A Descartes Algorithm for Polynomials with Bit-Stream Coefficients

Arno Eigenwillig[1], Lutz Kettner[1], Werner Krandick[2], Kurt Mehlhorn[1], Susanne Schmitt[1], and Nicola Wolpert[1]

[1] Max-Planck-Institut für Informatik, Saarbrücken, Germany
{arno,kettner,mehlhorn,sschmitt,wolpert}@mpi-inf.mpg.de
[2] Dept. of Computer Science, Drexel University, Philadelphia, PA, USA
krandick@cs.drexel.edu

Abstract. The Descartes method is an algorithm for isolating the real roots of square-free polynomials with real coefficients. We assume that coefficients are given as (potentially infinite) bit-streams. In other words, coefficients can be approximated to any desired accuracy, but are not known exactly. We show that a variant of the Descartes algorithm can cope with bit-stream coefficients. To isolate the real roots of a square-free real polynomial $q(x) = q_n x^n + \ldots + q_0$ with root separation ρ, coefficients $|q_n| \geq 1$ and $|q_i| \leq 2^\tau$, it needs coefficient approximations to $O(n(\log(1/\rho) + \tau))$ bits after the binary point and has an expected cost of $O(n^4(\log(1/\rho) + \tau)^2)$ bit operations.

1 Introduction

The isolation of the real roots of a real univariate polynomial $q(x) \in \mathbb{R}[x]$ is a fundamental task in computer algebra: given a polynomial q, 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. One of the best approaches to root isolation is the Descartes method. It is a bisection method based on the Descartes Rule of Signs to test for roots. Its modern form goes back to Collins and Akritas [1]. It can be formulated to operate on polynomials given in the usual power basis or in the Bernstein basis. For integer coefficients, it typically outperforms other methods. We review it in Section 3. We assume that the coefficients of our polynomials are given as potentially infinite bit-streams, i.e., coefficients are known to arbitrary precision, but, in general, never exactly.

We are the first to make a variant of the Descartes algorithm work in this setting. Our main tools are a sharper analysis of the rule of signs (Lemmas 5 and 6) and randomization (Sections 4.2 and 4.3). Our main result is as follows:

To isolate the real roots of a square-free (= no multiple roots) real polynomial $q(x) = q_n x^n + \ldots + q_0$ with root separation (= minimal distance between any two roots) ρ, coefficients $|q_n| \geq 1$ and $|q_i| \leq 2^\tau$, our algorithm needs coefficient approximations to $O(n(\log(1/\rho) + \tau))$ bits after the binary point and $O(n^4(\log(1/\rho) + \tau)^2)$ bit operations in expectancy.

The cost statement ignores the cost of computing the approximations of the coefficients with the required quality. Observe that the quantities n, ρ and τ are determined by the roots of our polynomial, i.e., the geometry of the problem, and hence the running time of our method is a function of the geometry of the problem.

The restriction to square-free inputs is inherent in the bit-stream setting, since detecting multiple roots is equivalent to testing for zero (cf. $x^2 - a$) and hence impossible.

The paper is structured as follows. In Section 2 we put our work into context and in Section 3 we review the Descartes method. Section 4 is the heart of the paper. We describe and analyze a variant of the Descartes method for polynomials with bit-stream coefficients. In Section 5 we report on some experimental observations.

2 Comparison to Related Work

The Descartes method can be formulated for polynomials in the usual power basis [2, 1, 3, 4] and for polynomials in the Bernstein basis [5–8]. The early work concentrated on polynomials with integer coefficients. More recent work [9, 4, 8] points out that the Descartes method can be combined with interval arithmetic for increased efficiency and to also handle some, but not all, polynomials with bit-stream coefficients [9, p. 152]. We are the first to exhibit a variant of the Descartes method handling all square-free polynomials with bit-stream coefficients.

Beyond the Descartes method, there is substantial work in numerical analysis on approximating the roots of a real polynomial [10]. Many algorithms were proposed for the simultaneous approximation of all complex roots of a polynomial with bit-stream coefficients. Most algorithms come without a guarantee of convergence. Weyl [11] exhibited the first complete algorithm and Pan [12] surveys the development till about 1995. The currently best algorithm is due to Pan [13]. It applies to polynomials with bit-stream coefficients. Given a polynomial $p(x) = \sum_i p_i x^i = p_n \prod_i (x - z_i)$ of degree n and a precision parameter b, his method computes approximate roots z_i^* such that after suitable renumbering $|z_i^* - z_i| < 2^{2-b/n}$. Here b must be at least $n \log n$ and the computational cost is $O(n \log^2 n(\log^2 n + \log b))$ arithmetic operations (additions and multiplications) on $O(b)$-bit numbers. It is assumed that all roots lie in the unit disk. The algorithm is, in the author's own words, *quite involved, and would require non-trivial implementation work*, and we are not aware of any implementation. Pan's algorithm can be used to isolate real roots. Thus it solves a more general problem (isolation of all roots and not only real roots) and is asymptotically much faster than our algorithm (quadratic dependence on n instead of quartic and linear dependence on $\log(1/\operatorname{sep}(p))$ instead of quadratic). Does this make our contribution obsolete? We believe not: First, because our algorithm is very simple and easily implemented. Second, we expect the algorithm to be superior for small to medium degree polynomials.

3 The Descartes Method in the Bernstein Basis

Fix an integer $n \geq 0$ and boundaries $c < d$ of an interval $[c, d]$. The *Bernstein basis* [14, 15] of the vector space $\mathbb{R}[x]_{\leq n}$ of polynomials of degree at most n consists of the *Bernstein polynomials* $B_0^n, B_1^n, \ldots, B_n^n$, where:

$$B_i^n(x) = B_i^n[c,d](x) = \binom{n}{i} \frac{(x-c)^i (d-x)^{n-i}}{(d-c)^n}, \quad 0 \leq i \leq n\ . \tag{1}$$

If $p(x) = \sum_{i=0}^{n} b_i B_i^n[c,d](x)$, we call $b = (b_n, \dots, b_0)$ the Bernstein representation of p with respect to interval $[c,d]$ and b_0 the *first* and b_n the *last* coefficient. We have $p(c) = b_0$ and $p(d) = b_n$. The Bernstein polynomials form a *non-negative partition of unity*, meaning that $\sum_{i=0}^{n} B_i^n(x) = 1$ and $B_i^n(x) \geq 0$ for all $x \in [c,d]$. This is helpful in bounding error propagation: If $p(x) = \sum_{i=0}^{n} b_i B_i^n(x)$ and $\tilde{p}(x) = \sum_{i=0}^{n} \tilde{b}_i B_i^n(x)$ with $|\tilde{b}_i - b_i| \leq \varepsilon$ for all i, then for all $x \in [c,d]$ it holds that

$$|\sum_i \tilde{b}_i B_i^n(x) - \sum_i b_i B_i^n(x)| \leq \varepsilon \sum_i |B_i^n(x)| = \varepsilon \sum_i B_i^n(x) = \varepsilon \ . \tag{2}$$

The most important property for our purposes is the Descartes Rule of Signs. Let $a = (a_0, \dots, a_n)$ be a finite sequence of real numbers. The *number of sign variations* in a, denoted $\mathrm{var}(a)$, is the number of pairs (i, j) of integers with $0 \leq i < j \leq n$ and $a_i a_j < 0$ and $a_{i+1} = \ldots = a_{j-1} = 0$.

Theorem 1 (The Descartes Rule of Signs). *Let $p(x) = \sum_{i=0}^{n} b_i B_i^n[c,d](x)$ be a polynomial. Then $\mathrm{var}(b)$ exceeds the number of zeroes of p in the open interval (c,d) by an even non-negative integer.*

This rule is traditionally stated for the power basis and the interval $(0,\infty)$; see [16] for a proof with historical references. The Bernstein formulation appears in [5–8].

Theorem 1 is the basis for a bisection method for root isolation in exact arithmetic. We start with an interval I guaranteed to contain all real zeroes of p and call *Descartes*(p, I). The procedure *Descartes*(p, I) works as follows: Let $I = (c,d)$, $p(x) = \sum_{i=0}^{n} b_i B_i^n[c,d](x)$ the Bernstein representation of p with respect to the interval (c,d), and $v = \mathrm{var}(b)$. If $v = 0$, return. If $v = 1$, report I as an isolating interval and return. If $v \geq 2$, choose a point $m = \alpha c + (1-\alpha)d$ with $1/4 \leq \alpha \leq 3/4$. (Any $\alpha \in (0,1)$ would work, but a choice near the middle guarantees linear convergence.) If $p(m) = 0$, report the exact root m. Call *Descartes*$(p, (c,m))$ and *Descartes*$(p, (m,d))$.

The Bernstein representations b' and b'' of p with respect to intervals $I' = [c,m]$ and $I'' = [m,d]$ are readily computed from b by *de Casteljau's algorithm* depicted in Figure 1. It operates on a triangular array of numbers. The top row $(b_{0,0}, \dots, b_{0,n})$ is initialized to $b = (b_0, \dots, b_n)$. For $1 \leq j \leq n$, the j-th row $(b_{j,0}, \dots, b_{j,n-j})$ is computed according to

$$b_{j,i} := \alpha b_{j-1,i} + (1-\alpha) b_{j-1,i+1} \ . \tag{3}$$

The result sequences are given by the two sloped sides $b' = (b_{0,0}, b_{1,0}, \dots, b_{n,0})$ and $b'' = (b_{n,0}, b_{n-1,1}, \dots, b_{0,n})$ of the triangle; see, e.g., [14, 3.2/3.3] and [15, Lemma 4.2].

The Descartes method terminates iff the polynomial p is square free. Termination proofs rest on partial converses of Theorem 1 such as the following.

Theorem 2 (Ostrowksi). *Consider a polynomial p and its roots in the complex plane $\mathbb{C}$. Let I be an interval with midpoint m and width $|I|$ and let $v = \mathrm{var}(b)$ be the number of sign variations in the Bernstein representation of p with respect to I.*

If the disc bounded by the circle C centered at m passing through the endpoints of I does not contain any root of p, then $v = 0$ (one-circle theorem).

If the union of the discs bounded by the circles $\underline{C}$ and $\overline{C}$ centered at $m \pm i(\sqrt{3}/6)|I|$ and passing through the endpoints of I contains precisely one simple root of p (which is then necessarily a real root), then $v = 1$ (two-circle theorem).

$$
\begin{array}{ccccccccccc}
b_{0,0} & & b_{0,1} & & b_{0,2} \;\cdots\; b_{0,n-2} & & b_{0,n-1} & & b_{0,n} \\
& b_{1,0} & & b_{1,1} & \cdots & b_{1,n-2} & & b_{1,n-1} & \\
& & b_{2,0} & & b_{2,1} \;\cdots\; b_{2,n-3} & & b_{2,n-2} & & \\
& & & b_{3,0} & \cdots & b_{3,n-3} & & & \\
& & & . & \cdots & . & & & \\
& & & & b_{n,0} & & & &
\end{array}
$$

Fig. 1. The de Casteljau triangle in which $b_{j,i} = \alpha b_{j-1,i} + (1-\alpha) b_{j-1,i+1}$.

For a proof see [16]. The circle theorems allow us to bound the depth of the Descartes recursion tree, depending on the distance between the roots. Let p be a non-zero polynomial with roots ξ_1 to ξ_n. We define its *root separation* $\operatorname{sep}(p) = \min\{|\xi_i - \xi_j| \mid i \neq j\}$ as the minimum distance between any two roots; $\operatorname{sep}(p) > 0$ iff p is square-free.

Corollary 3. *The Descartes method applied to any square-free polynomial p and start interval I_0 terminates. The interval at any internal node of the recursion tree has width at least $(\sqrt{3}/2)\operatorname{sep}(p)$, the interval at a leaf has width at least $(\sqrt{3}/8)\operatorname{sep}(p)$. Given $\sigma \leq \operatorname{sep}(p)$, recursion depth is at most $D(\sigma) := \lfloor \log(|I_0|/\sigma)/\log(4/3) + 3/2 \rfloor$.*

Proof. Consider any interval I for which the Bernstein representation has two or more sign variations. The contrapositive of Theorem 2 tells us: If p has no root in I then there must be a pair of conjugate roots $\xi, \overline{\xi}$ in the disc bounded by C. The diameter of C is $|I|$, hence $|I| \geq |\xi - \overline{\xi}| \geq \operatorname{sep}(p)$. If p has exactly one root ξ' in I, then p has a pair of conjugate roots $\xi, \overline{\xi}$ in the discs bounded by $\overline{C}$ and $\underline{C}$. The diameter of $\overline{C}$ and $\underline{C}$ is $(2/\sqrt{3})|I| \geq |\xi - \xi'| \geq \operatorname{sep}(p)$. If p has two roots ξ, ξ' in I, then $|I| \geq |\xi - \xi'| \geq \operatorname{sep}(p)$. In all three cases, *Descartes*(p,I) generates recursive calls only if $|I| \geq (\sqrt{3}/2)\operatorname{sep}(p)$. The interval at a leaf is at least one fourth the length of the interval at its parent. Since the interval length is multiplied by $3/4$ or less in each step, the depth k of an internal node satisfies $|I_0|(3/4)^k \geq (\sqrt{3}/2)\operatorname{sep}(p)$ or $k \leq \log(|I_0|/\operatorname{sep}(p))/\log(4/3) + 1/2$. $\square$

Proposition 4. *A Descartes recursion tree for a polynomial of degree n has at most n nodes at any depth.*

In the Bernstein basis, this easily seen from the well-known variation diminishing property of repeated linear interpolation. A proof appears in [7, Thm. 10.38].

4 The Descartes Method for Polynomials with Bit-Stream Coefficients

We present an algorithm *Descartes*$_{\text{approx}}$ to isolate the real roots of $p(x) = \sum_{i=0}^{n} b_i B_i^n(x)$ in $(0,1)$. We assume that the coefficients are given as bit-streams; in particular, for any fixed $\varepsilon > 0$, we can compute an approximate coefficient vector $\tilde{b} = (\tilde{b}_0, \ldots, \tilde{b}_n)$ with $|\tilde{b}_i - b_i| \leq \varepsilon$ for all i. We call $\tilde{b}$ an ε-approximate Bernstein representation of p. The pair $(\tilde{b}, \varepsilon)$ specifies an *interval polynomial* $\mathbf{p}(x) = \sum_{i=0}^{n} \mathbf{b}_i B_i^n(x)$ such that $\mathbf{p} \supseteq \{p\}$.

We start with a thought experiment. Consider executions of *Descartes* in exact arithmetic both on the exact Bernstein representation b and on its approximation $\tilde{b}$. The only computation *Descartes* ever does with the coefficients is repeated forming of averages as in Eq. (3). The absolute error in the result of such a convex combination is no larger than it is in the inputs. Hence the absolute errors in the de Casteljau triangles in all nodes of the Descartes tree for $\tilde{b}$ are ε-approximations of their counterparts in the tree starting from exact coefficients b. The shape of the exact Descartes tree depends on decisions based on the signs of exact entries. Can we mimic these decisions with intervals?

We call an interval *positive* $(+)$, if it contains only positive numbers; *negative* $(-)$, if it contains only negative numbers; and *indeterminate* (?), if it contains zero. A positive or negative interval is also called *determinate*. For a sequence of coefficient intervals $\mathbf{a} = (\mathbf{a}_0, \ldots, \mathbf{a}_n)$, we define its *set of potential numbers of sign variations* as $\mathrm{var}(\mathbf{a}) = \left\{ \mathrm{var}((a_0, \ldots, a_n)) \,\middle|\, a_i \in \mathbf{a}_i \text{ for } 0 \le i \le n \right\}$. For example, we have $\mathrm{var}(([2,3], [-1,1])) = \{0,1\}$, $\mathrm{var}(([2,3], [-1,1], [2,3])) = \{0,2\}$, and $\mathrm{var}(([2,3], [-1,1], [-2,-1])) = \{1\}$. The fact that some $\mathbf{a}_i$ is indeterminate does not imply that $\mathrm{var}(\mathbf{a})$ contains more than one value, as the third example shows.

Consider any node in the approximate Descartes tree. We have an approximate coefficient sequence $\tilde{b}$. Each $\tilde{b}_i$ stands for an interval $\tilde{\mathbf{b}}_i = [\tilde{b}_i - \varepsilon, \tilde{b}_i + \varepsilon]$. Define $\mathrm{var}_\varepsilon(\tilde{b}) = \mathrm{var}(\tilde{\mathbf{b}}_0, \ldots, \tilde{\mathbf{b}}_n)$. Observe that $\mathrm{var}_\varepsilon(\tilde{b})$ contains $\mathrm{var}(b)$. If $\mathrm{var}_\varepsilon(\tilde{b})$ is a singleton or disjoint from $\{0,1\}$, we know what the exact algorithm would do and can do the same.

But what should we do if $\mathrm{var}_\varepsilon(\tilde{b})$ is not a singleton and contains a number less than two? The first solution that comes to mind is to switch to a smaller ε. This will not always solve the problem: Assume we start with a degree-2-polynomial with Bernstein representation $(1, -\beta, \beta)$ with respect to $(0,1)$ where β is any positive irrational number less than one. We split the interval at $1/2$ and obtain $((1-\beta)/4, 0, \beta)$ for the right subinterval. For any approximation of β, the 0 will turn into an interval straddling zero and hence the potential sign variations are $\{0,2\}$. A second solution that comes to mind is to perform recursive calls whenever the set of potential sign variations contains a number larger than one. However, then the procedure might not terminate, namely, when ε is so large that $\mathbf{p}$ contains a non-square-free polynomial. Furthermore: what if the set of potential sign changes contain both zero and one? What if, after subdivision, the last coefficient of b' (first coefficient of b'') is indeterminate, i.e., our polynomial may be zero at the split point?

The last two problems disappear when first and last coefficients are determinate. All problems disappear when first and last coefficients are large. We call $\tilde{b}_i$ *large* if $|\tilde{b}_i| > C\varepsilon$ and *small* otherwise. We will fix the constant $C > 1$ later and prove that if $\tilde{b}_0$ and $\tilde{b}_n$ are large and $\mathrm{var}_\varepsilon(\tilde{b}) \cap \{0,1\} \ne \emptyset$ then $\mathrm{var}_\varepsilon(\tilde{b}'), \mathrm{var}_\varepsilon(\tilde{b}'') \in \{\{0\},\{1\}\}$.

Lemma 5. *Let $C \ge 4^{n+1}$ and consider subdivision at $\alpha \in [1/4, 3/4]$. If $\tilde{b}_0$ and $\tilde{b}_n$ are large and positive and $0 \in \mathrm{var}_\varepsilon(\tilde{b})$ then all elements of $\tilde{b}'$ and $\tilde{b}''$ are determinate and positive, i.e.,* $\mathrm{var}_\varepsilon(\tilde{b}') = \mathrm{var}_\varepsilon(\tilde{b}'') = \{0\}$.

Proof. Replace the $\tilde{b}_i$ by modified inputs c_i where $c_i = \tilde{b}_i$ if $\tilde{b}_i$ is determinate and $c_i = 0$ otherwise. This is a change by at most ε. As all entries of the modified de Casteljau triangle $c_{j,i}$ are convex combinations of the inputs, they are all non-negative, and not modified by more than ε. Due to the contribution of c_0 or c_n, resp., any element in the modified output sequences c' and c'' is greater than $4^{-n}C\varepsilon \ge 2\varepsilon$. Thus any element of $\tilde{b}'$ and $\tilde{b}''$ is greater than ε and thus determinate and positive. □

Lemma 6. *Let $C \geq 16^n$ and consider subdivision at $\alpha \in [1/4, 3/4]$. If $\tilde{b}_0$ is large and positive, $\tilde{b}_n$ is large and negative, $\tilde{b}'_n = \tilde{b}''_0$ at the tip of the de Casteljau triangle is large and negative, and $1 \in \mathrm{var}_\varepsilon(\tilde{b})$ then $\mathrm{var}_\varepsilon(\tilde{b}') = \{1\}$ and $\mathrm{var}_\varepsilon(\tilde{b}'') = \{0\}$.*

Proof. As above, we replace all indeterminate elements of $\tilde{b}$ by 0 and denote the elements of the so modified de Casteljau triangle by $c_{j,i}$. The modified input sequence c consists of non-negative followed by non-positive numbers. It is easy to see inductively that all rows of the modified de Casteljau triangle consist of zero or more non-negative elements followed by one or more non-positive elements. Once some row consists entirely of non-positive elements, the same holds for all further rows.

We first prove the claim about c''. The lower tip of the modified triangle is less than $-(C-1)\varepsilon$. A node cannot be less than the minimum of its parents, so there is a path $\mathcal{P}$ of elements less than $-(C-1)\varepsilon$ from row 0 to row n. The elements right of $\mathcal{P}$ are non-positive. Now consider the rightmost element c''_{n-i} in row i of the triangle, for arbitrary i. Go up $0 \leq k \leq i$ times to the left parent until you reach an element of $\mathcal{P}$ in row $i-k$ or end up in row 0 right of the path (with $k = i$). In either case, the last $k+1$ elements of row $i-k$ are non-positive, one of them, say c_*, is less than $-(C-1)\varepsilon$ (namely the path element or c_n), and c''_{n-i} is a convex combination of them. Due to the contribution of c_*, we have $c''_{n-i} < -4^{-k}(C-1)\varepsilon < -2\varepsilon$ and thus $\tilde{b}''_i < -\varepsilon$ holds for all i.

We turn to c'. It begins with $c'_0 > C\varepsilon$ and ends with $c'_n < -(C-1)\varepsilon$. Let

$$i = \min\left\{i \in \{0,\ldots,n\} \ ; \ c'_i \leq 0 \text{ or } |c'_i| \leq |c'_{i-1}|/16\right\} \ . \tag{4}$$

Since c'_n is negative, i exists. By minimality of i, we have for all $j < i$ that $c'_j > 0$ and $c'_j > c'_{j-1}/16 > c'_0/16^j > (C/16^{n-1})\varepsilon > 2\varepsilon$. Thus $c'_0, c'_1, \ldots, c'_{i-1} > 2\varepsilon$.

Next we will show $c'_{i+1}, \ldots, c'_n < -2\varepsilon$. For c'_n, this is already known, so assume $i \leq n-2$. By choice of i, we have $c'_i \leq c'_{i-1}/16$. From $c'_i = \alpha c'_{i-1} + (1-\alpha)c_{i-1,1}$ follows then $c_{i-1,1} = (c'_i - \alpha c'_{i-1})/(1-\alpha) \leq (1-16\alpha)/(16-16\alpha)c'_{i-1}$. This is negative for all $\alpha \in [1/4, 3/4]$, hence $c_{i-1,2} \leq 0$ as well. Now consider

$$\begin{aligned} c'_{i+1} &= \alpha^2 c'_{i-1} + 2\alpha(1-\alpha)c_{i-1,1} + (1-\alpha)^2 c_{i-1,2} \\ &\leq \alpha^2 c'_{i-1} + (\alpha/8)(1-16\alpha)c'_{i-1} = (-\alpha^2 + \alpha/8)c'_{i-1} \ . \end{aligned}$$

The first factor, seen as a function of $\alpha \in [1/4, 3/4]$, takes its maximum $-1/32$ at $\alpha = 1/4$. Hence $c'_{i+1} \leq -(1/32)c'_{i-1} < -(1/32)c'_0/16^{i-1}$. All entries in rows $i+1$ to n are negative and each c'_j from row $i+2$ on receives $\alpha^{j-(i+1)}$ from c'_{i+1}. Thus $c'_j \leq 4^{i-j+1}c'_{i+1} < -4^{i-j+1}32^{-1}16^{1-i}C\varepsilon = -2(C/2^{2i+2j})\varepsilon \leq -2\varepsilon$ for all $j \geq i+1$.

We modified $\tilde{b}$ by at most ε to get c. Hence $\tilde{b}'_0, \ldots, \tilde{b}'_{i-1} > \varepsilon$ and $\tilde{b}'_{i+1}, \ldots, \tilde{b}'_n < -\varepsilon$, and thus $\mathrm{var}_\varepsilon(\tilde{b}') = \{1\}$. □

Let us now fix $C := 16^n$, satisfying the premises of both lemmas for $\alpha \in [1/4, 3/4]$. Based on these lemmas, we formulate the following exact but yet incomplete procedure $\mathit{Descartes}_{\mathrm{approx}}(\tilde{p}, [c,d], \varepsilon)$: Let $\tilde{p}(x) = \sum_{i=0}^n \tilde{b}_i B_i^n[c,d](x)$ be an ε-approximate Bernstein representation of p with respect to the interval $I = [c,d]$. If $\tilde{b}_0$ or $\tilde{b}_n$ is small, abort and signal failure. Otherwise compute $V = \mathrm{var}_\varepsilon(\tilde{b})$, the set of potential values of $\mathrm{var}(b)$. If $V = \{0\}$, return. If $V = \{1\}$, report I as an isolating interval and return. Otherwise, choose a split point $m \in (c + \frac{d-c}{4}, d - \frac{d-c}{4})$ and invoke de Casteljau's algorithm on $\tilde{b}$ to compute approximate Bernstein representations $\tilde{b}'$ and $\tilde{b}''$ for p with respect to intervals $[c,m]$ and $[m,d]$. Call $\mathit{Descartes}_{\text{approx}}(\tilde{p},[c,m],\varepsilon)$ and $\mathit{Descartes}_{\text{approx}}(\tilde{p},[m,d],\varepsilon)$.

Observe that $\mathit{Descartes}_{\text{approx}}$ recurses whenever V contains a value larger than 1. We have shown that whenever $\mathit{Descartes}_{\text{approx}}$ cannot distinguish whether $\operatorname{var}(b)$ is less than two or more than one, this branch of the computation ends in the next recursion step, be it because the tip of the de Casteljau triangle is small or because both new coefficient sequences $\tilde{b}'$ and $\tilde{b}''$ have $\operatorname{var}_\varepsilon$ equal to $\{0\}$ or $\{1\}$. We conclude that $\mathit{Descartes}_{\text{approx}}$ applied to an ε-approximation of p always terminates (either successfully or by signalling failure) and that the internal nodes of its recursion tree form a subtree of the (exact) Descartes tree. Moreover, if the algorithm terminates successfully, it has determined isolating intervals for the real roots of p.

How can we guarantee that first and last coefficients are large? Key are the observations that first and last coefficients are the values of our polynomial at the interval endpoints, that a polynomial can be small only close to one of its complex roots (see Section 4.1), and that randomization can keep interval endpoints away from the roots (see Section 4.2). We describe two ways of randomization: a local one that selects each split point at random (procedure $\mathit{Descartes}_{\text{rndL}}$) and a global one that selects split points deterministically but runs the entire procedure on a random translate of our input polynomial (procedure $\mathit{Descartes}_{\text{rndG}}$).

4.1 The Smith bound

We make the link between the complex roots of p and the magnitude of its values through a corollary to the following theorem by Smith [17]. (We state a special case of his result. For its direct proof, see, e.g., [18, Thm. 13].) For a polynomial f, $\operatorname{lcf}(f)$ denotes the absolute value of the leading coefficient (= the coefficient of x^n).

Theorem 7 (Smith bound). *Let g be a polynomial of degree n and let $\xi_1, \ldots, \xi_n$ be pairwise distinct complex numbers. Then for any root z of g there is a ξ_i such that*

$$|z - \xi_i| \le \frac{n\,|g(\xi_i)|}{\operatorname{lcf}(g) \cdot \prod_{j\neq i}|\xi_j - \xi_i|} \,. \tag{5}$$

Corollary 8. *Let f be a square-free polynomial of degree n with complex roots ξ_1 to ξ_n and $\sigma \le \operatorname{sep}(f)$. Let $\tilde{f}(x) = f(x) + e(x)$ be an approximation of f with error term $e(x) = \sum_{i=0}^{n} \varepsilon_i B_i^n[c,d](x)$ where $|\varepsilon_i| \le \varepsilon$ for all i and some fixed $\varepsilon \ge 0$. Let $\gamma \ge 0$ and $z \in [c,d]$. If $|\tilde{f}(z)| \le \gamma$, then there is a root ξ_i of f such that*

$$|z - \xi_i| \le \frac{n\,(\gamma+\varepsilon)}{\operatorname{lcf}(f) \cdot \prod_{j\neq i}|\xi_j - \xi_i|} \le \frac{n\,(\gamma+\varepsilon)}{\operatorname{lcf}(f)\,\sigma^{n-1}} \,. \tag{6}$$

Proof. Let $g(x) = f(x) - f(z)$ so that $\operatorname{lcf}(g) = \operatorname{lcf}(f)$ and $g(z) = 0$. By Theorem 7, there is a root ξ_i of f satisfying (5). From (2), we can deduce $|g(\xi_i)| = |f(z)| \le |\tilde{f}(z)| + \varepsilon$. $\square$

4.2 Algorithm $Descartes_{\mathrm{rndL}}$

We obtain $Descartes_{\mathrm{rndL}}$ from $Descartes_{\mathrm{approx}}$ by specifying the choice of split point. In each recursive call, we select the split point m as $m = \alpha c + (1-\alpha)d$ with $\alpha = u/K$, $K = 2^{\lceil 5+\log n\rceil}$, and $u \in \{K/4, K/4+1, \ldots, 3K/4\}$ chosen uniformly at random.

We will show that for at least seven eighth of the possible values of u, m has distance at least $L := L(\varepsilon) := n(C+1)\varepsilon/(\mathrm{lcf}(p)\,\mathrm{sep}(p)^{n-1})$ from every root of p. By the contrapositive of Corollary 8 applied with $\gamma = C\varepsilon$, this guarantees that the approximate value of p at m is greater than $C\varepsilon$ in absolute value. Consider a fixed root ξ of p. Any two adjacent potential split points have distance $(d-c)/K$ and hence there are at most $\lceil 2L/((d-c)/K)\rceil$ values of u for which the distance of m and ξ is less than L. Thus all n roots of p exclude at most $n + 2LKn/(d-c)$ values of u. Since $d - c > \mathrm{sep}(p)/8$ by Corollary 3, this is less than $K/16$ and hence at most one eighth of the possible values for u if $n + 16LKn/\mathrm{sep}(p) \le K/16$ or $16L/\mathrm{sep}(p) \le 1/(16n) - 1/K$. Since $K \ge 32n$, this is fulfilled if $16L/\mathrm{sep}(p) \le 1/(32n)$ or

$$\varepsilon \le \frac{\mathrm{lcf}(p)\cdot \mathrm{sep}(p)^n}{512n^2(C+1)} \,. \tag{7}$$

However, $\mathrm{sep}(p)$ is unknown. Hence we maintain an estimate s for $\mathrm{sep}(p)$. We initialize s a negative power of two, to be specified later, and double $\log(1/s)$, i.e., replace s by s^2, whenever we have indication that s is still too big. For fixed s, we choose ε satisfying (7) by setting $\log(1/\varepsilon) = \lceil n\log(1/s) + 4n + 2\log n + 10\rceil = O(n\log(1/s))$; this assumes $\mathrm{lcf}(p) \ge 1$. We use two indicators for s being too big: First, we stop when the recursion depth exceeds the bound $D(s)$ from Corollary 3 by more than 1. Second, we call a choice of u and hence m a failure if the last coefficient of the resulting $\tilde{b}'$ (= first coefficient of $\tilde{b}''$) is small. Whenever a choice of u fails, we repeat it. We keep global counters of all choices and failed choices. Whenever the fraction of failed choices is more than half and we have tried at least twelve times, we stop, double $\log(1/s)$ and start over. Once $s \le \mathrm{sep}(p)$, the bound on the recursion depth is no longer a constraint, and the probability of a restart is less than 1/8. To see this, notice that more than half of r random choices failing has probability at most $\binom{r}{\lceil r/2\rceil}(1/8)^{\lceil r/2\rceil} \le 2^{-r/2}$, and as we try at least twelve times, the probability of restart is at most $\sum_{r\ge 12}(2^{1/2})^{-r} \le 1/8$.

Initialization. Let $q(x) = \sum_{i=0}^{n} q_i x^i$ be a square-free polynomial in power representation normalized to $q_n \in [1,4)$. (The obvious normalization would be $q_n \in [1,2)$, but with inexact data we need to avoid boundary cases.) We view the coefficients as infinite bitstrings. Let τ be the maximum number of bits before the binary point in any coefficient. All roots of q are bounded by $1 + \max_i |q_i|$ in absolute value (Cauchy bound, [19, Lemma 6.7]) and hence are contained in the open disc of radius $M := 2^{\tau+1}$ about the origin of the complex plane. In particular, the real roots of q are contained in the interval $(-M, +M)$. Let $p(x) = q(4Mx - 2M)/(4M)^n$. Then p has its real roots in $(1/4, 3/4)$ and hence the first and last coefficient of its Bernstein representation with respect to $[0,1]$ are large, $\mathrm{sep}(p) = \mathrm{sep}(q)/2^{\tau+3}$, and $\mathrm{lcf}(p) = \mathrm{lcf}(q)$.

We want to compute an $\varepsilon/2$-approximate Bernstein representation of p with respect to $[0,1]$. (Halving ε is motivated later on.) We compute in fixed-point notation with

$\log(1/\delta)$ bits after the binary point; δ to be determined later. Addition of two such numbers and multiplication with an integer can be done exactly. We start from approximations of the q_i's with error at most δ, compute p in power basis and then convert to Bernstein representation. We have

$$p(x) = \sum_j p_j x^j = q(4Mx - 2M)/(4M)^n = (4M)^{-n} \sum_{0 \le j \le n} x^j \sum_{j \le i \le n} \binom{i}{j} (-2)^{i+j} M^i q_i \ .$$

The factor $\binom{i}{j}$ is an integer less than 2^n and $(4M)^{-n} 2^{i+j} M^i = 2^{i+j+i(\tau+1)-n(\tau+3)}$ is a non-positive power of two (since $i \le n$ and $j \le n$). Hence the p_j's have $O(\tau + n)$ bits before the binary point and error at most $(n+1)2^n\delta$. (To see this, note that the error in each $\binom{i}{j} q_i$ is at most $2^n\delta$; the shifts do not increase the error; each p_j is the sum of at most $n+1$ terms; and the additions do not introduce errors.) We have $p_n = q_n \in [1,4)$. The Bernstein representation of p with respect to $[0,1]$ is given (see [14, 2.8]) by:

$$\sum_{l=0}^{n} b_l B_l^n[0,1](x) = \sum_{l=0}^{n} B_l^n(x) \sum_{k=0}^{l} \frac{\binom{l}{k}}{\binom{n}{k}} p_k = \sum_{l=0}^{n} B_l^n(x) \sum_{k=0}^{l} \frac{l(l-1)\cdots(l-k+1)(n-k)!}{n!} p_k \ .$$

To avoid divisions other than by powers of two, we compute $n!b_l$ instead b_l and later scale as to bring the leading coefficient back to $[1,4)$. We have $l(l-1)\cdots(l-k+1)(n-k)! \le n! \le 2^{n \log n}$. The leading coefficient of $\sum_l n! b_l B_l(x)$ is in $[1,4)n!$. It is brought back into $[1,4)$ by shifting all coefficients to the right by $r \in \log(n! q_n) \pm 1 = O(n \log n)$ bits. The results are the coefficients $\tilde{b}_l$ for use in $Descartes_{\mathrm{rndL}}$. The error in each $\tilde{b}_l$ is bounded by the sum of the errors of the p_j's multiplied by a small power of two accounting for the discrepancy between $1/n!$ and 2^{-r}. Hence the error is at most $(n+1)^2 2^{n+3}\delta$. We want this to be at most $\varepsilon/2$ and therefore choose δ as largest power of two such that $(n+1)^2 2^{n+3}\delta \le \varepsilon/2$. Thus $\log(1/\delta) = \log(1/\varepsilon) + O(n) = O(n\log(1/s))$.

The conversion requires $O(n^2)$ additions and multiplications. The coefficients have at most $O(\tau + n + \log(1/\delta)) = O(\tau + n\log(1/s))$ bits. Multiplications are with numbers of at most $n \log n$ bits and hence the bit complexity of conversion (using high-school multiplication) is $O(n^2 n \log n(\tau + n\log(1/s))$. This is $O(n^4 \log n \log(1/s))$, since we will choose $s_0 = 2^{-\max(1,\tau/n)}$.

Complexity Analysis. The estimate $\log(1/s)$ runs through the values $2^i \log(1/s_0)$, $i \ge 0$. For each s, we set ε such that $\log(1/\varepsilon) = O(n \log(1/s)) = O(n 2^i \log(1/s_0))$. We compute an $\varepsilon/2$-approximate Bernstein representation of p with respect to $(0,1)$ and call $Descartes_{\mathrm{rndL}}(p,(0,1),\varepsilon)$. Its recursion tree has depth at most $D := D(s) + 1 = O(\log(1/s))$ (Corollary 3), and there are at most n nodes on each level (Prop. 4). We perform the arithmetic with $\log(1/\delta_2)$ bits after the binary point; δ_2 fixed as follows. In each node, there are $n(n+1)/2 = O(n^2)$ operations, namely averages, that each introduce an additional error δ_2 but do not add bits before the binary point. The accumulated error in any value is at most $\varepsilon/2 + \frac{n(n+1)}{2} D\delta_2$. With $\frac{n(n+1)}{2} D\delta_2 \le \varepsilon/2$ or $\log(1/\delta_2) = \log(1/\varepsilon) + \log(n(n+1)) + \log(D) = O(n\log(1/s))$, any error is at most ε.

Each averaging operation has bit complexity dominated by the multiplication cost $O(\log n\,(\tau + n + n\log(1/s))) = O(n \log n \log(1/s))$ and hence for any fixed s, the $O(n^3 D)$ operations in total have bit complexity $O(n^4 D \log n \log(1/s)) = O(n^4 \log n \log^2(1/s))$.

The i-th estimate of $\log(1/s)$ is $2^i \log(1/s_0)$, and running $Descartes_{\mathrm{rndL}}$ for this value of s costs $O(4^i) \cdot h(n,s_0)$ where $h(n,s) = O(n^4 \log n \log^2(1/s))$. Let $i_1 \geq 0$ be minimal such that $s_1 := 2^{i_1} \log(1/s_0) \geq \log(1/\operatorname{sep}(p))$. The probability that the i-th estimate of s is used is at most $(1/8)^{i-i_1}$ since a call of $Descartes_{\mathrm{rndL}}$ fails with probability less than $1/8$ whenever $s \leq \operatorname{sep}(p)$. Hence the expected overall bit complexity is

$$\Big(\sum_{i \leq i_1} 4^i + \sum_{i > i_1} (1/8)^{i-i_1} 4^i\Big) \cdot h(n,s_0) = O(4^{i_1}) \cdot h(n,s_0) = h(n,s_1) \ .$$

This means that, asymptotically, the last iteration alone determines the expected cost. Since $\log(1/s_1) = O(\tau/n + \log(1/\operatorname{sep}(p)))$ and $\log(1/\operatorname{sep}(p)) = O(\tau + \log(1/\operatorname{sep}(q))$, we have thus shown

Theorem 9. *Let $q = \sum_{i=0}^{n} q_i x^i$ with $|q_n| \geq 1$ and $|q_i| \leq 2^\tau$ for all i. The expected bit cost of $Descartes_{\mathrm{rndL}}$ to isolate the real roots of q is is $O(n^4 \log n(\tau + \log(1/\operatorname{sep}(q)))^2)$.*

4.3 Algorithm $Descartes_{\mathrm{rndG}}$

Let us now consider another variant of *Descartes* in which we fix $\alpha := 1/2$ globally, meaning that always the interval midpoint is chosen as split point. To keep them away from the roots of p, we replace $p = p_0$ by a random translate $p_\beta(x) = p(x+\beta)$.

We can tighten the recursion depth bound of Corollary 3 to $D(\sigma) := \lfloor \log(1/\sigma) + 2 \rfloor$; and Lemma 6 already holds for $C := 8^n$. (For a proof, replace 8 by 16 in Eq. (4).)

As before, we maintain an estimate s of $\operatorname{sep}(p)$, starting from $s_0 := 2^{-\max(1,\tau/n)}$. The interval $(0,1)$ decomposes into $2^D = 4/s$ intervals of width $s/4$ which we call *elementary intervals*. Any interval I considered by $Descartes_{\mathrm{rndG}}$ is a union of elementary intervals (modulo the left endpoint). We call the endpoints of all elementary intervals the *elementary endpoints*. Our goal is to choose β such that any root of p_β has distance greater than L (as defined in Section 4.2) from both endpoints of the elementary interval containing it, so that the approximate value of p_β at all elementary endpoints is greater than $\gamma = C\varepsilon$ in absolute value, so that $Descartes_{\mathrm{rndG}}$ is successful. We choose β from

$$\left\{ \frac{u}{K} \cdot \frac{s}{4} \,\middle|\, u \in \{0,1,\ldots,K-1\} \right\} \tag{8}$$

uniformly at random, where the integer K is still to be determined. Consider a fixed root ξ_i of p_0. It excludes at most $1 + (2L/(s/4))K$ values of u. Thus all n roots of p_0 exclude at most $n + (8L/s)nK$ values of u. We want that at least $7/8$ of the values are good and hence require $n + (8L/s)nK \leq K/8$ or $8L/s \leq 1/(8n) - 1/K$. With $K := 2^{\lceil 4+\log n \rceil}$ we obtain the condition $8L(\varepsilon)/s \leq 1/(16n)$, or equivalently,

$$\varepsilon \leq \frac{\operatorname{lcf}(p_0) \cdot s^n}{128n^2(C+1)} \ . \tag{9}$$

We set $\log(1/\varepsilon) = \lceil n\log(1/s) + 3n + 2\log n + 8 \rceil = O(n\log(1/s))$ and limit the recursion depth to $D := D(s) + 1$. Whenever $Descartes_{\mathrm{approx}}$ aborts with failure, we double $\log(1/s)$ and start again, making a fresh choice for β. Once $s \leq \operatorname{sep}(p)$, every further call to $Descartes_{\mathrm{approx}}$ has success probability at least $7/8$.

Initialization. This is very similar to the initialization of *Descartes*$_{\mathrm{rndL}}$. One can compute p_β from p using only addition, multiplication by integers, and bit shifts, without introducing errors. The complexity stays at $O(n^4 \log n \log(1/s))$.

Complexity analysis. Also very similar to *Descartes*$_{\mathrm{rndL}}$. However, the bit complexity of *Descartes*$_{\mathrm{approx}}$ is $O(n^4 \log^2(1/s))$ and hence better by a factor of $\log n$, since $\alpha = u/K$ with numerator length $O(\log n)$ is replaced by $\alpha = 1/2$. This shows

Theorem 10. *Let $q = \sum_{i=0}^{n} q_i x^i$ with $|q_n| \geq 1$ and $|q_i| \leq 2^\tau$ for all i. The expected bit cost of Descartes$_{\mathrm{rndG}}$ to isolate the real roots of q is is $O(n^4(\tau + \log(1/\operatorname{sep}(q)))^2)$.*

5 Some Experiments

We are in the process of conducting experiments[3] on various classes of polynomials with our algorithms *Descartes*$_{\mathrm{rndL}}$ and *Descartes*$_{\mathrm{rndG}}$, implemented in fixed-point arithemtic as detailed in their respective complexity analyses. Some preliminary findings are as follows.

The random choice of the bisection parameter α in *Descartes*$_{\mathrm{rndL}}$ is quite costly in practice when compared to bisection at $\alpha = 1/2$, because the latter does not require multiplication with weights in the averaging step of the de Casteljau algorithm (addition and shift suffice). Hence we suggest to try small denominators of α first, starting with $\alpha = 1/2$, and increasing them in each try up to the original value K. The resulting variant of *Descartes*$_{\mathrm{rndL}}$, called *Descartes*$^{\mathrm{bias}}_{\mathrm{rndL}}$, can be one order of magnitude faster in practice.

Compared to the Bernstein Descartes method implemented with exact integer coefficients and subdivision at $\alpha = 1/2$, *Descartes*$^{\mathrm{bias}}_{\mathrm{rndL}}$ and *Descartes*$_{\mathrm{rndG}}$ tend to be faster for long coefficients.

As expected, experiments also indicate that the running time of our methods, unlike approaches with exact arithmetic, is mostly unaffected by the irrationality of coefficients.

References

1. Collins, G.E., Akritas, A.G.: Polynomial real root isolation using Descartes' rule of signs. In Jenks, R.D., ed.: Proceedings of the 1976 ACM Symposium on Symbolic and Algebraic Computation, ACM Press (1976) 272–275
2. Uspensky, J.: Theory of Equations. McGraw-Hill (1948)
3. Krandick, W.: Isolierung reeller Nullstellen von Polynomen. In Herzberger, J., ed.: Wissenschaftliches Rechnen. Akademie-Verlag (1995) 105–154
4. Rouillier, F., Zimmermann, P.: Efficient isolation of a polynomial's real roots. J. Computational and Applied Mathematics **162** (2004) 33–50
5. Lane, J.M., Riesenfeld, R.F.: Bounds on a polynomial. BIT **21** (1981) 112–117
6. Mourrain, B., Vrahatis, M.N., Yakoubsohn, J.C.: On the complexity of isolating real roots and computing with certainty the topological degree. J. Complexity **18** (2002) 612–640
7. Basu, S., Pollack, R., Roy, M.F.: Algorithms in Real Algebraic Geometry. Springer (2003)

[3] See `http://www.mpi-inf.mpg.de/˜sschmitt/Descartes`.

8. Mourrain, B., Rouillier, F., Roy, M.F.: Bernstein's basis and real root isolation. Rapport de recherche 5149, INRIA-Rocquencourt (2004) http://www.inria.fr/rrrt/rr-5149.html.
9. Collins, G.E., Johnson, J.R., Krandick, W.: Interval arithmetic in cylindrical algebraic decomposition. J. Symbolic Computation **34** (2002) 143–155
10. Henrici, P.: Applied and Computational Complex Analysis. Volume 1. Wiley (1974)
11. Weyl, H.: Randbemerkungen zu Hauptproblemen der Mathematik II: Fundamentalsatz der Algebra und Grundlagen der Mathematik. Math. Z. **20** (1924) 131–152
12. Pan, V.: Solving a polynomial equation: Some history and recent progress. SIAM Review **39** (1997) 187–220
13. Pan, V.: Univariate polynomials: Nearly optimal algorithms for numerical factorization and root finding. J. Symbolic Computation **33** (2002) 701–733
14. Prautzsch, H., Boehm, W., Paluszny, M.: Bézier and B-Spline Techniques. Springer (2002)
15. Hoschek, J., Lasser, D.: Fundamentals of computer aided geometric design. A K Peters (1996) Translation of: Grundlagen der geometrischen Datenverarbeitung, Teubner, 1989.
16. Krandick, W., Mehlhorn, K.: New bounds for the Descartes method. Technical report, Drexel University, Dept. of Computer Science (2004) To appear in J. Symbolic Computation, http://www.cs.drexel.edu/static/reports/DU-CS-04-04.html.
17. Smith, B.T.: Error bounds for zeros of a polynomial based upon Gerschgorin's theorems. J. ACM **17** (1970) 661–674
18. Bini, D.A., Fiorentino:, G.: Design, analysis, and implementation of a multiprecision polynomial rootfinder. Numerical Algorithms **23** (2000) 127–173
19. Yap, C.K.: Fundamental Problems of Algorithmic Algebra. Oxford University Press (2000)