HAL
archives-ouvertes.fr

Separating linear forms and Rational Univariate Representations of 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 and Rational Univariate Representations of bivariate systems. Journal of Symbolic Computation, Elsevier, 2015, 68, pp.84-119. .

HAL Id: hal-00977671

https://hal.inria.fr/hal-00977671

Submitted on 11 Apr 2014

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 and Rational Univariate Representations of bivariate systems*

Yacine Bouzidi[a,b], Sylvain Lazard[a,b], Marc Pouget[a,b], Fabrice Rouillier[a,c]

[a] *Inria, France.*
[b] *LORIA laboratory, Nancy, France.*
[c] *Institut de Mathématiques de Jussieu, Paris, France.*

Abstract

We address the problem of solving systems of bivariate polynomials with integer coefficients. We first present an algorithm for computing a separating linear form of such systems, 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.

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 of bitsize $O(\log d)$ in $\widetilde{O}_B(d^8 + d^7\tau)$ bit operations in the worst case, which decreases by a factor d^2 the best known complexity for this problem (where $\widetilde{O}$ refers to the complexity where polylogarithmic factors are omitted and O_B refers to the bit complexity).

We then present simple polynomial formulas for the Rational Univariate Representations (RURs) of such systems. This yields that, given a separating linear form of bitsize $O(\log d)$, the corresponding RUR can be computed in worst-case bit complexity $\widetilde{O}_B(d^7 + d^6\tau)$ and that its coefficients have bitsize $\widetilde{O}(d^2 + d\tau)$. We show in addition that isolating boxes of the solutions of the system can be computed from the RUR with $\widetilde{O}_B(d^8 + d^7\tau)$ bit operations in the worst case. Finally, we show how a RUR can be used to evaluate the sign of a bivariate polynomial (of degree at most d and bitsize at most τ) at one real solution of the system in $\widetilde{O}_B(d^8 + d^7\tau)$ bit operations and at all the $\Theta(d^2)$ real solutions in only $O(d)$ times that for one solution.

Keywords: Bivariate system, Separating Linear Form, Rational univariate representation

*A preliminary version of this article has been published in the 2013 *International Symposium on Symbolic and Algebraic Computation*.

Email addresses: Yacine.Bouzidi@inria.fr (Yacine Bouzidi), Sylvain.Lazard@inria.fr (Sylvain Lazard), Marc.Pouget@inria.fr (Marc Pouget), Fabrice.Rouillier@inria.fr (Fabrice Rouillier)

1. Introduction

In this paper, we address the problem of solving systems of *bivariate* polynomials with integer coefficients and we focus on the worst-case bit complexity of these methods (in the RAM model). We consider throughout the paper input polynomials of total degree at most d with integer coefficients of bitsize at most τ.

There exists many algorithms, in the literature, for "solving" algebraic systems of equations. Some focus on computing "formal solutions" such as rational parameterizations, Gröbner bases, and triangular sets, others focus on computing numerical approximations of the solutions. Such numerical approximations can be computed from formal solutions or directly from the input system using numerical methods such as subdivision or homotopy techniques. In this paper, we are interested in certified numerical approximations or, more precisely, isolating boxes of the solutions, that is axis-parallel boxes sets such that every real solution lies in a unique box and conversely.

It should be stressed that formal solutions do not necessarily yield, directly, isolating boxes of the solutions. In particular, from a theoretical complexity point of view, it is not proved that the knowledge of a triangular system or Gröbner basis of a system always simplifies the isolation of its solutions. The difficulty lies in the fact that isolating the solutions of a triangular system essentially amounts to isolating the roots of univariate polynomials with algebraic numbers as coefficients, which is not trivial when these polynomials have multiple roots. For recent work on this problem, we refer to Cheng et al. (2007), Boulier et al. (2009), Strzebonski and Tsigaridas (2011) and references therein. This difficulty also explains why it is not an easy task to precisely define what a formal solution of a system is, and why usage prevails in what is usually considered to be a formal solution.

One important approach, which can be traced back to Kronecker, for solving 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 some 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 Alonso et al. (1996), González-Vega and El Kahoui (1996), Rouillier (1999), Giusti et al. (2001), Bostan et al. (2003), Diochnos et al. (2009) 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, probabilistic Monte-Carlo algorithms can overlook this issue. In a deterministic setting, a separating linear form can easily be computed by considering a direction whose slope is larger than twice the ratio of an upper bound on the absolute values of the y-coordinates of the solutions over a lower bound on the distance between two consecutive x-coordinates of the solutions (see Cheng et al. (2009) for an adaptive version); however, this defines a change of variables that involves integers of bitsize $\Theta(d^3\tau)$ in the worst case,[1] which increases dramatically the bit complexity of the sheared polynomials and that of all

[1] The $\Theta(d^3\tau)$ bound follows from the Cauchy bound and the root separation bound of the resultant of the two input

subsequent computations (see e.g. Proposition 24). Surprisingly, in a deterministic setting, computing a linear separating form of small bitsize is the current bottleneck in the computation of parameterizations for bivariate systems, as discussed below, and this is thus a critical problem.

For systems of two bivariate polynomials of total degree at most d with integer coefficients of bitsize at most τ, the approach with best known worst-case bit complexity for computing a rational parameterization was first introduced by González-Vega and El Kahoui (1996) (see also González-Vega and Necula (2002)). Their algorithm first computes a separating linear form, then shears accordingly the two input polynomials, and computes a rational parameterization using the subresultant sequence of the sheared polynomials. Their initial analysis[2] of $\widetilde{O}_B(d^{16} + d^{14}\tau^2)$ was improved by Diochnos et al. (2009, Lemma 16 & Theorem 19)[3] to (i) $\widetilde{O}_B(d^{10} + d^9\tau)$ for computing a separating linear form and to (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. Computing a separating linear form is also a (non-strict) bottleneck when considering the additional phase of computing isolating boxes of the solutions.[3]

Note that, depending on the context, isolating boxes of the solutions may be sufficient and a rational parameterization of the solutions may not be needed. Then, for a system of two bivariate polynomials, the best known algorithm has complexity $\widetilde{O}_B(d^8 + d^7\tau)$ (Emeliyanenko and Sagraloff, 2012). Furthermore, the isolating boxes can easily be refined because the algorithm isolates the roots of the resultants of the two input polynomials with respect to each of the variables.

Main results. Our first 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 of bitsize $O(\log d)$ for a system of two bivariate polynomials of total degree at most d and integer coefficients of bitsize at most τ (Theorem 19). As discussed above, this decreases by a factor d^2 the best known complexity for this problem.

As a direct consequence, the overall bit complexity of computing a rational parameterization in the approach of González-Vega and El Kahoui (1996) decreases to $\widetilde{O}_B(d^8 + d^7\tau)$ (Diochnos et al., 2009).

We also consider the alternative Rational Univariate Representation (RUR for short) of Rouillier (1999). Although the parameterization of Gonzalez-Vega et al. consists in the worst case of $\Theta(d)$ univariate polynomials and their associated rational one-to-one mappings (that send the roots of the univariate polynomials to the solutions of the system), a RUR consists of a *single* univariate polynomial and its associated rational one-to-one mappings $t \mapsto (\frac{f_x(t)}{f_1(t)}, \frac{f_y(t)}{f_1(t)})$ defined by three polynomials. We show that (i) the RUR can be expressed with simple polynomial formulas, that (ii) it has a total bitsize which is asymptotically smaller than that of Gonzalez-Vega and El Kahoui by a factor d, and that (iii) it can be computed with the same complexity, that is $\widetilde{O}_B(d^7 + d^6\tau)$ (Theorem 22). Specifically, we prove that the four polynomials of the RUR have degree at most d^2 and bitsize $\widetilde{O}(d^2 + d\tau)$. Comparatively, the bounds on degrees and bitsizes of

polynomials; see e.g. Yap (2000, §6.2) and Lemmas 3 and 34.

[2] In González-Vega and El Kahoui (1996), the complexity of computing a separating form is in $\widetilde{O}_B(d^{16} + d^{14}\tau^2)$ (Lemma 4.4) and the complexity of computing a parameterization is in $\widetilde{O}_B(d^{10}\tau^2)$ (Lemma 4.1 and proof of Lemma 4.2).

[3] The overall bit complexity stated in Diochnos et al. (2009, Theorem 19) is $\widetilde{O}_B(d^{12} + d^{10}\tau^2)$ because it includes the isolation of the solutions of the system. Note, however, that the complexity of the isolation phase, and thus of the whole algorithm, trivially decreases to $\widetilde{O}_B(d^{10} + d^9\tau)$ using Pan (2002) results on the complexity of isolating the real roots of a univariate polynomial.

the polynomials in the parameterization of Gonzalez-Vega et al. are the same but, in the worst case, there are $\Theta(d)$ univariate polynomials instead of four. Moreover, we prove that this bound holds for any ideal containing P and Q (Proposition 28). Note that specializing the general result of Rouillier (1999, Proposition 4.1) to two variables gives the bounds $\widetilde{O}_B(D^5L)$ for the computation of the RUR (knowing a separating form) and $O(D^2L)$ for the bitsize of its coefficients, where $D = O(d^2)$ is the dimension of the quotient algebra and $L = O(D\tau')$ is the maximum bitsize of the coefficients in the multiplication tensor of the algebra, where $\tau' = \widetilde{O}(d^2\tau)$ is the maximum bitsize of a Gröbner basis of the input system (Lazard, 1983). Note that in the special case of radical systems, specializing the result of Dahan and Schost (2004, Theorem 1) to two variables yields a better bound in $\widetilde{O}(d^2\tau)$ for the bitsize of the RUR.

We also show that, given a RUR, isolating boxes of the solutions of the system can be computed with $\widetilde{O}_B(d^8 + d^7\tau)$ bit operations (Proposition 35). This decreases by a factor d^2 the best known complexity for this isolation phase of the algorithm (see the discussion above). Globally, this brings the overall bit complexity of all three phases of the solving algorithm, that computing (i) a separating linear form, (ii) a RUR, and (iii) isolating boxes, to $\widetilde{O}_B(d^8 + d^7\tau)$, which also improves by a factor d^2 the complexity. Note that this complexity matches the state-of-the-art complexity of Emeliyanenko and Sagraloff (2012) for computing isolating boxes, but our algorithm computes a rational parameterization as well as isolating boxes.

Finally, we show how a rational parameterization can be used to perform efficiently two important operations on the input system. We first show how a RUR can be used to perform efficiently the *sign_at* operation. Given a polynomial F of total degree at most d with integer coefficients of bitsize at most τ, we show that the sign of F at one real solution of the system can be computed in $\widetilde{O}_B(d^8 + d^7\tau)$ bit operations, while the complexity of computing its sign at all the $\Theta(d^2)$ solutions of the system is only $O(d)$ times that for one real solution (Theorem 40). This improves the best known complexities of $\widetilde{O}_B(d^{10} + d^9\tau)$ and $\widetilde{O}_B(d^{12} + d^{11}\tau)$ for these respective problems; see Diochnos et al. (2009, Th. 14 & Cor. 24) with the improvement of Sagraloff (2012) for the root isolation. Similar to the *sign_at* operation, we show that a RUR can be split in two parameterizations such that F vanishes at all the solutions of one of them and at none of the other. We also show that these rational parameterizations can be transformed back into RURs in order to reduce their total bitsize, within the same complexity, that is, $\widetilde{O}_B(d^8 + d^7\tau)$ (Proposition 44).

The paper is organized as follows. We introduce notation and recall classical material in Section 2. We present our results on separating linear forms in Section 3, those on the computation and bitsize of RURs in Section 4, and address in Section 5 the applications of the RURs on the isolation of real solutions, sign_at operations, and over-constrained systems.

2. 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 stands for 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. We refer to τ_γ as the bitsize of a polynomial, rational or integer γ. 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 the fraction of field 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 is 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 7. 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:

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

2.1. Subresultant sequences

We recall here the definition of subresultant sequences and some related properties. Note that we only use subresultants in Section 3.4.1 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 $Y^{p+q-1}, \ldots, Y, 1$.

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

Definition 1 (El Kahoui (2003, §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 $\mathbb{D}$) (Basu et al., 2006, §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 3.4.1. For clarity, we state this property for 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 El Kahoui (2003, 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] 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.

2.2. Complexity

We recall complexity results, using fast algorithms, on subresultants and gcd computations. We also analyze complexities related to the evaluation of a univariate polynomial at a given rational and the computation of the "sheared" polynomials and their resultant.

Lemma 3 (Basu et al. (2006, Proposition 8.46), Reischert (1997, §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 will 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 $\overline{P}$, provided that the characteristic of the coefficients ring is zero or sufficiently large (e.g., larger than the degree of P).

Lemma 4 (Basu et al. (2006, Corollary 10.12 & Remark 10.19)[5])**.** *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 $\mathbb{F}$. If P, $Q \in \mathbb{Z}[X]$ have degree at most d and bitsize at most τ, a gcd in $\mathbb{Z}[X]$ with coefficients of bitsize in $O(d + \tau)$ can be computed with $\widetilde{O}_B(d^2\tau)$ bit operations. The same bounds hold for the bitsize and the computation of the gcd-free part of P with respect to Q.*

The following is a refinement of the previous lemma in the case of two polynomials with different degrees and bitsizes. It is a straightforward adaptation of Lickteig and Roy (2001, Corollary 5.2) and it is only used in Section 5.3.

Lemma 5 (Lickteig and Roy (2001)[6])**.** *Let P and Q be two polynomials in $\mathbb{Z}[X]$ of degrees p and q and of bitsizes τ_P and τ_Q, respectively. A gcd of P and Q of bitsize $O(\min(p + \tau_P, q + \tau_Q))$*

[5] Basu et al. (2006, Corollary 10.12) states that P and Q have a gcd in $\mathbb{Z}[X]$ with bitsize in $O(d + \tau)$. Basu et al. (2006, Remark 10.19) claims that a gcd and gcd-free parts of P and Q can be computed in $\widetilde{O}_B(d^2\tau)$ bit operations. This remark refers to Lickteig and Roy (2001, Corollary 5.2) which proves that the last non-zero Sylvester-Habicht polynomial, which is a gcd of P and Q (Basu et al., 2006, Corollary 8.32), can be computed in $\widetilde{O}_B(d^2\tau)$ bit operations. Moreover, the corollary proves that the Sylvester-Habicht transition matrices can be computed within the same bit complexity, which gives the cofactors of P and Q in the sequence of the Sylvester-Habicht polynomials (i.e., $U_i, V_i \in \mathbb{Z}[X]$ such that $U_iP + V_iQ$ is equal to the i-th Sylvester-Habicht polynomials). The gcd-free part of P with respect to Q and conversely are the cofactors corresponding to the one-after-last non-zero Sylvester-Habicht polynomial (Basu et al., 2006, Proposition 10.14), and can thus be computed in $\widetilde{O}_B(d^2\tau)$ bit operations. The gcd (resp. gcd-free part) of P and Q computed this way is in $\mathbb{Z}[X]$, thus dividing it by the gcd of its coefficients yields a gcd (resp. gcd-free part) of P and Q of smallest bitsize in $\mathbb{Z}[X]$ which is known to be in $O(d + \tau)$. The gcd of the coefficients, which are of bitsize $\widetilde{O}(d\tau)$ (Basu et al., 2006, Proposition 8.46), follows from $O(d)$ gcds of two integers of bitsize $\widetilde{O}(d\tau)$ and each such gcd can be computed with $\widetilde{O}_B(d\tau)$ bit operations (Yap, 2000, §2.A.6). Therefore, a gcd (resp. gcd-free part) of P and Q of bitsize $O(d + \tau)$ can be computed in $\widetilde{O}_B(d^2\tau)$ bit complexity.

[6] The algorithm in Lickteig and Roy (2001) uses the well-known half-gcd approach to compute any polynomial in the Sylvester-Habicht and cofactors sequence in a softly-linear number of arithmetic operations, and it exploits Hadamard's bound on determinants to bound the size of intermediate coefficients. When the two input polynomials have different degrees and bitsizes, Hadamard's bound reads as $\widetilde{O}(p\tau_Q + q\tau_P)$ instead of simply $\widetilde{O}(d\tau)$ and, similarly as in Lemma 4, the algorithm in Lickteig and Roy (2001) yields a gcd and gcd-free parts of P and Q in $\widetilde{O}_B(\max(p, q)(p\tau_Q + q\tau_P))$ bit operations. Furthermore, the gcd and gcd-free parts computed this way are in $\mathbb{Z}[X]$ with coefficients of bitsize $\widetilde{O}(p\tau_Q + q\tau_P)$, thus, dividing them by the gcd of their coefficients can be done with $\widetilde{O}_B(\max(p, q)(p\tau_Q + q\tau_P))$ bit operations and yields a gcd and gcd-free parts in $\mathbb{Z}[X]$ with minimal bitsize, which is as claimed by Mignotte's bound; see e.g. Basu et al. (2006, Corollary 10.12).

in $\mathbb{Z}[X]$, can be computed in $\widetilde{O}_B(\max(p,q)(p\tau_Q + q\tau_P))$ bit operations. A gcd-free part of P with respect to Q, of bitsize $O(p+\tau_P)$ in $\mathbb{Z}[X]$, can be computed in the same bit complexity.

We now state a bound on the complexity of evaluating univariate polynomials; although this bound is ought to be known and straightforward in a divide-and-conquer scheme, we were not able to find a proper reference for it; see Bodrato and Zanoni (2011) and Hart and Novocin (2011) for recent references on the subject. For completeness, we provide a short and simple proof.

Lemma 6. *Let a be a rational of bitsize τ_a, the evaluation at a of a univariate polynomial f of degree d and rational coefficients of bitsize τ can be done in $\widetilde{O}_B(d(\tau+\tau_a))$ bit operations, while the value $f(a)$ has bitsize in $O(\tau + d\tau_a)$.*

Proof. The complexity $\widetilde{O}_B(d(\tau+\tau_a))$ can easily be obtained by recursively evaluating the polynomial $\sum_{i=0}^{d} a_i\, x^i$ as $\sum_{i=0}^{d/2} a_i\, x^i + x^{d/2} \sum_{i=1}^{d/2} a_{i+d/2}\, x^i$. Evaluating $x^{d/2}$ can be done in $O_B(d\tau_a \log^3 d\tau_a)$ time by recursively computing $\log \frac{d}{2}$ multiplications of rational numbers of bitsize at most $d\tau_a$, each of which can be done in $O_B(d\tau_a \log d\tau_a \log\log d\tau_a)$ time by Schönhage-Strassen algorithm; see e.g. von zur Gathen and Gerhard (2003, Theorem 8.24). $\sum_{i=0}^{d/2} a_{i+d/2}\, a^i$ has bitsize at most $d\tau_a+\tau$, hence its multiplication by $a^{d/2}$ can be done in $O_B((d\tau_a+\tau)\log^2(d\tau_a+\tau))$ time. Hence, the total complexity of evaluating f is at most $T(d,\tau,\tau_a) = 2T(d/2,\tau,\tau_a)+O_B((d\tau_a+\tau)\log^3(d\tau_a+\tau))$ which is in[7] $O_B(d(\tau_a+\tau)\log^4(d\tau_a+\tau))$ that is in $\widetilde{O}_B(d(\tau_a+\tau))$. □

Lemma 7. *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 $R(T,S)$. □

[7] Indeed, $T(d,\tau,\tau_a) = 2^{i+1}T(\frac{d}{2^{i+1}},\tau,\tau_a) + O_B((d\tau_a+\tau)\log^3(d\tau_a+\tau)+\cdots+2^i(\frac{d}{2^i}\tau_a+\tau)\log^3(\frac{d}{2^i}\tau_a+\tau))$
$\leqslant O_B(d\tau_a\log^3(d\tau_a+\tau)\log d + \tau\log^3(d\tau_a+\tau)\sum_{i=0}^{\log d}2^i)$
$\leqslant O_B(d(\tau_a+\tau)\log^4(d\tau_a+\tau))$.

3. Separating linear form

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.[8] 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).

3.1. Overview

We first outline a classical algorithm which is essentially the same as those proposed, for instance, in Diochnos et al. (2009, Lemma 16) and Kerber and Sagraloff (2012, Theorem 24)[9] 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 (Van der Waerden, 1930).

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 10). 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

[8] Note that the assumption that $I = \langle P, Q\rangle$ is zero-dimensional or equivalently that P and Q are coprime is implicitly tested during Algorithm 4 because they are coprime if and only if $R(T, S)$ does not identically vanish.

[9] Kerber and Sagraloff (2012, Theorem 24) states a complexity of $\widetilde{O}_B(d^9\tau)$ instead of $\widetilde{O}_B(d^{10} + d^9\tau)$ because the fact that sheared polynomials have bitsize $\widetilde{O}(d + \tau)$ (see Lemma 7) instead of $\widetilde{O}(\tau)$ had been missed. This was corrected in the version in the arXiv (see http://arxiv.org/abs/1104.1510).

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 3.2, a critical property (Proposition 9) 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 3.3 to 3.5 how to compute such a lucky prime number. We do that by first proving in Section 3.3 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 12) and then by computing a bound on the number of unlucky primes (Proposition 13). 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 3.4 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 3.5, 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 3.6). 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 19).

3.2. 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. Recall that the solutions are considered over the algebraic closure of the fraction field, $\mathbb{Z}_\mu$ or $\mathbb{Q}$, of the ring of coefficients. 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 8. *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*

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

Note that we consider μ in $\Omega(d^4)$ in Definition 8 because, in Algorithm 4, we want to ensure that there exists, for I_μ (resp. I), a separating form $X + aY$ *with* $a \in \mathbb{Z}_\mu$ (resp. $0 \leqslant a < \mu$ in $\mathbb{Z}$). The constant 2 in the bound $2d^4$ is an overestimate, which simplifies the proof of Proposition 12.

Proposition 9. *Let μ be a lucky prime for the ideal $I = \langle P, Q\rangle$ and let $a < \mu$ be an integer*[10] *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 9, as well as Propositions 12 and 13, is to prove the following inequalities (under the hypothesis that various leading terms do not vanish)

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

[10] 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 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 10 and 11. As mentioned in Section 3.1, Lemma 10 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 3.6. For completeness, we outline its proof; see Diochnos et al. (2009, Lemma 16) or Basu et al. (2006, Proposition 11.23) for details. Recall that P and Q are assumed to be coprime but not P_μ and Q_μ.

Lemma 10. *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 Cox et al. (1997, §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 $V(I)$, 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 11. *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 $\phi_\mu(R(T, S)) = Res_Y(\phi_\mu(P(T - SY, Y)), \phi_\mu(Q(T - SY, Y)))$ (Basu et al., 2006, 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 9. 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 10 and 11, 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 10, $X + aY$ separates $V(I)$. □

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

As shown in Proposition 9, 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 12. *Let $I = \langle P, Q \rangle$ be a zero-dimensional ideal in $\mathbb{Z}[X, Y]$. If a prime μ is larger*[11] *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 10 and 11 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 13. *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 8, 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 12 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 10 and 11 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 12 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 $L_P(a)\,L_Q(a)\,L_R(a) \neq 0$.[12] Similarly as in the proof of Proposition 12, 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 10 and 11, 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 $V(I)$.

[11] The constraint $\mu > 2d^4$ could be removed by proving that $\#V(I_\mu) = d_T(\overline{R_\mu(T, S)}) \leqslant d_T(\overline{R(T, S)}) = \#V(I)$ without specializing S at a (which requires generalizing Lemma 11 to bivariate polynomials).

[12] It can be shown that $L_P(a)\,L_Q(a) \neq 0$ implies $L_R(a) \neq 0$ (see Lemma 27) but this property does not simplify the proof.

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 $\phi_\mu(R(T,a)) = Res_Y(\phi_\mu(P(T-aY,Y)),\phi_\mu(Q(T-aY,Y)))$ (Basu et al., 2006, 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 12 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 $L_R(S)$.

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 Basu et al. (2006, 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 Yap (2000, 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[13] by $(2^{\tau'}\sqrt{d'+1})^{2d'+2}$. As noted above, the number of such primes μ is bounded by the bitsize of

[13] Yap (2000, 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 (González-Vega and El Kahoui, 1996; Li et al., 2011)

Input: P, Q in $\mathbb{F}[X, Y]$ coprime such that $Lc_Y(P)$ and $Lc_Y(Q)$ are coprime,[14] $d_Y(Q) \leqslant d_Y(P)$, 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** $\mathcal{T} = \{(A_i(X), B_i(X, Y))\}_{i\in\mathcal{I}}$

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 $2d^4$. □

3.4. 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.

3.4.1. 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 (Lazard, 1992; Li et al., 2011) and for the computation of topology of curves (González-Vega and El Kahoui, 1996).

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 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 14 (González-Vega and El Kahoui (1996); Li et al. (2011))**.** *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 Diochnos et al. (2009, Theorem 19) and its arithmetic complexity is thus implicitly analyzed as well; for clarity, we provide here a short proof.

Lemma 15. *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 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 $\widetilde{O}(d_X d_Y^3) = \widetilde{O}(d^4)$. □

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

We present here Algorithm 2, which 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 (see e.g. Aubry (1999)) 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 16. *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

[14] The hypothesis that $Lc_Y(P)$ and $Lc_Y(Q)$ are coprime can be relaxed by applying the algorithm recursively; see Li et al. (2011) 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.: $\{(A_{ij}(X), B_{ij}(X, Y))\}_{j\in\mathcal{J}_i}$ = 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** $\sum_{i\in\mathcal{I}} \left(i\, d_X(A_i) - \sum_{j\in\mathcal{J}_i} j\, d_X(A_{ij})\right)$

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 14, 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 14, $d_Y(B_i(\alpha, Y)) = i$, 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 14, 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 14), 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 14, $A_i(X)$ 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 14, 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 14) 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 17. *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 7, the sheared polynomials $P(T - S\,Y, Y)$ and $Q(T - S\,Y, Y)$ can be expanded in $\widetilde{O}_B(d^4 + d^3\tau)$ bit operations in $\mathbb{Z}$. Thus the sheared polynomials $P_\mu(X - S\,Y, Y)$ and $Q_\mu(X - S\,Y, Y)$ can obviously be computed in $\widetilde{O}(d^4)$ arithmetic operations in $\mathbb{Z}_\mu$.[15] The leading term $Lc_Y(P_\mu(X - S\,Y, Y)) \in \mathbb{Z}_\mu[S]$ 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.

According to Lemma 15, 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 (von zur Gathen and Gerhard, 2003, 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 (von zur Gathen and Gerhard, 2003, 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 (von zur Gathen and Gerhard, 2003, 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 15. 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 14, 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. □

[15] 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 13)
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

3.5. 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 18. *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 12 and Definition 8, $N_\mu \leqslant \#V(I)$ and the equality holds if μ is lucky for I. Since the set B of considered primes 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 7.

Proposition 13 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 $\widetilde{O}(r)$ (von zur Gathen and Gerhard, 2003, 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 7). 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 (Moenck and Borodin, 1974, 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 $O(\log d\tau)$. 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 $O(\log d\tau)$. Hence, the bit complexity of Line 4 is $\widetilde{O}_B(d^4 + d^3\tau)$.

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** $\Upsilon_\mu(a) \neq 0$[16] and $N_a = \#V(I)$
10: **return** The linear form $X + aY$

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 17 (since μ has bitsize $O(\log d\tau)$). The overall bit complexity of the algorithm is thus in $\widetilde{O}_B(d^8 + d^7\tau)$. □

3.6. 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 19. *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 $\widetilde{O}_B(d^8 + d^7\tau)$.*

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 (Basu et al., 2006, 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 10 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 9, that

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

$X + aY$ 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 18, 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 7, $R(T, S)$ 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 (von zur Gathen and Gerhard, 2003, 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 18).[17] 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 (von zur Gathen and Gerhard, 2003, 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 7). 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 18. 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 thus to a total bit complexity for the algorithm in $\widetilde{O}_B(d^8 + d^7\tau)$. □

4. Rational Univariate Representation

The idea of this section is to express the polynomials of a RUR of two polynomials in terms of a resultant defined from these polynomials. Given a separating form, this yields a new algorithm to compute a RUR and it also enables us to derive the bitsize of the polynomials of a RUR. In Section 4.1, we prove these expressions for the polynomials of a RUR and present the corresponding algorithm. We prove the bound on the bitsize of the RUR in Section 4.2. These results are summarized in Theorem 22.

Throughout this section we assume that the two input polynomials P and Q are coprime in $\mathbb{Z}[X, Y]$, that their maximum total degree d is at least 2 and that their coefficients have maximum bitsize τ.

We first recall the definition and main properties of Rational Univariate Representations. In the following, for any polynomial $v \in \mathbb{Q}[X, Y]$ and $\sigma = (\alpha, \beta) \in \mathbb{C}^2$, we denote by $v(\sigma)$ the image of σ by the polynomial function v (e.g. $X(\alpha, \beta) = \alpha$).

[17] 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 7) 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 $O(\log(d^4 + d^3\tau))$.

Definition 20 (Rouillier (1999, Definition 3.3)). *Let $I \subset \mathbb{Q}[X,Y]$ be a zero-dimensional ideal, $V(I) = \{\sigma \in \mathbb{C}^2, v(\sigma) = 0, \forall v \in I\}$ its associated variety, and a linear form $T = X + aY$ with $a \in \mathbb{Q}$. The RUR-candidate of I associated to $X + aY$ (or simply, to a), denoted $RUR_{I,a}$, is the following set of four univariate polynomials in $\mathbb{C}[T]$*

$$\begin{aligned} f_{I,a}(T) &= \prod_{\sigma \in V(I)} (T - X(\sigma) - aY(\sigma))^{\mu_I(\sigma)} \\ f_{I,a,v}(T) &= \sum_{\sigma \in V(I)} \mu_I(\sigma) v(\sigma) \prod_{\varsigma \in V(I), \varsigma \neq \sigma} (T - X(\varsigma) - aY(\varsigma)), \quad \text{for } v \in \{1, X, Y\} \end{aligned} \tag{6}$$

where, for $\sigma \in V(I)$, $\mu_I(\sigma)$ denotes the multiplicity of σ in I. If $(X,Y) \mapsto X + aY$ is injective on $V(I)$, we say that the linear form $X + aY$ separates $V(I)$ (or is separating for I) and $RUR_{I,a}$ is called a RUR (the RUR of I associated to a).

The following lemma states fundamental properties of RURs, which are all straightforward from the definition except for the fact that the RUR polynomials have rational coefficients (Rouillier, 1999, Theorem 3.1).

Lemma 21. *If $I \subset \mathbb{Q}[X,Y]$ is a zero-dimensional ideal and $a \in \mathbb{Q}$, the four polynomials of the RUR-candidate $RUR_{I,a}$, have rational coefficients. Furthermore, if $X + aY$ separates $V(I)$, the following mapping between $V(I)$ and $V(f_{I,a}) = \{\gamma \in \mathbb{C}, f_{I,a}(\gamma) = 0\}$*

$$\begin{array}{ccc} V(I) & \to & V(f_{I,a}) \\ (\alpha, \beta) & \mapsto & \alpha + a\beta \\ \left(\dfrac{f_{I,a,X}}{f_{I,a,1}}(\gamma), \dfrac{f_{I,a,Y}}{f_{I,a,1}}(\gamma)\right) & \leftarrow\!\shortmid & \gamma \end{array}$$

is a bijection, which preserves the real roots and the multiplicities.

We prove in this section the following theorem on the RUR of two polynomials. We state it for any separating linear form $X + aY$ with integer a of bitsize $O(\log d)$. Recall that there exists a separating form $X + aY$ with a positive integer $a < 2d^4$, which can be computed in $\widetilde{O}_B(d^8 + d^7\tau)$ bit operations (Theorem 19). Theorem 22 is a direct consequence of Propositions 24 and 28.

Theorem 22. *Let $P, Q \in \mathbb{Z}[X,Y]$ be two coprime bivariate polynomials of total degree at most d and maximum bitsize τ. Given a separating form $X + aY$ with integer a of bitsize $O(\log d)$, the RUR of $\langle P, Q\rangle$ associated to a can be computed using Proposition 23 with $\widetilde{O}_B(d^7 + d^6\tau)$ bit operations. Furthermore, the polynomials of this RUR have degree at most d^2 and bitsize in $\widetilde{O}(d^2 + d\tau)$.*

4.1. RUR computation

We show here that the polynomials of a RUR can be expressed as combinations of specializations of the resultant R and its partial derivatives. The seminal idea has already been used by several authors in various contexts for computing rational parameterizations of the radical of a given zero-dimensional ideal and mainly for bounding the size of a Chow form; see e.g. Canny (1987), Alonso et al. (1996) or Schost (2001). Based on the same idea but keeping track of multiplicities, we present a simple new formulation for the polynomials of a RUR, given a separating form.

Proposition 23. *For any rational a such that $L_P(a)L_Q(a) \neq 0$ and such that $X+aY$ is a separating form of $I = \langle P, Q\rangle$, the RUR of $\langle P, Q\rangle$ associated to a is as follows:*

$$f_{I,a}(T) = \frac{R(T,a)}{L_R(a)} \qquad f_{I,a,1}(T) = \frac{f'_{I,a}(T)}{\gcd(f_{I,a}(T), f'_{I,a}(T))}$$

$$f_{I,a,Y}(T) = \frac{\frac{\partial R}{\partial S}(T,a) - f_{I,a}(T)\frac{\partial L_R}{\partial S}(a)}{L_R(a)\gcd(f_{I,a}(T), f'_{I,a}(T))} \qquad f_{I,a,X}(T) = T f_{I,a,1}(T) - d_T(f_{I,a})\overline{f_{I,a}(T)} - a f_{I,a,Y}(T).$$

We postpone the proof of Proposition 23 to Section 4.1.1 and first analyze the complexity of the computation of the expressions therein.

Proposition 24. *The computation of the polynomials in Proposition 23 can be done with $\widetilde{O}_B(d^7 + d^6(\tau + \tau_a))$ bit operations, where τ_a is the bitsize of a.*

Proof of Proposition 24. According to Lemma 7, the resultant $R(T, S)$ of $P(T - SY, Y)$ and $Q(T - SY, Y)$ with respect to Y has degree $O(d^2)$ in T and S, has bitsize in $\widetilde{O}(d(d + \tau))$, and it can be computed in $\widetilde{O}_B(d^6(d + \tau))$ bit operations. We can now apply the formulas of Proposition 23 for computing the polynomials of the RUR.

Specializing $R(T, S)$ at $S = a$ can be done by evaluating $O(d^2)$ polynomials in S, each of degree in $O(d^2)$ and bitsize in $\widetilde{O}(d^2 + d\tau)$. By Lemma 6, each of the $O(d^2)$ evaluations can be done in $\widetilde{O}_B(d^2(d^2 + d\tau + \tau_a))$ bit operations and each result has bitsize in $\widetilde{O}(d^2 + d\tau + d^2\tau_a)$. Hence, $R(T, a)$ and $f_{I,a}(T)$ have degree in $O(d^2)$, bitsize in $\widetilde{O}(d^2 + d\tau + d^2\tau_a)$, and they can be computed with $\widetilde{O}_B(d^4(d^2 + d\tau + \tau_a))$ bit operations.

The complexity of computing the numerators of $f_{I,a,1}(T)$ and $f_{I,a,Y}(T)$ is clearly dominated by the computation of $\frac{\partial R}{\partial S}(T, a)$. Indeed, computing the derivative $\frac{\partial R}{\partial S}(T, S)$ can trivially be done in $O(d^4)$ arithmetic operations of complexity $\widetilde{O}_B(d^2 + d\tau)$, that is in $\widetilde{O}_B(d^6 + d^5\tau)$. Then, as for $R(T, a)$, $\frac{\partial R}{\partial S}(T, a)$ has degree in $O(d^2)$, bitsize in $\widetilde{O}(d^2 + d\tau + d^2\tau_a)$, and it can be computed within the same complexity as the computation of $R(T, a)$.

On the other hand, since $f_{I,a}(T)$ and $f'_{I,a}(T)$ have degree in $O(d^2)$ and bitsize in $\widetilde{O}(d^2 + d\tau + d^2\tau_a)$, and $f_{I,a}(T) = \frac{R(T,a)}{L_R(a)}$, one can multiply these two polynomials by $L_R(a)$ which is of bitsize $\widetilde{O}(d^2 + d\tau + d^2\tau_a)$ and by the denominator of the rational a to the power of $d_S(R(T, S))$ which is an integer of bitsize in $O(d^2\tau_a)$, to obtain polynomials with coefficients in $\mathbb{Z}$. Hence, according to Lemma 4, the gcd of $f_{I,a}(T)$ and $f'_{I,a}(T)$ can be computed in $\widetilde{O}_B(d^4(d^2 + d\tau + d^2\tau_a))$ bit operations and it has bitsize in $\widetilde{O}(d^2 + d\tau + d^2\tau_a)$.

Now, the bit complexity of the division of the numerators by the gcd is of the order of the square of their maximum degree times their maximum bitsize (von zur Gathen and Gerhard, 2003, Theorem 9.6 and subsequent discussion), that is, the divisions (and hence the computation of $f_{I,a,1}(T)$ and $f_{I,a,Y}(T)$) can be done in $\widetilde{O}_B(d^4(d^2 + d\tau + d^2\tau_a))$ bit operations.

Finally, computing $f_{I,a,X}(T)$ can be done within the same complexity as for $f_{I,a,1}(T)$ and $f_{I,a,Y}(T)$ since it is dominated by the computation of the squarefree part of $f_{I,a}(T)$, which can be computed similarly and with the same complexity as above, by Lemma 4.

The overall complexity is thus that of computing the resultant which is in $\widetilde{O}_B(d^6(d + \tau))$ plus that of computing the above gcd and Euclidean division which is in $\widetilde{O}_B(d^4(d^2 + d\tau + d^2\tau_a))$. This gives a total of $\widetilde{O}_B(d^7 + d^6(\tau + \tau_a))$. □

4.1.1. Proof of Proposition 23

Proposition 23 expresses the polynomials $f_{I,a}$ and $f_{I,a,v}$ of a RUR in terms of specializations (by $S = a$) of the resultant $R(T, S)$ and its partial derivatives. Since the specializations are

done after considering the derivatives of R, we study the relations between these entities before specializing S by a.

For that purpose, we first introduce the following polynomials which are exactly the polynomials $f_{I,a}$ and $f_{I,a,v}$ of (6) where the parameter a is replaced by the variable S. These polynomials can be seen as the RUR polynomials of the ideal I with respect to a "generic" linear form $X+SY$.

$$\begin{aligned} f_I(T,S) &= \prod_{\sigma\in V(I)} (T - X(\sigma) - S\,Y(\sigma))^{\mu_I(\sigma)} \\ f_{I,v}(T,S) &= \sum_{\sigma\in V(I)} \mu_I(\sigma)v(\sigma) \prod_{\varsigma\in V(I),\varsigma\neq\sigma} (T - X(\varsigma) - S\,Y(\varsigma)), \quad v \in \{1, X, Y\}. \end{aligned} \tag{7}$$

These polynomials are obviously in $\mathbb{C}[T,S]$, but they are actually in $\mathbb{Q}[T,S]$ because, when S is specialized at any rational value a, the specialized polynomials are those of $RUR_{I,a}$ which are in $\mathbb{Q}[T]$ (Lemma 21).

Before proving Proposition 23, we express the derivatives of $f_I(T,S)$ in terms of $f_{I,v}(T,S)$, in Lemma 25, and show that $f_I(T,S)$ is the monic form of the resultant $R(T,S)$, seen as a polynomial in T, in Lemma 27.

Lemma 25. *Let $g_I(T,S) = \prod_{\sigma\in V(I)}(T - X(\sigma) - S\,Y(\sigma))^{\mu_I(\sigma)-1}$. We have*

$$\frac{\partial f_I}{\partial T}(T,S) = g_I(T,S)f_{I,1}(T,S), \tag{8}$$

$$\frac{\partial f_I}{\partial S}(T,S) = g_I(T,S)f_{I,Y}(T,S). \tag{9}$$

Proof. It is straightforward that the derivative of f_I with respect to T is $\sum_{\sigma\in V(I)} \mu_I(\sigma)(T - X(\sigma) - S\,Y(\sigma))^{\mu_I(\sigma)-1} \prod_{\varsigma\in V(I),\varsigma\neq\sigma}(T - X(\varsigma) - S\,Y(\sigma))^{\mu_I(\varsigma)}$, which can be rewritten as the product of $\prod_{\sigma\in V(I)}(T - X(\sigma) - S\,Y(\sigma))^{\mu_I(\sigma)-1}$ and $\sum_{\sigma\in V(I)} \mu_I(\sigma) \prod_{\varsigma\in V(I),\varsigma\neq\sigma}(T - X(\varsigma) - S\,Y(\varsigma))$ which is exactly the product of $g_I(T,S)$ and $f_{I,1}(T,S)$.

The expression of the derivative of f_I with respect to S is similar to that with respect to T except that the derivative of $T - X(\sigma) - S\,Y(\sigma)$ is now $Y(\sigma)$ instead of 1. It follows that $\frac{\partial f_I}{\partial S}$ is the product of $\prod_{\sigma\in V(I)}(T - X(\sigma) - S\,Y(\sigma))^{\mu_I(\sigma)-1}$ and $\sum_{\sigma\in V(I)} \mu_I(\sigma)Y(\sigma) \prod_{\varsigma\in V(I),\varsigma\neq\sigma}(T - X(\varsigma) - S\,Y(\varsigma))$ which is the product of $g_I(T,S)$ and $f_{I,Y}(T,S)$. □

For the proof of Lemma 27, we will need the following lemma which states that when two polynomials have no common solution at infinity in some direction, the roots of their resultant with respect to this direction are the projections of the solutions of the system with cumulated multiplicities.

Lemma 26 (Busé et al. (2005, Prop. 2 and 5))**.** *Let $P, Q \in \mathbb{F}[X,Y]$ defining a zero-dimensional ideal $I = \langle P, Q\rangle$, such that their leading terms $Lc_Y(P)$ and $Lc_Y(Q)$ do not have common roots. Then $Res_Y(P,Q) = c\prod_{\sigma\in V(I)}(X - X(\sigma))^{\mu_I(\sigma)}$ where c is nonzero in $\mathbb{F}$.*

The following lemma links the resultant of $P(T - S\,Y, Y)$ and $Q(T - S\,Y, Y)$ with respect to Y and the polynomial $f_I(T,S)$ as defined above.

Lemma 27. *$R(T,S) = L_R(S)f_I(T,S)$ and, for any $a \in \mathbb{Q}$, $L_P(a)L_Q(a) \neq 0$ implies that $L_R(a) \neq 0$.*

Proof. The proof is organized as follows. We first prove that for any rational a such that $L_P(a)L_Q(a)$ does not vanish, $R(T,a) = c(a)f_I(T,a)$ where $c(a) \in \mathbb{Q}$ is a nonzero constant depending on a. This is true for infinitely many values of a and, since $R(T,S)$ and $f_I(T,S)$ are polynomials, we can deduce that $R(T,S) = L_R(S)f_I(T,S)$. This will also imply the second statement of the lemma since, if $L_P(a)L_Q(a) \neq 0$, then $R(T,a) = c(a)f_I(T,a) = L_R(a)f_I(T,a)$ with $c(a) \neq 0$, thus $L_R(a) \neq 0$ (since $f_I(T,a)$ is monic).

If a is such that $L_P(a)L_Q(a) \neq 0$, the resultant $R(T,S)$ can be specialized at $S = a$, in the sense that $R(T,a)$ is equal to the resultant of $P(T - aY, Y)$ and $Q(T - aY, Y)$ with respect to Y (Basu et al., 2006, Proposition 4.20).

We now apply Lemma 26 to these two polynomials $P(T-aY,Y)$ and $Q(T-aY,Y)$. These two polynomials satisfy the hypotheses of this lemma: first, their leading coefficients (in Y) do not depend on T, hence they have no common root in $\mathbb{Q}[T]$; second, the polynomials $P(T - aY, Y)$ and $Q(T - aY, Y)$ are coprime because $P(X,Y)$ and $Q(X,Y)$ are coprime by assumption and the change of variables $(X,Y) \mapsto (T = X + aY, Y)$ is a $\mathbb{Q}$-automorphism of $\mathbb{Q}[X,Y]$ (and a common factor will remain a common factor after the change of variables). Hence Lemma 26 yields that $R(T,a) = c(a) \prod_{\sigma \in V(I_a)} (T - T(\sigma))^{\mu_{I_a}(\sigma)}$, where $c(a) \in \mathbb{Q}$ is a nonzero constant depending on a, and I_a is the ideal generated by $P(T - aY, Y)$ and $Q(T - aY, Y)$.

We now observe that $\prod_{\sigma \in V(I_a)} (T - T(\sigma))^{\mu_{I_a}(\sigma)}$ is equal to $f_I(T,a) = \prod_{\sigma \in V(I)} (T - X(\sigma) - aY(\sigma))^{\mu_I(\sigma)}$ since any solution (α, β) of $P(X,Y)$ is in one-to-one correspondence with the solution $(\alpha + a\beta, \beta)$ of $P(T - aY, Y)$ (and similarly for Q) and the multiplicities of the solutions also match, i.e. $\mu_I(\sigma) = \mu_{I_a}(\sigma_a)$ when σ and σ_a are in correspondence through the mapping (Fulton, 2008, §3.3 Proposition 3 and Theorem 3). Hence,

$$L_P(a)L_Q(a) \neq 0 \quad \Rightarrow \quad R(T,a) = c(a)f_I(T,a) \quad \text{with} \quad c(a) \neq 0. \tag{10}$$

Since there are finitely many values of a such that $L_P(a)L_Q(a)L_R(a) = 0$ and since $f_I(T,S)$ is monic with respect to T, (10) implies that $R(T,S)$ and $f_I(T,S)$ have the same degree in T, say D. We write these two polynomials as

$$R(T,S) = L_R(S)T^D + \sum_{i=0}^{D-1} r_i(S)T^i, \qquad f_I(T,S) = T^D + \sum_{i=0}^{D-1} f_i(S)T^i. \tag{11}$$

If a is such that $L_P(a)L_Q(a)L_R(a) \neq 0$, (10) and (11) imply that $L_R(a) = c(a)$ and $r_i(a) = L_R(a)f_i(a)$, for all i. These equalities hold for infinitely many values of a, and $r_i(S), L_R(S)$ and $f_i(S)$ are polynomials in S, thus $r_i(S) = L_R(S)f_i(S)$ and, by (11), $R(T,S) = L_R(S)f_I(T,S)$. □

We can now prove Proposition 23, which we recall, for clarity.

Proposition 23. *For any rational a such that $L_P(a)L_Q(a) \neq 0$ and such that $X + aY$ is a separating form of $I = \langle P, Q \rangle$, the RUR of $\langle P, Q \rangle$ associated to a is as follows:*

$$f_{I,a}(T) = \frac{R(T,a)}{L_R(a)} \qquad f_{I,a,1}(T) = \frac{f'_{I,a}(T)}{\gcd(f_{I,a}(T), f'_{I,a}(T))}$$

$$f_{I,a,Y}(T) = \frac{\frac{\partial R}{\partial S}(T,a) - f_{I,a}(T)\frac{\partial L_R}{\partial S}(a)}{L_R(a)\gcd(f_{I,a}(T), f'_{I,a}(T))} \qquad f_{I,a,X}(T) = T f_{I,a,1}(T) - d_T(f_{I,a})\overline{f_{I,a}(T)} - a f_{I,a,Y}(T).$$

Proof. Since we assume that a is such that $L_P(a)L_Q(a) \neq 0$, Lemma 27 immediately gives the first formula.

Equation 8 states that $f_{I,1}(T,S)g_I(T,S) = \frac{\partial f_I(T,S)}{\partial T}$, where $g_I(T,S) = \prod_{\sigma\in V(I)}(T - X(\sigma) - SY(\sigma))^{\mu_I(\sigma)-1}$. In addition, g_I being monic in T, it never identically vanishes when S is specialized, thus the preceding formula yields after specialization: $f_{I,a,1}(T) = \frac{f'_{I,a}(T)}{g_I(T,a)}$. Furthermore, $g_I(T,a) = \gcd(f_{I,a}(T), f'_{I,a}(T))$. Indeed, $f_{I,a}(T) = \prod_{\sigma\in V(I)}(T - X(\sigma) - aY(\sigma))^{\mu_I(\sigma)}$ and all values $X(\sigma) + aY(\sigma)$, for $\sigma \in V(I)$, are pairwise distinct since $X + aY$ is a separating form, thus the gcd of $f_{I,a}(T)$ and its derivative is $\prod_{\sigma\in V(I)}(T - X(\sigma) - aY(\sigma))^{\mu_I(\sigma)-1}$, that is $g_I(T,a)$. This proves the formula for $f_{I,a,1}$.

Concerning the third equation, Lemma 27 together with Equation 9 implies:

$$f_{I,Y}(T,S) = \frac{\frac{\partial f_I(T,S)}{\partial S}}{g_I(T,S)} = \frac{\frac{\partial (R(T,S)/L_R(S))}{\partial S}}{g_I(T,S)} = \frac{\frac{\partial R(T,S)}{\partial S}L_R(S) - R(T,S)\frac{\partial L_R(S)}{\partial S}}{L_R(S)^2 g_I(T,S)}$$
$$= \frac{\frac{\partial R(T,S)}{\partial S} - f_I(T,S)\frac{\partial L_R(S)}{\partial S}}{L_R(S)g_I(T,S)}.$$

As argued above, when specialized, $g_I(T,a) = \gcd(f_{I,a}(T), f'_{I,a}(T))$ and it does not identically vanish. By Lemma 27, $L_R(a)$ does not vanish either, and the formula for $f_{I,a,Y}$ follows.

It remains to compute $f_{I,a,X}$. Lemma 21 implies that, for any root γ of $f_{I,a}$: $\gamma = \frac{f_{I,a,X}}{f_{I,a,1}}(\gamma) + a\frac{f_{I,a,Y}}{f_{I,a,1}}(\gamma)$, and thus $f_{I,a,X}(\gamma) + af_{I,a,Y}(\gamma) - \gamma f_{I,a,1}(\gamma) = 0$. Replacing γ by T, we have that the polynomial $f_{I,a,X}(T) + af_{I,a,Y}(T) - Tf_{I,a,1}(T)$ vanishes at every root of $f_{I,a}$, thus the squarefree part of $f_{I,a}$ divides that polynomial. In other words, $\overline{f_{I,a,X}(T)} = Tf_{I,a,1}(T) - af_{I,a,Y}(T) \bmod \overline{f_{I,a}(T)}$. We now compute $Tf_{I,a,1}(T)$ and $af_{I,a,Y}(T)$ modulo $\overline{f_{I,a}(T)}$.

Equation (6) implies that $f_{I,a,v}(T)$ is equal to $T^{\#V(I)-1}\sum_{\sigma\in V(I)}\mu_I(\sigma)v(\sigma)$ plus some terms of lower degree in T, and that the degree of $\overline{f_{I,a}(T)}$ is $\#V(I)$ (since $X + aY$ is a separating form). First, for $v = Y$, this implies that $d_T(f_{I,a,Y}) < d_T(\overline{f_{I,a}})$, and thus that $af_{I,a,Y}(T)$ is already reduced modulo $\overline{f_{I,a}(T)}$. Second, for $v = 1$, $\sum_{\sigma\in V(I)}\mu_I(\sigma)$ is nonzero and equal to $d_T(f_{I,a})$. Thus, $Tf_{I,a,1}(T)$ and $\overline{f_{I,a}(T)}$ are both of degree $\#V(I)$, and their leading coefficients are $d_T(f_{I,a})$ and 1, respectively. Hence $Tf_{I,a,1}(T) \bmod \overline{f_{I,a}(T)} = Tf_{I,a,1}(T) - d_T(f_{I,a})\overline{f_{I,a}(T)}$. We thus obtain the last equation, that is, $f_{I,a,X}(T) = Tf_{I,a,1}(T) - d_T(f_{I,a})\overline{f_{I,a}(T)} - af_{I,a,Y}(T)$. □

4.2. RUR bitsize

We prove here, in Proposition 28, a new bound on the bitsize of the coefficients of the polynomials of a RUR. This bound is interesting in its own right and is instrumental for our analysis of the complexity of computing isolating boxes of the solutions of the input system, as well as for performing *sign_at* evaluations. We state our bound for RUR-*candidates*, that is even when the linear form $X + aY$ is not separating. We only use this result when the form is separating, for proving Theorem 22, but the general result is interesting in a probabilistic context when a RUR-candidate is computed with a random linear form. We also prove our bound, not only for the RUR-candidates of an ideal defined by *two* polynomials P and Q, but for any ideal of $\mathbb{Z}[X,Y]$ that contains P and Q (for instance the radical of $\langle P, Q\rangle$ or the ideals obtained by decomposing $\langle P, Q\rangle$ according to the multiplicity of the solutions).

Proposition 28. *Let $P, Q \in \mathbb{Z}[X,Y]$ be two coprime polynomials of total degree at most d and maximum bitsize τ, let a be a rational of bitsize τ_a, and let J be any ideal of $\mathbb{Z}[X,Y]$ containing P and Q. The polynomials of the RUR-candidate of J associated to a have degree at most d^2*

and bitsize in $\widetilde{O}(d^2\tau_a + d\tau)$. Moreover, there exists an integer of bitsize in $\widetilde{O}(d^2\tau_a + d\tau)$ such that the product of this integer with any polynomial in the RUR-candidate yields a polynomial with integer coefficients.[18]

Before proving Proposition 28, we recall Mignotte's lemma and a notion of primitive part for polynomials in $\mathbb{Q}[X, Y]$ and some of its properties.

Lemma 29 (Basu et al. (2006, Corollary 10.12))**.** *Let $P \in \mathbb{Z}[X, Y]$ be of degree at most d in each variable with coefficients bitsize at most τ. If $P = Q_1 Q_2$ with Q_1, Q_2 in $\mathbb{Z}[X, Y]$, then the bitsize of Q_i, $i = 1, 2$, is in $\widetilde{O}(d + \tau)$.*

Primitive part. Consider a polynomial P in $\mathbb{Q}[X, Y]$ of degree at most d in each variable. It can be written $P = \sum_{i,j=0}^{d} \frac{a_{ij}}{b_{ij}} X^i Y^j$ with a_{ij} and b_{ij} coprime in $\mathbb{Z}$ for all i, j. We define the *primitive part* of P, denoted $pp(P)$, as P divided by the gcd of the a_{ij} and multiplied by the least common multiple (lcm) of the b_{ij}. (Note that this definition is not entirely standard since we do not consider contents that are polynomials in X or in Y.) We also denote by τ_P the bitsize of P (that is, the maximum bitsize of all the a_{ij} and b_{ij}). We prove three properties of the primitive part which will be useful in the proof of Proposition 28.

Lemma 30. *For any two polynomials P and Q in $\mathbb{Q}[X, Y]$, we have the following properties: (i) $pp(PQ) = pp(P)\, pp(Q)$. (ii) If P is monic then $\tau_P \leqslant \tau_{pp(P)}$ and, more generally, if P has one coefficient, ξ, of bitsize τ_ξ, then $\tau_P \leqslant \tau_\xi + \tau_{pp(P)}$. (iii) If P has coefficients in $\mathbb{Z}$, then $\tau_{pp(P)} \leqslant \tau_P$.*

Proof. Gauss Lemma states that if two univariate polynomials with integer coefficients are primitive, so is their product. This lemma can straightforwardly be extended to be used in our context by applying the mapping $X^i Y^j \to Z^{ik+j}$ with $k > 2\max(d_Y(P), d_Y(Q))$. Thus, if P and Q in $\mathbb{Q}[X, Y]$ are primitive (i.e., each of them has integer coefficients whose common gcd is 1), their product is primitive. It follows that $pp(PQ) = pp(P)\, pp(Q)$ because, writing $P = \alpha\, pp(P)$ and $Q = \beta\, pp(Q)$, we have $pp(PQ) = pp(\alpha\, pp(P) \beta\, pp(Q)) = pp(pp(P)\, pp(Q))$ which is equal to $pp(P)\, pp(Q)$ since the product of two primitive polynomials is primitive.

Second, if $P \in \mathbb{Q}[X, Y]$ has one coefficient, ξ, of bitsize τ_ξ, then $\tau_P \leqslant \tau_\xi + \tau_{pp(P)}$. Indeed, We have $P = \xi \frac{P}{\xi}$ thus $\tau_P \leqslant \tau_\xi + \tau_{\frac{P}{\xi}}$. Since $\frac{P}{\xi}$ has one of its coefficients equal to 1, its primitive part is $\frac{P}{\xi}$ multiplied by an integer (the lcm of the denominators), thus $\tau_{\frac{P}{\xi}} \leqslant \tau_{pp(\frac{P}{\xi})}$ and $pp(\frac{P}{\xi}) = pp(P)$ by definition, which implies the claim.

Third, if P has coefficients in $\mathbb{Z}$, then $\tau_{pp(P)} \leqslant \tau_P$ since $pp(P)$ is equal to P divided by an integer (the gcd of the integer coefficients). □

The idea of the proof of Proposition 28 is, for $J \supseteq I = \langle P, Q \rangle$, first argue that the polynomial f_J, that is the first polynomial of the RUR-candidate before specialization at $S = a$, is a factor of f_I which is a factor of the resultant $R(T, S)$ by Lemma 27. We then derive a bound of $\widetilde{O}(d^2 + d\tau)$ on the bitsize of f_J from the bitsize of this resultant using Lemma 29. The bound on the bitsize of the other polynomials of the non-specialized RUR-candidate of J follows from the bound on f_J and we finally specialize all these polynomials at $S = a$ which yields the result. We decompose this proof in two lemmas to emphasize that, although the bound on the bitsize of f_J uses the fact that J contains the polynomials P and Q, the second part of the proof only uses the bound on f_J.

[18] In other words, the mapping $\gamma \mapsto \left(\frac{f_{J,a,X}}{f_{J,a,1}}(\gamma), \frac{f_{J,a,Y}}{f_{J,a,1}}(\gamma)\right)$ sending the solutions of $f_{J,a}(T)$ to those of J (see Lemma 21) can be defined with polynomials with integer coefficients of bitsize $\widetilde{O}(d^2\tau_a + d\tau)$. This will be needed in the proof of Lemma 37.

Lemma 31. *Let $P, Q \in \mathbb{Z}[X, Y]$ be two coprime polynomials of total degree at most d and maximum bitsize τ, and J be any ideal of $\mathbb{Z}[X, Y]$ containing P and Q. The polynomial $f_J(T, S)$ (see (7)) and its primitive part have bitsize in $\widetilde{O}(d^2 + d\tau)$ and degree at most d^2 in each variable.*

Proof. Consider an ideal J containing $I = \langle P, Q\rangle$. Counted with multiplicity, the set of solutions of J is a subset of those of I thus, by Equation (7), polynomial $f_J(T, S)$ is monic in T and $f_J(T, S)$ divides $f_I(T, S)$. Furthermore, $f_I(T, S)$ divides $R(T, S)$ by Lemma 27. Thus $f_J(T, S)$ divides $R(T, S)$ and we consider $h \in \mathbb{Q}[T, S]$ such that $f_J\, h = R$. Taking the primitive part, we have $pp(f_J)\, pp(h) = pp(R)$ by Lemma 30. The bitsize of $pp(R)$ is in $\widetilde{O}(d^2 + d\tau)$ because R is of bitsize $\widetilde{O}(d^2 + d\tau)$ (Lemma 7) and, since R has integer coefficients, $\tau_{pp(R)} \leqslant \tau_R$ (Lemma 30). This implies that $pp(f_J)$ also has bitsize in $\widetilde{O}(d^2 + d\tau)$ by Lemma 29 because the degree of $pp(R)$ is in $O(d^2)$ (Lemma 7). Furthermore, since $f_J(T, S)$ is monic in T, $\tau_{f_J} \leqslant \tau_{pp(f_J)}$ (Lemma 30) which implies that both f_J and its primitive part have bitsize in $\widetilde{O}(d^2 + d\tau)$. Finally, the number of solutions (counted with multiplicity) of $\langle P, Q\rangle$ is at most d^2 by the Bézout bound, and this bound also holds for $J \supseteq \langle P, Q\rangle$. It then follows from Equation (7) that f_J has degree at most d^2 in each variable. □

Lemma 32. *Let J be any ideal such that polynomials $f_J(T, S)$ (see (7)) and its primitive part have degree $O(d^2)$ and bitsize in $\widetilde{O}(d^2 + d\tau)$ and a is a rational of bitsize τ_a. Then all the polynomials of the RUR-candidate $RUR_{J,a}$ have bitsize in $\widetilde{O}(d^2\tau_a + d\tau)$. Moreover, there exists an integer of bitsize in $\widetilde{O}(d^2\tau_a + d\tau)$ such that its product with any polynomial in the RUR-candidate yields a polynomial with integer coefficients.*

Proof. Bitsize of $f_{J,v}$, $v \in \{1, Y\}$. We consider the equations of Lemma 25 which can be written as $\frac{\partial f_J}{\partial u}(T, S) = g_J(T, S) f_{J,v}(T, S)$ where u is T or S, and v is 1 or Y, respectively. We first bound the bitsize of one coefficient, ξ, of $f_{J,v}$ so that we can apply Lemma 30 which states that $\tau_{f_{J,v}} \leqslant \tau_\xi + \tau_{pp(f_{J,v})}$. We consider the leading coefficient ξ of $f_{J,v}$ with respect to the lexicographic order (T, S). Since g_J is monic in T (see Lemma 25), the leading coefficient (with respect to the same ordering) of the product $g_J f_{J,v} = \frac{\partial f_J}{\partial u}$ is ξ which thus has bitsize in $\widetilde{O}(\tau_{f_J})$ (since it is bounded by τ_{f_J} plus the log of the degree of f_J). It thus follows from the hypothesis on τ_{f_J} that $\tau_{f_{J,v}}$ is in $\widetilde{O}(d^2 + d\tau + \tau_{pp(f_{J,v})})$.

We now take the primitive part of the above equation (of Lemma 25), which gives $pp(\frac{\partial f_J}{\partial u}(T, S))$ $= pp(g_J(T, S))\, pp(f_{J,v}(T, S))$. By Lemma 29, $\tau_{pp(f_{J,v})}$ is in $\widetilde{O}(d^2 + \tau_{pp(\frac{\partial f_J}{\partial u})})$. In order to bound the bitsize of $pp(\frac{\partial f_J}{\partial u})$, we multiply $\frac{\partial f_J}{\partial u}$ by the lcm of the denominators of the coefficients of f_J, which we denote by lcm_{f_J}. Multiplying by a constant does not change the primitive part and $\mathrm{lcm}_{f_J} \frac{\partial f_J}{\partial u}$ has integer coefficients, so the bitsize of $pp(\frac{\partial f_J}{\partial u}) = pp(\mathrm{lcm}_{f_J} \frac{\partial f_J}{\partial u})$ is thus at most that of $\mathrm{lcm}_{f_J} \frac{\partial f_J}{\partial u}$ which is bounded by the sum of the bitsizes of lcm_{f_J} and $\frac{\partial f_J}{\partial u}$. By hypothesis, the bitsize of f_J is in $\widetilde{O}(d^2 + d\tau)$ so the bitsize of $\frac{\partial f_J}{\partial u}$ is also in $\widetilde{O}(d^2 + d\tau)$. On the other hand, since f_J is monic (in T), $f_J\, \mathrm{lcm}_{f_J} = pp(f_J)$ and $\tau_{\mathrm{lcm}_{f_J}} \leqslant \tau_{pp(f_J)}$ which is in $\widetilde{O}(d^2 + d\tau)$ by hypothesis. It follows that $\tau_{pp(f_{J,v})}$ and $\tau_{f_{J,v}}$ are also in $\widetilde{O}(d^2 + d\tau)$ for $v \in \{1, Y\}$.

Bitsize of $f_{J,X}$. We obtain the bound for $f_{J,X}$ by symmetry. Similarly as we proved that $f_{J,Y}$ has bitsize in $\widetilde{O}(d^2 + d\tau)$, we get, by exchanging the role of X and Y in Equation (7) and Lemma 25, that $\sum_{\sigma\in V(J)} \mu_J(\sigma) X(\sigma) \prod_{\varsigma\in V(J), \varsigma\neq\sigma} (T - Y(\varsigma) - S\, X(\varsigma))$ has bitsize in $\widetilde{O}(d^2 + d\tau)$. This polynomial has degree $O(d^2)$ in T and S, by hypothesis, thus after replacing S by $\frac{1}{S}$ and then T by $\frac{T}{S}$, the

polynomial is of degree $O(d^2)$ in T and $\frac{1}{S}$. We multiply it by S to the power of $\frac{1}{S}$ and obtain $f_{J,X}$ which is thus of bitsize $\widetilde{O}(d^2 + d\tau)$.

Specialization at $S = a$. To bound the bitsize of the polynomials of $RUR_{J,a}$ (Definition 20), it remains to evaluate the polynomials f_J and $f_{J,v}$, $v \in \{1, X, Y\}$, at the rational value $S = a$ of bitsize τ_a. Since these polynomials have degree in S in $O(d^2)$ and bitsize in $\widetilde{O}(d^2 + d\tau)$, it follows from Lemma 6 that their specializations at $S = a$ have bitsize in $\widetilde{O}(d^2 + d\tau + d^2\tau_a) = \widetilde{O}(d^2\tau_a + d\tau)$.

The lcm of the denominators of all the coefficients in the polynomials of $RUR_{J,a}$ has bitsize $\widetilde{O}(d^2\tau_a + d\tau)$. We have already argued that lcm_{f_J}, the lcm of the denominators of the coefficients of f_J, is in $\widetilde{O}(d^2 + d\tau)$. For each of the other polynomials $f_{J,v}$, $v \in \{1, X, Y\}$, denote by $\mathrm{lcm}_{f_{J,v}}$ and $\gcd_{f_{J,v}}$ the lcm of the denominators of its coefficients and the gcd of its numerators. By definition, $pp(f_{J,v}) = \frac{\mathrm{lcm}_{f_J}}{\gcd_{f_{J,v}}} f_{J,v}$. Let c be any coefficient of $pp(f_{J,v}) \in \mathbb{Z}[S, T]$ and $\frac{a}{b}$ be the corresponding coefficient of $f_{J,v} \in \mathbb{Q}[S, T]$ (with a and b coprime integers); we have $\mathrm{lcm}_{f_J} = c\,\frac{b}{a}\,\gcd_{f_{J,v}} \leqslant c\,b$ since $\gcd_{f_{J,v}}$ divides a. It follows that $\tau_{\mathrm{lcm}_{f_J}} \leqslant \tau_{pp(f_{J,v})} + \tau_{f_{J,v}}$ which are both in $\widetilde{O}(d^2 + d\tau)$, as proved above. Hence the lcm of the denominators of all the coefficients in $RUR_{J,a}$ has bitsize $\widetilde{O}(d^2 + d\tau)$. Finally, since all these polynomials have degree $O(d^2)$, when specializing by $S = a$, the bitsize of the denominators of the coefficients of the polynomials increase by at most $O(d^2\tau_a)$ and thus the bitsize of their lcm also increases by at most $O(d^2\tau_a)$, which concludes the proof. □

Proof of Proposition 28. By Lemma 31 and Equation (7), f_J and $f_{J,v}$, $v \in \{1, X, Y\}$ have degree at most d^2 with respect to each variable. It follows from Equation (6) that all the polynomials of any RUR-candidate of J have degree at most d^2. The rest of the proposition is a corollary of Lemmas 31 and 32. □

5. Applications

We present three applications enlightening the advantages of computing a RUR of a system. The first one is the isolation of the solutions, that is computing boxes with rational coordinates that isolate the solutions. The second one is the evaluation of the sign of a bivariate polynomial at a real solution of the system. Finally, we address the problem of computing a rational parameterization of a system defined by several equality and inequality constraints. In all these applications, we take advantage of the RUR to transform bivariate operations on the system into univariate operations. We assume that the polynomials of the RURs satisfy the bitsize bound of Theorem 22.

We start by recalling the complexity of isolating the real roots of a univariate polynomial. Here, f denotes a univariate polynomial of degree d with integer coefficients of bitsize at most τ.

Lemma 33 (Mehlhorn et al. (2013, Theorem 5)[19])**.** *Isolating intervals of all the real roots of f can be computed and refined up to a width less than 2^{-L} with $\widetilde{O}_B(d^3 + d^2\tau + dL)$ bit operations.*

Let the minimum root separation bound of f (or simply the separation bound of f) be the minimum distance between two different complex roots of f: $\mathrm{sep}(f) = \min_{\{\gamma, \delta \text{ roots of } f,\ \gamma \neq \delta\}} |\gamma - \delta|$.

[19] Theorem 5 of Mehlhorn et al. (2013) is stated for complex roots, however it is straightforward to identify the boxes containing the real roots within the same complexity. Indeed, by considering L in $\widetilde{O}(d\tau)$ with 2^{-L} smaller than twice the root separation bound of f (which is possible by Lemma 34), the isolating boxes of the complex roots do not intersect the real axis.

Lemma 34 (Rump (1979, Theorem 4)). *One has* $sep(f) > 1/(2d^{d/2+2}(d2^\tau+1)^d)$*, which yields* $sep(f) > 2^{-\widetilde{O}(d\tau)}$.

5.1. Computation of isolating boxes

By Lemma 21, the RUR of an ideal I defines a mapping between the roots of a univariate polynomial and the solutions of I, which yields an algorithm to compute isolating boxes. Given a RUR $\{f_{I,a}, f_{I,a,1}, f_{I,a,X}, f_{I,a,Y}\}$ of the ideal I, isolating boxes for the real solutions can be computed by first computing isolating intervals for the real roots of the univariate polynomial $f_{I,a}$ and then, evaluating the rational fractions $\frac{f_{I,a,X}}{f_{I,a,1}}$ and $\frac{f_{I,a,Y}}{f_{I,a,1}}$ by interval arithmetic. However, for the simplicity of the proof, instead of evaluating by interval arithmetic each of these fractions of polynomials, we instead compute the product of its numerator with the inverted denominator modulo $f_{I,a}$, and then evaluate this resulting polynomial on the isolating intervals of the real roots of $f_{I,a}$ (note that we obtain the same complexity bound if we directly evaluate the fractions, but the proof is more technical, although not difficult, and we omit it here). When these isolating intervals are sufficiently refined, the computed boxes are necessarily disjoint and thus isolating. The following proposition analyzes the bit complexity of this algorithm.

Proposition 35. *Given a RUR of* $\langle P, Q\rangle$*, isolating boxes for the solutions of* $\langle P, Q\rangle$ *can be computed in* $\widetilde{O}_B(d^8 + d^7\tau)$ *bit operations, where* d *bounds the total degree of* P *and* Q*, and* τ *bounds the bitsize of their coefficients. The vertices of these boxes have bitsize in* $\widetilde{O}(d^3\tau)$.

Proof. For every real solution α of $I = \langle P, Q\rangle$, let $J_{X,\alpha} \times J_{Y,\alpha}$ be a box containing it. A sufficient condition for these boxes to be isolating is that the width of every interval $J_{X,\alpha}$ and $J_{Y,\alpha}$ is less than half the separation bound of the resultant of P and Q with respect to X and Y, respectively. Such a resultant has degree at most $2d^2$ and bitsize in $\widetilde{O}(d\tau)$ by Basu et al. (2006, Proposition 8.46). Lemma 34 thus yields a lower bound of $2^{-\varepsilon}$ with ε in $\widetilde{O}(d^3\tau)$ on the separating bound of such a resultant. It is thus sufficient to compute, for every α, a box $J_{X,\alpha} \times J_{Y,\alpha}$ that contains α and such that the widths of these intervals are smaller than half of $2^{-\varepsilon}$. For clarity and technical reasons, we define $\varepsilon' = \varepsilon + 2$. In fact, an explicit value of ε is not needed to compute isolating boxes since the algorithm uses adaptive refinements of the boxes and a test of box disjointness. On the other hand, an explicit value of ε will be used to reduce the bitsize of the box endpoints and an asymptotic estimate will be used for the complexity analysis. More precisely, the algorithm proceeds as follows. First, the real roots of $f_{I,a}$ are isolated. Then, we refine these intervals and, during the refinement, we routinely evaluate the polynomials of the mapping at these intervals, and we stop when all the resulting boxes are pairwise disjoint. It is of course critical not to evaluate the polynomials of the mapping too often; for every real root of $f_{I,a}$, we perform these evaluations every time the number of identical consecutive first bits of the two interval endpoints doubles or, in other words, every time the width of the interval becomes smaller than 2^{-2^k} for some positive integer k.

According to Lemma 21, given a RUR $\{f_{I,a}, f_{I,a,1}, f_{I,a,X}, f_{I,a,Y}\}$ of I, the mapping $\gamma \mapsto \left(\frac{f_{I,a,X}}{f_{I,a,1}}(\gamma), \frac{f_{I,a,Y}}{f_{I,a,1}}(\gamma)\right)$ defines a one-to-one correspondence between the real roots of $f_{I,a}$ and those of I. Thus every isolating interval J_γ of the real roots of $f_{I,a}$ is mapped through this mapping to a pair of intervals defining a box that contains the corresponding solution of I. We first show how to modify this rational mapping into a polynomial one. Second, we bound, in terms of the width of J_γ, the side length of the box obtained by interval arithmetic as the image of J_γ through the mapping. We will then deduce an upper bound on the width of J_γ that ensures that the side length of its box image is less than $2^{-\varepsilon'}$. This thus gives a worst-case refinement precision on the isolating

intervals of $f_{I,a}$ for the boxes to be disjoint. We then analyze the complexity of the proposed algorithm.

Polynomial mapping. By Proposition 23, the polynomials $f_{I,a}$ and $f_{I,a,1}$ are coprime and thus $f_{I,a,1}$ is invertible modulo $f_{I,a}$. The rational mapping can thus be transformed into a polynomial one by replacing $\frac{1}{f_{I,a,1}}$ by the inverse of $f_{I,a,1}$ modulo $f_{I,a}$. Since $\frac{1}{f_{I,a,1}}$ and the inverse of $f_{I,a,1}$ modulo $f_{I,a}$ coincide when $f_{I,a}$ vanishes (by Bézout's identity), this polynomial mapping still maps the real roots of $f_{I,a}$ to those of I.

This polynomial mapping can be computed in $\widetilde{O}_B(d^6 + d^5\tau)$ bit operations and these polynomials have degree less than $4d^2$ and bitsize in $\widetilde{O}(d^4 + d^3\tau)$. Indeed, the bit complexity of computing the inverse $\frac{1}{f_{I,a,1}}$ modulo $f_{I,a}$ is softly linear in the square of their maximum degree times their maximum bitsize (von zur Gathen and Gerhard, 2003, Corollary 11.11(ii)),[20] which yields a complexity of $\widetilde{O}_B((d^2)^2(d^2 + d\tau))$ by Theorem 22. The bitsize of this inverse is softly linear in the product of their maximum degree and maximum bitsize (von zur Gathen and Gerhard, 2003, Corollary 6.52), that is $\widetilde{O}(d^2(d^2 + d\tau))$. Furthermore, the product of this inverse and of $f_{I,a,X}$ or $f_{I,a,Y}$ can also be done with a bit complexity that is softly linear in the product of their maximum degree and maximum bitsize (von zur Gathen and Gerhard, 2003, Corollary 8.27), that is in $\widetilde{O}_B(d^2(d^4 + d^3\tau))$. This concludes the proof of the claim since the degree of the inverse modulo $f_{I,a}$ is less than that of $f_{I,a}$ and all the polynomials of the RUR have degrees at most d^2 by Theorem 22.

Width expansion through interval arithmetic evaluation. We recall a standard straightforward property of interval arithmetic for polynomial evaluation. We consider here exact interval arithmetic, that is, the arithmetic operations on the interval endpoints are considered exact. Let $J = [a, b]$ be an interval with rational endpoints such that $\max(|a|, |b|) \leqslant 2^\sigma$ and let $f \in \mathbb{Z}[T]$ be a polynomial of degree d_f with coefficients of bitsize τ_f. Denoting the width of J by $w(J) = |b - a|$, $f(J)$ can be evaluated by interval arithmetic into an interval $f_\square(J)$ whose width is at most $2^{\tau_f + d_f\sigma} d_f^2 w(J)$; see e.g. Cheng et al. (2010, Lemma 8). In other words, if $w(J) \leqslant 2^{-\varepsilon' - \tau_f - d_f\sigma - 2\log d_f}$, then $w(f_\square(J)) \leqslant 2^{-\varepsilon'}$.

We now apply this property to the polynomials of the mapping evaluated on isolating intervals of $f_{I,a}$. We denote by d_f and τ_f the maximum degree and bitsize of the polynomials of the mapping; as shown above $d_f < 4d^2$ and $\tau_f \in \widetilde{O}(d^4 + d^3\tau)$. The polynomial $f_{I,a}$ has bitsize $\tau_{f_{I,a}}$ in $\widetilde{O}(d^2 + d\tau)$ (Theorem 22), thus, by Cauchy's bound (see e.g. Yap (2000, §6.2)), the maximum absolute value of its roots is smaller than $1 + 2^{2\tau_{f_{I,a}}}$. Considering intervals of isolation for $f_{I,a}$ whose widths are bounded by a constant, we thus have that the maximum absolute value of the endpoints of the isolating intervals are smaller than 2^σ with $\sigma = \widetilde{O}(d^2 + d\tau)$. Now, consider any isolating interval of $f_{I,a}$ of width less than $2^{-\varepsilon' - \tau_f - d_f\sigma - 2\log d_f}$. The above property implies that we can evaluate by interval arithmetic the polynomials of the mapping on any such intervals and obtain an interval of width less than $2^{-\varepsilon'}$. In other words, the worst-case refinement precision of the isolating intervals of $f_{I,a}$ for the boxes to be disjoint is $L = \varepsilon' + \tau_f + d_f\sigma + 2\log d_f$. In addition, since ε' is in $\widetilde{O}(d^3\tau)$, L is in $\widetilde{O}(d^4 + d^3\tau)$.

[20] von zur Gathen and Gerhard (2003, Corollary 11.11(ii)) applies because this inverse is the cofactor of $f_{I,a,1}$ in the last line of the extended Euclidean algorithm corresponding to the resultant of $f_{I,a,1}$ and $f_{I,a}$. Note that this assumes that $f_{I,a,1}$ and $f_{I,a}$ have integer coefficients but this is not an issue because, by Proposition 28, all polynomials of the RUR can be transformed into integer polynomials with the same asymptotic bitsize by multiplying them by one and the same integer.

Analysis of the algorithm. For isolation and refinement, we consider the polynomial $\overline{pp(f_{I,a})}$, instead of $f_{I,a}$, which is also of degree bounded by d^2 and bitsize in $\widetilde{O}(d^2 + d\tau)$. Indeed, Proposition 28 implies that the integer polynomial $pp(f_{I,a})$ has bitsize in $\widetilde{O}(d^2 + d\tau)$ and Lemma 4 yields that its squarefree part (which the gcd-free part of itself and its derivative) is of the same bitsize and can be computed in $\widetilde{O}(d^6 + d^5\tau)$. According to Lemma 33, isolating intervals of the real roots of $\overline{pp(f_{I,a})}$ can be computed and refined up to a width less than 2^{-L} with $\widetilde{O}_B((d^2)^3 + (d^2)^2(d^2 + d\tau) + d^2L)$ bit operations which is in $\widetilde{O}_B(d^6 + d^5\tau)$ since $L = \widetilde{O}(d^4 + d^3\tau)$.

It remains to analyze the cost of the evaluations of the mapping and the cost of the box-disjointness tests. For a given root, an evaluation of the polynomials of the mapping is performed each time its isolating interval precision is doubled, the number of evaluations is thus logarithmic in the maximum precision reached, that is L. One evaluation by interval arithmetic of the polynomials of the mapping, which have degree $O(d^2)$ and bitsize $\widetilde{O}(d^4 + d^3\tau)$, on one isolating intervals whose endpoints have bitsize at most $L \in \widetilde{O}(d^4+d^3\tau)$ can be done in $\widetilde{O}_B(d^2(d^4+d^3\tau))$ bit operations by Lemma 6 and the resulting intervals have endpoints of bitsize in $\widetilde{O}(d^2(d^4 + d^3\tau))$. The cost of the $O(\log L)$ evaluations for the $O(d^2)$ roots is then in $\widetilde{O}_B(d^8 + d^7\tau)$. Moreover, the algorithm requires testing $O(\log L)$ times whether some of the $O(d^2)$ boxes intersect, which can be done, in total, with $O(\log L)$ times $\widetilde{O}(d^2)$ arithmetic operations (see e.g. Zomorodian and Edelsbrunner (2002, §3)) and thus with $\widetilde{O}_B(d^8 + d^7\tau)$ bit operations since the vertices of the box vertices have bitsize in $\widetilde{O}(d^6 + d^5\tau)$.

Therefore, we can compute isolating boxes for the solutions of $\langle P, Q\rangle$ in $\widetilde{O}_B(d^8 + d^7\tau)$ bit operations, and the box vertices have bitsize in $\widetilde{O}_B(d^6 + d^5\tau)$.

Bitsize of the box vertices. We finally show how to compute, from the isolated boxes with vertices of bitsize in $\widetilde{O}(d^6 + d^5\tau)$, some larger isolating boxes whose vertices have bitsize in $\widetilde{O}(d^3\tau)$. The method is identical for the X or the Y-coordinates of the boxes, thus we only consider the x-coordinates. We iteratively refine the boxes as describe above except that, once none of the boxes intersect, we carry on with the iterative refinement of the boxes until the distance in X between any two boxes that do not overlap in X is larger than $\frac{1}{2}2^{-\varepsilon}$ where ε, as defined at the beginning of the proof, is such that the distance between any two roots of the resultant of P and Q with respect to X is at least $2^{-\varepsilon}$; we use here an explicit value for ε which is given by Lemma 34. On the other hand, if we were to refine all the boxes until their widths are less than $2^{-\varepsilon'} = \frac{1}{4}2^{-\varepsilon}$, the distance between any two boxes that do not overlap in X would be ensured to be larger than $\frac{1}{2}2^{-\varepsilon}$. Hence the above analysis of the algorithm still applies since we considered that all boxes could be refined until their width (and height) do not exceed $2^{-\varepsilon'}$.

Now, for every box, all the other boxes that do not overlap in X are at distance more than $\frac{1}{2}2^{-\varepsilon}$ in X (before enlargement), so the considered box can be enlarged in X using coordinates in intervals of length at least $\frac{1}{4}2^{-\varepsilon}$ on the left and on the right sides of the box. We conclude the argument by noting that, given any such interval $[a, b]$ of width at least $2^{-\varepsilon'}$ with $\varepsilon' = \varepsilon + 2 \in \widetilde{O}(d^3\tau)$ and such that $|a|$ and $|b|$ are smaller than 2^σ with $\sigma = \widetilde{O}(d^2 + d\tau)$ (by Cauchy bound, as noted above), we can easily compute in that interval a rational of bitsize at most $\varepsilon' + \sigma \in \widetilde{O}(d^3\tau)$.[21] □

[21] A rational of bitsize at most $\varepsilon' + \sigma$ can be constructed as follows. We can assume without loss of generality that a and b are both positive since the case where they are both negative is symmetric and, otherwise, the problem is trivial. Let q_k be the truncation of b after the k-th digits of the mantissa, i.e. $q_k = \lfloor b2^k \rfloor 2^{-k}$, and let k_1 be the smallest nonnegative integer such that $q_{k_1} \geqslant a$. By construction $q_{k_1} \in [a, b]$ and we prove that its bitsize is at most $\varepsilon' + \sigma$. If $k_1 = 0$, $q_{k_1} = \lfloor b \rfloor \leqslant 2^\sigma$ thus q_{k_1} has bitsize at most σ. Otherwise, with $k_0 = k_1 - 1$, we have $q_{k_0} < a$ which implies that

Remark 36. *It is straightforward that the above proof and proposition also hold if a parameterization of González-Vega and El Kahoui (1996) is given instead of a RUR.*

5.2. Sign of a polynomial at the solutions of a system

This section addresses the problem of computing the sign (+, − or 0) of a given polynomial F at the solutions of a bivariate system defined by two polynomials P and Q. We consider in the following that all input polynomials, P, Q and F are in $\mathbb{Z}[X,Y]$, have degree at most d and coefficients of bitsize at most τ. We assume without loss of generality that the bound d is *even*. Recall that, as mentioned in the introduction, the best known complexity for this problem is to our knowledge $\widetilde{O}_B(d^{10}+d^9\tau)$ for the sign at one real solution and $\widetilde{O}_B(d^{12}+d^{11}\tau)$ for the sign at all the solutions; see Diochnos et al. (2009, Th. 14 & Cor. 24) with the improvement of Sagraloff (2012) for the root isolation. We first describe a naive RUR-based *sign_at* algorithm for computing the sign at one real solution of the system, which runs in $\widetilde{O}_B(d^9+d^8\tau)$ time. Then, using properties of generalized Sturm sequences, we analyze a more efficient algorithm that runs in $\widetilde{O}_B(d^8+d^7\tau)$ time. We also show that the sign of F at the $O(d^2)$ solutions of the system can be computed in only $O(d)$ times that for one real solution.

Once the RUR $\{f_{I,a}, f_{I,a,1}, f_{I,a,X}, f_{I,a,Y}\}$ of $I=\langle P,Q\rangle$ is computed, we can use it to translate a bivariate sign computation into a univariate sign computation. Indeed, let $F(X,Y)$ be the polynomial to be evaluated at the solution (α,β) of I that is the image of the root γ of $f_{I,a}$ by the RUR mapping. We first define the polynomial $f_F(T)$ roughly as the numerator of the rational fraction obtained by substituting $X=\frac{f_{I,a,X}(T)}{f_{I,a,1}(T)}$ and $Y=\frac{f_{I,a,Y}(T)}{f_{I,a,1}(T)}$ in the polynomial $F(X,Y)$, so that the sign of $F(\alpha,\beta)$ is the same as that of $f_F(\gamma)$.

Lemma 37. *The primitive part*[22] *of $f_F(T)=f_{I,a,1}^d(T)F(T-aY,Y)$, with $Y=\frac{f_{I,a,Y}(T)}{f_{I,a,1}(T)}$, has degree $O(d^3)$, bitsize in $\widetilde{O}(d^3+d^2\tau)$, and it can be computed with $\widetilde{O}_B(d^7+d^6\tau)$ bit operations. The sign of F at a real solution of $I=\langle P,Q\rangle$ is equal to the sign of $pp(f_F)$ at the corresponding root of $f_{I,a}$ via the mapping of the RUR.*

Proof. We first compute the polynomial $F(T-aY,Y)$ in the form $\sum_{i=0}^d a_i(T)Y^i$. Then, $f_F(T)$ is equal to $\sum_{i=0}^d a_i(T)f_{I,a,Y}(T)^i f_{I,a,1}(T)^{d-i}$. Consequently, computing an expanded form of $f_F(T)$ can be done by computing the $a_i(T)$, the powers $f_{I,a,Y}(T)^i$ and $f_{I,a,1}(T)^i$, and their appropriate products and sum.

Computing $a_i(T)$. According to Lemma 7, $P(T-SY,Y)$ can be expanded with $\widetilde{O}_B(d^4+d^3\tau)$ bit operations and its bitsize is in $\widetilde{O}(d+\tau)$. These bounds also apply to $F(T-SY,Y)$ and we deduce $F(T-aY,Y)$ by substituting S by a. Writing $F(T-SY,Y)=\sum_{i=0}^d f_i(T,Y)S^i$, the computation of $F(T-aY,Y)$ can be done by computing and summing the $f_i(T,Y)a^i$. Since a has bitsize in $O(\log d)$ by hypothesis, a^i has bitsize in $O(d\log d)\subseteq\widetilde{O}(d)$, and computing all the a^i can be done with $\widetilde{O}_B(d^2)$ bit operations. For each a^i, computing $f_i(T,Y)a^i$ can be done with $O(d^2)$ multiplications between integers of bitsize in $\widetilde{O}(d+\tau)$, and thus with $\widetilde{O}_B(d^2(d+\tau))$ bit operations. Thus, computing all the $f_i(T,Y)a^i$ can be done with $\widetilde{O}_B(d^3(d+\tau))$ bit operations, and summing, for every one of the $O(d^2)$ monomials in (T,Y), d coefficients (corresponding to every i) of bitsize in $\widetilde{O}(d+\tau)$ can also be done with $\widetilde{O}_B(d^3(d+\tau))$ bit operations, in total. It follows that, $F(T-aY,Y)$ and thus all the $a_i(T)$ can be computed with $\widetilde{O}_B(d^4+d^3\tau)$ bit operations.

$b-q_{k_0}>b-a\geqslant 2^{-\varepsilon'}$. On the other hand, $b-q_{k_0}=2^{-k_0}(b2^{k_0}-\lfloor b2^{k_0}\rfloor)<2^{-k_0}$, thus $2^{-\varepsilon'}<2^{-k_0}$ and $\varepsilon'>k_0$. It follows that the bitsize of q_{k_1}, which is k_1 plus the bitsize of $\lfloor b\rfloor$, is less than $\varepsilon'+1$ plus σ.

[22] See definition in Section 4.2.

Computing $f_{I,a,Y}(T)^i$ and $f_{I,a,1}(T)^i$. By Theorem 22, $f_{I,a,Y}(T)$ has degree $O(d^2)$ and bitsize $\widetilde{O}(d^2 + d\tau)$, thus $f_{I,a,Y}(T)^i$ has degree in $O(d^3)$ and bitsize in $\widetilde{O}(d^3 + d^2\tau)$. Computing all the $f_{I,a,Y}(T)^i$ can be done with $O(d)$ multiplications between these polynomials. Every multiplication can be done with a bit complexity that is softly linear in the product of the maximum degrees and maximum bitsizes (von zur Gathen and Gerhard, 2003, Corollary 8.27), thus all the multiplications can be done with $\widetilde{O}_B(d^4(d^3 + d^2\tau))$ bit operations in total. It follows that all the $f_{I,a,Y}(T)^i$, and similarly all the $f_{I,a,1}(T)^i$, can be computed using $\widetilde{O}_B(d^7 + d^6\tau)$ bit operations and their bitsize is in $\widetilde{O}(d^3 + d^2\tau)$.

Computing $f_F(T)$. Computing $a_i(T)f_{I,a,Y}(T)^i f_{I,a,1}(T)^{d-i}$, for $i = 0, \ldots, d$, amounts to multiplying $O(d)$ times, univariate polynomials of degree $O(d^3)$ and bitsize $\widetilde{O}(d^3 + d^2\tau)$, which can be done, similarly as above, with $\widetilde{O}(d^7 + d^6\tau)$ bit operations. Finally, their sum is the sum of d univariate polynomials of degree $O(d^3)$ and bitsize $\widetilde{O}(d^3 + d^2\tau)$, which can also be computed within the same bit complexity. Hence, $f_F(T)$ can be computed with $\widetilde{O}_B(d^7 + d^6\tau)$ bit operations and its coefficients have bitsize in $\widetilde{O}(d^3 + d^2\tau)$.

Primitive part of $f_F(T)$. According to Proposition 28, there exists an integer r of bitsize in $\widetilde{O}(d^2 + d\tau)$ such that its product with the RUR polynomials gives polynomials in $\mathbb{Z}[T]$ of bitsize in $\widetilde{O}(d^2 + d\tau)$. Consider the polynomial $r^d f_F(T) = (rf_{I,a,1}(T))^d F(T - aY, Y)$ with $Y = \frac{rf_{I,a,Y}(T)}{rf_{I,a,1}(T)}$. This polynomial has its coefficients in $\mathbb{Z}$ since $rf_{I,a,Y}(T)$ and $rf_{I,a,1}(T)$ are in $\mathbb{Z}[T]$. Moreover, since $rf_{I,a,Y}(T)$ and $rf_{I,a,1}(T)$ have bitsize in $\widetilde{O}(d^2 + d\tau)$, $r^d f_F(T)$ can be computed, similarly as above, in $\widetilde{O}_B(d^7 + d^6\tau)$ and it has bitsize in $\widetilde{O}(d^3 + d^2\tau)$. The primitive part of $f_F(T)$ has also bitsize in $\widetilde{O}(d^3 + d^2\tau)$ (since it is smaller than or equal to that of $r^d f_F(T)$) and it can be computed from $r^d f_F(T)$ with $\widetilde{O}_B(d^3(d^3 + d^2\tau))$ bit operations by computing $O(d^3)$ gcds of integers having bitsize $\widetilde{O}(d^3 + d^2\tau)$ (Yap, 2000, §2.A.6).

Signs of F and f_F. It remains to show that the sign of F at a real solution of $I = \langle P, Q\rangle$ is the sign of f_F at the corresponding root of $f_{I,a}$ via the mapping of the RUR. By Lemma 21, there is a one-to-one mapping between the roots of $f_{I,a}$ and those of $I = \langle P, Q\rangle$ that maps a root γ of $f_{I,a}$ to a solution $(\alpha, \beta) = (\frac{f_{I,a,X}(\gamma)}{f_{I,a,1}(\gamma)}, \frac{f_{I,a,Y}(\gamma)}{f_{I,a,1}(\gamma)})$ of I such that $\gamma = \alpha + a\beta$ and $f_{I,a,1}(\gamma) \neq 0$. For any such pair of γ and (α, β), $f_F(\gamma) = f_{I,a,1}^d(\gamma)F(\gamma - a\frac{f_{I,a,Y}(\gamma)}{f_{I,a,1}(\gamma)}, \frac{f_{I,a,Y}(\gamma)}{f_{I,a,1}(\gamma)})$ by definition of $f_F(T)$, and thus $f_F(\gamma) = f_{I,a,1}^d(\gamma)F(\alpha, \beta)$. It follows that $f_F(\gamma)$ and $F(\alpha, \beta)$ have the same sign since $f_{I,a,1}(\gamma) \neq 0$ and d is even by hypothesis. □

Naive algorithm. The knowledge of a RUR $\{f_{I,a}, f_{I,a,1}, f_{I,a,X}, f_{I,a,Y}\}$ of $I = \langle P, Q\rangle$ yields a straightforward algorithm for computing the sign of F at a real solution of I. Indeed, it is sufficient to isolate the real roots of $f_{I,a}$, so that the intervals are also isolating for $f_{I,a}f_F$, and then to evaluate the sign of $\overline{f_F}$ at the endpoints of these isolating intervals. We analyze the complexity of this straightforward algorithm before describing our more subtle and more efficient algorithm. We provide this analysis for several reasons: first it answers a natural question, second it shows that even a RUR-based naive algorithm performs better than the state of the art.

Lemma 38. *Given a RUR $\{f_{I,a}, f_{I,a,1}, f_{I,a,X}, f_{I,a,Y}\}$ of $I = \langle P, Q\rangle$ (satisfying the bounds of Theorem 22) and an isolating interval for a real root γ of $f_{I,a}$, the sign of F at the real solution of I that corresponds to γ can be computed with $\widetilde{O}_B(d^9 + d^8\tau)$ bit operations.*

Proof. By Lemma 37, $pp(f_F)$ has degree $O(d^3)$ and bitsize $\widetilde{O}(d^3 + d^2\tau)$, and it can be computed with $\widetilde{O}_B(d^7 + d^6\tau)$ bit operations. By Theorem 22, $f_{I,a}$ has degree $O(d^2)$ and bitsize $\widetilde{O}(d^2 + d\tau)$,

thus the product $pp(f_F)\, f_{I,a}$ has degree $O(d^3)$ and bitsize $\widetilde{O}(d^3 + d^2\tau)$. By Lemma 34, the root separation bound of $pp(f_F)\, f_{I,a}$ has bitsize $\widetilde{O}(d^6 + d^5\tau)$. We refine the isolating interval of γ for $f_{I,a}$ to a width less than the root separation bound of $pp(f_F)\, f_{I,a}$, which can be done with $\widetilde{O}_B((d^2)^3 + (d^2)^2(d^2 + d\tau) + d^2(d^6 + d^5\tau)) = \widetilde{O}_B(d^8 + d^7\tau)$ bit operations according to Lemma 33. Furthermore, we can ensure that the new interval has rational endpoints with bitsize $\widetilde{O}(d^6 + d^5\tau)$, similarly as in the proof of Proposition 35. On the other hand, by Lemma 4, since $pp(f_F)$ has bitsize $\widetilde{O}(d^3 + d^2\tau)$, its squarefree part $\overline{pp(f_F)}$ can be computed in complexity $\widetilde{O}_B((d^3)^2(d^3 + d^2\tau)) = \widetilde{O}_B(d^9 + d^8\tau)$ and it has bitsize in $\widetilde{O}_B(d^3 + d^2\tau)$. It then follows from Lemma 6 that the evaluation of $\overline{pp(f_F)}$ at the endpoints of the refined interval can be done with $\widetilde{O}_B(d^3(d^6 + d^5\tau))$ bit operations which concludes the proof by Lemma 37. □

Improved algorithm. Our more subtle algorithm is, in essence the one presented by Diochnos et al. for evaluating the sign of a univariate polynomial (here $pp(f_F)$) at the roots of a squarefree univariate polynomial (here $\overline{f_{I,a}}$) (Diochnos et al., 2009, Corollary 5). The idea of this algorithm comes originally from Lickteig and Roy (2001), where the Cauchy index of two polynomials is computed by means of sign variations of a particular remainder sequence called the Sylvester-Habicht sequence. In Diochnos et al. (2009), this approach is slightly adapted to deduce the sign from the Cauchy index (Yap, 2000, Theorem 7.3) and the bit complexity is given in terms of the two initial degrees and bitsizes. Unfortunately, the corresponding proof is problematic because the authors refer to two complexity results for computing parts of the Sylvester-Habicht sequences and none of them actually applies.[23] Following the spirit of their approach, we present in Lemma 39 a new (weaker) complexity result for evaluating the sign of a univariate polynomial at the roots of a squarefree univariate polynomial. This result is used to derive the bit complexity of evaluating the sign of a bivariate polynomial at the roots of the system. For clarity, we postpone the proof of this lemma to Section 5.2.1 after Theorem 40.

Lemma 39. *Let $f \in \mathbb{Z}[X]$ be a squarefree polynomial of degree d_f and bitsize τ_f, and (a, b) be an isolating interval of one of its real roots γ with a and b distinct rationals of bitsize in $\widetilde{O}(d_f\tau_f)$ and $f(a)f(b) \neq 0$. Let $g \in \mathbb{Z}[X]$ be of degree d_g and bitsize τ_g. The sign of $g(\gamma)$ can be computed in $\widetilde{O}_B((d_f^3 + d_g^2)\tau_f + (d_f^2 + d_f d_g)\tau_g)$ bit operations. The sign of g at all the real roots of f can be computed with $\widetilde{O}_B((d_f^3 + d_f^2 d_g + d_g^2)\tau_f + (d_f^3 + d_f d_g)\tau_g)$ bit operations.*

Theorem 40. *Given a RUR $\{f_{I,a}, f_{I,a,1}, f_{I,a,X}, f_{I,a,Y}\}$ of $I = \langle P, Q\rangle$ (satisfying the bounds of Theorem 22), the sign of F at a real solution of I can be computed with $\widetilde{O}_B(d^8 + d^7\tau)$ bit operations. The sign of F at all the solutions of I can be computed with $\widetilde{O}_B(d^9 + d^8\tau)$ bit operations.*

Proof. By Lemma 37, the sign of F at the real solutions of I, is equal to the sign of $pp(f_F)$ at the corresponding roots of $f_{I,a}$, or equivalently at those of $\overline{pp(f_{I,a})}$. Furthermore, $pp(f_F)$ has degree $O(d^3)$, bitsize in $\widetilde{O}(d^3 + d^2\tau)$, and it can be computed with $\widetilde{O}_B(d^7 + d^6\tau)$ bit operations. On the other hand, by Theorem 22 and Proposition 28, the primitive part of $f_{I,a}$ has degree at most d^2 and bitsize in $\widetilde{O}(d^2 + d\tau)$. Since $f_{I,a}$ is monic (see Equation (6)), its primitive part can be

[23] Precisely, their proof is based on their Proposition 1 which claims, based on Lickteig and Roy (2001) and Reischert (1997) that given two polynomials f and g of degree $p > q$ and bitsize in $O(\tau)$, any of their polynomial subresultants as well as the whole quotient chain corresponding to the subresultant sequence can be computed with $\widetilde{O}_B(pq\tau)$ bit operations. However, in Lickteig and Roy (2001) the complexity results are not stated in terms of p and q but only in terms of the maximum degree while in Reischert (1997), the result assumes that the $(q-1)^{th}$ subresultant of f and g is known.

computed by multiplying it by the lcm of the denominators of its coefficients. This lcm can be computed with $O(d^2)$ lcms of integers whose bitsizes remain in $\widetilde{O}(d^2 + d\tau)$ (since $f_{I,a}$ is monic and its primitive part has bitsize in $\widetilde{O}(d^2 + d\tau)$). Each lcm can be computed with $\widetilde{O}_B(d^2 + d\tau)$ bit operations (Yap, 2000, §2.A.6), thus $pp(f_{I,a})$ can be computed in $\widetilde{O}(d^4 + d^3\tau)$ bit operations.[24] The squarefree part of $pp(f_{I,a})$ can thus be computed in $\widetilde{O}_B(d^4(d^2 + d\tau))$ bit operations and it has bitsize in $\widetilde{O}(d^2 + d\tau)$, by Lemma 4. By Lemmas 33 and 34, the isolating intervals (if not given) of $\overline{pp(f_{I,a})}$ can be computed in $\widetilde{O}_B((d^2)^3 + (d^2)^2(d^2 + d\tau))$ bit operations with intervals endpoints of bitsize satisfying the hypotheses of Lemma 39. Indeed, we can ensure during the isolation of the roots of $f = \overline{pp(f_{I,a})}$ that the isolating intervals have endpoints with bitsize in $\widetilde{O}(d_f\tau_f)$, similarly as in the proof of Proposition 35. Applying Lemma 39 then concludes the proof. □

Remark 41. *Theorem 40 also holds if the solutions of $I = \langle P, Q\rangle$ are described by the rational parameterization of González-Vega and El Kahoui (1996) instead of a RUR. Indeed, such parameterization is defined, in the worst case, by $\Theta(d)$ univariate polynomials f_i of degree d_{f_i} whose sum d_f is at most d^2, and by associated rational one-to-one mappings which are defined, as for the RUR, by polynomials of degree $O(d^2)$ and bitsize $O(d^2 + d\tau)$. The result of Theorem 40 on the sign of F at* one *real solution of I thus trivially still holds. For the sign of F at* all *real solutions of I the result also follows from the following observation. In the proofs of Lemmas 42 and 39, the computation of one sequence of unevaluated Sylvester-Habicht transition matrices has complexity $\widetilde{O}_B(pH)$ (in proof of Lemma 42) where p is in $O(d_{f_i} + d_g)$ in the proof of Lemma 39. The sum of the pH over all i is thus $O((d_f + dd_g)H)$ instead of $O((d_f + d_g)H)$ as for the RUR. However, d_gH writes in the proof of Lemma 39 as $\widetilde{O}(d_g((d_f + d_g)\tau_f + d_f(\tau_f + \tau_g))) = \widetilde{O}(d_fd_g(\tau_f + \tau_g) + d_g^2\tau_f)$ which writes in the proof of Theorem 40 as $\widetilde{O}(d^2d^3(d^3 + d^2\tau) + (d^3)^2(d^2 + d\tau)) = \widetilde{O}(d^8 + d^7\tau)$. Thus multiplying this by d remains within the targeted bit complexity. On the other hand, the complexity of the evaluation phase in the proofs of Lemmas 42 and 39 does not increase when considering the representation of Gonzalez-Vega and El Kahoui instead of the RUR because the total complexity of the evaluations depends only on the number of solutions at which we evaluate the sign of the other polynomial and on the degree and bitsize of the polynomials involved (values which do not increase in Gonzalez-Vega and El Kahoui representation; only the number of polynomials is larger).*

5.2.1. Proof of Lemma 39

As shown in Basu et al. (2006, Theorem 2.61), the sign of $g(\gamma)$ is $V(SRemS(f, f'g; a, b))$ where $V(SRemS(P, Q; a, b))$ is the number of sign variations in the signed remainder sequence of P and Q evaluated at a minus the number of sign variations in this sequence evaluated at b; see Definition 1.7 in Basu et al. (2006) for the sequence and Notation 2.32 for the sign variation. On the other hand, for any P and Q such that $\deg(P) > \deg(Q)$ and $P(a)\,P(b) \neq 0$ or $Q(a)\,Q(b) \neq 0$, we have according to Roy (1996, Theorems 3.2, 3.18 & Remarks 3.9, 3.25)[25] that $V(SRemS(P, Q; a, b)) = W(SylH(P, Q; a, b))$ where $SylH$ is the Sylvester-Habicht sequence of P and Q, and W is the related sign variation function.[26] The following intermediate result is

[24] Notice that if $f_{I,a}$ has been computed using Proposition 23, then instead of computing $pp(f_{I,a})$ one can consider $R(T, a) = f_{I,a}(T)\,L_R(a)$ which is a polynomial of degree $O(d^2)$ with integer coefficients of bitsize $\widetilde{O}(d^2 + d\tau)$ by Lemma 7.

[25] The same result can be found directly stated, in French, in Lombardi (1990, Theorem 4).

[26] The Sylvester-Habicht sequence, defined in Basu et al. (2006, §8.3.2.2) as the Signed Subresultant sequence, can be derived from the classical subresultant sequence (El Kahoui, 2003) by multiplying the two starting subresultants by $+1$ the next two by -1 and so on. W is defined as the usual sign variation with the following modification for groups of

a consequence of an adaptation of Lickteig and Roy (2001, Theorem 5.2) in the case where the polynomials P and Q have different degrees and bitsizes.

Lemma 42. *Let P and Q in $\mathbb{Z}[X]$ with $deg(P) = p > q = deg(Q)$ and bitsize respectively τ_P, τ_Q. If a and b are two rational numbers of bitsize bounded by σ, the computation of $W(SylH(P, Q; a, b))$ can be performed with $\widetilde{O}_B((p + q^2)\sigma + p(p\tau_Q + q\tau_P))$ bit operations.*

Moreover, if a_ℓ and b_ℓ, $1 \leqslant \ell \leqslant u$, are rational numbers of bitsizes that sum to σ, the computation of $W(SylH(P, Q; a_\ell, b_\ell))$ can be performed for all ℓ with $\widetilde{O}_B((p+q^2)\sigma+(p+qu)(p\tau_Q+q\tau_P) + pu\tau_P)$ bit operations.

Proof. Following the algorithm in Lickteig and Roy (2001), we first compute the consecutive Sylvester-Habicht transition matrices of P and Q denoted by $\mathcal{N}_{j,i}$ with $0 \leqslant j < i \leqslant p$. These matrices link consecutive regular couples[27] (Sh_i, Sh_{i-1}) and (Sh_j, Sh_{j-1}) in the Sylvester-Habicht sequence as follows:

$$\begin{pmatrix} Sh_j \\ Sh_{j-1} \end{pmatrix} = \mathcal{N}_{j,i} \begin{pmatrix} Sh_i \\ Sh_{i-1} \end{pmatrix} \text{ such that } i \leqslant p \text{ and } (Sh_p, Sh_{p-1}) = (P, Q). \tag{12}$$

According to Lickteig and Roy (2001, Theorem. 5.2 & Corollary 5.2), computing all the matrices $\mathcal{N}_{j,i}$ of P and Q can be done with $\widetilde{O}_B(pH)$ bit operations, where $H \in \widetilde{O}(q\tau_P + p\tau_Q)$ is an upper bound on the bitsize appearing in the computations given by Hadamard's inequality.

We evaluate the Sylvester-Habicht sequence at a rational a by first evaluating P, Q, and all the matrices $\mathcal{N}_{j,i}$ at a, and then by applying iteratively the above formula. Doing the same at b yields $W(SylH(P, Q; a, b))$.

First, note that the evaluation of $P(a)$ and $Q(a)$ can be done with $\widetilde{O}_B(p(\tau_P+\sigma))$ plus $\widetilde{O}_B(q(\tau_Q+\sigma))$, that is $\widetilde{O}_B(p(\tau_P + \tau_Q + \sigma))$ bit operations (since $p > q$), by Lemma 6. The polynomials appearing in the matrices $\mathcal{N}_{j,i}$ have bitsize at most H and the sum of their degrees is equal to p (Lickteig and Roy, 2001, Corollary 4.3).[28] Thus, all $\mathcal{N}_{j,i}(a)$ have bitsize $\widetilde{O}(p\sigma + H)$ and they can be computed in a total of $\widetilde{O}_B(p(\sigma+H))$ bit operations, by Lemma 6. Moreover, by considering the matrices $\mathcal{N}_{j,i}$ other than the first one $\mathcal{N}_{k,p}$, as the consecutive transition matrices of the Sylvester-Habicht sequence of the first regular couple (Sh_k, Sh_{k-1}) after (Sh_p, Sh_{p-1}), we have that the polynomials appearing in these matrices have the sum of their degrees equal to that of Sh_k which is at most q (since $k \leqslant p - 1$ and $Sh_{p-1} = Q$). Thus, except the first one $\mathcal{N}_{k,p}(a)$, all evaluated matrices $\mathcal{N}_{j,i}(a)$ have bitsize $\widetilde{O}(q\sigma + H)$ and they can be computed in a total of $\widetilde{O}_B(q(\sigma + H))$ bit operations.

We now apply iteratively Equation (12) for computing all the $Sh_i(a)$. Since all Sylvester-Habicht polynomials have bitsize at most H and degree at most q except the first one $Sh_p = P$, the bitsize of $Sh_{i<p}(a)$ is in $O(q\sigma + H)$ and that of $Sh_p(a)$ is in $O(p\sigma + \tau_P)$. Given $P(a)$, $Q(a)$ and all $\mathcal{N}_{j,i}(a)$, it follows from their bitsizes that we can compute iteratively the $Sh_i(a)$ in time

two consecutive zeros: count *one* sign variation for the groups $[+, 0, 0, -]$ and $[-, 0, 0, +]$, and *two* sign variations for the groups $[+, 0, 0, +]$ and $[-, 0, 0, -]$; see Basu et al. (2006, §9.1.3 Notation 9.11).

[27] Regular couples in the Sylvester-Habicht sequence are the nonzero Sylvester-Habicht polynomials (Sh_i, Sh_{i-1}) such that $\deg(Sh_i) > \deg(Sh_{i-1})$.

[28] Lickteig and Roy (2001, Corollary 4.3) states that consecutive Sylvester-Habicht transition matrices consist of one zero, two integers and a polynomial which is, up to a coefficient, the quotient of the division of two consecutive Sylvester-Habicht polynomials. These polynomials being proportional to polynomials in the remainder sequence of (P, Q), the sum of the degrees of their quotients is equal to the degree of P.

$\widetilde{O}_B(p\sigma + H)$ for the first regular couple after $(Sh_p, Sh_{p-1}) = (P, Q)$ and in time $\widetilde{O}_B(q\sigma + H)$ for each of the others. Thus, for computing of $W(SylH(P, Q; a, b))$, the initial computation of all $\mathcal{N}_{j,i}$ takes $\widetilde{O}_B(pH)$ bit operations and the evaluation phase takes $\widetilde{O}_B(p(\tau_P + \tau_Q + \sigma))$ plus $\widetilde{O}_B(p(\sigma+H)+q(q\sigma+H))$ bit operations, which gives a total of $\widetilde{O}_B(p(\sigma+H)+q^2\sigma)$ bit operations.

We now consider the case of computing $W(SylH(P, Q; a_\ell, b_\ell))$ for $1 \leqslant \ell \leqslant u$. We slightly change the above algorithm as follows. We only change the way to evaluate the first regular couple (Sh_k, Sh_{k-1}) after (Sh_p, Sh_{p-1}) at the a_ℓ (and b_ℓ). Once the matrices $\mathcal{N}_{j,i}$ have been computed, we compute the (non-evaluated) first regular couple $(Sh_k, Sh_{k-1}) = \mathcal{N}_{k,p}(Sh_p, Sh_{p-1})$. Since the polynomials in $\mathcal{N}_{k,p}$ have degree at most p and bitsize at most H, the couple (Sh_k, Sh_{k-1}) can be computed in $\widetilde{O}_B(p(H + \tau_P + \tau_Q)) = \widetilde{O}_B(pH)$ time (von zur Gathen and Gerhard, 2003, Corollary 8.27). As noted above, Sh_k, and thus also Sh_{k-1}, have degree at most q and they have bitsize at most H, so they can be evaluated at a given a_ℓ in time $\widetilde{O}_B(q(\sigma_\ell + H))$ where σ_ℓ is the bitsize of a_ℓ. Now, the polynomials appearing in the matrices $\mathcal{N}_{j,i}$, other than the first one $\mathcal{N}_{k,p}$, have bitsize at most H and the sum of their degrees is at most q, so similarly as above, all the $\mathcal{N}_{j,i}(a_\ell)$, except $\mathcal{N}_{k,p}(a_\ell)$, can be computed in total bit complexity $\widetilde{O}_B(q(\sigma_\ell + H))$. Then, we compute as above each of the other regular couples evaluated at a_ℓ in time $\widetilde{O}_B(q\sigma_\ell + H)$. Hence, the initial computation of all $\mathcal{N}_{j,i}$ and of (Sh_k, Sh_{k-1}) takes $\widetilde{O}_B(pH)$ bit operations and the evaluation phase at all the a_ℓ takes the sum over ℓ, $1 \leqslant \ell \leqslant u$, of $\widetilde{O}_B(p(\tau_P + \tau_Q + \sigma_\ell))$ plus $\widetilde{O}_B(q(\sigma_\ell + H) + q(q\sigma_\ell + H))$ bit operations, that is $\widetilde{O}_B(p(\tau_P + \tau_Q) + (p + q^2)\sigma_\ell + qH)$ which sums to $\widetilde{O}_B(pu(\tau_P + \tau_Q) + (p + q^2)\sigma + quH)$. Hence the total bit complexity for computing all the $W(SylH(P, Q; a_\ell, b_\ell))$ for $1 \leqslant \ell \leqslant u$ is $\widetilde{O}_B((p + q^2)\sigma + (p + qu)H + pu\tau_P)$ which concludes the proof. □

Proof of Lemma 39. We may assume that g has degree greater than one since, if g is a constant the problem is trivial and, if $g(X) = cX - d$, then the sign of $g(\gamma)$ follows from (i) the sign of c if $\frac{d}{c} \notin (a, b)$ and from (ii) the signs of c, $f(a)$, and $f(\frac{d}{c})$ if $\frac{d}{c} \in (a, b)$; indeed, the signs of $f(a) \neq 0$ and $f(\frac{d}{c})$ determine whether γ lies in $(a, \frac{d}{c})$, $\{\frac{d}{c}\}$, or $(\frac{d}{c}, b)$. Hence, when g has degree one, the sign of $g(\gamma)$ can be computed with $\widetilde{O}_B(d_f(\tau_g + d_f\tau_f))$ bit operations according to Lemma 6.

Recall that the sign of $g(\gamma)$ is $V(SRemS(f, f'g; a, b))$ (Basu et al., 2006, Theorem 2.61). When g has degree greater than one, we cannot directly apply Lemma 42 since $\deg(f) < \deg(f'g)$. However, knowing the sign of f and $f'g$ at a and b and noticing that their signed remainder sequence starts with $[f, f'g, -f, -rem(f'g, -f), \ldots]$, we can easily compute the value c such that $V(SRemS(f, f'g; a, b)) = V(SRemS(f'g, -f; a, b)) + c$. Furthermore, as observed at the beginning of this section and since $f(a)f(b) \neq 0$ by hypothesis, $V(SRemS(f'g, -f; a, b)) = W(SylH(f'g, -f; a, b))$. We can now apply Lemma 42 which thus yields the sign of $g(\gamma)$ with a bit complexity in $\widetilde{O}_B((p+q^2)\sigma+p(p\tau_Q+q\tau_P))$ which simplifies into $\widetilde{O}_B((d_f^3+d_g^2)\tau_f+(d_f^2+d_fd_g)\tau_g)$.

For the sign of g at all the real roots of f, isolating intervals of these roots can be computed in complexity $\widetilde{O}_B(d_f^3+d_f^2\tau_f)$ (see Lemma 33) such that the bitsizes of the interval endpoints sum up to $\widetilde{O}(d_f^2 + d_f\tau_f)$ (a consequence of Davenport-Mahler-Mignotte bound, see e.g. Diochnos et al. (2009, Lemma 6)). Similarly as for one root, Lemma 42 then yields that the sign of g at all the real roots of f can be computed with a bit complexity in $\widetilde{O}_B((p+q^2)\sigma+(p+qu)(p\tau_Q+q\tau_P)+pu\tau_P)$ which writes as $\widetilde{O}_B((d_f+d_g+d_f^2)d_f\tau_f+(d_f+d_g+d_f^2)((d_f+d_g)\tau_f+d_f(\tau_f+\tau_g))+(d_f+d_g)d_f(\tau_g+\tau_f))$ and simplifies into $\widetilde{O}_B((d_f^3 + d_f^2d_g + d_g^2)\tau_f + (d_f^3 + d_fd_g)\tau_g)$ bit operations. □

5.3. Over-constrained systems

So far, we focused on systems defined by exactly two coprime polynomials. We now extend our results to compute rational parameterizations of zero-dimensional systems defined with additional equality or inequality. Let $P, Q \in \mathbb{Z}[X, Y]$ be two coprime polynomials of total degree at most d and maximum bitsize τ. In this section, we assume given $RUR_{I,a} = \{f_{I,a}, f_{I,a,1}, f_{I,a,X}, f_{I,a,Y}\}$ the RUR of the ideal $I = \langle P, Q\rangle$ associated to the separating form $X + aY$, we also assume that the polynomials of this RUR satisfy the bitsize bound of Theorem 22. Given another polynomial $F \in \mathbb{Z}[X, Y]$, we have seen in the previous section how to compute the sign of F at the solutions of I. With a similar approach, we now explain how to split $RUR_{I,a}$ according to whether F vanishes or not at the solutions of I.

Let $F \in \mathbb{Z}[X, Y]$ be of total degree at most d and maximum bitsize τ. Identifying the roots of $f_{I,a}$ with the solutions of the system I via the RUR, let $f_{F=0}$ (resp. $f_{F\neq 0}$) be the squarefree factor of $f_{I,a}$ such that its roots are exactly the solutions of the system I at which the polynomial F vanishes (resp. does not vanish).

Lemma 43. *Given $RUR_{I,a}$, the bit complexity of computing $f_{F=0}$ (resp. $f_{F\neq 0}$) is in $\widetilde{O}_B(d^8 + d^7\tau)$ and these polynomials have bitsize in $\widetilde{O}(d^2 + d\tau)$.*

Proof. The polynomial f_F (not to be confused with $f_{F=0}$ or $f_{F\neq 0}$), as defined in Lemma 37, has the same sign as F at the real solutions of the system I. The same holds for complex solutions by considering the "sign" as zero or nonzero. The roots of the squarefree polynomial $f_{F=0} = \gcd(\overline{f_{I,a}}, f_F)$ thus are the $\alpha + a\beta$ with (α, β) solution of I and $F(\alpha, \beta) = 0$. The polynomial $f_{F\neq 0}$ defined as the gcd-free part of $\overline{f_{I,a}}$ with respect to f_F is also squarefree and encodes the solutions such that $F(\alpha, \beta) \neq 0$.

According to Lemma 37 and the proof of Theorem 40, the primitive part of f_F and $\overline{f_{I,a}}$ can be computed in, respectively, $\widetilde{O}_B(d^7 + d^6\tau)$ and $\widetilde{O}_B(d^4(d^2 + d\tau))$ bit operations. Moreover, these integer polynomials have, respectively, bitsize $\widetilde{O}(d^3 + d^2\tau)$ and $\widetilde{O}(d^2 + d\tau)$ and degree $O(d^3)$ and $O(d^2)$. Thus, by Lemma 5, their gcd and the gcd-free part of $\overline{f_{I,a}}$ with respect to f_F, i.e. $f_{F=0}$ and $f_{F\neq 0}$, can be computed with $\widetilde{O}_B(d^8 + d^7\tau)$ bit operations and they have bitsize in $\widetilde{O}(d^2 + d\tau)$. □

For several equality or inequality constraints, iterating this splitting process gives a parameterization of the corresponding set of constraints. It is worth noticing that the set of polynomials $\{f_{F=0}, f_{I,a,1}, f_{I,a,X}, f_{I,a,Y}\}$ defines a rational parameterization of the solutions of the ideal $\langle P, Q, F\rangle$, but this is not a RUR of this ideal (in the sense of Definition 20). First, because multiplicities are lost in the splitting process and second because the coordinate polynomials of the parameterization are still those of the ideal I. Still, it is possible to compute a RUR of the radical of the corresponding ideal (and similarly for the ideal corresponding to $F \neq 0$):

Proposition 44. *Given $RUR_{I,a}$ and $F \in \mathbb{Z}[X, Y]$ of total degree at most d and maximum bitsize τ, the bit complexity of computing the RUR of the radical of the ideal $\langle P, Q, F\rangle$ is in $\widetilde{O}_B(d^8 + d^7\tau)$.*

Proof. Denote by J the radical of the ideal $\langle P, Q, F\rangle$. The polynomial $f_{F=0}$ computed in Lemma 43 is the first polynomial $f_{J,a}$ of $RUR_{J,a}$. Indeed, it vanishes at the solutions of this ideal (with identification of the roots of $f_{J,a}$ with the solutions of the system J) and it is squarefree. Then Proposition 23 yields that $f_{J,a,1}$ is the gcd-free part of $f'_{J,a}$ with respect to $f_{J,a}$. As in the proof of Theorem 40, $pp(f_{J,a})$ can be computed in $\widetilde{O}_B(d^4 + d^3\tau)$ and has bitsize in $\widetilde{O}(d^2 + d\tau)$. By Lemma 5, applied to $pp(f_{J,a})$ and its derivative, $f_{J,a,1}$ can be computed in $\widetilde{O}_B(d^6 + d^5\tau)$.

According to Lemma 21, the X-coordinates of the solutions of J are given by the polynomial fraction $\frac{f_{J,a,X}}{f_{J,a,1}}$ at the roots of $f_{J,a}$. On the other hand, the solutions of J, seen as solutions of I, have their X-coordinates defined by the polynomial fraction $\frac{f_{I,a,X}}{f_{I,a,1}}$. This thus implies that $f_{J,a,X} = f_{I,a,1}^{-1} f_{I,a,X} f_{J,a,1}$ modulo $f_{J,a}$. The computation of $f_{I,a,1}^{-1}$ together with the multiplication with other polynomials of the RUR has already been studied in the proof of Proposition 35; this can be done in $\widetilde{O}_B(d^6 + d^5\tau)$ time and gives a polynomial of degree $O(d^2)$ and bitsize $\widetilde{O}(d^4 + d^3\tau)$. It remains to compute the remainder of the division of this polynomial with $f_{J,a}$, which can be done in a soft bit complexity of the order of the square of the maximum degree times the maximum bitsize, i.e. $\widetilde{O}_B(d^8 + d^7\tau)$ (von zur Gathen and Gerhard, 2003, Theorem 9.6 and subsequent discussion). A similar computation gives the polynomial $f_{I,a,Y}$, hence the computation of $RUR_{J,a}$ can be done in $\widetilde{O}_B(d^8 + d^7\tau)$ bit operations. □

6. Conclusion

We addressed the problem of solving systems of bivariate polynomials with integer coefficients via rational parameterizations. Our first contribution concerns the computation of a separating linear form, problem which is at the core of approaches based on rational parameterizations. We presented an algorithm of worst-case bit complexity $\widetilde{O}_B(d^8 + d^7\tau)$ for finding a separating linear form of such systems (of polynomials of degree at most d and coefficients of bitsize at most τ), improving by a factor d^2 the best known algorithm for this problem. Our second contribution focuses on the Rational Univariate Representation (RUR) (Rouillier, 1999). We first showed that the polynomials of the RUR of a system of two polynomials can be expressed by simple formulas which yield a new simple method for computing the RUR and also yield a new bound on the bitsize of these polynomials. This new bound implies, in particular, that the total space complexity of such RURs is, in the worst case, $\Theta(d)$ smaller than the alternative rational parameterization introduced by González-Vega and El Kahoui (1996). Given a RUR, this new bound also yields some improvements on the complexity of computing isolating boxes and performing sign_at evaluations. These improvements also hold for the rational parameterization of Gonzalez-Vega and El Kahoui. We also addressed the problem of computing RURs of over-constrained systems.

Interestingly, computing a separating linear form remains the bottleneck, in terms of worst-case bit complexity, in the computation of rational parameterizations of bivariate systems (at least for the one of González-Vega and El Kahoui (1996) and for the RUR). Indeed, even though we have decreased this complexity to $\widetilde{O}_B(d^8 + d^7\tau)$, the worst-case complexity of computing the rational parameterization of Gonzalez-Vega and El Kahoui was in $\widetilde{O}_B(d^7 + d^6\tau)$ and we have decreased the complexity of computing RURs to the same bound.

Given these new worst-case bounds, two particular problems of interest are the design of theoretically efficient randomized algorithms and practically efficient algorithms and implementations. It should be stressed that the algorithm we presented for computing a RUR has presumably little practical interest because the computation of the resultant $R(T, S)$ of trivariate polynomials is not very efficient in practice. Concerning probabilistic algorithms, even though the computation of a separating form is the worst-case bit-complexity bottleneck, in a Monte-Carlo probabilistic setting, a linear form chosen uniformly at random in a set of cardinality kd^4 is separating with probability at least $1 - \frac{1}{k}$. However, checking that a linear form is separating is essentially as difficult as computing a separating form. One possible approach in a Las-Vegas probabilistic setting, it is to choose a candidate separating form randomly, compute a RUR-candidate and

verify a posteriori using the RUR-candidate if the chosen candidate separating form is actually separating. Furthermore, our new bound on the bitsize of RURs can be used to derive practically efficient algorithms using multi-modular arithmetic. Parallelization is also quite natural in this context. Such an approach is the topic of current research and we refer to Bouzidi et al. (2011) for preliminary work on the subject. Note that the best known Las-Vegas algorithm for computing a separating linear form has expected bit complexity $\widetilde{O}_B(d^7 + d^6\tau)$ (Diatta et al., 2008; Diatta, 2009).[29] Mehlhorn et al. (2013) also recently showed that isolating boxes of the real solutions can be computed (without a rational parameterization) with an expected bit-complexity $\widetilde{O}_B(d^6 + d^5\tau)$ in a Las-Vegas algorithm.

Alonso, M.-E., Becker, E., Roy, M.-F., Wörmann, T., 1996. Multiplicities and idempotents for zerodimensional systems. In: Algorithms in Algebraic Geometry and Applications. Vol. 143 of Progress in Mathematics. Birkhäuser, pp. 1–20.

Aubry, P., 1999. Ensembles triangulaires de polynômes et résolution de systèmes algébriques. implantation en axiom. Ph.D. thesis, Université P. et M. Curie.

Basu, S., Pollack, R., Roy, M.-F., 2006. Algorithms in Real Algebraic Geometry, 2nd Edition. Vol. 10 of Algorithms and Computation in Mathematics. Springer-Verlag.

Bodrato, M., Zanoni, A., 2011. Long integers and polynomial evaluation with estrin's scheme. In: Proceedings of the 13th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. SYNASC '11. pp. 39–46.

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

Boulier, F., Chen, C., Lemaire, F., Maza, M. M., 2009. Real root isolation of regular chains. In: Proceedings of the 2009 Asian Symposium on Computer Mathematics (ASCM 2009). Math for Industry. pp. 1–15.

Bouzidi, Y., Lazard, S., Pouget, M., Rouillier, F., 2011. New bivariate system solver and topology of algebraic curves. In: 27th European Workshop on Computational Geometry - EuroCG.

Busé, L., Khalil, H., Mourrain, B., Sep. 2005. Resultant-based methods for plane curves intersection problems. In: Computer Algebra in Scientific Computing (CASC). Vol. 3718 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, Kalamata, Greece, pp. 75–92.

Canny, J., 1987. A new algebraic method for robot motion planning and real geometry. In: Proceedings of the 28th Annual Symposium on Foundations of Computer Science. SFCS'87. pp. 39–48.

Cheng, J., Lazard, S., Peñaranda, L., Pouget, M., Rouillier, F., Tsigaridas, E., 2010. On the topology of real algebraic plane curves. Mathematics in Computer Science 4, 113–137.

Cheng, J.-S., Gao, X.-S., Li., J., 2009. Root isolation for bivariate polynomial systems with local generic position method. In: Proceedings of the 34th International Symposium on Symbolic and Algebraic Computation. ISSAC'09. pp. 103–110.

Cheng, J.-S., Gao, X.-S., Yap, C. K., 2007. Complete numerical isolation of real zeros in zero-dimensional triangular systems. In: Proc. Int. Symp. on Symbolic and Algebraic Computation. pp. 92–99.

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

Dahan, X., Schost, E., 2004. Sharp estimates for triangular sets. In: Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation. ISSAC'04. pp. 103–110.

Diatta, D. N., 2009. Calcul effectif de la topologie de courbes et surfaces algébriques réelles. Phd thesis, Université de Limoge, France.

Diatta, D. N., Mourrain, B., Ruatta, O., 2008. On the computation of the topology of a non-reduced implicit space curve. In: Proceedings of the 21th International Symposium on Symbolic and Algebraic Computation. ISSAC'08. pp. 47–54.

Diochnos, D. I., Emiris, I. Z., Tsigaridas, E. P., 2009. On the asymptotic and practical complexity of solving bivariate systems over the reals. J. Symb. Comput. 44 (7), 818–835.

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

Emeliyanenko, P., Sagraloff, M., 2012. On the complexity of solving a bivariate polynomial system. In: Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation. ISSAC'12. pp. 154–161.

[29] The complexity analysis of the algorithm presented in Diatta et al. (2008) is presented in Diatta (2009, Theorem 3.3.9). Note that the given proof of complexity is incorrect because it assumes that an Euclidean division between two univariate polynomials of degree at most d and bitsize at most τ can be done in $\widetilde{O}_B(d\tau)$ rather than in $\widetilde{O}_B(d^2\tau)$ bit operations. However, replacing the Euclidean division routine by a divisibility test, which has complexity $\widetilde{O}_B(d^2 + d\tau)$ (von zur Gathen and Gerhard, 2003, §9), yields the stated complexity.

Fulton, W., 2008. Algebraic curves: an introduction to algebraic geometry. Personal reprint available at http://www.math.lsa.umich.edu/~wfulton/CurveBook.pdf.

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

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

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

Hart, W., Novocin, A., 2011. Practical divide-and-conquer algorithms for polynomial arithmetic. In: Proceedings of the 13th International Conference on Computer Algebra in Scientific Computing. CASC'11. pp. 200–214.

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

Lazard, D., 1983. Gröbner bases, Gaussian elimination, and resolution of systems of algebraic equations. In: EUROCAL'83 European Computer Algebra Conference. Vol. 162 of LNCS. Springer, pp. 146–156.

Lazard, D., 1992. Solving zero-dimensional algebraic systems. J. Symb. Comput. 13 (2), 117–132.

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

Lickteig, T., Roy, M.-F., 2001. Sylvester-Habicht Sequences and Fast Cauchy Index Computation. J. Symb. Comput. 31 (3), 315–341.

Lombardi, H., 1990. Sous-résultant, suite de sturm, spécialisation. Ph.D. thesis, Université de Franche Comté.

Mehlhorn, K., Sagraloff, M., Wang, P., 2013. From approximate factorization to root isolation. In: Proceedings of the 38th International Symposium on International Symposium on Symbolic and Algebraic Computation. ISSAC'13. pp. 283–290, http://arxiv.org/abs/1301.4870.

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

Pan, V. Y., 2002. Univariate polynomials: Nearly optimal algorithms for numerical factorization and root-finding. Journal of Symbolic Computation 33 (5), 701 – 733.

Reischert, D., 1997. Asymptotically fast computation of subresultants. In: Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation. ISSAC'97. pp. 233–240.

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

Roy, M.-F., 1996. Basic algorithms in real algebraic geometry and their complexity: from Strum theorem to the existential theory of reals. Lectures on Real Geometry in memoriam of Mario Raimondo, Gruyter Expositions in Mathematics. 23, 1–67.

Rump, S. M., 1979. Polynomial minimum root separation. Mathematics of Computation 33 (145), 327–336.

Sagraloff, M., 2012. 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. pp. 297–304.

Schost, E., 2001. Sur la résolution des systèmes polynomiaux à paramètres. Ph.D. thesis, Ecole Polytechnique, France.

Strzebonski, A. W., Tsigaridas, E. P., 2011. Univariate real root isolation in an extension field. In: Proceedings of the 36th International Symposium on International Symposium on Symbolic and Algebraic Computation. ISSAC'11. pp. 321–328.

Van der Waerden, B.-L., 1930. Moderne Algebra I. Springer, Berlin.

von zur Gathen, J., Gerhard, J., 2003. Modern Computer Algebra, 2nd Edition. Cambridge Univ. Press, Cambridge, U.K.

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

Zomorodian, A., Edelsbrunner, H., 2002. Fast software for box intersections. Internat. J. Comput. Geom. Appl. 12, 143–172.