Separating Linear Forms for Bivariate Systems
Yacine Bouzidi, Sylvain Lazard, Marc Pouget, Fabrice Rouillier

To cite this version:
Yacine Bouzidi, Sylvain Lazard, Marc Pouget, Fabrice Rouillier. Separating Linear Forms for Bivariate Systems. ISSAC - 38th International Symposium on Symbolic and Algebraic Computation, Jun 2013, Boston, United States. pp.117-124. hal-00809425

HAL Id: hal-00809425
https://inria.hal.science/hal-00809425
Submitted on 9 Apr 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Separating Linear Forms for Bivariate Systems

Yacine Bouzidi
INRIA Nancy Grand Est
LORIA, Nancy, France
Yacine.Bouzidi@inria.fr

Sylvain Lazard
INRIA Nancy Grand Est
LORIA, Nancy, France
Sylvain.Lazard@inria.fr

Marc Pouget
INRIA Nancy Grand Est
LORIA, Nancy, France
Marc.Pouget@inria.fr

Fabrice Rouillier
INRIA Paris-Rocquencourt
IMJ, Paris, France
Fabrice.Rouillier@inria.fr

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 of a separating linear form is the bottleneck of these algorithms, in terms of worst-case bit complexity.

Our main contribution is a new deterministic approach with best known worst-case bit complexity for computing a separating linear form of a system of bivariate polynomials with integer coefficients of bitsize at most $\tau$. For such systems, the approach breaks down to (i) $O_B(d^{10} + d^7 \tau)$ for computing a separating linear form and then (ii) $O_B(d^6 + d^5 \tau)$ 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 $O_B(d^8 + d^7 \tau)$

Main results. Our main contribution is a new deterministic algorithm of worst-case bit complexity $O_B(d^8 + d^7 \tau + d^{12} \tau^2)$ 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 $\tau$ (Thm. 18). The system should be zero dimensional but this is tested in our case.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISSAC ’13, June 26–29, 2013, Boston, Massachusetts, USA.
Copyright 2013 ACM 978-1-4503-2059-7/13/06 ...$15.00.

Categories and Subject Descriptors
F.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems

Keywords
Bivariate system; Separating Linear Form

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 [4, 9, 13, 8, 3, 6] 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 $\tau$. For such systems, the approach breaks down to (i) $O_B(d^{10} + d^7 \tau)$ for computing a separating linear form and then (ii) $O_B(d^6 + d^5 \tau)$ 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 $O_B(d^8 + d^7 \tau)$

The overall bit complexity stated in [6, Thm. 19] is $O_B(d^{12} + d^{10} \tau^2)$ because it includes the isolation of the solutions of the system.
algorithm. When \( \tau \in \tilde{O}(d^2) \), this gives a complexity in \( \tilde{O}(d^3 + d^2\tau) \) which decreases by a factor \( d^2 \) the best known complexity for this problem (see the discussion above). Note furthermore that, while \( \tau \) is asymptotically negligible compared to \( d^3 \) (modulo polylogarithmic factors), i.e. \( \tau = O(d^3) \), the complexity of our algorithm is asymptotically better than the best known complexity for this problem, i.e. \( O(d^3 + d^2\tau + d^3\tau^2) \).

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. \[9,10\] and in that of Bouzidi et al. \[4\] for computing the alternative rational parameterization as defined in \[13\]. Moreover, this contribution is likely to impact the complexity of algorithmic rational parameterization within the same overall complexity as our approach of Gonzalez-Vega et al. \[9,10\].

As a direct consequence, using our algorithm for computing a separating form directly yields a rational parameterization within the same overall complexity as our approach of Gonzalez-Vega et al. \[9,10\].

Moreover, the stated complexity of \[10, Thm. 24\] is \( \tilde{O}((d^3 + d^2\tau + d^3\tau^2)) \) time (Section 4.5). This form, which is also separating for \( P \) and \( Q \) (the \( \tau^2 \) term results from the fact that we need to check that some coefficients do not vanish modulo \( \mu \)). Now, \( \mu \) is fixed, and we can apply the algorithm outlined above for computing a separating form for \( I_\mu \) in \( \tilde{O}(d^3 + d^2\tau) \) time (Section 4.5). This form, which is also separating for \( I \), is thus obtained with a total bit complexity of \( \tilde{O}(d^3 + d^2\tau + d^3\tau^2) \) (Thm. 18).

2. OVERVIEW AND ORGANIZATION

Let \( P \) and \( Q \) be two bivariate polynomials of total degree bounded by \( d \) and integer coefficients of maximum bitsize \( \tau \). 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 \[9,10\] Lemma 16 and \[10\] Thm. 24[4] and whose complexity, in \( \tilde{O}(d^{10} + d^3\tau) \), is the best known so far for this problem. This algorithm serves two purposes: it gives some insight on the more involved \( \tilde{O}(d^3 + d^2\tau + d^3\tau^2) \)-time algorithm that follows and it will be used in that algorithm but over \( \mathbb{Z}/\mu\mathbb{Z} \) instead of \( \mathbb{Z} \).

Known \( \tilde{O}(d^{10} + d^3\tau) \)-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 contexts; see for instance the related \( \mu \)-resultant \[14\].

It is known that, in a set \( S \) of \( d^3 \) integers, there exists at least one integer \( a \) such that \( X + aY \) is a separating form for \( I \) since \( I \) has at most \( d^3 \) solutions which define at most \( \binom{d^3}{2} \) directions in which two solutions are aligned. Hence, a separating form can be found by computing, for every \( a \) in \( S \), 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 5). The complexity of this algorithm is in \( O(\tilde{O}(d^{10} + d^3\tau)) \) because, for \( d^4 \) values of \( a \), the polynomial \( R(T,a) \) can be shown to be of degree \( O(d^3) \) and bitsize \( O(d^3 + d\tau) \), and its squarefree part can be computed in \( \tilde{O}(d^3 + d^2\tau) \) time.

As a byproduct, we obtain an algorithm for computing the number of distinct solutions of such systems within the same complexity, i.e. \( \tilde{O}(d^8 + d^2\tau + d^3\tau^2) \).

3. NOTATION AND PRELIMINARIES

We introduce notation and recall some classical material. The bitsize of an integer \( p \) is the number of bits needed to represent it, that is \( \lceil \log_2 p \rceil + 1 \) (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, \( \tilde{O} \) refers to the bit complexity and \( O \)
and $O_B$ refer to complexities where polylogarithmic factors are omitted, see [15] Definition 25.8 for details.

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

For any polynomial $P \in \mathbb{D}[X]$, let $LC_P$ ($P$) denote its leading coefficient with respect to the variable $X$, $d_X(P)$ its degree with respect to $X$, and $\mathcal{P}$ its squarefree part. The ideal generated by two polynomials $P$ and $Q$ is denoted $(P, Q)$, 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 $\sigma \in V(I)$, $\mu_\sigma(P)$ denotes the multiplicity of $\sigma$ in $I$. For simplicity, we refer indifferently to the ideal $(P, Q)$ 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 $Y$,

$$R(T,S) = \text{Res}_Y(P(T - SY,Y), Q(T - SY,Y)).$$

(1)

The complexity bounds on the degree, bitsize and computation of these polynomials are analyzed at the end of this section in Lemma 5. We introduce

$$L_P(S) = LC_P(P(T - SY,Y))$$

$L_Q(S) = LC_Q(Q(T - SY,Y)), L_R(S) = LC_R(R(T,S))$ (2)

and remark that these polynomials do not depend on $T$.

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

**Definition 1.** ([2] §3). For $i = 0, \ldots, \min(p,q-1)$, 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 $i = 0, \ldots, \min(p,q-1)$, the $i$-th polynomial subresultant of $P$ and $Q$, denoted by $Sres_{\alpha,i}(P,Q)$ is the polynomial determinant of $Sylv_i(P,Q)$. When $p = q$, the $q$-th polynomial subresultant of $P$ and $Q$ is $b_{q-q}^\perp$.

$Sres_{\alpha,i}(P,Q)$ has degree at most $i$ in $Y$, and the coefficient of its monomial of degree $i$ in $Y$, denoted by $sres_{\alpha,i}(P,Q)$, is called the $i$-th principal subresultant coefficient. Note that $sres_{\alpha,i}(P,Q) = sres_{\alpha,i}(P,Q)$ is the resultant of $P$ and $Q$ with respect to $Y$, which we also denote by $Res(P,Q)$.

We state below a fundamental property of subresultants which is instrumental in the triangular decomposition algorithm used in Section 1.3. For clarity, we state this property for bivariate polynomials $P = \sum_{i=0}^{\alpha} a_i Y^i$ and $Q = \sum_{i=0}^{\beta} b_i Y^i$ in $\mathbb{D}[X,Y]$, with $\beta \geq \alpha$. Note that this property is often stated with a stronger assumption that is that none of the leading terms $a_\alpha(\alpha)$ and $b_\beta(\beta)$ vanishes. This property is a direct consequence of the specialization property of subresultants and of the gap structure theorem; see for instance [2] Lemmas 2.3, 3.1 and Cor. 5.1.

**Lemma 2.** For any $\alpha$ such that $a_\alpha(\alpha)$ and $b_\beta(\beta)$ do not both vanish, the first $Sres_{\alpha,i}(P,Q)(\alpha,Y)$ (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 $\mathbb{D}(\alpha)$).

**Complexity.** We recall complexity results, using fast algorithms, on subresultants and gcd computations. We also state complexities related to the computation of the “sheared” polynomials and their resultant.

**Lemma 3.** ([2] Prop. 8.46] [12] [8]). Let $P$ and $Q$ be in $\mathbb{D}[X_1, \ldots, X_n][Y]$ (n fixed) with coefficients of bitsize at most $\sigma$ 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 $Sres_{\alpha,i}(P,Q)$ have bitsize in $O(d_Y+\tau)$.
- The degree in $X_i$ of $Sres_{\alpha,i}(P,Q)$ is at most $2d(d_Y+\tau)$.
- Any subresultant $Sres_{\alpha,i}(P,Q)$ can be computed in $O(d(d_Y+\tau))^2$ bit operations, and $O_B(d(d_Y+\tau))^2$ 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 $Q = P^\nu$, the latter is the squarefree part $P^\nu$.

**Lemma 4.** ([2], Rem. 10.19). Let $P$ and $Q$ in $\mathbb{D}[X]$ of degree at most $d$: $gcd(P,Q)$ or the gcd-free part of $P$ with respect to $Q$ can be computed with $O(d)$ operations in $\mathbb{D}$.

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

**Proof.** Writing $P(T - SY,Y) = \sum_{i=0}^{d} p_i(T)(T - SY)^i$ and considering the bitsize of the binomial coefficients, we easily get the first statement of the lemma. The second statement is a direct application of Lemma 3 on trivariate polynomials of partial degree at most $d$ in each variable.

4. SEPARATING LINEAR FORM

Throughout this section, we assume that the two input polynomials $P$ and $Q$ are coprime in $\mathbb{D}[X,Y]$, that they define the ideal $I$, that their maximum total degree $d$ is at least 2 and that their coefficients have maximum bitsize $\tau$. Note that the coprimality of $P$ and $Q$ is implicitly tested during Algorithm 1 because they are coprime if and only if $R(T,S)$ does not identically vanish. By abuse of notation, some complexity $O_B(d^\alpha)$ may refer to a complexity in which polylogarithmic factors in $d$ and in $\tau$ are omitted. $P_{\mu} = (P_{\mu}, Q_{\mu})$ 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 \( L_{\mu}(S), L_{Q_{\mu}}(S), \) and \( L_{R_{\mu}}(S) \), 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 \( \mu \) which are, roughly speaking, primes \( \mu \) for which the number of distinct solutions of \( (P, Q) \) does not change when considering the polynomials modulo \( \mu \). We then show the critical property that, if a linear form is separating modulo such a \( \mu \), then it is also separating over \( \mathbb{Z} \).

**Definition 6.** A prime number \( \mu \) is said to be **lucky** for an ideal \( I = \langle P, Q \rangle \) if it is larger than \( 2d^4 \) and satisfies
\[
\phi_\mu(L_P(S)) \phi_\mu(L_Q(S)) \neq 0 \quad \text{and} \quad \#V(I) = \#V(I_\mu).
\]

**Proposition 7.** Let \( \mu \) be a lucky prime for the ideal \( I = \langle P, Q \rangle \) and let \( a < \mu \) be an integer such that
\[
\phi_\mu(L_P(a)) \phi_\mu(L_Q(a)) \phi_\mu(L_R(a)) \neq 0.
\]

If \( X + aY \) separates \( V(I_\mu) \), it also separates \( V(I) \).

The key idea of the proof of Prop. 7 as well as Prop. 11 and 12 is to prove the following inequalities (under the hypothesis that various leading terms do not vanish)
\[
\#V(I_{\mu}) \leq \#V(R_{\mu}(T, a)) \leq \#V(\mathbb{R}(T)) \leq \#V(I)
\]
and argue that the first (resp. last) one is an equality if \( X + aY \) separates \( V(I_\mu) \) (resp. \( V(I) \)), and that the middle one is an equality except for finitely many \( \mu \). We establish these claims in Lemmas 9 and 10. 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 \( I_\mu \) in Section 4.5. We refer to Lemma 16 or 10.3.23 for a proof. Recall that \( P \) and \( Q \) are assumed to be coprime but not \( P_\mu \) and \( Q_\mu \); we address this issue in Lemma 9.

**Lemma 8.** If \( a \in \mathbb{Z} \) is such that \( L_P(a) L_Q(a) \neq 0 \) then \( \#V(R_{\mu}(T, a)) \leq \#V(I) \) and they are equal if and only if \( X + aY \) separates \( V(I) \). The same holds over \( \mathbb{Z}_\mu \), that is for \( P_\mu \), \( Q_\mu \), \( R_\mu \), and \( I_\mu \), provided that \( P_\mu \) and \( Q_\mu \) are coprime.

**Lemma 9.** If \( \phi_\mu(L_P(S)) \phi_\mu(L_Q(S)) \neq 0 \) and \( \mu > 4d^2 \) then \( P_\mu \) and \( Q_\mu \) are coprime in \( \mathbb{Z}_\mu[X, Y] \).

**Proof.** Since \( \phi_\mu(L_P(S)) \phi_\mu(L_Q(S)) \neq 0 \), the property of specialization of resultants (2 Prop. 4.20)[/\ref{footnote:2}] yields that \( \phi_\mu(R_{\mu}(T, S)) = R_{\mu}(T, S) \) and \( \phi_\mu(s) \neq 0 \) implies that \( R_{\mu}(T, S) \neq 0 \). We can thus choose a value \( S = a \in \mathbb{Z}_\mu \) so that \( R_{\mu}(T, a) \neq 0 \) and \( L_{P_\mu}(a) L_{Q_\mu}(a) \neq 0 \); indeed, \( \mu > 4d^2 \) and \( \phi_\mu(L_{R_\mu}(S)) \), \( L_{P_\mu}(S) \), and \( L_{Q_\mu}(S) \) have degree at most \( 2d^2 \) and \( d \) respectively (Lemma 8). For such a value, the resultant of \( P_\mu(T - SY) \) and \( Q_\mu(T - SY) \) is \( R_{\mu}(T, a) \). This resultant is not identically zero, the leading coefficients (in \( Y \)) \( L_{P_\mu}(a) \) and \( L_{Q_\mu}(a) \) do not depend on \( T \) (see Eq. (2)) and are not zero, thus \( P_\mu(T - aY, Y) \) and \( Q_\mu(T - aY, Y) \) are coprime. The result follows.

The following lemma is a direct consequence of the property of specialization of resultants (2 Prop. 4.20) and of the fact that the degree of the gcd cannot decrease when the polynomials are reduced modulo \( \mu \) (16 Lemma 4.8).
Prime numbers such that \( \#V(I) \neq \#V(I_\mu) \). Let \( a \) be such that the form \( X + aY \) separates \( V(I) \) and \( L_P(a), L_Q(a), L_R(a) \neq 0 \). Similarly as in the proof of Prop. 11 we can choose \( a \leq 2d^4 \).

We consider any prime \( \mu \) such that \( \phi_\mu(L_P(a)) \phi_\mu(L_Q(a)) \phi_\mu(L_R(a)) \neq 0 \), so that we can apply Lemmas 9 and 10. Since \( X + aY \) separates \( V(I) \), these lemmas yield that
\[
\#V(I_\mu) \geq d_T(R_\mu(T, a)) = d_T(R(T, a)) = \#V(I).
\]

Now, \( d_T(R(T, a)) = d_T(R(T, a) - d_T(gcd(R(T, a), R'(T, a))) \), and similarly for \( R_\mu(T, a) \). The leading coefficient of \( R(T, S) \) with respect to \( T \) is \( L_\mu(S) \), and since it does not vanish at \( S = a \), \( L_\mu(T) \) is the leading coefficient of \( R(T, a) \). In addition, we have \( R_\mu(T, a) = \phi_\mu(R(T, a)) \), hence 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 \( \mu \) 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 \( \mu \), we have
\[
\#V(I_\mu) \geq d_T(R_\mu(T, a)) = d_T(R(T, a)) = \#V(I).
\]

Since \( \phi_\mu(L_P(a)) \phi_\mu(L_Q(a)) \phi_\mu(L_R(a)) \neq 0 \), we can apply Prop. 11 which yields that \( \#V(I_{\mu i}) \leq \#V(I) \) and thus \( \#V(I_{\mu i}) = \#V(I) \).

Therefore, the primes \( \mu \) such that \( \#V(I_\mu) \neq \#V(I) \) are among those such that \( L_P(a), L_Q(a) \), or \( L_R(a) \) vanishes modulo \( \mu \) 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 \( \mu \). Note that if \( L_P(a), L_Q(a), \) and \( L_R(a) \) do not vanish modulo \( \mu \), then \( L_P(S), L_Q(S), \) and \( L_R(S) \) do not identically vanish modulo \( \mu \). It is straightforward to prove that we can compute an explicit bound, in \( O(d^4 + d\tau) \), on the number of prime divisors of \( L_P(a), L_Q(a), \) or \( L_R(a) \).

Bounding the number of prime \( \mu \) such that the degree of \( gcd(R(T, a), R'(T, a)) \) changes when \( R(T, a) \) and \( R'(T, a) \) are reduced modulo \( \mu \). By 11 Lemma 4.12, given two univariate polynomials in \( \mathbb{Z}[X] \) of degree at most \( d' \) and bitsize at most \( \tau' \), the product of all \( \mu \), such that the degree of the gcd of the two polynomials changes when the polynomials are considered modulo \( \mu \), is bounded by \( (2^{\tau'} + d')^2 \cdot d' + 2 \). The number of such primes \( \mu \) is bounded by the bitsize of this bound, and thus is bounded by \( (d' + 1)(2^{\tau' + \log(d' + 1)}) + 1 \). Here \( d' \leq 2d^4 \) and \( \tau' \) is in \( O(d^4 + d\tau) \) since our explicit bound on the bitsize of \( L_\mu(a) \) holds as well for the bitsize of \( R(T, a) \), and, since \( R(T, a) \) is of degree at most \( 2d^4 \), the bitsize of \( R'(T, a) \) is bounded by that of \( R(T, a) \) plus \( 1 + \log 2d^4 \). Thus we obtain an explicit bound in \( O(d^4 + d\tau) \) on the number of primes \( \mu \) such that the degree of \( gcd(R(T, a), R'(T, a)) \) changes when \( R(T, a) \) and \( R'(T, a) \) are reduced modulo \( \mu \).

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 of \( I_\mu \)

In the following lemma, we analyze the complexity of Algorithm 1 for \( P \) and \( Q \) of degree at most \( d_0 \) in \( X \) and \( d_Y \) in \( Y \) and \( A \) of degree at most \( d_0 \), where \( d_0 \) denotes a bound on the total degree of \( P \) and \( Q \). We will use Algorithm 2 with polynomials with coefficients in \( \mathbb{Z} \) and \( \mathbb{Z}[X] \), and we will only consider its arithmetic complexity in \( \mathbb{F} \). The bit complexity of this algorithm over \( \mathbb{Z} \) is analyzed in [6, Thm. 19] and its arithmetic complexity is thus implicitly analyzed as well; see also [5].

**Algorithm 1** Triangular decomposition

**Input:** \( P, Q \) in \( \mathbb{F}[X, Y] \) coprime such that \( LCP(P) \) and \( LCP(Q) \) are coprime, \( dy(Q) \leq dy(P) \), and \( A \in \mathbb{F}[X] \) squarefree.

**Output:** Triangular decomps. \( \{(A_i(X), B_i(X, Y))\}_{i \in \mathbb{T}} \) such that \( \{(P(A_i(X), B_i(X, Y))\}_{i \in \mathbb{T}} \)

1. Compute the subresultant sequence of \( P \) and \( Q \) with respect to \( Y \): \( B_i = Sres_y(P, Q) \)

2. \( G_0 = gcd(Rres_y(P, Q)) \) and \( T = \emptyset \)

3. for \( i = 1 \) to \( dy(Q) \) do

4. \( G_i = gcd(G_{i-1}, Sres_y(P, Q)) \)

5. \( A_i = G_{i-1}/G_i \)

6. if \( dX(A_i) > 0 \), add \( (A_i, B_i) \) to \( T \)

7. return \( T = \{(A_i(X), B_i(X, Y))\}_{i \in \mathbb{T}} \)

**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 1** (\([9], [11]\)). Algorithm 1 computes a triangular decomposition \( \{(A_i(X), B_i(X, Y))\}_{i \in \mathbb{T}} \) such that

(i) the set \( \{V(P, Q, A)\} \) is the disjoint union of the sets \( \{(A_i(X), B_i(X, Y))\}_{i \in \mathbb{T}} \)

(ii) \( \bigcap_{i \in \mathbb{T}} A_i = \{0\} \)

(iii) \( \forall \alpha \in \{V(A_i(X), B_i(X, Y))\} \) is of degree \( i \) and is equal to \( gcd(P(\alpha, Y), Q(\alpha, Y)) \)

(iv) \( A(X) \) and \( LCP(b(X, Y)) \) are coprime.

In the following lemma, we analyze the complexity of Algorithm 1 for \( P \) and \( Q \) of degree at most \( d_0 \) in \( X \) and \( d_Y \) in \( Y \) and \( A \) of degree at most \( d_0 \), where \( d_0 \) denotes a bound on the total degree of \( P \) and \( Q \). We will use Algorithm 2 with polynomials with coefficients in \( \mathbb{Z} \) and \( \mathbb{Z}[X] \), and we will only consider its arithmetic complexity in \( \mathbb{F} \). The bit complexity of this algorithm over \( \mathbb{Z} \) is analyzed in [6, Thm. 19] and its arithmetic complexity is thus implicitly analyzed as well; see also [5].

**Lemma 14**. Algorithm 1 performs \( O(dx dy^2) = O(d^4) \) arithmetic operations in \( \mathbb{F} \).

**Counting the number of solutions of \( I_\mu \)**. Algorithm 2 computes the number of distinct solutions of an ideal \( I_\mu \).
Algorithm 2 Number of distinct solutions of \( \langle P, Q \rangle \)

Input: \( P, Q \) in \( \mathbb{Z}_\mu[X, Y] \) coprime, \( \mu \) larger than their total degree

Output: Number of distinct solutions of \( \langle P, Q \rangle \)

1. Shear \( P \) and \( Q \) by replacing \( X \) by \( X-bY \) with \( b \in \mathbb{Z}_\mu \) so that \( \text{LCY}(P(X-bY,Y)) \in \mathbb{Z}_\mu \)
2. Triangular decomposition: \( \{ (A_i(X), B_i(X,Y)) \} \) \( i \in I \)
   Algorithm 3 \( (P, Q, \mu) \)
3. for all \( i \in I \) do
   4. \( C_i(X) = \text{LCY}(B_i(X,Y))^{-1} \mod A_i(X) \)
   5. \( \tilde{B}_i(X,Y) = C_i(X)B_i(X,Y) \mod A_i(X) \)
   6. Triangular decomposition: \( \{ (A_{ij}(X), B_{ij}(X,Y)) \} \) \( j \in J_i \)
   Algorithm 4 \( (\tilde{B}_i(X,Y), \frac{\partial B_i(X,Y)}{\partial X}, A_i(X)) \)
7. return \( \sum_{i \in I} \left( i dx(A_i) - \sum_{j \in J_i} j \ dx(A_{ij}) \right) \) 

\( \langle P, Q \rangle \) of \( \mathbb{Z}_\mu[X, Y] \). Roughly speaking, this algorithm first performs one triangular decomposition with the input polynomials \( P \) and \( Q \), 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 \( I_m \) can be read, with a simple formula, from the degrees of the polynomials in the decomposition.

**Lemma 15.** Algorithm 3 computes the number of distinct solutions of \( \langle P, Q \rangle \).

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 generally sheared polynomial \( P(X - SY, Y) \) 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 \( L_{P,b}(S) = \text{LCY}(P(X-SY, SY)) \) 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 13, the triangular decompositions \( \{ (A_i(X), B_i(X,Y)) \} \) \( i \in I \) computed in Line 2 is such that the solutions of \( \{ P(X, Q) \} \) is the disjoint union of the solutions of the \( \{ (A_i(X), B_i(X,Y)) \} \) for \( i \in I \). It follows that the number of (distinct) solutions of \( I_m \) is \( \langle P, Q, \mu \rangle \) is
\[
\#V(I_m) = \sum_{i \in I} \sum_{\alpha \in V(A_i)} dx(B_{ij}(\alpha, Y)).
\]
Since \( B_i(X,Y) \) is a univariate polynomial in \( Y \), \( dx(B_{ij}(\alpha, Y)) \) is equal to \( dy(B_{ij}(\alpha, Y)) \) for all \( \alpha \in V(A_i) \) and \( B_{ij}(\alpha, Y) \) is the derivative of \( B_i(X,Y) \), which is also equal to \( \frac{\partial B_i}{\partial Y}(\alpha, Y) \). By Lemma 13, \( dy(B_i(\alpha, Y)) = i \), and since the degree of the gcd is zero when \( B_i(\alpha, Y) \) is squarefree, we have
\[
\#V(I_m) = \sum_{i \in I} \sum_{\alpha \in V(A_i)} i - \sum_{i \in I} \sum_{\alpha \in V(A_i)} \text{gcd}(B_i(\alpha, Y), \frac{\partial B_i}{\partial Y}(\alpha, Y)).
\]

The polynomials \( A_i(X) \) are squarefree by Lemma 13 so \( \sum_{\alpha \in V(A_i)} i \) is equal to \( i dx(A_i) \).

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 \( \frac{\partial B_i}{\partial Y}(\alpha, Y) \), for every \( i \in I \), which computes a triangular decomposition \( \{ (A_{ij}(X), B_{ij}(X,Y)) \} \) \( j \in J_i \) such that, for \( \alpha \in V(A_{ij}), dy(\text{gcd}(B_i(\alpha, Y), \frac{\partial B_i}{\partial Y}(\alpha, Y))) = j \) (by Lemma 13), which simplifies Equation (5) into
\[
\#V(I_m) = \sum_{i \in I} i dx(A_i) + \sum_{\alpha \in V(A_i)} y j = dx(A_{ij}) \frac{\partial B_i}{\partial Y}(\alpha, Y).
\]
However, we cannot directly apply Algorithm 1 to \( B_i(X,Y) \) and \( \frac{\partial B_i}{\partial Y}(\alpha, Y) \) because their leading coefficients in \( Y \) have no reason to be coprime.

By Lemma 13, \( A_i(X) \) and \( \text{LCY}(B_i(X,Y)) \) are coprime, thus \( \text{LCY}(B_i(X,Y)) \) is invertible modulo \( A_i(X) \) (by Bézout's identity); let \( C_i(X) \) be this inverse and define \( \tilde{B}_i(X,Y) = C_i(X)B_i(X,Y) \text{mod } A_i(X) \) (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 \( C_i(\alpha) \text{LCY}(B_i(\alpha, Y)) = 1 \). Equation (5) can thus be rewritten by replacing \( \tilde{B}_i \) by \( B_i \).

By Lemma 13, for every \( i \in I \), Algorithm 4 computes a triangular decomposition \( \{ (A_{ij}(X), B_{ij}(X,Y)) \} \) \( j \in J_i \) such that \( V((\tilde{B}_i, \frac{\partial \tilde{B}_i}{\partial Y}, A_i)) \) is the disjoint union of the sets \( V(A_{ij}(X), B_{ij}(X,Y)) \), \( j \in J_i \), and for all \( \alpha \in V(A_{ij}) \), \( dy(\text{gcd}(B_i(\alpha, Y), \frac{\partial B_i}{\partial Y}(\alpha, Y))) = j \). 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 \( (\alpha, \beta) \in V((\tilde{B}_i, \frac{\partial \tilde{B}_i}{\partial Y}, A_i)) \) we get
\[
\#V(I_m) = \sum_{i \in I} \sum_{\beta \in V(A_i)} \sum_{j \in J_i} \sum_{\alpha \in V(A_{ij})} j = dx(A_{ij}),
\]
which concludes the proof.

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

**Lemma 16.** Given \( P, Q \) in \( \mathbb{Z}_\mu[X, Y] \) of total degree at most \( d \), Algorithm 2 performs \( \tilde{O}(d^4) \) operations in \( \mathbb{Z}_\mu \).

Proof. According to Lemma 3, the sheared polynomials \( P(T - SY, Y) \) and \( Q(T - SY, Y) \) can be expanded in \( \tilde{O}(d^d + d^4 \pi) \) bit operations in 2. Thus the sheared polynomials \( P(X - SY, Y) \) and \( Q(X - SY, Y) \) can obviously be computed in \( \tilde{O}(d^4) \) arithmetic operations in \( \mathbb{Z}_\mu \). The leading term \( \text{LCY}(P(X - SY, Y)) \) is a polynomial of degree at most \( d \) and a value \( b \in \mathbb{Z}_\mu \) that does not vanish 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 \( \tilde{O}(d^5) \) 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 softly linear in \( d \), this gives a total complexity in \( \tilde{O}(d^4) \) for Line 1.

According to Lemma 13, the triangular decomposition in Line 2 can be done in \( \tilde{O}(d^4) \) arithmetic operations. In Lines 4 and 5, \( C_i(X) \) and \( B_i(X,Y) \) 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 softly linear in the degree of the operands [10 Cor. 11.6], which is \( \tilde{O}(d^2) \) by Lemma 3. The reduction of
Algorithm 3 Number of distinct solutions and lucky prime for \((P, Q)\)

**Input:** \(P, Q \in \mathbb{Z}[X, Y]\) coprime of total degree at most \(d\) and bitwise at most \(\tau\)

**Output:** Number of solutions and lucky prime \(\mu\) for \((P, Q)\)

1: Compute \(P(T - SY, Y), Q(T - SY, Y)\), \(R(T, S) = \text{Res}_T(P(T - SY, Y), Q(T - SY, Y))\)
2: Compute a set of primes larger than \(2d^4\) and of cardinality \(\widetilde{O}(d^4 + d^3\tau)\) that contains a lucky prime for \((P, Q)\)
   (see Prop. 12)
3: for all \(\mu \in B\) do
4: if \(\phi_\mu(L_P(S)) \phi_\mu(L_Q(S)) \phi_\mu(L_R(S)) \neq 0\) then
5: Compute \(N_\mu = \text{Algorithm 2}\phi_\mu(P), \phi_\mu(Q))\)
6: return \((\mu, N_\mu)\) such that \(N_\mu\) is maximum

Algorithm 4 Separating form for \((P, Q)\)

**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 \(\mu\) for \(I\)
2: Compute \(P(T - SY, Y), Q(T - SY, Y)\) and \(R(T, S) = \text{Res}_T(P(T - SY, Y), Q(T - SY, Y))\)
3: Compute \(R_\mu(T, S) = \phi_\mu(R(T, S))\)
4: Compute \(\Upsilon_\mu(S) = \phi_\mu(L_P(S)) \phi_\mu(L_Q(S)) \phi_\mu(L_R(S))\)
5: \(a := 0\)
6: repeat
7: Compute the degree \(N_\mu\) of the squarefree part of \(R_\mu(T, a)\)
8: \(a := a + 1\)
9: until \(\Upsilon_\mu(a) \neq 0\) and \(N_\mu = \#V(I)\)
10: return The linear form \(X + aY\)

We now prove the correctness of the algorithm. First note that for all \(\mu \in B\) satisfying the constraint of Line 1, Lemma 9 implies that \(\phi_\mu(P)\) and \(\phi_\mu(Q)\) are coprime. It follows that Algorithm 2 computes the number of distinct solutions \(N_\mu = \#V(I_\mu)\) of \(I_\mu\). By Prop. 11 and Def. 6, \(N_\mu \leq \#V(I)\) and the equality holds if \(\mu\) is lucky for \(I\).

We now prove the complexity of the algorithm. The polynomials \(P(T - SY, Y), Q(T - SY, Y)\) and their resultant

Algorithm 3 is thus \(O(d^4 + d^3\tau)\) arithmetic operations by Lemma 14. Prop. 12 states that we can compute an explicit bound \(\Xi(d, \tau)\) in \(O(d^3 + d^3\tau)\) on the number of unlucky primes for \((P, Q)\). We want to compute in Line 2 a set 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 first prime numbers is in \(\widetilde{O}(\tau)\) and their maximum is in \(\widetilde{O}(\tau)\) (Thm. 18.10). We can thus compute the set of primes \(B\) with \(O_B(d^4 + d^3\tau)\) bit operations and these primes are in \(\widetilde{O}(d^4 + d^3\tau)\).

In Line 4 we test to zero the reduction modulo \(\mu\) of three polynomials in \(\mathbb{Z}[S]\) which have been computed in Line 1 and which are of degree \(O(d^3)\) and bitwise \(O(d^4 + d\tau)\) in the worst case (by Lemma 5). For each of these polynomials, the test to zero can be done by first computing (once for all) the gcf of its \(O(d^3)\) integer coefficients of bitwise \(O(d^4 + d\tau)\). Each gcf can be computed with a bit complexity that is softly linear in the bitwise of the integers \([15, \text{Thm. 12.6}]\) (and the bitsize clearly does not increase), hence all the gcf's can be done with a bit complexity of \(\widetilde{O}_B(d^4 + d^3\tau)\). Then the reduction of each of the three gcf's modulo \(\mu\) is performed, for each of the \(\widetilde{O}(d^4 + d\tau)\) choices of \(\mu\), in a bit complexity that is softly linear in the maximum bitsize, that is \(\widetilde{O}(d^4 + d\tau)\)

Thm. 9.8 since \(\mu\) has bitsize in \(O(\log(d^4 + d^3\tau))\). Hence, the tests in Line 4 can be done with a total bit complexity in \(\widetilde{O}_B((d^4 + d^3\tau)(d^4 + d\tau)) = \widetilde{O}_B(d^4 + d^3\tau)\).

In Line 5 we compute, for \(O(d^4 + d^3\tau)\) prime numbers \(\mu, \phi_\mu(P)\) and \(\phi_\mu(Q)\) and call Algorithm 2 to compute the number of their common solutions. For every \(\mu\), the computation of \(\phi_\mu(P)\) and \(\phi_\mu(Q)\) can be done with \(O_B(d^4 + d\tau)\) bit operations, since the reduction modulo \(\mu\) of each of the \(O(d^3)\) coefficients is softly linear in its bitsize. By Lemma 16, the bit complexity of Algorithm 2 is in \(O_B(d^4)\). Hence, the total bit complexity of Line 5 is in \(O_B(d^4 + d^3\tau + d^3\tau^2)\), and so is the overall bit complexity of Algorithm 3.
We now focus on the complexity of the algorithm. By Lemma 17, the bit complexity of evaluating in $\mathbb{Z}_{\mu}(S)$ and we consider $\Upsilon_{\mu}(a)$ in $\mathbb{Z}_{\mu}$.

5. REFERENCES


