# Univariate Real Root Isolation in Multiple Extension Fields

Adam Strzebonski, Elias Tsigaridas

# Univariate Real Root Isolation in Multiple Extension Fields

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

Elias P. Tsigaridas
POLSYS Project INRIA Paris-Rocquencourt
UPMC, Univ Paris 06, LIP6, FRANCE
elias@polsys.lip6.fr

## ABSTRACT

We present algorithmic, complexity and implementation results for the problem of isolating the real roots of a univariate polynomial in $B_{\boldsymbol{\alpha}} \in L[y]$, where $L = \mathbb{Q}(\alpha_1, \ldots, \alpha_\ell)$ is an algebraic extension of the rational numbers. Our bounds are single exponential in $\ell$ and match the ones presented in [34] for the case $\ell = 1$. We consider two approaches. The first, indirect approach, using multivariate resultants, computes a univariate polynomial with integer coefficients, among the real roots of which are the real roots of $B_{\boldsymbol{\alpha}}$. The Boolean complexity of this approach is $\widetilde{\mathcal{O}}_B(N^{4\ell+4})$, where $N$ is the maximum of the degrees and the coefficient bitsize of the involved polynomials. The second, direct approach, tries to solve the polynomial directly, without reducing the problem to a univariate one. We present an algorithm that generalizes Sturm algorithm from the univariate case, and modified versions of well known solvers that are either numerical or based on Descartes' rule of sign. We achieve a Boolean complexity of $\widetilde{\mathcal{O}}_B(\min\{N^{4\ell+7}, N^{2\ell^2+6}\})$ and $\widetilde{\mathcal{O}}_B(N^{2\ell+4})$, respectively. We implemented the algorithms in C as part of the core library of MATHEMATICA and we illustrate their efficiency over various data sets.

**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

We consider the problem of isolating the real roots of a univariate polynomial the coefficients of which are polynomial functions of real algebraic numbers, that is they belong to multiple algebraic extensions.

We use $\mathbf{x}^{\mathbf{e}}$ 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 $f = \sum_{j=1}^{m} c_j \mathbf{x}^{\mathbf{e}_j} \in \mathbb{Z}[\mathbf{x}]$, 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. $(\#Q) = |Q \cap \mathbb{Z}^\ell|$. The problem that we consider is the following:

**Problem 1.** *Let $\alpha_j$, $1 \leq j \leq \ell$, be real algebraic numbers. Their isolating interval representation is $\alpha_j \cong (A_j, \mathcal{I}_j)$, where $A_j = \sum_{i=0}^{m} a_i x_j^i$, $\mathcal{I}_j = [\mathsf{a}_{j,1}, \mathsf{a}_{j,2}]$, $\mathsf{a}_{1,2} \in \mathbb{Q}$, $\mathsf{deg}(A_j) = m$, and $\mathcal{L}(A_j) = \tau$. Let $B_{\boldsymbol{\alpha}} = \sum_{i=0}^{n} b_i(\alpha_1, \ldots, \alpha_\ell)\, y^i \in \mathbb{Z}(\alpha)[y]$, be square-free, where $b_i(\mathbf{x}) = \sum_{j=0}^{\eta} c_{ij}\, \mathbf{x}^{\boldsymbol{e}_j} \in \mathbb{Z}[\mathbf{x}]$, $\mathcal{L}(c_{i,j}) \leq \sigma$, $\boldsymbol{e}_j = (e_{j,1}, \ldots, e_{j,\ell})$, and $e_{j,i} < m$, for $0 \leq i \leq d$. What is the Boolean complexity of isolating the real roots of $B_{\boldsymbol{\alpha}}$?*

We denote by $\mathbf{a}_i$ the coefficients of $A_i$, where $1 \leq i \leq \ell$, and by $\mathbf{c}$ the coefficients of $B$. The problem of isolating the real roots of a univariate polynomial with coefficients that are polynomial functions of real algebraic numbers, is not as well studied as the case of polynomials with integer coeffiecients. Nevetheless is a problem of great importance as it appears as a subproblem in various algorithms, for example cylindrical algebraic decomposition, computing the topology of curves and surfaces, etc.

One of the first systematic studies of the problem appeared in Rump [29, 30]. A complete treatment of various direct and indirect algorithms appeared in Johnson PhD thesis [19], see also [18]. A modern treatment of these algorithms and improved separation bounds were presented in [34]. The algorithms that we present are a generalization of the algorithms in [19, 34] in the multivariate case. Along the same lines, it is worth mentioning the work of Johnson and Krandick [18] and Rouillier and Zimmermann [28] that introduced variants for isolating the real roots of a univariate polynomial which are based on certified use of approximate arithmetic and are very efficient in practice. The latter approach has an optimal memory usage.

A similar approach is a bitstream version of Descartes' algorithm [14, 15, 22], where the coefficients of the input polynomial could be real numbers that we approximate up to arbitrary precision. In our implementation we use a variant due to Sagraloff [31] as a subalgorithm for tackling Problem 1. An even more recent variant, that also exploits Newton's iteration seems to be even more efficient [32].

The state-of-the-art algorithms, at least from the complexity point of view, for solving polynomials are the numerical algorithms due to Pan [24] and Schönhage [33]. We use these algorithms for analysing the complexity of the various approaches for tackling Problem 1. For a recent approach we refer the reader to [32].

The approaches for solving Problem 1 could be used as


*ISSAC'12,* July 22–25, 2012, Grenoble, FRANCE.

subalgorithms in the process of real root isolation of triangular systems and regular chains. Hence, many researchers working on algorithms for triangular systems and/or regular chains considered procedures for solving algebraic polynomials. There is a sequence of papers [7, 8, 9, 10, 21, 35, 36] with many algorithms and implementations for real solving of triangular polynomial systems. Their common point of efficiency is the clever use of interval arithmetic and the so-called *sleeve* polynomials. The isolation process is based on exclusion and inclusion predicates. In [10] the authors introduce evaluation bounds that they use to guarantee the termination of the subdivision process. These algorithms are adaptive and are based on the worst case bound only from a theoretical point of view. In the case where the triangular systems do not have multiple roots, they provide an efficient algorithm which does not need evaluation bounds but only test the sign of the derivative(s).

We should also mention algorithms for isolating the real roots of zero-dimensional regular chains [2]. In this line of research, the author modify and generalize the algorithm of Vincent-Collins-Akritas (or Descartes) algorithm to isolate the real roots of polynomials with polynomial functions of real algebraic numbers as coefficients. The choice of algorithm for solving, as in [34], is similar to the direct algorithms that we present in Sec. 3. However, to our knowledge, a complexity analysis of the algorithms for solving of regular chains is missing. Let us also mention that there is a complete and efficient implementation of the algorithms in [2] in MAPLE, that is also part of the official release.

There is also the work of Rioboo [26, 27] that provides purely symbolic algorithms for various operations with real algebraic numbers, based on quasi Sylvester sequences. These algorithms could also be used for solving Problem 1, and they are closely connected with the Sturm algorithm that we present (Sec. 3.4). However, we are not aware of any complexity analysis and the subalgorithms that we use for sign evaluations and computation of polynomial sequences are different.

**Our Results.** The results in the current paper generalize the results of [34] to multiple extension fields.

We consider two approaches for isolating the real roots of $B_{\boldsymbol{\alpha}}$. An indirect approach (Sec. 4), where we compute the minimal polynomial of the real roots using (multivariate or succesive) resultant computations and sign evaluations. We obtain a bound of $\widetilde{\mathcal{O}}_B(N^{4\ell+4})$ (Th. 14), where $N$ is the maximum of the degrees and the maximum coefficient bitisize of the involved polynomials and $\ell$ is the number of the extensions. The direct approach (Sec. 3) consists of two solvers. One based on Sturm's algorithm, with bit complexity $\widetilde{\mathcal{O}}_B(\min\{N^{4\ell+7}, N^{2\ell^2+6}\})$ (Th. 11) and one based on a modification of univariate real root isolation algorithms, with bit complexity $\widetilde{\mathcal{O}}_B(N^{2\ell+4})$ (Th. 9). For both approaches we prove exact (aggregate) separation bounds (Lem. 8 and Th. 12). We also present two algorithms for computing the sign of multivariate polynomial evaluated over real algebraic numbers. One based on approximations (Cor. 7) and one based on recursive Sturm sequence computations (Th. 10).

All the complexity bounds are single exponential with respect to the number of variables (that is the number of simple extensions) and match the ones presented in [34] in the case of a single extension.

We have implemented the indirect approach and one of the direct approaches in C as part of the core library of MATHEMATICA and we present experiments with various datasets (Sec. 5).

**Notation** $\mathcal{O}_B$ means bit complexity and the $\widetilde{\mathcal{O}}_B$-notation means that we are ignoring logarithmic factors. For $A = \sum_{i=1}^{d} a_i x^i \in \mathbb{Z}[x]$, $\mathsf{deg}(A)$ 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 $\alpha_1, \ldots, \alpha_d$ are the distinct, possible complex, roots of $A$, then $\Delta_i = |\alpha_i - \alpha_{c_i}|$, where $\alpha_{c_i}$ is the roots closest to $\alpha_i$. $\Delta = \min_i \Delta_i$ is the separation bound of $A$, that is the smallest distance between two (real or complex, depending on the context) roots of $A$. The following quantity is also useful $\Sigma(A) = -\sum_{i=1}^{n} \lg \Delta_i$, that expresses the numbers of bits that we need in order 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. THE DMM BOUND

We present lower/upper bounds and separation bounds for the real roots of univariate polynomials and polynomial systems. For the univariate case we refer the reader to e.g. [11, 13, 19, 23]. The multivariate separation bounds that we use, were introduced in [17]. For other multivariate separation bounds we refer the reader to [3, 5, 37].

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

$$
\begin{aligned}
|\gamma_i| &\leq 2\|f\|_\infty \leq 2^{\tau+1} \ , && (1)\\
-\lg \Delta(f) &\leq -\frac{1}{2}\lg|3\,\mathsf{disc}(f_r)| + \frac{p+2}{2}\lg(p) + (p-1)\lg\|f_r\|_2 \\
&\leq 2p\lg p + p\tau \ , && (2)\\
-\sum_i \lg \Delta_i(f) &\leq -\frac{1}{2}\lg|\mathsf{disc}(f_r)| + \frac{p^2-p-2}{2} + (2p-1)\lg\|f_r\|_2 \\
&\leq 3p^2 + 3p\tau + 4p\lg p \ , && (3)
\end{aligned}
$$

*where $f_r$ is the square-free part of $f$, and for the second inequalities we assume $f \in \mathbb{Z}[x]$ and $\mathcal{L}(f) = \tau$.*

Let $n > 1$ be the number of variables. We use $\mathbf{x}^{\boldsymbol{e}}$ to denote the monomial $x_1^{e_1}\cdots x_n^{e_\ell}$, with $\boldsymbol{e} = (e_1, \ldots, e_\ell) \in \mathbb{N}^\ell$. For a polynomial $f = \sum_{j=1}^{\nu} c_j \mathbf{x}^{\mathbf{e}_j} \in \mathbb{Z}[\mathbf{x}]$, let $\{\mathbf{e}_1, \ldots, \mathbf{e}_\nu\} \subset \mathbb{N}^\ell$ 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. $(\#Q) = |Q \cap \mathbb{Z}^\ell|$. Consider the polynomials $f_i = \sum_{j=1}^{\nu_i} c_{i,j}\mathbf{x}^{a_{i,j}}$, $1 \leq i \leq n$, each of total degree $d_i$. Let $\{a_{i,1}, \ldots, a_{i,\nu_i}\} \subset \mathbb{Z}^n$ be the support of $f_i$; its Newton polytope $Q_i$ is the convex hull of the support. Let $\mathsf{MV}(Q_1, \ldots, Q_n) > 0$ be the *mixed volume* of convex polytopes $Q_1, \ldots, Q_n \subset \mathbb{R}^n$. By $Q_0$ we denote the standard simplex. We consider polynomial system

$$(\Sigma): \ f_1(\mathbf{x}) = f_2(\mathbf{x}) = \cdots = f_n(\mathbf{x}) = 0, \qquad (4)$$

where $f_i \in \mathbb{Z}[\mathbf{x}^{\pm 1}]$, Let $\mathsf{M}_i = \mathsf{MV}(Q_0, \ldots, Q_{i-1}, Q_{i+1}, \ldots, Q_n)$. Wlog, assume $\dim \sum_{i=0}^{n} Q_i = n$ and $\dim \sum_{i\in I} Q_i \geq j$ for any $I \subset \{0, \ldots, n\}$ with $|I| = j$. Let $\mathrm{vol}(\cdot)$ stand for the Euclidean volume. We need the following inequalities: $\mathsf{M}_0 \leq \prod_{i=1}^{n} d_i$, $\mathcal{B} \leq n\prod_{i=1}^{n} d_i^2$, $\mathsf{M}_i \leq \prod_{\substack{1\leq j\leq n\\ j\neq i}} d_j$, $(\#Q_i) \leq n!\,\mathrm{vol}(Q_i) + n \leq 2d_i^n$, $\mathcal{A} = \prod_{i=1}^{n}\sqrt{\mathsf{M}_i}\,2^{\mathsf{M}_i}$, $\mathcal{C} = \prod_{i=1}^{n}\|f_i\|_\infty^{M_j}$, $h \leq (n+1)^D\varrho$, and $\varrho = \prod_{i=1}^{n}(\#Q_i)^{M_i}$.

**Theorem 2 (DMM$_n$).** *[17] Consider the polynomial system $(\Sigma)$ in (4), which is not necessarily 0-dimensional, and where it holds that $f_i \in \mathbb{Z}[\boldsymbol{x}]$, $\deg(f_i) \leq d$, and $\mathcal{L}(f_i) \leq \tau$. Let $D$ be the number of the isolated points of the solution set in $(\mathbb{C}^*)^n$, which are $0 < |\gamma_1| \leq |\gamma_2| \leq \cdots \leq |\gamma_D|$. Let $\Omega$ be any set of $\ell$ couples of indices $(i,j)$ such that $1 \leq i < j \leq D$, and $\gamma_{j,k}$ stands for the $k$-th coordinate of $\gamma_j$. Then*

$$\prod_{(i,j)\in\Omega} \|\gamma_i - \gamma_j\|_2 \geq 2^{-\ell-(\mathsf{M}_0-1)(\mathsf{M}_0+2)/2}\,(h\,\mathcal{C}\,\mathcal{A})^{1-\mathsf{M}_0-\ell}\,\mathcal{B}^{(1-n)(\mathsf{M}_0^2+\mathsf{M}_0(\ell-1)+\ell)}, \quad (5)$$

$$(2^{\mathsf{M}_0}\,\varrho\,\mathcal{C}\,\mathcal{A})^{-1} \leq |\gamma_{j,k}| \leq 2^{\mathsf{M}_0}\,\varrho\,\mathcal{C}\,\mathcal{A}, \quad (6)$$

$$\Delta(\Sigma) \geq 2^{-(3\mathsf{M}_0+2)(\mathsf{M}_0-1)/2}\,(\sqrt{\mathsf{M}_0+1}\,\varrho\,\mathcal{C}\,\mathcal{A})^{-\mathsf{M}_0}. \quad (7)$$

*The univariate polynomial that has $R_i$, $1 \leq i \leq n$, that has $\gamma_{i,k}$ as complex solutions is of degree $\mathsf{M}_0$ and $\mathcal{L}(R_i) \leq \lg(2\,h\,\mathcal{A}\,\mathcal{C}\,\mathcal{B}^{\ell\,\mathsf{M}_0})$. If the system is 0-dim then we can skip $\mathcal{A}$ from the previous bounds.*

# 3. DIRECT METHODS

## 3.1 Coefficient bounds and sign computation

The following proposition gives an extension of the BFMSS bound [4] providing an additional rule for polynomial expressions in algebraic numbers.

**Proposition 3.** *Let $\xi_1, \ldots, \xi_\ell$ be algebraic numbers and let $\xi = \sum_{e\in S} c_e \xi_1^{e_1} \ldots \xi_\ell^{e_\ell}$, where $S \subseteq \mathbb{N}^\ell \cap ([0,m_1]\times\ldots\times[0,m_\ell])$ and $c_e \in \mathbb{Z}$ for $e \in S$. Suppose that for each $1 \leq i \leq \ell$ there exist algebraic integers $\beta_i$ and $\gamma_i$ and positive reals $u_i$ and $v_i$ such that $\xi_i = \frac{\beta_i}{\gamma_i}$ and any conjugates $\beta_i^*$ and $\gamma_i^*$ of $\beta_i$ and $\gamma_i$ satisfy $|\ \beta_i^*\ | \leq u_i$ and $|\ \gamma_i^*\ | \leq v_i$. Let $u = \sum_{e\in S} |\ c_e\ | u_1^{e_1} \ldots u_\ell^{e_\ell} v_1^{m_1-e_1} \ldots v_\ell^{m_\ell-e_\ell}$ and $v = v_1^{m_1} \ldots v_\ell^{m_\ell}$. Then there exist algebraic integers $\beta$ and $\gamma$ such that $\xi = \frac{\beta}{\gamma}$ and any conjugates $\beta^*$ and $\gamma^*$ of $\beta$ and $\gamma$ satisfy $|\ \beta^*\ | \leq u$ and $|\ \gamma^*\ | \leq v$. Moreover, if $d$ is an upper bound on the algebraic degree of $\xi$ then*

$$\frac{1}{u^{d-1}v} \leq |\ \xi\ | \leq uv^{d-1} \quad (8)$$

**Proof:** Let $\beta := \sum_{e\in S} c_e \beta_1^{e_1} \ldots \beta_\ell^{e_\ell} \gamma_1^{m_1-e_1} \ldots \gamma_\ell^{m_\ell-e_\ell}$ and $\gamma := \gamma_1^{m_1} \ldots \gamma_\ell^{m_\ell}$. Then $\beta$ and $\gamma$ are algebraic integers and $\xi = \frac{\beta}{\gamma}$. For $1 \leq i \leq \ell$, let $g_i$ and $h_i$ be the minimal polynomials of $\beta_i$ and $\gamma_i$. Let $f := z - \sum_{e\in S} c_e x_1^{e_1} \ldots x_\ell^{e_\ell} y_1^{m_1-e_1} \ldots y_\ell^{m_\ell-e_\ell}$, $f \in \mathbb{Z}[x_1,\ldots,x_\ell;y_1,\ldots,y_\ell;z]$, and let

$$r := res_{x_1}(\ldots(res_{x_\ell}(res_{y_1}(\ldots(res_{y_\ell}(f,h_\ell),\ldots),h_1),g_\ell),\ldots),g_1) \in \mathbb{Z}[z],$$

Then the roots of $r$ are

$$\sum_{e\in S} c_e (\beta_1^*)^{e_1} \ldots (\beta_\ell^*)^{e_\ell} (\gamma_1^*)^{m_1-e_1} \ldots (\gamma_\ell^*)^{m_\ell-e_\ell} \quad (9)$$

with arbitrary conjugates $\beta_i^*$ and $\gamma_i^*$ of $\beta_i$ and $\gamma_i$ for $1 \leq i \leq \ell$. Since $r \in \mathbb{Z}[z]$ and $r(\beta) = 0$, the minimal polynomial of $\beta$ divides $r$. Hence any conjugate $\beta^*$ of $\beta$ has the form (9) and so $|\ \beta^*\ | \leq u$. By a similar reasoning, for any conjugate $\gamma^*$ of $\gamma$, $|\ \gamma^*\ | \leq v$. Inequality (8) follows from Lem. 2 of [4]. □

**Remark 4.** *The BFMSS bound is computed one arithmetic operation at a time. Proposition 3 provides an additional rule which computes bounds on polynomial expressions in one step. To show that adding the rule improves the resulting bounds let us consider a simplified problem where $S = \{e \in \mathbb{N}^\ell : e_1 + \ldots + e_\ell = m\}$, $u_1 = \ldots = u_\ell$, $v_1 = \ldots = v_\ell$, $|\ c_e\ | \leq a$ for all $e \in S$, and $\xi_1, \ldots, \xi_\ell$ all have the algebraic degree $d_1$. For $e \in S$ let $E_e := c_e \xi_1^{e_1} \ldots \xi_\ell^{e_\ell}$. With the notation of [4], we get $u(E_e) = a u_1^m$, $l(E_e) = v_1^m$, and $D(E_e) = d_1^m$. Let $E := \sum_{e\in S} E_e$. Note that $\xi$ is the value of $E$. The BFMSS rules yield $u(E) = k a u_1^m v_1^{k-1}$, $l(E) = v_1^k$, and $D(E) = d_1^{km}$, where $k = \#S = \binom{m+\ell-1}{\ell-1}$. Hence the BFMSS bound becomes*

$$\frac{1}{(k a u_1^m v_1^{k-1})^{d_1^{km}-1} v_1^k} \leq |\ \xi\ | \leq k a u_1^m v_1^{k-1} v_1^{d_1^{km}-1}$$

*The rule of Proposition 3 yields $u = k a u_1^m v_1^{(\ell-1)m}$, $v = v_1^{lm}$, and $d = d_1^\ell$. Hence (8) becomes*

$$\frac{1}{(k a u_1^m v_1^{(\ell-1)m})^{d_1^\ell-1} v_1^{\ell m}} \leq |\ \xi\ | \leq k a u_1^m v_1^{(\ell-1)m} v_1^{d_1^\ell-1}$$

*If $m > 1$ and $\ell > 1$ then $k = \ell(\ell+1)\cdot\ldots\cdot(\ell+m-1) > \ell m$ and hence the bounds given by Proposition 3 are tighter than the BFMSS bounds.*

**Corollary 5.** *With the notation of Problem 1,*

$$2^{-m^\ell(\sigma+\ell m(\tau+2))} \leq |b_i(\alpha_1,\ldots,\alpha_\ell)| \leq 2^{\sigma+\ell m(\tau+2)} \quad (10)$$

**Proof:** For each $1 \leq j \leq l$ let $\gamma_j \in \mathbb{Z}$ be the absolute value of the leading coefficient of $A_j$. Then $\gamma_j \leq 2^\tau$, $\beta_j := \gamma_j \alpha_j$ is an algebraic integer and any conjugate $\beta_j^*$ of $\beta_j$ satisfies $|\ \beta_j^*\ | \leq 2^{\tau+1}$. By Proposition 3 $|\ b_i(\alpha_1,\ldots,\alpha_l)\ | \geq \frac{1}{u^{d-1}v}$, where $u \leq m^\ell 2^\sigma 2^{\ell m(\tau+1)}$, $v \leq 2^{\ell m\tau}$, and $d \leq m^\ell$.

Therefore $\lg(\frac{1}{u^{d-1}v}) \geq -(m^\ell-1)(\ell\,\lg(m) + \sigma + \ell m(\tau+1)) - \ell m\tau \geq -m^\ell(\sigma + \ell m(\tau+2))$. Since $|\alpha_j| \leq 2^{\tau+1}$, $|b_i(\alpha_1,\ldots,\alpha_l)| \leq m^\ell\, 2^\sigma 2^{(\tau+1)(m-1)\ell} \leq 2^{\sigma+\ell m(\tau+2)}$. □

**Lemma 6.** *We can approximate $b_i(\alpha_1,\ldots,\alpha_\ell)$ to accuracy of $L$ bits in $\widetilde{\mathcal{O}}_B(\ell m^{\ell+1}(L+\sigma+\ell m\tau))$.*

**Proof:** We have $b_i(\alpha_1,\ldots,\alpha_\ell) = \sum_{j=0}^{\eta_i} c_{i,j} \alpha_1^{e_{j,1}} \ldots \alpha_\ell^{e_{j,\ell}}$. Suppose $|\ \alpha_k^* - \alpha_k\ | \leq 2^{-p}$ for $1 \leq k \leq \ell$ and some $p \in \mathbb{N}_+$. Since $|\ c_{i,j}\ | \leq 2^\sigma$, $|\ \alpha_k\ | \leq 2^{\tau+1}$, $|\ \alpha_k^*\ | \leq 2^{\tau+2}$, $e_{j,k} < m$ and $\eta_i \leq m^\ell$

$$|\ b_i(\alpha_1^*,\ldots,\alpha_\ell^*) - b_i(\alpha_1,\ldots,\alpha_\ell)\ | \leq m^\ell \ell 2^\sigma 2^{-p} 2^{(\tau+2)(m-1)(\ell-1)}$$

Hence

$$\begin{aligned} &-\lg|\ b_i(\alpha_1^*,\ldots,\alpha_\ell^*) - b_i(\alpha_1,\ldots,\alpha_\ell) \geq \\ &\quad p - \ell\lg(m) - \lg(\ell) - \sigma - (\tau+2)(m-1)(\ell-1) \geq \\ &\quad\quad p - \sigma - \ell m(\tau+3) \end{aligned}$$

Therefore to obtain an approximation of $b_i(\alpha_1,\ldots,\alpha_\ell)$ to accuracy of $L$ bits it is sufficient to approximate each $\alpha_k$ to accuracy of $L + \sigma + \ell m(\tau+3)$ bits. The complexity of computing $c_{i,j}(\alpha_1^*)^{e_{j,1}} \ldots (\alpha_\ell^*)^{e_{j,\ell}}$ is bounded by $\widetilde{\mathcal{O}}_B(\ell m(L+\sigma+\ell m\tau))$ and the complexity of computing $b_i(\alpha_1^*,\ldots,\alpha_\ell^*)$ is bounded by $\widetilde{\mathcal{O}}_B(m^\ell \ell m(L+\sigma+\ell m\tau))$.

Using [20] the bit complexity of approximating $\alpha_k$ to accuracy $a$ is $\widetilde{\mathcal{O}}_B(m^3\tau^2 + m^2 a)$. However, we can ignore the first summand if we assume fast root isolation algorithm. Then the bound on approximating all $\alpha_k$ to accuracy of $L + \sigma + \ell m(\tau+3)$ bits becomes $\widetilde{\mathcal{O}}_B(\ell m^2(L+\sigma+\ell m\tau))$ which is dominated. □

**Corollary 7.** *The bit complexity of computing the sign of $b_i(\alpha_1,\ldots,\alpha_\ell)$ is bounded by $\widetilde{\mathcal{O}}_B(\ell m^{2\ell+1}(\sigma+\ell m\tau))$.*

**Proof:** By Prop. 3, to compute the sign of $b_i(\alpha_1,\ldots,\alpha_\ell)$ it is sufficient to approximate $b_i(\alpha_1,\ldots,\alpha_\ell)$ to bit accuracy of $L = m^\ell(\sigma+\ell m(\tau+2))+1$. By Lemma 6 this can be done with bit complexity of $\widetilde{\mathcal{O}}_B(\ell m^{\ell+1}(m^\ell(\sigma+\ell m\tau)+\sigma+\ell m\tau))$. □

## 3.2 Separation bounds

**Lemma 8.** *Let $B_\alpha$ be as in Problem 1, and $\xi_i$ be its roots. Then, it holds*

$$\begin{aligned}
\lg|\xi_i| &\leq 2\sigma m^\ell + 2\ell^2\tau m^\ell + 8\ell^2 m^\ell \lg(m\ell) \ , && (11)\\
-\lg\Delta(B_{\boldsymbol{\alpha}}) &\leq 4n\sigma m^\ell + 4\tau\ell^2 n m^\ell + 12n\ell^2 m^\ell \lg(mn\ell) \ , && (12)\\
-\sum_i \lg\Delta_i(B_{\boldsymbol{\alpha}}) &\leq 4n\sigma m^\ell + 4\tau\ell^2 n m^\ell + 16n\ell^2 m^\ell \lg(mn\ell) + n^2 && (13)
\end{aligned}$$

$$\begin{aligned}
\lg|\xi_i| &= \widetilde{\mathcal{O}}(m^\ell(\sigma+\ell^2\tau)) \ , && (14)\\
-\lg\Delta(B_{\boldsymbol{\alpha}}) &= \widetilde{\mathcal{O}}(m^\ell n(\sigma+\ell^2\tau)) \ , && (15)\\
\Sigma(B_{\boldsymbol{\alpha}}) = -\sum_i \lg\Delta_i(B_{\boldsymbol{\alpha}}) &= \widetilde{\mathcal{O}}(m^\ell n(\sigma+\ell^2\tau)+n^2) && (16)
\end{aligned}$$

**Proof:** We compute various bounds on the roots of $B_{\boldsymbol{\alpha}}$ based on the first inequalities of Prop. 1. For this we need to bound $|\mathsf{disc}(B_{\boldsymbol{\alpha}})|$ and $\|B_{\boldsymbol{\alpha}}\|_2$. First we bound the coefficients of $B_{\boldsymbol{\alpha}}$. We consider the following systems

$$(S_{b_i}) \quad \begin{cases} A_1(\mathbf{x}) = \sum_{i=0}^{m} a_{1,i}\, x_1^i = 0 \\ \quad\vdots \\ A_\ell(\mathbf{x}) = \sum_{i=0}^{m} a_{\ell,i}\, x_\ell^i = 0 \\ A_{\ell+1} = z - b_i(x_1,\ldots,x_\ell) = 0 \end{cases} ,$$

where $1 \leq i \leq n$. The variables are $(x_1,\ldots,x_\ell,y)$. Upper and lower bounds on the coordinates of the isolated solutions of $(S_{b_i})$ bound $b_i(\alpha_1,\ldots,\alpha_\ell)$. To compute such bounds we use Th. 2. The system is zero dimensional so we can skip $\mathcal{A}$.

It holds $\|A_i\|_\infty \leq 2^\tau$ for $1 \leq i \leq \ell$ and $\|A_{\ell+1}\|_\infty \leq 2^\sigma$.

The number of possible isolated solutions of $(S_{b_i})$ is bounded by $\mathsf{M}_0 = m^\ell$, and $\mathsf{M}_i = \ell(m-1)m^{\ell-1} \leq \ell m^\ell$ and $\mathsf{M}_\ell \leq m^\ell$. Moreover, $C = \prod_{i=1}^{\ell+1}\|A_i\|_\infty^{M_j} = \|A_{\ell+1}\|_\infty^{M_{\ell+1}}\prod_{i=1}^{\ell}\|A_i\|_\infty^{M_j}$, and so $\lg(C) \leq m^\ell\sigma + \ell^2 m^\ell\tau$. Finally, $\varrho \leq (\ell m)^{3\ell^2 m^\ell}$. Combining the previous inequalities with (6) we get that for every $i$ it holds

$$2^{-\sigma m^\ell - \ell^2\tau m^\ell - 4\ell^2 m^\ell \lg(m\ell)} \leq |b_i(\alpha_1,\ldots,\alpha_\ell)| \ .$$

For the upper bound we have:

$$\begin{aligned}
|b_i(\boldsymbol{\alpha})| = \left|\sum_{j=0}^{\eta} c_{ij}\,\boldsymbol{\alpha}^{\mathbf{e}_j}\right| &\leq \left|\sum_{j_1,\ldots,j_\ell=0}^{m-1} c_{i,j}\,\alpha_1^{j_1}\cdots\alpha_\ell^{j_\ell}\right| \\
&\leq \sum_{j_1,\ldots,j_\ell=0}^{m-1}\left|2^\sigma\,(2^\tau)^{j_1+\cdots+j_\ell}\right| \leq 2^{\sigma+\ell m\tau} \ ,
\end{aligned}$$

and overall

$$2^{-\sigma m^\ell - \ell^2\tau m^\ell - 4\ell^2 m^\ell\lg(m\ell)} \leq |b_i(\alpha_1,\ldots,\alpha_\ell)| \leq 2^{\sigma+\ell m\tau} \ . \quad (17)$$

Cauchy's bound indicates $|\xi_i| \leq \max_{i\neq n}|b_i(\boldsymbol{\alpha})/b_n(\boldsymbol{\alpha})|$, and combining it with 17 we prove (11) and (14). To bound $\|B_{\boldsymbol{\alpha}}\|_2$ we use the definition of the 2-norm and (17), that is

$$\|B_{\boldsymbol{\alpha}}\|_2^2 \leq \sum_{i=0}^{n}(b_i(\boldsymbol{\alpha}))^2 \leq (n+1)\,2^{2\sigma m^\ell + 2\ell^2\tau m^\ell + 8\ell^2 m^\ell\lg(m\ell)} \ . \quad (18)$$

We bound $|\mathsf{disc}(B_{\boldsymbol{\alpha}})|$ using the identity

$$\begin{aligned}
\mathsf{disc}(B_{\boldsymbol{\alpha}}) &= (-1)^{\frac{1}{2}n(n-1)}\frac{1}{b_n(\boldsymbol{\alpha})}\mathtt{res}_y(B_{\boldsymbol{\alpha}}, \tfrac{d}{dy}B_{\boldsymbol{\alpha}}(y)) \\
&= (-1)^{\frac{1}{2}n(n-1)}\frac{1}{b_n(\boldsymbol{\alpha})}\,R_B(\boldsymbol{\alpha}) \ ,
\end{aligned} \quad (19)$$

where the resultant, $R_B \in \mathbb{Z}[\boldsymbol{\alpha}]$, is the determinant of the Sylvester matrix of $B_{\boldsymbol{\alpha}}$ and $\frac{d}{dy}B_{\boldsymbol{\alpha}}(y)$, evaluated over $\boldsymbol{\alpha}$. The matrix is of size $(2n-1)\times(2n-1)$, the elements of which belong to $\mathbb{Z}[\boldsymbol{\alpha}]$. The determinant consists of $(2n-1)! \leq (2n-1)^{2n-1}$ terms. Each term is a product of $n-1$ polynomials in $\ell$ variables of total degree $\ell(m-1)$ and bitsize $\sigma$ times the product of $n$ polynomials in $\ell$ variables of total degree at most $\ell(m-1)$ and bitsize at most $\sigma+\lg(n)$. The first product results polynomial total degree $\leq (n-1)\ell(m-1)$ and bitsize $\leq n\sigma + n\ell\lg(\ell nm)$. The second product results polynomials of total degree $\leq n\ell(m-1)$ and bitsize $\leq n\sigma + 2n\ell\lg(\ell nm)$ Hence, each term is a polynomial in $\ell$ variables of total degree at most $\ell(2n-1)(m-1)$ and bitsize at most $(2n-1)\sigma + 6n\ell\lg(\ell nm)$. We conclude that determinant is a polynomial in $\ell$ variables, $R_B \in \mathbb{Z}[x_1,\ldots,x_\ell]$, of degree at most $(2n-1)(m-1) = \mathcal{O}(mn)$ in each variable, of total degree at most $\ell(2n-1)(m-1) = \mathcal{O}(\ell mn)$ and bitsize at most $(2n-1)\sigma + 6n\ell\lg(\ell nm) + \lg((2n-1)!) \leq (2n-1)\sigma + 9n\ell\lg(\ell nm) = \mathcal{O}(n\sigma + n\ell)$.

To bound $R_B(\boldsymbol{\alpha}) = R_B(\alpha_1,\ldots,\alpha_\ell)$ we consider the following system

$$(S_{R_B}) \quad \begin{cases} A_1(\mathbf{x}) = \sum_{i=0}^{m} a_{1,i}\, x_1^i = 0 \\ \quad\vdots \\ A_\ell(\mathbf{x}) = \sum_{i=0}^{m} a_{\ell,i}\, x_\ell^i = 0 \\ A_{\ell+1} = z - R_B(x_1,\ldots,x_\ell) = 0 \end{cases} ,$$

where the variables are $(x_1,\ldots,x_\ell,z)$. Upper and lower bounds on the coordinates of isolated solutions of $(S_{R_B})$ also bound $R_B(\alpha_1,\ldots,\alpha_\ell)$. Since the system is zero dimensional, we do not need the extended version of Th. 2, that is we skip $\mathcal{A}$, to compute such bounds.

It holds $\|A_i\|_\infty \leq 2^\tau$ for $1 \leq i \leq \ell$ and $\lg\|A_{\ell+1}\|_\infty \leq (2n-1)\sigma + 9n\ell\lg(\ell nm)$.

The number of possible isolated solutions of $(S_{R_B})$ is bounded by $\mathsf{M}_0 \leq m^\ell$, and $\mathsf{M}_i \leq 2\ell nm^\ell$ and $\mathsf{M}_{\ell+1} \leq m^\ell$. Moreover, $\mathcal{C} = \prod_{i=1}^{\ell+1}\|A_i\|_\infty^{M_j} = \|A_{\ell+1}\|_\infty^{M_{\ell+1}}\prod_{i=1}^{\ell}\|A_i\|_\infty^{M_j}$, and so $\lg(C) \leq 2n\sigma m^\ell + 2\tau\ell^2 n m^\ell + 9n\ell m^\ell\lg(mn\ell)$. Finally, $\log|\varrho| \leq 2\ell^2 n m^\ell\lg(nm\ell)$. Combining all the previous inequalities with (6) we get

$$\begin{aligned}
&2^{-2n\sigma m^\ell - 2\tau\ell^2 n m^\ell - 12n\ell^2 m^\ell\lg(mn\ell)} \leq |R_B(\alpha_1,\ldots,\alpha_\ell)| \\
&|R_B(\alpha_1,\ldots,\alpha_\ell)| \leq 2^{2n\sigma m^\ell + 2\tau\ell^2 n m^\ell + 12n\ell^2 m^\ell\lg(mn\ell)}
\end{aligned} \quad (20)$$

Using (20) and (17), (19) becomes

$$\begin{aligned}
&2^{-3n\sigma m^\ell - 3\tau\ell^2 n m^\ell - 16n\ell^2 m^\ell\lg(mn\ell)} \leq |\mathsf{disc}(B_{\boldsymbol{\alpha}})| = \left|\tfrac{R_B(\boldsymbol{\alpha})}{b_n(\boldsymbol{\alpha})}\right| \\
&|\mathsf{disc}(B_{\boldsymbol{\alpha}})| = \left|\tfrac{R_B(\boldsymbol{\alpha})}{b_n(\boldsymbol{\alpha})}\right| \leq 2^{3n\sigma m^\ell + 3\tau\ell^2 n m^\ell + 16n\ell^2 m^\ell\lg(mn\ell)}
\end{aligned} \quad (21)$$

If we plug in (21) and (18) to (2) and (3) we get (12), (13), (15) and (16). □

## 3.3 A modified Univariate algorithm

If we approximate the coefficients of $y^n + \frac{b_{n-1}(\boldsymbol{\alpha})}{b_n(\boldsymbol{\alpha})} + \ldots + \frac{b_0(\boldsymbol{\alpha})}{b_n(\boldsymbol{\alpha})}$ to accuracy $\mathcal{O}(-\sum_i \lg\Delta_i(B_{\boldsymbol{\alpha}}) + n\tau_B)$, where $|\frac{b_i(\boldsymbol{\alpha})}{b_n(\boldsymbol{\alpha})}| \leq 2^{\tau_B}$, we obtain a (univariate) polynomial $\widetilde{B_{\boldsymbol{\alpha}}}$ with binary rational coefficients. By [31] to isolate the real roots of $B_{\boldsymbol{\alpha}}$ it suffices to isolate the real roots of $\widetilde{B_{\boldsymbol{\alpha}}}$.

Polynomial $\widetilde{B_{\boldsymbol{\alpha}}}$ is a univariate of degree $n$ and maximum coefficient bitsize $\mathcal{O}(-\sum_i \lg\Delta_i(B_{\boldsymbol{\alpha}}) + n\tau_B)$. We can isolate

the real roots of this polynomial in $\widetilde{\mathcal{O}}_B(n^3(-\sum_i \lg \Delta_i(B_{\boldsymbol{\alpha}})+ n\tau_B))$ [24, 33]. From (16), $-\sum_i \lg \Delta_i(B_{\boldsymbol{\alpha}}) = \widetilde{\mathcal{O}}(m^\ell n(\sigma + \ell^2\tau) + n^2)$ and from (17) $\tau_B = \widetilde{\mathcal{O}}(m^\ell(\sigma + \ell^2\tau))$. The complexity of isolating the real roots of $\widetilde{B_{\boldsymbol{\alpha}}}$, and hence $B_{\boldsymbol{\alpha}}$, is $\widetilde{\mathcal{O}}_B(m^\ell n^4(\sigma + \ell^2\tau) + n^5)$. It remains to estimate the cost of computing the successive approximations of $b_i(\boldsymbol{\alpha})/b_n(\boldsymbol{\alpha})$. Lem. 6 indicates that we can approximate the coefficients of $B_{\boldsymbol{\alpha}}$ up to accuracy $L$ in $\widetilde{\mathcal{O}}_B(\ell\, m^{\ell+1}(L+\sigma+\ell m\tau))$. By (17) to approximate $b_i(\boldsymbol{\alpha})/b_n(\boldsymbol{\alpha})$ to accuracy $L$ it suffices to approximate $b_i(\alpha)$, for $0 \leq i \leq n$, to accuracy $\widetilde{\mathcal{O}}(L + m^\ell(\sigma + \ell^2\tau))$. In our case $L = \widetilde{\mathcal{O}}(m^\ell n(\sigma + \ell^2\tau) + n^2)$ and there are $n+1$ coefficients to approximate, so the cost of approximation is $\widetilde{\mathcal{O}}_B(\ell\, m^{2\ell+1}n^2(\sigma + \ell^2\tau) + \ell m^{\ell+1}n^3)$.

**Theorem 9.** *We can solve Problem 1 using the algorithms of Pan [24] or Schönhage [33] in* $\widetilde{\mathcal{O}}_B(m^\ell n^4(\sigma + \ell^2\tau) + n^5 + \ell\, m^{2\ell+1}n^2(\sigma + \ell^2\tau) + \ell\, m^{\ell+1}n^3)$, *or* $\widetilde{\mathcal{O}}_B(N^{2\ell+4})$, *where* $N = \max\{m, n, \sigma, \tau\}$.

## 3.4 The Sturm solver

We study STURM algorithm, a pure symbolic subdivision-based algorithm, for isolating the real roots of $B_{\boldsymbol{\alpha}}$, and as before we assume that $B_{\boldsymbol{\alpha}}$ is square-free. First we prove a theorem for multivariate sign evaluation using Sturm sequences recursively, which is of independent interest.

**Theorem 10.** *Let* $F \in \mathbf{Z}[x_1, \ldots, x_n]$ *be polynomial in* $n$ *variables with integer coefficients of maximum bitsize* $\sigma$ *and of degree* $< m$, *wrt every variable* $x_i$. *Let* $\alpha_i$ *be a real root a the polynomial* $A_i \in \mathbf{Z}[x]$, *such that* $\deg(A_i) = m$ *and* $\mathcal{L}(A_i) = \tau$. *The cost of computing* $\operatorname{sgn}(F(\alpha_1, \ldots, \alpha_n))$, *using Sturm(-Habicht) sequences is* $\widetilde{\mathcal{O}}_B(m^{2n(n-1)}(\sigma + m\tau))$.

**Proof:** We want to compute the sign of the evaluation $F(\alpha_1, \ldots, \alpha_n)$. The real algebraic number $\alpha_i$, that is real root of $A_i$, is represented by an isolating interval, $I_i = [\mathsf{a}_{i,1}, \mathsf{a}_{i,2}]$, the endpoints of which are bitsize $\mathcal{O}(m\tau)$. We compute the sign using nested Sturm sequence computation.

*Stage 1.* Initially, we consider the polynomial $F$ as univariate in $x_1$, and we call it $F_1$. That is $F_1 \in (\mathbf{Z}[x_2, \ldots, x_n])[x_1]$. In order to compute the sign of $F_1$ over a real root, $\alpha_1$, of $A_1(x_1)$, we compute the Sturm-Habicht sequence of $A_1$ and $F_1$ wrt to $x_1$, and evaluate it over the two endpoints of $I_1$. We denote these (evaluated) sequences by $S_{1,1} = \mathbf{SR}(A_1, F_1|e_{1,1})$ and $S_{1,2} = \mathbf{SR}(A_1, F_1|e_{1,2})$. The sequences contain less than $2m$ polynomials in $\mathbf{Z}[x_2, \ldots, x_n]$ of degree bounded by $\leq m^2$, wrt to $x_i$, where $i \in \{2, \ldots, n\}$, and bitsize $\widetilde{\mathcal{O}}(m(\sigma + m\tau))$. This computation costs $\widetilde{\mathcal{O}}_B(m^{2n}(\sigma + m\tau))$ [12, Th. 10].

*Stage 2.* At the second step, we consider every polynomial in $S_{1,1}$ and $S_{1,2}$ as univariate wrt to $x_2$, and we call it $B_2$. That is $F_2 \in (\mathbf{Z}[x_3, \ldots, x_n])[x_2]$. To compute the sign of $F_2$ evaluate it over a real root, $\alpha_2$, of $A_2$, we compute the Sturm-Habicht sequence of $A_2$ and $F_2$ wrt to $x_2$, and evaluate it over the two endpoints of $I_2$. We denote these (evaluated) sequences by $S_{2,1} = \mathbf{SR}(A_2, F_2|e_{2,1})$ and $S_{2,2} = \mathbf{SR}(A_2, F_2|e_{2,2})$. The sequences contain less than $2m$ polynomials in $\mathbf{Z}[x_3, \ldots, x_n]$ of degree bounded by $\leq m^3$, wrt to $x_i$, where $i \in \{3, \ldots, n\}$, and bitsize $\widetilde{\mathcal{O}}(m^3(\sigma + m\tau))$. This computation costs $\widetilde{\mathcal{O}}_B(m^{4n-4}(\sigma + m\tau))$ [12, Th. 10]. We have to perform this computation $\mathcal{O}(m)$ times, because this is the number of polynomials in $S_{1,1}$ and $S_{1,2}$ and so the total cost is $\widetilde{\mathcal{O}}_B(m^{4n-3}(\sigma + m\tau))$.

We have $\mathcal{O}(m^2)$ polynomials in $\mathbf{Z}[x_3, \ldots, x_n]$ at this stage.

*Stage k.* At stage $k$, we have $\mathcal{O}(m^{k-1})$ polynomials from stage $k-1$. We consider each of them, $F_k$, as univariate in $x_k$, that is $F_k \in (\mathbf{Z}[x_{k+1}, \ldots, x_n])[x_k]$. The degree of $F_k$ wrt to $x_k$ is $\mathcal{O}(m^k)$ and bitsize $\widetilde{\mathcal{O}}_B(m^{k(k-1)/2}(\sigma + m\tau))$.

As before, to compute the sign of $F_k$ evaluate it over a real root, $\alpha_k$, of $A_k$, we compute the Sturm-Habicht sequence of $A_k$ and $F_k$ wrt to $x_k$, and evaluate it over the two endpoints of $I_k$. We denote these (evaluated) sequences by $S_{k,1} = \mathbf{SR}(A_k, F_k|e_{k,1})$ and $S_{k,2} = \mathbf{SR}(A_k, F_k|e_{k,2})$. The sequences contain less than $2m$ polynomials in $\mathbf{Z}[x_{k+1}, \ldots, x_n]$ of degree bounded by $\leq m^{k+1}$, wrt to $x_i$, where $i \in \{k+1, \ldots, n\}$, and bitsize $\widetilde{\mathcal{O}}(m^{k(k+1)/2}(\sigma + m\tau))$. This computation costs $\widetilde{\mathcal{O}}_B(m^{(2n+1/2)k-3k^2/2+1}(\sigma + m\tau))$ [12, Th. 10]. We have to perform this computation $\mathcal{O}(m^{k-1})$ times, because this is the number of polynomials from the previous stage, and so the total cost is $\widetilde{\mathcal{O}}_B(m^{(2n+3/2)k-3k^2/2}(\sigma + m\tau))$.

We have $\mathcal{O}(m^k)$ polynomials in $\mathbf{Z}[x_{k+1}, \ldots, x_n]$ at this stage.

*Overall cost.* To derive the overall cost, we sum over all $k$, where $1 \leq k \leq n$. Then
$\widetilde{\mathcal{O}}_B(\sum_{k=1}^n m^{k-1}\, m^{(2n+1/2)k-3k^2/2+1}(\sigma + m\tau))$,
which is $\widetilde{\mathcal{O}}_B(m^{2n(n-1)}(\sigma + m\tau))$. □

We consider $B_{\boldsymbol{\alpha}}$ as a polynomial in $y$ and we evaluate the Sturm sequence of $B(\boldsymbol{\alpha}, y)$ and its derivatice, $\partial B(\boldsymbol{\alpha}, y)/\partial y$, over various rational numbers. The number of steps of the algorithm, $(\#T)$ depends on the separation bound [11, 13]. In our case

$$(\#T) \leq 2r + \lg \mathsf{B} - \lg \prod \Delta_i(B_{\boldsymbol{\alpha}}) = \widetilde{\mathcal{O}}(m^\ell n(\sigma + \ell^2\tau)) \ , \quad (22)$$

where $r$ is the number of real roots, $\mathsf{B}$ is an upper bound on their magnitude, and the last equality follows from Lem 8.

It remains to estimate the complexity of each step, i.e. the cost of evaluating the Sturm sequence over a rational number, say of bitsize $L$. In the worst case $L$ equals the bitsize of the separation bound, i.e. $\widetilde{\mathcal{O}}(m^\ell n(\sigma + \ell^2\tau))$.

We consider $B$ as polynomial in $\mathbf{Z}[x_1, \ldots, x_\ell][y]$. The Sturm sequence of $B$ and $\partial B/\partial y$ contains $\mathcal{O}(n)$ polynomials. At each step we evaluate these polynomials over rational numbers of bitsize at most $L$. This means that we get $\mathcal{O}(n)$ polynomials in $\mathbf{Z}[x_1, \ldots, x_\ell]$ of degree $\mathcal{O}(nm)$ in $x_i$ and of maximum bitsize $\widetilde{\mathcal{O}}(n(\tau + L)) = \mathcal{O}(m^\ell n^2(\sigma + \ell^2\tau))$.

We should compute their sign when we perform the substitution $x_i = \alpha_i$. For this we use Th. 10 and we deduce that the sign evaluation costs $\mathcal{O}(m^{2\ell^2-\ell}n^2(\sigma + \ell^2\tau))$. For the whole sequence the costs becomes $\mathcal{O}(m^{2\ell^2-\ell}n^3(\sigma + \ell^2\tau))$.

For the overall cost of STURM we should multiply the previous bound with the number of steps, $(\#T)$, and the cost becomes $\widetilde{\mathcal{O}}_B(m^{2\ell^2}n^4(\sigma^2 + \ell^4\tau^2)$.

The previous bound matches the one presented in [34] for $\ell = 1$. However, if $\ell > 2$ we can do better. If instead of Th. 10 we use Cor. 7, then the cost of one sign evaluation becomes $\widetilde{\mathcal{O}}_B(\ell m^{3\ell+1}n^2(\sigma + \ell^2\tau))$, and the cost of evaluating the whole Sturm sequence is $\widetilde{\mathcal{O}}_B(\ell m^{3\ell+2}n^3(\sigma + \ell^2\tau))$. If we multiply by the number of steps, we end up with a bound

of $\widetilde{\mathcal{O}}_B(\ell m^{4\ell+1}n^4(\sigma^2+\ell^4\tau^2))$, or $\widetilde{\mathcal{O}}_B(N^{4\ell+7})$. This bound is better that the one of Th. 11 when $\ell > 2$.

**Theorem 11.** *We can solve Problem 1 using* STURM *solver and sign evaluations which exploit recursive Sturm sequences in $\widetilde{\mathcal{O}}_B(m^{2\ell^2}n^4(\sigma^2+\ell^4\tau^2)$, or $\widetilde{\mathcal{O}}_B(N^{2\ell^2+6})$, where $N = \max\{m, n, \sigma, \tau\}$. If we use approximations for sign evaluations then we obtain a bound of $\widetilde{\mathcal{O}}_B(\ell m^{4\ell+1}n^4(\sigma^2+\ell^4\tau^2))$, or $\widetilde{\mathcal{O}}_B(N^{4\ell+7})$.*

The aforementioned bounds suggest that recursive Sturm sequences should, if at all, used only in the presense of a small number of variables, 1 or 2. Beyond this approximation should be used for sign evaluations. This agrees with the practical experience.

# 4. REDUCTION TO INTEGER COEFFICIENTS

In this section we tackle Problem 1 using a reduction to a polynomial with integer coefficients. The analysis improves the bounds from [34] and also applies to degenerate cases.

**Theorem 12.** *Let $B_{\boldsymbol{\alpha}}$ as in Problem 1. The minimal polynomial, $C \in \mathbb{Z}[x]$, of the, possible complex, roots of $B_{\boldsymbol{\alpha}}$, $\gamma_i$, has degree $\leq nm^\ell$ and bitsize $\leq m^{\ell-1}\ell(\ell m+n)\tau + m^\ell\sigma + 9n\ell^2m^\ell\lg(mn\ell)$, or $\widetilde{\mathcal{O}}(m^{\ell-1}(\ell^2m\tau+\ell n\tau+m\sigma))$. It holds*

$$|\lg|\gamma_i|| \leq \widetilde{\mathcal{O}}(m^{\ell-1}(\ell^2m\tau+\ell n\tau+m\sigma)) \ , \quad (23)$$

$$-\lg\Delta(C) = \widetilde{\mathcal{O}}(n\,m^{2\ell-1}(\ell^2m\tau+\ell n\tau+m\sigma)) \quad (24)$$

$$\Sigma(C) = -\sum_i \lg\Delta_i(C) = \widetilde{\mathcal{O}}(n\,m^{2\ell-1}(\ell^2m\tau+\ell n\tau+m\sigma)) \quad (25)$$

**Proof:** We consider the following polynomial system:

$$(S) \quad \begin{cases} A_1(\mathbf{x}) = \sum_{i=0}^{m} a_{1,i}\,x_1^i = 0 \\ \vdots \\ A_\ell(\mathbf{x}) = \sum_{i=0}^{m} a_{\ell,i}\,x_\ell^i = 0 \\ A_{\ell+1} = B(\mathbf{x}, y) = \sum_{i=0}^{n} b_i(x_1,\ldots,x_\ell)\,y^i = 0 \end{cases},$$

where the variables are $(x_1, \ldots, x_\ell, y)$. We compute the various quantities of Th. 2 which appear just before the statement of the theorem.

We notice that $\|A_1\|_\infty = \cdots = \|A_\ell\|_\infty = 2^\tau$ and $\|A_{\ell+1}\|_\infty = \sigma$, and the total degrees of the first $\ell$ polynomials is $m$, and $\deg(A_{\ell+1}) \leq n + \ell(m-1)$. Moreover $\mathsf{M}_1 = \cdots = \mathsf{M}_\ell = m^{\ell-1}(n+\ell(m-1)) \leq m^{\ell-1}(n+\ell m)$, $\mathsf{M}_{\ell+1} = m^\ell$, and $\mathsf{M}_0 = nm^\ell$. The latter bound is an upper bound on the number of solution of the system.

The following two inequalities are easy to obtain:

$$\begin{aligned}\mathcal{C} &= \textstyle\prod_{i=1}^{\ell}\|A_i\|_\infty^{\mathsf{M}_1}\cdot\|A_{\ell+1}\|_\infty^{\mathsf{M}_{\ell+1}} \leq 2^{\tau\ell\mathsf{M}_1}\cdot 2^{\sigma\mathsf{M}_{\ell+1}} \\ &\leq 2^{m^{\ell-1}(\tau\ell^2m+\tau\ell n+\sigma m)}\end{aligned} .$$

$$\begin{aligned}\mathcal{A} &= \textstyle\prod_{i=1}^{\ell}\sqrt{\mathsf{M}_1}\cdot 2^{\mathsf{M}_1}\cdot\sqrt{\mathsf{M}_{\ell+1}}\cdot 2^{\mathsf{M}_{\ell+1}} \\ &\leq m^{\ell-1}[\ell m+n)2^{2m^{\ell-1}(\ell m+n)}]^{\ell/2}\cdot m^{\ell/2}\cdot 2^{m^\ell} \\ &\leq 2^{\ell m^{\ell-1}(m\ell+n)+m\ell+\ell^2\lg(\ell mn)}\end{aligned} .$$

The first $\ell$ polynomials of $(S)$ are univariate, hence their Newton polytope is a segment and so it is easy to estime the number of lattice points it contains. There are $(\#Q_1) = \cdots = (\#Q_\ell) = m+1$. For the last polynomial, it holds $(\#Q_{\ell+1}) \leq 2(\ell m+n)^{\ell+1}$.
Finally, $\varrho \leq (m+1)^{m^\ell}[2(\ell m+n)^{\ell+1}]^{m^\ell} \leq 2^{\ell m^{\ell-1}(2m\ell+n)\lg(mn\ell)}$, $h \leq (\ell+2)^{\mathsf{M}_0}\cdot\varrho \leq 2^{m^{\ell-1}(mn+m\ell^2+n\ell)\lg(mn\ell)}$, $\mathcal{B} \leq (l+1)\mathsf{M}_0^2$, and $\mathcal{B}^{\ell\mathsf{M}_0} \leq 2^{4n\ell^2m^\ell\lg(mn\ell)}$.

The univariate polynomial that has the roots of $B_{\boldsymbol{\alpha}}$ as solution, $R_{\ell+1}$ is of degree $\leq \mathsf{M}_0 \leq nm^\ell$ and $\mathcal{L}(R_{\ell+1}) \leq m^{\ell-1}\ell(\ell m+n)\tau + m^\ell\sigma + 9n\ell^2m^\ell\lg(mn\ell)$. □

**Remark 13.** *The bitsize of $C$ could be slightly improved if we take into account that we need project on one variable and use a u-resultant of special form. Then its bitsize becomes $\lg(2^{\mathsf{M}_0}\varrho\mathcal{C}\mathcal{A})$. However, this only affects the constants and does not alter the asymptotic behavior of the bound(s).*

We can eliminate the variables $x_1, \ldots, x_\ell$ in $(S)$ to obtain the polynomial $C$ using various methods, for example Gröbner basis computation, (sparse) resultants. The best complexity bound is obtained using multivariate (sparse) resultants [6, 16]. However, this bound is dominated by the complexity of isolating the real roots of $C$ and checking if they are roots of $B_{\boldsymbol{\alpha}}$.

We can isolate the real roots of $C$ in $\widetilde{\mathcal{O}}_B(n^3m^{4\ell-1}(\ell^2m\tau+\ell n\tau+m\sigma))$ [24, 33]. The endpoints of the isolating intervals have (total) bitsize $\widetilde{\mathcal{O}}(m^{2\ell-1}n(\ell^2m\tau+\ell n\tau+m\sigma))$.

It remains to check which of the real roots of $C$ are roots of $B_{\boldsymbol{\alpha}}$. For this, since we assume that $B_{\boldsymbol{\alpha}}$ is square free, given an isolating interval of a real root of $C$, it suffices to check if $B_{\boldsymbol{\alpha}}$ changes sign when it is evaluated over the endpoints of the interval. At this point we may also use a change of ordering algorithm for triangular sets to avoid working with possible non-squre-free polynomials. For the bivariate case of this algorithm we refer the reader to [25].

Let $\mathsf{c}_k$ be the $k$-th isolating point and $\mathcal{L}(\mathsf{c}_k) = s_k$. We perform the substitution $y = \mathsf{c}_k$, and after clearing denominators, we get a number in $\mathbb{Z}[\alpha]$, for which we want to compute its sign. This is the same as computing the sign of the (multivariate) polynomial $B_k = B(x_1, \ldots, x_\ell, \mathsf{c}_k)$ evaluated over the (real) algebraic numbers $\alpha_1, \ldots, \alpha_\ell$.

The degree of $B_k$ with respect to $x_i$ is bounded by $m$ and $\mathcal{L}(B_k) = ns_k + \tau$.

One sign evaluation costs $\widetilde{\mathcal{O}}_B(\ell m^{2\ell-1}(ns_k+\tau))$ (Cor. 7), and all of them cost

$$\widetilde{\mathcal{O}}_B(\ell m^{2\ell-1}(n\sum_k s_k+\tau)) = \widetilde{\mathcal{O}}_B(\ell m^{4\ell-1}n(\ell^2m\tau+\ell n\tau+m\sigma)) \ .$$

**Theorem 14.** *We can solve Problem 1 using RIC algorithm in $\widetilde{\mathcal{O}}_B(n^3m^{4\ell-1}(\ell^2m\tau+\ell n\tau+m\sigma))$, or $\widetilde{\mathcal{O}}_B(N^{4\ell+4})$, where $N = \max\{m, n, \sigma, \tau\}$.*

The derived bound is single exponential wrt the number of variables. Moreover, they match the bounds for the single extension case, $\ell = 1$, presented in [34].

# 5. IMPLEMENTATION AND EXPERIMENTS

We compare implementations of two methods of real root isolation for squarefree polynomials over simple algebraic extensions of rationals. The first method, *RIC* (for Reduction to Integer Coefficients), performs reduction to integer coefficients described in Section 4. For isolating roots of polynomials with integer coefficients it uses the *Mathematica* implementation of the Continued Fractions algorithm [1]. The

| $\ell$ | $n$ | Algorithm | $m=2$ | $m=3$ | $m=5$ | $m=10$ |
|---|---|---|---|---|---|---|
| 2 | 10 | *RIC* | 0.004 | 0.025 | 0.482 | 83.3 |
| | | *BMD* | 0.003 | 0.006 | 0.019 | 0.090 |
| | 20 | *RIC* | 0.008 | 0.060 | 0.155 | 334 |
| | | *BMD* | 0.006 | 0.013 | 0.032 | 0.179 |
| | 50 | *RIC* | 0.039 | 0.303 | 10.4 | $> 3600$ |
| | | *BMD* | 0.027 | 0.040 | 0.087 | 0.423 |
| | 100 | *RIC* | 0.196 | 0.980 | 47.7 | $> 3600$ |
| | | *BMD* | 0.100 | 0.139 | 0.228 | 0.970 |
| 3 | 10 | *RIC* | 0.017 | 0.359 | 421 | $> 3600$ |
| | | *BMD* | 0.005 | 0.017 | 0.096 | 1.10 |
| | 20 | *RIC* | 0.051 | 0.916 | 1612 | $> 3600$ |
| | | *BMD* | 0.009 | 0.031 | 0.161 | 2.17 |
| | 50 | *RIC* | 0.348 | 5.61 | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.034 | 0.102 | 0.437 | 5.28 |
| | 100 | *RIC* | 1.89 | 31.6 | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.136 | 0.242 | 1.03 | 11.4 |
| 4 | 10 | *RIC* | 0.233 | 38.7 | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.010 | 0.055 | 0.657 | 18.5 |
| | 20 | *RIC* | 0.729 | 129 | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.023 | 0.120 | 1.27 | 43.9 |
| | 50 | *RIC* | 4.73 | 764 | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.067 | 0.310 | 2.99 | 89.1 |
| | 100 | *RIC* | 25.3 | $> 3600$ | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.164 | 0.722 | 6.46 | 201 |
| 5 | 10 | *RIC* | 1.35 | $> 3600$ | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.022 | 0.254 | 4.50 | 233 |
| | 20 | *RIC* | 6.18 | $> 3600$ | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.039 | 0.391 | 7.76 | 461 |
| | 50 | *RIC* | 57.2 | $> 3600$ | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.115 | 1.27 | 19.4 | 1207 |
| | 100 | *RIC* | 408 | $> 3600$ | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.238 | 2.43 | 38.9 | $> 3600$ |

**Table 1.** Randomly generated dense polynomials

second method, *BMD* (for Bitstream Modified Descartes), uses Sagraloff's modified version of Descartes' algorithm [31]. Zero testing of the leading coefficients is done using the method described in Section 3.1. The algorithm has been implemented in C as a part of the *Mathematica* system. We have not performed experiments with Sturm's algorithm because it is not efficient at all, and, at least from our experience, can not handle even small degree cases.

The experiments have been run on 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 1 hour of CPU time are reported as $> 3600$.

**(Randomly generated dense polynomials)** For given values of $l$, $m$ and $n$ each problem was generated as follows. First, univariate polynomials of degree $m$ with uniformly distributed random 10-bit integer coefficients were generated until $l$ relatively prime irreducible polynomials were obtained, each of which had real roots. Then $\alpha_1, \ldots, \alpha_l$ were obtained by randomly selecting one real root of each of the polynomials. Finally, a polynomial $f \in \mathbb{Z}[x_1, \ldots, x_l; y]$ of degree $n$ in $y$ and degree $m-1$ in each of $x_i$ with 10-bit random integer coefficients was generated and $B_\alpha$ was defined as $B_\alpha(y) := f(\alpha_1, \ldots, \alpha_l; y)$. The results of the experiment 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 $L_n^\lambda(y)$, where $\lambda = \alpha_1 + \ldots + \alpha_l$ and $\alpha_i$ is the smallest root of the generalized Laguerre polynomial $L_m^{i-1}(x)$. $B_\alpha(y)$ is taken to be $L_n^\lambda(y)$ reduced modulo all the minimal polynomials of $\alpha_i$ to ensure that the degree of $B_\alpha$ in $\alpha_i$ is at most $m-1$. Note that $L_n^\lambda(y)$ has $n$ positive roots for any positive $\lambda$ and $L_m^{i-1}(x)$ has $m$ positive roots, so this example maximizes the number of real roots of both the input

| $\ell$ | $n$ | Algorithm | $m=2$ | $m=3$ | $m=5$ | $m=10$ |
|---|---|---|---|---|---|---|
| 2 | 10 | *RIC* | 0.009 | 0.102 | 1.59 | 299 |
| | | *BMD* | 0.005 | 0.007 | 0.014 | 0.021 |
| | 20 | *RIC* | 0.046 | 0.618 | 15.8 | $> 3600$ |
| | | *BMD* | 0.020 | 0.041 | 0.041 | 0.091 |
| | 50 | *RIC* | 0.617 | 13.1 | 458 | $> 3600$ |
| | | *BMD* | 0.205 | 0.306 | 0.288 | 0.454 |
| | 100 | *RIC* | 7.76 | 137 | $> 3600$ | $> 3600$ |
| | | *BMD* | 2.04 | 1.57 | 2.30 | 2.37 |
| 3 | 10 | *RIC* | 0.015 | 1.55 | 908 | $> 3600$ |
| | | *BMD* | 0.006 | 0.013 | 0.063 | 0.067 |
| | 20 | *RIC* | 0.064 | 25.0 | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.020 | 0.053 | 0.126 | 0.674 |
| | 50 | *RIC* | 0.923 | $> 3600$ | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.273 | 0.273 | 0.590 | 4.15 |
| 4 | 10 | *RIC* | 0.087 | 112 | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.008 | 0.031 | 0.147 | 0.235 |
| | 20 | *RIC* | 0.426 | 1695 | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.034 | 0.119 | 0.729 | 4.75 |
| | 50 | *RIC* | 8.43 | 764 | $> 3600$ | ? |
| | | *BMD* | 0.270 | 0.395 | 2.90 | ? |
| 5 | 10 | *RIC* | 0.465 | $> 3600$ | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.011 | 0.115 | 0.522 | 1.13 |
| | 20 | *RIC* | 4.11 | $> 3600$ | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.047 | 0.269 | 7.03 | 38.6 |

**Table 2.** Generalized Laguerre polynomials

polynomial with algebraic number coefficients and the polynomial with integer coefficients obtained by *RIC*. The results of the experiment are given in Table 2. For some values of $l$, $m$ and $n$ the computation of $B_\alpha(y)$, that is the reduction of $L_n^\lambda(y)$ modulo all the minimal polynomials of $\alpha_i$, failed due to insufficient memory. The corresponding table entries are marked with "?". If the computation of $B_\alpha(y)$ failed for some $l$ and $n$ and all values of $m$, the corresponding row has been omitted.

**(Generalized Wilkinson Polynomials)** This example uses the following generalized Wilkinson polynomials $W_{n,\lambda}(y) := \prod_{k=1}^{n}(y - k\lambda)$, where $\lambda = \alpha_1 + \ldots + \alpha_l$ and $\alpha_i$ is the smallest root of the generalized Laguerre polynomial $L_m^{i-1}(x)$. $B_\alpha(y)$ is taken to be $W_{n,\lambda}(y)$ reduced modulo all the minimal polynomials of $\alpha_i$ to ensure that the degree of $B_\alpha$ in $\alpha_i$ is at most $m-1$. The results of the experiment are given in Table 3. For some values of $l$ and $n$ the computation of $B_\alpha(y)$, that is the reduction of $W_{n,\lambda}(y)$ modulo all the minimal polynomials of $\alpha_i$, failed due to insufficient memory for all values of $m$. The corresponding rows have been omitted.

In all the experiments *RIC* is faster than *BMD* and the timing ratio increases with the number and the degrees of the algebraic numers. The computation time of *RIC* depends directly on $l$ and $m$, since it isolates roots of a polynomial of degree $m^l n$. On the other hand, the main root isolation loop of *BMD* depends only on the geometry of roots, which depends on $\ell$ and $m$ only through the worst case lower bound on root separation. The only part of *BMD* that depends directly on $\ell$ and $m$ is the computation of approximations of coefficients. We can observe that, unlike for the case of $\ell = 1$ [34], this part of the algorithm can dominate the computation time of *BMD* for large values of $\ell$ and $m$. This is due to the exponential growth of the size of expressions representing $b_i(\alpha_1, \ldots, \alpha_\ell)$.

**Acknowledgments.** ET is partially supported by an individual postdoctoral grant from the Danish Agency for Science, Technology and Innovation, and also acknowledges support from the Danish National Research Foundation and the National Science Foundation of China (under grant 61061130540) for the Sino-Danish Center for the

| $\ell$ | $n$ | Algorithm | $m=2$ | $m=3$ | $m=5$ | $m=10$ |
|---|---|---|---|---|---|---|
| 2 | 10 | *RIC* | 0.023 | 0.066 | 2.78 | 383 |
| | | *BMD* | 0.006 | 0.007 | 0.020 | 0.016 |
| | 20 | *RIC* | 0.051 | 0.411 | 18.2 | $> 3600$ |
| | | *BMD* | 0.021 | 0.024 | 0.042 | 0.169 |
| | 50 | *RIC* | 0.659 | 8.23 | 617 | $> 3600$ |
| | | *BMD* | 0.217 | 0.214 | 0.288 | 1.99 |
| | 100 | *RIC* | 5.42 | 131 | $> 3600$ | $> 3600$ |
| | | *BMD* | 1.38 | 1.89 | 3.41 | 9.06 |
| 3 | 10 | *RIC* | 0.015 | 1.77 | 834 | $> 3600$ |
| | | *BMD* | 0.006 | 0.013 | 0.039 | 0.036 |
| | 20 | *RIC* | 0.067 | 18.6 | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.021 | 0.043 | 0.260 | 0.564 |
| | 50 | *RIC* | 0.748 | 828 | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.215 | 0.470 | 0.995 | 8.51 |
| | 100 | *RIC* | 8.63 | $> 3600$ | $> 3600$ | $> 3600$ |
| | | *BMD* | 1.58 | 2.76 | 7.72 | 46.4 |
| 4 | 10 | *RIC* | 0.079 | 107 | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.008 | 0.042 | 0.133 | 0.101 |
| | 20 | *RIC* | 0.445 | 2087 | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.027 | 0.179 | 0.735 | 3.45 |
| | 50 | *RIC* | 7.78 | $> 3600$ | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.288 | 0.788 | 3.64 | 78.0 |
| 5 | 10 | *RIC* | 0.515 | $> 3600$ | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.010 | 0.118 | 0.546 | 0.531 |
| | 20 | *RIC* | 3.62 | $> 3600$ | $> 3600$ | $> 3600$ |
| | | *BMD* | 0.035 | 0.345 | 5.67 | 34.2 |

**Table 3.** Generalized Wilkinson polynomials

Theory of Interactive Computation, within which part of this work was performed, and from the EXACTA grant of the National Science Foundation of China (NSFC 60911130369) and the French National Research Agency (ANR-09-BLAN-0371-01). ET performed part of this work while he was with the Aarhus University, Denmark.

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

[3] W. D. Brownawell and C. K. Yap. Lower bounds for zero-dimensional projections. In *Proc. 34th ACM Int'l Symp. on Symbolic & Algebraic Comp. (ISSAC)*, Seoul, Korea, 2009.

[4] C. Burnikel, S. Funke, K. Mehlhorn, S. Schirra, and S. Schmitt. A separation bound for real algebraic expressions. *Algorithmica*, 55(1):14–28, 2009.

[5] J. Canny. *The Complexity of Robot Motion Planning*. ACM Doctoral Dissertation Award Series. MIT Press, 1987.

[6] J. Canny, E. Kaltofen, and Y. Lakshman. Solving systems of non-linear polynomial equations faster. In *Proc. ACM Intern. Symp. on Symbolic & Algebraic Comput.*, pages 121–128, 1989.

[7] C. Chen and M. Moreno Maza. Algorithms for computing triangular decompositions of polynomial systems. In *Proc. 36th ACM Int'l Symp. on Symbolic & Algebraic Comp. (ISSAC)*, pages 83–90, USA, 2011.

[8] C. Chen and M. Moreno Maza. Algorithms for computing triangular decompositions of polynomial systems by. *J. Symbolic Computation*, 2012.

[9] J.-S. Cheng and X.-S. Gao. Multiplicity preserving triangular set decomposition of two polynomials. Technical report, MM-Preprints, 2011.

[10] 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.

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

[12] 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.

[13] 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, Beihang University, Beijing, China, 2005. Birkhauser.

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

[15] 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.

[16] I. Emiris and V. Pan. Improved algorithms for computing determinants and resultants. *J. Complexity, Special Issue*, 21:43–71, 2005. Special Issue on FOCM-02.

[17] 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.

[18] 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.

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

[20] M. Kerber and M. Sagraloff. Efficient real root approximation. In *Proc. 36th ACM Int'l Symp. on Symbolic & Algebraic Comp. (ISSAC)*, pages 209–216, San Jose, CA, USA, June 2011. ACM.

[21] 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.

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

[25] 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.

[26] R. Rioboo. Towards faster real algebraic numbers. In T. Mora, editor, *Proc. Annual ACM ISSAC*, pages 221–228, New York, NY 10036, USA, 2002. ACM Press.

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

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

[29] S. Rump. On the sign of a real algebraic number. In *SYMSAC '76: Proceedings of the third ACM symposium on Symbolic and algebraic computation*, pages 238–241, New York, NY, USA, 1976. ACM Press.

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

[31] M. Sagraloff. On the complexity of real root isolation. *CoRR*, abs/1011.0344v1, 2010.

[32] M. Sagraloff. When Newton meets Descartes: A simple and fast algorithm to isolate the real roots of a polynomial. To appear in *Proc. 37th ACM Int'l Symp. on Symbolic & Algebraic Comp.(ISSAC)*, 2012.

[33] A. Schönhage. The fundamental theorem of algebra in terms of computational complexity. Manuscript. Univ. of Tübingen, Germany, 1982. URL: http://www.iai.uni-bonn.de/~schoe/fdthmrep.ps.gz.

[34] A. Strzeboński and E. P. Tsigaridas. Univariate real root isolation in an extension field. In A. Leykin, editor, *Proc. 36th ACM Int'l Symp. on Symbolic & Algebraic Comp. (ISSAC)*, pages 321–328, San Jose, CA, USA, June 2011. ACM.

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

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

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