A Deterministic Descartes Algorithm for Real Polynomials

Kurt Mehlhorn

Max Planck Institute for Informatics

Michael Sagraloff

Max Planck Institute for Informatics

Abstract

We describe a Descartes algorithm for root isolation of polynomials with real coefficients. It is assumed that the coefficients of the polynomial can be approximated with arbitrary precision; exact computation in the field of coefficients is not required. We refer to such coefficients as bitstream coefficients. The algorithm is deterministic and has almost the same asymptotic complexity as the randomized bitstream-Descartes algorithm of Eigenwillig et al. (2005). Besides being deterministic, the algorithm is also somewhat simpler to analyze.

Key words: real polynomial, root isolation, Descartes' rule of sign, bitstream coefficients

1. Introduction

The isolation of the real roots of a real polynomial is a fundamental task in computer algebra and numerical analysis: given a polynomial p, compute for each of its real roots an interval with rational endpoints containing it and being disjoint from the intervals computed for the other roots. There are many methods for isolating the real roots of a real polynomial. One of the best approaches to root isolation is the Descartes method. It is a bisection method based on Descartes' Rule of Signs to test for roots. Its modern form goes back to Collins and Akritas (1976), see also Basu et al. (2006). It can be formulated to operate on polynomials given in the usual power basis or in the Bernstein basis and works over any ring of coefficients in which addition and sign test (with result $+$, 0, and $-$) are computable.

⋆ The work of both authors was partially supported by the IST Programme of the EU as Shared-cost RTD (FET Open) Project under Contract No IST-006413 (ACS - Algorithms for Complex Shapes).

Email addresses: mehlhorn@mpi-inf.mpg.de (Kurt Mehlhorn), msagralo@mpi-inf.mpg.de (Michael Sagraloff).

For integer coefficients, it is currently one of the most efficient methods (Rouillier and Zimmermann, 2004). We review it in Section 3 and give further references to related and previous work along the way. For algebraic coefficients, the cost of exact arithmetic may render the method useless; hence it was suggested to replace the coefficients by small intervals and to execute the method using interval arithmetic. The first proposals (Johnson and Krandick, 1997; Collins et al., 2002; Mourrain et al., 2004; Rouillier and Zimmermann, 2004) were incomplete; they all had to resort to exact arithmetic in the ring of coefficients for some input polynomials. In (Eigenwillig et al., 2005; Eigenwillig, 2008) it was shown that randomization leads to a complete algorithm with no need for exact arithmetic. The only requirement is that coefficients can be approximated to any specified error bound. Following Eigenwillig et al. (2005); Eigenwillig (2008), we call such coefficients bitstream coefficients. In (Eigenwillig et al., 2005; Eigenwillig, 2008), the following result was shown: *To isolate the real roots of a square-free real polynomial $p(x) = p_n x^n + \ldots + p_0$ with root separation (= the minimal distance between any two roots) σ, coefficients $|p_n| \geq 1$ and $|p_i| < 2^\tau$, the algorithm needs coefficient approximations to $O(n(\log(1/\sigma)+\tau))$ bits after the binary point and has an expected cost of $O(n^4(\log(1/\sigma)+\tau)^2)$ bit operations.* The cost statement ignores the cost of computing the approximations of the coefficients with the required quality. The algorithm is readily derandomized, but this increases the running time by a factor of n.

We describe a deterministic algorithm with running time $O(n^4(\log n + \log(1/\sigma) + \tau)^2)$; up to the $\log n$ term, this is the same as for the randomized algorithms of Eigenwillig et al. (2005); Eigenwillig (2008). The precision requirement is the same as for the randomized algorithm. Besides being deterministic, the algorithm is also more intuitive and somewhat simpler to analyze. Moreover, it works directly over the monomial basis and there is no need for conversion to the Bernstein basis.

The roots of a polynomial depend continuously on its coefficients. The algorithms of Eigenwillig et al. (2005); Eigenwillig (2008) use this fact only indirectly; our new algorithm uses this fact directly. It constructs a rational polynomial p^* from the input polynomial p by approximating the coefficients to some carefully chosen precision ε. It then runs a variant of Descartes algorithm on p^* and determines isolating intervals for the roots of p^*. Finally, it returns suitably enlarged intervals as isolating intervals for the roots of p.

This paper is organized as follows. In Section 2 we discuss related work and in Section 3 we review Descartes method. Section 4 discusses the extension to real polynomials with bitstream coefficients. Finally, Section 5 deals with a partial extension to polynomials with multiple roots.

2. Related Work

Root isolation is a fundamental problem in computer algebra and numerical analysis. For a survey, we refer the reader to (Pan, 1997). On a top level, there are two kinds of algorithms. Algorithms that always solve the task and algorithms that solve the task if some additional information is available, e.g., approximations of the roots. Aberth's method (Aberth, 1988; Bini and Fiorentino, 2000) is a representative of the second class with excellent practical behavior. The origin of the first class of algorithms dates back to Descartes, Sturm, Bundan, Fourier, and Vincent. For modern accounts, see (Collins and Akritas, 1976), (Yap, 1999, Chapter 7), and (Tsigaridas and Emiris, 2008). There are asymptotically faster algorithms available (Pan, 2002; Schönhage, 1982). However, the asymptotically faster algorithms are quite involved and no implementation was attempted yet.

The bisection and continued fraction algorithms based on Descartes' rule of sign work well for polynomials with integer coefficients. However, for polynomials with nonrational coefficients, the high cost of arithmetic makes the approaches less attractive. It was therefore suggested (Johnson and Krandick, 1997; Collins et al., 2002; Mourrain et al., 2004; Rouillier and Zimmermann, 2004) to approximate the coefficients by intervals and to use interval arithmetic instead of real arithmetic. This led to Descartes solvers for polynomials with nonrational coefficients or long integer coefficients with improved efficiency. However, all methods mentioned above have to resort to exact arithmetic for some inputs, namely for inputs for which certain decisions (counting sign changes in a sequence of coefficients and determining the sign of the polynomial at subdivision points) could not made reliably with interval arithmetic. The first Descartes algorithm that is guaranteed to work with approximate arithmetic was presented in (Eigenwillig et al., 2005; Eigenwillig, 2008). Eigenwillig et al. termed their method bitstream-Descartes algorithm. It uses randomization to overcome the problems mentioned above. The choice of random subdivision points guarantees that the polynomial is "sufficiently large" at subdivision points and that sign changes in coefficient sequences can be counted with sufficient reliability. Kerber et. al. (Kerber, 2006; Eigenwillig et al., 2007) introduce a partial extension to polynomials with multiple roots. The variant works for polynomials with exactly one multiple root; it requires the number m of distinct real roots and the value $k = \deg\gcd(p, p')$ as additional inputs. The bitstream-Descartes algorithm and its extension will become part of the algebraic kernel of CGAL (CGAL, 2008) and are key ingredients for the topology computation of algebraic plane curves (Eigenwillig et al., 2007) and algebraic surfaces in space (Berberich et al., 2008).

In comparison to the randomized bitstream-Descartes method, our new algorithm is deterministic, conceptually simpler, simpler to analyze, and has almost the same running time. The gain in simplicity stems from the fact that the algorithm runs on a concrete approximation of the input polynomial and not on an interval polynomial that represents all possible approximations of the input polynomial with a certain precision.

3. Preliminaries

For a real root z of p, let $\sigma(z,p)$ be the minimal distance of z to another root of p. For a nonreal root z of p, let $\sigma(z,p)$ be the absolute value of its imaginary coordinate. Let $\sigma(p)$ be the minimal value of $\sigma(z,p)$ over all roots of p. For an interval $I = (a,b)$, let $w(I) := b - a$ be its length or width.

Algorithm 1 shows the Descartes algorithm for isolating the roots of a real polynomial p in an open interval I_0; see (Basu et al., 2006; Eigenwillig, 2008) for extensive treatments and references. The algorithm requires that the real roots of p in I_0 are simple. If the requirement is not met, the algorithm diverges. It maintains a set A of active intervals. Initially, A contains I_0, and the algorithm stops as soon as A is empty. In each iteration, some interval $I \in A$ is processed. The action taken depends on the integer $var(p,I)$, the outcome of Descartes' rule of signs applied to p and I.

Descartes' rule of signs states that for a real polynomial $q(x) = \sum_{0 \le i \le n} q_i x^n$, the number of sign changes in the coefficient sequence of q, i.e., the number of pairs (i,j) with $i < j$, $q_i q_j < 0$, and $q_{i+1} = \ldots = q_{j-1} = 0$, is no smaller than and of the same parity as the number of positive real roots of q. Let $var(q)$ denote the number of sign changes in the coefficient sequence of q. If $var(q) = 0$, q has no positive real root, and if $var(q) = 1$, q has exactly one positive real root. The rule is easily extended to arbitrary open intervals by a suitable coordinate transformation. Let

Algorithm 1 Descartes Algorithm for Isolating Real Roots

Require: $p = \sum_{0 \le i \le n} p_i x^i$ is a real polynomial and I_0 is an open interval. The real roots of p in I_0 are simple.

Ensure: returns a list O of isolating intervals for the real roots of p in I.

```
A := { I_0 }                                   {list of active intervals}
O := ∅                                         {list of isolating intervals}
repeat
  I:= some interval in A; delete I from A;
  if var(p,I) = 0 do nothing;
  if var(p,I) = 1 add I to O;
  if var(p,I) ≥ 2 then
    let I = (a,b) and set m := (a+b)/2;
    if p(m) = 0 add [m,m] to O;
    add (a,m) and (m,b) to A;
  end if
until A is empty
return O
```

$I = (a,b)$ be an arbitrary open interval. The mapping $x \mapsto (ax+b)/(x+1)$ maps $(0,\infty)$ bijectively onto (a,b) and hence the positive real roots of

$$q_I(x) := (1+x)^n \cdot p\left(\frac{ax+b}{x+1}\right)$$

correspond bijectively to the real roots of p in I. We define $var(p,I)$ as $var(q_I)$. The factor $(1+x)^n$ in the definition of q_I clears denominators and guarantees that q_I is a polynomial.

Having defined $var(p,I)$, we continue our explanation of the algorithm. If there is no sign change, I contains no root of p and we discard it. If there is exactly one sign change, I contains exactly one root of p and hence is an isolating interval for it. We add I to the list O of isolating intervals. If there is more than one sign change, we divide I at its midpoint and add the subintervals to the set of active intervals. If the midpoint m is a zero of p, we add the trivial interval $[m,m]$ to the list of isolating intervals.

Correctness of the algorithm is obvious. Termination and complexity analysis of Descartes algorithm rest on the following theorem, see also Figure 1.

Theorem 1 (Obreschkoff (1963); Obrechkoff (2003))**.** Let p be a polynomial of degree n, I an open interval, and $v = var(p,I)$. If the Obreshkoff lens L_{n-q} (see Figure 1) contains at least q roots (counted with multiplicity) of p, then $v \ge q$. If the Obreshkoff area A_q (see Figure 1) contains at most q roots (counted with multiplicity) of p, then $v \le q$. In particular,

$$\#\text{ of roots of } p \text{ in } L_n \le var(p,I) \le \#\text{ of roots of } p \text{ in } A_n.$$

Theorem 2 (Obreschkoff (1925); Ostrowski (1950))**.** Consider a real polynomial $p(x)$ and an interval $I = (a,b)$ with midpoint $m_I = (a+b)/2$ and let $v = var(p,I)$.

- (One-Circle Theorem) If the open disc bounded by the circle C_0 centered at m_I and passing through the endpoints of I contains no root of $p(x)$, then $v = 0$.
- (Two-Circle Theorem) If the union of the open discs bounded by the circles $\underline{C}_1$ and $\overline{C}_1$ centered at $m_I \pm i(1/(2\sqrt{3}))w(I)$ and passing through the endpoints of I contains precisely one root of $p(x)$, then $v = 1$.

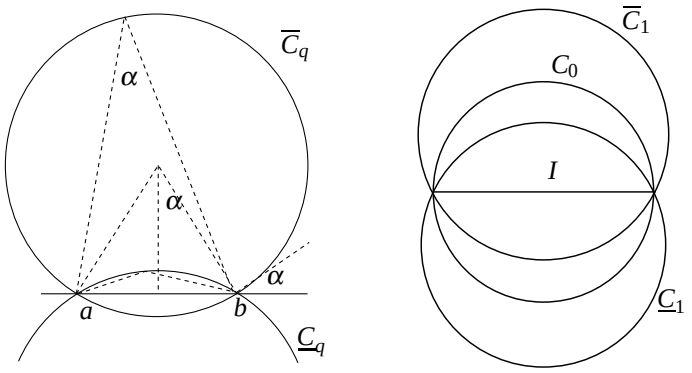

Fig. 1. For any q with $0 \le q \le n$, the Obreshkoff disks $\overline{C}_q$ and $\underline{C}_q$ for I have the endpoints of I on their boundary; their centers see the line segment $[a,b]$ under the angle $2\alpha = 2\pi/(q+2)$. The Obreshkoff lens L_q is the interior of $\overline{C}_q \cap \underline{C}_q$ and the Obreshkoff area A_q is the interior of $\overline{C}_q \cup \underline{C}_q$. Any point (except for a and b) on the boundary of A_q sees $[a,b]$ under an angle $\pi/(q+2)$ (= half the angle at the center) and any point (except for a and b) on the boundary of L_q sees $[a,b]$ under angle $\pi - \pi/(q+2)$ (= half the complementary angle at the center). We have $L_n \subset L_{n-1} \subset \ldots \subset L_1 \subset L_0$ and $A_0 \subset A_1 \subset \ldots \subset A_{n-1} \subset A_n$. The circles $\overline{C}_0$ and $\underline{C}_0$ coincide. They have their center at the midpoint of I. The circles $\overline{C}_1$ and $\underline{C}_1$ are the circumcircles of the two equilateral triangles having I as one of their edges. We call A_1 the *two-circle region* of I.

The one-circle and two-circle theorems are special cases of Theorem 1, namely: if A_0 contains no root of p, then $var(p,I) = 0$ and if A_1 contains exactly one root of p, then $var(p,I) = 1$. Proofs of the one- and two-circle theorems can be found in (Obreschkoff, 1925, 1963; Obrechkoff, 2003; Ostrowski, 1950; Krandick and Mehlhorn, 2006; Eigenwillig, 2008). We also need the property that $var(p,I)$ is subadditive, i.e., $var(p,I_1) + var(p,I_2) \le var(p,I)$ whenever I_1 and I_2 are disjoint subintervals of I. For a simple self-contained proof, we refer the reader to (Eigenwillig, 2008, Corollary 2.27).

Theorem 3 (Schoenberg (1934))**.** Let p be a real polynomial. If the pairwise disjoint open intervals $J_1, \ldots, J_l$ are subsets of the open interval I, then

$$var(p,I) \ge \sum_{1 \le i \le \ell} var(p,J_i).$$

Theorem 2 implies that no interval I of length $\sigma(p)$ or less is split. Such an interval, recall that is is open, cannot contain two real roots and its two-circle region cannot contain any nonreal root. Thus $var(p,I) \le 1$ by Theorem 2. We conclude that the depth of the recursion tree is bounded by $\log w(I_0)/\sigma(p)$. The number of internal nodes in the recursion tree is bounded by n times the depth. This follows from $var(p,I_1) + var(p,I_2) \le var(p,I)$, where I_1 and I_2 are the two subintervals of I. Thus there cannot be more than $n/2$ intervals I with $var(p,I) \ge 2$ at any level of the recursion.

The computation of q_I from p at every node of the recursion is costly. It is better to store with every interval $I = (a,b)$ the polynomial $p_I(x) := p(a + x(b-a))$, whose roots in $(0,1)$ correspond to the roots of p in I. If I is split at $m = (a+b)/2$ into $I_\ell = (a,m)$ and $I_r = (m,b)$, the polynomials associated with the subintervals are

$$p_{I_\ell}(x) = 2^n p_I(x/2) \quad \text{and} \quad p_{I_r}(x) = 2^n p_I((1+x)/2) = p_{I_\ell}(1+x)\ .$$

Also, $q_I(x) = (1+x)^n p_I(1/(1+x))$. The polynomials p_{I_ℓ}, p_{I_r}, and q_I can be obtained from p_I by n^2 additions. Also, if the coefficients are integral, the coefficients grow by $O(n)$ bits in every node.

4. A Descartes Algorithm for Polynomials with Bitstream Coefficients

We will extend the Descartes algorithm to polynomials with real coefficients. It is assumed that the coefficients can be approximated to any precision. We generalize in two steps. In Section 4.1, we assume that a lower bound for the root separation of the polynomial is known. We remove this assumption in Section 4.2. We make use of a result of Schönhage that bounds the change of roots in terms of a change of coefficients. For a polynomial $p = \sum_{0\le i\le n} p_i x^i$, $|p| = \sum_{0\le i\le n} |p_i|$ denotes the maximum norm of p.

Theorem 4 (Schönhage (1985))**.** Let $p = \sum_{0\le i\le n} p_i x^i = p_n \prod_{1\le i\le n}(x - z_i)$ be a polynomial of degree n with $|z_i| < 1$ for all i. Let μ be a positive real with $\mu \le 2^{-7n}$ and let $p^*(x) = \sum_{0\le i\le n} p_i^* x^i = p_n^* \prod_{1\le i\le n}(x - z_i^*)$ be such that

$$|p - p^*| < \mu |p| \ .$$

Then up to a permutation of the indices of the z_i^*

$$|z_i^* - z_i| < 9\sqrt[n]{\mu} \ .$$

4.1. Known Root Separation

The idea of our algorithm is now as follows. In order to isolate the roots of a polynomial $P = \sum_{0\le i\le n} P_i x^i$, we first determine an interval I_0 containing all real roots of P. It is well-known that the modulus of any root of P is bounded by

$$B := 2\max\left\{ \frac{|P_i|}{|P_n|}; 0 \le i < n \right\} \ .$$

Let $p(x) := P(4B(x - 1/2))$. Then all roots of p are contained in the disc of radius 1/4 centered at $1/2 + 0i$. Isolating the real roots of P is equivalent to isolating the real roots of p. We achieve the latter by isolating the real roots of p^* in $(0,1)$ for a suitable approximation p^* of p. i.e., $|p^* - p| < \mu|p|$ for a suitable μ. Then corresponding roots of p^* and p are less than $9\sqrt[n]{\mu}$ apart. What properties should μ have?

- $9\sqrt[n]{\mu} < 1/4$. This guarantees that the real roots of p^* are in $(0,1)$. The condition is certainly satisfied for $\mu \le 2^{-7n}$.
- $9\sqrt[n]{\mu} \le \sigma(p)/12$; the choice of constant 12 will become clear below. This guarantees that real roots of p correspond to real roots of p^* and that nonreal roots of p correspond to nonreal roots of p^*. Moreover, $\sigma(p^*) \ge \sigma(p) - 18\sqrt[n]{\mu} \ge (5/6)\sigma(p) \ge 90\sqrt[n]{\mu}$.

We choose μ such that $9\sqrt[n]{\mu} \le \min(\sigma(p)/12, 1/8)$.

We call a binary fraction a an *ε-approximation* of a real number x if $|a - x| \le \varepsilon$. With $L = \lceil \log 1/\varepsilon \rceil$, we call $\lceil x2^L \rceil 2^{-L}$ and $\lfloor x2^L \rfloor 2^{-L}$ *ε-approximates* of x. Of course, an ε-approximate is an ε-approximation. Moreover, if $\varepsilon < 1$, an ε-approximate has only L bits after the binary point and if $\varepsilon > 1$, an ε-approximate is an integer whose binary representation ends with L zeros. An ε-approximate of a polynomial p is a polynomial p^* such that each coefficient[1] of

[1] Of course, if $\varepsilon > 1$, the last L bits of every coefficient of p^* can be dropped.

Algorithm 2 Descartes Algorithm for a Real Polynomial p with Root Separation Estimate σ

Require: p is a real polynomial with roots in the disc of radius 1/4 centered at $1/2+0i$, $\sigma \leq \sigma(p)$, $\mu = \min(2^{-7n}, (\sigma/108)^n)$, $\varepsilon \leq \mu/(n+2)|p|$, p^* is an ε-approximate of p

Ensure: returns a list O^* of well-separated isolating intervals for the real roots of p^*.

$A := \{(0,1)\}$ {list of active intervals}
$O^* := \emptyset$ {list of isolating intervals}
repeat
  $I :=$ some interval in A; delete I from A;
  $I^+ = (a-2(b-a), b+2(b-a))$, where $I = (a,b)$;
  if $var(p^*, I^+) > 1$ and $var(p^*, I) \geq 1$ **then**
    add (a,m) and (m,b) to A where $m = (a+b)/2$;
    if $p^*(m) = 0$ add $[m,m]$ to O^*;
  else
    if $var(p^*, I) = 0$ do nothing;
    if $var(p^*, I) = 1$ add I to O^*;
  end if
until A is empty
return O^*

p^* is an ε-approximate of the corresponding coefficient of p. If p^* is an ε-approximate of p, $|p^* - p| \leq (n+1)\varepsilon$. We let $\varepsilon \leq \mu|p|/(n+2)$. Then $|p^* - p| < \mu|p|$ and Theorem 4 applies.

For an interval I, let $\tilde{I}$ be its expansion by $9\sqrt[n]{\mu}$ on both sides, i.e, if $I = (a,b)$, then $\tilde{I} = (a - 9\sqrt[n]{\mu}, b + 9\sqrt[n]{\mu})$ and if $I = [m,m]$, then $\tilde{I} = (m - 9\sqrt[n]{\mu}, m + 9\sqrt[n]{\mu})$. If I is an isolating interval for a real root of p^*, $\tilde{I}$ contains the corresponding root of p. Our goal is to compute isolating intervals for the roots of p^* such that their expansions are pairwise disjoint. The *expanded intervals* are then isolating intervals for the roots of p.

A simple modification of Algorithm 1 computes sufficiently separated isolating intervals for the roots of p^*. We simply subdivide an interval as long as $var(p^*, I^+) \geq 2$ and $var(p^*, I) \geq 1$, where I^+ is the interval of length $5w(I)$ enlarging I by $2w(I)$ on either side[2], i.e., if $I = (a,b)$, $I^+ = (a-2(b-a), b+2(b-a))$. We call I^+ the *extension* of I or an *extended interval*. We subdivide an interval I if $var(p^*, I^+) > 1$ and $var(p^*, I) \geq 1$. If I is not split, we have $var(p^*, I) \leq var(p^*, I^+) \leq 1$. We obtain Algorithm 2.

Lemma 5. Algorithm 2 splits only intervals with length greater than $\sigma(p^*)/5$.

Proof. Consider any interval I with $w(I) \leq \sigma(p^*)/5$. Then $w(I^+) \leq \sigma(p^*)$ and hence either the one- or the two-circle theorem applies to I^+. Thus $var(p^*, I^+) \leq 1$ and I is not split. □

Let O^* be the list of isolating intervals computed for p^*. Any interval in O^* is either a singleton or has length at least $\sigma(p^*)/10$. The expansions of the intervals in O^* are isolating intervals for p. Let

$$O := \left\{ \tilde{I};\, I \in O^* \right\} .$$

Lemma 6. O is a set of isolating intervals for p.

[2] We have no particular reason for enlarging by $2w(I)$. Enlarging by $\ell w(I)$ for any fixed $\ell \in \mathbb{N}$ would also work. We have not tried to optimize the choice of ℓ.

Proof. By our choice of μ, p and p^* have the same number of real roots and each expanded interval contains a real root of p. We need to argue disjointness.

Let I and J be two intervals in O^*. If I and J are singletons, they have distance at least $\sigma(p^*)$ from each other. Our choice of μ certainly guarantees $\sigma(p^*) \geq 18\sqrt[n]{\mu}$ and hence disjointness is preserved after expanding both intervals.

So assume, that at least one of the intervals is not a singleton, say I. We may also assume $w(I) \geq w(J)$. Since I and J are in O^*, both contain a real root of p^*. If I^+ would contain J, it would contain two real roots, and we would have $var(p^*, I^+) \geq 2$. So I would be split. Thus I^+ does not contain J and hence is disjoint from J (since $w(I) \geq w(J)$). Thus the distance of I and J is at least $2w(I)$. Also,

$$2w(I) \geq \frac{\sigma(p^*)}{5} \geq \frac{\sigma(p) - 18\sqrt[n]{\mu}}{5} \geq 18\sqrt[n]{\mu}$$

by our choice of μ and hence $\tilde{I}$ and $\tilde{J}$ are disjoint. □

We next turn to the complexity analysis. We first bound the size of the tree generated by Algorithm 2 and then bound the bit complexity.

Theorem 7. Let T be the tree generated by Algorithm 2. Then

$$|T| = \mathrm{O}\left(n + \sum_{z \text{ is a root of } p} \log \frac{1}{\sigma(p,z)}\right),$$

where $|T|$ denotes the number of nodes of T.

Proof. The argument is a minor modification of an argument in (Eigenwillig et al., 2006). The nodes of T at depth d correspond to intervals of length 2^{-d}. We use I_v to denote the interval corresponding to node v. If v is an internal node, $var(p^*, I_v^+) > 1$ and $var(p^*, I_v) \geq 1$. For a root $z = x + iy$ of p^* and depth d, the *canonical interval for z at depth d* is such that $x \in [k2^{-d}, (k+1)2^{-d})$. Then $k = \lfloor x2^d \rfloor$. We call a node v of T *canonical* if I_v is canonical for one of the roots contained in the two-circle figure of I_v^+. If v is canonical, the parent of v is too. The canonical subtree T_c of T consists of all internal canonical nodes. In order to bound the size of T, we show first that $|T| = \mathrm{O}(|T_c|)$ and then estimate the size of the canonical subtree. Since T is a binary tree, it suffices to estimate the number of internal nodes.

We define a mapping from internal nodes to canonical internal nodes. Let I be any internal node. If I contains a real root, I is canonical and we map I to itself. So assume that I contains no real root. Then $var(p^*, I) \geq 2$, since $var(p^*, I) = 1$ implies the existence of a real root contained in I, and hence the two-circle figure of I contains a pair of complex roots $x \pm iy$. Let K be the interval of length $w(I)$ that is canonical for $x \pm iy$. Since the projection of the two-circle figure onto the real axis is contained in an interval of length $(\sqrt{3}/2)w(I)$, K is either equal to I or adjacent to I. The roots $x \pm iy$ are clearly contained in the two-circle figure of K and hence $var(p^*, K) > 1$. We map I to K. At most three intervals are mapped to K in this way.

We conclude that the number of internal nodes of the Descartes tree is at most 3 times the number of internal nodes of the canonical subtree.

We will next estimate the size of the canonical subtree. Consider a leaf v of the canonical subtree and let z be a root of p^* corresponding to this leaf. If there are several, z is the root with minimal value of $\sigma(p^*, z)$. Since the canonical subtree consists only of internal nodes of the Descartes tree, we have $var(p^*, I_v^+) > 1$ and hence $\sigma(p^*, z) \leq w(I_v^+) \leq 5w(I_v)$. The depth d_v is

equal to $\log 1/w(I_v)$. Thus $d_v \le \log 5 + \log 1/\sigma(p^*,z)$. Since any root of p^* is associated with at most one leaf of the canonical tree, we conclude

$$|T_c| = \mathrm{O}\left(n + \sum_{z \text{ is a root of } p^*} \log \frac{1}{\sigma(p^*,z)}\right) = \mathrm{O}\left(n + \sum_{z \text{ is a root of } p} \log \frac{1}{\sigma(p,z)}\right) ,$$

where the last equality follows from the fact that corresponding roots of p and p^* have distance at most $9\sqrt[n]{\mu}$ from each other and that $9\sqrt[n]{\mu} \le \sigma(p)/12$. □

The bound on the tree size readily translates into a bound on the bit complexity of Algorithm 2.

Theorem 8. Let q be be a polynomial with $|q_n| \ge 1$, and $|q_i| \le 2^{\tau-1}$ for all i. If a quantity σ with $\sqrt{\sigma(q)} \le \sigma \le \sigma(q)$ is known, the bit complexity of isolating the real roots of q is

$$\mathrm{O}(n^4(\tau + \log(1/\sigma(q)))^2) .$$

The coefficients of q need to be approximated with $\mathrm{O}(n(\tau + \log 1/\sigma(q)))$ bits after the binary point.

Proof. All roots of q have modulus at most $B := 2^\tau$. Hence the polynomial $p(x) = q(4Bx - 2B)$ has its real roots in $(1/4, 3/4)$. The root separations are scaled by $4B$ and the coefficients of p have $\mathrm{O}(n\tau)$ bits before the binary point. Also, $|p| \ge 1$. We set $\mu := \min((\sigma/(4B))/108, 2^{-7})^n \le \min(\sigma(p)/108, 2^{-7})^n$ and $\varepsilon := \mu/(n+2) \le \mu|p|/(n+2)$. Then

$$\log(1/\varepsilon) = \mathrm{O}(n(\tau + \log 1/\sigma(q))) .$$

So the coefficients of p^* have length $\mathrm{O}(n(\tau + \log 1/\sigma(q)))$. The size of the Descartes tree is

$$\mathrm{O}\left(n + \sum_{z \text{ is a root of } p} \log \frac{1}{\sigma(p,z)}\right) = \mathrm{O}\left(n + n\log \frac{B}{\sigma}\right) = \mathrm{O}(n(\tau + \log 1/\sigma(q))) ,$$

and its depth is bounded by $\mathrm{O}(\tau + \log(1/\sigma(q)))$. In each node of the tree, n^2 additions are performed and the coefficient size grows by n. So the total growth in coefficient length is $n(\tau + \log 1/\sigma(q))$, and hence the coefficient length is $\mathrm{O}(n(\tau + \log 1/\sigma(q)))$ at all nodes. We conclude that the bit complexity is

$$\mathrm{O}(|T| \cdot n^2 \cdot n(\tau + \log 1/\sigma(q))) = \mathrm{O}(n^4(\tau + \log(1/\sigma(q)))^2) .$$

□

4.2. The Case of Unknown Separation

Algorithm 2 works fine as long as $9\sqrt[n]{\mu} \le \min(1/8, \sigma(p)/12)$. *How can we ensure this condition (or a similar condition) or how can we learn that it is not satisfied*?

Is the smallest length of an interval generated by the subdivision algorithm a good indicator of $\sigma(p)$? For the standard Descartes algorithm (Algorithm 1) this is not the case, as the following example shows. Let $p(x) = (4x - 4x + 1 + \delta^2) = (2x - 1 - i\delta)(2x - 1 - i\delta)$ with $\delta \approx 0$. This polynomial has a pair of conjugate complex roots at $1/2 \pm i\delta/2$ and hence separation $\delta/2$. However, $var(p,(0,1)) = 2$ and $var(p,(0,1/2)) = var(p,(1/2,1)) = 0$. Thus the algorithm ends with intervals of length $1/2$, although the separation may be arbitrarily small. The situation is different for Algorithm 2. For it, the length of the shortest interval produced by the algorithm is a good indicator of the separation.

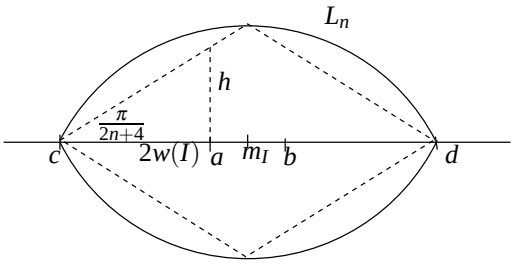

Fig. 2. $I=(a,b)$, $I^+=(c,d)$, and L_n denotes the Obreshkoff lens $L_n(I^+)$. The height of L_n at endpoint of I is at least h, where $h=2w(I)\tan(\pi/(2(n+2)))\geq 2w(I)/(n+2)\geq w(I)/n$. By Theorem 1, $var(p^*,I^+)$ is at least the number of roots of p^* in the rectangle $I\times[-hi,+hi]$.

Lemma 9. Algorithm 2 splits no interval of length $\sigma(p^*)/5$ or less and refines at least one interval to a length less than $n\sigma(p^*)$.

Proof. We have already shown the first part. For the second part, we distinguish cases. If $\sigma(p^*)$ is equal to the distance of two real roots, let I be the separating interval computed for one of them. Then $w(I)\leq\sigma(p^*)/2$ because otherwise I^+ would contain both roots and I would be split.

If $\sigma(p^*)$ is equal to the imaginary coordinate of a nonreal root, let I be the canonical leaf for this root. Then $var(p^*,I^+)\leq 1$ and hence $\sigma(p^*)\geq w(I)/n$, see Figure 2. □

Assume now that we run Algorithm 2 with some value of μ that is known to satisfy $\mu\leq 2^{-7n}$, but not known to satisfy $\mu\leq(\sigma(p)/108)^n$, and the algorithm does not generate an interval of length less than $120n\sqrt[n]{\mu}$. Then $n\sigma(p^*)>120n\sqrt[n]{\mu}$ by Lemma 9 and hence $\sigma(p^*)>120\sqrt[n]{\mu}$ and $\sigma(p)>\sigma(p^*)-18\sqrt[n]{\mu}>108\sqrt[n]{\mu}$. In other words, if the algorithm does not generate an interval of length less than $120n\sqrt[n]{\mu}$, we have a proof that the precondition for μ was satisfied. Actually, a more refined argument shows that we can use the threshold $L_0=18n\sqrt[n]{\mu}$. Let us call an interval *short* if its length is less than L_0 and *long* otherwise.

Lemma 10. If Algorithm 2 generates no short interval, O is a set of isolating intervals for the real roots of p.

Proof. Since no short interval is produced, we have $n\sigma(p^*)>18n\sqrt[n]{\mu}$ and hence $\sigma(p^*)>18\sqrt[n]{\mu}$. Since corresponding roots of p and p^* are less than $9\sqrt[n]{\mu}$ apart, p and p^* have the same number of real roots. Also, each interval in O contains a real root of p. It remains to argue disjointness of the expanded intervals.

Let I and J be two intervals in O^*. If I and J are singletons, they have distance at least $\sigma(p^*)$ from each other. By the above, $\sigma(p^*)>18\sqrt[n]{\mu}$.

So assume, that at least one of the intervals is not a singleton, say I. We may also assume $w(I)\geq w(J)$. Since I and J are in O^*, both contain a real root of p^*. If I^+ would contain J, it would contain two real roots, and we would have $var(p^*,I^+)\geq 2$. So I would be split. Thus I^+ does not contain J and hence is disjoint from J (since $w(I)\geq w(J)$). Thus the distance of I and J is at least $2w(I)$. Also, I is long and hence $w(I)\geq 18\sqrt[n]{\mu}$. □

Algorithm 3 shows the Descartes algorithm for a real polynomial p with bitstream coefficients. It embeds Algorithm 2 into a loop that determines an appropriate value of μ. We initialize μ to 2^{-7n}. In any iteration, we run Algorithm 2 on an ε-approximate p^* of p, where $\varepsilon\leq\mu|p|/(n+2)$, and start interval $(0,1)$. If a short interval is produced, we square μ and repeat. Otherwise, we return the expanded versions of the isolating intervals of p^*. How small can μ become?

Algorithm 3 Descartes Algorithm for Real Polynomials

Require: $p = \sum_{0 \le i \le n} p_i x^i$ and all roots of p lie in a disc of radius 1/4 centered at $1/2 + 0i$. Real roots are distinct.

Ensure: returns isolating intervals for the real roots of p.

```
μ = 2^{-7n};
while (true) do
  choose ε ≤ μ|p|/(n+2) and let p* be an ε-approximate of p;
  run Algorithm 2 on p* and start interval I = (0,1); //we do not guarantee μ ≤ (σ(p)/108)^n
  if the algorithm does not produce a short interval then
    exit from the loop;
  else
    μ = μ^2;
  end if
end while
return O := { Ĩ; I ∈ O* }
```

Lemma 11. Algorithm 3 stops with

$$\mu \ge \min\left(2^{-7n}, \left(\frac{\sigma(p)}{100n}\right)^{2n}\right) .$$

Proof. If the algorithm stops in the first iteration, it stops with $\mu = 2^{-7n}$. So assume that the algorithm performs more than one iteration. By Lemma 9, no interval of length less than $\sigma(p^*)/5$ is split. Thus, if an iteration with a particular value of μ is not the last, we must have $18n\sqrt[n]{\mu} \ge \sigma(p^*)/5$ or $\mu \ge (\sigma(p^*)/(90n))^n$.

Consider now any iteration with $\mu < (\sigma(p)/(100n))^n \le (\sigma(p)/200)^n$. Then $\sigma(p^*) \ge \sigma(p) - 18\sqrt[n]{\mu} \ge (9/10)\sigma(p)$ and hence $\mu < (\sigma(p^*)/(90n))^n$. Thus the iteration must be the last and therefore $\mu \ge (\sigma(p)/(100n))^n$ in the next to last iteration. Since μ is squared from one iteration to the next, we have $\mu \ge (\sigma(p)/(100n))^{2n}$ in the last iteration. □

In order to determine the bit complexity of Algorithm 3 we follow the same approach as for Algorithm 2, that is, we will first estimate the size of the canonical subtree (for the definition see the proof of Theorem 7). Consider a leaf v of the canonical subtree and let z be a root of p^* corresponding to this leaf. If there are several, let z be the one with minimal value of $\sigma(p^*, z)$. The leaf v is either *regular*, that is, the extended intervals of its children exhibit at most one sign variation, or *forced*, that is, we would get a short interval if we further subdivide v.

Consider a regular leaf first. Then v is an internal node of the Descartes tree, $var(p^*, I_v^+) > 1$ and therefore $\sigma(p^*, z) \le w(I_v^+) \le 5w(I_v)$. The depth d_v is equal to $\log(1/w(I_v))$ and $w(I_v) \ge L_0$. Thus $d_v \le \log 5 + \log 1/\max(L_0, \sigma(p^*, z))$. Next consider a forced leaf. Its interval has length at most $2L_0$ and thus $\sigma(p^*, z) \le w(I_v^+) \le 5w(I_v) \le 10L_0$. The depth d_v is equal to $\log 1/L_0$, and hence $d_v \le \log 10 + \log 1/\max(L_0, \sigma(p^*, z))$. Since any root of p^* is associated with at most one leaf of the canonical tree, we conclude

$$|T_c| = \mathrm{O}\left(n + \sum_{z \text{ is a root of } p^*} \log \frac{1}{\max(L_0, \sigma(p^*, z))}\right) .$$

We next argue that we may replace p^* by p in the expression above. Fix a correspondence between the roots z_i^* of p^* and the roots z_i of p such that $|z_i^* - z_i| \leq 9\sqrt[n]{\mu}$. Then $\sigma(p^*, z_i^*) > L_0$ implies $\sigma(p, z_i) \leq 2\sigma(p^*, z_i^*)$ and $\sigma(p^*, z_i^*) < L_0$ implies $\sigma(p, z_i) \leq 2L_0$. Thus we may replace p^* by p in the bound above. The following Lemma summarizes the discussion.

Lemma 12. In any iteration, the size of the Descartes tree of Algorithm 3 is

$$\mathrm{O}\left(n + \sum_{z \text{ is a root of } p} \log \frac{1}{\max(L_0, \sigma(p,z))}\right) ,$$

where $L_0 = 18n\sqrt[n]{\mu}$.

Now again, let q be a polynomial with $|q_n| \geq 1$ and $|q_i| \leq 2^{\tau-1}$ for all i. All roots of q have modulus at most $B := 2^\tau$. So $p(x) = q(4Bx - 2B)$ has its real roots in $(1/4, 3/4)$. Also root separations are scaled by $4B$ and the coefficients of p have $\mathrm{O}(n\tau)$ bits before the binary point.

We start with $\mu = 2^{-7n}$, μ is squared from one iteration to the next, and we terminate with $\mu \geq \min(2^{-7n}, (\sigma(p)/(100n))^{2n})$ (see Lemma 11).

In an iteration with a particular value of μ, the coefficients of p^* have length $\mathrm{O}(n\tau + \log 1/\mu)$. The depth of the tree is bounded by $\log 1/L_0 = \mathrm{O}((\log 1/\mu)/n)$. We use Lemma 12 to bound its size by

$$\mathrm{O}\left(n + n\log \frac{1}{\max(18n\sqrt[n]{\mu}, \sigma(p))}\right) = \mathrm{O}\left(n\log 1/\sigma(p) + \log 1/\mu\right) .$$

In each node of the tree we have to perform n^2 additions and the coefficient size grows by n. So the total growth in coefficient length is $\log 1/\mu$, thus the coefficient length is $\mathrm{O}(n\tau + \log 1/\mu)$ at all nodes. We conclude that the bit complexity for a fixed value of μ is

$$\mathrm{O}\left((n\log 1/\sigma(p) + \log 1/\mu) \cdot n^2 \cdot (n\tau + \log 1/\mu)\right) .$$

The quantity $\log(1/\mu)$ starts at $7n$, doubles in each iteration, and ends at $\mathrm{O}(n(\log n + \tau + \log(1/\sigma(q)))$. Therefore the total bit complexity is $\mathrm{O}(n^4(\log n + \tau + \log 1/\sigma(q))^2)$. Except for the $\log n$ term, this is the same as in the case of known separation.

Theorem 13. Let q be be a polynomial with $|q_n| \geq 1$, and $|q_i| \leq 2^{\tau-1}$ for all i. The bit complexity of isolating the real roots of q is

$$\mathrm{O}\left(n^4 \left(\log n + \tau + \log \frac{1}{\sigma(q)}\right)^2\right) .$$

The coefficients of q need to be approximated with $\mathrm{O}(n(\tau + \log 1/\sigma(q)))$ bits after the binary point.

5. Multiple Roots: A Special Case

We give a partial extension to polynomials with multiple roots. It requires additional inputs m_0 and k_0 (obtained, for example, by a precomputation step) and works under the *precondition* that p has exactly m_0 distinct real roots, that k_0 is the degree of the gcd of p and its derivative p', and that $m_0 \geq 2$ and $k_0 \geq 1$. If $k_0 = 0$, p has no multiple roots and we know already how to isolate the roots of p. If $m_0 \leq 1$, there is no need for isolation. The *postcondition* is as follows:

(1) If the precondition holds, the algorithm either returns a set of m_0 intervals or a failure indicator. If intervals are returned, all but one interval is marked as "simple root".

(2) If the algorithm returns a set of intervals, these intervals isolate the real roots of p. Moreover, p has at most one multiple real root and the interval marked "simple root" contains simple roots.

(3) If the algorithm returns a failure indicator, any real root of p has multiplicity at most k_0.

In other words, if the precondition holds and p has a real root of multiplicity k_0+1 (which is then the unique multiple real root), the algorithm must return isolating intervals. If the precondition holds and p has several multiple real roots, the algorithm must return a failure indicator. If the precondition holds, p has at most one multiple real root, and this root has multiplicity less than k_0+1, the algorithm may either return a failure indicator or isolating intervals.

Why are we interested in an algorithm with this seemingly strange functionality? The answer is that it is exactly this behavior that is needed in computing cylindrical algebraic decompositions (cad for short) of (semi-)algebraic sets (Eigenwillig et al., 2007; Berberich et al., 2008). For a general definition and discussion of cads we refer the reader to the books (Basu et al., 2006; Caviness and Johnson, 1998). Here, we briefly review cads for algebraic curves in the plane and how the functionality defined above is useful for computing them.

Let $f \in \mathbb{Z}[x,y]$ be a polynomial in two variables and let $S = V(f)$ be its zero set. A cad for f consists of decompositions S_i of $\mathbb{R}^i$, $i = 1,2$, into semi-algebraic sets such that (1) the y-projection of each cell in S_2 projects onto a cell in S_1 and such that (2) S is the union of cells in S_2.

The construction of a cad for f consists of two steps. In the first step, the x-coordinates of the critical points of f, i.e., the points where y-projection does not describe a local diffeomorphism, are determined. The critical points of f are self-intersections, isolated points, points with vertical tangents, ... Their x-coordinates are among the real roots of the y-resultant of f and f_y, the partial derivative of f with respect to y. Let C be the set of these roots; C decomposes the real line into singletons and open intervals between these singletons, see Figure 3. The decomposition S_1 of $\mathbb{R}^1$ consists of these singletons and intervals. We call the singletons *critical* cells and the intervals *noncritical cells*. In the second step, the lifting step, one constructs above each cell of S_1 a stack of cells of S_2 and also computes adjacency information between cells in neighboring stacks.

Above each cell of S_1, $V(f)$ consists of a fixed number of connected components. For each cell Γ of S_1, let x_Γ be a point in the cell. If Γ is an interval, we may choose x_Γ arbitrarily in Γ. The connected components above Γ correspond to the real roots of $f(x_\Gamma,y)$. For a noncritical cell Γ, $f(x_\Gamma,y)$ is square-free. Hence, we can determine its real roots by either a Descartes algorithm for integer polynomials if we succeeded in choosing a rational x_Γ or the approximate root isolation algorithm described in the preceding section.

However, in case of a critical Γ, we are faced with two problems: First, $f(x_\Gamma,y) \in \mathbb{R}[y]$ may have multiple real roots, and second, its coefficients are algebraic numbers that are nonrational in general. Thus making $f(x_\Gamma,y)$ square-free is a costly computation. If S is in generic position with respect to the projection direction (genericity can be achieved by a suitably chosen, e.g., randomly chosen, linear transformation $x := x + ay$), $f(x_\Gamma,y)$ will have only one multiple root and moreover the number m_0 of distinct real roots and the multiplicity k_0+1 of the multiple root can be obtained in a preprocessing step; see (Eigenwillig et al., 2007) for details. The number m_0 of distinct real roots is also valid if the projection direction is not generic. However, for a nongeneric projection direction, there will be more than one multiple root. We conclude that for resolving the nature of $f(x,y)$ above x_Γ, the functionality defined at the beginning of the section is precisely what is needed. When the algorithm stops with a failure indication, the projection direction is not generic. A different linear transformation is tried and the computation is repeated.

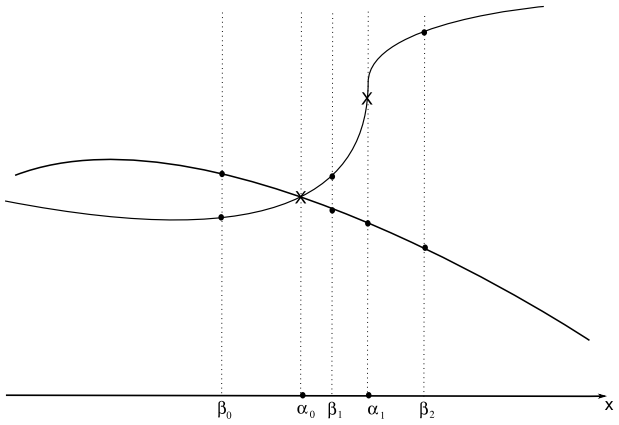

Fig. 3. The critical points of the set $S = V(f)$ are indicated as crosses. Their x-coordinates are α_0 and α_1. The α_i decompose the real line into five cells, namely $(-\infty,\alpha_0)$, $\{\alpha_0\}$, (α_0,α_1), $\{\alpha_1\}$, and $(\alpha_1,+\infty)$. Above each cell, $V(f)$ consists of a fixed number of connected components; in the example, there is one component above $\{\alpha_0\}$ and there are two components above each other cell. The points β_0, β_1 and β_2 are arbitrary points in the three open intervals. The y-values of points over β_i are simple roots of the square free polynomial $f(\beta_i,y)$. However, the polynomials $f(\alpha_j,y)$ may have multiple roots. If for $f(\alpha_j,y)$, the number m of points above α_j and the degree k_0 of $\gcd(f(\alpha_j,y),\frac{\partial}{\partial y}f(\alpha_j,y))$ is known, the algorithm of this section isolates the roots of $f(\alpha_j,y)$ or returns a failure indicator in the case where $f(\alpha_j,y)$ has more than one multiple root; these multiple roots may be real or complex.

Kerber et al. (Kerber, 2006; Eigenwillig et al., 2007; Eigenwillig, 2008) showed that a variant, which they termed *m-k* bitstream-Descartes method, of the randomized bitstream-Descartes method (Eigenwillig et al., 2005; Eigenwillig, 2008) can provide the desired functionality. However, they did not fully analyze the bit complexity of the variant. We will show that a variant of our algorithm can provide the desired functionality. We also analyze its bit complexity.

The idea is simple. Consider a sufficiently good approximation p^* of p. Then simple real roots of p turn into simple real roots of p^* and nonreal roots of p turn into nonreal roots of p^*. A real root z of p of multiplicity ℓ turns into ℓ roots (counted with multiplicity) of p^* contained in a disc D of radius $9\sqrt[n]{\mu}$ with center z. The roots of p^* corresponding to z may be real or nonreal, simple or multiple. We need to discern the roots of p^* corresponding to multiple real roots of p from those corresponding to simple real roots of p^*. If μ is small enough, we will be able to do so.

Again, we assume that all roots of p are contained in a disc of radius 1/4 with center $1/2+0i$. and have our algorithm driven by a parameter μ. We start with $\mu = 2^{-7n}$. For a fixed value of μ, let p^* be an ε-approximate of p, where $\varepsilon \le \mu|p|/(n+2)$. Then $|p-p^*| < \mu|p|$. As in the preceding section, let $L_0 = 18n\sqrt[n]{\mu}$. We call an interval I *long* if $w(I) \ge L_0$ and *short* otherwise.

Consider any long subdivision interval I. What should we expect as the value of $var(p^*,I^+)$? Corresponding roots of p and p^* lie within $9\sqrt[n]{\mu}$ of each other and $9\sqrt[n]{\mu}$ is small compared to L_0. Thus if a real root z of multiplicity k lies in I, I^+ should *see all* k roots of p^* corresponding to z and hence $var(p,I^+)$ should be at least k. This suggests that we should return a failure indicator as soon as $var(p^*,I^+) \le k_0$ for all subdivision intervals.

Consider next a long subdivision interval that is almost short, i.e., $L_0 \le w(I) < 2L_0$. If $9\sqrt[n]{\mu} \ll \sigma(p)$, we might hope that I^+ *only sees* the k roots of p^* corresponding to z and hence $var(p,I^+)$

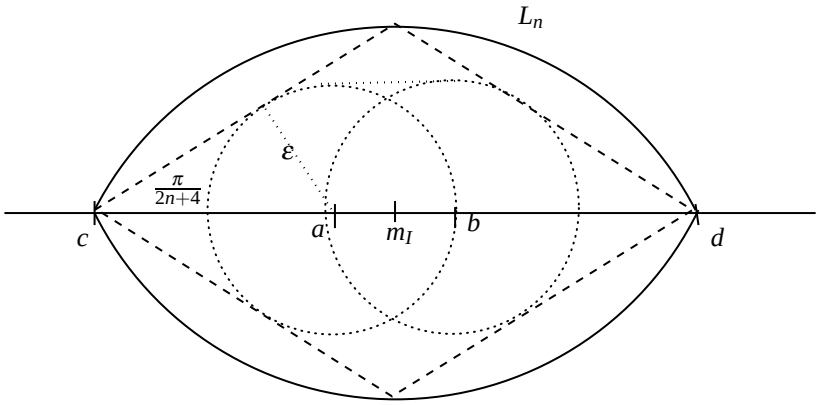

Fig. 4. $I=(a,b)$, $I^+=(c,d)$, and L_n denotes the Obreshkoff lens $L_n(I^+)$. For any $\varepsilon \le 2w(I)\sin(\pi/(2n+4))$, we have $U_\varepsilon(I) \subseteq L_n$ where $U_\varepsilon(I)$ denotes the ε-neighborhood of I.

should be exactly k. This suggests that we can return isolating intervals if we have found m_0-1 intervals I_ℓ with $var(p^*,I_\ell)=1$ and one interval I with $var(p^*,I)=k_0+1$.

Assume now that none of the two cases above arises, i.e., if we consider all subdivision intervals I with $L_0 \le w(I) < 2L_0$, there is still at least one that counts at least k_0+1 sign changes and there are no m_0-1 yet that count only one. We should then conclude that our current approximation is not good enough. So we stop, square μ, and start over.

The following theorem shows that $var(p^*,I^+)$ is strongly related to the number of roots of p near I and captures the intuition underlying the reasoning above.

Theorem 14. Let p be a polynomial of degree $n \ge 2$ and roots of modulus less than one and let p^* be an approximation of p with $|p-p^*| < \mu|p|$ and $\mu \le 2^{-7n}$. Let I be a long interval and let m_I be its midpoint. Then

$$\#\text{ of roots of } p \text{ in } U_{w(I)/(2n)}(I) \le var(p^*,I^+) \le \#\text{ of roots of } p \text{ in } U_{6nw(I)}(m_I).$$

Roots are counted with multiplicity and, for a set S, $U_\varepsilon(S)$ is the ε-neighborhood of S.

Proof. Let v be the number of roots of p in $U_{w(I)/(2n)}(I)$. Then $U_{w(I)/(2n)+9\sqrt[n]{\mu}}(I)$ contains at least v roots of p^*. Assume $U_{w(I)/(2n)+9\sqrt[n]{\mu}}(I) \subseteq L_n(I^+)$. Then $var(p^*,I^+) \ge v$ by Theorem 1. For the inclusion, we refer to Figure 4. We have $U_\varepsilon(I) \subseteq L_n(I^+)$ if $\varepsilon \le 2w(I)\sin(\pi/(2(n+2))$. Since $\sin(\pi(/(2(n+2)) \ge 1/(n+2) \ge 1/(2n)$, the inclusion certainly holds whenever $\varepsilon \le w(I)/n$. Thus it holds for $\varepsilon = w(I)/(2n)+9\sqrt[n]{\mu}$. This proves the first inequality.

By Theorem 1, $var(p^*,I^+)$ is bounded by the number of roots of p^* in $A_n(I^+)$ and hence by the number of roots of p in $X := U_{9\sqrt[n]{\mu}}(A_n(I^+))$. Let r be the radius of the Obreshkoff discs D_n for I^+. The extended sine theorem yields $2r = w(I^+)/\sin(\pi/(n+2)) < (n+2)w(I^+)/2 \le 2n\cdot 5w(I)/2 = 5nw(I)$. Hence the distance from m_I to an arbitrary point in X is at most $5nw(I)+9\sqrt[n]{\mu}$. This is at most $6nw(I)$. □

The next lemma is a consequence of Theorem 14. It tells us under what circumstances an isolating interval I for a real root of p^* gives rise to an isolating interval $\tilde{I}$ for a simple real root of p.

Lemma 15. Let I either be a long interval with $var(p^*, I^+) = var(p^*, I) = 1$ or a singleton interval[3] $[m^*, m^*]$ for which $p^*(m^*) = 0$ and $var(p^*, I_\ell^+) = 1 = var(p^*, I_r^+) = 1$ for long intervals $I_\ell = (\ , m^*)$ and $I_r = (m^*, \ )$ ending and starting in m^*, respectively. Then $\tilde{I}$ contains a unique zero of p.

If I and I' are disjoint long intervals satisfying one of the conditions above, $\tilde{I}$ and $\tilde{I'}$ are disjoint.

Proof. In the first case, I contains a zero of p^*, and, by Theorem 14, $U_{w(I)/2n}(I)$ contains at most one root of p. Since corresponding roots have distance less than $9\sqrt[n]{\mu}$ and $9\sqrt[n]{\mu} \le w(I)/2n$, $U_{9\sqrt[n]{\mu}}(I)$ contains exactly one root of p. This root must be real and hence lies in $\tilde{I}$. The second case is also a direct consequence of Theorem 14 applied to the interval I_ℓ or I_r, respectively.

We come to the second part. If I and I' are singletons, they are at least L_0 apart and hence $\tilde{I}$ and $\tilde{I'}$ are disjoint. So assume that I is not a singleton and $w(I) \ge w(I')$. If I^+ contains I', $var(p^*, I^+) \ge 2$, a contradiction. So I^+ does not contain I and hence I and I' are at least $2L_0$ apart. Thus $\tilde{I}$ and $\tilde{I'}$ are disjoint. □

We can now give the details of the *m-k deterministic bitstream-Descartes algorithm*; see Algorithm 4 for pseudocode. The algorithm maintains an output list and a candidate output list. The candidate output list contains singleton intervals $[m, m]$ with $p^*(m) = 0$ that have not been verified yet to correspond to simple roots of p. We have an approximation p^* of p with $|p - p^*| < \mu|p|$ and $\mu \le 2^{-7n}$. We proceed in rounds. At the beginning of each round all active intervals have the same length. We first process the active intervals I with $var(p^*, I^+) = 1$. We remove them from the list of active intervals and add I to the output list if $var(p^*, I) = 1$, otherwise we discard them. A singleton interval $[m, m]$ is moved from the candidate output list to the output list if there is no active interval either ending or starting in m. Then we process the intervals I with $var(p^*, I^+) \ge 2$. We split I at m_I into I_ℓ and I_r. A subinterval I_ℓ or I_r is added to the list of active intervals if its extension I_ℓ^+ or I_r^+ counts at least one sign change. The singleton $[m_I, m_I]$ is added to the candidate output list, if it is a zero of p^*. At all times, let J denote the minimal interval containing all active intervals. We stop when we reach one of the following situations:

- (R) When a short interval is added to the list of active intervals, we square μ and start over.
- (F) $var(p^*, I^+) \le k_0$ for all active intervals I. We stop and return a failure indicator.
- (S) $m_0 - 1$ intervals $I_1, \ldots, I_{m_0-1}$ have been added to the output list O^*, there is at least one active interval I with $var(p^*, I^+) \ge k_0 + 1$, and the expanded intervals $\tilde{I}_1, \ldots, \tilde{I}_{m_0-1}$, and $\tilde{J}$ are disjoint.[4] We return the expanded intervals and mark $\tilde{I}_1$ to $\tilde{I}_{m_0-1}$ as "simple root".

We proceed to the analysis. We proceed in two steps. We first show that if the precondition holds and the algorithm stops in cases (F) or (S), the postcondition holds. In a second step, we show that the algorithm terminates if the precondition holds.

Correctness: Assume that the algorithm stops. If it stops in case (F), every active interval I counts at most k_0 sign changes for I^+. Thus, by Theorem 14, there is no real root of p of multiplicity $k_0 + 1$.

So assume that it stops in case (S). The output list O^* contains $m_0 - 1$ intervals $I_1, \ldots, I_{m_0-1}$, J is nonempty and contains an interval I with $var(p^*, I^+) \ge k_0 + 1$, and the extended intervals $\tilde{I}_1, \ldots, \tilde{I}_{m_0-1}$, and $\tilde{J}$ are disjoint. An I_j is either a singleton $[m, m]$ with $p^*(m) = 0$ and there exists

[3] A real zero m^* of p^* might correspond to a multiple zero of p. For such a zero, at least one of the subdivision intervals having m^* as an endpoint counts more than one sign change.

[4] The subdivision intervals containing a multiple root of p are contained in J. As long as O^* has fewer than $m_0 - 1$ elements or $\tilde{J}$ is not disjoint from the extended intervals in O^*, there might be more than one multiple real root.

Algorithm 4 *m-k* deterministic bitstream-Descartes Algorithm

Require: $p = \sum_{0 \le i \le n} p_i x^i$ and all roots of p lie in a disc of radius 1/4 centered at $1/2 + 0i$; p has exactly m_0 distinct real roots and $k_0 = \deg \gcd(p, p') \ge 1$.

Ensure: postcondition is as stated at the beginning of Section 5

$\mu = 2^{-7n}$;
while (**true**) **do**
  choose $\varepsilon \le \mu |p|/(n+2)$ and let p^* be an ε-approximate of p;
  $O^* := \emptyset$; $CO^* := \emptyset$; *output and candidate output list*
  $A := \{ (0,1) \}$; $A_{new} := \emptyset$; *A_{new} is the A of the next iteration*
  while (**true**) **do**
    all intervals in A have the same length
    remove all I with $var(p^*, I^+) = 1$ from A; move the ones with $var(p^*, I) = 1$ to O^*;
    move any $[m,m]$ from CO^* to O^* for which there is no active interval with endpoint m;
    if $var(p^*, I^+) \le k_0$ for all $I \in A$ **then**
      return a failure indicator; *Case (F)*
    end if
    Let J be a minimal interval containing all active intervals;
    if O^* contains exactly $m_0 - 1$ intervals I_1 to I_{m_0-1}, $var(p^*, I^+) \ge k_0 + 1$ for at least one active interval I, and $\tilde{I}_\ell$ is disjoint from $\tilde{J}$ for all ℓ **then**
      return $\tilde{I}_1$ to $\tilde{I}_{m_0-1}$ and $\tilde{J}$; *Case (S)*
    end if
    while ($A \neq \emptyset$) **do**
      let $I \in A$ be arbitrary; remove I from A;
      add I_ℓ to A_{new} if $var(p^*, I_\ell^+) \ge 1$; add I_r to A_{new} if $var(p^*, I_r^+) \ge 1$; add $[m_I, m_I]$ to CO^* if $p^*(m_I) = 0$;
    end while
    $A := A_{new}$; $A_{new} := \emptyset$;
    if A contains a short interval **then**
      $\mu = \mu^2$; break from the inner while loop and restart with a better approximation;
    end if
  end while
end while

no active interval with endpoint m or it satisfies $var(p^*, I_j) = var(p^*, I_j^+) = 1$. Then according to Lemma 15, each $\tilde{I}_j$ contains exactly one real root of p. By the precondition, p has exactly m_0 real roots. Let z_0 be the remaining real root of p and let k be its multiplicity. Then $1 \le k \le k_0 + 1$ by assumption. We need to show that z_0 is contained in $\tilde{J}$. By definition, $z_0 \notin I_j$ for $1 \le j \le m_0 - 1$. Consider the chain $K_0, K_1, \ldots$ of subdivision intervals containing z_0 in their closure. All of them are long and hence $var(p^*, K_i^+) \ge k$ for all i by Theorem 14. If some interval containing z_0 in its closure is active when the algorithm terminates, $z_0 \in \tilde{J}$. So assume, there is no active interval containing z_0 in its closure when the algorithm terminates. Then $k = 1$ and either an interval containing z_0 was added to O^* or the singleton $[z_0, z_0]$ was added to CO^* and then moved to O^*. In either case, we have a contradiction.

Termination: The approximation parameter μ is called *small* if

$$\mu \leq \min\left(\left(\frac{\sigma(p)}{72\cdot 25n^2}\right)^n, 2^{-7n}\right).$$

If μ is small and $n \geq 2$, we have:

- $4L_0 = 72n\sqrt[n]{\mu} \leq \sigma(p)/(25n) \leq \sigma(p)/50$.
- $9\sqrt[n]{\mu} \leq \sigma(p)/2$. Hence nonreal roots of p correspond to nonreal roots of p^*, simple real roots of p correspond to simple real roots of p^*, and a real root z of p of multiplicity k corresponds to k roots of p^* in a disk of radius $9\sqrt[n]{\mu}$ with center z.
- $4L_0 \cdot 6n + 2L_0 \leq \sigma(p)/4$ and hence $I \subseteq U_{6nw(I)}(I) \subseteq U_{\sigma/4}(m_I)$ for any I with $w(I) \leq 4L_0$.

Theorem 16. Let p be a polynomial of degree n and roots with modulus less than one. Let $k_0 = \deg\gcd(p,p')$, let μ be small, and let p^* be such that $|p^* - p| < \mu|p|$. Then for any interval I with $L_0 \leq w(I) < 4L_0$, $var(p^*, I^+) \leq k$, where k is the multiplicity of the unique real root of p in $U_{\sigma/4}(m_I)$; $k = 0$, if the disk contains no real root.

Proof. Since μ is small and $w(I) < 4L_0$, we have $U_{6nw(I)}(m_I) \subseteq U_{\sigma/4}(m_I)$. The latter disk can contain at most one root of p. The root must be real and, by Theorem 14, $var(p^*, I^+)$ is at most its multiplicity. □

Theorem 17. Let p be a polynomial of degree n and roots of modulus at most one; p has exactly m_0 distinct real roots and $k_0 = \deg\gcd(p,p')$. If μ is small, Algorithm 4 terminates. The algorithm terminates with

$$\mu \geq \min\left(\left(\frac{\sigma(p)}{72\cdot 25n^2}\right)^{2n}, 2^{-7n}\right).$$

Proof. If the algorithm does not terminate, a short interval is added to the list of active intervals. Just before this happens, any active intervals I has length L with $L_0 \leq L < 2L_0$ and $var(p^*, I^+)$ is at least two. We argue that the algorithm would have terminated in this iteration.

If p has no real root of multiplicity $k_0 + 1$, $var(p^*, I^+) \leq k_0$ for all intervals of length L by Theorem 16 and the algorithm terminates in case (F).

So assume that p has exactly $m_0 - 1$ simple real roots $z_1, \ldots, z_{m_0-1}$ and one real root, say z_0, of multiplicity $k_0 + 1$ with $k_0 \geq 1$. For i with $1 \leq i \leq m_0 - 1$, let z_i^* be the simple real root of p^* corresponding to z_i. Then under the given assumptions the following Lemma holds.

Lemma 18. The output list contains exactly $m_0 - 1$ intervals I_1 to I_{m_0-1}. The extended intervals $\tilde{I}_j$ contain one real root of p each and are disjoint. Each proper interval on the output list has length at least $2L_0$.

Proof. For i with $1 \leq i \leq m_0 - 1$, let K_i be a subdivision interval of length $2L$ containing z_i^* in its closure. Then $var(p^*, K_i^+) \leq 1$ by Theorem 16. If $z^* \in K_i$, $var(p^*, K_i) = 1$ and hence some ancestor of K_i was added to the output list. If z^* is an endpoint of K_i, $var(p^*, K_i) = 0 = var(p^*, K_i')$, where K_i' is the other subdivision interval of length $2L$ with endpoint z^* and hence $[z^*, z^*]$ was added to the output list. We conclude that the output list contains at least $m_0 - 1$ elements. Also, by Lemma 15, each extended interval contains exactly one real root of p and the extended intervals are disjoint. The same Lemma shows that the output list cannot contain m_0 or more elements, as the extension of each of them would contain a simple root of p. □

The preceding Lemma shows that the intervals $\tilde{I}_j$, $1 \le j \le m_0 - 1$, isolate the simple roots of p. It remains to show that $\tilde{J}$ contains z_0 and is disjoint from the $\tilde{I}_j$'s.

Lemma 19. *$\tilde{J}$ contains z_0 and is disjoint from any $\tilde{I}_j$ on the output list.*

Proof. Let I_0 be a subdivision interval of length L containing z_0 in its closure; if z_0 is a subdivision point, there is a choice of two intervals for I_0, otherwise it is uniquely defined. Then $var(p^*, I_0^+) \ge k_0 + 1$ by Theorem 14. Thus I_0 is active and hence $\tilde{J}$ contains z_0.

Now consider any active interval I; $U_{\sigma/4}(m_I)$ contains at most one root of p and $var(p^*, I^+)$ is at most the multiplicity of this root (Theorem 16). As $var(p^*, I^+) > 1$, this root has to be z_0. Thus $|m_I - z_0| \le \sigma/4$ and hence $w(\tilde{J}) \le \sigma/4 + w(I)/2 + 18n\sqrt[n]{\mu} \le \sigma/4 + 36n\sqrt[n]{\mu} < \sigma$. If J would contain an interval I_j from the output list, then $\tilde{J}$ would contain the simple real root z_j of p. But this contradicts $\sigma \ge |z_0 - z_j|$ as $\tilde{J}$ also contains z_0 and $w(\tilde{J}) < \sigma$.

We must still exclude the case that J shares an endpoint with any proper interval I_j from the output list. So assume that I_j shares an endpoint with some active interval I. Then $w(I) = L < 2L_0 \le w(I_j)$. Thus $I^+ \subseteq I_j^+$ and hence $var(I_j^+) \ge var(I^+) > 1$, a contradiction. □

It remains to prove the bound on μ. If the algorithm terminates in the first iteration, it stops with $\mu = 2^{-7n}$. If μ is small in an iteration, the iteration is the last. So if the algorithm requires more than one iteration, μ is not small in the next to last iteration. The bound on μ follows. □

We turn to the complexity analysis. The analysis of Algorithm 3 essentially carries over. Both algorithms start with $\log(1/\mu) = 7n$ and double μ in every iteration. For any fixed value of μ, they generate the same subdivision tree and hence incur the same cost of

$$\mathrm{O}\left((n\log 1/\sigma(p) + \log 1/\mu) \cdot n^2 \cdot (n\tau + \log 1/\mu)\right)$$

bit operations, as shown in Section 4.2. Algorithm 3 stops with $\mu \ge \min((\sigma(p)/(100n))^{2n}, 2^{-7n})$, the algorithm of this section stops with $\mu \ge \min((\sigma(p)/(72 \cdot 25n^2))^{2n}, 2^{-7n})$. In both cases, $\log(1/\mu)$ stops at $\mathrm{O}(n(\log n + \tau + \log(1/\sigma(q))))$. Therefore we obtain the same bit complexity as for Algorithm 3.

Theorem 20. Let q be a polynomial with root separation $\sigma(q)$, $|q_n| \ge 1$, and $|q_i| \le 2^{\tau-1}$ for all i. Furthermore, let m_0 be the number of distinct real roots of q and $k_0 = \deg\gcd(q, q')$. The bit complexity of the *m-k* deterministic bitstream-Descartes algorithm is

$$\mathrm{O}\left(n^4 \left(\log n + \tau + \log \frac{1}{\sigma(q)}\right)^2\right).$$

The coefficients of q need to be approximated with $\mathrm{O}(n(\tau + \log 1/\sigma(q)))$ bits after the binary point.

We remark that no bound on the bit complexity of the *m-k* randomized bitstream-Descartes algorithm of Eigenwillig et al. (2007) is available. As this algorithm is the main workhorse in the algorithms of (Eigenwillig et al., 2007; Berberich et al., 2008) for determining the topology of algebraic curves and surfaces, our complexity result may pave the way for determining the complexity of the entire topology computation.

6. Conclusions

The randomized bitstream-Descartes method as presented in (Eigenwillig et al., 2005) has already shown its effectiveness and strength in practice; it is a main ingredient of the algorithms of (Eigenwillig et al., 2007; Berberich et al., 2008) for cad (Cylindrical Algebraic Decomposition) computation. An implementation will become available in (CGAL, 2008). It remains to be seen, whether our deterministic algorithm is competitive with its randomized cousin. A first implementation is encouraging.

Collins and Krandick (1992) have described a natural extension of the Descartes algorithm for isolating all (complex) roots of a polynomial. Does their algorithm generalize to the bitstream model and can it be extended to the situation where multiple roots are allowed and the number of distinct complex roots is given as an additional input?

The continued fraction method (Akritas, 1980; Tsigaridas and Emiris, 2008) is an alternative to the Descartes method for root isolation. It would be interesting to generalize it to the bitstream model and to the *m-k* scenario.

References

Aberth, O., 1988. Precise Numerical Analysis. William C. Brown Company Publishers, Dubuque, IA, USA.

Akritas, A. G., 1980. The fastest exact algorithms for the isolation of the real roots of a polynomial equation. Computing 24 (4), 299–313.

Basu, S., Pollack, R., Roy, M.-F., 2006. Algorithms in Real and Algebraic Geometry, 2nd Edition. Springer, electronic version available at `www.math.gatech.edu/~saugata/bpr-posted1.html`.

Berberich, E., Kerber, M., Sagraloff, M., 2008. Exact geometric-topological analysis of algebraic surfaces. In: Proceedings of the twenty-fourth Annual Symposium on Computational Geometry (SoCG 08). pp. 164–173.

Bini, D., Fiorentino, G., 2000. Design, analysis and implementation of a multiprecision polynomial rootfinder. Numerical Algorithms 23, 127–173.

Caviness, B. F., Johnson, J. R. (Eds.), 1998. Quantifier Elimination and Cylindrical Algebraic Decomposition. Springer.

CGAL, 2008. CGAL (Computational Geometry Algorithms Library). `www.cgal.org`.

Collins, G., Krandick, W., 1992. An efficient algorithm for infallible polynomial complex root isolation. In: Proc. Intl. Symp. Symbolic and Algebraic Comp. (ISSAC). pp. 189–194.

Collins, G. E., Akritas, A. G., 1976. Polynomial real root isolation using Descartes' rule of signs. In: Jenks, R. D. (Ed.), SYMSAC. ACM Press, Yorktown Heights, NY, pp. 272–275.

Collins, G. E., Johnson, J. R., Krandick, W., 2002. Interval arithmetic in cylindrical algebraic decomposition. J. Symbolic Computation 34 (2), 143–155.

Eigenwillig, A., 2008. Real root isolation for exact and approximate polynomials using Descartes' rule of signs. Ph.D. thesis, Saarland University, Saarbrücken, Germany.

Eigenwillig, A., Kerber, M., Wolpert, N., 2007. Fast and exact analysis of real algebraic plane curves. In: Proc. Intl. Symp. Symbolic and Algebraic Comp. (ISSAC). pp. 151–158.

Eigenwillig, A., Kettner, L., Krandick, W., Mehlhorn, K., Schmitt, S., Wolpert, N., 2005. An Exact Descartes Algorithm with Approximate Coefficients (Extended Abstract). In: CASC. Vol. 3718 of LNCS. pp. 138–149.

Eigenwillig, A., Sharma, V., Yap, C., 2006. Almost tight complexity bounds for the Descartes method. In: Proc. Intl. Symp. Symbolic and Algebraic Comp. (ISSAC). pp. 151–158.

Johnson, J. R., Krandick, W., 1997. Polynomial real root isolation using approximate arithmetic. In: Proc. Intl. Symp. Symbolic and Algebraic Comp. (ISSAC). ACM Press, pp. 225–232.

Kerber, M., 2006. Analysis of real algebraic plane curves. Master's thesis, Saarland University.

Krandick, W., Mehlhorn, K., 2006. New Bounds for the Descartes Method. Journal of Symbolic Computation 41 (1), 49–66.

Mourrain, B., Rouillier, F., Roy, M.-F., March 2004. Bernstein's basis and real root isolation. Rapport de recherche 5149, INRIA-Rocquencourt, `http://www.inria.fr/rrrt/rr-5149.html`.

Obrechkoff, N., 2003. Zeros of Polynomials. Marina Drinov, Sofia, translation of the Bulgarian original.

Obreschkoff, N., 1925. Über die Wurzeln von algebraischen Gleichungen. Jahresbericht der Deutschen Mathematiker-Vereinigung 33, 52–64.

Obreschkoff, N., 1963. Verteilung und Berechnung der Nullstellen reeller Polynome. VEB Deutscher Verlag der Wissenschaften.

Ostrowski, A. M., 1950. Note on Vincent's theorem. Annals of Mathematics, Second Series 52 (3), 702–707, reprinted in: Alexander Ostrowski, Collected Mathematical Papers, vol. 1, Birkhäuser Verlag, 1983, pp. 728–733.

Pan, V., 1997. Solving a polynomial equation: Some history and recent progress. SIAM Review 39 (2), 187–220.

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

Rouillier, F., Zimmermann, P., 2004. Efficient isolation of a polynomial's real roots. J. Computational and Applied Mathematics 162, 33–50.

Schoenberg, I. J., 1934. Zur Abzählung der reellen Wurzeln algebraischer Gleichungen. Mathematische Zeitschrift, 546–564.

Schönhage, A., 1982. The fundamental theorem of algebra in terms of computational complexity, `http://www.informatik.uni-bonn.de/~schoe/fdmthmrep.ps.gz`.

Schönhage, A., 1985. Quasi-GCD computations. Journal of Complexity 1 (1), 118–137.

Tsigaridas, E., Emiris, I., 2008. On the complexity of real root isolation using continued fractions. Theor. Comput. Sci. 392, 158–173.

Yap, C., 1999. Fundamental Problems in Algorithmic Algebra. Oxford University Press.