Separating linear forms for bivariate systems

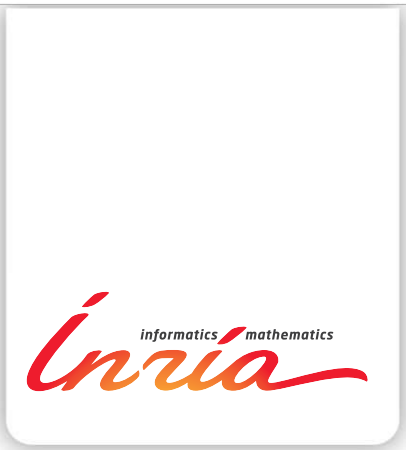
Yacine Bouzidi, Sylvain Lazard, Marc Pouget, Fabrice Rouillier

Separating linear forms for bivariate systems

Yacine Bouzidi, Sylvain Lazard, Marc Pouget, Fabrice Rouillier

RESEARCH REPORT
N° 8261
March 2013
Project-Team Vegas

ISSN 0249-6399 ISRN INRIA/RR--8261--FR+ENG



Separating linear forms for bivariate systems

Yacine Bouzidi*, Sylvain Lazard*, Marc Pouget*, Fabrice Rouillier†

Project-Team Vegas

Research Report n° 8261 — March 2013 — 19 pages

Abstract: We present an algorithm for computing a separating linear form of a system of bivariate polynomials with integer coefficients, that is a linear combination of the variables that takes different values when evaluated at distinct (complex) solutions of the system. In other words, a separating linear form defines a shear of the coordinate system that sends the algebraic system in generic position, in the sense that no two distinct solutions are vertically aligned. The computation of such linear forms is at the core of most algorithms that solve algebraic systems by computing rational parameterizations of the solutions and, moreover, the computation a separating linear form is the bottleneck of these algorithms, in terms of worst-case bit complexity.
Given two bivariate polynomials of total degree at most d with integer coefficients of bitsize at most τ, our algorithm computes a separating linear form in $\widetilde{O}_B(d^8 + d^7\tau)$ bit operations in the worst case, where the previously known best bit complexity for this problem was $\widetilde{O}_B(d^{10} + d^9\tau)$ (where $\widetilde{O}$ refers to the complexity where polylogarithmic factors are omitted and O_B refers to the bit complexity).

Key-words: computer algebra, polynomial system solving, linear separating form

* INRIA Nancy Grand Est, LORIA laboratory, Nancy, France. Firstname.Name@inria.fr
† INRIA Paris-Rocquencourt and IMJ (Institut de Mathématiques de Jussieu, Université Paris 6, CNRS), Paris, France. Firstname.Name@inria.fr

RESEARCH CENTRE
NANCY – GRAND EST

615 rue du Jardin Botanique
CS20101
54603 Villers-lès-Nancy Cedex

Forme linéaire séparante de systèmes bivariés

Résumé : Nous présentons un algorithme pour calculer une forme linéaire séparante d'un système de polynômes à deux variables à coefficients entiers, c'est-à-dire une combinaison linéaire des variables qui prend des valeurs différentes quand elle est évaluée en des solutions (complexes) distinctes du système. En d'autres termes, une forme linéaire séparante définit un changement de coordonnées qui met le système algébrique en position générique, au sens où deux solutions distinctes ne sont jamais verticalement alignées. Le calcul de ces formes linéaires est au coeur de la plupart des algorithmes qui permettent de résoudre des systèmes algébriques au moyen de paramétrisations rationnelles des solutions et, de plus, le calcul d'une forme linéaire séparante domine la complexité binaire de ces algorithmes.

Etant donnés deux polynômes à deux variables de degré total au plus d avec des coefficients entiers de taille binaire au plus τ, notre algorithme calcule une forme linéaire séparante en $\widetilde{O}_B(d^8 + d^7\tau)$ opérations binaires dans le pire des cas, améliorant la meilleure complexité connue pour ce problème d'un facteur d^2 (où $\widetilde{O}$ se réfère à la complexité où les facteurs polylogarithmiques sont omis et O_B se réfère à la complexité binaire).

Mots-clés : calcul formel, résolution de systèmes polynomiaux, forme linéaire séparante

1 Introduction

One approach, that can be traced back to Kronecker, to solve a system of polynomials with a finite number of solutions is to compute a rational parameterization of its solutions. Such a representation of the (complex) solutions of a system is given by a set of univariate polynomials and associated rational one-to-one mappings that send the roots of the univariate polynomials to the solutions of the system. Such parameterizations enable to reduce computations on the system to computations with univariate polynomials and thus ease, for instance, the isolation of the solutions or the evaluation of other polynomials at the solutions.

The computation of such parameterizations has been a focus of interest for a long time; see for example [ABRW96, GVEK96, Rou99, GLS01, BSS03, DET09] and references therein. Most algorithms first shear the coordinate system, with a linear change of variables, so that the input algebraic system is in generic position, that is such that no two solutions are vertically aligned. These algorithms thus need a *linear separating form*, that is a linear combination of the coordinates that takes different values when evaluated at different solutions of the system. Since a random linear form is separating with probability one, probabilist Monte-Carlo algorithms can overlook this issue. However, for deterministic algorithms, computing a linear separating form is critical, especially because this is, surprisingly, the current bottleneck for bivariate systems, as discussed below.

We restrict our attention to systems of two bivariate polynomials of total degree bounded by d with integer coefficients of bitsize bounded by τ. For such systems, the approach with best known worst-case bit complexity for computing a rational parameterization was first introduced by Gonzalez-Vega and El Kahoui [GVEK96] (see also [GVN02]): their initial analysis of $\widetilde{O}_B(d^{16} + d^{14}\tau^2)$ was improved by Diochnos et al. [DET09, Lemma 16 & Theorem 19][1] to (i) $\widetilde{O}_B(d^{10} + d^9\tau)$ for computing a separating linear form and then (ii) $\widetilde{O}_B(d^7 + d^6\tau)$ for computing a parameterization. Computing a separating linear form is thus the bottleneck of the computation of the rational parameterization. This is still true even when considering the additional phase of computing isolating boxes of the solutions (from the rational parameterization), which state-of-the-art complexity is in $\widetilde{O}_B(d^8 + d^7\tau)$ [BLPR13, Proposition 19].

Main results. Our main contribution is a new deterministic algorithm of worst-case bit complexity $\widetilde{O}_B(d^8 + d^7\tau)$ for computing a separating linear form of a system of two bivariate polynomials of total degree at most d and integer coefficients of bitsize at most τ (Theorem 17). This decreases by a factor d^2 the best known complexity for this problem.

As a direct consequence, using our algorithm for computing a separating linear form directly yields a rational parameterization within the same overall complexity as our algorithm, both in the approach of Gonzalez-Vega et al. [GVEK96, DET09] and in that of Bouzidi et al. [BLPR13] for computing the alternative rational parameterization as defined in [Rou99]. As a byproduct, we obtain an algorithm for computing the number of (complex) distinct solutions of such systems within the same complexity, i.e. $\widetilde{O}_B(d^8 + d^7\tau)$.

[1] The overall bit complexity stated in [DET09, Theorem 19] is $\widetilde{O}_B(d^{12} + d^{10}\tau^2)$ because it includes the isolation of the solutions of the system. Note that this complexity trivially decreases to $\widetilde{O}_B(d^{10} + d^9\tau)$ by the recent result of Sagraloff [Sag12] which improves the complexity of isolating the real roots of a univariate polynomial. Note also that Diochnos et al. [DET09] present two algorithms, the M_RUR and G_RUR algorithms, both with bit complexity $\widetilde{O}_B(d^{12} + d^{10}\tau^2)$. However, this complexity is worst case only for the M_RUR algorithm. As pointed out by Emeliyanenko and Sagraloff [ES12], the G_RUR algorithm uses a modular gcd algorithm over an extension field whose considered bit complexity is expected.

2 Overview and organization

Let P and Q be two bivariate polynomials of total degree bounded by d and integer coefficients of maximum bitsize τ. Let $I = \langle P, Q\rangle$ be the ideal they define and suppose that I is zero-dimensional. The goal is to find a linear form $T = X + aY$, with $a \in \mathbb{Z}$, that separates the solutions of I.

We first outline a classical algorithm which is essentially the same as those proposed, for instance, in [DET09, Lemma 16] and [KS12, Theorem 24][2] and whose complexity, in $\widetilde{O}_B(d^{10} + d^9\tau)$, is the best known so far for this problem. This algorithm serves two purposes: it gives some insight on the more involved $\widetilde{O}_B(d^8 + d^7\tau)$-time algorithm that follows and it will be used in that algorithm but over $\mathbb{Z}/\mu\mathbb{Z}$ instead of $\mathbb{Z}$.

Known $\widetilde{\mathbf{O}}_\mathbf{B}(\mathbf{d^{10}} + \mathbf{d^9}\tau)$-time algorithm for computing a separating linear form. The idea is to work with a "generic" linear form $T = X + SY$, where S is an indeterminate, and find conditions such that the specialization of S by an integer a gives a separating form. We thus consider $P(T - SY, Y)$ and $Q(T - SY, Y)$, the "generic" sheared polynomials associated to P and Q, and $R(T, S)$ their resultant with respect to Y. This polynomial has been extensively used and defined in several context; see for instance the related u-resultant [VdW30].

It is known that, in a set $\mathcal{S}$ of d^4 integers, there exists at least one integer a such that $X + aY$ is a separating form for I since I has at most d^2 solutions which define at most $\binom{d^2}{2}$ directions in which two solutions are aligned. Hence, a separating form can be found by computing, for every a in $\mathcal{S}$, the degree of the squarefree part of $R(T, a)$ and by choosing one a for which this degree is maximum. Indeed, for any (possibly non-separating) linear form $X + aY$, the number of distinct roots of $R(T, a)$, which is the degree of its squarefree part, is always smaller than or equal to the number of distinct solutions of I, and equality is attained when the linear form $X + aY$ is separating (Lemma 8). The complexity of this algorithm is in $\widetilde{O}_B(d^{10} + d^9\tau)$ because, for d^4 values of a, the polynomial $R(T, a)$ can be shown to be of degree $O(d^2)$ and bitsize $\widetilde{O}(d^2 + d\tau)$, and its squarefree part can be computed in $\widetilde{O}_B(d^6 + d^5\tau)$ time.

$\widetilde{\mathbf{O}}_\mathbf{B}(\mathbf{d^8} + \mathbf{d^7}\tau)$-time algorithm for computing a separating linear form. To reduce the complexity of the search for a separating form, one can first consider to perform naively the above algorithm on the system $I_\mu = \langle P \bmod \mu, Q \bmod \mu\rangle$ in $\mathbb{Z}_\mu = \mathbb{Z}/\mu\mathbb{Z}$, where μ is a prime number upper bounded by some polynomial in d and τ (so that the bit complexity of arithmetic operations in $\mathbb{Z}_\mu$ is polylogarithmic in d and τ). The resultant $R_\mu(T, S)$ of $P(X - SY, Y) \bmod \mu$ and $Q(X - SY, Y) \bmod \mu$ with respect to Y can be computed in $\widetilde{O}_B(d^6 + d^5\tau)$ bit operations and, since its degree is at most $2d^2$ in each variable, evaluating it at $S = a$ in $\mathbb{Z}_\mu$ can be easily done in $\widetilde{O}_B(d^4)$ bit operations. Then, the computation of its squarefree part does not suffer anymore from the coefficient growth, and it becomes softly linear in its degree, that is $\widetilde{O}_B(d^2)$. Considering d^4 choices of a, we get an algorithm that computes a separating form for I_μ in $\widetilde{O}_B(d^8)$ time in $\mathbb{Z}_\mu$. However, a serious problem remains, that is to ensure that a separating form for I_μ is also a separating form for I. This issue requires to develop a more subtle algorithm.

We first show, in Section 4.1, a critical property (Proposition 7) which states that a separating linear form over $\mathbb{Z}_\mu$ is also separating over $\mathbb{Z}$ when μ is a *lucky* prime number, which is, essentially, a prime such that the number of solutions of $\langle P, Q\rangle$ is the same over $\mathbb{Z}$ and over $\mathbb{Z}_\mu$. We then show in Sections 4.2 to 4.4 how to compute such a lucky prime number. We do that by first

[2]The stated complexity of [KS12, Theorem 24] is $\widetilde{O}_B(d^9\tau)$, but it seems the fact that the sheared polynomials have bitsize in $\widetilde{O}(d + \tau)$ (see Lemma 5) instead of $\widetilde{O}(\tau)$ has been overlooked in their proof.

proving in Section 4.2 that, under mild conditions on μ, the number of solutions over $\mathbb{Z}_\mu$ is always less than or equal to the number of solutions over $\mathbb{Z}$ (Proposition 10) and then by computing a bound on the number of unlucky primes (Proposition 11). Computing a lucky prime can then be done by choosing a μ that maximizes the number of solutions over $\mathbb{Z}_\mu$ among a set of primes of cardinality $\widetilde{\Theta}(d^4 + d^3\tau)$. For that purpose, we present in Section 4.3 a new algorithm, of independent interest, for computing in $\widetilde{O}(d^4)$ arithmetic operations the number of distinct solutions of the system I_μ in $\mathbb{Z}_\mu$; this algorithm is based on a classical triangular decomposition. This yields, in Section 4.4, a $\widetilde{O}_B(d^8 + d^7\tau)$-time algorithm for computing a lucky prime μ in $\widetilde{O}(d^4 + d^3\tau)$. Now, μ is fixed, and we can apply the algorithm outlined above for computing a separating form for I_μ in $\mathbb{Z}_\mu$ in $\widetilde{O}_B(d^8)$ time (Section 4.5). This form, which is also separating for I, is thus obtained with a total bit complexity of $\widetilde{O}_B(d^8 + d^7\tau)$ (Theorem 17).

3 Notation and preliminaries

We introduce notation and recall classical material about subresultant sequences.

The bitsize of an integer p is the number of bits needed to represent it, that is $\lfloor \log p \rfloor + 1$ (log refers to the logarithm in base 2). For rational numbers, we refer to the bitsize as to the maximum bitsize of its numerator and denominator. The bitsize of a polynomial with integer or rational coefficients is the *maximum* bitsize of its coefficients. As mentioned earlier, O_B refers to the bit complexity and $\widetilde{O}$ and $\widetilde{O}_B$ refer to complexities where polylogarithmic factors are omitted.

In the following, μ is a prime number and we denote by $\mathbb{Z}_\mu$ the quotient $\mathbb{Z}/\mu\mathbb{Z}$. We denote by ϕ_μ: $\mathbb{Z} \to \mathbb{Z}_\mu$ the reduction modulo μ, and extend this definition to the reduction of polynomials with integer coefficients. We denote by $\mathbb{D}$ a unique factorization domain, typically $\mathbb{Z}[X, Y]$, $\mathbb{Z}[X]$, $\mathbb{Z}_\mu[X]$, $\mathbb{Z}$ or $\mathbb{Z}_\mu$. We also denote by $\mathbb{F}$ a field, typically $\mathbb{Q}$, $\mathbb{C}$, or $\mathbb{Z}_\mu$.

For any polynomial $P \in \mathbb{D}[X]$, let $Lc_X(P)$ denote its leading coefficient with respect to the variable X, $d_X(P)$ its degree with respect to X, and $\overline{P}$ its squarefree part. The ideal generated by two polynomials P and Q is denoted $\langle P, Q\rangle$, and the affine variety of an ideal I is denoted by $V(I)$; in other words, $V(I)$ is the set of distinct solutions of the system $\{P, Q\}$. The solutions are always considered in the algebraic closure of $\mathbb{D}$ and the number of distinct solutions is denoted by $\#V(I)$. For a point $\sigma \in V(I)$, $\mu_I(\sigma)$ denotes the multiplicity of σ in I. For simplicity, we refer indifferently to the ideal $\langle P, Q\rangle$ and to the system $\{P, Q\}$.

We finally introduce the following notation which are extensively used throughout the paper. Given the two input polynomials P and Q, we consider the "generic" change of variables $X = T - SY$, and define the "sheared" polynomials $P(T - SY, Y)$, $Q(T - SY, Y)$, and their resultant with respect to Y,

$$R(T, S) = Res_Y(P(T - SY, Y), Q(T - SY, Y)). \tag{1}$$

The complexity bounds on the degree, bitsize and computation of these polynomials are analyzed at the end of this section in Lemma 5. Let $L_R(S)$ be the leading coefficient of $R(T, S)$ seen as a polynomial in T. Let $L_P(S)$ and $L_Q(S)$ be the leading coefficients of $P(T - SY, Y)$ and $Q(T - SY, Y)$, seen as polynomials in Y; it is straightforward that these leading coefficients do not depend on T. In other words:

$$L_P(S) = Lc_Y(P(T - SY, Y)), \qquad L_Q(S) = Lc_Y(Q(T - SY, Y)), \qquad L_R(S) = Lc_T(R(T, S)). \tag{2}$$

3.1 Subresultant sequences

We recall here the definition of subresultant sequences and some related properties. Note that we only use subresultants in Section 4.3.1 in which we recall a classical triangular decomposition algorithm.

We first recall the concept of *polynomial determinant* of a matrix which is used in the definition of subresultants. Let M be an $m \times n$ matrix with $m \leqslant n$ and M_i be the square submatrix of M consisting of the first $m-1$ columns and the i-th column of M, for $i = m, \ldots, n$. The *polynomial determinant* of M is the polynomial defined as $\det(M_m)Y^{n-m} + \det(M_{m+1})Y^{n-(m+1)} + \ldots + \det(M_n)$.

Let $P = \sum_{i=0}^{p} a_i Y^i$ and $Q = \sum_{i=0}^{q} b_i Y^i$ be two polynomials in $\mathbb{D}[Y]$ and assume without loss of generality that $p \geqslant q$. The Sylvester matrix of P and Q, $Sylv(P,Q)$ is the $(p+q)$-square matrix whose rows are $Y^{q-1}P, \ldots, P, Y^{p-1}Q, \ldots, Q$ considered as vectors in the basis $Y^{p+q-1}, \ldots, Y, 1$.

$$Sylv(P,Q) = \overbrace{\begin{pmatrix} a_p & a_{p-1} & \cdots & \cdots & a_0 & & \\ & a_p & a_{p-1} & \cdots & \cdots & a_0 & \\ & & \ddots & & & & \ddots \\ & & & a_p & a_{p-1} & \cdots & \cdots & a_0 \\ b_q & b_{q-1} & \cdots & & b_0 & & & \\ & b_q & b_{q-1} & \cdots & & b_0 & & \\ & & \ddots & & & & \ddots & \\ & & & \ddots & & & & \ddots \\ & & & & b_q & b_{q-1} & \ldots & b_0 \end{pmatrix}}^{\text{p+q columns}} \left.\begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix}\right\} \begin{matrix} \text{q rows} \\ \\ \\ \\ \text{p rows} \\ \\ \\ \\ \end{matrix}$$

Definition 1. *([EK03, §3]). For $i = 0, \ldots, \min(q, p-1)$, let $Sylv_i(P,Q)$ be the $(p+q-2i) \times (p+q-i)$ matrix obtained from $Sylv(P,Q)$ by deleting the i last rows of the coefficients of P, the i last rows of the coefficients of Q, and the i last columns.*

For $i = 0, \ldots, \min(q, p-1)$, the i-th polynomial subresultant of P and Q, denoted by $Sres_{Y,i}(P,Q)$ is the polynomial determinant of $Sylv_i(P,Q)$. When $q = p$, the q-th polynomial subresultant of P and Q is $b_q^{-1}Q$.[3]

$Sres_{Y,i}(P,Q)$ has degree at most i in Y, and the coefficient of its monomial of degree i in Y, denoted by $sres_{Y,i}(P,Q)$, is called the i-th *principal subresultant coefficient*. Note that $Sres_{Y,0}(P,Q) = sres_{Y,0}(P,Q)$ is the *resultant* of P and Q with respect to Y, which we also denote by $Res_Y(P,Q)$. Furthermore, the first (with respect to increasing i) nonzero subresultant of $P, Q \in \mathbb{D}[Y]$ is equal to their gcd in $\mathbb{F}_{\mathbb{D}}[Y]$, up to a multiplicative factor in $\mathbb{F}_{\mathbb{D}}$, where $\mathbb{F}_{\mathbb{D}}$ is the fraction field of $\mathbb{D}$ (e.g., if $\mathbb{D} = \mathbb{Z}[X]$, then $\mathbb{F}_{\mathbb{D}} = \mathbb{Q}(X)$, the field of fractions of polynomials in $\mathbb{Q}[X]$); more generally, the subresultants of P and Q are equal to either 0 or to polynomials in the remainder sequence of P and Q in Euclid's algorithm (up to multiplicative factors in $\mathbb{D}$) [BPR06, §8.3.3 & Cor. 8.32].[4]

We state below a fundamental property of subresultants which is instrumental in the triangular decomposition algorithm used in Section 4.3.1. For clarity, we state this property for

[3] It can be observed that, when $p > q$, the q-th subresultant is equal to $b_q^{p-q-1}Q$, however it is not defined when $p = q$. In this case, following El Kahoui, we extend the definition to $b_q^{-1}Q$ assuming that the domain $\mathbb{D}$ is integral, which is the case in this paper. Note that it is important to define the q-th subresultant to be a multiple of Q so that Lemma 2 holds when $Q(\alpha, Y)$ is of degree q and divides $P(\alpha, Y)$ for some α.

[4] For efficiency, the computation of subresultant sequences are usually performed by computing the polynomial remainder sequences using some variants of Euclid algorithm instead of the aforementioned determinants.

bivariate polynomials $P = \sum_{i=0}^{p} a_i Y^i$ and $Q = \sum_{i=0}^{q} b_i Y^i$ in $\mathbb{D}[X, Y]$, with $p \geqslant q$. Note that this property is often stated with a stronger assumption that is that *none* of the leading terms $a_p(\alpha)$ and $b_q(\alpha)$ vanishes. This property is a direct consequence of the specialization property of subresultants and of the gap structure theorem; see for instance [EK03, Lemmas 2.3, 3.1 and Corollary 5.1].

Lemma 2. *For any α such that $a_p(\alpha)$ and $b_q(\alpha)$ do not both vanish, the first $Sres_{Y,k}(P, Q)(\alpha, Y)$ (for k increasing) that does not identically vanish is of degree k and it is the gcd of $P(\alpha, Y)$ and $Q(\alpha, Y)$ (up to a nonzero constant in the fraction field of $\mathbb{D}(\alpha)$).*

3.2 Complexity

We recall complexity results, using fast algorithms, on subresultants and gcd computations. We also analyze complexities related to the computation of the "sheared" polynomials and their resultant.

Lemma 3 ([BPR06, Proposition 8.46] [Rei97, §8, Algorithm 7.3])**.** *Let P and Q in $\mathbb{Z}[X_1, \ldots, X_n][Y]$ of coefficient bitsize τ such that their degrees in Y are bounded by d_Y and their degrees in the other variables are bounded by d.*

- *The coefficients of $Sres_{Y,i}(P, Q)$ have bitsize in $\widetilde{O}(d_Y \tau)$.*
- *The degree in X_j of $Sres_{Y,i}(P, Q)$ is at most $2d(d_Y - i)$.*
- *Any subresultants $Sres_{Y,i}(P, Q)$ can be computed in $\widetilde{O}(d^n d_Y^{n+1})$ arithmetic operations, and $\widetilde{O}_B(d^n d_Y^{n+2} \tau)$ bit operations.*

In the sequel, we often consider the gcd of two univariate polynomials P and Q and the gcd-free part of P with respect to Q, that is, the divisor D of P such that $P = \gcd(P, Q)D$. Note that when $Q = P'$, the latter is the squarefree part $\overline{P}$.

Lemma 4 ([BPR06, Remark 10.19])**.** *Let P and Q in $\mathbb{F}[X]$ of degree at most d. $\gcd(P, Q)$ or the gcd-free part of P with respect to Q can be computed with $\widetilde{O}(d)$ operations in $\mathbb{F}$.*

Lemma 5. *Let P and Q in $\mathbb{Z}[X, Y]$ be of total degree at most d and maximum bitsize τ. The sheared polynomials $P(T - SY, Y)$ and $Q(T - SY, Y)$ can be expanded in $\widetilde{O}_B(d^4 + d^3\tau)$ and their bitsizes are in $\widetilde{O}(d + \tau)$. The resultant $R(T, S)$ can be computed in $\widetilde{O}_B(d^7 + d^6\tau)$ bit operations and $\widetilde{O}(d^5)$ arithmetic operations in $\mathbb{Z}$; its degree is at most $2d^2$ in each variable and its bitsize is in $\widetilde{O}(d^2 + d\tau)$.*

Proof. Writing P as $\sum_{i=0}^{d} p_i(Y) X^i$, expending the substitution of X by $T - SY$ needs the computation of the successive powers $(T - SY)^i$ for i from 1 to d. The binomial formula shows that each polynomial $(T - SY)^i$ is the sum of $i + 1$ monomials, with coefficients of bitsize in $O(i \log i)$. Using the recursion formula $(T - SY)^i = (T - SY)^{i-1}(T - SY)$, given the polynomial $(T - SY)^{i-1}$, the computation of $(T - SY)^i$ requires $2i$ multiplications of coefficients having bitsize in $O(i \log i)$, which can be done in $\widetilde{O}_B(i^2 \log i)$ bit operations. The complexity of computing all the powers is thus in $\widetilde{O}_B(d^3 \log d)$. The second step is to multiply $p_i(Y)$ by $(T - SY)^i$ for $i = 1, \ldots, d$. Each polynomial multiplication can be done with $O(d^2)$ multiplications of integers of bitsize in $O(\tau)$ or in $O(d \log d)$, and thus it can be done in $\widetilde{O}_B(d^2(\tau + d \log d))$ bit operations and yields polynomials of bitsize $O(\tau + d \log d)$. For the d multiplications the total cost is in $\widetilde{O}_B(d^3(\tau + d \log d))$. Consequently the computation of $P(T - SY, Y)$ and $Q(T - SY, Y)$ can be done in $\widetilde{O}_B(d^3(\tau + d))$ bit operations and these polynomials have bitsize in $\widetilde{O}(\tau + d)$. In addition, since $P(T - SY, Y)$ and $Q(T - SY, Y)$ are trivariate polynomials of partial degree in all variables bounded by d, Lemma 3 implies the claims on $R(T, S)$. □

4 Separating linear form

Throughout this section, we assume that the two input polynomials P and Q are coprime in $\mathbb{Z}[X,Y]$, that they define the ideal I, that their maximum total degree d is at least 2 and that their coefficients have maximum bitsize τ. Note that the coprimality of P and Q is implicitly tested during Algorithm 4 because they are coprime if and only if $R(T,S)$ does not identically vanish. By abuse of notation, some complexity $\widetilde{O}_B(d^k)$ may refer to a complexity in which polylogarithmic factors in d *and in* τ are omitted. $I_\mu = \langle P_\mu, Q_\mu\rangle$ denotes the ideal generated by $P_\mu = \phi_\mu(P)$ and $Q_\mu = \phi_\mu(Q)$. Similarly as in Equation (1), we define $R_\mu(T,S)$ as the resultant of $P_\mu(T - SY, Y)$ and $Q_\mu(T - SY, Y)$ with respect to Y, and we define $L_{P_\mu}(S)$ and $L_{Q_\mu}(S)$ similarly as in (2). We refer to the overview in Section 2 for the organization of this section.

4.1 Separating linear form over $\mathbb{Z}_\mu$ versus $\mathbb{Z}$

We first introduce the notion of lucky prime numbers μ which are, roughly speaking, primes μ for which the number of distinct solutions of $\langle P, Q\rangle$ does not change when considering the polynomials modulo μ. We then show the critical property that, if a linear form is separating modulo such a μ, then it is also separating over $\mathbb{Z}$.

Definition 6. *A prime number μ is said to be* **lucky** *for an ideal $I = \langle P, Q\rangle$ if it is larger than $2d^4$ and satisfies*

$$\phi_\mu(L_P(S))\ \phi_\mu(L_Q(S)) \not\equiv 0 \quad \textit{and} \quad \#V(I) = \#V(I_\mu).$$

Proposition 7. *Let μ be a lucky prime for the ideal $I = \langle P, Q\rangle$ and let $a < \mu$ be an integer*[5] *such that $\phi_\mu(L_P(a))\ \phi_\mu(L_Q(a)) \neq 0$. If $X + aY$ separates $V(I_\mu)$, it also separates $V(I)$.*

The key idea of the proof of Proposition 7, as well as Propositions 10 and 11, is to prove the following inequalities (under the hypothesis that various leading terms do not vanish)

$$\#V(I_\mu) \geqslant d_T(\overline{R_\mu(T,a)}) \leqslant d_T(\overline{R(T,a)}) \leqslant \#V(I) \tag{3}$$

and argue that the first (resp. last) one is an equality if $X + aY$ separates $V(I_\mu)$ (resp. $V(I)$). We establish these claims in Lemmas 8 and 9. As mentioned in Section 2, Lemma 8 is the key property in the classical algorithm for computing a separating form for I, which algorithm we will use over $\mathbb{Z}_\mu$ to compute a separating form for I_μ in Section 4.5. For completeness, we outline its proof (see [DET09, Lemma 16] or [BPR06, Proposition 11.23] for details). Recall that P and Q are assumed to be coprime but not P_μ and Q_μ.

Lemma 8. *If $a \in \mathbb{Z}$ is such that $L_P(a)\, L_Q(a) \neq 0$ then $d_T(\overline{R(T,a)}) \leqslant \#V(I)$ and they are equal if and only if $X + aY$ separates $V(I)$. The same holds over $\mathbb{Z}_\mu$, that is for P_μ, Q_μ, R_μ and I_μ, provided P_μ and Q_μ are coprime.*

Proof. Since $L_P(a)\ L_Q(a) \neq 0$, the resultant $R(T,S)$ can be specialized at $S = a$, that is $R(T,a) = Res_Y(P(T - aY, Y), Q(T - aY, Y))$. On the other hand, the sheared polynomials $P(T - aY, Y)$ and $Q(T - aY, Y)$ are coprime (since P and Q are coprime) and since $L_P(a)\ L_Q(a) \neq 0$, they have no common solution at infinity in the Y-direction. Thus the roots of their resultant with respect to Y are the T-coordinates of the (affine) solutions of $I_a = \langle P(T - aY, Y), Q(T - aY, Y)\rangle$ (see for instance [CLO97, §3.6 Proposition 3]). Hence, $d_T(\overline{R(T,a)}) \leqslant \#V(I_a) = \#V(I)$. Moreover, if $X + aY$ separates $V(I)$, $T = X + aY$ takes distinct values for every solution in $V(I)$,

[5] We assume $a < \mu$ for clarity so that the linear form $X + aY$ is "identical" in $\mathbb{Z}$ and in $\mathbb{Z}_\mu$. This hypothesis is however not needed and we actually prove that if $X + \phi_\mu(a)Y$ separates $V(I_\mu)$, then $X + aY$ separates $V(I)$.

and since these values of T are roots of $R(T,a)$, $d_T(\overline{R(T,a)}) \geqslant \#V(I)$ and thus they are equal. Conversely, if $d_T(\overline{R(T,a)}) = \#V(I)$, $R(T,a)$ admits $\#V(I)$ distinct roots $T = X + aY$ which means that $X + aY$ separates all the solutions of $V(I)$. The same argument holds over $\mathbb{Z}_\mu$. □

The following lemma states a rather standard properties. For completeness and readers' convenience, we provide a proof for which we could not find accurate references.

Lemma 9. *Let μ be a prime and a be an integer such that $\phi_\mu(L_P(a))\ \phi_\mu(L_Q(a)) \neq 0$, then $d_T(\overline{R_\mu(T,a)}) \leqslant d_T(\overline{R(T,a)})$.*

Proof. By hypothesis, $\phi_\mu(L_P(S))$ and $\phi_\mu(L_Q(S))$ do not identically vanish, thus we can specialize the resultant R by ϕ_μ, that is $\phi_\mu(R(T,S)) = Res_Y(\phi_\mu(P(T-SY,Y)), \phi_\mu(Q(T-SY,Y)))$ [BPR06, Proposition 4.20]. Hence, $\phi_\mu(R(T,S)) = R_\mu(T,S)$. The evaluation at $S = a$ and the reduction modulo μ commute (in $\mathbb{Z}_\mu$), thus $\phi_\mu(R(T,a)) = R_\mu(T,a)$ in $\mathbb{Z}_\mu[T]$.

We now show that for any polynomial $f \in \mathbb{Z}[X]$ and prime μ, $\deg(\overline{\phi_\mu(f)}) \leqslant \deg(\overline{f})$, which will implie the lemma.

Let $f = c\prod_i f_i^{m_i}$ be the squarefree decomposition of f in $\mathbb{Z}[X]$. Considering its reduction modulo μ, we obtain that $\phi_\mu(f) = \phi_\mu(c)\prod_i \phi_\mu(f_i)^{m_i}$. Hence, $\deg(\overline{\phi_\mu(f)}) \leqslant \sum_i \deg(\phi_\mu(f_i))$. Furthermore, since $\deg(\phi_\mu(f_i)) \leqslant \deg(f_i)$, we have that $\deg(\overline{\phi_\mu(f)}) \leqslant \sum_i \deg(f_i)$. On the other hand, since $f = c\prod_i f_i^{m_i}$ is the squarefree decomposition of f, we have $\deg(\overline{f}) = \sum_i \deg(f_i)$ so $\deg(\overline{\phi_\mu(f)}) \leqslant \deg(\overline{f})$. □

Proof of Proposition 7. If μ is a lucky prime, then by definition $\#V(I) = \#V(I_\mu)$, thus I_μ is zero-dimensional since I is. Thus, by Lemmas 8 and 9, if μ is a lucky prime and a is an integer such that $X + aY$ separates $V(I_\mu)$ and $\phi_\mu(L_P(a))\ \phi_\mu(L_Q(a)) \neq 0$, then

$$\#V(I_\mu) = d_T(\overline{R_\mu(T,a)}) \leqslant d_T(\overline{R(T,a)}) \leqslant \#V(I).$$

Since μ is lucky, $\#V(I_\mu) = \#V(I)$ thus $d_T(\overline{R(T,a)}) = \#V(I)$ and by Lemma 8, $X + aY$ separates $V(I)$. □

4.2 Number of solutions over $\mathbb{Z}_\mu$ versus $\mathbb{Z}$

As shown in Proposition 7, the knowledge of a lucky prime permits to search for separating linear forms over $\mathbb{Z}_\mu$ rather than over $\mathbb{Z}$. We prove here two propositions that are critical for computing a lucky prime, which state that the number of solutions of $I_\mu = \langle P_\mu, Q_\mu\rangle$ is always at most that of $I = \langle P, Q\rangle$ and give a bound on the number of unlucky primes.

Proposition 10. *Let $I = \langle P, Q\rangle$ be a zero-dimensional ideal in $\mathbb{Z}[X,Y]$. If a prime μ is larger than $2d^4$ such that I_μ is zero-dimensional and $\phi_\mu(L_P(S))\ \phi_\mu(L_Q(S)) \not\equiv 0$ then $\#V(I_\mu) \leqslant \#V(I)$.*

Proof. Let μ be a prime that satisfies the hypotheses of the proposition. We also consider an integer $a < \mu$ such that $\phi_\mu(L_P(a))\ \phi_\mu(L_Q(a)) \neq 0$ and such that the linear form $X + aY$ is separating for I_μ. Such an integer exists because (i) $\phi_\mu(L_P(S))$ and $\phi_\mu(L_Q(S))$ are not identically zero by hypothesis and they have degree at most d and, since I_μ is zero dimensional, (ii) I_μ has at most d^2 solutions which define at most $\binom{d^2}{2}$ directions in which two solutions are aligned. Since $2d + \binom{d^2}{2} < 2d^4$ (for $d \geqslant 2$), there exists such an integer $a \leqslant 2d^4 < \mu$. With such an a, we can apply Lemmas 8 and 9 which imply that $\#V(I_\mu) = d_T(\overline{R_\mu(T,a)}) \leqslant d_T(\overline{R(T,a)}) \leqslant \#V(I)$. □

Next, we bound the number of primes that are unlucky for the ideal $\langle P, Q\rangle$.

Proposition 11. *An upper bound on the number of unlucky primes for the ideal $\langle P, Q\rangle$ can be explicitly computed in terms of d and τ, and this bound is in $\widetilde{O}(d^4 + d^3\tau)$.*

Proof. According to Definition 6, a prime μ is unlucky if it is smaller than $2d^4$, if $\phi_\mu(L_P(S))$ $\phi_\mu(L_Q(S)) \not\equiv 0$, or if $\#V(I) \neq \#V(I_\mu)$. In the following, we consider $\mu > 2d^4$. We first determine some conditions on μ that ensure that $\#V(I) = \#V(I_\mu)$, and we then bound the number of μ that do not satisfy these conditions. As we will see, under these conditions, $L_P(S)$ and $L_Q(S)$ do not vanish modulo μ and thus this constraint is redundant.

The first part of the proof is similar in spirit to that of Proposition 10 in which we first fixed a prime μ and then specialized the polynomials at $S = a$ such that the form $X + aY$ was separating for I_μ. Here, we first choose a such that $X + aY$ is separating for I. With some conditions on μ, Lemmas 8 and 9 imply Equation (4) and we determine some more conditions on μ such that the middle inequality of (4) is an equality. We thus get $\#V(I_\mu) \geqslant \#V(I)$ which is the converse of that of Proposition 10 and thus $\#V(I_\mu) = \#V(I)$. In the second part of the proof, we bound the number of μ that violate the conditions we considered.

Prime numbers such that $\#V(I) \neq \#V(I_\mu)$. Let a be such that the form $X + aY$ separates $V(I)$ and $L_P(a)\, L_Q(a)\, L_R(a) \neq 0$.[6] Similarly as in the proof of Proposition 10, since $L_R(S)$ has degree at most $2d^2$ (Lemma 3) and $2d + 2d^2 + \binom{d^2}{2} < 2d^4$ (for $d \geqslant 2$), we can choose $a \leqslant 2d^4$.

We consider any prime $\mu > 2d^4$ such that $\phi_\mu(L_P(a))$ $\phi_\mu(L_Q(a))$ $\phi_\mu(L_R(a)) \neq 0$. By Lemmas 8 and 9, we have

$$\#V(I_\mu) \geqslant d_T(\overline{R_\mu(T,a)}) \leqslant d_T(\overline{R(T,a)}) = \#V(I), \tag{4}$$

since the first inequality trivially holds when I_μ is not zero-dimensional and since $X+aY$ separates $V(I)$.

Now, $d_T(\overline{R(T,a)}) = d_T(R(T,a)) - d_T(\gcd(R(T,a), R'(T,a)))$, and similarly for $R_\mu(T,a)$. The leading coefficient of $R(T,S)$ with respect to T is $L_R(S)$, and since it does not vanish at $S = a$, $L_R(a)$ is the leading coefficient of $R(T,a)$. In addition, since $\phi_\mu(L_P(a))\, \phi_\mu(L_Q(a)) \neq 0$, we can specialize the resultant R by ϕ_μ, thus $\phi_\mu(R(T,a)) = Res_Y(\phi_\mu(P(T-aY,Y)), \phi_\mu(Q(T-aY,Y)))$ [BPR06, Proposition 4.20]. Hence, $\phi_\mu(R(T,a)) = R_\mu(T,a)$ and the hypothesis $\phi_\mu(L_R(a)) \neq 0$ implies that $R_\mu(T,a)$ and $R(T,a)$ have the same degree. It follows that, *if μ is such that the degree of* $\gcd(R(T,a), R'(T,a))$ *does not change when $R(T,a)$ and $R'(T,a)$ are reduced modulo* μ, we have

$$\#V(I_\mu) \geqslant d_T(\overline{R_\mu(T,a)}) = d_T(\overline{R(T,a)}) = \#V(I).$$

Since $\phi_\mu(R(T,a)) = R_\mu(T,a)$ and $\phi_\mu(L_R(a)) \neq 0$, the resultant $R_\mu(T,a)$ does not identically vanish and thus I_μ is zero-dimensional. Furthermore, since $\mu > 2d^4$ and $\phi_\mu(L_P(a))\, \phi_\mu(L_Q(a)) \neq 0$, we can apply Proposition 10 which yields that $\#V(I_\mu) \leqslant \#V(I)$ and thus $\#V(I_\mu) = \#V(I)$.

Therefore, the primes μ such that $\#V(I_\mu) \neq \#V(I)$ are among those such that $\mu \leqslant 2d^4$, or $L_P(a)$, $L_Q(a)$ or $L_R(a)$ vanishes modulo μ or such that the degree of $\gcd(R(T,a), R'(T,a))$ changes when $R(T,a)$ and $R'(T,a)$ are reduced modulo μ. Note that if $L_P(a)$ and $L_Q(a)$ do not vanish modulo μ, then $L_P(S)$ and $L_Q(S)$ do not identically vanish modulo μ.

Bounding the number of prime divisors of $L_P(a)$, $L_Q(a)$ or $L_R(a)$. The number of prime divisors of an integer z is bounded by its bitsize. Indeed, its bitsize is $\lfloor \log z \rfloor + 1$ and its factorization into w (possibly identical) prime numbers directly yields that $2^w \leqslant \prod_{i=1}^{w} z_i = z = 2^{\log z} \leqslant 2^{\lfloor \log z \rfloor + 1}$. We can thus bound the number of prime divisors by bounding the bitsize of $L_P(a)$, $L_Q(a)$ and $L_R(a)$. We start by bounding the bitsize of $L_P(S)$, $L_Q(S)$ and $L_R(S)$.

[6] It can be shown that $L_P(a)\, L_Q(a) \neq 0$ implies $L_R(a) \neq 0$ (see for instance [BLPR13, Lemma 11]) but this property does not simplify the proof.

Each coefficient of $P(T-SY,Y)$ has bitsize at most $\tau' = \tau + d\log d + \log(d+1) + 1$. Indeed, $(T-SY)^i$ is a sum of $i+1$ monomials whose coefficients are binomials $\binom{i \leqslant d}{j} < d^d$. The claim follows since each coefficient of $P(T-SY,Y)$ is the sum of at most $d+1$ such binomials, each multiplied by a coefficient of $P(X,Y)$ which has bitsize at most τ. We get the same bound for the coefficients of $Q(T-SY,Y)$ and thus for $L_P(S)$ and $L_Q(S)$ as well. Concerning $L_R(S)$, we have that $R(T,S)$ is the resultant of $P(T-SY,Y)$ and $Q(T-SY,Y)$ thus, by Lemma 3, its coefficients are of bitsize $\widetilde{O}(d\tau')$. In fact, an upper bound can be explicitly computed using, for instance, the bound of [BPR06, Theorem 8.46] which implies that the resultant of two trivariate polynomials of total degree d' and bitsize τ' has bitsize at most $2d'(\tau' + \lfloor \log 2d' \rfloor + 1) + 2(\lfloor \log(2d'^2+1) \rfloor + 1)$, which is in $\widetilde{O}(d^2 + d\tau)$ in our case. Therefore, $L_P(S)$, $L_Q(S)$ and $L_R(S)$ have degree at most $2d^2$ and their bitsizes can be explicitly bounded by a function of d and τ in $\widetilde{O}(d^2 + d\tau)$.

Finally, since $a \leqslant 2d^4$, its bitsize is at most $\sigma = 4\log d + 2$. It is straightforward that the result of an evaluation of a univariate polynomial of degree at most d' and bitsize τ' at an integer value of bitsize σ has bitsize at most $d'\sigma + \tau' + \log(d'+1) + 1$. Here $d' \leqslant 2d^2$ and τ' is in $\widetilde{O}(d^2 + d\tau)$. We thus proved that we can compute an explicit bound, in $\widetilde{O}(d^2 + d\tau)$, on the number of prime divisors of $L_P(a)$, $L_Q(a)$, or $L_R(a)$.

Bounding the number of prime μ such that the degree of $\gcd(R(T,a), R'(T,a))$ *changes when $R(T,a)$ and $R'(T,a)$ are reduced modulo μ.* By [Yap00, Lemma 4.12], given two univariate polynomials in $\mathbb{Z}[X]$ of degree at most d' and bitsize at most τ', the degree of their gcd changes when the polynomials are considered modulo μ on a set of μ whose product is bounded[7] by $(2^{\tau'}\sqrt{d'+1})^{2d'+2}$. As noted above, the number of such primes μ is bounded by the bitsize of this bound, and thus is bounded by $(d'+1)(2\tau' + \log(d'+1)) + 1$. Here $d' \leqslant 2d^2$ and τ' is in $\widetilde{O}(d^2 + d\tau)$ since our explicit bound on the bitsize of $L_R(a)$ holds as well for the bitsize of $R(T,a)$, and, since $R(T,a)$ is of degree at most $2d^2$, the bitsize of $R'(T,a)$ is bounded by that of $R(T,a)$ plus $1 + \log 2d^2$. We thus obtain an explicit bound in $\widetilde{O}(d^4 + d^3\tau)$ on the number of primes μ such that the degree of $\gcd(R(T,a), R'(T,a))$ changes when $R(T,a)$ and $R'(T,a)$ are reduced modulo μ.

The result follows by summing this bound with the bounds we obtained on the number of prime divisors of $L_P(a)$, $L_Q(a)$, or $L_R(a)$, and a bound (e.g. $2d^4$) on the number of primes smaller than $2d^4$. □

4.3 Counting the number of solutions over $\mathbb{Z}_\mu$

For counting the number of (distinct) solutions of $\langle P_\mu, Q_\mu \rangle$, we use a classical algorithm for computing a triangular decomposition of an ideal defined by two bivariate polynomials. We first recall this algorithm, slightly adapted to our needs, and analyze its arithmetic complexity.

4.3.1 Triangular decomposition

Let P and Q be two polynomials in $\mathbb{F}[X,Y]$. A decomposition of the solutions of the system $\{P,Q\}$ using the subresultant sequence appears in the theory of triangular sets [Laz91, LMMRS11] and for the computation of topology of curves [GVEK96].

The idea is to use Lemma 2 which states that, after specialization at $X = \alpha$, the first (with respect to increasing i) nonzero subresultant $Sres_{Y,i}(P,Q)(\alpha, Y)$ is of degree i and is equal to the gcd of $P(\alpha,Y)$ and $Q(\alpha,Y)$. This induces a decomposition of the system $\{P,Q\}$ into triangular subsystems $(\{A_i(X), Sres_{Y,i}(P,Q)(X,Y)\})$ where a solution α of $A_i(X) = 0$ is such that the

[7] [Yap00, Lemma 4.12] states the bound as $N^{2d'+2}$ where N is the maximum Euclidean norm of the vectors of coefficients of the polynomials.

Algorithm 1 Triangular decomposition [GVEK96, LMMRS11]

Input: P, Q in $\mathbb{F}[X,Y]$ coprime such that $Lc_Y(P)$ and $Lc_Y(Q)$ are coprime,[8] $d_Y(Q) \leqslant d_Y(P)$, and
$A \in \mathbb{F}[X]$ squarefree.

Output: Triangular decomposition $\{(A_i(X),\ B_i(X,Y))\}_{i\in\mathcal{I}}$ such that $V(\langle P,Q,A\rangle)$ is the disjoint union of the sets $V(\langle A_i(X), B_i(X,Y)\rangle)_{i\in\mathcal{I}}$

1: Compute the subresultant sequence of P and Q with respect to Y: $B_i = Sres_{Y,i}(P,Q)$
2: $G_0 = \gcd(\overline{Res_Y(P,Q)}, A)$ and $\mathcal{T} = \emptyset$
3: **for** $i = 1$ **to** $d_Y(Q)$ **do**
4: $\quad G_i = \gcd(G_{i-1}, sres_{Y,i}(P,Q))$
5: $\quad A_i = G_{i-1}/G_i$
6: $\quad$ if $d_X(A_i) > 0$, add (A_i, B_i) to $\mathcal{T}$
7: **end for**
8: **return** $\mathcal{T} = \{(A_i(X), B_i(X,Y))\}_{i\in\mathcal{I}}$

system $\{P(\alpha,Y), Q(\alpha,Y)\}$ admits exactly i roots (counted with multiplicity), which are exactly those of $Sres_{Y,i}(P,Q)(\alpha,Y)$. Furthermore, these triangular subsystems are regular chains, i.e., the leading coefficient of the bivariate polynomial (seen in Y) is coprime with the univariate polynomial. For clarity and self-containedness, we recall this decomposition in Algorithm 1, where, in addition, we restrict the solutions of the system $\{P,Q\}$ to those where some univariate polynomials $A(X)$ vanishes (A could be identically zero).

The following lemma states the correctness of Algorithm 1 which follows from Lemma 2 and from the fact that the solutions of P and Q project on the roots of their resultant.

Lemma 12 ([GVEK96, LMMRS11]). *Algorithm 1 computes a triangular decomposition* $\{(A_i(X), B_i(X,Y))\}_{i\in\mathcal{I}}$ *such that*
(i) the set $V(\langle P,Q,A\rangle)$ *is the disjoint union of the sets* $V(\langle A_i(X), B_i(X,Y)\rangle)_{i\in\mathcal{I}}$,
(ii) $\prod_{i\in\mathcal{I}} A_i$ *is squarefree,*
(iii) $\forall\alpha \in V(A_i)$, $B_i(\alpha,Y)$ *is of degree* i *and is equal to* $\gcd(P(\alpha,Y), Q(\alpha,Y))$, *and*
(iv) $A_i(X)$ *and* $Lc_Y(B_i(X,Y))$ *are coprime.*

In the following lemma, we analyze the complexity of Algorithm 1 for P and Q of degree at most d_X in X and d_Y in Y and A of degree at most d^2, where d denotes a bound on the total degree of P and Q. We will use Algorithm 1 with polynomials with coefficients in $\mathbb{F} = \mathbb{Z}_\mu$ and we thus only consider its arithmetic complexity in $\mathbb{F}$. Note that the bit complexity of this algorithm, over $\mathbb{Z}$, is analyzed in [DET09, Theorem 19] and its arithmetic complexity is thus implicitly analyzed as well; for clarity, we provide here a short proof.

Lemma 13. *Algorithm 1 performs* $\widetilde{O}(d_X d_Y^3) = \widetilde{O}(d^4)$ *arithmetic operations in* $\mathbb{F}$.

Proof. From Lemma 3 (note that this lemma is stated for the coefficient ring $\mathbb{Z}$, but the arithmetic complexity is the same for any field $\mathbb{F}$), the subresultant sequence of P and Q can be computed in $\widetilde{O}(d_X d_Y^3)$ arithmetic operations, and the resultant as well as the principal subresultant coefficients have degrees in $O(d_X d_Y)$. The algorithm performs at most d_Y gcd computations between these univariate polynomials. The arithmetic complexity of one such gcd computation is soft linear in their degrees, that is $\widetilde{O}(d_X d_Y)$ (Lemma 4). Hence the arithmetic complexity of computing the

[8] The hypothesis that $Lc_Y(P)$ and $Lc_Y(Q)$ are coprime can be relaxed by applying the algorithm recursively (see [LMMRS11] for details). We require here this hypothesis for complexity issues.

Algorithm 2 Number of distinct solutions of $\langle P_\mu, Q_\mu \rangle$

Input: P_μ, Q_μ in $\mathbb{Z}_\mu[X,Y]$ coprime, μ larger than their total degree
Output: Number of distinct solutions of $\langle P_\mu, Q_\mu \rangle$

1: Shear P_μ and Q_μ by replacing X by $X - bY$ with $b \in \mathbb{Z}_\mu$ so that $Lc_Y(P_\mu(X - bY, Y)) \in \mathbb{Z}_\mu$
2: Triangular decomposition: $\{(A_i(X), B_i(X,Y))\}_{i\in\mathcal{I}}$ = Algorithm 1 $(P_\mu, Q_\mu, 0)$
3: **for all** $i \in \mathcal{I}$ **do**
4: $\quad C_i(X) = Lc_Y(B_i(X,Y))^{-1} \bmod A_i(X)$
5: $\quad \tilde{B}_i(X,Y) = C_i(X)B_i(X,Y) \bmod A_i(X)$
6: $\quad$ Triangular decomp.:
$\quad \{(A_{ij}(X), B_{ij}(X,Y))\}_{j\in\mathcal{J}_i}$ = Algorithm 1 $\left(\tilde{B}_i(X,Y), \frac{\partial \tilde{B}_i(X,Y)}{\partial Y}, A_i(X)\right)$
7: **end for**
8: **return** $\sum_{i\in\mathcal{I}} \left(i\, d_X(A_i) - \sum_{j\in\mathcal{J}_i} j\, d_X(A_{ij}) \right)$

systems $\{S_i\}_{i=1\ldots d}$ is $\widetilde{O}(d_X d_Y^2)$. The total complexity of the triangular decomposition is hence dominated by the cost of the subresultant computation, that is $\widetilde{O}(d_X d_Y^3) = \widetilde{O}(d^4)$. □

4.3.2 Counting the number of solutions over $\mathbb{Z}_\mu$

Algorithm 2 computes the number of distinct solutions of an ideal $I_\mu = \langle P_\mu, Q_\mu \rangle$ of $\mathbb{Z}_\mu[X,Y]$. Roughly speaking, this algorithm first performs one triangular decomposition with the input polynomials P_μ and Q_μ, and then performs a sequence of triangular decompositions with polynomials resulting from this decomposition. The result is close to a radical triangular decomposition and the number of solutions of I_μ can be read, with a simple formula, from the degrees of the polynomials in the decomposition. Note that Algorithm 2, as Algorithm 1, is valid for any base field $\mathbb{F}$ but, since we will only use it over $\mathbb{Z}_\mu$, we state it and analyze its complexity in this case.

Lemma 14. *Algorithm 2 computes the number of distinct solutions of* $\langle P_\mu, Q_\mu \rangle$.

Proof. The shear of Line 1 allows to fulfill the requirement of the triangular decomposition algorithm, called in Line 2, that the input polynomials have coprime leading coefficients. Once the generically sheared polynomial $P_\mu(X - SY, Y)$ is computed (in $\mathbb{Z}_\mu[S,X,Y]$), a specific shear value $b \in \mathbb{Z}_\mu$ can be selected by evaluating the univariate polynomial $L_{P_\mu}(S) = Lc_Y(P_\mu(X - SY, Y))$ at $d+1$ elements of $\mathbb{Z}_\mu$. The polynomial does not vanish at one of these values since it is of degree at most d and $d < \mu$. Note that such a shear clearly does not change the number of solutions.

According to Lemma 12, the triangular decomposition $\{(A_i(X), B_i(X,Y))\}_{i\in\mathcal{I}}$ computed in Line 2 is such that the solutions of $\langle P_\mu, Q_\mu \rangle$ is the disjoint union of the solutions of the $\langle A_i(X), B_i(X,Y) \rangle$, for $i \in \mathcal{I}$. It follows that the number of (distinct) solutions of $I_\mu = \langle P_\mu, Q_\mu \rangle$ is

$$\#V(I_\mu) = \sum_{i\in\mathcal{I}} \sum_{\alpha\in V(A_i)} d_Y(\overline{B_i(\alpha, Y)}).$$

Since $B_i(\alpha, Y)$ is a univariate polynomial in Y,
$d_Y(\overline{B_i(\alpha,Y)}) = d_Y(B_i(\alpha,Y)) - d_Y(\gcd(B_i(\alpha,Y), B_i'(\alpha,Y)))$, where $B_i'(\alpha,Y)$ is the derivative of $B_i(\alpha,Y)$, which is also equal to $\frac{\partial B_i}{\partial Y}(\alpha,Y)$. By Lemma 12, $d_Y(B_i(\alpha,Y)) = i$, and since the

degree of the gcd is zero when $B_i(\alpha, Y)$ is squarefree, we have

$$\#V(I_\mu) = \sum_{i\in\mathcal{I}} \left(\sum_{\alpha\in V(A_i)} i - \sum_{\substack{\alpha\in V(A_i)\\ B_i(\alpha,Y) \text{ not sqfr.}}} d_Y(\gcd(B_i(\alpha,Y), \tfrac{\partial B_i}{\partial Y}(\alpha,Y))) \right). \tag{5}$$

The polynomials $A_i(X)$ are squarefree by Lemma 12, so $\sum_{\alpha\in V(A_i)} i$ is equal to $i\, d_X(A_i)$.

We now consider the sum of the degrees of the gcds. The rough idea is to apply Algorithm 1 to $B_i(X,Y)$ and $\frac{\partial B_i}{\partial Y}(X,Y)$, for every $i \in \mathcal{I}$, which computes a triangular decomposition $\{(A_{ij}(X), B_{ij}(X,Y))\}_{j\in\mathcal{J}_i}$ such that, for $\alpha \in V(A_{ij})$, $d_Y(\gcd(B_i(\alpha,Y), \frac{\partial B_i}{\partial Y}(\alpha,Y))) = j$ (by Lemma 12), which simplifies Equation (5) into $\#V(I_\mu) = \sum_{i\in\mathcal{I}} \left(i\, d_X(A_i) - \sum_{j\in\mathcal{J}_i} \sum_{\alpha\in V(A_{ij})} j \right)$. However, we cannot directly apply Algorithm 1 to $B_i(X,Y)$ and $\frac{\partial B_i}{\partial Y}(X,Y)$ because their leading coefficients in Y have no reason to be coprime.

By Lemma 12, $A_i(X)$ and $Lc_Y(B_i(X,Y))$ are coprime, thus $Lc_Y(B_i(X,Y))$ is invertible modulo $A_i(X)$ (by Bézout's identity); let $C_i(X)$ be this inverse and define $\tilde{B}_i(X,Y) = C_i(X)B_i(X,Y) \bmod A_i(X)$ (such that every coefficient of $C_i(X)B_i(X,Y)$ with respect to Y is reduced modulo $A_i(X)$). The leading coefficient in Y of $\tilde{B}_i(X,Y)$ is equal to 1, so we can apply Algorithm 1 to $\tilde{B}_i(X,Y)$ and $\frac{\partial \tilde{B}_i}{\partial Y}(X,Y)$. Furthermore, if $A_i(\alpha) = 0$, then $\tilde{B}_i(\alpha,Y) = C_i(\alpha)B_i(\alpha,Y)$ where $C_i(\alpha) \neq 0$ since $C_i(\alpha)Lc_Y(B_i(\alpha,Y)) = 1$. Equation (5) can thus be rewritten by replacing B_i by $\tilde{B}_i$.

By Lemma 12, for every $i \in \mathcal{I}$, Algorithm 1 computes a triangular decomposition $\{(A_{ij}(X), B_{ij}(X,Y))\}_{j\in\mathcal{J}_i}$ such that $V(\langle \tilde{B}_i, \frac{\partial \tilde{B}_i}{\partial Y}, A_i\rangle)$ is the disjoint union of the sets $V(\langle A_{ij}(X), B_{ij}(X,Y)\rangle)$, $j \in \mathcal{J}_i$, and for all $\alpha \in V(A_{ij})$, $d_Y(\gcd(\tilde{B}_i(\alpha,Y), \frac{\partial \tilde{B}_i}{\partial Y}(\alpha,Y))) = j$. Since the set of $\alpha \in V(A_i)$ such that $\tilde{B}_i(\alpha,Y)$ is not squarefree is the projection of the set of solutions $(\alpha,\beta) \in V(\langle \tilde{B}_i, \frac{\partial \tilde{B}_i}{\partial Y}, A_i\rangle)$ we get

$$\#V(I_\mu) = \sum_{i\in\mathcal{I}} \left(i\, d_X(A_i) - \sum_{j\in\mathcal{J}_i} \sum_{\alpha\in V(A_{ij})} j \right).$$

$A_{ij}(X)$ is squarefree (Lemma 12) so $\sum_{\alpha\in V(A_{ij})} j = j\, d_X(A_{ij})$, which concludes the proof. □

The next lemma gives the arithmetic complexity of the above algorithm.

Lemma 15. *Given P_μ, Q_μ in $\mathbb{Z}_\mu[X,Y]$ of total degree at most d, Algorithm 2 performs $\widetilde{O}(d^4)$ operations in $\mathbb{Z}_\mu$.*

Proof. According to Lemma 5, the sheared polynomials $P(T-SY,Y)$ and $Q(T-SY,Y)$ can be expanded in $\widetilde{O}_B(d^4+d^3\tau)$ bit operations in $\mathbb{Z}$. Thus the sheared polynomials $P_\mu(X-SY,Y)$ and $Q_\mu(X-SY,Y)$ can obviously be computed in $\widetilde{O}(d^4)$ arithmetic operations in $\mathbb{Z}_\mu$.[9] The leading term $Lc_Y(P_\mu(X-SY,Y)) \in \mathbb{Z}_\mu[S]$ is a polynomial of degree at most d and a value $b \in \mathbb{Z}_\mu$ that does not vanish it can be found by at most $d+1$ evaluations. Each evaluation can be done with $O(d)$ arithmetic operations, thus the shear value b can be computed in $\widetilde{O}(d^2)$ operations. It remains to evaluate the generically sheared polynomials at this value $S = b$. These polynomials have $O(d^2)$ monomials in X and Y, each with a coefficient in $\mathbb{Z}_\mu[S]$ of degree at most d; since the evaluation of each coefficient is soft linear in d, this gives a total complexity in $\widetilde{O}(d^4)$ for Line 1.

[9] It can easily be proved that these polynomials can be computed in $\widetilde{O}(d^3)$ arithmetic operations but the $\widetilde{O}(d^4)$ bound is sufficient here.

Algorithm 3 Number of distinct solutions and lucky prime for $\langle P, Q\rangle$

Input: P, Q in $\mathbb{Z}[X, Y]$ coprime of total degree at most d and bitsize at most τ
Output: The number of solutions and a lucky prime μ for $\langle P, Q\rangle$

1: Compute $P(T - SY, Y)$ and $Q(T - SY, Y)$
2: Compute a set B of primes larger than $2d^4$ and of cardinality $\widetilde{O}(d^4 + d^3\tau)$ that contains a lucky prime for $\langle P, Q\rangle$ (see Proposition 11)
3: **for all** μ in B **do**
4: Compute the reduction modulo μ of $P, Q, L_P(S), L_Q(S)$ and $Res_Y(\phi_\mu(P), \phi_\mu(Q))$
5: **if** $Res_Y(\phi_\mu(P), \phi_\mu(Q)) \not\equiv 0$ and $\phi_\mu(L_P(S))\ \phi_\mu(L_Q(S)) \not\equiv 0$ **then**
6: Compute N_μ = Algorithm 2$(\phi_\mu(P), \phi_\mu(Q))$
7: **end if**
8: **end for**
9: **return** (μ, N_μ) such that N_μ is maximum

According to Lemma 13, the triangular decomposition in Line 2 can be done in $\widetilde{O}(d^4)$ arithmetic operations. In Lines 4 and 5, $C_i(X)$ and $\tilde{B}_i(X, Y)$ can be computed by first reducing modulo $A_i(X)$ every coefficient of $B_i(X, Y)$ (with respect to Y). There are at most i coefficients (by definition of subresultants) and the arithmetic complexity of every reduction is soft linear in the degree of the operands [vzGG99, Corollary 11.6], which is $\widetilde{O}(d^2)$ by Lemma 3. The reduction of $B_i(X, Y)$ modulo $A_i(X)$ can thus be done with $\widetilde{O}(d^3)$ arithmetic operations in $\mathbb{Z}_\mu$. Now, in Line 4, the arithmetic complexity of computing the inverse of one of these coefficients modulo $A_i(X)$ is soft linear in its degree [vzGG99, Corollary 11.8], that is $\widetilde{O}(d_i)$ where d_i denotes the degree of $A_i(X)$. Furthermore, computing the product modulo $A_i(X)$ of two polynomials which are already reduced modulo $A_i(X)$ can be done in $\widetilde{O}(d_i)$ arithmetic operations [vzGG99, Corollary 11.8]. Thus, in Line 5, the computation of $\tilde{B}_i(X, Y)$ can be done with i such multiplications, and thus with $\widetilde{O}(id_i)$ arithmetic operations. Finally, in Line 6, the triangular decomposition can be done with $\widetilde{O}(i^3 d_i)$ arithmetic operations by Lemma 13. The complexity of Lines 4-6 is thus in $\widetilde{O}(d^3 + i^3 d_i)$ which is in $\widetilde{O}(d^3 + d^2 i d_i)$. The total complexity of the loop in Line 3 is thus $\widetilde{O}(d^4 + d^2 \sum_i i d_i)$ which is in $\widetilde{O}(d^4)$ because the number of solutions of the triangular system $(A_i(X), B_i(X, Y))$ is at most the degree of A_i times the degree of B_i in Y, that is id_i, and the total number of these solutions for $i \in \mathcal{I}$ is that of (P, Q), by Lemma 12, which is at most d^2 by Bézout's bound. This concludes the proof because the sum in Line 8 can obviously be done in linear time in the size of the triangular decompositions that are computed during the algorithm. □

4.4 Computing a lucky prime and the number of solutions over $\mathbb{Z}$

We now show how to compute the number of solutions of $I = \langle P, Q\rangle$ over $\mathbb{Z}$ and a lucky prime for that ideal.

Lemma 16. *Algorithm 3 computes the number of distinct solutions and a lucky prime for $\langle P, Q\rangle$ in $\widetilde{O}_B(d^8 + d^7\tau)$ bit operations. Moreover, this lucky prime is upper bounded by $\widetilde{O}(d^4 + d^3\tau)$.*

Proof. We first prove the correctness of the algorithm. Note first that for all $\mu \in B$ satisfying the constraint of Line 5, $\phi_\mu(P)$ and $\phi_\mu(Q)$ are coprime. It follows that Algorithm 2 computes the number of distinct solutions $N_\mu = \#V(I_\mu)$ of I_μ. By Proposition 10 and Definition 6, $N_\mu \leqslant \#V(I)$ and the equality holds if μ is lucky for I. Since the set B of considered primes

Algorithm 4 Separating form for $\langle P, Q\rangle$

Input: P, Q in $\mathbb{Z}[X, Y]$ of total degree at most d and defining a zero-dimensional ideal I
Output: A linear form $X + aY$ that separates $V(I)$, with $a < 2d^4$ and $L_P(a)\, L_Q(a) \neq 0$

1: Apply Algorithm 3 to compute the number of solutions $\#V(I)$ and a lucky prime μ for I
2: Compute $P(T - SY, Y)$, $Q(T - SY, Y)$ and $R(T, S) = Res_Y(P(T - SY, Y), Q(T - SY, Y))$
3: Compute $R_\mu(T, S) = \phi_\mu(R(T, S))$
4: Compute $\Upsilon_\mu(S) = \phi_\mu(L_P(S))\ \phi_\mu(L_Q(S))$
5: $a := 0$
6: **repeat**
7: Compute the degree N_a of the squarefree part of $R_\mu(T, a)$
8: $a := a + 1$
9: **until** $\Upsilon_\mu(a) \neq 0$[10]and $N_a = \#V(I)$
10: **return** The linear form $X + aY$

contains a lucky one by construction, the maximum of the computed value of N_μ is equal to $\#V(I)$. Finally, the μ associated to any such maximum value of N_μ is necessarily lucky by the constraint of Line 5 and since μ is larger than $2d^4$.

We now prove the complexity of the algorithm. The polynomials $P(T - SY, Y)$ and $Q(T - SY, Y)$ can be computed in $\widetilde{O}_B(d^4 + d^3\tau)$ bit operations by Lemma 5.

Proposition 11 states that we can compute an explicit bound $\Xi(d, \tau)$ in $\widetilde{O}(d^4 + d^3\tau)$ on the number of unlucky primes for $\langle P, Q\rangle$. We want to compute in Line 2 a set B of at least $\Xi(d, \tau)$ primes (plus one) that are larger than $2d^4$. For computing B, we can thus compute the first $\Xi(d, \tau) + 2d^4 + 1$ prime numbers and reject those that are smaller than $2d^4$. The bit complexity of computing the r first prime numbers is in $\widetilde{O}(r)$ and their maximum is in $\widetilde{O}(r)$ [vzGG99, Theorem 18.10]. We can thus compute the set of primes B with $\widetilde{O}_B(d^4 + d^3\tau)$ bit operations and these primes are in $\widetilde{O}(d^4 + d^3\tau)$.

Polynomials P, Q, $L_P(S)$ and $L_Q(S)$ are of degree at most d in one or two variables and they have bitsize at most $\widetilde{O}(d + \tau)$ (Lemma 5). The reduction of all their $O(d^2)$ coefficients modulo all the primes in B can be computed via a remainder tree in a bit complexity that is soft linear in the total bitsize of the input [MB74, Theorem 1], which is dominated by the sum of the bitsizes of the $\widetilde{O}(d^4 + d^3\tau)$ primes in B each of bitsize $\widetilde{O}(1)$. Furthermore, computing the resultant of $\phi_\mu(P)$ and $\phi_\mu(Q)$ can be done with $\widetilde{O}(d^3)$ arithmetic operations in $\mathbb{Z}_\mu$ (Lemma 3) and thus in $\widetilde{O}_B(d^3)$ bit operations since μ has bitsize $\widetilde{O}(1)$. Hence, the bit complexity of Line 4 is $\widetilde{O}_B(d^4 + d^3\tau)$.

Finally, the total bit complexity of Line 6 is $\widetilde{O}_B(d^8 + d^7\tau)$, since each call to Algorithm 2 has bit complexity $\widetilde{O}_B(d^4)$ by Lemma 15 (since μ has bitsize $\widetilde{O}(1)$). The overall bit complexity of the algorithm is thus in $\widetilde{O}_B(d^8 + d^7\tau)$. □

4.5 Computing a separating linear form

Using Algorithm 3, we now present our algorithm for computing a linear form that separates the solutions of $\langle P, Q\rangle$.

Theorem 17. *Algorithm 4 returns a separating linear form $X + aY$ for $\langle P, Q\rangle$ with $a < 2d^4$. The bit complexity of the algorithm is in $\widetilde{O}_B(d^8 + d^7\tau)$.*

[10] $\Upsilon_\mu(S)$ is a polynomial in $\mathbb{Z}_\mu[S]$ and we consider $\Upsilon_\mu(a)$ in $\mathbb{Z}_\mu$.

Proof. We first prove the correctness of the algorithm. We start by proving that the value a returned by the algorithm is the smallest nonnegative integer such that $X + aY$ separates $V(I_\mu)$ with $\Upsilon_\mu(a) \neq 0$. Note first that, in Line 3, $\phi_\mu(R(T,S))$ is indeed equal to $R_\mu(T,S)$ which is defined as $Res_Y(P_\mu(T - SY, Y), Q_\mu(T - SY, Y))$ since the leading coefficients $L_P(S)$ and $L_Q(S)$ of $P(T - SY, Y)$ and $Q(T - SY, Y)$ do not identically vanish modulo μ (since μ is lucky), and thus $L_{P_\mu}(S) = \phi_\mu(L_P(S))$, similarly for Q, and the resultant can be specialized modulo μ [BPR06, Proposition 4.20]. Now, Line 9 ensures that the value a returned by the algorithm satisfies $\Upsilon_\mu(a) \neq 0$, and we restrict our attention to nonnegative such values of a. Note that $\Upsilon_\mu(a) \neq 0$ implies that $\phi_\mu(L_P(a))\ \phi_\mu(L_Q(a)) \neq 0$ because the specialization at $S = a$ and the reduction modulo μ commute (in $\mathbb{Z}_\mu$). For the same reason, $L_{P_\mu}(S) = \phi_\mu(L_P(S))$ implies $L_{P_\mu}(a) = \phi_\mu(L_P(a))$ and thus $L_{P_\mu}(a) \neq 0$ and, similarly, $L_{Q_\mu}(a) \neq 0$. On the other hand, Line 9 implies that the value a is the smallest that satisfies $d_T(\overline{R_\mu(T,a)}) = \#V(I)$, which is also equal to $\#V(I_\mu)$ since μ is lucky. Lemma 8 thus yields that the returned value a is the smallest nonnegative integer such that $X + aY$ separates $V(I_\mu)$ and $\Upsilon_\mu(a) \neq 0$, which is our claim.

This property first implies that $a < 2d^4$ because the degree of Υ_μ is bounded by $2(d^2 + d)$, the number of non-separating linear forms is bounded by $\binom{d^2}{2}$ (the maximum number of directions defined by any two of d^2 solutions), and their sum is less than $2d^4$ for $d \geqslant 2$. Note that, since μ is lucky, $2d^4 < \mu$ and thus $a < \mu$. The above property thus also implies, by Proposition 7, that $X + aY$ separates $V(I)$. This concludes the proof of correctness of the algorithm since $a < 2d^4$ and $L_P(a)\, L_Q(a) \neq 0$ (since $\Upsilon_\mu(a) \neq 0$).

We now focus on the complexity of the algorithm. By Lemma 16, the bit complexity of Line 1 is in $\widetilde{O}_B(d^8 + d^7\tau)$. The bit complexity of Lines 2 to 5 is in $\widetilde{O}_B(d^7 + d^6\tau)$. Indeed, by Lemma 5, $R(T,S)$ has degree $O(d^2)$ in T and in S, bitsize $\widetilde{O}(d^2 + d\tau)$, and it can be computed in $\widetilde{O}_B(d^7 + d^6\tau)$ time. Computing $R_\mu(T,S) = \phi_\mu(R(T,S))$ can thus be done in reducing $O(d^4)$ integers of bitsize $\widetilde{O}(d^2 + d\tau)$ modulo μ. Each reduction is soft linear in the maximum of the bitsizes [vzGG99, Theorem 9.8] thus the reduction of $R(T,S)$ can be computed in $\widetilde{O}_B(d^4(d^2+d\tau))$ time (since μ has bitsize in $O(\log(d^4+d^3\tau))$ by Lemma 16).[11] The computation of Υ_μ can clearly be done with the same complexity since each reduction is easier than the one in Line 3, and the product of the polynomials (which does not actually need to be computed since we are only interested in whether $\Upsilon_\mu(a)$ vanishes) can be done with a bit complexity that is soft linear in the product of the maximum degrees and maximum bitsizes [vzGG99, Corollary 8.27].

We proved that the value a returned by the algorithm is less than $2d^4$, thus the loop in Line 6 is performed at most $2d^4$ times. Each iteration consists of computing the squarefree part of $R_\mu(T,a)$ which requires $\widetilde{O}_B(d^4)$ bit operations. Indeed, computing $R_\mu(T,S)$ at $S = a$ amounts to evaluating, in $\mathbb{Z}_\mu$, $O(d^2)$ polynomials in S, each of degree $O(d^2)$ (by Lemma 5). Note that a does not need to be reduced modulo μ because $a < 2d^4$ and $2d^4 < \mu$ since μ is lucky. Thus, the bit complexity of evaluating in $\mathbb{Z}_\mu$ each of the $O(d^2)$ polynomials in S is the number of arithmetic operations in $\mathbb{Z}_\mu$, which is linear the degree that is $O(d^2)$, times the (maximum) bit complexity of the operations in $\mathbb{Z}_\mu$, which is in $O_B(\log d\tau)$ since μ is in $\widetilde{O}(d^4 + d^3\tau)$ by Lemma 16. Hence, computing $R_\mu(T,a)$ can be done in $\widetilde{O}_B(d^4)$ bit operations. Once $R_\mu(T,a)$ is computed, the arithmetic complexity of computing its squarefree part in $\mathbb{Z}_\mu$ is soft linear in its degree (Lemma 4), that is $\widetilde{O}(d^2)$, which yields a bit complexity in $\widetilde{O}_B(d^2)$ since, again, μ is in $\widetilde{O}(d^4 + d^3\tau)$. This leads to a total bit complexity of $\widetilde{O}_B(d^8)$ for the loop in Lines 6 to 9, and

[11] Note that $R_\mu(T,S)$ can be computed more efficiently in $\widetilde{O}_B(d^5 + d^3\tau)$ bit operations as the resultant of $P_\mu(T - SY, Y)$ and $Q_\mu(T - SY, Y)$ because computing these two polynomials and their reduction can be done in $\widetilde{O}_B(d^4 + d^3\tau)$ bit operations (Lemma 5) and their resultant can be computed with $\widetilde{O}(d^5)$ arithmetic operations in $\mathbb{Z}_\mu$ (Lemma 3) and thus with $\widetilde{O}_B(d^5)$ bit operations since μ has bitsize in $O(\log(d^4 + d^3\tau))$.

thus to a total bit complexity for the algorithm in $\widetilde{O}_B(d^8 + d^7\tau)$. □

5 Conclusion

We presented an algorithm of bit complexity $\widetilde{O}_B(d^8 + d^7\tau)$ for finding a separating linear form of a bivariate system, improving by a factor d^2 the best known algorithm for this problem. Finding a separating linear form is at the core of approaches based on rational parametrizations for solving such systems and, as mentionned in the introduction, our algorithm directly improves the bit complexity of the classical method for computing rational parametrizations via subresultants [GVEK96]. Interestingly, computing a separating linear form remains the bit-complexity bottleneck in this algorithm [DET09] and we show in [BLPR13] that this is also the bottleneck for computing the rational parameterization of [Rou99]. This thus yields algorithms of bit complexity $\widetilde{O}_B(d^8 + d^7\tau)$ for computing rational parameterizations of bivariate systems and we show in [BLPR13] that isolating boxes can be computed with a smaller bit complexity. It should be stressed that this complexity matches the recent one presented by Emeliyanenko and Sagraloff [ES12] for "only" computing isolating boxes of the real solutions. Furthermore, rational parameterizations yield efficient algorithms for various related problems, such as evaluating the sign of a polynomial at the solutions of the system, or solving over-constrained systems [BLPR13].

One interesting open problem is to determine how, or whether, this contribution may impact the complexity of algorithms, on plane algebraic curves, that require finding a shear that ensures the curves to be in "generic" position (such as [KS11, GVN02]). In particular, we hope that this result may improve the complexity of computing the topology of an algebraic plane curve.

References

[ABRW96] M.-E Alonso, E. Becker, M.-F. Roy, and T. Wörmann. Multiplicities and idempotents for zerodimensional systems. In *Algorithms in Algebraic Geometry and Applications*, volume 143 of *Progress in Mathematics*, pages 1–20. Birkhäuser, 1996.

[BLPR13] Y. Bouzidi, S. Lazard, M. Pouget, and F. Rouillier. Solving bivariate systems: Efficient worst-case algorithm for computing rational univariate representations and applications. INRIA Research Report 8262, 2013.

[BPR06] S. Basu, R. Pollack, and M.-R. Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer-Verlag, 2nd edition, 2006.

[BSS03] A. Bostan, B. Salvy, and É. Schost. Fast algorithms for zero-dimensional polynomial systems using duality. *Applicable Algebra in Engineering, Communication and Computing*, 14(4):239–272, 2003.

[CLO97] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, 2nd edition, 1997.

[DET09] 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.

[EK03] M. El Kahoui. An elementary approach to subresultants theory. *J. Symb. Comput.*, 35(3):281–292, 2003.

[ES12] P. Emeliyanenko and M. Sagraloff. On the complexity of solving a bivariate polynomial system. In *Proceedings of the 37th international symposium on Symbolic and algebraic computation*, ISSAC '12, 2012.

[GLS01] M. Giusti, G. Lecerf, and B. Salvy. A Gröbner free alternative for solving polynomial systems. *J. of Complexity*, 17(1):154–211, 2001.

[GVEK96] L. González-Vega and M. El Kahoui. An improved upper complexity bound for the topology computation of a real algebraic plane curve. *J. Complexity*, 12(4):527–544, 1996.

[GVN02] L. González-Vega and I. Necula. Efficient topology determination of implicitly defined algebraic plane curves. *Computer Aided Geometric Design*, 19(9), 2002.

[KS11] M. Kerber and M. Sagraloff. A worst-case bound for topology computation of algebraic curves. *CoRR*, abs/1104.1510, 2011.

[KS12] M. Kerber and M. Sagraloff. A worst-case bound for topology computation of algebraic curves. *J. Symb. Comput.*, 47(3):239 – 258, 2012.

[Laz91] D. Lazard. A new method for solving algebraic systems of positive dimension. *Discrete Appl. Math.*, 33:147–160, October 1991.

[LMMRS11] X. Li, M. Moreno Maza, R. Rasheed, and É. Schost. The modpn library: Bringing fast polynomial arithmetic into maple. *J. Symb. Comput.*, 46(7):841 – 858, 2011.

[MB74] R. Moenck and A. Borodin. Fast modular transforms. *Journal of Computer and System Sciences*, 8, 1974.

[Rei97] D. Reischert. Asymptotically fast computation of subresultants. In *Proceedings of the 1997 international symposium on Symbolic and algebraic computation*, ISSAC '97, pages 233–240, New York, NY, USA, 1997. ACM.

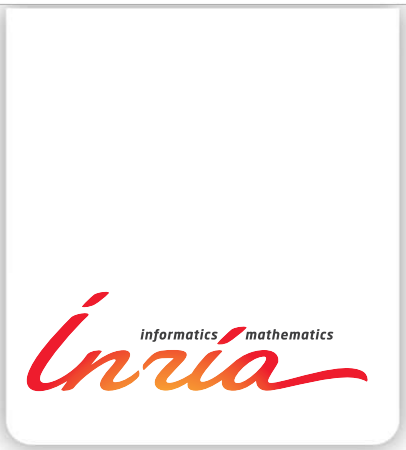
[Rou99] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *J. of Applicable Algebra in Engineering, Communication and Computing*, 9(5):433–461, 1999.

[Sag12] M. Sagraloff. When Newton meets Descartes: A Simple and Fast Algorithm to Isolate the Real Roots of a Polynomial. In *Proceedings of the 37th international symposium on Symbolic and algebraic computation*, ISSAC '12, 2012.

[VdW30] B. L. Van der Waerden. *Moderne Algebra I.* Berlin, 1930.

[vzGG99] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra.* Cambridge Univ. Press, Cambridge, U.K., 1st edition, 1999.

[Yap00] C.K. Yap. *Fundamental Problems of Algorithmic Algebra.* Oxford University Press, Oxford-New York, 2000.



RESEARCH CENTRE
NANCY – GRAND EST

615 rue du Jardin Botanique
CS20101
54603 Villers-lès-Nancy Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399