A quasi-optimal Monte Carlo algorithm for the symbolic solution of polynomial systems in $\mathbb{Z}[X, Y]$

Esmaeil Mehrabi Éric Schost
emehrab@uwo.ca eschost@uwo.ca
Computer Science Department,
Western University, London, ON, Canada

February 22, 2015

Abstract

We give an algorithm for the symbolic solution of polynomial systems in $\mathbb{Z}[X, Y]$. Following previous work with Lebreton, we use a combination of lifting and modular composition techniques, relying in particular on Kedlaya and Umans' recent quasi-linear time modular composition algorithm.

The main contribution in this paper is an adaptation of a deflation algorithm of Lecerf's, that allows us to treat singular solutions for essentially the same cost as the regular ones. Altogether, for an input system with degree d and coefficients of bit-size h, we obtain Monte Carlo algorithms that achieve probability of success at least $1 - 1/2^{\mathcal{P}}$, with running time $d^{2+\varepsilon}O^{\sim}(d^2 + dh + d\mathcal{P} + \mathcal{P}^2)$ bit operations, for any $\varepsilon > 0$.

1 Introduction

Overview. Newton iteration is one of the most popular components of polynomial system solvers, from either the numeric or symbolic points of view. The usual version of this procedure handles situations without multiplicities only, since it requires that the Jacobian matrix of the given system be invertible at the roots we are looking for. To handle singular roots, various forms of *deflation techniques* have been developed (we will review some of them below).

In this paper, we are interested in applying such techniques to the symbolic solution of bivariate polynomial systems $F = G = 0$, with F and G in $\mathbb{Z}[X, Y]$. This is in the continuation of previous work with Lebreton [29], where Newton iteration techniques were used to handle solutions without multiplicities of the system $F = G = 0$. In this work, using results and ideas from [29], as well as Lecerf's deflation algorithm [30], we extend this approach to all solutions.

Motivated by applications to computational topology or computer graphics, recent years have witnessed the publication of a large body of work on bivariate systems. While some algorithms rely mostly on numerical techniques such as subdivision [1], many recent results involve symbolic elimination techniques, possibly in combination with real or complex root isolation [22, 16, 14, 43, 3, 15, 6, 5, 24]; we will discuss some these results further below.

Our interest here is on complexity of the symbolic side of such algorithms. In a nutshell, our main result says that bivariate systems with integer coefficients can be solved "symbolically" in essentially optimal time by Monte Carlo algorithms.

Over an arbitrary field. Let us first discuss known results for solving a bivariate system $F = G = 0$ over $\mathbb{K}[X, Y]$, where $\mathbb{K}$ is an arbitrary perfect field. Suppose that the zero-set $V(F, G)$ of F and G in an algebraic closure $\overline{\mathbb{K}}$ of $\mathbb{K}$ is finite. In this case, if F and G have total degree at most d, the Bézout theorem implies that the system $F = G = 0$ has at most d^2 solutions.

Several approaches exist to describe the solutions of our system: Gröbner bases, triangular representations, or descriptions based on univariate polynomials. For instance, in [29], together with Lebreton, we relied on a canonical description of a zero-dimensional variety, called the equiprojectable decomposition [11], using triangular sets; this is close to the decomposition based on subresultant calculations used in [22], but as a geometric notion, it does not take into account multiplicities in the input system.

Although it would be natural to use this kind of description here as well, the techniques we rely on are slightly easier to apply when working in generic coordinates. Indeed, if we are in generic coordinates, the zeros of $F = G = 0$ can simply be described by a pair of polynomials in $\mathbb{K}[X]$, of the form

$$P(X) = 0, \quad Y = S(X). \tag{1}$$

Remark that for such a description to make sense, no two points on $V(F, G)$ should have the same abscissa; this is precisely what is ensured once we are in generic coordinates. In such an output, our choice is to take P squarefree; in other words, our representation of the solutions does not reflect multiplicities (as a matter of fact, if $\mathbb{K}$ were not perfect, P could still have multiple roots in $\overline{\mathbb{K}}$ while being squarefree in $\mathbb{K}[X]$).

The input polynomials F and G have degree d in two variables; the polynomials P and S have degree at most d^2 in one variable. Thus, representing both input and output involves only $O(d^2)$ elements in $\mathbb{K}$. One would then naturally hope that P and S could be computed within $O^{\sim}(d^2)$ operations in $\mathbb{K}$, where the $O^{\sim}(\ )$ notation omits polylogarithmic factors.

However, no such result is known; the very close problem of computing the resultant of F and G using $O^{\sim}(d^2)$ operations is given as a research problem in [18, Problem 11.11]. For the latter resultant problem, the best algorithm known so far [41] uses $O^{\sim}(d^3)$ operations in $\mathbb{K}$.

Systems over the integers. In this paper, we are going to work in the particular case where $\mathbb{K} = \mathbb{Q}$. In such cases, it becomes crucial to take into account the bit-size of the input

and output as well; cost estimates will then be given in a boolean model (explicitly, a RAM with logarithmic cost).

For a nonzero integer a, we write $\mathrm{len}(a) = \lceil \log(|a|) \rceil$, and we call this the *length* of a; for $a = 0$, we write $\mathrm{len}(0) = 1$. This quantity essentially represents the amount of bits needed to store a (one may also work with the *height* of a, written $\mathrm{ht}(a) = \log(|a|)$, but the fact that $\mathrm{len}(a)$ takes integer values will be useful to us). It will be convenient to introduce a notion of length for polynomials with coefficients in $\mathbb{Q}$ as well: if P is such a polynomial, the length $\mathrm{len}(P)$ denotes the maximum of the lengths $\mathrm{len}(d)$ and $\mathrm{len}(n_i)_{i\in I}$, where d is a minimal common denominator for all coefficients of P and $(n_i)_{i\in I}$ are the coefficients of dP (which are integers). Thus, degree and length combined give us an upper bound on the total amount of bits, or machine words, needed to store P.

Suppose then that F and G have coefficients in $\mathbb{Z}$, with degree at most d and length at most h. Assuming that F and G have no nontrivial factor in $\mathbb{Q}[X, Y]$ and that we are in generic coordinates, so that a representation of the solutions of $F = G = 0$ as in (1) makes sense, both P and S having coefficients in $\mathbb{Q}$. As is well-known (since at least [2, 42]), the bit-size bounds for the coefficients of S are much worse than those for P, and this is reflected in practice very accurately. Explicitly, the following results are known (we will reprove them):

$$\mathrm{len}(P) = \tilde{O}(dh + d^2), \quad \mathrm{len}(S) = \tilde{O}(d^3h + d^4).$$

The usual workaround is to replace S by another polynomial R, defined as $R = P'S \bmod P$; equivalently, the solutions are now described by

$$P(X) = 0, \quad Y = \frac{R(X)}{P'(X)}.$$

This construction was highlighted in [2, 42], but goes back to early work of Kronecker [26] and Macaulay [33]. For the polynomial R, much better length bounds are known, of the form $\mathrm{len}(R) = \tilde{O}(dh + d^2)$. Thus, representing (P, R) involves $\tilde{O}(d^3h + d^4)$ bits; a similar construction for triangular representations is in [12].

Following Rouillier [42], we will call this representation the Rational Univariate Representation of $V(F, G)$. Rouillier's definition is more general, in that it allows one to take multiplicities into account, working with ideals rather than varieties: in Rouillier's terminology, P and R as defined above would be called a Rational Univariate Representation of the radical of the ideal $\langle P, Q\rangle$; we will stick to the slightly more compact terminology above.

Let us finally say a word about generic position questions. As was mentioned above, our requirement for the existence of an output such as polynomials (P, S) or (P, R) is that the coordinates X *separates* the points in $V(F, G)$, that is, that any two distinct points in $V(F, G)$ have different abscissas. A change of variables of the form $X \leftarrow X + tY$ will ensure that this is the case, for almost all values of $t \in \mathbb{Z}$ (that is, all values except a finite number).

Main result. In all that follows, we will say that a solution (x, y) of the system $F = G = 0$ is *simple* if the Jacobian determinant of (F, G) is nonzero at (x, y).

Let us denote by Z the set of such simple solutions. In [29], we gave with Lebreton an algorithm to compute a triangular representation of Z; this algorithm could be adapted to give a univariate representation as we do here, after putting the equations in generic coordinates.

In a nutshell, the idea of that algorithm is to compute the output modulo a prime p, then *lift* this representation modulo powers of p using a suitable form of Newton iteration. Looking only at points in Z makes it straightforward to apply such techniques, since by assumption, at such points, the Jacobian matrix of (F, G) is invertible. One of the two main results of [29] was that for any $\varepsilon > 0$, one could compute a description of Z by means of triangular sets using $d^{3+\varepsilon}O^{\tilde{}}(d + h)$ bit operations, by a Monte Carlo algorithm, with probability of success greater than $1/2$.

In this paper, we show that that we can extend these ideas to find a univariate representation of the whole $V(F, G)$, with a running time that matches the results of [29] in the case where all solutions of $F = G = 0$ are simple. As in [29], the algorithm uses a modification of the Kedlaya-Umans modular composition algorithm [23], the new ingredient being a deflation algorithm by Lecerf [30] to handle multiple roots.

Our algorithm is probabilistic of the Monte Carlo kind: one can choose an arbitrary threshold, say $1/2^{\mathcal{P}}$, and the algorithm guarantees that the result is correct with probability at least $1 - 1/2^{\mathcal{P}}$. Part of the randomness simply amounts to choosing an integer in a finite set. Another component is more involved, as it amounts to choosing primes. Since this is a delicate question in itself, and not the topic of this paper, we will use the following device: we assume that we are given an oracle $\mathscr{O}$, which takes as input an integer B, and returns a prime number in $\{B + 1, \ldots, 2B\}$, uniformly distributed within this set of primes.

Theorem 1. *Let $F, G \in \mathbb{Z}[X, Y]$ with degree at most d and length at most h, that have no nontrivial factor in $\mathbb{Q}[X, Y]$.*

For any $\varepsilon > 0$, there exists an algorithm with the following characteristics. Given $\mathcal{P} \geq 1$, the algorithm computes the Rational Univariate Representation of the system $F_t = G_t = 0$, where t is an integer of length $O(\log(d))$, $F_t = F(X + tY, Y)$ and $G_t = G(X + tY, Y)$. The running time is $d^{2+\varepsilon}O^{\tilde{}}(d^2 + dh + d\mathcal{P} + \mathcal{P}^2)$ bit operations, and the probability of success is at least $1 - 1/2^{\mathcal{P}}$.

The algorithm makes two calls to the oracle $\mathscr{O}$, with input integers in $(2^{\mathcal{P}}dh)^{O(1)}$.

From such an output, using the techniques of [40], one may then compute a triangular decomposition of $V(F, G)$, or compute a Rational Univariate Representation of the system $F_{t'} = G_{t'} = 0$ for $t' \in \mathbb{Z}$ of small height; we leave the details to the reader.

For fixed $\mathcal{P}$, the running time of our algorithm is $d^{3+\varepsilon}O^{\tilde{}}(d + h)$ bit operations, which almost matches the known upper bounds on the output size. It is worth noting that these upper bounds are actually sharp, up to polylogarithmic terms. Consider the system

$$F^{(d)} = \prod_{i=1}^{d}(X - i), \quad G^{(d)} = \prod_{j=1}^{d}(Y - j);$$

these polynomials have degree d and length $\Theta^{\tilde{}}(d)$, where the $\Theta^{\tilde{}}(\ )$ notation is inspired by the $O^{\tilde{}}$ one, and also indicates the omission of polylogarithmic factors.

For any $t \in \mathbb{Z}$ such that the integers $\{i - tj\}_{1 \le i,j \le d}$ are pairwise distinct, the first polynomial in the Rational Univariate Representation of the system $F_t^{(d)} = G_t^{(d)} = 0$ is precisely $R^{(d)} = \prod_{1 \le i,j \le d}(X - i + tj)$.

If we take t of length $O(\log(d))$ as in the theorem, one sees that $R^{(d)}$ has degree d^2 and length $\tilde{\Theta}(d^2)$. More precisely, if we assume that $t < 0$, so all roots of $R^{(d)}$ are pairwise distinct positive integers, we claim that the sum of the lengths of the coefficients of $R^{(d)}$ is $\tilde{\Theta}(d^4)$. The upper bound follows from the previous remarks; to obtain the lower bound, note that this sum is greater than the sum of the lengths of the coefficients of $(X+1)^{d^2}$, which is $\tilde{\Theta}(d^4)$, see for instance [27]. Thus, simply writing the output for such examples requires $\tilde{\Theta}(d^4)$ bit operations.

As mentioned above, there exist a host of algorithms for bivariate systems, some of them mixing symbolic and numerical techniques. On the symbolic side, many previous works, starting in particular from [22], rely on resultant and subresultant calculations, or closely related triangular decomposition algorithms, as in [7]. The reader will find a detailed review of previous work in [24] (which is one of those references that involve symbolic-numerical techniques, relying on root isolation ideas).

For the "symbolic" problem we consider, we are not aware of previous results that would be in the same complexity class as ours. To our knowledge the best deterministic algorithm is from [5], with cost $\tilde{O}(d^6h + d^7)$; a Las Vegas version of the same algorithm has cost $\tilde{O}(d^4h + d^5)$.

It is relatively easy to write a Monte Carlo algorithm that would run in time $\tilde{O}(d^4h+d^5)$: pick many small primes ($O(dh+d^2)$ of them) or a large one (of length $O(dh+d^2)$), and solve the system $F = G = 0$ in $\tilde{O}(d^3)$ operations modulo each of these primes. Our results show that when using Monte Carlo algorithms, we can almost save a further factor d.

Practical aspects and extensions. It remains a challenge to make the algorithms presented here competitive in practice. The main difficulty is that we rely on Kedlaya and Umans' algorithm for modular composition [23] and its bivariate extension by Poteaux and Schost [39]. Unfortunately, the constants hidden in the cost estimates of these algorithms make a direct implementation of these techniques slower than naive algorithms for inputs of realistic size; further work is needed to solve this issue.

In [29], we also gave with Lebreton an alternative result (still only for computing the solutions without multiplicities), that did not make use of Kedlaya-Umans and Poteaux-Schost's modular composition algorithm; instead, that result used an extension of Brent and Kung's modular composition algorithm [8]. For polynomials in $\mathbb{Z}[X,Y]$, the resulting cost was $\tilde{O}(d^{(\omega+5)/2}h + d^{(\omega+7)/2}) \subset \tilde{O}(d^{3.69}h + d^{4.69})$ bit operations, where ω is such that we can multiply $n \times n$ matrices using $O(n^\omega)$ ring operations, over any ring (the best known bound to date is $\omega < 2.38$ from [28]).

That result was not as good as the one based on Kedlaya and Umans' ideas, but had the advantage of being easy to implement, yielding an efficient practical solution. In addition, that algorithm admitted extensions to input systems in $k[T][X,Y]$ (for a field k) instead of $\mathbb{Z}[X,Y]$, with a similar running time (now counted in terms of operations in k); this is not

known to be possible for Kedlaya and Umans' algorithm so far.

In our situation, it remains possible to extend the general principle of our algorithm to inputs in $k[T][X,Y]$, but we were not able to apply ideas *à la* Brent and Kung to obtain the same result $O^{\sim}(d^{(\omega+5)/2}h + d^{(\omega+7)/2})$ as in [29]. Explicitly, the adaptation of the results in Subsection 4.1 to such a context, with a suitable complexity, still eludes us.

Multiplicities. Before giving an overview of our algorithm, it will be useful for us to recall the definition of the *multiplicity* of an isolated solution of a polynomial system. We will only need to discuss systems in one or two variables.

We assume that our polynomials have coefficients in an algebraically closed field $\overline{\mathbb{K}}$ (since this will include the case of polynomials with coefficients in any subfield $\mathbb{K}$ of $\overline{\mathbb{K}}$).

First, consider a nonzero univariate polynomial F in $\overline{\mathbb{K}}[X]$, and a root x of F. The *multiplicity* of F at x, sometimes written $\mu(F,x)$, is the largest integer M such that $(X-x)^M$ divides F. The multiplicity M is one if and only if $F'(x)$ is nonzero.

Next, consider an ideal ψ in $\overline{\mathbb{K}}[X,Y]$, and an isolated solution (x,y) of the system of equations $\psi = 0$. Define the ideal $\psi' = \{f(X+x, Y+y) \mid f \in \psi\}$, so that $(0,0)$ is a solution of the system $\psi' = 0$. Then, the multiplicity of the system ψ at (x,y), that we will denote by $\mu(\psi,(x,y))$, is the dimension of the $\overline{\mathbb{K}}$-vector space $\overline{\mathbb{K}}[[X,Y]]/\psi'$, see for instance [9, Chapter 4]; the fact that (x,y) is an isolated solution is equivalent to this dimension being finite. If $\psi = \langle F, G\rangle$, for some polynomials F, G in $\overline{\mathbb{K}}[X,Y]$, the multiplicity M is one if and only if the Jacobian determinant of (F,G) is nonzero at (x,y).

It will also be useful to remember the following extension of the Bézout bound on the number of isolated solutions of a bivariate system: if F, G are polynomials in $\overline{\mathbb{K}}[X,Y]$, with both F and G having degree at most d, then the sum of the multiplicities of the isolated solutions of the system $F = G = 0$ is at most d^2. Examples such as $F = X^d, G = Y^d$ show that multiplicities as large as d^2 are possible.

Overview of the algorithm. As was hinted at above, the basic idea of our algorithm is simple: we use modular techniques. A common way to put this idea to practice is straightforward: given input polynomials F and G in $\mathbb{Z}[X,Y]$, that we assume for simplicity to be in generic coordinates, we compute the Rational Univariate Representation of $V(F,G)$ modulo p_i for sufficiently many primes p_i, and recombine them by Chinese Remaindering.

The main idea in [29] was that using lifting techniques could result in better algorithms. Denoting the Rational Univariate Representation of $V(F,G)$ by $(P,R) \in \mathbb{Q}[X]^2$, this boils down to essentially computing (P,R) mod p for a prime p, then lifting these polynomials to (P,R) mod p^2, (P,R) mod $p^4, \ldots,$ (P,R) mod p^{2^κ} and eventually recover (P,R) by rational number reconstruction. The bulk of the computational effort is the lifting step, which is a form of Newton iteration; the basic approach follows previous work from [20, 21, 45], but the actual computation uses modular composition techniques derived from Kedlaya and Umans' algorithm. (Note that the lifting algorithm is actually slightly more involved, as the polynomial S from (1) also appears in the calculation, but the main idea is the same.)

However, the lifting algorithms from [20, 21, 45] require that all points (x,y) described

by the Rational Univariate Representation (P, R) are simple solutions of the input system (F, G); this is why the algorithm of [29] was restricted to such points. In order to handle all solutions, including the multiple ones, we need to employ a form of Newton iteration for multiple roots.

Our approach is based on a result of Lecerf's [30], which generalizes the usual Newton iterator to multiple roots, in a context of $\mathfrak{m}$-adic lifting, that is, modulo the powers of a maximal ideal $\mathfrak{m}$ in a domain $\mathbb{A}$. The main idea behind this approach is classical: it boils down to replacing the given polynomial system ψ with a new one, say $\tilde{\psi}$, such that for a given root (x, y) of ψ with multiplicity M, (x, y) is still a root of the new system $\tilde{\psi}$, but with multiplicity $\tilde{M} < M$. We can then find a new deflated system for which (x, y) is a non-singular root, by repeating the process sufficiently many times.

We are however not able to directly use the complexity results in [30]. Indeed, while they handle the general case of n-variate systems, these algorithms assume that the input system is given by means of a straight-line program, rather than the dense representation we use here — we could of course build a straight-line program representation from our dense polynomials in a naive manner, but the cost overhead would be too large. In addition, the running time given in [30] grows like the square of the multiplicity M of the root; this is too much for us to achieve the results claimed in Theorem 1, as we saw above that multiplicities as large as d^2 are possible for a bivariate system of degree d.

Our main technical contribution in this paper lies in the adaptation of Lecerf's algorithm to our context of lifting bivariate systems, with an admissible complexity. The fact that we only consider bivariate systems simplifies the description of the algorithm considerably.

Other deflation algorithms. Generalizing Newton iteration to singular situations, and in particular designing an efficient iterator with quadratic convergence in degenerate cases, are still research problems; we briefly review some of the previous work on this question. Remark that all algorithms below work for an arbitrary number of variables, not only bivariate systems.

An early result in this area is due to Ojika, Watanabe, and Mitsui [36]: by applying a triangulation preprocessing step on the Jacobian matrix at the approximate root, minors of the Jacobian matrix are added to the system to reduce the multiplicity.

In [31, 32], instead of triangulating the Jacobian matrix, the number of variables is doubled and new equations are introduced, which are linear in the new variables; it is proved that the multiplicity decreases through this process. In [13], this construction in related to Macaulay's *inverse systems*; Macaulay's dialytic method [33] is revisited for this purpose. These deflation methods are applied iteratively until the root becomes simple, increasing each time the number of variables.

Other algorithms for the construction of inverse systems are described e.g. in [34], reducing the size of the intermediate linear systems, or in [35] using an integration method. In [38], a minimization approach is used to reduce the value of the equations and their derivatives at the approximate root, assuming a basis of the inverse system is known. In [46], the inverse system is constructed via Macaulay's method; multiplication tables of the local algebras are

deduced and their eigenvalues are used to improve the approximate solution; the convergence of this process is quadratic when the Jacobian has co-rank one at the multiple root.

Unfortunately, even when the input system is bivariate, it seems difficult to control the complexity of the above algorithms. In addition, several of these results rely on purely numerical techniques, such as the Singular Value Decomposition, which will not carry over to our context.

Organization of the paper. Section 2 gives quantitative results on output size and primes of good reduction for various constructions – some of them being well-known. We continue in Section 3 with simple algorithms for calculations with reducible polynomials, and in Section 4 with some normal forms algorithms, which crucially rely on Kedlaya and Umans' modular composition algorithm.

Section 5 states a deflation result inspired by Lecerf's, which leads us to define a notion of *signature* attached to isolated zeros of a bivariate system; in Section 6, we give an algorithm to compute a partition of the zeros of a zero-dimensional bivariate system induced by this signature.

Section 7 shows how the previous constructions can be used to write a Newton iteration for multiple roots of a bivariate system (again inspired by Lecerf's work), and Section 8 gives the main algorithm and proves Theorem 1.

Acknowledgements. We wish to thank Yacine Bouzidi, Sylvain Lazard, Guillaume Moroz, Marc Pouget and Fabrice Rouillier for helpful discussions. This work was supported by NSERC and the Canada Research Chairs program.

2 Quantitative estimates

This section is devoted to first recall some classical properties of the length of polynomials, then use them to prove some basic quantitative estimates, such as on the size of the output of our algorithm, primes of "bad reduction" for various constructions, etc. The results in the first two subsections are hardly new; those of the next two sections are very much in the same spirit.

2.1 Length bounds

With regards to primes of bad reduction, the typical kind of statement we will need to make is of the form "there exists a nonzero integer A such that if a prime p does not divide A, ... (some desirable properties are guaranteed)". In all such cases, we will have to estimate the length of A, since such estimates are required in order to choose p that satisfies the non-divisibility condition with a prescribed probability.

Writing down these length estimates entirely explicitly is however rather tedious, error-prone, and hardly useful: for any practical purposes, the implementation itself should determine the bounds (foregoing in particular any simplifying overestimate one could be tempted

to do when writing these bounds, as for instance in [11, 10]). Following [7], we will thus mostly refrain from writing explicit bounds here; instead, we will give asymptotic estimates, and indicate how to derive the actual bounds.

The techniques we use are extremely classical, such as factor bounds, or bounds on determinants of polynomial matrices. Lemma 1.2 in [25] and the discussion that follows it provide the following results:

b_1. For $F_1, \ldots, F_s$ in $\mathbb{Z}[X_1, \ldots, X_n]$, all of degrees at most d and length at most h, and G in $\mathbb{Z}[Y_1, \ldots, Y_s]$, of degree at most e and length at most ℓ, $G(F_1, \ldots, F_s)$ has degree at most de and length at most $\ell + e(h + \mathrm{len}(s+1) + \mathrm{len}(n+1)d)$. For fixed s and n, this is $O(\ell + e(h+d))$.

b_2. For F in $\mathbb{Z}[X, Y]$ of degree d and length h, any factor of F in $\mathbb{Z}[X, Y]$ has length at most $h + 3d = O(h + d)$.

b_3. If M is an $n \times n$ matrix with entries in $\mathbb{Z}[X, Y]$ of degree at most d and length at most h, $\det(M)$ has degree at most nd and length at most $n(h+2d+\mathrm{len}(n)) = O\tilde{}(n(h+d))$.

In most cases where we apply these results, we will typically handle input polynomials F, G of degree d and length h, with resulting bounds in $(dh)^{O(1)}$. Often, writing polynomial bounds as $(dh)^{O(1)}$ above will be sufficient, but in some cases such as Corollary 1 below, we will write down the actual exponents.

Although we do not write explicit bounds, we said above that our main algorithm will need to be able to evaluate them, in order to be able to choose prime numbers that satisfy some luckiness properties with prescribed probability. In all that follows, a function of the form $\Delta : \mathbb{N}^k \to \mathbb{N}$, for some integer k, that can be computed in time $\log(\Delta)^{O(1)}$, will be called *efficiently computable*. All bounds we will obtain by applications of b_1, b_2 and b_3 will be efficiently computable, since this is the case for all expressions in b_1, b_2 and b_3 (this is why we introduced the length of integers as a function taking integer values); for example, all bounds of the form $(dh)^{O(1)}$ as above will be computable in $\log(dh)^{O(1)}$ bit operations. Thus, the costs incurred by these calculations will be negligible compared to all other ones.

2.2 Polynomials in general position

In this subsection, we describe a classical notion of system in *general position*, and we discuss conditions that ensure that this property is preserved through reduction at a prime. These results are classical (they go back to Kronecker and Macaulay), and their quantitative versions appear for instance in [42, 25, 45, 11, 14, 6], among many other references. Nevertheless, we give self-contained proofs of the facts we need, as we did not find the exact statements we needed in the literature. The main result in this subsection is Corollary 1 below.

In all this section, π denotes the mapping $(x, y) \mapsto x$ of projection on the first factor; although the points x, y will be taken in various fields, we keep the same notation throughout, since no ambiguity can arise. In the beginning of this section, $\mathbb{A}$ is a domain with field of fractions $\mathbb{K}$; we let $\overline{\mathbb{K}}$ denote an algebraic closure of $\mathbb{K}$.

Representing zero-dimensional algebraic sets. Let $V \subset \overline{\mathbb{K}}^2$ be a finite set, and assume that V can be written as $V = V(F_1, \ldots, F_t)$ for some $F_1, \ldots, F_t$ in $\mathbb{K}[X, Y]$. Suppose that the following conditions are satisfied:

G_1. $\mathbb{K}$ is perfect;

G_2. X is a *separating element* for V, that is, the restriction of π to V is one-to-one.

Under these assumptions, there exist uniquely defined polynomials (P, S) in $\mathbb{K}[X]$, with P monic and squarefree (in $\mathbb{K}[X]$, or equivalently in $\overline{\mathbb{K}}[X]$, under our perfectness assumption), and S of degree less than $\deg(P)$, such that the ideal $\langle P, Y - S\rangle$ is the defining ideal of V in $\overline{\mathbb{K}}[X, Y]$.

Following [19], we call polynomials (P, S) the *Shape Lemma representation* of V, and denote them by $(P, S) = \mathsf{SL}(V)$. Over a field such as $\mathbb{K} = \mathbb{Q}$, it is well known that this representation suffers from coefficient size bloat [2, 42]: the coefficients of S are in many cases significantly larger than those of P. A workaround is to use an alternative description, the *Rational Univariate Representation* of V, for which this issue usually disappears. It consists in polynomials (P, R), with $R = SP' \bmod P \in \mathbb{K}[X]$; we denote these polynomials by $(P, R) = \mathsf{RUR}(V)$. One can always deduce $\mathsf{RUR}(V)$ from $\mathsf{SL}(V)$; since we took P squarefree over a perfect field, P' is a unit modulo P, so one can conversely deduce $\mathsf{SL}(V)$ from $\mathsf{RUR}(V)$.

As mentioned in the introduction, the term "Rational Univariate Representation" is from [42]; the original definition is able to incorporate multiplicities, which we do not take into consideration here.

Polynomials in general position. Recall that $\mathbb{A}$ is our domain, with fraction field $\mathbb{K}$. Let then F and G be in $\mathbb{A}[X, Y]$ and let $V = V(F, G) \subset \overline{\mathbb{K}}^2$. We say that F, G are in *general position* if the following holds:

H_1. F and G have no common factor in $\mathbb{K}[X, Y]$, so V is finite;

H_2. The leading coefficients f and g of respectively F and G with respect to Y are in $\mathbb{A}$;

H_3. V satisfies G_1 and G_2.

When this is the case, by H_3, the polynomials P, S and R associated to V as above are well-defined. H_2 then implies that the polynomial P appearing in the Shape Lemma representation of V is the squarefree part of the resultant of F and G with respect to Y (once made monic in X). As a matter of notation, when F, G are in general position, we will write $(P, S) = \mathsf{SL}(F, G)$ and $(P, R) = \mathsf{RUR}(F, G)$.

The following shearing operation is the usual device used to put polynomials in general position. For t in $\mathbb{A}$, we will denote by F_t and G_t the polynomials $F_t = F(X + tY, Y)$ and $G_t = G(X + tY, Y)$; similarly, we will write $V_t = V(F_t, G_t)$, so that

$$V_t = \{(x, y) \in \overline{\mathbb{K}}^2 \mid (x + ty, y) \in V\} = \phi_t(V),$$

where ϕ_t is the mapping $\overline{\mathbb{K}}^2 \to \overline{\mathbb{K}}^2$ given by $\phi_t(x, y) = (x - ty, y)$. Letting T be an indeterminate over $\mathbb{A}$, we use the same notation, using a subscript T instead of t, to denote the polynomials

$$F_T = F(X + TY, Y) \quad \text{and} \quad G_T = G(X + TY, Y),$$

and their zero-set V_T in $\overline{\mathbb{K}(T)}^2$; remark that V_T actually lies in $\overline{\mathbb{K}}(T)^2 \subset \overline{\mathbb{K}(T)}^2$. If F and G are polynomials in $\mathbb{A}[X, Y]$, with no common factor in $\mathbb{K}[X, Y]$ (so they satisfy H_1), one easily verifies that over the ring $\mathbb{A}[T]$ instead of $\mathbb{A}$, F_T and G_T satisfy H_1 and H_2, V_T satisfies G_2, and V_T has the same cardinality as V.

Over the integers. Let us now restrict our attention to the case $\mathbb{A} = \mathbb{Z}$ and $\mathbb{K} = \mathbb{Q}$; as before, we take F and G that satisfy H_1 and we write $V = V(F, G)$. The following construction is classical; see for instance [6, 7] for a very close presentation.

Let $\mathfrak{A}$ be the resultant of F_T and G_T with respect to Y; this is a nonzero polynomial in $\mathbb{Z}[T, X]$, and we denote by $\mathfrak{a} \in \mathbb{Z}[T]$ its leading coefficient with respect to X. Let next $\mathfrak{B} \in \mathbb{Z}[T, X]$ be the *squarefree part* of $\mathfrak{A}$, that is, $\mathfrak{B} = \mathfrak{A}/\gcd(\mathfrak{A}, \mathfrak{A}')$, where $\mathfrak{A}'$ is the derivative of $\mathfrak{A}$ with respect to X, and where the gcd is taken in the unique factorization domain $\mathbb{Z}[T, X]$. The gcd, and thus $\mathfrak{B}$ itself, are *a priori* defined only up to sign, but this will be inconsequential.

In what follows, the *content* of a polynomial with integer coefficients is the gcd of its coefficients (so it is defined up to sign). Gauss' lemma states that the content is multiplicative; this is well-known for univariate polynomials, and the multivariate case follows for instance by using Kronecker's substitution. A polynomial with unit content is called *primitive*.

Lemma 1. *The polynomial $\mathfrak{B}$ is primitive in $\mathbb{Z}[T, X]$. In $\overline{\mathbb{Q}}[T, X]$, $\mathfrak{B}$ factors as*

$$\mathfrak{B} = \mathfrak{b} \prod_{(x,y)\in V} (X - (x - Ty)), \tag{2}$$

where $\mathfrak{b}$ is the leading coefficient of $\mathfrak{B}$ with respect to X, belongs to $\mathbb{Z}$ and divides the content of $\mathfrak{a}$.

Proof. Let $(x_i, y_i)_{i\in I}$ be the coordinates of all points in V, and let $(m_i)_{i\in I}$ be their corresponding multiplicities, with respect to the ideal $\langle F, G\rangle$. Then, the resultant $\mathfrak{A}$ factors as

$$\mathfrak{A} = \mathfrak{a} \prod_{i\in I} (X - (x_i - Ty_i))^{m_i}.$$

Since $\mathfrak{A}$ is in $\mathbb{Z}[T, X]$ and $\mathfrak{a}$ in $\mathbb{Z}[T]$, we see in particular that the product $\mathfrak{A}^\star = \prod_{i\in I}(X - (x_i - Ty_i))^{m_i}$ is in $\mathbb{Q}[T, X]$.

For a given integer m, let V_m be the subset of V consisting of all those points of multiplicity m (so all V_m are empty, except finitely many of them). Then, the product $\prod_{(x,y)\in V_m}(X - (x - Ty))$ is in $\mathbb{Q}[T, X]$, since we can obtain as a factor in the squarefree factorization of $\mathfrak{A}^\star$. Since this product is monic in X, we can write it as

$$\prod_{(x,y)\in V_m} (X - (x - Ty)) = \frac{B_m}{\beta_m},$$

for some polynomial $B_m \in \mathbb{Z}[T, X]$ of content 1, and some integer β_m. As a consequence, we can write

$$\mathfrak{A} = \mathfrak{a} \prod_m \frac{B_m^m}{\beta_m^m}, \tag{3}$$

where the product involves only finitely many m's. Since all B_m's have content one, and $\mathfrak{A}$ has integer coefficients, we deduce in particular that the product of all β_m^m divides the content of $\mathfrak{a}$.

Since $\mathfrak{a}$ is in $\mathbb{Z}[T]$, we deduce from the expression of $\mathfrak{A}$ given above that $\mathfrak{B}$ can be written as

$$\mathfrak{B} = \prod_m B_m = \left(\prod_m \beta_m \right) \prod_{(x,y)\in V} (X - (x - yT)). \tag{4}$$

The polynomial $\mathfrak{B}$ is primitive because all B_m's are; we also deduce that $\mathfrak{b} = \prod_m \beta_m$, which divides $\mathfrak{a}$. This concludes the proof of the lemma. □

The following lemmas show how the polynomial $\mathfrak{B}$ and its factors allow us to give formulas for the Rational Univariate Representation of V_t and its subsets, when F_t and G_t are in general position. To state these lemmas, remark that if W if a subset of V, we may rewrite the factorization in (2) as

$$\mathfrak{B} = \mathfrak{b}\, \mathfrak{C}_W\, \mathfrak{C}_{W^c},$$

where we write $W^c = V \setminus W$ and

$$\mathfrak{C}_W = \prod_{(x,y)\in W} (X - (x - Ty)) \quad \text{and} \quad \mathfrak{C}_{W^c} = \prod_{(x,y)\in W^c} (X - (x - Ty)).$$

Lemma 2. *Let W be a subset of V defined over $\mathbb{Q}$. Then, $\mathfrak{C}_W$ is in $\mathbb{Q}[T, X]$ and it can be written as $\mathfrak{B}_W/\mathfrak{b}_W$, where $\mathfrak{B}_W$ is a primitive polynomial in $\mathbb{Z}[T, X]$ that divides $\mathfrak{B}$ in $\mathbb{Z}[T, X]$, and $\mathfrak{b}_W$ is a nonzero integer that divides $\mathfrak{b}$.*

Proof. The formula defining the polynomial $\mathfrak{C}_W$ shows that this polynomial is in $\overline{\mathbb{Q}}[T, X]$; to prove that it is in $\mathbb{Q}[T, X]$, it is then enough to prove that this polynomial lies in $\mathbb{Q}(T)[X]$. The latter claim is clear, since the monicity of $\mathfrak{C}_W$ with respect to X implies that it is the characteristic polynomial of the multiplication-by-X map in the coordinate ring $\mathbb{Q}(T)[W_T]$ of W_T, and W_T is defined over $\mathbb{Q}(T)$.

Next, we start from the factorization $\mathfrak{B} = \mathfrak{b}\, \mathfrak{C}_W\, \mathfrak{C}_{W^c}$, which holds between polynomials in $\mathbb{Q}[T, X]$. Since $\mathfrak{C}_W$ and $\mathfrak{C}_{W^c}$ are in $\mathbb{Q}[T, X]$, and monic in X, they can be written as $\mathfrak{C}_W = \mathfrak{B}_W/\mathfrak{b}_W$ and $\mathfrak{C}_{W^c} = \mathfrak{B}_{W^c}/\mathfrak{b}_{W^c}$, with $\mathfrak{b}_W$ and $\mathfrak{b}_{W^c}$ in $\mathbb{Z}$, and $\mathfrak{B}_W$ and $\mathfrak{B}_{W^c}$ primitive in $\mathbb{Z}[T, X]$. Clearing denominators, we obtain $\mathfrak{b}_W \mathfrak{b}_{W^c} \mathfrak{B} = \mathfrak{b}\, \mathfrak{B}_W \mathfrak{B}_{W^c}$. Using Gauss' Lemma over $\mathbb{Z}[T, X]$, we deduce that $\mathfrak{b}_W \mathfrak{b}_{W^c} = \mathfrak{b}$ and $\mathfrak{B}_W \mathfrak{B}_{W^c} = \mathfrak{B}$. □

Remark that the polynomial $\mathfrak{B}_W$ in the previous lemma is uniquely defined up to sign only; this will be harmless in what follows.

The explicit factorization the polynomials $\mathfrak{B}_W$ or $\mathfrak{C}_W$ allows us to give formulas for the Rational Univariate Representation of V, or of one of its subsets W.

Lemma 3. *Let $t \in \mathbb{Z}$ be such that F_t and G_t are in general position and let W be a subset of V, defined over $\mathbb{Q}$. Then, W_t satisfies G_1 and G_2, the associated Rational Univariate Representation $(P_{W_t}, R_{W_t}) = \mathsf{RUR}(W_t)$ is given by*

$$P_{W_t} = \frac{1}{\mathfrak{b}_W}\mathfrak{B}_W(t, X) \quad \text{and} \quad R_{W_t} = \frac{1}{\mathfrak{b}_W}\frac{\partial \mathfrak{B}_W}{\partial T}(t, X),$$

and the Shape Lemma Representation $(P_{W_t}, S_{W_t}) = \mathsf{SL}(W_t)$ is given by

$$P_{W_t} = \frac{1}{\mathfrak{b}_W}\mathfrak{B}_W(t, X) \quad \text{and} \quad S_{W_t} = \frac{\frac{\partial \mathfrak{B}_W}{\partial T}(t, X)}{\frac{\partial \mathfrak{B}_W}{\partial X}(t, X)} \mod P_{W_t}.$$

Proof. If F_t and G_t are in general position, V_t satisfies G_1 and G_2; it is then also the case for its subset W_t, which proves the first point. To prove the next points, we use the fact that $(P_{W_t}, R_{W_t}) = \mathsf{RUR}(W_t)$ is given by

$$P_{W_t} = \mathfrak{C}_W(t, X) \quad \text{and} \quad R_{W_t} = \frac{\partial \mathfrak{C}_W}{\partial T}(t, X);$$

this claim is classical (see for instance [2, 42, 45] in the recent literature, which actually apply in more general cases) and can also be recovered immediately from the definition of $\mathfrak{C}_W$. Using Lemma 2 then gives the assertions for $\mathsf{RUR}(W_t)$. Finally, to conclude for $\mathsf{SL}(W_t)$, one uses the fact that $R_{W_t} = S_{W_t}P'_{W_t} \mod P_{W_t}$. □

Let now $\Delta \in \mathbb{Z}[T]$ be the discriminant of $\mathfrak{B}$ with respect to X and define finally Γ as the product of Δ by the leading coefficients $\mathfrak{f}$ and $\mathfrak{g}$ of respectively F_T and G_T with respect to Y, and by the leading coefficient $\mathfrak{a}$ of $\mathfrak{A}$ in X. Because F_T and G_T satisfy H_2, $\mathfrak{f}$ and $\mathfrak{g}$ are in $\mathbb{Z}[T]$, so Δ is a nonzero element of $\mathbb{Z}[T]$.

The following lemma gives upper bounds on the degree and length of Γ and of the various polynomials $\mathfrak{B}_W$, for W a subset of V.

Lemma 4. *Suppose that F and G have degree at most d and length at most h. Then, the following holds:*

- *for any subset W of V defined over $\mathbb{Q}$, the polynomial $\mathfrak{B}_W \in \mathbb{Q}[T, X]$ has degree bounded by d^2 and length bounded by an efficiently computable integer $B_{\mathfrak{B}}(d, h) = O(dh + d^2)$.*
- *Γ has degree bounded by $6d^4$ and length bounded by an efficiently computable integer $B_\Gamma(d, h) = O(d^3h + d^4)$.*

Proof. The polynomials F_T and G_T have degree at most d in Y and (T, X) and, by inequality b_1 above, length at most $h + 4d = O(h + d)$. Their resultant $\mathfrak{A} \in \mathbb{Z}[T, X]$ has total degree at most $2d^2$, and the determinant bound b_3 implies that its length is at most $2dh + 16d^2 = O(dh + d^2)$. The factor bound b_2 then implies that $\mathfrak{B}$, which divides $\mathfrak{A}$ in $\mathbb{Z}[T, X]$, has length bounded by an efficiently computable integer in $O(dh + d^2)$.

From this, we can prove our bounds on $\mathfrak{B}_W$. The degree bound in (T, X) is obvious, since at most d^2 linear factors appear in the product giving $\mathfrak{C}_W$. For the length bound, remark that the polynomial $\mathfrak{B}_W$ divides $\mathfrak{B}$, and thus $\mathfrak{A}$ in $\mathbb{Z}[T, X]$, so it admits the same length upper bound as $\mathfrak{B}$ itself.

On the other hand, the determinant bound b_3, together with the degree and length bounds for the polynomial $\mathfrak{B}$, imply that Δ has degree at most $2d^4$ and length $O(d^3h + d^4)$. Multiplying by the leading coefficients $\mathfrak{f}$ and $\mathfrak{g}$ of respectively F_T and G_T with respect to Y, and by the leading coefficient $\mathfrak{a}$ of $\mathfrak{A}$ in X, which are all in $\mathbb{Z}[T]$ and whose sum of degrees is at most $2d + 2d^2 \leq 4d^4$, we deduce that Γ has degree at most $6d^4$ and length still $O(d^3h + d^4)$. □

The last technical lemma we need is the following specialization result, which will show how Γ controls (in particular) primes of bad reduction. Although we give it in the general form we will need below, our presentation is inspired by that in [7], which was given for $\mathbb{A} = \mathbb{F}_p$, for a prime p.

Lemma 5. *Let ϕ be a ring morphism $\mathbb{Z}[T] \to \mathbb{A}$, where $\mathbb{A}$ is a domain; this morphism extends to a ring morphism $\phi : \mathbb{Z}[T, X, Y] \to \mathbb{A}[X, Y]$.*

Let $V' = V(\phi(F_T), \phi(G_T)) \subset \overline{\mathbb{K}}^2$, where $\overline{\mathbb{K}}$ is an algebraic closure of the fraction field $\mathbb{K}$ of $\mathbb{A}$. Then:

- *if $\phi(\mathfrak{f})$ and $\phi(\mathfrak{g})$ are nonzero, the cardinality of $\pi(V')$ is the number of pairwise distinct roots of $\phi(\mathfrak{A})$ in $\overline{\mathbb{K}}$;*
- *if $\phi(\Gamma)$ is nonzero, $\phi(F_T)$ and $\phi(G_T)$ satisfy H_1 and H_2, and the cardinality of $\pi(V') \subset \overline{\mathbb{K}}$ is equal to the cardinality of V.*

Proof. First, let us establish that the cardinality of $\pi(V')$ is the number of pairwise distinct roots of $\phi(\mathfrak{A})$ in $\overline{\mathbb{K}}$ when $\phi(\mathfrak{f})$ and $\phi(\mathfrak{g})$ are nonzero. In this case, indeed, they remain the leading coefficients of respectively $\phi(F_T)$ and $\phi(G_T)$ with respect to Y (which proves H_2); in addition, the resultant $\mathrm{res}(\phi(F_T), \phi(G_T), Y)$ coincides with the image $\phi(\mathrm{res}(F_T, G_T, Y)) = \phi(\mathfrak{A})$. On the other hand, because $\phi(\mathfrak{f})$ and $\phi(\mathfrak{g})$ are nonzero, the number of pairwise distinct roots of $\mathrm{res}(\phi(F_T), \phi(G_T), Y)$ is the cardinality of $\pi(V') \subset \overline{\mathbb{K}}$. Our claim above is thus proved.

Let us further suppose that $\phi(\Gamma)$ is nonzero. Then, $\phi(\mathfrak{a})$ is nonzero, where $\mathfrak{a}$ is the leading coefficient of $\mathfrak{A}$ with respect to X (recall that $\mathfrak{a}$ divides Γ). Thus, $\phi(\mathfrak{A})$ itself is nonzero, which implies that $\mathrm{res}(\phi(F_T), \phi(G_T), Y)$ is nonzero. As a result, the only possible common factors of $\phi(F_T)$ and $\phi(G_T)$ in $\mathbb{K}[X, Y]$ must lie in $\mathbb{K}[X]$. However, since their leading coefficients with respect to Y lie in $\mathbb{A}$, they have no such common factor. This proves H_1.

Since $\mathfrak{B} = \mathfrak{A}/\gcd(\mathfrak{A}, \mathfrak{A}')$, we deduce that $\mathfrak{B}$ divides $\mathfrak{A}$, and that $\mathfrak{A}$ divides some polynomial of the form $\mathfrak{a}\mathfrak{B}^k$ in $\mathbb{Z}[T, X]$, for some positive integer k; we can for instance deduce this from the factorizations of $\mathfrak{A}$ and $\mathfrak{B}$ given in (3) and (4). This relationship remains true through ϕ; this implies that $\phi(\mathfrak{B})$ and $\phi(\mathfrak{A})$ have the same roots in $\overline{\mathbb{K}}$.

Now, we claim that because $\phi(\Delta)$ is nonzero, $\phi(\mathfrak{B})$ has no multiple root in $\overline{\mathbb{K}}$. Indeed, the leading term $\mathfrak{b}$ of $\mathfrak{B}$ divides $\mathfrak{a}$ (Lemma 1), so $\phi(\mathfrak{b})$ must be nonzero. This implies that

the resultant computation that gives $\Delta = \mathrm{res}(\mathfrak{B}, \mathfrak{B}', X)$ carries over through ϕ, where $\mathfrak{B}'$ is the derivative of $\mathfrak{B}$ with respect to X. Thus, $\phi(\Delta)$ is the discriminant of $\phi(\mathfrak{B})$, and since it does not vanish, $\phi(\mathfrak{B})$ has no multiple root in $\overline{\mathbb{K}}$, as claimed above.

The latter claim implies that the number of roots $\phi(\mathfrak{B})$, or equivalently, as we saw above, of $\phi(\mathfrak{A})$, is equal to the degree of $\phi(\mathfrak{B})$. Since $\phi(\mathfrak{b})$ is nonzero, this degree is the degree of $\mathfrak{B}$ in X, which is the cardinality of V, by Eq. (2). On the other hand, the first paragraph proved that the number of roots of $\phi(\mathfrak{A})$ is equal to the cardinality of $\pi(V')$, so we are done. □

We can finally conclude this subsection with the following corollary, which summarizes the facts we will need below to control the bit-size of the output of our algorithm, as well as primes of bad reduction and unlucky changes of coordinates. As said before, we make no claim of originality here, but the lack of a reference that provided all exact claims we need led us to write all proofs. Some of these results, or close variants, are known: see for instance [6, Proposition 10] for bit-size estimates for Rational Univariate Representations, in a context where multiplicities are still kept into account, but where the results are stated only for $V(F, G)$, not its $\mathbb{Q}$-definable subsets.

Corollary 1. *Let F and G be in $\mathbb{Z}[X, Y]$, that satisfy H_1, with degree at most d and length at most h. Then, the following holds.*

- *For t in $\mathbb{Z}$, if $\Gamma(t)$ is nonzero, then F_t and G_t are in general position.*
- *For t as above, if t has length at most ℓ, and if W is a subset of V defined over $\mathbb{Q}$, then the polynomials $(P_{W_t}, R_{W_t}) = \mathsf{RUR}(W_t)$ have degree bounded by d^2 and length bounded by an efficiently computable integer $B_{\mathsf{RUR}}(d, h, \ell) = \tilde{O}(dh + d^2\ell)$.*

  In addition, the polynomial S_{W_t} appearing in $(P_{W_t}, S_{W_t}) = \mathsf{SL}(W_t)$ has degree at most d^2 and length bounded by an efficiently computable integer $B_{\mathsf{SL}}(d, h, \ell) = \tilde{O}(d^3h + d^4\ell)$.

  In particular, the polynomials in $\mathsf{RUR}(F_t, G_t)$ and $\mathsf{SL}(F_t, G_t)$ satisfy these bounds.
- *Let in addition p be a prime. If $\Gamma(t) \bmod p$ is nonzero, then $F_t \bmod p$ and $G_t \bmod p$ are in general position, and the leading terms of $F_t \bmod p$ and $G_t \bmod p$ with respect to Y are the images of those of F_t and G_t modulo p.*
- *Let W be a subset of V defined over $\mathbb{Q}$. For t and p as above, write again $(P_{W_t}, R_{W_t}) = \mathsf{RUR}(W_t)$ and $(P_{W_t}, S_{W_t}) = \mathsf{SL}(W_t)$. Then, p cancels no denominator in P_{W_t}, R_{W_t} or S_{W_t}, and $P_{W_t} \bmod p$ remains squarefree.*

  In addition, for $W = V$, we have

$$\mathsf{SL}(F_t, G_t) \bmod p = \mathsf{SL}(F_t \bmod p, G_t \bmod p)$$

  and

$$\mathsf{RUR}(F_t, G_t) \bmod p = \mathsf{RUR}(F_t \bmod p, G_t \bmod p).$$

Proof. Suppose that $t \in \mathbb{Z}$ is such that $\Gamma(t)$ is nonzero. Properties H_1 and G_1 clearly hold for F_t and G_t. Applying the previous lemma to $\phi : \mathbb{Z}[T] \to \mathbb{Z}$ given by $\phi(f) = f(t)$, we deduce that H_2 holds for F_t and G_t, and that the cardinality of $\pi(V_t) \subset \overline{\mathbb{Q}}$ is equal to the cardinality of V. Since V and V_t have the same cardinality, because V_t is obtained from V by a change of variables, this proves that V_t satisfies G_2. Thus, F_t and G_t are in general position.

To prove the second item, recall the formulas for P_{W_t}, R_{W_t} and S_{W_t} given in Lemma 3:

$$P_{W_t} = \frac{1}{\mathfrak{b}_W}\mathfrak{B}_W(t, X), \quad R_{W_t} = \frac{1}{\mathfrak{b}_W}\frac{\partial \mathfrak{B}_W}{\partial T}(t, X) \quad \text{and} \quad S_{W_t} = \frac{\frac{\partial \mathfrak{B}_W}{\partial T}(t, X)}{\frac{\partial \mathfrak{B}_W}{\partial X}(t, X)} \mod P_{W_t}.$$

Recall also the bounds on the degree and length of $\mathfrak{B}_W$ given in Lemma 4, which are respectively d^2 and $O(dh + d^2)$; the lengths of both derivatives of $\mathfrak{B}_W$ admit the same bound as that of $\mathfrak{B}_W$, up to a negligible additional $\mathrm{len}(d)$ term. Using the evaluation bound b_1, we deduce that evaluation at $T = t$ incurs a length growth of $O(d^2\ell)$, so that the lengths of P_{W_t} and R_{W_t} are $O^{\sim}(dh + d^2\ell)$, as claimed.

Next, we deduce bounds for S_{W_t}. The expression given above for S_{W_t} shows that we can obtain its coefficients by solving a linear system with matrix the Sylvester matrix of $\mathfrak{B}_W(t, X)$ and $\frac{\partial \mathfrak{B}_W}{\partial X}(t, X)$, and with right-hand side made up from the coefficients of $\frac{\partial \mathfrak{B}_W}{\partial T}(t, X)$; this proves in particular that the denominators of all coefficients of S_{W_t} divide the discriminant of $\mathfrak{B}_W(t, X)$. We can then use Cramer's formulas and apply the determinant bound b_3 to estimate the length of the coefficients of S_{W_t}; the Sylvester matrix and the right-hand side have size $O(d^2)$, with integer entries of length $O^{\sim}(dh + d^2\ell)$, so all determinants we need are integers of length $O^{\sim}(d^3h + d^4\ell)$.

The results of the last two paragraphs allow us to define B_{RUR} and B_{SL}; since both bounds are derived by direct applications of b_1, b_2 and b_3, they are indeed efficiently computable; this proves the second item.

Suppose next that the prime p is such that $\Gamma(t) \bmod p$ is nonzero. Consider first $\phi' : \mathbb{Z}[T] \to \mathbb{F}_p[T]$ given by $\phi(f) = f \bmod p$. Because $\Gamma(t) \bmod p$ is nonzero, we have in particular that $\phi'(\Gamma)$ is nonzero. The previous lemma then implies that $(F_T \bmod p, G_T \bmod p)$ satisfy the coprimality assumption H_1. Since these polynomials are obtained from $(F \bmod p, G \bmod p)$ through the substitution $X \mapsto X + TY$, we deduce that $(F \bmod p, G \bmod p)$ satisfy H_1 as well. Let V'_T be the zero-set of $(F_T \bmod p, G_T \bmod p)$ in an algebraic closure of $\mathbb{F}_p[T]$; then, as pointed out previously, V'_T satisfies G_2, so that $|\pi(V'_T)| = |V'_T|$. On the other hand, applying the previous lemma to ϕ' implies that the cardinality of $|\pi(V'_T)|$ is equal to $|V|$. In particular, we deduce that $|V'_T| = |V|$.

Let further V'_t be the zero-set of $(F_t \bmod p, G_t \bmod p)$ in an algebraic closure of $\mathbb{F}_p$. Because evaluation of T at t commutes with reduction modulo p, we deduce that $(F_t \bmod p, G_t \bmod p)$ are obtained by evaluating $(F_T \bmod p, G_T \bmod p)$ at $T = t \bmod p$, so $|V'_T| = |V'_t|$. Since we saw that $|V'_T| = |V|$, we deduce that $|V| = |V'_t|$.

Consider now the mapping $\phi'' : \mathbb{Z}[T] \to \mathbb{F}_p$ given by $\phi(f) = f(t) \bmod p$. Applying the previous lemma to ϕ'', we deduce that $(F_t \bmod p, G_t \bmod p)$ satisfy H_1 and H_2, and that $|\pi(V'_t)| = |V|$. Since we saw above that $|V| = |V'_t|$, this proves that $|\pi(V'_t)| = |V'_t|$, so that $(F_t \bmod p, G_t \bmod p)$ are in general position.

This almost proves the third item; the missing assertion from that item (that the leading terms of $F_t \bmod p$ and $G_t \bmod p$ with respect to Y are the images of those of F_t and G_t modulo p) is straightforward, since both $\mathfrak{f}$ and $\mathfrak{g}$ divide Γ in $\mathbb{Z}[T]$.

To conclude, consider a subset W of V, defined over $\mathbb{Q}$, together with the formulas that yield $(P_{W_t}, R_{W_t}) = \mathsf{RUR}(W_t)$. In particular, we have $(P_{V_t}, R_{V_t}) = \mathsf{RUR}(V_t) = \mathsf{RUR}(F_t, G_t)$ and $(P_{V_t}, S_{V_t}) = \mathsf{SL}(V_t) = \mathsf{SL}(F_t, G_t)$.

Recall that $\mathfrak{b} \bmod p$ is nonzero (we established this in the proof of the previous lemma, applied to ϕ''). Since $\mathfrak{b}_W$ divides $\mathfrak{b}$, $\mathfrak{b}_W$ does not vanish modulo p. Using Lemma 3, this proves that none of the denominators of the coefficients of either P_{W_t} or R_{W_t} vanishes at p. On the other hand, $\mathfrak{B}(t, X)$, or equivalently P_{V_t}, remains squarefree modulo p (this was established as well in the proof of the previous lemma), so this is the case as well for the polynomial P_{W_t} appearing in the Shape Lemma representation of W_t. We saw above that all denominators appearing in the coefficients of S_{W_t} divide the discriminant of $\mathfrak{B}_W(t, X)$, so they are nonzero modulo p, as claimed.

To conclude, notice that the polynomials F_t and G_t reduce to zero modulo $(P_{V_t}, Y - S_{V_t})$. This relationship remains true modulo p, so that the polynomials $(P_{V_t} \bmod p, Y - S_t \bmod p)$ define a subset of $V'_t = V(F_t \bmod p, G_t \bmod p)$. However, both sets have the same cardinality $|V|$, so they are equal. By uniqueness, we conclude that $\mathsf{SL}(F_t, G_t) \bmod p = \mathsf{SL}(F_t \bmod p, G_t \bmod p)$; multiplying by $P'_{V_t} \bmod P_{V_t}$, this carries over to $\mathsf{RUR}(F_t, G_t) \bmod p = \mathsf{RUR}(F_t \bmod p, G_t \bmod p)$. The proof is complete. □

Finally, we state a partial converse to these claims.

Lemma 6. *Let F and G be in $\mathbb{Z}[X, Y]$, that satisfy H_1, with degree at most d and length at most h. Suppose that a prime p and $t \in \mathbb{Z}$ are such that:*

- *$\Gamma \bmod p$ is nonzero,*
- *$\mathfrak{a}(t)\mathfrak{f}(t)\mathfrak{g}(t) \neq 0 \bmod p$*
- *X is a separating element for $V(F_t \bmod p, G_t \bmod p) \subset \overline{\mathbb{F}_p}^2$.*

Then, $\Gamma(t) \bmod p$ is nonzero.

Proof. Because $\mathfrak{f}(t)\mathfrak{g}(t) \neq 0 \bmod p$, $\mathfrak{A}(t, X) \bmod p$ is the resultant of $F_t \bmod p$ and $G_t \bmod p$ with respect to Y, computed over $\mathbb{F}_p$, and its roots are precisely the X-coordinates of the points in $V(F_t \bmod p, G_t \bmod p) \subset \overline{\mathbb{F}_p}^2$.

The last assumption then implies that the number of these roots is equal to the cardinality of $V(F_t \bmod p, G_t \bmod p) \subset \overline{\mathbb{F}_p}^2$, or equivalently of $V(F_T \bmod p, G_T \bmod p)$. Using Lemma 5 with $\phi : \mathbb{Z}[T] \to \mathbb{F}_p[T]$ given by reduction modulo p, the first assumption implies that this is precisely the cardinality of $V(F, G)$ (because the projection π is one-to-one on $V(F_T \bmod p, G_T \bmod p)$).

Now, recall from Lemma 1 and its proof that there exist polynomial $(B_m)_{m \in M}$ in $\mathbb{Z}[T, X]$ (for some finite set M) and integers $(\beta_m)_{m \in M}$ such that we have

$$\mathfrak{A} = \frac{\mathfrak{a}}{\prod_{m \in M} \beta_m^m} \prod_{m \in M} B_m^m \quad \text{and} \quad \mathfrak{B} = \prod_{m \in M} B_m,$$

where the first fraction is an exact division in $\mathbb{Z}[T]$.

Since $\mathfrak{a}(t) \bmod p$ is nonzero, we deduce that the number of roots of $\mathfrak{A}(t, X) \bmod p$ and $\mathfrak{B}(t, X) \bmod p$ in $\overline{\mathbb{F}_p}$ are the same. We saw above that this number is equal to the cardinality of $V(F, G)$, which is equal to the degree of $\mathfrak{B}$ with respect to X.

This implies that $\mathfrak{B}(t, X) \bmod p$ is squarefree, so its discriminant does not vanish. Since $\mathfrak{a}(t) \bmod p$ is not zero, the leading coefficient $\mathfrak{b}$ of $\mathfrak{B}$ does not vanish modulo p; this implies that this discriminant is equal to $\Delta(t) \bmod p$. Thus, this quantity is nonzero, and this is enough to deduce that $\Gamma(t) \bmod p$ itself is nonzero. □

2.3 Non-vanishing conditions

Let $\mathbb{K}$ be a field, let P and S be in $\mathbb{K}[X]$, with P monic of degree e, and S of degree less than e. Consider a further polynomial H in $\mathbb{K}[X, Y]$, and assume that the following properties hold:

C_1. P is squarefree.

C_2. H vanishes nowhere on the set $V = V(P, Y - S)$.

In this short section, we mainly focus on the case $\mathbb{K} = \mathbb{Q}$. Assuming that H has integer coefficients, we give conditions under which these two properties are maintained through reduction at a prime p.

Proposition 1. *There exists an efficiently computable function $\Delta_1(d, h, e, \ell) = (dhe\ell)^{O(1)}$ such that the following holds.*

Suppose that P and S are in $\mathbb{Q}[X]$ and have degree at most e and length at most ℓ, and that $H \in \mathbb{Z}[X, Y]$ has degree at most d and length at most h. If (P, S, H) satisfy C_1 and C_2, there exists a nonzero integer δ_1 such that:

- *δ_1 has length at most $\Delta_1(d, h, e, \ell)$;*
- *for any prime p that does not divide δ_1, $P \bmod p$ and $S \bmod p$ are well-defined, and $(P, S, H) \bmod p$ satisfy C_1 and C_2 over $\mathbb{F}_p$.*

The proof of this result occupies the rest of this section. Let c_P and c_S be minimal common denominators for the coefficients of respectively P and S, so that we can write $P = P^\star/c_P$ and $S = S^\star/c_S$, with $P^\star$ and $S^\star$ in $\mathbb{Z}[X]$. Remark that the integers c_P and c_S have length at most ℓ, and that the same holds for the polynomials $P^\star$ and $S^\star$.

Suppose that p is a prime that does not divide $c_P c_S$, and that P remains squarefree modulo p. Thus, C_1 is maintained through reduction at such a prime.

Starting from $H = \sum_{i+j \le d} h_{i,j} X^i Y^j$, let us then define the polynomial with integer coefficients

$$H^\star = \sum_{i+j \le d} c_S^{d-j} h_{i,j} X^i Y^j,$$

so that $K = c_S^d H(X, S)$ satisfies $K = H^\star(X, S^\star) \in \mathbb{Z}[X]$. By assumption C_2, this polynomial is coprime with P, and C_2 holds modulo p if K and P remain coprime modulo p. Because p does not divide the leading coefficient c_P of $P^\star$, this is the case as soon as p does not divide the resultant of K and $P^\star$, which is a nonzero integer. Thus, we can define δ_1 as the nonzero integer

$$\delta_1 = c_P c_S \operatorname{res}(P^\star, P^{\star\prime}, X) \operatorname{res}(P^\star, K, X).$$

It remains to estimate the length of this integer. First, recall that c_P and c_S have length at most ℓ and that the same holds for $P^\star$ and $S^\star$; this implies that $P^{\star\prime}$ has length at most $\ell + \operatorname{len}(e)$.

- The matrix giving the resultant $\operatorname{res}(P^\star, P^{\star\prime}, X)$ has size at most $2e$ and integer entries of length at most $\ell + \operatorname{len}(e)$. Hence, we can use the determinant bound b_3 to deduce that its determinant is a nonzero integer of length $O\tilde{}(e\ell)$.
- The polynomial $H^\star$ has degree at most d, and coefficients of length at most $h + d\ell$.
- The polynomial $K = H^\star(X, S^\star)$ is obtained by evaluating a polynomial of degree at most d in two variables, with coefficients of length at most $h + d\ell$, at univariate polynomials of degree at most e and length at most ℓ. Using the evaluation bound b_1, we deduce that K has degree at most de and length $O(h + d(e + \ell))$.
- As a result, using again the determinant bound b_3, we obtain that the matrix giving the resultant $\operatorname{res}(P^\star, K, X)$ has for determinant a nonzero integer of length $O(de(h + d(e + \ell)))$.

Adding all estimates gives an explicit formula for the upper bound Δ_1, which is easily seen to be polynomial in d, h, e, ℓ and computable in time $\log(dhe\ell)^{O(1)}$.

2.4 Conservation of intersection multiplicity

Our context in this section is similar to the one of the previous section. We consider a perfect field $\mathbb{K}$, P and S in $\mathbb{K}[X]$, with P monic of degree e, and S of degree less than e. Now, we also take two further polynomials H, K in $\mathbb{K}[X, Y]$, not necessarily coprime, and we assume that the following properties hold:

M_1. P is squarefree.

M_2. All points in $V = V(P, Y - S)$ are isolated points of $V(H, K)$.

We are interested in describing situations under which the following extra property is verified:

$\mathsf{M}_3(n)$. There exists $n \geq 1$ such that for all (x, y) in V, $\langle H, K\rangle$ has multiplicity n at (x, y).

Define the new polynomials $G = \gcd(H, K) \in \mathbb{Z}[X, Y]$, $H^\dagger = H/G$ and $K^\dagger = K/G$. Then, $V(H^\dagger, K^\dagger)$ is finite, the points in V are still isolated points of $V(H^\dagger, K^\dagger)$, and for

$(x, y) \in V$, the intersection multiplicities $\mu((H, K), (x, y))$ and $\mu((H^\dagger, K^\dagger), (x, y))$ are the same.

Intersection multiplicity is invariant through linear change of coordinates. Thus, reusing the notation of Subsection 2.2, we deduce that for any value of t in $\mathbb{K}$, and for (x, y) in V, the equality $\mu((H^\dagger, K^\dagger), (x, y)) = \mu((H_t^\dagger, K_t^\dagger), (x - ty, y))$ holds.

As per our convention, V_t denotes the image of $V = V(P, Y - S)$ under the change of coordinates $\phi_t : (x, y) \mapsto (x - ty, y)$.

If $t \in \mathbb{Z}$ is such that $H_t^\dagger$ and $K_t^\dagger$ are in general position, then since V_t is a subset of $V(H_t^\dagger, K_t^\dagger)$ of cardinality e defined over the perfect field $\mathbb{K}$, it admits a Shape Lemma representation: there exist polynomials $P_{[t]}$ and $S_{[t]}$ in $\mathbb{K}[X]$, with $P_{[t]}$ monic and squarefree of degree e, such that $V_t = V(P_{[t]}, Y - S_{[t]})$. Note that we use the symbol $[t]$ in our subscripts for P and S, since the subscript t is reserved for polynomials obtained by applying a linear change of variable. The same will hold below for the polynomial $A_{[t]}$.

Lemma 7. *Let t be such that $H_t^\dagger$ and $K_t^\dagger$ are in general position, and let $A_{[t]} \in \mathbb{K}[X]$ be their resultant with respect to Y. For $n \geq 1$, condition $\mathsf{M}_3(n)$ holds if and only if we have both:*

- *$P_{[t]}^n$ divides $A_{[t]}$ in $\mathbb{K}[X]$;*
- *$P_{[t]}$ and $A_{[t]}/P_{[t]}^n$ are coprime in $\mathbb{K}[X]$.*

Proof. Because $H_t^\dagger$ and $K_t^\dagger$ are in general position, for any (x, y) in V_t, we know that $\mu((H_t^\dagger, K_t^\dagger), (x, y))$ is the valuation of the resultant $A_{[t]} = \mathrm{res}(H_t^\dagger, K_t^\dagger, Y)$ at x, that is, the highest exponent n such that $(X - x)^n$ divides $A_{[t]}$. Equivalently, $\mu((H_t^\dagger, K_t^\dagger), (x, y))$ is characterized as being the unique integer n such that $(X - x)^n$ divides $A_{[t]}$ and $(X - x)$ and $A_{[t]}/(X - x)^n$ are coprime.

Taking all (x, y) in V into account, this leads to the condition given in the statement of the lemma. □

We will now focus on the particular case where $\mathbb{K} = \mathbb{Q}$. We suppose that H and K are in $\mathbb{Z}[X, Y]$, that P and S are in $\mathbb{Q}[X]$, and that P, S, H, K satisfy M_1, M_2 and $\mathsf{M}_3(n)$, for some $n \geq 1$. Our goal is to give conditions on a prime p such that the same polynomials taken modulo p are well-defined and still satisfy M_1, M_2 and $\mathsf{M}_3(n)$.

Proposition 2. *There exists an efficiently computable function $\Delta_2(d, h, e, \ell) = (dhe\ell)^{O(1)}$ such that the following holds.*

Suppose that P and S are in $\mathbb{Q}[X]$ and have degree at most e and length at most ℓ, and that $H, K \in \mathbb{Z}[X, Y]$ have degree at most d and length at most h. If (P, S, H, K) satisfy M_1, M_2 and $\mathsf{M}_3(n)$, for some $n \geq 1$, there exists a nonzero integer δ_2 such that:

- *δ_2 has length at most $\Delta_2(d, h, e, \ell)$;*
- *for any prime p that does not divide δ_2, $P \bmod p$ and $S \bmod p$ are well-defined, and $(P, S, H, K) \bmod p$ satisfy M_1, M_2 and $\mathsf{M}_3(n)$ over $\mathbb{F}_p$.*

The proof of this proposition will occupy the rest of this section. As a preliminary remark, we will still let G be the gcd of H and K in $\mathbb{Z}[X,Y]$, and write $H^\dagger = H/G$ and $K^\dagger = K/G$.

Since $V = V(P, Y - S)$ consists entirely of isolated points of $V(H,K)$, the polynomials (P,S,G) satisfy conditions C_1 and C_2 of the previous section. Our first constraint is that p does not divide the integer δ_1 defined in Proposition 1. For such a prime p, the polynomials (P,S,G) mod p are well-defined, P remains squarefree modulo p, and (P,S,G) mod p still satisfy conditions C_1 and C_2. In particular, the polynomials (P,S,H,K) mod p still satisfy M_1, but we cannot conclude that they satisfy M_2 yet.

Let then Γ be the polynomial in $\mathbb{Z}[T]$ associated to the coprime polynomials $H^\dagger$ and $K^\dagger$ by the construction of Section 2.2. In all that follows, we take t in $\mathbb{Z}$ such that $\Gamma(t)$ is nonzero; in particular, by Corollary 1, $H_t^\dagger$ and $K_t^\dagger$ are in general position. We let $A_{[t]} = \mathrm{res}(H^\dagger, K^\dagger, Y) \in \mathbb{Z}[X]$ and $P_{[t]} \in \mathbb{Q}[X]$ be as defined above; then, by the previous lemma, $P_{[t]}^n$ divides $A_{[t]}$ in $\mathbb{Q}[X]$, and $P_{[t]}$ and $A_{[t]}/P_{[t]}^n$ are coprime in $\mathbb{Q}[X]$.

We will give conditions on p for which the same statement remains true modulo p; then, using the converse direction in the previous lemma will allow us to conclude.

The resultant $A_{[t]}$ is in $\mathbb{Z}[X]$, not necessarily monic. The polynomial $P_{[t]}$ is monic in $\mathbb{Q}[X]$, so we may write it as $P_{[t]} = P_{[t]}^\star / c_{[t]}$, with $c_{[t]}$ in $\mathbb{Z}$ and $P_{[t]}^\star$ primitive in $\mathbb{Z}[X]$. Since $P_{[t]}^n$ divides $A_{[t]}$ in $\mathbb{Q}[X]$, we deduce that ${P_{[t]}^\star}^n$ divides $A_{[t]}$ in $\mathbb{Z}[X]$, so $N_{[t]} = A_{[t]}/{P_{[t]}^\star}^n$ is a polynomial with integer coefficients. By assumption, $P_{[t]}$ and $N_{[t]}$ are coprime, and thus so are $P_{[t]}^\star$ and $N_{[t]}$. We deduce that their resultant is a nonzero integer.

Let us then add the following conditions on our prime p: $\Gamma(t)$ mod p is nonzero, and the resultant $\mathrm{res}(P_{[t]}^\star, N_{[t]}, X)$ does not vanish modulo p. We will prove that M_2 and $\mathsf{M}_3(n)$ are satisfied for (P,S,H,K) mod p.

Since $\Gamma(t)$ mod p is nonzero, we can apply Corollary 1 to $H^\dagger$ and $K^\dagger$, and we deduce the following facts:

- $H_t^\dagger$ mod p and $K_t^\dagger$ mod p are in general position; in particular, $H^\dagger$ mod p and $K^\dagger$ mod p have finitely many common solutions. Since $H = GH^\dagger$ and $K = GK^\dagger$, and since by Proposition 1 the points defined by $(P \bmod p, Y - S \bmod p)$ do not cancel G mod p, we deduce that these points are isolated points on $V(H \bmod p, K \bmod p)$, and that the multiplicities of (H,K) mod p and $(H^\dagger, K^\dagger)$ mod p are the same at these points. In particular, we have proved that M_2 still holds.

- Let $\alpha_{[t]} \in \mathbb{F}_p[X]$ be the resultant of $H_t^\dagger$ mod p and $K_t^\dagger$ mod p with respect to Y. By Corollary 1, the leading terms of $H_t^\dagger$ mod p and $K_t^\dagger$ mod p are the reductions modulo p of those of $H_t^\dagger$ and $K_t^\dagger$. As a consequence, $\alpha_{[t]} = A_{[t]} \bmod p$.

Since P mod p is squarefree, the equations $(P \bmod p, Y - S \bmod p)$ define a subset $V' \subset \overline{\mathbb{F}_p}^2$ of cardinality e of $V(H^\dagger \bmod p, K^\dagger \bmod p)$. Applying the change of coordinates ϕ_t, we obtain a subset $V_t' \subset \overline{\mathbb{F}_p}^2$ of $V(H_t^\dagger \bmod p, K_t^\dagger \bmod p)$ of cardinality e. Since we saw that the equations $(H_t^\dagger \bmod p, K_t^\dagger \bmod p)$ are in general position, we deduce as in the discussion prior to Lemma 7 that V_t' admits a Shape Lemma representation.

Lemma 8. *None of the denominators of the coefficients of $P_{[t]}$ or $S_{[t]}$ vanishes modulo p, $P_{[t]} \bmod p$ is squarefree and the Shape Lemma representation of V'_t is $(P_{[t]} \bmod p, S_{[t]} \bmod p)$.*

Proof. The facts that none of the denominators of the coefficients of $P_{[t]}$ or $S_{[t]}$ vanishes modulo p and that $P_{[t]} \bmod p$ is squarefree are consequences of Corollary 1 applied to $H^\dagger$, $K^\dagger$, and the $\mathbb{Q}$-definable subset V of $V(H^\dagger, K^\dagger)$.

By construction, V_t is the zero-set of $(P_{[t]}, Y - S_{[t]})$. Thus, applying ϕ_t^{-1}, we deduce that $P_{[t]}(X - tY)$ and $Y - S_{[t]}(X - tY)$ vanish on V; this implies that $P_{[t]}(X - tY)$ and $Y - S_{[t]}(X - tY)$ reduce to zero modulo $(P, Y - S)$.

Because no denominator reduces to zero modulo p in these membership equalities, they remain true modulo p. This shows that V' is contained in the zero-set of $(P_{[t]}(X - tY) \bmod p, Y - S_{[t]}(X - tY) \bmod p)$. The set V' has cardinality e (because $P \bmod p$ is squarefree), and so does the zero-set of $(P_{[t]}(X - tY) \bmod p, Y - S_{[t]}(X - tY) \bmod p)$, because $P_{[t]} \bmod p$ is squarefree; thus, the inclusion is an equality.

Applying ϕ_t, we deduce that V'_t is the zero-set of $(P_{[t]} \bmod p, Y - S_{[t]} \bmod p)$. By uniqueness of the Shape Lemma representation, we are done. □

We can now prove that $\mathsf{M}_3(n)$ is satisfied for $(P, S, H, K) \bmod p$. In what follows, we write $\pi_{[t]} = P_{[t]} \bmod p \in \mathbb{F}_p[X]$.

Recall that $N_{[t]} \in \mathbb{Z}[X]$ is given by $N_{[t]} = A_{[t]}/{P^\star_{[t]}}^n$. By assumption, $P^\star_{[t]}$ is primitive, so $P^\star_{[t]} \bmod p$ is nonzero, which implies that $N_{[t]} \bmod p = (A_{[t]} \bmod p)/({P^\star_{[t]}}^n \bmod p)$; in addition, $P^\star_{[t]} \bmod p$ coincides with $\pi_{[t]}$ up to a nonzero constant. We saw above that $A_{[t]} \bmod p$ is the resultant $\alpha_{[t]} = \mathrm{res}(H_t^\dagger \bmod p, K_t^\dagger \bmod p, Y)$, so that $N_{[t]} \bmod p$ and $\alpha_{[t]}/\pi^n_{[t]}$ are the same, up to a nonzero constant.

Since p divides the denominator of no coefficient of $P_{[t]}$, the degree of $P^\star_{[t]} \bmod p$ remains equal to e, so $\mathrm{res}(P^\star_{[t]}, N_{[t]}, X) \bmod p$ is equal (up to a nonzero constant) to the resultant $\mathrm{res}(\pi_{[t]}, \alpha_{[t]}/\pi^n_{[t]}, X)$ computed in $\mathbb{F}_p[X]$. By assumption on p, the resultant $\mathrm{res}(P^\star_{[t]}, N_{[t]}, X)$ does not vanish modulo p, so that $\mathrm{res}(\pi_{[t]}, \alpha_{[t]}/\pi^n_{[t]}, X)$ is a nonzero element of $\mathbb{F}_p$. The previous lemma shows that $\pi_{[t]}$ is precisely the first polynomial appearing in the Shape Lemma representation of V'_t, and Lemma 7 then implies that $\mathsf{M}_3(n)$ is satisfied for $(P, S, H, K) \bmod p$. Thus, we are done.

It remains to quantify the conditions on p. The first constraint is that p does not divide the integer δ_1 defined in Proposition 1; recall that δ_1 has length $(dh\ell)^{O(1)}$. Our other constraints are that $\Gamma(t)$ and $\mathrm{res}(P^\star_{[t]}, N_{[t]}, X)$ do not vanish modulo p, where t is any integer that does not cancel Γ. Let us then deal with these two terms.

- Since Lemma 4 shows that $\deg(\Gamma) \leq 6d^4$, there exists $t \in \mathbb{N}$ that does not cancel Γ and such that $t \leq 6d^4 + 1$; its length is $O(\mathrm{len}(d))$. By Lemma 4 again, the length of Γ is $O(hd^3 + d^4)$, so the length of $\Gamma(t)$ is of the same order: an upper bound $(hd)^{O(1)}$ can be calculated for it using the evaluation bound b_1.

- Using the factor bound b_2, we obtain an upper bound $(hd)^{O(1)}$ for the length of both $H^\dagger$ and $K^\dagger$. From this, the determinant bound b_3 gives an efficiently computable upper bound on the length of $A_{[t]}$, which is still $(dh)^{O(1)}$; the degree of $A_{[t]}$ is at most d^2. The

polynomials $P^\star_{[t]}$ and $N_{[t]}$ both divide $A_{[t]}$, so we can apply again the factor bound b_2 to deduce upper bounds for their length, which are again $(dh)^{O(1)}$. Finally, using once again the determinant bound b_3, we can deduce bounds of the form $(hd)^{O(1)}$ for the length of the integer $\mathrm{res}(P^\star_{[t]}, N_{[t]}, X)$.

Since all bounds obtained here are direct consequences b_1, b_2 and b_3, they can all be computed efficiently, so the proof is complete.

3 Finding nonzeros in a list

In this section, we present simple algorithms for bookkeeping computations with univariate polynomials. The process given below of partitioning an algebraic set into parts indexed by indices that are here integers, but will later on become more complex, is a template for several further constructions. By convention, here and in what follows, our array indices start at one.

Consider the following question: take a field $\mathbb{K}$, an element x in $\mathbb{K}$ (or, as below, in an algebraic closure of it called $\overline{\mathbb{K}}$) and polynomials $r = [r_1, \ldots, r_N]$ in $\mathbb{K}[X]$. To x and r, we can associate the index $v(x, r) \in \{1, \ldots, N\}$, defined as the smallest i such that $r_i(x)$ is nonzero; if no such i exists, take $v(x, r) = \infty$. Computing $v(x, r)$ is easy, by evaluating all r_i's at x one after the other.

Let r be as before and let now P be non-constant and squarefree in $\mathbb{K}[X]$; let also V be the set of roots of P in $\overline{\mathbb{K}}$. The finite set V can be partitioned into non-empty sets $V_{v_1}, \ldots, V_{v_s}$, for some indices $v_i \in \{1, \ldots, N\} \cup \{\infty\}$, where V_{v_i} is the subset of all points x in V such that $v(x, r) = v_i$. Computing the partition $V_{v_1}, \ldots, V_{v_s}$ amounts to factoring P into (non-necessarily irreducible) factors $P_1, \ldots, P_s$, and finding indices $v_1, \ldots, v_s$ in $\{1, \ldots, N\} \cup \{\infty\}$, such that for all i in $\{1, \ldots, s\}$, the set of roots of P_i in $\overline{\mathbb{K}}$ is precisely V_{v_i} (remark that the P_i's and v_i's are uniquely defined, up to order). This is the object of the following algorithm called nonzero_index.

Lemma 9. *Suppose that P is squarefree of degree e, and that all r_i have degree less than e. Algorithm* nonzero_index *correctly returns $(P_1, v_1), \ldots, (P_s, v_s)$ as specified above, using $O^\sim(eN)$ operations in $\mathbb{K}$.*

Proof. Correctness is proved by seeing that at the beginning of each step i of the **for** loop, the roots of C are exactly the roots x of P for which $v(x, r) \geq i$, and that the roots of the gcd Z are then those roots x of P for which $v(x, R) > i$. Each pass through the loop takes $O^\sim(e)$ operations for gcd and exact division [18], so the cost estimate follows. □

Slightly more generally, consider polynomials

$$R = [\,[R_{1,1}, \ldots, R_{1,N}], \ldots, [R_{M,1}, \ldots, R_{M,N}]\,]$$

in $\mathbb{K}[X]$, and $x \in \overline{\mathbb{K}}$ as above. Then, to x and R, we want to associate the smallest index $i \in \{1, \ldots, M\}$ such that the vector $[R_{i,1}(x), \ldots, R_{i,N}(x)]$ is not identically zero (if it exists);

Algorithm 1: nonzero_index(P, r)

Input: P in $\mathbb{K}[X]$, $r = [r_1, \ldots, r_N]$ in $\mathbb{K}[X]^N$
Output: $L = [(P_1, v_1), \ldots, (P_s, v_s)]$, with $v_i \in \{1, \ldots, N\} \cup \{\infty\}$

```
1  L = [ ]
2  C = P
3  for i = 1, ..., N do
4  |   Z = gcd(C, r_i)
5  |   if Z is not constant then
6  |   |   append (C/Z, i) to L
7  |   C = Z
8  end
9  if C is not constant then
10 |   append (C, ∞) to L
11 return L
```

we also want to compute the smallest index $j \in \{1, \ldots, N\}$ such that $R_{i,j}(x)$ is nonzero, so that our output is $w(x, R) = (i, j)$. If no such i exists, instead of the pair (i, j), we return $w(x, R) = (\infty, \infty)$.

Given R and a squarefree polynomial P as before, we can then partition the zero-set $V \subset \overline{\mathbb{K}}$ of P into $V_{w_1}, \ldots, V_{w_t}$, such that for r in $\{1, \ldots, t\}$, V_{w_r} is the set of all $x \in V$ for which $w(x, R) = w_r$. As output, we thus return a sequence a polynomials $P_1, \ldots, P_t$, together with indices $w_1, \ldots, w_t$ in $(\{1, \ldots, M\} \times \{1, \ldots, N\}) \cup \{(\infty, \infty)\}$, such that for all i in $\{1, \ldots, t\}$, the set of roots of P_i in $\overline{\mathbb{K}}$ is precisely V_{w_i}.

This is done by the following algorithm, called nonzero_index_vectorial, which now takes as input P and the sequence of sequences of polynomials R. We use a subroutine called infinity(L) which takes as input a sequence $[(P_1, v_1), \ldots, (P_s, v_s)]$ such as the one computed by nonzero_index, and returns the polynomial P_i in it corresponding to $v_i = \infty$, if one such polynomial exists; otherwise, this subroutine returns 1.

Lemma 10. *Suppose that P is squarefree of degree e, and that all $R_{j,i}$ have degree less than e. Algorithm* nonzero_index_vectorial *correctly returns $(P_1, w_1), \ldots, (P_t, w_t)$ as specified above, using $O\tilde{}(eMN)$ operations in $\mathbb{K}$.*

Proof. Correctness is proved by seeing that at the beginning of each step i of the **for** loop, the roots of C are exactly the roots x of P for which all $R_{i',j}(x)$ vanish, for any $i' < i$ and $j \in \{1, \ldots, N\}$. After the call nonzero_index(C, r), $L' = [(P_{i,j}, v_{i,j}) \mid j \in D_i]$ contains the nonzero indices for $[R_{i,1} \bmod C, \ldots, R_{i,N} \bmod C]$, for some index set D_i. We remove from it the factor $C =$ infinity(L') (if it exists), which corresponds to those roots for which we will continue the process. At the end of the loops, C defines those roots of P that cancel all $R_{j,i}$, so we associate it with (∞, ∞).

For a given index i, the reductions at step 4 take $O\tilde{}(Ne)$ operations in $\mathbb{K}$, using fast Euclidean division. Calling nonzero_index takes $O\tilde{}(Ne)$ operations as well, in view of the

Algorithm 2: nonzero_index_vectorial(P, R)

Input: P in $\mathbb{K}[X]$, $R = [R_{1,1}, \ldots, R_{1,N}], \ldots, [R_{M,1}, \ldots, R_{M,N}]$ in $\mathbb{K}[X]^{M \times N}$
Output: $L = [(P_1, w_1), \ldots, (P_t, w_t)]$, $w_i \in (\{1, \ldots, M\} \times \{1, \ldots, N\}) \cup \{(\infty, \infty)\}$

$L = [\,]$
$C = P$
for $i = 1, \ldots, M$ **do**
  $r = [R_{i,j} \bmod C \mid j \in [1, \ldots, N]]$
  $L' = \mathsf{nonzero\_index}(C, r)$ L' has the form $L' = [(P_{i,j}, v_{i,j})]_{j \in D_i}$, $v_{i,j} \in \mathbb{N} \cup \{\infty\}$
  $C = \mathsf{infinity}(L')$
  if C is not constant **then**
    remove (C, ∞) from L'
  $L = L$ cat $[(P, (i, v)) \mid (P, v) \in L']$
end
if C is not constant **then**
  append $(C, (\infty, \infty))$ to L
return L

previous lemma. Summing these costs, we conclude the proof. □

4 Normal forms for derivatives

We now discuss some algorithms to compute normal forms of derivatives, inspired by techniques from [29]. These results will be crucial for our main algorithm in further sections.

The following notation will be useful: for positive integers $n_1, \ldots, n_s$, and for a ring $\mathbb{A}$, $\mathbb{A}[X_1, \ldots, X_s]_{n_1, \ldots, n_s}$ denotes the set of all $F \in \mathbb{A}[X_1, \ldots, X_s]$ such that $\deg(F, X_i) < n_i$ holds for all i. In all instances where we use this notation, we will have $s \in \{1, 2, 3\}$ (variables may then carry other names than $X_1, \ldots, X_s$).

In all this section, we work over the ring $\mathbb{A} = \mathbb{Z}/N\mathbb{Z}$, for some prime power $N = p^\ell$, using indeterminates X, ξ, ζ. Our input is as follows:

- $L = [(n_1, m_1), \ldots, (n_t, m_t)]$ is a list of pairs of integers.
- $L' = [P_1, \ldots, P_t]$ is a list of polynomials, with for all i, P_i monic of degree e_i in $\mathbb{A}[X]$. In addition, we suppose that for all i, j, with $i \neq j$, P_i and P_j generate the unit ideal in $\mathbb{A}[X]$. Equivalently, $P_i \bmod p$ and $P_j \bmod p$ are coprime in $\mathbb{F}_p[X]$.
- $L'' = [J_1, \ldots, J_t]$ is a list of polynomials, with for all i, J_i in $\mathbb{A}[X, \xi]_{e_i, n_i+1}$.
- F is a polynomial in $\mathbb{A}[X, Y]$.

As output, we want to compute the normal forms

$$D_{i,\mu} = \frac{\partial^\mu F}{\partial Y^\mu}(X + \xi, J_i) \bmod \langle P_i(X), \xi^{n_i+1} \rangle \in \mathbb{A}[X, \xi]_{e_i, n_i+1},$$

for all $i = 1, \ldots, t$ and $\mu = 0, \ldots, m_i$. This will be done by computing

$$F_i = F(X + \xi, J_i + \zeta) \bmod \langle P_i(X), \xi^{n_i+1}, \zeta^{m_i+1} \rangle, \tag{5}$$

for all $i = 1, \ldots, t$, since Taylor expansion shows that

$$D_{i,\mu} = \mu! \, \mathsf{cf}(F_i, \zeta^j),$$

where $\mathsf{cf}(P, \zeta^j)$ denotes the coefficient of ζ^j in a polynomial P. We will focus on the computation of the F_i's, since the overhead to deduce all $D'_{i,\mu}s$ by coefficient extraction and multiplication by $\mu!$'s will be negligible.

If for instance J_i does not depend on ξ, so it lies in $\mathbb{A}[X]_{e_i}$, and $2, \ldots, n_i$ are units in $\mathbb{A}$, $D_{i,\mu}$ can be written

$$D_{i,\mu} = \sum_{\nu=0}^{n_i} \frac{1}{\nu!} \frac{\partial^{\mu+\nu} F}{\partial X^\nu \partial Y^\mu}(X, J_i) \xi^\nu \bmod \langle P_i(X) \rangle;$$

knowing $D_{i,\mu}$ thus allows us to compute the normal forms of the derivatives $\frac{\partial^{\mu+\nu} F}{\partial X^\nu \partial Y^\mu}$ modulo $\langle P_i(X), Y - J_i(X) \rangle$, for all $i = 1, \ldots, t$, $\nu = 0, \ldots, n_i$ and $\mu = 0, \ldots, m_i$.

Suppose that F has degree d. We make the following assumption regarding the quantities n_i, m_i, e_i:

H_{NF}. The inequality $\sum_{1 \le i \le t} (n_i + 1)(m_i + 1)e_i = O(d^2)$ holds.

Representing F requires approximately d^2 coefficients in $\mathbb{A}$, and representing all P_i's and J_i's uses about $\sum_{1 \le i \le t} (n_i + 1)e_i$ extra coefficients. On the other hand, for all i, F_i lies in $\mathbb{A}[X, \xi, \zeta]_{e_i, n_i+1, m_i+1}$, so representing all of them uses $\sum_{1 \le i \le t} (n_i + 1)(m_i + 1)e_i$ coefficients in $\mathbb{A}$. Thus, assumption H_{NF} means that input and output sizes add up to about d^2 elements of $\mathbb{A} = \mathbb{Z}/N\mathbb{Z}$, or $d^2 \log(N)$ bits.

The main result in this section is the following proposition, which shows that all F_i can be computed in essentially linear time.

Proposition 3. *Under assumption* H_{NF}*, for any* $\varepsilon > 0$*, there exists an algorithm* normal_forms *that takes as input a prime power* $N = p^\ell$*, sequences* L, L', L'' *and polynomial* F *as above, and returns all* F_i*, for* i *in* $\{1, \ldots, t\}$*, using* $d^{2+\varepsilon} O\tilde{}(\log(N))$ *bit operations.*

4.1 Auxiliary results

A first normal form algorithm. The central problem for these normal form questions is normal form computation modulo a single *triangular set* $\mathbf{T} = (P(X), Q(X, Y))$, where $P \in \mathbb{A}[X]$ is monic in X and $Q \in \mathbb{A}[X, Y]$ is monic in Y, reduced with respect to P.

Suppose that $\deg(P, X) = f$ and $\deg(Q, Y) = e$. Given F in $\mathbb{A}[X, Y]$, the question is to compute $F \bmod \langle P, Q \rangle \in \mathbb{A}[X, Y]_{f,g}$. This apparently simple question is actually quite challenging; so far, no algorithm is known to solve it in optimal time in an *algebraic* complexity model.

In our particular context of computations modulo N, however, better results are available. Building on seminal results by Kedlaya and Umans [23], Theorem 6 in [39] gives a quasi-linear *bit complexity* result for such a task (as pointed out in [29], this result was originally proved for N a prime, but carries over without modification to the case of a prime power).

Lemma 11. *For any $\varepsilon > 0$, there exists an algorithm* normal_form_bivariate *with the following input:*

- *a prime power N;*
- *F in $\mathbb{A}[X,Y]_{m,n}$, with $\mathbb{A} = \mathbb{Z}/N\mathbb{Z}$,*
- *a triangular set $\mathbf{T} = (P(X), Q(X,Y))$, with P in $\mathbb{A}[X]$, monic of degree f, and Q in $\mathbb{A}[X,Y]$, monic in Y of degree g and of degree in X less than e.*

This algorithm returns $F \bmod \langle \mathbf{T} \rangle \in \mathbb{A}[X,Y]_{f,g}$ using $(mn+fg)^{1+\varepsilon} O^{\sim}(\log(N))$ bit operations.

Remark that up to the exponent ε, this algorithm is optimal, since storing input and output involves $\Theta(mn + ef)$ coefficients in $\mathbb{A} = \mathbb{Z}/N\mathbb{Z}$, for a total of $\Theta((mn + ef)\log(N))$ bits of storage.

Using this result, Proposition 3 in [29] states the following extension towards the reduction of one polynomial F modulo *several* bivariate triangular sets.

Lemma 12. *Let $\mathbf{T}_1, \ldots, \mathbf{T}_s$ be triangular sets in $\mathbb{A}[X,Y]$, where for $i = 1, \ldots, s$ $\mathbf{T}_i = (P_i(X), Q_i(X,Y))$, with P_i monic in X of degree f_i and $Q_i(X,Y)$ monic in Y of degree g_i, and reduced with respect to X. Suppose that for all i, j in $\{1, \ldots, s\}$, with $i \neq j$, P_i and P_j generate the unit ideal in $\mathbb{A}[X]$.*

Let F be in $\mathbb{A}[X,Y]$ with degree d, and suppose that $\sum_{i \leq s} f_i g_i = O(d^2)$. Then, for any $\varepsilon > 0$, there exists an algorithm normal_forms_bivariate *that takes as input the prime power N, $\mathbf{T}_1, \ldots, \mathbf{T}_s$ and F as above, and returns all $F \bmod \langle \mathbf{T}_i \rangle$, for i in $\{1, \ldots, s\}$, using $d^{2+\varepsilon} O^{\sim}(\log(N))$ bit operations.*

As in Proposition 3, the input and output sizes are $\Theta(d^2)$ elements of $\mathbb{A}$, so the running time is close to optimal. This lemma will be our main tool to prove Proposition 3; most of the work in this section will consist in reducing our original problem to an instance of the bivariate problem above.

Remark that there are two slight differences between the lemma above and the one stated in reference [29]. First, that result seemingly required another assumption, namely that all g_i should satisfy $g_i \leq d$. This is actually not needed: the article [29] gave an alternative solution to this problem, valid in an algebraic complexity model (over an arbitrary ring), that did require such an assumption; the property $g_i \leq d$ was assumed to hold throughout for simplicity. In our context, we can safely omit it.

Another slight difference is that the result in [29] required as an extra input the inverses of $(P_1 \cdots P_{i-1} P_{i+1} \cdots P_s)$ modulo P_i, for all $i = 1, \ldots, s$. It was then pointed out that in the case $\mathbb{A} = \mathbb{Z}/N\mathbb{Z}$, for N a prime power, they can be computed in $O^{\sim}(d^2 \log(N))$ operations, which will be negligible. Thus, our assumptions are not restrictive.

An easy change of order. Our next auxiliary result is an explicit change of order algorithm for a particular bivariate ideal in $\mathbb{A}[X, Z]$. Several references give algorithms to perform this kind of operations [4, 37, 39], but we are not aware of a complexity result that would apply in our particular case (for instance, the change of order algorithms of [37, 39] require a radical ideal over a field, but none of these conditions apply here). Nevertheless, the situation is simple enough that we can give an explicit solution.

Lemma 13. *Let P be monic of degree e in $\mathbb{A}[X]$, such that $P \bmod p$ is squarefree in $\mathbb{F}_p[X]$, and let n be a positive integer.*

There exists an algorithm change_order_special *that computes using $\tilde{O}(en\log(N))$ bit operations a polynomial V in $\mathbb{A}[Z]$ of degree less than en, such that in $\mathbb{A}[X, Z]$, we have the following equality between ideals:*

$$\langle P(X), (Z - X)^n\rangle = \langle P(Z)^n, X - V(Z)\rangle.$$

Proof. Let $P^\star$ be an arbitrary monic lift of P to $\mathbb{Z}_p[X]$, where $\mathbb{Z}_p$ is the ring of p-adic integers. Because $P \bmod p$ is squarefree, $P^\star$ is squarefree as well. In the first part of the proof, we work over $\mathbb{Z}_p$, its field of fractions $\mathbb{Q}_p$, and an algebraic closure of it, $\overline{\mathbb{Q}_p}$.

Let $a_1, \ldots, a_e$ be the (unknown) pairwise distinct roots of $P^\star$ in $\overline{\mathbb{Q}_p}$. Then, the ideal $\langle P^\star(X), (Z - X)^n\rangle$ is the product of the pairwise coprime ideals

$$\left|\begin{array}{ll} (Z - a_i)^n & \\ X - a_i, & i = 1, \ldots, e. \end{array}\right.$$

For such ideals, changing the order of X and Z is straightforward. We deduce that the polynomial $V^\star$ of degree less than en defined by the Chinese Remainder conditions

$$V^\star \bmod (Z - a_i)^n = a_i, \quad i = 1, \ldots, e$$

satisfies the equality $\langle P^\star(X), (Z - X)^n\rangle = \langle P^\star(Z)^n, X - V^\star(Z)\rangle$, except that $V^\star$ is *a priori* in $\overline{\mathbb{Q}_p}[Z]$, and the equality holds in $\overline{\mathbb{Q}_p}[X, Z]$.

Let us write $Q = P^\star(Z)^n$. To compute $V^\star$, we define the polynomials of degree at most $(e - 1)n$

$$A = \sum_{i=1}^{e} a_i \prod_{i' \neq i} (Z - a_{i'})^n \quad \text{and} \quad B = \sum_{i=1}^{e} \prod_{i' \neq i} (Z - a_{i'})^n.$$

First, let us show how to compute A and B; we will show as we go that both A and B are in $\mathbb{Z}_p[Z]$.

Let $\tilde{A}$ and $\tilde{B}$ be the reverse polynomials $Z^{(e-1)n}A(1/Z)$ and $Z^{(e-1)n}B(1/Z)$; define similarly $\tilde{Q} = Z^{en}Q(1/Z)$, so that we have

$$\tilde{A} = \sum_{i=1}^{e} a_i \prod_{i' \neq i} (1 - a_{i'}Z)^n, \quad \tilde{B} = \sum_{i=1}^{e} \prod_{i' \neq i} (1 - a_{i'}Z)^n$$

and

$$\tilde{Q} = \prod_{i=1}^{e} (1 - a_i Z)^n.$$

Let us first show how to compute the power series expansions of the rational functions $\tilde{A}/\tilde{Q}$ and $\tilde{B}/\tilde{Q}$. Consider the power series

$$\frac{1}{(1-Z)^n} = \sum_{j \geq 0} c_j Z^j \quad \text{and} \quad S = \sum_{j \geq 0} s_j Z^j,$$

where $s_j = a_1^j + \cdots + a_n^j$ is the jth power sum of $P^\star$, so that all c_j's and s_j's are in $\mathbb{Z}_p$. The rational functions $\tilde{A}/\tilde{Q}$ and $\tilde{B}/\tilde{Q}$ can then be written as

$$\frac{\tilde{A}}{\tilde{Q}} = \sum_{i=1}^{e} \frac{a_i}{(1-a_i Z)^n} = \sum_{i=1}^{e} \sum_{j \geq 0} a_i^{j+1} c_j Z^j = \sum_{j \geq 0} c_j s_{j+1} Z^j,$$

$$\frac{\tilde{B}}{\tilde{Q}} = \sum_{i=1}^{e} \frac{1}{(1-a_i Z)^n} = \sum_{i=1}^{e} \sum_{j \geq 0} a_i^{j} c_j Z^j = \sum_{j \geq 0} c_j s_j Z^j,$$

so they lie in $\mathbb{Z}_p[[Z]]$. Upon multiplication by $\tilde{Q}$, we deduce that $\tilde{A}$ and $\tilde{B}$ are both in $\mathbb{Z}_p[Z]$, and so are A and B, as claimed.

In addition, we claim that B is invertible modulo Q, not only in $\mathbb{Q}_p[Z]$, but actually in $\mathbb{Z}_p[Z]$. Indeed, the resultant of Q and B is (up to sign) the n^2-th power of the discriminant of $P^\star$, which is by assumption a unit in $\mathbb{Z}_p$.

Finally, one verifies that $A/B \bmod (Z - a_i)^n = a_i$, which implies that $V^\star = A/B \bmod Q$. In particular, $V^\star$ is in $\mathbb{Z}_p[Z]$, as announced before.

So far, the Chinese Remainder conditions we used have established the equality $\langle P^\star(X), (Z - X)^n \rangle = \langle Q(Z), X - V^\star(Z) \rangle$ in $\overline{\mathbb{Q}_p}[X, Z]$. However, since all polynomials are in $\mathbb{Z}_p[X, Z]$, and monic in their leading variables, we deduce that the underlying membership identities hold in $\mathbb{Z}_p[X, Z]$ as well. Truncating modulo N, and defining $V = V^\star \bmod N \in \mathbb{A}[Z]$, we conclude that the equality $\langle P(X), (Z - X)^n \rangle = \langle P(Z)^n, X - V(Z) \rangle$ holds in $\mathbb{A}[X, Z]$.

Finally, we turn to the cost analysis. We can compute all coefficients c_j and s_j at precision en using $O^{\tilde{}}(en)$ operations in $\mathbb{A}$, and thus $O^{\tilde{}}(en \log(N))$ bit operations: for the former, this is for instance done by computing $(1 - X)^n$ by binary powering and inverting it; for the latter, this is in [44].

Once we know the coefficients c_j and s_j, we recover $\tilde{A}$ and $\tilde{B}$ through multiplication by Q and reversal, for another $O^{\tilde{}}(en)$ operations in $\mathbb{A}$, and A and B are deduced for free. The last non-obvious step is the computation of $1/B \bmod Q$ (since the rest is just another multiplication modulo Q). This is done using Newton iteration: the inverse of B modulo $\langle p, Q \rangle$ can be computed using the fast extended gcd algorithm in $\mathbb{F}_p[Z]$ in $O^{\tilde{}}(en)$ operations modulo p; then, Newton iteration for inverse gives us $1/B \bmod Q$ in $\mathbb{A}[Z]$ in quasi-linear time $O^{\tilde{}}(en \log(N))$. Summing all costs above gives the claimed overall running time. □

All notation being as in the lemma, we deduce that we have an isomorphism

$$\psi : \mathbb{A}[X,Z]/\langle P(X), (Z-X)^n\rangle \to \mathbb{A}[Z]/\langle P(Z)^n\rangle.$$

Taking $\mathbb{A}[X,Z]_{e,n}$ and $\mathbb{A}[Z]_{en}$ for representatives of respectively the left and right-hand sides, ψ is given by

$$\psi(R) = R \bmod \langle P(Z)^n, X - V(Z)\rangle$$

for R in $\mathbb{A}[X,Z]_{n,e}$, and

$$\psi^{-1}(S) = S \bmod \langle P(X), (Z-X)^n\rangle$$

for S in $\mathbb{A}[Z]_{en}$. Once V is known, applying Lemma 11, we deduce in particular that for any $\varepsilon > 0$, both change of bases ψ and ψ^{-1} can be performed in $(en)^{1+\varepsilon}O^{\tilde{}}(\log(N))$ bit operations.

4.2 Proof of Proposition 3

Recall that on input sequences L, L', L'' our goal is to compute normal forms

$$F_i = F(X+\xi, J_i+\zeta) \bmod \langle P_i(X), \xi^{n_i+1}, \zeta^{m_i+1}\rangle,$$

for $i = 1, \ldots, t$. We now show how to perform this operation, thereby proving Proposition 3: assuming that H_{NF} holds, that is, $\sum_{1\le i\le t}(n_i+1)(m_i+1)e_i = O(d^2)$, we can compute all F_i, for i in $\{1,\ldots,t\}$, using $d^{2+\varepsilon}O^{\tilde{}}(\log(N))$ bit operations. This will be done by reducing this problem to an instance of a bivariate normal form computation, that can be handled with Algorithm normal_forms_bivariate from Lemma 12.

Let us fix i in $\{1,\ldots,t\}$, and let Z and T be two new variables. We will use them through the change of variables $Z = X+\xi$, $T = J_i+\zeta$.

First change of variables. First, we consider the introduction of the variable Z, that stands for $X+\xi$. In most of this paragraph, the index $i \in \{1,\ldots,t\}$ is fixed. For any such i, there is an $\mathbb{A}$-algebra isomorphism

$$\phi_i : \mathbb{A}[X,\xi]/\langle P_i(X), \xi^{n_i+1}\rangle \to \mathbb{A}[X,Z]/\langle P_i(X), (Z-X)^{n_i+1}\rangle.$$

The left-hand side and right-hand side respectively admit the polynomials in $\mathbb{A}[X,\xi]_{e_i,n_i+1}$ and $\mathbb{A}[X,Z]_{e_i,n_i+1}$ as canonical representatives. With these representatives, we have, for R in $\mathbb{A}[X,\xi]_{e_i,n_i+1}$, $\phi_i(R) = R(X, Z-X) \bmod P_i$. The inverse mapping is given by $\phi_i^{-1}(S) = S(X,\xi+X) \bmod P_i$, for S in $\mathbb{A}[X,Z]_{e_i,n_i+1}$.

Lemma 14. *The following holds:*

- *For R in $\mathbb{A}[X,\xi]_{e_i,n_i+1}$, one can compute $\phi_i(R)$ using $O^{\tilde{}}(e_i n_i \log(N))$ bit operations.*
- *For S in $\mathbb{A}[X,Z]_{e_i,n_i+1}$, one can compute $\phi_i^{-1}(S)$ using $O^{\tilde{}}(e_i n_i \log(N))$ bit operations.*

Proof. We give the proof for ϕ_i; that for ϕ_i^{-1} is entirely similar. Define $\mathbb{B}_i = \mathbb{A}[X]/\langle P_i \rangle$. Computing $\phi_i(R)$ amounts to seeing R in $\mathbb{B}_i[\xi]$, and computing $R(\xi - X)$ in that ring (and finally, formally replacing ξ by Z). This is thus an instance of *shifting* a polynomial, in this case by $-X$. Since R has degree less than n_i in ξ, the divide-and-conquer algorithm of [17] solves this problem in $O^{\sim}(n_i)$ operations $(+, \times)$ in $\mathbb{B}_i$, which is $O^{\sim}(e_i n_i)$ operations $(+, \times)$ in $\mathbb{A}$, and thus $O^{\sim}(e_i n_i \log(N))$ bit operations. □

The mapping ϕ_i can then be extended to a change of a variables

$$\Phi_i : \mathbb{A}[X, \xi, \zeta]/\langle P_i(X), \xi^{n_i+1}, \zeta^{m_i+1} \rangle \to \mathbb{A}[X, Z, \zeta]/\langle P_i(X), (Z - X)^{n_i+1}, \zeta^{m_i+1} \rangle,$$

which acts coefficient-wise in ζ; by the previous lemma, both Φ_i and its inverse Φ_i^{-1} can thus be computed in $O^{\sim}(e_i n_i m_i \log(N))$ bit operations. Taking all $i \in \{1, \ldots, t\}$ into account, and using assumption H_{NF}, the cost becomes $O^{\sim}(d^2 \log(N))$ bit operations.

For i in $\{1, \ldots, t\}$, let us finally write $J_i^{(1)} = \phi_i(J_i)$, so that $J_i^{(1)}$ lies in $\mathbb{A}[X, Z]_{e_i, n_i+1}$. Defining

$$F_i^{(1)} = F(Z, J_i^{(1)} + \zeta) \bmod \langle P_i(X), (Z - X)^{n_i+1}, \zeta^{m_i+1} \rangle,$$

we see that F_i as showed in (5) can be recovered as $\Phi_i^{-1}(F_i^{(1)})$.

Since we saw that applying all changes of variables ϕ_i, or even Φ_i, and their inverses, takes time $O^{\sim}(d^2 \log(N))$, we can now focus on computing the polynomials $F_i^{(1)}$, for $i = 1, \ldots, t$.

Second change of variables. Our second change of variables is actually a change of order. As before, for the following discussion, we fix an index i in $\{1, \ldots, t\}$.

Applying Lemma 13, we deduce that we can compute in $O^{\sim}(e_i n_i \log(N))$ bit operations a polynomial V_i in $\mathbb{A}[Z]$ such that we have the equality between ideals

$$\langle P_i(X), (Z - X)^{n_i+1} \rangle = \langle P_i(Z)^{n_i+1}, X - V_i(X) \rangle$$

in $\mathbb{A}[X, Z]$. In addition, we saw that for any $\varepsilon > 0$, the change of basis

$$\psi_i : \mathbb{A}[X, Z]/\langle P_i(X), (Z - X)^{n_i+1} \rangle \to \mathbb{A}[Z]/\langle P_i(Z)^{n_i+1} \rangle$$

and its inverse can be performed in $(e_i n_i)^{1+\varepsilon} O^{\sim}(\log(N))$ bit operations. As above, the mapping ψ_i can be extended to a change of basis

$$\Psi_i : \mathbb{A}[X, Z, \zeta]/\langle P_i(X), (Z - X)^{n_i+1}, \zeta^{m_i+1} \rangle \to \mathbb{A}[Z, \zeta]/\langle P_i(Z)^{n_i+1}, \zeta^{m_i+1} \rangle$$

which acts coefficient-wise in ζ. Both Ψ_i and its inverse Ψ_i^{-1} can thus be computed in $(e_i n_i)^{1+\varepsilon} O^{\sim}(m_i \log(N))$ bit operations; using assumption H_{NF}, the cost for all i in $\{1, \ldots, t\}$ is thus $d^{2+\varepsilon} O^{\sim}(\log(N))$ bit operations.

For i in $\{1, \ldots, t\}$, let us finally write $J_i^{(2)} = \psi_i(J_i^{(1)})$, so that $J_i^{(2)}$ lies in $\mathbb{A}[Z]_{e_i(n_i+1)} \simeq \mathbb{A}[Z]/\langle P_i(Z)^{n_i+1} \rangle$. Defining

$$F_i^{(2)} = F(Z, J_i^{(2)} + \zeta) \bmod \langle P_i(Z)^{n_i+1}, \zeta^{m_i+1} \rangle,$$

we see that $F_i^{(1)}$ can be recovered as $\Psi_i^{-1}(F_i^{(2)})$. Thus, since the change of bases take quasi-linear time $d^{2+\varepsilon} O^{\sim}(\log(N))$, we can now focus on computing the normal forms $F_i^{(2)}$, for $i = 1, \ldots, t$.

Third change of variables. Our last change of variables introduces a new variable T which will stand for $J_i^{(2)} + \zeta$. In the same vein as what we said for the introduction of variable Z, we can now notice that there is an $\mathbb{A}$-algebra isomorphism

$$\gamma_i : \mathbb{A}[Z, \zeta]/\langle P_i(Z)^{n_i+1}, \zeta^{m_i+1}\rangle \to \mathbb{A}[Z, T]/\langle P_i(Z)^{n_i+1}, (T - J_i^{(2)})^{m_i+1}\rangle.$$

The left-hand side and right-hand side admit respectively the elements of $\mathbb{A}[Z, \zeta]_{e_i(n_i+1),m_i+1}$ and $\mathbb{A}[Z, T]_{e_i(n_i+1),m_i+1}$ as canonical representatives. With these representatives, we have, for R in $\mathbb{A}[Z, \zeta]_{e_i(n_i+1),m_i+1}$, $\gamma_i(R) = R(Z, T - J_i^{(2)}) \bmod P_i^{n_i+1}$. The inverse mapping is given by $\gamma_i^{-1}(S) = S(Z, \zeta + J_i^{(2)}) \bmod P_i^{n_i+1}$, for S in $\mathbb{A}[Z, T]_{e_i(n_i+1),m_i+1}$.

Proceeding exactly as in Lemma 14, we deduce that we can compute γ_i or its inverse in $O^{\tilde{}}(e_i n_i m_i \log(N))$ bit operations. Defining finally

$$F_i^{(3)} = F(Z, T) \bmod \langle P_i(Z)^{n_i+1}, (T - J_i^{(2)})^{m_i+1}\rangle,$$

we deduce that $F_i^{(2)} = \gamma_i^{-1}(F_i^{(3)})$. Once more, the changes of variables take quasi-linear time, so we are left with the problem of computing the polynomials

$$F(Z, T) \bmod \langle P_i(Z)^{n_i+1}, (T - J_i^{(2)})^{m_i+1}\rangle,$$

for $i = 1, \ldots, t$. Since the polynomials $(P_i(Z)^{n_i+1}, P_j(Z)^{n_j+1})$ generate the unit ideal in $\mathbb{A}[Z]$ (for $i \neq j$), this can be done as a direct application of Algorithm normal_forms_bivariate from Lemma 12.

For i in $\{1, \ldots, t\}$, the polynomials defining the triangular set $(P_i(Z)^{n_i+1}, (T - J_i^{(2)})^{m_i+1})$ have respective degrees in their main variables $e_i(n_i + 1)$ and $m_i + 1$. Using once more assumption H_{NF}, we deduce that the total cost for all indices i is $d^{2+\varepsilon} O^{\tilde{}}(\log(N))$ bit operations, for any $\varepsilon > 0$. Adding up all costs seen so far, we conclude the proof of Proposition 3.

5 A deflation lemma

Consider a polynomial system $F = G = 0$ in $\mathbb{K}[X, Y]$, where $\mathbb{K}$ is a field, and an isolated solution of it $(x, y) \in \overline{\mathbb{K}}^2$, where $\overline{\mathbb{K}}$ is an algebraic closure of $\mathbb{K}$. Most extensions of Newton iteration to the case where (x, y) has multiplicity $M > 1$ seek to replace the given system with a new one, say ψ, such that the multiplicity of ψ at the root (x, y) is less than M; eventually, we reach $M = 1$, where we can apply Newton iteration without difficulty. Such a process is called *deflation*.

Since our main algorithm is based on a form of Newton iteration, we will need to employ such techniques in order to handle multiple roots. This section presents a particular deflation process initially due to Lecerf [30], which will be at the heart of our main algorithm.

In this approach, the deflated systems are constructed by considering suitable derivatives of the given system $\langle F, G\rangle$. The following construction assigns to an isolated solution (x, y) of $F = G = 0$ a *signature* $\sigma(x, y)$, of the form $\sigma(x, y) = (m, \mathtt{H}, n, a, \mathtt{K})$ (defined precisely below). In essence, this signature predicts which derivatives of F, G should be taken to

reach a deflated ideal ψ satisfying the multiplicity reduction requirement. This definition is inspired by that of a *generic trace* in [30].

The deflation lemma below is the key to this construction; it follows very closely [30, Lemma 4]. We introduce a small modification: the initial approach works in generic coordinates, so following it would require us to perform the corresponding probability analysis, which appears not to be straightforward. Instead, in our bivariate setting, it is possible to bypass generic coordinates; the resulting proof is very similar to Lecerf's, up to minor changes (for instance, in the definition of the integer m below, or the monomial order we use).

Lemma 15. *Let $\langle F, G\rangle \subset$ be an ideal in $\mathbb{K}[X, Y]$, with F and G of degree at most d, and let $(x, y) \in \overline{\mathbb{K}}^2$ be an isolated root of $\langle F, G\rangle$ with multiplicity M. Define*

$$m = \min\left\{\mu : \frac{\partial^\mu F}{\partial Y^\mu}(x, y) \neq 0 \quad or \quad \frac{\partial^\mu G}{\partial Y^\mu}(x, y) \neq 0\right\}.$$

If $\mathbb{K}$ has characteristic greater than d, then

(a) $m \geq 1$;

(b) $m \leq d$;

(c) (x, y) *is a root of* ψ *with multiplicity* n, *for some integer* n *satisfying* $1 \leq n \leq M/m$, *where*

$$\psi = \left\langle F, G, \frac{\partial F}{\partial Y}, \frac{\partial G}{\partial Y}, \ldots, \frac{\partial^{m-1} F}{\partial Y^{m-1}}, \frac{\partial^{m-1} G}{\partial Y^{m-1}}\right\rangle.$$

Proof. In what follows, we denote by I the ideal $\langle F, G\rangle$. Upon translating the origin to (x, y), we can assume without loss of generality that $x = y = 0$. To prove the first item, note that $F(0, 0) = G(0, 0) = 0$, which implies $m > 0$.

Let us next prove that m is finite, and bounded from above by d. Because $\mathbb{K}$ has characteristic greater than d, if all partial derivatives of F with respect to $Y, Y^2, \ldots, Y^d$ vanish at $(0, 0)$, $F(0, Y)$ must be the zero polynomial (recall that F has degree at most d), so that X divides F. If this is the case for G as well, X divides both F and G, so $(0, 0)$ is not an isolated solution of $F = G = 0$, a contradiction. Thus, we have proved (a) and (b).

Using the equalities

$$\frac{\partial^\mu F}{\partial Y^\mu}(0, 0) = 0 \quad \text{and} \quad \frac{\partial^\mu G}{\partial Y^\mu}(0, 0) = 0, \quad 0 \leq \mu \leq m - 1$$

and

$$\psi = \left\langle F, G, \frac{\partial F}{\partial Y}, \frac{\partial G}{\partial Y}, \cdots, \frac{\partial^{m-1} F}{\partial Y^{m-1}}, \frac{\partial^{m-1} G}{\partial Y^{m-1}}\right\rangle,$$

it is clear that $(0, 0)$ is a root of ψ, so $n \geq 1$. It remains to prove the upper bound $n \leq M/m$.

We are going to work locally, by looking at F, G and their derivatives in $\overline{\mathbb{K}}[[X,Y]]$. By the definition of the multiplicity, we have

$$M = \dim_{\overline{\mathbb{K}}} \overline{\mathbb{K}}[[X,Y]]/I \quad \text{and} \quad n = \dim_{\overline{\mathbb{K}}} \overline{\mathbb{K}}[[X,Y]]/\psi.$$

We are going to describe more precisely these residue class rings. Let us endow $\overline{\mathbb{K}}[[X,Y]]$ with the order defined by

$$X^{a_1}Y^{b_1} > X^{a_2}Y^{b_2} \iff a_1 < a_2 \quad \text{or} \quad a_1 = a_2 \text{ and } b_1 < b_2.$$

One verifies that this order is compatible with multiplication, and that $1 > X$ and $1 > Y$ both hold. This is thus a *local monomial order*, as in [9, Chapter 4]; precisely, this is a reverse lexicographic order.

To any power series S in $\overline{\mathbb{K}}[[X,Y]]$, we can associate its *leading monomial* $\mathrm{lm}(S)$ with respect to this order; as usual, this notation carries over to ideals in $\overline{\mathbb{K}}[[X,Y]]$.

From [9, Theorem 4.3], we infer that the monomials in $\mathrm{lm}(I)^c$ and $\mathrm{lm}(\psi)^c$ – where the exponent ^{c} denotes complement – form bases of respectively $\overline{\mathbb{K}}[[X,Y]/I$ and $\overline{\mathbb{K}}[[X,Y]/\psi$. In particular, the numbers of these monomials are respectively M and n. Define

$$T = \{X^a Y^{m-1} \in \mathrm{lm}(I)^c \mid a \geq 0\}.$$

Because $\mathrm{lm}(I)$ is stable by multiplication, for each element X^aY^{m-1} of T, all monomials X^aY^b, for $0 \leq b \leq m-1$, are in $\mathrm{lm}(I)^c$, whence $M = |\mathrm{lm}(I)^c| \geq m|T|$. Equivalently, we have $|T| \leq M/m$.

We now prove that n is at most $|T|$, which is enough to conclude, since we then have $n \leq |T| \leq M/m$. To establish that $n \leq |T|$, we will prove in the two items below that $\mathrm{lm}(\psi)^c$ is contained in $\{X^a \mid 0 \leq a < |T|\}$.

- The definition of m implies that at least one of $\frac{\partial^m F}{\partial Y^m}$ or $\frac{\partial^m G}{\partial Y^m}$ does not vanish at $(0,0)$; let us assume without loss of generality that this is the case for $\frac{\partial^m F}{\partial Y^m}$. This implies that for $b = 0, \ldots, m-1$, the coefficient of the monomial Y^b in F is zero, while that of Y^m is nonzero. The definition of our local order then implies that Y^m is the leading term of F. Thus, Y^m is in $\mathrm{lm}(I)$, so that X^aY^m is in $\mathrm{lm}(I)$ for any $a \geq 0$.

  Take $a \geq 0$ and consider an element $P \in I$ having leading monomial X^aY^m. Because $m \leq d$, and due to our assumption on the characteristic of $\overline{\mathbb{K}}$, we deduce that the leading monomial of $\frac{\partial^{m-1} P}{\partial Y^{m-1}}$ is $m!X^aY$. Because $\frac{\partial^{m-1} P}{\partial Y^{m-1}}$ is in the ideal ψ, this shows that for $a \geq 0$, X^aY is in $\mathrm{lm}(\psi)$. In other words, all elements of $\mathrm{lm}(\psi)^c$ are of the form X^a for some $a \geq 0$.

- By definition of T, $X^{|T|}Y^{m-1}$ is in $\mathrm{lm}(I)$. Differentiating $m-1$ times as above, we deduce that $X^{|T|}$ is in $\mathrm{lm}(\psi)$.

This proves that $\mathrm{lm}(\psi)^c$ is contained in $\{X^a \mid 0 \leq a < |T|\}$, as claimed above. □

This lemma allows us to define the first components $m, \mathtt{H}$ of the signature $\sigma(x,y)$:

- m is defined as in the lemma;
- the string $\mathtt{H} \in \{\text{"}\mathtt{F}\text{"}, \text{"}\mathtt{G}\text{"}\}$ indicates which of $\frac{\partial^m F}{\partial Y^m}$ and $\frac{\partial^m G}{\partial Y^m}$ is nonzero at (x, y); in case of a tie, for definiteness, we choose F.

Using the same notation as above, let us define the polynomial $H = F$ (if $\mathtt{H} = \text{"}\mathtt{F}\text{"}$) or $H = G$ (if $\mathtt{H} = \text{"}\mathtt{G}\text{"}$), so that

$$m = \min\left\{\mu : \frac{\partial^\mu H}{\partial Y^\mu}(x, y) \neq 0\right\};$$

in particular, (x, y) is a root of $\frac{\partial^{m-1} H}{\partial Y^{m-1}}$, but not of $\frac{\partial^m H}{\partial Y^m}$. We also define H^c (the "complement" of H) as either $H^c = G$ if $H = F$ and $H^c = F$ if $H = G$.

The invertibility assumption of $\frac{\partial^m H}{\partial Y^m}(x, y)$ allows us to apply the implicit function theorem to $\frac{\partial^{m-1} H}{\partial Y^{m-1}}$ at (x, y). Replacing X by $x + \xi$, where ξ is a new variable, we can find a power series J^∞ in $\overline{\mathbb{K}}[[\xi]]$ such that

$$\frac{\partial^{m-1} H}{\partial Y^{m-1}}(x + \xi, J^\infty) = 0, \quad J^\infty(0) = y;$$

in particular, there exists A in $\overline{\mathbb{K}}[[\xi]][Y]$ such that

$$\frac{\partial^{m-1} H}{\partial Y^{m-1}}(x + \xi, Y) = (Y - J^\infty)A \quad \text{and} \quad A(0, y) \neq 0.$$

Let us further replace Y by $y + \zeta$, where ζ is a new variable, and let us work in the power series ring $\overline{\mathbb{K}}[[\xi, \zeta]]$. Since $A(0, y)$ is nonzero, $A(\xi, y + \zeta)$ is a unit in $\overline{\mathbb{K}}[[\xi, \zeta]]$. We deduce that in $\overline{\mathbb{K}}[[\xi, \zeta]]$, we have the equality between ideals

$$\left\langle \frac{\partial^{m-1} H}{\partial Y^{m-1}}(x + \xi, y + \zeta) \right\rangle = \langle \zeta - (J^\infty - y) \rangle .$$

Consider now the following system:

$$\psi = \left\langle F, G, \frac{\partial F}{\partial Y}, \frac{\partial G}{\partial Y}, \cdots, \frac{\partial^{m-1} F}{\partial Y^{m-1}}, \frac{\partial^{m-1} G}{\partial Y^{m-1}} \right\rangle .$$

The previous lemma implies that (x, y) is a root of ψ of multiplicity n, with $n \leq M/m$. Replacing as above X by $x + \xi$ and Y by $y + \zeta$, and noticing that $\{F, G\} = \{H, H^c\}$, the previous remark shows that in $\overline{\mathbb{K}}[[\xi, \zeta]]$, $(0, 0)$ is a root of multiplicity n of the ideal generated by

$$\left(\frac{\partial^\alpha H}{\partial Y^\alpha}(x + \xi, J^\infty)\right)_{0 \leq \alpha < m-1}, \quad \zeta - (J^\infty - y), \quad \left(\frac{\partial^\alpha H^c}{\partial Y^\alpha}(x + \xi, J^\infty)\right)_{0 \leq \alpha < m} .$$

For $\alpha \geq 0$, define the power series in $\overline{\mathbb{K}}[[\xi]]$

$$H_\alpha = \frac{\partial^\alpha H}{\partial Y^\alpha}(x + \xi, J^\infty) \quad \text{and} \quad H_\alpha^c = \frac{\partial^\alpha H^c}{\partial Y^\alpha}(x + \xi, J^\infty),$$

so that the above ideal is generated by

$$\left\langle H_0, H_0^c, H_1, H_1^c, \ldots, H_{m-2}, H_{m-2}^c, \zeta - (J^\infty - y), H_{m-1}^c \right\rangle .$$

Remark that, with the exception of $\zeta - (J^\infty - y)$, all the above generators are in $\overline{\mathbb{K}}[[\xi]]$. Since $\zeta - (J^\infty - y)$ has degree one in ζ, we deduce that 0 is a root of multiplicity n of the ideal

$$\left\langle H_0, H_0^c, H_1, H_1^c, \ldots, H_{m-2}, H_{m-2}^c, H_{m-1}^c \right\rangle \subset \overline{\mathbb{K}}[[\xi]].$$

This proves in particular the following lemma.

Lemma 16. *The integer n satisfies*

$$n = \min\left(\{\mathrm{val}(H_\alpha)\}_{0 \le \alpha < m-1} \ \cup \ \{\mathrm{val}(H_\alpha^c)\}_{0 \le \alpha < m} \right),$$

where val *denotes the ξ-adic valuation.*

This allows us to to complete the definition of the signature $\sigma(x, y)$: the last three components are n, the index $a \in \{0, \ldots, m-1\}$ that realizes the minimum above (in case of a tie, choose the smallest index), and a string $\mathtt{K} \in \{\text{"H"}, \text{"H}^\mathtt{c}\text{"}\}$ that indicates whether this occurs for H or H^c (in case of a tie, choose H). Associated to string $\mathtt{K}$, we have the corresponding polynomial $K \in \{F, G\}$, obviously defined as $K = H$ if $\mathtt{K} = \text{"H"}$ and $K = H^c$ otherwise.

Finally, the following lemma will help us give conditions on the preservation of the signature through specialization at primes, when for instance $\mathbb{K} = \mathbb{Q}$.

Lemma 17. *For any P in $\mathbb{K}[X, Y]$, the ξ-adic valuation of $P(x + \xi, J^\infty)$ is equal to the multiplicity of the ideal $\langle \frac{\partial^{m-1} H}{\partial Y^{m-1}}, P \rangle$ at (x, y).*

Proof. The proof follows essentially the same derivation as above. We have by definition

$$\mu\left(\left\langle \frac{\partial^{m-1} H}{\partial Y^{m-1}}, P \right\rangle, (x, y) \right) = \dim_{\overline{\mathbb{K}}} \overline{\mathbb{K}}[[\xi, \zeta]] / \left\langle \frac{\partial^{m-1} H}{\partial Y^{m-1}}(x + \xi, y + \zeta), P(x + \xi, y + \zeta) \right\rangle .$$

Now, in $\overline{\mathbb{K}}[[\xi, \zeta]]$, we saw that we also have the equality between ideals

$$\left\langle \frac{\partial^{m-1} H}{\partial Y^{m-1}}(x + \xi, y + \zeta) \right\rangle = \langle \zeta - (J^\infty - y) \rangle ,$$

so that the above multiplicity can be rewritten as

$$\dim_{\overline{\mathbb{K}}} \overline{\mathbb{K}}[[\xi, \zeta]] / \langle \zeta - (J^\infty - y), P(x + \xi, J^\infty) \rangle .$$

Since $P(x + \xi, J^\infty)$ depends only on ξ, this dimension is given by $\mathrm{val}(P(x + \xi, J^\infty))$. □

6 The σ-decomposition

In this section, we consider two polynomials F and G in $\mathbb{K}[X,Y]$, over some field $\mathbb{K}$, with degree at most d. We assume that $\mathbb{K}$ is perfect and has characteristic greater than $\deg(F)$ and $\deg(G)$, and that F and G have no nontrivial common factor in $\mathbb{K}[X,Y]$.

In this case, by Lemma 15, the signature $\sigma(x,y)$ of any element (x,y) of $V = V(F,G)$ is well-defined. Since V is finite, we can use the equivalence relation "having the same signature" to partition it into finitely many equivalence classes. This decomposition will be called the *σ-decomposition* of V.

We give here an algorithm that computes this decomposition. We will work under the extra assumption that F an G are in general position: this assumption implies that there exist polynomials $(P,S) = \mathsf{SL}(F,G)$ in $\mathbb{K}[X]$ such that the defining ideal of V admits the generators $\langle P(X), Y - S(X)\rangle$; in particular, P is squarefree.

To define more precisely our output, remember from the last section that the signature $\sigma(x,y) = (m,\mathtt{H},n,a,\mathtt{K})$ of a point (x,y) is obtained by defining $(m,\mathtt{H})$ first, then $(n,a,\mathtt{K})$. Thus, the partition of V we look for will be obtained by first decomposing it into classes having same values for $(m,\mathtt{H})$, and refining this partition into the σ-decomposition proper. Accordingly, given (F,G) and $(P,S) = \mathsf{SL}(F,G)$ as input, we compute tuples $[\,(C_{i,j},T_{i,j},m_i,\mathtt{H}_i,n_{i,j},a_{i,j},\mathtt{K}_{i,j})\,]_{1\le i\le s, j\in D_i}$, such that for all i,j, $C_{i,j}$ and $T_{i,j}$ are in $\mathbb{K}[X]$, and for each i,j, $(C_{i,j},T_{i,j})$ is the Shape Lemma representation of the subset $V_{m_i,\mathtt{H}_i,n_{i,j},a_{i,j},\mathtt{K}_{i,j}}$ of all elements of V having signature $(m_i,\mathtt{H}_i,n_{i,j},a_{i,j},\mathtt{K}_{i,j})$. By a slight abuse of notation, we still call this sequence the σ-decomposition of V, and we denote it by $\sigma\text{-}\mathsf{dec}(F,G)$ (it is uniquely defined up to order).

The first main result of this section is the following complexity bound on this calculation, when working over a finite field.

Proposition 4. *Suppose that F and G are polynomials in $\mathbb{K}[X,Y]$, of degree at most d, with no nontrivial common factor and in general position over a perfect field $\mathbb{K}$. Suppose further that $\mathbb{K}$ has characteristic greater than d.*

There exists an algorithm $\sigma-\mathsf{decomposition}$ that takes as input F, G and $\mathsf{SL}(F,G)$, and returns the σ-decomposition of $V(F,G)$. When $\mathbb{K} = \mathbb{F}_p$, with p a prime, this algorithm can be implemented so as to take $d^{3+\varepsilon}\tilde{O}(\log(p))$ bit operations, for any $\varepsilon > 0$.

Remark that when $\mathbb{K} = \mathbb{F}_p$, this algorithm is far from being optimal, as both input and output occupy $\Theta(d^2\log(p))$ bits; however, when we apply this algorithm, this will not be the bottleneck of the whole process. Since the notion of signature we are using comes from Lecerf's paper [30], it should also come as no surprise that that reference gives an algorithm for such a calculation; however, the corresponding complexity result (Proposition 19 of [30]) is not suitable for us: that result is aimed toward multivariate systems, given by straight-line programs. Adapted to our particular situation, the cost reported there can be as high as $\tilde{O}(d^8\log(p))$ bit operations (which would then become the bottleneck).

When $\mathbb{K} = \mathbb{Q}$ and F,G have coefficients in $\mathbb{Z}$, we also give conditions under which the computation reduces well at a prime p; for this to make sense, we must first ensure that p is

greater than d, since the base field must have characteristic large enough for the signature of points to be defined.

The data $\sigma\text{-}\mathsf{dec}(F, G)$ consists of a sequence of polynomials, integers and strings; by *reducing* such an object modulo p, we refer to the sequence obtained by reducing the coefficients of all polynomials in $\sigma\text{-}\mathsf{dec}(F, G)$ modulo p, if no denominator vanishes. We denote this new sequence $\sigma\text{-}\mathsf{dec}(F, G) \bmod p$.

Proposition 5. *There exists an efficiently computable function $\Delta_3(d, h, \ell) = (dh\ell)^{O(1)}$ such that the following holds.*

Suppose that F and G are polynomials in $\mathbb{Z}[X, Y]$, with no nontrivial common factor in $\mathbb{Q}[X, Y]$ and in general position, with degree at most d and length at most h. Suppose as well that all polynomials appearing in $\mathsf{SL}(W)$, for any subset W of $V(F, G)$ defined over $\mathbb{Q}$, have length at most ℓ. Then, there exists a nonzero integer δ_3 such that:

- *δ_3 has length at most $\Delta_3(d, h, \ell)$;*
- *for any prime p that satisfies the following conditions:*
  - *p does not divide δ_3,*
  - *for any subset W of $V(F, G)$ defined over $\mathbb{Q}$, p cancels no denominator in $\mathsf{SL}(W)$,*
  - *$(F \bmod p, G \bmod p)$ are in general position and $(P \bmod p, S \bmod p)$ is the Shape Lemma representation of $V(F \bmod p, G \bmod p)$,*

  p is greater than d and the equality $\sigma\text{-}\mathsf{dec}(F, G) \bmod p = \sigma\text{-}\mathsf{dec}(F \bmod p, G \bmod p)$ holds.

The proof of these propositions occupies the whole section, so in all that follows, we assume that the assumptions of Proposition 4 are satisfied. First, we mention the following lemma, which will be used repeatedly.

Lemma 18. *Writing the σ-decomposition of $V(F, G)$ as $[\,(C_{i,j}, T_{i,j}, m_i, \mathtt{H}_i, n_{i,j}, a_{i,j}, \mathtt{K}_{i,j})\,]_{1 \le i \le s, j \in D_i}$, the inequality*

$$\sum_{1 \le i \le s, j \in D_i} n_{i,j} m_{i,j} \deg(C_{i,j}) \le d^2$$

holds.

Proof. The deflation lemma shows that for any root x of $C_{i,j}$, $n_{i,j}m_{i,j}$ is a lower bound on the multiplicity of (F, G) at $(x, T_{i,j}(x))$. The above inequality then follows from Bézout's theorem. □

6.1 Computing all m_i's and $\mathtt{H}_i$'s

In order to motivate the general algorithm, we first briefly explain how to compute the integer m and string $\mathtt{H}$ at a rational point $(x, y) \in \mathbb{K}^2$ of $V(F, G)$, assuming such a point

exists. In this case, the process is straightforward: simply evaluate all required derivatives at (x, y), and stop as soon as we find a nonzero value.

This is detailed in Algorithm compute_m_H_rational below, where we use a function nonzero_index$((x, y), [r_1, \ldots, r_N])$ that returns the smallest index i such that $r_i(x, y)$ does not vanish; the index is calculated following the definition of m and $\mathtt{H}$, by choosing F over G in case of ambiguity (recall that by convention, array indices start at one).

Algorithm 3: compute_m_H_rational(F, G, x, y)

Input: (F, G) in $\mathbb{K}[X, Y]$, a point (x, y) in $V = V(F, G)$
Output: $(m, \mathtt{H})$

$d = \max(\deg(F), \deg(G))$
$R = [\frac{\partial F}{\partial Y}, \frac{\partial G}{\partial Y}, \ldots, \frac{\partial^d F}{\partial Y^d}, \frac{\partial^d G}{\partial Y^d}]$
$n =$ nonzero_index$((x, y), R)$
if n is odd **then**
  return $((n + 1)/2, \text{"}\mathtt{F}\text{"})$
else
  return $(n/2, \text{"}\mathtt{G}\text{"})$
end

Given the Shape Lemma representation (P, S) of V, we follow the same approach. The only significant difference is that zero-tests are replaced by the splitting mechanism of Algorithm nonzero_index of Section 3.

To describe the output, note that we can partition V into subsets $V_{m_1,\mathtt{H}_1}, \ldots, V_{m_s,\mathtt{H}_s}$, for pairwise distinct $(m_i, \mathtt{H}_i)$, where $V_{m_i,\mathtt{H}_i}$ is the subset of V consisting of all (x, y) such that $\sigma(x, y) = (m_i, \mathtt{H}_i, \ldots)$; this partition is coarser than the σ-decomposition, and will be refined later on.

The output of the following algorithm compute_m_H is the sequence $[\,(P_i, S_i, m_i, \mathtt{H}_i)\,]_{1 \le i \le s}$ such that (P_i, S_i) is the Shape Lemma representation of $V_{m_i,\mathtt{H}_i}$. This output, just like the partition $V_{m_1,\mathtt{H}_1}, \ldots, V_{m_s,\mathtt{H}_s}$, is uniquely defined up to order.

Notice for further use that for all i, $\frac{\partial^{m_i-1} H_i}{\partial Y^{m_i-1}}(X, S_i) = 0$ modulo P_i and $\frac{\partial^{m} H_i}{\partial Y^{m_i}}(X, S_i)$ is a unit modulo P_i.

Algorithm 4: compute_m_H(F, G, P, S)

Input: (F, G) in $\mathbb{K}[X, Y]$, the Shape Lemma representation (P, S) of $V = V(F, G)$
Output: a sequence $[\,(P_i, S_i, m_i, \mathtt{H}_i)\,]_{1 \le i \le s}$

1 $d = \max(\deg(F), \deg(G))$
2 $R_0 = [\frac{\partial F}{\partial Y}, \frac{\partial G}{\partial Y}, \ldots, \frac{\partial^d F}{\partial Y^d}, \frac{\partial^d G}{\partial Y^d}]$
3 $R = [r \bmod \langle P, Y - S\rangle \mid r \in R_0]$
4 $K = \mathsf{nonzero\_index}(P, R)$ — K is a sequence of the form $[\,(P_i, n_i)\,]$
5 $S = [\,]$
6 **for** (P_i, n_i) in K **do**
7 **if** n_i is odd **then**
8 append $(P_i, S_i, (n_i + 1)/2, \text{"}\mathtt{F}\text{"})$ to S
9 **else**
10 append $(P_i, S_i, n_i/2, \text{"}\mathtt{G}\text{"})$ to S
11 **end**
12 **end**
13 **return** S

Lemma 19. *Algorithm* compute_m_H *is correct. When* $\mathbb{K} = \mathbb{F}_p$*, one can implement it so as to take* $d^{3+\varepsilon} O^{\tilde{}}(\log(p))$ *bit operations, for any* $\varepsilon > 0$.

Proof. Correctness of the algorithm directly follows from the correctness of nonzero_index, and the fact that all m_i's are at most d, as proved in the deflation lemma.

For the complexity analysis in the particular case $\mathbb{K} = \mathbb{F}_p$, remark first that P has degree $e \le d^2$. We deduce from Lemma 9 that the cost of nonzero_index is $O^{\tilde{}}(d^3)$ operations in $\mathbb{K}$, that is, $O^{\tilde{}}(d^3 \log(p))$ bit operations (arithmetic operations in $\mathbb{F}_p$ can all be done in $O^{\tilde{}}(\log(p))$ bit operations. Thus, all that remains is the cost of computing polynomials R at steps 2 and 3.

This is achieved by combining these two steps into one call to Algorithm normal_forms of Proposition 3, with input $t = 1$, and L, L', L'', F, where L is the list $[(0, d)]$, L' is the list $[P]$ and L'' is the list $[S]$. The output (F_1) of this algorithm is

$$F_1 = F(X + \xi, S + \zeta) \bmod \langle P(X), \xi, \zeta^{d+1}\rangle = F(X, S + \zeta) \bmod \langle P(X), \zeta^{d+1}\rangle.$$

As noted in Section 4, we deduce that we can compute

$$D_\mu = \frac{\partial^\mu F}{\partial Y^\mu}(X, S) \bmod \langle P(X)\rangle$$

as

$$D_\mu = \mu!\, \mathsf{cf}(F_1, \zeta^\mu),$$

so we have obtained half the polynomials we wanted; doing the same with G, we obtain all normal forms we required.

The multiplications by the various constants $0!, 1!, \ldots, d!$ take $O^{\tilde{}}(d^3 \log(p))$ bit operations, so we can focus on the call to Algorithm normal_forms. In order to satisfy Assumption

H_{NF} of Proposition 3, let us write $d' = \lceil d^{3/2} \rceil$. Since, with the notation of that proposition, we have $n_1 = 0$, $m_1 = d$ and $e \leq d^2$, we see that $(n_1+1)(m_1+1)e$ is $O(d^3) = O(d'^2)$. Thus, we are under the assumptions of that proposition, up to replacing d by d'. For any $\varepsilon > 0$, calling Proposition 3 can be done in $d'^{2+\varepsilon} O^{\sim}(\log(p))$ bit operations; this is $d^{3+\varepsilon} O^{\sim}(\log(p))$, as claimed. □

It would be possible to reduce the cost to $d^{2+\varepsilon} O^{\sim}(\log(p))$ bit operations, using Lemma 18 and some amortization techniques; however, the cost $d^{3+\varepsilon} O^{\sim}(\log(p))$ we obtained above is sufficient to make this calculation negligible in the total cost of the main algorithm. Note that we *will* use amortization techniques for some algorithms in the next subsections, for otherwise they would become a bottleneck.

Suppose now that we are over $\mathbb{K} = \mathbb{Q}$, and that F and G are in $\mathbb{Z}[X, Y]$. The following discussion gives conditions under which the above calculation admits a good reduction at a prime p.

Lemma 20. *There exists an efficiently computable function $\Delta_{3,1}(d, h, \ell) = (dh\ell)^{O(1)}$ such that the following holds.*

Suppose that F and G are polynomials in $\mathbb{Z}[X, Y]$, with no nontrivial common factor in $\mathbb{Q}[X, Y]$ and in general position, with degree at most d and length at most h. Let $(P, S) = \mathsf{SL}(F, G)$ and suppose that P and S have length at most ℓ. There exists a nonzero integer $\delta_{3,1}$ such that:

- *$\delta_{3,1}$ has length at most $\Delta_{3,1}(d, h, \ell)$;*
- *for any prime p that satisfies the following conditions:*
  - *p does not divide $\delta_{3,1}$,*
  - *for any subset W of $V(F, G)$ defined over $\mathbb{Q}$, p cancels no denominator in $\mathsf{SL}(W)$,*
  - *$(F \bmod p, G \bmod p)$ are in general position and $(P \bmod p, S \bmod p)$ is the Shape Lemma representation of $V(F \bmod p, G \bmod p)$,*

  p is greater than d and the sequence obtained from $\mathsf{compute\_m\_H}(F, G, P, S) \bmod p$ coincides with the output of $\mathsf{compute\_m\_H}(F \bmod p, G \bmod p, P \bmod p, S \bmod p)$.

Proof. Let $[\,(P_i, S_i, m_i, \mathtt{H}_i)\,]_{1 \leq i \leq s}$ be the output of $\mathsf{compute\_m\_H}(F, G, P, S)$. For a given index i in $\{1, \ldots, s\}$, the corresponding integer m_i is characterized as follows: for each entry A coming before $\frac{\partial^{m_i} H_i}{\partial Y^{m_i}}$ in the sequence $[\frac{\partial F}{\partial Y}, \frac{\partial G}{\partial Y}, \ldots, \frac{\partial^d F}{\partial Y^d}, \frac{\partial^d G}{\partial Y^d}]$, we have $A(X, S_i(X)) = 0 \bmod P_i$; for the entry $\frac{\partial^{m_i} H_i}{\partial Y^{m_i}}$, we have $\gcd(\frac{\partial^{m_i} H_i}{\partial Y^{m_i}}(X, S_i(X)), P_i) = 1$. This latter condition is equivalent to $\frac{\partial^{m_i} H_i}{\partial Y^{m_i}}$ vanishing nowhere on $V(P_i(X), Y - S_i(X))$.

Thus, the polynomials $P_i, S_i, \frac{\partial^{m_i} H_i}{\partial Y^{m_i}}$ satisfy conditions C_1 and C_2 of Proposition 1. We claim that we can take for $\delta_{3,1}$ the product of the integers δ_1 associated by that proposition to the systems $P_i, S_i, \frac{\partial^{m_i} H_i}{\partial Y^{m_i}}$, for $i = 1, \ldots, s$, multiplied by $d!$.

Let indeed p be a prime that satisfies the assumptions listed in the statement of the present lemma; in particular, it does not divide $\delta_{3,1}$.

Then, by these assumptions, all P_i's and S_i's can be reduced modulo p; besides, because the polynomial P in $\mathsf{SL}(F,G)$ remains squarefree modulo p, this is also the case for all P_i's, and these polynomials remain pairwise coprime modulo p. Thus, the polynomials $[\,(P_i \bmod p, S_i \bmod p)\,]_{1\le i\le s}$ form the Shape Lemma representations of *some* partition of $V(F \bmod p, G \bmod p)$. It remains to see whether this is the same partition as the one obtained by running the algorithm over $\mathbb{F}_p$, with input $(F,G,P,S) \bmod p$.

Let us first point out that because p does not divide $d!$, p is greater than d. Thus, by the deflation lemma, every point (x,y) in $V(F \bmod p, G \bmod p)$ admits a well-defined signature $(m, \mathtt{H}, n, a, \mathtt{K})$; m is characterized as the smallest integer such that either $\frac{\partial^m F}{\partial Y^m}$ or $\frac{\partial^m G}{\partial Y^m}$ does not vanish at (x,y).

On input $(F,G,P,S) \bmod p$, Algorithm $\mathsf{compute\_m\_H}$ returns the partition $V(F \bmod p, G \bmod p)$ according to these values of m and $\mathtt{H}$. Now, the assumption that p does not divide $\delta_{3,1}$ shows that for any (x,y) root of $(P_i \bmod p, Y - S_i \bmod p)$, the values of m and $\mathtt{H}$ are precisely m_i and $\mathtt{H}_i$, so by uniqueness of the Shape Lemma representation, the tuples $[\,(P_i \bmod p, S_i \bmod p, m_i, \mathtt{H}_i)\,]_{1\le i\le s}$ are indeed the output obtained by running the algorithm over $\mathbb{F}_p$, with input $(F,G,P,S) \bmod p$.

Thus, our claims are proved, except for the upper bound $\Delta_{3,1}$ on the length of $\delta_{3,1}$. There are at most d^2 families $P_i, S_i, \frac{\partial^{m_i} H_i}{\partial Y^{m_i}}$ to take into account. Each of the polynomials $\frac{\partial^{m_i} H_i}{\partial Y^{m_i}}$ has degree at most d, and length $h' \le h + \mathrm{len}(d!)$, which is $O\tilde{}(h+d)$, where the term $\mathrm{len}(d!)$ accounts for the length growth through differentiation. We can then take $\Delta_{3,1} = d^2\Delta_1(d,h',d^2,\ell) + \mathrm{len}(d!)$, where Δ_1 is the function defined in Proposition 1 and the term $\mathrm{len}(d!)$ accounts for the multiplication by $d!$. □

6.2 Computing all J_i's

Suppose that we have determined the sequence $[\,(P_i, S_i, m_i, \mathtt{H}_i)\,]_{1\le i\le s}$ of the previous subsection; we now want to compute power series J_i's as defined in Section 5. Compared to the presentation section, there is a slight difference: we are not working at a point (x,y) with coordinates in $\overline{\mathbb{K}}^2$, but with points given through Shape Lemma representations.

Let us thus fix an index i in $\{1,\dots,s\}$. Associated to P_i, one can define the ring $\mathbb{B}_i = \mathbb{K}[X]/\langle P_i\rangle$; this is in general not a field, but only a product of fields. Two elements will be highlighted in $\mathbb{B}_i$: the residue class x_i of X, and the residue class y_i of $S_i(X)$. Thus, by construction, $F(x_i,y_i) = G(x_i,y_i) = 0$ (where F and G are viewed as polynomials in $\mathbb{B}_i$, through the canonical injection $\mathbb{K} \to \mathbb{B}_i$). We noted in the previous subsection that the polynomial H_i associated to P_i and S_i is such that

$$\frac{\partial^{m_i-1} H_i}{\partial Y^{m_i-1}}(x_i,y_i) = 0 \text{ in } \mathbb{B}_i \quad \text{and} \quad \frac{\partial^{m_i} H_i}{\partial Y^{m_i}}(x_i,y_i) \text{ is a unit in } \mathbb{B}_i. \tag{6}$$

This is sufficient for us to apply Newton iteration, and compute a power series J_i^∞ in $\mathbb{B}_i[[\xi]]$ such that

$$\frac{\partial^{m_i-1} H_i}{\partial Y^{m_i-1}}(x_i+\xi, J_i^\infty) = 0 \quad \text{and} \quad J_i^\infty(0) = y_i. \tag{7}$$

The following algorithm describes this process. Remark that one can relate this construction to the one in Section 5: if $x \in \overline{\mathbb{K}}$ is a root of P_i and $y = T_i(x)$, so that (x, y) is in $V(F, G)$, the power series $J^\infty \in \overline{\mathbb{K}}[[\xi]]$ associated to (x, y) in that section is obtained by letting $x_i = x$ in J_i^∞ (which is valid because x is a root of P_i).

The algorithm will actually be used in a slightly more general context than the one just described. First, instead of taking as input only the polynomials P_i and S_i computed in the previous subsection, we will as well call this algorithm using input polynomials (C_i, T_i), where C_i may be only a factor of P_i, and T_i will replace S_i (if C_i is a factor of P_i, we may for example take $T_i = S_i \bmod C_i$). In any case, we will solve for J_i^∞ that satisfies Eq. (7) above, with y_i now being the residue class of T_i modulo C_i. The second minor extension is that we will not necessarily work over a field $\mathbb{K}$, but possibly only a ring denoted $\mathbb{A}$ (which will be of the form $\mathbb{Z}/N\mathbb{Z}$ below); this is a harmless assumption, since the algorithm only requires that (6) above holds. As a matter of notation, $\mathbb{B}_i$ will then denote $\mathbb{B}_i = \mathbb{A}[X]/\langle C_i \rangle$.

As input, the algorithm takes positive integers n_i as extra parameters, which give the required precision in ξ for the power series J_i^∞. Since our output is truncated modulo ξ^{n_i}, we denote it by J_i, and keep the notation J_i^∞ for the infinite-precision solution of equations such as (7).

The computations themselves are a simple form of Newton iteration; the only important point is to control of the cost of the evaluations of the functions $\frac{\partial^{m_i-1} H_i}{\partial Y^{m_i-1}}$ and their derivatives.

Algorithm 5: compute_J$(F, G, [\,(C_i, T_i, m_i, \mathtt{H}_i, n_i)\,]_{1 \le i \le s})$

Input: F, G, a sequence of polynomials C_i and T_i in $\mathbb{A}[X]$, strings $\mathtt{H}_i$ and indices m_i and n_i

Output: a sequence $[J_i]_{1 \le i \le s}$ with $J_i \in \mathbb{B}_i[[\xi]]$ known mod ξ^{n_i} that satisfies (7) mod ξ^{n_i}

$\lambda = 1$

$[\,J_i\,]_{1 \le i \le s} = [\,T_i\,]_{1 \le i \le s}$

$I = [\,i \mid 1 \le i \le s \text{ and } 1 < n_i\,]$ indices for which we need to lift further

$I_F = [\,i \mid 1 \le i \le s \text{ and } \mathtt{H}_i = \text{"F"}\,]$ indices for which $H = F$

$I_G = [\,i \mid 1 \le i \le s \text{ and } \mathtt{H}_i = \text{"G"}\,]$ indices for which $H = G$

while I is not empty **do**

  $[\,\eta_i\,]_{i \in I} = [\,\frac{\partial^{m_i-1} F}{\partial Y^{m_i-1}}(X + \xi, J_i) \bmod \langle C_i(X), \xi^{2\lambda} \rangle\,]_{i \in I \cap I_F}$ cat
  $[\,\frac{\partial^{m_i-1} G}{\partial Y^{m_i-1}}(X + \xi, J_i) \bmod \langle C_i(X), \xi^{2\lambda} \rangle\,]_{i \in I \cap I_G}$

  $[\,\eta'_i\,]_{i \in I} = [\,\frac{\partial^{m_i} F}{\partial Y^{m_i}}(X + \xi, J_i) \bmod \langle C_i(X), \xi^{2\lambda} \rangle\,]_{i \in I \cap I_F}$ cat
  $[\,\frac{\partial^{m_i} G}{\partial Y^{m_i}}(X + \xi, J_i) \bmod \langle C_i(X), \xi^{2\lambda} \rangle\,]_{i \in I \cap I_G}$

  for i in I **do**

    $J_i = J_i - \eta_i / \eta'_i \bmod \langle C_i(X), \xi^{2\lambda} \rangle$

  end

  $\lambda = 2\lambda$

  $I = [\,i \mid 1 \le i \le s \text{ and } \lambda < n_i\,]$ indices for which we need to lift further

end

return $[\,J_i \bmod \xi^{n_i}\,]_{1 \le i \le s}$ we may know J_i at a slightly higher precision than n_i

---

Lemma 21. *Algorithm* compute_J *is correct. When* $\mathbb{A} = \mathbb{Z}/N\mathbb{Z}$*, when* N *is a power of a prime* p*, for any* $\varepsilon > 0$*, one can implement this algorithm so that it takes* $d^{2+\varepsilon}O^{\tilde{}}(\log(N))$ *bit operations, provided* $\sum_{1\le i\le s} n_i(m_i+1)\deg(C_i) = O(d^2)$ *holds.*

Proof. The algorithm essentially implements Newton iteration, over all $\mathbb{B}_i[[\xi]]$ independently. By assumption, for all i, $\frac{\partial^{m_i-1}H_i}{\partial Y^{m_i-1}}(x_i, y_i) = 0$ in $\mathbb{B}_i$ and $\frac{\partial^{m_i}H_i}{\partial Y^{m_i}}(x_i, y_i)$ is a unit in $\mathbb{B}_i$, so even when we work over a ring $\mathbb{A}$ rather than a field, we can indeed run Newton iteration. The sequence I indicates the indices for which we have not reached the required precision yet; these are the indices for which we do further iteration steps. Sequences I_F and I_G indicate which indices use F or G to do the lifting.

It remains to do the cost analysis, in the case where $\mathbb{A} = \mathbb{Z}/N\mathbb{Z}$; all the cost is spent in the main loop (at the beginning, the T_i's are already reduced modulo the respective C_i's; at the end, truncation is free). First, remark that the highest value λ will reach will be $O(\max_i n_i)$, which is $O(d^2)$ by assumption. As a consequence, the number of times we will enter the loop is $O(\log(d))$, which we will be able to absorb in the term $d^{2+\varepsilon}$. Thus, we can focus on the cost of a single pass through the loop.

The inversion and multiplication at Step 10 take $O^{\tilde{}}(\sum_{i\in I}\deg(C_i)\lambda)$ operations in $\mathbb{Z}/N\mathbb{Z}$, or $O^{\tilde{}}(\sum_{i\in I}\deg(C_i)\lambda\log(N))$ bit operations; this will be absorbed in the cost of Steps 7 and 8, which are more delicate to analyze.

Since we are over $\mathbb{Z}/N\mathbb{Z}$, we use algorithm normal_forms of Proposition 3 to compute the values η_i and η_i', simultaneously for all indices i in I. We call this algorithm twice: once using F for the indices i in I_F, then using G for those indices i in I_G; it is enough to analyze the cost for, say, F.

We call algorithm normal_forms with an input size t_F (the cardinality of $I \cap I_F$), and lists L, L', L'' and polynomial F as follows: we take $L = [\,(2\lambda-1, m_i)\,]_{i\in I\cap I_F}$, $L' = [\,C_i\,]_{i\in I\cap I_F}$ and $L'' = [\,J_i\,]_{i\in I\cap I_F}$. The key remark is that when we are at precision λ, all indices i remaining in $I \cap I_F$ satisfy $\lambda < n_i$; thus, the input size satisfies

$$\sum_{i\in I\cap I_F} 2\lambda(m_i+1)\deg(C_i) \le \sum_{i\in I\cap I_F} 2n_i(m_i+1)\deg(C_i),$$

which is $O(d^2)$ by assumption. Thus, assumption H_{NF} of Proposition 3 is satisfied, so for any $\varepsilon > 0$, we can compute

$$D_{i,\mu} = \frac{\partial^\mu F}{\partial Y^\mu}(X+\xi, J_i) \bmod \langle C_i(X), \xi^{2\lambda}\rangle,$$

for all i in $I \cap I_F$ and $\mu = 0, \ldots, m_i$, using $d^{2+\varepsilon}O^{\tilde{}}(\log(N))$ bit operations. Keeping those derivatives of order m_i and m_{i-1} gives us the requires values η_i and η_i'. □

6.3 Computing all n_i's, a_i's and $\mathtt{K}_i$'s

Finally, we want to compute the values of n, a and $\mathtt{K}$ at all points in V. As input, we start from the sequence $[\,(P_i, S_i, m_i, \mathtt{H}_i)\,]_{1\le i\le s}$ computed in Section 6.1; recall that this sequence defines the partition of V into sets $(V_{m_i,\mathtt{H}_i})_{1\le i\le s}$.

The σ-decomposition of V that we wish to compute is a refinement of the partition $(V_{m_i,\mathtt{H}_i})_{1\le i\le s}$; in other words, we obtain it by partitioning further each $V_{m_i,\mathtt{H}_i}$ into subsets $(V_{\sigma_{i,j}})_{j\in D_i}$, for some index set D_i; each $\sigma_{i,j}$ takes the form $\sigma_{i,j} = (m_i, \mathtt{H}_i, n_{i,j}, a_{i,j}, \mathtt{K}_{i,j})$. Our output will consist in a similarly indexed array of the form $[\,(C_{i,j}, T_{i,j}, m_i, \mathtt{H}_i, n_{i,j}, a_{i,j}, \mathtt{K}_{i,j})\,]_{1\le i\le s, j\in D_i}$, such that for all i, j, $(C_{i,j}, T_{i,j})$ is the Shape Lemma representation of $V_{\sigma_{i,j}}$.

To describe the idea the algorithm, let us fix the index i in $\{1, \ldots, s\}$. Then, we need to compute the power series J_i defined in the previous section at some suitable precision λ, and deduce the expansions of $\frac{\partial^\mu F}{\partial Y^\mu}(X+\xi, J_i) \bmod \xi^\lambda$ and $\frac{\partial^\mu G}{\partial Y^\mu}(X+\xi, J_i) \bmod \xi^\lambda$ for suitable values of μ; this will be done at successive precisions $\lambda = 1, 2, 4, \ldots$ in ξ.

Suppose we have obtained these expansions modulo ξ^λ. If we were over a field, using Lemma 16, we would then look for the expansion with smallest valuation in ξ; however, we are working over $\mathbb{B}_i = \mathbb{K}[X]/\langle P_i\rangle$, which is not necessarily a field. Thus, we apply Algorithm nonzero_index_vectorial of Section 3; it returns factors of P_i for which we have found the correct valuation, together with possibly a residual factor, for which we have to increase the precision λ in ξ. Thus, we replace P_i by this factor, multiply λ by 2, and start over. In order to distinguish between the input polynomials (P_i, S_i) and their factors, we use new variables called (C_i, T_i) as our current polynomials.

The following algorithm called compute_n_a_K implements this idea; the fact that we have to handle as well the strings $\mathtt{H}_i$ to decide with partial derivatives to consider makes for some admittedly clumsy bookkeeping (which we explain in the proof of the following lemma). In the pseudo-code, we use the subroutine $\mathsf{cf}(P, \xi^j)$ which returns the coefficient of ξ^j in polynomial P; further subroutines infinity, index_of and polynomial, that are only designed for said bookkeeping purposes, are explained in the proof of the lemma.

Lemma 22. *Algorithm* compute_n_a_K *terminates and is correct. When* $\mathbb{K} = \mathbb{F}_p$*, for any* $\varepsilon > 0$*, one can implement it so that it takes* $d^{2+\varepsilon}O^{\sim}(\log(p))$ *bit operations.*

Proof. To establish correctness, we first prove that the following invariant is preserved throughout the **while** loop: at the beginning of the loop,

- for all indices i in I, (C_i, T_i) is the Shape Lemma representation of the union of all subsets $V_{\sigma_{i,j}}$, for $\sigma_{i,j}$ of the form $\sigma_{i,j} = (m_i, \mathtt{H}_i, n, a, \mathtt{K})$, for some indices $n, a, \mathtt{K}$ such that $n \ge \lambda$;
- L contains the entries $[\,(C_{i,j}, T_{i,j}, m_i, \mathtt{H}_i, n_{i,j}, a_{i,j}, \mathtt{K}_{i,j})\,]_{i,j}$, for all indices i in $\{1, \ldots, s\}$ and j in D_i such that $n_{i,j} < \lambda$.

Initially, $\lambda = 1$, $I = [1, \ldots, s]$ and L is empty; since all $n_{i,j}$ are at least equal to one, our loop invariant holds. Supposing that we maintained the invariant up to some exponent λ, we prove that they will be maintained through the next pass in the loop.

Step 8 computes the sequence $[\,J_i\,]_{i\in I}$; those are power series known modulo $\langle C_i, \xi^{2\lambda}\rangle$, such that, for all i,

$$\frac{\partial^{m_i-1} H_i}{\partial Y^{m_i-1}}(X+\xi, J_i) = 0 \bmod \xi^{2\lambda} \quad \text{and} \quad J_i(0) = T_i$$

Algorithm 6: compute_n_a_K $(F, G, [\,(P_i, S_i, \mathtt{H}_i, m_i)\,]_{1\le i\le s})$

Input: the sequence $[\,(P_i, S_i, m_i, \mathtt{H}_i)\,]_{1\le i\le s}$ computed in Section 6.1

Output: a sequence $[\,(C_{i,j}, T_{i,j}, m_i, \mathtt{H}_i, n_{i,j}, a_{i,j}, \mathtt{K}_{i,j})\,]_{1\le i\le s, j\in D_i}$

$\lambda = 1$
$L = [\,]$
$I = [\,1, \ldots, s\,]$
$I_F = [\,i \mid 1 \le i \le s \text{ and } \mathtt{H}_i = \text{"}\mathtt{F}\text{"}\,]$
$I_G = [\,i \mid 1 \le i \le s \text{ and } \mathtt{H}_i = \text{"}\mathtt{G}\text{"}\,]$
$[\,C_i, T_i\,]_{i\in I} = [\,P_i, S_i]_{i\in I}$
while I is not empty **do**
  $[\,J_i\,]_{i\in I} = \mathsf{compute\_J}(F, G, [\,C_i, T_i, m_i, \mathtt{H}_i, 2\lambda\,]_{i\in I})$
  $[\,\eta_{i,\alpha}\,]_{i\in I, \alpha\in[0,\ldots,m_i-1]} = [\,\frac{\partial^\alpha F}{\partial Y^\alpha}(X+\xi, J_i) \bmod \langle C_i(X), \xi^{2\lambda}\rangle\,]_{i\in I, \alpha\in[0,\ldots,m_i-1]}$
  $[\,\gamma_{i,\alpha}\,]_{i\in I, \alpha\in[0,\ldots,m_i-1]} = [\,\frac{\partial^\alpha G}{\partial Y^\alpha}(X+\xi, J_i) \bmod \langle C_i(X), \xi^{2\lambda}\rangle\,]_{i\in I, \alpha\in[0,\ldots,m_i-1]}$
  for i in I **do**
    if i is in I_F **then**
      $R_i =$
      $[\,[\,\mathsf{cf}(\eta_{i,0}, \xi^j), \mathsf{cf}(\gamma_{i,0}, \xi^j), \ldots, \mathsf{cf}(\eta_{i,m_i-2}, \xi^j), \mathsf{cf}(\gamma_{i,m_i-2}, \xi^j), \mathsf{cf}(\gamma_{i,m_i-1}, \xi^j)\,]\,]_{j=0,\ldots,2\lambda-1}$
    else
      $R_i =$
      $[\,[\,\mathsf{cf}(\gamma_{i,0}, \xi^j), \mathsf{cf}(\eta_{i,0}, \xi^j), \ldots, \mathsf{cf}(\gamma_{i,m_i-2}, \xi^j), \mathsf{cf}(\eta_{i,m_i-2}, \xi^j), \mathsf{cf}(\eta_{i,m_i-1}, \xi^j)\,]\,]_{j=0,\ldots,2\lambda-1}$
    $L_i = \mathsf{nonzero\_index\_vectorial}(C_i, R_i)$ L_i has the form $[\,(C_{i,j}, (n_{i,j}, \ell_{i,j}))\,]_{j\in D_i}$
    $C_i = \mathsf{infinity}(L_i)$
    if C_i is not constant **then**
      remove C_i from L_i update C_i and T_i
      $T_i = T_i \bmod C_i$
    else
      remove i from I we are done with this index
    $T_{i,j} = [\,T_i \bmod A_{i,j}\,]_{A_{i,j}\in L_i}$
    $L = L$ cat $[\,(C_{i,j}, T_{i,j}, m_i, \mathtt{H}_i, n_{i,j}, \mathsf{index\_of}(n_{i,j}, R_i), \mathsf{polynomial}(n_{i,j}, R_i))\,]_{j\in D_i}$
  end
  $\lambda = 2\lambda$
end
return L

hold modulo C_i. In what follows, it will be useful to write J_i^∞ for the power series in $\mathbb{D}_i[[\xi]]$ that satisfies $\frac{\partial^{m_i-1}H_i}{\partial Y^{m_i-1}}(X+\xi, J_i) = 0$ and $J_i(0) = T_i$, with $\mathbb{D}_i = \mathbb{K}[X]/\langle C_i\rangle$; in particular, $J_i = J_i^\infty \bmod \xi^{2\lambda}$ for all i.

Consider an index i in I; without loss of generality, we assume that I is in I_F, that is, that $\mathtt{H}_i$ is the string "`F`"; the case where $\mathtt{H}_i$ is equal to "`G`" is handled similarly, up to the difference in indices.

In this case, Lemma 16 shows that for any (x, y) in $V(C_i, Y - T_i) \subset \overline{\mathbb{F}_p}^2$, the integer n appearing in its signature satisfies

$$n = \min\left(\left\{\mathrm{val}\left(\frac{\partial^\alpha F}{\partial Y^\alpha}(x+\xi, J_i^\infty(x))\right)\right\}_{0\le\alpha<m_i-1} \cup \left\{\mathrm{val}\left(\frac{\partial^\alpha G}{\partial Y^\alpha}(x+\xi, J_i^\infty(x))\right)\right\}_{0\le\alpha<m_i}\right),$$

where val denotes the ξ-adic valuation and $J_i^\infty(x)$ denotes the power series in $\overline{\mathbb{K}}[[\xi]]$ obtained by evaluating X at x in J_i^∞ (this is valid, since J_i^∞ has coefficients in $\mathbb{D}_i$, and x is a root of C_i). Define similarly $\eta_{i,\alpha}(x)$ and $\gamma_{i,\alpha}(x)$ as the polynomials in $\overline{\mathbb{K}}[\xi]_{2\lambda}$ obtained by evaluating respectively $\eta_{i,\alpha}$ and $\gamma_{i,\alpha}$ at $X = x$. Then, in view of the calculations at Steps 9 and 10, we see that either $n \ge 2\lambda$, in which case all of

$$(\eta_{i,\alpha}(x))_{0\le\alpha<m_i-1} \quad \text{and} \quad (\gamma_{i,\alpha}(x))_{0\le\alpha<m_i}$$

vanish, or n can be rewritten as

$$n = \min\left(\{\mathrm{val}(\eta_{i,\alpha}(x))\}_{0\le\alpha<m_i-1} \cup \{\mathrm{val}(\gamma_{i,\alpha}(x))\}_{0\le\alpha<m_i}\right).$$

The sequence R_i then precisely contains the coefficients of power series $\eta_{i,\alpha}$ and $\gamma_{i,\alpha}$ that we have to test for zero at the roots x of C_i (the way that R_i is sorted follows from our definition of the index a and the polynomial `K` in the signature of a point).

The call to nonzero_index_vectorial(C_i, R_i) returns a sequence $L_i = [\,(C_{i,j}, (n_{i,j}, \ell_{i,j}))\,]_{j\in D_i}$, such that $C_i = \prod_{j\in D_i} C_{i,j}$, with either $(n_{i,j}, \ell_{i,j})$ in $\{0,\dots,2\lambda-1\}\times\{1,\dots,2m_i-1\}$ (the former length is the length of R_i, the latter the length of the entries in R_i) or $(n_{i,j}, \ell_{i,j}) = (\infty, \infty)$; the roots of $C_{i,j}$ are the roots x of C_i having $n = n_{i,j}$, when $n_{i,j} < \infty$, or for which all we can say is that $n \ge 2\lambda$, when $n_{i,j} = \infty$.

If $n_{i,j} = \infty$ does not show up, we have found the valuation for all roots of C_i; otherwise, the roots corresponding to $n_{i,j} = \infty$ will have to enter the next pass in the **while** loop. This is decided at Step 17, where subroutine infinity extracts the entry $(C_{i,j}, (n_{i,j}, \ell_{i,j}))$ in L_i having $n_{i,j} = \infty$, if such an entry exists, and replaces C_i by this polynomial $C_{i,j}$. If no such entry exists, infinity returns and assign 1 to C_i.

If the new value of C_i has positive degree, we update T_i as well, so that (C_i, T_i) is the Shape Lemma representation of all roots of P_i having $n \ge 2\lambda$ — this proves that the first half of our loop invariant will be satisfied for the next iteration. If the new value of C_i is equal to 1, we are done with all roots of P_i, so we can remove i from index set I.

It remains to update the sequence L with those entries of L_i corresponding to $n_{i,j} < \infty$. For all these entries, the index $\ell_{i,j} \in \{1,\dots,2m_i-1\}$ tells us which polynomial in R_i yielded

a nonzero value. The subroutines index_of and polynomial simply deduce the corresponding index $a_{i,j}$ (in either $\{0,\dots,m_i-2\}$ or $\{0,\dots,m_i-1\}$), and the corresponding string $\mathtt{K}_{i,j}$ indicates whether the non-vanishing occurred for one of the $\eta_{i,\alpha}$'s or $\gamma_{i,\alpha}$'s. This construction shows that the second half of our loop invariant will be satisfied for the next iteration, so we are done with our induction proof of correctness.

Next, remark that the algorithm terminates: indeed, all $n_{i,j}$ satisfy the crude upper bound $n_{i,j} \leq d^2$. At the end of the algorithm, I is empty and our loop invariant proves that the output is correct.

It remains to do the cost analysis, when $\mathbb{K} = \mathbb{F}_p$. Because all $n_{i,j}$ are at most d^2, the number of passes through the **while** loop is $O(\log(d))$, which we will be able to absorb in the term d^ε. We can thus focus on a given pass through the loop, for some precision λ. The key inequality to notice is that

$$\sum_{i\in I} 2\lambda m_i \deg(C_i) = O(d^2). \tag{8}$$

Indeed, for all indices i remaining in I at this stage, the index n of any root x of C_i is at least λ (as per our loop invariant), so our claim follows from Lemma 18.

As a consequence, we can apply Lemma 21, which proves that the cost of computing all J_i is $d^{2+\varepsilon}O^{\sim}(\log(p))$ bit operations. Similarly, the inequality above shows that the computation of all $[\,\eta_{i,\alpha}\,]_{i\in I,\alpha\in[0,\dots,m_i-1]}$ can handled by Algorithm normal_forms, with input lists L, L', L'' given by $L = [\,(2\lambda-1, m_i-1)\,]_{i\in I}$, $L' = [\,C_i\,]_{i\in I}$ and $L'' = [\,J_i\,]_{i\in I}$; in view of Proposition 3, the cost is $d^{2+\varepsilon}O^{\sim}(\log(p))$ bit operations as well.

The sequence R_i has length $O(\lambda)$, with entries of length $O(m_i)$, containing polynomials of degree less than $\deg(C_i)$. Then, Lemma 10 implies that the cost of the call to nonzero_index_vectorial(C_i, R_i) is $O^{\sim}(\lambda m_i \deg(C_i))$ operations in $\mathbb{F}_p$. Using (8) again, we deduce that the total cost over all i's is $O^{\sim}(d^2)$ operations in $\mathbb{F}_p$, that is, $O^{\sim}(d^2\log(p))$ bit operations.

All other arithmetic operations (remainder at Step 20 and multiple remainders at Step 23) take time $O^{\sim}(\deg(C_i))$ operations in $\mathbb{F}_p$, for a total of $O^{\sim}(d^2\log(p))$ bit operations. Summing all costs seen so far, we conclude the proof of the lemma. □

The main algorithm of this section, Algorithm σ-dec, is simply the combination of compute_m_H and compute_n_a_K. Combining the results of Lemmas 19, 21 and 22 proves the complexity statement in Proposition 4. It remains to prove Proposition 5, about primes of good reduction.

Let us thus assume that $\mathbb{K} = \mathbb{Q}$ and let p be a prime that does not divide the integer $\delta_{3,1}$ of Lemma 20, and that satisfies the following assumptions, that are taken from either Lemma 20 or Proposition 5:

- for any subset W of $V(F,G)$ defined over $\mathbb{Q}$, p cancels no denominator in $\mathsf{SL}(W)$,
- $(F \bmod p, G \bmod p)$ are in general position and $(P \bmod p, S \bmod p)$ is the Shape Lemma representation of $V(F \bmod p, G \bmod p)$.

Further assumptions will be put on p, but we can already deduce the following facts: by the first point above p divides no denominator in the coefficients of any polynomial $C_{i,j}$ or $T_{i,j}$; since P is the product of all $C_{i,j}$'s and it remains squarefree modulo p, the second item shows that the pairs $(C_{i,j} \bmod p, T_{i,j} \bmod p)_{1\le i\le s, j\in D_i}$ are the Shape Lemma representations of a partition of $V(F \bmod p, G \bmod p)$. It remains to see whether this is the σ-decomposition of this set (which is well-defined, since p not dividing $\delta_{3,1}$ implies that $p > d$).

Consider a component $(C_{i,j}, T_{i,j}, m_i, \mathtt{H}_i, n_{i,j}, a_{i,j}, \mathtt{K}_{i,j})$ of the σ-decomposition of $V(F, G)$, with i and j fixed throughout. By Lemma 20, another consequence of the fact that p does not divide $\delta_{3,1}$ is that the sequence obtained from $\mathsf{compute\_m\_H}(F, G, P, S)$ mod p coincides with the output of $\mathsf{compute\_m\_H}(F \bmod p, G \bmod p, P \bmod p, S \bmod p)$.

This implies that for any (x, y) in $V(C_{i,j} \bmod p, Y - T_{i,j} \bmod p) \subset \overline{\mathbb{F}_p}^2$, the signature of (x, y) has the form $(m_i, \mathtt{H}_i, \dots)$. To conclude, it is enough to prove that this signature is indeed $(m_i, \mathtt{H}_i, n_{i,j}, a_{i,j}, \mathtt{K}_{i,j})$.

Let $\mathbb{D}_{i,j}$ denote the ring $\mathbb{Q}[X]/\langle C_{i,j}\rangle$ and let $J^\infty_{i,j}$ be the power series in $\mathbb{D}_{i,j}[[\xi]]$ that satisfies

$$\frac{\partial^{m_i-1} H_i}{\partial Y^{m_i-1}}(x_{i,j} + \xi, J^\infty_{i,j}) = 0 \quad \text{and} \quad J^\infty_{i,j}(0) = y_{i,j},$$

where $x_{i,j}$ and $y_{i,j}$ is the respective images of X and $T_{i,j}$ in $\mathbb{D}_{i,j}$. Then, we know from Lemma 16 that $n_{i,j}$, $a_{i,j}$ and $\mathtt{K}_{i,j}$ are determined as follows: consider the power series in $\mathbb{D}_{i,j}[[\xi]]$ given by

$$\begin{aligned} &H_i(x_{i,j} + \xi, J^\infty_{i,j}), H^c_i(x_{i,j} + \xi, J^\infty_{i,j}), \dots, \\ &\frac{\partial^{m_i-2} H_i}{\partial Y^{m_i-1}}(x_{i,j} + \xi, J^\infty_{i,j}), \frac{\partial^{m_i-2} H^c_i}{\partial Y^{m_i-2}}(x_{i,j} + \xi, J^\infty_{i,j}), \frac{\partial^{m_i-1} H^c_i}{\partial Y^{m_i-1}}(x_{i,j} + \xi, J^\infty_{i,j}). \end{aligned}$$

Then, $n_{i,j}$ is such that for all $n < n_{i,j}$, the coefficient of ξ^n of all these power series vanishes. For $n = n_{i,j}$, the coefficient of $\xi^{n_{i,j}}$ of all power series obtained from $H_i, H^c_i, \frac{\partial H_i}{\partial Y}, \frac{\partial H^c_i}{\partial Y}, \dots$ vanishes, until we reach $\frac{\partial^{a_{i,j}} K_{i,j}}{\partial Y^{a_{i,j}}}(x_{i,j} + \xi, J^\infty_{i,j})$, for which that coefficient is a unit.

To prove our claim, it suffices to give conditions on p under which this remains the case modulo p. The calculation of $J^\infty_{i,j}$ commutes with reduction modulo p, so the vanishing conditions will continue to hold modulo p, and we are left to ensure that the coefficient of $\xi^{n_{i,j}}$ in $\frac{\partial^{a_{i,j}} K_{i,j}}{\partial Y^{a_{i,j}}}(x_{i,j} + \xi, J^\infty_{i,j})$ remains a unit modulo p.

That coefficient being a unit is equivalent to $\frac{\partial^{a_{i,j}} K_{i,j}}{\partial Y^{a_{i,j}}}(x + \xi, J^\infty_{i,j}(x))$ having valuation $n_{i,j}$ for all (x, y) in $V(C_{i,j}(X), Y - T_{i,j}(X))$, where $J^\infty_{i,j}(x)$ is the power series in $\overline{\mathbb{K}}[[\xi]]$ obtained by evaluating $x_{i,j}$ at x in $J^\infty_{i,j}$. By Lemma 17, this is equivalent to all points in $V(C_{i,j}(X), Y - T_{i,j}(X))$ being roots of multiplicity $n_{i,j}$ of the system $(\frac{\partial^{m_i-1} H_i}{\partial Y^{m_i-1}}, \frac{\partial^{a_{i,j}} K_{i,j}}{\partial Y^{a_{i,j}}})$.

The polynomials $C_{i,j}, T_{i,j}, \frac{\partial^{m_i-1} H_i}{\partial Y^{m_i-1}}, \frac{\partial^{a_{i,j}} K_{i,j}}{\partial Y^{a_{i,j}}}$ thus satisfy the assumptions of Proposition 2. We deduce that if p does not divide the integer, say $\delta_{2,i,j}$, associated to these polynomials by that proposition, the multiplicity of $(\frac{\partial^{m_i-1} H_i}{\partial Y^{m_i-1}}, \frac{\partial^{a_{i,j}} K_{i,j}}{\partial Y^{a_{i,j}}})$ at any root of $C_{i,j}(X), Y - T_{i,j}(X)$ modulo p remains $n_{i,j}$. As a result, applying again Lemma 17 shows that the corresponding coefficient of $\frac{\partial^{a_{i,j}} K_{i,j}}{\partial Y^{a_{i,j}}}(x_{i,j} + \xi, J^\infty_{i,j})$ remains invertible in $\mathbb{F}_p[X]/\langle C_{i,j} \bmod p\rangle$, and we are done.

Taking all $C_{i,j}$ into account, we see that to ensure success it is sufficient to impose the new condition that p divides none of the integers $\delta_{2,i,j}$, or equivalently that it does not divide their product δ_2. Thus, we naturally define $\delta_3 = \delta_2\delta_{3,1}$.

To quantify this construction, recall that we suppose that F and G have degree at most d and length at most h. There are at most d^2 systems of the form $C_{i,j}, T_{i,j}, \frac{\partial^{m_i-1}H_i}{\partial Y^{m_i-1}}, \frac{\partial^{a_{i,j}}K_{i,j}}{\partial Y^{a_{i,j}}}$ to take into account. In any of these systems, the degree and length of polynomials $\frac{\partial^{m_i}H_i}{\partial Y^{m_i}}$ and $\frac{\partial^{a_{i,j}}K_{i,j}}{\partial Y^{a_{i,j}}}$ are respectively at most d and $h + \mathrm{len}(d!)$, which is $O\tilde{}(h+d)$.

If we assume that $C_{i,j}$ and $T_{i,j}$ have length at most ℓ, the bounds given in Proposition 2 show that the product δ_2 of all $\delta_{2,i,j}$'s we consider admits an efficiently computable upper bound of the form $\Delta_2 = (dh\ell)^{O(1)}$. Together with the bound given in Lemma 20 for $\delta_{3,1}$, this finally leads us to define $\Delta_3 = \Delta_{3,1}\Delta_2$, and allows us to conclude the proof of Proposition 5.

7 Newton iteration

In this section, we give the details of a Newton iteration for systems with multiplicities that follows naturally from the deflation lemma. The following is essentially a particular case of the general algorithm from [30]; however, we give a simpler, self-contained presentation of the result we need, which will make it easy for us to give a cost analysis in the next section (as in the previous section, reusing directly the complexity estimates of [30] in our setting would lead to costs much higher than the one we will obtain).

Let $\mathbb{D}$ be a commutative ring and let $(x^\star, y^\star)$ in $\mathbb{D}$, m in $\mathbb{N}$ and H in $\mathbb{D}[X,Y]$ be such that the following holds:

N_1. $\frac{\partial^{m-1}H}{\partial Y^{m-1}}(x^\star, y^\star) = 0$.

N_2. $\frac{\partial^{m}H}{\partial Y^{m}}(x^\star, y^\star)$ is a unit in $\mathbb{D}$.

We suppose in addition that $\mathfrak{m}$ is an ideal in $\mathbb{D}$ such that $\mathfrak{m}^2 = (0)$. In what follows, we suppose that we know (x, y) in $\mathbb{D}$, with both $x - x^\star$ and $y - y^\star$ in $\mathfrak{m}$, and we will show how to recover $x^\star$ and $y^\star$, using a second equation K; further assumptions on $(x^\star, y^\star)$ will be introduced when needed.

Solving for Y. Given (x, y) in $\mathbb{D}$, with both $x - x^\star$ and $y - y^\star$ in $\mathfrak{m}$, Newton iteration applied (with respect to Y) to the polynomial $\frac{\partial^{m-1}H}{\partial Y^{m-1}}$ shows that there exist a unique y_x in $\mathbb{D}$ such that

$$\frac{\partial^{m-1}H}{\partial Y^{m-1}}(x, y_x) = 0, \quad y_x = y \bmod \mathfrak{m}. \tag{9}$$

Explicitly, y_x is given by

$$y_x = y - \frac{\frac{\partial^{m-1}H}{\partial Y^{m-1}}(x, y)}{\frac{\partial^{m}H}{\partial Y^{m}}(x, y)}. \tag{10}$$

This is well-defined, since $\frac{\partial^{m}H}{\partial Y^{m}}(x^\star, y^\star)$ being a unit implies that $\frac{\partial^{m}H}{\partial Y^{m}}(x, y)$ is a unit as well. Remark also that since $\mathfrak{m}^2 = (0)$ in $\mathbb{D}$, doing just one step of Newton iteration is sufficient to find the root y_x.

For $x = x^\star$, we obviously get $y_{x^\star} = y^\star$. Finally, note that in any case, y_x as defined above also satisfies $y_x = y^\star \bmod \mathfrak{m}$.

The implicit functions J^∞. Let ξ be a new variable, which we will use for power series over $\mathbb{D}$. Given x in $\mathbb{D}$, with $x - x^\star$ in $\mathfrak{m}$, our next goal is to compute, if it exists, a power series J^∞ in $\mathbb{D}[[\xi]]$ such that

$$\frac{\partial^{m-1} H}{\partial Y^{m-1}}(x + \xi, J^\infty) = 0, \quad J^\infty(0) = y^\star \bmod \mathfrak{m}. \tag{11}$$

Remark that very similar power series were already considered in Sections 5 and 6.

Lemma 23. *A power series J^∞ satisfies (11) if and only if it satisfies*

$$\frac{\partial^{m-1} H}{\partial Y^{m-1}}(x + \xi, J^\infty) = 0, \quad J^\infty(0) = y_x. \tag{12}$$

Proof. Of course, if J^∞ satisfies (12), it satisfies the seemingly weaker condition (11) (recall that $y_x = y^\star \bmod \mathfrak{m}$) . Conversely, suppose that J^∞ satisfies (11); we only have to prove that $J^\infty(0) = y_x$. Start from condition $\frac{\partial^{m-1} H}{\partial Y^{m-1}}(x + \xi, J^\infty) = 0$, and evaluate ξ at 0. This shows that the element $J^\infty(0) \in \mathbb{D}$ satisfies $\frac{\partial^{m-1} H}{\partial Y^{m-1}}(x, J^\infty(0)) = 0$ and $J^\infty(0) = y \bmod \mathfrak{m}$; the uniqueness of the solution of (9) proves that $J^\infty(0) = y_x$. □

Applying again Newton iteration, this time modulo the powers of ξ, we deduce that there exits a unique power series J_x^∞ in $\mathbb{D}[[\xi]]$ that satisfies (12), or equivalently (11). Of particular interest will be the power series $J^\star = J_{x^\star}^\infty$ associated to $x^\star$, which thus satisfies

$$\frac{\partial^{m-1} H}{\partial Y^{m-1}}(x^\star + \xi, J^\star) = 0, \quad J^\star(0) = y^\star.$$

Of course, since $x^\star$ and $y^\star$ are unknown to us, we cannot compute $J^\star$. However, for any x such that $x - x^\star \in \mathfrak{m}$, we claim that we have $J_x^\infty = J^\star \bmod \mathfrak{m}$. We will actually prove something more precise:

Lemma 24. *Let x be in $\mathbb{D}$, such that $x = x^\star \bmod \mathfrak{m}$. Then the equality*

$$J_x^\infty = J^\star + (x - x^\star)\frac{dJ^\star}{d\xi}$$

holds.

Proof. We prove that the power series $C_x = J^\star + (x - x^\star)\frac{dJ^\star}{d\xi}$ is equal to J_x^∞. In view of the uniqueness property, it suffices to prove that C_x satisfies both conditions in (11). We start by evaluating the functional in (11) at $x + \xi$ and C_x; this gives

$$\frac{\partial^{m-1} H}{\partial Y^{m-1}}(x + \xi, C_x) \quad = \quad \frac{\partial^{m-1} H}{\partial Y^{m-1}}(x^\star + \xi + (x - x^\star), J^\star + (x - x^\star)\frac{dJ^\star}{d\xi}).$$

Because $x - x^\star$ is in $\mathfrak{m}$, and $\mathfrak{m}^2 = (0)$, we can do a Taylor expansion at the first order, and deduce that the previous quantity is

$$\frac{\partial^{m-1} H}{\partial Y^{m-1}}(x^\star + \xi, J^\star) + (x - x^\star)\frac{\partial^m H}{\partial X \partial Y^{m-1}}(x^\star + \xi, J^\star) + (x - x^\star)\frac{dJ^\star}{d\xi}\frac{\partial^m H}{\partial Y^m}(x^\star + \xi, J^\star).$$

The first term above vanishes by definition of $J^\star$, so we are left with

$$(x - x^\star)\left(\frac{\partial^m H}{\partial X \partial Y^{m-1}}(x^\star + \xi, J^\star) + \frac{dJ^\star}{d\xi}\frac{\partial^m H}{\partial Y^m}(x^\star + \xi, J^\star)\right).$$

The right-hand factor is identically zero, since it is the derivative with respect to ξ of the defining equation for $J^\star$. Thus, we are done with the first condition for C_x.

To prove the second one, note that because $x - x^\star$ is in $\mathfrak{m}$, $C_x = J^\star \bmod \mathfrak{m}$. As a consequence, $C_x(0) = J^\star(0) \bmod \mathfrak{m}$, and thus $C_x(0) = y^\star \bmod \mathfrak{m}$. This proves the second condition for C_x, and thus that $C_x = J_x^\infty$. □

Using the second equation. Let us now consider a further polynomial K in $\mathbb{D}[X, Y]$, together with an integer $a \geq 0$. To x in $\mathbb{D}$, such that $x = x^\star \bmod \mathfrak{m}$, we now associate

$$S_x^\infty = \frac{\partial^a K}{\partial Y^a}(x + \xi, J_x^\infty),$$

which is a well-defined power series in $\mathbb{D}[[\xi]]$. Inspired by the notation above, we write $S^\star$ for the particular case $x = x^\star$. The following lemma shows that S_x^∞ is a first-order approximation of $S^\star$.

Lemma 25. *Let x be in $\mathbb{D}$, such that $x = x^\star \bmod \mathfrak{m}$. The equality*

$$S_x^\infty = S^\star + (x - x^\star)\frac{dS_x^\infty}{d\xi}$$

holds.

Proof. Let us write $P = \frac{\partial^a K}{\partial Y^a}$. We have to prove that

$$P(x + \xi, J_x^\infty) = P(x^\star + \xi, J^\star) + (x - x^\star)\frac{d}{d\xi}P(x + \xi, J_x^\infty).$$

The proof is similar to that in the previous lemma. Recall that $J_x^\infty = J^\star + (x - x^\star)\frac{dJ^\star}{d\xi}$. Thus, the left-hand side is

$$P\left(x^\star + \xi + (x - x^\star), J^\star + (x - x^\star)\frac{dJ^\star}{d\xi}\right),$$

which gives, after a first-order Taylor expansion,

$$\begin{aligned} P(x^\star + \xi, J^\star) + (x - x^\star)\frac{\partial P}{\partial X}(x^\star + \xi, J^\star) + (x - x^\star)\frac{dJ^\star}{d\xi}\frac{\partial P}{\partial Y}(x^\star + \xi, J^\star) \\ = P(x^\star + \xi, J^\star) + (x - x^\star)\frac{d}{d\xi}P(x^\star + \xi, J^\star). \end{aligned}$$

Because $x = x^\star \bmod \mathfrak{m}$ and $J_x^\infty = J^\star \bmod \mathfrak{m}$, we deduce that $P(x+\xi, J_x^\infty) = P(x^\star + \xi, J^\star) \bmod \mathfrak{m}$, and this remains true after differentiation with respect to ξ. Since $x - x^\star$ is in $\mathfrak{m}$, and $\mathfrak{m}^2 = (0)$, we obtain that

$$(x-x^\star)\frac{d}{d\xi}P(x^\star+\xi, J^\star) = (x-x^\star)\frac{d}{d\xi}P(x+\xi, J_x^\infty).$$

Thus, the lemma is proved. □

As in Section 6, let us write $\mathsf{cf}(S, \xi^j)$ for the the coefficient of ξ^j in a power series $S \in \mathbb{D}[[\xi]]$. We can then make our last assumptions on $(x^\star, y^\star)$: there exists an integer $n \geq 1$ such that

N_3. $\mathsf{cf}(S^\star, \xi^{n-1}) = 0$,

N_4. $n\,\mathsf{cf}(S^\star, \xi^n)$ is a unit in $\mathbb{D}$.

The following lemma finally allows us to compute $x^\star$, assuming we know $x^\star \bmod \mathfrak{m}$ and $y^\star \bmod \mathfrak{m}$.

Lemma 26. *Let x be in $\mathbb{D}$, such that $x = x^\star \bmod \mathfrak{m}$. If $(x^\star, y^\star)$ satisfies N_1 - N_4, then $n\,\mathsf{cf}(S_x^\infty, \xi^n)$ is a unit in $\mathbb{D}$ and*

$$x^\star = x - \frac{1}{n}\frac{\mathsf{cf}(S_x^\infty, \xi^{n-1})}{\mathsf{cf}(S_x^\infty, \xi^n)}.$$

Proof. To prove the first item, remark that the previous lemma shows in particular that $S_x^\infty = S^\star \bmod \mathfrak{m}$ and extract the coefficient of ξ^n, we deduce that $\mathsf{cf}(S^\star, \xi^n) = \mathsf{cf}(S_x^\infty, \xi^n) \bmod \mathfrak{m}$, and thus $n\,\mathsf{cf}(S^\star, \xi^n) = n\,\mathsf{cf}(S_x^\infty, \xi^n) \bmod \mathfrak{m}$. Since the former is a unit, the latter must be a unit too.

To conclude, start from the equality $S_x^\infty = S^\star + (x - x^\star)\frac{dS_x^\infty}{d\xi}$ proved above, and extract the coefficient of degree $n-1$ (with respect to ξ) on both sides. Using N_3 gives

$$\mathsf{cf}(S_x^\infty, \xi^{n-1}) = (x - x^\star)\, n\,\mathsf{cf}(S_x^\infty, \xi^n).$$

Since we proved that $n\,\mathsf{cf}(S_x^\infty, \xi^n)$ is a unit in $\mathbb{D}$, the claim follows. □

Once $x^\star$ is known, we can also recover $y^\star$. One option is to apply the Newton iteration of Eq. (10), but we will prefer the following method, which will not require further evaluations. We know that $J_x^\infty = J^\star + (x - x^\star)\frac{dJ^\star}{d\xi}$; since $J^\star = J_x^\infty \bmod \mathfrak{m}$, we can rewrite $J_x^\infty = J^\star + (x - x^\star)\frac{dJ_x^\infty}{d\xi}$. Since $\mathsf{cf}(J^\star, \xi^0) = y^\star$, we deduce

$$y^\star = \mathsf{cf}(J_x, \xi^0) - (x - x^\star)\mathsf{cf}(J_x^\infty, \xi^1). \tag{13}$$

To conclude this section, we state the following result, which will be used in various contexts (such as reduction modulo an integer N).

Lemma 27. *Let $\mathbb{D}'$ be a commutative ring, φ a ring morphism $\mathbb{D} \to \mathbb{D}'$, and let Φ denote the extension of φ to a ring morphism $\mathbb{D}[X,Y] \to \mathbb{D}'[X,Y]$. Suppose that $\varphi(n)$ is a unit in $\mathbb{D}'$.*

If $(x^\star, y^\star)$ and (m, H, n, a, K) satisfy N_1 - N_4 over $\mathbb{D}$, $(\varphi(x^\star), \varphi(y^\star))$ and $(m, \Phi(H), n, a, \Phi(K))$ satisfy N_1 - N_4 over $\mathbb{D}'$.

Proof. Since they describe vanishing, resp. invertibility properties of either $\frac{\partial^{m-1}H}{\partial Y^{m-1}}$ or $\frac{\partial^m H}{\partial Y^m}$ at $(x^\star, y^\star)$, N_1 and N_2 clearly remain true through the application of φ. To prove that N_3 and N_4, let us denote by $J^\star \in \mathbb{D}[[\xi]]$ the power series associated to $(x^\star, y^\star)$ as above, and $J'^\star \in \mathbb{D}'[[\xi]]$ the power series associated to $(\varphi(x^\star), \varphi(y^\star))$.

By uniqueness of the solution of (12), we deduce that $\Phi'(J^\star) = J'^\star$, where Φ' denotes the extension of φ to a ring morphism $\mathbb{D}[[\xi]] \to \mathbb{D}'[[\xi]]$. From there, we deduce that $\Phi'(S^\star) = S'^\star$, where $S^\star$ is as above, and $S'^\star$ is the power series

$$S'^\star = \frac{\partial^a \Phi(K)}{\partial Y^a}(\varphi(x^\star) + \xi, J'^\star).$$

Extracting coefficients of $S'^\star$, we see that $\mathsf{cf}(S'^\star, \zeta^{n-1}) = 0$ and that $\mathsf{cf}(S'^\star, \zeta^n)$ is a unit in $\mathbb{D}'$. Since $\varphi(n)$ is invertible as well, we are done. □

8 Main algorithm

We can finally present the main algorithm and analyze its complexity. All along this section, we use the following notation. The input is a pair of polynomials F and G with coefficients in $\mathbb{Z}$, with degree at most d and length at most h. We suppose that these polynomials satisfy the coprimality assumption H_1 of Section 2, so that the associated polynomial $\Gamma \in \mathbb{Z}[T]$ is well-defined.

We are going to apply a random change of variables, and compute modulo a random prime. These choices will be done in two steps: first we choose a large enough shearing coefficient t_0 to ensure that it is "lucky" with high probability, and a first prime p_0; in a second time, we use t_0 to look for another change of variables t with smaller length, together with another prime p, and we apply the lifting process to F_t and G_t starting from the solution modulo p.

8.1 Choosing t_0 and p_0

In this first section, we discuss the choices of t_0 and p_0, and quantify their probability of success. As said in the introduction, we suppose that we have an oracle $\mathscr{O}$, which on input an integer B returns a random prime in the interval $\{B+1, \ldots, 2B\}$, chosen uniformly among these primes.

Given $\mathcal{P} \geq 1$, our overall goal is to obtain a probability of success of at least $1 - 1/2^{\mathcal{P}}$. Let us first choose an integer t_0 at random in the set $\{1, \ldots, 2^{\mathcal{P}+5}d^4\}$; remark in particular that the length of t_0 is at most $\mathcal{P} + 5 + 4\mathrm{len}(d) = O(\mathcal{P} + \log(d))$. Because Γ is nonzero of

degree at most $6d^4$, the probability that t_0 cancels it is at most $1/2^{\mathcal{P}+2}$. In what follows, let us assume that $\Gamma(t_0)$ is nonzero.

Since t_0 is in $\{1, \ldots, 2^{\mathcal{P}+5}d^4\}$, using the length bound on Γ given in Lemma 4 and the evaluation bound b_1, the length of $\Gamma(t_0)$ can be bounded by an efficiently computable integer $H_0(\mathcal{P}, d, h) = (\mathcal{P}dh)^{O(1)}$. Let us then define

$$\Delta_0(\mathcal{P}, d, h) = \max(12d^4, 2^{\mathcal{P}+3}H_0(\mathcal{P}, d, h)),$$

and the set

$$\Lambda_0(\mathcal{P}, d, h) = \{\Delta_0(\mathcal{P}, d, h) + 1, \ldots, 2\Delta_0(\mathcal{P}, d, h)\};$$

the reason for requiring $\Delta_0(\mathcal{P}, d, h) \geq 12d^4$ is explained in Subsection 8.3. With these choices, we have the following quantitative estimates:

- The set $\Lambda_0(\mathcal{P}, d, h)$ contains at least $\Delta_0(\mathcal{P}, d, h)/(2\log(\Delta_0(\mathcal{P}, d, h)))$ primes [18, proof of Theorem 18.8].
- There are at most $\log_{\Delta_0(\mathcal{P},d,h)}(|\Gamma(t_0)|)$ primes in $\Lambda_0(\mathcal{P}, d, h)$ that divide $\Gamma(t_0)$.

Let us call the oracle $\mathscr{O}$, with input $\Delta_0(\mathcal{P}, d, h)$; as output, we get a random prime p_0 in $\Lambda_0(\mathcal{P}, d, h)$. The probability of p_0 dividing $\Gamma(t_0)$ is thus at most

$$\frac{\log_{\Delta_0(\mathcal{P},d,h)}(|\Gamma(t_0)|)}{\frac{\Delta_0(\mathcal{P},d,h)}{2\log\Delta_0(\mathcal{P},d,h)}} = 2\frac{\log(|\Gamma(t_0)|)}{\Delta_0(\mathcal{P}, d, h)} \leq \frac{1}{2^{\mathcal{P}+2}}.$$

Taking into account the choices of both t_0 and p_0, the probability that $\Gamma(t_0)$ is zero modulo p_0 is thus at most $1/2^{\mathcal{P}+1}$.

When $\Gamma(t_0)$ does not vanish modulo p_0, by Corollary 1, we deduce that the polynomials $F_{t_0} \bmod p_0$ and $G_{t_0} \bmod p_0$ are in general position, and that their zero-set has the same cardinality as $V(F, G)$. This will be the main properties we will use about them.

8.2 Computations modulo p_0

In all that follows, we suppose that t_0 and p_0 have been chosen such that $\Gamma(t_0)$ is nonzero modulo p_0. The first part of the algorithm consists in computing the polynomials $\mathsf{SL}(F_{t_0} \bmod p_0, G_{t_0} \bmod p_0)$.

Lemma 28. *Given F and G, one can compute $F_{t_0} \bmod p_0$ and $G_{t_0} \bmod p_0$, as well as $\mathsf{SL}(F_{t_0} \bmod p_0, G_{t_0} \bmod p_0)$, using $O^{\tilde{}}(\mathcal{P} + d^2h + d^3\log(p_0))$ bit operations.*

Proof. First, we have to compute $t_0 \bmod p_0$; this is done in time $O^{\tilde{}}(\log(t_0) + \log(p_0)) = O^{\tilde{}}(\mathcal{P} + \log(d) + \log(p_0))$ by fast Euclidean division [18].

Next, we reduce F and G modulo p_0; this takes $O^{\tilde{}}(d^2(h + \log(p_0))$ bit operations, by fast Euclidean division of each coefficient. Then, we apply the change of variable $X \mapsto X + t_0Y$ to $F \bmod p_0$ and $G \bmod p_0$, which gives $F_{t_0} \bmod p_0$ and $G_{t_0} \bmod p_0$ in $O^{\tilde{}}(d^2\log(p_0))$ bit operations.

We know that $(F_{t_0} \bmod p_0, G_{t_0} \bmod p_0)$ are in general position. To compute their Shape Lemma representation, we apply the algorithm of [29, Proposition 1], which runs in time $O^{\sim}(d^3 \log(p_0))$. There is only one minor difference: one step in that algorithm should be avoided (Step 6, which removes multiple solutions from $V(F_{t_0} \bmod p_0, G_{t_0} \bmod p_0)$ – we do not want to discard them here). □

Our next step is then to find an integer t of smaller length than t_0 such that $\Gamma(t) \neq 0$; we will as well have to change our prime.

8.3 Choosing t and p

To find a suitable t, we recall notation from Section 2: we let $\mathfrak{f}$ and $\mathfrak{g}$ be the leading coefficients of respectively F_T and G_T with respect to Y; they lie in $\mathbb{Z}[T]$. We also let $\mathfrak{a} \in \mathbb{Z}[T]$ be the leading coefficient of the resultant $\mathfrak{A} = \mathrm{res}(F_T, G_T, Y)$ with respect to X.

Lemma 29. *Let $t \in \mathbb{Z}$ be such that neither $\mathfrak{f}(t) \bmod p_0$ nor $\mathfrak{g}(t) \bmod p_0$ vanishes, and such that X is a separating element for $V(F_t \bmod p_0, G_t \bmod p_0) \subset \overline{\mathbb{F}}_{p_0}{}^2$. Then $\Gamma(t) \bmod p_0$ is nonzero.*

Proof. We know that $\Gamma \bmod p_0$ is nonzero, so in view of Lemma 6, it is enough to prove that $\mathfrak{a}(t) \bmod p_0$ is nonzero.

Writing $\phi : \mathbb{Z}[T] \to \mathbb{F}_{p_0}$ given by $\phi(r) = r(t) \bmod p_0$, the first item in Lemma 5 implies that the cardinality of $\pi(V(F_t \bmod p_0, G_t \bmod p_0))$ is the number of pairwise distinct roots of $\mathfrak{A}(t, X) \bmod p_0$ in $\overline{\mathbb{F}_{p_0}}$. Now, we know that the cardinality of $V(F_{t_0} \bmod p_0, G_{t_0} \bmod p_0)$ is finite, so it must also be the same for $V(F_t \bmod p_0, G_t \bmod p_0)$, and thus for its projection. This implies that $\mathfrak{A}(t, X) \bmod p_0$ has finitely many roots, so it is not the zero polynomial. As a consequence, the leading coefficient $\mathfrak{a}$ of $\mathfrak{A}$ does not vanish through ϕ, which is what we wanted to prove. □

To find a suitable t, the algorithm is now straightforward: we try enough such values at random, since the test implied by the above lemma can be performed efficiently enough.

Lemma 30. *One can find with probability at least $1 - 1/2^{\mathcal{P}+2}$ an integer $t \in \{1, \ldots, 12d^4\}$ such that $\Gamma(t) \bmod p_0$ is nonzero using $d^{2+\varepsilon} O^{\sim}(\mathcal{P} \log(p_0))$ bit operations.*

Proof. Since $\Gamma(t_0) \bmod p_0$ is nonzero, $\Gamma \bmod p_0$ is not the zero polynomial. Since Γ has degree at most $6d^4$ (Lemma 4), and since p_0 has been chosen at least equal to $12d^4$, there are at most $6d^4$ integers t among $\{1, \ldots, 12d^4\}$ for which $\Gamma(t) \bmod p_0 = 0$.

The algorithm is then simple: pick at random $\mathcal{P} + 2$ values in this set, test if they cancel $\Gamma \bmod p_0$, and return one for which we find this is not the case, if any (otherwise, the whole algorithm may just return fail). This succeeds with probability at least $1 - 1/2^{\mathcal{P}+2}$, so we are left with the cost analysis.

Given t in the set $\{1, \ldots, 12d^4\}$, to test whether $\Gamma(t) \bmod p_0$ vanishes, we use the previous lemma.

We first test whether $\mathfrak{f}(t) \bmod p_0$ or $\mathfrak{g}(t) \bmod p_0$ vanish. The polynomials $\mathfrak{f}$ and $\mathfrak{g}$ are the highest degree forms of F and G, and since Γ does not vanish modulo p_0, neither do they; thus, $\mathfrak{f} \bmod p_0$ and $\mathfrak{g} \bmod p_0$ are the highest degree forms of respectively $F \bmod p_0$ and $G \bmod p_0$. We computed the latter polynomials in the course of Lemma 28, so all we have to do is evaluate them at t; this takes $O(d)$ operations in $\mathbb{F}_{p_0}$, so $O^{\sim}(d \log(p_0))$ bit operations.

Next, we verify whether X is a separating element for $V(F_t \bmod p_0, G_t \bmod p_0)$. Since we know $(P_0, S_0) = \mathsf{SL}(F_{t_0} \bmod p_0, G_{t_0} \bmod p_0)$, performing the above test amounts to testing whether the characteristic polynomial of $X + (t - t_0)S_0$ modulo P_0 is squarefree. We can compute this characteristic polynomial using Kedlaya and Umans' power projection algorithm [23], for $d^{2+\varepsilon}O^{\sim}(\log(p_0))$ bit operations; the squarefreeness test is not more expensive.

Altogether, the cost for testing one value of t is $d^{2+\varepsilon}O^{\sim}(\log(p_0))$ bit operations; the conclusion follows. □

We will now assume that we have found a suitable t, and we will now replace our prime p_0 by a new prime p: our choice of p_0 did not take into account issues related to the σ-decomposition of $V(F_{t_0}, G_{t_0})$; even if it had, it would not be enough to guarantee good reduction for $V(F_t, G_t)$. In order to choose p, note that we have the following length bounds:

- Since t is in $\{1, \dots, 12d^4\}$, using the length bound on Γ given in Lemma 4 and the evaluation bound b_1, the length of $\Gamma(t)$ can be bounded by an efficiently computable integer $H_1(d, h) = (dh)^{O(1)}$.
- From Corollary 1 again, all polynomials appearing in the Shape Lemma representation of $V_t = V(F_t, G_t)$, and of all its $\mathbb{Q}$-definable subsets, have length at most $H_2(d, h) = B_{\mathsf{SL}}(d, h, \mathrm{len}(12d^4))$, which is $O^{\sim}(dh + d^2)$, and thus $(dh)^{O(1)}$.
- The polynomials F_t and G_t have degree d; using the evaluation bound b_1 again, we see that their length is at most $H_3(d, h) = h + d(\mathrm{len}(12d^4) + 3)$, which is again $(dh)^{O(1)}$.

Let us then define $\Delta_4(d, h) = \Delta_3(d, H_3(d, h), H_2(d, h))\, H_1(d, h)$, where Δ_3 is the function defined in Proposition 5. Since all of Δ_3, H_2, H_1, H_3 are efficiently computable, Δ_4 itself is an efficiently computable function of d, h, and we have $\Delta_4(d, h) = (dh)^{O(1)}$.

Consider now the integer $\delta_4 = \delta_3\Gamma(t)$, where δ_3 is the nonzero integer associated to F_t and G_t by Proposition 5. The construction above shows that δ_4 is a nonzero integer of length at most $\Delta_4(d, h)$. Let us finally define

$$\Delta_4'(\mathcal{P}, d, h) = \max(d^2, 2^{\mathcal{P}+3}\Delta_4(d, h)),$$

and the set

$$\Lambda_4(\mathcal{P}, d, h) = \{\Delta_4'(d, h) + 1, \dots, 2\Delta_4'(d, h)\};$$

the reason for requiring $\Delta_4' > d^2$ appears in Lemma 33. Let us call the oracle $\mathscr{O}$, with input $\Delta_4'(\mathcal{P}, d, h)$; as output, we get a random prime p in $\Lambda_4(\mathcal{P}, d, h)$. Proceeding as in Subsection 8.1, we see that the probability of p dividing δ_4 is at most

$$\frac{\log_{\Delta_4'(\mathcal{P},d,h)}(\delta_4)}{\frac{\Delta_4'(\mathcal{P},d,h)}{2\log\Delta_4'(\mathcal{P},d,h)}} = 2\frac{\log(\delta_4)}{\Delta_4'(\mathcal{P}, d, h)} \leq \frac{1}{2^{\mathcal{P}+2}}.$$

Let us assume that this is not the case; we will see that the algorithm necessarily succeeds. This concludes the description of the randomized part of the algorithm. The probability of choosing "lucky" (t_0, p_0) was at least $1 - 1/2^{\mathcal{P}+1}$, and the same holds for (t, p), so the overall probability of success is at least $1 - 1/2^{\mathcal{P}}$, as claimed.

For what follows, it will be useful to remember that the prime p is at most $2^{\mathcal{P}+4}\Delta_4(\mathcal{P}, d, h)$, which is $2^{O(\mathcal{P})}(dh)^{O(1)}$.

8.4 Computations modulo p

In all that follows, we suppose that t and p have been chosen such that $\Gamma(t)$ is nonzero, and such that the integer δ_4 defined above does not vanish modulo p.

In particular, by Corollary 1, both systems (F_t, G_t) and $(F_t \bmod p, G_t \bmod p)$ are in general position, over respectively $\mathbb{Q}[X, Y]$ and $\mathbb{F}_p[X, Y]$, and we have the specialization property $\mathsf{SL}(F_t, G_t) \bmod p = \mathsf{SL}(F_t \bmod p, G_t \bmod p)$. Our next step consists in computing the latter polynomials, together with the corresponding σ-decomposition.

Lemma 31. *Given F and G, one can compute $F_t \bmod p$ and $G_t \bmod p$, as well as $\mathsf{SL}(F_t \bmod p, G_t \bmod p)$, using $O\tilde{}(d^2h + d^3\log(p))$ bit operations.*

Proof. The proof is similar to that of Lemma 31; the only difference is that t has length $O(\log(d))$, whereas t_0 had length $O(\mathcal{P} + \log(d))$. □

Let $\Sigma = [\,(C_{i,j}, T_{i,j}, m_i, \mathtt{H}_i, n_{i,j}, a_{i,j}, \mathtt{K}_{i,j})\,]_{1 \le i \le s, j \in D_i}$ be the σ-decomposition of $V(F_t, G_t)$. Since p does not divide $\Gamma(t)$, Corollary 1 proves that p cancels the denominator of none of the coefficients of the polynomials in Σ. In addition, Corollary 1 and the fact that p does not divide the integer δ_3 of Proposition 5 (applied to F_t and G_t) imply that $\Sigma \bmod p$ is the σ-decomposition of $V(F_t \bmod p, G_t \bmod p)$.

Lemma 32. *Given $F_t \bmod p$ and $G_t \bmod p$, as well as $\mathsf{SL}(F_t \bmod p, G_t \bmod p)$, one can compute $\Sigma \bmod p$ using $d^{3+\varepsilon}O\tilde{}(\log(p))$ bit operations.*

Proof. This is Proposition 4, applied to $F_t \bmod p$ and $G_t \bmod p$. □

Since none of the denominators of the coefficients of the polynomials in either $\mathsf{SL}(F_t, G_t)$ or Σ vanishes modulo p, all these polynomials can be seen in $\mathbb{Z}_p[X]$; the same obviously holds for F and G, which have integer coefficients.

For fixed indices i, j, we can then define the residue class ring $\mathbb{E}_{i,j} = \mathbb{Z}_p[X]/\langle C_{i,j}\rangle$, as well as $x_{i,j}^{(p)}$ as the residue class of X in $\mathbb{E}_{i,j}$, and $y_{i,j}^{(p)}$ as the residue class of $T_{i,j}$. We saw a similar construction at the end of Section 6, in the proof of Proposition 5; at the time, we were working over $\mathbb{D}_{i,j} = \mathbb{Q}[X]/\langle C_{i,j}\rangle$, with symbols $x_{i,j}$ and $y_{i,j}$, whereas we now use $\mathbb{E}_{i,j} = \mathbb{Z}_p[X]/\langle C_{i,j}\rangle$.

Lemma 33. *For any index i, j, the point $(x_{i,j}^{(p)}, y_{i,j}^{(p)})$ and $(m_i, H_i, n_{i,j}, a_{i,j}, K_{i,j})$ satisfy conditions N_1 - N_4 of Section 7 over the ring $\mathbb{E}_{i,j}$.*

Proof. In all that follows, i and j are fixed. Recall that by definition of the signature, we know that, in $\mathbb{D}_{i,j}$,

$$\frac{\partial^{m_i-1}H_i}{\partial Y^{m_i-1}}(x_{i,j}, y_{i,j}) = 0 \quad \text{and} \quad \frac{\partial^{m_i}H_i}{\partial Y^{m_i}}(x_{i,j}, y_{i,j}) \text{ is a unit.}$$

To prove N_1 and N_2 for $(x_{i,j}^{(p)}, y_{i,j}^{(p)})$, we have to prove that similar identities hold in $\mathbb{E}_{i,j}$, that is, with coefficients in $\mathbb{Z}_p$ instead of $\mathbb{Q}$.

The first condition obviously carries over to $\mathbb{E}_{i,j}$, since it is an equality involving polynomials with coefficients in $\mathbb{Q}$, and p divides no denominator in these polynomials. The second condition is dealt with similarly, upon noticing that by assumption on p, the term $\frac{\partial^{m_i}H_i}{\partial Y^{m_i}}(X, T_{i,j}) \bmod p$ remains a unit modulo $C_{i,j} \bmod p$.

Returning to calculations over $\mathbb{D}_{i,j}$, we can as before define a power series $J_{i,j}^{\infty} \in \mathbb{D}_{i,j}[[\xi]]$ characterized by the conditions

$$\frac{\partial^{m_i-1}H_i}{\partial Y^{m_i-1}}(x_{i,j} + \xi, J_{i,j}^{\infty}) = 0 \quad \text{and} \quad J_{i,j}^{\infty}(0) = y_{i,j}.$$

Defining further the power series

$$S_{i,j} = \frac{\partial^{a_{i,j}}K_{i,j}}{\partial Y^{a_{i,j}}}(x_{i,j} + \xi, J_{i,j}^{\infty}) \in \mathbb{D}_{i,j}[[\xi]],$$

we saw (for instance in the proof of Proposition 5) that

$$\mathsf{cf}(S_{i,j}, \xi^{n_{i,j}-1}) = 0 \quad \text{and} \quad \mathsf{cf}(S_{i,j}, \xi^{n_{i,j}}) \text{ is a unit.}$$

Once more, to prove N_3 and N_4, we have to derive similar statements in $\mathbb{E}_{i,j}$. First, remark that the fact that $\frac{\partial^{m_i}H_i}{\partial Y^{m_i}}(x_{i,j}, y_{i,j})$ remains a unit in $\mathbb{E}_{i,j}$ implies that all coefficients of $J_{i,j}^{\infty}$ are well-defined modulo p, so that we may consider $J_{i,j}^{\infty}$ in $\mathbb{E}_{i,j}[[\xi]]$. Then, the equality $\mathsf{cf}(S_{i,j}, \xi^{n_{i,j}-1}) = 0$ in $\mathbb{E}_{i,j}$ remains obviously true in this context.

To prove that $\mathsf{cf}(S_{i,j}, \xi^{n_{i,j}})$ is a unit, it is sufficient to prove that it is a unit modulo p, that is, in $\mathbb{F}_p[X]/\langle C_{i,j} \bmod p\rangle$. This is a direct consequence of the fact that p does not divide the integer δ_3 of Proposition 5.

The last point we need to prove is that $n_{i,j}$ is a unit in $\mathbb{E}_{i,j}$. This follows from the constraint that $p > d^2$, combined with the inequality $n_{i,j} \leq d^2$. □

8.5 Analysis of one lifting step

In the previous section, we showed how to compute $\Sigma \bmod p$. Supposing that we know $\Sigma \bmod N$, for N some power of p, we now show how to compute $\Sigma \bmod N^2$. The main result of this section is the following proposition.

Proposition 6. *Given F, G and $\Sigma \bmod N$, for N a power of p, one can compute $\Sigma \bmod N^2$ using $d^{2+\varepsilon}\tilde{O}(\log(N))$ bit operations.*

In what follows, we denote by $\mathbb{A}$ the ring $\mathbb{Z}/N^2\mathbb{Z}$, over which all computations will be done. For any i in $\{1,\ldots,s\}$ and j in D_i, we thus assume that we know $c_{i,j} = C_{i,j} \bmod N$ and $t_{i,j} = T_{i,j} \bmod N$. Our goal is to compute $c'_{i,j} = C_{i,j} \bmod N^2$ and $t'_{i,j} = T_{i,j} \bmod N^2$.

Let us write

$$\mathbb{F}_{i,j} = \mathbb{A}[X]/\langle c_{i,j}\rangle \quad \text{and} \quad \mathbb{F}'_{i,j} = \mathbb{A}[X]/\langle c'_{i,j}\rangle,$$

where in the former case, we view $c_{i,j}$ as a polynomial in $\mathbb{A}[X] = \mathbb{Z}/N^2\mathbb{Z}$ by considering its canonical lift. Thus, we can compute in $\mathbb{F}_{i,j}$, but not in $\mathbb{F}'_{i,j}$, since $c'_{i,j}$ is unknown. In what follows, we write $\alpha_{i,j}$ for the residue class of X in $\mathbb{F}_{i,j}$, and $\beta_{i,j}$ for that of $t_{i,j}$; similarly, we let $\alpha'_{i,j}$ for the residue class of X in $\mathbb{F}'_{i,j}$, and $\beta'_{i,j}$ that of $t'_{i,j}$.

Starting from the claim in Lemma 33 and applying Lemma 27 to the homomorphism $\mathbb{E}_{i,j} \to \mathbb{F}'_{i,j}$ of reduction modulo N^2, we deduce that over the ring $\mathbb{F}'_{i,j}$, the point $(\alpha'_{i,j}, \beta'_{i,j}) \in {\mathbb{F}'_{i,j}}^2$ and $(m_i, H_i \bmod N^2, n_{i,j}, a_{i,j}, K_{i,j} \bmod N^2)$ satisfy assumptions N_1 - N_4 of Section 7.

Now, Proposition 1 in [30] proves further that there exists a ring isomorphism $\phi : \mathbb{F}'_{i,j} \to \mathbb{F}_{i,j}$, which reduces to the identity modulo N, and leaves $\mathbb{A} = \mathbb{Z}/N^2\mathbb{Z}$ invariant. Thus, by Lemma 27 again, there exists $(x^\star_{i,j}, y^\star_{i,j}) = (\phi(\alpha'_{i,j}), \phi(\beta'_{i,j})) \in {\mathbb{F}_{i,j}}^2$ such that:

- $x^\star_{i,j} = \alpha_{i,j} \bmod N$ and $y^\star_{i,j} = \beta_{i,j} \bmod N$;
- $(x^\star_{i,j}, y^\star_{i,j})$ and $(m_i, H_i \bmod N^2, n_{i,j}, a_{i,j}, K_{i,j} \bmod N^2)$ satisfy conditions N_1 - N_4 of Section 7 over the ring $\mathbb{F}_{i,j}$ (remark that since $H_{i,j}$ and $K_{i,j}$ have coefficients in $\mathbb{Z}$, $H_{i,j} \bmod N^2$ and $K_{i,j} \bmod N^2$ are left unchanged by ϕ).

Since $N^2 = 0$ in $\mathbb{A}$, and thus in $\mathbb{F}_{i,j}$, we will apply the algorithm of Section 7 with the ideal $\mathfrak{m} = \langle N\rangle$, in order to first compute $x^\star_{i,j}$ and $y^\star_{i,j}$ in $\mathbb{F}_{i,j}$; in a second stage, we will deduce $c'_{i,j}$ and $t'_{i,j}$. The computation of $x^\star_{i,j}$ and $y^\star_{i,j}$ proceeds itself in several steps, which follow the description in Section 7.

Computing $F_t \bmod N^2$ and $G_t \bmod N^2$. Remark that all polynomials $H_i \bmod N^2$ and $K_{i,j} \bmod N^2$ are in $\{F_t \bmod N^2, G_t \bmod N^2\}$, so we need to compute these two polynomials. This is done exactly as in Lemma 28, using $\tilde{O}(\mathcal{P} + d^2h + d^2\log(N))$ bit operations (the cost stated in Lemma 28 involves an extra term $d^3\log(p_0)$ for solving the system (F_{t_0}, G_{t_0}) modulo p_0, which we do not need here).

In what follows, we can thus assume that these polynomials are known.

Computing all $v_{i,j}$'s. First, applying Eq. (10), we compute the elements $v_{i,j}$ in $\mathbb{F}_{i,j}$, such that for all i, j, we have

$$\frac{\partial^{m_i-1} H_i}{\partial Y^{m_i-1}}(\alpha_{i,j}, v_{i,j}) = 0$$

in $\mathbb{F}_{i,j}$, and such that $v_{i,j} = \beta_{i,j} \bmod N$. This is done in Algorithm $\mathsf{lift\_y}$ below.

Algorithm 7: $\mathsf{lift\_y}(F_t, G_t, \Sigma)$

Input: polynomials $F_t, G_t \bmod N^2$, the σ-decomposition $\Sigma \bmod N$
Output: $[\, v_{i,j} \,]_{1 \le i \le s, j \in D_i}$

$I = [\, (i,j) \mid 1 \le i \le s,\ j \in D_i]$
$I_F = [\, (i,j) \mid 1 \le i \le s,\ j \in D_i$ and $\mathtt{H}_i = \text{"}\mathtt{F}\text{"}\,]$
$I_G = [\, (i,j) \mid 1 \le i \le s,\ j \in D_i$ and $\mathtt{H}_i = \text{"}\mathtt{G}\text{"}\,]$
$[\, \eta_{i,j} \,]_{(i,j)\in I_F} = [\, \frac{\partial^{m_i-1} F_t}{\partial Y^{m_i-1}}(\alpha_{i,j}, \beta_{i,j}) \,]_{(i,j)\in I_F}$ calculations are done in $\mathbb{F}_{i,j}$
$[\, \eta'_{i,j} \,]_{(i,j)\in I_F} = [\, \frac{\partial^{m_i} F_t}{\partial Y^{m_i}}(\alpha_{i,j}, \beta_{i,j}) \,]_{(i,j)\in I_F}$
$[\, \gamma_{i,j} \,]_{(i,j)\in I_G} = [\, \frac{\partial^{m_i-1} G_t}{\partial Y^{m_i-1}}(\alpha_{i,j}, \beta_{i,j}) \,]_{(i,j)\in I_G}$
$[\, \gamma'_{i,j} \,]_{(i,j)\in I_G} = [\, \frac{\partial^{m_i} G_t}{\partial Y^{m_i}}(\alpha_{i,j}, \beta_{i,j}) \,]_{(i,j)\in I_G}$
return $[\, v_{i,j} \,]_{(i,j)\in I} = [\, \beta_{i,j} - \eta_{i,j}/\eta'_{i,j} \,]_{(i,j)\in I_F}$ cat $[\, \beta_{i,j} - \gamma_{i,j}/\gamma_{i,j} \,]_{(i,j)\in I_G}$

Correctness follows from Eq. (10). Regarding running time, the bulk of the cost is the computation of the sequences $[\, \eta_{i,j} \,]$, $[\, \eta'_{i,j} \,]$, $[\, \gamma_{i,j} \,]$, $[\, \gamma'_{i,j} \,]$. Indeed, at the last step, the division corresponding to index (i,j) can be done in $O^{\sim}(\deg(C_{i,j}))$ operations in $\mathbb{A}$, which is $O^{\sim}(\deg(C_{i,j}) \log(N))$ bit operations: summing over all (i,j) and using Lemma 18, the total adds up to $O^{\sim}(d^2 \log(N))$.

Let us thus for instance explain how to compute $[\, \eta_{i,j} \,]$ and $[\, \eta'_{i,j} \,]$. This is a direct application of Algorithm $\mathsf{normal\_forms}$ of Proposition 3, with input the lists L, L', L'' and F, with $L = [\, (0, m_i) \,]_{(i,j)\in I}$, $L' = [\, c_{i,j} \,]_{(i,j)\in I}$ and $L'' = [\, y_{i,j} \,]_{(i,j)\in I}$. Lemma 18 implies that

$$\sum_{1 \le i \le s, j \in D_i} (m_{i,j} + 1) \deg(C_{i,j}) = O(d^2),$$

so we are under the conditions of Proposition 3. This implies that for any $\varepsilon > 0$, all $[\, \eta_{i,j} \,]$ and $[\, \eta'_{i,j} \,]$ can be computed in $d^{2+\varepsilon} O^{\sim}(\log(N))$ bit operations, and the same holds for all $[\, \gamma_{i,j} \,]$ and $[\, \gamma'_{i,j} \,]$. This concludes the cost analysis of this step.

Computing all $x^\star_{i,j}$ and $y^\star_{i,j}$'s. Next, we consider Algorithm $\mathsf{lift\_x\_y}$, which computes all $x^\star_{i,j}$ and $y^\star_{i,j}$ in $\mathbb{F}_{i,j}$. This stage of the algorithm directly uses the formula derived in Lemma 26, and the subsequent Equation (13): for every index (i,j) we compute a (truncated) power series, say $J_{i,j}$ that satisfies (12), from which we deduce the corresponding power series

$$S_{i,j} = \frac{\partial^{a_{i,j}} K_{i,j}}{\partial Y^{a_{i,j}}}(\alpha_{i,j} + \xi, J_{i,j}) \in \mathbb{F}_{i,j}[[\xi]].$$

We only need $S_{i,j}$ modulo $\xi^{n_{i,j}+1}$, which in turn means that we need $J_{i,j}$ at the same precision.

Algorithm 8: $\mathsf{lift\_x\_y}(F_t, G_t, \Sigma)$

Input: polynomials $F_t, G_t \bmod N^2$, the σ-decomposition $\Sigma \bmod N$
Output: $[\,(x^\star_{i,j}, y^\star_{i,j}\,]_{1\le i\le s, j\in D_i}$

1 $I = [\,(i,j) \mid 1 \le i \le s,\ j \in D_i]$
2 $I_F = [\,(i,j) \mid 1 \le i \le s,\ j \in D_i$ and $\mathtt{H}_i = \text{"}\mathtt{F}\text{"}\,]$
3 $I_G = [\,(i,j) \mid 1 \le i \le s,\ j \in D_i$ and $\mathtt{H}_i = \text{"}\mathtt{G}\text{"}\,]$
4 $[\,v_{i,j}\,]_{(i,j)\in I} = \mathsf{lift\_y}(F_t, G_t, \Sigma)$
5 $[\,J_{i,j}\,]_{(i,j)\in I} = \mathsf{compute\_J}(F_t, G_t, [\,(c_{i,j}, v_{i,j}, m_i, \mathtt{H}_i, n_{i,j}+1)\,]_{(i,j)\in I})$
6 $\eta = [\,\eta_{i,j,\alpha}\,]_{(i,j)\in I,\alpha\in[0,\dots,m_i]} = [\,\frac{\partial^\alpha F_t}{\partial Y^\alpha}(X+\xi, J_{i,j}) \bmod \langle c_{i,j}(X), \xi^{n_{i,j}+1}\rangle\,]_{(i,j)\in I,\alpha\in[0,\dots,m_i]}$
7 $\gamma = [\,\gamma_{i,j,\alpha}\,]_{(i,j)\in I,\alpha\in[0,\dots,m_i]} = [\,\frac{\partial^\alpha G_t}{\partial Y^\alpha}(X+\xi, J_{i,j}) \bmod \langle c_{i,j}(X), \xi^{n_{i,j}+1}\rangle\,]_{(i,j)\in I,\alpha\in[0,\dots,m_i]}$
8 $L = [\,]_{(i,j)\in I}$
9 **for** (i,j) in I **do**
10 $\quad S_{i,j} = \mathsf{select}(\mathtt{H}_i, \mathtt{K}_{i,j}, a_{i,j}, \eta, \gamma)$
11 $\quad x^\star_{i,j} = \alpha_{i,j} - \frac{1}{n_{i,j}} \frac{\mathsf{cf}\left(S_{i,j}, \xi^{n_{i,j}-1}\right)}{\mathsf{cf}\left(S_{i,j}, \xi^{n_{i,j}}\right)}$
12 $\quad y^\star_{i,j} = \mathsf{cf}(J_{i,j}, \xi^0) - (\alpha_{i,j} - x^\star_{i,j})\mathsf{cf}(J_{i,j}, \xi^1)$
13 $\quad$ append $(x^\star_{i,j}, y^\star_{i,j})$ to L
14 **end**
15 **return** L

The power series $J_{i,j}$ are computed at Step 5 using Algorithm compute_J, using the output $v_{i,j}$ of the previous step as a starting value. For Algorithm compute_J, the bound in Lemma 18 shows that we are under the assumptions of Lemma 21, over the base ring $\mathbb{A} = \mathbb{Z}/N\mathbb{Z}$, so we deduce that Step 5 can be executed using $d^{2+\varepsilon}O^{\tilde{}}(\log(N))$ bit operations.

Steps 6 and 7 are done using Algorithm normal_forms. Once more, Lemma 18 shows that we are under the assumptions of Proposition 3, so these steps take $d^{2+\varepsilon}O^{\tilde{}}(\log(N))$ bit operations. Subroutine select then extracts the power series

$$S_{i,j} = \frac{\partial^{a_{i,j}} K_{i,j}}{\partial Y^{a_{i,j}}}(\alpha_{i,j} + \xi, J_{i,j}) \in \mathbb{E}_{i,j}[[\xi]]$$

from the vectors η and γ, using $\mathtt{H}_i$, $\mathtt{K}_{i,j}$, and $a_{i,j}$ as indices to find the proper entry. Finally, the updates necessary to compute all $x^\star_{i,j}$ and $y^\star_{i,j}$ take a total of $O^{\tilde{}}(d^2\log(N))$ bit operations, just like inversions in Algorithm lift_y did. To summarize, the cost of this step is $d^{2+\varepsilon}O^{\tilde{}}(\log(N))$ bit operations.

Computing all $c'_{i,j}$'s and $t'_{i,j}$'s. Recall that we started from $c_{i,j} = C_{i,j} \bmod N$ and $t_{i,j} = T_{i,j} \bmod N$. At this stage, for any index (i,j), we have found the root $(x^\star_{i,j}, y^\star_{i,j})$ with coordinates in $\mathbb{E}_{i,j} = \mathbb{A}[X]/\langle c_{i,j}\rangle$. Our goal is to recover $c'_{i,j} = C_{i,j} \bmod N^2$, together with $t'_{i,j} = T_{i,j} \bmod N^2$.

Explicit formulas exist for this conversion, see for instance [21, Section 6]. In our case, we will write $\delta_{i,j} = x^\star_{i,j} - \alpha_{i,j} \in \mathbb{E}_{i,j}$, and we let $\Delta_{i,j}$ be its canonical preimage in $\mathbb{A}[X]$; similarly,

we write $\lambda_{i,j} = y^\star_{i,j} - \beta_{i,j} \in \mathbb{E}_{i,j}$, write $\Lambda_{i,j} \in \mathbb{A}[X]$ for its canonical preimage. Then, from [21, Section 6], we deduce

$$c'_{i,j} = c_{i,j} - \left(\Delta_{i,j}\frac{dc_{i,j}}{dX} \bmod c_{i,j}\right) \quad \text{and} \quad t'_{i,j} = t_{i,j} + \Lambda_{i,j} - \left(\Delta_{i,j}\frac{dt_{i,j}}{dX} \bmod c_{i,j}\right).$$

In particular, the new polynomials $c'_{i,j}$ and $t'_{i,j}$ can now all be computed for a total of $O^\sim(d^2\log(N))$ bit operations. Summing all costs seen so far, we conclude the proof of Proposition 6.

8.6 Concluding the proof of Theorem 1

We can finally finish the cost analysis of our algorithm. Since t has been chosen with length $O(\log(d))$, Corollary 1 shows that the output $\mathsf{RUR}(F_t, G_t)$ has length $O^\sim(dh + d^2)$. Let us then review the cost of the steps in our algorithm:

- Choosing t_0 and p_0 (Subsection 8.1) involves only computing integers which were shown to be efficiently computable. Precisely, we can compute the integer $H_0(\mathcal{P}, d, h)$ in $\log(\mathcal{P}dh)^{O(1)}$ bit operations; then, we also need to compute $\Delta_0(\mathcal{P}, d, h)$, which takes an extra $O^\sim(\mathcal{P}\log(dh))$ bit operations.

- Computations modulo p_0 in Subsection 8.2 involve $O^\sim(\mathcal{P} + d^2h + d^3\log(p_0))$ bit operations (Lemma 28). The prime p_0 is in $2^{O(\mathcal{P})}(dh)^{O(1)}$, so $\log(p_0)$ is $O^\sim(\mathcal{P} + \log(dh))$. As a result, the cost of this step is $O^\sim(d^2h + d^3\mathcal{P})$ bit operations.

- Finding t and p (Subsection 8.3) use $d^{2+\varepsilon}O^\sim(\mathcal{P}\log(p_0))$ bit operations to find t; since $\log(p_0)$ is $O^\sim(\mathcal{P}+\log(dh))$, this is $d^{2+\varepsilon}O^\sim(\mathcal{P}^2+\mathcal{P}\log(dh))$. We also have to compute the bounds Δ_4 and $\Delta'_4 = 2^{\mathcal{P}+3}\Delta_4$; this involves $\log(dh)^{O(1)} + O^\sim(\mathcal{P}\log(dh))$ bit operations.

- Computations modulo p in Subsection 8.4 involve $O^\sim(d^2h+d^3\log(p))$ bit operations for Lemma 31 and $d^{3+\varepsilon}O^\sim(\log(p))$ for Lemma 32. We know that $\log(p)$ is $O^\sim(\mathcal{P}+\log(dh))$, so this is $O^\sim(d^2h + d^{3+\varepsilon}\mathcal{P})$ bit operations.

- We run lifting steps (Subsection 8.5) until the precision N goes beyond twice the bound $O^\sim(dh+d^2)$ on the length of the coefficients of the output $(P, R) = \mathsf{RUR}(F_t, G_t)$. The previous section shows that the cost of this lifting is $d^{2+\varepsilon}O^\sim(dh + d^2)$, or simply $d^{3+\varepsilon}O^\sim(h + d)$.

- Using Chinese remaindering, we can then deduce from $\Sigma \bmod N$ the polynomials $(P \bmod N, S \bmod N) = \mathsf{SL}(F_t, G_t) \bmod N$, using $O^\sim(d^2\log(N))$ bit operations, which is $O^\sim(d^3h + d^4)$. From this, we compute $R = P'S \bmod P$ modulo N, and we apply rational reconstruction to all coefficients. This takes again time $O^\sim(d^3h + d^4)$.

Summing all costs, we obtain a total of $d^{2+\varepsilon}O^\sim(d^2 + dh + d\mathcal{P} + \mathcal{P}^2)$ bit operations, which proves Theorem 1.

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.

[2] M. E. Alonso, E. Becker, M.-F. Roy, and T. Wörmann. Zeroes, multiplicities and idempotents for zerodimensional systems. In *MEGA'94*, volume 142 of *Progress in Mathematics*, pages 1–15. Birkhäuser, 1996.

[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.

[4] F. Boulier, F. Lemaire, and M. Moreno Maza. Pardi! In *ISSAC'01*, pages 38–47. ACM, 2001.

[5] Y. Bouzidi, S. Lazard, G. Moroz, M. Pouget, and F. Rouillier. Improved algorithm for computing separating linear forms for bivariate systems. In *ISSAC'14*, pages 75–82. ACM, 2014.

[6] Y. Bouzidi, S. Lazard, M. Pouget, and F. Rouillier. Rational univariate representations of bivariate systems and applications. In *ISSAC'13*, pages 109–116. ACM, 2013.

[7] Y. Bouzidi, S. Lazard, M. Pouget, and F. Rouillier. Separating linear forms for bivariate systems. In *ISSAC'13*, pages 117–124. ACM, 2013.

[8] R. P. Brent and H. T. Kung. Fast algorithms for manipulating formal power series. *J. ACM*, 25(4):581–595, 1978.

[9] D. A. Cox, J. B. Little, and D. O'Shea. *Using Algebraic Geometry*, volume 185 of *Graduate Texts in Mathematics*. Springer, New-York, 1998.

[10] X. Dahan, A. Kadri, and É. Schost. Bit-size estimates for triangular sets in positive dimension. *Journal of Complexity*, 28(1):109–135, 2012.

[11] X. Dahan, M. Moreno Maza, É. Schost, W. Wu, and Y. Xie. Lifting techniques for triangular decompositions. In *ISSAC'05*, pages 108–115. ACM, 2005.

[12] X. Dahan and É. Schost. Sharp estimates for triangular sets. In *ISSAC'04*, pages 103–110. ACM, 2004.

[13] B. H. Dayton and Z. Zeng. Computing the multiplicity structure in solving polynomial systems. In *ISSAC'05*, pages 116–123. ACM, 2005.

[14] 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.

[15] P. Emeliyanenko and M. Sagraloff. On the complexity of solving a bivariate polynomial system. In *ISSAC'12*, pages 154–161. ACM, 2012.

[16] I. Z. Emiris and E. P. Tsigaridas. Real solving of bivariate polynomial systems. In *CASC*, pages 150–161. Springer, 2005.

[17] J. von zur Gathen and J. Gerhard. Fast algorithms for Taylor shifts and certain difference equations. In *ISSAC'97*, pages 40–47. ACM, 1997.

[18] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, Cambridge, second edition, 2003.

[19] P. Gianni and T. Mora. Algebraic solution of systems of polynomial equations using Groebner bases. In *Applied algebra, algebraic algorithms and error-correcting codes*, volume 356 of *Lecture Notes in Computer Science*, pages 247–257. Springer, 1989.

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

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

[22] 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.

[23] K. S. Kedlaya and C. Umans. Fast polynomial factorization and modular composition. *SIAM J. Computing*, 40(6):1767–1802, 2011.

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

[25] T. Krick, L. M. Pardo, and M. Sombra. Sharp estimates for the arithmetic Nullstellensatz. *Duke Math. J.*, 109:521–598, 2001.

[26] L. Kronecker. Grundzüge einer arithmetischen Theorie der algebraischen Grössen. *J. reine angew. Math.*, 92:1–122, 1882.

[27] J. C. Lagarias and H. Mehta. Products of binomial coefficients and unreduced farey fractions, 2014.

[28] F. Le Gall. Powers of tensors and fast matrix multiplication. In *ISSAC'14*, pages 296–303. ACM, 2014.

[29] 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.

[30] G. Lecerf. Quadratic Newton iteration for systems with multiplicity. *Found. Comp. Math.*, 2:247–293, 2002.

[31] A. Leykin, J. Verschelde, and A. Zhao. Newton's method with deflation for isolated singularities of polynomial systems. *Theor. Comput. Sci.*, 359(1-3):111–122, 2006.

[32] A. Leykin, J. Verschelde, and A. Zhao. Higher-order deflation for polynomial systems with isolated singular solutions. In *Algorithms in Algebraic Geometry*, volume 146 of *The IMA Volumes in Mathematics and its Applications*, pages 79–97. 2008.

[33] F. S. Macaulay. *The Algebraic Theory of Modular Systems*. Cambridge University Press, 1916.

[34] M. G. Marinari, T. Mora, , and H. M. Möller. Groebner duality and multiplicities in polynomial system solving. In *ISSAC'95*, pages 167–179. ACM, 1995.

[35] B. Mourrain. Isolated points, duality and residues. *J. of Pure and Applied Algebra*, 117–118:469–493, 1997.

[36] T. Ojika, S. Watanabe, and T. Mitsui. Deflation algorithm for multiple roots of a system of nonlinear equations. *Math. An. and Appls.*, 96(2):463–479, 1983.

[37] C. Pascal and É. Schost. Change of order for bivariate triangular sets. In *ISSAC'06*, pages 277–284. ACM, 2006.

[38] S. R. Pope and A. Szanto. Nearest multivariate system with given root multiplicities. *J. Symb. Comput.*, 44(6):606–625, 2009.

[39] A. Poteaux and É. Schost. Modular composition modulo triangular sets and applications. *Computational Complexity*, 22(3):463–516, 2013.

[40] A. Poteaux and É. Schost. On the complexity of computing with zero-dimensional triangular sets. *Journal of Symbolic Computation*, 50(0):110 – 138, 2013.

[41] D. Reischert. Asymptotically fast computation of subresultants. In *ISSAC'97*, pages 233–240. ACM, 1997.

[42] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Appl. Algebra Engrg. Comm. Comput.*, 9(5):433–461, 1999.

[43] F. Rouillier. On solving systems of bivariate polynomials. In *ICMS*, volume 6327 of *Lecture Notes in Computer Science*, pages 100–104. Springer, 2010.

[44] A. Schönhage. The fundamental theorem of algebra in terms of computational complexity. Technical report, Univ. Tübingen, 1982.

[45] É. Schost. Computing parametric geometric resolutions. *Appl. Algebra Engrg. Comm. Comput.*, 13(5):349–393, 2003.

[46] X. Wu and L. Zhi. Computing the multiplicity structure from geometric involutive form. In *ISSAC'08*, pages 325–332. ACM, 2008.