \\ x^4. \end{array} \right.$$

If $\mathbb{A} = \mathbb{Z}$, there are finitely many primes p for which this is not the case. In the case $\mathbb{A} = \Bbbk[t_1, \ldots, t_m]$, all maximal ideals of the form $\langle t_1 - \tau_1, \ldots, t_m - \tau_m \rangle$ are good, except for those $(\tau_1, \ldots, \tau_m)$ lying on a certain hypersurface in $\Bbbk^m$ (a quantitative analysis of the number of bad maximal ideals will be the subject of future work).

Our main result is an efficient lifting procedure based on Newton iteration to compute $\mathcal{G}$ rem $\mathfrak{m}^K$, given $f_1, \ldots, f_t$, $\mathcal{G}_{\mathfrak{m}}$ and K. Lifting methods are widely used in computer algebra, for instance, to solve linear systems or compute polynomial GCDs, and serve two purposes. First, while solving the problem (here, computing the Gröbner basis of I) may be nontrivial from the outset, working directly over $\mathbb{K}$, our result will show that lifting an approximate solution modulo powers of $\mathfrak{m}$ is a relatively simple problem. Second, these techniques are usually used in cases where elements in $\mathbb{A}$, and $\mathbb{K}$, have a natural notion of "size" (such as the height when $\mathbb{A} = \mathbb{Z}$, or degree when $\mathbb{A} = \Bbbk[t_1, \ldots, t_m]$). Then, direct computations in $\mathbb{K}$ often induce a significant "intermediate expression swell", where polynomials computed throughout the algorithm may have larger coefficients than the final output; $\mathfrak{m}$-adic approximation schemes avoid this issue.

Previous forms of Newton iteration have been proposed in the context of Gröbner basis computation, but all have limitations (they may work with solutions of multiplicity one only, or compute a Gröbner basis of the radical of the input ideal, or display linear convergence only); we discuss them below. Our algorithm applies in the bivariate case only, but it features the quadratic convergence typical of Newton iteration, in the sense that it computes $\mathcal{G}$ rem $\mathfrak{m}^2, \mathcal{G}$ rem $\mathfrak{m}^4, \ldots$ (hence, without loss of generality, we assume that $K = 2^\kappa$ is a power of two). The cost of the algorithm is expressed in terms of two kinds of quantities:

- number of operations in the rings $\mathbb{A}/\mathfrak{m}^{2^i}$ (for which we discuss our computational model in more detail at the end of the introduction)
- the cost of reducing the coefficients of the polynomials f_i modulo $\mathfrak{m}^{2^i}$: we will assume that for $i \geq 0$, each such coefficient can be reduced modulo $\mathfrak{m}^{2^i}$ in time T_{2^i} (for $\mathbb{A} = \mathbb{Z}$, this time would depend on the bit-size of these coefficients; over $\mathbb{A} = \Bbbk[t_1, \ldots, t_m]$, it would depend on their degree, and the number m of parameters).

Throughout, the $O^{\tilde{}}$ notation indicates that we omit polylogarithmic factors, and ω is a feasible exponent for linear algebra.

Theorem 1.4. *Let* $f_1, \ldots, f_t$ *be of degree at most* d *in* $\mathbb{A}[x, y]$*, with* $\mathbb{A}$ *a domain, that generate an ideal* J *in* $\mathbb{K}[x, y]$*, with* $\mathbb{K}$ *the fraction field of* $\mathbb{A}$*. Let* I *be the intersection of some of the zero-dimensional primary components of* J*, with minimal, reduced Gröbner basis* $\mathcal{G}$*, for the lexicographic order induced by* $y \succ x$*.*

Let further $\boldsymbol{E} = (y^{n_0}, x^{m_1}y^{n_1}, \ldots, x^{m_{s-1}}y^{n_{s-1}}, x^{m_s})$ *be the initial terms of* $\mathcal{G}$*, and let* $\delta = \dim_{\mathbb{K}} \mathbb{K}[x,y]/I$*.*

Let $\mathfrak{m} \subseteq \mathbb{A}$ *be a good maximal ideal for* $\mathcal{G}$*. For* K *of the form* $K = 2^k$*, given* $\mathcal{G}$ rem $\mathfrak{m}$*, one can find* $\mathcal{G}$ rem $\mathfrak{m}^K$ *with the following cost:*

- $O^{\sim}(s^2\delta n_0 m_s + t\delta(d^2 + dm_s + s\delta + \delta^{\omega-1}))$ *operations in* $\mathbb{A}/\mathfrak{m}^{2^i}$*, for* $i = 1, \ldots, k$*;*
- $td^2T_{2^i}$ *steps for coefficient reduction, for* $i = 1, \ldots, k$*.*

From a high level perspective, Newton iteration is based on the solution of linearizations of a certain set of non-linear equations; in our case, these equations involve $\Theta(\delta)$ unknowns. The runtime given above is rather complex, but we can give a rough interpretation of its components: the first term $s^2\delta n_0 m_s$ describes the cost of setting up a "parametric Gröbner basis", that depends on our $\Theta(\delta)$ unknowns. The term $t\delta(d^2 + dm_s + s\delta)$ gives the runtime taken by reducing the input equations modulo this parametric Gröbner basis. We do all computations at order one in our unknowns, which amounts to the linearization inherent to Newton iteration. It remains to find what values of these unknowns cancel the degree-one component; this results in the term $t\delta^\omega$, from linear system solving.

Remark 1.5. *When* I *is the* $\langle x, y\rangle$*-primary component of* J*, runtimes can be sharpened, giving*

- $O^{\sim}(s^2\delta n_0 m_s + t\delta^2(m_s + \delta^{\omega-2}))$ *operations in* $\mathbb{A}/\mathfrak{m}^{2^i}$*, for* $i = 1, \ldots, k$*;*
- $t\delta m_s T_{2^i}$ *steps for coefficient reduction, for* $i = 1, \ldots, k$*.*

Since $m_s \leq \delta$*, these are in particular* $O^{\sim}(s^2\delta n_0 m_s + t\delta^3) \subset O^{\sim}((s^2 + t)\delta^3)$*, resp.* $t\delta^2 T_{2^i}$*. For the latter, we also have the bound* $td^2T_{2^i}$ *stated in the theorem, but here we prefer to express the cost in terms of the multiplicity* δ *only.*

This paper focuses on those cases where the ideal I is not radical (that is, where some points $p \in V(I)$ are singular), with the intent of describing the localizations of $\mathbb{K}[x,y]/I$ at such points. If the sole interest is to find $V(I)$, then our approach is unnecessarily complex: the algorithms in [38, 47] use Newton iteration to compute a set-theoretic description of the solutions in an efficient manner.

Example 1.6. *An extreme case has* $t = 2$ *and* f_1, f_2 *"generic" in the sense that they define a radical ideal in* $\mathbb{K}[x,y]$ *with* d^2 *solutions in general position. In this case, if we take* $I = J$*, we have* $s = 1$*,* $m_s = \delta = d^2$ *and* $n_0 = 1$*. Then, the complexity in the first item of the theorem becomes* $O^{\sim}(d^5)$ *operations modulo each* $\mathfrak{m}^{2^i}$*. This is to be compared with the sub-cubic cost* $O^{\sim}(d^{(\omega+3)/2})$ *reported in [38] for a similar task.*

Clearly, for these generic situations, our algorithm does not compare favourably with the state of the art. For the situation in Example 1.6, some techniques from [38] could be put to use in our situation as well, but they would at best give a runtime of $O^{\sim}(d^{2+(\omega+3)/2})$ operations in $\mathbb{A}/\mathfrak{m}^{2^i}$, still leaving a quadratic overhead. This is due to the different ways

these papers apply Newton iteration: in our case, we linearize the problem in dimension d^2 (or, in, general, δ), and thus work with matrices of such size, whereas [38] work with matrices of size 2 (albeit with polynomial entries).

The results of Theorem 1.4 are of interest in the presence of intersection with multiplicities, where approaches such as [38] do not apply. The algorithm in [47] does not solve our problem in such cases, as it does not compute a Gröbner basis of I, but of its radical.

Remark that to derive a complete algorithm from our result, further ingredients are needed: quantitative bounds on the number of bad ideals $\mathfrak{m}$ (if $\mathbb{A} = \mathbb{Z}$ or $\mathbb{A} = \Bbbk[t_1, \ldots, t_m]$, for instance), a cost analysis for computing the starting point $\mathcal{G}_\mathfrak{m}$ and bounds on a sufficient precision K that will allow us to recover $\mathcal{G}$ from its approximation $\mathcal{G}$ rem $\mathfrak{m}^K$. To avoid this paper growing to an excessive length, we will address these questions in a separate manuscript.

We now review previous work on bivariate systems and Newton iteration for Gröbner bases. As we will see, there is a marked difference between Newton iteration algorithms for "simple" solutions (where the Jacobian of the input equations has full rank) in generic position and those that can handle arbitrary situations.

Newton iteration for non-degenerate solutions. Following an early discussion in [14], p-adic techniques for Gröbner bases were introduced by Trinks in the 1980's [58]. That article focuses on zero-dimensional *radical* ideals with generators in $\mathbb{Z}[x_1, \ldots, x_n]$, in shape lemma position, that is, with a Gröbner basis of the form $x_1 - G_1(x_n), \ldots, x_{n-1} - G_{n-1}(x_n), G_n(x_n)$, for the lexicographic order $x_1 \succ \cdots \succ x_n$. Under this assumption, given a "lucky" prime p, one can apply a symbolic form of Newton iteration to lift $(G_1, \ldots, G_n)$ rem p to $(G_1, \ldots, G_n)$ rem p^K, for an arbitrary $K \geq 0$. Similar techniques were used in the geometric resolution algorithm of [19, 21, 20, 22]; the scope of this symbolic form of Newton iteration was then extended in [56] to *triangular sets*, which are here understood as those particular lexicographic Gröbner bases $(G_1, \ldots, G_n)$ with respective initial terms of the form $x_1^{e_1}, \ldots, x_n^{e_n}$, for some positive integers $e_1, \ldots, e_n$. In [38], these techniques were studied in detail for the case $n = 2$ that concerns us in this paper, with a focus on the complexity of the lifting process.

Computationally, these algorithms have the advantage of working with simple data structures: they mainly perform matrix multiplications in size n with entries that are polynomials with coefficients in $\mathbb{Z}/p^K\mathbb{Z}$ (or more generally $\mathbb{A}/\mathfrak{m}^K$). These methods also share their numerical counterpart's quadratic convergence (in one iteration, the precision doubles, from p^K to p^{2K}), but none of them can directly handle solutions with multiplicities.

Lifting algorithms for general inputs. [62] introduced an algorithm that handles arbitrary inputs: given a Gröbner basis $\mathcal{G}$ for $f_1, \ldots, f_t$ reduced modulo a "lucky" prime p, it recovers the Gröbner basis of the same system modulo p^K, for any $K \geq 0$. No assumption is made on the dimension of $V(\langle f_1, \ldots, f_t \rangle)$ or the rank of the Jacobian matrix of the equations. The computations are more complex as the ones above, as they involve lifting not only the Gröbner basis $\mathcal{G}$ itself, but also all quotients in the division of $f_1, \ldots, f_t$, and of the S-polynomials of $\mathcal{G}$, by $\mathcal{G}$.

In follow-up work, [51] discussed the choice of lucky primes; for homogeneous inputs,

or graded orderings, [2] gave an efficient criterion to stop lifting and simplified the lifting algorithm itself, using ideas of Pauer's (the S-polynomials are not needed anymore).

To our knowledge, the algorithms mentioned here only perform *linear* lifting, going from an approximation modulo p^K to precision p^{K+1}; whether quadratic convergence is possible is unclear to us. No cost analysis was made.

Deflation. Ojika, Watanabe and Mitsui introduced the idea of deflation in a numerical context [50], to restore Newton iteration's quadratic convergence even for multiple roots. The core idea is to replace the system we are given by another set of equations, having multiplicity one at the root we are interested in, possibly introducing new variables. There are now many references discussing this approach, see for instance [64, 39, 42, 43, 52, 11, 45, 63].

We are in particular going to use an idea from [25]. In that reference, Hauenstein, Mourrain and Szanto designed a deflation operator for an n-variate system $f_1, \ldots, f_t$, that converges quadratically to an augmented root (ξ, ν), where ν is a vector that specifies the local structure at a point $\xi \in V(\langle f_1, \ldots, f_t \rangle)$, through the coefficients of multiplication matrices in the local algebra at ξ. If ξ is known, this gives in particular an operator with quadratic convergence to compute the structure constants (that is, the entries of the multiplication matrices).

Our contribution. The lifting algorithm we propose is so far specific to lexicographic orders in two variables, but has the advantage of being simpler than those in [62, 2]. Indeed, compared to these two references, we do not need to p-adically lift the polynomial quotients in the division of $f_1, \ldots, f_t$ by $\mathcal{G}$; instead, we work with a family of free parameters that describe bivariate Gröbner bases with given initial terms in a one-to-one manner (these Gröbner bases form a *Gröbner cell*). In particular, the number of parameters we work with is tight: this number is precisely the dimension of the Gröbner cell, whereas the polynomial quotients involve a (necessarily) larger number of coefficients, that depends not only on $\mathcal{G}$ but also on the input equations we are given.

However, identifying a suitable family of parameters is not straightforward. The coefficients that appear in the Gröbner basis do not form such a family, as there are nontrivial relations between them. However, for lexicographic orders in two variables, Conca and Valla explicitly constructed a one-to-one parametrization of a given Gröbner cell by an affine space [9], from a description of canonical generators of the syzygy module. Our Newton iteration computes the parameters corresponding to $\mathcal{G}_{\mathfrak{m}}$ and lifts them modulo $\mathfrak{m}^K$.

This is done by adapting the approach of [25]: the coefficients of the normal forms of $f_1, \ldots, f_t$ modulo the unknown Gröbner basis $\mathcal{G}$ are polynomials in the parameters of the Gröbner cell; we prove that they admit as a (not necessarily unique) solution the parameters corresponding to $\mathcal{G}$, and that their Jacobian matrix has full rank at this solution. We can then apply Newton iteration to these polynomials. We need in particular a starting point for the iteration, that is, the reduction modulo $\mathfrak{m}$ of the Conca-Valla parameters corresponding to $\mathcal{G}$: we derive it from the knowledge of $\mathcal{G}_{\mathfrak{m}}$, using ad-hoc conversion formulas.

Computationally, the core operation involved in our Newton iteration is simply reduction modulo a lexicographic Gröbner basis: as was mentioned after Theorem 1.4, reducing the input polynomials modulo a certain parametric Gröbner basis gives us the linear equations we

need to solve at each lifting step. While we have algorithms with quasi-linear cost for reduction modulo a single polynomial (this is fast Euclidean division [61, Chapter 9]), or modulo two polynomials with respective initial terms y^n and x^m (by a direct bivariate extension, see for instance [44] for the case of an arbitrary number of variables), we are not aware of specific results for arbitrary lexicographic bases. Another contribution of this paper is a reduction algorithm, where we use techniques developed by van der Hoeven and Larrieu [59] for certain weighted orderings, adapted to our purposes.

Leitfaden. In Section 2, we discuss *initial segments* in $\mathbb{N}^2$; they allow us to describe polynomials reduced modulo a Gröbner basis. We give in particular an algorithm for multiplying two such polynomials, which is used in the lexicographic Gröbner basis reduction algorithm.

In Section 3, we review known results on the structure of bivariate lexicographic Gröbner bases: Lazard's theorem [37], and Conca and Valla's description of Gröbner cells: this introduces the parameters of our Newton iteration, namely the coefficients of a canonical family of syzygies between the elements of our Gröbner basis.

Section 4 then presents our algorithm for reduction modulo a lexicographic Gröbner basis. In Section 5 and Section 6, we give algorithms to compute the Gröbner basis corresponding to a set of parameters in the Gröbner cell, and conversely; they derive directly from the definition of the Conca-Valla parameters.

Finally, we describe Newton iteration for the Gröbner cell parameters in Section 7, proving Theorem 1.4. This is based on the description of a family of polynomial equations for which the Conca-Valla parameters form a solution of multiplicity one; the core of this section explains how to apply Newton iteration to these equations in an efficient manner.

Computational model. In the whole paper, the costs of algorithms are measured using numbers of operations in the base ring or base field.

We will first and foremost count $\mathbb{Z}$*-algebra operations*. For an algorithm with inputs and outputs in a (unital) ring $\mathbb{A}$, these are additions and multiplications involving the inputs, previously computed quantities, and constants taken from the image of the canonical mapping $\mathbb{Z} \to \mathbb{A}$ (e.g., integers if $\mathbb{A}$ has characteristic zero); they will be simply be called "$(+, \times)$ operations". If an algorithm performs only this kind of operations, its outputs are in the subring of $\mathbb{A}$ generated by its inputs.

Important examples are addition, multiplication and Euclidean division (by a monic divisor) in $\mathbb{A}[x]$; they can all be done using a softly linear number of $(+, \times)$ operations in $\mathbb{A}$, over any base ring $\mathbb{A}$. For background, see Chapters 8 and 9 in [61].

Other operations we will occasionally use are invertibility tests and inversions (to solve linear systems). Finally, if $\mathfrak{m}$ is an ideal in a ring $\mathbb{A}$, given a in $\mathbb{A}/\mathfrak{m}$, we assume that we can find A in $\mathbb{A}$ with $A \text{ rem } \mathfrak{m} = a$ using one operation in $\mathbb{A}$.

Notation. The following notation is used throughout the paper. In the following items, $\mathbb{A}$ is an arbitrary ring.

- For $d \geq 1$, We let $\mathbb{A}[x]_{<d}$ be the free $\mathbb{A}$-module of all polynomials in $\mathbb{A}[x]$ of degree less than d.

- For f, g in $\mathbb{A}[x]$, with f monic, we define f rem g and f div g as respectively the remainder and quotient in the Euclidean division of f by g.
- For f in $\mathbb{A}[x, y]$, $\deg(f, x)$ and $\deg(f, y)$ respectively denote its partial degrees with respect to x and y.
- For f in $\mathbb{A}[x, y]$ and $i \geq 0$, the *polynomial coefficient* of y^i in f will refer to the coefficient f_i in the expression $f = \sum_{i=0}^{d} f_i y^i$, with $f_0, \ldots, f_d$ in $\mathbb{A}[x]$. In the pseudo-code, we write PolynomialCoefficient$(f, y^i) \in \mathbb{A}[x]$ for this polynomial coefficient.
- If $f \in \mathbb{A}[x, y]$ has degree d in y, we say that f is *monic in* y if the polynomial coefficient of y^d is 1 (this definition and the previous one carry over to coefficients with respect to x instead, but we will not need this).
- If T is a subset of $\mathbb{N}^2$, we write $\mathbb{A}[x, y]_{\mathsf{T}}$ for the $\mathbb{A}$-module of polynomials *supported on* T, that is, all polynomials of the form $\sum_{(u,v)\in\mathsf{T}} a_{u,v} x^u y^v$, with only finitely many non-zero coefficients $a_{u,v}$.

We will not need to define Gröbner bases over rings. In particular, for the reduction of bivariate polynomials, we only work over fields: if $\mathcal{G}$ is a Gröbner basis in $\mathbb{K}[x, y]$, where $\mathbb{K}$ is a field and $\mathbb{K}[x, y]$ is endowed with a monomial order, f rem $\mathcal{G}$ denotes the remainder of f through reduction by $\mathcal{G}$.

2. Initial segments in $\mathbb{N}^2$

In this section, we first introduce terminology and basic constructions regarding subsets of $\mathbb{N}^2$ called initial segments. In the second part, we give algorithms to multiply polynomials supported on such initial segments.

2.1. Basic definitions

Initial segments. We say that a set $\mathsf{T} \subset \mathbb{N}^2$ is an *initial segment* if for all (m, n) in T, any pair (m', n') with $m' \leq m$ and $n' \leq n$ is also in T.

Suppose that T is an initial segment in $\mathbb{N}^2$, let $\mathbb{K}$ be a field and x, y be variables over $\mathbb{K}$. The elements in $\mathbb{K}[x, y]$ supported on $\mathbb{N}^2 - \mathsf{T}$ form a monomial ideal $I \subset \mathbb{K}[x, y]$. Conversely, any initial segment T in $\mathbb{N}^2$ can be obtained in this manner from a monomial ideal I, as the set of exponents of monomials not in I. If T is finite, we write the minimal monomial generators of I as

$$\boldsymbol{E} = (y^{n_0}, x^{m_1} y^{n_1}, \ldots, x^{m_{s-1}} y^{n_{s-1}}, x^{m_s})$$

with the m_i's increasing and the n_i's decreasing, and we set $m_0 = n_s = 0$. We call n_0 the *height* of T and m_s its *width*. We say that T is *determined* by I, or equivalently by $\boldsymbol{E}$.

For $i = 1, \ldots, s$, we set $d_i = m_i - m_{i-1}$, so that $m_i = d_1 + \cdots + d_i$. Then, the cardinal δ of T can be written as $\sum_{i=1}^{s} d_i n_{i-1}$; δ is also called the *degree* of $\boldsymbol{E}$. Similarly, for $i = 1, \ldots, s$, we write $e_i = n_{i-1} - n_i$. These definitions are illustrated in Figure 1, where the monomials in $\boldsymbol{E}$ are the initial terms of the Gröbner basis in Eq. (1).

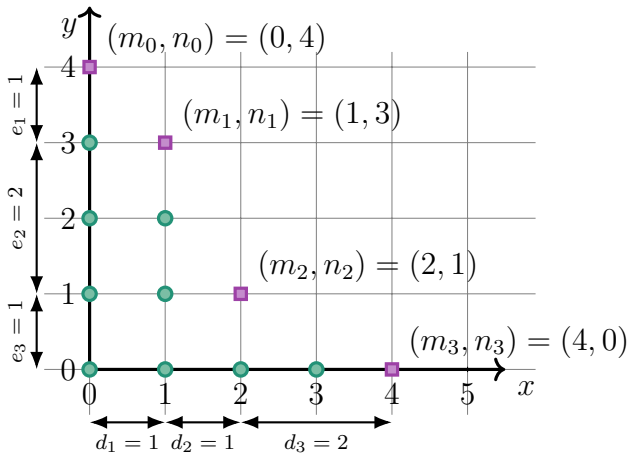

Figure 1: An initial segment T (green) and the monomials $\boldsymbol{E} = (y^4, xy^3, x^2y, x^4)$ (purple), with $s = 3$ and $\delta = 9$.

The cost analyses in this paper will be done using in particular the parameters s and δ. If desired, one can simplify such expressions using the following explicit upper bound for s.

Lemma 2.1. *The integer s is in $O(\sqrt{\delta})$, and this bound is sharp in some instances.*

Proof. Start from the equality $\delta = \sum_{i=1}^{s} d_i n_{i-1}$, which implies $\delta \geq \sum_{i=1}^{s} n_{i-1}$. Since $n_s = 0$ and $n_{i-1} > n_i$, we get by induction $n_i \geq s - i$ for all i. This implies $\delta \geq s(s-1)/2$, so that s is in $O(\sqrt{\delta})$. For the lower bound, for any integer d we can take $\boldsymbol{E} = (x^i y^{d-i},\ i = 0, \ldots, d)$, for which $s = d$ and $\delta = d(d+1)/2$. □

Translates of an initial segment. We will occasionally make use of the following construction. Let T be a finite initial segment in $\mathbb{N}^2$, and suppose that T is determined by a monomial ideal I, with minimal monomial generators $\boldsymbol{E}$ as above. For $i = 0, \ldots, s$ we let $\mathsf{T}_{\leftarrow i}$ be the initial segment determined by the colon ideal $I : x^{m_i}$, with minimal monomial generators

$$\boldsymbol{E}_{\leftarrow i} = (y^{n_i}, x^{m_{i+1}-m_i} y^{n_{i+1}}, \ldots, x^{m_{s-1}-m_i} y^{n_{s-1}}, x^{m_s - m_i}).$$

The set $\mathsf{T}_{\leftarrow i}$ has height n_i and width $m_s - m_i$; its cardinal will be written δ_i, and is equal to $\sum_{j=i+1}^{s} d_j n_{j-1}$. We call $\mathsf{T}_{\leftarrow i}$ the ith *translate* of T.

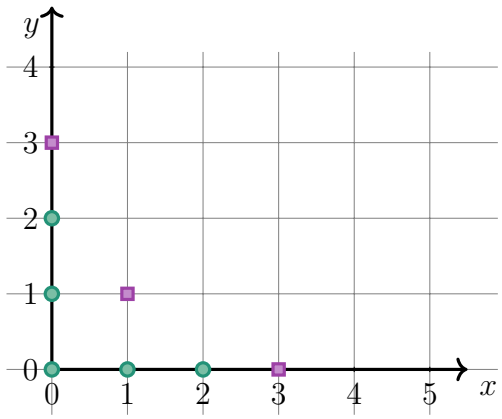

Figure 2: The first translate $\mathsf{T}_{\leftarrow 1}$ of T from Figure 1.

The shell of an initial segment. Let T be a finite initial segment in $\mathbb{N}^2$. In this paragraph, we define its *shell* T', which is another initial segment that forms an outer approximation of T with few generators, while at most doubling the cardinality of T. The definition and the lemma below are from [30, **A.2**]; this construction will be used in the next subsection, to devise an algorithm for the multiplication of polynomials supported on T.

As we did before, we let

$$\boldsymbol{E} = (y^{n_0}, x^{m_1}y^{n_1}, \dots, x^{m_{s-1}}y^{n_{s-1}}, x^{m_s})$$

be the minimal monomial generating set associated to T. We define T' by introducing indices $i_\sigma < i_{\sigma-1} < \cdots < i_0$, defined as follows. Set $i_0 = s$. We let $i_1 \geq 0$ be the largest index less than i_0 and such that $m_{i_1} < m_{i_0}/2$, and iterate the process to define a sequence $i_\sigma = 0 < i_{\sigma-1} < \cdots < i_0 = s$. We can then consider the monomials

$$\boldsymbol{E}' = (y^{n_{i_\sigma}}, x^{m_{i_{\sigma-1}}}y^{n_{i_{\sigma-1}}}, \dots, x^{m_{i_0}}) = (y^{n_0}, x^{m_{i_{\sigma-1}}}y^{n_{i_{\sigma-1}}}, \dots, x^{m_s}),$$

and let T' be the initial segment determined by $\boldsymbol{E}'$.

Lemma 2.2. *The initial segment T' contains T, its cardinal is at most 2δ and σ is in $O(\log(\delta))$.*

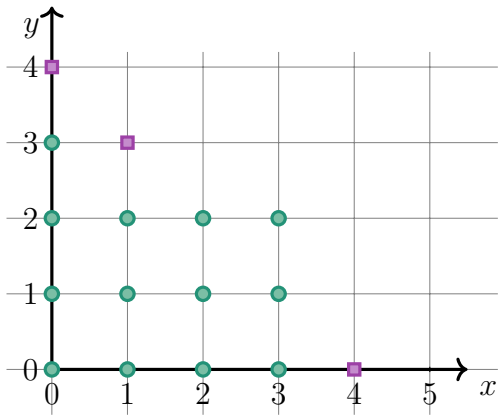

Figure 3: The shell of T from Figure 1.

In our pseudo-code, we will write $\mathsf{T}' \leftarrow \textsc{Shell}(\mathsf{T})$ to indicate that T' is the shell of T. The algorithm Shell does not use any base field or base ring operation, only index manipulations (in particular, it does not show up in our cost analyses).

2.2. Structured polynomial multiplication

We now prove two propositions regarding polynomial multiplication in $\mathbb{A}[x, y]$, for an arbitrary ring $\mathbb{A}$, which will be the basis of the runtime analysis of several algorithms. We mention in all propositions below that the algorithms in this section only use additions and multiplications in $\mathbb{A}$, as we will need this property in the sequel. In what follows, given two sets S, T in $\mathbb{N}^2$, $\mathsf{S} + \mathsf{T}$ denotes their Minkowski sum.

The main prerequisite is the following fact: if $\mathsf{S} \subset \mathbb{N}^2$ is a rectangle, given A and B in $\mathbb{A}[x, y]_{\mathsf{S}}$, we can compute $AB \in \mathbb{A}[x, y]_{\mathsf{S}+\mathsf{S}}$ using $O\tilde{}(|\mathsf{S}|)$ operations $(+, \times)$ in $\mathbb{A}$: if S contains the origin, this is done using Kronecker substitution to reduce to multiplication in $\mathbb{A}[x]$, see [61, Corollary 8.28]; in the general case, we reduce to the situation where S contains the origin by factoring out $x^u y^v$ from A and B, with (u, v) being the unique minimal element of S.

This being said, the first result we highlight here gives the cost of computing the product AB, for A and B supported on the same initial segment T. Note that AB is supported on $\mathsf{T} + \mathsf{T}$, and that if T has height n and width m, $\mathsf{T} + \mathsf{T}$ has cardinal $\Theta(nm)$. Indeed, this set contains the rectangle $\{0, \dots, m-1\} \times \{0, \dots, n-1\}$ of cardinal nm, and is contained in the rectangle $\{0, \dots, 2m-2\} \times \{0, \dots, 2n-2\}$ of cardinal less than $4nm$, so that $|\mathsf{T}+\mathsf{T}| \in \Theta(nm)$. This is to be contrasted with the cardinal of T itself, which can range anywhere between $n + m$ and nm.

Proposition 2.3. *Consider a finite initial segment $\mathsf{T} \subset \mathbb{N}^2$, of height n and width m. Given A and B in $\mathbb{A}[x, y]_{\mathsf{T}}$, one can compute AB using $O\tilde{}(|\mathsf{T} + \mathsf{T}|) = O\tilde{}(nm)$ operations $(+, \times)$ in $\mathbb{A}$.*

Proof. Let S be the rectangle $\{0, \dots, m-1\} \times \{0, \dots, n-1\}$, so that S contains T. Then, A and B are in $\mathbb{A}[x, y]_{\mathsf{S}}$, so we can multiply them using $O\tilde{}(|\mathsf{S} + \mathsf{S}|) = O\tilde{}(nm)$ operations

$(+,\times)$ in $\mathbb{A}$ with Kronecker substitution, as pointed out above, and this runtime is also $O^{\tilde{}}(|\mathsf{T}+\mathsf{T}|)$. □

Our second proposition gives an algorithm to compute $AB \in \mathbb{A}[x,y]$, where A is supported on a rectangle containing the origin and B on an initial segment.

Proposition 2.4. *Consider a rectangle $\mathsf{S} \subset \mathbb{N}^2$ and a finite initial segment $\mathsf{T} \subset \mathbb{N}^2$. Given A in $\mathbb{A}[x,y]_{\mathsf{S}}$ and B in $\mathbb{A}[x,y]_{\mathsf{T}}$, one can compute AB using $O^{\tilde{}}(|\mathsf{S}+\mathsf{T}|)$ operations $(+,\times)$ in $\mathbb{A}$.*

Without loss of generality, we assume that S contains the origin $(0,0)$; if not, as above, factor out the monomial $x^u y^v$ from A, with (u,v) the minimal element in S. We can thus suppose that S is the rectangle $\{0,\dots,\ell-1\}\times\{0,\dots,h-1\}$, for some integers $\ell, h \geq 1$, so in particular $|\mathsf{S}| = \ell h$, and that T is an initial segment of cardinal $|\mathsf{T}| = \delta$, with height n and width m.

If $\mathbb{A}$ is a field of characteristic zero, this result follows directly from the sparse evaluation and interpolation algorithms of [7]. More generally, if $\mathbb{A}$ is a field of cardinal at least $\max(\ell+m, h+n)-1$, this is also the case, using the algorithm in [60]. The algorithm below achieves the same asymptotic runtime, without assumption on $\mathbb{A}$. The proof is slightly more involved than that of the previous proposition, and occupies the rest of this section.

An algorithm when T is a rectangle. Suppose first that $\mathsf{T} = \{0,\dots,m-1\}\times\{0,\dots,n-1\}$, so that $\delta = nm$; then the cardinal of $\mathsf{S}+\mathsf{T}$ is $(\ell+m-1)(h+n-1)$.

Take A in $\mathbb{A}[x,y]_{\mathsf{S}}$ and B in $\mathbb{A}[x,y]_{\mathsf{T}}$. Then, both A and B are in $\mathbb{A}[x,y]_{\mathsf{S}+\mathsf{T}}$. Since $\mathsf{S}+\mathsf{T}$ is a rectangle, we saw in the preamble of this section that using Kronecker's substitution, we can compute their product using $O^{\tilde{}}(|\mathsf{S}+\mathsf{T}|) = O^{\tilde{}}((\ell+m-1)(h+n-1))$ operations $(+,\times)$ in $\mathbb{A}$. In the main algorithm below, this is written KroneckerMultiply(A,B).

A first general algorithm. We now suppose that T is an arbitrary initial segment, and that it is determined by the monomials

$$\boldsymbol{E} = (y^{n_0}, x^{m_1}y^{n_1}, \dots, x^{m_{s-1}}y^{n_{s-1}}, x^{m_s}),$$

with the m_i's increasing, the n_i's decreasing, and $m_0 = n_s = 0$; note that we also have $n_0 = n$ and $m_s = m$. As before, for $i = 1,\dots,s$, we set $d_i = m_i - m_{i-1}$, so that $m_i = d_1 + \cdots + d_i$.

The input $B \in \mathbb{A}[x,y]_{\mathsf{T}}$ can then be written as $B = \sum_{0\leq i<s} B_i x^{m_i}$, with B_i supported on $\mathsf{T}_i = \{0,\dots,d_{i+1}-1\}\times\{0,\dots,n_i-1\}$. To compute AB, with A in $\mathbb{A}[x,y]_{\mathsf{S}}$, we thus compute all AB_i and add up the results.

Algorithm 2.1 MultiplyNaive$(A,\mathsf{S},B,\mathsf{T})$

Input: A in $\mathbb{A}[x,y]_{\mathsf{S}}$, B in $\mathbb{A}[x,y]_{\mathsf{T}}$
Output: AB in $\mathbb{A}[x,y]_{\mathsf{S}+\mathsf{T}}$
1: write $B = B_0 + B_1 x^{m_1} + \cdots + B_{s-1}x^{m_{s-1}}$ with $B_i \in \mathbb{A}[x,y]_{\{0,\dots,d_{i+1}-1\}\times\{0,\dots,n_i-1\}}$ for all i
2: **for** $i = 0,\dots,s-1$ **do** $C_i \leftarrow$ KroneckerMultiply(A,B_i)
3: **return** $C_0 + C_1 x^{m_1} + \cdots + C_{s-1}x^{m_{s-1}}$

By the result in the previous paragraph, each product AB_i can be computed in

$$O\tilde{}((\ell + d_{i+1} - 1)(h + n_i - 1)) = O\tilde{}((\ell - 1)(h - 1) + (\ell - 1)n_i + d_{i+1}(h - 1) + d_{i+1}n_i)$$

operations in $\mathbb{A}$, and the cost of adding this product to the final result fits into the same bound. Using the inequality $n_i \leq n_0 = n$ for all i, as well as $d_1 + \cdots + d_s = m_s = m$ and $d_1 n_0 + \cdots + d_s n_{s-1} = \delta$ (the cardinal of T), we see that the total cost is

$$O\tilde{}(s(\ell - 1)(h - 1) + s(\ell - 1)n + m(h - 1) + \delta).$$

On the other hand, we can determine the cardinal of the sum $\mathsf{U} = \mathsf{S} + \mathsf{T}$ as follows. The set U is the disjoint union of the following sets:

- $\mathsf{U}_1 = \{0, \ldots, \ell - 2\} \times \{0, \ldots, h - 2\}$,
- $\mathsf{U}_2 = (0, h - 1) + \{0, \ldots, \ell - 2\} \times \{0, \ldots, n - 1\}$
- $\mathsf{U}_3 = (\ell - 1, 0) + \{0, \ldots, m - 1\} \times \{0, \ldots, h - 2\}$
- $\mathsf{U}_4 = (\ell - 1, h - 1) + \mathsf{T}$.

This is established by taking (i, j) in S, (v, w) in T, and discussing according to the signs of $v - (\ell - 1 - i)$ and $w - (h - 1 - j)$. As a result, we obtain

$$|\mathsf{S} + \mathsf{T}| = (\ell - 1)(h - 1) + (\ell - 1)n + m(h - 1) + \delta.$$

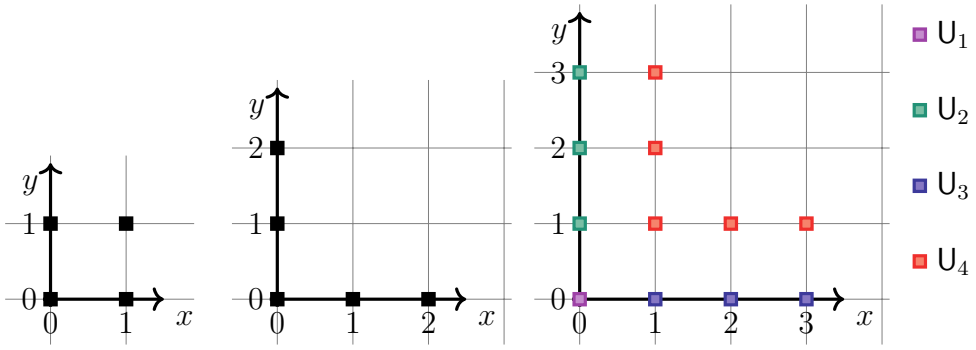

Figure 4: the sets S, T and $\mathsf{U} = \mathsf{S} + \mathsf{T}$, with $\ell = h = 2$ and $n = m = 3$.

The main algorithm.. The runtime reported above does not fit in the target cost $O\tilde{}(|\mathsf{S} + \mathsf{T}|)$, as s could be large. To circumvent this issue, we apply the algorithm of the previous paragraph, but we replace T by its shell T'. We know (Lemma 2.2) that the cardinal of T' is at most 2δ, that its width and height are the same as those of T, and that it is generated by $\sigma \in O(\log(s)) \subset O(\log(\delta))$ terms.

Algorithm 2.2 MULTIPLY$(A, \mathsf{S}, B, \mathsf{T})$

INPUT: A in $\mathbb{A}[x,y]_{\mathsf{S}}$, B in $\mathbb{A}[x,y]_{\mathsf{T}}$
OUTPUT: AB in $\mathbb{A}[x,y]_{\mathsf{S}+\mathsf{T}}$
1: $\mathsf{T}' \leftarrow$ SHELL(T)
2: **return** MULTIPLYNAIVE$(A, \mathsf{S}, B, \mathsf{T}')$

The algorithm of the previous paragraph still applies (since T is contained in T'), and its runtime is then $O^{\tilde{}}((\ell-1)(h-1)\log(\delta)+(\ell-1)n\log(\delta)+m(h-1)+\delta)$ operations $(+,\times)$ in $\mathbb{A}$. Since we saw that $|\mathsf{S}+\mathsf{T}| = (\ell-1)(h-1)+(\ell-1)n+m(h-1)+\delta$, the above expression is indeed in $O^{\tilde{}}(|\mathsf{S}+\mathsf{T}|)$. This finishes the proof of Proposition 2.4.

3. Lexicographic Gröbner bases

In this section, we first review Lazard's structure theorem [37] for lexicographic Gröbner bases in $\mathbb{K}[x,y]$, for a field $\mathbb{K}$, then a parametrization of such bases due to [9]. While the core of the discussion makes no assumption on the ideals we consider, we also highlight the case of ideals that are primary at the origin, that is, $\langle x,y\rangle$-primary.

In all that follows, we use the lexicographic monomial order $\succ$ on $\mathbb{K}[x,y]$ induced by $y \succ x$.

3.1. The structure theorem

Consider a zero dimensional ideal $I \subseteq \mathbb{K}[x,y]$, and let $\mathcal{G} = (g_0, \ldots, g_s)$ be its reduced Gröbner basis, listed in decreasing order. Let further

$$\boldsymbol{E} = (y^{n_0}, x^{m_1}y^{n_1}, \ldots, x^{m_{s-1}}y^{n_{s-1}}, x^{m_s})$$

be the minimal reduced basis of the initial ideal $in(I)$ of I, listed in decreasing order, so the n_i's are decreasing and the m_i's are increasing; as before, we set $m_0 = n_s = 0$.

It follows that g_i has initial term $x^{m_i}y^{n_i}$ for all i; in particular g_0 is monic in y with initial term y^{n_0}.

As in Section 2.1, for $i = 1, \ldots, s$, we set $d_i = m_i - m_{i-1}$, with thus $m_i = d_1 + \cdots + d_i$, and $e_i = n_{i-1} - n_i$.

Lazard proved in [37, Theorem 1] the existence of polynomials $D_1, \ldots, D_s$ in $\mathbb{K}[x]$, all monic in x and of respective degrees $d_1, \ldots, d_s$, such that for $i = 0, \ldots, s$, g_i can be written as M_iG_i, with $M_i = D_1 \cdots D_i \in \mathbb{K}[x]$ and $G_i \in \mathbb{K}[x,y]$ monic of degree n_i in y (for $i = 0$, we set $D_0 = 1$). In particular, for $i = s$, this gives $g_s = M_s = D_1 \cdots D_s$ and $G_s = 1$. In addition, for $i = 0, \ldots, s-1$, we have the membership relation

$$G_i \in \langle G_{i+1},\ D_{i+2}G_{i+2},\ \ldots,\ D_{i+2}\cdots D_s\rangle = \left\langle \frac{g_{i+1}}{M_{i+1}}, \frac{g_{i+2}}{M_{i+1}}, \ldots, \frac{g_s}{M_{i+1}} \right\rangle, \tag{2}$$

where the polynomials $G_{i+1}, D_{i+2}G_{i+2}, \ldots, D_{i+2}\cdots D_s$ also form a Gröbner basis of the ideal they generate (which is the colon ideal $(I : M_{i+1})$, but we do not need this fact). Besides, for all i, $G_i(0,y)$ vanishes only at $y = 0$, *i.e.* $G_i(0,y) = y^{n_i}$, see [37, Theorem 2].

Remark 3.1. *If $\mathcal{G}$ generates an $\langle x, y\rangle$-primary ideal, we have $g_s = x^{m_s} = x^{d_1+\cdots+d_s}$, with thus $D_i = x^{d_i}$ and $M_i = x^{m_i}$ for all i.*

In terms of data structures, representing $\mathcal{G} = (g_0, \ldots, g_s)$ involves $O(s\delta)$ field elements, with δ the degree of I. As a remark, we note that it would be sufficient to store the polynomials $\boldsymbol{D} = (D_1, \ldots, D_s)$ and $\boldsymbol{G} = (G_0, \ldots, G_s)$ instead. If $\mathsf{T} \subset \mathbb{N}^2$ is the initial segment determined by $\boldsymbol{E}$, the structure theorem implies that for $i = 0, \ldots, s$, $G_i - y^{n_i}$ is supported on the ith translate $\mathsf{T}_{\leftarrow i}$ of T. In particular, δ_i field elements are needed to store it, with $\delta_i = |\mathsf{T}_{\leftarrow i}|$, hence a slightly improved total of $O(\sum_{i=0}^{s} \delta_i)$ field elements for $\boldsymbol{D}$ and $\boldsymbol{G}$.

3.2. Conca and Valla's parametrization

In this subsection, we suppose that the tuple $\boldsymbol{E} = (y^{n_0}, x^{m_1}y^{n_1}, \ldots, x^{m_{s-1}}y^{n_{s-1}}, x^{m_s})$ is fixed. Following [9], we are interested in describing the set of ideals I in $\mathbb{K}[x, y]$ that have initial ideal generated by $\boldsymbol{E}$. We call this set the *Gröbner cell* of $\boldsymbol{E}$, and we write it $\mathcal{C}(\boldsymbol{E}) := \{I \mid in(I) = \langle \boldsymbol{E}\rangle\}$. We will also mention a subset of it, the set of ideals I in $\mathbb{K}[x, y]$ with initial ideal generated by $\boldsymbol{E}$ and that are $\langle x, y\rangle$-primary; this is called the *punctual* Gröbner cell of $\boldsymbol{E}$, and is written $\mathcal{C}_0(\boldsymbol{E})$.

The idea of describing ideals with a prescribed initial ideal goes back to [6, 5, 31] for ideals in $\mathbb{K}[\![x, y]\!]$ and [8] for $\mathbb{K}[x_1, \ldots, x_n]$; it was developed in many further references, such as [16, 24, 49, 29, 28, 53, 41, 40]. It is known that these Gröbner cells, also called strata, have corresponding moduli spaces that are affine schemes (see [40, Section 8]), but to our knowledge, no general an explicit description has not yet been given. In our case, however, Conca and Valla obtained in [9] a complete description of Gröbner cells and punctual Gröbner cells for bivariate ideals under the lexicographic order (following previous work of [16], where the dimensions of these cells were already made explicit).

Example 3.2. *For an example of a punctual Gröbner cell, taking $\boldsymbol{E} = (y^4, xy^3, x^2y, x^4)$ as in Figure 1, using the facts that $g_i = x^{m_i}G_i$ and that $G_i(0, y) = y^{n_i}$, we deduce that the lexicographic Gröbner basis of an ideal in $\mathcal{C}_0(\boldsymbol{E})$ necessarily has the following shape, for some coefficients $c_1, \ldots, c_8$ in $\mathbb{K}$:*

$$\begin{aligned} g_1 &= y^4 + c_1xy^2 + c_2xy + c_3x^3 + c_4x^2 + c_5x \\ g_2 &= xy^3 + c_6x^3 + c_7x^2 \\ g_3 &= x^2y + c_8x^3 \\ g_4 &= x^4 \end{aligned}$$

So far, though, we have not taken into account the membership equality in (2), which imposes relations on the coefficients c_i. The parametrizations of $\mathcal{C}(\boldsymbol{E})$ and $\mathcal{C}_0(\boldsymbol{E})$ given below resolve this issue.

Recall that we write $d_i = m_i - m_{i-1}$ and $e_i = n_{i-1} - n_i$, for $i = 1, \ldots, s$. Given I in $\mathcal{C}(\boldsymbol{E})$, Conca and Valla prove in [9, Lemma 3.6] the existence and uniqueness of polynomials $(\sigma_{j,i})_{0\le i\le s-1, i\le j\le s}$ in $\mathbb{K}[x, y]$ with the following degree constraints:

- for all $i = 0, \ldots, s-1$ and $j = i, \ldots, s$, $\deg(\sigma_{j,i}, x) < d_{i+1}$
- for all $i = 0, \ldots, s-1$, $\sigma_{i,i}$ is in $\mathbb{K}[x]$ and $\deg(\sigma_{j,i}, y) < e_j$ holds for $j = i+1, \ldots, s$,

and such that the following properties hold. Define polynomials $\mathcal{H} = (h_0, \ldots, h_s)$ in $\mathbb{K}[x, y]$ by

- $h_s = (x^{d_1} - \sigma_{0,0}) \cdots (x^{d_s} - \sigma_{s-1,s-1})$
- for $i = 0, \ldots, s-1$,

$$x^{d_{i+1}} h_i - y^{e_{i+1}} h_{i+1} = \sigma_{i,i} h_i + \sigma_{i+1,i} h_{i+1} + \cdots + \sigma_{s,i} h_s; \tag{3}$$

then, all polynomials h_i's are in I. Since the relations above imply that for $i = 0, \ldots, s$, h_i has initial term $x^{m_i} y^{n_i}$, $\mathcal{H} = (h_0, \ldots, h_s)$ is a minimal Gröbner basis of I. (Note that Eq. (3) then gives the normal form of the syzygy between h_i and h_{i+1}.)

Conversely, for any choice of the polynomials $\sigma_{j,i}$ satisfying the degree constraints above, the resulting polynomials $\mathcal{H}$ form a minimal Gröbner basis of an ideal I in $\mathcal{C}(\boldsymbol{E})$.

Let us briefly mention some properties of the polynomials $h_0, \ldots, h_s$. First, we claim that they have x-degree either exactly m_s (for h_s), or less than m_s, for $h_0, \ldots, h_{s-1}$. This is true for h_s by construction. For the other indices, this follows from a decreasing induction, by rewriting (3) as

$$(x^{d_{i+1}} - \sigma_{i,i}) h_i = y^{e_{i+1}} h_{i+1} + \sigma_{i+1,i} h_{i+1} + \cdots + \sigma_{s,i} h_s, \tag{4}$$

where all terms $\sigma_{j,i} h_j$ on the right have x-degree less than $d_{i+1} + m_s$.

Next, note that for $i = 0, \ldots, s$, $(x^{d_1} - \sigma_{0,0}) \cdots (x^{d_i} - \sigma_{i-1,i-1})$ divides h_i, and thus all polynomials $h_i, \ldots, h_s$; this follows from (4) by a decreasing induction (for $i = 0$, the empty product is set to 1). Since h_i has initial term $x^{m_i} y^{n_i} = x^{d_1 + \cdots + d_i} y^{n_i}$, we deduce that $(x^{d_1} - \sigma_{0,0}) \cdots (x^{d_i} - \sigma_{i-1,i-1})$ is precisely the polynomial coefficient of y^{n_i} in h_i.

Let then $\mathcal{G} = (g_0, \ldots, g_s)$ be the reduced Gröbner basis obtained by inter-reducing $\mathcal{H}$. Since none of the terms in $(x^{d_1} - \sigma_{0,0}) \cdots (x^{d_i} - \sigma_{i-1,i-1}) y^{n_i}$ can be reduced by $h_0, \ldots, h_{i-1}$ or $h_{i+1}, \ldots, h_s$, we see that $(x^{d_1} - \sigma_{0,0}) \cdots (x^{d_i} - \sigma_{i-1,i-1})$ is also the polynomial coefficient of y^{n_i} in g_i. Hence, the polynomials D_i and M_i that appear in Lazard's structure theorem are respectively given by $D_i = x^{d_i} - \sigma_{i-1,i-1}$ and $M_i = (x^{d_1} - \sigma_{0,0}) \cdots (x^{d_i} - \sigma_{i-1,i-1})$.

Remark 3.3. *We can recover Lazard's result, that M_i divides g_i for all i, from this discussion: the reduction of h_i by $\mathcal{H}$ can only involve $h_{i+1}, \ldots, h_s$ (since the y-degree of the other polynomials $h_0, \ldots, h_{i-1}$ is too large). We saw that M_i divides h_i, but then also all of $h_{i+1}, \ldots, h_s$; as a result, it divides the remainder g_i.*

Altogether, the total number N of coefficients that appear in the polynomials $(\sigma_{j,i})_{0\leq i\leq s-1, i\leq j\leq s}$, for the Gröbner cell $\mathcal{C}(\boldsymbol{E})$, is given by

$$\begin{aligned} N &= \sum_{i=0}^{s-1}\left(\sum_{j=i+1}^{s} d_{i+1}e_j + d_{i+1}\right) \\ &= \sum_{i=0}^{s-1} d_{i+1}n_i + \sum_{i=0}^{s-1} d_{i+1} \\ &= \delta + m_s, \end{aligned}$$

with δ the degree of $\boldsymbol{E}$. These coefficients will be written $\lambda_1, \ldots, \lambda_N$ and called *Gröbner parameters*; this gives us a bijection $\Phi_{\boldsymbol{E}}$ between $\mathbb{K}^N$ and $\mathcal{C}(\boldsymbol{E})$.

The elements in the *punctual* Gröbner cell $\mathcal{C}_0(\boldsymbol{E})$ are obtained by setting some of the Gröbner parameters to zero, corresponding to the following extra conditions:

- the polynomials $\sigma_{0,0}, \ldots, \sigma_{s-1,s-1}$ vanish (recall that for the punctual Gröbner cell, we have $D_i = x^{d_i}$ and $M_i = x^{m_i}$ for all i, see Remark 3.1)
- $\sigma_{i+1,i}$ is divisible by x, for $i = 0, \ldots, s-1$.

The number of remaining coefficients in $\sigma_{1,0}, \ldots, \sigma_{s,s-1}$ is

$$\begin{aligned} N_0 &= \sum_{i=0}^{s-1}\left(\sum_{j=i+1}^{s} d_{i+1}e_j - e_{i+1}\right) \\ &= \sum_{i=0}^{s-1} d_{i+1}n_i - \sum_{i=0}^{s-1} e_{i+1} \\ &= \delta - n_0, \end{aligned}$$

establishing a bijection between $\mathbb{K}^{N_0}$ and $\mathcal{C}_0(\boldsymbol{E})$. In the $\langle x, y\rangle$-primary case, the degree δ of $\boldsymbol{E}$ is by definition the common multiplicity of all ideals in $\mathcal{C}_0(\boldsymbol{E})$ at the origin.

Example 3.4. *Let us describe the punctual Gröbner cell of $\boldsymbol{E}$ in our running example (Example 1.1). It has dimension $N_0 = 9 - 4 = 5$, so that we can use parameters $\lambda_1, \ldots, \lambda_5$, with polynomials $(\sigma_{j,i})$ of the form*

$$\sigma_{0,0} = \sigma_{1,0} = 0, \quad \sigma_{2,0} = \lambda_1 y + \lambda_2, \quad \sigma_{3,0} = \lambda_3, \quad \sigma_{1,1} = n_{2,1} = 0, \quad \sigma_{3,1} = \lambda_4, \quad \sigma_{2,2} = 0, \quad \sigma_{3,2} = \lambda_5 x.$$

Then, the ideals in $\mathcal{C}_0(\boldsymbol{E})$ are exactly those ideals with Gröbner bases as follows:

$$\begin{aligned} h_0 &= y^4 + \lambda_5 xy^3 + \lambda_1 xy^2 + (\lambda_1\lambda_5 + \lambda_4)x^2y + \lambda_2 xy + \lambda_3 x^3 + \lambda_2\lambda_5 x^2 \\ h_1 &= xy^3 + \lambda_5 x^2y^2 + \lambda_4 x^3 \\ h_2 &= x^2y + \lambda_5 x^3 \\ h_3 &= x^4. \end{aligned}$$

As expected, these are not reduced Gröbner bases. After reduction, we obtain the following polynomials $\mathcal{G}$:

$$\begin{aligned} g_0 &= y^4 + \lambda_1 xy^2 + \lambda_2 xy + (-\lambda_1\lambda_5^2 + \lambda_3 - 2\lambda_4\lambda_5)x^3 + \lambda_2\lambda_5 x^2 \\ g_1 &= xy^3 + \lambda_4 x^3 \\ g_2 &= x^2 y + \lambda_5 x^3 \\ g_3 &= x^4. \end{aligned} \tag{5}$$

4. Reduction modulo a lexicographic Gröbner basis

As before, suppose that $\mathcal{G} = (g_0, \ldots, g_s)$ is a lexicographic Gröbner basis in $\mathbb{K}[x, y]$, with initial segment $\mathsf{T} \subset \mathbb{N}^2$. Given f in $\mathbb{K}[x, y]$, we are interested in computing the remainder $r = f \text{ rem } \mathcal{G} \in \mathbb{K}[x, y]_{\mathsf{T}}$; this will be used on multiple occasions in this paper, and is also an interesting question in itself.

Polynomial reduction has been discussed in the literature, for an arbitrary order in [26], more specifically in the bivariate setting for certain weighted orderings in [59], and for the degree lexicographic ordering in [27]. The latter two articles used a dichotomic scheme, from which we will draw our inspiration.

We start by developing the necessary background as a problem in plane geometry, closely following [59]. We continue with algorithms to convert polynomials into a so-called *mixed-radix* representation, and back; the reduction algorithm itself is then given in the last subsection.

4.1. A paving problem

For $\mathcal{G}$ as above and f in $\mathbb{K}[x, y]$, the remainder $r = f \text{ rem } \mathcal{G}$ is uniquely defined, but the quotients Q_i in the relation $f = Q_0 g_0 + \cdots + Q_s g_s + r$ are not. The reduction algorithm will obtain r by computing the Q_i's one after the other. Hence, to completely specify the algorithm, we need to make these quotients unambiguous: whenever a monomial $x^u y^v$ can be reduced by more than one of the Gröbner basis elements, we must prescribe which of the g_i's is used. The cost of the resulting algorithm will depend in an essential manner on these decisions.

In [59], van der Hoeven and Larrieu introduced a dichotomic scheme, in the context of reduction modulo certain "nice" Gröbner bases (called *vanilla Gröbner bases*), for weighted degree orderings. In this subsection, we adapt their construction to our situation; prior to that, let us briefly point out what vanilla Gröbner bases are: for a weighted ordering, the Gröbner basis $\mathcal{G}$ of an ideal I of degree δ is vanilla if the standard monomials modulo $\mathcal{G}$ are precisely the δ smallest monomials. This definition makes it possible for van der Hoeven and Larrieu to give a compact representation of such Gröbner bases, by means of certain "retraction coefficients" that specify relations between the elements of $\mathcal{G}$; it would be of interest to understand to what extent these can be used in the description of Gröbner cells in the weighted ordering context.

Back to our situation, suppose as before that the initial terms of $\mathcal{G}$ are the monomials

$$\boldsymbol{E} = (y^{n_0}, x^{m_1} y^{n_1}, \ldots, x^{m_{s-1}} y^{n_{s-1}}, x^{m_s});$$

we still write $d_i = m_i - m_{i-1}$ and $e_i = n_{i-1} - n_i$, for $i = 1, \ldots, s$. The set of monomials to which we will apply the main reduction algorithm is $\{x^u y^v,\ 0 \le u < m_s,\ 0 \le v < n_0\}$, so it has cardinal $n_0 m_s$ (the general case will be reduced to this situation). In particular, neither g_0 nor g_s can reduce any of these monomials.

We can then translate our question into a paving problem in the plane. We want to cover $\mathsf{S} = \{0, \ldots, m_s - 1\} \times \{0, \ldots, n_0 - 1\} - \mathsf{T}$ by rectangles, under the following constraints:

- we use $s - 1$ pairwise disjoint rectangles, $\mathsf{R}_1, \ldots, \mathsf{R}_{s-1}$, so that R_i will index the set of monomials that are reduced using g_i
- for all i, R_i has the form $\{m_i, \ldots, m_{i+\ell_i} - 1\} \times \{n_i, \ldots, n_{i-h_i} - 1\}$, for some positive integers ℓ_i, h_i such that $i + \ell_i \le s$ and $i - h_i \ge 0$
- the union of all R_i's covers S.

The sequence $((\ell_1, h_1), \ldots, (\ell_{s-1}, h_{s-1}))$ is sufficient to specify such a paving. Our goal is then to minimize the quantity

$$c := n_0 \sum_{i=1}^{s-1} (m_{i+\ell_i} - m_i) + m_s \sum_{i=1}^{s-1} (n_{i-h_i} - n_i),$$

where $(m_{i+\ell_i} - m_i)$ and $(n_{i-h_i} - n_i)$ are respectively the width and height of R_i. This quantity will turn out to determine the cost of the reduction algorithm; the target is to keep c in $O\tilde{}(n_0 m_s)$, since we mentioned that $n_0 m_s$ in an upper bound on the number of monomials in the polynomials we want to reduce.

The following figure shows two possible pavings, for the case $d = 4$ of the family already seen in the proof of Lemma 2.1, with $\boldsymbol{E} = (y^d, xy^{d-1}, \ldots, x^d)$. For this family, $n_0 = m_s = d$ and $n_0 m_s = d^2$; the strategies shown in the example below have either $\sum_{i=1}^{s-1}(m_{i+\ell_i} - m_i)$ or $\sum_{i=1}^{s-1}(n_{i-h_i} - n_i)$ in $\Theta(d^2)$, so c is in $\Theta(d^3) = \Theta((n_0 m_s)^{1.5})$ in either case.

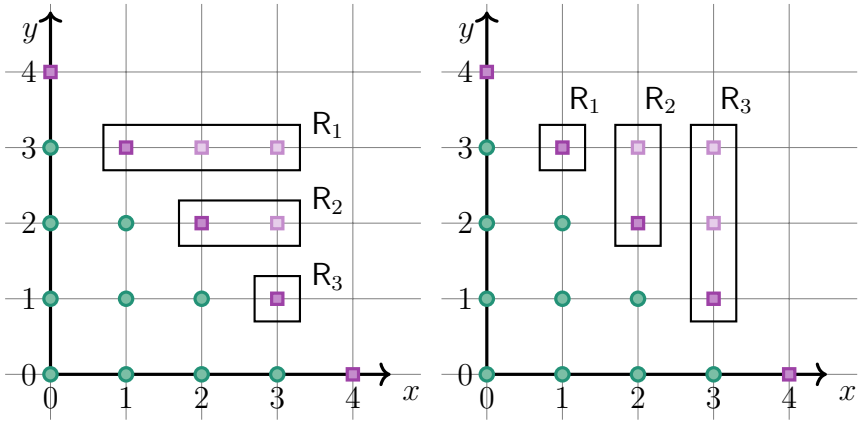

Figure 5: two possible pavings with $d = 4$.

For this family, a better solution is given below.

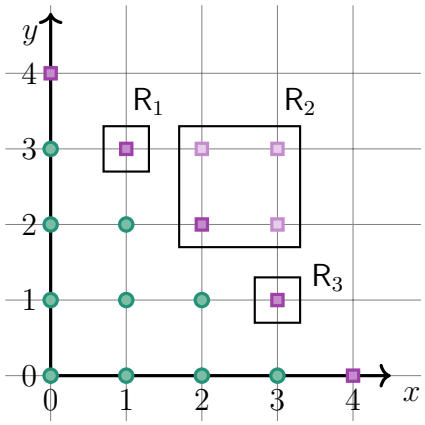

Figure 6: an improved paving.

This design was introduced in [59], for "vanilla" families $\boldsymbol{E}$ similar to the one in the example, where the d_i (which are the widths of the steps in the Gröbner staircase, see Figure 1) are almost constant, and all e_i (which are the heights of these steps) are equal to 1. The construction we give below for arbitrary inputs is derived from it directly, replacing vertices with coordinates such as (i, j) by vertices with coordinates of the form (n_i, m_j). In what follows, $\mathrm{val}_2(i)$ denotes the 2-adic valuation of a positive integer i.

Definition 4.1. *For $i = 1, \ldots, s-1$, define:*

- $h_i = 2^{\mathrm{val}_2(i)}$
- $\ell_i = \min(h_i, s - i)$

As a result, the rectangle R_i is given by

$$\begin{aligned}\mathsf{R}_i &= \{m_i, \ldots, m_{i+\ell_i} - 1\} \times \{n_i, \ldots, n_{i-h_i} - 1\} \\ &= \{m_i, \ldots, m_{\min(i+h_i,s)} - 1\} \times \{n_i, \ldots, n_{i-h_i} - 1\}.\end{aligned}$$

The following three propositions give the main properties of these rectangles. First, we prove that they cover all points not in T.

Proposition 4.2. *For any s and any choices of $m_1, \ldots, m_s$ and $n_0, \ldots, n_{s-1}$, the rectangles $\mathsf{R}_1, \ldots, \mathsf{R}_{s-1}$ are pairwise disjoint, cover $\mathsf{S} = \{0, \ldots, m_s - 1\} \times \{0, \ldots, n_0 - 1\} - \mathsf{T}$, and satisfy $i + \ell_i \leq s$ and $i - h_i \geq 0$ for all i.*

Proof. The last claim is a direct consequence of the definitions. We prove the rest of the proposition by reduction to the case where all d_i's and e_i's are equal to one. The proof is technical but raises no special difficulty.

For any positive integer s, we define the monomials $\mathscr{E}_s = (x^i y^{s-i},\ 0 \leq i \leq s)$, the initial segment $\mathscr{T}_s$ determined by $\mathscr{E}_s$ and $\mathscr{S}_s = \{0, \ldots, s-1\} \times \{0, \ldots, s-1\} - \mathscr{T}_s$; note that $\mathscr{T}_s$ is the set of all pairs of non-negative integers (a, b) with $b < s - a$. Finally, for $i = 1, \ldots, s-1$ we define the rectangle $\mathscr{R}_{i,s} = \{i, \ldots, \min(i + h_i, s) - 1\} \times \{s - i, \ldots, s - i + h_i - 1\} \subset \mathscr{S}_s$.

We start from $m_1, \dots, m_s$ and $n_0, \dots, n_{s-1}$ as in the proposition's statement, with corresponding sets T and S in $\mathbb{N}^2$. Take a point (u, v) in S. Because $u < m_s$, there exists a unique pair (α, u') such that $u = m_\alpha + u'$, with $0 \leq \alpha \leq s-1$ and $0 \leq u' < d_{\alpha+1}$. Similarly, because $v < n_0$, there exists a unique pair (β, v') such that $v = n_\beta + v'$, with $1 \leq \beta \leq s$ and $0 \leq v' < e_\beta$. We claim that $(\alpha, s-\beta)$ is in the set $\mathscr{S}_s$ defined in the previous paragraph, and that for $i = 1, \dots, s-1$, (u, v) is in the rectangle R_i if and only if $(\alpha, s-\beta)$ is in the rectangle $\mathscr{R}_{i,s}$.

- For the first claim, we already pointed out the inequalities $0 \leq \alpha \leq s-1$ and $1 \leq \beta \leq s$, which gives $0 \leq s-\beta \leq s-1$, so that $(\alpha, s-\beta)$ is in the square $\{0, \dots, s-1\} \times \{0, \dots, s-1\}$. On the other hand, we have $v \geq n_\alpha$ (otherwise (α, β) would be in T), and so $\beta \leq \alpha$ and $s-\beta \geq s-\alpha$. This proves that the point $(\alpha, s-\beta)$ is not in $\mathscr{T}_s$, so altogether, it lies in $\mathscr{S}_s$.

- For the second claim, note that since $u = m_\alpha + u'$, with $0 \leq u' < d_{\alpha+1}$, $m_i \leq u < m_{\min(i+h_i, s)}$ is equivalent to $i \leq \alpha < \min(i+h_i, s)$. Similarly, the inequalities $n_i \leq v < n_{i-h_i}$ are equivalent to $s-i \leq s-\beta < s-i+h_i$. This proves the claim.

To conclude, it is now sufficient to prove that for all s, the following property, written $P(s)$, holds: the rectangles $\mathscr{R}_{1,s}, \dots, \mathscr{R}_{s-1,s}$ are pairwise disjoint and cover $\mathscr{S}_s$. First, we prove it for s a power of two, of the form $s = 2^k$, by induction on $k \geq 1$. For $k = 1$ (so $s = 2$), there is nothing to prove, as $\mathscr{S}_2 = \{1\} \times \{1\} = \mathscr{R}_{1,2}$.

Supposing that $P(s)$ is true for $s = 2^k$, we now prove it for $s' = 2s$. For $\mathscr{S}$ a subset of $\mathbb{N}^2$, we write $\mathscr{S} \cap \{x \leq t\}$ for the set of all (x, y) in $\mathscr{S}$ with $x \leq t$. The sets $\mathscr{S} \cap \{x \geq t\}$, $\mathscr{S} \cap \{x \leq t, y \leq t'\}$, etc, are defined similarly.

First, we note that for any power of two $\sigma = 2^t$ and $i = 1, \dots, \sigma-1$, we have $i + h_i \leq \sigma$, so the rectangle $\mathscr{R}_{i,\sigma}$ is simply $\mathscr{R}_{i,\sigma} = \{i, \dots, i+h_i-1\} \times \{\sigma-i, \dots, \sigma-i+h_i-1\}$. As a result, the rectangles $\mathscr{R}_{1,s'}, \dots, \mathscr{R}_{s-1,s'}$ are translates of $\mathscr{R}_{1,s}, \dots, \mathscr{R}_{s-1,s}$ by $(0, s)$, so by the induction assumption, they are pairwise disjoint, cover $\mathscr{S}_{s'} \cap \{x \leq s-1\}$, and do not meet $\mathscr{S}_{s'} \cap \{x \geq s\}$ (on Figure 6, we have $s = 2$, $s' = 4$, and there is only one such rectangle, written R_1). Since $h_i = h_{i+s}$ for $i = 1, \dots, s-1$, we also deduce that the rectangles $\mathscr{R}_{s+1,s'}, \dots, \mathscr{R}_{2s-1,s'}$ are translates of $\mathscr{R}_{1,s}, \dots, \mathscr{R}_{s-1,s}$ by $(s, 0)$. Thus, they are pairwise disjoint, cover $\mathscr{S}_{s'} \cap \{x \geq s, y \leq s-1\}$, and do not meet $\mathscr{S}_{s'} \cap \{x \geq s, y \geq s\}$ (on Figure 6, this is R_3). Finally, $\mathscr{R}_{s,s'}$ is the rectangle $\{s, 2s-1\} \times \{s, 2s-1\}$ (on Figure 6, this is R_2). Altogether, $P(s')$ holds and the induction is complete.

The last step is to prove that $P(s)$ holds for all s, knowing that it holds for all powers of two. Let s be arbitrary and let s' be the first power of two greater than or equal to s, so that we know that $P(s')$ holds. Let $s'' = s'/2$. Since $s' < 2s$, $s'' \leq s$. For $i < s''$, $\mathscr{R}_{i,s} = \mathscr{R}_{i,s'} - (s'-s, 0)$, whereas for $s'' \leq i \leq s-1$, $\mathscr{R}_{i,s} = \mathscr{R}_{i,s'} \cap \{x \leq s-1\} - (s'-s, 0)$. Knowing $P(s')$, this implies that all these sets are pairwise disjoint. In addition, they cover $\mathscr{S}_{s'} \cap \{x \leq s-1\} - (s'-s, 0)$, which is none other that $\mathscr{S}_s$. Thus, $P(s)$ is proved. □

Second, we prove a monotonicity property: going from left to right along any given horizontal line, the indices of the R_i's that we meet increase. To state the exact property we need, we will also write $\mathsf{R}_0 = \mathsf{T}$.

Proposition 4.3. *Given u, u' in $\{0, \dots, m_s - 1\}$ and v in $\{0, \dots, n_0 - 1\}$, with $u \leq u'$, if (u, v) is in R_i and (u', v) is in $\mathsf{R}_{i'}$, for some indices i, i' in $\{0, \dots, s\}$, then $i \leq i'$.*

Proof. If $i = 0$, there is nothing to prove, since $i' \geq 0$. If $i' = 0$, this means that (u', v) is in T, but then so is (u, v), and thus $i = 0$ as well. Altogether, we can now assume that both i and i' are positive, that is, neither (u, v) nor (u', v) is in $\mathsf{R}_0 = \mathsf{T}$.

The proof proceeds as in the previous proposition, so we will freely reuse some objects introduced there, such as the rectangles $\mathscr{R}_{i,s}$. As in the previous proof, to (u, v) and (u', v), we associate α, α' in $\{0, \dots, s-1\}$ and β in $\{1, \dots, s\}$ such that $(\alpha, s - \beta)$ is in $\mathscr{R}_{i,s}$ and $(\alpha', s - \beta)$ is in $\mathscr{R}_{i',s}$. Assuming that $u \leq u'$, we get that $\alpha \leq \alpha'$, so that we have reduced our question to its analogue for the rectangles $\mathscr{R}_{1,s}, \dots, \mathscr{R}_{s-1,s}$. Explicitly, we prove that indices increase as we travel left-to-right along horizontal lines through $\mathscr{R}_{1,s} \cup \cdots \cup \mathscr{R}_{s-1,s}$.

The strategy is again a proof by induction, starting with the case of s a power of two. There is nothing to prove for $s = 2$; going from a power of two s to $s' = 2s$, the conclusion follows from the observations made in the proof of the previous proposition, that $\mathscr{R}_{1,s'}, \dots, \mathscr{R}_{s-1,s'}$ and $\mathscr{R}_{s+1,s'}, \dots, \mathscr{R}_{2s-1,s'}$ are translates of $\mathscr{R}_{1,s}, \dots, \mathscr{R}_{s-1,s}$ by respectively $(0, s)$ and $(s, 0)$, and that $\mathscr{R}_{s,s'}$ is the rectangle $\{s, 2s-1\} \times \{s, 2s-1\}$. Thus, if indices increase as we travel left-to-right along horizontal lines through $\mathscr{R}_{1,s} \cup \cdots \cup \mathscr{R}_{s-1,s}$, it remains true for $\mathscr{R}_{1,s'} \cup \cdots \cup \mathscr{R}_{s'-1,s'}$.

Finally, we prove the claim for an arbitrary s using the fact that it holds for the next power of two s'. As in the previous proposition, we observe that $\mathscr{R}_{1,s}, \dots, \mathscr{R}_{s-1,s}$ are obtained from $\mathscr{R}_{1,s'}, \dots, \mathscr{R}_{s-1,s'}$ by translation by $(s - s', 0)$, and a right truncation; this is enough to conclude. □

Finally, the key property of this construction is that the corresponding value of $c = n_0 \sum_{i=1}^{s-1}(m_{i+\ell_i} - m_i) + m_s \sum_{i=1}^{s-1}(n_{i-h_i} - n_i)$ is softly linear in $n_0 m_s$. This is close to optimal, since the inequalities $\sum_{i=1}^{s-1}(m_{i+\ell_i} - m_i) \geq m_s - 1$ and $\sum_{i=1}^{s-1}(n_{i-h_i} - n_i) \geq n_0 - 1$ imply that c is in $\Omega(n_0 m_s)$.

Proposition 4.4. *For $\mathsf{R}_1, \dots, \mathsf{R}_{s-1}$ as above, $c = n_0 \sum_{i=1}^{s-1}(m_{i+\ell_i} - m_i) + m_s \sum_{i=1}^{s-1}(n_{i-h_i} - n_i)$ is in $O\tilde{}(n_0 m_s)$.*

Proof. We prove that with the choices in Definition 4.1, $\sum_{i=1}^{s-1}(m_{i+\ell_i} - m_i)$ is in $O\tilde{}(m_s)$; we omit the remaining part of the argument that proves that $\sum_{i=1}^{s-1}(n_{i-h_i} - n_i)$ is in $O\tilde{}(n_0)$ in a similar manner.

First, we reduce to the case where s is a power of 2. For $i \geq s$, set $\ell_i = 0$ and $m_i = m_s$; the sum $\sum_{i=1}^{s-1}(m_{i+\ell_i} - m_i)$ is then equal to $\sum_{i=1}^{s'-1}(m_{i+\ell_i} - m_i)$, where $s' = 2^k$ is the first power of two greater than or equal to s. Besides, this convention implies $m_{i+\ell_i} = m_{i+h_i}$ for all i.

For a given κ in $\{0, \dots, k-1\}$, the indices $i \in \{1, \dots, s'-1\}$ of 2-adic valuation κ are the integers $2^\kappa(1 + 2j)$, for $j = 1, \dots, 2^{k-\kappa-1} - 1$, so we can rewrite the sum $\sum_{i=1}^{s'-1}(m_{i+\ell_i} - m_i)$ as

$$\sum_{\kappa=0}^{k-1} \sum_{j=0}^{2^{k-\kappa-1}-1} (m_{2^\kappa(1+2j)+2^\kappa} - m_{2^\kappa(1+2j)}) = \sum_{\kappa=0}^{k-1} \sum_{j=0}^{2^{k-\kappa-1}-1} (d_{2^\kappa(1+2j)+1} + \cdots + d_{2^\kappa(1+2j)+2^\kappa}),$$

where we set $d_i = 0$ for $i > s$. In particular, for a fixed κ, the last index occurring at the summation step j is less than the first index occurring at $j+1$, so the inner sum is bounded above by $\sum_{i=1}^{s'} d_i = m_s$. It follows that $\sum_{i=1}^{s-1}(m_{i+\ell_i} - m_i) \leq \sum_{\kappa=0}^{k-1} m_s \in O(m_s \log(s))$. Since $s \leq m_s$, our claim is proved. □

4.2. Mixed radix representation

In this subsection, we discuss an alternative basis for our polynomials. Our motivation is the following: if $\mathcal{G} = (g_0, \ldots, g_s)$ is the minimal, reduced lexicographic Gröbner basis that we want to use in our reduction algorithm, we saw that for $i = 0, \ldots, s$, g_i can be written as $M_i G_i$, with M_i of degree m_i in $\mathbb{K}[x]$ and $G_i \in \mathbb{K}[x, y]$ monic in y, of degree n_i in y. Recall also that for $i = 1, \ldots, s$ we write $D_i = M_i / M_{i-1}$, which is a polynomial of degree $d_i = m_i - m_{i-1}$ in $\mathbb{K}[x]$.

The main reduction algorithm will perform many univariate reductions modulo the polynomials $M_1, \ldots, M_s$. When working with $\langle x, y\rangle$-primary ideals, all M_i's are powers of x, so these operations are free of arithmetic cost. In general, though, this is not the case anymore, if the inputs are represented on the monomial basis. In this paragraph, we introduce a mixed radix representation where reductions by the M_i's are free, and we discuss conversion algorithms.

Given polynomial $\boldsymbol{K} = (K_1, \ldots, K_t)$ in $\mathbb{K}[x]$, with respective degrees $k_1, \ldots, k_t$, and writing $h = k_1 + \cdots + k_t$, we consider the $\mathbb{K}$-linear mapping

$$\begin{aligned}\Phi_{\boldsymbol{K}} : \mathbb{K}[x]_{<k_1} \times \cdots \times \mathbb{K}[x]_{<k_t} &\to \mathbb{K}[x]_{<h} \\ (F_1, \ldots, F_t) &\mapsto F_1 + K_1 F_2 + K_1 K_2 F_3 + \cdots + K_1 \cdots K_{t-1} F_t.\end{aligned}$$

The domain and codomain both have dimension h; from this, we easily deduce that $\Phi_{\boldsymbol{K}}$ is a $\mathbb{K}$-vector space isomorphism. For F in $\mathbb{K}[x]_{<h}$, we call $(F_1, \ldots, F_t) = \Phi_{\boldsymbol{K}}^{-1}(F)$ its *mixed radix* representation with respect to the basis $\boldsymbol{K}$.

We will rely on the following fact: given $(F_1, \ldots, F_t) = \Phi_{\boldsymbol{K}}^{-1}(F)$, for i in $\{1, \ldots, t\}$, the mixed radix representation of F div $K_1 \cdots K_i$, with respect to the basis $(K_{i+1}, \ldots, K_t)$, is $(F_{i+1}, \ldots, F_t)$, so we have access to it free of cost. Similarly, the mixed radix representation of F rem $K_1 \cdots K_i$, with respect to the basis $(K_1, \ldots, K_i)$, is $(F_1, \ldots, F_i)$. In particular, if F is given in its mixed radix representation, quotient and remainder by the product $K_1 \cdots K_i$ are free; we still denote these operations by div and rem.

Conversely, for F of degree less than $k_{i+1} + \cdots + k_t$, given on the mixed radix basis associated to $(K_{i+1}, \ldots, K_t)$ as a vector $(F_{i+1}, \ldots, F_t)$, the mixed radix representation of $K_1 \cdots K_i F$, for the basis $(K_1, \ldots, K_t)$, is $(0, \ldots, 0, F_{i+1}, \ldots, F_t)$, so it can be computed for free.

For completeness, we give algorithms with softly linear runtime to apply $\Phi_{\boldsymbol{K}}$ and its inverse. These are elementary variants of the algorithms for Chinese remaindering in [61, Chapter 10.3], or generalized Taylor expansion [61, Chapter 9.2]. We start with the conversion from the mixed radix to monomial representation.

Algorithm 4.1 FromMixedRadix$((F_1, \ldots, F_t), (K_1, \ldots, K_t))$

Input: $(F_1, \ldots, F_t)$ in $\mathbb{K}[x]_{<k_1} \times \cdots \times \mathbb{K}[x]_{<k_t}$, $\boldsymbol{K} = (K_1, \ldots, K_t)$ of respective degrees $k_1, \ldots, k_t$
Output: $\Phi_{\boldsymbol{K}}(F_1, \ldots, F_t) \in \mathbb{K}[x]_{<h}$, with $h = k_1 + \cdots + k_t$
1: **if** $t = 1$ **then return** F_1
2: $t' \leftarrow \lceil t/2 \rceil$
3: $L \leftarrow$ FromMixedRadix$((F_1, \ldots, F_{t'}), (K_1, \ldots, K_{t'}))$
4: $R \leftarrow$ FromMixedRadix$((F_{t'+1}, \ldots, F_t), (K_{t'+1}, \ldots, K_t))$
5: **if** $R = 0$ **then**
6: **return** L
7: **else**
8: **return** $L + K_1 \cdots K_{t'} R$

Correctness is clear: if we write $F = \Phi_{\boldsymbol{K}}(F_1, \ldots, F_t)$, then the previous discussion shows that $L = F$ rem $K_1 \cdots K_{t'}$ and $R = F$ div $K_1 \cdots K_{t'}$, so that the output is indeed F. If we enter Line 8, computing P takes $O^{\sim}(k_1 + \cdots + k_{t'})$ operations $(+, \times)$ in $\mathbb{K}$ [61, Lemma 10.4]; however, in this case R is nonzero, so F has degree at least $k_1 + \cdots + k_{t'}$, and $O^{\sim}(k_1 + \cdots + k_{t'})$ is $O^{\sim}(\deg(F))$. It follows that, excluding the recursive calls, the cost of a single call to Algorithm FromMixedRadix is $O^{\sim}(\deg(F))$ if $\deg(F) \geq k_1 + \cdots + k_{t'}$, and zero otherwise.

There are $O(\log(\deg(F)))$ levels of the recursion tree that will incur a nonzero cost, and the degrees of the polynomials computed at any of these levels add up to at most $\deg(F)$. Hence, the overall cost is $O^{\sim}(\deg(F))$ operations $(+, \times)$ in $\mathbb{K}$.

For the inverse operation, the algorithm is recursive as well. Using the test at Line 3, we avoid doing any computation if F has degree less than $k_1 + \cdots + k_{t'}$. The discussion is as above, yielding a runtime of $O^{\sim}(\deg(F))$ operations $(+, \times)$ in $\mathbb{K}$.

Algorithm 4.2 ToMixedRadix$(F, (K_1, \ldots, K_t))$

Input: F in $\mathbb{K}[x]_{<h}$, $\boldsymbol{K} = (K_1, \ldots, K_t)$ of respective degrees $k_1, \ldots, k_t$, with $h = k_1 + \cdots + k_t$
Output: $(F_1, \ldots, F_t) = \Phi_{\boldsymbol{K}}^{-1}(F)$
1: **if** $t = 1$ **then return** (F)
2: $t' \leftarrow \lceil t/2 \rceil$
3: **if** $\deg(F) < k_1 + \cdots + k_{t'}$ **then**
4: **return** ToMixedRadix$(F, (K_1, \ldots, K_{t'}))$ cat $(0, \ldots, 0)$ ▷ $t - t'$ *zeros*
5: **else**
6: $P \leftarrow K_1 \cdots K_{t'}$
7: $Q, R \leftarrow F$ div P, F rem P
8: **return** ToMixedRadix$(R, (K_1, \ldots, K_{t'}))$ cat ToMixedRadix$(Q, (K_{t'+1}, \ldots, K_t))$

In the next paragraphs, we apply these algorithms to polynomials in $\mathbb{K}[x, y]$ (we use the same names for the algorithms). In this case, we simply proceed coefficient-wise with respect to y, the mixed-radix representation of $F \in \mathbb{K}[x, y]$ being now a two-dimensional array. If the

sum of the degrees of $K_1, \ldots, K_t$ is h, and for F in $\mathbb{K}[x, y]$ supported on an initial segment U, with also $\deg(F, x) < h$, the runtime of both algorithms is $O^{\sim}(|\mathsf{U}|)$.

4.3. The main algorithm

We can now use the results from the previous subsections to give an algorithm for the reduction of a polynomial $f \in \mathbb{K}[x, y]$ modulo a minimal reduced lexicographic Gröbner basis $\mathcal{G} = (g_0, \ldots, g_s)$. For the time being, we only consider the "balanced" case, where f is already reduced modulo g_0 and g_s. Let us write, as usual, the initial terms of $\mathcal{G}$ as

$$\boldsymbol{E} = (y^{n_0}, x^{m_1} y^{n_1}, \ldots, x^{m_{s-1}} y^{n_{s-1}}, x^{m_s})$$

with the m_i's increasing and the n_i's decreasing, and let S be the rectangle $\{0, \ldots, m_s - 1\} \times \{0, \ldots, n_0 - 1\}$. Then, we assume that f is in $\mathbb{K}[x, y]_{\mathsf{S}}$. More general inputs can be handled by performing a reduction by (g_0, g_s) first; this is discussed in the last paragraph of this section.

In what follows, we let T be the initial segment determined by $\mathcal{G}$, and $\delta = \dim_{\mathbb{K}}(\mathbb{K}[x, y]/\mathcal{G})$ be the degree of $\mathcal{G}$.

Overview of the algorithm. Given f in $\mathbb{K}[x, y]$ with $\deg(f, x) < m_s$ and $\deg(f, y) < n_0$, our main algorithm Reduction computes $r = f$ rem $\mathcal{G}$ by calling $s - 1$ times a procedure called PartialReduction, which is described further. The main algorithm returns the remainder r, together with quotients $Q_1, \ldots, Q_{s-1}$, such that $f = Q_1 g_1 + \cdots + Q_s g_s + r$. While we do not need the quotients in this paper, we return them as a byproduct that could possibly be of use in other contexts (the algorithm does not compute the last quotient Q_s, but one can easily deduce it from the other Q_i's and the remainder). Since we assume $\deg(f, y) < n_0$, g_0 does not appear in the reduction equality.

The mixed radix basis is used throughout the algorithm to handle intermediate data; input and output are on the usual monomial basis.

Algorithm 4.3 Reduction$(f, \mathcal{G})$

Input: f in $\mathbb{K}[x, y]$, $\mathcal{G} = (g_0, \ldots, g_s)$ as above
Assumptions: $\deg(f, x) < m_s$, $\deg(f, y) < n_0$
Output: f rem $\mathcal{G}$ and quotients $Q_1, \ldots, Q_{s-1}$

1: $M_0 \leftarrow 1$, $G_0 \leftarrow g_0$
2: **for** $i = 1, \ldots, s$ **do**
3: $\quad M_i \leftarrow$ PolynomialCoefficient$(g_i, y^{n_i}) \in \mathbb{K}[x]$
4: $\quad G_i \leftarrow g_i$ div M_i
5: $\quad D_i \leftarrow M_i$ div M_{i-1}
6: $f^{(0)} \leftarrow$ ToMixedRadix$(f, (D_1, \ldots, D_s))$ ▷ *$f^{(0)}$ is on the mixed radix basis*
7: **for** $i = 1, \ldots, s - 1$ **do**
8: $\quad f^{(i)}, Q_i \leftarrow$ PartialReduction$(f^{(i-1)}, i)$ ▷ *all $f^{(i)}$ are on the mixed radix basis*
9: **return** FromMixedRadix$(f^{(s-1)}, D_1, \ldots, D_s), Q_1, \ldots, Q_{s-1}$

To simplify notation, the polynomials $g_0, \ldots, g_s$, $G_0, \ldots, G_s$, $M_0, \ldots, M_s$ and $D_1, \ldots, D_s$, the latter of which are computed at the beginning of the main algorithm, are assumed to be known in our calls to Algorithm PartialReduction, rather than passed as arguments.

The main result in this section is the following proposition. The runtime given here is softly linear in $n_0 m_s$ and $s\delta$: the former represents the size of the input polynomial f, and the latter is the upper bound on the number of coefficients needed to represent $\mathcal{G}$ discussed in Section 3.1. Whether a better algorithm is possible (which would not need all coefficients of $\mathcal{G}$, but only, for instance, its Gröbner parameters) is not clear to us.

Proposition 4.5. *Given f and $\mathcal{G}$, with $\deg(f, x) < m_s$ and $\deg(f, y) < n_0$, Algorithm* Reduction *returns f rem $\mathcal{G}$ using $O^{\sim}(n_0 m_s + s\delta)$ operations $(+, \times)$ in $\mathbb{K}$.*

Before proving the proposition, we mention an important particular case, where a simplified runtime is available. Suppose that $e_i = 1$ for all i, that is, that all steps in the staircase have height 1. In this case, $n_0 = s$, and since we have $m_s \leq \delta$, we obtain $n_0 m_s \leq s\delta$. In other words, the runtime of the algorithm is simply $O^{\sim}(s\delta)$.

A single reduction step. We start with a description of the key subroutine, Algorithm PartialReduction. We are given $f \in \mathbb{K}[x, y]_{\mathsf{S}}$, and an index i in $\{1, \ldots, s-1\}$; the essential operation is a Euclidean division by g_i with respect to the variable y.

We cannot directly divide by g_i, since this polynomial is not monic in y. However, we know that it factors as $g_i = M_i G_i$, with M_i in $\mathbb{K}[x]$ and G_i monic in y, so our first step is to take the quotient of f by M_i (or more exactly, of a suitable truncation of f with respect to y, as will we know that the higher-degree coefficients will not have to be modified); this is an operation involving division with respect to x only. Then, we do a Euclidean division by G_i with respect to the variable y, keeping coefficients reduced modulo $D_{i+1} \cdots D_{i+\ell_i}$ in $\mathbb{K}[x]$. We can then use the quotient Q obtained this way to reduce the input polynomial f itself by g_i, obtaining a polynomial r.

This gives only a sketch of the algorithm. Its details are complicated by the use of the mixed radix basis, which is however needed for us to control the cost of the operations with respect to x.

We prove below that the output r has the same remainder as f modulo $\mathcal{G}$, but also modulo M_i (both properties will be useful); we also return the quotient Q, which is supported on a translate of R_i. In the analysis of the main algorithm, we will study the support of the polynomials obtained by repeated applications of Algorithm PartialReduction, and show that zeros must progressively appear on $\mathsf{R}_1, \mathsf{R}_2, \ldots$

Lemma 4.6. *Calling* PartialReduction(f, i) *takes*

$$O^{\sim}(n_0(m_{i+\ell_i} - m_i) + m_s(n_{i-h_i} - n_i) + \delta)$$

operations $(+, \times)$ in $\mathbb{K}$, with $h_i = 2^{\mathrm{val}_2(i)}$ and $\ell_i = \min(h_i, s - i)$. The output r, Q satisfies the following properties:

1. $\deg(r, x) < m_s$ *and* $\deg(r, y) < n_0$

Algorithm 4.4 PartialReduction(f, i)

Input: f in $\mathbb{K}[x, y]$, i in $\{1, \ldots, s-1\}$

Assumptions: $\deg(f, x) < m_s$, $\deg(f, y) < n_0$, i in $\{1, \ldots, s-1\}$. f is given on the mixed radix basis associated to $D_1, \ldots, D_s$

Output: r and Q in $\mathbb{K}[x, y]$. r is given on the mixed radix basis associated with $D_1, \ldots, D_s$

1: $h_i \leftarrow 2^{\mathrm{val}_2(i)}$, $\ell_i \leftarrow \min(h_i, s-i)$

2: $F_1 \leftarrow f$ div M_i ▷ *division in the mixed radix basis*
▷ *F_1 is given on the mixed radix basis associated to $D_{i+1}, \ldots, D_s$*

3: $F_2 \leftarrow F_1$ rem $D_{i+1} \cdots D_{i+\ell_i}$ ▷ *division in the mixed radix basis*
▷ *F_2 is given on the mixed radix basis associated to $D_{i+1}, \ldots, D_{i+\ell_i}$*

4: $F_3 \leftarrow F_2$ rem $y^{n_{i-h_i}}$

5: $F_4 \leftarrow$ FromMixedRadix$(F_3, (D_{i+1}, \ldots, D_{i+\ell_i}))$ ▷ *F_4 is on the monomial basis*

6: $q \leftarrow F_4$ div G_i in $\mathbb{A}[y]$ ▷ *G_i such that $g_i = M_i G_i$, $\mathbb{A} = \mathbb{K}[x]/\langle D_{i+1} \cdots D_{i+\ell_i} \rangle$*

7: let Q be the canonical lift of q to $\mathbb{K}[x, y]$ ▷ $\deg(Q, x) < m_{i+\ell_i} - m_i$

8: $V \leftarrow$ Multiply$(Q, \{0, \ldots, m_{i+\ell_i} - m_i - 1\} \times \{0, \ldots, n_{i-h_i} - n_i - 1\}, G_i, \mathsf{T})$
▷ *$V = QG_i$ on the monomial basis*

9: $V_1 \leftarrow V$ rem $(D_{i+1} \cdots D_s)$ ▷ *$V_1 = QG_i$ rem $(D_{i+1} \cdots D_s)$ on the monomial basis*

10: $V_2 \leftarrow$ ToMixedRadix$(V_1, (D_{i+1}, \ldots, D_s))$
▷ *$V_2 = QG_i$ rem $(D_{i+1} \cdots D_s)$, given on the mixed radix basis associated to $D_{i+1}, \ldots, D_s$*

11: $V_3 \leftarrow M_i V_2$ ▷ *multiplication in the mixed radix basis*
▷ *$V_3 = Qg_i$ rem M_s, given on the mixed radix basis associated to $D_1, \ldots, D_s$*

12: $r \leftarrow f - V_3$ ▷ *subtraction in the mixed radix basis*
▷ *$r = (f - Qg_i)$ rem M_s, given on the mixed radix basis associated to $D_1, \ldots, D_s$*

13: **return** r, Q

2. r rem $\mathcal{G} = f$ rem $\mathcal{G}$
3. r rem $M_i = f$ rem M_i
4. r div $y^{n_{i-h_i}} = f$ div $y^{n_{i-h_i}}$
5. $((r \text{ div } M_i) \text{ div } y^{n_i}) \text{ rem } (D_{i+1} \cdots D_{i+\ell_i}, y^{n_{i-h_i}-n_i}) = 0$
6. $\deg(Q, x) < m_{i+\ell_i} - m_i$ *and* $\deg(Q, y) < n_{i-h_i} - n_i$

Proof. We first verify that all steps are well-defined, and discuss degree properties of the polynomials in the algorithm.

As per our discussion in the preamble, the division and remainder at Lines 2 and 3 output a bivariate polynomial F_2 on the mixed radix basis associated to $D_{i+1}, \ldots, D_{i+\ell_i}$. The polynomial F_3 is written on the same basis; F_4 represents the same polynomial, this time on the monomial basis.

That polynomial has y-degree less than n_{i-h_i}; since G_i has y-degree n_i, q, and thus Q, have y-degree less than $n_{i-h_i} - n_i$. Since Q also has x-degree less than $m_{i+\ell_i} - m_i$, it is supported on the rectangle $\{0, \ldots, m_{i+\ell_i} - m_i - 1\} \times \{0, \ldots, n_{i-h_i} - n_i - 1\}$ (which is the translate of R_i to the origin). This proves the last claim in the lemma.

On the other hand, G_i is supported on T (this is true because $i \geq 1$; for $i = 0$, the initial term of G_0, which is y^{n_0}, is not in T), so altogether, the call to Multiply at Line 8 is justified. The variables V_1 and V_2 then represent the same polynomial, namely QG_i rem $(D_{i+1} \cdots D_s)$, on two different bases (resp. monomial and mixed radix). It follows that V_3 represents the polynomial

$$\begin{aligned} M_i(QG_i \text{ rem } (D_{i+1} \cdots D_s)) &= M_i QG_i \text{ rem } (M_i D_{i+1} \cdots D_s) \\ &= Qg_i \text{ rem } M_s. \end{aligned}$$

As we noted in the previous subsection, since V_2 is written on the mixed basis associated to $(D_{i+1}, \ldots, D_s)$, V_3 is written on the mixed basis associated to $(D_1, \ldots, D_s)$. Since this is also the case for f, the subtraction at Line 12 is done coefficient-wise, and results in the polynomial $(f - Qg_i)$ rem M_s, written on the same mixed basis.

This being said, we establish properties 1-5. First item: We have $\deg(f, y) < n_0$. On the other hand, the degree bound on Q implies that Qg_i has y-degree less than n_{i-h_i}. Since $n_{i-h_i} \leq n_0$, the product Qg_i has y-degree less than n_0 as well, and it is then also the case for r. The bound $\deg(r, x) < m_s$ holds by construction.

Second item: we can write $r = f - Qg_i + hM_s = f - Qg_i + hg_s$, for some h in $\mathbb{K}[x, y]$, so that $r - f$ is in the ideal $\langle \mathcal{G} \rangle$.

Third item: consider again the expression $r = f - Qg_i + hM_s$, and notice that M_i divides both g_i and M_s.

Fourth item: because $\deg(f, x) < m_s$, the quotient h in the relation $r = f - Qg_i + hM_s$ is $-Qg_i$ div M_s. Since Qg_i has y-degree less than n_{i-h_i}, it is thus also the case for h. This shows that r div $y^{n_{i-h_i}} = f$ div $y^{n_{i-h_i}}$, as claimed.

Fifth item: since $r = f - Qg_i + hM_s = f - QM_iG_i + hM_s$, we have r div $M_i = F_1 - QG_i + hD_{i+1} \cdots D_s$. By definition, we have $F_1 = F_2 + LD_{i+1} \cdots D_{i+\ell_i}$ and $F_2 = F_3 + Ky^{n_{i-h_i}}$ for some K, L in $\mathbb{K}[x, y]$. F_4 is the same polynomial as F_3, written on a different basis, and

satisfies $F_4 = QG_i + P + L'D_{i+1}\cdots D_{i+\ell_i}$, for some P and L' in $\mathbb{K}[x,y]$, with P of y-degree less than n_i. Altogether, we obtain r div $M_i = P + (L+L')D_{i+1}\cdots D_{i+\ell_i} + hD_{i+1}\cdots D_s + Ky^{n_{i-h_i}}$. As a result,

$$(r \text{ div } M_i) \text{ div } y^{n_i} = ((L+L') \text{ div } y^{n_i})D_{i+1}\cdots D_{i+\ell_i} + (h \text{ div } y^{n_i})D_{i+1}\cdots D_s + Ky^{n_{i-h_i}-n_i}.$$

Because $i + \ell_i \leq s$, this expression taken modulo $(D_{i+1}\cdots D_{i+\ell_i}, y^{n_{i-h_i}-n_i})$ vanishes, as claimed.

It remains to estimate the cost of the algorithm. The divisions with remainders at Lines 2 and 3 are free of cost (because we work in the suitable mixed radix bases); the same holds for Line 4, since it only involves a power of y.

Since $D_{i+1}\cdots D_{i+\ell_i}$ has degree $m_{i+\ell_i} - m_i$, the conversion at Line 5 uses $O^{\tilde{}}(n_{i-h_i}(m_{i+\ell_i} - m_i))$ operations $(+,\times)$ in $\mathbb{K}$, which is $O^{\tilde{}}(n_0(m_{i+\ell_i} - m_i))$.

Prior to the division at Line 6, G_i has to be reduced modulo $D_{i+1}\cdots D_{i+\ell_i}$; proceeding coefficient-wise in y, this takes $O^{\tilde{}}(|\mathsf{T}|) = O^{\tilde{}}(\delta)$ operations $(+,\times)$ in $\mathbb{K}$. Then, the division in $\mathbb{A}[y]$ takes $O^{\tilde{}}(n_{i-h_i})$ operations $(+,\times)$ in $\mathbb{A}$, which is $O^{\tilde{}}(n_{i-h_i}(m_{i+\ell_i} - m_i))$ operations $(+,\times)$ in $\mathbb{K}$. For this expression, it will be enough to use the same upper bound $O^{\tilde{}}(n_0(m_{i+\ell_i} - m_i))$ as above.

Next, we consider the cost of computing the product V in $\mathbb{K}[x,y]$. The input Q has x-degree less than $m_{i+\ell_i} - m_i$ and y-degree less than $n_{i-h_i} - n_i$, whereas G_i is supported on the initial segment T of height n_0, width m_s, and cardinal δ. Hence, using Proposition 2.4 (and the remarks that follow the proposition on the size of the support of QG_i), we see that QG_i can be computed in $O^{\tilde{}}((m_{i+\ell_i} - m_i)(n_{i-h_i} - n_i) + n_0(m_{i+\ell_i} - m_i) + m_s(n_{i-h_i} - n_i) + \delta)$ operations $(+,\times)$ in $\mathbb{K}$. This is also $O^{\tilde{}}(n_0(m_{i+\ell_i} - m_i) + m_s(n_{i-h_i} - n_i) + \delta)$.

The Euclidean division at Line 9 is done on the monomial basis, proceeding coefficient-wise in y. Computing $D_{i+1}\cdots D_s$ takes $O^{\tilde{}}(m_s)$ operations $(+,\times)$ in $\mathbb{K}$. Then, the reduction is done in quasi-linear time in the size of the support of V, that is, $O^{\tilde{}}(n_0(m_{i+\ell_i} - m_i) + m_s(n_{i-h_i} - n_i) + \delta)$ again. Recall that for polynomials supported on an initial segment U, the conversion to the mixed radix basis takes quasi-linear time in the size of U. Here, the support U is contained in the support of $V = QG_i$, so the conversion at Line 10 takes time $O^{\tilde{}}(n_0(m_{i+\ell_i} - m_i) + m_s(n_{i-h_i} - n_i) + \delta)$ once more.

The multiplication by M_i in the mixed radix basis is free, as we simply prepend a vector of zeros to each entry of V_2 to obtain V_3. Finally, the polynomial subtraction at the last step involves one subtraction in $\mathbb{K}$ for each nonzero coefficient of V_3, so $O^{\tilde{}}(n_0(m_{i+\ell_i} - m_i) + m_s(n_{i-h_i} - n_i) + \delta)$ altogether. □

Correctness of the main algorithm. The properties stated above allow us to prove that Algorithm Reduction correctly computes the remainder of f by $\mathcal{G}$.

We define indices $(b_{i,j})_{0\leq i<s, 0\leq j<n_0}$ in $\{1,\ldots,s\}$ as follows (they will be needed to state the recurrence property that proves correctness). For $i = 0,\ldots,s-1$, let $\mathsf{T}_i \subset \mathbb{N}^2$ be the union of the initial segment T and the rectangles $\mathsf{R}_1,\ldots,\mathsf{R}_i$; in particular, $\mathsf{T}_0 = \mathsf{T}$ and T_{s-1} is the rectangle $\{0,\ldots,m_s-1\}\times\{0,\ldots,n_0-1\}$. Then, for $i = 0,\ldots,s-1$ and $j = 0,\ldots,n_0-1$, we let $b_{i,j} \in \{1,\ldots,s\}$ be the smallest index k such that (m_k, j) is not in T_i. In particular,

$b_{s-1,j} = s$ for all $j < n_0$. On the other hand, for $i = 0$, we see that any pair (u, j) with $u < m_{b_{0,j}}$ is in T, so $x^u y^j$ is reduced modulo $\mathcal{G}$.

Let $f^{(0)}, \ldots, f^{(s-1)}$ be the polynomials computed throughout the algorithm (the first item of Lemma 4.6 proves that these polynomials are well-defined, and all supported on the rectangle $\{0, \ldots, m_s - 1\} \times \{0, \ldots, n_0 - 1\}$). We prove the following claim, written $A(i)$ in the sequel, by induction on $i = 0, \ldots, s-1$: for $n_i \leq j < n_0$, the polynomial PolynomialCoefficient$(f^{(i)}, y^j)$ rem $M_{b_{i,j}} \in \mathbb{K}[x]$ has degree less than $m_{b_{0,j}}$. For $i = 0$, there is nothing to prove (since no index j needs to be considered). Suppose that $A(i-1)$ holds, for some i in $\{1, \ldots, s-1\}$; we prove $A(i)$.

For $j \geq n_{i-h_i}$, Item 4 of Lemma 4.6 shows that PolynomialCoefficient$(f^{(i)}, y^j) =$ PolynomialCoefficient$(f^{(i-1)}, y^j)$. In that case, though, we also have $b_{i,j} = b_{i-1,j}$: indeed, these two quantities are the smallest indices k such that (m_k, j) is not in T_i, resp. T_{i-1}, and the definition of R_i shows that in the area of points with ordinate at least equal to n_{i-h_i}, T_i and T_{i-1} coincide. Thus, in this case, our induction property holds.

Now, suppose that j is in $\{n_i, \ldots, n_{i-h_i} - 1\}$; in this case, by Items 3 and 5 of the same lemma, PolynomialCoefficient$(f^{(i)}, y^j)$ rem $M_{i+\ell_i}$ is equal to PolynomialCoefficient $(f^{(i-1)}, y^j)$ rem M_i. On the other hand, we claim that we also have $b_{i-1,j} = i$ and $b_{i,j} = i + \ell_i$; for this discussion, it will be convenient to reuse the notation $\mathsf{R}_0 = \mathsf{T}$ from Proposition 4.3.

- First claim. By definition, $b_{i-1,j}$ is the smallest index k such that (m_k, j) is not in $\mathsf{T}_{i-1} = \mathsf{R}_0 \cup \mathsf{R}_1 \cup \cdots \cup \mathsf{R}_{i-1}$. We know that (m_i, j) is in R_i, and thus in $\mathsf{R}_0 \cup \cdots \cup \mathsf{R}_i$. Now, (m_{i-1}, j) is also in one of $\mathsf{R}_0, \mathsf{R}_1, \ldots, \mathsf{R}_{s-1}$ (Proposition 4.2) but since $m_{i-1} < m_i$, (m_{i-1}, j) is more precisely in $\mathsf{T}_i = \mathsf{R}_0 \cup \cdots \cup \mathsf{R}_i$. On the other hand, it is not in R_i (because $m_{i-1} < m_i$, and using Proposition 4.3), so it is in $\mathsf{T}_{i-1} = \mathsf{R}_0 \cup \cdots \cup \mathsf{R}_{i-1}$. In other words, we have $i - 1 < b_{i-1,j}$. However, (m_i, j) being in R_i, it does not lie in $\mathsf{T}_{i-1} = \mathsf{R}_0 \cup \cdots \cup \mathsf{R}_{i-1}$ (since the R_k's are pairwise disjoint). Altogether, this shows $b_{i-1,j} = i$.

- Second claim. Again, we first spell out the definition: $b_{i,j}$ is the smallest index k such that (m_k, j) is not in $\mathsf{T}_i = \mathsf{R}_0 \cup \mathsf{R}_1 \cup \cdots \cup \mathsf{R}_i$. For $k = i + \ell_i$, we know that by construction, $(m_{i+\ell_i}, j)$ lies on the right of $\mathsf{R}_i = \{m_i, \ldots, m_{i+\ell_i} - 1\} \times \{n_i, \ldots, n_{i-h_i} - 1\}$ (recall that j is in the range $\{n_i, \ldots, n_{i-h_i} - 1\}$, and thus is not in T_i, by Proposition 4.3. On the other hand, for $k = i + \ell_i - 1$, we start by observing that $(m_{i+\ell_i} - 1, j)$ is by construction in R_i, and thus in T_i. Since $i + \ell_i - 1 < i + \ell_i$, we know that $m_{i+\ell_i-1} < m_{i+\ell_i}$, so that $m_{i+\ell_i-1} \leq m_{i+\ell_i} - 1$. It follows again from Proposition 4.3 that $(m_{i+\ell_i} - 1, j)$ is also in T_i. Altogether, this shows $b_{i,j} = i + \ell_i$.

As a result, the left-hand side above is the term PolynomialCoefficient$(f^{(i)}, y^j)$ rem $M_{b_{i,j}}$ that appears in our claim. Thus, to conclude the induction proof, it is enough to show that PolynomialCoefficient$(f^{(i-1)}, y^j)$ rem M_i has degree less than $m_{b_{0,j}}$. We do this using a further case discussion:

- if $j \geq n_{i-1}$, we can use the induction assumption. It implies that the remainder PolynomialCoefficient$(f^{(i-1)}, y^j)$ rem $M_{b_{i-1,j}}$ has degree less than $m_{b_{0,j}}$. Since we saw that have $b_{i-1,j} = i$, we are done.

- if $j < n_{i-1}$, we have $b_{0,j} = i$, so that $m_{b_{0,j}} = m_i = \deg(M_i)$, and our claim holds as well.

Having established our induction claim, we can take $i = s-1$. Then, $A(s-1)$ shows that for j in $n_{s-1}, \ldots, n_0 - 1$, PolynomialCoefficient$(f^{(s-1)}, y^j)$ rem M_s has degree less than $m_{b_{0,j}}$. By construction, $f^{(s-1)}$ is reduced modulo M_s, so that PolynomialCoefficient$(f^{(s-1)}, y^j)$ itself has degree less than $m_{b_{0,j}}$. Now, for j in $0, \ldots, n_{s-1} - 1$, we have $b_{0,j} = s$, so PolynomialCoefficient$(f^{(s-1)}, y^j)$ has degree less than $m_{b_{0,j}}$ as well in this case. Altogether, as we pointed out when we introduced $m_{b_{0,j}}$, this proves that $f^{(s-1)}$ is reduced modulo $\mathcal{G}$.

The second item of Lemma 4.6 finally shows that f rem $\mathcal{G} = f^{(s-1)}$ rem $\mathcal{G}$, so $f^{(s-1)}$ is indeed the normal form of f modulo $\mathcal{G}$. This finishes the correctness proof.

Cost analysis. For the cost analysis, we start with the computation of polynomials M_i, G_i and D_i, at the beginning of the main algorithm. Since divisions by a monic univariate polynomial take softly linear time, each pass in the loop at Line 2 of Reduction takes $O\tilde{}(\delta)$ operations, for a total of $O\tilde{}(s\delta)$.

The conversions to and from the mixed radix basis take quasi-linear time in the size of the support of f, that is, $O\tilde{}(n_0 m_s)$ operations. Then, it suffices to add the costs of the calls to PartialReduction. By Lemma 4.6, deducing $f^{(i)}$ from $f^{(i-1)}$ takes $O\tilde{}(n_0(m_{i+\ell_i} - m_i) + m_s(n_{i-h_i} - n_i) + \delta)$ operations in $\mathbb{K}$, with $\delta = |\mathsf{T}|$, so it suffices to sum this quantity for $i = 1$ to $s-1$. The first two terms add up to a total of $O\tilde{}(n_0 \sum_{i=1}^{s-1}(m_{i+\ell_i} - m_i) + m_s \sum_{i=1}^{s-1}(n_{i-h_i} - n_i))$. Proposition 4.4 shows that this sum is in $O\tilde{}(n_0 m_s)$, so taking into account the term $O\tilde{}(\delta)$ in each summand, the total is $O\tilde{}(n_0 m_s + s\delta)$, as claimed.

Generalization to arbitrary inputs and discussion. If the input f does not satisfy the conditions $\deg(f, x) < m_s$ and $\deg(f, y) < n_0$, we fall back to this case by reduction modulo the pair of polynomials (g_0, g_s), which have respective initial terms y^{n_0} and x^{m_s}. The following algorithm achieves this; we discuss possible improvements below.

Algorithm 4.5 ReductionGeneralInput$(f, \mathcal{G})$

Input: f in $\mathbb{K}[x, y]$, $\mathcal{G} = (g_0, \ldots, g_s)$
Output: f rem $\mathcal{G}$

1: $f_1 \leftarrow f$ rem g_s
2: $f_2 \leftarrow f_1$ rem g_0 in $\mathbb{A}[y]$ ▷ $\mathbb{A} = \mathbb{K}[x]/\langle g_s \rangle$
3: let f_3 be the canonical lift of f_2 to $\mathbb{K}[x, y]$ ▷ $\deg(f_3, x) < m_s$
4: **return** Reduction$(f_3, \mathcal{G})$

Proposition 4.7. *Given f and $\mathcal{G}$, with $\deg(f, x) < d$ and $\deg(f, y) < e$, Algorithm* ReductionGeneralInput *returns f rem $\mathcal{G}$ using $O\tilde{}(ed + em_s + n_0 m_s + s\delta)$ operations $(+, \times)$ in $\mathbb{K}$. If $\mathcal{G}$ generates an $\langle x, y \rangle$-primary ideal, the runtime becomes $O\tilde{}(\delta m_s)$ operations $(+, \times)$ in $\mathbb{K}$.*

Proof. Reducing f modulo g_s takes $O^{\sim}(ed)$ operations (and is actually free if $d < m_s$). Then, Euclidean division by g_0 in $\mathbb{A}[y]$ uses $O^{\sim}(e)$ steps in $\mathbb{A}$, which is $O^{\sim}(em_s)$ steps in $\mathbb{K}$. Finally, Proposition 4.5 gives a cost of $O^{\sim}(n_0 m_s + s\delta)$ for the last step.

If $\mathcal{G}$ generates an $\langle x, y\rangle$-primary ideal, all terms of y-degree at least δ vanish through the reduction (so we can replace e by δ), as do all terms of x-degree at least m_s (so we can replace d by m_s). □

In the runtime for the general case, ed is the size of the support of input f, and $s\delta$ our bound on the size of $\mathcal{G}$, so they are essentially unavoidable (unless of course one could avoid using $\mathcal{G}$ itself but only its Gröbner parameters). The runtime also features the extra terms em_s and $n_0 m_s$, but getting rid of them and improving the runtime to $O^{\sim}(ed + s\delta)$ unconditionally seems to be very challenging.

Indeed, consider the *modular composition* problem: given F, G, H in $\mathbb{K}[x]$, with F monic of degree n and G, H of degrees less than n, this amounts to computing $G(H)$ rem F. A direct approach takes quadratic time, and Brent-Kung's baby-steps / giant-steps algorithm uses $O(n^{1.69})$ operations (and relies on fast matrix arithmetic). Bringing this down to a quasi-linear runtime has been an open question since 1978: it is so far known to be feasible only over finite $\mathbb{K}$ [34], with the best algorithm for an arbitrary $\mathbb{K}$ to date featuring a Las Vegas cost of $O(n^{1.43})$ [48].

It turns out that modular composition is a particular case of the reduction problem we are considering here. With F, G, H as above, if we consider $\mathcal{G} = (y - H(x), F(x))$ and the polynomial $f = G(y)$, then the remainder f rem $\mathcal{G}$ is precisely $G(H)$ rem F. Here, we have $n_0 = 1$, $s = 1$, $m_s = n$, $\delta = n$, $d = 1$ and $e = \deg(G, y) + 1$, so that in general $e = n$; on such input, the runtime of our algorithm is $O^{\sim}(n^2)$. Improving our result to $O^{\sim}(ed + s\delta)$ would give a softly linear modular composition algorithm, thus solving a long-standing open question.

On the other hand, the case where f has large degree in both x and y, *i.e.* when $m_s \leq d$ and $n_0 \leq e$, is particularly favourable, since then the runtime does become $O^{\sim}(ed + s\delta)$. Another favourable situation is when all e_i's are equal to 1, since we said before that we have $n_0 m_s \leq s\delta$ in this case, with thus a runtime of $O^{\sim}(ed + em_s + s\delta)$ – this is for instance the case if we apply a generic change of coordinates, as the initial ideal is then *Borel-fixed* (see the discussions in [15, Chapter 15.9] for the case of homogeneous polynomials, and in [57] for our situation).

Finally, we point out an application of Proposition 4.7 to modular multiplication: given A, B in $\mathbb{K}[x, y]_{\mathsf{T}}$, where T is the initial segment determined by $\mathcal{G}$, compute $f = AB$ rem $\mathcal{G} \in \mathbb{K}[x, y]_{\mathsf{T}}$. In this case, we have $d < 2m_s$ and $e < 2n_0$, so the runtime is $O^{\sim}(n_0 m_s + s\delta)$; when all e_i's are equal to 1, this becomes $O^{\sim}(s\delta)$. We are not aware of previous results for this question.

5. From Gröbner parameters to Gröbner basis

In this section, we fix a given Gröbner cell (or equivalently, the monomials $\boldsymbol{E}$). We show how to make explicit the mapping $\Phi_{\boldsymbol{E}} : \mathbb{K}^N \to \mathcal{C}(\boldsymbol{E})$, which takes as input Gröbner parameters and outputs the corresponding reduced Gröbner basis (see Section 3.2).

First, we fix a way to index the N coefficients of the polynomials $(\sigma_{j,i})_{0\le i\le s-1, i\le j\le s}$ that appear in the syzygy (3); this will be done in the mutually inverse routines given below. Here, for simplicity, we assume that given the monomials $\boldsymbol{E}$, we can directly access the integers s, $(d_i)_{1\le i\le s}$ and $(e_i)_{1\le i\le s}$.

Algorithm 5.1 SigmaFromParameters$(\boldsymbol{E}, (\lambda_1, \ldots, \lambda_N))$

Input: monomials $\boldsymbol{E}$, $(\lambda_1, \ldots, \lambda_N)$ in $\mathbb{K}^N$
Output: polynomials $(\sigma_{j,i})_{0\le i\le s-1, i\le j\le s}$ in $\mathbb{K}[x, y]$

$k \leftarrow 1$
for $i = 0, \ldots, s-1$ **do**
  $\sigma_{i,i} \leftarrow \sum_{0\le \ell < d_{i+1}} \lambda_{k+\ell} x^\ell$
  $k \leftarrow k + d_{i+1}$
  for $j = i+1, \ldots, s$ **do**
    $\sigma_{j,i} \leftarrow 0$
    for $m = 0, \ldots, e_{j-1}$ **do**
      $\sigma_{j,i} \leftarrow \sigma_{j,i} + \sum_{0\le \ell < d_{i+1}} \lambda_{k+\ell} x^\ell y^m$
      $k \leftarrow k + d_{i+1}$
return $(\sigma_{j,i})_{0\le i\le s-1, i\le j\le s}$

Algorithm 5.2 ParametersFromSigma$(\boldsymbol{E}, (\sigma_{j,i})_{i,j})$

Input: monomials $\boldsymbol{E}$, polynomials $(\sigma_{j,i})_{i,j}$ in $\mathbb{K}[x, y]$
Output: $(\lambda_1, \ldots, \lambda_N)$ in $\mathbb{K}^N$

$k \leftarrow 1$
for $i = 0, \ldots, s-1$ **do**
  for $\ell = 0, \ldots, d_{i+1} - 1$ **do** $\lambda_{k+\ell} \leftarrow$ Coefficient$(\sigma_{i,i}, x^\ell)$
  $k \leftarrow k + d_{i+1}$
  for $j = i+1, \ldots, s$ **do**
    for $m = 0, \ldots, e_{j-1}$ **do**
      for $\ell = 0, \ldots, d_{i+1} - 1$ **do** $\lambda_{k+\ell} \leftarrow$ Coefficient$(\sigma_{j,i}, x^\ell y^m)$
      $k \leftarrow k + d_{i+1}$
return $(\lambda_1, \ldots, \lambda_N)$

To deal with the particular case of punctual Gröbner parameters, a few obvious modifications are needed, such as setting $\sigma_{0,0}, \ldots, \sigma_{s-1,s-1}$ to zero and ensuring that x divides $\sigma_{1,0}, \ldots, \sigma_{s,s-1}$ in SigmaFromParameters. We call SigmaFromPunctualParameters and PunctualParametersFromSigma the resulting procedures.

We can now give an algorithm called ReducedBasisFromParameters, which describes the mapping $\Phi_{\boldsymbol{E}} : \mathbb{K}^N \to \mathcal{C}(\boldsymbol{E})$. This procedure is rather straightforward; the algorithm for the inverse operation, called ParametersFromReducedBasis, is slightly more involved, and is described in the next section. We still use the notation of Section 3.2, writing in

particular $M_i \in \mathbb{K}[x]$ for the polynomial coefficient of y^{n_i} in both g_i and h_i, for all i, and m_i for its degree.

We compute the h_i's, and then the g_i's, in descending order. To obtain the former, we simply use Eq. (3). For any $i = s-1, \ldots, 0$, assuming we know h_i and $g_{i+1}, \ldots, g_s$, let us show how to obtain g_i by reducing h_i (for $i = s$, we have $g_s = h_s$), using procedure Reduction from the previous section.

Using Euclidean division with respect to x, the polynomial h_i can be written as $h_i = A_i M_{i+1} + B_i$, with A_i and B_i in $\mathbb{K}[x, y]$ and $\deg(B_i, x) < m_{i+1}$.

Recall now that all polynomials $g_{i+1}, \ldots, g_s$ are multiples of M_{i+1}, and that the family $\mathcal{G}_i = (g_{i+1}/M_{i+1}, \ldots, g_s/M_{i+1})$ is a zero-dimensional Gröbner basis (as pointed out after Eq. (2)). Set $\bar{h}_i = (A_i \text{ rem } \mathcal{G}_i)M_{i+1} + B_i$; we claim that $\bar{h}_i = g_i$. First, we determine its initial term: all monomials in A_i rem $\mathcal{G}_i$, and thus in $(A_i \text{ rem } \mathcal{G}_i)M_{i+1}$, have y-degree less than n_{i+1}, whereas B_i contains the initial term $x^{m_i}y^{n_i}$ of h_i. Thus the initial term of $\bar{h}_i$ is still $x^{m_i}y^{n_i}$. Next, we verify that $\bar{h}_i$ is reduced modulo $g_1, \ldots, g_{i-1}, g_{i+1}, \ldots, g_s$.

- None of $g_1, \ldots, g_{i-1}$ can reduce any term in $\bar{h}_i$, since this polynomial has y-degree n_i.
- Since A_i rem $\mathcal{G}_i$ is reduced modulo $\mathcal{G}_i$, $(A_i \text{ rem } \mathcal{G}_i)M_{i+1}$ is reduced modulo $g_{i+1}, \ldots, g_s$.
- Since B_i has x-degree less than m_{i+1}, it is also reduced modulo $g_{i+1}, \ldots, g_s$.

The last observation is that the difference $\bar{h}_i - h_i$ is in the ideal $\langle g_{i+1}, \ldots, g_s \rangle$. Altogether, this establishes $\bar{h}_i = g_i$.

Example 5.1. *We revisit the polynomials of Example 3.4, with*

$$\begin{aligned} h_0 &= y^4 + \lambda_5 xy^3 + \lambda_1 xy^2 + (\lambda_1\lambda_5 + \lambda_4)x^2y + \lambda_2 xy + \lambda_3 x^3 + \lambda_2\lambda_5 x^2 \\ h_1 &= xy^3 + \lambda_5 x^2y^2 + \lambda_4 x^3 \\ h_2 &= x^2y + \lambda_5 x^3 \\ h_3 &= x^4, \end{aligned}$$

doing computations over the field $\mathbb{Q}(\lambda_1, \ldots, \lambda_5)$. We have $g_2 = h_2$ and $g_3 = h_3$; let us then take $i = 1$ and compute g_1. Starting from $h_1 = xy^3 + \lambda_5 x^2y^2 + \lambda_4 x^3$ and $M_2 = x^2$, we write $h_1 = A_1 M_2 + B_1$ with

$$A_1 = \lambda_5 y^2 + \lambda_4 x \quad \text{and} \quad B_1 = xy^3.$$

We reduce A_1 modulo $(g_2/M_2, g_3/M_2) = (y + \lambda_5 x, x^2)$, and obtain the remainder $\lambda_4 x$. Finally, this gives

$$g_1 = \lambda_4 x M_2 + B_1 = xy^3 + \lambda_4 x^3,$$

as already seen in Example 3.4.

We can now give our algorithm to compute $g_0, \ldots, g_s$. For the reduction of the bivariate polynomial A_i modulo $\mathcal{G}_i$, we use our procedure Reduction. Note that the degree assumptions for that procedure are satisfied: the polynomial A_i has x-degree less than $m_s - m_{i+1}$

and y-degree less than n_{i+1}, which are precisely the maximal x-degrees and y-degrees of the elements in $\mathcal{G}_i$.

As before, we assume that given $\boldsymbol{E}$, we can directly access the integers s, $(d_i)_{1\le i\le s}$ and $(e_i)_{1\le i\le s}$ and use them freely in the pseudo-code.

Algorithm 5.3 ReducedBasisFromParameters($\boldsymbol{E}, (\lambda_1, \dots, \lambda_N)$)

Input: monomials $\boldsymbol{E}$, $(\lambda_1, \dots, \lambda_N)$ in $\mathbb{K}^N$
Output: the minimal reduced Gröbner basis of $\Phi_{\boldsymbol{E}}(\lambda_1, \dots, \lambda_N)$

1: $(\sigma_{j,i})_{i,j} \leftarrow$ SigmaFromParameters($\boldsymbol{E}, (\lambda_1, \dots, \lambda_N)$)
2: $M_0 \leftarrow 1$
3: **for** $i = 1, \dots, s$ **do** $M_i \leftarrow (x^{d_i} - \sigma_{i-1,i-1})M_{i-1}$
4: $h_s \leftarrow M_s$; $g_s \leftarrow M_s$
5: **for** $i = 0, \dots, s-1$ **do**
6: $\quad T_i \leftarrow$ KroneckerMultiply$(y^{e_{i+1}}, h_{i+1}) + \cdots +$ KroneckerMultiply$(\sigma_{s,i}, h_s)$
7: $\quad h_i \leftarrow T_i$ div $(x^{d_{i+1}} - \sigma_{i,i})$
8: $\quad \mathcal{G}_i \leftarrow (g_{i+1}$ div $M_{i+1}, \dots, g_s$ div $M_{i+1})$
9: $\quad A_i, B_i \leftarrow h_i$ div M_{i+1}, h_i rem M_{i+1}
10: $\quad \bar{A}_i \leftarrow$ Reduction$(A_i, \mathcal{G}_i)$
11: $\quad g_i \leftarrow \bar{A}_i M_{i+1} + B_i$
12: **return** $(g_0, \dots, g_s)$

Proposition 5.2. *Given monomials $\boldsymbol{E}$ and $(\lambda_1, \dots, \lambda_N)$ in $\mathbb{K}$,* ReducedBasisFromParameters$(\boldsymbol{E}, (\lambda_1, \dots, \lambda_N))$ *returns the reduced Gröbner basis of $\Phi_{\boldsymbol{E}}(\lambda_1, \dots, \lambda_N)$ using $O^{\sim}(s^2 n_0 m_s)$ operations $(+, \times)$ in $\mathbb{K}$.*

Proof. Correctness follows from the previous discussion. Regarding the runtime, the first step does no arithmetic operation, and computing each polynomial M_i takes $O^{\sim}(\delta)$ operations, for a total of $O^{\sim}(s\delta)$.

For a given index i, computing T_i involves at most s polynomial multiplications, each of which uses $O^{\sim}(n_0 m_s)$ operations $(+, \times)$ in $\mathbb{K}$; we can deduce h_i in the same asymptotic time. The Euclidean divisions needed to compute $\mathcal{G}_i$ cost $O^{\sim}(s\delta)$ operations (since all polynomials in $\mathcal{G}$ are supported on an initial segment of size δ), and the one for A_i and B_i costs $O^{\sim}(n_0 m_s)$, for the same reason. Proposition 4.5 shows that we compute $\bar{A}_i$ in $O^{\sim}(n_0 m_s + s\delta)$ operations $(+, \times)$. Finally, the product and sum giving g_i take $O^{\sim}(n_0 m_s)$ operations $(+, \times)$ as well.

Altogether, the cost at step i is $O^{\sim}(sn_0 m_s + s\delta)$, which is $O^{\sim}(sn_0 m_s)$, and the overall runtime estimate follows. □

Unfortunately, this bound is not linear in the output size: each polynomial in the output has $O(n_0 m_s)$ coefficients, so $O(sn_0 m_s)$ coefficients are sufficient to represent the output, to be compared with our $O^{\sim}(s^2 n_0 m_s)$ runtime.

On another hand, it will be useful to note that the algorithm does not perform divisions, so if the input parameters lie in a ring $\mathbb{A} \subset \mathbb{K}$, the output polynomials $\mathcal{G}$ all have coefficients in $\mathbb{A}$.

The whole procedure can be adapted to deal with punctual Gröbner cells in a straightforward manner, by using SigmaFromPunctualParameters at Line 1. The resulting function is called ReducedBasisFromPunctualParameters, and features a similar runtime.

6. Computing the Gröbner parameters

We can now give our algorithms to compute the Gröbner parameters of a zero-dimensional ideal I.

We do this in two different contexts. The first situation is the recovery of these parameters starting from the reduced Gröbner basis of I (*i.e.*, computing the map $\Phi_{\boldsymbol{E}}^{-1}$ defined in the previous sections). This operation, based on a sequence of Euclidean divisions, is aimed to be used on a Gröbner basis in a given base field, for example over $\mathbb{A}/\mathfrak{m}$ one may use it to find initial parameters prior to lifting.

The second variant we present is the core ingredient of our main algorithm. We describe a system of polynomials which admits the Gröbner parameters of any zero-dimensional ideal I contained in $\langle f_1, \ldots, f_t\rangle$, for some bivariate polynomials f_i, as a solution with multiplicity one. In a nutshell, we obtain these equations by defining a parametric Gröbner basis (whose coefficients are polynomials in the Gröbner parameters, similar to those given in Example 3.4 in the punctual case), and reducing the polynomials f_i modulo this basis. The coefficients of the remainders are also polynomials in the Gröbner parameters: these are our equations.

In this, we follow previous work of Hauenstein, Mourrain, Szanto [25] that was in the context of border bases representations (for which there is no notion of monomial order, and reductions are done using repeated applications of multiplication matrices; the entries of these matrices are then suitably parameterized, rather than the coefficients of a Gröbner basis).

The equations we derive are in general too complex to be dealt with directly. In the next section, we will use them to describe our main algorithm, a version of Newton iteration to compute the Gröbner parameters of a zero-dimensional ideal I.

6.1. Starting from a reduced basis

In this subsection, we assume that we are given the reduced Gröbner basis $\mathcal{G} = (g_0, \ldots, g_s)$ of a zero-dimensional ideal I, and we show how to compute its Gröbner parameters. We also indicate how the procedure simplifies slightly when I is $\langle x, y\rangle$-primary.

Our notation is as before: the initial terms of the polynomials $(g_0, \ldots, g_s)$ are written $\boldsymbol{E} = (y^{n_0}, x^{m_1}y^{n_1}, \ldots, x^{m_{s-1}}y^{n_{s-1}}, x^{m_s})$, the degree of $\langle\mathcal{G}\rangle$ is δ and $N = \delta + m_s$ is the number of Gröbner parameters. In what follows, we compute the polynomials $(\sigma_{j,i})_{i,j}$ appearing in the syzygies (3), whose coefficients are the Gröbner parameters of I. Recall that we write $D_i = x^{d_i} - \sigma_{i-1,i-1}$ for $i = 1, \ldots, s$, and $M_i = (x^{d_1} - \sigma_{0,0}) \cdots (x^{d_i} - \sigma_{i-1,i-1})$ for $i = 0, \ldots, s$, with the empty product being equal to 1.

Deriving the algorithm. Knowing the reduced Gröbner basis $\mathcal{G} = (g_0, \ldots, g_s)$, some of the polynomials $(\sigma_{j,i})$ are easy to compute: for $i = 1, \ldots, s$, we saw in the previous section that the polynomial coefficient of y^{n_i} in g_i is none other than M_i. Knowing $M_1, \ldots, M_s$ gives us $D_1, \ldots, D_s$, and thus $\sigma_{0,0}, \ldots, \sigma_{s-1,s-1}$, by successive divisions.

Let now $h_0, \ldots, h_s$ be the non-reduced Gröbner basis already used previously, that satisfies Eq. (3), and recall that for $i = 0, \ldots, s$, M_i divides h_i. We define $H_i = h_i/M_i$, and consider again Eq. (4), which is a rewriting of (3):

$$D_{i+1}h_i - y^{e_{i+1}}h_{i+1} = \sum_{j=i+1}^{s} \sigma_{j,i}h_j,$$

in which both left- and right-hand sides can be divided by M_{i+1}. Carrying out the division, we obtain

$$H_i - y^{e_{i+1}}H_{i+1} = \sum_{j=i+1}^{s} \sigma_{j,i}D_{i+2}\cdots D_jH_j. \tag{6}$$

Fix i in $\{0, \ldots, s-1\}$, and assume that we have computed $H_{i+1}, \ldots, H_s$; we show how to compute $\sigma_{i+1,i}, \ldots, \sigma_{s,i}$, and then H_i.

By construction, the polynomials $(g_0, \ldots, g_i, h_{i+1}, \ldots, h_s)$ also form a minimal Gröbner basis of I. The polynomial $h_i - g_i$ is in I, so it reduces to zero through division by these polynomials. Since g_i and h_i both have M_i as polynomial coefficient of y^{n_i}, $h_i - g_i$ has degree less than n_i in y. This implies that the only polynomials in the list that can reduce it are $h_{i+1}, \ldots, h_s$. We reduce $h_i - g_i$ by h_{i+1}, then h_{i+2}, etc, in this order; for $j = i, \ldots, s$, write $R_{i,j}$ for the remainder obtained after reduction by $h_{i+1}, \ldots, h_j$, so that $R_{i,i} = h_i - g_i$.

Lemma 6.1. *For $j = i, \ldots, s$, $R_{i,j}$ has y-degree less than n_j.*

Proof. We pointed out that this is true for $j = i$, so we suppose that the claim holds for some index $j < s$ and prove it for index $j + 1$. To obtain $R_{i,j+1}$, we reduce $R_{i,j}$ by h_{j+1}, which has initial term $x^{m_{j+1}}y^{n_{j+1}}$, so that we can write $R_{i,j+1} = A_{j+1} + B_{j+1}$, with $\deg(B_{j+1}, y) < n_{j+1}$, $\deg(A_{j+1}, x) < m_{j+1}$ and all terms in A_{j+1} having y-degree at least n_{j+1}. To conclude, we prove that $A_{j+1} = 0$.

Since we use the lexicographic order $y \succ x$, reduction of a term by h_{j+1} does not increase its y-degree; since $R_{i,j}$ had y-degree less than n_j by assumption, it is also the case for A_{j+1}. In particular, A_{j+1} is reduced modulo $\mathcal{H}$, where $\mathcal{H} = (h_0, \ldots, h_s)$. Since $R_{i,j}$ reduces to zero modulo $\mathcal{H}$, it follows that $A_{j+1} + (B_{j+1} \text{ rem } \mathcal{H}) = 0$. Now, for the same reason as above, $(B_{j+1} \text{ rem } \mathcal{H})$ has y-degree less than n_{j+1}, so that the supports of A_{j+1} and $(B_{j+1} \text{ rem } \mathcal{H})$ do not overlap. This implies that $A_{j+1} = (B_{j+1} \text{ rem } \mathcal{H}) = 0$, as claimed. □

This lemma shows that the reduction of $h_i - g_i$ induces an equality of the form

$$h_i - g_i = \sum_{j=i+1}^{s} q_{j,i}h_j,$$

for some polynomials $q_{j,i}$ in $\mathbb{K}[x,y]$ satisfying $\deg(q_{j,i},y) < n_{j-1} - n_j = e_j$ for all j. Equivalently, we may rewrite this as

$$h_i = g_i + \sum_{j=i+1}^{s} q_{j,i} M_j H_j,$$

whence, after dividing by M_i,

$$H_i = G_i + \sum_{j=i+1}^{s} q_{j,i} D_{i+1} \cdots D_j H_j. \tag{7}$$

Combining (6) and (7), we get

$$G_i - y^{e_{i+1}} H_{i+1} = \sum_{j=i+1}^{s} Q_{j,i} H_j, \quad \text{with} \quad Q_{j,i} = (\sigma_{j,i} - q_{j,i} D_{i+1}) D_{i+2} \cdots D_j. \tag{8}$$

Notice in particular that for all j, we have $\deg(Q_{j,i},y) < e_j$ and thus $\deg(Q_{j,i}H_j,y) < n_{j-1}$.

In this paragraph, for F in $\mathbb{K}[x,y]$, we write $\bar{F}$ for its residue class in $\mathbb{B}[y]$, with $\mathbb{B} = \mathbb{K}[x]/\langle D_{i+1} \cdots D_s \rangle$. Take j in $i+1, \ldots, s-1$ and suppose that we know $\bar{Q}_{i+1,i}, \ldots, \bar{Q}_{j-1,i}$. Split the sum in (8) as $A = Q_{j,i}H_j + R$ with

$$A = G_i - y^{e_{i+1}} H_{i+1} - \sum_{k=i+1}^{j-1} Q_{k,i} H_k \quad \text{and} \quad R = \sum_{k=j+1}^{s} Q_{k,i} H_k.$$

Over $\mathbb{B}[y]$, $\bar{R}$ has degree (in y) less than n_j; since $\bar{H}_j$ is monic of degree n_j, the relation $\bar{A} = \bar{Q}_{j,i}\bar{H}_j + \bar{R}$ describes the Euclidean division of $\bar{A}$, which is known, by $\bar{H}_j$, which is known as well. If we let $Q^*_{i,j}$ be the canonical lift of $\bar{Q}_{i,j}$ to $\mathbb{K}[x,y]$, we obtain

$$\begin{aligned} Q^*_{j,i} &= Q_{j,i} \text{ rem } D_{i+1} \cdots D_s \\ &= (\sigma_{j,i} - q_{j,i} D_{i+1}) D_{i+2} \cdots D_j \text{ rem } D_{i+1} \cdots D_s. \end{aligned}$$

It follows that $Q^*_{i,j}$ is divisible by $D_{i+2} \cdots D_j$, and that

$$Q^*_{i,j} \text{ div } (D_{i+2} \cdots D_j) = (\sigma_{j,i} - q_{j,i} D_{i+1}) \text{ rem } D_{i+1} D_{j+1} \cdots D_s. \tag{9}$$

Since $\deg(\sigma_{j,i},x) < d_{i+1}$, reducing this modulo D_{i+1} finally gives us $\sigma_{j,i}$. Noticing also that the remainder $\bar{R}$ gives us the next value of $\bar{A}$, we obtain Algorithm ParametersFromReducedBasis.

In the following proposition, in preparation for the discussion in the next subsection, we point out in particular that the algorithm does not perform any division.

Proposition 6.2. *Given a minimal reduced Gröbner basis $\mathcal{G} = (g_0, \ldots, g_s)$ in $\mathbb{K}[x,y]$, ParametersFromReducedBasis($\mathcal{G}$) returns the Gröbner parameters of $\mathcal{G}$ using $O^{\tilde{}}(s^2 n_0 m_s)$ operations $(+,\times)$ in $\mathbb{K}$.*

Algorithm 6.1 ParametersFromReducedBasis($\mathcal{G}$)

Input: $\mathcal{G} = (g_0, \dots, g_s)$ in $\mathbb{K}[x, y]^s$

Assumptions: $\mathcal{G}$ is a minimal reduced Gröbner basis, with initial terms $(y^{n_0}, \dots, x^{m_s})$ listed in decreasing order

Output: $(\lambda_1, \dots, \lambda_N)$ in $\mathbb{K}^N$

1: **for** $i = 0, \dots, s$ **do** $x^{m_i} y^{n_i} \leftarrow$ InitialTerm(g_i)
2: $M_0 \leftarrow 1$, $G_0 \leftarrow g_0$
3: **for** $i = 1, \dots, s$ **do**
4: $\quad M_i \leftarrow$ PolynomialCoefficient(g_i, y^{n_i}) ▷ *M_i monic in $\mathbb{K}[x]$*
5: $\quad G_i \leftarrow g_i$ div M_i
6: $\quad D_i \leftarrow M_i$ div M_{i-1} ▷ *D_i monic in $\mathbb{K}[x]$*
7: $\quad n_{i-1,i-1} \leftarrow x^{d_i} - D_i$ ▷ *$d_i = m_i - m_{i-1}$*
8: $H_s \leftarrow 1$
9: **for** $i = s-1, \dots, 0$ **do**
10: $\quad H_i \leftarrow y^{e_{i+1}} H_{i+1}$ ▷ *$e_{i+1} = n_i - n_{i+1}$*
11: $\quad \bar{A} \leftarrow \bar{G}_i - y^{e_{i+1}} \bar{H}_{i+1}$ ▷ *computation done in $\mathbb{B}[y]$, with $\mathbb{B} = \mathbb{K}[x]/\langle D_{i+1} \cdots D_s \rangle$*
12: $\quad$ **for** $j = i+1, \dots, s$ **do**
13: $\quad\quad \bar{Q}_{j,i} \leftarrow \bar{A}$ div $\bar{H}_j$, $\bar{A} \leftarrow \bar{A}$ rem $\bar{H}_j$ ▷ *Euclidean division done in $\mathbb{B}[y]$*
14: $\quad\quad Q^*_{j,i} \leftarrow$ canonical lift of $\bar{Q}_{j,i}$ to $\mathbb{K}[x, y]$
15: $\quad\quad \sigma_{j,i} \leftarrow (Q^*_{j,i}$ div $D_{i+2} \cdots D_j)$ rem D_{i+1} ▷ *by reducing Eq. (9) modulo D_{i+1}*
16: $\quad\quad H_i \leftarrow H_i +$ KroneckerMultiply$(\sigma_{j,i}, D_{i+2} \cdots D_j H_j)$ ▷ *by Eq. (6)*
17: **return** ParametersFromSigma$((y^{n_0}, \dots, x^{m_s}), (\sigma_{j,i})_{0 \le i \le s-1, i \le j \le s})$

As before, the modifications needed to deal with the punctual Gröbner cell are elementary; it suffices to invoke PunctualParametersFromSigma at the last step. The resulting procedure will be written PunctualParametersFromReducedBasis. Before proving the proposition, we give an example of computation of punctual Gröbner coefficients.

Example 6.3. *Given $\mathcal{G}$ as in the introduction from Example 1.1,*

$$\begin{aligned} &y^4 + \tfrac{17}{14} xy - \tfrac{17}{7} x^2, \\ &xy^3 - \tfrac{10}{9} x^3, \\ &x^2 y - 2x^3, \\ &x^4, \end{aligned}$$

Algorithm PunctualParametersFromReducedBasis *computes*

$$\begin{aligned} \sigma_{0,0} = \sigma_{1,0} = 0, \quad \sigma_{2,0} = \frac{17}{14}, \quad \sigma_{3,0} = \frac{40}{9}, \\ \sigma_{1,1} = \sigma_{2,1} = 0, \quad \sigma_{3,1} = -\frac{10}{9}, \\ \sigma_{2,2} = 0, \quad \sigma_{3,2} = -2x \end{aligned}$$

and thus

$$\lambda_1 = 0, \quad \lambda_2 = \frac{17}{14}, \quad \lambda_3 = \frac{40}{9}, \quad \lambda_4 = -\frac{10}{9}, \quad \lambda_5 = -2. \tag{10}$$

Proof. We already established the correctness of the algorithm. By inspection, we see that all steps involve only additions and multiplications in $\mathbb{K}$, using only integer constants, since all that is done are multiplications or Euclidean divisions by monic polynomials, either in $\mathbb{K}[x]$ or in $\mathbb{B}[y]$, with $\mathbb{B}$ of the form $\mathbb{K}[x]/\langle D_{i+1} \cdots D_s \rangle$ (this in turn reduces to additions and multiplications in $\mathbb{K}$).

It remains to establish the runtime of the algorithm. Each pass in the loop at Line 3 uses $O^{\tilde{}}(\delta)$ operations $(+, \times)$, for a total of $O^{\tilde{}}(s\delta)$. To continue the analysis, we first note that for all i, the polynomial H_i computed by the algorithm has x-degree less than $d_{i+1}+\cdots+d_s$, which is less than m_s, and y-degree n_i. The same bounds holds for $\deg(Q^*_{j,i}, x)$ (by construction); the y-degree of this polynomial is less than e_j, as mentioned for $Q_{j,i}$ during the derivation of the algorithm.

Since G_i satisfies the same degree bound $\deg(G_i, x) < d_{i+1} + \cdots + d_s$ as H_i, the reduction of $G_i - y^{e_{i+1}} H_{i+1}$ modulo $D_{i+1} \cdots D_s$ at is free. At each pass through Line 13, the Euclidean division takes $O^{\tilde{}}(n_{j-1}) \subset O^{\tilde{}}(n_0)$ operations $(+, \times)$ in $\mathbb{B}$, which is $O^{\tilde{}}(n_0 m_s)$ operations $(+, \times)$ in $\mathbb{K}$. The degree bounds given above show that the cost of computing $\sigma_{j,i}$ and updating H_i admits the same upper bound $O^{\tilde{}}(n_0 m_s)$. Since we enter the inner For loop at Line 12 $O(s^2)$ times, this gives a total cost $O^{\tilde{}}(s^2 n_0 m_s)$. □

Let us now see how to formalize the observation that the coefficients computed by Algorithm ParametersFromReducedBasis are polynomial expressions of the coefficients of $\mathcal{G}$.

Assume that the terms $\boldsymbol{E}$ are fixed, let $\mu_1, \ldots, \mu_\delta$ be the monomials not in $\langle \boldsymbol{E} \rangle$, ordered in an arbitrary fashion, and let $\Gamma_{0,1}, \ldots, \Gamma_{s,\delta}$ be $(s+1)\delta$ new variables over $\mathbb{Z}$. We set $\mathbb{A}_{\boldsymbol{E}} = \mathbb{Z}[\Gamma_{0,1}, \ldots, \Gamma_{s,\delta}]$.

Because the algorithm only performs additions and multiplications, and uses constants from the image of $\mathbb{Z}$ in $\mathbb{K}$, we deduce that there exist $P_{1,\boldsymbol{E}}, \ldots, P_{N,\boldsymbol{E}}$ in $\mathbb{A}_{\boldsymbol{E}} = \mathbb{Z}[\Gamma_{0,1}, \ldots, \Gamma_{s,\delta}]$ such that given ¡¡¡¡¡¡¡ HEAD any reduced minimal Gröbner basis $\mathcal{G} = (g_0, \ldots, g_s)$ with ======= any minimal, reduced Gröbner basis $\mathcal{G} = (g_0, \ldots, g_s)$ with ¿¿¿¿¿¿¿ 82d34faae719e68ecf5331f957167c910d916bb initial terms $\boldsymbol{E}$ and with coefficients in $\mathbb{K}$ (or any extension of it, as we choose below), the Gröbner parameters of $\mathcal{G}$ are obtained by evaluating $P_{1,\boldsymbol{E}}, \ldots, P_{N,\boldsymbol{E}}$ at the coefficients of $\mathcal{G}$.

The correctness of the algorithm can then be restated as follows. Let $\Lambda_1, \ldots, \Lambda_N$ be another set of new variables over $\mathbb{K}$, that stand for "generic" Gröbner parameters, and define $\mathbb{L} = \mathbb{K}(\Lambda_1, \ldots, \Lambda_N)$. Let further $g_{0,\mathbb{L}}, \ldots, g_{s,\mathbb{L}}$ be the polynomials obtained as output of ReducedBasisFromParameters$(\boldsymbol{E}, (\Lambda_1, \ldots, \Lambda_N))$. Since that algorithm as well performs only additions and subtractions (Proposition 5.2), these polynomials have coefficients in $\mathbb{K}[\Lambda_1, \ldots, \Lambda_N] \subset \mathbb{L}$. For $i = 0, \ldots, s$ and $j = 1, \ldots, \delta$, let then $R_{i,j} \in \mathbb{K}[\Lambda_1, \ldots, \Lambda_N]$ be the coefficient of the monomial μ_j in $g_{i,\mathbb{L}}$. We deduce from our discussion that $P_{i,\boldsymbol{E}}(R_{0,1}, \ldots, R_{s,\delta}) = \Lambda_i$ holds for all i. We will use this observation in the next subsection.

6.2. Polynomial equations for the Gröbner parameters

Let now $f_1, \ldots, f_t$ be polynomials in $\mathbb{K}[x, y]$; in this subsection, those are our inputs, and we denote by J the ideal they generate in $\mathbb{K}[x, y]$. Let further I be an ideal in $\mathbb{K}[x, y]$ such that the following properties hold:

A_1. I has dimension zero;

A_2. there exists an ideal $I' \subset \mathbb{K}[x, y]$ such that $I + I' = \langle 1 \rangle$ and $II' = J$.

Equivalently, I is the intersection (or product) of some zero-dimensional primary components of J. This is for instance the case if the origin $(0, 0)$ is isolated in $V(J)$ and I is the $\langle x, y \rangle$-primary component of J, or if $I = J$ and $V(J)$ is finite.

Let $\mathcal{G} = (g_0, \ldots, g_s) \subset \mathbb{K}[x, y]$ be the reduced lexicographic Gröbner basis of I. We denote by $\boldsymbol{E}$ the initial terms of the polynomials in $\mathcal{G}$, written as before as

$$\boldsymbol{E} = (y^{n_0}, x^{m_1} y^{n_1}, \ldots, x^{m_{s-1}} y^{n_{s-1}}, x^{m_s}).$$

In what follows, we assume that $\boldsymbol{E}$ is known, but not $\mathcal{G}$; we show how to recover the Gröbner parameters of I (and thus $\mathcal{G}$ itself).

We let δ be the degree of I, and $\mu_1, \ldots, \mu_\delta$ be the monomials not in $\langle \boldsymbol{E} \rangle$, ordered in an arbitrary way. Let further $N = \delta + m_s$ be the number of parameters for the Gröbner cell $\mathcal{C}(\boldsymbol{E})$, and let $(\lambda_1, \ldots, \lambda_N) = \phi_{\boldsymbol{E}}^{-1}(I) \in \mathbb{K}^N$ be the Gröbner parameters associated to I. In this subsection, we define a system of $t\delta$ equations $\mathscr{E}_{1,1}, \ldots, \mathscr{E}_{t,\delta}$ in $\mathbb{K}[\Lambda_1, \ldots, \Lambda_N]$, where $\Lambda_1, \ldots, \Lambda_N$ are new variables, and we prove that $(\lambda_1, \ldots, \lambda_N)$ is a solution of multiplicity 1 to these equations.

As in the previous subsection, let $\mathbb{L} = \mathbb{K}(\Lambda_1, \ldots, \Lambda_N)$ and let $g_{0,\mathbb{L}}, \ldots, g_{s,\mathbb{L}}$ be the parametric Gröbner basis of $\mathcal{C}(\boldsymbol{E})$ given by ReducedBasisFromParameters$(\boldsymbol{E}, (\Lambda_1, \ldots, \Lambda_N))$. Recall that all polynomials $g_{0,\mathbb{L}}, \ldots, g_{s,\mathbb{L}}$ have coefficients in $\mathbb{K}[\Lambda_1, \ldots, \Lambda_N]$; this implies in particular that for A in $\mathbb{K}[x, y]$, the remainder A rem $\langle g_{0,\mathbb{L}}, \ldots, g_{s,\mathbb{L}} \rangle$, which is well-defined in $\mathbb{L}[x, y] = \mathbb{K}(\Lambda_1, \ldots, \Lambda_N)[x, y]$, is in $\mathbb{K}[\Lambda_1, \ldots, \Lambda_N][x, y]$. For $j = 1, \ldots, \delta$, we then denote by $\mathcal{N}_i$ the following $\mathbb{K}$-linear map:

$$\begin{aligned} \mathcal{N}_j : \mathbb{K}[x, y] &\to \mathbb{K}[\Lambda_1, \ldots, \Lambda_N] \\ A &\mapsto \operatorname{coeff}(A \text{ rem } \langle g_{0,\mathbb{L}}, \ldots, g_{s,\mathbb{L}} \rangle, \mu_j), \end{aligned}$$

with $\mu_1, \ldots, \mu_\delta$ the monomials not in $\langle \boldsymbol{E} \rangle$, as defined above. For $i = 1, \ldots, t$, we then let

$$\mathscr{E}_{i,1}, \ldots, \mathscr{E}_{i,\delta} = \mathcal{N}_1(f_i), \ldots, \mathcal{N}_\delta(f_i),$$

thus defining $t\delta$ polynomials $\mathscr{E}_{1,1}, \ldots, \mathscr{E}_{t,\delta}$ in $\mathbb{K}[\Lambda_1, \ldots, \Lambda_N]$. The following key property for these equations was inspired by [25, Theorem 4.8], which was stated in the context of border bases. The conclusion in the theorem in that reference is that the ideal generated by the equations used therein is maximal, which means that it admits a unique solution, and this solution has multiplicity 1. In our case, we recover the multiplicity 1 property, which is what we will need for Newton iteration.

Proposition 6.4. *$(\lambda_1, \ldots, \lambda_N)$ is a solution of $\mathscr{E}_{1,1}, \ldots, \mathscr{E}_{t,\delta}$ of multiplicity 1.*

Proof. Let $\mathcal{I}$ be the ideal generated by all polynomials $\mathcal{N}_i(g_j)$, for $i = 1, \ldots, \delta$ and $j = 0, \ldots, s$, and let $R_{0,1}, \ldots, R_{s,\delta} \in \mathbb{K}[\Lambda_1, \ldots, \Lambda_N]$ be the coefficients of $(g_{0,\mathbb{L}}, \ldots, g_{s,\mathbb{L}})$, as in the previous subsection. Then, for $i = 1, \ldots, \delta$ and $j = 0, \ldots, s$, the polynomial $\mathcal{N}_i(g_j)$ is equal to $R_{j,i}(\lambda_1, \ldots, \lambda_N) - R_{j,i}$. In particular, $(\lambda_1, \ldots, \lambda_N)$ is in the zero-set of $\mathcal{I}$.

Recall further from the previous subsection the existence of polynomials $P_{1,\boldsymbol{E}}, \ldots, P_{N,\boldsymbol{E}}$, with $P_{k,\boldsymbol{E}}(R_{0,1}, \ldots, R_{s,\delta}) = \Lambda_k$ for all k. The fact that $R_{j,i}(\lambda_1, \ldots, \lambda_N) - R_{j,i}$ is in $\mathcal{I}$ for all i, j implies that

$$P_{k,\boldsymbol{E}}(R_{0,1}(\lambda_1, \ldots, \lambda_N), \ldots, R_{s,\delta}(\lambda_1, \ldots, \lambda_N)) - P_{k,\boldsymbol{E}}(R_{0,1}, \ldots, R_{s,\delta})$$

is in $\mathcal{I}$ as well, for all $k = 1, \ldots, N$. The left-hand side is λ_k, and the right-hand side Λ_k, so that $\mathcal{I}$ contains all polynomials $\Lambda_1 - \lambda_1, \ldots, \Lambda_N - \lambda_N$. Taken together, the two paragraphs so far establish that $\mathcal{I} = \langle \Lambda_1 - \lambda_1, \ldots, \Lambda_N - \lambda_N \rangle$.

Let now $\mathcal{J}$ be the ideal generated in $\mathbb{K}[\Lambda_1, \ldots, \Lambda_N]$ by the polynomials $\mathscr{E}_{1,1}, \ldots, \mathscr{E}_{t,\delta}$. Remark first that for any $a, b \geq 0$ and $i = 1, \ldots, t$,

$$(x^a y^b f_i) \text{ rem } \langle g_{0,\mathbb{L}}, \ldots, g_{s,\mathbb{L}} \rangle = \sum_{j=1}^{\delta} \mathcal{N}_j(f_i)(x^a y^b \mu_j \text{ rem } \langle g_{0,\mathbb{L}}, \ldots, g_{s,\mathbb{L}} \rangle).$$

It follows that for any A in $J = \langle f_1, \ldots, f_t \rangle$, and for $j = 1, \ldots, \delta$, $\mathcal{N}_j(A)$ is in $\mathcal{J}$. For the same reason, for A in $I = \langle g_0, \ldots, g_s \rangle$, and for $j = 1, \ldots, \delta$, $\mathcal{N}_j(A)$ is in $\mathcal{I}$. We will also need the fact that for A in I^2, and for all j, $\mathcal{N}_j(A)$ is in $\mathcal{I}^2$; this is established similarly.

Recall now our second assumption on I': there exists an ideal $I' \subset \mathbb{K}[x, y]$ such that $I + I' = \langle 1 \rangle$ and $II' = J$. Since J is contained in I, the statements in the previous paragraph imply that $\mathcal{J}$ is contained in $\mathcal{I} = \langle \Lambda_1 - \lambda_1, \ldots, \Lambda_N - \lambda_N \rangle$, so that $(\lambda_1, \ldots, \lambda_N)$ is in the zero-set of $\mathcal{J}$. This proves the first claim of the proposition.

Let further K, K' be in resp. I and I' such that $K + K' = 1$. For $i = 0, \ldots, s$, g_i is in I, so that $g_i K' = g_i - g_i K$ is in $II' = J$. By the remark above, for $j = 1, \ldots, \delta$, $A_{j,i} := \mathcal{N}_j(g_i) - \mathcal{N}_j(g_i K)$ is then in $\mathcal{J}$, whereas $\mathcal{N}_j(g_i K)$ is in $\mathcal{I}^2$.

Consider the Jacobian matrix $\boldsymbol{J}$ of all polynomials $A_{j,i}$ at $(\lambda_1, \ldots, \lambda_N)$. Because all terms $\mathcal{N}_j(g_i K)$ are in $\mathcal{I}^2 = \langle \Lambda_1 - \lambda_1, \ldots, \Lambda_N - \lambda_N \rangle^2$, their Jacobian matrix vanishes at $(\lambda_1, \ldots, \lambda_N)$, so that $\boldsymbol{J}$ is simply the Jacobian matrix of the polynomials $\mathcal{N}_j(g_i)$ at $(\lambda_1, \ldots, \lambda_N)$. Because these polynomials generate the ideal $\mathcal{I} = \langle \Lambda_1 - \lambda_1, \ldots, \Lambda_N - \lambda_N \rangle$, this matrix has trivial kernel. Thus, $\mathcal{J}$ has multiplicity 1 at $(\lambda_1, \ldots, \lambda_N)$. □

In the particular case where $I = J$, we have a slightly stronger result.

Corollary 6.5. *Suppose that $I = \langle f_1, \ldots, f_t \rangle$. Then, $\langle \mathscr{E}_{1,1}, \ldots, \mathscr{E}_{t,\delta} \rangle = \langle \Lambda_1 - \lambda_1, \ldots, \Lambda_N - \lambda_N \rangle$ in $\mathbb{K}[\Lambda_1, \ldots, \Lambda_N]$.*

Proof. Using the notation in the proof of the proposition, we see that if $I = J$, then $\mathcal{I} = \mathcal{J}$, and we proved that $\mathcal{I} = \langle \Lambda_1 - \lambda_1, \ldots, \Lambda_N - \lambda_N \rangle$. □

In our other particular case, where I is the $\langle x, y\rangle$-primary component of J, we can obtain a similar stronger statement. Recall that the punctual Gröbner cell $\mathcal{C}_0(\boldsymbol{E})$ has dimension $N' = \delta - n_0$, and that the parameters for $\mathcal{C}_0(\boldsymbol{E})$ are obtained by setting $N - N'$ parameters to zero in the parameters $\Lambda_1, \ldots, \Lambda_N$ of $\mathcal{C}(\boldsymbol{E})$.

Let $\tau_1, \ldots, \tau_{N-N'}$ be the indices of these parameters set to zero, and let $\Lambda_{\sigma_1}, \ldots, \Lambda_{\sigma_{N'}}$ be the remaining N' parameters. For $i = 1, \ldots, t$ and $j = 1, \ldots, \delta$, let $\mathscr{F}_{i,j}$ be the polynomial in $\mathbb{K}[\Lambda_{\sigma_1}, \ldots, \Lambda_{\sigma_{N'}}]$ obtained by setting $\Lambda_{\tau_1}, \ldots, \Lambda_{\tau_{N-N'}}$ to zero in $\mathscr{E}_{i,j}$. Then, we have the following.

Corollary 6.6. *Suppose that I is $\langle x, y\rangle$-primary. Then, $\langle \mathscr{F}_{1,1}, \ldots, \mathscr{F}_{t,\delta}\rangle = \langle \Lambda_{\sigma_1} - \lambda_{\sigma_1}, \ldots, \Lambda_{\sigma_{N'}} - \lambda_{\sigma_{N'}}\rangle$ in $\mathbb{K}[\Lambda_{\sigma_1}, \ldots, \Lambda_{\sigma_{N'}}]$.*

Proof. We proved in Proposition 6.4 that $\lambda_{\sigma_1}, \ldots, \lambda_{\sigma_{N'}}$ is a solution of $\mathscr{F}_{1,1}, \ldots, \mathscr{F}_{t,\delta}$. Besides, since the Jacobian matrix of $\mathscr{E}_{1,1}, \ldots, \mathscr{E}_{t,\delta}$ has trivial kernel at $(\lambda_1, \ldots, \lambda_N)$ (with thus $\lambda_{\tau_1} = \cdots = \lambda_{\tau_{N-N'}} = 0$), it is also the case for that of $\mathscr{F}_{1,1}, \ldots, \mathscr{F}_{t,\delta}$ at $(\lambda_{\sigma_1}, \ldots, \lambda_{\sigma_{N'}})$. The only missing property is thus that $(\lambda_{\sigma_1}, \ldots, \lambda_{\sigma_{N'}})$ is the only common solution to these equations. Let $(\lambda^\star_{\sigma_1}, \ldots, \lambda^\star_{\sigma_{N'}}) \in \overline{\mathbb{K}}^{N'}$ be such a solution, let $\mathcal{G}^\star$ be the corresponding reduced Gröbner basis, and let $I^\star$ be the ideal it generates (in particular, $V(I^\star) = \{(0,0)\}$). Since by assumption $\mathcal{G}^\star$ reduces $f_1, \ldots, f_t$ to zero, we have $J \subset I^\star$.

By assumption on I, there exists an ideal $I' \subset \mathbb{K}[x, y]$ such that $I + I' = \langle 1\rangle$ and $II' = J$. Let K, K' be in resp. I and I' such that $K + K' = 1$; in particular, K' does not vanish at $(0,0)$. Since $V(I^\star) = \{(0,0)\}$, it follows that K' is a unit modulo $I^\star$.

Recall that we write $\mathcal{G} = (g_0, \ldots, g_s)$ for the reduced lexicographic Gröbner basis of I. Then, for $i = 0, \ldots, s$, the polynomial $g_i K'$ is in II', so in J, and thus in $I^\star$. Since K' is a unit modulo $I^\star$, this means that g_i is in $I^\star$. Altogether, this proves that I is contained in $I^\star$. Since these ideals have the same initial ideals for the lexicographic order, they are then equal. This in turn proves that $(\lambda_{\sigma_1}, \ldots, \lambda_{\sigma_{N'}}) = (\lambda^\star_{\sigma_1}, \ldots, \lambda^\star_{\sigma_{N'}})$. □

Example.. In our running example, we consider only the punctual Gröbner cell, and we take f_1 and f_2 as in the introduction. To write the equations for the punctual Gröbner parameters, we consider $g_{0,\mathbb{L}}, \ldots, g_{3,\mathbb{L}}$ and set to zero the parameters written $\Lambda_{\tau_1}, \ldots, \Lambda_{\tau_{N-N'}}$ above; the resulting polynomials were given in (5), written in variables $\lambda_1, \ldots, \lambda_5$ (recall that $N' = 5$ here). After reducing f_1 and f_2 by these polynomials, and taking coefficients (we discard those that are identically zero), we obtain

$$14\Lambda_1, \quad 14\Lambda_2 - 17, \quad -14\Lambda_1\Lambda_5^2 + 14\Lambda_3 - 28\Lambda_4\Lambda_5, \quad 14\Lambda_2\Lambda_5 + 34, \quad -18\Lambda_4 + 10\Lambda_5. \tag{11}$$

As claimed, these polynomials generate the maximal ideal

$$\Lambda_1, \quad \Lambda_2 - 17/14, \quad \Lambda_3 - 40/9, \quad \Lambda_4 + 10/9, \quad \Lambda_5 + 2.$$

Because the input f_1, f_2 and $\mathcal{G}$ have rather small degrees, the equations in (11) can be solved by hand. There is of course no reason for this to be the case in general, although on several other examples, we observed the presence of some linear equations (understanding the structure of this system of equations is an interesting, but nontrivial, question).

7. Newton iteration

We can finally describe our main algorithm, which computes Gröbner parameters using Newton iteration. For this, we will suppose that $\mathbb{K}$ is the field of fractions of a domain $\mathbb{A}$, and we consider a maximal ideal $\mathfrak{m}$ in $\mathbb{A}$, with residual field $\Bbbk = \mathbb{A}/\mathfrak{m}$.

Consider the following objects: polynomials $(f_1, \ldots, f_t)$ in $\mathbb{A}[x, y]$ and a minimal, reduced Gröbner basis $\mathcal{G}$ in $\mathbb{K}[x, y]$, with initial terms $\boldsymbol{E}$. We make the following assumptions:

- A'_1. the ideal generated by $\mathcal{G}$ in $\mathbb{K}[x, y]$ is the intersection of some of the primary components of $\langle f_1, \ldots, f_t\rangle$,
- A'_2. all polynomials in $\mathcal{G}$ are in $\mathbb{A}_\mathfrak{m}[x, y]$, where $\mathbb{A}_\mathfrak{m}$ is the localization of $\mathbb{A}$ at the maximal ideal $\mathfrak{m}$,
- A'_3. the ideal generated by $\mathcal{G}_\mathfrak{m} = \mathcal{G}$ rem $\mathfrak{m}$ in $\Bbbk[x, y]$ is the intersection of some of the primary components of the ideal $\langle f_1$ rem $\mathfrak{m}, \ldots, f_t$ rem $\mathfrak{m}\rangle$.

The last two items express that $\mathfrak{m}$ is good for $\mathcal{G}$, in the sense of Definition 1.2. Important cases where the first and third assumptions are satisfied are as in the previous subsection, viz. when $\mathcal{G}_\mathfrak{m}$ and $\mathcal{G}$ generate the ideals $\langle f_1$ rem $\mathfrak{m}, \ldots, f_t$ rem $\mathfrak{m}\rangle$, resp. $\langle f_1, \ldots, f_t\rangle$ themselves, or when they describe the $\langle x, y\rangle$-primary components of these ideals.

Given $\mathfrak{m}$, $(f_1, \ldots, f_t)$ and $\mathcal{G}_\mathfrak{m}$, we show here how to compute $\mathcal{G}$ rem $\mathfrak{m}^K$, for an arbitrary $K \geq 1$.

Algorithm LiftOneStep describes the core lifting procedure; it takes as input the Gröbner parameters of $\mathcal{G}$, known modulo $\mathfrak{m}^\kappa$, for some $\kappa \geq 0$, and returns these parameters modulo $\mathfrak{m}^{2\kappa}$ (note that since $\mathcal{G}$ has coefficients in $\mathbb{A}_\mathfrak{m}$ by A'_2, its Gröbner parameters are in $\mathbb{A}_\mathfrak{m}$ as well, so reducing them modulo powers of $\mathfrak{m}$ makes sense).

The algorithm simply applies Newton's iteration to the equations $\mathscr{E}_{i,j}$ introduced in the previous subsection: at each iteration, given the Gröbner parameters of $\mathcal{G}$ modulo $\mathfrak{m}^\kappa$, we solve the linearization of these equations in order to lift the Gröbner parameters modulo $\mathfrak{m}^{2\kappa}$. Note however that we never explicitly write down the equations $\mathscr{E}_{i,j}$, as they may involve a large number of terms: instead, we reduce the input equations $f_1, \ldots, f_t$ modulo a Gröbner basis $\mathcal{G}^*$ with parametric coefficients, and extract coefficients in the remainder. We only compute the first order Taylor expansions of these coefficients, as this is enough to conduct the iteration; this explains why below, we work modulo the ideal $\langle \Lambda_1, \ldots, \Lambda_N\rangle^2$.

Since we want to give a cost estimate that counts operations in $\mathbb{A}_{2\kappa}$, we here assume that we already know the reductions of the input equations $f_1, \ldots, f_t$ modulo $\mathfrak{m}^{2\kappa}$; they are written $f'_1, \ldots, f'_t \in \mathbb{A}_{2\kappa}[x, y]$. Some steps in the algorithm require a few further comments, namely the calls to ReducedBasisFromParameters at Line 5, Reduction at Line 8 and LinearSolve at Line 11.

- At Line 5, we are working with Gröbner parameters written $(\ell_1, \ldots, \ell_N)$, that are in $\mathbb{B} = \mathbb{A}_{2\kappa}[\Lambda_1, \ldots, \Lambda_N]/\langle \Lambda_1, \ldots, \Lambda_N\rangle^2$ (in the algorithm, elements of $\mathbb{B}$ are written as $b_0 + \sum_{i=1}^n b_i \Lambda_i$, for some b_i's in $\mathbb{A}_{2\kappa}$). Recall that Algorithm ReducedBasisFromParameters only does additions and multiplications, and uses constants from $\mathbb{Z}$, so we can

run this algorithm with inputs in $\mathbb{B}$; however the proof of correctness (Proposition 7.1) use the original properties (Proposition 5.2) which were only established for inputs in a field; the proof of the following proposition addresses this.

The same remark applies at Line 8, for Algorithm ReductionGeneralInput.

- The last subroutine solves a linear system over $\mathbb{A}_{2\kappa}$: the inputs are elements of $\mathbb{B}$, which we recall take the form $b_0 + \sum_{i=1}^{n} b_i \Lambda_i$, for some b_i's in $\mathbb{A}_{2\kappa}$. Procedure LinearSolve then sees these elements are linear equations in the Λ_i's. We will prove the existence and uniqueness of the solution, by showing that the corresponding matrix admits a maximal minor that does not vanish modulo $\mathfrak{m}$.

Algorithm 7.1 LiftOneStep$((f'_1, \ldots, f'_t), \boldsymbol{E}, (\alpha_1, \ldots, \alpha_N))$

Input: $(f'_1, \ldots, f'_t)$ in $\mathbb{A}_{2\kappa}[x, y]$, monomials $\boldsymbol{E}$, $(\alpha_1, \ldots, \alpha_N)$ in $\mathbb{A}_{\kappa}^N$
Output: $(\alpha''_1, \ldots, \alpha''_N)$ in $\mathbb{A}_{2\kappa}^N$
1: $(\alpha'_1, \ldots, \alpha'_N) \leftarrow$ lift of $(\alpha_1, \ldots, \alpha_N)$ to $\mathbb{A}_{2\kappa}^N$
2: $\mu_1, \ldots, \mu_\delta \leftarrow$ monomials not in $\langle \boldsymbol{E} \rangle$
3: **for** $i = 1, \ldots, N$ **do**
4: $\quad \ell_i \leftarrow \alpha'_i + \Lambda_i$ ▷ *all* ℓ_i *in* $\mathbb{B} = \mathbb{A}_{2\kappa}[\Lambda_1, \ldots, \Lambda_N]/\langle \Lambda_1, \ldots, \Lambda_N \rangle^2$
5: $\mathcal{G}^* \leftarrow$ ReducedBasisFromParameters$(\boldsymbol{E}, (\ell_1, \ldots, \ell_N))$ ▷ *computations done over* $\mathbb{B}$
6: $\mathscr{R} \leftarrow [\,]$
7: **for** $i = 1, \ldots, t$ **do**
8: $\quad r_i \leftarrow$ ReductionGeneralInput$(f'_i, \mathcal{G}^*)$ ▷ *computations done over* $\mathbb{B}$
9: $\quad$ **for** $j = 1, \ldots, \delta$ **do** $r_{i,j} \leftarrow \mathrm{coeff}(r_i, \mu_j)$ ▷ *all* $r_{i,j}$ *in* $\mathbb{B}$
10: $\quad \mathscr{R} \leftarrow \mathscr{R}$ cat $[r_{i,1}, \ldots, r_{i,\delta}]$ ▷ $\mathscr{R}$ *is an array with entries in* $\mathbb{B}$
11: $(\epsilon_1, \ldots, \epsilon_N) \leftarrow$ LinearSolve$(\mathscr{R})$ ▷ *all* ϵ_i *in* $\mathbb{A}_{2\kappa}$
12: **for** $i = 1, \ldots, N$ **do** $\alpha''_i \leftarrow \alpha'_i + \epsilon_i$ ▷ *all* α''_i *in* $\mathbb{A}_{2\kappa}$
13: **return** $(\alpha''_1, \ldots, \alpha''_N)$

Proposition 7.1. *Suppose that* A'_1, A'_2, A'_3 *hold, and let* $(\lambda_1, \ldots, \lambda_N) \in \mathbb{A}_{\mathfrak{m}}^N$ *be the Gröbner parameters of* $\mathcal{G}$. *Given* $(f_1, \ldots, f_t)$ rem $\mathfrak{m}^{2\kappa}$ *and* $(\lambda_1$ rem $\mathfrak{m}^{\kappa}, \ldots, \lambda_N$ rem $\mathfrak{m}^{\kappa})$, *Algorithm* LiftOneStep *correctly returns* $(\lambda_1$ rem $\mathfrak{m}^{2\kappa}, \ldots, \lambda_N$ rem $\mathfrak{m}^{2\kappa})$.

Proof. Let $\lambda = (\lambda_1, \ldots, \lambda_N) \in \mathbb{A}_{\mathfrak{m}}^N$ be the Gröbner parameters associated to $\mathcal{G}$. By assumption, the vector $\alpha = (\alpha_1, \ldots, \alpha_N)$ satisfies $\alpha = \lambda$ rem $\mathfrak{m}^{\kappa}$, and the same holds for α'. We prove that the output $\alpha'' = (\alpha''_1, \ldots, \alpha''_N)$ is equal to λ rem $\mathfrak{m}^{2\kappa}$.

This is simply the classical proof of the validity of Newton's iteration. Let δ be the degree of $\rangle \mathcal{G} \langle$, and let $\mathscr{E} = (\mathscr{E}_{1,1}, \ldots, \mathscr{E}_{t,\delta})$ be the equations introduced in the previous subsection for the polynomials $f_1, \ldots, f_t$ and the Gröbner cell $\mathcal{C}(\boldsymbol{E})$, over the field $\mathbb{K}$. Since all f_i's have coefficients in $\mathbb{A}$, and since the reduction process introduces no new denominator, the polynomials $\mathscr{E}$ are in $\mathbb{A}[\Lambda_1, \ldots, \Lambda_N]$. Using Proposition 6.4, assumption A'_1 shows that λ is a

solution to these equations (and that their Jacobian matrix at λ has trivial kernel, but we will not need this fact directly).

Let further $\mathscr{E}_{\mathfrak{m}} = (\mathscr{E}_{\mathfrak{m},1,1}, \ldots, \mathscr{E}_{\mathfrak{m},t,\delta})$ be these same equations, but this time for the polynomials f_1 rem $\mathfrak{m}, \ldots, f_t$ rem $\mathfrak{m}$ and $\mathcal{G}_{\mathfrak{m}}$. These are polynomials in $\Bbbk[\Lambda_1, \ldots, \Lambda_N]$, with $\mathscr{E}_{\mathfrak{m}} = \mathscr{E}$ rem $\mathfrak{m}$. Using Proposition 6.4, assumption A'_3 shows that λ rem $\mathfrak{m}$ is a solution to these equations (which we already could deduce from the previous paragraph) and that their Jacobian matrix at λ rem $\mathfrak{m}$ has trivial kernel. We will use this below.

We claim that for all i, j, the coefficient $r_{i,j}$ computed at Line 9 is equal to $\mathscr{E}_{i,j}(\ell_1, \ldots, \ell_N)$, computed in $\mathbb{B} = \mathbb{A}_{2\kappa}[\Lambda_1, \ldots, \Lambda_N]/\langle \Lambda_1, \ldots, \Lambda_N \rangle^2$. The only point we have to be careful with is that the output of Algorithm ReducedBasisFromParameters is specified as being a Gröbner basis only if the inputs are in a field. To deal with this, let $\ell'_1, \ldots, \ell'_N$ be arbitrary lifts of $\ell_1, \ldots, \ell_N$ to the domain $\mathbb{A}[\Lambda_1, \ldots, \Lambda_N]$, and let $\mathcal{G}'$ be the output of ReducedBasisFromParameters$(\boldsymbol{E}, (\ell'_1, \ldots, \ell'_N))$. These polynomials form a Gröbner basis in $\mathbb{K}(\Lambda_1, \ldots, \Lambda_N)[x, y]$, which happens to have all its coefficients in $\mathbb{A}[\Lambda_1, \ldots, \Lambda_N]$, and $\mathcal{G}^*$ computed at Line 5 is the reduction of $\mathcal{G}'$ modulo $\mathfrak{m}^{2\kappa} + \langle \Lambda_1, \ldots, \Lambda_N \rangle^2$.

Similarly, at Line 8, Algorithm ReductionGeneralInput can take as input polynomials with coefficients in $\mathbb{B}$, but its output was only specified for polynomials with coefficients in a field. This is handled as before, and gives us that for all i, r_i is the reduction modulo $\mathfrak{m}^{2\kappa} + \langle \Lambda_1, \ldots, \Lambda_N \rangle^2$ of the polynomial f_i rem $\mathcal{G}'$. Now, the coefficients of f_i rem $\mathcal{G}'$ are the polynomials $\mathscr{E}_{i,j}$ evaluated at $(\ell'_1, \ldots, \ell'_N)$, so altogether, for all i, j, $r_{i,j} = \mathscr{E}_{i,j}(\ell_1, \ldots, \ell_N)$, as an element of $\mathbb{B}$. Taking all i, j at once, we obtain the following equalities over $\mathbb{B}$:

$$\begin{aligned} \mathscr{R} &= \mathscr{E}(\alpha'_1 + \Lambda_1, \ldots, \alpha'_N + \Lambda_N) \\ &= \mathscr{E}(\alpha') + \mathrm{jac}(\mathscr{E}, \alpha')[\Lambda_1 \ \cdots \ \Lambda_N]^T, \end{aligned}$$

where $\mathrm{jac}(\mathscr{E}, \alpha')$ is the Jacobian matrix of $\mathscr{E}$ evaluated at α'. First, we show that the system of linear equations $\mathscr{R}$ has a unique solution $\epsilon = (\epsilon_1, \ldots, \epsilon_N)$ in $\mathbb{A}_{2\kappa}^N$. Indeed, given two solution vectors ϵ and ϵ' in $\mathbb{A}_{2\kappa}^N$, we obtain the relation

$$\mathrm{jac}(\mathscr{E}, \alpha')[\epsilon_1 - \epsilon'_1 \ \cdots \ \epsilon_N - \epsilon'_N]^T = [0 \ \cdots \ 0]^T$$

over $\mathbb{A}_{2\kappa}$. We pointed out above that $\mathrm{jac}(\mathscr{E}$ rem $\mathfrak{m}, \lambda$ rem $\mathfrak{m})$ has trivial kernel, so it admits a non-zero N-minor in $\Bbbk = \mathbb{A}/\mathfrak{m} = \mathbb{A}_{2\kappa}/\mathfrak{m}$. Now, by assumption, α' rem $\mathfrak{m} = \lambda$ rem $\mathfrak{m}$, so that $\mathrm{jac}(\mathscr{E}, \alpha')$ itself admits an N-minor invertible modulo $\mathfrak{m}$, and thus in $\mathbb{A}_{2\kappa}$. This in turn implies that $\epsilon = \epsilon'$, as vectors over $\mathbb{A}'/\mathfrak{m}^{2\kappa}$. Our first claim is proved.

Second, we show that $\epsilon = (\lambda - \alpha')$ rem $\mathfrak{m}^{2\kappa}$ is a solution to these linear equations. Indeed, modulo $\mathfrak{m}^{2\kappa}$, we have the Taylor expansion $\mathscr{E}(\alpha' + \epsilon) = \mathscr{E}(\alpha') + \mathrm{jac}(\mathscr{E}, \alpha')[\epsilon_1 \ \cdots \ \epsilon_N]^T$: higher-order terms vanish, since all entries of ϵ are by assumption in $\mathfrak{m}^\kappa$. Now, $\alpha' + \epsilon = \lambda$ rem $\mathfrak{m}^{2\kappa}$, so $\mathscr{E}(\alpha' + \epsilon) = 0$ rem $\mathfrak{m}^{2\kappa}$, and our claim follows.

The two previous paragraphs prove that at the end of the while loop, the value α'' satisfies $\alpha'' = \alpha' + (\lambda - \alpha')$ rem $\mathfrak{m}^{2\kappa} = \lambda$ rem $\mathfrak{m}^{2\kappa}$, so the proof is complete. □

Proposition 7.2. *Let $\boldsymbol{E} = (y^{n_0}, x^{m_1}y^{n_1}, \ldots, x^{m_{s-1}}y^{n_{s-1}}, x^{m_s})$ be the initial terms of $\mathcal{G}$, and suppose that all f_i's have degree at most d.*

Under assumptions A'_1, A'_2, A'_3, *Algorithm* LiftOneStep *uses* $O^{\sim}(s^2\delta n_0 m_s + t\delta(d^2 + dm_s + s\delta + \delta^{\omega-1}))$ *operations in* $\mathbb{A}_{2\kappa}$.

Proof. By convention (see the introduction), lifting each α_i to α'_i takes one operation in $\mathbb{A}_{2\kappa}$, for a total of $O(N) = O(\delta)$ operations.

By Proposition 5.2, computing $\mathcal{G}^*$ takes $O^{\sim}(s^2 n_0 m_s)$ operations $(+, \times)$ in $\mathbb{B}$, with each such operation taking $O(\delta)$ operations in $\mathbb{A}_{2\kappa}$.

At Line 8, by Proposition 4.7, Algorithm ReductionGeneralInput uses $O^{\sim}(d^2 + dm_s + n_0 m_s + s\delta)$ operations $(+, \times)$ in $\mathbb{B}$. Here, we know that n_0 is at most d, so the runtime for all f_i's becomes $O^{\sim}(t(d^2 + dm_s + s\delta))$ operations in $\mathbb{B}$, which is $O^{\sim}(t\delta(d^2 + dm_s + s\delta))$ operations in $\mathbb{A}_{2\kappa}$.

Finally, we have to solve the linear system defined by $\mathscr{R} = 0$ over $\mathbb{A}_{2\kappa}$. This is a system in $t\delta$ equations and N unknowns, and we know that it admits a unique solution in $\mathbb{A}_{2\kappa}^N$, since the corresponding matrix has trivial kernel modulo $\mathfrak{m}$. Even though $\mathbb{A}_{2\kappa}^N$ is not a field, we may still apply fast algorithms, such as the one in [32] (as extended in [33]), replacing zero-tests by invertibility tests; this takes $O^{\sim}(t\delta^{\omega})$ operations in $\mathbb{A}_{2\kappa}$. □

As usual, if $\mathcal{G}$ (and thus $\mathcal{G}_{\mathfrak{m}}$) is $\langle x, y\rangle$-primary, we may use a variant of this lifting procedure, called LiftOneStepPunctualParameters, which uses ReducedBasisFromPunctualParameters as the first key subroutine. It allows us to work with N' rather than N unknown Gröbner parameters; the proof now relies on Corollary 6.6, and the runtime becomes $O^{\sim}(s^2\delta n_0 m_s + t\delta^2(m_s + \delta^{\omega-2}))$ operations in $\mathbb{A}_{2\kappa}$ (see Proposition 4.7).

At this stage, we are almost done with the proof of Theorem 1.4: for $K = 2^k$, the algorithm simply computes $\mathcal{G}$ rem $\mathfrak{m}^K$ through repeated calls to Algorithm LiftOneStep. However, this procedure works with Gröbner parameters as input and output. Hence, prior to entering Algorithm LiftOneStep for the first time, we compute the Gröbner parameters of $\mathcal{G}$ rem $\mathfrak{m}$, and after the last call to Algorithm LiftOneStep, we compute $\mathcal{G}$ rem $\mathfrak{m}^K$ using Algorithm ReducedBasisFromParameters. This extra work does not affect the asymptotic runtime, so that we do $O^{\sim}(s^2\delta n_0 m_s + t\delta(d^2 + dm_s + s\delta))$ operations in $\mathbb{A}/\mathfrak{m}^{2^i}$, for $i = 1, \ldots, k$.

The only operations not accounted for so far are the coefficient-wise reductions of the polynomials $f_1, \ldots, f_t$ modulo $\mathfrak{m}^2, \ldots, \mathfrak{m}^{2^k}$. These cannot be expressed in terms of operations in the residue class rings $\mathbb{A}/\mathfrak{m}^{2^i}$; instead, as per the convention in the introduction, we assume that each coefficient reduction modulo $\mathfrak{m}^{2^i}$ takes time T_{2^i}, for a total of $td^2T_{2^i}$ for each $i = 1, \ldots, k$. This concludes the proof of our main theorem. When we work with the punctual Gröbner cell, we saw in Proposition 4.7 that only δm_s coefficients of each input polynomial are needed, whence $t\delta m_s T_{2^i}$ steps for coefficient reduction, for all indices i.

Remark 7.3. *If one wishes to work only with Gröbner bases as input and output, it is straightforward to design algorithms called* LiftOneStepGroebnerBasis *(and* LiftOneStepPunctualGroebnerBasis*), that take* $f'_1, \ldots, f'_t$ *and* $\mathcal{G}$ mod $\mathfrak{m}^{\kappa}$ *as input and return* $\mathcal{G}$ mod $\mathfrak{m}^{2\kappa}$. *It suffices to call Algorithm* ParametersFromReducedBasis *when entering the procedure, then Algorithm* LiftOneStep*, and finally Algorithm* ReducedBasisFromParameters *before exiting (or their punctual variants). This does not affect asymptotic runtimes, but is not useful in the context of our main theorem.*

Remark 7.4. *When $\mathfrak{m}$ is principal, we can slightly improve the lifting procedure by using either divide-and-conquer techniques (folklore) or relaxed algorithms [4, Section 4] to solve the linear system that gives $\epsilon_1, \ldots, \epsilon_N$. The downside is that the runtime is not written in terms of operations in $\mathbb{A}_{2\kappa}$ anymore. Instead, we give runtimes for the common cases $\mathbb{A} = \mathbb{Z}$ and $\mathfrak{m} = \langle p \rangle$, and $\mathbb{A} = \Bbbk[t]$ and $\mathfrak{m} = \langle t - \tau \rangle$:*

- *In the former case, solving the system uses $O^{\sim}(t\delta^\omega \log(p))$ bit operations, for a one-time computation (matrix inversion) done modulo p, and $O^{\sim}(\delta^2 \kappa \log(p))$ for subsequently solving the system modulo $p^{2\kappa}$.*
- *In the latter case, the one time computation takes $O^{\sim}(t\delta^\omega)$ operations in $\Bbbk$, after which linear system solving takes $O^{\sim}(\delta^2\kappa)$ operations in $\Bbbk$.*

To wit, each operation in $\mathbb{A}_{2\kappa}$, as reported in Proposition 7.2, takes $O^{\sim}(\kappa \log(p))$ bit operations in the former case, and $O^{\sim}(\kappa \log(p))$ operations in the latter. The net effect is that in both cases, the cost of solving the linear system can be neglected (up to the one-time computation we perform at the beginning).

Example 7.5. *We show one step of the algorithm for our running example (Example 1.1), focusing on the punctual Gröbner parameters. Our input is the polynomials f_1, f_2 as in the introduction, together with the Gröbner basis of the $\langle x, y \rangle$-primary component of $\langle f_1 \bmod p, f_2 \bmod p \rangle$, with $p = 11$; namely:*

$$\left| \begin{array}{l} y^4 + 2xy + 7x^2, \\ xy^3 + 5x^3, \\ x^2y + 9x^3, \\ x^4. \end{array} \right.$$

We deduce the punctual Gröbner parameters modulo 11, $\alpha = (0, 2, 2, 5, 9) \in \mathbb{Z}/11\mathbb{Z}^5$ (recall that $N' = 5$ here). Following the algorithm, we set $(\ell_1, \ldots, \ell_5) = (\Lambda_1, 2 + \Lambda_2, 2 + \Lambda_3, 5 + \Lambda_4, 9 + \Lambda_5)$ and we compute the corresponding punctual Gröbner basis, with coefficients truncated modulo 11^2 and $\langle \Lambda_1, \ldots, \Lambda_N \rangle^2$. We obtain the polynomials written $\mathcal{G}^*$ in the pseudo-code:

$$\left| \begin{array}{l} y^4 + \Lambda_1 xy^2 + (\Lambda_2 + 2)xy + (40\Lambda_1 + \Lambda_3 + 103\Lambda_4 + 111\Lambda_5 + 33)x^3 + (9\Lambda_2 + 2\Lambda_5 + 18)x^2, \\ xy^3 + (\Lambda_4 + 5)x^3, \\ x^2y + (\Lambda_5 + 9)x^3, \\ x^4. \end{array} \right.$$

Reducing f_1 and f_2 modulo $\mathcal{G}^*$ (with calculations done modulo 11^2 and $\langle \Lambda_1, \ldots, \Lambda_N \rangle^2$), and keeping coefficients, we obtain the linear equations $\mathscr{R}$ (we only show the non-zero ones)

$$14\Lambda_1 = 14\Lambda_2 + 11 = 76\Lambda_1 + 14\Lambda_3 + 111\Lambda_4 + 102\Lambda_5 + 99 = 5\Lambda_2 + 28\Lambda_5 + 44 = 103\Lambda_4 + 10\Lambda_5 = 0.$$

They admit the following unique solution modulo 11^2:

$$\epsilon_1 = 0, \ \epsilon_2 = 77, \ \epsilon_3 = 110, \ \epsilon_4 = 88, \ \epsilon_5 = 110;$$

as expected, all ϵ_i *vanish modulo* 11. *From this,* α *is updated to take the value* $\alpha + \epsilon = [0, 79, 112, 93, 119]$ *modulo* 11^2. *One can verify that this coincides modulo* 11^2 *with the values given in 10.*

8. Conclusion

A natural question is whether our approach can be used for ideals in more than two variables. As of now, several ingredients are missing: the known structure results are not as complete as Lazard's [46], and there is no known explicit description of Gröbner cells. Algorithmically, the key operation (reduction modulo an n-variate lexicographic Gröbner basis) seems to be a challenging problem in itself.

As already mentioned in the introduction, using our results in order to recover $\mathcal{G}$ itself, rather than $\mathcal{G}$ rem $\mathfrak{m}^K$, including in particular the quantification of bad maximal ideals $\mathfrak{m}$, is the subject of future work. Beyond this, the main algorithmic improvement we would like to achieve is reducing the overall cost so that it matches that of [38], in cases where both approaches are applicable. This would require several improvements in our algorithm, such as for instance improving the dense linear algebra we use to perform each step in Newton iteration.

Acknowledgments We thank our reviewer for their diligent reading and the helpful suggestions that improved the readability and general quality of this manuscript. Schost is supported by an NSERC Discovery Grant. St-Pierre thanks NSERC, the Alexander Graham Bell Canada Graduate Scholarship, and FQRNT for their funding.

References

[1] L. Alberti, B. Mourrain, and J. Wintz. Topology and arrangement computation of semi-algebraic planar curves. *Computer Aided Geometric Design*, 25(8):631–651, 2008. doi: 10.1016/j.cagd.2008.06.009. URL https://doi.org/10.1016/j.cagd.2008.06.009.

[2] E. A. Arnold. Modular algorithms for computing Gröbner bases. *J. Symb. Comp.*, 35(4): 403–419, 2003. doi: 10.1016/S0747-7171(02)00140-2. URL https://doi.org/10.1016/S0747-7171(02)00140-2.

[3] E. Berberich, P. Emeliyanenko, and M. Sagraloff. An elimination method for solving bivariate polynomial systems: Eliminating the usual drawbacks. In *ALENEX*, pages 35–47. SIAM, 2011. doi: 10.1137/1.9781611972917.4. URL https://doi.org/10.1137/1.9781611972917.4.

[4] J. Berthomieu and R. Lebreton. Relaxed p-adic Hensel lifting for algebraic systems. In *ISSAC'12*, pages 59–66. ACM, 2012. doi: 10.1145/2442829.2442842. URL https://dl.acm.org/doi/pdf/10.1145/2442829.2442842.

[5] J. Briançon. Description de Hilb^n $\mathbb{C}\{x, y\}$. *Inventiones Mathematicae*, 41:45–90, 1977.

[6] J. Briançon and A. Galligo. Déformations distinguées d'un point de $\mathbb{C}^2$ ou $\mathbb{R}^2$. In *Singularités à Cargèse*, number 7-8 in Astérisque, pages 129–138. Société mathématique de France, 1973.

[7] J. Canny, E. Kaltofen, and Y. Lakshman. Solving systems of non-linear polynomial equations faster. In *ISSAC'89*, pages 121–128. ACM, 1989. doi: 10.1145/74540.74556. URL https://dl.acm.org/doi/pdf/10.1145/74540.74556.

[8] G. Carrà Ferro. Gröbner bases and Hilbert schemes. i. *Journal of Symbolic Computation*, 6(2):219–230, 1988. ISSN 0747-7171. doi: 10.1016/S0747-7171(88)80044-0. URL https://doi.org/10.1016/S0747-7171(88)80044-0.

[9] A. Conca and G. Valla. Canonical Hilbert-Burch matrices for ideals of $k[x, y]$. *Michigan Mathematical Journal*, 57:157 – 172, 2008. doi: 10.1307/mmj/1220879402. URL https://doi.org/10.1307/mmj/1220879402.

[10] X. Dahan. Lexicographic Gröbner bases of bivariate polynomials modulo a univariate one. *Journal of Symbolic Computation*, 110:24–65, 2022. ISSN 0747-7171. doi: 10.1016/j.jsc.2021.10.001. URL https://doi.org/10.1016/j.jsc.2021.10.001.

[11] B. Dayton, T.-Y. Li, and Z. Zeng. Multiple zeros of nonlinear systems. *Mathematics of Computation*, 80(276):2143–2168, 2011.

[12] D. N. Diatta, S. Diatta, F. Rouillier, M.-F. Roy, and M. Sagraloff. Bounds for polynomials on algebraic numbers and application to curve topology, 2021.

[13] D. I. Diochnos, I. Z. Emiris, and E. P. Tsigaridas. On the asymptotic and practical complexity of solving bivariate systems over the reals. *J. Symb. Comput.*, 44(7):818–835, 2009. doi: 10.1016/j.jsc.2008.04.009. URL https://doi.org/10.1016/j.jsc.2008.04.009.

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

[15] David Eisenbud. *Commutative algebra: with a view toward algebraic geometry*, volume 150 of *GTM*. Springer, 2013.

[16] G. Ellingsrud and S. Strømme. On the homology of the Hilbert scheme of points in the plane. *Inventiones Mathematicae*, 87:343–352, 1987. doi: 10.1007/BF01389419. URL https://doi.org/10.1007/BF01389419.

[17] P. Emeliyanenko and M. Sagraloff. On the complexity of solving a bivariate polynomial system. In *ISSAC'12*, pages 154–161. ACM, 2012. doi: 10.1145/2442829.2442854. URL https://doi.org/10.1145/2442829.2442854.

[18] I. Z. Emiris and E. P. Tsigaridas. Real solving of bivariate polynomial systems. In *CASC*, pages 150–161. Springer, 2005. doi: 10.1007/11555964_13. URL https://doi.org/10.1007/11555964_13.

[19] 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 *LNCS*, pages 205–231. Springer, 1995.

[20] M. Giusti, K. Hägele, J. Heintz, J.-E. Morais, J.-L. Montaña, and L.-M. Pardo. Lower bounds for diophantine approximation. *J. of Pure and Applied Algebra*, 117/118:277–317, 1997. doi: 10.1016/S0022-4049(97)00015-7. URL https://doi.org/10.1016/S0022-4049(97)00015-7.

[21] M. Giusti, J. Heintz, J.-E. Morais, J. Morgenstern, and L.-M. Pardo. Straight-line programs in geometric elimination theory. *Journal of Pure and Applied Algebra*, 124: 101–146, 1998. doi: 10.1016/S0022-4049(96)00099-0. URL https://doi.org/10.1016/S0022-4049(96)00099-0.

[22] M. Giusti, G. Lecerf, and B. Salvy. A Gröbner-free alternative for polynomial system solving. *Journal of Complexity*, 17(1):154–211, 2001. doi: 10.1006/jcom.2000.0571. URL https://doi.org/10.1006/jcom.2000.0571.

[23] L. González-Vega and M. El Kahoui. An improved upper complexity bound for the topology computation of a real algebraic plane curve. *Journal of Complexity*, 12(4):527 – 544, 1996. doi: 10.1006/jcom.1996.0032. URL https://doi.org/10.1006/jcom.1996.0032.

[24] Mark Haiman. t,q-Catalan numbers and the Hilbert scheme. *Discrete Math.*, 193 (1-3):201–224, 1998. ISSN 0012-365X,1872-681X. doi: 10.1016/S0012-365X(98)00141-1. URL https://doi.org/10.1016/S0012-365X(98)00141-1. Selected papers in honor of Adriano Garsia (Taormina, 1994).

[25] J. Hauenstein, B. Mourrain, and A. Szanto. On deflation and multiplicity structure. *Journal of Symbolic Computation*, 83:228–253, 2017. doi: 10.1016/j.jsc.2016.11.013. URL https://doi.org/10.1016/j.jsc.2016.11.013. Special issue on the conference ISSAC 2015: Symbolic computation and computer algebra.

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

[27] J. van der Hoeven and R. Larrieu. Fast Gröbner basis computation and polynomial reduction for generic bivariate ideals. *AAECC*, 30(6):509–539, 2019. doi: 10.1007/s00200-019-00389-9. URL https://doi.org/10.1007/s00200-019-00389-9.

[28] Mark Huibregtse. On Ellingsrud and Strømme's cell decomposition of $\mathrm{Hilb}^n_{A^2_{\mathbb{C}}}$. *Invent. Math.*, 160(1):165–172, 2005. doi: 10.1007/s00222-004-0409-9. URL https://doi.org/10.1007/s00222-004-0409-9.

[29] Mark E. Huibregtse. A description of certain affine open subschemes that form an open covering of $\mathrm{Hilb}^n_{\mathbf{A}^2_k}$. *Pacific J. Math.*, 204(1):97–143, 2002. doi: 10.2140/pjm.2002.204.97. URL http://dx.doi.org/10.2140/pjm.2002.204.97.

[30] S. G. Hyun, S. Melczer, É. Schost, and C. St-Pierre. Change of basis for $\mathfrak{m}$-primary ideals in one and two variables. In *ISSAC'19*, pages 227–234. ACM Press, 2019. doi: 10.1145/3326229.3326268. URL https://dl.acm.org/doi/10.1145/3326229.3326268.

[31] A. Iarrobino. *Punctual Hilbert Schemes*. Number 188 in Memoirs of the American Mathematical Society. American Mathematical Society, 1977.

[32] O. H Ibarra, S. Moran, and R. Hui. A generalization of the fast LUP matrix decomposition algorithm and applications. *J. Algorithms*, 3(1):45–56, 1982. doi: 10.1016/0196-6774(82)90007-4. URL https://doi.org/10.1016/0196-6774(82)90007-4.

[33] C.-P. Jeannerod. LSP matrix decomposition revisited. 2006.

[34] K. Kedlaya and C. Umans. Fast polynomial factorization and modular composition. *SIAM Journal on Computing*, 40(6):1767–1802, 2011. doi: 10.1137/08073408X. URL https://doi.org/10.1137/08073408X.

[35] A. Kobel and M. Sagraloff. Improved complexity bounds for computing with planar algebraic curves. *CoRR*, abs/1401.5690, 2014.

[36] A. Kobel and M. Sagraloff. On the complexity of computing with planar algebraic curves. *Journal of Complexity*, 31(2):206–236, 2015. ISSN 0885-064X. doi: https://doi.org/10.1016/j.jco.2014.08.002.

[37] D. Lazard. Ideal bases and primary decomposition: case of two variables. *J. Symbolic Comput.*, 1(3):261–270, 1985. doi: 10.1016/S0747-7171(85)80035-3. URL https://doi.org/10.1016/S0747-7171(85)80035-3.

[38] R. Lebreton, E. Mehrabi, and É. Schost. On the complexity of solving bivariate systems: the case of non-singular solutions. In *ISSAC'13*, pages 251–258. ACM, 2013. doi: 10.1145/2465506.2465950. URL https://doi.org/10.1145/2465506.2465950.

[39] G. Lecerf. Quadratic Newton iteration for systems with multiplicity. *Foundations of Computational Mathematics*, 2(3):247–293, 2002. doi: 10.1007/s102080010026. URL https://doi.org/10.1007/s102080010026.

[40] Mathias Lederer. Gröbner strata in the Hilbert scheme of points. *J. Commut. Algebra*, 3(3):349–404, 2011. ISSN 1939-0807,1939-2346. doi: 10.1216/JCA-2011-3-3-349. URL https://doi.org/10.1216/JCA-2011-3-3-349.

[41] P. Lella and M. Roggero. Rational components of Hilbert schemes. *Rend. Semin. Mat. Univ. Padova*, 126:11–45, 2011. doi: 10.4171/RSMUP/126-2. URL https://doi.org/10.4171/RSMUP/126-2.

[42] A. Leykin, J. Verschelde, and A. Zhao. Newton's method with deflation for isolated singularities of polynomial systems. *Theoretical Computer Science*, 359(1):111–122, 2006. doi: https://doi.org/10.1016/j.tcs.2006.02.018.

[43] A. Leykin, J. Verschelde, and A. Zhao. Higher-order deflation for polynomial systems with isolated singular solutions. In *Algorithms in algebraic geometry*, pages 79–97. Springer, 2008.

[44] X. Li, M. Moreno Maza, and É. Schost. Fast arithmetic for triangular sets: from theory to practice. *J. Symb. Comp.*, 44(7):891–907, 2009. doi: 10.1016/j.jsc.2008.04.019. URL https://doi.org/10.1016/j.jsc.2008.04.019.

[45] A. Mantzaflaris and B. Mourrain. Deflation and certified isolation of singular zeros of polynomial systems. In *ISSAC'11*, page 249–256. ACM Press, 2011. doi: 10.1145/1993886.1993925. URL https://dl.acm.org/doi/abs/10.1145/1993886.1993925.

[46] M. G. Marinari and T. Mora. Cerlienco-Mureddu correspondence and Lazard structural theorem. *Investigación Operacional*, 27(1):75–98, 2013.

[47] E. Mehrabi and É. Schost. A softly optimal Monte Carlo algorithm for solving bivariate polynomial systems over the integers. *Journal of Complexity*, 34:78–128, 2016. ISSN 0885-064X. doi: 10.1016/j.jco.2015.11.009. URL https://doi.org/10.1016/j.jco.2015.11.009.

[48] V. Neiger, B. Salvy, É. Schost, and G. Villard. Faster modular composition, 2021. URL https://arxiv.org/abs/2110.08354.

[49] R Notari and M. L. Spreafico. A stratification of Hilbert schemes by initial ideals and applications. *Manuscripta Mathematica*, 101:429–448, 2000. doi: 0.1007/s002290050225. URL https://doi.org/10.1007/s002290050225.

[50] T. Ojika, S. Watanabe, and T. Mitsui. Deflation algorithm for the multiple roots of a system of nonlinear equations. *Journal of Mathematical Analysis and Applications*, 96(2):463–479, 1983. ISSN 0022-247X. doi: 10.1016/0022-247X(83)90055-0. URL https://doi.org/10.1016/0022-247X(83)90055-0.

[51] F. Pauer. On lucky ideals for Gröbner basis computations. *J. Symb. Comp.*, 14(5):471–482, 1992. doi: 10.1016/0747-7171(92)90018-Y. URL https://doi.org/10.1016/0747-7171(92)90018-Y.

[52] S. Pope and A. Szanto. Nearest multivariate system with given root multiplicities. *Journal of Symbolic Computation*, 44(6):606–625, 2009. ISSN 0747-7171. doi: 10.1016/j.jsc.2008.03.005. URL https://doi.org/10.1016/j.jsc.2008.03.005.

[53] L. Robbiano. On border basis and Gröbner basis schemes. *Collectanea Mathematica*, 60:11–25, 2009. doi: 10.1007/BF03191213. URL https://doi.org/10.1007/BF03191213.

[54] F. Rouillier. Solving zero-dimensional systems through the Rational Univariate Representation. *Applicable Algebra in Engineering, Communication and Computing*, 9(5):433–461, 1999. doi: 10.1007/s002000050114. URL https://doi.org/10.1007/s002000050114.

[55] F. Rouillier. On solving systems of bivariate polynomials. In *ICMS*, volume 6327 of *Lecture Notes in Computer Science*, pages 100–104. Springer, 2010. doi: 10.1007/978-3-642-15582-6_21. URL https://doi.org/10.1007/978-3-642-15582-6_21.

[56] É. Schost. Computing parametric geometric resolutions. *Applicable Algebra in Engineering, Communication and Computing*, 13(5):349–393, 2003. doi: 10.1007/s00200-002-0109-x. URL https://doi.org/10.1007/s00200-002-0109-x.

[57] É. Schost and C. St-Pierre. p-adic algorithm for bivariate Gröbner bases. In *ISSAC '23*, page 508–516. ACM Press, 2023. doi: 10.1145/3597066.3597086. URL https://doi.org/10.1145/3597066.3597086.

[58] W. Trinks. On improving approximate results of Buchberger's algorithm by Newton's method. *SIGSAM Bull.*, 18(3):7–11, 1984. doi: 10.1145/1089389.1089392. URL https://doi.org/10.1145/1089389.1089392.

[59] J. van der Hoeven and R. Larrieu. Fast reduction of bivariate polynomials with respect to sufficiently regular Gröbner bases. In *ISSAC '18*, page 199–206. ACM Press, 2018. doi: 10.1145/3208976.3209003. URL https://doi.org/10.1145/3208976.3209003.

[60] J. van der Hoeven and É. Schost. Multi-point evaluation in higher dimensions. *Applicable Algebra in Engineering, Communication and Computing*, 24(1):37–52, 2013. doi: 10.1007/s00200-012-0179-3. URL https://doi.org/10.1007/s00200-012-0179-3.

[61] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, third edition, 2013.

[62] F. Winkler. A p-adic approach to the computation of Gröbner bases. *J. Symb. Comput.*, 6(2/3):287–304, 1988. doi: 10.1016/S0747-7171(88)80049-X. URL https://doi.org/10.1145/1089389.1089392.

[63] X. Wu and L. Zhi. Determining singular solutions of polynomial systems via symbolic–numeric reduction to geometric involutive forms. *Journal of Symbolic Computation*, 47(3):227–238, 2012. ISSN 0747-7171. doi: https://doi.org/10.1016/j.jsc.2011.10.001.

[64] N. Yamamoto. Regularization of solutions of nonlinear equations with singular Jacobian matrices. *J. Infor. Processing*, 7:16–21, 1984.