Univariate Real Root Isolation in an Extension Field

Adam Strzeboński
Wolfram Research Inc.,100 Trade Centre Drive,
Champaign, IL 61820, U.S.A.
adams (AT) wolfram.com

Elias P. Tsigaridas
Computer Science Department
Aarhus University, Denmark
elias (AT) cs.au.dk

ABSTRACT

We present algorithmic, complexity and implementation results for the problem of isolating the real roots of a univariate polynomial in $B_\alpha \in L[y]$, where $L = \mathbb{Q}(\alpha)$ is a simple algebraic extension of the rational numbers. We revisit two approaches for the problem. In the first approach, using resultant computations, we perform a reduction to a polynomial with integer coefficients and we deduce a bound of $\widetilde{\mathcal{O}}_B(N^{10})$ for isolating the real roots of B_α, where N is an upper bound on all the quantities (degree and bitsize) of the input polynomials. In the second approach we isolate the real roots working directly on the polynomial of the input. We compute improved separation bounds for the roots and we prove that they are optimal, under mild assumptions. For isolating the real roots we consider a modified Sturm algorithm, and a modified version of DESCARTES' algorithm introduced by Sagraloff. For the former we prove a complexity bound of $\widetilde{\mathcal{O}}_B(N^8)$ and for the latter a bound of $\widetilde{\mathcal{O}}_B(N^7)$. We implemented the algorithms in C as part of the core library of MATHEMATICA and we illustrate their efficiency over various data sets. Finally, we present complexity results for the general case of the first approach, where the coefficients belong to multiple extensions.

Categories and Subject Descriptors: F.2 [Theory of Computation]: Analysis of Algorithms and Problem Complexity; I.1 [Computing Methodology]: Symbolic and algebraic manipulation: Algorithms

Keywords real root isolation, algebraic polynomial, field extension, separation bounds, Sturm, Descartes' rule of sign

General Terms Algorithms, Experimentation, Theory

1. INTRODUCTION

Real root isolation is a very important problem in computational mathematics. Many algorithms are known for isolating the real roots of a polynomial with integer or rational coefficients that are either based solely on operations with rational numbers, [8, 13, 24, 29] and references therein,

or they follow a numerical, but certified approach, [25, 33] and references therein. In this paper we consider a variation of the problem in which the coefficients of the polynomial are polynomial functions of a real algebraic number, that is they belong to a simple algebraic extension of the rationals.

Problem 1. *Let α be a real algebraic number with isolating interval representation $\alpha \cong (A, \mathcal{I})$, where $A = \sum_{i=0}^{m} a_i\, x^i$, $\mathcal{I} = [\mathsf{a}_1, \mathsf{a}_2]$, $\mathsf{a}_{1,2} \in \mathbb{Q}$ and $\deg(A) = m$ and $\mathcal{L}(A) = \tau$. Let $B_\alpha = \sum_{i=0}^{n} b_i(\alpha)\, y^i \in \mathbb{Z}(\alpha)[y]$ be square-free, where $b_i(x) = \sum_{j=0}^{\eta_i} c_{i,j}\, x^j \in \mathbb{Z}[x]$, $\mathcal{L}(c_{i,j}) \leq \sigma$, and $\eta_i < m$, for $0 \leq i \leq d$. What is the Boolean complexity of isolating the real roots of B_α?*

Rump [31], see also [30], presented an algorithm for the problem that is an extension of Collins and Loos [6] algorithm for integral polynomials. Johnson [17] presented and compared various algorithms for Problem 1. He considered a norm based algorithm that reduces the problem to root isolation of integral polynomial (this is the approach that we consider in Sec. 3) and extended three algorithms used for integral polynomials, i.e. Sturm (we present it in Sec. 4.2), the algorithm based on derivative sequence and Rolle's theorem [6], and the algorithm based on Descartes' rule of sign [5] (we present a modified version in Sec. 4.3). Johnson and Krandick [16] modified the latter and managed to replace exact arithmetic, when possible, with certified floating point operations; a novelty that speeds up considerably the computations. Along the same lines, Rouillier and Zimmermann [29] presented an optimal in terms of memory used algorithm for integral polynomials that exploits adaptive multiprecision techniques that could be used for Problem 1, if we approximate the real algebraic number up to a sufficient precision. In a series of works [11, 12, 22] a bitstream version of Descartes' algorithm was introduced. The coefficients of the input polynomial are considered to be real numbers that we can approximate up to arbitrary precision. We use the most recent version of this approach, which is due to Sagraloff [32], to tackle Problem 1. Last but not least, let us also mention the numerical algorithms due to Pan [25] and Schönhage [33], that could be also used if approximate α in our problem up to a sufficient precision.

Rioboo [28] considered various symbolic algorithms for operations with real algebraic numbers, based on quasi Sylvester sequences. These algorithms could be used for Problem 1, and they are closely connected with the Sturm algorithm

that we present (Sec. 4.2). However, we use different subalgorithms for sign evaluations and solving polynomials. The focus in [28] is on efficient implementation of the real closure in AXIOM.

Problem 1 is closely related to real root isolation of of triangular systems and regular chains. In [4, 20, 38, 39] algorithms and implementations are presented for isolating the real roots of triangular polynomial systems, based on interval arithmetic and the so-called *sleeve* polynomials. In the case of two variables the problem at study is similar to Problem 1. In this line of research the coefficients of the algebraic polynomial are replaced with sufficiently refined intervals, hence obtaining upper and lower bounds (i.e. a sleeve) for the polynomial. Isolation is performed using evaluations and exclusion predicates that involve the non-vanishing of the derivative. To our knowledge there is no complexity analysis of the algorithms. Neverthelss in [4] evaluation bounds are presented, which are crucial for the termination of the algorithm, based on separation bounds of polynomial systems. However, the systems used for the bounds involve the derivative of the polynomial (this is needed for the exclusion criterion), which is not the case for our approach. In [3] the problem of real root isolation of 0-dim square-free regular chains is considered. A generalization of Vincent-Collins-Akritas (or Descartes) algorithm is used to isolate the real roots of of polynomials with real algebraic numbers as coefficients. This approach is similar to the direct strategy that we study. To our knowledge the authors do not present a complexity analysis since they focus on efficient algorithms and implementation in MAPLE.

We revisit two approaches for isolating the real roots of a square-free polynomial with coefficients in a simple algebraic extension of the rational numbers. The first, indirect, approach (Sec. 3), already presented in [17], is to find a polynomial with integer coefficients which is zero at all roots of B_α, isolate its real roots, and identify the intervals which contain the roots of B_α. We compute (aggregate) separation bounds for the resulting polynomial (Lem. 7), that are slightly better than the ones in [31], and prove that the complexity of the algorithm is $\widetilde{\mathcal{O}}_B(N^{10})$, where N is an upper bound on all the quantities (degrees and bitsizes) of the input. The second approach (Sec. 4.1) is to isolate the roots of the input polynomial directly, using either the Sturm's algorithm or Sagraloff's modified Descartes algorithm. We analyze the worst-case asymptotic complexity of the algorithms and we obtained a bound of $\widetilde{\mathcal{O}}_B(N^8)$ and $\widetilde{\mathcal{O}}_B(N^7)$, respectively. We obtain these complexity bounds by estimating improved separation bounds for the roots (Sec. 4.1 and Lem. 9), that we also prove that they are optimal (Sec. 4.4). The bounds are better than the previously known ones [17, 30] by a factor of N. We empirically compare the performance of the indirect approach and the direct approach based on Sagraloff's modified Descartes algorithm. The algorithms were implemented in C as part of the core library of MATHEMATICA, and we illustrate their behavior on various datasets (Sec. 5). The complexity bounds that we present are many factors better that the previously known ones. However, a fair and explicit comparison with the bounds in [17] is rather difficult, if possible at all, since, besides the improved separation bounds that we present, the complexity bounds of many subalgorithms that are used have been dramatically improved over the last 20 years, and it is not clear how to take this into account in the comparison.

Finally, we present a generalization of the first approach to the case where the input polynomials are univariate, but with coefficients that belong to multiple extensions (Sec. 6). We derive (aggregate) separation bounds for this case (Lem. 12) and we sketch the overall complexity of the algorithm. The bounds are single exponential with respect to the number of extensions.

Notation. $\mathcal{O}_B$ means bit complexity and the $\widetilde{\mathcal{O}}_B$-notation means that we are ignoring logarithmic factors. For $A = \sum_{i=1}^d a_i x^i \in \mathbb{Z}[x]$, $\deg(A)$ denotes its degree. $\mathcal{L}(A)$ denotes an upper bound on the bitsize of the coefficients of A, including a bit for the sign. For $\mathsf{a} \in \mathbb{Q}$, $\mathcal{L}(\mathsf{a}) \geq 1$ is the maximum bitsize of the numerator and the denominator.

If $\alpha_1, \dots, \alpha_d$ are the distinct, possible complex, roots of A, then $\Delta_i = |\alpha_i - \alpha_{c_i}|$, where α_{c_i} is the roots closest to α_i. $\Delta(A) = \min_i \Delta_i(A)$ is the separation bound, that is the smallest distance between two (real or complex, depending on the context) roots of A. By $\Sigma(A) = -\sum_{i=1}^n \lg \Delta_i(A)$, we denote the numbers of bits needed to represent isolating rational numbers for all the roots of A.

Given two polynomials, possible multivariate, f and g, then $\mathtt{res}_x(f, g)$ denotes their resultant with respect to x.

2. PRELIMINARIES

Real algebraic numbers are the real roots of univariate polynomials with integer coefficients; let their set be $\mathbb{R}_{\mathsf{alg}}$. We represent them in the so-called *isolating interval representation*. If $\alpha \in \mathbb{R}_{\mathsf{alg}}$ then the representation consists of a square-free polynomial with integer coefficients, $A \in \mathbb{Z}[x]$, that has α as a real root, and an isolating interval with rational endpoints, $\mathcal{I} = [\mathsf{a}_1, \mathsf{a}_2]$, that contains α and no other root of the polynomial. We write $\alpha \cong (A, \mathcal{I})$.

The following proposition provides various bounds for the roots of a univariate polynomial. Various versions of the proposition could be found in e.g. [8, 10, 36]. We should mention that the constants that appear are not optimal. For multivariate bounds we refer to [15].

Proposition 1. *Let f be a univariate polynomial of degree p. If γ_i are the distinct real roots of f, then it holds*

$$|\gamma_i| \leq 2\|f\|_\infty \leq 2^{\tau+1} , \tag{1}$$

$$-\lg \Delta(f) \leq -\frac{1}{2} \lg |3\, \mathsf{disc}(f_{red})| + \frac{p+2}{2} \lg(p) +$$
$$(p-1) \lg \|f_{red}\|_2 \tag{2}$$
$$\leq 2p \lg p + p\tau ,$$

$$-\sum_i \lg \Delta_i(f) \leq -\frac{1}{2} \lg |\mathsf{disc}(f_{red})| + \frac{p^2 - p - 2}{2} +$$
$$(2p-1) \lg \|f_{red}\|_2 \tag{3}$$
$$\leq 3p^2 + 3p\tau + 4p \lg p ,$$

where f_{red} is the square-free part of f, and the second inequalities hold if we consider $f \in \mathbb{Z}[x]$ and $\mathcal{L}(f) = \tau$.

Proposition 2. *Let $f \in \mathbb{Z}[x]$ have degree p and bitsize τ. We compute the isolating interval representation of its real roots and their multiplicities in $\widetilde{\mathcal{O}}_B(p^5 + p^4\tau + p^3\tau^2)$ [32, 35]. The endpoints of the isolating intervals have bitsize $\mathcal{O}(p^2 + p\tau)$ and $\mathcal{L}(f_{red}) = \mathcal{O}(p + \tau)$, where f_{red} is the square-free part of f. If $N = \max\{p, \tau\}$ then complexity bound for isolation becomes $\widetilde{\mathcal{O}}_B(N^5)$.*

Proposition 3. *[9, 14] Given a real algebraic number $\alpha \cong$ $(f, [\mathsf{a}, \mathsf{b}])$, where $\mathcal{L}(\mathsf{a}) = \mathcal{L}(\mathsf{b}) = \mathcal{O}(p^2 + p\tau)$, and $g \in \mathbb{Z}[x]$, such that $\deg(g) = q$, $\mathcal{L}(g) = \sigma$, we compute $\mathrm{sign}(g(\alpha))$ in bit complexity $\widetilde{\mathcal{O}}_B(pq \max\{\tau, \sigma\} + p \min\{p, q\}^2 \tau)$.*

For the proofs of the following results the reader may refer to [9]. Let $f, g \in (\mathbb{Z}[x])[y]$ such that $\deg_x(f) = p$, $\deg_x(g) = q$, $\deg_y(f), \deg_y(g) \le d$, $\tau = \max(\mathcal{L}(f), \mathcal{L}(g))$. By $\mathbf{SR}(f, g\,; \mathsf{a})$ we denote the evaluation of the signed polynomial remainder sequence of f and g with respect to x over a, and by $\mathbf{SR}_j(f, g\,; \mathsf{a})$ the j-th element in this sequence.

Proposition 4. *We can compute $\mathrm{res}(f, g)$ w.r.t. x or y in $\widetilde{\mathcal{O}}_B(pq \max\{p, q\} d\tau)$.*

Proposition 5. *We compute $\mathbf{SR}(f, g\,; \mathsf{a})$, where $\mathsf{a} \in \mathbb{Q} \cup \{\infty\}$ and $\mathcal{L}(\mathsf{a}) = \sigma$, in $\widetilde{\mathcal{O}}_B(pq \max\{p, q\} d \max\{\tau, \sigma\})$. For the polynomials $\mathbf{SR}_j(f, g\,; \mathsf{a}) \in \mathbb{Z}[y]$, except for f, g, we have $\deg_y(\mathbf{SR}_j(f, g\,; \mathsf{a})) = \mathcal{O}((p + q)d)$ and $\mathcal{L}(\mathbf{SR}_j(f, g\,; \mathsf{a})) = \mathcal{O}(\max\{p, q\}\tau + \min\{p, q\}\sigma)$.*

3. REDUCTION TO INTEGER COEFFICIENTS

3.1 Some useful bounds

The roots of B_α in Problem 1 are algebraic numbers, hence they are roots of a polynomial with integer coefficients. We estimate bounds on the degree and the bitsize of this polynomial, and we will use them to analyze the Boolean complexity of the real root isolation algorithm.

Consider a real algebraic number $\alpha \in \mathbb{R}_{\mathsf{alg}}$, in isolating interval representation $\alpha \cong (A, \mathcal{I})$, where $A = \sum_{i=0}^m a_i x^i$, $\mathcal{I} = [\mathsf{a}_1, \mathsf{a}_2]$, $\mathsf{a}_{1,2} \in \mathbb{Q}$ and $\deg(A) = m$ and $\mathcal{L}(A) = \tau$. Since A is square-free, has m, possible complex, roots, say $\alpha_1, \alpha_2, \dots, \alpha_m$ and after a (possible) reordering let $\alpha = \alpha_1$.

Let $B_\alpha \in \mathbb{Z}(\alpha)[y]$, be a univariate polynomial in y, with coefficients that are polynomials in α with integer coefficients. More formally, let $B_\alpha = \sum_{i=0}^n b_i(\alpha) y^i$, where $b_i(x) = \sum_{j=0}^{\eta_i} c_{ij} x^j$ and $\eta_i < m$, $0 \le i \le d$. The restriction $\eta_i < m$ comes from the fact that $\mathbb{Z}(\alpha)$ is a vector space of dimension[1] m and the elements of one of its bases are $1, \alpha, \dots, \alpha^{m-1}$. Finally, let $\mathcal{L}(B_\alpha) = \max_{i,j} \mathcal{L}(c_{ij}) = \sigma$. We assume that B_α is a square-free.

Our goal is to isolate the real roots of B_α (Problem 1). Since B_α has algebraic numbers as coefficients, its roots are algebraic numbers as well. Hence, there is a polynomial with integer coefficients that has as roots the roots of B_α, and possible other roots as well. To construct this polynomial, e.g. [8, 17, 19], we consider the following resultant w.r.t. x

$$R(y) = \mathrm{res}_x(B(x, y), A(x)) = (-1)^{mn} a_m^\eta \prod_{j=1}^m B(\alpha_j, y), \quad (4)$$

where $\eta = \max\{\eta_i\}$, and $B(x, y) \in \mathbb{Z}[x]$ is obtained from B_α after replacing all the occurrences of α with x. Interpreting the resultant using the Poisson formula, $R(y)$ is the product of polynomials $B(\alpha_j, y)$, where j ranges over all the roots of A. Our polynomial $B_\alpha \in \mathbb{Z}(\alpha)[y]$ is the factor in this product for $j = 1$. Hence, R has all the roots that B_α has and maybe more.

---

[1]If A is the minimal polynomial of α then the dimension is exactly m. In general it is not (computational) easy to compute the the minimal polynomial of a real algebraic number, thus we work with a square-free polynomial that has it as real root.

Remark 6. *Notice that $R(y)$ is not square-free in general. For example consider the polynomial $B_\alpha = y^4 - \alpha^2$, where α is the positive root of $A = x^2 - 3$. In this case $R(y) = \mathrm{res}_x(A(x), B(x, y) = \mathrm{res}_x(x^2 - 3, y^2 - x^2) = (y^4 - 3)^2$.*

Using Prop. 14 and by taking into account that $\eta_i < m$, we get $\deg(R) \le mn$ and $\mathcal{L}(R) \le m(\tau + \sigma) + 2m \lg(4mn)$. We may also write $\deg(R) = \mathcal{O}(mn)$ and $\mathcal{L}(R) = \widetilde{\mathcal{O}}(m(\sigma + \tau))$.

In order to construct an isolating interval representation for the real roots of B_α, we need a square-free polynomial. This polynomial, $C(y) \in \mathbb{Z}[y]$, is a square factor of $R(y)$, and so it holds $\deg(C) \le mn$ and $\mathcal{L}(C) \le m(\tau + \sigma) + 3m \lg(4mn)$, where the last inequality follows from Mignotte's bound [23].

Using the Prop. 1, we deduce the following lemma:

Lemma 7. *Let B_α be as in Problem 1. The minimal polynomial, $C \in \mathbb{Z}[x]$, of the, possible complex, roots of B_α, γ_i, has degree $\le mn$ and bitsize $\le m(\tau + \sigma) + 3m \lg(4mn)$ or $\widetilde{\mathcal{O}}(m(\tau + \sigma))$. Moreover, it holds*

$$|\gamma_i| \le 2^{m(\tau+\sigma)+2m\lg(4mn)} , \quad (5)$$
$$-\lg \Delta(C) \le m^2 n(\tau + \sigma + 4\lg(4mn)) , \quad (6)$$
$$-\sum_i \lg \Delta_i(C) \le 3m^2 n(n + \tau + \sigma + 6\lg(4mn)) , \quad (7)$$

$$|\gamma_i| \le 2^{\widetilde{\mathcal{O}}(m(\tau+\sigma))} , \quad (8)$$
$$-\lg \Delta(C) = \widetilde{\mathcal{O}}(m^2 n(\tau + \sigma)) , \quad (9)$$
$$\Sigma(C) = -\sum_i \lg \Delta_i(C) = \widetilde{\mathcal{O}}(m^2 n(n + \tau + \sigma)) . \quad (10)$$

3.2 The algorithm

The indirect algorithm for Problem 2, follows closely the procedure described in the previous section to estimate the various bounds on the roots of B_α. First, we compute the univariate polynomial with integer coefficients, R, such that the set of its real roots includes those of B_α. We isolate the real roots of R and we identify which ones are roots of B_α.

Let us present in details the three steps and their complexity. We compute R using resultant computation, as presented in (4). For this we consider B as a bivariate polynomial in $\mathbb{Z}[x, y]$ and we compute $\mathrm{res}_x(B(x, y), A(x))$, using Prop. 4. Since $\deg_x(B) < m$, $\deg_y(B) = n$, $\mathcal{L}(B) = \sigma$, $\deg_x(A) = m$, $\deg_y(A) = 0$ and $\mathcal{L}(A) = \tau$, this computation costs $\widetilde{\mathcal{O}}_B(m^3 n(\sigma + \tau))$, using Prop. 4.

Now we isolate the real roots of R. This can be done in $\widetilde{\mathcal{O}}_B(m^4 n^3(mn^2 + mn\tau + n^2\sigma + m\tau^2 + m\sigma^2 + m\tau\sigma))$, by Prop. 2. In the same complexity bound we can also compute the multiplicities of the real roots, if needed [14].

The rational numbers that isolate the real roots of R have bitsize bounded by $\widetilde{\mathcal{O}}(m^2 n(n + \sigma + \tau))$, which is also a bound on the bitsize of all of them, as Prop. 1 and Lem. 7 indicate.

It is possible that R can have more roots that B_α, thus it remains to identify which real roots of R are roots of B_α. For sure all the real roots of B_α are roots of R. Consider a real root γ of R and its isolating interval $[\mathsf{c}_1, \mathsf{c}_2]$. If γ is a root of B_α, then since B_α is square-free, by Rolle's theorem it must change signs if we evaluate it over the endpoints of the isolating interval of γ. Hence, in order to identify the real roots of R that are roots of B_α it suffices to compute the sign of B_α over all the endpoints of the isolating intervals.

We can improve the step that avoids the non-relevant roots of R by applying the algorithm for chainging the ordering of

a bivariate regular chain [26]. However, currently, this step is not the bottleneck of the algorithm so we do not elaborate further.

Consider an isolating point of R, say $c_j \in \mathbb{Q}$, of bitsize s_j. To compute the sign of the evaluation of B_α over it, we proceed as follows. First we perform the substitution $y = c_j$, and after clearing denominators, we get a number in $\mathbb{Z}[\alpha]$, for which we want to compute its sign. This is equivalent to consider the univariate polynomial $B(x, c_j)$ and to compute its sign if we evaluate it over the real algebraic number α. We have $\deg(B(x, c_j)) = \mathcal{O}(m)$ and $\mathcal{L}(B(x, c_j)) = \widetilde{\mathcal{O}}(\sigma + ns_j)$. Hence the sign evaluation costs $\widetilde{\mathcal{O}}_B(m^3\tau + m^2\sigma + m^2ns_j)$ using Prop. 3. Summing up over all s_j's, there are $\mathcal{O}(mn)$, and taking into account that $\sum_j s_j = \widetilde{\mathcal{O}}(m^2n(\sigma + \tau + n))$ (Lem. 7), we conclude that the overall complexity of identifying the real roots of B_α is $\widetilde{\mathcal{O}}_B(m^4n^3 + m^4n\tau + m^3n\sigma + m^4n^2(\sigma + \tau))$.

The overall complexity of the algorithm is dominated by that of real solving. We can state the following theorem:

Theorem 8. *The complexity of isolating the real roots of $B \in \mathbb{Z}(\alpha)[y]$ using the indirect method is $\widetilde{\mathcal{O}}_B(m^4n^3(mn^2 + mn\tau + n^2\sigma + m\tau^2 + m\sigma^2 + m\tau\sigma))$. If $N = \max\{m, n, \sigma, \tau\}$, then the previous bounds become $\widetilde{\mathcal{O}}_B(N^{10})$.*

If the polynomial B_α is not square-free then we can apply the algorithm of [37] to compute its square-free factorization and then we apply the previous algorithm either to the square-free part or to each polynomial of the square-free factorization. The complexity of the square-free factorization is $\widetilde{\mathcal{O}}_B(m^2n(\sigma^2 + \tau^2) + mn^2(\sigma + \tau))$, and does not dominate the aforementioned bound.

4. TWO DIRECT APPROACHES

The computation of R, the polynomial with integer coefficients that has the real roots of B_α is a costly operation that we usually want to avoid. If possible, we would like to try to solve the polynomial B_α directly, using one of the well-known subdivision algorithms, for example STRUM or DESCARTES and BERNSTEIN, specially adopted to handle polynomials that have coefficients in an extension field. In practice, this is accomplished by obtaining, repeatedly improved, approximations of the real algebraic number α and subsequently apply DESCARTES or BERNSTEIN for polynomials with interval coefficients, e.g. [16, 29].

The fact that we compute the roots using directly the representation of B_α allows us to avoid the complexity induced by the conjugates of α. This leads to improved separation bounds, and to faster algorithms for real root isolation.

4.1 Separation bounds for B_α

We compute various bounds on the roots of B_α based on the first inequalities of Prop. 1. For this we need to compute a lower bound for $|\text{disc}(B_\alpha)|$ and an upper bound for $\|B_\alpha\|_2$.

First we compute bounds on the coefficients on B_α. Let $\alpha_1 = \alpha, \alpha_2, \ldots, \alpha_m$ be the roots of A. We consider the resultants

$$r_i := \text{res}_x(A(x), z - b_i(x)) = \text{res}_x\left(A(x), z - \sum_{j=0}^{\eta_i} c_{i,j}x^j\right) \in \mathbb{Z}[z] \ .$$

It holds that

$$r_i(z) = a_m^\eta \prod_{k=1}^m (z - b_i(\alpha_k)) \ ,$$

where $\eta = \max\{\eta_i\} < m$. The roots of r_i are the numbers $b_i(\alpha_k)$, where k runs over all the roots of A. We use Prop. 14 to bound the degree and bitsize of r_i. The degree of r_i is bounded by m and their coefficient are of bitsize $\leq m\sigma + m\tau + 5m\lg(m)$. Using Cauchy's bound, we deduce

$$2^{-m\sigma - m\tau - 5m\lg(m)} \leq |b_i(\alpha_k)| \leq 2^{m\sigma + m\tau + 5m\lg(m)} \ , \quad (11)$$

for all i and k. To bound $|\text{disc}(B_\alpha)|$ we consider the identity

$$\begin{aligned}
\text{disc}(B_\alpha) &= (-1)^{\frac{1}{2}n(n-1)} \frac{1}{b_n(\alpha)} \text{res}_y(B_\alpha, \partial B_\alpha(y)/\partial y) \\
&= (-1)^{\frac{1}{2}n(n-1)} \frac{1}{b_n(\alpha)} R_B(\alpha) \ ,
\end{aligned}$$

where the resultant, $R_B \in \mathbb{Z}[\alpha]$, can be computed as the determinant of the Sylvester matrix of B_α and $\partial B_\alpha(y)/\partial y$, evaluated over α.

The Sylvester matrix is of size $(2n-1) \times (2n-1)$, the elements of which belong to $\mathbb{Z}[\alpha]$. The determinant consists of $(2n-1)!$ terms. Each term is a product of $n-1$ polynomials in α of degree at most $m-1$ and bitsize at most σ, times a product of n polynomials in α of degree at most $m-1$ and bitsize at most $\sigma + \lg n$. The first product results a polynomial of degree $(n-1)(m-1)$ and bitsize $(n-1)\sigma + (n-1)\lg m$. The second product results polynomials of degree $n(m-1)$ and bitsize $n\sigma \lg n + n\lg m$. Thus, any term in the determinant expansion is a polynomial in α of degree at most $(2n-1)(m-1)$, or $\mathcal{O}(mn)$, and bitsize at most $4(2n-1)\sigma \lg(mn)$ or $\widetilde{\mathcal{O}}(n\sigma)$. The determinant itself, is a polynomial in α of degree at most mn and of bitsize $4(2n-1)\sigma \lg(mn) + (2n-1)\lg(2n-1) \leq 5(2n-1)\sigma \lg(mn) = \widetilde{\mathcal{O}}(n\sigma)$.

To compute a bound on $R_B(\alpha)$ we consider R_B as a polynomial in $\mathbb{Z}[y]$, and we compute a bound on its evaluation over α. For this we use resultants. It holds

$$D = \text{res}_x(A(x), y - R_B(x)) = a_m^{\deg(R_B)} \prod_{i=1}^m (y - R_B(\alpha_i)) \ .$$

We notice that the roots of $D \in \mathbb{Z}[x]$ are the evaluations of R_B over the roots of A. So it suffices to compute bounds on the roots of D. Using Prop. 14 we deduce that $\deg(D) \leq m$ and $\mathcal{L}(D) \leq 13mn\sigma \lg(mn) + mn\tau$ or $\mathcal{L}(D) = \widetilde{\mathcal{O}}(mn(\sigma + \tau))$. Using Cauchy bound, refer to Eq. (1), we conclude that

$$2^{-13mn\sigma \lg(mn) - mn\tau} \leq |R_B(\alpha)| \leq 2^{13mn\sigma \lg(mn) + mn\tau} \ .$$

Using this inequality and (11), we can bound $|\text{disc}(B_\alpha)|$, i.e.

$$2^{-13mn\sigma \lg(mn) - 2mn\tau} \leq |\text{disc}(B_\alpha)| \leq 2^{13mn\sigma \lg(mn) + 2mn\tau} \ . \tag{12}$$

It remains to bound $\|B_\alpha\|_2$. Using Eq. (11) we get

$$\|B_\alpha\|_2^2 \leq \sum_{i=0}^n (b_i(\alpha))^2 \leq (n+1) \, 2^{2m(\sigma + \tau + 5\lg(m))} \ .$$

The previous discussion leads to the following lemma

Lemma 9. *Let B_α be as in Problem 1, and ξ_i be its roots. Then, it holds*

$$|\xi_i| \leq 2^{m(\tau+\sigma+5\lg m)} , \qquad (13)$$
$$-\lg\Delta(B_\alpha) \leq 12mn(\sigma\lg(mn) + \tau + 5\lg m) ,(14)$$
$$-\sum_i \lg\Delta_i(B_\alpha) \leq 14mn(\sigma\lg(mn)+\tau+5\lg m) ,(15)$$

or

$$|\xi_i| \leq 2^{\widetilde{\mathcal{O}}(m(\tau+\sigma))} , \qquad (16)$$
$$-\lg\Delta(B_\alpha) = \widetilde{\mathcal{O}}(mn(\tau+\sigma)) , \quad (17)$$
$$\Sigma(B_\alpha) = -\sum_i \lg\Delta_i(B_\alpha) = \widetilde{\mathcal{O}}(mn(\tau+\sigma)) . \quad (18)$$

4.2 The STURM algorithm

Let us first study the STURM algorithm. We assume B_α as in Problem 1 to be square-free. To isolate the real roots of B_α using the STURM algorithm, we need to evaluate the Sturm sequence of $B(\alpha, y)$ and its derivative with respect to y, $\partial B(\alpha, y)/\partial y$, over various rational numbers. For the various bounds needed we will use Lem. 9.

The number of steps that a subdivision-based algorithm, and hence STURM algorithm, performs to isolate the real roots of a polynomial depends on the separation bound. To be more specific, the number of steps, $(\#T)$, that STURM performs is $(\#T) \leq 2r + r\lg\mathsf{B} + \Sigma(B_\alpha)$ [8, 10], where r is the number of real roots and B is an upper bound on the real roots. Using (14) and (15) we deduce that $(\#T) = \widetilde{\mathcal{O}}(mn(\tau+\sigma))$.

To complete the analysis of the algorithm it remains to compute the complexity of each step, i.e. the cost of evaluating the Sturm sequence over a rational number, of the worst possible bitsize. The latter is induced by the separation bound, and in our case is $\widetilde{\mathcal{O}}(mn(\tau+\sigma))$.

We consider B as polynomial in $\mathbb{Z}[x, y]$ and we evaluate the Sturm-Habicht sequence of B and $\frac{\partial B}{\partial y}$, over rational numbers of bitsize $\widetilde{\mathcal{O}}(mn(\tau+\sigma))$. The cost of this operation is $\widetilde{\mathcal{O}}_B(m^2n^4(\tau+\sigma))$ (Prop. 5).

It produces $\mathcal{O}(n)$ polynomials in $\mathbb{Z}[x]$, of degrees $\mathcal{O}(mn)$ and bitsize $\widetilde{\mathcal{O}}(n\tau+n\sigma)$. For each polynomial we have to compute its sign if we evaluate it over α. Using Prop. 3 each sign evaluation costs $\widetilde{\mathcal{O}}_B(m(m^2+n^2)\tau + mn^2\sigma)$, and so the overall cost is $\widetilde{\mathcal{O}}_B(mn(m^2+n^2)\tau + mn^3\sigma)$. If we multiply the latter bound with the number of steps, $\widetilde{\mathcal{O}}(mn(\tau+\sigma))$, we get the following theorem.

Theorem 10. *The complexity of isolating the real roots of $B \in \mathbb{Z}(\alpha)[y]$ using the STURM algorithm is $\widetilde{\mathcal{O}}_B(m^2n^2(m^2+n^2)(\tau^2+\sigma^2))$, or $\widetilde{\mathcal{O}}_B(N^8)$, where $N = \max\{m, n, \sigma, \tau\}$.*

4.3 A modified DESCARTES algorithm

We consider Sagraloff's modified version of Descartes' algorithm [32], that applies to polynomials with bitstream coefficients. We also refer the reader to [12, 21].

As stated in Problem 1, let α be a real root of $A = \sum_{i=0}^m a_i x^i \in \mathbb{Z}[x]$, where $a_m \neq 0$ and $|a_i| < 2^\tau$ for $0 \leq i \leq m$, and let $B_\alpha = \sum_{i=0}^n b_i(\alpha)y^i \in \mathbb{Z}[\alpha][y]$, where $b_i = \sum_{j=0}^{\eta_i} c_{i,j}x^j \in \mathbb{Z}[x]$, $\eta_i < m$ and $|c_{i,j}| < 2^\sigma$ for $0 \leq i \leq n$ and $0 \leq j \leq \eta_i$, where we also assume that B_α is square-free.

Let $\xi_1, \ldots, \xi_n$ be all (complex) roots of B, and $\Delta_i(B_\alpha) := \min_{j\neq i}|\xi_j - \xi_i|$. By Theorem 19 of [32], the complexity of

isolating real roots of B_α is

$$\widetilde{\mathcal{O}}_B(n(\Sigma(B_\alpha) + n\tau_B)^2) ,$$

where $\left|\frac{b_i(\alpha)}{b_n(\alpha)}\right| \leq 2^{\tau_B}$ and $\Sigma(B_\alpha) = -\sum_{i=1}^n \lg(\Delta_i(B_\alpha))$. From Lem. 9 we get that

$$\Sigma(B_\alpha) \leq 14mn(\tau + \sigma\lg(mn)) + n\lg n = \widetilde{\mathcal{O}}(mn(\tau+\sigma)) . \qquad (19)$$

To compute a bound on τ_B, we use Eq. (11). It holds $\left|\frac{b_i(\alpha_k)}{b_n(\alpha_k)}\right| \leq 2^{2m\sigma+2m\tau+6m\lg(m)}$, for all i and k. Hence,

$$\tau_B \leq 2m\sigma + 2m\tau + 6m\lg(m) = \widetilde{\mathcal{O}}(m(\sigma+\tau)) . \qquad (20)$$

Finally, by combining (19) and (20), we deduce that the cost of isolating real roots of B is

$$\widetilde{\mathcal{O}}_B(n(\Sigma(B_\alpha) + n\tau_B)^2) = \widetilde{\mathcal{O}}_B(n(mn\tau + mn\sigma)^2)$$
$$= \widetilde{\mathcal{O}}_B(m^2n^3(\sigma^2+\tau^2)) .$$

If $N = \max\{m, n, \sigma, \tau\}$, then the bound becomes $\widetilde{\mathcal{O}}_B(N^7)$.

It remains to estimate the cost of computing the successive approximations of $b_i(\alpha)/b_n(\alpha)$. The root isolation algorithm requires approximations of $b_i(\alpha)/b_n(\alpha)$ to accuracy of $\mathcal{O}(\Sigma(B_\alpha) + n\tau_B)$ bits after the binary point. Since $|b_i(\alpha)/b_n(\alpha)| \leq 2^{\tau_B}$, to approximate each fraction, for $0 \leq i \leq n-1$, to accuracy L, it is sufficient to approximate $b_i(\alpha)$, for $0 \leq i \leq n$, up to precision $\mathcal{O}(L + \tau_B)$. Hence, the algorithm requires approximation of $b_i(\alpha)$, for $0 \leq i \leq n$, to precision $\mathcal{O}(\Sigma(B)+n\tau_B)$. By inequality (11), $|b_i(\alpha)| \geq 2^{-\tau_B}$, and therefore it is sufficient to approximate $b_i(\alpha)$ to accuracy $\mathcal{O}(\Sigma(B_\alpha) + n\tau_B)$.

Approximation of $c_{i,j}\alpha^j$ to accuracy of L bits requires approximation of α to accuracy $L + \lg|c_{i,j}| + \lg(j) + (j-1)\lg|\alpha| \leq L + \sigma + \lg(m) + (m-1)(\tau+1) = \widetilde{\mathcal{O}}(L+\sigma+m\tau)$ bits. Hence the accuracy of approximations of α required by the algorithm is

$$\mathcal{O}(\Sigma(B_\alpha) + n\tau_B) = \widetilde{\mathcal{O}}(mn(\sigma+\tau)) .$$

By Lemmata 4.4, 4.5 and 4.11 of [18], the bit complexity of approximating α to accuracy L is

$$\widetilde{\mathcal{O}}(m^4\tau^2 + m^2L) .$$

Therefore, the bit complexity of computing the required approximations of $b_i(\alpha)/b_n(\alpha)$ is

$$\widetilde{\mathcal{O}}(m^4\tau^2 + m^2mn(\sigma+\tau)) = \widetilde{\mathcal{O}}(m^3(m\tau^2 + n\sigma + n\tau)) .$$

Theorem 11. *The bit complexity of isolating the real roots of B_α of Problem 1 using the modified Descartes' algorithm in [32] is $\widetilde{\mathcal{O}}_B(m^2n^3(\sigma^2+\tau^2) + m^3(m\tau^2+n\sigma+n\tau))$, or $\widetilde{\mathcal{O}}_B(N^7)$, where $N = \max\{m, n, \sigma, \tau\}$.*

4.4 Almost tight separation bounds

Let α be the root of $A(x) = x^m - ax^{m-1} - 1$, in $(a, a+1)$, for $a \geq 3$, $m \geq 3$. Then the Mignotte polynomial $B_\alpha(y) = y^n - 2(\alpha^k y - 1)^2$, where $k = \lfloor(m-1)/2\rfloor$, has two roots in $(1/\alpha^k - h, 1/\alpha^k + h)$, where $h = \alpha^{-k(n+2)/2} < a^{-(m-2)(n+2)/4}$.

If $a \leq 2^\tau$ and $\tau = \Omega(\lg(mn))$, then $-\lg\Delta(B_\alpha) = \Omega(mn\tau)$, which matches the upper bound in (15) of Lem. 9. This quantity, $\Omega(mn\tau)$, is also a tight lower bound for the number of steps that an subdivision based algorithm performs, following the arguments used in [13] to prove a similar bound for polynomials with integer coefficients.

n	Algorithm	$m=2$	$m=3$	$m=5$	$m=10$	$m=20$
10	ICF	0.003	0.006	0.013	0.082	0.820
	BMD	0.002	0.002	0.003	0.006	0.019
20	ICF	0.004	0.010	0.048	1.49	2.80
	BMD	0.008	0.008	0.010	0.017	0.053
50	ICF	0.014	0.044	0.271	8.29	20.5
	BMD	0.046	0.050	0.061	0.079	0.213
100	ICF	0.047	0.173	1.09	33.1	108
	BMD	0.165	0.206	0.137	0.246	0.546
200	ICF	0.144	0.612	4.90	141	626
	BMD	0.746	0.701	1.00	0.824	1.55

Table 1. Randomly generated polynomials

n	Algorithm	$m=2$	$m=3$	$m=5$	$m=10$	$m=20$
10	ICF	0.011	0.008	0.032	0.208	1.75
	BMD	0.007	0.007	0.009	0.010	0.015
20	ICF	0.019	0.041	0.193	1.50	13.9
	BMD	0.075	0.071	0.080	0.088	0.106
50	ICF	0.122	0.270	1.51	25.8	338
	BMD	1.78	1.63	1.83	1.90	2.27
100	ICF	0.834	2.17	16.1	365	10649
	BMD	54.7	51.3	56.0	74.7	92.4
200	ICF	7.53	31.2	246	8186	> 36000
	BMD	2182	3218	3830	4280	4377

Table 2. Generalized Laguerre polynomials

n	Algorithm	$m=2$	$m=3$	$m=5$	$m=10$	$m=20$
10	ICF	0.017	0.012	0.035	0.285	2.09
	BMD	0.015	0.013	0.011	0.015	0.008
20	ICF	0.029	0.069	0.262	2.23	18.3
	BMD	0.059	0.052	0.069	0.039	0.027
50	ICF	0.137	0.356	2.04	45.4	429
	BMD	1.84	1.35	1.29	0.703	0.561
100	ICF	0.808	2.84	24.6	674	8039
	BMD	47.0	38.6	32.0	23.3	8.38
200	ICF	8.48	35.1	348	11383	> 36000
	BMD	3605	2566	2176	927	565

Table 3. Generalized Wilkinson polynomials

n	Algorithm	$m=3$	$m=5$	$m=10$	$m=20$
10	ICF	0.003	0.008	0.049	0.594
	BMD	0.010	0.006	0.014	0.036
20	ICF	0.006	0.027	0.288	8.83
	BMD	0.015	0.020	0.049	0.137
50	ICF	0.041	0.441	12.2	777
	BMD	0.112	0.147	0.321	0.854
100	ICF	0.866	11.6	729	28255
	BMD	0.702	0.868	2.32	5.99
200	ICF	35.7	684	23503	> 36000
	BMD	3.12	5.30	13.8	46.1

Table 4. Mignotte polynomials

5. IMPLEMENTATION AND EXPERIMENTS

We compare implementations of two methods of real root isolation for square-free polynomials over simple algebraic extensions of rationals. The first method, *ICF* (for Integer Continued Fractions), performs reduction to integer coefficients described in Section 3.2. For isolating roots of polynomials with integer coefficients it uses the MATHEMATICA implementation of the Continued Fractions algorithm [1]. The second method, *BMD* (for Bitstream Modified Descartes), uses Sagraloff's modified version of Descartes' algorithm ([32], see Section 4.3). The algorithm has been implemented in C as a part of the MATHEMATICA system.

For the experiments we used a 64-bit Linux virtual machine with a 3 GHz Intel Core i7 processor and 6 GB of RAM. The timings are in given seconds. Computations that did not finish in 10 hours of CPU time are reported as > 36000.

Randomly generated polynomials. For given values of m and n each instance was generated as follows. First, univariate polynomials of degree m with uniformly distributed random 10-bit integer coefficients were generated until an irreducible polynomial which had real roots was obtained. A real root r of the polynomial was randomly selected as the extension generator. Finally, a polynomial in $\mathbb{Z}[r,y]$ of degree n in y and degree $m-1$ in r with 10-bit random integer coefficients was generated. The results of the experiments are given in Table 1. Each timing is an average for 10 randomly generated problems.

Generalized Laguerre Polynomials. This example compares the two root isolation methods for generalized Laguerre polynomials $L_n^\alpha(x)$, where α was chosen to be the smallest root of the Laguerre polynomial $L_m(x)$. Note that $L_n^\alpha(x)$ has n positive roots for any positive α and $L_m(x)$ has m positive roots, so this example maximizes the number of real roots of both the input polynomial with algebraic number coefficients and the polynomial with integer coefficients obtained by *ICF*. The results of the experiment are given in Table 2.

Generalized Wilkinson Polynomials. This example uses the following generalized Wilkinson polynomials $W_{n,\alpha}(x) := \prod_{k=1}^{n}(x - k\alpha)$, where α is the smallest root of the Laguerre polynomial $L_m(x)$. The timings are presented in Table 3.

Mignotte Polynomials. The variant of Mignotte polynomials used in this example is given by $M_{n,\alpha}(x) := y^n - 2(\alpha^k y - 1)^2$, where α is the root of $A_m(x) := x^m - 3x^{m-1} - 1$ in $(3,4)$, $m \geq 3$ and $k = \lfloor (m-1)/2 \rfloor$ (see Section 4.4). The results of the experiment are given in Table 4.

The experiments suggest that for low degree extensions *ICF* is faster than *BMD*, but in all experiments as the degree of extension grows *BMD* becomes faster than *ICF*. Another fact worth noting is that *ICF* depends directly on the extension degree m, since it isolates roots of a polynomial of degree mn. On the other hand, the only part of *BMD* that depends directly on m is computing approximations of coefficients, which in practice seems to take a very small proportion of the running time. The main root isolation loop depends only on the geometry of roots, which depends on m only through the worst case lower bound on root separation. Indeed, in all examples the running time of *ICF* grows substantially with m, but the running time of *BMD* either grows at a much slower pace or, in case of generalized Wilkinson polynomials, it even decreases with m (because the smallest root α of $L_m(x)$, and hence the root separation of $W_{n,\alpha}(x)$, increase with m). The superiority of the direct approach was also observed in [17].

6. MULTIPLE EXTENSIONS

In this section we consider the problem of real root isolation of a polynomials with coefficients in multiple extensions. We tackle the problem using a reduction to a polynomial with integer coefficients. The technique could be considered as a generalization of the one presented in Sec. 3.

We use $\mathbf{x^e}$ to denote the monomial $x_1^{e_1} \cdots x_n^{e_\ell}$, with $\mathbf{e} = (e_1, \ldots, e_\ell) \in \mathbb{N}^\ell$. For a polynomial $f = \sum_{j=1}^{m} c_j \mathbf{x^{e_j}} \in \mathbb{Z}[\mathbf{x}]$, let $\{\mathbf{e}_1, \ldots, \mathbf{e}_m\} \subset \mathbb{N}^\ell$ be the support of f; its Newton polytope Q is the convex hull of the support. By $(\#Q)$ we

denote the integer points of the polytope Q, i.e. $(\#Q) = |Q \cap \mathbb{Z}^\ell|$.

Problem 2. *Let α_j, where $1 \leq j \leq \ell$, be a real algebraic numbers. Their isolating interval representation is $\alpha_j \cong (A_j, \mathcal{I}_j)$, where $A_j = \sum_{i=0}^m a_i x_j^i$, $\mathcal{I}_j = [a_{j,1}, a_{j,2}]$, $a_{1,2} \in \mathbb{Q}$, $\deg(A_j) = m$, and $\mathcal{L}(A_j) = \tau$. Let*

$$B_\alpha = \sum_{i=0}^n b_i(\alpha_1, \ldots, \alpha_\ell) y^i \in \mathbb{Z}(\alpha)[y],$$

be square-free, where $b_i(\mathbf{x}) = \sum_{\mathbf{e}} c_{ij} \mathbf{x}^{\mathbf{e}} \in \mathbb{Z}[\mathbf{x}]$, $\mathcal{L}(c_{i,j}) \leq \sigma$, for $0 \leq i \leq d$, and for $\mathbf{e} = (e_1, \ldots, e_\ell)$, it holds $e_j \leq \eta < m$, What is the Boolean complexity of isolating the real roots of B_α?

We denote by $\mathbf{a}_i$ the coefficients of A_i, where $1 \leq i \leq \ell$, and by $\mathbf{c}$ the coefficients of B. We compute separation bounds following the technique introduced [15].

We consider the zero dimensional polynomial system (S): $A_1(\mathbf{x}) = \cdots = A_\ell(\mathbf{x}) = A_{\ell+1}(\mathbf{x}) = 0$, where $A_k(\mathbf{x}) = \sum_{i=0}^m a_{k,i} x_k^i = 0$, $1 \leq k \leq \ell$, and $A_{\ell+1} = B(\mathbf{x}, y) = \sum_{i=0}^n b_i(x_1, \ldots, x_\ell) y^i = 0$. We should mention that we make the assumption that B does not become identically zero when $\alpha_1, \ldots, \alpha_l$ are replaced with some set of their conjugates (otherwise the resultant is zero).

We hide variable y, that is we consider (S) as an overdetermined system of $\ell + 1$ equations in ℓ variables. We consider the resultant, R, with respect to $x_1, \ldots, x_\ell$, that is we eliminate these variables, and we obtain a polynomial $R \in \mathbb{Z}[\mathbf{a}_1, \ldots, \mathbf{a}_\ell, \mathbf{c}, y]$. We interpret the resultant using the Poisson formula [7], see also [27], i.e.

$$R(y) = \mathbf{res}_x(A_1, \ldots, A_\ell, B) = \prod B(\alpha_{1,i_1}, \ldots, \alpha_{\ell,i_\ell}, y),$$

and $R(y) \in (\mathbb{Z}[\mathbf{a}_1, \ldots, \mathbf{a}_\ell, \mathbf{c}])[y]$. Similar to the single extension case, B_α, is among the factors of R, hence it suffices to compute bounds for the roots of $R(y)$.

We consider R as a univariate polynomial in y. The resultant is a homogeneous polynomial in the coefficients of (S), we refer to e.g. [7, 27] for more details and to [15] for a similar application. To be more specific, the structure of the coefficients of R is

$$R(y) = \cdots + \varrho_k \mathbf{a}_1^{M_1} \cdots \mathbf{a}_\ell^{M_\ell} \mathbf{c}^{M_{\ell+1}-k} (y^i)^k + \cdots,$$

where $1 \leq k \leq M_{\ell+1} = m^\ell$, and i is a number in $\{1, \ldots, n\}$. The semantics of $\mathbf{a}_i^{M_i}$ are that it is a monomial in the coefficients of A_i of total degree M_i. Similarly, $\mathbf{c}^{M_{\ell+1}-k}$ stands for a monomial in the coefficients of B of total degree $M_{\ell+1} - k$. Moreover, $M_i \leq \ell \eta m^{\ell-1} < \ell(m-1)m^{\ell-1} < \ell m^\ell$. The degree of R with respect to y is at most $n M_{\ell+1} = nm^\ell$.

Since $|a_{i,j}| \leq 2^\tau$, it holds

$$\lg \prod_{i=i}^\ell |\mathbf{a}_i|^{M_i} \leq \tau \ell^2 m^\ell. \qquad (21)$$

Similarly, since $|c_{i,j}| \leq 2^\sigma$, we get

$$\lg |\mathbf{c}|^{M_{\ell+1}-k} \leq \sigma(m^\ell - k) \leq \sigma m^\ell. \qquad (22)$$

Finally, $|\varrho_k| \leq \prod_{i=1}^{\ell+1} (\#Q_i)^{M_i}$ [34], where $(\#Q_i)$ is the number of integer points of the Newton polytope of the polynomial A_i. We let $A_{\ell+1} = B$. It is $(\#Q_i) = m + 1$ for

$1 \leq i \leq \ell$, so $\prod_{i=1}^\ell (\#Q_i)^{M_i} \leq (m+1)^{\ell(m-1)m^{\ell-1}} \leq m^{\ell m^\ell}$, and $(\#Q_{\ell+1}) \leq (\ell(m-1) + n)^{\ell+1} + \ell + 1$. Hence,

$$(\#Q_i)^{M_{\ell+1}} \leq \left((\ell(m-1) + n)^{\ell+1} + \ell + 1 \right)^{m^\ell}$$
$$\leq (2\ell m + n)^{(\ell+1)m^\ell} \leq (\ell mn)^{\ell m^\ell},$$

and so for every k

$$\lg |\varrho_k| \leq \lg \prod_{i=1}^{\ell+1} (\#Q_i)^{M_i} \leq 2\ell m^\ell \lg(mn\ell). \qquad (23)$$

By combining (21), (22) and (23) we can bound the coefficients of R and its square-free factors. Using also Prop. 1 we get the following lemma.

Lemma 12. *Let B_α be as in Problem 2. The minimal polynomial, C_ℓ of the, possible complex, roots of B_α, γ_i, has degree $\leq n m^\ell$ and bitsize $\leq m^\ell(\tau \ell^2 + \sigma + 3\ell \lg(mn\ell))$ or $\widetilde{\mathcal{O}}(m^\ell(\ell^2 \tau + \sigma))$. Moreover, it holds*

$$|\gamma_i| \leq 2^{m^\ell(\ell^2 \tau + \sigma + 2\ell \lg(mn\ell))}, \qquad (24)$$
$$-\lg \Delta(C_\ell) \leq m^{2\ell} n(\ell^2 \tau + \sigma + 4\ell \lg(mn\ell)), \qquad (25)$$
$$-\sum_i \lg \Delta_i(C_\ell) \leq m^{2\ell} n(\ell^2 \tau + \sigma + n + 6\ell \lg(mn\ell)) \qquad (26)$$

$$|\gamma_i| \leq 2^{\widetilde{\mathcal{O}}(m^\ell(\ell^2 \tau + \sigma))}, \qquad (27)$$
$$-\lg \Delta(C_\ell) = \widetilde{\mathcal{O}}(m^{2\ell} n(\ell^2 \tau + \sigma)), \qquad (28)$$
$$-\sum_i \lg \Delta_i(C_\ell) = \widetilde{\mathcal{O}}(m^{2\ell} n(\ell^2 \tau + \sigma + n)). \qquad (29)$$

Remark 13. *To match exaclty the bounds derived in Lem. 7 one should use for M_i the more accurate inequality $M_i < \ell(m-1)m^{\ell-1}$.*

We can isolate the real roots of C_ℓ in $\widetilde{\mathcal{O}}_B(n^5 m^{5\ell} + n^4 m^{5\ell} \tau \ell^2 + n^4 m^{4\ell} \sigma + + n^3 m^{5\ell} \tau^2 \ell^4 + + n^3 m^{3\ell} \sigma^2)$. That is we get a single exponential bound with respect to the number of the real algebraic numbers involved.

Acknowledgement

References

[1] A. G. Akritas and A. Strzeboński. A comparative study of two real root isolation methods. *Nonlinear Analysis: Modelling and Control*, 10:297–304, 2005.

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

[3] F. Boulier, C. Chen, F. Lemaire, and M. Maza. Real root isolation of regular chains. In *Proc. Asian Symposium on Computer Mathematics (ASCM)*, pages 1–15, 2009.

[4] J.-S. Cheng, X.-S. Gao, and C.-K. Yap. Complete numerical isolation of real roots in zero-dimensional triangular systems. *J. Symbolic Computation*, 44:768–785, July 2009.

[5] G. Collins and A. Akritas. Polynomial real root isolation using Descartes' rule of signs. In *SYMSAC '76*, pages 272–275, New York, USA, 1976. ACM Press.

[6] G. E. Collins and R. Loos. Polynomial real root isolation by differentiation. In *Proc. of the 3rd Int'l Symp. on Symbolic and Algebraic Computation*, SYMSAC '76, pages 15–25, New York, NY, USA, 1976. ACM.

[7] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry.* Number 185 in GTM. Springer, New York, 2nd edition, 2005.

[8] J. H. Davenport. Cylindrical algebraic decomposition. Technical Report 88–10, School of Mathematical Sciences, University of Bath, England, URL: http://www.bath.ac.uk/masjhd/, 1988.

[9] D. I. Diochnos, I. Z. Emiris, and E. P. Tsigaridas. On the asymptotic and practical complexity of solving bivariate systems over the reals. *J. Symbolic Computation*, 44(7):818–835, 2009. (Special issue on ISSAC 2007).

[10] Z. Du, V. Sharma, and C. K. Yap. Amortized bound for root isolation via Sturm sequences. In D. Wang and L. Zhi, editors, *Int. Workshop on Symbolic Numeric Computing*, pages 113–129, School of Science, Beihang University, Beijing, China, 2005.

[11] A. Eigenwillig. *Real root isolation for exact and approximate polynomials using Descartes´ rule of signs.* PhD thesis, Doktorarbeit, Universität des Saarlandes, Saarbrücken, 2008.

[12] A. Eigenwillig, L. Kettner, W. Krandick, K. Mehlhorn, S. Schmitt, and N. Wolpert. A Descartes Algorithm for Polynomials with Bit-Stream Coefficients. In V. Ganzha, E. Mayr, and E. Vorozhtsov, editors, *CASC*, volume 3718 of *LNCS*, pages 138–149. Springer, 2005.

[13] A. Eigenwillig, V. Sharma, and C. K. Yap. Almost tight recursion tree bounds for the Descartes method. In *Proc. Annual ACM ISSAC*, pages 71–78, New York, USA, 2006.

[14] I. Z. Emiris, B. Mourrain, and E. P. Tsigaridas. Real Algebraic Numbers: Complexity Analysis and Experimentation. In P. Hertling, C. Hoffmann, W. Luther, and N. Revol, editors, *Reliable Implementations of Real Number Algorithms: Theory and Practice*, volume 5045 of *LNCS*, pages 57–82. Springer Verlag, 2008. (also available in www.inria.fr/rrrt/rr-5897.html).

[15] I. Z. Emiris, B. Mourrain, and E. P. Tsigaridas. The DMM bound: Multivariate (aggregate) separation bounds. In S. Watt, editor, *Proc. 35th ACM Int'l Symp. on Symbolic & Algebraic Comp. (ISSAC)*, pages 243–250, Munich, Germany, July 2010.

[16] J. Johnson and W. Krandick. Polynomial real root isolation using approximate arithmetic. In *Proc. Int'l Symp. on Symbolic and Algebraic Comp. (ISSAC)*, pages 225–232. ACM, 1997.

[17] J. R. Johnson. *Algorithms for Polynomial Real Root Isolation.* PhD thesis, The Ohio State University, 1991.

[18] M. Kerber. On the complexity of reliable root approximation. In V. P. Gerdt, E. W. Mayr, and E. V. Vorozhtsov, editors, *CASC*, volume 5743 of *Lecture Notes in Computer Science*, pages 155–167. Springer, 2009.

[19] R. Loos. Computing in algebraic extensions. In B. Buchberger, G. E. Collins, R. Loos, and R. Albrecht, editors, *Computer Algebra: Symbolic and Algebraic Computation*, pages 173–187. Springer-Verlag, 1983.

[20] Z. Lu, B. He, Y. Luo, and L. Pan. An algorithm of real root isolation for polynomial system. In D. Wang and L. Zhi, editors, *Proc. 1st ACM Int'l Work. Symbolic Numeric Computation (SNC)*, pages 94–107, 2005.

[21] K. Mehlhorn and S. Ray. Faster algorithms for computing Hong's bound on absolute positiveness. *J. Symbolic Computation*, 45(6):677 – 683, 2010.

[22] K. Mehlhorn and M. Sagraloff. A deterministic algorithm for isolating real roots of a real polynomial. *J. Symbolic Computation*, 46(1):70–90, 2011.

[23] M. Mignotte. *Mathematics for Computer Algebra.* Springer-Verlag, New York, 1991.

[24] B. Mourrain, M. Vrahatis, and J. Yakoubsohn. On the complexity of isolating real roots and computing with certainty the topological degree. *J. Complexity*, 18(2), 2002.

[25] V. Pan. Univariate polynomials: Nearly optimal algorithms for numerical factorization and rootfinding. *J. Symbolic Computation*, 33(5):701–733, 2002.

[26] C. Pascal and E. Schost. Change of order for bivariate triangular sets. In *Proc. 31th ACM Int'l Symp. on Symbolic & Algebraic Comp. (ISSAC)*, pages 277–284, New York, NY, USA, 2006.

[27] P. Pedersen and B. Sturmfels. Product formulas for resultants and Chow forms. *Math. Zeitschrift*, 214:377–396, 1993.

[28] R. Rioboo. Towards faster real algebraic numbers. *J. Symb. Comput.*, 36(3-4):513–533, 2003.

[29] F. Rouillier and Z. Zimmermann. Efficient isolation of polynomial's real roots. *J. of Computational and Applied Mathematics*, 162(1):33–50, 2004.

[30] S. Rump. On the sign of a real algebraic number. In *SYMSAC '76: Proc. of the 3rd ACM Symposium on Symbolic and Algebraic Computation*, pages 238–241, New York, USA, 1976.

[31] S. M. Rump. Real root isolation for algebraic polynomials. *ACM SIGSAM Bulletin*, 11(2):327–336, 1977.

[32] M. Sagraloff. On the complexity of real root isolation. *CoRR*, abs/1011.0344, 2010.

[33] A. Schönhage. The fundamental theorem of algebra in terms of computational complexity. Manuscript. Univ. of Tübingen, Germany, 1982.

[34] M. Sombra. The height of the mixed sparse resultant. *Amer. J. Math.*, 126:1253–1260, 2004.

[35] E. P. Tsigaridas. Improved complexity bounds for real root isolation using Continued Fractions. *Arxiv preprint arXiv:1010.2006*, 2010.

[36] E. P. Tsigaridas and I. Z. Emiris. On the complexity of real root isolation using Continued Fractions. *Theor. Comput. Sci.*, 392:158–173, 2008.

[37] M. van Hoeij and M. Monagan. A modular GCD algorithm over number fields presented with multiple extensions. In *Proc. Annual ACM ISSAC*, pages 109–116, July 2002.

[38] B. Xia and L. Yang. An algorithm for isolating the real solutions of semi-algebraic systems. *J. Symbolic Computation*, 34:461–477, November 2002.

[39] B. Xia and T. Zhang. Real solution isolation using interval arithmetic. *Comput. Math. Appl.*, 52:853–860, September 2006.

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

APPENDIX

Proposition 14. *Let $B = \sum_{i,j} c_{i,j} x^i y^j \in \mathbb{Z}[x,y]$ of degree n with respect to y and of degree η with respect to x, and of bitsize σ. Let $A = \sum_{i=0}^{m} a_i x^i \in \mathbb{Z}[x]$ of degree m and bitsize τ. The resultant of B and A with respect to x is univariate polynomial in y of degree at most mn and bitsize at most $m\sigma + \eta\tau + m\lg(n+1) + (m+\eta)\lg(m+\eta)$ or $\widetilde{\mathcal{O}}(m\sigma + \eta\tau)$.*

Proof: The proof follows closely the proof in [2, Prop. 8.15] that provides a bound for general multivariate polynomials. We can compute the resultant of $B(x,y)$ and $A(x)$ with respect to x from the determinant of the Sylvester matrix, by considering them as univariate polynomial in x, with coefficients that are polynomial in y, which is

$$\begin{pmatrix} b_\eta & b_{\eta-1} & \dots & b_0 & & & & \\ & b_\eta & b_{\eta-1} & \dots & b_0 & & & \\ & & \ddots & \ddots & & \ddots & & \\ & & & b_\eta & b_{\eta-1} & \dots & b_0 & \\ a_m & a_{m-1} & \dots & a_0 & & & & \\ & a_m & a_{m-1} & \dots & a_0 & & & \\ & & \ddots & \ddots & & \ddots & & \\ & & & a_m & a_{m-1} & \dots & a_0 & \end{pmatrix} \quad \begin{matrix} x^{m-1}B \\ x^{m-2}B \\ \vdots \\ x^0 B \\ x^{\eta-1}A \\ x^{\eta-2}A \\ \vdots \\ x^0 A \end{matrix}$$

where $b_k = \sum_{i=0}^{n} c_{i,k} y^i$.

The resultant is a factor of the determinant of the Sylvester matrix. The matrix is of size $(\eta+m) \times (\eta+m)$, hence the determinant consists of $(\eta+m)!$ terms. Each term is a product of m univariate polynomials in y, of degree n and bitsize σ, times the product of n numbers, of bitsize τ. The first product results in polynomials in y of degree at most mn and bitsize at most $m\sigma + m\lg(n+1)$; since there are at most $(n+1)^m$ terms with bitsize at most $m\sigma$ each. The second product results in numbers of bitsize at most $\eta\tau$. Hence each term of the determinant is, in the worst case a univariate polynomial in y of degree m and bitsize $m\sigma + \eta\tau + m\lg(n+1)$. We conclude that the resultant is of degree at most mn in y and of bitsize $m\sigma + \eta\tau + m\lg(n+1) + (m+\eta)\lg(m+\eta)$ or $\widetilde{\mathcal{O}}(m\sigma + \eta\tau)$. □