A General Approach to Isolating Roots of a Bitstream Polynomial

Michael Sagraloff

Abstract. We describe a new approach to isolate the roots (either real or complex) of a square-free polynomial F with real coefficients. It is assumed that each coefficient of F can be approximated to any specified error bound and refer to such coefficients as *bitstream coefficients*. The presented method is exact, complete and deterministic. Compared to previous approaches [10, 12, 23] we improve in two aspects. Firstly, our approach can be combined with any existing subdivision method for isolating the roots of a polynomial with rational coefficients. Secondly, the approximation demand on the coefficients and the bit complexity of our approach is considerably smaller. In particular, we can replace the worst-case quantity $\sigma(F)$ by the average-case quantity $\prod_{i=1}^{n} \sqrt[n]{\sigma_i}$, where σ_i denotes the minimal distance of the $i-$th root ξ_i of F to any other root of F, $\sigma(F) := \min_i \sigma_i$, and $n = \deg F$. For polynomials with integer coefficients, our method matches the best bounds known for existing practical algorithms that perform exact operations on the input coefficients.

Keywords. real polynomial, root isolation, bitstream coefficients, root perturbation bounds, adaptive precision management.

1. Introduction

Root isolation is considered a fundamental problem in computer algebra, numerical analysis and geometric computing (cf. [30, 33, 37]). Given a polynomial $F(x) \in \mathbb{R}[x]$, we want to determine a set of disjoint intervals (discs) such that each of them contains exactly one root and their union covers all real (complex) roots of F.

We describe a new approach to isolate the roots of a square-free polynomial

$$F(x) = \sum_{i=0}^{n} A_i x^i \text{ with real coefficients } A_i, \text{ where } |A_i| < 2^L \text{ for all } i \text{ and } |A_n| \geq 1. \qquad (1)$$

We assume that each A_i can be approximated to any specified error bound and refer to such coefficients as *bitstream coefficients*. Our method is exact, complete and deterministic. As in [23], we isolate the roots of F by isolating the roots of a carefully chosen

	Approximation demand	Bit complexity
New Method	$\tilde{O}(nL+\Sigma(F))$	$\tilde{O}(n^2(L-\log\sigma(F))(nL+\Sigma(F)))$
Previous Method	$\tilde{O}(n(L-\log\sigma(F))$	$\tilde{O}(n^3(L-\log\sigma(F))^2)$

TABLE 1. $\Sigma(F) := \sum_{i=1}^{n} \log(\sigma_i^{-1})$ and $\tilde{O}$ omits polylogarithmic factors.

rational approximation $\tilde{F} \in \mathbb{Q}[x]$ of F. Compared to previous approaches [10, 12, 23] we improve in two aspects. Firstly, our approach can be combined with *any subdivision method* for root isolation of polynomials with rational coefficients (namely, the approximation $\tilde{F}$) such as the Descartes method, continued fraction solvers, or the Bolzano method; see the section on related work for a more comprehensive overview. Thus, it profits directly from the effectiveness of these methods. Secondly, the approximation demand on the coefficients and the bit complexity of our approach is considerably smaller. More precisely, for the i-th root ξ_i of F, let $\sigma_i = \sigma(\xi_i, F)$ be the minimal distance of ξ_i to any other root and let $\sigma(F) := \min_{i=1,\dots,n} \sigma_i$ be the root separation of F. We show that the worst-case quantity $\sigma(F)$ can be replaced by the average-case quantity $\prod_{i=1}^{n} \sqrt[n]{\sigma_i}$ in the complexity bounds. More precisely, in the corresponding bounds on the approximation demand and the bit complexity, we managed to replaced a factor $n(L-\log\sigma(F))$ by $nL - \log(\prod_{i=1}^{n} \sqrt[n]{\sigma_i}) = nL + \Sigma(F)$, where $\Sigma(F) := \sum_i \log(\sigma_i^{-1})$; see Table 1 for detailed complexity statements. The geometric mean of the root separations is never larger and frequently much smaller than the smallest root separation; we give specific examples in Section 2.3. Integer coefficients are a special case of Bitstream coefficients, that is, the streams are finite. For polynomials with integer coefficients, our method matches the best bounds known for existing practical algorithms which operate exactly on the input coefficients, that is, $\tilde{O}(n^4L^2)$. For very large coefficients, the Bitstream approaches sometimes outperform the methods for integer polynomials as the full precision of the coefficients might not be needed. This was already observed [16, 28] for the previous Bitstream solvers and should even more be true for the new method, in particular, for polynomials of larger degree. Our results are crucially based on the usage of an adaptive precision management in comparison to the usage of worst case perturbation bounds as proposed in [12] or [23].

Related Work. There are mainly two efficient methods to isolate the *real roots* of a square-free polynomial $F \in \mathbb{R}[x]$, namely, subdivision methods based on Descartes' Rule of Signs [3, 7, 13, 25, 28] and Sturm's Theorem [8, 21]. They both start on an initial interval and perform a recursive binary search. In practice, methods based on Descartes' Rule of Signs have proven to be more efficient [16, 17, 28] than Sturm's approach, but both approaches behave equally in terms of worst case complexity [8, 13, 20]. More precisely, for F a polynomial of degree n with integer coefficients of bitsize L, the induced subdivision tree has size $O(n(\log n + L))$ and isolating all real roots demands for $\tilde{O}(n^4L^2)$ bit operations. There exist several variants of subdivision methods dating back to Vincent's work [36], either using a Bernstein representation instead of the monomial basis or subdividing an interval not at the midpoint but at an arbitrary point by the use of root bounds (continued fraction methods [2, 32, 35, 36]). Although continued fraction methods yield

the same theoretical worst case bounds [22, 32, 35], experiments [1, 16, 35] have shown that they outperform other methods for hard instances such as Mignotte polynomials.
For the problem of isolating all *complex roots*, there is a considerable discrepancy between asymptotically fast algorithms and algorithms which are fast in practice. In the eighties, Schönhage [30] and Pan [26, 27] proposed algorithms with almost optimal complexity bounds $\tilde{O}(n^3L)$ but both methods lack evidence of being efficient in practice; see [15] for an implementation of the splitting circle method within the Computer Algebra system Pari/GP. In the numerical literature, there are many algorithms that are widely used and effective in practice but lack a guarantee on the global behavior; see [27] for discussion. Some global methods such as the Weierstrass or Durant-Kerner method that simultaneously approximate all roots seem ideal to achieve good complexity bounds and work well in practice, but their convergence and/or complexity analysis are open. In [29], a simple and efficient complex root isolation method is formulated. It is denoted CEVAL due to its similarities to its real counterpart EVAL [24]. EVAL and CEVAL are analytic but exact subdivision methods based on Weyl's approach (1924) (see [27] for a discussion) and both apply to a wider class of analytic functions as well. Recent results [29] show that, for polynomials with integer coefficients, both methods achieve the same complexity bounds as the Descartes or continued fraction method for isolating the real roots only.

Common to all above mentioned methods is that they operate exactly on the coefficients of the input polynomial. It is required that addition and sign test (with results $+$, 0, or $-$) are computable over the ring of coefficients. For the continued fraction strategy, multiplication, division, and approximate logarithm are also required. The previously mentioned practical methods are complete, exact, and efficient as long as the coefficients are rational (and of moderate complexity), but are infeasible for coefficients which are too large, algebraic, or even transcendental. It was suggested to replace the coefficients by small intervals and to execute the method using interval arithmetic. The first proposals [6, 18, 25, 28] were incomplete, in general; they all had to resort to exact arithmetic in the ring of coefficients for some input polynomials, namely, for inputs for which certain decisions (counting sign changes in a sequence of coefficients and determining the sign of the polynomial at subdivision points) could not be made reliably with interval arithmetic (e.g, for irrational coefficients). Recently, two different complete and exact methods [12, 23] to isolate the real roots of a square-free bitstream polynomial have been designed. The algorithm from 2005 is a randomized algorithm whereas that of 2009 is deterministic. Both methods can be considered as approximate versions of the classical VCA-bisection algorithm (due to Vincent, Collins and Akritas) based on Descartes' Rule of Signs. They both exploit that the roots of a polynomial continuously depend on its coefficients and, thus, use approximations $\tilde{F}$ of F to determine isolating intervals for the real roots of F. While in the first version $\tilde{F}$ is a polynomial with interval coefficients, the newer version works on a certain concrete rational approximation $\tilde{F} \in \mathbb{Q}[x]$. Both algorithms are driven by a guess of the separation of F and adaptively improve this guess and the approximation error during the algorithm. For their precision management, these solvers use worst case perturbation bounds for the roots of F and its approximation $\tilde{F}$. As a consequence, in most situations, the precision demand as well as the resulting running times are unnecessarily large. Whereas for polynomials of moderate degree this disadvantage

does not carry too much weight, these approaches become impractical for higher degrees. In his PhD Thesis [10], Eigenwillig presents a considerably improved, even though more complicated, version of the original algorithm [12]. Most importantly, the precision management is mostly decoupled from the guess on the separation. The latter is achieved by introducing additional tests based on the evaluation of the polynomial at randomly chosen subdivision points. Following this approach, the solver is allowed to run much longer for a chosen precision. In addition, it is guaranteed that the induced recursion tree is almost of the same size as the recursion tree which would have been induced by the exact Descartes method. We further remark that the analysis from [10, Section 3.3.8-3.3.9] can be modified to obtain similar bounds on the expected complexity and precision demand as the deterministic bounds achieved by our solver.

Common to all existing approaches is that they are crucially based on the Descartes method and do not directly extend to other methods such as continued fraction or even to methods for isolating complex roots. In this paper, we present a deterministic algorithm which addresses these tasks.

Outline. Section 2 introduces some notation, a root perturbation bound, the main predicates and their functionality. Our algorithm is presented in Section 3. Section 3.1 is dedicated to the problem of isolating the real roots only whereas Section 3.2 sketches how to extend our method for complex root isolation. In Section 4, we provide the results of our complexity analysis. We conclude in Section 5.

2. Preliminaries

Instead of isolating the roots of the given polynomial F as in (1), we consider the equivalent task of isolating the roots of a "scaled" polynomial f which is defined as follows: Let Γ be an integer bound on the modulus of all roots ξ_i of F and

$$f(x) = \sum_{i=0}^{n} a_i x^i := \frac{F(8\Gamma \cdot x)}{A_n}. \tag{2}$$

Then, the roots $z_1 = \xi_1/(8\Gamma), \ldots, z_n = \xi_n/(8\Gamma)$ of f are all contained within the disc $\Delta_{1/8}(0)$. We can assume that $\Gamma \leq \Gamma_{CB}$, where $\Gamma_{CB} := 1 + \max_i |A_i/A_n| < 2^{L+1}$ denotes the Cauchy Bound [37] for the roots of F. It follows that the absolute value of each coefficient of f is bounded by $(8\Gamma)^n 2^L = 2^{O(nL)}$. In practice, it might be worth to investigate in a more tight root bound Γ as described in [10, Section 2.4] in order to prevent the coefficients of f to become unnecessarily large. We further remark that the separations of corresponding roots of F and f scale by a factor 8Γ, that is, $\sigma(\xi_i, F) = 8\Gamma \cdot \sigma(z_i, f)$ and $\Sigma(f) = \sum_{i=1}^{n} \log \sigma(z_i, f)^{-1} = \Sigma(F) + n \log(8\Gamma) = O(nL + \Sigma(F))$.

We assume that the coefficients of F are given as infinite bitstreams, that is, for a given $\rho \in \mathbb{N}$, we can ask for an approximation of F to ρ bits after the binary point. More precisely, for each coefficient A_i, there exists 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., $\tilde{A}_i = \operatorname{sgn}(A_i) \lfloor |A_i| 2^{\rho} \rfloor 2^{-\rho}$. We call a polynomial $\tilde{F} \in \mathbb{Q}[x]$

obtained in this way a ρ-*binary approximation of* F. We remark that, in order to get a ρ-binary approximation of f, we have to approximate F to $O(nL+\rho)$ bits after the binary point because of the scaling operation $x \mapsto 8\Gamma x$.

2.1. Notation

We use two geometric objects throughout this paper, that is, intervals and discs. For an interval $I=(a,b)$, we denote $m(I) := \frac{a+b}{2}$ its midpoint, $w(I) := b-a$ its width and $r(I) := \frac{w(I)}{2}$ its radius. Discs in $\mathbb{C}$ are denoted by $\Delta_r(m)$, where r indicates the radius and m the center of the disc $\Delta_r(m)$.

For a univariate polynomial $g = g_n x^n + \ldots + g_0 \in \mathbb{C}[x]$ of degree n and an arbitrary non-negative $\mu \in \mathbb{R}_0^+$, we denote the family of all μ-*approximations of* g by

$$[g]_\mu := \left\{ \tilde{g}(x) = \sum_{i=0}^{n} \tilde{g}_i x^i \in \mathbb{C}[x] : \tilde{g}_n = g_n \text{ and } |g_i - \tilde{g}_i| \le \mu \text{ for all } i < n \right\} \tag{3}$$

and each $\tilde{g} \in [g]_\mu$ a μ-*approximation of* g. We remark that any ρ-binary approximation $\tilde{f}$ of f is contained in $[f]_{2^{-\rho}}$ because f has has an integer leading coefficient $a_n = (8\Gamma)^n$ and the coefficients of $\tilde{f}$ approximate those of f to an error of $2^{-\rho}$.

Finally, for arbitrary values $m \in \mathbb{C}$ and $\lambda \in \mathbb{R}\backslash\{0\}$, we define

$$g_{[m,\lambda]}(x) := g(m+\lambda x). \tag{4}$$

the polynomial obtained from shifting g by m followed by the scaling $x \mapsto \lambda \cdot x$.

2.2. Approximate Taylor Shift

For a μ-approximation $\tilde{g} \in [g]_\mu$ and $|\lambda| \le 1$, it is obvious that $\tilde{g}_{[0,\lambda]}$ is a μ-approximation of $g_{[0,\lambda]}$ as well. The following considerations will show that this result generalizes to arbitrary m and λ of magnitude less than or equal to $1/2$.

Lemma 1. *Let $m \in \mathbb{C}$, $\lambda \in \mathbb{R}\backslash\{0\}$ with $|m|, |\lambda| \le 1/2$, and $\tilde{g} \in [g]_\mu$ a μ-approximation of g. Then, $\tilde{g}_{[m,\lambda]}$ is a 2μ-approximation of $g_{[m,\lambda]}$, that is, $\tilde{g}_{[m,\lambda]} \in [g_{[m,\lambda]}]_{2\mu}$.*

Proof. For $h(x) := (g-\tilde{g})(x) = \mu_{n-1}x^{n-1} + \ldots + \mu_1 x + \mu_0$, the absolute value of each coefficient μ_i is bounded by μ. The following computation shows that all coefficients of $h_{[m,\lambda]}(x)$ have absolute value less than 2μ which proves our claim:

$$h(m+\lambda x) = \sum_{i=0}^{n-1} \mu_i (m+\lambda x)^i = \sum_{i=0}^{n-1} \mu_i \sum_{k=0,\ldots,i} x^k \lambda^k m^{i-k} \binom{i}{k} = \sum_{k=0}^{n-1} x^k \sum_{i=k}^{n-1} \mu_i m^{i-k} \lambda^k \binom{i}{k}$$

Thus, the absolute value of the coefficient of x^k is bounded by

$$\mu \cdot \sum_{i \ge k} \frac{1}{2^i} \binom{i}{k} = \frac{\mu}{2^k} \cdot \sum_{i \ge 0} 2^{-i} \binom{k+i}{k} = \frac{\mu}{2^k} \cdot \frac{1}{(1-1/2)^{k+1}} = 2\mu,$$

where we used

$$\left(1-\frac{1}{2}\right)^{-(k+1)} = \sum_{i \ge 0} \binom{-(k+1)}{i} (-1)^i 2^{-i} = \sum_{i \ge 0} \binom{k+i}{i} 2^{-i} = \sum_{i \ge 0} \binom{k+i}{k} 2^{-i}.$$

□

2.3. Root Perturbation Bounds

As already mentioned in the introduction, we want to isolate the roots of f by isolating the roots of a carefully chosen approximation $\tilde{f} \in [f]_\mu$ first and then enlarging the so obtained isolating regions to isolating regions for the roots of f. Here, carefully chosen means that μ should be sufficiently small such that each root z of f does not move too far compared to its separation $\sigma(z,f)$ when passing from f to $\tilde{f}$. Our idea to check for sufficient approximation is to apply perturbation bounds for the roots of f and $\tilde{f}$. In [23], the following worst case perturbation bound from Schönhage [31] has been used: *If all roots of f are contained in the unit disc and $\left\|\tilde{f}-f\right\|_1 < \mu \|f\|_1$ for a $\mu < 2^{-7n}$, then corresponding roots of f and $\tilde{f}$ differ by at most $9\sqrt[n]{\mu}$.* Similar as the method in [23], the randomized algorithm in [12] uses a precision management that is also exclusively controlled by a guess on the separation of f. Instead of considering Schönhage's bound it uses a worst case perturbation bound derived from Smith's bound [34] which is asymptotically equivalent to the one from Schönhage. The usage of worst case bounds forces the above algorithms to consider approximations $\tilde{f} \in [f]_\mu$ with $\mu < \sigma(f)^n$ in order to ensure that the roots stay at almost the same place when passing from f to $\tilde{f}$. However, it can be shown that the latter property can already be achieved if μ is smaller than the product of all separations. For many polynomials, not all roots realize the minimum distance to a nearest root but only some do. For these polynomials, the geometric mean of the root separations will be significantly larger than the minimal distance (see Lemma 2 and the subsequent examples). The algorithms described in [12, 23] are ignorant of such a situation. In his PhD Thesis [10], Eigenwillig presented a variant of [12] which mostly decouples the precision management from a guess on the separation and, thus, succeeds for considerably less precision. For this approach, randomization is essential. We aim to design a deterministic method that is based on a precision management which takes the separations of all roots into account. In order to do so, we introduce a root perturbation bound which is similar to Smith's bound but suits better our needs.

Throughout the following considerations, $g(x) = \sum_{i=0}^{n} g_i x^i \in \mathbb{R}[x]$ denotes an arbitrary square-free polynomial of degree $n \geq 2$ with roots $\alpha_1, \ldots, \alpha_n$. We will later apply our results mostly to $g = f$ but also to some transformations of f.

Definition 1. *For $t \geq 1$ an arbitrary real value, we define*

$$\mu(g,t) := \frac{1}{t} \cdot \min_{i=1,\ldots,n} \left| \frac{\sigma(\alpha_i, g) g'(\alpha_i)}{4n^2} \right|. \tag{5}$$

We call a $\mu \in \mathbb{R}_0^+$ sufficiently small with respect to g *if $\mu \leq \mu(g) := \mu(g, 2^7 n^2)$. Furthermore, each $\tilde{g} \in [g]_{\mu(g)}$ is called* a sufficiently good approximation of g.

We remark that, for arbitrary values $m \in \mathbb{C}$ and $\lambda \in \mathbb{R}\setminus\{0\}$, $\mu(g,t) = \mu(g_{[m,\lambda]}, t)$ due to the following argument: The roots of $g_{[m,\lambda]}(x) = g(m + \lambda x)$ are given by $\alpha_i^* := \frac{\alpha_i - m}{\lambda}$, thus, their separations scale by a factor λ^{-1}. Since $(g_{[m,\lambda]})'(\alpha_i^*) = \lambda g'(\alpha_i)$, it follows that $\mu(g_{[m,\lambda]}, t) = \mu(g,t)$. In particular, $\mu(f) = \mu(F)$. We further remark that, for a polynomial

f as in (2), we have $\log(\mu(f)^{-1}) = O(\log n + \Sigma(F))$ because

$$\sigma(z_i,f)\cdot|f'(z_i)| = \sigma(z_i,f)\cdot|a_n|\prod_{j\neq i}|z_i-z_j| \geq \sigma(z_i,f)\cdot|a_n|\prod_{j\neq i}\sigma(z_j,f) = |a_n|2^{-\Sigma(f)}$$

and $\Sigma(f)-\log|a_n| = \Sigma(F)+n\log(8\Gamma)-\log(8\Gamma)^n = \Sigma(F)$.

Lemma 2. *For fixed $t \geq 1$, suppose that each disc $\Delta_i := \Delta_{\frac{\sigma(\alpha_i,g)}{tn}}(\alpha_i)$, $i = 1,\ldots,n$, is contained within the unit disc $\Delta_1(0)$. Then, for each $\tilde{g} \in [g]_\mu$ with $\mu \leq \mu(g,t)$, it holds that:*

(i) *Each root $\tilde{\alpha}_i$ of $\tilde{g}$ differs by less than $\frac{\sigma(\alpha_i,g)}{tn}$ from a corresponding counterpart α_i.*
(ii) *For each $i = 1,\ldots,n$, we have*

$$\left(1-\frac{2}{tn}\right)\sigma(\alpha_i,g) < \sigma(\tilde{\alpha}_i,\tilde{g}) < \left(1+\frac{2}{tn}\right)\sigma(\alpha_i,g).$$

(iii) *If μ is sufficiently small with respect to g, then $(1-2^{-9})\mu$ is sufficiently small with respect to any $\tilde{g} \in [g]_\mu$.*

Proof. For an arbitrary point $z \in \partial\Delta_i$ on the boundary of Δ_i, we have

$$\begin{aligned}
|g(z)| &= |g_n|\prod_{j=1}^{n}|z-\alpha_j| = \frac{\sigma(\alpha_i,g)}{tn}\left(\prod_{1\leq j\leq n, j\neq i}\left|\frac{z-\alpha_j}{\alpha_i-\alpha_j}\right|\right)\cdot|g_n|\left(\prod_{1\leq j\leq n, j\neq i}|\alpha_i-\alpha_j|\right)\\
&= \frac{\sigma(\alpha_i,g)|g'(\alpha_i)|}{tn}\prod_{1\leq j\leq n, j\neq i}\left|\frac{z-\alpha_j}{\alpha_i-\alpha_j}\right| \geq \frac{\sigma(\alpha_i,g)|g'(\alpha_i)|}{tn}\prod_{1\leq j\leq n, j\neq i}\frac{|\alpha_i-\alpha_j|-|z-\alpha_i|}{|\alpha_i-\alpha_j|}\\
&\geq \frac{\sigma(\alpha_i,g)|g'(\alpha_i)|}{tn}\left(1-\frac{1}{tn}\right)^{n-1} > \frac{\sigma(\alpha_i,g)|g'(\alpha_i)|}{3tn} > n\mu(g,t).
\end{aligned}$$

In addition, since $\tilde{g} \in [g]_\mu$, it follows that $|(g-\tilde{g})(z)| < n\mu$ because $|z| < 1$. Now, using Rouché's Theorem for the discs Δ_i and the functions g and $\tilde{g}$, we obtain: If $\mu \leq \mu(g,t)$, then $|(g-\tilde{g})(z)| < |g(z)|$ for all $z \in \partial\Delta_i$ and, thus, g and $\tilde{g}$ have the same number of roots, namely one, within Δ_i. Since g and $\tilde{g}$ have the same degree, both have the same number of roots which proves (i). The estimate (ii) on the separation is an immediate consequence from our above consideration because, under the assumption $\mu \leq \mu(g,t)$, we must have

$$\left(1-\frac{2}{tn}\right)\sigma(\alpha_i,g) < \sigma(\tilde{\alpha}_i,\tilde{g}) < \left(1+\frac{2}{tn}\right)\sigma(\alpha_i,g)$$

for each root $\tilde{\alpha}_i$. Now, because of

$$\frac{\tilde{g}'(\tilde{\alpha}_i)}{g'(\alpha_i)} = \frac{\tilde{g}_n}{g_n}\prod_{j\neq i}\frac{\tilde{\alpha}_i-\tilde{\alpha}_j}{\alpha_i-\alpha_j} = \prod_{j\neq i}\frac{\tilde{\alpha}_i-\tilde{\alpha}_j}{\alpha_i-\alpha_j} = \prod_{j\neq i}\left(1+\frac{\tilde{\alpha}_i-\alpha_i}{\alpha_i-\alpha_j}-\frac{\tilde{\alpha}_j-\alpha_j}{\alpha_i-\alpha_j}\right)$$

and $\left|\frac{\tilde{\alpha}_j-\alpha_i}{\alpha_i-\alpha_j}\right| \leq \frac{1}{tn}$ for all $i \neq j$, it follows that

$$\left(1-\frac{2}{tn}\right)^n|g'(\alpha_i)|\sigma(\alpha_i,g) < |\tilde{g}'(\tilde{\alpha}_i)|\sigma(\tilde{\alpha}_i,\tilde{g}) < \left(1+\frac{2}{tn}\right)^n|g'(\alpha_i)|\sigma(\alpha_i,g).$$

For $t \geq 2^7 n^2 \geq 2^9$, the left side of the latter inequality implies that $(1-2^{-9})\mu(g,t) < \mu(\tilde{g},t)$ since $(1-\frac{2}{tn})^n \geq (1-\frac{1}{t})^1 \geq (1-2^{-9})$. It follows that $(1-2^{-9})\mu$ is sufficiently small with respect to any $\tilde{g} \in [g]_\mu$ if μ is sufficiently small with respect to g. □

From the last theorem and our remarks subsequent to Definition 1, it follows that, for a given polynomial f as in (2), any approximation $\tilde{f}$ of f to $\rho = \lceil \log(\mu(f)^{-1}) \rceil = O(\Sigma(F) + \log n)$ bits after the binary point has its roots at almost the same location as f (with respect to the corresponding separations).

Corollary 3. *Let f be a polynomial as in (2) and $\tilde{f} \in [f]_{\mu(f)}$ a sufficiently good approximation of f. Then, each root z_i of f moves by at most $\sigma(z_i,f)/(2^7 n^3)$ when passing from f to $\tilde{f}$. In particular, real roots of f stay real and non-real roots stay non-real.*

Remark: The reader may notice that, using the worst case perturbation bound from Schönhage, we would need $\mu < (\sigma(f)/(9tn))^n \cdot \|f\|_1$ to ensure that z_i and $\tilde{z}_i$ do not differ by more than $\sigma(z_i,f)/(tn)$ for all i. Thus, we have to approximate the coefficients of f to $O(n(\log n - \log \sigma(f)))$ bits after the binary point. In Section 3, we will see that our algorithm needs approximations of f to only $\rho = \lceil \log(\mu(f)^{-1}) \rceil = O(\Sigma(f) + \log n)$ bits after the binary point. The following two examples demonstrate how these approximation demands compare to each other.

Examples. a) The polynomial $f := x^{100} - 1/2$ has exactly 100 distinct roots $z_k := \sqrt[100]{1/2} e^{I\frac{2k\pi}{100}}$, $k = 1,\ldots,100$, on the boundary of the disc $\Delta_{\sqrt[100]{1/2}}(0)$. Since $\sigma(f) = \sigma(z_k,f) \approx 0.062$ for all k, it follows that $\Delta_{\sigma(z_k,f)/100}(z_k) \subset \Delta_1(0)$. Furthermore, we have $|f'(z_k)| = 99\sqrt[100]{(1/2)^{99}} \approx 49.844$ and, thus, due to Lemma 2, we need $\mu < 0.000077$ to ensure that corresponding roots z_k and $\tilde{z}_k$ of f and $\tilde{f} \in [f]_\mu$ do not differ by more than $\sigma(z_k,f)/100$. However, using the worst case perturbation bound from Schönhage, we need $\mu < (\sigma(f)/900)^{100} \|f\|_1 \approx 9.795 \cdot 10^{-417}$. This shows that, even in case where all roots have minimal separation, the approximation demand can be of magnitudes smaller.

b) Let $f := 1048576x^{20} - 2(200x-1)^2$ be a Mignotte-like polynomial. We state without proof that $\Delta_{\sigma(z_i,f)/20}(z_i) \subset \Delta_1(0)$ for all roots z_i of f. Furthermore, f has two nearby roots z_1 and z_2 close to 0.005. Their separations are given by $\sigma(f) = \sigma(z_1,f) = \sigma(z_2,f) \approx 7.071 \cdot 10^{-23}$. For the derivative of f at z_1 and z_2, we have $|f'(z_1)|, |f'(z_2)| \approx 5.657 \cdot 10^{-18}$, thus, we need $\mu < 2.5 \cdot 10^{-42}$ to ensure that each z_i does not move by more than $\sigma(z_i,f)/20$ when passing from f to $\tilde{f} \in [f]_\mu$. Again, using the worst case bound from Schönhage, we need $\mu < (\sigma(f)/180)^{20} \|f\|_\infty \approx 8.647 \cdot 10^{-483}$, a significantly higher approximation of the coefficients of f.

2.4. The $\mathscr{T}^{\mathbf{g}}_{\mathbf{K}}(\mathbf{m},\mathbf{r})$-Test: Existence of Roots

For $m \in \mathbb{C}$ and positive real values K and r, we consider the test

$$\mathscr{T}^g_K(m,r): \quad t^g_K(m,r) := |g(m)| - K \sum_{k \geq 1} \left| \frac{g^{(k)}(m)}{k!} \right| r^k > 0. \tag{6}$$

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 $\Delta = \Delta_r(m)$ or $I = (a,b)$ an interval with midpoint m and radius r. 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 $\mathscr{T}_K^g(m,r)$ and $\mathscr{T}_K'(m,r)$ for $\mathscr{T}_K^{g'}(m,r)$. We often use $K = 3/2$. Therefore, whenever the "K" is suppressed (i.e., we write $\mathscr{T}^g(m,r)$ instead of $\mathscr{T}_{3/2}^g(m,r)$), we consider $K = 3/2$. Before presenting the main technical lemmata, we first summarize the following useful properties of $\mathscr{T}_K^g(m,r)$:

- If $\mathscr{T}_K^g(m,r)$ holds, then $\mathscr{T}_{K'}^g(m,r)$ holds for all $K' \le K$ and all $r' \le r$.
- For arbitrary values m, r and $\lambda \neq 0$, the test $\mathscr{T}_K^g(m,r)$ is equivalent to $\mathscr{T}_K^{g_{[m,\lambda]}}(0, r/\lambda)$ because of $t_K^{g_{[m,\lambda]}}(0, r/\lambda) = t_K^g(m,r)$.
- For $\lambda \in \mathbb{R}^+$, $t_K^g(m,r) = t_K^{\lambda g}(m,r)/\lambda$ and, thus, $\mathscr{T}_K^g(m,r)$ is equivalent to $\mathscr{T}_K^{\lambda g}(m,r)$.

The above test serves as an exclusion predicate but might also give a guarantee that a certain disc contains at most one root. We refer to [4, Theorem 3.2] for a proof of the following lemma.

Lemma 4. *Consider a disk $\Delta = \Delta_m(r) \subset \mathbb{C}$:*

(i) *If $\mathscr{T}_1(\Delta)$ holds, then Δ contains no root of g and*

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

for all $z \in \bar{\Delta}$ in the closure of Δ.

(ii) *If $\mathscr{T}_{3/2}'(\Delta)$ hold, then $\bar{\Delta}$ contains at most one root of g.*

We can also give a lower bound on the radius r of Δ in terms of the separation $\sigma(g)$ of g such that either $\mathscr{T}(\Delta)$ or $\mathscr{T}'(\Delta)$ holds.

Lemma 5. *Let $\Delta = \Delta_r(m) \subset \mathbb{C}$ be a disc of radius r centered at $m \in \mathbb{C}$.*

(i) *If $r < \sigma(g)/(4n^2)$, then $\mathscr{T}(\Delta)$ or $\mathscr{T}'(\Delta)$ holds.*

(ii) *If Δ contains a root α_i of g and $r < \sigma(\alpha_i, g)/(4n^2)$, then $\mathscr{T}'(\Delta)$ holds.*

(iii) *If neither $\mathscr{T}(\Delta)$ nor $\mathscr{T}'(\Delta)$ holds, then $\Delta_{6n^2 r}(m)$ contains at least two roots α_1 and α_2 of g with $|\alpha_1 - \alpha_2| < 4n^2 r$.*

Proof. For the proof of (i) and (ii), we use a result from [9, 37] which shows that, for each root α_i of g, the disc $\Delta_{\sigma(\alpha_i,g)/n}(\alpha_i)$ does not contain any of the roots $\alpha_1', \ldots, \alpha_{n-1}'$ of the derivative g'. Thus, an arbitrary point $m \in \mathbb{C}$ is at least $\sigma(g)/(2n)$ away from all α_i or from all z_i'. We first consider the case where $|m - \alpha_i| \ge \sigma(g)/(2n)$ for all i. In this situation,

$$\left|\frac{g^{(k)}(m)}{g(m)}\right| = \left|\sum\nolimits'_{i_1,\ldots,i_k} \frac{1}{(m-\alpha_{i_1})\ldots(m-\alpha_{i_k})}\right| \le \left(\sum\nolimits_{i=1,\ldots,n} \frac{1}{|m-\alpha_i|}\right)^k \le \left(\frac{2n^2}{\sigma(g)}\right)^k,$$

where the prime means that the i_j's ($j = 1 \ldots k$) are chosen to be distinct. For a disc Δ of radius $r < \sigma(g)/(4n^2)$ and midpoint m, it follows that $\sum_{k\ge1} \left|\frac{g^{(k)}(m)}{g(m)}\right| \frac{r^k}{k!} < \sum_{k\ge1} \frac{1}{k!} \left(\frac{2n^2 r}{\sigma(g)}\right)^k < e^{\frac{1}{2}} - 1 < \frac{2}{3}$. Thus, $\mathscr{T}(\Delta)$ holds in this case. If $|m - \alpha_i'| \ge \sigma(g)/(2n)$ for all i, then a similar consideration shows that $\sum_{k\ge2} \left|\frac{g^{(k)}(m)}{g'(m)}\right| \frac{r^{k-1}}{(k-1)!} < \sum_{k\ge2} \frac{1}{(k-1)!} \left(\frac{2n(n-1)r}{\sigma(g)}\right)^{(k-1)} < \frac{2}{3}$,

hence, $\mathscr{T}'(\Delta)$ holds. (ii) follows in analogous manner because, in this case, $|m-\alpha'_j| \geq \sigma(\alpha_i,g)/(2n) > 2nr$ for all j. For (iii), suppose that neither $\mathscr{T}(\Delta)$ nor $\mathscr{T}'(\Delta)$ holds. Our proof of (i) and (ii) shows that there must exist at least one root α of g in the disc $\Delta_{2nr}(m)$. Otherwise, m would be separated from any root of g by at least $2nr$ and, thus, $\left|\frac{g^{(k)}(m)}{g(m)}\right| < \left(\frac{1}{2r}\right)^k$ for all k. This would imply the success of $\mathscr{T}(\Delta)$. In complete analogous manner, it follows that $\Delta_{2nr}(m)$ also contains a root α' of g'. If we assume that α is the only root of g within $\Delta_{6n^2r}(m)$, then $\sigma(\alpha,g) > 6n^2r - 2nr \geq 4n^2r$ and, thus, $|\alpha-\alpha'| \geq 4nr$, contradicting the fact that α and α' are both contained in $\Delta_{2nr}(m)$. It follows that $\Delta_{6n^2r}(m)$ contains two roots α_1, α_2 of g with distance less than $4n^2r$ from each other. □

2.5. Testing for Sufficient Precision

Most subdivision algorithms for isolating the roots of a polynomial g do not create regions which are much smaller than the separation $\sigma(g)$ of g. For our algorithm to isolate the roots of f, we aim for a similar behavior. In particular, we do not want to create regions which are much smaller than $\sigma(f)$. However, since we do not run the subdivision algorithm on f directly but on a certain μ-approximation $\tilde{f}$ of f, this is non-trivial at all because $\sigma(\tilde{f})$ may be much smaller than $\sigma(f)$. Certainly, if μ has been chosen sufficiently small (see Definition 1), then $\sigma(\tilde{f}) \approx \sigma(f)$, however, $\mu(f)$ is initially unknown. In order to prevent our algorithm to subdivide too small regions for values $\mu > \mu(f)$, we introduce a guard which prevents us from subdividing very small regions by informing us that μ is not small enough. We can then restart our algorithm with an improved approximation of f.

Let $g = \sum_{i=0}^n g_i x^i$ be a polynomial, $\Delta := \Delta_r(m)$ be a disc in $\mathbb{C}$ and μ an arbitrary non-negative number. We aim to check whether there exists a $g^* \in [g_{[m,1/2]}]_{2\mu} = [g(m+x/2)]_{2\mu}$ for which $\mathscr{T}^{g^*}(0,2r)$ holds. In order to do so, we consider a polynomial in $[g_{[m,1/2]}]_{2\mu}$ for which $t_{3/2}(0,2)$ is maximal; see (6) for the definition of $t_K(m,r)$. For $k = 1,\ldots,n-1$, we define

$$g_k^{[\mu,m]} := \frac{g^{(k)}(m)}{2^k k!} - \operatorname{sgn}(g^{(k)}(m)) \cdot \min\left(\frac{|g^{(k)}(m)|}{2^k k!}, 2\mu\right),$$

and

$$g^*(x) := g^{[\mu,m]}(x) := (g(m) + 2\operatorname{sgn}(g(m)) \cdot \mu) + \sum_{k=1}^{n-1} g_k^{[\mu,m]} x^k + \frac{g^{(n)}(m)}{2^n n!} x^n \qquad (7)$$

Hence, the polynomial g^* is a 2μ-approximation of $g_{[m,1/2]}(x) = \sum_{k\geq 0} \frac{g^{(k)}(m)}{2^k k!} x^k$ for which $t_{3/2}(0,2)$ becomes maximal. In order words, we obtain g^* by increasing or decreasing the coefficients of $g_{[m,1/2]} = g(m+x/2)$ by at most 2μ such that the absolute value of the constant coefficient becomes maximal and that of all other coefficients minimal.

Example. For $g = x^3 - x^2 + 2x - 1$, we have $g_{[1/2,1/2]}(x) = \frac{1}{8}x^3 + \frac{1}{8}x^2 + \frac{7}{8}x - \frac{1}{8}$ and, thus,

$$g^{[1/4,1/2]}(x) = \frac{3}{8}x^3 + \left(\frac{1}{8} - \frac{1}{8}\right)x^2 + \left(\frac{7}{8} - 2\cdot\frac{1}{4}\right)x + \left(-\frac{1}{8} - 2\cdot\frac{1}{4}\right) = \frac{3}{8}x^3 + \frac{3}{8}x - \frac{5}{8}.$$

We now extend the test $\mathscr{T}^g(m,r)$ from (6) to the family $[g]_\mu$ of all μ-approximations of g, where g^* is the polynomial as defined in (7):

$$\mathscr{T}^{[g]_\mu}(m,r): \quad t^{g^*}(0,2r) > 0 \tag{8}$$

which is equivalent to $\mathscr{T}^{g^*}(0,2r)$. We remark that $\mathscr{T}^{[g]_\mu}(m,r)$ is equivalent to $\mathscr{T}^g(m,r)$ for $\mu = 0$, justifying the term "extension of $\mathscr{T}^g(m,r)$". We immediately obtain the following useful property:

Lemma 6. *$\mathscr{T}^{[g]_\mu}(m,r)$ succeeds if and only if there exists an approximation $g^* \in [g_{[m,1/2]}]_{2\mu}$ such that $\mathscr{T}^{g^*}(0,2r)$ holds.*

The next lemma is central for the preceding Theorem 8 where we present our guard to check whether a certain μ was chosen sufficiently small with respect to f.

Lemma 7. *Suppose that $\mathscr{T}^{[g]_\mu}(m,r)$ or $\mathscr{T}^{[g']_{n\mu}}(m,r)$ succeeds. Then, there exists an $n\mu$-approximation g^* of $g_{[m,1/2]}$ that has at most one root within $\Delta_{2r}(0)$.*

Proof. We distinguish two cases:

- $\mathscr{T}^{[g]_\mu}(m,r)$ holds: Due to Lemma 6 there exists a 2μ-approximation g^* of $g_{[m,1/2]}$ for which $\mathscr{T}^{g^*}(0,2r)$ holds, thus, $\Delta_{2r}(0)$ contains no root of g^*.
- $\mathscr{T}^{[g']_\mu}(m,r)$ holds: In this case, there exists a $2n\mu$-approximation $\bar{g}(x)$ of $g'(m+x/2) = 2\cdot(g_{[m,1/2]}(x))'$ for which $\mathscr{T}^{\bar{g}}(0,2r)$ holds. Since $\bar{g}/2$ is an $n\mu$-approximation of $(g_{[m,1/2]}(x))'$, integrating $\bar{g}/2$ leads to an $n\mu$-approximation g^* of $g_{[m,1/2]}$ such that $\mathscr{T}^{(g^*)'}(0,2r)$ holds. Then, according to Lemma 4, the disc $\Delta_{2r}(0)$ contains at most one root of g^*.

□

This motivates the following definition:

Definition 2. *$\Delta = \Delta_r(m)$ is called* terminal for $[g]_\mu$ *if $\mathscr{T}^{[g]_\mu}(m,r)$ or $\mathscr{T}^{[g']_{n\mu}}(m,r)$ holds.*

Theorem 8 ("The Guard"). *Let f be a polynomial as in (2), $\tilde{f} \in [f]_\mu$ and $|m| \le 1/4$.*

(i) *Any disc $\Delta := \Delta_r(m)$ with radius $r < \sigma(f)/(4n^2)$ is terminal for $[\tilde{f}]_\mu$.*
(ii) *If Δ contains a root z_i of f and $r < \sigma(z_i,f)/(4n^2)$, then Δ is terminal for $[\tilde{f}]_\mu$.*
(iii) *If $\Delta_{6n^2r}(m)$ contains no root z of f with $\sigma(z,f) < 4n^2r$, then Δ is terminal for $[\tilde{f}]_\mu$.*
(iv) *If $\mathscr{T}^{\tilde{f}}(\Delta)$ and $\mathscr{T}^{\tilde{f}'}(\Delta)$ do not hold and $\Delta_{8n^2r}(m)$ is terminal for $[\tilde{f}]_\mu$, then $\mu \ge \mu(f)$.*

Proof. Due to Lemma 5, (i) $\mathscr{T}^f(\Delta)$ or $\mathscr{T}^{f'}(\Delta)$ holds for $r < \sigma(f)/(4n^2)$. Thus, at least one of the tests, $\mathscr{T}^{f_{[m,1/2]}}(0,2r)$ or $\mathscr{T}^{f'_{[m,1/2]}}(0,2r)$, succeeds. Since $\tilde{f}_{[m,1/2]}$ and $\tilde{f}'_{[m,1/2]}$ are 2μ- and $2n\mu$-approximations of $f_{[m,1/2]}$ and $f'_{[m,1/2]}$, respectively, Lemma 6 implies that Δ is terminal for $[\tilde{f}]_\mu$. (ii) follows in complete analogous manner by considering Lemma 5 (ii). (iii) is a direct implication of Lemma 5 (iii): Under the given conditions, either $\mathscr{T}^f(\Delta)$ or $\mathscr{T}^{f'}(\Delta)$ holds, thus, Δ must be terminal for $[\tilde{f}]_\mu$. For the proof of (iv), we assume that $\mu \le \mu(f)$, that is, $\mu \le \mu(f,2^7n^2) = \min_i \left|\frac{\sigma(z_i,f)f'(z_i)}{2^9n^4}\right|$. The roots of $g := f_{[m,1/2]}(x) = f(m+x/2)$ are given by $\xi_i = 2(z_i - m)$ and, thus, all ξ_i are contained within $\Delta_{3/4}(0)$ and

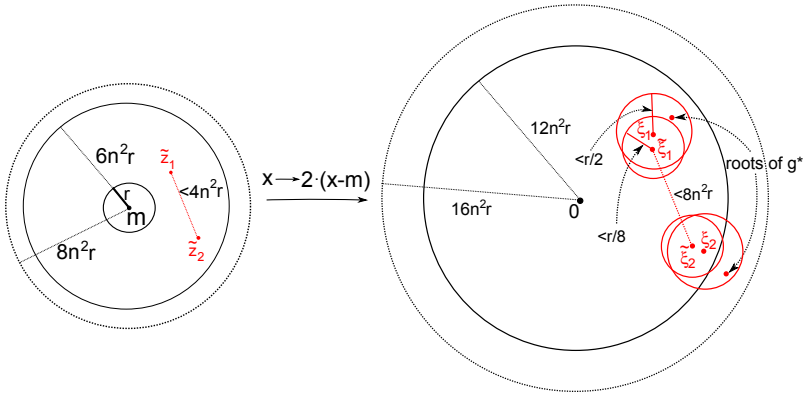

FIGURE 1. The left figure shows the situation for $\tilde{f}$, the figure on the right shows the situation for the transformed polynomial $\tilde{f}_{(m,1/2)}$ with corresponding roots $\tilde{\xi}_i := 2(\tilde{z}_i - m)$.

$\sigma(\xi_i, g) = 2\sigma(z_i, f) < 1/2$ for all i. Hence, the conditions from Lemma 2 are fulfilled for g. We have already shown that $\mu(g) = \mu(f_{[m,1/2]}) = \mu(f)$ and, thus, μ is also sufficiently small with respect to g. Now, if neither $\mathscr{T}^{\tilde{f}}(\Delta)$ nor $\mathscr{T}^{\tilde{f}'}(\Delta)$ holds then, according to Lemma 5, the disc $\Delta_{6n^2r}(m)$ contains two roots $\tilde{z}_1$ and $\tilde{z}_2$ of $\tilde{f}$ with $|\tilde{z}_1 - \tilde{z}_2| < 4n^2r$; see Figure 1. It follows that the polynomial $\tilde{g} := \tilde{f}_{[m,1/2]}$ has two roots $\tilde{\xi}_i := 2(\tilde{z}_i - m)$, $i = 1, 2$, with separation $\sigma(\tilde{\xi}_i, g) < 8n^2r$ and both of them are contained in $\Delta_{12n^2r}(0)$. According to Lemma 1, $\tilde{g}$ is a 2μ-approximation of g. Thus, Lemma 2 (applied to g) ensures the existence of two roots ξ_1 and ξ_2 of g with

$$\left(1 - \frac{1}{2^6 n^3}\right) \sigma(\xi_i, g) < \sigma(\tilde{\xi}_i, \tilde{g}) < 8n^2r, \text{ thus, } \sigma(\xi_i, g) < 9n^2r,$$

and $|\xi_i - \tilde{\xi}_i| < \sigma(\xi_i, g) 2^{-7} n^{-3} < 9n^{-1}2^{-7}r < \frac{r}{8}$. Now, let us consider an arbitrary $n\mu$-approximation g^* of $\tilde{g}$. Then, g^* is an $(n+2)\mu$-approximation of g since $\tilde{g} \in [g]_{2\mu}$. When passing from g to g^*, the roots ξ_i do not move by more than $(n+2)\sigma(\xi_i, g)2^{-7}n^{-3} < \frac{r}{2}$. Thus, each disc $\Delta_{5r/8}(\tilde{\xi}_i)$ contains at least one root of g^* that corresponds to ξ_i. It follows that, for any $n\mu$-approximation g^* of g, the disc $\Delta_{16n^2r}(0)$ contains at least two roots of g^*. Hence, $\Delta_{8n^2r}(m)$ cannot be terminal for $|\tilde{f}|_\mu$ due to Lemma 7. □

2.6. Estimating Separation and Derivative

Given a polynomial $g(x) = g_n x^n + \ldots + g_0$ and a disc $\Delta := \Delta_r(m)$ which is isolating for a root ξ of g, we aim to estimate the values $\sigma(\xi, g)$ and $|g'(\xi)|$. We will also give a lower bound on the value of $|g|$ on the boundary of Δ.

Lemma 9. *With the notations from above, suppose that Δ contains a root ξ of g and*

$$R := \sup\{t \in \mathbb{R}^+ : \mathscr{T}^{g'}(m,t) \text{ holds}\} \geq 8nr, \text{ then}$$

(i) *ξ is the unique root of g within $\Delta_R(m)$,*

(ii)

$$\frac{1}{4n^2}\sigma(\xi,g) \leq R \leq \frac{16}{15}\sigma(\xi), \text{ and}$$

(iii) *for an arbitrary $\tilde{r} \in \mathbb{R}$ with $\frac{R}{4n} \leq \tilde{r} < \frac{R}{2n}$ and an arbitrary point z on the boundary of $\Delta_{\tilde{r}}(m)$, the following inequality holds*

$$|g(z)| > \frac{\tilde{r}|g'(m)|}{8} > \frac{\sigma(\xi,g)|g'(\xi)|}{135n^3}.$$

Proof. Due to Lemma 4, ξ is the unique root of g within $\Delta_R(m)$. We now estimate $|g'(\xi)|$. Since $\mathscr{T}^{g'}(m,8nr)$ holds, we have

$$|g'(m)| > \frac{3}{2}\sum_{k\geq 1}\frac{|g^{(k+1)}(m)|}{k!}(8n)^k r^k > \frac{24n}{2}\sum_{k\geq 1}\frac{|g^{(k+1)}(m)|}{k!}r^k$$

and, thus, $\mathscr{T}_{24}^{g'}(m,r)$ holds as well. It follows that $\frac{23}{24}|g'(m)| < |g'(\xi)| < \frac{25}{24}|g'(m)|$. Since $\Delta_R(m)$ contains no other root than ξ, we have $\sigma(\xi,g) > R - r \geq R - \frac{R}{8n} = R(1-\frac{1}{8n}) \geq \frac{15}{16}R$ which proves the right inequality of (ii). For the left inequality, we assume that $R < \frac{\sigma(\xi,g)}{4n^2}$. Then, for any $\tilde{R}$ with $R < \tilde{R} < \frac{\sigma(\xi,g)}{4n^2}$, Lemma 5 states that $\mathscr{T}^{g'}(m,\tilde{R})$ holds, a contradiction to our definition of R. The proof of (iii) consists of two steps. First, we prove that each point z on the boundary of $\Delta_{\tilde{r}}(m)$ fulfills the inequality

$$\frac{\sigma(\xi,g)}{2^5 n^3} \leq \frac{R}{8n} < |z-\xi| < \frac{\sigma(\xi,g)}{n} \tag{9}$$

We denote $d := |\xi - m| < r$ the distance between m and ξ. We have $\sigma(\xi,g) \geq R - d$ and $|z-\xi| \leq |z-m| + |\xi - m| = \tilde{r} + d$. In order to prove the right inequality in 9 it suffices to show that $R - d > n(\tilde{r}+d)$. This is equivalent to $R - n\tilde{r} > (n+1)d$ which is true since $R - n\tilde{r} > \frac{R}{2}$ and $2(n+1)d < 8nd < 8nr \leq R$. For the left inequality in 9, we consider the following computation

$$|z-\xi| \geq |z-m| - |m-\xi| = \tilde{r} - d > \tilde{r} - r \geq \tilde{r} - \frac{R}{8n} \geq \tilde{r} - \frac{\tilde{r}}{2} = \frac{\tilde{r}}{2},$$

thus, we get $|z-\xi| > \frac{\tilde{r}}{2} \geq \frac{R}{8n} \geq \frac{\sigma(\xi,g)}{32n^3}$. We denote the roots of g by $\xi_1 := \xi, \xi_2, \ldots, \xi_n$ and consider the following computation which is similar to the one in the proof of Lemma 2:

$$\begin{aligned}
|g(z)| &= |g_n|\prod_{i=1}^{n}|z-\xi_i| = |z-\xi|\cdot|g_n|\prod_{i=2}^{n}|\xi-\xi_i|\cdot\prod_{i=2}^{n}\frac{|z-\xi_i|}{|\xi-\xi_i|} \\
&> \frac{\tilde{r}}{2}\cdot|g'(\xi)|\cdot\prod_{i=2}^{n}\left(1-\frac{|z-\xi|}{|\xi-\xi_i|}\right) > \frac{\tilde{r}}{2}\cdot|g'(\xi)|\cdot\prod_{i=2}^{n}\left(1-\frac{\sigma(\xi,g)}{n\sigma(\xi,g)}\right) \\
&= \frac{\tilde{r}}{2}\cdot|g'(\xi)|\cdot(1-\frac{1}{n})^{n-1} > \frac{\tilde{r}}{2}\cdot|g'(\xi)|\cdot\frac{1}{3} > \frac{\tilde{r}|g'(m)|}{8} > \frac{\sigma(\xi,g)|g'(\xi)|}{135n^3}.
\end{aligned}$$

The above computation uses $|z-\xi| < \sigma(\xi)/n$ and $\frac{23}{24}|g'(m)| < |g'(\xi)| < \frac{25}{24}|g'(m)|$. □

In order to apply Lemma 9 to $g := \tilde{f}$ we first compute a rational approximation $\tilde{R}$ of R with $R \le \tilde{R} < 2R$. For a given rational r, we can perform binary search to find such an $\tilde{R} = 8n \cdot 2^\tau r$, $\tau \in \mathbb{N}_0$, in $O(\log|\log R| + \log\log n) = O(\log|\log \sigma(\xi)| + \log\log n)$ steps. Then, $\tilde{r} := 2^{\tau+1} r$ fulfills the condition in Lemma 9 (iii).

Theorem 10. *We use the same notations as in Lemma 9 with $g := \tilde{f} \in [f]_\mu$. Suppose that $\Delta = \Delta_r(m) \subset \mathbb{C}$, $|m| \le 1/4$, contains a root $\tilde{z}_i$ of $\tilde{f}$, and $\mathscr{T}'(m, 8nr)$ holds. With $\tilde{R} = 8n \cdot 2^\tau r$, $\tau \in \mathbb{N}_0$, an approximation of R such that $R \le \tilde{R} < 2R$ and $\tilde{r} := 2^{\tau+1} r$, we get*

(i)
$$|\tilde{f}(z)| > \frac{\tilde{r}|\tilde{f}'(m)|}{8} > \frac{\sigma(\tilde{z}_i, \tilde{f})|\tilde{f}'(\tilde{z}_i)|}{135n^3}$$
for all z on the boundary of the disc $\Delta_{\tilde{r}}(m)$.

(ii) *If $\tilde{r} < \frac{3}{4}$ and $\mu < \frac{\tilde{r}|\tilde{f}'(m)|}{8n}$, then f has exactly one root in $\Delta_{\tilde{r}}(m)$.*

(iii) *If $\mu < \mu_f$ and all roots of f are contained in $\Delta_{1/8}(0)$, then $\tilde{r} < \frac{1}{2}$ and $\mu < \frac{\tilde{r}|f'(m)|}{24n}$, thus, (ii) applies to each root $\tilde{z}_i$ of $\tilde{f} \in f_\mu$.*

Proof. (i) follows directly from Lemma 9 (iii). For (ii), we apply Rouché's Theorem to the functions f and $\tilde{f}$. If $\tilde{r}$ and μ fulfill the conditions in (ii), then $|(f-\tilde{f})(z)| < n\mu < |\tilde{f}(z)|$ for all z on the boundary of $\Delta_{\tilde{r}}(m) \subset \Delta_1(0)$. Hence, both functions must have the same numbers of roots within this disc, namely, one. It remains to prove (iii). If μ is sufficiently small with respect to f (i.e., $\mu \le \mu(f)$), then each root z_i of f moves less than $\frac{\sigma(z_i,f)}{n} < \frac{1}{8}$ when passing from f to $\tilde{f}$ due to Corollary 3. Thus, the disc $\Delta_{1/4}(0)$ contains all roots of $\tilde{f}$. It follows that $\tilde{r} < \frac{1}{2}$ because, otherwise, $\mathscr{T}^{\tilde{f}'}(m, \tilde{r})$ cannot succeed. Furthermore, due to Lemma 2 (iii), $(1-2^{-9})\mu$ is sufficiently small with respect to $\tilde{f}$. Hence, with (i), we have
$$\mu < (1-2^{-9})^{-1} \cdot \frac{\sigma(\tilde{z}_i, \tilde{f})|\tilde{f}'(\tilde{z}_i)|}{2^9 n^4} < \frac{\tilde{r}|\tilde{f}'(m)|}{24n}.$$
□

Theorem 10 is crucial for the second step in our algorithm (see Section 3.1.2). After isolating a root $\tilde{z}_i$ of $\tilde{f}$ by means of a corresponding disc $\Delta_r(m)$ such that $\mathscr{T}^{\tilde{f}'}(m, 8nr)$ holds, we can check whether the condition in 10 (ii) holds. If it holds, $\Delta_{\tilde{r}}(m)$ isolates a corresponding root z_i of f. Otherwise, we know that μ is not sufficiently small.

3. Algorithm

As already sketched before, we want to isolate the roots $z_1, \ldots, z_n$ of f via isolating the roots $\tilde{z}_1, \ldots, \tilde{z}_n$ of a ρ-binary approximation $\tilde{f} \in [f]_{2^{-\rho}}$ (see Section 2 for definitions) first and, then, enlarging the so obtained isolating regions. Following this approach, we can only succeed if each root z_i of f does not move too far compared to its separation $\sigma(z_i, f)$ when passing from f to $\tilde{f}$. In particular, for isolating only the real roots of f, we must ensure that real roots stay real and non-real roots stay non-real. Corollary 3 gives a bound on ρ such that this is guaranteed, however, its usage assumes that we are aware of the values $\sigma(z_i, f)$ and $|f'(z_i)|$ which are both initially unknown. The idea to get over this

hurdle is to start with a certain $\rho = \rho_0 \geq 1$ and to apply a subdivision algorithm to isolate the roots of $\tilde{f}$. The subdivision procedure is guarded in the following way:

- Regions which are much smaller than the separation $\sigma(f)$ of f are never subdivided; see Theorem 8 (iv) and Sections 3.1.1 and 3.1.3.
- Whenever we observe that $\tilde{f}$ is not a sufficiently good approximation of f, that is, $2^{-\rho} > \mu(f)$ with $\mu(f)$ as defined in Definition 1, we double ρ and start over the entire algorithm.

After isolating the roots of $\tilde{f}$, we aim to derive isolating regions for the roots of f; see Theorem 10 and the subroutine CERTIFY in Section 3.1.2. We remark that, also for this step, it is crucial that ρ has been chosen large enough. Again, if we observe that $\tilde{f}$ is not a sufficiently good approximation of f, we double ρ and start over the entire algorithm. It is important to mention that the above steps in our algorithm may succeed even if $2^{-\rho} > \mu(f)$. Unfortunately, this means that there is no guarantee that we have captured all roots of f. Hence, in a final certification step, we check whether this is the case. This is easy for complex root isolation. Namely, if we have found n disjoint isolating regions for the roots of f, all roots must have been captured. Isolating the real roots only is more tricky because no global counting argument applies in this case. Therefore, we present an additional method in Section 3.1.3 to show that f cannot vanish outside the determined isolating intervals.

Remark: In [23], the overall method is also guarded. There, the guard is based on the perturbation bound due to Schönhage; see Section 2.3. Essentially, intervals are not allowed to become smaller than the possible worst case perturbation $9 \cdot 2^{-\rho/n}$. If this happens, the subdivision process stops and the algorithm restarts with a better approximation $\tilde{f}$ of f. In the description of our algorithm (Sections 3.1.1- 3.1.3), we use a guard based on our result in Theorem 8 (iv). The corresponding test allows a more adaptive precision management based upon the root perturbation bound from Lemma 2.

Before we start with the description of the algorithm, we remark that, for an arbitrary ρ-binary approximation $\tilde{f}$ of f, the roots of $\tilde{f}$ are contained within the disc $\Delta_{1/4}(0)$. Namely, since each point z on the boundary of $\Delta_{1/4}(0)$ is at least $1/8$ away from any root of f, we have $|f(z)| \geq |a_n| 8^{-n} = (8\Gamma)^n 8^{-n} = \Gamma^n$ and $|(f-\tilde{f})(z)| \leq \sum_{i=0}^{n-1} 2^{-\rho}|z|^i \leq 2^{-\rho}\sum_{i=0}^{n-1}(1/4)^i < 1 \leq \Gamma^n$. Hence, our claim follows from Rouché's Theorem.

3.1. Real Root Isolation

The proposed method applies to a wide class of subdivision solvers to isolate the real roots of a polynomial g with *rational* coefficients. The user has the freedom to choose his favorite method such as the Descartes algorithm (also termed as VCA-bisection algorithm due to Vincent, Collins and Akritas), a continued fraction solver, EVAL (see the proceeding description) or Sturm. Throughout the following considerations, we denote this method ISO. ISO provides exclusion predicates denoted $\mathbf{P}_{\text{ex}}^{\text{ISO}}$ and inclusion predicates denoted $\mathbf{P}_{\text{in}}^{\text{ISO}}$ which both apply to the polynomial g and an interval $I \subset \mathbb{R}$ with rational

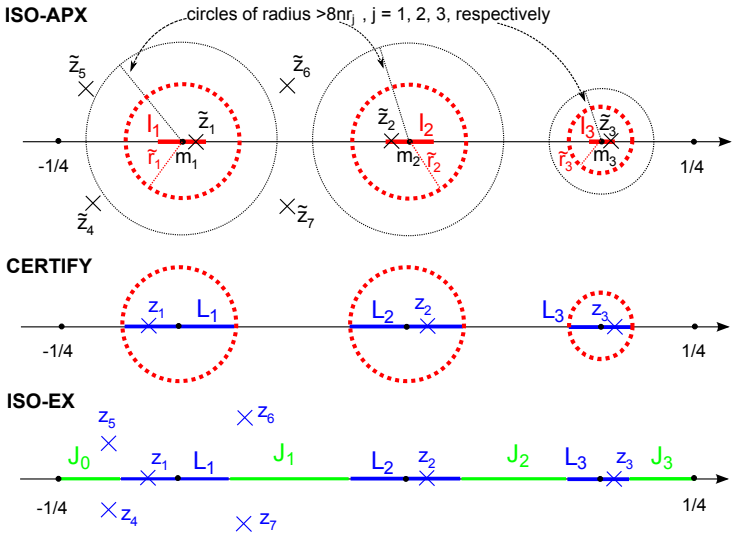

FIGURE 2. ISO-APX determines isolating intervals I_j for the real roots $\tilde{z}_j$ of $\tilde{f}$ such that $\mathscr{T}'_{3/2}(m_j, 8nr_j)$ holds. CERTIFY checks whether $\Delta_{\tilde{r}_j}(m_j)$ ($\tilde{r}_j > r_j$ as in Theorem 10) is isolating for a corresponding root z_j of f and returns isolating intervals L_j for z_j. ISO-EX ensures that the intervals $J_0, \ldots, J_m$ "in between" do not contain any further root of f.

endpoints. More precisely, if $\mathbf{P}^{\mathrm{ISO}}_{\mathrm{ex}}(g, I)$ holds, then I does not contain a root of g whereas I is isolating for a real root of g if $\mathbf{P}^{\mathrm{ISO}}_{\mathrm{in}}(g, I)$ holds. For instance, the VCA bisection algorithm and the continued fraction solvers apply Descartes' Rule of Signs as an exclusion as well as an inclusion predicate. However, in order to speed up computation, continued fraction solvers additionally use root bounds to discard intervals but integrate this step into the subdivision strategy.

The proposed algorithm denoted ISO* can be considered as combination of ISO and EVAL. We provide a brief description of EVAL:

EVAL is based on the tests $\mathscr{T}_1$ and $\mathscr{T}'_1$ as defined in Section 2.4. If, for an interval $I = (a, b)$, the test $\mathscr{T}_1(I)$ holds, then I contains no root of g. Thus, $\mathscr{T}_1$ serves as an exclusion predicate. $\mathscr{T}_1$ in combination with sign computation of g at the endpoints of I constitutes both an inclusion and exclusion predicate: If $\mathscr{T}'_1(I)$ holds then g' has no root in I, thus, g is monotone. Hence, it suffices to compare the signs of $g(a)$ and $g(b)$ in order to discard I or to certify I to be isolating. Recent work [29] shows that, despite its simpleness, EVAL achieves similar complexity bounds for the size of the induced recursion tree and the bit complexity as the more sophisticated real root isolation methods.

ISO* consists of three subroutines, namely, ISO-APX, CERTIFY and ISO-EX. ISO-APX and ISO-EX are subdivision methods based on ISO. CERTIFY comprises only one test

which is defined independently from ISO. We briefly outline their functionalities: ISO-APX determines isolating intervals $I_1,\ldots,I_m$ for the real roots of an approximation $\tilde{f}$ of f. CERTIFY extends these intervals to isolating intervals $L_1,\ldots,L_m$ for corresponding real roots of f. Finally, ISO-EX ensures that all real roots of f are captured; see Figure 2.

3.1.1. ISO-APX: Isolating roots of $\tilde{f}$. ISO-APX is a subdivision method to determine isolating intervals for the real roots of $\tilde{f}$. As exclusion predicate, it uses a combination of $\mathbf{P}_{\text{ex}}^{\text{ISO}}$ (provided by ISO) and the exclusion predicate provided by EVAL. Following this approach, we can achieve efficiency of ISO-APX with respect to the size of the induced subdivision tree. As inclusion predicate, we only use the test $\mathscr{T}'_{3/2}$ in combination with sign computation of $\tilde{f}$ at the endpoints of an interval I; the inclusion predicate $\mathbf{P}_{\text{in}}^{\text{ISO}}$ of ISO is dropped, the subdivision strategy is carried over from ISO. We give an outline for our subroutine (see Appendix, Algorithm 1 for pseudo-code)

ISO-APX maintains a dynamic list $\mathscr{A}$ of *active* intervals and a list $\mathscr{O}$ of *isolating* intervals for the roots of $\tilde{f}$. It initially starts with a $\rho=\rho_0\in\mathbb{N}$, a corresponding ρ-binary approximation $\tilde{f}$ of f, and lists $\mathscr{A}=\{(-1/4,1/4)\}$, $\mathscr{O}=\emptyset$. For each interval $I=(a,b)\in\mathscr{A}$ with center $m=m(I)$ and radius $r=r(I)$, we proceed as follows:

1. First, I is deleted from $\mathscr{A}$;
2. If $\mathbf{P}_{\text{ex}}^{\text{ISO}}(f,I)$ and $\tilde{f}(a)\cdot\tilde{f}(b)\neq 0$, do nothing // $\bar{I}=[a,b]$ contains no root of $\tilde{f}$;
3. If $\mathscr{T}_1(m,r)$ holds, do nothing // $\bar{I}$ contains no root of $\tilde{f}$;
4. If $\mathscr{T}'_{3/2}(m,8nr)$ holds, then $\bar{I}$ is added to $\mathscr{O}$ *if and only if* $\tilde{f}(a)\cdot\tilde{f}(b)\leq 0$ and I is not adjacent to any element in $\mathscr{O}$ // $\bar{I}$ contains a root ξ of $\tilde{f}$ and $\Delta_{8nr}(m)$ isolates ξ;
5. If none of the tests in 2.-4. holds, check whether the disc $\Delta_{64n^3r}(m)$ is terminal for $[\tilde{f}]_{2^{-\rho}}$. If it is terminal, we stop and start over again with $\rho\leftarrow 2\rho$ // because then $2^{-\rho}>\mu(f)$ due to Theorem 8 (iv);
6. Otherwise, subdivide I in accordance with the subdivision strategy of ISO and add all subintervals to $\mathscr{A}$. [1]

ISO-APX stops if $\mathscr{A}=\emptyset$ and returns $(\rho,\tilde{f},\mathscr{O})$.

Theorem 11. ISO-APX *never subdivides an interval of radius less than* $\sigma(f)2^{-9}n^{-5}$. *The list* $\mathscr{O}:=\{\bar{I}_1,\ldots,\bar{I}_m\}$ *contains pairwise disjoint isolating intervals for all roots of* $\tilde{f}$. *Furthermore,* $\mathscr{T}_{3/2}(m_j,8nr_j)$, *with* m_j *the center* r_j *the radius of* I_j, *succeeds for all* j.

Proof. Let I be an interval of radius $r(I)<\sigma(f)2^{-9}n^{-5}$. Because of $64n^3r(I)<\frac{\sigma(f)}{4n^2}$, the disc $\Delta_{64n^3r(I)}(m(I))$ is critical for $[f]_{2^{-\rho}}$; see Theorem 8 (i). Hence, if I passes Steps 2.-4. in ISO-APX, we must detect in Step 5 that $2^{-\rho}>\mu(f)$. It follows that the subdivision process stops and we start over with a larger ρ, thus, I is not subdivided. It remains to show that the intervals $\bar{I}_1\ldots,\bar{I}_m$ isolate *all* real roots of $\tilde{f}$. First, each of them isolates a real root because $\tilde{f}$ is monotone on each $\bar{I}_j$ and there is a sign change of $\tilde{f}$ at the endpoints of $\bar{I}_j$. Since we only add intervals to $\mathscr{O}$ that are disjoint to all intervals in $\mathscr{O}$, the $\bar{I}_j$'s are

[1] For classical bisection, we just add the two subintervals to $\mathscr{A}$. If ISO uses an additional exclusion predicate in the subdivision step (e.g., the continued fraction method applies a root bound to discard a certain subinterval), then we just add the remaining intervals to $\mathscr{A}$.

also pairwise disjoint. The following consideration further shows that no real root ξ of $\tilde{f}$ has been lost: Suppose that there exists a root ξ of $\tilde{f}$ which is not contained in any $\bar{I}_j$. Since all real roots of $\tilde{f}$ are contained within $(-1/4, 1/4)$, we must have discarded an interval I whose closure contains ξ. This is only possible if I is adjacent to some interval $\bar{I}_j \in \mathscr{O}$. If the width of I_j is larger than or equal to I, it follows that $\Delta_{8nr_j}(m_j)$ contains I. Since $\Delta_{8nr_j}(m_j)$ is isolating for a real root of $\tilde{f}$, this root must be ξ. It follows that ξ is one of the endpoints of $\bar{I}_j$, a contradiction to our assumption. The case where I is larger than I_j is treated in exactly the same way. Namely, the disc with radius $8nr(I)$ and center $m(I)$ then contains $\bar{I}_j$ and no other root than ξ. Hence, $\bar{I}_j$ isolates ξ, a contradiction. □

3.1.2. CERTIFY: Isolating intervals for the roots of f. We now aim to enlarge each of the intervals $\bar{I}_j$ to obtain intervals L_j which isolate corresponding real roots of f. The subroutine CERTIFY mainly comprises one test based on our results in Theorem 10. Each interval $\bar{I}_j \in \mathscr{O}$ with radius r_j and center m_j isolates a real root $\tilde{z}_j$ of $\tilde{f}$. Furthermore, the conditions in Theorem 10 are fulfilled, that is, each disc $\Delta_{r_j}(m_j)$ contains a root of $\tilde{f}$ and $\mathscr{T}'_{3/2}(m_j, 8nr_j)$ holds.

CERTIFY. For each $j = 1, \ldots, m$, determine a $\tau_j \in \mathbb{N}_0$ such that $R_j \le 2^{\tau_j+3} n r_i < 2R_j$, with

$$R_j := \sup\{t \in \mathbb{R}^+ : \mathscr{T}^{\tilde{f}'}_{3/2}(m_j, t) \text{ holds}\}.$$

This can be done in a number of $O(\log|\log\sigma(\tilde{z}_j)| + \log\log n) = O(\log|\log r_j| + \log\log n)$ steps via binary search. For $\tilde{r}_j := 2^{\tau_i+1} r_j$, check whether

$$2^{-\rho} < \frac{\tilde{r}_j |\tilde{f}'(m_j)|}{8n}. \tag{10}$$

If (10) holds for all j, then each disc $\Delta_{\tilde{r}_j}(m_j)$ and, thus, each interval

$$L_j := (m_j - \tilde{r}_j, m_j + \tilde{r}_j) \tag{11}$$

isolates a real root z_j of f; In case that (10) does not hold for any j, we must have $2^{-\rho} > \mu(f)$; see Theorem 10. In the latter case, we return "insufficient precision".

Lemma 12. *The intervals L_j defined in (11) are pairwise disjoint.*

Proof. We consider a pair $\Delta_{\tilde{r}_j}(m_j)$ and $\Delta_{\tilde{r}_k}(m_k)$ of discs. W.l.o.g., we can assume that $R_j \ge R_k$. It suffices to show that $|m_j - m_k| > \tilde{r}_j + \tilde{r}_k$. Since $\tilde{r}_i < R_i/(2n)$ for all i, the latter inequality is fulfilled if $|m_j - m_k| \ge (R_j + R_k)/(2n)$. From the definition of R_j the disc $\Delta_{R_j}(m_j)$ contains $\tilde{z}_j$ and no other root of $\tilde{f}$. It follows that $\Delta_{R_j}(m_j)$ cannot completely contain $\Delta_{r_k}(m_k)$ since $\Delta_{r_k}(m_k)$ is isolating for the root $z_k \neq z_j$. Thus,

$$|m_j - m_k| \ge R_j - r_k \ge R_j - \frac{R_k}{8n} \ge R_j - \frac{R_j}{8n} \ge \frac{15}{16} R_j \ge \frac{15}{32}(R_j + R_k) > \frac{(R_j + R_k)}{2n}.$$

Obviously, the intervals L_j are then also pairwise disjoint. □

We remark that CERTIFY succeeds under guarantee if $2^{-\rho} < \mu(f)$. In case of success, we proceed with ISO-EX; otherwise, we restart the overall algorithm with $\rho \leftarrow 2\rho$. For an actual implementation, we propose to integrate CERTIFY into ISO-APX. That is,

whenever we add an interval $\bar{I}_j$ to $\mathscr{O}$, we check whether (10) holds for I_j. Only for the sake of clarity, we decided to separate ISO-APX and CERTIFY in the presentation.

3.1.3. ISO-EX: All real roots of f are captured. It remains to show that the intervals L_j returned by CERTIFY contain all real roots of f. The crucial idea is to check whether $|\tilde{f}(x)| > 2^{-\rho+1}$ for all $x \in \Omega := (-1/4,1/4)\backslash\bigcup_j L_j$. If the latter inequality holds, then f has no root in Ω because $|(f-\tilde{f})(x)| \le \sum_{i=0}^{n} 2^{-\rho}(1/4)^{-i} < 2^{-\rho+1}$ for all $x \in (-1/4,1/4)$.

Lemma 13. *If $\tilde{f}$ is a sufficiently good ρ-binary approximation of f, that is, $2^{-\rho} < \mu(f)$, then $|\tilde{f}(x)| > 3\cdot 2^{-\rho+1}$ for all $x \in \Omega$.*

Proof. If $2^{-\rho} < \mu(f)$, then $2^{-\rho} < (1-2^{-9})^{-1}\mu(\tilde{f}) < \mu(\tilde{f},9)$; see Definition 1 and Lemma 2 (iii). Thus, when passing from $\tilde{f}$ to f, the roots $\tilde{z}_i$ of $\tilde{f}$ do not move by more than $\sigma(\tilde{z}_i)/(9n)$. It follows that the real roots $\tilde{z}_1,\ldots,\tilde{z}_m$ of $\tilde{f}$ stay real and the non-real roots $\tilde{z}_{m+1},\ldots,\tilde{z}_n$ stay non-real. For $j=1,\ldots,m$, let $\Delta_j := \Delta_{\tilde{r}_j}(m_j)$ be the discs which intersect the real axes at the endpoints of L_j, and, for $j=m+1,\ldots,n$, let $\Delta_j := \Delta_{\sigma(\tilde{z}_j)/3n}(\tilde{z}_j)$. Due to Theorem 10, $|\tilde{f}(z)| > 3n\cdot 2^{-\rho}$ for any point z on the boundary of one of the discs $\Delta_1,\ldots,\Delta_m$. In addition, similar as in the proof of Lemma 2 (i), it follows that $|\tilde{f}(z)| > n\mu(\tilde{f},3) > 3n2^{-\rho}$ for any $z \in \partial\Delta_j$, $j=m+1,\ldots,n$. $\tilde{f}$ is holomorphic and $\lim_{z\to\infty}|\tilde{f}(z)| = \infty$. Hence, on $\mathbb{C}\backslash\bigcup_j\Delta_j$, the function $|\tilde{f}|$ becomes minimal for a z on the boundary of one of the discs Δ_j and, thus, $|\tilde{f}(z)| > 3n2^{-\rho}$ for $z \in \mathbb{C}\backslash\bigcup_j\Delta_j$. Since $\Delta_{m+1},\ldots,\Delta_n$ do not intersect the real axes, our claim follows. □

Ω consists of $m+1$ disjoint intervals $J_0,\ldots,J_m$ where each L_j separates J_{j-1} and J_j; see Figure 2. We formulate the final subroutine ISO-EX (see Appendix, Algorithm 2 for pseudo-code).

ISO-EX. Similar to ISO-APX, ISO-EX maintains a dynamic list $\mathscr{A}$ of active intervals. It initially starts with $\mathscr{A} := \{J_0,\ldots,J_m\}$ and, in each step, ISO-EX operates on an interval $I \in \mathscr{A}$ with endpoints a and b, center m and radius r as follows:

1. First, I is deleted from $\mathscr{A}$;
2. If $\mathbf{P}_{\text{ex}}^{\text{ISO}}(\tilde{f}',I)$ or $\mathscr{T}_1'(I)$ holds, check whether $\min(|\tilde{f}(a)|,|\tilde{f}(b)|) > 2^{-\rho+1}$. If the inequality holds, do nothing. Otherwise, return "insufficient precision" // $\tilde{f}$ is monotone on I and, thus, the minimum of $|\tilde{f}|$ on I is taken at one of the endpoints of I.;
3. If $\mathscr{T}_{3/2}(I)$ holds, check whether $\frac{1}{3}|\tilde{f}(m)| > 2^{-\rho+1}$. If the inequality holds, do nothing. Otherwise, return "insufficient precision" // $|f(x)| \ge \frac{1}{3}|\tilde{f}(m)|$ for all $x \in I$ due to Lemma 4 (i);
4. If none of the predicates in 2. or 3. holds, check whether $\Delta_{8n^2r}(m)$ is terminal for $[\tilde{f}]_{2^{-\rho}}$. If it is terminal, return "insufficient precision" // because then $2^{-\rho} > \mu(f)$ due to Theorem 8 (iv);
5. Otherwise, subdivide I in accordance with the subdivision strategy of ISO and add all subintervals to $\mathscr{A}$. [2]

[2] If subintervals $I' = (a',b') \subset I$ are discarded in this step (e.g., a continued fraction solver discards intervals due to the application of a root bound), then we must also check whether $\min(|\tilde{f}(a')|,|\tilde{f}(b')|) > 2^{-\rho+1}$.

ISO-EX stops if $\mathscr{A} = \emptyset$ and returns "$f(x) \neq 0$ for all $x \in \Omega$".

Remark: Instead of subdividing Ω, we can alternatively subdivide $(-1/4, 1/4)$. More precisely, we initially set $\mathscr{A} = (-1/4, 1/4)$ in the above algorithm and proceed on intervals $I = (a,b)$ in a similar way as described above. The only difference is that instead of evaluating $\tilde{a}$ and $\tilde{f}(b)$, we have to evaluate $\tilde{f}$ at the endpoints of $I \cap \Omega$. For instance, in Step 2 of ISO-EX, we check whether $\mathbf{P}_{\text{ex}}^{\text{ISO}}(\tilde{f}', I)$ or $\mathscr{T}_1'(I)$ holds. If one of these predicates apply, $\tilde{f}$ is monotone on I. The intersection of I and Ω decomposes into intervals (a', b'), where $\tilde{f}$ is monotone as well. Since $\tilde{f}$ has no root in Ω, $|\tilde{f}|$ becomes minimal at a' or b'. Hence, if $\min(|\tilde{f}(a')|, |\tilde{f}(b')|) > 2^{-\rho+1}$, I contains no root of f. For an actual implementation, we propose not to follow this approach, however, for our complexity analysis as presented in Section 4, the argument becomes much simpler.

If ISO-EX succeeds, then $f(x) \neq 0$ for all $x \in \Omega$ and, thus, the intervals $L_1, \ldots, L_m$ isolate *all* real roots of f. We remark that ISO-EX succeeds if $2^{-\rho} \leq \mu(f)$; see Lemma 13. Furthermore, no interval of width $r(I) < \sigma(f)/(32n^4)$ is further subdivided. Namely, for each such interval, the disc $\Delta_{8n^2 r(I)}$ must be terminal for $[f]_{2^{-\rho}}$. We can now present our overall algorithm to isolate the roots of F:

ISO*. For given F as in (1), let $f(x) := F(8\Gamma x)/A_n$ as defined in (2) and $\rho_0 := 1$. Then, ISO-APX returns a $\rho \in \mathbb{N}$, a ρ-binary approximation $\tilde{f}$ of f, and a list $\mathscr{O} = \{\bar{I}_1, \ldots, \bar{I}_m\}$ of isolating intervals for the real roots of $\tilde{f}$. If CERTIFY returns "insufficient precision" for $(\rho, \tilde{f}, \mathscr{O})$, we set $\rho_0 := 2\rho$ and start over again. Otherwise, CERTIFY returns disjoint intervals $L_1, \ldots, L_m$ each isolating a real root of f. Let $\Omega = \{J_0, \ldots, J_m\} = [-1/4, 1/4] \setminus \bigcup_i L_i$ be the set of intervals in between the L_i's. If ISO-EX returns "insufficient precision" for $(\rho, \tilde{f}, \Omega)$, we set $\rho_0 := 2\rho$ and start over. Otherwise, the intervals $L_1, \ldots, L_m$ isolate all roots of f and, thus, the scaled intervals $8\Gamma \cdot L_1, \ldots, 8\Gamma \cdot L_m$ isolate all real roots of F.

We summarize:

Theorem 14. *Let F be a polynomial as in (1) and $f(x) = F(8\Gamma x)/A_n$ as defined in (2).*

(i) ISO* *isolates all real roots of F and demands for an approximation of F to*

$$n\log(8\Gamma) + \log(\mu(F)) = O(nL + \Sigma(F))$$

bits after the binary point.

(ii) ISO* *increases the precision for the approximation of F at most*

$$\lceil \log\log(\mu(f)) \rceil = O(\log\log(nL + \Sigma(F)))$$

many times.

(iii) *The subdivision routines* ISO-APX *and* ISO-EX *do not subdivide intervals of width less than* $\sigma(f)2^{-8}n^{-5} = \sigma(F)2^{-11}n^{-5}\Gamma^{-1}$.

Proof. It remains to prove (ii). In the first round, we start with $\rho := 1$. Since ρ is doubled in each round, we have $2^{-\rho} < \mu(f)$ after $\lceil \log\log(\mu(f)) \rceil$ many steps. Then, $\tilde{f} \in [f]_{2^{-\rho}}$ is a sufficiently good approximation of f and, thus, all subroutines ISO-APX, CERTIFY and ISO-EX succeed. □

3.2. Complex Root Isolation

We first outline some recent results on CEVAL [29], a complex root isolation method for square-free polynomials $g \in \mathbb{Q}[x]$, $n := \deg g$. CEVAL is a subdivision method based on Weyl's approach and can be considered as the complex counterpart of EVAL. CEVAL uses the predicates $T_1(\Delta)$ and $T'_{3/2}(\Delta)$ for discs $\Delta := \Delta_r(m)$. We have already seen that Δ does not contain any root of g if $T_1(\Delta)$ holds and Δ contains at most one root if $T'_{3/2}(\Delta)$ holds; see Lemma 4 and 5. Furthermore, $T_1(\Delta)$ also applies as inclusion predicate. Namely, if $\Delta_{2nr}(m)$ contains no root of g, then $\frac{1}{m-z_i} \leq \frac{1}{2nr}$ for all $i = 1,\ldots,n$. Then, as in the proof of Lemma 5, it follows that $T_1(\Delta)$ holds. We fix this result:

Lemma 15. *If $T_1(\Delta)$ does not hold, then $\Delta_{2nr}(m)$ contains a root of g.*

The CEVAL algorithm is now formulated as follows: Let B_0 be a squared box that contains all roots of g and $\mathscr{A} := \{B_0\}$, $\mathscr{O} := \emptyset$. For a box $B \in \mathscr{A}$ with center m and size s (= length of a side), we proceed as follows: If $\mathscr{T}_1(m, 3s/4)$ holds, then B contains no root of g and, thus, we discard B. If $\mathscr{T}_1(m, 3s/4)$ does not hold and $\mathscr{T}'_{3/2}(m, 3ns)$ holds, then $\Delta := \Delta(m, 3ns/2)$ contains a root ξ due to the above Lemma and $\Delta^+ := \Delta(m, 3ns)$ isolates ξ. We add Δ to $\mathscr{O}$ if Δ intersects no other disc in $\mathscr{O}$. CEVAL terminates when $\mathscr{A}$ becomes empty and returns the list $\mathscr{O}$ of isolating disc.

Correctness and termination are easy to see. Furthermore, for integer polynomials of degree n with integer coefficients of bitsize L, CEVAL induces a subdivision tree of size $\tilde{O}(n^2L)$ and isolating all roots demands for $\tilde{O}(n^4L^2)$ bit operations; see [29] for details.

Similar as in the case of real root isolation, we now aim to extend an arbitrary exact subdivision method $\text{ISO}_{\mathbb{C}}$ (e.g., CEVAL) for isolating the complex roots of a polynomial $g \in \mathbb{Q}[x]$ to a corresponding method $\text{ISO}^*_{\mathbb{C}}$ for isolating the roots of a bitstream polynomial. Let $\mathbf{P}^{\text{ISO}}_{\text{ex}}$ denote the exclusion predicate used by $\text{ISO}_{\mathbb{C}}$ to discard boxes that do not contain a root of g. $\text{ISO}^*_{\mathbb{C}}$ decomposes into two subroutines denoted $\text{ISO-APX}_{\mathbb{C}}$ and $\text{CERTIFY}_{\mathbb{C}}$. $\text{ISO-APX}_{\mathbb{C}}$ can be considered a natural extension of its real counterpart ISO-APX. The subroutine $\text{CERTIFY}_{\mathbb{C}}$ is equal to CERTIFY as defined in Section 3.1.2.

$\text{ISO}_{\mathbb{C}}$. $\text{ISO-APX}_{\mathbb{C}}$ starts with $\rho = \rho_0$ and a ρ-binary approximation $\tilde{f}$ of f, where we initially set $\rho_0 = 1$. Similar as in ISO-APX, we maintain a list of active boxes $\mathscr{A}$ and a list $\mathscr{O}$ of isolating discs. Initially, we set $\mathscr{A} = \{B_0\}$ and $\mathscr{O} := \emptyset$, where B_0 denotes the box with center $m = 0$ and size $1/2$. Then, B_0 contains all roots of $\tilde{f}$. For a box $B \in \mathscr{A}$ with center m and size s, we proceed as follows:

1. First, B is deleted from $\mathscr{A}$;
2. If $\mathbf{P}^{\text{ISO}}_{\text{ex}}(f, B)$ or $\mathscr{T}_1(m, 3s/4)$ holds, do nothing // B contains no root of $\tilde{f}$;
3. If $\mathscr{T}'_{3/2}(m, 12n^2s)$ holds, then $\Delta := \Delta_{3ns/2}(m)$ is added to $\mathscr{O}$ *if and only if* Δ does not intersect any other disc in $\mathscr{O}$ // Δ isolates a root not already captured by a disc in $\mathscr{O}$;
4. If none of the tests in 2. or 3. holds, check whether the disc $\Delta_{96n^4s}(m)$ is terminal for $[\tilde{f}]_{2^{-\rho}}$. If it is terminal, we stop and start over again with $\rho \leftarrow 2\rho$ // $2^{-\rho} > \mu(f)$ due to Theorem 8 (iv);

5. Otherwise, subdivide B in accordance with the subdivision strategy of ISO and add all sub boxes to $\mathscr{A}$;

ISO-APX$_{\mathbb{C}}$ terminates when $\mathscr{A}$ becomes empty and returns a list $\mathscr{O} = \{\Delta_1, \ldots, \Delta_n\}$ of pairwise disjoint discs which isolate all complex roots of $\tilde{f}$. For each disc $\Delta_j = \Delta_{r_j}(m_j)$, the test $\mathscr{T}'_{3/2}(m_j, 8nr_j)$ holds. CERTIFY$_{\mathbb{C}}$ is now equal to CERTIFY, that is, for each j, we compute a corresponding $\tilde{r}_j$ as in the description of CERTIFY and check whether the inequality (10) holds. If it holds for all j, the enlarged discs $\Delta_{\tilde{r}_j}(m_j)$ are pairwise disjoint and isolate all roots of f. If (10) does not hold for one of the j's, we stop and restart the overall algorithm with $\rho \leftarrow 2\rho$.

We remark that, for isolating all complex roots of f, there is no need for a third subroutine such as ISO-EX in ISO* since a global counting argument applies. Namely, having determined n disjoint isolating regions for the roots of f, this implies that all roots of f are captured. The following Theorem summarizes our results. Its proof is completely similar as the proof for the corresponding result in Theorem 14 for real root isolation.

Theorem 16. *Let F be a polynomial as in (1) and $f(x) = F(8\Gamma x)/A_n$ as defined in (2).*

(i) ISO$_{\mathbb{C}}$ *isolates all real roots of F and demands for an approximation of F to*

$$n\log(8\Gamma) + \log(\mu(F)) = O(nL + \Sigma(F))$$

bits after the binary point.

(ii) ISO* *increases the precision for the approximation of F at most*

$$\lceil \log\log(\mu(f)) \rceil = O(\log\log(nL + \Sigma(F)))$$

many times.

(iii) ISO-APX$_{\mathbb{C}}$ *does not subdivide boxes of size less than* $\frac{\sigma(f)}{2^{10}n^6} = \frac{\sigma(F)}{2^{13}n^6\Gamma}$.

4. Complexity Analysis

We provide a complexity analysis for VCA*, that is, our extension of the classical bisection algorithm VCA (also termed as the Descartes method in the literature) for isolating the roots of a square-free polynomial with bitstream coefficients.

4.1. Descartes' Rule of Signs

We first resume some basic facts about Descartes' Rule of Signs. For a polynomial $g(x) = \sum_{i=0}^{n} g_i x^n \in \mathbb{R}[x]$, it states that the number $var(g)$ of sign changes in the coefficient sequence of g, that is, the number of pairs (i, j) with $i < j$, $g_i g_j < 0$, and $g_{i+1} = \ldots = g_{j-1} = 0$, is no smaller than and of the same parity as the number of positive real roots of g. If $var(g) = 0$, then g has no positive real root, and if $var(g) = 1$, g 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 g in I exactly correspond to those of

$$g_I(x) := g(a + (b-a)x) \tag{12}$$

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

$$g_{I,\mathrm{rev}}(x) := (1+x)^n g_I\left(\frac{1}{x+1}\right) = (1+x)^n \cdot g\left(\frac{ax+b}{x+1}\right)$$

correspond bijectively to the real roots of g in I. The factor $(1+x)^n$ in the definition of $g_{I,\mathrm{rev}}$ clears denominators and guarantees that $g_{I,\mathrm{rev}}$ is a polynomial. We define $var(g,I)$ as $var(g_{I,\mathrm{rev}})$.

We fix the following property of $var(g,I)$ which will turn out crucial for the preceding analysis of VCA*. For a simple self-contained proof, we refer to [10, Corollary 2.27].

Theorem 17. *If the pairwise disjoint open intervals $J_1, \ldots, J_l$ are subsets of the open interval I, then*

$$var(g,I) \geq \sum_{1 \leq i \leq \ell} var(g,J_i).$$

4.2. Analysis of the Recursion Tree

We now analyze the recursion trees induced by VCA-APX and VCA-EX when applied to a polynomial f as defined in (2). In order to simplify the argument, we assume that VCA-EX is modified in a way such that it starts subdividing the interval $(-1/4,1/4)$ (instead of the intervals $J_0,\ldots,J_m$) and "merges" the subintervals with Ω in each step; see the remark following the description of ISO-EX in Section 3.1.3. For a fixed $\rho \in \mathbb{N}$ and a ρ-binary approximation $\tilde{f}$ of f, we denote the recursion trees for VCA-APX and VCA-EX by T_{apx} and T_{ex}, respectively. We first sketch our approach: For each internal node (interval) I of T_{ex}, we have $var(\tilde{f},I) > 0$ and, in addition, a certain neighborhood of I contains at least two roots of f. We now define a recursion tree $T(\tilde{f})$ in accordance to the latter two properties such that T_{apx} is a subtree of $T(\tilde{f})$. Eventually, we show that $|T_{\mathrm{apx}}| \leq |T(\tilde{f})| = O(n \log n + \Sigma(f))$. For T_{ex}, we proceed in exactly the same manner.

Definition 3. *For g an arbitrary polynomial, let $T(g)$ be the subdivision tree consisting of all intervals obtained by recursive bisection of the interval $I_0 := (-1/4,1/4)$ in accordance with the following rule: At depth $h \in \mathbb{N}_0$, an interval*

$$I = (-1/4 + i2^{-h-1}, -1/4 + (i+1)2^{-h-1}), \quad i \in \{0,\ldots,2^h-1\},$$

is subdivided if and only if $var(g,I) \geq 1$ and $\Delta_{2^9 n^5 r(I)}(m(I))$ contains a root ξ of f with separation $\sigma(\xi,f) < 2^8 n^5 r(I)$.

Lemma 18. *T_{apx} is a subtree of $T(\tilde{f})$ and T_{ex} is a subtree of $T(\tilde{f}')$.*

Proof. For the first claim, it suffices to show that each internal node $I=(a,b)$ of T_{apx} is also an internal node of $T(\tilde{f})$. If I is further subdivided by VCA-APX, then $var(\tilde{f},I) > 0$ and the disc $\Delta = \Delta_{64n^3 r(I)}(m(I))$ is not terminal for $[\tilde{f}]_{2^{-\rho}}$. Then, from Theorem 8 (iii), it follows that the disc $\Delta_{384n^5 w(I)}(m(I))$ contains a root of f with separation less than $256n^5 w(I)$. Hence, I is an internal node of $T(\tilde{f})$. The second claim follows in analogous manner. □

The following considerations will show that, for arbitrary g with $\deg g \leq n$, the size of $T(g)$ is bounded by $O(n \log n + \Sigma(f))$. We introduce the following notations: Let $T_h \subset T(g)$ be the set of all nodes at depth h and T_h^* the set of all leaves at the same depth. Furthermore, let

$$\begin{aligned}
\lambda(h) &:= |T_h| \text{ the number of nodes at depth } h,\\
\lambda^*(h) &:= |T_h^*| \text{ the number of leaves at depth } h,\\
\lambda^\#(h) &:= \lambda(h) - \lambda(h)^* \text{ the number of internal nodes at depth } h, \text{ and}\\
\nu(h) &:= \sum_{I \in T_h} var(g, I) \text{ the sum of all sign variations for } g \text{ at depth } h.
\end{aligned}$$

W.l.o.g., we can assume that $\sigma(z_1, f) \leq \ldots \leq \sigma(z_n, f)$. We define

$$h^* := \lceil 5 \log n \rceil + 8 \quad \text{and} \quad h_i := \left\lceil \log \sigma(z_i, f)^{-1} \right\rceil$$

for all $i = 1 \ldots, n$. For an arbitrary $h \in \mathbb{N}$, $k(h)$ *denotes the number of roots* z_i *with* $h_i \geq h - h^*$. Thus, we get

$$\sigma(z_i, f) \geq 2^{-h_i} > 2^{h^* - h} > (n^5 2^8) \cdot 2^{-h-2} \text{ for } i = k(h)+1, \ldots, n.$$

We say that z_i of f is *critical for an interval* I if $z_i \in \Delta_{2^9 n^5 r(I)}(m(I))$ and $\sigma(z_i, f) < 2^8 n^5 r(I)$. Hence, since $\sigma(z_i, f) \geq (n^5 2^8) \cdot 2^{-h-2}$ for $i \geq k(h)+1$, z_i cannot be critical for an interval I of radius $r(I) \leq 2^{-h-2}$. From the definition of $T(g)$, it immediately follows:

Lemma 19. *A node I in $T(g)$ is terminal if there exists no root z_i of f that is critical for I. For nodes $I \in T_{h'}$ at depth $h' \geq h$, only the roots $z_1, \ldots, z_{k(h)}$ can be critical.*

For each root z_i, all but at most two intervals $I \in T_h$ fulfill the inequality $|z_i - m(I)| > 2^{-(h+1)}$. It follows that, for all but at most $2k(h)$ intervals $I \in T_h$, we must have $|z_i - m(I)| > 2^{-(h+1)}$ for all $i = 1, \ldots, k(h)$. We consider an internal node $I \in T_h \backslash T_h^*$ that fulfills this inequality. Then, the following consideration shows that, at depth $h' := h + h^*$, there cannot be any interval $I' \in T_{h'}$ with $I' \subset I$: Assume that there exists such an interval I'. Then, I' is one of the two children of a $J \in T_{h'-1}$ and $J \subset I$. For the midpoint of J, we get $|m(J) - z_i| > 2^{-(h+1)} = 2^{h^*-1} 2^{-h'} \geq n^5 2^{7-h'} = n^5 2^8 r(J)$ for all $i = 1, \ldots, k(h)$. Hence, it follows that none of the roots $z_1, \ldots, z_{k(h)}$ is critical for J. According to Lemma 19, none of the roots $z_{k(h)+1}, \ldots, z_n$ is critical for J as well and, thus, J must be terminal, a contradiction. Since I is an internal node, we must have $var(g, I) \geq 1$, and since I has no children in $T_{h'}$, Theorem 17 implies that, at depth h', Descartes' Rule of Sign counts at least one sign variation less for g than at depth h. The latter applies to at least $\lambda^\#(h) - 2k(h)$ intervals at depth h and, thus, we obtain:

Lemma 20. *For $r(h) := \lambda^\#(h) - 2k(h)$, it holds that $\nu(h + h^*) \leq \nu(h) - r(h)$.*

We can now bound the size of $T(g)$: From Lemma 19 and $k(h) = 0$ for all $h > h_1 + h^*$, it follows that $T(g)$ has no nodes at depth $h > h_1 + h^*$. For a certain depth h with $1 \leq h \leq h^*$, we consider the sequence $T_h, T_{h+h^*}, T_{h+2h^*}, \ldots$ corresponding to the levels

$h, h+h^*, h+2h^*, \ldots$ in the recursion tree $T(g)$. From Lemma 20 and $v(h) \leq \deg g \leq n$ for all h, we obtain the following computation:

$$\begin{aligned}\sum_{i=0}^{\lceil h_1/h^*\rceil} \lambda^{\#}(h+ih^*) &= \sum_{i=0}^{\lceil h_1/h^*\rceil} \lambda^{\#}(h+ih^*) = \sum_{i=0}^{\lceil h_1/h^*\rceil} (r(h+ih^*)+2k(h+ih^*)) \\ &= 2\sum_{i=0}^{\lceil h_1/h^*\rceil} k(h+ih^*) + \sum_{i=0}^{\lceil h_1/h^*\rceil} r(h+ih^*) \\ &\leq 2\sum_{i=0}^{\lceil h_1/h^*\rceil} k(h+ih^*) + \sum_{i=0}^{\lceil h_1/h^*\rceil} (v(h+ih^*) - v(h+(i+1)h^*)) \\ &\leq v(h) + 2\sum_{i=0}^{\lceil h_1/h^*\rceil} k(h+ih^*) \leq n + 2\sum_{i=0}^{\lceil h_1/h^*\rceil} k(h+ih^*)\end{aligned}$$

Since $T(g)$ is a binary tree, the number of nodes is bounded by two times the number of internal nodes. Hence, summing up over all levels leads to the following result:

$$\begin{aligned}|T(g)| &\leq 2\cdot \sum_{h=1}^{h^*} \sum_{i=0}^{\lceil h_1/h^*\rceil+1} \lambda^{\#}(h+ih^*) \leq 2h^* + 4\sum_{h=1}^{h^*} \sum_{i=0}^{\lceil h_1/h^*\rceil} k(h+ih^*) \\ &= 2h^*n + 4\sum_{i=0}^{h_1+h^*} k(i) = O(n\log n) + 4\sum_{i=0}^{h_1+h^*} k(i)\end{aligned}$$

It remains to bound the sum on the right side of the last inequality. From the definition of $k(i)$, it holds that each root z_j contributes to this sum at most $h^* + h_j = O(\log n - \log \sigma(z_j, f))$ many times. Namely, for all $i > h^* + h_j$, the root z_j is no longer counted in $k(i)$. It follows that the size of $|T(g)|$ is bounded by $O(n\log n + \Sigma(f))$. Hence, using Lemma 18, we conclude:

Theorem 21. *Let F and f be defined as in (2) and (1), respectively. Then, for an arbitrary but fixed $\rho \in \mathbb{N}$, the subroutines* VCA-APX *and* VCA-EX *induce recursion trees of size*

$$O(n\log n + \Sigma(f)) = O(n(\log n + L) + \Sigma(F)).$$

4.3. Bit Complexity

In the previous section, we already derived bounds on the size of the recursion trees T_{apx} and T_{ex} induced by VCA-APX and VCA-EX, respectively. In the last step, we aim for a bound on the number of bit operations needed at a certain node of these trees.

Before we start with our analysis, we fix the following definition:

Definition 4 (bitsize). *A polynomial $g(x) := \sum_{i=0}^{n} g_i z^i$ with coefficients $g_i = m_i \cdot 2^{-\tau_i}$, $m_i \in \mathbb{Z}$ and $\tau_i \in \mathbb{N}_0$,* has bitsize $\tau(g)$ *if multiplication of g by the common denominator $2^{\max_i \tau_i}$ of all g_i leads to an integer polynomial with coefficients of at most $\tau(g)$ bits.*

Now let f be a polynomial as in (2), $\rho \in \mathbb{N}$ be fixed and $\tilde{f}$ a ρ-binary approximation of f. For the analysis of our algorithm, we can assume that $\rho = O(\log n + \Sigma(f))$; see Theorem 14 (i). In a first step, we bound the costs for computing $\tilde{f}_I(x) = \tilde{f}(a+(b-a)x)$ and $\tilde{f}_{[m(I),r(I)]}(x) = f(m(I)+r(I)x)$ at a node $I = (a,b)$ of the recursion trees T_{apx} and T_{ex}.

Lemma 22. *For fixed ρ, the cost for computing the polynomials $\tilde{f}_I(x)$ and $\tilde{f}_{[m(I),r(I)]}$ for all nodes $I=(a,b)$ of T_{apx} and T_{ex} is bounded by*

$$\tilde{O}(n^2(L-\log\sigma(f))(n+\Sigma(f))).$$

Furthermore, all these polynomials have maximal bitsize

$$\tau_{\max}=\tilde{O}(n(L-\log\sigma(f))).$$

Proof. For our starting interval $I_0:=(-1/4,1/4)$, the polynomial $\tilde{f}_{I_0}(x)=f(-1/4+x/2)$ has bitsize $O(nL+\Sigma(f))$ because f has coefficients of absolute value bounded by $2^{O(nL)}$ and $\tilde{f}$ approximates f to $\rho=O(\log n+\Sigma(f))$ bits after the binary point; see the preceding remark. Now, for given $\tilde{f}_I=\tilde{f}(a+(b-a)x)$, we compute $\tilde{f}_{I_\ell}=\tilde{f}(a+r(I)x)$, $I_\ell=(a,m(I))$, from $\tilde{f}_I$ via the substitution $x\mapsto x/2$. For the right subinterval $I_r=(m(I),b)$, $\tilde{f}_{I_r}=\tilde{f}(m(I)+r(I)x)=\tilde{f}_{[m(I),r(I)]}(x)$ is derived from $\tilde{f}_{I_\ell}$ via the substitution $x\mapsto x+1$. Hence, the bitsize of $\tilde{f}_I$ increases by at most n in each subdivision step. It follows that, for a node I at depth h, both polynomials $\tilde{f}_I$ and $\tilde{f}_{[m(I),r(I)]}$ have bitsize $O(n(L+h)+\Sigma(f))$. The depth of the recursion trees T_{apx} and T_{ex} is bounded by $O(\log n-\log\sigma(f))$ according to the proof of Theorem 21, hence, our claim on the bitsize follows. For the cost, we remark that the scaling by $1/2$ is easy because this is just a shift of the coefficients in the binary representation. The more costly step is the Taylor shift by 1, that is, $x\mapsto x+1$. Using asymptotically fast Taylor shift [14], we therefor need

$$\tilde{O}(n(n+\tau(\tilde{f}_I)))=\tilde{O}(n(n(L+h)+\Sigma(f)))=\tilde{O}(n^2(L-\log\sigma(f)))$$

bit operations at a node of level $h=O(\log n-\log\sigma(f))$ because $-n\log\sigma(f)\geq\Sigma(f)$. For the cost of computing the polynomials at all nodes, we thus get the bound

$$(|T_{\mathrm{apx}}|+|T_{\mathrm{ex}}|)\cdot\tilde{O}(n^2(L-\log\sigma(f)))=\tilde{O}(n^2(L-\log\sigma(f))(n+\Sigma(f)))$$

due to Theorem 21. □

We can now directly derive the bit complexity for one iteration of VCA-APX and VCA-EX.

Lemma 23. *For fixed ρ, the bit complexity of* VCA-APX *and* VCA-EX *is bounded by*

$$\tilde{O}(n^2(L-\log\sigma(f))(n+\Sigma(f))). \tag{13}$$

Proof. For $I=(a,b)$ an arbitrary node in the recursion tree T_{ex}, we have to compute

(A) $var(\tilde{f},I)$ (Step 2),
(B) the signs of $t_1:=t_1^{\tilde{f}}(m(I),r(I))$ and $t'_{3/2}:=t_{3/2}^{\tilde{f}'}(m(I),8nr(I))$ (Step 3 and 4),
(C) the signs of $\tilde{f}(a)$ and $\tilde{f}(b)$ (Step 2 and 4),
(D) whether the disc $\Delta:=\Delta_{64n^3r(I)}(m(I))$ is terminal for $[\tilde{f}]_{2^{-\rho}}$ (Step 5).

For (A), we have to evaluate $\tilde{f}_{I,\mathrm{rev}}(x)=(1+x)^n\tilde{f}(1/(1+x))$. $\tilde{f}_{I,\mathrm{rev}}(x)$ is obtained by reversing the coefficients of $\tilde{f}_I$ followed by a Taylorshift by 1. Since $\tilde{f}_I(x)$ has bitsize less than or equal $\tau_{\max}$ (see Lemma 22 for the definition), the corresponding costs are bounded by $\tilde{O}(n^2(L-\log\sigma(f)))$. For (B), instead of evaluating the sign of t_1 we can alternatively evaluate the sign of $t_1^{\tilde{f}_{[m(I),r(I)]}}(0,1)$. Since $\tau(\tilde{f}_{[m(I),r(I)]})\leq\tau_{\max}$, the cost for

this step also achieves the latter bound. Similarly, we compute $t_1^{(\tilde{f}_{[m(I),r(I)]})'}(0,8n)$ instead of $t'_{3/2} := t_{3/2}^{\tilde{f}'}(m(I),8nr(I))$. This is equivalent to the evaluation of a polynomial of bitsize $\leq \log n + \tau_{\max}$ at a $\log n$-bit number, hence, the bound $\tilde{O}(n^2(L-\log\sigma(f)))$ for the cost for this step applies as well. The evaluations of $\tilde{f}(a) = \tilde{f}_I(0)$ and $\tilde{f}(a) = \tilde{f}_I(1)$ amount for $O(n\tau_{\max})$ bit operations. It remains to analyze the costs for Step (E). For the test $\mathscr{T}^{[\tilde{f}]_{2^{-\rho}}}(m(I),64n^3r(I))$, we have to compute the polynomial $\tilde{f}(m(I)+x/2) = \tilde{f}_{[m(I),r(I)]}(x/w(I))$ first. This is just a shift of the coefficients of $\tilde{f}_{[m(I),r(I)]}$ by at most $-n\log w(I) = O(n(\log n - \log\sigma(f)))$ bits in the binary representation. Thus, $\tilde{f}_{[m(I),1/2]}$ has bitsize $O(n\log n+\sigma(f))$. In the next step, we enlarge (or decrease) its coefficients by $2^{-\rho}$. The resulting polynomial $g^{[2^{-\rho},m(I)]}$ as defined in (7) has also bitsize $O(n\log n+\sigma(f))$ because $\rho \leq \tau_{\max}$. We then compute $p(x) := g^{[2^{-\rho},m(I)]}(r(I)x)$ which is again a shift of the coefficients by $O(n(\log n - \log\sigma(f)))$ bits. Then, the test $\mathscr{T}^{[\tilde{f}]_{2^{-\rho}}}(m(I),64n^3r(I))$ is equivalent to $\mathscr{T}^p(0,128n^3)$ which amounts for an evaluation of a polynomial of bitsize $O(n(\log n - \log\sigma(f)))$ at a $\log n$-bit number. Again, the costs are in $\tilde{O}(n^2(L-\log\sigma(f)))$. In analogous manner, it follows that the evaluation of $\mathscr{T}^{[\tilde{f}]_{n2^{-\rho}}}(m(I),64n^3r(I))$ achieves the same bound. Thus, our claim follows since the number of nodes are in $O(n+\Sigma(f))$. □

It remains to consider the cost for the subroutine CERTIFY. The computation of $\tilde{r}_j$ (see Section 3.1.2)) demands for

$$O(\log|\log r_j| + \log\log n) = O(\log\log n + \log(\sigma(f)+\log n))$$

evaluations, each of cost $\tilde{O}(n^2(L-\log\sigma(f))$. Namely, since all intervals have size $\geq \sigma(f)2^{-9}n^{-5}$ (Theorem 14), we have $|\log r_j| = O(\log n + \log\sigma(f))$ and, thus, the cost for each evaluation $\mathscr{T}_{3/2}^{\tilde{f}'}(m_j,2^{\tau_j+3}nr_i) = \mathscr{T}_{3/2}^{\tilde{f}'_{[m_j+r_jx]}}(0,2^{\tau_j+3}n)$ is $\tilde{O}(n^2(L-\log\sigma(f))$. Obviously, the same bound also applies to the verify inequality (10). Summing up over all $j=1,\ldots,m\leq n$ shows that CERTIFY needs at most $\tilde{O}(n^3(L-\log\sigma(f)))$ bit operations.

We summarize:

Theorem 24. *Let F be a bistream polynomial as defined in (1). For isolating the real roots of F, we need coefficient approximations of F to $\tilde{O}(nL+\Sigma(F))$ bits after the binary point and the computational cost is bounded by*

$$\tilde{O}(n^2(L-\log\sigma(F))(nL+\Sigma(F))).$$

In case that F has integer coefficients, the computational costs are bounded by $\tilde{O}(n^4L^2)$.

Proof. Due to Lemma 23 and the above considerations, the total cost for a fixed precision ρ is bounded by $\tilde{O}(n^2(L-\log\sigma(F))(nL+\Sigma(F)))$. Since ρ is increased at most $O(\log\log(nL+\Sigma(F)))$ times (see Theorem 14), the claim follows. For integer polynomials, the above bound is a direct consequence of $\Sigma(F) = \tilde{O}(nL)$; see [10, 13] for a proof. □

5. Conclusion

We presented a new deterministic approach to isolate the (real or complex) roots of a square-free bitstream polynomial $F \in \mathbb{R}[x]$. Our method is formulated in a way such that it extends any exact subdivision method for rational polynomials to a version that isolates the roots of a bitstream polynomial. Previous methods only modify Descartes method to treat bitstream polynomials. Since continued fraction solvers sometimes outperform the classical bisection methods for rational polynomials, it would be interesting to see whether this behavior carries over to the bitstream setting.

In the description of our algorithm, we abstained from using the inclusion predicate $\mathbf{P}_{\text{in}}^{\text{ISO}}$ provided by the external solver ISO. However, in practice, such inclusion predicates often succeed much earlier than the one provided by the EVAL algorithm. Hence, for an actual implementation, we propose we propose to integrate $\mathbf{P}_{\text{in}}^{\text{ISO}}$ into our method in order to safe many subdivision steps.

We further expect to see many applications of our approach in the topology computation of algebraic curves and surfaces. One of the main drawbacks of the former approaches was their huge precision demand. As a consequence, for polynomials with complex algebraic terms as coefficients (e.g., subresultant polynomials), the approximation of the polynomial and not the root isolation itself becomes a bottleneck. This has already been observed in practice [5, 11, 19] as well as in the complexity analysis for topology computation [19]. Our algorithm works with a considerably improved precision management, thus we are confident that it will improve the overall efficiency of the algorithms for topology computations.

Finally, we aim for an extension of our approach to the *m-k* scenario [11], that is, we want to isolate the roots of a *not necessarily square-free* bitstream polynomial F for which the number m of distinct real roots and the degree k of $\gcd(F, F')$ is known from a precomputation step.

Acknowledgements. We want to thank the reviewers for their careful review and their helpful comments. Your comments helped us improving the presentation of the paper and simplifying notation.

References

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

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

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

[4] E. Beberich, P. Emeliyanenko, and M. Sagraloff. An elimination method for solving bivariate polynomial systems: Eliminating the usual drawbacks. In *ALENEX*, pages 35–47, Philadelphia, USA, 2011. SIAM.

[5] E. Berberich, M. Kerber, and M. Sagraloff. An efficient algorithm for the stratification and triangulation of an algebraic surface. *Computational Geometry: Theory and Applications (CGTA)*, 43(3):257–278, 2009.

[6] G. Collins, J. Johnson, and W. Krandick. Interval arithmetic in cylindrical algebraic decomposition. *JSC.*, 34:143–155, 2002.

[7] G. E. Collins and A. G. Akritas. Polynomial real root isolation using Descartes' rule of signs. In R. D. Jenks, editor, *Proceedings of the 1976 ACM Symposium on Symbolic and Algebraic Computation*, pages 272–275. ACM Press, 1976.

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

[9] 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, March 2007.

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

[11] A. Eigenwillig, M. Kerber, and N. Wolpert. Fast and Exact Geometric Analysis of Real Algebraic Plane Curves. In *Proc. of the 2007 International Symposium on Symbolic and Algebraic Computation (ISSAC 2007)*, pages 151–158, 2007.

[12] A. Eigenwillig, L. Kettner, W. Krandick, K. Mehlhorn, S. Schmitt, and N. Wolpert. An exact descartes algorithm with approximate coefficients. In *CASC*, volume 3718 of *LNCS*, pages 138–149, 2005.

[13] A. Eigenwillig, V. Sharma, and C. Yap. Almost tight complexity bounds for the Descartes method. In *ISSAC*, pages 71–78, 2006.

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

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

[16] 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*, pages 45–54, 2009.

[17] J. Johnson. Algorithms for polynomial real root isolation. In B. Caviness and J. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts and monographs in Symbolic Computation, pages 269–299. Springer, 1998.

[18] J. R. Johnson and W. Krandick. Polynomial real root isolation using approximate arithmetic. In W. Küchlin, editor, *ISSAC*, pages 225–232. ACM Press, 1997.

[19] M. Kerber. *Geometric Algorithms for Algebraic Curves and Surfaces*. PhD thesis, Universität des Saarlandes, 2009.

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

[21] T. Lickteig and M.-F. Roy. Sylvester-Habicht sequences and fast Cauchy index computation. *J. of Symbolic Computation*, 31:315–341, 2001.

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

[23] K. Mehlhorn and M. Sagraloff. Isolating real roots of real polynomials. In *ISSAC '09*, pages 247–254, New York, NY, USA, 2009. ACM. an extended version of this paper appears in J. of Symbolic Computation 2011.

[24] D. P. Mitchell. Robust ray intersection with interval arithmetic. In *Graphics Interface'90*, pages 68–74, 1990.

[25] B. Mourrain, F. Rouillier, and M.-F. Roy. The Bernstein basis and real root isolation. In J. E. Goodman, J. Pach, and E. Welzl, editors, *Combinatorial and Computational Geometry*, number 52 in MSRI Publications, pages 459–478. Cambridge University Press, 2005.

[26] V. Y. Pan. Sequential and parallel complexity of approximate evaluation of polynomial zeros. *Comput. Math. Applic.*, 14(8):591–622, 1987.

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

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

[29] M. Sagraloff and C. K. Yap. A simple but exact and efficient algorithm for complex root isolation, 2009. Submitted. http://mpi-inf.mpg.de/ msagralo/ceval.pdf.

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

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

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

[33] S. Smale. The fundamental theorem of algebra and complexity theory. *Bulletin (N.S.) of the AMS*, 4(1):1–36, 1981.

[34] B. T. Smith. Error bounds for zeros of a polynomial based upon Gerschgorin's theorems. *Journal of the ACM*, 17(4):661–674, Oct. 1970.

[35] E. P. Tsigaridas and I. Z. Emiris. On the complexity of real root isolation using continued fractions. *Theor. Comput. Sci.*, 392(1-3):158–173, 2008.

[36] A. J. H. Vincent. Sur la résolution des equations numériques. *Journal de Mathématiques Pures et Appliquées*, 1:341–372, 1836.

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

Michael Sagraloff
Max-Planck-Institut für Informatik
Campus E1 4
66123 Saarbrücken
Germany
e-mail: msagralo@mpi-inf.mpg.de

Algorithm 1 ISO-APX

Require: Real Polynomial $f \in \mathbb{R}[x]$ with roots contained in $\Delta_{1/8}(0)$ and a $\rho_0 \in \mathbb{N}$. All real roots of f are simple.

Ensure: Returns a $\rho \in \mathbb{N}$, a ρ-binary approximation $\tilde{f}$ of f, and a list $\mathscr{O}$ of isolating intervals $\bar{I}_1, \ldots, \bar{I}_m$ for the real roots of $\tilde{f}$ such that $\mathscr{T}^{\tilde{f}'}(m(I_j), 8nr(I_j))$ holds for all j

$\mathscr{A} := \{(-1/4, 1/4)\}$; $\mathscr{O} := \emptyset$; {list of active and isolating intervals}
$\rho := \rho_0$;
repeat
  $\rho := 2\rho$; {precision is doubled}
  $G := 0$; {guard is set neutral}
  $\tilde{f}$ a ρ-binary approximation of f;
  repeat
    $I := (a,b)$ some interval in $\mathscr{A}$ with center m and radius r; delete I from $\mathscr{A}$;
    if $\mathbf{P}_{\text{ex}}^{\text{ISO}}(\tilde{f}, I)$ or $\mathscr{T}_1(m,r)$ holds and $\tilde{f}(a) \cdot \tilde{f}(b) \neq 0$ **then**
      do nothing
    else if $\mathscr{T}'_{3/2}(m, 8nr)$ holds **then**
      if $\tilde{f}(a) \cdot \tilde{f}(b) \leq 0$ and I is not adjacent to any interval in $\mathscr{O}$ **then**
        add $\bar{I} = [a,b]$ to $\mathscr{O}$
      else
        do nothing
      end if
    else if $\Delta_{64n^3 r}(m)$ is terminal for $[\tilde{f}]_{2^{-\rho}}$ **then**
      $G := 1$; {guard "fires" because $2^{-\rho} > \mu(f)$}
    else
      subdivide I in accordance to the subdivision strategy of ISO and add all subintervals to $\mathscr{A}$
    end if
  until $\mathscr{A}$ is empty or $G = 1$
until $\mathscr{A}$ is empty
return $(\rho, \tilde{f}, \mathscr{O})$

Algorithm 2 ISO-EX

Require: A $\rho \in \mathbb{N}$, a ρ-binary approximation $\tilde{f}$ of f, and intervals $J_0,\ldots,J_m \subset (-1/4,1/4)$ with rational endpoints.

Ensure: either returns "insufficient precision" or the guarantee that $f(x) \neq 0$ for all $x \in \bigcup_i J_i$.

$\mathscr{A} := \{J_0,\ldots,J_m\}$; {list of active intervals}
repeat
  $I = (a,b)$ some interval in $\mathscr{A}$ with center m and radius $r =$; delete I from $\mathscr{A}$;
  if $\mathbf{P}^{\text{ISO}}_{ex}(\tilde{f}',I)$ or $\mathscr{T}'_1(m,r)$ holds **then**
    if $\min(|\tilde{f}(a)|,|\tilde{f}(b)|) > 2^{-\rho+1}$ **then**
      do nothing
    else
      return "insufficient precision"
    end if
  else if $\mathscr{T}_{3/2}(m,r)$ holds **then**
    if $\frac{1}{3}\tilde{f}(m) > 2^{-\rho+1}$ **then**
      do nothing
    else
      return "insufficient precision"
    end if
  else if $\Delta_{8n^2r}(m)$ is terminal for $[f]_{2^{-\rho}}$ **then**
    return "insufficient precision"
  else
    subdivide I in accordance to the subdivision strategy of ISO (with respect to $\tilde{f}'$); for each interval $I' = (a',b') \subset I$ that is discarded in this step, **return** "insufficient precision" **if** $\min(|f(a')|,|f(b')|) \leq 2^{-\rho+1}$; finally, add all remaining subintervals to $\mathscr{A}$
  end if
until $\mathscr{A}$ is empty
return $f(x) \neq 0$ for all $x \in \bigcup_i J_i$.

Algorithm 3 Subroutine ISO-APX$_{\mathbb{C}}$

Require: A bitstream polynomial $f \in \mathbb{R}[x]$ with roots $z_1, \ldots, z_m \in \Delta_{1/8}(0)$ and a $\rho \in \mathbb{N}$.

Ensure: either returns "insufficient precision" or a $\rho \in \mathbb{N}$, ρ-binary approximation $\tilde{f}$ of f and a list $\mathscr{O} = \{\Delta_1, \ldots, \Delta_n\}$ of isolating discs for the complex roots $z_1, \ldots, z_n$ of $\tilde{f}$ such that $\mathscr{T}'_{3/2}(m_j, 8nr_j)$ holds for all j, with m_j the center and r_j the radius of Δ_j

$\mathscr{A} := \{B_0\}$, B_0 a box with center 0 and size $1/2$; $\mathscr{O} := \emptyset$; {list of active boxes and isolating discs}
$\tilde{f}$ an arbitrary ρ-binary approximation of f;
repeat
  B some box in $\mathscr{A}$ with center m and of size s; delete B from $\mathscr{A}$;
  if $\mathbf{P}_{ex}^{\mathrm{ISO}_{\mathbb{C}}}(\tilde{f}, B)$ or $\mathscr{T}_1(m, 3s/4)$ holds **then**
    do nothing
  else if $\mathscr{T}'_{3/2}(m, 12n^2 s)$ holds and $\Delta_{3ns/2}(m)$ intersects no disc in $\mathscr{O}$ **then**
    add $\Delta_{3ns/2}(m)$ to $\mathscr{O}$
  else
    do nothing
  else if $\Delta_{96n^4 s}(m)$ is terminal for $[\tilde{f}]_{2^{-\rho}}$ **then**
    return "insufficient precision"
  else
    subdivide B due to the subdivision strategy of ISO$_{\mathbb{C}}$ and add all sub boxes to $\mathscr{A}$
  end if
until $\mathscr{A}$ is empty
return $(\rho, \tilde{f}, \mathscr{O})$