On the Complexity of the Descartes Method when using Approximate Arithmetic

Michael Sagraloff

MPI for Informatics, Saarbrücken, Germany

Abstract

In this paper, we introduce a variant of the Descartes method to isolate the real roots of a square-free polynomial $F(x) = \sum_{i=0}^{n} A_i x^i$ with arbitrary real coefficients. It is assumed that each coefficient of F can be approximated to any specified error bound. Our algorithm uses approximate arithmetic only, nevertheless, it is certified, complete and deterministic. We further provide a bound on the complexity of our method which exclusively depends on the geometry of the roots and not on the complexity of the coefficients of F. For the special case, where F is a polynomial of degree n with integer coefficients of maximal bitsize τ, our bound on the bit complexity writes as $\tilde{O}(n^3\tau^2)$. Compared to the complexity of the classical Descartes method from Collins and Akritas (based on ideas dating back to Vincent), which uses exact rational arithmetic, this constitutes an improvement by a factor of n. The improvement mainly stems from the fact that the maximal precision that is needed for isolating the roots of F is by a factor n lower than the precision needed when using exact arithmetic.

Key words: Root isolation, Descartes method, subdivision methods, numerical computation, complexity bounds, approximate coefficients

1. Introduction

Computing the roots of a univariate polynomial can be considered as one of the fundamental problems in computational algebra, and numerous approaches have been proposed in the last decades to solve this problem. In this paper, we focus on the problem of *isolating the real roots* of a square-free polynomial $F \in \mathbb{R}[x]$ with arbitrary real coefficients. More precisely, given approximations of the coefficients of F to an arbitrary precision, we aim to compute disjoint intervals $J_1, \ldots, J_m$ such that each J_i contains exactly one root of F and such that their union contains all real roots of F. For polynomials with integer coefficients, the so-called Descartes method (or

Email address: msagralo@mpi-inf.mpg.de (Michael Sagraloff).

"Vincent-Collins-Akritas" method) [1], first introduced by Collins and Akritas [10], constitutes one of the simplest and most efficient algorithms. In order to better understand the contribution of this paper, we briefly review the algorithm: It starts with an interval $\mathscr{I}$ containing all real roots of F and recursively proceeds as follows: For an interval $I = (a,b) \subset \mathscr{I}$, Descartes' Rule of Sign is used to test I for roots of F. If it yields that the number m of roots contained in I equals zero, I is discarded. If it yields that $m = 1$, then I is stored as an isolating interval. In all other cases, I is subdivided into two equally sized subintervals $I_\ell := (a, m(I))$ and $I_r := (m(I), b)$, where $m(I)$ denotes the midpoint of I. For a polynomial F of degree n with integer coefficients of bit-size τ, the Descartes method induces a recursion tree of size $O(n(\tau + \log n))$, where the latter bound has shown to be optimal [15]. For Descartes' Rule of Signs, we need to compute the polynomial [2]

$$F_{I,\mathrm{rev}}(x) := (x+1)^n \cdot F\left(\frac{ax+b}{x+1}\right). \tag{1.1}$$

Using asymptotically fast Taylor shifts [16, 45, 39], the cost for this computation is bounded by

$$\tilde{O}(n^2(\log^+ \max(|a|,|b|) + \log^+ |b-a|^{-1})) = \tilde{O}(n^3\tau) \tag{1.2}$$

bit operations, [3] where we define $\log^+(x) := \log \max(2,|x|) \geq 1$ for all $x \in \mathbb{C}$ and $\log := \log_2$. The bound in (1.2) follows from the fact that we have to perform $\tilde{O}(n)$ arithmetic operations and that $F_{I,\mathrm{rev}}$ has rational coefficients of bit-size $O(n(\log^+ \max(|a|,|b|) + \log^+ |b-a|^{-1})) = \tilde{O}(n^2\tau)$. Multiplication of the bound on the recursion tree and the bound (1.2) on the bit complexity for the computations at each node yields the bound $\tilde{O}(n^4\tau^2)$ on the overall bit complexity of the Descartes method.

The advantages of the Descartes method are its simplicity and that the size of the recursion tree adapts well to the geometric locations of the roots, that is, the recursion tree becomes large if and only if some of the roots are clustered. A disadvantage of the Descartes method is that the exact computation of the polynomials $F_{I,\mathrm{rev}}$ needs a precision of $\tilde{\Theta}(n^2\tau)$ in the worst case, whereas separating the roots from each other needs only $\tilde{O}(n\tau)$ bits. In fact, the binary representation of the endpoints of all isolating intervals returned by the algorithm needs no more than $\tilde{O}(n\tau)$ bits. This brings up the question whether approximate computation of the polynomials $F_{I,\mathrm{rev}}$ yields any improvement with respect to the precision demand during the computation and, thus, also with respect to the bit complexity of the Descartes method. This question has been addressed in a series of previous papers: Johnson and Krandick [19] introduced a hybrid method that uses interval arithmetic based on floating point computation (up to a certain fixed precision) to compute the polynomials $F_{I,\mathrm{rev}}$. This allows to determine the signs of the coefficients of $F_{I,\mathrm{rev}}$ (and, thus, to use Descartes' Rule of Signs) for most of the considered intervals within the subdivision process by using approximate arithmetic, whereas, for the remaining intervals, the method falls back to exact computation. Hence, floating point arithmetic is used as a filter

[1] There exist numerous discussions (e.g. [1]) about whether "Descartes method" is the correct term since Descartes did not introduce any algorithm to isolate the roots but (only) a method to estimate the number of positive roots of a univariate polynomial (i.e. Descartes' Rule of Signs). However, because of the fact that the algorithm from Collins and Akritas (based on ideas dating back to Vincent) exclusively uses this rule as inclusion and exclusion predicate, it is reasonable to name the algorithm after Descartes without using the possessive "s" following his name.

[2] Descartes' Rule of Sign states that the number m of roots contained in I is upper bounded by the number v of sign changes in the coefficient sequence of $F_{I,\mathrm{rev}}$ and that $v \equiv m \bmod 2$. For more details, we refer to Section 2.6.

[3] According to Cauchy's Root Bound (see e.g. [47]), we can assume that $\mathscr{I} \subset (-1-2^\tau, 1+2^\tau)$, and thus $\max(1,|a|,|b|) \leq 1+2^\tau$. In addition, Descartes method does not subdivide intervals of size less than half of the minimal distance between two distinct roots of F (i.e. the separation σ_F of F), and $\log \max(1,\sigma_F) = O(n(\tau+\log n))$; see Section 2.6 for details.

which allows to decrease the precision demand for most intervals, however, no improvement is achieved with respect to worst case bit complexity. Rouillier and Zimmermann [34] modified the latter approach by arbitrarily increasing the working precision at each stage of the algorithm. It is currently one of the fastest algorithms in practice (e.g. the univariate solver in MAPLE is based upon this method), however, no result on the needed precision demand and its computational complexity is known, and we expect that, without further modifications, there is no improvement upon the bound $\tilde{O}(n^4\tau^2)$ in the worst case. There also exist "approximate versions" of the Descartes methods for which complexity results are known: In [14], Eigenwillig et al. proposed a randomized algorithm which is similar to the one from Rouillier and Zimmermann in the sense that it computes interval approximations of the polynomials $F_{I,\mathrm{rev}}$; in fact, the method works with the Bernstein representation and not with the monomial representation of F. However, the main difference is that the subdivision points are randomly chosen in order to avoid unnecessarily large working precisions. The algorithms from [25, 35] are both deterministic, and they both start with a specific rational approximation $\tilde{F}$ of F for which isolating intervals are computed. Eventually, the isolating intervals for F are obtained by enlarging the isolating intervals for $\tilde{F}$. It has been shown that, for integer polynomials, all of the latter methods (i.e. [14, 25, 35]) need $\tilde{O}(n^4\tau^2)$ bit operations to isolate all real roots. In summary, there exists no theoretical proof for the improved efficiency of an "approximate" Descartes method as observed in practice.

Main Results. A main contribution of this paper is to close the above described gap between theory and practice by introducing a modified Descartes method, denoted $\mathbb{R}$ISOLATE, which combines the Descartes and the Bolzano method [9, 38, 46]. More precisely, for discarding intervals that do not contain any root, our method mainly uses Descartes' Rule of Signs, whereas an interval is confirmed to be isolating via a sign-change test at the endpoints of the interval and Rouché's Theorem. $\mathbb{R}$ISOLATE succeeds under guarantee (i.e. the method returns an exact result for any given F) with a working precision bounded by $\tilde{O}(n\tau)$ in the worst case and the size on the recursion tree is bounded by $\tilde{O}(n\tau)$. This eventually yields the improved bound $\tilde{O}(n^3\tau^2)$ for isolating all real roots of F. Before we give more details, we briefly sketch how this improvement is possible: For an interval $I=(a,b)$, let

$$F_I(x) := F(a+(b-a)\cdot x). \tag{1.3}$$

Then, it holds that $F_{I,\mathrm{rev}}(x) = (x+1)^n \cdot F_I(1/(x+1))$. Hence, from an approximation $\tilde{F}_I$ of F_I to $L+n$ bits after the binary point[4], we can directly compute an approximation $\tilde{F}_{I,\mathrm{rev}}(x)$ of $F_{I,\mathrm{rev}}(x)$ to L bits after the binary point (see Lemma 1 (c)). Thus, in essence, we can restrict to the computation of sufficiently good approximations of the polynomials F_I. In Section 2.3, we show that, for an arbitrary approximation of F to $\rho_0 = \tilde{O}(n\tau)$ bits after the binary point, corresponding roots of F and its approximation are almost at the same location with respect to their separations; see Theorem 3 and Appendix 6.2 for a more precise result. We conclude that, for isolating the roots of F, it should also suffice to consider approximations $\tilde{F}_I$ of F_I to ρ_0 bits after the binary point. But how can we compute such approximations $\tilde{F}_I$ in an efficient manner? Let h_0, with $h_0 = \tilde{O}(n\tau)$, denote an upper bound on the depth of the recursion tree induced by the (modified) Descartes method. If we start with an approximation of F to $\rho_0 + 2h_0 = \tilde{O}(n\tau)$ bits after the binary point, then we can recursively compute approximations $\tilde{F}_I$ of F_I such that the approximation error quadruples at most in each bisection step (Lemma 1), and thus each F_I is approximated to at least ρ_0 bits after the binary point. The polynomials $\tilde{F}_I$ have bitsize $\tilde{O}(n\tau)$

[4] More precisely, each coefficient of F_I is approximated to an absolute error of less than 2^{-L-n}

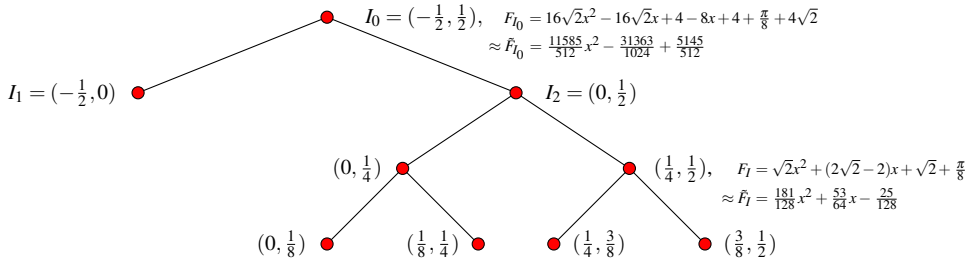

Fig. 1.1. Consider the recursion tree induced by the Descartes method when applied to $F(x) := 16\sqrt{2} \cdot x^2 - 8 \cdot x + \frac{\pi}{4}$ (with roots $x_1 = 0.06\ldots$ and $x_2 = 0.29\ldots$). For each interval $I = (a,b)$ in the subdivision process, we have to compute the polynomial $F_I(x)$. For instance, for $I = (\frac{1}{4}, \frac{1}{2})$, it holds that $F_I(x) = F(\frac{1}{4} + \frac{x}{4}) = \sqrt{2}x^2 + (2\sqrt{2} - 2)x + \sqrt{2} + \frac{\pi}{8}$. We start with an approximation $\tilde{F}_{I_0}$ of F_{I_0} to a certain number $\rho_{I_0} = \rho$ of bits after the binary point. Then, we recursively compute approximations $\tilde{F}_I$ of F_I to ρ_I bits, where ρ_I is updated in each step. Notice that the polynomials $\tilde{F}_I$ do not necessarily correspond to a specific initial approximation G of F, that is, there might exist no polynomial G such that $G_I = \tilde{F}_I$ for all considered I. In the above example, we start with $\tilde{F}_{I_0}(x) = \frac{11585}{512}x^2 - \frac{31363}{1024} + \frac{5145}{512}$ which approximates $F_{I_0}(x) = f(-\frac{1}{2} + x) = 16\sqrt{2}x^2 - 16\sqrt{2}x + 4 - 8x + 4 + \frac{\pi}{8} + 4\sqrt{2}$ to $\rho_{I_0} = 10$ bits. Then, $\tilde{F}_{I_0}(\frac{x}{2})$ and $\tilde{F}_{I_0}(\frac{1}{2} + \frac{x}{2})$ are evaluated and the result is rounded to 9 bits after the binary point. The resulting polynomials are then approximations of $F_{I_1}(x) = F(\frac{1}{2} + \frac{x}{2})$ and $F_{I_2}(x) = F(\frac{x}{2})$ to $\rho_{I_1} = \rho_{I_2} = 8$ bits, respectively; see Lemma 1 for details. In the following bisection steps, we proceed in exactly the same manner. For instance, for the interval $I = (\frac{1}{4}, \frac{1}{2})$, we obtain $\tilde{F}_I(x) = \frac{181}{128}x^2 + \frac{53}{64}x - \frac{25}{128}$ which approximates F_I to $\rho_I = 6$ bits after the binary point.

(instead of $\tilde{O}(n^2\tau)$ for the exact counterpart F_I) which allows to reduce the cost at each node by a factor n; see also Figure 1.1 for an example in the more general setting, where F has arbitrary real coefficients. The main difficulty is that the values h_0 and ρ_0 are not known in advance and that considering worst case bounds for these values (e.g. for integer polynomials) yields an algorithm which might achieve an improved worst case complexity bound but is not practical at all; see also Section 4.4.1 for more details. In contrast, we propose an adaptive algorithm which succeeds with a working precision comparable to the precision that is actually needed for the given input. In addition, the size of the recursion tree directly depends on the actual separations of the roots; cf. the complexity bounds in (1.6) and (1.7) for a more precise result.

In the above considerations, we mainly focused on polynomials with integer coefficients. However, the proposed algorithm $\mathbb{R}$ISOLATE does not only apply to integer polynomials but also to arbitrary square-free polynomials [5]

$$F(x) := \sum_{i=0}^{n} A_i x^i \in \mathbb{R}[x], \qquad \frac{1}{4} \leq |A_n| \leq 1, \tag{1.4}$$

with *real valued* coefficients A_i, where we assume the existence of a coefficient oracle that provides arbitrary good approximations of the coefficients at the cost of reading the approximation. In this setting, the complexity results are exclusively stated in terms of the geometry of the roots and not in the complexity of the coefficients. More precisely, let

- $\xi_1, \ldots, \xi_n \in \mathbb{C}$ denote the (complex) roots of F,
- $\Gamma_F := \log^+ \max_i |\xi_i|$ the *logarithmic root bound of* F,

[5] The additional requirement for the leading coefficient A_n yields a simpler overall presentation. Notice that, for general values A_n, we first have to multiply the polynomial F by some 2^t, with $t \in \mathbb{Z}$, such that $2^t \cdot |A_n|$ is contained in $[1/4, 1]$.

- $\sigma_i := \sigma(\xi_i, F) := \min_{j \neq i} |\xi_i - \xi_j|$ the *separation of the root* ξ_i,
- $\sigma_F := \min_i \sigma_i$ the *separation of* F, and
- $\Sigma_F := \sum_{i=1}^{n} \log^+ \sigma_i^{-1}$, (1.5)

then $\mathbb{R}$ISOLATE induces a recursion tree of size

$$\tilde{O}(n\Gamma_F + \Sigma_F), \tag{1.6}$$

and it needs

$$\tilde{O}(n(n\Gamma_F + \Sigma_F)^2) \tag{1.7}$$

bit operations to isolate all real roots of F. The coefficients of F have to be approximated to $\tilde{O}(n\Gamma_F + \Sigma_F)$ bits after the binary point. We remark that the bound in (1.7) factorizes into the bound (1.6) on the size of the recursion tree, the precision $\tilde{O}(n\Gamma + \Sigma_F)$ to carry out the computations, and a factor $\tilde{O}(n)$ for the number of arithmetic operations needed to process an interval I in the recursion tree. Furthermore, for a polynomial F with integer coefficients of bit size τ or less, we can first divide F by its leading coefficient (to meet the requirements in (1.4)) and then apply $\mathbb{R}$ISOLATE to the polynomial $F/|A_n|$. For this special case, the bounds on the size of the recursion tree and the bit complexity simplify to $\tilde{O}(n\tau)$ and $\tilde{O}(n^3\tau^2)$, respectively, because $\Gamma_F = O(\tau)$, $\log \mathrm{Mea}(F) = O(\tau + \log n)$, and $\Sigma_F = \tilde{O}(n\tau)$; see also Appendix 6.2.

Related Work. The literature on root finding mainly distinguishes between *numerical* methods that use approximate computation and methods that use *exact arithmetic*. Many numerical algorithms (e.g. based on Newton-Raphson iteration, the Weierstrass-Durand-Kerner method, (inverse) power iteration, Eigenvalue computation, etc.) are widely used and effective in practice [6] but lack a guarantee on the global behavior. A prominent example is the Weierstrass-Durand-Kerner method, where there is still no proof known that the method converges for arbitrary given starting values. For a more detailed discussion, we refer to [32]. In parallel, there is a steady ongoing research on subdivision algorithms which perform rational operations on the input coefficients. Algorithms of the latter kind are the Descartes method (e.g. [10, 13, 34]), the Bolzano method [9, 38], the Sturm method [11, 46], or the continued fraction method [2, 24, 41, 44]. [7] Many of these methods have been integrated into computer algebra systems, and experiments have shown their practical evidence [18, 34]. In addition, their computational complexity has been well-studied [11, 15, 38, 44]. Current experimental data shows that an approximate variant of the Descartes method [34] performs best for most polynomials, whereas, for some particular hard instances (e.g. Mignotte polynomials), the continued fraction approach is more efficient.

From a theoretical complexity point of view, the first benchmark was set by A. Schönhage [39] in 1982. He combines a newly introduced concept denoted *splitting circle method* with techniques from numerical analysis (Newton iteration, Graeffe's method, discrete Fourier transforms) and fast algorithms for polynomial and integer multiplication. With respect to the benchmark problem (i.e. isolating all roots of a polynomial F of degree n with integer coefficients of bit size τ or less), his method achieves the bit complexity bound $\tilde{O}(n^3\tau)$. Pan and others [32, 33] gave theoretical improvements which yield record bounds with respect to bit complexity and arithmetic complexity. In particular, [33, Theorem 2.1.1] implies that isolating all complex roots

[6] E.g. MPSOLVE [8] is a highly efficient implementation of the Aberth-Ehrlich method.

[7] The literature on root solving is extensive, hence, we decided to restrict to a selection of representative papers and refer the reader to the references given therein.

of F needs no more than $\tilde{O}(n^2\tau)$ bit operations. Very recent work [26] turns Pan's factorization algorithm into a root isolation method that achieves a bit complexity bound which adapts directly to the geometry of the roots. That is, similar to the bound in (1.7), the bound exclusively depends on the absolute values and the pairwise distances between roots. However, the main drawback of the asymptotically fast algorithms above is that they are rather involved and difficult to implement. In fact, Pan's method has not been implemented, whereas Schönhage's method has not proven to be efficient in practice so far; see [17] for a "proof of concept" implementation of the splitting circle method within the Computer Algebra system Pari/GP.

All of the above mentioned exact subdivision algorithms (i.e. Sturm, Bolzano, Descartes, or the continued fraction method) need $\tilde{O}(n^4\tau^2)$ bit operations to isolate all real roots of F, thus they lag behind the (asymptotically) fastest method by three magnitudes. In a very recent work [37], we introduced a variant of the Descartes method which uses Newton iteration to speed up convergence. The method exclusively performs exact rational arithmetic and has bit complexity $\tilde{O}(n^3\tau)$ which is still by one magnitude worse than the method from Pan. When compared to other exact subdivision methods, the improvement in [37] mainly stems from the fact that it achieves quadratic convergence in most iterations which yields a recursion tree of almost optimal size. In contrast, the improvement with respect to bit complexity (i.e. from $\tilde{O}(n^4\tau^2)$ to $\tilde{O}(n^3\tau^2)$) as achieved by the algorithm $\mathbb{R}$ISOLATE in this paper is due to the use of approximate arithmetic with a considerably smaller working precision as needed for the exact counterpart.

A first version [36] of this paper appeared in arXiv in November 2010. Since that time, the algorithm $\mathbb{R}$ISOLATE has been implemented as a core function in MATHEMATICA (see [42]) and the complexity results have been applied in a series of papers (e.g. [20, 21, 37, 42, 43]). In this context, we would also like to remark that our complexity results are already confirmed by experiments [42, 43] showing that the complexity of $\mathbb{R}$ISOLATE is exclusively related to the geometry of the roots. Furthermore, the adaptiveness of our bound has turned out to be very useful in the analysis [21] of an algorithm to compute the topology of an algebraic curve which makes extensive use of amortization. For the future, we expect that there will be a series of further complexity results based on our adaptive complexity bound for real root isolation.

Acknowledgments

A special thank goes to all anonymous reviewers for their constructive and detailed criticism that has helped to improve the quality and exposition of this contribution.

2. Preliminaries

2.1. Notations

In addition to the definitions from (1.4) and (1.5), we define $w(I) := b-a$ the *width*, $m(I) := \frac{a+b}{2}$ the *center*, and $r(I) = \frac{w(I)}{2}$ the *radius* of an interval $I = (a,b)$. Furthermore,

$$I^+ = (a^+,b^+) := (a - \frac{w(I)}{4n}, b + \frac{w(I)}{4n}) \quad \text{and} \quad \tilde{I} = (\tilde{a},\tilde{b}) := (a - \frac{w(I)}{2n}, b + \frac{w(I)}{2n})$$

denote extensions of I by $\frac{w(I)}{4n}$ and $\frac{w(I)}{2n}$ (to both sides), respectively. We will need these intervals for our modified version of the Descartes method as presented in Section 3. For an arbitrary point $m \in \mathbb{C}$ and a positive real value r, we define $\Delta = \Delta_r(m)$ to be the open disk with center m and radius r. $\bar{\Delta}$ and $\bar{I}$ denote the closure of a disk Δ and an interval I, respectively.

2.2. *Scaling the Polynomial*

Instead of isolating the roots of the given polynomial $F(x) = \sum_{i=0}^{n} A_i x^i$ as defined in (1.4), we consider the equivalent task of isolating the roots of a "scaled" polynomial

$$f(x) = \sum_{i=0}^{n} a_i x^i := \sum_{i=0}^{n} (A_i \cdot 2^{i\cdot\Gamma}) \cdot x^i = F(2^\Gamma \cdot x), \tag{2.1}$$

where $\Gamma \in \mathbb{N}$ is an integer approximation of the exact logarithmic root bound $\Gamma_F = \log^+(\max_i |\xi_i|)$ of F such that

$$\Gamma_F + 1 \leq \Gamma \leq \Gamma_F + 8\log n + 1. \tag{2.2}$$

According to Appendix 6.1, we can compute such a Γ with $\tilde{O}(n^2\Gamma_F)$ bit operations from an approximation of F to $\tilde{O}(n\Gamma_F)$ bits after the binary point. From the definition of Γ, it follows that the roots $z_1 := \xi_1 \cdot 2^{-\Gamma}, \ldots, z_n := \xi_n \cdot 2^{-\Gamma}$ of f are contained within the disk $\Delta_{1/2}(0)$. Furthermore, the absolute value of each coefficient a_i is upper bounded by $2^{O(n\Gamma)}$ since $|A_i| \leq \binom{n}{i}\mathrm{Mea}(F) \leq 2^{n+n\Gamma_F} \leq 2^{n\Gamma}$ for all i. We further remark that the separations of corresponding roots of F and f scale by a factor of 2^Γ (i.e. $\sigma(\xi_i, F) = 2^\Gamma \cdot \sigma(z_i, f)$). Thus, we have

$$\Sigma_f = \sum_{i=1}^{n} \log^+ \sigma(z_i, f)^{-1} \leq \Sigma_F + n\Gamma = O(n\Gamma_F + n\log n + \Sigma_F) = \tilde{O}(n\Gamma_F + \Sigma_F). \tag{2.3}$$

2.3. *Approximating Polynomials*

We assume the existence of a coefficient oracle which, for a given $\rho \in \mathbb{N}$, provides approximations of the coefficients of F to ρ bits after the binary point. More precisely, each coefficient A_i is approximated by a binary fraction $\tilde{A}_i = m_i \cdot 2^{-\rho}$ with $m_i \in \mathbb{Z}$ and $|A_i - \tilde{A}_i| \leq 2^{-\rho}$, e.g., $\tilde{A}_i = \mathrm{sign}(A_i) \cdot \lfloor |A_i \cdot 2^\rho| \rfloor \cdot 2^{-\rho}$. We call a polynomial $\tilde{F} \in \mathbb{Q}[x]$ obtained in this way a *ρ-binary approximation of F*. We only consider the cost for reading (i.e. $O(n(n\Gamma_F + \rho))$) but not for computing such an approximation. Notice that, in order to obtain a ρ-binary approximation of the scaled polynomial f, we have to approximate F to $n\Gamma + \rho$ bits after the binary point since the i-th coefficient of F is shifted by $i \cdot \Gamma$ bits.

For an arbitrary polynomial $g(x) := \sum_{i=0}^{m} g_i x^i \in \mathbb{C}[x]$ with complex coefficients and an arbitrary non-negative real number $\mu \in \mathbb{R}_{\geq 0}$, we define

$$[g]_\mu := \left\{ \tilde{g}(x) = \sum_{i=0}^{n} \tilde{g}_i x^i \in \mathbb{C}[x] : |g_i - \tilde{g}_i| \leq \mu \text{ for all } i = 0, \ldots, n \right\}$$

the set of all μ-approximations of g. We remark that, since the coefficients of modulus less than μ can be approximated by zero, a μ-approximation $\tilde{g}$ of g might have lower degree than g.

Example. For $g(x) := \frac{12256}{65589}x^{10} - 2x^2 + \frac{1}{243}x - \frac{9}{16}$, the polynomial $\tilde{g}(x) := \frac{11}{64}x^{10} - 2x^2 - \frac{9}{16}$ constitutes a 6-binary approximation and $\tilde{g}(x) := -2x^2 - \frac{3}{4}$ a 2-binary approximation of g.

2.4. *Taylor Shifts*

The following lemma provides error bounds on how the absolute approximation error μ of a polynomial $\tilde{g} \in [g]_\mu$ scales under the transformation $x \mapsto m + \lambda \cdot x$ for some special values for $m \in \mathbb{C}$ and $\lambda \in \mathbb{R}\setminus\{0\}$:

Lemma 1. *For $\mu \in \mathbb{R}_0^+$ and $\tilde{g} \in [g]_\mu$ an arbitrary μ-approximation of a polynomial $g \in \mathbb{C}[x]$ of degree n, it holds that*

(a) $\tilde{g}(\frac{1}{2} + \frac{1}{2} \cdot x) \in [g(\frac{1}{2} + \frac{1}{2} \cdot x)]_{2\mu}$,

(b) $\tilde{g}(-\frac{1}{4n} + (1 + \frac{1}{2n}) \cdot x) \in [g(-\frac{1}{4n} + (1 + \frac{1}{2n}) \cdot x)]_{4\mu}$,

(c) $\tilde{g}(-\frac{1}{2} + x) \in [g(-\frac{1}{2} + x)]_{2^n\mu}$, *and* $\tilde{g}(1 + x) \in [g(1 + x)]_{2^n\mu}$.

Proof. For $\mu(x) := (g - \tilde{g})(x) = \mu_n x^n + \ldots + \mu_1 x + \mu_0$, the absolute value of each coefficient μ_i is bounded by μ. Let $m \in \mathbb{C}$ and $\lambda \in \mathbb{R} \setminus \{0\}$ be arbitrary values, then

$$\mu(m + \lambda x) = \sum_{i=0}^{n} \mu_i (m + \lambda x)^i = \sum_{i=0}^{n} \mu_i \sum_{k=0}^{i} x^k \lambda^k m^{i-k} \binom{i}{k} = \sum_{k=0}^{n} x^k \sum_{i=k}^{n} \mu_i m^{i-k} \lambda^k \binom{i}{k} \tag{2.4}$$

Thus, for $|m| < 1$, the absolute value of the coefficient of x^k is bounded by

$$\mu |\lambda|^k \cdot \sum_{i \geq k} |m|^{i-k} \binom{i}{k} = \mu |\lambda|^k \cdot \sum_{i \geq 0} |m|^i \binom{k+i}{k} = \mu |\lambda|^k \cdot \frac{1}{(1 - |m|)^{k+1}}, \tag{2.5}$$

where we used

$$(1 - |m|)^{-(k+1)} = \sum_{i \geq 0} \binom{-(k+1)}{i} (-1)^i |m|^i = \sum_{i \geq 0} \binom{k+i}{i} |m|^i = \sum_{i \geq 0} \binom{k+i}{k} |m|^i.$$

For $m = \lambda = 1/2$, it follows that the absolute value of all coefficients of $\mu(x)$ is bounded by 2μ. This shows (a). For $m = -\frac{1}{4n}$ and $\lambda = 1 + \frac{1}{2n}$, (2.5) implies that

$$\tilde{g}(-\frac{1}{4n} + (1 + \frac{1}{2n}) \cdot x) \in \left[g(-\frac{1}{4n} + (1 + \frac{1}{2n}) \cdot x) \right]_{\mu \frac{8}{7} \cdot \left(\frac{1+1/(2n)}{1-1/(4n)}\right)^n} \subset \left[g(-\frac{1}{4n} + (1 + \frac{1}{2n}) \cdot x) \right]_{4\mu}$$

because $\frac{8}{7} \cdot \left(\frac{1+1/(2n)}{1-1/(4n)}\right)^n \leq \frac{8^3}{7^3} \cdot \sqrt{e} \leq 4$. Hence, (b) follows. The first part of (c) is also a direct implication of (2.5). The second claim in (c) follows from the computation in (2.4) since μ_i is then ($m = \lambda = 1$) bounded by $\mu \cdot \sum_{i=k}^{n} \binom{i}{k} = \mu \cdot \sum_{i=k}^{n} \binom{i}{i-k} = \mu \cdot \sum_{i=0}^{n-k} \binom{i+k}{i} \leq \mu \cdot \sum_{i=0}^{n-k} \binom{n}{i} \leq 2^n \cdot \mu$. □

2.5. On Sufficiently Good Approximation

In the next step, we derive a bound on how good f has to be approximated by some $\tilde{f}$ such that, for all i, the distance of corresponding roots z_i and $\tilde{z}_i$ of f and $\tilde{f}$ is small with respect to the separation $\sigma(z_i, f)$. There exist general worst-case perturbation bounds (e.g. [40, Thm. 2.7] or [23, Chapter 15]) that apply to polynomials with multiple roots and which only depend on the distance $\|f - \tilde{f}\|_1$ between f and $\tilde{f}$. [8] For polynomials with roots of very large multiplicity, these bounds are nearly optimal. However, they often constitute vast overestimations of the amount of perturbation, in particular, for polynomials with well separated roots. In contrast, we provide a more adaptive, but implicit, bound depending on parameters, such as the separations of the roots and the absolute values of the derivatives at the roots, which can not directly be derived from the coefficients of f (or $\tilde{f}$). However, our algorithm as presented in Section 4 is designed in a way such that it eventually succeeds with a working precision that is related to our adaptive bound. We further remark that our bound cannot be directly derived from the bound in [40, Thm. 2.7] and vice versa.

[8] In the context of real root isolation, the bound in [40, Thm. 2.7] has been used in [25] in order to derive isolating intervals for the roots of a real polynomial $f \in \mathbb{R}[x]$ from corresponding isolating intervals for the roots of a rational approximation $\tilde{f} \in \mathbb{Q}[x]$.

The following considerations are mainly adopted from our studies in [35]. For the sake of comprehensibility, we decided to briefly review the results in this paper as well. We start with the following definition:

Definition 2. For t, with $t \geq 1$, an arbitrary real value and f a polynomial as in (2.1), we define

$$\mu(f,t) := \frac{1}{t} \cdot \min_{i=1,\ldots,n} \left| \frac{\sigma(z_i,f) \cdot f'(z_i)}{8n^2} \right| \tag{2.6}$$

We call a $\rho \in \mathbb{N}$ *sufficiently large with respect to* f if [9]

$$\rho \geq \rho_f := \lceil -\log \mu(f, 64n^2) \rceil. \tag{2.7}$$

Notice that $\rho_f = O(\Sigma_f + \log n - \log |a_n|)$ and $\rho_f = O(\Sigma_F + \log n)$ because of

$$\begin{aligned} \sigma(z_i,f) \cdot |f'(z_i)| = \sigma(z_i,f) \cdot |a_n| \cdot \prod_{j \neq i} |z_i - z_j| \geq \sigma(z_i,f) \cdot |a_n| \cdot \prod_{j \neq i} \sigma(z_j,f) = \\ = \frac{|a_n|}{2^{n\Gamma}} \sigma(\xi_i,F) \prod_{j \neq i} \sigma(\xi_j,F) \geq \frac{1}{4} \cdot 2^{-\Sigma_F}. \end{aligned} \tag{2.8}$$

The following theorem gives an answer to our initial question how good f has to be approximated by some $\tilde{f}$ in order to ensure that corresponding roots stay at almost "the same place" with respect to their separations:

Theorem 3. *Let f be the polynomial as defined in (2.1), $t \geq 1$ and $\tilde{f} \in [f]_{\mu(f,t)}$.*

(a) For all $i = 1, \ldots, n$, the disk

$$\Delta_i := \Delta_{\frac{\sigma(z_i,f)}{tn}}(z_i)$$

contains the root z_i of f and a corresponding root $\tilde{z}_i$ of $\tilde{f}$.

(b) For each $z \in \mathbb{C} \setminus \bigcup_{i=1}^n \Delta_i$, it holds that $|f(z)| > (n+1)\mu(f,t)$.

(c) If $\rho \geq \rho_f$, then each root z_i moves by at most $\frac{\sigma(z_i,f)}{64n^3}$ when passing from f to an arbitrary $\tilde{f} \in [f]_{2^{-\rho}}$. In particular, real roots of f stay real and non-real roots stay non-real. Furthermore, for any $z \in \mathbb{C}$ with $|z - z_i| \geq \frac{\sigma(z_i,f)}{64n^3}$ for all i, it holds that $|f(z)| > (n+1)2^{-\rho_f}$.

Proof. Since all roots of f are contained within $\Delta_{1/2}(0)$, it follows that $\sigma(z_i,f) < 1$ for all i and, thus, each disk Δ_i is completely contained within the unit disk. For an arbitrary point $z \in \partial\Delta_i$ on the boundary of Δ_i, we have

$$\begin{aligned} |f(z)| = |a_n| \prod_{j=1}^n |z - z_j| &= \frac{\sigma(z_i,f)}{tn} \left(\prod_{1 \leq j \leq n, j \neq i} \left| \frac{z - z_j}{z_i - z_j} \right| \right) \cdot |a_n| \cdot \left(\prod_{1 \leq j \leq n, j \neq i} |z_i - z_j| \right) \\ &= \frac{\sigma(z_i,f) \cdot |f'(z_i)|}{tn} \prod_{1 \leq j \leq n, j \neq i} \left| \frac{z - z_j}{z_i - z_j} \right| \geq \frac{\sigma(z_i,f) \cdot |f'(z_i)|}{tn} \prod_{1 \leq j \leq n, j \neq i} \frac{|z_i - z_j| - |z - z_i|}{|z_i - z_j|} \\ &\geq \frac{\sigma(z_i,f) \cdot |f'(z_i)|}{tn} \left(1 - \frac{1}{tn}\right)^{n-1} > \frac{\sigma(z_i,f) \cdot |f'(z_i)|}{2.72 \cdot tn} > (n+1)\mu(f,t). \end{aligned}$$

In addition, since $\tilde{f} \in [f]_{\mu(f,t)}$ and $|z| < 1$, we have $|(f - \tilde{f})(z)| < (n+1)\mu(f,t) < |f(z)|$. Hence, (a) follows from Rouché's Theorem applied to the disks Δ_i and the functions f and $\tilde{f}$. For (b), we remark that f is a holomorphic function on $\mathbb{C} \setminus \bigcup_{i=1}^n \Delta_i$ and, thus, $|f(z)|$ becomes minimal for a

[9] This definition is motivated by our results in Theorem 3 and Section 4.1

point z on the boundary of one of the disks Δ_i. (c) follows directly from (a), (b) and the definition of ρ_f in (2.7). $\square$

We conclude from the last theorem that it suffices to approximate the coefficients of f to ρ, with some $\rho = O(\Sigma_f + \log n - \log|a_n|)$, bits after the binary point to guarantee that each approximation $\tilde{f} \in [f]_{2^{-\rho}}$ has its roots at almost the same location as f.

2.6. The Descartes Method

We first resume some basic facts about the Descartes method for isolating the real roots of a polynomial $f(x) = \sum_{i=0}^{n} a_i x^n \in \mathbb{R}[x]$. Descartes' Rule of Signs states that the number $\mathrm{var}(f)$ of sign changes in the coefficient sequence of f, that is, the number of pairs (i, j) with $i < j$, $a_i a_j < 0$, and $a_{i+1} = \ldots = a_{j-1} = 0$, is not smaller than and of the same parity as the number of positive real roots of f. If $\mathrm{var}(f) = 0$, then f has no positive real root, and if $\mathrm{var}(f) = 1$, f has exactly one positive real root. The rule easily extends to an arbitrary open interval $I = (a, b)$ via a suitable coordinate transformation: The mapping $x \mapsto a + (b-a)x$ maps $(0, 1)$ bijectively onto I, that is, the roots of f in I exactly correspond to those of

$$f_I(x) := f(a + w(I)x) = f(a + (b-a)x) \tag{2.9}$$

in $(0, 1)$. Hence, the composition of $x \mapsto a + (b-a) \cdot x$ and $x \mapsto 1/(1+x)$ constitutes a bijective map from $(0, \infty)$ to I. It follows that the positive real roots of

$$f_{I,\mathrm{rev}}(x) := (1+x)^n f_I\left(\frac{1}{x+1}\right) = (1+x)^n \cdot f\left(\frac{ax+b}{x+1}\right)$$

correspond bijectively to the real roots of f in I. The factor $(1+x)^n$ in the definition of $f_{I,\mathrm{rev}}$ clears denominators and guarantees that $f_{I,\mathrm{rev}}$ is a polynomial. $f_{I,\mathrm{rev}}$ is computed from f_I by reversing the coefficients (i.e. the i-th coefficient is replaced by the $(n-i)$-th coefficient) followed by a Taylor shift by 1 (i.e. $x \mapsto x+1$). We now define $\mathrm{var}(f, I) := \mathrm{var}(f_{I,\mathrm{rev}})$.

Based on Descartes' Rule of Sign, Collins and Akritas introduced a bisection algorithm [10] for isolating the roots of f in an interval I_0 (here, we assume that $I_0 = (-1/2, 1/2)$). We refer the reader to [3, 4, 5, 6, 10, 13] for extensive treatments and references.

VCA. The algorithm requires that the real roots of f in I_0 are simple, otherwise it diverges. In each step, a set $\mathscr{A}$ of active intervals is maintained. Initially, $\mathscr{A}$ contains I_0, and the algorithm stop as soon as A becomes empty. In each iteration, some interval $I \in \mathscr{A}$ is processed; If $\mathrm{var}(f, I) = 0$, then I contains no root of f and we discard I. If $\mathrm{var}(f, I) = 1$, then I contains exactly one root of f and, hence, is an isolating interval for it. We add I to a list $\mathscr{O}$ of isolating intervals. If there is more than one sign change, we divide I at its midpoint $m(I)$ and add the subintervals to the set of active intervals. If $m(I)$ is a root of f, we add the trivial interval $[m(I), m(I)]$ to the list of isolating intervals.

Correctness of the algorithm follows immediately from Descartes' Rule of Signs. Termination and complexity analysis of VCA rest on the following theorem:

[10] Based on the fact that Collins and Akritas used ideas dating back to Vincent (see [5]), the algorithm has been named Vincent-Collins-Akritas method (or VCA for short). In this paper, we use both denotations, that is, VCA and Descartes method, in an interchangeable way.

Theorem 4 ([28, 31]). *For a polynomial $f \in \mathbb{R}[x]$ and an interval $I = (a,b)$, let $v := \mathrm{var}(f,I)$.*

(a) (One-Circle Theorem) If the open disk bounded by the circle centered at $m(I)$ and passing through the endpoints of I contains no root of $f(x)$, then $v = 0$.

(b) (Two-Circle Theorem) If the union of the open disks bounded by the two circles centered at $m(I) \pm i(1/(2\sqrt{3}))w(I)$ and passing through the endpoints of I contains exactly one root of $f(x)$, then $v = 1$.

Proofs of the one- and two-circle theorems can be found in [3, 13, 22, 28, 29, 30, 31]. Theorem 4 implies that no interval I of length $\sigma_f/2$ or less is split. Such an interval cannot contain two real roots and its two-circle region cannot contain any nonreal root. Thus, $\mathrm{var}(f,I) \leq 1$ by Theorem 4. We conclude that the depth of the recursion tree is bounded by $1 + \log \sigma_f^{-1}$. Furthermore, it holds (see [13, Cor. 2.27] or [27, Prop. 3.1] self-contained proofs):

Theorem 5. *Let I be an interval and I_1 and I_2 be two disjoint subintervals of I. Then,*

$$\mathrm{var}(f,I_1) + var(f,I_2) \leq \mathrm{var}(f,I).$$

According to the above theorem, there cannot be more than $n/2$ intervals I with $\mathrm{var}(f,I) \geq 2$ at any level of the recursion. Therefore, the size of the *recursion tree* T_{VCA} is bounded by $n(1+\log \sigma_f^{-1})$. For polynomials with integer coefficients of maximal bitsize τ, it has been shown that $-\log \sigma_f = O(n(\log n + \tau))$, thus, the latter bound writes as $\tilde{O}(n^2\tau)$. However, a more refined argumentation [13] shows that $|T_{\mathrm{VCA}}|$ is even bounded by $\tilde{O}(n\tau)$ which is due to the fact that there are amortization effects over the separations of all roots; see Appendix 6.2.

The computation of $f_{I,\mathrm{rev}}$ at each node of the tree is costly. It is better to store with every interval $I = (a,b)$ the polynomial $f_I(x) = f(a + x\cdot(b-a))$. If I is split at its midpoint $m(I)$ into $I_\ell = (a, m(I))$ and $I_r = (m(I), b)$, the polynomials associated with the subintervals are $f_{I_\ell}(x) = f_I(\frac{x}{2})$ and $f_{I_r}(x) = f_I(\frac{1+x}{2}) = f_{I_\ell}(1+x)$. Also, $f_{I,\mathrm{rev}}(x) = (1+x)^n f_I(\frac{1}{1+x})$. If the coefficients of f are integers (or dyadic fractions) of bitsize τ, then the coefficients grow by n bits in every bisection step. Thus, for a node I of depth h, the bitsize τ_h of the coefficients of f_I is bounded by $\tau_h = \tau + nh$. Hence, using asymptotically fast Taylor shift (see [45, 16]), the number of bit operations needed to compute f_{I_ℓ}, f_{I_r} and $f_{I,\mathrm{rev}}$ from f_I is $\tilde{O}(n(nh+\tau))$. Since the depth of the recursion tree is $\tilde{O}(n\tau)$, each f_I has coefficients of bitsize $\tilde{O}(n^2\tau)$ and, thus, the cost at each node is bounded by $\tilde{O}(n^3\tau)$. Eventually, the total cost for VCA is in $\tilde{O}(n^3\tau)\cdot\tilde{O}(n\tau) = \tilde{O}(n^4\tau^2)$.

3. A Modified Descartes Method

In c computational model, where exact operations on real numbers are assumed to be available at unit costs, the Descartes method can be directly used to isolate the real roots of the polynomial f as defined in (2.1). Namely, in such a model, we can compute the number of sign variations for the polynomial $f_{I,\mathrm{rev}}$ and the sign of f at the midpoint $m(I)$ for each node I of the recursion tree no matter whether f has rational, algebraic, or transcendental coefficients. However, for an actual implementation, these computations turn out to be hard, or even infeasible, in general. Namely, if one of the coefficients of $f_{I,\mathrm{rev}}$ equals zero (e.g. this is the case if one of the endpoints of I is a root of f), then deciding the sign of this coefficient becomes infeasible since we can only ask for approximations of f. The decision problem becomes hard if one of the coefficients has a very small value because, in this case, we have to run our computations with a very large working precision. We further remark that, even for algebraic coefficients (with known algebraic

representation), the decision problem might be hard because this amounts to comparing algebraic numbers of large degree. In order to overcome these issues, we do not consider the original version of the Descartes method but a modified variant which completely avoids such difficult decision problems. More precisely, we will show that our method always succeeds with a working precision comparable to ρ_f. A crucial step in our approach is to replace the inclusion predicate $\mathrm{var}(f,I) = 1$, which is used in the Descartes method to confirm an interval to be isolating, by a predicate used in the Bolzano method.

Section 3.1 resumes some useful results which are adopted from our studies on the Bolzano method [38], whereas, in Section 3.2, our modified Descartes method is formulated.

3.1. The $\mathscr{T}[g,K](\cdot)$-Test: Existence of Roots

For $g \in \mathbb{C}[x]$, $m \in \mathbb{C}$ and positive real values K and r, we consider the following test which has already been introduced in [46] in a less general form: [11]

$$\mathscr{T}[g,K](m,r): \quad \mathbf{t}[g,K](m,r) := |g(m)| - K \sum_{k\geq 1} \left| \frac{g^{(k)}(m)}{k!} \right| r^k > 0. \tag{3.1}$$

In order to simplify notation, we also write $\mathscr{T}[g,K](\Delta)$ or $\mathscr{T}[g,K](I)$ instead of $\mathscr{T}[g,K](m,r)$, where $\Delta = \Delta_r(m)$ is disk or $I = (a,b)$ an interval with midpoint $m = m(I)$ and radius $r = r(I)$. If the polynomial g is fixed and no mix-up is possible, we further omit the "g" and write $\mathscr{T}[K](m,r)$ for $\mathscr{T}[K](m,r)$ and $\mathscr{T}'[K](m,r)$ for $\mathscr{T}[g',K](m,r)$. We mainly use $K = 3/2$. Therefore, whenever the "K" is suppressed (i.e. we write $\mathscr{T}[g](m,r)$ instead of $\mathscr{T}[g,3/2](m,r)$), we consider $K = 3/2$. Before presenting the main technical lemmata, we first summarize the following useful properties of $\mathscr{T}[g,K](\cdot)$:

- If $\mathscr{T}[g,K](m,r)$ holds, then $\mathscr{T}[g,K'](m,r')$ holds for all $K' \leq K$ and all $r' \leq r$.
- For arbitrary values m, r and $\lambda \neq 0$, the tests $\mathscr{T}[g,K](m,r)$ and $\mathscr{T}[g(m+\lambda\cdot x),K](0,\frac{r}{\lambda})$ are equivalent since $\mathbf{t}[g(m+\lambda x),K](0,\frac{r}{\lambda}) = \mathbf{t}[g,K](m,r)$. In particular, for an interval $I = (a,b)$, the test $\mathscr{T}[g_I,K](0,r)$ is equivalent to $\mathscr{T}[g,K](a,r\cdot w(I))$, where $g_I(x) = g(a+w(I)\cdot x)$.
- For $\lambda \in \mathbb{R}^+$, it holds that $\mathbf{t}[g,K](m,r) = \mathbf{t}[\lambda g,K](m,r)\cdot\lambda^{-1}$ and, thus, $\mathscr{T}[g,K](m,r)$ is equivalent to $\mathscr{T}[\lambda g,K](m,r)$. Hence, $\mathscr{T}[(g')_I,K](m,r)$ and $\mathscr{T}[(g_I)',K](m,r)$ are equivalent since $(g_I)' = (g(a+w(I)\cdot x))' = w(I)\cdot(g')_I$.

The $\mathscr{T}[g,K](\cdot)$-test serves as an exclusion predicate but might also guarantee that a certain disk contains at most one root. We refer to [7, Theorem 3.2] for a proof of the following lemma.

Lemma 6. *Consider a disk $\Delta = \Delta_r(m) \subset \mathbb{C}$ and a polynomial $g \in \mathbb{R}[x]$:*
(a) If $\mathscr{T}[K](\Delta)$ holds for some $K \geq 1$, then $\bar{\Delta}$ contains no root of g and

$$\left(1 - \frac{1}{K}\right)\cdot|g(m)| < |g(z)| < \left(1 + \frac{1}{K}\right)\cdot|g(m)|$$

for all z in the closure $\bar{\Delta}$ of Δ.
(b) If $\mathscr{T}'[3/2](\Delta)$ holds, then $\bar{\Delta}$ contains at most one root of g.

[11] In [46], Yakoubsohn introduces a quadtree (Weyl) construction for computing the complex roots of an analytic function, where the test $\mathscr{T}[g,1](\cdot)$ is exclusively used as an exclusion predicate. In [46, Section 9], he also provides bounds on the arithmetic complexity and the precision that is needed by his algorithm to isolate all complex roots of a square-free polynomial f. In particular, the bound on the precision is stated in terms of the degree of f, the absolute value of the roots, and the distance of f to the variety of all polynomials that have a multiple root, and thus, it is similar to our bound.

The $\mathscr{T}'(\cdot)$-test now easily applies as an inclusion predicate:

Corollary 7. *Let $I=(a,b)$ be an interval and $r \geq 1$ such that $\mathscr{T}[g_I'](0,r)$ holds. Then, I contains a root ξ of g if and only if $g(a)\cdot g(b)<0$. In the latter case, the disk $\Delta_{r\cdot w(I)}(a)$ is isolating for ξ.*

Proof. If $\mathscr{T}[g_I'](0,r)$ holds, then $\mathscr{T}[g'](a,r\cdot w(I))$ holds as well according to the above properties of $\mathscr{T}[g,K](\cdot)$. It follows that the disk $\Delta_{r\cdot w(I)}(a)$ and, thus, I contains no root of the derivative g'. Now, since g is monotone on I, it suffices to check for a sign change of g at the endpoints of I. Namely, there exists a root ξ of g in I if and only if $g(a)\cdot g(b)<0$. In case of existence, $\Delta_{r\cdot w(I)}(a)$ is isolating for ξ due to Lemma 6. □

In order to show that the $\mathscr{T}[g'](m,r)$-test in combination with sign evaluation is an efficient inclusion predicate, we give lower bounds on r in terms of σ_g such that the predicate succeeds under guarantee.

Lemma 8. *For g a polynomial of degree n, a disk $\Delta=\Delta_r(m)\subset\mathbb{C}$, an interval $I=(a,b)$ and $I^+=(a-\frac{w(I)}{4n},b+\frac{w(I)}{4n})$, it holds:*

(a) If $r\leq\frac{\sigma_g}{4n^2}$, then $\mathscr{T}(\Delta)$ or $\mathscr{T}'(\Delta)$ holds.

(b) If Δ contains a root ξ of g and $r\leq\frac{\sigma(\xi,g)}{4n^2}$, then $\mathscr{T}'(\Delta)$ holds.

(c) If $\mathrm{var}(g,I^+)>0$ and $\mathscr{T}[g_I'](0,2)$ fails, $\Delta_{2w(I)}(a)$ contains a root ξ of g with $\sigma(\xi,g)<8n^2w(I)$.

(d) If $\mathrm{var}(g',I)>0$ and $\mathscr{T}[g_I](0,1)$ fails, $\Delta_{2nw(I)}(a)$ contains a root ξ of g with $\sigma(\xi,g)<4n^2w(I)$.

Proof. For the proof of (a) and (b), we refer to [35, Lemma 5]. For (c), suppose that $\mathrm{var}(g,I^+)>0$ and $\mathscr{T}[g_I'](0,2)$ does not hold. Then, according to Theorem 4 (a), the disk $\Delta_{r(I^+)}(m(I))\subset\Delta_{2w(I)}(a)$ contains a root ξ of g. With (b), it follows that $2w(I)>\frac{\sigma(\xi,g)}{4n^2}$ and, thus, $\sigma(\xi,g)<8n^2w(I)$. For (d), we first argue by contradiction that the disk $\Delta_{2nw(I)}(a)$ contains a root ξ of g: If $|a-x_i|\geq 2nw(I)$ for all roots x_i of g, then

$$\left|\frac{g^{(k)}(a)}{g(a)}\right| = \left|\sum\nolimits'_{i_1,\ldots,i_k}\frac{1}{(a-x_{i_1})\ldots(a-x_{i_k})}\right| \leq \left(\sum\nolimits_{i=1}^{n}\frac{1}{|a-x_i|}\right)^k \leq \left(\frac{1}{2w(I)}\right)^k,$$

where the prime means that the i_j's ($j=1\ldots k$) are chosen to be distinct. It follows that $\mathscr{T}(a,w(I))$ holds because of $\sum_{k=1}^{n}\left|\frac{g^{(k)}(a)}{g(a)}\right|w(I)^k\leq\sum_{k=1}^{n}2^{-k}<1<\frac{3}{2}$. In addition, Theorem 4 guarantees the existence of a root $\xi'\in\Delta_{r(I)}(m(I))$ of g'. Hence, we have $|\xi-\xi'|<2nw(I)+w(I)<4nw(I)$ which implies $\sigma(\xi,g)<4n^2w(I)$ due to the fact [12, 47] that there exists no root of the derivative g' in $\Delta_{\frac{\sigma(\xi,g)}{n}}(\xi)$. □

3.2. DCM*: A Modified Descartes Algorithm*

We introduce our modified Descartes method DCM (short for "Descartes modified") to isolate the real roots of a polynomial f. We formulate the algorithm in the REAL-RAM model, thus, it still does not directly apply to bitstream polynomials. However, in Section 4.1, we will present a corresponding version DCM$^\rho$ of DCM which resolves this issue; see also Appendix, Algorithm 1 for pseudo-code of DCM.

DCM. DCM maintains a list $\mathscr{A}$ of active nodes and a list $\mathscr{O}$ of isolating intervals, where we initially set $\mathscr{O} = \emptyset$ and $\mathscr{A} := \{(I_0, f_{I_0})\}$, with $I_0 := (-\frac{1}{2}, \frac{1}{2})$. For each active node (I, f_I) from $\mathscr{A}$, we proceed as follows. We first remove (I, f_I) from the list $\mathscr{A}$. Then, we compute the number $v_{I^+} := \mathrm{var}(f, I^+) = \mathrm{var}(f_{I^+,\mathrm{rev}})$ of sign variations for f on the extended interval I^+ (notice that $f_{I^+}(x) = f_I(-\frac{1}{4n} + (1+\frac{1}{2n})x)$ and $f_{I^+,\mathrm{rev}}(x) = (1+x)^n f_{I^+}(\frac{1}{1+x})$). [12] If $v_{I^+} = 0$, we do nothing, that is, I is discarded. If $v_{I^+} \geq 1$, we consider the test $\mathscr{T}[f_I'](0,2)$ which is equivalent to $\mathscr{T}[f'](a, 2w(I))$. If it fails, then I is subdivided into $I_\ell = (a, m(I))$ and $I_r = (m(I), b)$ and we add $(I_\ell, f_{I_\ell}) = (I_\ell, f_I(\frac{x}{2}))$ and $(I_r, f_{I_r}) = (I_r, f_{I_\ell}(x+1))$ to $\mathscr{A}$. Otherwise, we evaluate the sign s of $f(a^+) \cdot f(b^+) = f_{I^+}(0) \cdot f_{I^+}(1)$. If $s < 0$ and I^+ is disjoint from any other interval in $\mathscr{O}$, we add I^+ to $\mathscr{O}$. If $s \geq 0$ or I^+ intersects an interval in $\mathscr{O}$, we do nothing (i.e. I is discarded). The algorithm stops when $\mathscr{A}$ becomes empty.

Theorem 9. *For the polynomial f as defined in (2.1), the algorithm* DCM *terminates and returns a list $\mathscr{O} = \{I_1, \ldots, I_m\}$ of disjoint isolating intervals for* all *real roots of f.*

Proof. If the width $w(I)$ of an interval $I = (a,b)$ is smaller or equal to $\frac{\sigma_f}{8n^2}$, then, according to Theorem 4 and Lemma 8 (c), $\mathrm{var}(f, I^+) = 0$ or $\mathscr{T}[f_I'](0,2)$ holds. Thus, I is not further subdivided. This shows termination of DCM. From our construction and Corollary 7, each interval in $\mathscr{O}$ is isolating for a real root of f and all intervals in $\mathscr{O}$ are pairwise disjoint. It remains to show that, for each real root ξ of f, there exists a corresponding isolating interval in $\mathscr{O}$. Since all roots of f have absolute value bounded by $1/2$ and DCM terminates, there must be an interval $I = (a,b)$ of minimal (positive) length whose closure $\bar{I}$ contains ξ. Since $v_{I^+} > 0$, I cannot be discarded in the first step of DCM. Hence, $\mathscr{T}[f_I'](0,2)$ holds and, thus, f is monotone on I^+. Since I^+ contains the root ξ, we have $f(a^+) \cdot f(b^+) < 0$. It follows that either I^+ is added to the list of isolating intervals or I^+ intersects an interval $J^+ = (c^+, d^+) \in \mathscr{O}$ which has been added to $\mathscr{O}$ before. Let $J = (c,d)$ be the corresponding smaller interval for J^+. Since the $\frac{w(I)}{4n}$-neighborhood of I intersects the $\frac{w(J)}{4n}$-neighborhood of J, the following Lemma 10 shows that one of the disks $\Delta_{2w(I)}(a)$ or $\Delta_{2w(J)}(c)$ contains both intervals I^+ and J^+. Since both, $\mathscr{T}[f_I'](0,2)$ and $\mathscr{T}[f_J'](0,2)$, hold, each of the latter two disks contains at most one root due to Corollary 7. It follows that $J^+ \in \mathscr{O}$ already isolates ξ. $\square$

Lemma 10. [13] *Let $I = (a,b)$ and $J = (c,d)$ be two intervals (not necessarily of equal length) of the form $\left(-\frac{1}{2} + i2^{-h}, -\frac{1}{2} + (i+1)2^{-h}\right)$, where $h \in \mathbb{N}$ and $i \in \{0, \ldots, 2^h - 1\}$. If the $\frac{w(I)}{2n}$-neighborhood $U_{\frac{w(I)}{2n}}(I)$ of I intersects the $\frac{w(J)}{2n}$-neighborhood $U_{\frac{w(J)}{2n}}(J)$ of J, then one of the disks $\Delta_{2w(I)}(a)$ or $\Delta_{2w(J)}(c)$ contains the intervals $(a - w(I), b + w(I))$ and $(c - w(J), d + w(J))$.*

Proof. W.l.o.g., we can assume that $w(J) \geq w(I)$ and, thus, $w(J) = 2^l w(I)$ with an $l \in \mathbb{N}_0$. Let δ denote the distance between I and J. If $\delta = 0$, then $\Delta_{2w(J)}(c)$ contains $(a - w(I), b + w(I))$ and $(c - w(J), d + w(J))$. If $\delta \neq 0$, then $\delta = 2^k w(I)$ with a $k \in \mathbb{N}_0$. Since $U_{\frac{w(I)}{2n}}(I) \cap U_{\frac{w(J)}{2n}}(J) \neq \emptyset$,

---

[12] Remember that the polynomial f_{I^+} is obtained from f by mapping all roots of f in I^+ one-to-one and onto the interval $(0,1)$. When further mapping the roots one-to-one and onto the positive real axis, we obtain $f_{I^+,\mathrm{rev}}$.

[13] Lemma 10 proves a slightly stronger result than necessary for the proof of Theorem 9. The stronger result applies in the proof of Theorem 14 in Section 4.2.

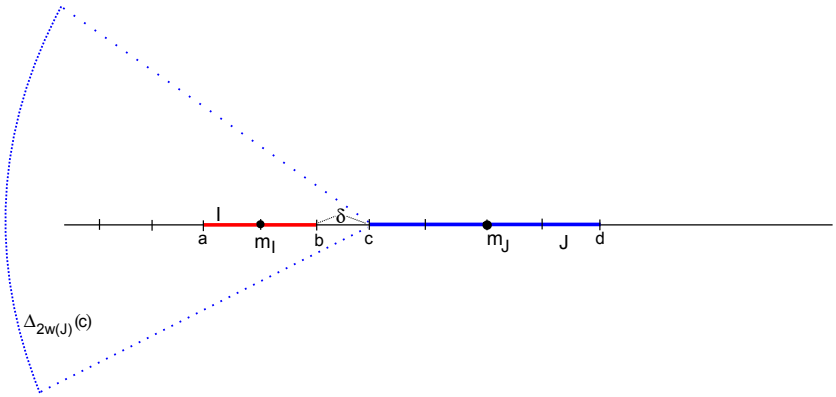

Fig. 3.1. Wlog., we can assume that $w(J) \geq w(I)$. $w(I)$, $w(J)$ and the distance δ between I and J differ by a power of 2. For $\delta = 0$, the disk $\Delta := \Delta_{2w(J)}(c)$ certainly contains $\tilde{I}$ and $\tilde{J}$. If $\delta \neq 0$, then $w(J) \geq 2w(I)$ and $w(J) \geq 4\delta$, hence $\tilde{I}, \tilde{J} \subset \Delta$.

we must have $\frac{w(J)}{2n} > \frac{\delta}{2}$. In particular, we have $\frac{w(J)}{4} > \frac{\delta}{2} = 2^{k-1}w(I)$. Since $w(I)$ and $w(J)$ differ by a power of 2, it follows that $w(J) \geq 2^{k+2}w(I) = 4\delta$ and, thus, $2w(J) = w(J) + \frac{w(J)}{2} + \frac{w(J)}{2} \geq w(J) + 2w(I) + 2\delta$. From the latter inequality our claim follows. □

Theorem 11. *For a polynomial f as in (2.1),* DCM *induces a subdivision tree T_{DCM} of*

$$\text{height } h(T_{\mathrm{DCM}}) = O(\log n - \log \sigma_f) \text{ and size } |T_{\mathrm{DCM}}| = O(\Sigma_f + n \log n).$$

Proof. The result on the height of T_{DCM} follows directly from the proof of Theorem 9. Namely, we have shown that DCM never subdivides an interval of width less than or equal to $\frac{\sigma_f}{8n^2}$. For the bound on $|T_{\mathrm{DCM}}|$, we use a similar argument as in [15] and [25]. Namely, for a root ξ of f and a certain $h \in \mathbb{N}_0$ we say that $I = (-\frac{1}{2} + i2^{-h}, -\frac{1}{2} + (i+1)2^{-h})$, $i = \{0, \ldots, 2^h - 1\}$, is *a canonical interval* for ξ if the real part of ξ is contained in $[-\frac{1}{2} + i2^{-h}, -\frac{1}{2} + (i+1)2^{-h})$ and $\sigma(\xi, f) < 8n^2 2^{-h} = 8n^2 w(I)$. We denote T_c the *canonical tree* which consists of all canonical intervals. We remark that, for a canonical interval I, the parent interval of I is canonical as well. The following considerations will show that $|T_{\mathrm{DCM}}| = O(|T_c|)$ and $|T_c| = O(\Sigma_f + n \log n)$: For the size of the canonical tree, consider a leaf $I \in T_c$ and let ξ_I be a root of f corresponding to this leaf. If there are several, then ξ_I is the root with minimal separation. Then, $\sigma(\xi_I, f) < 8n^2 2^{-h}$ and, thus, $h \leq 2 \log n + 4 - \log \sigma(\xi_I, f)$. Since each root of f is associated with at most one leaf of the canonical tree, we conclude that $|T_c| = O(n \log n + \Sigma_f)$. It remains to show that $|T_{\mathrm{DCM}}| = O(|T_c|)$. Consider the following mapping of internal nodes (intervals) of T_{DCM} to canonical nodes (intervals) in T_c: Let I be a non-terminal interval of width $w(I) = 2^{-h}$. Then, $\mathrm{var}(f, I^+) > 0$ and $\mathscr{T}[f_I'](0,2)$ does not hold. According Lemma 8 (c), the disk $\Delta_{2w(I)}(a)$ contains a root ξ of f with $\sigma(\xi, f) < 8n^2 w(I) = 8n^2 2^{-h}$. Hence, one of the four intervals, $I_1 := (a - 2w(I), a - w(I))$, $I_2 := (a - w(I), a)$, $I_3 := I$ or $I_4 := (b, b + (b - a))$, is canonical for ξ. We map I to the corresponding interval. This defines a mapping from the internal nodes of T_{DCM} to the nodes of the canonical tree T_c. Furthermore, each node in the canonical tree has at most four preimages in T_{DCM} and, thus, the number of internal nodes of T_{DCM} is bounded by $O(n \log n + \Sigma_f)$. Since T_{DCM} is a binary tree, the bound on the number of internal nodes applies to the whole tree as well. □

4. Algorithm

We first outline our algorithm $\mathbb{R}$ISOLATE to isolate the roots of f. $\mathbb{R}$ISOLATE decomposes into two subroutines DCM$^\rho$ and CERTIFY$^\rho$, where ρ indicates the actual working precision. We proceed in rounds: In the first round, we start with a low working precision (e.g. $\rho = 16$). If our algorithm does not succeed in a certain round, the precision is doubled in the next round. Following this approach, we can eventually guarantee an adaptive behavior of our method, that is, it eventually succeeds for a working precision which is at most twice the size of the actually needed precision.

The first subroutine DCM$^\rho$ is essentially identical to DCM with the main difference that, at each node $I = (a,b)$ of the recursion tree, we only consider approximations $\tilde{f}_I(x)$ of $f_I(x) = f(a+w(I)\cdot x)$ to a certain number ρ_I of bits after the binary point, where $\rho + 2\log w(I) \le \rho_I \le \rho$. We remark that we process I in a way such that I is not subdivided by DCM$^\rho$ if it is not subdivided by the the exact counterpart DCM. This ensures that, for any ρ, DCM$^\rho$ induces a subtree T_{DCM^ρ} of T_{DCM} and, thus, $|T_{\text{DCM}^\rho}| = O(\Sigma_f + n\log n)$ due to Theorem 11. We further show that, for a precision $\rho \ge \rho_f^{\max} = O(\Sigma_f + \log n)$, DCM$^\rho$ returns isolating intervals for *all* real roots of f; see Theorem 14 for the definition of $\rho_f^{\max}$ and further details. However, for smaller ρ, DCM$^\rho$ may return isolating intervals only for some roots but without any information whether all real roots are captured or not. In order to overcome such an undesirable situation, we consider an additional subdivision method CERTIFY$^\rho$ similar to DCM$^\rho$ which aims to certify that all roots are captured. We further show that CERTIFY$^\rho$ also induces a recursion tree of size $O(\Sigma_f + n\log n)$ and that it succeeds if $\rho \ge \rho_f^{\max}$.

4.1. DCM$^\rho$*: An Approximate Version of* DCM

We present our first subroutine DCM$^\rho$. Comments to support the approach are in *italic* and marked by a "//" at the beginning.

DCM$^\rho$. Let $I_0 = (-\frac{1}{2}, \frac{1}{2})$ be the starting interval which, by construction of f, contains all real roots of f. In a first step, we choose a $(\rho+n+1)$-binary approximation $\tilde{f}$ of f and evaluate $\tilde{f}(-\frac{1}{2}+x)$. Then, the resulting polynomial is approximated by a $(\rho+1)$-binary approximation $\tilde{f}_{I_0} \in [\tilde{f}(-\frac{1}{2}+x)]_{2^{-\rho-1}}$ and, according to Lemma 1, we have $\tilde{f}_{I_0} \in [f_{I_0}]_{2^{-\rho}}$.

DCM$^\rho$ maintains a list $\mathscr{A}$ of active nodes $(I, \tilde{f}_I, \rho_I)$, where $I = (a,b) \subset I_0$ is an interval, $\tilde{f}_I$ approximates f_I to ρ_I bits after the binary point and $\rho + 2\log w(I) \le \rho_I \le \rho$. DCM$^\rho$ eventually returns a list $\mathscr{O}$ of tuples $(J, s_{J,\ell}, s_{J,r}, B_J)$, where $J = (c,d)$ is an isolating interval for a root of f, $s_{J,\ell} = \text{sign}(f(c))$, $s_{J,r} = \text{sign}(f(d))$ and $0 < B_J \le \min(|f(c)|, |f(d)|)$. We initially start with $\mathscr{A} := \{(I_0, \tilde{f}_{I_0}, \rho)\}$ and $\mathscr{O} := \emptyset$. For each active node, we proceed as follows:

(1) Remove $(I, \tilde{f}_I, \rho_I)$ from $\mathscr{A}$.
(2) Compute the polynomials

$$\tilde{f}_{I^+}(x) = \tilde{f}_I\left(-\frac{1}{4n} + \left(1+\frac{1}{2n}\right)\cdot x\right) \quad \text{and} \quad \tilde{h}(x) = \sum_{i=0}^{n} \tilde{h}_i x^i := (1+x)^n \cdot \tilde{f}_{I^+}\left(\frac{1}{1+x}\right). \tag{4.1}$$

(3) If $\tilde{h}_i > -2^{n+2-\rho_I}$ for all i or $\tilde{h}_i < 2^{n+2-\rho_I}$ for all i, do nothing (i.e., I is dicarded).

// A simple computation (see the following Lemma 12 (a)) shows that $\tilde{h}$ is an approximation of $f_{I^+,\mathrm{rev}}(x) = (x+1)^n f_{I^+}(\frac{1}{1+x})$ to $\rho_I - n - 2$ bits after the binary point. Thus, if $\mathrm{var}(f, I^+) = 0$, all coefficients $\tilde{h}_i$ are either smaller than $2^{n+2-\rho_I}$ or larger than $-2^{n+2-\rho_I}$. Since we want to induce a subtree of the recursion tree T_{DCM} induced by f, we discard I if all coefficients of $\tilde{h}$ are larger than $-2^{n+2-\rho_I}$ (or smaller than $2^{n+2-\rho_I}$).

(4) If there exist $\tilde{h}_i$ and $\tilde{h}_j$ with $\tilde{h}_i \le -2^{n+2-\rho_I}$ and $\tilde{h}_j \ge 2^{n+2-\rho_I}$, consider the test $\mathscr{T}[(\tilde{f}_I)'](0,2)$, that is, evaluate $\mathbf{t}[(\tilde{f}_I)'](0,2) = \mathbf{t}[(\tilde{f}_I)', 3/2](0,2)$.

// Due to Lemma 12 (i), it holds that $|\mathbf{t}[(f_I)'](0,2) - \mathbf{t}[(\tilde{f}_I)'](0,2)| < n2^{n+1-\rho_I}$. Hence, if $\mathscr{T}[(f_I)'](0,2)$ holds, then $\mathbf{t}[(\tilde{f}_I)'](0,2) > -n2^{n+1-\rho_I}$. Thus, we proceed as follows:

(a) If $\mathbf{t}[(\tilde{f}_I)'](0,2) > -n2^{n+1-\rho_I}$, consider the polynomial

$$\hat{f}_I(x) := \tilde{f}_I(x) + \mathrm{sign}((\tilde{f}_I)'(0)) \cdot n \cdot 2^{n+1-\rho_I} \cdot x, \tag{4.2}$$

// Then, $\mathscr{T}[(\hat{f}_I)'](0,2)$ holds and, in particular, $\hat{f}_I$ is monotone on $(-2,2)$.

evaluate

$$\lambda^- := \hat{f}_I\left(-\frac{1}{4n}\right) = \tilde{f}_{I^+}(0) - 2^{n-1-\rho_I} \tag{4.3}$$

$$\lambda^+ := \hat{f}_I\left(1+\frac{1}{4n}\right) = \tilde{f}_{I^+}(1) + (4n+1)2^{n-1-\rho_I} \tag{4.4}$$

$$\lambda := \hat{f}_I\left(-\frac{1}{n}\right) = \tilde{f}_I\left(-\frac{1}{n}\right) - 2^{n+1-\rho_I}, \tag{4.5}$$

and check whether the following conditions are fulfilled:

$$\tilde{I} = (\tilde{a}, \tilde{b}) = (a - \frac{w(I)}{2n}, b + \frac{w(I)}{2n}) \text{ intersects no } J \text{ for any } (J, s_{J,\ell}, s_{J,r}, B_J) \in \mathscr{O}, \tag{4.6}$$

$$\lambda^- \cdot \lambda^+ < 0, \tag{4.7}$$

$$\min(|\lambda^-|, |\lambda^+|) > 2^{n+3-\rho_I} n, \text{ and} \tag{4.8}$$

$$|\lambda| > 2^{\deg \hat{f}_I + n + 7 - \rho_I} n^2. \tag{4.9}$$

If any of the conditions (4.6)-(4.9) fails, do nothing. If all conditions are fulfilled, then add $(\tilde{I}, \mathrm{sign}(\lambda^-), \mathrm{sign}(\lambda^+), \min(|\lambda^-|, |\lambda^+|) - 2^{n+3-\rho_I} n)$ to $\mathscr{O}$.

// If (4.7)-(4.9) hold, $\tilde{I}$ is isolating for a root ξ of f (Lemma 12 (c)). Furthermore, since $\hat{f}_I$ is monotone on $(-2,2)$, we have $|\hat{f}_I(-\frac{1}{2n})| > |\lambda^-|$ and $|\hat{f}_I(1+\frac{1}{2n})| > |\lambda^+|$. Then, inequality (4.8) and Lemma 12 (b) yields that $\mathrm{sign}(f(\tilde{a})) = \mathrm{sign}(\lambda^-)$, $\mathrm{sign}(f(\tilde{b})) = \mathrm{sign}(\lambda^+)$, and $\min(|f(\tilde{a})|, |f(\tilde{b})|) > \min(|\lambda^-|, |\lambda^+|) - 2^{n+3-\rho_I} n$.

(b) If $\mathbf{t}[(\tilde{f}_I)'](0,2) \le -n2^{n+1-\rho_I}$, subdivide I into $I_\ell := (a, m_I)$ and $I_r := (m_I, b)$. Compute a ρ_I-binary approximation $\tilde{f}_{I_\ell}$ of $\tilde{f}_I(\frac{x}{2})$ and a $(\rho_I - 1)$-binary approximation $\tilde{f}_{I_r}$ of $\tilde{f}_I(\frac{x+1}{2})$, and add $(I_\ell, \tilde{f}_{I_l}, \rho_I - 1)$ and $(I_r, \tilde{f}_{I_r}, \rho_I - 2)$ to $\mathscr{A}$. If $\rho_I < 2$, return "insufficient precision".

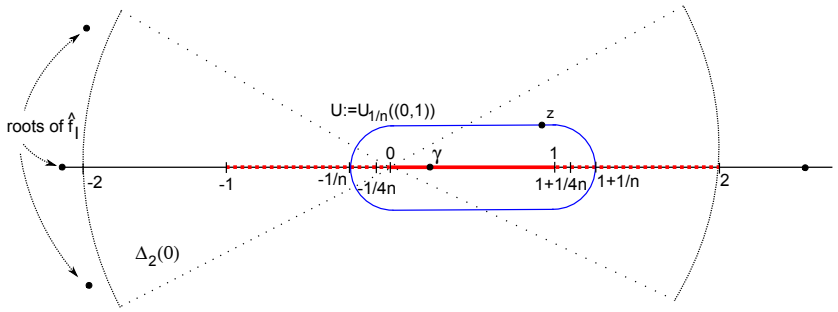

Fig. 4.1. If $\lambda^- \cdot \lambda^+ = \hat{f}_I(-\frac{1}{4n}) \cdot \hat{f}_I(1+\frac{1}{4n}) < 0$, then there exists a root $\gamma \in (-\frac{1}{4n}, 1+\frac{1}{4n})$ of $\hat{f}_I$. Furthermore, $\Delta_2(0)$ contains no further root of $\hat{f}_I$. A computation shows that $|\hat{f}_I(z)| > |f_I(z) - \hat{f}_I(z)|$ for all z on the boundary of the $\frac{1}{n}$-neighborhood U of $(0,1)$ if the inequality (4.9) holds. Then, due to Rouché's Theorem, U isolates a root of f_I.

// Due to Lemma 1, we have $\tilde{f}_{I_l} \in [f_{I_l}]_{2^{-\rho_I - 1}}$ and $\tilde{f}_{I_r} \in [f_{I_r}]_{2^{-\rho_I - 2}}$. Hence, by induction, it follows that $\rho + 2\log w(I) \le \rho_I \le \rho$ for all active nodes.

DCM$^\rho$ stops when $\mathscr{A}$ becomes empty. It may either return "insufficient precision" (in Step 4 (b)) or a list $\mathscr{O}$ of isolating intervals $\tilde{I}$ for some of the roots of f together with the signs of f and a lower bound on $|f|$ at the endpoints of $\tilde{I}$.

Lemma 12. *Let f be a polynomial as in (2.1), $I = (a,b)$ an interval considered by* DCM$^\rho$ *and $\tilde{h}$ the polynomial as defined in (4.1). Then,*

(a) $\tilde{h}(x) \in [f_{I^+,\mathrm{rev}}]_{2^{n+2-\rho_I}}$ *and* $|\mathbf{t}[(f_I)'](0,2) - \mathbf{t}[(\tilde{f}_I)'](0,2)| < n \cdot 2^{n+1-\rho_I}$.

(b) For an arbitrary value t with $|t| \le 1 + \frac{1}{n}$, it holds that $|f(a + t \cdot w(I)) - \hat{f}_I(t)| < 2^{n+3-\rho_I} n$, with $\hat{f}_I$ as defined in (4.2). In particular,

$$|f(a^+) - \lambda^-|,\ |f(a - \frac{w(I)}{n}) - \lambda|,\ |f(b^+) - \lambda^+| < 2^{n+3-\rho_I} n,$$

with λ^-, λ^+ and λ as defined in (4.3)-(4.5).

(c) Suppose that $\mathbf{t}[(\tilde{f}_I)'](0,2) > -n2^{n+1-\rho_I}$ and the inequalities (4.7)-(4.9) hold. Then, I^+ contains a real root ξ of f and the $\frac{w(I)}{n}$-neighborhood of I is isolating for ξ.

(d) For any tuple $(J, s_{J,\ell}, s_{J,r}, B_J) \in \mathscr{O}$, the endpoints of J are located outside the union of the disks

$$\Delta_i := \Delta_{\frac{\sigma(z_i,f)}{64n^3}}(z_i),\ \text{where } i = 1,\dots,n.$$

Proof. Since $\tilde{f}_I \in [f_I]_{2^{-\rho_I}}$, we have $\tilde{f}_{I^+} \in [f_{I^+}]_{2^{-\rho_I+2}}$ due to Lemma 1 (b). Reversing the coefficients and replacing x by $x+1$ increases the error by a factor of at most 2^n (see Lemma 1 (c)), thus $\tilde{h} \in [f_{I^+,\mathrm{rev}}]_{2^{-\rho_I+2+n}}$. For the second part of (a), consider the following simple computation:

$$|\mathbf{t}[(f_I)'](0,2) - \mathbf{t}[(\tilde{f}_I)'](0,2)| \le \frac{3}{2} \cdot n \cdot 2^{-\rho_I} \sum_{i=0}^{n-1} 2^i = \frac{3}{2} \cdot n \cdot 2^{-\rho_I}(2^n - 1) < n2^{n+1-\rho_I},$$

where the first inequality uses $(\tilde{f}_I)' \in [(f_I)']_{n \cdot 2^{-\rho_I}}$. For (b), we have

$$|f(a+t\cdot w(I))-\hat{f}_I(t)| = |f_I(t)-\hat{f}_I(t)| \le |f_I(t)-\tilde{f}_I(t)| + |t|\cdot 2^{n+1-\rho_I} n$$
$$\le 2^{-\rho_I}\sum_{i=0}^{n}|t|^i + \left(1+\frac{1}{n}\right)\cdot 2^{n+1-\rho_I} n$$
$$\le n2^{-\rho_I}\left(1+\frac{1}{n}\right)^{n+1} + n2^{n+2-\rho_I} < n2^{n+3-\rho_I}.$$

Now, if the inequalities (4.7) and (4.8) hold, then $\operatorname{sign}(f(a^+)) = \operatorname{sign}(\lambda^-)$, $\operatorname{sign}(f(b^+)) = \operatorname{sign}(\lambda^+)$ and $f(a^+)\cdot f(b^+) < 0$, hence, f has a real root in I^+. We next show that (4.9) implies the uniqueness of this root. From $\mathbf{t}[(\tilde{f}_I)'](0,2) > -n2^{n+1-\rho_I}$, it follows that $\mathscr{T}[(\hat{f}_I)'](0,2)$ succeeds and, thus, $\Delta_2(0)$ contains at most one root of $\hat{f}_I$. Since $\lambda^- = \hat{f}_I(-\frac{1}{4n})$ and $\lambda^+ = \hat{f}_I(1+\frac{1}{4n})$ have different signs, the interval $(-\frac{1}{4n}, 1+\frac{1}{4n})$ contains a root γ of $\hat{f}_I$. We consider the $\frac{1}{n}$-neighborhood $U \subset \mathbb{C}$ of $(0,1)$ and an arbitrary point z on its boundary; see Figure 4.1. It holds that $|-\frac{1}{n}-\gamma|/|z-\gamma| < (1+\frac{5}{4n})/(\frac{1}{4n}) = 4n+5 < 8n$ and, for any root $\tilde{\gamma} \neq \gamma$ of $\hat{f}_I$, we have

$$\frac{|-\frac{1}{n}-\tilde{\gamma}|}{|z-\tilde{\gamma}|} \le \frac{|-\frac{1}{n}-z|+|z-\tilde{\gamma}|}{|z-\tilde{\gamma}|} \le 1+\frac{1+\frac{2}{n}}{1-\frac{1}{n}} = 2\frac{1+\frac{1}{2n}}{1-\frac{1}{n}}.$$

Hence, it follows that

$$\left|\frac{\lambda}{\hat{f}_I(z)}\right| = \left|\frac{\hat{f}_I(-\frac{1}{n})}{\hat{f}_I(z)}\right| = \frac{|-\frac{1}{n}-\gamma|}{|z-\gamma|}\prod_{\tilde{\gamma}\neq\gamma:\hat{f}_I(\tilde{\gamma})=0}\frac{|-\frac{1}{n}-\tilde{\gamma}|}{|z-\tilde{\gamma}|}$$
$$< (4n+5)\cdot 2^{\deg\hat{f}_I-1}\left(1+\frac{1}{2n}\right)^{\deg\hat{f}_I-1}\left(1-\frac{1}{n}\right)^{-\deg\hat{f}_I+1}$$
$$< (4n+5)\cdot 2^{\deg\hat{f}_I-1}\cdot\sqrt{2.72}\cdot 2.72 < n2^{\deg\hat{f}_I+4}$$

and, thus, $|\hat{f}_I(z)| > |\lambda|\cdot 2^{-\deg\hat{f}_I-4}n^{-1}$. Since $|z| \le 1+\frac{1}{n}$, we have $|f_I(z)-\hat{f}_I(z)| < n2^{n+3-\rho_I}$ according to (b). Then, from Rouché's Theorem, it follows that f_I has exactly one root within U if (4.9) holds. This shows (c). It remains to prove (d): Let $\tilde{I} = (\tilde{a},\tilde{b})$ and $I = (a,b)$ the corresponding smaller interval. From our construction and (c), I^+ contains a root $\xi = z_{i_0}$ of f and the $\frac{w(I)}{n}$-neighborhood of I is isolating for this root. Thus, $|\tilde{a}-z_i| > \frac{w(I)}{4n}$ for all i. If there exists an $i \neq i_0$ with $\tilde{a} \in \Delta_i$, then $w(I) < 4n|\tilde{a}-z_i| < \sigma(z_i,f)/(16n^2)$. Hence, we obtain

$$|\xi - z_i| \le |\xi-\tilde{a}| + |\tilde{a}-z_i| < \left(1+\frac{1}{n}\right)\cdot w(I) + \frac{\sigma(z_i,f)}{64n^3}$$
$$< \left(1+\frac{1}{n}\right)\cdot\frac{\sigma(z_i,f)}{16n^2} + \frac{\sigma(z_i,f)}{64n^3} < \sigma(z_i,f),$$

a contradiction. It remains to show that $\tilde{a} \notin \Delta_{i_0}$. If $\tilde{a} \in \Delta_{i_0}$, then $w(I) < \frac{\sigma(\xi,f)}{16n^2}$. According to Lemma 8 (b), $T[(f_J)'](0,2)$ already holds for a parent node J of I and, thus, $\mathbf{t}[(\tilde{f}_J)'](0,2) > -n2^{n+1-\rho_J}$ because of (a). This contradicts the fact that J is not terminal. In completely analogous manner, one shows that $\tilde{b}$ is also not contained in any Δ_i. This proves (d). □

We close this section with a result on the size of the recursion tree induced by DCM$^\rho$ and the bit complexity of DCM$^\rho$:

Theorem 13. *Let f be a polynomial as in (2.1) and $\rho \in \mathbb{N}$ an arbitrary positive integer. Then, the recursion tree T_{DCM^ρ} induced by* DCM$^\rho$ *is a subtree of the tree T_{DCM} induced by* DCM*, thus,*

$$|T_{\mathrm{DCM}^\rho}| \le |T_{\mathrm{DCM}}| = O(\Sigma_f + n\log n).$$

Furthermore, DCM$^\rho$ *demands for a number of bit operations bounded by*

$$\tilde{O}(n(\Sigma_f + n\log n)(n\Gamma + \rho - \log\sigma_f)).$$

Proof. For the first claim, we remark that DCM$^\rho$ never splits an interval I which is not split by DCM when applied to the exact polynomial f. Namely, if I is terminal for DCM, then either $\mathbf{t}[(f_I)'](0,2) > 0$ or $\mathrm{var}(f, I^+) = \mathrm{var}(f_{I^+,\mathrm{rev}}) = 0$. In the first case, we must have $\mathbf{t}[(\tilde{f}_I)'](0,2) > -n2^{n+1-\rho_I}$ whereas, in the second case, all coefficients $\tilde{h}_i$ of $\tilde{h}(x) = (1+x)^n \cdot \tilde{f}_{I^+}(\frac{1}{1+x})$ are either larger than $-2^{n+2-\rho_I}$ or smaller than $2^{n+2-\rho_I}$; see Lemma 12 (a). Thus, I is terminal for DCM$^\rho$ as well. The result on the size of T_{DCM^ρ} then follows directly from Theorem 11.

For the bit complexity, we first consider the cost in each iteration: For an active node $(I, \tilde{f}_I, \rho_I) \in \mathscr{A}$, $I = (a,b)$, the polynomial $\tilde{f}_I$ approximates f_I to $\rho_I \le \rho$ bits after the binary point. The absolute value of each coefficient of f_I is bounded by $2^{n\Gamma}$ because the shift operation $x \mapsto a + (b-a)\cdot x$ does not increase the coefficients of f by a factor of more than 2^n and the absolute value of the coefficients of f is bounded by $2^{n\Gamma}$; see Section 2.2. It follows that the bitsize of the coefficients of $\tilde{f}_I$ is bounded by $O(n\Gamma) + \rho$. Hence, the cost for computing $\tilde{h}(x)$, $\tilde{f}_{I_\ell}$ and $\tilde{f}_{I_r}(x)$ is bounded by $\tilde{O}(n(n\Gamma + \rho))$. Namely, the latter constitutes a bound on the cost for a fast asymptotic Taylor shift by an $O(\log n)$-bit number. The cost for evaluating $\mathbf{t}[(f_I)'](0,2)$, λ^-, λ^+ and λ matches the same bound because all these computations are evaluations of a polynomial of bitsize $O(n\Gamma + \rho)$ at an $O(\log n)$-bit number. We further remark that, in each iteration, $\mathscr{O}$ contains disjoint isolating intervals J for some of the real roots of f and, thus, $|\mathscr{O}| \le n$. Hence, the endpoints of the interval J have to be compared with those of at most n intervals stored in $\mathscr{O}$. Since DCM$^\rho$ does not produce any interval of size less than $\frac{\sigma_f}{8n^2}$, these comparisons demand for at most $O(n(\log n - \log\sigma_f))$ bit operations. It follows that the total cost at each node is bounded by $\tilde{O}(n(n\Gamma + \rho - \log\sigma_f))$ bit operations. The bound on the total cost then follows from our result on the size of the recursion tree. □

4.2. Known ρ_f and σ_f

From Lemma 3 (c), we already know that, for $\rho \ge \rho_f$, each root z_i of f moves by at most $\frac{\sigma(z_i,f)}{64n^3}$ when passing from f to an arbitrary approximation $\tilde{f} \in [f]_{2^{-\rho_f}}$; see Definition 2 for the definition of ρ_f. Hence, we expect it to be possible to isolate the roots of f by only considering approximations of f (and the intermediate results f_I) to ρ_f bits after the binary point. The following theorem proves a corresponding result.

Theorem 14. *Let f be a polynomial as in (2.1) and $\rho \in \mathbb{N}$ an integer with*

$$\rho \ge \rho_f^{\max} := \lceil \rho_f - 3\log\sigma_f + 16n \rceil = O(\Sigma_f + n). \tag{4.10}$$

Then, DCM$^\rho$ *isolates* all *real roots of f and $B_J > 2^{-\rho_f}$ for all $(J, s_{J,\ell}, s_{J,r}, B_J) \in \mathscr{O}$.*

Proof. Due to Theorem 11 and 13, the height $h(\mathrm{DCM}^\rho)$ of T_{DCM^ρ} is bounded by

$$h(\mathrm{DCM}^\rho) \le \log\frac{16n^2}{\sigma_f} = 2\log n + 4 - \log\sigma_f \le 4n - \log\sigma_f.$$

Then, for any interval $I = (a,b)$ produced by $\textsc{Dcm}^\rho$, we have

$$\rho_I \geq \rho + 2\log w(I) \geq \rho - 2h(\textsc{Dcm}^\rho) \geq \rho_f^{\min} := \lceil \rho_f + 8n - \log \sigma_f \rceil > 0. \tag{4.11}$$

The latter inequality guarantees that $\textsc{Dcm}^\rho$ does not return "insufficient precision". Now let I be an interval whose closure $\bar{I}$ contains a root $\xi = z_{i_0}$ of f. We aim to show the following facts:

(1) I is not discarded in Step 3 of $\textsc{Dcm}^\rho$.
(2) If $\mathbf{t}[(\tilde{f}_I)'](0,2) > -n2^{n+1-\rho_I}$, then all inequalities (4.7)-(4.9) are fulfilled.
(3) In the latter case, either $\tilde{I} = (a - \frac{w(I)}{2n}, b + \frac{w(I)}{2n})$ is added to $\mathscr{O}$ or $\tilde{I}$ only intersects intervals J, with a corresponding $(J, s_{J,\ell}, s_{J,r}, B_J) \in \mathscr{O}$, which already isolate ξ.

If (1)-(3) hold, then $\textsc{Dcm}^\rho$ outputs isolating intervals for *all* real roots of f. Namely, $\textsc{Dcm}^\rho$ starts subdividing $I_0 = (-\frac{1}{2}, \frac{1}{2})$ which contains all real roots of f. Thus, for each root ξ of f, we eventually obtain an interval I such that $\bar{I}$ contains ξ and $\mathbf{t}[(\tilde{f}_I)'](0,2) > -n2^{n+1-\rho_I}$. Then, either $\tilde{I}$ is added to the list of isolating intervals or $\mathscr{O}$ already contains an isolating interval for ξ.

For the proof of (1), we have already shown that $w(I) > \frac{\sigma(\xi,f)}{16n^2}$. Lemma 3 (c) then ensures that an arbitrary $g \in [f]_{2^{-\rho_f}}$ has a root $\xi' \in I^+$. Namely, the root $\xi \in \bar{I}$ stays real and moves by at most $\frac{\sigma(\xi,f)}{64n^3} < \frac{w(I)}{4n}$ when passing from f to g. Now, suppose that all coefficients $\tilde{h}_i$ of $\tilde{h}(x) = (1+x)^n \cdot \tilde{f}_{I^+}(\frac{1}{1+x})$ are larger than $-2^{n+2-\rho_I}$; see (4.1) for definitions. Since $|h_i - \tilde{h}_i| < 2^{n+2-\rho_I}$ for all coefficients h_i of $f_{I^+,\mathrm{rev}} = \sum_{i=0}^n h_i x^i$ (see Lemma 12 (a)), it follows that $h_i > -2^{n+3-\rho_I}$ for all i. Hence, for the polynomial

$$g(x) := f(x) + 2^{n+3-\rho_I} \in [f]_{2^{-\rho_f}},$$

we have $g_{I^+,\mathrm{rev}}(x) = f_{I^+,\mathrm{rev}}(x) + 2^{n+3-L_I}(x+1)^n$ and, thus, $g_{I^+,\mathrm{rev}}$ has only positive coefficients. In the case where $\tilde{h}_i < 2^{n+2-\rho_I}$ for all i, we consider $g(x) := f(x) - 2^{n+3-\rho_I} \in [f]_{2^{-\rho_f}}$ and, thus, $g_{I^+,\rho}$ has only negative coefficients. Hence, in both cases, there exists a $g \in [f]_{2^{-\rho_f}}$ which has no root in I^+, a contradiction. It follows that I cannot be discarded in Step 3.

For (2), suppose that $\mathbf{t}[(\tilde{f}_I)'](0,2) > -n2^{n+1-\rho_I}$. Due to Lemma 12 (a), we have $\mathbf{t}[(f_I)'](0,2) > -n2^{n+2-\rho_I}$, and since $\log \frac{n2^{n+2-\rho_I}}{w(I)} \leq 6 + 3\log n + n - \rho_I - \log \sigma_f < -\rho_f$, it follows that

$$g(x) := f(x) + x \cdot \mathrm{sign}((f_I)'(0)) \cdot \frac{n2^{n+2-\rho_I}}{w(I)} \in [f]_{2^{-\rho_f}}.$$

Hence, g has a root ξ' in I^+. Since $\mathbf{t}[(g_I)'](0,2) = \mathbf{t}[(f_I)'](0,2) + n2^{n+2-\rho_I} > 0$, the disk $\Delta_{2w(I)}(a)$ is isolating for ξ'. The following argument shows that $\Delta_{\frac{3w(I)}{2}}(a)$ isolates ξ: Suppose that $\Delta_{\frac{3w(I)}{2}}(a)$ contains an additional root $z_j \neq \xi$ of f. Then, $\sigma(\xi,f) < 3w(I)$ and, thus, ξ and z_j would move by at most $\frac{3w(I)}{64n^3} < \frac{w(I)}{2}$ when passing from f to g. It follows that g would have at least two roots within $\Delta_{2w(I)}(a)$, a contradiction. Now, since $\Delta_{\frac{3w(I)}{2}}(a)$ is isolating for $\xi \in \bar{I}$, we have

$$\frac{\sigma(\xi,f)}{16n^2} < w(I) < 2\sigma(\xi,f).$$

The left inequality implies that the distance of ξ to any of the points $a^+ = a - \frac{w(I)}{4n}$, $b^+ = b + \frac{w(I)}{4n}$ and $c := a - \frac{w(I)}{n}$ is larger than or equal to $\frac{w(I)}{4n} > \frac{\sigma(\xi,f)}{64n^3}$. Let d_i denote the distance between a root $z_i \neq \xi$ and the disk $\Delta_{\frac{3w(I)}{2}}(a)$. Then,

$$\frac{\sigma(z_i,f)}{64n^3} \leq \frac{|z_i - \xi|}{64n^3} \leq \frac{d_i + 3w(I)}{64n^3} < d_i + \frac{w(I)}{4}.$$

It follows that the points a^+, b^+, $c \in \Delta_{\frac{5w(I)}{4}}(a)$ are located outside the disk $\Delta_i := \Delta_{\frac{\sigma(z_i,f)}{64n^3}}(z_i)$. In summary, none of the disks Δ_i, $i = 1,\dots,n$, contains any of the points a^+, b^+ and c. Hence, due to Lemma 3 (c), it follows that each of the values $|f(c)|$, $|f(a^+)|$ and $|f(b^+)|$ is larger than $(n+1)2^{-\rho_f}$. A simple computation now shows that $(n+1)2^{-\rho_f} > 2^{2n+8-\rho_I}n^2$. Thus, according to Lemma 12 (b), each of the absolute values $|\lambda|$, $|\lambda^-|$ and $|\lambda^+|$ is larger than

$$(n+1)2^{-\rho_f} - 2^{n+3-\rho_I}n > 2^{2n+8-\rho_I}n^2 - 2^{n+3-\rho_I}n > 2^{2n+7-\rho_I}n^2. \tag{4.12}$$

It follows that the inequalities (4.8) and (4.9) hold. Since I^+ is isolating for ξ, $f(a^+)$ and $f(b^+)$ must have different signs and, thus, the same holds for λ^- and λ^+. Hence, the inequality (4.7) holds as well. In addition, we have $B_{\tilde{I}} = \min(|\lambda^-|,|\lambda^+|) - 2^{n+3-\rho_I}n > 2^{-\rho_f}$ because of (4.12). It remains to show (3): If $\mathbf{t}[(\tilde{f}_I)'](0,2) > -n2^{n+1-\rho_I}$, then due to (2) and Lemma 12 (b), the interval $\tilde{I}$ and the $\frac{w(I)}{n}$-neighborhood of I is isolating for ξ. If $\tilde{I}$ does not intersect any other interval in $\mathscr{O}$, then $\tilde{I}$ is added to $\mathscr{O}$ and, thus, DCM$^\rho$ outputs an isolating interval for ξ. We still have to consider the case where $\tilde{I}$ intersects an interval J from $\mathscr{O}$. From the construction of $\mathscr{O}$, J is the extension $(\tilde{c},\tilde{d})$ of an interval $J' = (c,d)$. Now, suppose that J is isolating for a root $\gamma \neq \xi$. The roots ξ and γ move by at most $\frac{w(I)}{4n}$ and $\frac{w(J')}{4n}$, respectively, when passing from f to an arbitrary $g \in [f]_{2^{-\rho_f}}$ (see the proof of (1)). Hence, it follows that the union of $(a-w(I),b+w(I))$ and $(c-w(J'),d+w(J'))$ contains at least two roots of any $g \in [f]_{2^{-\rho_f}}$. Due to Lemma 10, one of the disks $\Delta_{2w(I)}(a)$ or $\Delta_{2w(J')}(c)$ then also contains at least two roots of g contradicting the fact that $\mathbf{t}[(p_I)'](0,2) > 0$ for $p(x) := f(x) + x \cdot \operatorname{sign}((f_I)'(0)) \cdot \frac{n2^{n+2-\rho_I}}{w(I)} \in [f]_{2^{-\rho_f}}$ and $\mathbf{t}[(q_{J'})'](0,2) > 0$ for $q(x) := f(x) + x \cdot \operatorname{sign}((f_I)'(0)) \cdot \frac{n2^{n+2-\rho_{J'}}}{w(J')} \in [f]_{2^{-\rho_f}}$. It follows that J already isolates ξ. □

4.3. Unknown ρ_f and σ_f

For unknown ρ_f and σ_f, we proceed as follows: We start with an initial precision ρ (e.g. $\rho = 16$) and run DCM$^\rho$. If DCM$^\rho$ returns "insufficient precision", we double ρ and start over. Otherwise, DCM$^\rho$ returns a list $\mathscr{O} = \{(J_k, s_{k,\ell}, s_{k,r}, B_k)\}_{k=1,\dots,m}$, where each interval $J_k = (c_k,d_k)$ isolates a real root of f, $s_{k,\ell} = \operatorname{sign} f(c_k)$, $s_{k,r} = \operatorname{sign} f(d_k)$ and $0 < B_k < \min(|f(c_k)|,|f(d_k)|)$. As already mentioned, there is no guarantee that all roots of f are captured. Hence, in a second step, we use the subsequently described method CERTIFY$^\rho$ to check whether the *region of uncertainty*

$$\mathscr{R} := \left[-\frac{1}{2},\frac{1}{2}\right] \setminus \bigcup_{k=1}^{m} J_k$$

contains a root of f. If we can guarantee that $f(x) \neq 0$ for all $x \in \mathscr{R}$, we return the list $\mathscr{L} = \{J_k\}_{k=1,\dots,m}$ of isolating intervals. Otherwise, we double ρ and start over the entire algorithm. We have already proven in Theorem 14 that DCM$^\rho$ isolates all real roots of f if $\rho \geq \rho_f^{\max}$ (i.e., ρ fulfills the inequality (4.10)). The following considerations will show that, for $\rho \geq \rho_f^{\max}$, CERTIFY$^\rho$ succeeds as well.

How can we guarantee that f does not vanish on $\mathscr{R}$? The crucial idea is to consider a decomposition of $[-\frac{1}{2},\frac{1}{2}]$ into subintervals I and corresponding μ_I-approximations g of f_I such that g is monotone on $[0,1]$ or $\mathscr{T}[g](0,1)$ holds. Namely, for such an interval I, we can easily estimate the image $g([0,1])$ and, thus, conclude that f either contains no root in $I \cap \mathscr{R}$ or that $\rho < \rho_f^{\max}$ because $g(t)$ and $f_I(t)$ differ by at most $(n+1)\mu_I$ for all $t \in [0,1]$. More precisely, we have:

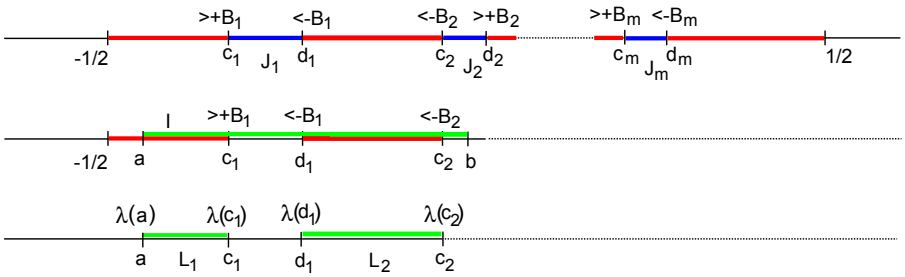

Fig. 4.2. DCM$^\rho$ returns a list $\mathscr{O} = \{J_k, s_{k,\ell}, s_{k,r}, B_k\}_k$, where J_k is isolating for a real root of f, $s_{k,\ell} = \operatorname{sign} f(c_k)$, $s_{k,r} = \operatorname{sign} f(d_k)$ and $\min(|f(c_k)|, |f(d_k)|) > B_k > 0$. The intervals in between define the *region of uncertainty* $\mathscr{R}$. In CERTIFY$^\rho$, we subdivide $(-1/2, 1/2)$ into intervals I such that, for a μ-approximation g of f_I, either $\mathscr{T}[g](0,1)$ holds or g is monotone on $(0,1)$. If $\mathscr{T}[g](0,1)$ holds and $|g(0)| > 8m\mu$, then I contains no root of f; see Lemma 15 (a). If g is monotone on $(0,1)$, we consider all intervals L_i in the intersection of I with $\mathscr{R}$ and check whether the conditions in Lemma 15 (b) are fulfilled. If they are fulfilled, then f has no root in L_i; otherwise, we must have $\rho < \rho_f^{\max}$.

Lemma 15. *Let $I = (a,b)$ be an interval and $g(x)$ a μ-binary approximation of f_I with*

$$-\log \mu \geq \rho - 2(4n - \log \sigma_f). \tag{4.13}$$

(a) Suppose that $\mathscr{T}[g](0,1)$ holds and I is not entirely contained in one of the J_k. If

$$|g(0)| > 8n\mu, \tag{4.14}$$

then $\bar{I}$ contains no root of f. Otherwise, $\rho < \rho_f^{\max}$.

(b) Suppose that g is monotone on $[0,1]$ and let $\bar{I} \cap \mathscr{R} = \bigcup_{i=1}^{s} L_i$ be the intersection of $\bar{I}$ and $\mathscr{R}$. For each endpoint q of an arbitrary L_i, we define

$$\lambda(q) := \begin{cases} s_{k,\ell} \cdot B_k, & \text{if } q \notin \{a,b\} \text{ and } q \text{ is the left endpoint of an interval } J_k \\ s_{k,r} \cdot B_k, & \text{if } q \notin \{a,b\} \text{ and } q \text{ is the right endpoint of an interval } J_k \\ g(0), & \text{if } q = a \\ g(1), & \text{if } q = b. \end{cases} \tag{4.15}$$

If, for all $L_i = [q_\ell, q_r]$, $\min(|\lambda(q_\ell)|, |\lambda(q_r)|) > 4n\mu$ and $\lambda(q_l) \cdot \lambda(q_r) > 0$, then $\bar{I} \cap \mathscr{R}$ contains no root of f. Otherwise, we have $\rho < \rho_f^{\max}$.

Proof. If $\mathscr{T}[g](0,1)$ holds, then $\frac{1}{3}|g(0)| < |g(t)| < \frac{5}{3}|g(0)|$ for all $t \in [0,1]$ according to Lemma 6. Suppose that $g(0) > 8n\mu$, then it follows that

$$|f_I(t)| \geq |g(t)| - |g(t) - f_I(t)| > \frac{1}{3}|g(0)| - |g(t) - f_I(t)| > \frac{8}{3}n\mu - (n+1)\mu > 0,$$

hence, f has no root in $\bar{I}$. Now suppose that $|g(0)| \leq 8n\mu$. Since I is not contained in any J_k, there exists a $t \in [0,1]$ with $x = a + t(b-a) \in \mathscr{R}$ and $|f(x)| = |f_I(t)| \leq |g(t)| + (n+1)\mu \leq \frac{5}{3}|g(0)| + (n+1)\mu < 16n\mu$. If $\rho \geq \rho_f^{\max}$, then from (4.13) and the definition of $\rho_f^{\max}$, it follows that $-\log \mu \geq \rho_f^{\min} = \lceil \rho_f + 8n - \log \sigma_f \rceil$; see the computation in (4.11). Hence, we have $|f(x)| < 2^{-\rho_f}$. In addition, Lemma 12 (d) and Theorem 14 guarantee that DCM$^\rho$ returns isolating intervals for all real roots of f, and each point in $\mathscr{R}$ has distance $\geq \sigma(z_i, f)/(64n^3)$ from each root z_i. Thus, $|f(x)| > (n+1)2^{-\rho_f}$ due to Lemma 3 (c), a contradiction. This proves (a).

For (b), we consider an arbitrary interval $L_i = [q_\ell, q_r]$. Let t_ℓ and t_r be corresponding values in $[0,1]$ with $q_l = a + t_\ell \cdot w(I)$ and $q_r = a + t_r \cdot w(I)$. If $\min(|\lambda(q_\ell)|, |\lambda(q_r)|) > 4n\mu$, then

$$\min(|g(t_\ell)|, |g(t_r)|) \geq \min(|\lambda(q_\ell)|, |\lambda(q_r)|) - (n+1)\mu > 2n\mu.$$

Namely, for $q_\ell = a$, we obviously have $|g(t_\ell)| = |\lambda(q_\ell)|$; otherwise, $|g(t_\ell)| \geq |f_I(t_\ell)| - (n+1)\mu \geq |\lambda(q_\ell)| - (n+1)\mu$. For q_r, an analogous argument applies. If, in addition, $\lambda(q_\ell) \cdot \lambda(q_r) > 0$, then $g(t_\ell) \cdot g(t_r) > 0$ as well because $\lambda(q_\ell)$ and $\lambda(q_r)$ have the same sign as $g(t_\ell)$ and $g(t_r)$, respectively. Since we assumed that g is monotone on $[0,1]$, it follows that $|g(t)| > 2n\mu$ for all $t \in [t_\ell, t_r]$. This shows that $|f_I(t)| \geq |g(t)| - (n+1)\mu > 0$ for all $t \in [t_\ell, t_r]$, thus the first part of (b) follows. For the second part, suppose that $\rho \geq \rho_f^{\max}$. Then, $B_k > 2^{-\rho_f} > 4n\mu$ for all k and $|f(x)| > 2^{-\rho_f}(n+1)$ for all $x \in \mathscr{R}$ according to Lemma 3 (c) and Theorem 14. Thus, if $a \in \mathscr{R}$, we have

$$|g(0)| \geq |f_I(0)| - (n+1)\mu = |f(a)| - (n+1)\mu > 2^{-\rho_f} \cdot (n+1) - (n+1)\mu > 4n\mu.$$

An analogous argument applies to b. It follows that $|\lambda(q)| > 4n\mu$ for all endpoints q of an arbitrary interval $L_i = [q_\ell, q_r]$. It remains to show that $\lambda(q_\ell) \cdot \lambda(q_r) > 0$. We have already shown that $|\lambda(q)| > 4n\mu$ for each endpoint q, thus, $f(q)$ must have the same sign as $\lambda(q)$. Namely, if $q \in \{a,b\}$, then $f(q)$ differs from $\lambda(q) > 4n\mu$ by at most $(n+1)\mu < 4n\mu$, and, for $q \notin \{a,b\}$, we have $\operatorname{sign}(\lambda(q)) = s_{k,\ell}$ or $\operatorname{sign}(\lambda(q)) = s_{k,r}$ depending on whether q is the left or the right endpoint of an interval J_k. Since $\rho \geq \rho_f^{\max}$, $\mathscr{R}$ contains no root of f, thus, we must have $\lambda(q_\ell) \cdot \lambda(q_r) = f(q_\ell) \cdot f(q_r) > 0$. □

We can now formulate the subroutine CERTIFY$^\rho$ (see Algorithm 3 in the Appendix for pseudocode). CERTIFY$^\rho$ is similar to DCM$^\rho$ in the sense that we recursively subdivide $I_0 = (-\frac{1}{2}, \frac{1}{2})$ into intervals I and consider corresponding ρ_I-binary approximations $\tilde{f}_I$ of f_I. Then, in each iteration, we aim to apply Lemma 15 in order to certify that $\bar{I} \cap \mathscr{R}$ contains no root of f or $\rho < \rho_f^{\max}$. Throughout the following consideration, we *assume* that

$$\text{CERTIFY}^\rho \text{ never produces an interval } I \text{ of width } w(I) \leq \frac{\sigma_f}{8n^2}. \tag{4.16}$$

We will prove this fact in Theorem 16 (b). Again, we mark comments which should help to follow the approach by an "//" at the beginning.

CERTIFY$^\rho$. In a first step, we choose a $(\rho + n + 1)$-binary approximation $\tilde{f}$ of f and evaluate $\tilde{f}(-\frac{1}{2} + x)$. Then, the resulting polynomial is approximated by a $(\rho+1)$-binary approximation $\tilde{f}_{I_0} \in [\tilde{f}(-\frac{1}{2} + x)]_{2^{-\rho-1}}$, thus, $\tilde{f}_{I_0} \in [f_{I_0}]_{2^{-\rho}}$ according to Lemma 1.

CERTIFY$^\rho$ maintains a list $\mathscr{A}$ of active nodes $(I, \tilde{f}_I, \rho_I)$, where $I = (a,b) \subset I_0$ is an interval, $\tilde{f}_I$ approximates f_I to ρ_I bits after the binary point and $\rho + 2\log w(I) \leq \rho_I \leq \rho$. We initially start with $\mathscr{A} := \{(I_0, \tilde{f}_{I_0}, \rho)\}$. For each active node, we proceed as follows:

(1) Remove $(I, \tilde{f}_I, \rho_I)$ from $\mathscr{A}$.
(2) If $I \cap \mathscr{R} = \emptyset$, do nothing (i.e., discard I). Otherwise, compute $\mathbf{t}[\tilde{f}_I](0,1)$.

// If $I \cap \mathscr{R} = \emptyset$, I is contained in one of the intervals J_k, and thus we can discard I.

(3) If $\mathbf{t}[\tilde{f}_I](0,1) > -2^{-\rho_I+2}n$, check whether

$$|\tilde{f}_I(0) + \operatorname{sign}(\tilde{f}_I(0)) \cdot 2^{-\rho_I+2}n| > 2^{-\rho_I+6}n^2. \tag{4.17}$$

If (4.17) holds, do nothing (i.e., discard I); otherwise, return "insufficient precision".

// For $g(x) := \tilde{f}_I(x) + \operatorname{sign}(\tilde{f}_I(0)) \cdot 2^{-\rho_I+2}n \in [f_I]_{2^{-\rho_I+3}n}$, the predicate $\mathscr{T}[g](0,1)$ holds. From our assumption on $w(I)$, we further have

$$\begin{aligned}\rho_I - 3 - \log n &\geq \rho + 2\log w(I) - 3 - \log n \geq \rho - 2(3 + 2\log n - \log\sigma_f) - 3 - \log n \\ &\geq \rho - 2\log\sigma_f - 8n,\end{aligned}$$

and thus $2^{-\rho_I+3}n \leq 2^{-\rho-2(4n-\log\sigma_f)}$. It follows that g fulfills the condition (4.13) from Lemma 15 and, therefore, $\bar{I}$ contains no root of f if (4.17) holds; otherwise, $\rho < \rho_f^{\max}$.

(4) If $\mathbf{t}[\tilde{f}_I](0,1) \leq -2^{-\rho_I+2}n$, compute $\tilde{h}(x) = \sum_{i=0}^{n} \tilde{h}_i x^i := (1+x)^n \cdot (\tilde{f}_I)'(\frac{1}{1+x})$ and consider the following distinct cases:

(a) If $\tilde{h}_i > -n2^{n-\rho_I}$ for all i (or $\tilde{h}_i < n2^{n-\rho_I}$ for all i), consider

$$g(x) := \tilde{f}_I(x) + n2^{n-\rho_I} \cdot x \in [f_I]_{n2^{n+1-\rho_I}}$$

($g(x) := \tilde{f}_I(x) - n2^{n-\rho_I} \cdot x$, respectively). Then, for each interval $L_i = [q_\ell, q_r]$, determine $\lambda(q_\ell)$ and $\lambda(q_r)$ as defined in (4.15). If $\min(|\lambda(q_\ell)|, |\lambda(q_r)|) > n2^{n+3-\rho_I}$ and $\lambda(q_\ell) \cdot \lambda(q_r) > 0$ for all L_i, discard I; otherwise, return "insufficient precision".

// Suppose $\tilde{h}_i > -n2^{n-\rho_I}$ for all i and define $g(x) := \tilde{f}_I(x) + n2^{n-\rho_I}x$. Then, the polynomial $(1+x)^n \cdot (g)'(\frac{1}{1+x}) = (1+x)^n \cdot (\tilde{f}_I)'(\frac{1}{1+x}) + n2^{n-\rho_I}(1+x)^n$ has only positive coefficients. It follows that $\operatorname{var}(g', (0,1)) = 0$ and, therefore, g is monotone on $[0,1]$. In addition, from our assumption on $w(I)$, we have $n2^{n+1-\rho_I} \leq 2^{-\rho-2(4n-\log\sigma_f)+1}$ (see the computation above). Hence, we can apply Lemma 15 (b) to g: That is, $\bar{I} \cap \mathscr{R}$ contains no root of f if $\min(|\lambda(q_\ell)|, |\lambda(q_r)|) > n2^{n+3-\rho_I}$ and $\lambda(q_\ell) \cdot \lambda(q_r) > 0$ for all $L_i = [q_l, q_r]$. If one of the latter two inequalities does not hold, then $\rho < \rho_f^{\max}$. The case $\tilde{h}_i < n2^{n-\rho_I}$ for all i is treated in exactly the same manner.

(b) If there exist $\tilde{h}_i$ and $\tilde{h}_j$ with $\tilde{h}_i \leq -n2^{n-\rho_I}$ and $\tilde{h}_j \geq n2^{n-\rho_I}$, then I is subdivided into $I_\ell := (a, m_I)$ and $I_r := (m_I, b)$. We add $(I_\ell, \tilde{f}_{I_\ell}, \rho_I - 1)$ and $(I_r, \tilde{f}_{I_r}, \rho_I - 2)$ to $\mathscr{A}$, where $\tilde{f}_{I_\ell}$ is an ρ_I-binary approximation of $\tilde{f}_I(\frac{x}{2})$ and $\tilde{f}_{I_r}$ an $(\rho_I - 1)$-binary approximation of $\tilde{f}_I(\frac{x+1}{2})$; see Step 4 (b) of DCM$^\rho$ for details. If $\rho_I < 2$, return "insufficient precision".

// Due to Lemma 1, $\tilde{f}_{I_l} \in [f_{I_l}]_{2^{-\rho_I-1}}$ and $\tilde{f}_{I_r} \in [f_{I_r}]_{2^{-\rho_I-2}}$. Hence, by induction, it follows that $\rho + 2\log w(I) \leq \rho_I \leq \rho$ for all active nodes.

CERTIFY$^\rho$ stops when $\mathscr{A}$ becomes empty. If CERTIFY$^\rho$ returns "insufficient precision", we know for sure that $\sigma < \sigma_f^{\max}$. Otherwise, the region of uncertainty $\mathscr{R}$ contains no root of f.

The following theorem proves that our assumption (4.16) for the intervals produced by CERTIFY$^\rho$ is correct. Furthermore, we show that CERTIFY$^\rho$ is also efficient with respect to bit complexity matching the worst case bound obtained for DCM$^\rho$; see Theorem 13.

Theorem 16. *For a polynomial f as defined in (2.1) and an arbitrary $\rho \in \mathbb{N}$,*

(a) CERTIFY$^\rho$ *does not produce an interval I of width $w(I) \le \sigma_f/(8n^2)$ and it induces a recursion tree of size $O(\Sigma_f + n\log n)$.*

(b) CERTIFY$^\rho$ *needs no more than $\tilde{O}(n(\Sigma_f + n\log n)(n\Gamma + \rho - \log\sigma_f))$ bit operations.*

(c) For $\rho \ge \rho_f^{\max}$, CERTIFY$^\rho$ *succeeds.*

Proof. An interval I is only subdivided if $\mathbf{t}[\tilde{f}_I](0,1) \le -2^{-\rho_I+2}n$ (Step (3)) and if there exist coefficients $\tilde{h}_i$ and $\tilde{h}_j$ of $\tilde{h}(x) = \sum_{i=0}^n \tilde{h}_i x^i = (1+x)^n \cdot (\tilde{f}_I)'(\frac{1}{1+x})$ with $\tilde{h}_i < -n2^{n-\rho_I}$ and $\tilde{h}_j > n2^{n-\rho_I}$ (Step 4 (b)). In the first case, we must have $\mathbf{t}[f_I](0,1) < 0$ since $|\mathbf{t}[f_I](0,1) - \mathbf{t}[\tilde{f}_I](0,1)| < 2^{-\rho_I+2}n$. Hence, $\mathscr{T}[f_I](0,1)$ does not hold. For the second case, we have $\mathrm{var}((f_I)',(0,1)) \ne 0$ since corresponding coefficients of $\tilde{h}(x) = (1+x)^n \cdot (\tilde{f}_I)'(\frac{1}{1+x})$ and $(1+x)^n \cdot (f_I)'(\frac{1}{1+x})$ differ by at most $n2^{n-\rho_I}$, and thus $\mathrm{var}(f',I) = \mathrm{var}((f')_I,(0,1)) = \mathrm{var}((f_I)',(0,1)) \ne 0$. Hence, the first part of (a) follows from Lemma 8 (d). This proves that our assumption (4.16) is always fulfilled, and thus Lemma 15 applies in Step 3 and Step 4 (a) of CERTIFY$^\rho$. It follows that the algorithm only fails (i.e. it returns "insufficient precision") if $\rho < \rho_{\max}^f$. For the second part of (a), we remark that, due to the above argument, an interval I is terminal if the disk $\Delta_{2nw(I)}(m(I))$ does not contain a root ξ of f with $\sigma(\xi,f) < 4n^2 w(I)$. In [35, Section 4.2], it has been shown that the recursion tree $T(f')$ induced by the latter property [14] has size $O(\Sigma_f + n\log n)$. Hence, the same holds for the recursion tree induced by CERTIFY$^\rho$ which is a subtree of $T(f')$. Finally, (c) follows in completely analogous manner as the result on the bit complexity for DCM$^\rho$ as shown in the proof of Theorem 13. □

Eventually, we present our overall root isolation method $\mathbb{R}$ISOLATE. It applies to a polynomial F as given in (1.4) and returns isolating intervals for all real roots of F.

$\mathbb{R}$ISOLATE: Choose a starting precision $\rho \in \mathbb{N}$ (e.g., $\rho = 16$) and run DCM$^\rho$ on the polynomial f as defined in (2.1). If DCM$^\rho$ returns "insufficient precision", we double ρ and start over again. Otherwise, DCM$^\rho$ returns a list $\mathscr{O} = \{(J_k, s_{k,\ell}, s_{k,r}, B_k)\}_{k=1,\dots,m}$ with isolating intervals J_k for some of the real roots of f. If CERTIFY$^\rho$ returns "insufficient precision", we double ρ and start over the algorithm. If CERTIFY$^\rho$ succeeds, the intervals $J_k = (c_k, d_k)$ isolate all real roots of f. Hence, we return the intervals $(2^\Gamma c_k, 2^\Gamma d_k)$, $k = 1,\dots,m$, which isolate the real roots of F.

The following theorem summarizes our results:

Theorem 17. *Let F be a polynomial as given in (1.4). Then,* $\mathbb{R}$ISOLATE *determines isolating intervals $J_1,\dots,J_m$ for all real roots of F and, for each interval $J \in \{J_1,\dots,J_m\}$ containing a root ξ of F, it holds that*

$$\frac{\sigma(\xi,F)}{16n^2} < w(J) < 2n\sigma(\xi,F).$$

[14] In [35, Section 4.2], $T(f')$ is defined as subdivision tree obtained by recursive bisection of the interval $(-\frac{1}{4},\frac{1}{4})$ in accordance with the following rule: At depth $h \in \mathbb{N}_0$, an interval $I = (-\frac{1}{4} + i2^{-h-1}, -\frac{1}{4} + (i+1)2^{-h-1})$ is subdivided if and only if $\mathrm{var}(f',I) \ne 0$ and $\Delta_{2^8 n^5 w(I)}(m(I))$ contains a root ξ of f with separation $\sigma(\xi,f) < 2^7 n^5 w(I)$. For the given situation, we can alternatively define $T(f')$ as the (even smaller) tree obtained by recursive bisection of $(-\frac{1}{2},\frac{1}{2})$, where an interval I is subdivided if $\mathrm{var}(I,f') \ne 0$ and $\Delta_{2nw(I)}(m(I))$ contains a root ξ of f with $\sigma(\xi,f) < 4n^2 w(I)$.

$\mathbb{R}$ISOLATE *demands for coefficient approximations of* F *to* $\tilde{O}(\Sigma_F + n\Gamma_F)$ *bits after the binary point and the total cost is bounded by*

$$\tilde{O}(n(\Sigma_F + n\Gamma_F)^2) = \tilde{O}(n(\Sigma_F + n\tau)^2)$$

bit operations. For $F \in \mathbb{Z}[x]$ *a polynomial with integer coefficients of bitsize* τ, $\mathbb{R}$ISOLATE *computes isolating intervals with* $\tilde{O}(n^3\tau^2)$ *bit operations.*

Proof. It remains to prove the complexity bounds and the claim on the width of the isolating intervals. According to Appendix 6.1, the computation of an approximate logarithmic root bound $\Gamma \in \mathbb{N}$ as defined in Section 2.2 needs $\tilde{O}(n^2\Gamma_F)$ bit operations. For a fixed precision ρ, the total cost for running DCM$^\rho$ and CERTIFY$^\rho$ is bounded by

$$\tilde{O}(n(\Sigma_f + n\log n)(n\Gamma + \rho - \log\sigma_f)) = \tilde{O}(n(\Sigma_F + n\Gamma)(n\Gamma + \rho - \log\sigma_F))$$

bit operations; see Theorem 13 and Theorem 16. Since we double ρ in each step and succeed for $\rho \geq \rho_f^{\max}$, ρ is always bounded by $2\rho_f^{\max} = O(\Sigma_f + n) = O(\Sigma_F + n\Gamma)$ and we need at most a logarithmic number of rounds. Hence, it follows that (up to logarithmic factors) the total cost is dominated by the cost for the last run which is $\tilde{O}(n(\Sigma_F + n\Gamma)^2)$. Furthermore, we have to approximate the coefficients of f to $O(\Sigma_f + n) = O(\Sigma_F + n\Gamma)$ bits after the binary point. Hence, the coefficients of F have to be approximated to $O(\Sigma_F + n\Gamma)$ bits after the binary point; see Section 2.3 for more details. From our construction of f and Γ, it holds that $\Gamma < 8\log n + 1 + \Gamma_F$. Hence, we can replace Γ by Γ_F in the above complexity bounds.

For the special case where $F = A_n \cdot x^n + \cdots + A_0$ is a polynomial with integer coefficients of bit size τ, we first divide F by its leading coefficient to meet the requirements in (1.4). Then, the bound on the bit complexity follows from the above general bound (applied to $F(x)/A_n$) and the fact that $\Sigma_F = \tilde{O}(n\tau)$; see Appendix 6.2.

The estimate on the size of the isolating intervals is due to the following consideration: An interval I which contains the root $z = \frac{\xi}{2^{\Gamma+1}}$ of f is not subdivided by DCM$^\rho$ if $w(I) \leq \sigma(z,f)/(8n^2)$. Hence, any interval J_k which is returned by DCM$^\rho$ as an isolating interval for z is the extension $\tilde{I} = (a - \frac{w(I)}{2n}, b + \frac{w(I)}{2n})$ of an interval $I = (a,b)$ with $w(I) > \sigma(z,f)/(16n^2)$, and thus $w(J) = 2^{\Gamma+1}w(I) > \sigma(\xi,F)/(16n^2)$. From our construction, the $\frac{w(I)}{n}$-neighborhood of I isolates z as well. This shows that $w(J) = 2^{\Gamma+1}w(I) < 4\Gamma n\sigma(z_i,f) = 2n\sigma(\xi,F)$. $\square$

4.4. Some Remarks

4.4.1. On the Complexity Analysis for Integer Polynomials

We remark that, in order to achieve the complexity bound $\tilde{O}(n^3\tau^2)$ for integer polynomials, the subroutine CERTIFY$^\rho$ and its analysis is actually not needed. Namely, due to our considerations in Appendix 6.2, we can compute explicit upper bounds for Σ_f (in terms of n and τ) and, thus, also an explicit upper bound $\rho^*(n,\tau)$ for $\rho_f^{\max}$ which matches $\rho_f^{\max}$ at least with respect to worst case complexity. Then, Theorem 14 guarantees that DCM$^{\rho^*(n,\tau)}$ computes isolating intervals for all real roots of f. Unfortunately, this approach cannot be considered practical at all because such upper bounds usually tend to be much larger than the actual $\rho_f^{\max}$. We would like to emphasize on the fact that our algorithm is output sensitive in the way that it demands for a precision which is not much larger than ρ_f.

At this point, we even conjecture that, for any bisection method, the bound on the bit complexity as achieved by our algorithm is optimal (up to log-factors). We are not aware of any lower

bounds for the bit complexity of root isolation algorithms, and we can also not provide a rigorous proof for the optimality of our bound. However, the intuition behind our claim is that, for a Mignotte polynomial F, the bounds on the precision demand and the size of the recursion tree seem to be optimal. Namely, any bisection algorithm needs $\Theta(n\tau)$ steps to separate two roots of F with pairwise distance $2^{-\Theta(n\tau)}$. In addition, if we perturb the coefficients of F by more than $2^{-\Theta(n\tau)}$, the two ordinary roots can move by more than their initial separations, hence it seems that a precision of size $\Theta(n\tau)$ is needed to isolate these roots from each other.

We finally remark that, in practice, it may be advantageous to start with the exact representation of the rational polynomial F/A_n and not with an (artificial) approximation to a certain number ρ of bits. In this case, we propose to use the classical Descartes method with exact arithmetic for the first iterations, and only if the exact representation of the polynomials constructed during the subdivision process exceeds the given working precision ρ, we switch to our modified Descartes method, where approximations are considered. However, as already argued above, we do not expect that this approach yields any improvement with respect to worst case bit complexity.

4.4.2. On Efficient Implementation

We formulated our algorithm in a way to make it accessible to the complexity analysis but still feasible and efficient for an implementation. Nevertheless, we recommend to consider a slight modification of our algorithm when actually implementing it.

For our certification step CERTIFY$^\rho$, the most obvious modification is to only subdivide the region $\mathscr{R}$ instead of the entire interval $(-\frac{1}{2},\frac{1}{2})$. More precisely, $\mathscr{R}$ decomposes into intervals L_j "in between" the isolating intervals J_k. Then, we approximate the polynomials f_{L_j} to ρ bits after the binary point and recursively proceed each L_j in a similar way as proposed in CERTIFY$^\rho$. An experimental implementation of our algorithm in MAPLE has shown that, following this approach, the running time for the certification step is almost negligible, whereas, for the original formulation, it is approximately of the same magnitude as the running time for DCM$^\rho$.

Furthermore, we propose to also use the inclusion predicate based on Descartes' Rule of Signs. With respect to complexity, our inclusion predicate based on the $\mathscr{T}'[3/2](\cdot)$-test (see Corollary 7) is comparable to Descartes' Rule of Signs, where we check whether f has exactly one sign variation for a certain interval. However, in practice, this subtle difference is crucial because already $\log n$ bisection steps more for each root may render an algorithm inefficient. As an alternative, for an interval I, we propose to check whether there exists a "good" approximation g of f_I with $\mathrm{var}(g,(0,1)) = 1$.[15] Namely, if there exists such a g, we can proceed with $\tilde{f}_I := g$ which has exactly one root in I. Thus, it is easy to refine I (via simple bisection or quadratic interval refinement) such that $\mathscr{T}[g'](0,2)$ holds as well.

We finally report on an interesting behavior of the proposed method. It is easy to see that, for small intervals $I = (a,b)$, the leading coefficients of $f_I(x) = f(a + w(I)\cdot x)$ are considerably smaller than the first-order coefficients. Since we only consider a certain number $\rho_I \le \rho$ of bits after the binary point, the approximations $\tilde{f}_I$ can be chosen of a considerably lower degree than f_I. As a consequence, the cost at such an interval is considerably reduced because we have to compute the polynomial $\tilde{f}_{I^+} = \tilde{f}_I(\frac{1}{4n} + (1+\frac{1}{2n})\cdot x)$ which is expensive for large degrees. In

[15] Since $\tilde{f}_I$ approximates f_I to ρ_I bits after the binary point, we can derive an interval approximation $[f_{I,\mathrm{rev}}] := [a_0^-, a_0^+] + \cdots [a_n^-, a_n^+]\cdot x^n$ of $f_{I,\mathrm{rev}}$ to $\rho_I - n$ bits after the binary point. We can then easily check whether there exists a polynomial $h \in [f_{I,\mathrm{rev}}]$ with $\mathrm{var}(h) = 1$, and transform h back to $g(x) = (x-1)^n \cdot h(1/(x-1))$. The so-obtained polynomial g approximates f_I to at least $\rho_I - 2(n+1)$ bits after the binary point and $\mathrm{var}(g,[0,1]) = 1$. Notice that, following this approach, the precision ρ_I has to be updated accordingly.

particular, for a polynomial with two very nearby roots (such as Mignotte polynomials), this behavior can be clearly observed. More precisely, when refining an interval I that contains two nearby roots, the degree of $\tilde{f}_I$ decreases in each bisection step and eventually equals 2 for I small enough. We consider this behavior as quite natural because f_I implicitly captures the information on the location of the roots in a neighborhood of I, whereas the influence of all other roots becomes almost negligible.

5. Conclusion

We presented a novel deterministic algorithm to isolate the real roots of a square-free polynomial F with arbitrary real coefficients. Our analysis shows that the hardness of isolating the real roots exclusively depends on the location of the roots and not on the coefficient type of F. Furthermore, the overall running time is significantly reduced by considering approximations at each node of the recursion tree. In particular, for integer polynomials, we achieve an improvement with respect to worst case bit complexity by a factor $n = \deg F$ compared to other practical methods such as the Descartes method, the continued fraction method, or the Sturm method. The improvement stems from the fact that exact arithmetic produces too much information for the task of root isolation and, thus, a significant overhead of computation. Hence, for the main part, we consider our result to be the missing theoretical proof of a fact that has already been observed in practice, namely, that using approximate but certified arithmetic instead of exact arithmetic yields a significant improvement. We are confident that this result does not only hold for the Descartes method but also for a majority of the known real roots solvers and encourage other researchers to develop corresponding approximate variants.

Very recent work [37] shows how to combine the Descartes method with Newton iteration to improve upon the linear convergence of the original Descartes method. The algorithm NEWDSC from [37] can only be used to isolate the real roots of an integer polynomial F, and it achieves a worst case bit complexity of $\tilde{O}(n^3\tau)$. Hence, the major remaining research question is whether combining the two approaches, that is, using approximate arithmetic as proposed in this paper and using Newton iteration as proposed in [37], yields a practical algorithm that achieves record complexity bounds comparable to the bounds achieved by Pan's method.

References

[1] A. G. Akritas. There is no "Descartes' method". In *Computer Algebra in Education*, pages 19–35, 2008.

[2] A. G. Akritas and A. Strzeboński. A comparative study of two real root isolation methods. *Nonlinear Analysis:Modelling and Control*, 10(4):297–304, 2005.

[3] A. Alesina and M. Galuzzi. A new proof of Vicent's theorem. *L'Enseignement Mathematique*, 44:219–256, 1998.

[4] A. Alesina and M. Galuzzi. Addendum to the paper "A new proof of Vicent's theorem". *L'Enseignement Mathematique*, 45:379–380, 1999.

[5] A. Alesina and M. Galuzzi. Vincent's theorem from a modern point of view. *Categorical Studies in Italy 2000, Rendiconti del Circolo Matematico di Palermo, Serie II*, 64:179–191, 2000.

[6] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real and Algebraic Geometry*. Springer, 2nd edition, 2006.

[7] E. Berberich, P. Emeliyanenko, and M. Sagraloff. An elimination method for solving bivariate polynomial systems: Eliminating the usual drawbacks. In *ALENEX: Algorithm Engineering and Experiments*, pages 35–47, 2011.

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

[9] M. Burr and F. Krahmer. SqFreeEVAL: An (almost) optimal real-root isolation algorithm. *Journal of Symbolic Computation*, 47(2):153–166, 2012.

[10] G. E. Collins and A. G. Akritas. Polynomial real root isolation using descarte's rule of signs. In *SYMSAC: Symposium on Symbolic and Algebraic Computation*, pages 272–275, 1976.

[11] Z. Du, V. Sharma, and C. Yap. Amortized bounds for root isolation via Sturm sequences. In *SNC: Symbolic and Numeric Computation*, pages 113–130. 2007.

[12] A. Eigenwillig. On multiple roots in Descartes' rule and their distance to roots of higher derivatives. *Journal of Computational and Applied Mathematics*, 200(1):226–230, 2007.

[13] A. Eigenwillig. *Real Root Isolation for Exact and Approximate Polynomials using Descartes' Rule of Signs*. PhD thesis, Universität des Saarlandes, May 2008.

[14] A. Eigenwillig, L. Kettner, W. Krandick, K. Mehlhorn, S. Schmitt, and N. Wolpert. An exact descartes algorithm with approximate coefficients. In *CASC: Computer Algebra in Scientific Computing*, pages 138–149, 2005.

[15] A. Eigenwillig, V. Sharma, and C. K. Yap. Almost tight recursion tree bounds for the descartes method. In *ISSAC: International Symposium on Symbolic and Algebraic Computation*, pages 71–78, 2006.

[16] J. Gerhard. Modular algorithms in symbolic summation and symbolic integration. *LNCS, Springer*, 3218, 2004.

[17] X. Gourdon. *Combinatoire, Algorithmique et Géométrie des Polynômes*. Thèse, École polytechnique, 1996.

[18] M. Hemmer, E. P. Tsigaridas, Z. Zafeirakopoulos, I. Z. Emiris, M. I. Karavelas, and B. Mourrain. Experimental evaluation and cross-benchmarking of univariate real solvers. In *SNC: Symbolic Numeric Computation*, pages 45–54, 2009.

[19] J. R. Johnson and W. Krandick. Polynomial real root isolation using approximate arithmetic. In *ISSAC: International Symposium on Symbolic and Algebraic Computation*, pages 225–232, 1997.

[20] M. Kerber and M. Sagraloff. Efficient real root approximation. In *ISSAC: International Symposium on Symbolic and Algebraic Computation*, pages 209–216, 2011.

[21] M. Kerber and M. Sagraloff. A worst-case bound for topology computation of algebraic curves. *Journal of Symbolic Computation*, 47(3):239–258, 2012.

[22] W. Krandick and K. Mehlhorn. New bounds for the Descartes method. *Journal of Symbolic Computation*, 41(1):49–66, 2006.

[23] J. McNamee and V. Pan. *Numerical Methods for Roots of Polynomials -*. Number 2 in Studies in Computational Mathematics. Elsevier Science, 2013.

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

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

[26] K. Mehlhorn, M. Sagraloff, and P. Wang. From approximate factorization to root isolation. In *ISSAC: International Symposium on Symbolic and Algebraic Computation*, pages 283–290, 2013.

[27] B. Mourrain, F. Rouillier, and M.-F. Roy. Bernsteins basis and real root isolation. *Combinatorial and Computational Geometry*, 52:459–478, 2005.

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

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

[30] N. Obreschkoff. *Zeros of Polynomials*. Marina Drinov, Sofia, 2003. Translation of the Bulgarian original.

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

[32] V. Y. Pan. Solving a polynomial equation: some history and recent progress. *SIAM Review*, 39(2):187–220, 1997.

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

[34] F. Roullier and P. Zimmermann. Efficient isolation of a polynomial's real roots. *Journal of Computational and Applied Mathematics*, pages 33–50, 2004.

[35] M. Sagraloff. A general approach to isolating roots of a bitstream polynomial. *Mathematics in Computer Science*, 4(4):481–506, 2010.

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

[37] M. Sagraloff. When Newton meets Descartes: a simple and fast algorithm to isolate the real roots of a polynomial. In *ISSAC: International Symposium on Symbolic and Algebraic Computation*, pages 297–304, 2012.

[38] M. Sagraloff and C.-K. Yap. A simple but exact and efficient algorithm for complex root isolation. In *ISSAC: International Symposium on Symbolic and Algebraic Computation*, pages 353–360, 2011.

[39] A. Schönhage. The fundamental theorem of algebra in terms of computational complexity, 1982. Manuscript, Department of Mathematics, University of Tübingen. Updated 2004.

[40] A. Schönhage. Quasi-GCD computations. *Journal of Complexity*, 1(1):118–137, 1985.

[41] V. Sharma. Complexity of real root isolation using continued fractions. *Theoretical Computer Science*, 409(2):292–310, 2008.

[42] A. W. Strzebonski and E. P. Tsigaridas. Univariate real root isolation in an extension field. In *ISSAC: International Symposium on Symbolic and Algebraic Computation*, pages 321–328, 2011.

[43] A. W. Strzebonski and E. P. Tsigaridas. Univariate real root isolation in multiple extension fields. In *ISSAC: International Symposium on Symbolic and Algebraic Computation*, pages 343–350, 2012.

[44] E. Tsigaridas and I. Emiris. On the complexity of real root isolation using continued fractions. *Theoretical Computer Science*, pages 158–173, 2008.

[45] J. von zur Gathen and J. Gerhard. Fast algorithms for Taylor shifts and certain difference equations. In *ISSAC: International Symposium on Symbolic and Algebraic Computation*, pages 40–47, New York, NY, USA, 1997. ACM.

[46] J.-C. Yakoubsohn. Numerical analysis of a bisection-exclusion method to find zeros of univariate analytic functions. *Journal of Complexity*, 21(5):652 – 690, 2005.

[47] C. Yap. *Fundamental Problems in Algorithmic Algebra*. Oxford University Press, 2000.

6. Appendix

6.1. Approximating Γ_F

Theorem 1. An integer $\Gamma \in \mathbb{N}$ with

$$\Gamma_p \leq \Gamma < 8\log n + \Gamma_p \tag{6.1}$$

can be computed with $\tilde{O}(n^2\Gamma_p)$ bit operations. The computation uses an approximation of F to $L = O(n\Gamma_F)$ bits after the binary point.

Proof. Consider the *Cauchy polynomial*

$$\bar{F}(x) := |A_n|x^n - \sum_{i=0}^{n-1} |A_i|x^i$$

of F. Then, according to [13, Proposition 2.51], $\bar{F}$ has a unique positive real root $\xi \in \mathbb{R}^+$, and the following inequality holds:

$$\max_{i=1,\ldots,n} |\xi_i| \leq \xi < \frac{n}{\ln 2} \cdot \max_{i=1,\ldots,n} |z_i| < 2n \cdot \max_{i=1,\ldots,n} |z_i|.$$

It follows that $\bar{F}(x) > 0$ for all $x \geq \xi$ and $\bar{F}(x) < 0$ for all $x < \xi$. Furthermore, since $\bar{F}$ coincides with its own Cauchy polynomial, each complex root of $\bar{F}$ has absolute value less than or equal to ξ. Let k_0 be the smallest non-negative integer k with $\bar{F}(2^k) > 0$ (which is equal to the smallest k with $2^k > \xi$). Our goal is to compute an integer Γ with $k_0 \leq \Gamma \leq k_0 + 1$. Namely, if Γ fulfills the latter inequality, then

$$\max(1, \max_i |z_i|)) \leq \max(1, \xi) \leq 2^\Gamma < 4\max(1, \xi) < 8n \cdot \max(\max_i |z_i|, 1),$$

and thus Γ fulfills inequality (6.1). In order to compute a Γ with $k_0 \leq \Gamma \leq k_0 + 1$, we use exponential and binary search (try $k = 1, 2, 4, 8, \ldots$ until $\bar{F}(2^k) > 0$ and, then, perform binary search on the interval $k/2$ to k) and approximate evaluation of $\bar{F}$ at the points 2^k: More precisely, we evaluate $\bar{F}(2^k)$ using interval arithmetic with a precision ρ (using fixed point arithmetic) which guarantees that the width w of $\mathfrak{B}(\bar{F}(2^k), \rho)$ is smaller than $1/4$, where $\mathfrak{B}(E, \rho)$ is the interval obtained by evaluating a polynomial expression E via interval arithmetic with precision ρ for the basic arithmetic operations; see [20, Section 4] for details. We use [20, Lemma 3] to estimate the cost for each such evaluation: Since $\bar{F}$ has coefficients of size less than $2^n\mathrm{Mea}(F)$, we have to choose ρ such that

$$2^{-\rho+2}(n+1)^2\mathrm{Mea}(F)2^{n+nk} < 1/4$$

in order to ensure that $w < 1/4$. Hence, ρ is bounded by $O(\log \mathrm{Mea}(F) + nk)$ and, thus, each interval evaluation needs $\tilde{O}(n(\log \mathrm{Mea}(F) + nk))$ bit operations. We now use exponential plus binary search to find the smallest k such that $\mathfrak{B}(\bar{F}(2^k), \rho)$ contains only positive values. The following argument then shows that $k_0 \leq k \leq k_0 + 1$: Obviously, we must have $k \geq k_0$ since $\bar{F}(2^k) < 0$ and $\bar{F}(2^k) \in \mathfrak{B}(\bar{F}(2^k), \rho)$ for all $k < k_0$. Furthermore, the point $x = 2^{k_0+1}$ has distance more than 1 to each of the roots of $\bar{F}$, and thus $|\bar{F}(2^{k_0+1})| \geq |A_n| \geq 1/4$. Hence, it follows that $\mathfrak{B}(\bar{F}(2^{k_0} + 1), \rho)$ contains only positive values. For the search, we need

$$O(\log k_0) = O(\log\log \xi) = O(\log(\log n + \Gamma_F))$$

iterations, and the cost for each of these iterations is bounded by $\tilde{O}(n(\log \mathrm{Mea}(F) + nk_0)) = \tilde{O}(n^2\Gamma_F)$ bit operations. □

6.2. Integer Polynomials

For a polynomial F with integer coefficients of bit size τ, we aim to show that $\Sigma_F = \tilde{O}(n\tau)$. We proceed in two steps: First, we cluster the roots ξ_i of F into subsets consisting of nearby roots. Second, we apply the generalized Davenport-Mahler bound [11, 13] to the roots of F.

W.l.o.g., we can assume that $\sigma(\xi_1, F) \leq \cdots \leq \sigma(\xi_n, F)$. For $h \in \mathbb{N}$, we denote $i(h)$ the maximal index i with $\sigma(\xi_i, F) \leq 2^{-h}$ and $R(h) := \{\xi_1, \ldots, \xi_{i(h)}\}$ the corresponding set of roots. If $h \leq \log(1/\sigma_F)$, then R contains at least two roots. For a fixed h, we are interested in a partition of $R := R(h)$ into disjoint subsets $R_1, \ldots, R_l$ that consist of nearby points, only.

Lemma 18. *Suppose that $h \leq \log(1/\sigma_F)$. Then, there exists a partition of $R := R(h)$ into disjoint sets $R_1, \ldots, R_l$ such that $|R_i| \geq 2$ for all $i \in \{1, \ldots, l\}$ and $|\xi - \xi'| \leq n2^{-h}$ for all ξ, $\xi' \in R_i$.*

Proof. We initially set $R_1 := \{\xi_1\}$. Then, we add all roots ξ_i to R_1 that satisfy $|\xi_i - \xi_1| \leq 2^{-h}$. For each root in R_1, we proceed in the same way. More precisely, for each $\xi \in R_1$, we add those roots $\xi' \in R$ to R_1 with $|\xi - \xi'| \leq 2^{-h}$. If no further root can be added to R_1, we consider the set $R \backslash R_1$ of the remaining roots and treat it in exactly the same manner. Finally, we end up with a partition $R_1, \ldots, R_l$ of R such that, for any two points in any R_i, their distance is less than or equal to $(|R_i| - 1)2^{-h} < n \cdot 2^{-h}$. Furthermore, each of the sets R_i must contain at least two roots since $\sigma(\xi_i, F) \leq 2^{-h}$ for all $i = 1, \ldots, i(h)$. $\square$

We now fix a $h > 4 \log n$ and consider a partitioning of $R := R(h)$ as in the above lemma. Then, we define $\mathscr{G}_i$ to be the directed graph on each R_i which connects consecutive roots of R_i in ascending order of their absolute values. We further define $\mathscr{G} := (R, E)$ as the union of all $\mathscr{G}_i$. Then, $\mathscr{G}$ is a directed graph on R with the following properties:

(1) each edge $(\alpha, \beta) \in E$ satisfies $|\alpha| \leq |\beta|$,
(2) $\mathscr{G}$ is acyclic, and
(3) the in-degree of each node is at most 1.

Hence, we can apply the generalized Davenport-Mahler bound [11, 13] to $\mathscr{G}$:

$$\prod_{(\alpha,\beta) \in E} |\alpha - \beta| \geq \frac{1}{(\sqrt{n+1} \cdot 2^{\tau})^{n-1}} \cdot \left(\frac{\sqrt{3}}{n}\right)^{\#E} \cdot \left(\frac{1}{n}\right)^{n/2}$$

Since each set R_i contains at least 2 roots, we must have $i(h) > \#E \geq i(h)/2$. Furthermore, for each edge $(\alpha, \beta) \in E$, we have $|\alpha - \beta| \leq n \cdot 2^{-h}$. Thus, it follows that (notice that $n \cdot 2^{-h} < 1$)

$$(n \cdot 2^{-h})^{\frac{i(h)}{2}} > \frac{1}{(\sqrt{n+1} \cdot 2^{\tau})^{n-1}} \cdot \left(\frac{\sqrt{3}}{n}\right)^{i(h)} \cdot \left(\frac{1}{n}\right)^{n/2} > \frac{1}{(n+1)^n \cdot 2^{n\tau}} \cdot \left(\frac{3}{n^2}\right)^{i(h)/2},$$

and thus

$$i(h) < \frac{2n(\tau + \log(n+1))}{\log 3 - 3\log n + h} < \frac{2n(\tau + \log(n+1))}{h/4} = \frac{8n(\tau + \log(n+1))}{h}.$$

From the latter inequality, we conclude that $\log(1/\sigma_F) < 4n(\tau + \log(n+1)) + 1$ since, otherwise, there exists an h with $4n(\tau + \log(n+1)) \leq h \leq \log(1/\sigma_F)$ and $i(h) < 2$ which is not possible. For the bound on Σ_F, it suffices to consider only the roots $\xi_1, \ldots, \xi_k$ with separation less than $1/n^4$ since each root with a larger separation contributes with at most $\max(\tau + 1, 4\log n)$ to Σ_F. Thus, $\Sigma_F = \tilde{O}(n\tau)$ follows from

$$-\sum_{i=1}^{k} \log \sigma(\xi_i, F) < \sum_{h=1}^{\lceil 4n(\tau + \log(n+1)) \rceil} i(h) < 8n(\tau + \log(n+1)) \sum_{h=1}^{\lceil 4n(\tau + \log(n+1)) \rceil} \frac{1}{h} = O(n\tau \log(n\tau)).$$

Algorithm 1 DCM

Require: polynomial $f = \sum_{0 \le i \le n} a_i x^i \in \mathbb{R}[x]$ as defined in (2.1)

Ensure: returns a list $\mathscr{O}$ of disjoint isolating intervals for all real roots of f
{only in the REAL-RAM model}

$I_0 := (-\frac{1}{2}, \frac{1}{2})$
$f_{I_0}(x) := f(-\frac{1}{2} + x)$
$\mathscr{A} := \{ (I_0, f_{I_0}) \}$; $\mathscr{O} := \emptyset$ {list of active and isolating intervals}
repeat
  (I, f_I) some element in $\mathscr{A}$ with $I = (a,b)$; delete (I, f_I) from $\mathscr{A}$
  $f_{I^+} := f_I\left(-\frac{1}{4n} + \left(1 + \frac{1}{2n}\right)x\right)$ and $f_{I^+,\mathrm{rev}}(x) = \sum_{i=0}^{n} h_i x^i := (1+x)^n \cdot f_{I^+}\left(\frac{1}{1+x}\right)$
  if $\mathrm{var}(f_{I^+,\mathrm{rev}}) = 0$ **then**
    do nothing
  else
    if $\mathbf{t}[(f_I)', 3/2](0,2) > 0$ **then**
      $s := \mathrm{sign}(f_{I^+}(0) \cdot f_{I^+}(1))$
      if $s \ge 0$ **then**
        do nothing
      else
        if I^+ does not intersect any interval in $\mathscr{O}$ **then**
          add I^+ to $\mathscr{O}$
        else
          do nothing
        end if
      end if
    else
      subdivide I into $I_l := (a, m_I)$ and $I_r := (m_I, b)$
      $f_{I_\ell} := f_I\left(\frac{x}{2}\right)$ and $f_{I_r} := f_I\left(\frac{x+1}{2}\right) = f_{I_\ell}(x+1)$
      add (I_l, f_{I_ℓ}) and (I_r, f_{I_r}) to $\mathscr{A}$
    end if
  end if
until $\mathscr{A}$ is empty
return $\mathscr{O}$

Algorithm 2 DCM$^{\rho}$

Require: polynomial $f=\sum_{0\le i\le n} a_i x^i \in \mathbb{R}[x]$ as in (2.1) and a $\rho \in \mathbb{N}$

Ensure: returns "insufficient precision" or a list $\mathscr{O}=\{J_k, s_{k,\ell}, s_{k,r}, B_k\}$ of disjoint isolating intervals $J_k=(c_k,d_k)$ for some of the real roots of f (and $s_{k,\ell}=\mathrm{sign}(f(c_k))$, $s_{k,r}=\mathrm{sign}(f(d_k))$ and $0<B_k\le \min(|f(c_k)|,|f(d_k)|)$.

$I_0:=(-\frac{1}{2},\frac{1}{2})$
$\tilde{f}$ a $(\rho+n+1)$-binary approximation of f
$\tilde{f}_{I_0}$ a $(\rho+1)$-binary approximation of $\tilde{f}(-\frac{1}{2}+x)$ $\{\Rightarrow \tilde{f}_{I_0}\in [f_{I_0}]_{2^{-\rho}}\}$
$\mathscr{A}:=\{(I_0,\tilde{f}_{I_0},\rho)\}$; $\mathscr{O}:=\emptyset$ {list of active and isolating intervals}

repeat
  $(I,\tilde{f}_I,\rho_I)$, where $I:=(a,b)$, some element in $\mathscr{A}$; delete $(I,\tilde{f}_I,\rho_I)$ from $\mathscr{A}$
  $\tilde{f}_{I^+}(x):=\tilde{f}_I\left(-\frac{1}{4n}+\left(1+\frac{1}{2n}\right)x\right)$ and $\tilde{h}(x)=\sum_{i=0}^n \tilde{h}_i x^i:=(1+x)^n\cdot \tilde{f}_{I^+}\left(\frac{1}{1+x}\right)$
  if $\tilde{h}_i>-2^{n+2-\rho_I}$ for all i or $\tilde{h}_i<2^{n+2-\rho_I}$ for all i **then**
    do nothing
  else
    if t$[(\tilde{f}_I)',3/2](0,2)>-n\cdot 2^{n+1-\rho_I}$ **then**
      $\hat{f}_I(x):=\tilde{f}_I(x)+\mathrm{sign}((\tilde{f}_I)'(0))\cdot n\cdot 2^{n+1-\rho_I}\cdot x$
      $\lambda^-:=\tilde{f}_{I^+}(0)-2^{n-1-\rho_I}$, $\lambda^+:=\tilde{f}_{I^+}(1)+(4n+1)2^{n-1-\rho_I}$ and $\lambda:=\tilde{f}_I(-1/n)-2^{n+1-\rho_I}$.
      if $\tilde{I}:=\left(a-\frac{w(I)}{2n},b+\frac{w(I)}{2n}\right)$ intersects no J for all $(J,s_{J,\ell},s_{J,r},B_J)\in\mathscr{O}$ **and** $\lambda^-\cdot\lambda^+<0$
      and $\min(|\lambda^-|,|\lambda^+|)>n2^{n+3-\rho_I}$ **and** $|\lambda|>n^2 2^{\deg(\hat{f}_I)+7+n-\rho_I}$ **then**
        add $(\tilde{I},\mathrm{sign}(\lambda^-),\mathrm{sign}(\lambda^+),\min(|\lambda^-|,|\lambda^+|)-2^{n+3-\rho_I}n)$ to $\mathscr{O}$
            $\{\Rightarrow \tilde{I}$ contains a root ξ of f and the $\frac{w(I)}{n}$-neighborhood of I is isolating for $\xi\}$
        else
          do nothing $\{\tilde{J}$ is already isolating for $\xi\}$
        end if
      else
        do nothing
      end if
    else
      if $\rho_I<0$ **then**
        return "insufficient precision"
      else
        if $\rho_I<2$ **then**
          return "insufficient precision"
        else
          Subdivide I into $I_\ell:=(a,m_I)$ and $I_r:=(m_I,b)$
          $\tilde{f}_{I_\ell}$ an ρ_I-binary approximation of $\tilde{f}_I\left(\frac{x}{2}\right)$ $\{\Rightarrow \tilde{f}_{I_l}\in[f_{I_\ell}]_{2^{-(\rho_I-1)}}\}$
          $\tilde{f}_{I_r}$ an (ρ_I-1)-binary approximation of $\tilde{f}_I\left(\frac{1+x}{2}\right)$ $\{\Rightarrow \tilde{f}_{I_r}\in[f_{I_r}]_{2^{-(\rho_I-2)}}\}$
          Add $(I_\ell,\tilde{f}_{I_\ell},\rho_I-1)$ and $(I_r,\tilde{f}_{I_r},\rho_I-2)$ to $\mathscr{A}$
        end if
      end if
    end if
until $\mathscr{A}$ is empty
return $\mathscr{O}$

Algorithm 3 CERTIFY$^{\rho}$

Require: polynomial $f = \sum_{0\le i\le n} a_i x^i \in \mathbb{R}[x]$ as defined in (2.1), an integer $\rho \in \mathbb{N}$, and the list $\mathscr{O} = \{(J_k, s_{k,\ell}, s_{k,r}, B_k)\}_{k=1,\dots,s}$ returned by DCM$^{\rho}$.

Ensure: returns "insufficient precision" or the list $\mathscr{L} = \{J_k\}_{k=1,\dots,s}$ of isolating intervals with the guarantee that, for each real root of f, there exists a corresponding interval in $\mathscr{L}$.

$I_0 := (-\frac{1}{2}, \frac{1}{2})$
$\tilde{f}$ a $(\rho+n+1)$-binary approximation of f
$\tilde{f}_{I_0}$ a $(\rho+1)$-binary approximation of $\tilde{f}(-\frac{1}{2}+x)$ $\{\Rightarrow \tilde{f}_{I_0} \in [f_{I_0}]_{2^{-\rho}}\}$
$\mathscr{A} := \{(I_0, \tilde{f}_{I_0}, \rho)\}$ {list of active intervals}

repeat
  $(I, \tilde{f}_I, \rho_I)$, where $I := (a,b)$, some element in $\mathscr{A}$; delete $(I, \tilde{f}_I, \rho_I)$ from $\mathscr{A}$.
  if $\bar{I} \cap \mathscr{R} = \bigcup_{i=1}^{s} L_i = \emptyset$ **then**
    do nothing
  else
    if $\mathbf{t}[\tilde{f}_I, 3/2](0,1) > -n \cdot 2^{-\rho_I+2}$ **then**
      if $|\tilde{f}_I(0) + \mathrm{sign}(\tilde{f}_I(0)) \cdot 2^{-\rho_I+2} n| > n^2 \cdot 2^{-\rho_I+6}$ **then**
        do nothing {$\bar{I}$ contains no root of f}
      else
        return "insufficient precision" $\{\rho < \rho_f^{\max}\}$
      end if
    else
      $\tilde{h}(x) := \sum_{i=0}^{n} \tilde{h}_i x^i = (1+x)^n \cdot (\tilde{f}_I)'(\frac{1}{1+x})$
      if $\tilde{h}_i < n \cdot 2^{n-\rho_I}$ for all i (or $\tilde{h}_i > -n \cdot 2^{n-\rho_I}$ for all i) **then**
        $g(x) := \tilde{f}_I(x) - n \cdot 2^{n-\rho_I}$ (or $g(x) := \tilde{f}_I(x) + n \cdot 2^{n-\rho_I}$, respectively);
        if for each $L_i = [q_\ell, q_r]$, $\min(|\lambda(q_\ell)|, |\lambda(q_r)|) > n \cdot 2^{n+3-\rho_I}$ **and** $\lambda(q_l) \cdot \lambda(q_r) < 0$ **then**
          do nothing {$\bar{I} \cap \mathscr{R}$ contains no root of f; $\lambda(q_l)$, $\lambda(q_r)$ defined as in (4.15)}
        else
          return "insufficient precision" $\{\rho < \rho_f^{\max}\}$
        end if
      else
        if $\rho_I < 2$ **then**
          return "insufficient precision"
        else
          Subdivide I into $I_l := (a, m_\ell)$ and $I_r := (m_\ell, b)$
          $\tilde{f}_{I_\ell}$ an ρ_I-binary approximation of $\tilde{f}_I\left(\frac{x}{2}\right)$ $\{\Rightarrow \tilde{f}_{I_\ell} \in [f_{I_\ell}]_{2^{-(\rho_I-1)}}\}$
          $\tilde{f}_{I_r}$ an (ρ_I-1)-binary approximation of $\tilde{f}_\ell\left(\frac{1+x}{2}\right)$ $\{\Rightarrow \tilde{f}_{I_r} \in [f_{I_r}]_{2^{-(\rho_I-2)}}\}$
          Add $(I_l, \tilde{f}_{I_\ell}, \rho_I - 1)$ and $(I_r, \tilde{f}_{I_r}, \rho_I - 2)$ to $\mathscr{A}$
        end if
      end if
    end if
  end if
until $\mathscr{A}$ is empty
return "certification successful" {The region of uncertainty $\mathscr{R}$ contains no root of f}