When Newton meets Descartes: A Simple and Fast Algorithm to Isolate the Real Roots of a Polynomial

Michael Sagraloff
Max-Planck-Institut für Informatik, Germany
msagralo@mpi-inf.mpg.de

ABSTRACT

We introduce a novel algorithm denoted NEWDSC to isolate the real roots of a univariate square-free polynomial f with integer coefficients. The algorithm iteratively subdivides an initial interval which is known to contain all real roots of f and performs *exact* operations on the coefficients of f in each step. For the subdivision strategy, we combine Descartes' Rule of Signs and Newton iteration. More precisely, instead of using a fixed subdivision strategy such as bisection in each iteration, a Newton step based on the number of sign variations for an actual interval is considered, and, only if the Newton step fails, we fall back to bisection. Following this approach, our analysis shows that, for most iterations, quadratic convergence towards the real roots is achieved. In terms of complexity, our method induces a recursion tree of almost optimal size $O(n \cdot \log(n\tau))$, where n denotes the degree of the polynomial and τ the bitsize of its coefficients. The latter bound constitutes an improvement by a factor of τ upon all existing subdivision methods for the task of isolating the real roots. In addition, we provide a bit complexity analysis showing that NEWDSC needs only $\tilde{O}(n^3\tau)$ bit operations[1] to isolate all real roots of f. This matches the best bound known for this fundamental problem. However, in comparison to the significantly more involved *numerical* algorithms by V. Pan and A. Schönhage, which achieve the same bit complexity for the task of isolating all complex roots, NEWDSC focuses on real root isolation, is much easier to access and to implement.

1. INTRODUCTION

Finding the roots of a univariate polynomial f is considered as one of the most important tasks in computational algebra. This is justified by the fact that many problems from mathematics, engineering, computer science, and the natural sciences can be reduced to solving a system of polynomial equations which in turn, by means of elimination techniques such as resultants or Gröbner Bases, reduces to solving a polynomial equation in one variable. Hence, it is not surprising that numerous approaches are dedicated to this fundamental problem. We mainly distinguish between (1) numerical and exact methods, and (2) methods to find all complex roots and methods which are especially tuned to search for real roots. The numerical literature lists many algorithms, such as Newton iteration or the Weierstrass-Durand-Kerner method, that are widely used and effective in practice but lack a guarantee on the global behavior (see [14] for a discussion). In particular, the global convergence and/or the complexity of the Weierstrass-Durand-Kerner method is still open.

The work of A. Schönhage [19] from 1982 marks the beginning of the complexity-theoretic approaches. It combines a newly introduced concept denoted *splitting circle method* with techniques from numerical analysis (Newton iteration, Graeffe's method, discrete Fourier transforms) and fast algorithms for polynomial and integer multiplication. For the benchmark problem of isolating all complex roots of a polynomial f of degree n with integer coefficients of modulus 2^τ or less, the proposed method achieves the record bound of $\tilde{O}(n^3\tau)$ bit operations. V. Pan and others [13, 14] gave theoretical improvements in the sense of achieving record bounds simultaneously in both bit complexity and arithmetic complexity, but the initial bound $\tilde{O}(n^3\tau)$ on the number of bit operations has still remained intact. Common to all asymptotically fast algorithms is (as the authors themselves admit) that they are rather involved and very difficult to implement. The latter is also due to the fact that, in order to control the precision errors in the considered numerical subroutines, one has to carefully work out many details of their implementation. Hence, it is not surprising that, despite their theoretical richness, these algorithms have so far not been used, or not proven to be efficient in practice; see [8] for an implementation of the splitting circle method within the Computer Algebra system Pari/GP. A further reason might be that the benchmark problem is inappropriate for most applications. For instance, in ray shooting in computer graphics, we are only interested in the first positive root or in the roots in some specified neighborhood.

In parallel to the development of purely numerical methods, there is a steady ongoing research on exact subdivision algorithms, such as[2] the Descartes method (e.g. [4, 6, 15]), the Bolzano method [18], Sturm Sequences [5], or the continued fraction method [2, 20]. These methods from the exact computation literature are widely used in various algebraic applications (e.g. cylindrical algebraic decomposition), and many of them have been integrated into computer algebra systems. In addition, their computational complexity has been well-studied [5, 7, 18, 20], and many experiments have shown their practical evidence [9, 15]. Current experimental data shows that a version of the classical Descartes method (i.e. the univariate solver in RS based on [15], integrated into MAPLE) which

[1] $\tilde{O}$ indicates that we omit logarithmic factors.

[2] The literature on root solving is extensive. Hence, due to space limitations, we decided to restrict to some recent and/or important papers and refer the reader to the references given therein.

uses approximate computation performs best for most polynomials, whereas, for harder instances (e.g. Mignotte polynomials), the continued fraction approach seems to be more efficient. With respect to the benchmark problem, all of the above mentioned exact algorithms demand for $\tilde{O}(n^4\tau^2)$ bit operations to isolate all real roots, hence they tend to lag behind the asymptotically fast algorithms by a factor of $n\tau$. Recent work [17] shows that the bound on the bit complexity for the Descartes method can be lowered to $\tilde{O}(n^3\tau^2)$ when replacing exact by approximate computation (without abstaining from correctness). This result partially explains the success of such a modified Descartes method in practice. However, as long as we restrict to the bisection strategy, the latter bound seems to be optimal. A. Schönhage already made a similar observation: In the introduction of [19], he argued that "a factor τ^2 inevitably occurs if nothing better than linear convergence is achieved".

In fact, the analysis of the classical Descartes method [4], which exclusively uses bisection in each iteration, shows that the induced recursion tree is large ($\approx n\tau$) if there exists a long sequences $I_1 \supset I_2 \supset \cdots \supset I_s$ of intervals in the subdivision process, where $v = v_{I_1} = \cdots = v_{I_s}$; see Section 2. Such a sequence implies the existence of a cluster $\mathscr{C}$ of v nearby roots, and vice versa (see Theorem 1). Hence, it seems reasonable to obtain a good approximation (i.e. a considerably smaller interval $I' \subset I$ close to $\mathscr{C}$) of such a cluster by considering a corresponding Newton step to approximate a v-fold root. Our novel algorithm, denoted NEWDSC, is exactly based on the latter idea. More precisely, in each iteration, we combine Descartes' Rule of Signs, Newton iteration and a subdivision technique similar to the one as proposed by J. Abbott for the quadratic interval refinement method [1] (QIR), to derive an interval I' as above and to check whether I' should be kept or not. In case of success (i.e. we can show that I' contains all roots within I), we proceed with I', whereas we fall back to bisection otherwise. Our analysis further shows that NEWDSC achieves quadratic convergence in most iterations. As a consequence, the induced recursion tree has almost optimal size $O(n\log(n\tau))$ which improves upon the bisection strategy by a factor of τ. We further provide a detailed bit complexity analysis which yields the bound $\tilde{O}(n^3\tau)$ for NEWDSC, thus matching the record bound achieved by the aforementioned asymptotically fast numerical algorithms. Combining NEWDSC with a recently presented approximate variant of the QIR method [10] for root refinement, we further improve the bit complexity for refining all isolating intervals to a width 2^{-L} or less from $\tilde{O}(n^3\tau^2 + n^2L)$ (as achieved by AQIR) to $\tilde{O}(n^3\tau + n^2L)$.

We consider our contribution of importance mainly because of the following two reasons: (1) The proposed algorithm is the first exact subdivision method which achieves the bound $\tilde{O}(n^3\tau)$ for the bit complexity of the fundamental problem of isolating the real roots of a polynomial. (2) In addition, it is much easier to access as well as to implement than the asymptotically fast numerical algorithms that are available so far. In comparison to the existing practical methods for real root isolation, the modifications are moderate, and thus we expect that a careful implementation of our new approach will outperform the existing ones.

2. OVERALL IDEA AND RESULTS

We first provide a high-level description of our algorithm and, then, outline the argument why our approach improves upon existing methods such as the classical Descartes method. For details, we refer to Section 3. Throughout the following considerations, let

$$f(x) := \sum_{i=0}^{n} a_i x^i \in \mathbb{Z}[x], \text{ with } |a_i| < 2^\tau, \tag{2.1}$$

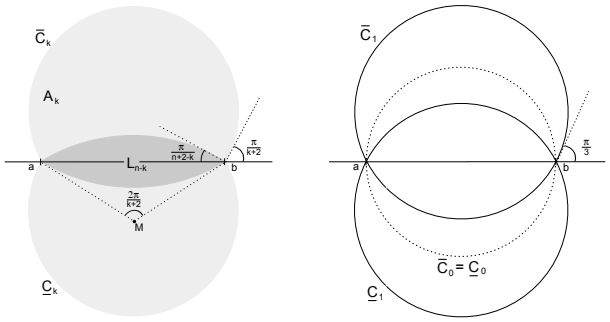

Figure 2.1: For any k, $0 \le k \le n$, the *Obreshkoff discs* $\overline{C}_k$ and $\underline{C}_k$ for $I = (a,b)$ have the endpoints of I on their boundaries; their centers see the line segment (a,b) under the angle $\frac{2\pi}{k+2}$. The *Obreshkoff lens* L_k is the interior of $\overline{C}_k \cap \underline{C}_k$, and the *Obreshkoff area* A_k is the interior of $\overline{C}_k \cup \underline{C}_k$. Any point (except a and b) on the boundary of A_k sees I under the angle $\frac{\pi}{k+2}$, and any point (except a and b) on the boundary of L_k sees I under the angle $\pi - \frac{\pi}{k+2}$. We have $L_n \subset \ldots \subset L_1 \subset L_0$ and $A_0 \subset A_1 \subset \ldots \subset A_n$. The cases $k = 0$ and $k = 1$ are of special interest: The circles $\overline{C}_0$ and $\underline{C}_0$ coincide. They have their centers at the midpoint of I. The circles $\overline{C}_1$ and $\underline{C}_1$ are the circumcircles of the two equilateral triangles having I as one of their edges. We call A_0 and A_1 the *one* and *two-circle regions* for I, respectively.

be a square-free polynomial of degree n with integer coefficients of bit-length τ or less. We further denote $z_1, \ldots, z_n$ the complex roots of f, $\sigma(z_i) := \min_{j \ne i} |z_i - z_j|$ the *separation of* z_i, and $\sigma_f := \min_i \sigma(z_i)$ the *separation of* f. According to Cauchy's root bound (e.g. [22]), the modulus of each root z_i is bounded by $1 + 2^\tau \le 2^{\tau+1}$, and thus, for the task of isolating the real roots of f, we can restrict our search to the initial interval $\mathscr{I}_0 := (-2^{\tau+1}, 2^{\tau+1})$.

We now consider an arbitrary root isolation method denoted ISO which recursively performs subdivision on $\mathscr{I}_0$ in order to determine isolating intervals for the real roots of f. ISO uses a counting function $\mathrm{var}(f,I)$ for m, *the number of roots within an interval* $I \subset \mathscr{I}_0$, where $v := \mathrm{var}(f,I) \in \mathbb{N}$ fulfills the following properties:

(P1) v is an upper bound for m (i.e. $v \ge m$), and

(P2) v has the same parity as m (i.e. $v \equiv m \mod 2$).

The latter two properties imply that $v = m$ if $v \le 1$. Hence, in each step of the recursion, an interval I is stored as isolating if $v = 1$, and I is discarded if $v = 0$. If $v > 1$, $I = (a,b)$ is subdivided (according to some subdivision strategy) into subintervals $I_1 = (a, \lambda_1)$, $I_2 = (\lambda_1, \lambda_2), \ldots, I_l := (\lambda_{l-1}, b)$, where $1 \le l \le l_0$, $l_0 \in \mathbb{N}$ is a global constant, and the λ_i are rational values. In order to detect roots at the subdivision points λ_i, we also check whether $f(\lambda_i) = 0$ or not. Throughout the following considerations, T_{ISO} denotes the recursion tree induced by ISO. The terms *nodes of* ISO and *intervals produced by* ISO are interchangeable.

The definition of $\mathrm{var}(f,I)$ is based on Descartes' Rule of Signs: *For an arbitrary polynomial* $p = \sum_{i=0}^{n} p_i x^i \in \mathbb{R}[x]$*, the number* m *of positive real roots of* p *is bounded by the number* v *of sign variations in its coefficient sequence* $(p_0, \ldots, p_n)$ *and, in addition,* $v \equiv m \mod 2$. In order to extend the latter rule to arbitrary intervals $I = (a,b)$, the Möbius transformation $x \mapsto \frac{ax+b}{x+1}$ which maps $(0, +\infty)$ one-to-one onto I is considered. Thus, for

$$f_I(x) = \sum_{i=0}^{n} c_i x^i := (x+1)^n \cdot f\left(\frac{ax+b}{x+1}\right), \tag{2.2}$$

and $\mathrm{var}(f,I)$ defined as *the number of sign variations in the coefficient sequence* $(c_0, \ldots, c_n)$ *of* f_I, $\mathrm{var}(f,I)$ fulfills the properties (P1) and (P2). Because of the latter two properties and the fact that

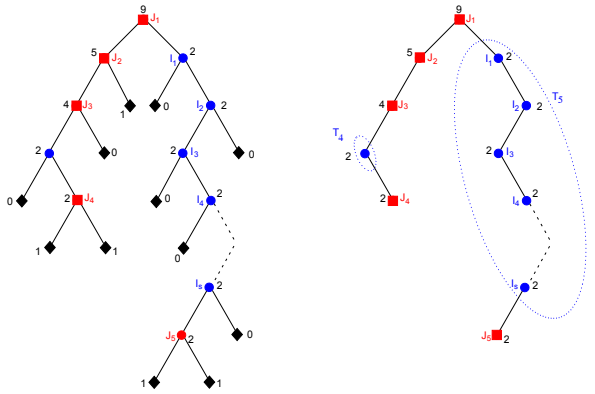

Figure 2.2: The left figure shows the subdivision tree T_{DSC} induced by the Descartes method, where, for each node I, the number $\mathrm{var}(f,I)$ of sign variations is given (e.g. $\mathrm{var}(f,J_1)=9$ or $\mathrm{var}(f,J_4)=2$). Milestones and terminal nodes are indicated by red squares and black diamonds, respectively. Blue dots indicate the ordinary nodes. The right figure shows the subtree T^*_{DSC} obtained by removing all terminal nodes. The ordinary nodes in T^*_{DSC} partition into connected components T_4 and T_5.

we never discard intervals that contain a real root of f, correctness of ISO follows immediately.

The classical Descartes method (DSC for short) due to Collins and Akritas [4] uses bisection in each iteration, that is, in each step, we have $l=l_0=2$, $I_1=(a,\lambda_1):=(a,m(I))$ and $I_2:=(\lambda_1,b)=(m(I),b)$, with $m(I):=(a+b)/2$ the midpoint of I. Termination and complexity analysis of DSC rest on the following result:

THEOREM 1 ([12, 11]). *Let $I=(a,b)$ be an open interval and $v=\mathrm{var}(f,I)$. If the Obreshkoff lens L_{n-k} (see Figure 2.1 for the definition of L_{n-k}) contains at least k roots (counted with multiplicity) of f, then $v\geq k$. If the Obreshkoff area A_k contains at most k roots (counted with multiplicity) of k, then $v\leq k$. In particular,*

(P3) *# of roots of f in $L_n\leq \mathrm{var}(p,I)\leq$ # of roots of f in A_n.*

We remark that the special cases $k=0$ and $k=1$ appear as the one- and two-circle theorems in the literature (e.g. [3, 6]). For the Descartes method, Theorem 1 implies that no interval I of length $w(I)\leq \sigma_f/2$ is split. Namely, in this case, the two-circle region A_1 for I cannot contain more than one root of f (which must be real). If I contains a real root, then $\mathrm{var}(f,I)=1$, otherwise $\mathrm{var}(f,I)=0$ by Theorem 1. We conclude that the depth of the recursion tree T_{DSC} induced by the Descartes method is bounded by $\log w(\mathscr{I}_0)+\log\sigma_f^{-1}+1=\tau+\log\sigma_f^{-1}+3$. Furthermore, it holds (e.g. see [6, Corollary 2.27] for a self-contained proof):

THEOREM 2. *Let I be an interval and I_1 and I_2 be two disjoint subintervals of I. Then,*

(P4) $\mathrm{var}(f,I_1)+\mathrm{var}(f,I_2)\leq \mathrm{var}(f,I)$.

According to Theorem 2, there cannot be more than $n/2$ intervals I with $\mathrm{var}(f,I)\geq 2$ at any level of the recursion. Hence, the size of T_{DSC} is bounded by $n(\tau+\log\sigma_f^{-1}+3)$. Using Davenport-Mahler bound, one can further show [6, 17] that $\log\sigma_f^{-1}=O(n(\log n+\tau))$, and thus the bound for $|T_{\mathrm{DSC}}|$ writes as $\tilde{O}(n^2\tau)$. A more refined argument [6, 7] yields $|T_{\mathrm{DSC}}|=\tilde{O}(n\tau)$ which is optimal for the bisection strategy.

In the next step, we study the situation where the recursion tree T_{DSC} for DSC is large (i.e. $|T_{\mathrm{DSC}}|\approx n\tau$). We then outline how to address this situation via combining Newton iteration and Descartes' Rule of Signs, and we sketch the argument why our approach improves upon DSC. The following definition is essential for the argument; see also Figure 2.2:

DEFINITION 1. *Let T_{ISO} denote the recursion tree induced by some subdivision algorithm* ISO. *A node (interval) $I\in T_{\mathrm{ISO}}$ is called* terminal *if $\mathrm{var}(f,I)\leq 1$. A non-terminal node I with children $I_1,\ldots,I_l$ is called a* milestone *if either $I=\mathscr{I}_0$ (i.e. I is the root of T_{ISO}) or*

$$\mathrm{var}(f,I_j)<\mathrm{var}(f,I)\ \text{for all } j=1,\ldots,l.$$

A non-terminal node which is not a milestone is called ordinary.

Due to (P4) in Theorem 2, we have $\sum_{j=1}^{l}\mathrm{var}(f,I_j)\leq\mathrm{var}(f,I)$. Thus, a non-terminal node $I\neq\mathscr{I}_0$ is ordinary if and only if, for one of its children, we count the same number of sign variations as for I (and thus no sign variation for all other children). The *number n' of milestones* is bounded by $\mathrm{var}(f,\mathscr{I}_0)\leq n$. Namely, when subdividing a milestone which is not the root $\mathscr{I}_0$, the non-negative value $\mu:=\sum_I\mathrm{var}(f,I)-\#\{I:\mathrm{var}(f,I)>0\}$ decreases by at least one, where we sum over all leafs in the actual iteration, and we initially start with $\mu=\mathrm{var}(f,\mathscr{I}_0)-1$. We denote the milestones by $J_1,\ldots,J_{n'}$ and assume, w.l.o.g., that $w(J_i)\geq w(J_k)$ if $i<k$. In particular, we have $J_1=\mathscr{I}_0$. We further define T^*_{ISO} the subtree of T_{ISO} obtained from T_{ISO} via removing all terminal nodes. T^*_{ISO} partitions into

(1) the milestones $J_1,\ldots,J_{n'}$ (red squares in Figure 2.2), and
(2) subtrees $T_i\subset T^*_{\mathrm{ISO}}$, with $i=2,\ldots,n'$, consisting of ordinary nodes $I\in T^*_{\mathrm{ISO}}$, with $J_i\subset I$ and $J_k\not\subset I$ for all milestones J_k with $J_k\supsetneq J_i$ (blue dots).

From our definition of a milestone, each T_i constitutes a chain of ordinary intervals $I_1\supset\cdots\supset I_s$ that connects two milestones. More precisely, T_i connects J_i with J_k, where J_k is a milestone of minimal width that contains J_i. Since each interval has at most l_0 children, $|T_{\mathrm{ISO}}|$ is bounded by $(l_0+1)\cdot|T^*_{\mathrm{ISO}}|=O(|T^*_{\mathrm{ISO}}|)$. Hence,

$$O(|T_{\mathrm{ISO}}|)=O(n'+\sum_{i=2}^{n'}|T_i|)=O(n)+O(\sum_{i=2}^{n'}|T_i|). \tag{2.3}$$

The latter consideration shows that the size of the subdivision tree crucially depends on the length of the chains T_i. For the classical Descartes method, it might happen that some of these chains are very large (i.e. $|T_i|\approx n\tau$) which is due to the following situation (see also Figure 2.3): For a polynomial f as in (2.1), it is possible that there exists a $\xi\in\mathbb{R}$ and a small complex neighborhood (of size $\varepsilon\approx 2^{-n\tau}$) of ξ that contains a cluster $\mathscr{C}$ of v nearby roots of f (e.g. when f is a Mignotte polynomial). Thus, separating these roots from each other via bisection requires at least $\log\varepsilon^{-1}\approx n\tau$ steps. Furthermore, due to (P3) in Theorem 1, there exists a long sequence $I_1\supset I_2\supset\cdots\supset I_s$ of intervals with $\xi\in I_j$ for all j, and thus the number

$$v:=\mathrm{var}(f,I_1)=\mathrm{var}(f,I_2)=\cdots=\mathrm{var}(f,I_s)$$

of sign variations does not change for the intervals in this sequence. Namely, for each I_j in the above sequence, the Obreshkoff lens L_n contains $\mathscr{C}$. Vice versa, according to Theorem 1, such a long sequence of non-special intervals implies the existence of a cluster $\mathscr{C}$ consisting of v nearby roots as above because the Obreshkoff area A_n of each I_j must contain at least v roots.[3] Since a cluster $\mathscr{C}$ of

[3]The thoughtful reader may notice that the latter two statements are not completely rigorous: If ξ, and thus also the cluster $\mathscr{C}$, is very close to one of the endpoints of some I_j, some of the roots might not be considered by the counting function $\mathrm{var}(f,I_j)$ since they are located outside the Obreshkoff lens/area. We will address this issue in our algorithm as defined in Section 3.

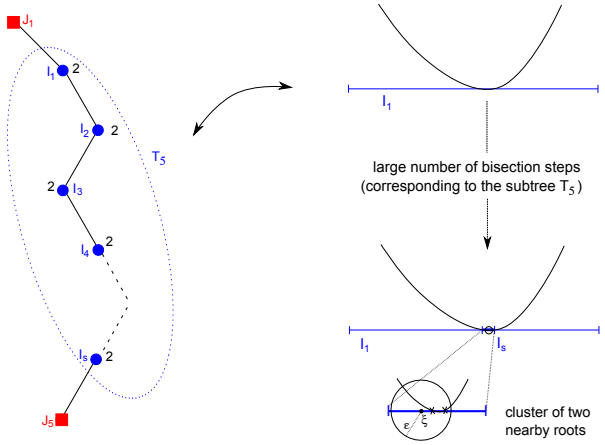

Figure 2.3: The long chain $I_1 \supset I_2 \supset \cdots \supset I_s$ of intervals in T_5 with $\mathrm{var}(f,I_1) = \cdots = \mathrm{var}(f,I_s) = 2$ corresponds to a large number of bisection steps to isolate two very nearby roots from each other; see the figure on the right with the graph of f over I_1.

v nearby roots at ξ behaves similar to a v-fold root at ξ, it seems reasonable to obtain a good approximation of $\mathscr{C}$ by considering Newton iteration instead of bisection. Namely, for a polynomial $p(x) \in \mathbb{R}[x]$ with a v-fold root at ξ and a starting value x_0 sufficiently close to ξ, it is well-known from numerical analysis that the sequence $(x_i)_{i\in\mathbb{N}_0}$ recursively defined by

$$x_{i+1} := x_i - v \cdot \frac{p(x_i)}{p'(x_i)}$$

converges quadratically to ξ. Unfortunately, when isolating the roots of f, the situation differs considerably from the latter one: First, the above result only holds for a v-fold root ξ and does not directly extend to a cluster $\mathscr{C}$ of v roots near ξ. Second, in an early stage of the subdivision process, the existence of such a cluster $\mathscr{C}$ is not guaranteed, and even if one exists, we do not know what "sufficiently close to ξ" means in this situation. In order to address the above mentioned problems and to finally turn the purely numerical Newton method into an exact and complete algorithm, we propose the following approach: Let $v = \mathrm{var}(f,I)$ be the number of sign variation for an actual interval $I = (a,b)$ in a certain iteration. Hence, there might exist a cluster of v nearby roots. In order to check whether this is actually the case, we compute $\lambda := t - v\cdot f(t)/f'(t)$ for some $t \in [a,b]$ (e.g. an endpoint of I) and consider a subinterval $I' = (a',b') \subset I$ of width $w(I') \ll w(I)$ that contains λ. If $\mathrm{var}(f,I') = v$, we keep I' and discard the intervals $(a,a']$ and $[b',b)$. Otherwise, we split I into $I_1 := (a,m(I))$ and $I_2 := (m(I),b)$ and finally check whether $f(m(I)) = 0$ or not. Following this approach, no root is lost and intervals are at least bisected in each iteration. Furthermore, if a cluster $\mathscr{C}$ of nearby roots actually exists, we can hope to achieve fast convergence to this cluster when choosing I' in an appropriate manner. In our algorithm, we choose I' in a similar way as proposed by J. Abbott [1] for the task of further refining isolating intervals. Namely, we decompose I into a certain number N_I, $N_I \geq 4$, of subintervals and pick the subinterval I' of size $w(I)/N_I$ which contains λ. If $\mathrm{var}(f,I') = v$, we keep I' and decompose I' into $N_{I'} = N_I^2$ subintervals in the next iteration. Otherwise, we continue with the intervals $I_1 = (a,m(I))$ and $I_2 = (m(I),b)$ which are now decomposed into only $N_{I_1} = N_{I_2} := \max(4,\sqrt{N_I})$ many subintervals, etc.

In the next section, we give the exact definition of our new algorithm denoted NEWDSC, and we show that it induces a subdivision tree T_{NEWDSC} of considerably smaller size than T_{DSC}. In particular, the size of each $T_i \subset T_{\mathrm{NEWDSC}}$ is bounded by $O(\log(n\tau))$ which is due to the fact that, for most iterations, we have quadratic convergence, and the width of each interval is lower bounded by $2^{-\tilde{O}(n\tau)}$; see Lemma 3 and Theorem 6 for proofs. Hence, according to (2.3), the size of the overall recursion tree is bounded by

$$O(n'(\log n\tau)) = O(\mathrm{var}(f,\mathscr{I}_0)\cdot(\log n\tau)) = O(n(\log n\tau)). \quad (2.4)$$

The latter result particularly shows that the size of the recursion tree is directly correlated to *the number n^* of non-zero coefficients of f* because instead of considering $\mathscr{I}_0 = (-2^{\tau+1},2^{\tau+1})$, we can start with $(-2^{\tau+1},0)$ and $(0,2^{\tau+1})$, and thus the total number of sign variations counted for both intervals is upper bounded by $2\cdot n^*$.

For the bit complexity of our algorithm, we have to consider the cost for computing the polynomials $f_I(x)$ as defined in (2.2), where $I = (a,b)$ is an interval to be processed. In Section 3.3, we will show that the computation of $f(x+a)$ constitutes the most costly step. For $I \in T_i$, the endpoints of I are dyadic numbers of bitsize $O(\tau + \log w(J_i)^{-1})$ or less, and thus the computation of f_I demands for $\tilde{O}(n^2(\tau+\log w(J_i)^{-1}))$ bit operations, where we assume asymptotically fast Taylor shift [21]. Lemma 8 shows that, for all $i = 1,\ldots,n'$, $\log w(J_i)^{-1} > \sigma(z_{n-(n'-i)})^{-1} + O(\log^2 n)$ if the roots are ordered with respect to separation (i.e. $\sigma(z_1) \geq \cdots \geq \sigma(z_n)$). Hence, computing f_I demands for $\tilde{O}(n^2(\tau+\log\sigma(z_{n-(n'-i)})^{-1} + \log^2 n))$ bit operations. For the total cost, we thus obtain the bound

$$\tilde{O}(n^3\tau + n^2\sum_{i=1}^{n'}\log\sigma(z_{n-(n'-i)})^{-1}) = \tilde{O}(n^3\tau)$$

since the number of special nodes is bounded by n, the length of each T_i is bounded by $\log(n\tau)$, and $\sum_{i=1}^{n'}\log\sigma(z_i)^{-1} = \tilde{O}(n\tau)$ (e.g. see in [17, Lemma 19], or [18] for the latter bound).

3. ALGORITHM AND ANALYSIS

3.1 The Algorithm

We first present the algorithm. For pseudo-code, see Appendix 6.1.

NEWDSC maintains a list $\mathscr{A}$ of active intervals I with corresponding integers $N_I = 2^{2^{n_I}}$, $n_I \in \mathbb{N}_{\geq 1}$, and a list $\mathscr{O}$ of isolating intervals, where, initially, $\mathscr{A} := \{(\mathscr{I}_0, N_{\mathscr{I}_0})\} := \{((-2^{\tau+1},2^{\tau+1}),4)\}$ and $\mathscr{O} := \emptyset$. For $(I,N_I) \in \mathscr{A}$, $I = (a,b)$, we proceed as follows:
We remove I from $\mathscr{A}$ and compute $v := \mathrm{var}(f,I)$.

1. If $v = 0$, we do nothing (i.e. I is discarded).
2. If $v = 1$, then I isolates a real root of f. Thus, we add I to the list $\mathscr{O}$ of isolating intervals.
3. For $v > 1$, we proceed as follows: Let

$$B_1 := (a, a + \frac{w(I)}{N_I}) \text{ and } B_2 := (b - \frac{w(I)}{N_I}, b) \quad (3.1)$$

be the left- and rightmost interval of size $w(I)/N_I$ contained in I, respectively. We compute $v_i := \mathrm{var}(f,B_i)$ for $i = 1,2$:

   (a) If one of the values v_i equals v, then the (unique) corresponding interval B_i contains all roots of f within I. Hence, we keep $I' := B_i$ and set $N_{I'} := N_I^2$. That is, $(I',N_{I'}) := (B_i,N_I^2)$ is added to $\mathscr{A}$.

   (b) If both values v_1 and v_2 differ from v, we compute

$$\lambda_1 := a - v\cdot\frac{f(a)}{f'(a)} \text{ and } \lambda_2 := b - v\cdot\frac{f(b)}{f'(b)}. \quad (3.2)$$

If there exists a cluster $\mathscr{C}$ of v nearby roots and a (or b) has "reasonable" distance to $\mathscr{C}$, then λ_1 (or λ_2) constitutes a considerably better approximation of $\mathscr{C}$ than a (or b). In order to check whether this is actually the case, we consider the $4N_I-3$ grid points $a+k\cdot\frac{w(I)}{4N_I}$ in I, with $k\in\{2,\dots,4N_I-2\}$ and choose one which is "close" to λ_i. More precisely, for $i=1,2$, we define

$$k_i := \min(\max(\lfloor 4N_I(\lambda_i-a)\rfloor,2),4N_I-2) \qquad (3.3)$$

and I_i' to be the interval of length $w(I)/N_I$ centered at the point $a+k_i\cdot w(I)/4N_I$, that is,

$$I_i' := (a+(k_i-2)\frac{w(I)}{4N_I}, a+(k_i+2)\frac{w(I)}{4N_I}) \subset I. \quad (3.4)$$

We remark that it is crucial for our approach that I_i' contains λ_i and λ_i has distance at least $w(I)/4N_I$ to both endpoints of I_i', given that $a+w(I)/4N_I \le \lambda_i \le b-w(I)/4N_I$. In the next step, we compute $v_i' := \mathrm{var}(f,I_i')$ for $i=1,2$. If one of the two values v_i' equals v, we keep the corresponding interval $I' := I_i'$ with $v_i'=v$ (if we count v sign variations for I_1' as well as I_2', we just keep I_1') and add $(I',N_{I'}) := (I',N_I^2)$ to $\mathscr{A}$.

(c) If all values v_1, v_2, v_1' and v_2' differ from v, we consider this an indicator that there is either no cluster of v nearby roots or that such a cluster is not separated well enough from the remaining roots. Hence, in this situation, we fall back to bisection. That is, we split I into the intervals $I_1=(a,m(I))$ and $I_2=(m(I),b)$ and add $(I_1,\max(4,\sqrt{N_I}))$ and $(I_2,\max(4,\sqrt{N_I}))$ to $\mathscr{A}$. Finally, if $f(m(I))=0$, we also add $[m(I),m(I)]$ to $\mathscr{O}$.

Correctness of NEWDSC follows immediately from the fact that our starting interval $\mathscr{I}_0$ contains all real roots of f and we never discard intervals (or endpoints) that contain a root of f. NEWDSC terminates because an interval I is at least bisected in each iteration, and thus eventually $\mathrm{var}(f,I)\le 1$. In addition, we have:

LEMMA 3. *For each interval I produced by* NEWDSC*, we have*

$$2^{\tau+2} \ge w(I) \ge \frac{\sigma_f^3}{2^{2(\tau+4)}} = 2^{-\tilde{O}(n\tau)} \text{ and } 4 \le N_I \le \frac{2^{2(\tau+3)}}{\sigma_f^2} = 2^{\tilde{O}(n\tau)}.$$

In particular, for each interval I in the subtree $T_i \subset T_{\text{NEWDSC}}$ (see Section 2 for the definition), it holds that

$$2^{\tau+2} \ge w(I) \ge w(J_i) \text{ and } 4 \le N_I \le 2^{2(\tau+4)}\cdot w(J_i)^{-2},$$

with J_i the milestone corresponding to T_i.

PROOF. The inequalities $2^{\tau+2}=w(\mathscr{I}_0)\ge w(I)$ and $N_I\ge 4$ are trivial. For $N_I>4$, there must exist an interval J with $J\supset I$ and $N_J=\sqrt{N_I}$, and J was replaced by an interval $J'\supseteq I$ of size $w(J)/N_J$. Since J is non-terminal, J' is also non-terminal because $\mathrm{var}(f,J')=\mathrm{var}(f,J)>1$. Thus, $\sigma_f\le 2w(J')=2w(J)/N_J\le 2^{\tau+3}/\sqrt{N_I}$. This shows the upper bound for N_I. For the lower bound for $w(I)$, we consider the parent interval J of I. Since J is non-terminal, we have $2w(J)\ge\sigma_f$, and thus $w(I)\ge w(J)/N_J\ge(\sigma_f/2)\cdot\sigma_f^2\cdot 2^{-2(\tau+3)}$. For $I\in T_i$, the bounds for $w(I)$ are trivial, and, in completely similar manner as above, we conclude that $2^{\tau+2}/\sqrt{N_I}\ge w(J_i)$. □

Throughout the following considerations, we call a subdivision step at I *quadratic* if I is replaced by an interval I' of width $w(I')=w(I)/N_I$ and *linear* if I is split into two equally sized intervals I_1 and I_2. In a quadratic step, the integer N_I is squared whereas, in a linear step, $N_{I'}:=\max(4,\sqrt{N_I})$ for each subinterval $I'=I_{1/2}$.

3.2 Analysis of the Recursion Tree

In this section, we show that the size of each subtree $T_i\subset T_{\text{NEWDSC}}$ as defined in Section 2 is bounded by $O(\log(n\tau))$. We start with the following two technical lemmata whose proofs are given in Appendix 6.2. For Lemma 5, see also Figures 2.1 and 3.1.

LEMMA 4. *Let w, $w'\in\mathbb{R}^+$ be two positive reals with $w>w'$, and let $m\in\mathbb{N}_{\ge 1}$ be a positive integer. We further define the sequence $(s_i)_{i\in\mathbb{N}_{\ge1}} := ((x_i,n_i))_{i\in\mathbb{N}_{\ge1}}$ as follows: $s_1:=(w,m)$, and*

$$s_i=(x_i,n_i) := \begin{cases} \left(\frac{x_{i-1}}{N_{i-1}}, n_{i-1}+1\right), & \text{if } \frac{x_{i-1}}{N_{i-1}}\ge w' \\ \left(\frac{x_{i-1}}{2}, \max(1,n_{i-1}-1)\right), & \text{if } \frac{x_{i-1}}{N_{i-1}}<w', \end{cases}$$

where $N_i:=2^{2^{n_i}}$ and $i\ge 2$. Then, the smallest index i_0 with $x_{i_0}\le w'$ is upper bounded by $8(n_1+\log\log\max(4,\frac{w}{w'}))$.

LEMMA 5. *Let $I=(a,b)$ be an arbitrary interval, A_n the corresponding Obreshkoff area and L_n the Obreshkoff lens for I.*

(1) For $I'=(a',b')\subset I$ with $a\ne a'$ and $b\ne b'$, the Obreshkoff area A_n' for I' is completely contained within the lens L_n for I if

$$\min(|a-a'|,|b-b'|) > 8n^2 w(I').$$

If the latter inequality if fulfilled, then, for all $x\notin L_n$ and all $\xi\in A_n'$,

$$|x-\xi| > \frac{1}{4n}\cdot\left(\min(|a-a'|,|b-b'|)-8n^2w(I')\right)$$

(2) For $I'=(a',b')$ with $I'\cap I=\emptyset$, the Obreshkoff area A_n' for I' does not intersect A_n if $\mathrm{dist}(I,I')>4n^2\cdot\min(w(I),w(I'))$, where $\mathrm{dist}(I,I')$ denotes the distance between the intervals I and I'.

In Section 2, we already argued that each T_i constitutes a chain of intervals $I_1=(a_1,b_1)\supset I_2=(a_2,b_2)\supset\cdots\supset I_s=(a_s,b_s)$ "connecting" the milestone J_i with the milestone J_k of minimal width that contains J_i. In the proof of the following theorem, we will show that, for all but $O(\log(n\tau))$ many j, the sequence $(w(I_j),n_{I_j})=(w(I_j),\log\log N_{I_j})$ behaves similarly to the sequence (x_j,n_j) as defined in Lemma 4. This results in the following bound for $|T_i|$:

THEOREM 6. *Each subtree $T_i\subset T_{\text{NEWDSC}}$ has size $O(\log(n\tau))$.*

PROOF. We first consider the case where $a_1=a_2=\cdots=a_s$, that is, in each subdivision step, the leftmost interval has been chosen. Since $v=\mathrm{var}(I_1)=\cdots=\mathrm{var}(I_s)$, Theorem 2 implies that $\mathrm{var}(f,I)=v$ for each interval I with $I_s\subset I\subset I_1$. In particular, if $w(I_j)/N_{I_j}\ge w(I_s)$, we count v sign variations for the interval $B_1=(a_j,a_j+w(I_j)/N_{I_j})=(a_s,a_s+w(I_j)/N_{I_j})$ as defined in (3.1). It follows that the subdivision step at I_j is quadratic, and thus, for $j=1,\dots,s-1$, the sequence $(w(I_j),n_{I_j})$ coincides with the sequence (x_j,n_j) as defined in Lemma 4, where $w:=w(I_1)$, $w':=w(I_s)$ and $n_1=m:=n_{I_1}$. Namely, if $w(I_j)/N_{I_j}\ge w'$, we have $w(I_{j+1})=w(I_j)/N_{I_j}$ and $n_{I_{j+1}}=1+n_{I_j}$, and, otherwise, we have $w(I_{j+1})=w(I_j)/2$ and $n_{I_{j+1}}=\max(1,n_{I_j}-1)$. Hence, according to Lemma 3 and Lemma 4, it follows that s is bounded by

$$8(n_{I_1}+\log\log\max(4,w(I_1)/w(I_s)))=O(\log(n\tau)).$$

An analogous argument shows the same bound for s in the case where $b_1=b_2=\cdots=b_s$. We now turn to the more general case, where $a_1\ne a_s$ and $b_1\ne b_s$: Let $s_1\in\{2,\dots,s\}$ be the smallest index with $a_{s_1}\ne a_1$ and $b_{s_1}\ne b_1$. Then, the above argument shows that s_1 is bounded by $O(\log(n\tau))$. Since $\min(|a_1-a_{s_1}|,|b_1-b_{s_1}|)\ge w(I_{s_1})/4$, we have $\min(|a_1-a_j|,|b_1-b_j|)\ge 2^{j-s_1-2}w(I_j)$ for all $j\ge s_1$, and thus

$$\frac{\left(\min(|a_1-a_j|,|b_1-b_j|)-8n^2w(I_j)\right)}{4n} \ge w(I_j)\left(\frac{2^{j-s_1-4}}{n}-2n\right).$$

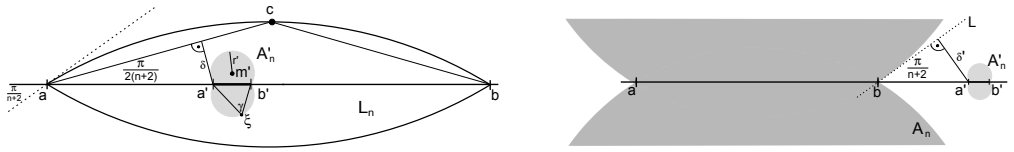

Figure 3.1: On the left figure, c denotes the topmost point of the Obreshkoff lens L_n for $I=(a,b)$. If $|a-a'|\le w(I)/2$, then the distance from a' to the boundary of L_n is bounded by the distance δ from a' to $\overline{ac}$. The radius r' of the Obreshkoff discs $\underline{C}'_n$ and $\overline{C}'_n$ for $I'=(a',b')$ is bounded by $n\cdot w(I')$ due to the extended Sine Theorem. The right figure shows the Obreshkoff areas for the intervals I and I', respectively.

Hence, with $s_2 := s_1+\lceil\log(16n^3)\rceil+4 = O(\log(n\tau))$, we have

$$\frac{\big(\min(|a_1-a_j|,|b_1-b_j|)-8n^2w(I_j)\big)}{4n} \ge 8n^2w(I_j) \text{ for all } j\ge s_2,$$

or $s<s_2$ in which case we are done. Then, from Theorem 1 and Lemma 5, we conclude that, for $j\ge s_2$, the Obreshkoff area for I_j (denoted $A_n^{(j)}$) contains exactly v roots $z_1,\dots,z_v$ of f because the Obreshkoff lens $L_n^{(1)}$ for I_1 contains at most v roots, $A_n^{(j)}$ contains at least v roots, and $A_n^{(j)}\subset L_n^{(1)}$. In particular, the Obreshkoff area $A_n^{(s)}$ for I_s must contain $z_1,\dots,z_v$. In the proof of Lemma 5, we already argued that each point in $A_n^{(s)}$ has distance less than $2nw(I_s)$ from any point within I_s, and thus

$$|x-z_i|<2nw(I_s), \text{ for all } i=1,\dots,v \text{ and all } x\in I_s. \tag{3.5}$$

The remaining roots $z_{v+1},\dots,z_n$ of f are located outside the Obreshkoff lens $L_n^{(1)}$ for I_1, and thus their distance to an arbitrary point within I_j is larger than $8n^2w(I_{s_2})$. Namely, according to Lemma 5, the distance from any of the roots $z_{v+1},\dots,z_n$ to an arbitrary point within I_{s_2} is lower bounded by $8n^2w(I_{s_2})$ and I_{s_2} contains I_j. The following consideration further shows that the existence of an $s_3 = s_2+O(\log n\tau)$ such that $w(I_j)\le w(I_{s_2})/N_{I_j}$ for all $j\ge s_3$ (or that $s<s_3$ in which case we are again done), and thus

$$|x-z_i|>8n^2N_{I_j}w(I_j) \text{ for all } i\ge v+1,\ j\ge s_3, \text{ and } x\in I_j: \tag{3.6}$$

Due to Lemma 3, we have $N_{I_j}\le N_{\max} := \lceil 2^{2(\tau+2)}/\sigma_f\rceil = 2^{\tilde O(n\tau)}$ for all j. Hence, if the sequence I_{s_2}, $I_{s_2+1},\dots$ starts with more than $m_{\max} := \log\log N_{\max}+1 = O(\log(n\tau))$ consecutive linear subdivision steps, then $N_{I_{j'}}=4$ and $w(I_{j'})\le w(I_{s_2})/4 = w(I_{s_2})/N_{I_{j'}}$ for some $j'\le s_2+m_{\max}$. Otherwise, there exists a j' with $s_2\le j'\le s_2+m_{\max}$ such that the subdivision step at $I_{j'}$ is quadratic. Since the length of a sequence of consecutive quadratic subdivision steps is also bounded by $m_{\max}$, there must exist a j'' with $j'+1\le j''\le j'+m_{\max}+1$ such that the step at $I_{j''-1}$ is quadratic, whereas the step at $I_{j''}$ is linear. Then, $N_{I_{j''+1}}=\sqrt{N_{I_{j''}}}=N_{I_{j''-1}}$, and $w(I_{j''+1})=w(I_{j''})/2=w(I_{j''-1})/(2N_{I_{j''-1}})<w(I_{s_2})/N_{I_{j''+1}}$. Hence, in both cases, we have shown that there exists an $s_3\le s_2+2m_{\max}+1 = O(\log(n\tau))$ with $w(I_{s_3})\le w(I_{s_2})/N_{I_{s_3}}$. Then, by induction, it follows that $w(I_j)\le w(I_{s_2})/N_{I_j}$ for all $j\ge s_3$ which shows (3.6).

We are now ready to prove that the subdivision step at I_j is quadratic if $j\ge s_3$ and $w(I_j)\ge 68nN_{I_j}w(I_s)$: Namely, if the latter two inequalities hold, then one of the endpoints of I_j (w.l.o.g. assume a_j) has distance at least $w(I_j)/2\ge 34nN_{I_j}w(I_s)$ from a_s. Thus, the distance from a_j to any of the roots $z_1,\dots,z_v$ is larger than $34nN_{I_j}w(I_s)-2nw(I_s)\ge 32nN_{I_j}w(I_s)$. In addition, it holds that $|a_j-z_i|>8n^2N_{I_j}w(I_j)$ for all $i>v$ according to (3.6). Thus,

$$\left|\frac{1}{v}\cdot\frac{(a_j-a_s)f'(a_j)}{f(a_j)}-1\right| = \left|\frac{1}{v}\sum_{i=1}^{v}\frac{a_j-a_s}{a_j-z_i}+\frac{1}{v}\sum_{i>v}\frac{a_j-a_s}{a_j-z_i}-1\right|$$
$$= \frac{1}{v}\left|\sum_{i=1}^{v}\frac{z_i-a_s}{a_j-z_i}+\sum_{i>v}\frac{a_j-a_s}{a_j-z_i}\right| \le \frac{1}{v}\sum_{i=1}^{v}\frac{|z_i-a_s|}{|a_j-z_i|}+\frac{1}{v}\sum_{i>v}\frac{|a_j-a_s|}{|a_j-z_i|}$$
$$< \frac{2nw(I_s)}{32nN_{I_j}w(I_s)}+(n-v)\cdot\frac{w(I_j)}{8n^2N_{I_j}w(I_j)} \le \frac{1}{8N_{I_j}},$$

where we used that $f'(a)/f(a)=\sum_{i=1}^n(a-z_i)^{-1}$ for all $a\in\mathbb{C}$ with $f(a)\ne 0$. This yields the existence of an $\varepsilon\in\mathbb{R}$ with $|\varepsilon|<1/(8N_j)\le 1/32$ and $\frac{1}{v}\cdot\frac{(a_j-a_s)f'(a_j)}{f(a_j)}=1+\varepsilon$. We can now derive the following bound on the distance between the approximation obtained by the Newton iteration and a_s:

$$\left|a_s-(a_j-v\cdot\frac{f(a_j)}{f'(a_j)})\right| = |a_s-a_j|\cdot\left|1-\frac{1}{\frac{1}{v}\cdot\frac{(a_j-a_s)f'(a_j)}{f(a_j)}}\right|$$
$$= |a_s-a_j|\cdot\left|1-\frac{1}{1+\varepsilon}\right| = \left|\frac{\varepsilon\cdot(a_s-a_j)}{1+\varepsilon}\right| < \frac{w(I_j)}{7N_{I_j}}, \tag{3.7}$$

where the latter inequality follows from $|a_s-a_j|\le w(I_j)$ and $|\varepsilon|<1/(8N_j)$. If $a_s\ge b_j-w(I_j)/N_{I_j}$, then the interval (a_s,b_s) is contained in $B_2=(b_j-w(I_j)/N_{I_j},b_j)$, and thus $\mathrm{var}(f,B_2)=v$. Hence, in Step 3 (a) of the algorithm, we keep the interval $I_{j+1}=B_2$ which has width $w(I_j)/N_{I_j}$. If $a_s<b_j-w(I_j)/N_{I_j}$, then it holds that

$$a_j-v\frac{f(a_j)}{f'(a_j)}\in(a_s-\frac{w(I_j)}{7N_{I_j}},a_s+\frac{w(I_j)}{7N_{I_j}})\subset(a_j+\frac{w(I_j)}{4N_{I_j}},b_j-\frac{w(I_j)}{4N_{I_j}})$$

according to (3.7) and $|a_j-a_s|\ge w(I_j)/2$. It follows that the interval I'_1 as defined in Step 3 (b) of the algorithm contains (a_s,b_s), and thus $\mathrm{var}(f,I'_1)=v$. Hence, the subdivision step at I_j is quadratic.

We now consider the sequence $(w(I_{s_3+i}),n_{I_{s_3+i}})_{i\le i^*}$, where i^* is defined as the largest index with $w(I_{s_3+i^*})\ge 68nw(I_s)$. The above argument implies that the sequence $(w(I_{s_3+i}),n_{I_{s_3+i}})_{1\le i\le i^*-1}$ coincides with the sequence $(x_i,n_i)_{1\le i\le i^*-1}$ as defined in Lemma 4, where $n_1=n_{I_{s_3+1}}$ and $w' := 68nw(I_s)$. Namely, if $w(I_{s_3+i})/N_{I_{s_3+i}}\ge w'$, then $68nw(I_s)\le w(I_{s_3+i+1})=w(I_{s_3+i})/N_{s_3+i}$ and $n_{I_{s_3+i+1}}=1+n_{I_{s_3+i}}$, whereas, for $w(I_{s_3+i})/N_{I_{s_3+i}}<w'$, it holds that $w(I_{s_3+i+1})=w(I_{s_3+i})/2$ and $n_{I_{s_3+i+1}}=\max(n_{I_{s_3+i}}-1,1)$. It follows that i^* is bounded by $8(n_1+\log\log\max(4,w(I_{s_3+1})/w'))=O(\log(n\tau))$. It follows that there exists an $s_4=s_3+i^*+1=O(\log(n\tau))$ with $w(I_j)<68nw(I_s)$ for all $j\ge s_4$. Since intervals are at least bisected in each iteration, this shows that s, and thus the size of T_i is bounded by $s_4+\log(68n)=O(\log(n\tau))$. □

Combining the latter theorem and (2.3) now immediately yields the following result on the size of the induced recursion tree:

THEOREM 7. *For f of degree n with integer coefficients of bitsize τ,* NEWDSC *induces a recursion tree T_{NEWDSC} of size*

$$|T_{\text{NEWDSC}}| = \mathrm{var}(f,\mathscr{I}_0)\cdot O(\log(n\tau)) = O(n\cdot\log(n\tau)),$$

where $\mathscr{I}_0 := (-2^{\tau+1},2^{\tau+1})$ denotes the initial interval known to contain all real roots of f.

3.3 Bit Complexity Analysis

We now derive an upper bound for the number of bit operations that are needed to determine isolating intervals for all real roots of f. We will show that, in each iteration, the costs are dominated by the computation of the polynomial f_I as defined in (2.2). The costs for this step mainly depend on the bitsize of the endpoints and, thus, on the width of the interval $I = (a,b)$. The following lemma provides a lower bound on the width of the milestones J_i, and thus also for the nodes $I \in T_i$, in terms of the separations of the roots $z_1, \dots, z_n$:

LEMMA 8. *For a polynomial f as defined in (2.1), suppose that its roots $z_1, \dots, z_n$ are ordered with respect to their separations (i.e. $\sigma(z_1) \geq \cdots \geq \sigma(z_n)$). Then,*

$$w(J_i) > \frac{\sigma(z_{n-(n'-i)})}{4} \cdot n^{-5-2\log n}, \tag{3.8}$$

where $J_1, \dots, J_{n'}$ are the milestones in the recursion tree T_{NEWDSC} which are ordered with respect to length (i.e. $w(J_1) \geq \cdots \geq w(J_{n'})$). Furthermore, the endpoints of each interval $I \in T_i \cup \{J_i\}$ are dyadic numbers representable by $O(\tau + \log \sigma^{-1}_{n-(n'-i)} + \log^2 n)$ bits.

PROOF. Let $k_0, \dots, k_s$ be integers with $k_0 := 0 < k_1 < k_2 < \cdots < k_s$ such that

$$\sigma(z_1) = \cdots = \sigma(z_{n-k_s}) > \sigma(z_{n-k_s+1}) = \cdots = \sigma(z_{n-k_{s-1}}) > \cdots$$
$$> \sigma(z_{n-k_2+1}) = \cdots = \sigma(z_{n-k_1}) > \sigma(z_{n-k_1+1}) = \cdots = \sigma(z_{n-k_0}).$$

For fixed $k := k_i$ and $\sigma := \sigma(z_{n-k_i})$, exactly the k roots $z_{n-k+1}, \dots, z_n$ have separation less than σ. For $k = 0$, there exists no root with separation less than σ. We further denote $I_1 := J_{i_1}, \dots, I_m := J_{i_m}$ the milestones of T_{NEWDSC} such that

- $w(I_l) < w_{\min} := \frac{\sigma}{4} \cdot n^{-5-2\log n}$ for all $l = 1, \dots, m$, and
- each milestone J_i which contains I_l has width $w(J_i) \geq w_{\min}$.

In addition, $v_l := \text{var}(f, I_l) \geq 2$ denotes the number of sign variations for I_l. Since the intervals I_l are disjoint, we have $v_1 + \cdots + v_m \leq n$, and thus $m \leq n/2$. According to Theorem 1, v_l is a lower bound for the number of roots within the Obreshkoff area $A_n^{(l)}$ for I_l. Furthermore, the proof of Lemma 5 yields that any two points within $A_n^{(l)}$ have distance less than $4nw(I_l) < \sigma$, and thus each root contained in $A_n^{(l)}$ must be one of the k roots $z_{n-k+1}, \dots, z_n$. Let S_l denote the set of all roots which are contained in $A_n^{(l)}$.

We first consider the case, where the Obreshkoff areas $A_n^{(l)}$ are pairwise disjoint. Then, the subsets $S_l \subset A_n^{(l)}$ are also pairwise disjoint, and thus $v_1 + \cdots + v_s \leq |S_1| + \cdots + |S_m| \leq k$. In the case where some of the $A_n^{(l)}$ overlap, it is possible that some of the roots contained in these areas are counted more than once, and thus the above argument does not directly apply. However, the following consideration shows that the sum of all v_l is still upper bounded by k: Let $\mathscr{A}$ be a list of active intervals, where we initially set $\mathscr{A} := \{I_1, \dots, I_m\}$. In each iteration, we pick two intervals $I = (a,b)$ and $I' = (a',b')$ from $\mathscr{A}$ whose corresponding Obreshkoff areas overlap. Then, we remove I, I', and all intervals $J \in \mathscr{A}$ in between I and I'. Finally, we add the smallest interval K to $\mathscr{A}$ which contains I and J (i.e. $K = (\min(a,a'), \max(b,b'))$). We proceed in this way until we obtain intervals $I'_1, \dots, I'_{m'}$ such that the corresponding Obreshkoff areas do not overlap. From our construction, the intervals $I_1, \dots, I_m$ are covered by $I'_1, \dots, I'_{m'}$. Now using Lemma 5 (2) in an inductive manner further shows that each of the so-obtained intervals I'_l has width $w(I'_l) < w_{\min} \cdot n^{4+2\log n}$; see also Appendix 6.2 for a rigorous proof. Hence, the same argument as above (applied to $I'_1, \dots, I'_{m'}$ instead of $I_1, \dots, I_m$) yields $v_1 + \cdots + v_m \leq v' := \sum_{l=1}^{m'} \text{var}(f, I'_l) \leq k$, where the latter inequality follows from the fact that the Obreshkoff area A_n for each I'_l contains only roots with separation less than $4nw(I'_l) < 4nw_{\min} \cdot n^{4+2\log n} < \sigma(z_{n-k})$. Hence, in total, we count at most k sign variations for the intervals I_l. In Section 2, we argued that there exist at most $\text{var}(f, \mathscr{I}_0)$ milestones. Now, the same argument also shows that the number of milestones J with $w(J) \leq \sigma$ is bounded by k. Namely, we start with milestones $I_1, \dots, I_m$ with $\sum_{l=1}^m \text{var}(f, I_l) \leq k$, and whenever a milestone is subdivided, the value $\sum_I \text{var}(f,I) - \#\{I : \text{var}(f,I) \geq 2\}$ decreases by at least one. In summary, there exists no milestone of width less than $\sigma(z_{n-k_0}) n^{-4-2\log n}/4$, and there exists at most k_i milestones of width less than $\sigma(z_{n-k_i}) n^{-4-2\log n}/4$ for each $i > 0$. Since the J_i are ordered with respect to their lengths, the bound (3.8) now follows by an inductive argument. For an interval $I = (a,b) \in T_i \cap \{J_i\}$, we remark that, due to our construction, a and b are both dyadic numbers of absolute value bounded by $2^{\tau+1}$. In addition, the absolute value of the denominators of a and b in their dyadic representations are both bounded by $\max(1, \log w(I)^{-1})$. Hence, we can represent a and b with $O(\tau + \log \sigma^{-1}_{n-(n'-i)} + \log^2 n)$ many bits. □

We will now derive our final result on the bit complexity of NEWDSC: In each step of the algorithm, we have to compute the polynomial $f_I(x) = (x+1)^n \cdot f((ax+b)/(x+1))$, where $I = (a,b)$ is the interval that is actually processed. The latter computation decomposes into computing $f_I^* := f(a + (b-a)x)$, reversing the coefficients, and then applying a Taylor shift by 1 (i.e. $x \mapsto x+1$). For the computation of f_I^*, we first shift f by a, and then scale by a factor $b - a = w(I)$ which is a power of two. Using asymptotically fast Taylor shift [21], the computation of $f(x+a)$ demands for $\tilde{O}(n^2(\tau + \log w(I)^{-1}))$ bit operations. The scaling $x \mapsto (b-a) \cdot x$ is achieved by shifting the i-th coefficient of $f(x+a)$ by $i \cdot \log(b-a)^{-1} = i \cdot \log w(I)^{-1}$ many bits. Then, the resulting polynomial f_I^* has coefficients whose absolute value is bounded by $2^{O(n\tau)}$, and the corresponding denominators are powers of two bounded by $2^{O(n(\tau + \log w(I)^{-1}))}$. Hence, reversing the coefficients of f_I^*, and then applying a Taylor shift by 1, demands for $\tilde{O}(n^2(\tau + \log w(I)^{-1}))$ bit operations. In summary, the cost for computing f_I is bounded by $\tilde{O}(n^2(\tau + \log w(I)^{-1}))$. The same bound further applies to the computation of λ_1 and λ_2 in Step 3 (b) of the algorithm because, in this step, we have to evaluate a polynomial of degree n and bitsize τ at a $(\tau + \log w(I)^{-1})$-bit number. If I is non-terminal, we also have to compute the number of sign variations for the intervals B_1, B_2, I'_1 and I'_2. The same argument as above also shows that we can do so using $\tilde{O}(n^2(\tau + \log N_I + \log w(I)^{-1}))$ bit operations since the latter mentioned intervals have size $w(I)/N_I$. From Lemma 3 and 8, we conclude that $\log N_I + \log \frac{1}{w(I)}$ is bounded by

$$O(\log \frac{N_I}{w(J_i)}) = O(\tau + \log \frac{1}{\sigma(z_{n-(n'-i)})} + \log^2 n)) \text{ for all } I \in T_i \cup J_i.$$

Thus, all computations at I demand for $\tilde{O}(n^2(\tau + \sigma(z_{n-(n'-i)})^{-1}))$ bit operations. It remains to consider a terminal interval I which is one of the two children of a milestone J_i. In this case, it suffices to bound the cost for the computation of f_I because $\text{var}(f,I) \leq 1$. Since $w(I) = w(J_i)/2$, the latter computation needs $\tilde{O}(n^2(\tau + \log w(J_i)^{-1})) = \tilde{O}(n^2(\tau + \log \sigma(z_{n-(n'-i)})^{-1}))$ bit operations as well.

Due to Theorem 6, we have $|T_i| = O(\log(n\tau))$ for all i. Thus, the total cost for isolating all real roots of f is bounded by

$$O(\log(n\tau)) \cdot \sum_{i=1}^{n'} \tilde{O}(n^2(\tau + \log \sigma(z_{n-(n'-i)})^{-1} + \log^2 n)) = \tilde{O}(n^3\tau)$$

since $\sum_{i=1}^{n'} \log \sigma(z_{n-(n'-i)})^{-1} = O(n\tau + \sum_{i=1}^{n} \log \sigma(z_i)^{-1}) = \tilde{O}(n\tau)$; see Lemma 19 in [17] or [18] for the latter bound. We fix this result:

THEOREM 9. *For a polynomial f as in (2.1),* NEWDSC *isolates the real roots of f using no more than $\tilde{O}(n^3\tau)$ bit operations.*

Remark. A similar argument as in the proof of Lemma 3 shows that, for each real root ξ of f, NEWDSC returns an isolating interval I of width $\sigma(\xi)^3 \cdot 2^{-2(\tau+2)} < w(I) \leq 2^{\tau+2}$. For some applications, it is necessary to further refine I to a width of 2^{-L} or less, where $L \in \mathbb{N}$ is given. We can directly use NEWDSC for the refinement, that is, I is processed in the same manner [4] as in the isolation routine, but we do not stop until $w(I) < 2^{-L}$. In order to do so, we need $O(\log(n\tau) + \log L)$ iterations because all but $O(\log(n\tau))$ many subdivision steps are quadratic. Namely, the same argument as in the proof of Theorem 6 to show that the subtrees T_i have length $O(\log(n\tau))$ also applies in this case (just consider $I_1 := I$ and I_s to be the first interval of width less than 2^{-L}). The cost for each refinement step is bounded by $\tilde{O}(n^2(L+\tau))$ since we have to perform n arithmetic operations with $O(n(L+\tau))$ bit numbers. Hence, the cost to obtain an approximation of ξ to L bits after the binary point is bounded by $\tilde{O}(n^2(L+\tau))$, and thus $\tilde{O}(n^3(L+\tau))$ for *all real roots* of f. When L is dominating, the latter bound is by a factor of n larger than the bound $\tilde{O}(n^3\tau^2 + n^2L)$ achieved by the AQIR-method [10], a variant of the QIR method from J. Abbott which uses approximate instead of exact computation in each step. AQIR eventually achieves quadratic convergence, however the total cost for the initial sequences, where only linear convergence can be guaranteed, is $\tilde{O}(n^3\tau^2)$. In order to improve upon the result from [10], we propose to combine NEWDSC and AQIR, that is, we use NEWDSC for the initial refinement steps until we can guarantee that AQIR achieves quadratic convergence. Following this approach, we eventually obtain our main result (see Appendix 6.2 for a rigorous proof):

THEOREM 10. *For a square-free polynomial f with integer coefficients of modulus less than 2^τ, we can compute isolating intervals (for all real root of f) of width less than 2^{-L} using no more than $\tilde{O}(n^3\tau + n^2L)$ bit operations.*

4. CONCLUSION

We introduced the first exact real root isolation method which achieves the record bound $\tilde{O}(n^3\tau)$ for the bit complexity of this problem. In comparison to the asymptotically fast numerical algorithms from the 1980s, which compute all complex roots, our approach entirely relies on exact computation, is much simpler, and can be considered very practical. The algorithm is based on a novel subdivision technique combining Descartes' Rule of Signs and Newton iteration. As a consequence, our algorithm shows quadratic convergence towards the roots in most steps.

So far, our algorithm applies to polynomials with integer (or rational) coefficients. In [16], it is shown how to modify an exact subdivision method such that it can also be used to isolate the roots of a polynomial $f \in \mathbb{R}[x]$ whose coefficients can be approximated to any specified error bound (so-called bitstream coefficients). In the bitstream setting, it is more reasonable to express the bit complexity in terms of the geometry of the roots (i.e. the separations $\sigma(z_i)$ and the maximum of all $|z_i|$) instead of the input size of the polynomial; see [17, Theorem 18] for a corresponding bound for the bisection approach. We are confident that the bound given in [17] can be further improved by considering a modified NEWDSC-method.

[4] Since I is already isolating, it certainly suffices to check for a sign change of f at the endpoints of the subintervals $I' \subset I$ instead of computing $\mathrm{var}(f, I')$ directly.

5. REFERENCES

[1] J. Abbott. Quadratic interval refinement for real roots. Poster presented at ISSAC, 2006.

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

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

[4] G. Collins and A. Akritas. Polynomial real root isolation using Descartes' rule of signs. In *ISSAC*, pages 272–275, 1976.

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

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

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

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

[9] 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.

[10] M. Kerber and M. Sagraloff. Efficient real root approximation. In *ISSAC*, pages 209–216, 2011.

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

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

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

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

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

[16] M. Sagraloff. A general approach to isolating roots of a bitstream polynomial. *Math. in Comput. Sci.*, 4(4):481–506, 2010.

[17] M. Sagraloff. On the complexity of real root isolation. arXiv:1011.0344v2, submitted to J. Symb. Comput., 2011.

[18] M. Sagraloff and C.-K. Yap. A simple but exact and efficient algorithm for complex root isolation. In *ISSAC*, pages 353–360, 2011.

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

[20] 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.

[21] J. von zur Gathen and J. Gerhard. Fast algorithms for Taylor shifts and certain difference equations. In *ISSAC*, pages 40–47, 1997.

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

6. APPENDIX

6.1 Algorithm

Algorithm 1 NEWDSC

Require: polynomial $f = \sum_{0\le i\le n} a_i x^i \in \mathbb{Z}[x]$ with integer coefficients a_i, $|a_i| < 2^\tau$ for all i.

Ensure: returns a list $\mathscr{O}$ of disjoint isolating intervals for all real roots of f

$I_0 := (-2^{\tau+1}, 2^{\tau+1})$; $N_{I_0} := 4$
$\mathscr{A} := \{(I_0, N_{I_0})\}$; $\mathscr{O} := \emptyset$ {*list of active and isolating intervals*}
repeat
  (I, N_I) some element in $\mathscr{A}$, $I = (a,b)$; delete (I, f_I) from $\mathscr{A}$
  $v := \mathrm{var}(f, I)$
  if $v = 0$ **then**
    do nothing {*I contains no root*}
  else if $v = 1$ **then**
    add I to $\mathscr{O}$ {*I isolates a real root*}
  else if $v > 1$ **then**
    $B_1 := (a, a + \frac{w(I)}{N_I})$; $B_2 := (b - \frac{w(I)}{N_I}, b)$
    if $\mathrm{var}(f, B_1) = v$ **or** $\mathrm{var}(f, B_2) = v$ **then**
      for the unique $i \in \{1,2\}$ with $\mathrm{var}(f, B_i) = v$, add $(B_i, N_{B_i}) := (B_i, N_I^2)$ to $\mathscr{A}$
      {*B_i contains all real roots within I*}
    else
      $\lambda_1 := a - v \cdot \frac{f(a)}{f'(a)}$; $\lambda_2 := b - v \cdot \frac{f(b)}{f'(b)}$
      for $i = 1, 2$:
      $k_i := \min(\max(\lfloor 4N_I \cdot \frac{\lambda_i - a}{b-a} \rfloor, 2), 4N_I - 2)$;
      $I_i' := (a + (k_i - 2) \cdot \frac{w(I)}{4N_I}, a + (k_i + 2) \cdot \frac{w(I)}{4N_I})$;
        {*$m_k := a + k \cdot \frac{w(I)}{4N_I}$ is the k-th subdivision point when decomposing I into $4N_I$ equally sized intervals; m_{k_i} is one of the two closest points to λ_i; I_i' has width $w(I)/N_I$ and is centered at m_{k_i}.*}
      if $\mathrm{var}(f, I_1') = v$ **or** $\mathrm{var}(f, I_2') = v$ **then**
        choose the smallest $i \in \{1,2\}$ with $\mathrm{var}(f, I_i') = v$ and add $(I_i', N_{I_i'}) := (I_i', N_I^2)$ to $\mathscr{A}$
          {*If $\mathrm{var}(f, I_i') = v$, then I_i' contains all roots within I*}
      else
        add $(I_1, N_{I_1}) := ((a, m(I)), \max(4, \sqrt{N_I}))$ and
        add $(I_2, N_{I_2}) := ((m(I), b), \max(4, \sqrt{N_I}))$ to $\mathscr{A}$
        if $f(m(I)) = 0$ **then**
          add $[m(I), m(I)]$ to $\mathscr{O}$
        end if
      end if
    end if
  end if
until $\mathscr{A}$ is empty
return $\mathscr{O}$

6.2 Complete Proofs

PROOF OF LEMMA 4. Throughout the following consideration, we call an index i *strong* (**S**) if $x_i/N_i \ge w'$ and *weak* (**W**), otherwise. If $w/4 < w'$, then each i with $i \ge 1$ is weak, and thus $i_0 \le 3$. For $w/4 \ge w'$, let k be the unique integer with $2^{-2^{k+1}} < w'/w \le 2^{-2^k}$. Then, $1 \le k \le \log\log \frac{w}{w'}$, and since $x_i \le x_{i-1}/2$ for all i, there exists an index i which is weak. Let k' denote the smallest weak index.

Claim 1: $k' \le k+1$
Assume otherwise, then the indices 1 to k are all strong. Hence,

$$x_{k+1} = w \cdot 2^{-(2^m + 2^{m+1} + \cdots + 2^{m+k-1})} = w \cdot 2^{-2^m(2^0 + 2^1 + \cdots + 2^{k-1})}$$
$$= w \cdot 2^{-2^m(2^k - 1)} \le 4w \cdot 2^{-2^{k+1}} < 4w',$$

and $n_{k+1} > 1$. It follows that $k+1$ is weak, a contradiction.

Let us now consider the subsequence $\mathscr{S} = k', k'+1, \ldots, i_0 - 3$:

Claim 2: $\mathscr{S}$ contains no subsequence of type ...**SS**... or ...**SWSWS**...
If there exists a weak index i and two strong indices $i+1$ and $i+2$, then $x_i/N_i > x_{i+2}/N_i \ge x_{i+2}/N_{i+2} \ge w'$ contradicting the fact that $x_i/N_i < w'$. Since $\mathscr{S}$ starts with a weak index, the first part of our claim follows. For the second part, assume that i, $i+2$ and $i+4$ are strong, and $i+1$ and $i+3$ are weak. Then,

$$w' \le \frac{x_{i+4}}{N_{i+4}} < \frac{x_{i+2}}{N_{i+2} \cdot N_{i+4}} < \frac{x_i}{N_i \cdot N_{i+2} \cdot N_{i+4}} = \frac{x_i}{N_i^3} = \frac{x_{i+1}}{N_{i+1}}$$

contradicting the fact that $i+1$ is weak.

Claim. 3: If i is weak and $i < i_0$, then $n_i \ge 2$.
Namely, if i is weak and $n_i = 1$, then $x_i/4 = x_i/N_i < w'$, and thus $x_{i_0-1} < w'$ which contradicts the definition of i_0.

We now partition the sequence $\mathscr{S}$ into maximal subsequences $\mathscr{S}_1$, $\mathscr{S}_2, \ldots, \mathscr{S}_r$ such that each $\mathscr{S}_j$, $j = 1, \ldots, r$, contains no two consecutive weak elements. Then, according to our above results, each $\mathscr{S}_j$, with $j < r$, is of type **W**, **WSW**, or **WSWSW**. The last subsequence $\mathscr{S}_r$ (with last index $i_0 - 3$) is of type **W**, **WS**, **WSW**, **WSWS**, or **WSWSW**. After each S_j, with $j < r$, the number n_i decreases by one, and thus we must have $r \le n_1 + k'$ since we start with $n_{k'} = n_1 + k' - 1$ and, in addition, $n_i \ge 2$ for all weak i. Since the length of each $\mathscr{S}_j$ is bounded by 5, it follows that

$$i_0 = i_0 - 3 + 3 \le k' + 5r + 3 \le 5(n_1 + k') + k' + 3 \le 8(n_1 + k).$$

□

PROOF OF LEMMA 5. (1) In a first step, we compute the radius r' of the Obreshkoff discs $\underline{C}_n'$ and $\overline{C}_n'$ for the interval $I' = (a', b')$: A point ξ on the boundary of the Obreshkoff area A_n' (except a' and b') sees I' under an angle $\gamma = \pi/(n+2)$; see Figure 2.1 and 3.1. Hence, from the extended Sine Theorem, it follows that

$$r' = \frac{w(I')}{2\sin(\gamma)} = \frac{w(I')}{2\sin(\pi/(n+2))} < \frac{(n+2)w(I')}{\pi} < n \cdot w(I')$$

since $\sin x > x/2$ for all $x \in (0, \pi/4]$. In particular, each point z within the Obreshkoff area A_n' has distance at most $2r' < 2n \cdot w(I')$ from any point within I. W.l.o.g., we assume that $|a - a'| \le |b - b'|$. Then, the distance from a' to the boundary of the Obreshkoff lens L_n for I is bounded by the distance δ from a' to the line $\overline{ac}$, where c denotes the topmost point of L_n. Since $\overline{ac}$ intersects the x-axis in an angle of $\pi/(2(n+2))$, we have $\delta = |a - a'| \sin \pi/(2(n+2)) > |a - a'|/(4n)$. Thus, $A_n' \subset L_n$ if $|a - a'|/(4n) > 2n \cdot w(I')$ or $|a - a'| > 8n^2 w(I')$. For $|a - a'| > |b - b'|$, a similar argument shows that $A_n' \subset L_n$ if $|b - b'| > 8n^2 w(I')$. In addition, if the inequality $\min(|a - a'|, |b - b'|) > 8n^2 w(I')$ holds, then each $\xi \in A_n'$ has distance at least $\delta - 2nw(I') > \min(|a - a'|, |b - b'|)/(4n) - 2nw(I')$

from any point located outside of L_n.
(2) W.l.o.g., we can assume that $w(I') \le w(I)$ and $a' \ge b$. Let L be the line passing through b which intersects the x-axis in an angle of $\pi/(n+2)$. Then, the upper part of the Obreshkoff area A_n lies completely on one side of this line. Now, if A'_n lies completely on the other side of L, then, by symmetry, A_n and A'_n do not share a common point. We have already argued that A'_n is contained within the disc of radius $2nw(I')$ centered at a'. Hence, if the distance $\delta' := \mathrm{dist}(a', L)$ from a' to L is larger than $2nw(I')$, then $A_n \cap A'_n = \emptyset$. We have $\delta' = |a'-b|\sin(\pi/(n+2)) > |a'-b|/2n = \mathrm{dist}(I, I')/2n$, and thus our claim follows. □

PROOF OF LEMMA 8. In addition to the proof as given in the main paper, it remains to show that the length of each of the intervals $I'_1, \ldots, I'_{m'}$ is bounded by $w_{\min} \cdot n^{4+2\log n}$: At any stage of the "merging process", an interval $J \in \mathscr{A}$ covers a certain number s of intervals from $I_1, \ldots, I_m$. By induction on s, we prove that

$$w(J) < w_{\min} \cdot s^4 n^{2\log s}, \text{ and thus } w(J) \le w_{\min} \cdot n^{2\log n+4}. \quad (6.1)$$

If J covers only one interval I_l, then $J = I_l$ which proves the claim for $s = 1$. An interval J covering $s+1$ intervals from $I_1, \ldots, I_m$ is obtained by merging two intervals I and I' which cover s_1 and s_2 intervals, respectively, where $s_1 + s_2 \le s+1$ and, w.l.o.g., $1 \le s_1 \le s_2$. From Lemma 5, it follows that the distance between I and I' is bounded by $4n^2 \cdot \min(w(I'), w(I))$ since the corresponding Obreshkoff areas overlap. Hence, J has width

$$\begin{aligned} w(J) &\le w(I) + w(I') + 4n^2 \cdot \min(w(I'), w(I)) \\ &< w_{\min} \cdot (s_2^4 n^{2\log s_2} + (4n^2+1) s_1^4 n^{2\log s_1}) \\ &< w_{\min} \cdot (s_2^4 n^{2\log s_2} + 8 s_1^4 n^{2\log s_1 + 2}) \\ &= w_{\min} \cdot (s_2^4 n^{2\log s_2} + 8 s_1^4 n^{2\log 2 s_1}) \end{aligned}$$

If $2s_1 \ge s_2$, then

$$s_2^4 n^{2\log s_2} + 8 s_1^4 n^{2\log 2 s_1} \le (8 s_1^4 + s_2^4) n^{2\log 2 s_1} \le (s_1+s_2)^4 n^{2\log(s_1+s_2)}.$$

Otherwise, we have

$$s_2^4 n^{2\log s_2} + 8 s_1^4 n^{2\log 2 s_1} \le (8 s_1^4 + s_2^4) n^{2\log s_2} < (s_1+s_2)^4 n^{2\log(s_1+s_2)},$$

and thus (6.1) follows. □

PROOF OF THEOREM 10. The AQIR method from [10] can be considered as a variant of the QIR method, that is, in each iteration, the algorithm aims to refine an isolating interval $I = (a,b)$ (for a root ξ) via approximating f by a line segment passing through $(a, f(a))$ and $(b, f(b))$. However, in comparison to the initial QIR method, AQIR is exclusively based on approximate instead of exact arithmetic. In [10, Corollary 14], it has been shown that each step in the AQIR sequence (see [10, Definition 11] for the definition)

$$S_0 := (I_0, N_0) = (I, 4), S_i = (I_i, N_i) := \text{AQIR}(f, I_{i-1}, N_{i-1}, s) \text{ for } i \ge 1,$$

is successful (i.e. $N_i = N_{i-1}^2$ and $w(I_i) = w(I_{i-1})/N_{i-1}$) if we start with an interval $I_0 := I$ of width

$$w(I_0) < w_\xi := \frac{|f'(\xi)|}{32 e n^3 2^\tau \max\{|\xi|, 1\}^{n-1}}, \quad (6.2)$$

where $e \approx 2.71\ldots$ denotes the Eulerian number. Furthermore, [10, Lemma 21] provides a bound for the number of bit operations to compute the sequence $(S_i)_{0 \le i \le i_0}$, where i_0 denotes the smallest index with $w(I_i) < 2^{-L}$. For a polynomial f as in (2.1), this bound writes as $\tilde{O}(nL + n^2\tau)$. Hence, given isolating intervals I_ξ of width $w(I_\xi) < w_\xi$ for all real roots ξ of f, the refinement of all these intervals to a width less than 2^{-L} demands for $\tilde{O}(n^3\tau + n^2 L)$ bit operations. It remains to show that such intervals I_ξ can be computed using no more than $\tilde{O}(n^3\tau + n^2 L)$ bit operations. In order to do so, we first isolate the real roots (w.l.o.g. we assume that these are the roots $z_1, \ldots, z_m$) of f using NEWDSC. For each of the corresponding isolating intervals $I = I_\xi = (a,b)$, $\xi = z_i$, we now proceed as follows: We first refine I using NEWDSC until $w(I) < 1/(2n)$ and we count one sign variation for the enlarged interval

$$I^+ := \left(a - \frac{w(I)}{2} \cdot (2^{3\lceil \log n \rceil + 6} - 1), b + \frac{w(I)}{2} \cdot (2^{3\lceil \log n \rceil + 6} - 1)\right),$$

that is, $\mathrm{var}(f, I^+) = 1$. The interval $I^+ = (a^+, b^+)$ has width $w(I^+) = 2^{3\lceil \log n \rceil + 6} \cdot w(I) \ge 64 n^3 w(I)$ and is centered at I. The cost for this refinement is bounded by $\tilde{O}(n^2(\tau + \log n + \log \sigma(\xi)^{-1}))$ since the endpoints of the interval I (and I^+) are dyadic numbers that can be represented by $O(\tau + \log n + \log \sigma(\xi)^{-1})$ many bits and we need at most $O(\log(n\tau) + \log\log \sigma(\xi)^{-1})$ many iterations. Hence, the total cost for all real roots is bounded by $\tilde{O}(n^3\tau)$. Since $\mathrm{var}(f, I^+) = 1$, the Obreshkoff lens L_n^+ for I^+ contains exactly one root, namely, $\xi \in I$. According to Lemma 5, the distance from an arbitrary point within I to an arbitrary point outside L_n^+ is lower bounded by

$$\frac{1}{4n} \cdot \Big(\min(|a^+ - a|, |b^+ - b|) - 8n^2 w(I)\Big) > 4n^2 w(I),$$

and thus $w(I) < \sigma(\xi)/(4n^2)$. It is well-known (e.g. [22]) that the disc $\Delta_{\sigma(\xi)/n}(\xi)$ of radius $\sigma(\xi)/n$ centered at $\xi \in I$ contains no root of the derivative f', hence the disc $\Delta_{2nw(I)}(m(I)) \subset \Delta_{\sigma(z_i)/n}(\xi)$ contains no root of f' as well. It follows that

$$|f'(\xi)|/2 < |f'(a)| < 2|f'(\xi)| \quad (6.3)$$

since, for each root z'_j of the derivative f', we have $|a - z'_j|/|\xi - z'_j| \in (1 - 1/(2n), 1 + 1/(2n))$, and $(1 + 1/(2n))^{n-1} < \sqrt{e} < 2$ and $(1 - 1/(2n))^{n-1} > 1/\sqrt{e} > 1/2$. In addition, we have

$$(1 - 1/(2n)) \cdot \max(1, |\xi|) < \max(1, |a|) < (1 + 1/(2n)) \cdot \max(1, |\xi|)$$

since $w(I) < 1/(2n)$. Hence, it follows that

$$\frac{1}{2} \cdot \max(1, |a|)^{n-1} < \max\{|\xi|, 1\}^{n-1} < 2 \max(1, |a|)^{n-1}. \quad (6.4)$$

Now, we can easily check whether $w(I) < w_\xi$: Namely, according to (6.2), (6.3) and (6.4), the latter inequality holds for sure if

$$w(I) < \frac{|f'(a)|}{128 e n^3 2^\tau \max\{|a|, 1\}^{n-1}}, \quad (6.5)$$

Hence, we refine I using NEWDSC until the inequality (6.5) is fulfilled. Since $w(I) < w_\xi/16$ implies that (6.5) holds, we have to refine I to a width $w_\xi/16$ or less. This step demands for $\tilde{O}(n^2(\tau + n \log \max(1, |\xi|) - \log |f'(\xi)|))$ bit operations due to the quadratic convergence of NEWDSC for all but $O(\log(n\tau))$ many iterations, and the fact that the endpoints of the interval I are representable by $O(\tau + n + n \log \max(1, |\xi|) - \log |f'(\xi)|)$ many bits. Using the Mahler bound [22] yields $\log \prod_{i=1}^m \max(1, |z_i|) = O(\tau)$. The product of all $f'(z_i)$, $i = 1, \ldots, n$, equals $\mathrm{lcf}(f)^{2-n} \mathrm{Disc}(f)$, where $\mathrm{lcf}(f)$ denotes the leading coefficient and $\mathrm{Disc}(f) \in \mathbb{Z}$ the discriminant of f. Since $|f'(z_i)| \le n^2 2^\tau \max(1, |z_i|)^n$ for all i, it follows that

$$\begin{aligned} \prod_{i=1}^m |f'(z_i)| &\ge \prod_{i>m} (n^2 2^\tau \max(1, |z_i|)^n)^{-1} \cdot \prod_{i=1}^n |f'(z_i)| \\ &\ge \prod_{i=1}^n (n^2 2^\tau \max(1, |z_i|)^n)^{-1} \cdot \mathrm{lcf}(f)^{2-n} \mathrm{Disc}(f) = 2^{-O(n(\log n + \tau))} \end{aligned}$$

Thus, the total cost for the refinement is bounded by $\tilde{O}(n^3\tau)$. □