

# Efficient Real Root Approximation

Michael Kerber  
IST (Institute of Science and Technology) Austria  
3400 Klosterneuburg, Austria  
mkerber@ist.ac.at

Michael Sagraloff  
Max-Planck-Institute for Informatics  
66123 Saarbrücken, Germany  
msagralo@mpi-inf.mpg.de

## ABSTRACT

We consider the problem of approximating all real roots of a square-free polynomial  $f$ . Given isolating intervals, our algorithm refines each of them to a certain width  $2^{-L}$ , that is, each of the roots is approximated to  $L$  bits after the binary point. Our method provides a certified answer for arbitrary real polynomials, only considering finite approximations of the polynomial coefficient and choosing a suitable working precision adaptively. In this way, we get a correct algorithm that is simple to implement and practically efficient. Our algorithm uses the quadratic interval refinement method; we adapt that method to be able to cope with inaccuracies when evaluating  $f$ , without sacrificing its quadratic convergence behavior. We prove a bound on the bit complexity of our algorithm in terms of degree, coefficient size and discriminant. Our bound improves previous work on integer polynomials by a factor of  $\deg f$  and essentially matches best known theoretical bounds on root approximation which are obtained by very sophisticated algorithms.

## Categories and Subject Descriptors

G.1.5 [Numerical Analysis]: Roots of Nonlinear Equations; F.2.1 [Analysis of Algorithms and Problem Complexity]: Numerical Algorithms and Problems

## General Terms

Algorithms, Reliability, Theory

## Keywords

Root isolation, Root approximation, Quadratic Interval Refinement

## 1. INTRODUCTION

The problem of computing the real roots of a polynomial in one variable is one of the best studied problems in mathematics. If one asks for a *certified* method that finds all roots, it is common to write the solutions as a set of disjoint *isolating* intervals, each containing exactly one root; for that reason, the term *real root isolation* is common in the literature. Simple, though efficient methods for

this problem have been presented, for instance, based on Descartes' rule of signs [7], or on Sturm's theorem [9]. Recently, the focus of research shifted to polynomials with real coefficients which are approximated during the algorithm. It is worth to remark that this approach does not just generalize the integer case but has also lead to practical [11, 16] and theoretical [17] improvements of it.

We consider the related *real root refinement problem*: assuming that isolating intervals of a polynomial are known, *refine* them to a width of  $2^{-L}$  (where  $L \geq 0$  is an additional input parameter). Clearly, the combination of root isolation and root refinement, also called *strong root isolation*, yields a certified approximation of all roots of the polynomial to an absolute precision of  $2^{-L}$  or, in other words, to  $L$  bits after the binary point in binary representation.

We present a solution to the root refinement problem for arbitrary square-free polynomial with real coefficients. Most of the related approaches are formulated in the REAL-RAM model where exact operations on real numbers are assumed to be available at unit costs. In contrast, our approach considers the coefficients as *bit-streams*, that is, it only works with finite prefixes of its binary representation, and we also quantify how many bits are needed in the worst case. The refinement uses the quadratic interval refinement method [1] (QIR for short), which is a quadratically converging hybrid of the bisection and secant method. We adapt the method to work with a increasing working precisions and use interval arithmetic to validate the correctness of the outcome. In this way, we obtain an algorithm that always returns a correct root approximation, is simple to implement on an actual computer (given that arbitrary approximations of the coefficients are accessible), and is adaptive in the sense that it might succeed with a much lower working precision than asserted by the worst-case bound.

We provide a bound on the bit complexity of our algorithm. Let

$$f(x) := \sum_{i=0}^d a_i x^i \in \mathbb{R}[x] \quad (1)$$

be a polynomial of degree  $d \geq 2$  with leading coefficient  $|a_d| \geq 1$  and  $a_i < 2^\tau$  for all  $i$ . Given initial isolating intervals, our algorithm refines one interval to width  $2^{-L}$  using

$$\tilde{O}(d(d\tau + R)^2 + dL)$$

bit operations and refines all intervals using

$$\tilde{O}(d(d\tau + R)^2 + d^2L)$$

bit operations, where  $R := \log(|\text{res}(f, f')|)^{-1}$  and  $\tilde{O}$  means that we ignore logarithmic factors in  $d$ ,  $\tau$ , and  $L$ . To do so, our algorithm requires the coefficients of  $f$  in a precision of at most

$$O(d\tau + R + L)$$

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISSAC '11 San Jose, California USA

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$10.00.

bits after the binary point. We remark that the costs of obtaining approximations for the coefficients are unaccounted in this bound. For the analysis, we divide the sequence of QIR steps in the refinement process into a *linear sequence* where the method behaves like bisection in the worst case, and a *quadratic sequence* where the interval is converging quadratically towards the root, following the approach in [12]. We do not require any conditions on the initial intervals except that they are disjoint and cover all real roots of  $f$ ; an initial *normalization phase* modifies the intervals to guarantee the efficiency of our refinement strategy.

We remark that, using the recently presented root solver from [17], obtaining initial isolating intervals can be done within  $\tilde{O}(d(d+R)^2)$  bit operations using coefficient approximations of  $O(d\tau+R)$ . Combined with that result, our complexity result also gives a bound on the strong root isolation problem.

The case of integer coefficients is often of special interest, and the problem has been investigated by previous work [12] for this restricted case. In that work, the complexity of root refinement was bounded by  $\tilde{O}(d^4\tau^2+d^3L)$ . We improve this bound to

$$\tilde{O}(d^3\tau^2+d^2L)$$

because  $R$  as defined above becomes negative for integer polynomials. The difference in the complexities is due to a different approach to evaluate the sign of  $f$  at rational points, which is the main operation in the refinement procedure: for an interval of size  $2^{-\ell}$ , the evaluation has a complexity of  $\tilde{O}(d^2(\tau+\ell))$  when using exact rational arithmetic because evaluated function values can consist of up to  $d(\tau+\ell)$  bits. However, we show that we can still compute the sign of the function value with certified numerical methods using the substantially smaller working precision of  $O(d\tau+\ell)$ .

**Related work.** The problem of accurate root approximation is omnipresent in mathematical applications; certified methods are of particular importance in the context of computations with algebraic objects, e.g., when computing the topology of algebraic curves [10, 6] or when solving systems of multivariate equations [3].

The idea of combining bisection with a faster converging method to find roots of continuous functions has been first introduced in *Dekker's method* and elaborated in *Brent's method*; see [5] for a summary. However, these approaches assume exact arithmetic for their convergence results.

For polynomial equations, numerous algorithms are available, for instance, the *Jenkins-Traub algorithm* or *Durant-Kerner iteration*; although they usually approximate the real roots very fast in practice, general worst-case bounds on their arithmetic complexity are not available. In fact, for some variants, even termination cannot be guaranteed in theory; we refer to the survey [15] for extensive references on these and further methods.

The theoretical complexity of root approximation has been investigated by Pan [14]. Assuming all roots to be in the unit disc, he achieves a bit complexity of  $\tilde{O}(n^3+n^2L)$  for approximating all roots to an accuracy of  $2^{-L}$ , which matches our bound if  $L$  is the dominant input parameter. His approach even works for polynomials with multiple roots. However, as Pan admits in [15], the algorithm is difficult to implement and so is the complexity analysis when taking rounding errors in intermediate steps into account. Moreover, it appears unclear whether his bound can be improved if only a single root needs to be approximated.

We finally remark that a slightly simplified version of our approach (for integer coefficients) is included in the recently introduced CGAL<sup>1</sup>-package on algebraic computations [4]. Experiment-

---

### Algorithm 1 EQIR: Exact Quadratic Interval Refinement

---

INPUT:  $f \in \mathbb{R}[x]$  square-free,  $I = (a, b)$  isolating,  $N = 2^{2^i} \in \mathbb{N}$

OUTPUT:  $(J, N')$  with  $J \subseteq I$  isolating for  $\xi$  and  $N' \in \mathbb{N}$

```

1: procedure EQIR( $f, I = (a, b), N$ )
2:   if  $N = 2$ , return (BISECTION( $f, I$ ), 4).
3:    $\omega \leftarrow \frac{b-a}{N}$ 
4:    $m' \leftarrow a + \text{round}(N \frac{f(a)}{f(a)-f(b)}) \omega \triangleright m' \approx a + \frac{f(a)}{f(a)-f(b)}(b-a)$ 
5:    $s \leftarrow \text{sign}(f(m'))$ 
6:   if  $s = 0$ , return ( $[m', m']$ ,  $\infty$ )
7:   if  $s = \text{sign}(f(a))$  and  $\text{sign}(f(m' + \omega)) = \text{sign}(f(b))$ , return ( $[m', m' + \omega]$ ,  $N^2$ )
8:   if  $s = \text{sign}(f(b))$  and  $\text{sign}(f(m' - \omega)) = \text{sign}(f(a))$ , return ( $[m' - \omega, m']$ ,  $N^2$ )
9:   Otherwise, return ( $I, \sqrt{N}$ ).

```

---

tal comparisons in the context of [3] have shown that the approximate version of QIR gives significantly better running times than its exact counterpart. These observations underline the practical relevance of our approximate version and suggest a practical comparison with state-of-the-art solvers mentioned above as further work.

**Notation.** Additional to  $f$ ,  $d$ ,  $\tau$ ,  $a_i$ , and  $R$  as above, we use the following terminology throughout the paper: We denote the complex roots of  $f$  by  $z_1, \dots, z_d$  with  $z_1, \dots, z_m, m \leq d$ , exactly the real roots. For each  $z_i$ ,  $\sigma_i = \sigma(z_i, f) := \min_{j \neq i} |z_i - z_j|$  denotes the *separation* of  $z_i$  and  $\Sigma_f := \sum_{i=1}^n \log \sigma_i^{-1}$ . An interval  $I = (a, b)$  is called *isolating* for a root  $z_i$  if  $I$  contains  $z_i$  and no other root of  $f$ . We set  $m(I) = \frac{a+b}{2}$  for the *center* and  $w(I) := b - a$  for the *width* of  $I$ .

**Outline.** We summarize the (exact) QIR method in Section 2. A variant using only approximate coefficients is described in Section 3. Its precision demand is analyzed in Section 4. Based on that analysis of a single refinement step, the complexity bound of root refinement is derived in Section 5. The appendix provides proofs of some theorems in this work which are left out for brevity.

## 2. REVIEW ON EXACT QIR

Abbott's QIR method [1, 12] is a hybrid of the simple (but inefficient) bisection method with a quadratically converging variant of the *secant method*. We refer to this method as EQIR, where "E" stands for "exact" in order to distinguish from the variant presented in Section 3. Given an isolating interval  $I = (a, b)$  for a real root  $\xi$  of  $f$ , we consider the secant through  $(a, f(a))$  and  $(b, f(b))$  (see also Figure 1). This secant intersects the real axis in the interval  $I$ , say at  $x$ -coordinate  $m$ . For  $I$  small enough, the secant should approximate the graph of the function above  $I$  quite well and, so,  $m \approx \xi$  should hold. An EQIR step tries to exploit this fact:

The isolating interval  $I$  is (conceptually) subdivided into  $N$  subintervals of same size, using  $N+1$  equidistant grid points. Each subinterval has width  $\omega := \frac{w(I)}{N}$ . Then  $m'$ , the closest grid point to  $m$ , is computed and the sign of  $f(m')$  is evaluated. If that sign equals the sign of  $f(a)$ , the sign of  $f(m' + \omega)$  is evaluated. Otherwise,  $f(m' - \omega)$  is evaluated. If the sign changes between the two computed values, the interval  $(m', m' + \omega)$  or the interval  $(m' - \omega, m')$ , respectively, is set as new isolating interval for  $\xi$ . In this case, the EQIR step is called *successful*. Otherwise, the isolating interval remains unchanged, and the EQIR step is called *failing*. See Algorithm 1 for a description in pseudo-code.

In [12], the root refinement problem is analyzed using the just described EQIR method for the case of integer coefficients and ex-

<sup>1</sup>Computational Geometry Algorithms Library, [www.cgal.org](http://www.cgal.org)



step *successful*. Otherwise, we call the step *failing* and keep the old isolating interval. As in the exact case, we square up  $N$  after a successful step, and reduce it to its square root after a failing step. See Algorithm 3 for a complete description.

Note that, in case of a successful step, the new isolating interval  $I^*$  satisfies  $\frac{1}{8N}w(I) \leq w(I^*) \leq \frac{1}{N}w(I)$ . Also, similar to the bisection method, the function can only be zero at one of the chosen subdivision points, and the function is guaranteed to be reasonably large for all but one of them, which leads to a bound on the necessary precision (Lemma 6). The reader might wonder why we have chosen a non-equidistant grid involving the subdivision points  $m^* \pm \frac{7}{8}\omega$ . The reason is that these additional points allow us to give a success guarantee of the method under certain assumptions in the following lemma, which is the basis to prove quadratic convergence if the interval is smaller than a certain threshold (Section 5.2).

**Lemma 2.** *Let  $I = (a, b)$  be an isolating interval for some root  $\xi$  of  $f$ ,  $s = \text{sign}(f(a))$  and  $m$  as before. If  $|m - \xi| < \frac{b-a}{8N} = \frac{\omega}{8}$ , then AQIR( $f, I, N, s$ ) succeeds.*

**PROOF.** Let  $m^*$  be the subdivision point selected by the AQIR method. We assume that  $m^* \notin \{a, b\}$ ; otherwise, a similar (simplified) argument applies. By Lemma 1  $m \in [m^* - \frac{3}{4}\omega, m^* + \frac{3}{4}\omega]$  and, thus,  $\xi \in (m^* - \frac{7}{8}\omega, m^* + \frac{7}{8}\omega)$ . It follows that the leftmost two points of (2) have a different sign than the rightmost two points of (2). Since the sign of  $f$  is evaluated for at least one value on each side, the algorithm detects a sign change and, thus, succeeds.  $\square$

#### 4. ANALYSIS OF AN AQIR STEP

The running time of an AQIR step depends on the maximal precision  $\rho$  needed in the two while loops (Step 5, Steps 11-14) of Algorithm 3. The termination criterion of both loops is controlled by evaluations of the form  $\mathfrak{B}(E, \rho)$ , where  $E$  is some polynomial expression and  $\rho$  is the current working precision.

We specify recursively what we understand by evaluating  $E$  in precision  $\rho$  with interval arithmetic. For that, we define  $\text{down}(x, \rho)$  for  $x \in \mathbb{R}$  and  $\rho \in \mathbb{N}$  to be the maximal  $x_0 \leq x$  such that  $x_0 = \frac{k}{2^\rho}$  for some integer  $k$ . The same way  $\text{up}(x, \rho)$  is the minimal  $x_0 \geq x$  with  $x_0$  of the same form. We extend this definition to arithmetic expressions by the following rules (we leave out  $\rho$  for brevity):

$$\begin{aligned} \text{down}(E_1 + E_2) &:= \text{down}(E_1) + \text{down}(E_2) \\ \text{up}(E_1 + E_2) &:= \text{up}(E_1) + \text{up}(E_2) \\ \text{down}(E_1 \cdot E_2) &:= \text{down}(\min\{\text{down}(E_1)\text{down}(E_2), \text{up}(E_1)\text{up}(E_2), \\ &\quad \text{up}(E_1)\text{down}(E_2), \text{down}(E_1)\text{up}(E_2)\}) \\ \text{up}(E_1 \cdot E_2) &:= \text{up}(\max\{\text{down}(E_1)\text{down}(E_2), \text{down}(E_1)\text{up}(E_2), \\ &\quad \text{up}(E_1)\text{down}(E_2), \text{up}(E_1)\text{up}(E_2)\}) \\ \text{down}(1/E_1) &:= \text{down}(1/\text{up}(E_1)) \\ \text{up}(1/E_1) &:= \text{up}(1/\text{down}(E_1)) \end{aligned}$$

Finally, we define the interval  $\mathfrak{B}(E, \rho) := [\text{down}(E, \rho), \text{up}(E, \rho)]$ . By definition, the exact value of  $E$  is guaranteed to be contained in  $\mathfrak{B}(E, \rho)$ . We assume that polynomials are evaluated according to the Horner scheme. We give a bound on the size of the resulting interval under certain conditions next. It is similar to the more general result [13, Thm. 12]; see Appendix B for a proof.

**Lemma 3.** *Let  $f$  be a polynomial as in (1),  $c \in \mathbb{R}$  with  $|c| \leq 2^\tau$ , and  $\rho \in \mathbb{N}$ . Then,*

$$|f(c) - \text{down}(f(c), \rho)| \leq 2^{-\rho+1}(d+1)^2 2^{\tau d} \quad (3)$$

$$|f(c) - \text{up}(f(c), \rho)| \leq 2^{-\rho+1}(d+1)^2 2^{\tau d} \quad (4)$$

*In particular,  $\mathfrak{B}(f(c), \rho)$  has a width of at most  $2^{-\rho+2}(d+1)^2 2^{\tau d}$ .*

---

#### Algorithm 3 Approximate Quadratic interval refinement

---

INPUT:  $f \in \mathbb{R}[x]$  square-free,  $I = (a, b)$  isolating,  $N = 2^{2^\ell} \in \mathbb{N}$ ,  $s = \text{sign}(f(a))$

OUTPUT:  $(J, N')$  with  $J \subseteq I$  isolating and  $N' \in \mathbb{N}$

```

1: procedure AQIR( $f, I = (a, b), N$ )
2:   if  $N = 2$ , return (APPROXIMATE_BISECTION( $f, I, s$ ), 4).
3:    $\omega \leftarrow \frac{b-a}{N}$ 
4:    $\rho \leftarrow 2$ 
5:   while  $J \in \mathfrak{B}(N \frac{f(a)}{f(a)-f(b)}, \rho)$  has width  $> \frac{1}{4}$ , set  $\rho \leftarrow 2\rho$ 
6:    $m^* \leftarrow a + \text{round}(m(J)) \cdot \omega$ 
7:   if  $m^* = a$ ,  $s \leftarrow 4, V \leftarrow [m^*, m^* + \frac{1}{2}\omega, m^* + \frac{7}{8}\omega, m^* + \omega], S \leftarrow [s, 0, 0, 0]$ 
8:   if  $m^* = b$ ,  $s \leftarrow 4, V \leftarrow [m^* - \omega, m^* - \frac{7}{8}\omega, m^* - \frac{1}{2}\omega, m^*], S \leftarrow [0, 0, 0, -s]$ 
9:   if  $a < m^* < b$ ,  $s \leftarrow 7, V \leftarrow [m^* - \omega, m^* - \frac{7}{8}\omega, m^* - \frac{1}{2}\omega, m^*, m^* + \frac{1}{2}\omega, m^* + \frac{7}{8}\omega, m^* + \omega], S \leftarrow [0, 0, 0, 0, 0, 0, 0]$ 
10:   $\rho \leftarrow 2$ 
11:  while  $S$  contains more than one zero do
12:    for  $i=1, \dots, s$  do
13:      If  $S[i] = 0$ , set  $S[i] \leftarrow \text{sign } \mathfrak{B}(f(V[i]), \rho)$ 
14:     $\rho \leftarrow 2\rho$ 
15:  If  $\exists v, w : S[v] \cdot S[w] = -1 \wedge (v+1 = w \vee (v+2 = w \wedge S[v+1] = 0))$  return  $((V[v], V[w]), N^2)$ 
16:  Otherwise, return  $(I, \sqrt{N})$ 

```

---

We analyze the required working precision of approximate bisection and of an AQIR step next. We exploit that, whenever we evaluate  $f$  at  $t$  subdivision points,  $t-1$  of them have a certain minimal distance to the root in the isolating interval. The following lemma gives a lower bound on  $|f(x_0)|$  for such a point  $x_0$ , given that it is sufficiently far away from any other root of  $f$ .

**Lemma 4.** *Let  $f$  be as in (1),  $\xi = z_{i_0}$  a real root of  $f$  and  $x_0$  a real value with distance  $|x_0 - z_i| \geq \frac{\sigma_i}{4}$  to all real roots  $z_i \neq z_{i_0}$ . Then,*

$$|f(x_0)| > |\xi - x_0| \cdot 2^{-(2d+\tau+\Sigma_f)}.$$

(recall the notations from Section 1 for the definitions of  $\sigma_i$  and  $\Sigma_f$ )

**PROOF.** For each non-real root  $z_i$  of  $f$ , there exists a complex conjugate root  $\bar{z}_i$  and, thus, we have  $|x_0 - z_i| \geq \text{Im}(z_i) \geq \frac{\sigma_i}{2} > \frac{\sigma_i}{4}$  for all  $i = m+1, \dots, d$  as well. It follows that

$$\begin{aligned} |f(x_0)| &= |a_d \prod_{i=1}^d (x_0 - z_i)| = |a_d| \cdot |\xi - x_0| \cdot \prod_{i=1, \dots, d; i \neq i_0} |x_0 - z_i| \\ &\geq |\xi - x_0| \cdot \frac{4}{\sigma_{i_0}} \cdot \prod_{i=1}^d \frac{\sigma_i}{4} > |\xi - x_0| \cdot 2^{-2d-\tau} \cdot 2^{-\Sigma_f}, \end{aligned}$$

where the last inequality uses that  $|z_i| < 1 + \frac{\max a_i}{a_d} < 2^{\tau+1}$  by Cauchy's Bound [18] and, thus,  $\sigma(z_{i_0}) \leq 2^{\tau+2}$ .  $\square$

We next analyze an approximate bisection step.

**Lemma 5.** *Let  $f$  be a polynomial as in (1),  $I = (a, b)$  be an isolating interval for a root  $\xi = z_{i_0}$  of  $f$  and  $s = \text{sign}(f(a))$ . Then, Algorithm 2 applied on  $(f, I, s)$  requires a maximal precision of*

$$\begin{aligned} \rho_0 &:= 2 \log(b-a)^{-1} + 4 \log(d+1) + 4d + 10 + 2(d+1)\tau + 2\Sigma_f \\ &= O(\log(b-a)^{-1} + d\tau + \Sigma_f), \end{aligned}$$

*and its bit complexity is bounded by  $\tilde{O}(d(\log(b-a)^{-1} + d\tau + \Sigma_f))$ .*

PROOF. Consider the three subdivision points  $m_j := a + j \cdot \frac{b-a}{4}$ , where  $1 \leq j \leq 3$ , and an arbitrary real root  $z_i \neq \xi$  of  $f$ . Note that  $|m_j - z_i| > \frac{b-a}{4}$  because the segment from  $m_j$  to  $z_i$  spans at least over a quarter of  $(a, b)$ . Moreover,  $|\xi - m_j| \leq \frac{3}{4}(b-a)$ , and so

$$\sigma_i \leq |\xi - z_i| \leq |\xi - m_j| + |m_j - z_i| \leq \frac{3}{4}(b-a) + |m_j - z_i| \leq 4|m_j - z_i|.$$

It follows that  $m_j$  has a distance to  $z_i$  of at least  $\frac{\sigma_i}{4}$ . Hence, we can apply Lemma 4 to each  $m_j$ , that is, we have  $|f(m_j)| > |\xi - m_j| \cdot 2^{-(2d+\tau+\Sigma_f)}$ . Since the signs of  $f$  at the endpoints of  $I$  are known, it suffices to compute the signs of  $f$  at two of the three subdivision points. For at least two of these points, the distance of  $m_j$  to  $\xi$  is at least  $\frac{b-a}{8}$ , thus, we have  $|f(m_j)| > |b-a| \cdot 2^{-(2d+3+\tau+\Sigma_f)}$  for at least two points. Then, due to Lemma 3, we can use interval arithmetic with a precision  $\rho$  to compute these signs if  $\rho$  satisfies

$$2^{-\rho+2}(d+1)^2 2^{d\tau} \leq (b-a) \cdot 2^{-(2d+3+\tau+\Sigma_f)},$$

which is equivalent to  $\rho \geq \frac{\rho_0}{2}$ . Since we double the precision in each step, we will eventually succeed with a precision smaller than  $\rho_0$ . The bit complexity for an arithmetic operation with precision  $\rho$  is  $\tilde{O}(\rho)$ . At each subdivision point, we have to perform  $O(d)$  arithmetic operations for the computation of  $f(m_j)$ , thus, the costs for these evaluations are bounded by  $\tilde{O}(d\rho)$ . Since we double the precision in each iteration, the total costs are dominated by the last successful evaluation and, thus, we have to perform  $\tilde{O}(d\rho_0) = \tilde{O}(d(\log(b-a)^{-1} + d\tau + \Sigma_f))$  bit operations.  $\square$

We proceed with the analysis of an AQIR step. In order to bound the required precision, we need additional properties of the isolating interval.

*Definition 1.* Let  $f$  be as in (1) and let  $I := (a, b)$  be an isolating interval of a root  $\xi$  of  $f$ . We call  $I$  *normal* if

- $I \subseteq (-2^{\tau+3}, 2^{\tau+3})$ ,
- $|p - z_i| > \frac{\sigma_i}{4}$  for every  $p \in I$  and  $z_i \neq \xi$ , and
- $\min\{|f(a)|, |f(b)|\} \geq 2^{-(32d\tau+2\Sigma_f-5\log(b-a))}$ .

In simple words, a normal isolating interval has a reasonable distance to any other root of  $f$ , and the function value at the endpoints is reasonably large. We will later see that it is possible to get normal intervals by a sequence of approximate bisection steps.

**Lemma 6.** *Let  $f$  be a polynomial as in (1),  $I = (a, b)$  be a normal isolating interval for a root  $\xi = z_{i_0}$  of  $f$  with  $s = \text{sign}(f(a))$ , and let  $N \leq 2^{2(\tau+5\log(b-a))}$ . Then, the AQIR step for  $(f, I, N, s)$  requires a precision of at most*

$$\rho_{\max} := 99d\tau + 4\Sigma_f - 14\log(b-a)$$

and, therefore, its bit complexity is bounded by

$$\tilde{O}(d(d\tau + \Sigma_f - \log(b-a))).$$

Moreover, the returned interval is again normal.

PROOF. We have to distinguish two cases. For  $N > 2$ , we consider the two while-loops in Algorithm 3. In the first loop (Step 5), we evaluate  $N \frac{f(a)}{f(a)-f(b)}$  via interval arithmetic, doubling the precision  $\rho$  until the width of the resulting interval  $J$  is less than or equal to  $1/4$ . The following considerations show that we can achieve this if  $\rho$  fulfills

$$2^{-\rho+1}(d+1)^2 2^{d\tau} \leq \frac{\min(|f(a)|, |f(b)|)}{32N}. \quad (5)$$

W.l.o.g., we assume  $f(a) > 0$ . If  $\rho$  fulfills the above condition, then, due to Lemma 3,  $\mathfrak{B}(N \cdot f(a), \rho)$  is contained within the interval

$$\left[ Nf(a) - \frac{|f(a)|}{32}, Nf(a) + \frac{|f(a)|}{32} \right] = Nf(a) \cdot \left[ 1 - \frac{1}{32N}, 1 + \frac{1}{32N} \right]$$

and  $\mathfrak{B}(f(a) - f(b), \rho)$  is contained within

$$\begin{aligned} & \left[ f(a) - f(b) - \frac{|f(a) - f(b)|}{32N}, f(a) - f(b) + \frac{|f(a) - f(b)|}{32N} \right] \\ & = (f(a) - f(b)) \cdot \left[ 1 - \frac{1}{32N}, 1 + \frac{1}{32N} \right], \end{aligned}$$

where the latter result uses the fact that  $f(a)$  and  $f(b)$  have different signs. It follows that  $\mathfrak{B}(N \frac{f(a)}{f(a)-f(b)}, \rho)$  is contained within the interval  $\frac{Nf(a)}{f(a)-f(b)} \cdot \left[ (1 - \frac{1}{32N}) / (1 + \frac{1}{32N}), (1 + \frac{1}{32N}) / (1 - \frac{1}{32N}) \right]$ , and a simple computation shows that  $N \cdot \left[ (1 - \frac{1}{32N}) / (1 + \frac{1}{32N}), (1 + \frac{1}{32N}) / (1 - \frac{1}{32N}) \right]$  has width less than  $1/4$ . Hence, since  $\frac{f(a)}{f(a)-f(b)}$  has absolute value less than 1,  $\mathfrak{B}(N \frac{f(a)}{f(a)-f(b)}, \rho)$  has width less than  $1/4$  as well. The bound (5) on  $\rho$  also writes as

$$\rho \geq 7 + \log(d+1) + d\tau + \log N + \log \min(|f(a), f(b)|)^{-1}$$

and since we double  $\rho$  in each iteration, computing  $N \frac{f(a)}{f(a)-f(b)}$  via interval arithmetic up to an error of  $1/4$  demands for a precision

$$\begin{aligned} \rho & < 14 + 2\log(d+1) + 2d\tau + 2\log N + 2\log \min(|f(a), f(b)|)^{-1} \\ & < 13d\tau + 2\log N + 2\log \min(|f(a), f(b)|)^{-1}, \end{aligned}$$

Because  $I$  is normal and because of the posed condition on  $N$  we can bound this by

$$\begin{aligned} \rho & < 13d\tau + 4(\tau + 5 - \log(b-a)) + 2(32d\tau + 2\Sigma_f - 5\log(b-a)) \\ & < 73d\tau + 4\Sigma_f - 14\log(b-a) = \rho_{\max}. \end{aligned}$$

We turn to the second while loop of Algorithm 3 (Steps 11-14) where  $f$  is evaluated at the subdivision points  $m^* - \omega, m^* - \frac{7\omega}{8}, \dots, m^* + \omega$  as defined in (2). Since the interval is normal, we can apply Lemma 4 to each of the seven subdivision points. Furthermore, at least six of these points have distance  $\geq \frac{b-a}{16N}$  to the root  $\xi$  and, thus, for these points,  $|f|$  is larger than  $\frac{b-a}{16N} \cdot 2^{-(2d+\tau+\Sigma_f)}$ . Then, according to Lemma 5, it suffices to use a precision  $\rho$  that fulfills

$$2^{-\rho+1}(d+1)^2 2^{d\tau} \leq \frac{b-a}{16N} \cdot 2^{-(2d+\tau+\Sigma_f)}, \text{ or}$$

$$\rho \geq \rho_1 := 2\log(d+1) + (d+1)\tau + 2d + 1 + \Sigma_f + 4 + \log N - \log(b-a).$$

The same argumentation as above then shows that the point evaluation will be performed with a maximal precision of less than

$$\begin{aligned} 2\rho_1 & < 2(10d\tau + \Sigma_f + \log N - \log(b-a)) \\ & \leq 20d\tau + 2\Sigma_f + 4(\tau + 5 - \log(b-a)) - \log(b-a) \\ & \leq 32d\tau + 2\Sigma_f - 5\log(b-a) \end{aligned}$$

which is bounded by  $\rho_{\max}$ . Moreover, at the new endpoints  $a'$  and  $b'$ ,  $|f|$  is at least

$$2^{-2\rho_1} \geq 2^{-(32d\tau+2\Sigma_f-5\log(b-a))} \geq 2^{-(32d\tau+2\Sigma_f-5\log(b'-a'))}$$

which proves that  $I' = (a', b')$  is again normal.

It remains the case of  $N = 2$ , where a bisection step is performed. It is straight-forward to see with Lemma 5 that the required precision is bounded by  $\rho_{\max}$ , and in an analogue way as for the point evaluations for  $N > 2$ , we can see that the resulting interval is again

---

**Algorithm 4** Normalization

---

INPUT:  $f \in \mathbb{R}[t]$  a polynomial as in (1),  $I_1, \dots, I_m$  disjoint isolating intervals,  $s_1, \dots, s_m$  with  $s_k = \text{sign}(f(\min I_k))$

OUTPUT: normal isolating intervals  $J_1, \dots, J_m$  with  $z_k \in I_k \cap J_k$

```
1: procedure NORMALIZE( $f, I_1, \dots, I_m$ )
2:   for  $k=1, \dots, m-1$  do
3:     while  $\min I_{k+1} - \max I_k < 3 \max\{w(I_k), w(I_{k+1})\}$  do
4:       if  $w(I_k) > w(I_{k+1})$ 
5:         then APPROXIMATE_BISECTION( $f, I_k, s_k$ )
6:         else APPROXIMATE_BISECTION( $f, I_{k+1}, s_{k+1}$ )
7:   for  $k=1, \dots, m$  do
8:      $J_k \leftarrow [m(I_k) - \frac{3w(I_k)}{2}, m(I_k) + \frac{3w(I_k)}{2}]$   $\triangleright$  scale by 3
9:   return  $J_1, \dots, J_m$ 
```

---

normal. By the same argument as in Lemma 5, the overall bit complexity of the AQIR step is bounded by

$$\tilde{O}(d\rho_{\max}) = \tilde{O}(d(\tau + \Sigma_f - \log(b-a))). \quad \square$$

## 5. ROOT REFINEMENT

We next analyze the complexity of our original problem: Given a polynomial  $f$  as in (1) and isolating intervals for all its real roots, refine the intervals to a size of at most  $2^{-L}$ . Our refinement method consists of two steps. First, we turn the isolating intervals into normal intervals by applying bisections repeatedly. Second, we call the AQIR method repeatedly on the intervals until each has a width of at most  $2^{-L}$ . Algorithm 5 summarizes our method for root refinement. We remark that depending on the properties of the root isolator used to get initial isolating intervals, the normalization can be skipped; this is for instance the case when using the isolator from [17]. We also emphasize that the normalization is unnecessary for the correctness of the algorithm; its purpose is to prevent the working precision in a single AQIR step of growing too high.

### 5.1 Normalization

The normalization (Algorithm 4) consists of two steps: first, the isolating intervals are refined using approximate bisection until the distance between two consecutive intervals is at least three times larger than the size of the larger of the two involved intervals. This ensures that all points in an isolating interval are reasonably far away from any other root of  $f$ . In the second step, each interval is enlarged by a factor of three, keeping the same midpoint. This ensures that the endpoints are sufficiently far away from any root of  $f$  to prove a lower bound of  $f$  at the endpoints. W.l.o.g., we assume that the input intervals are contained in  $(-2^{\tau+1}, 2^{\tau+1})$  because by Cauchy's bound [18], all roots are contained in that interval, so the leftmost and rightmost intervals can just be cut if necessary.

Obviously, the resulting intervals are still isolating and disjoint from each other. Moreover, they do not become too small during the bisection process:

**Lemma 7.** For  $J_1, \dots, J_m$  as returned by Alg. 4,  $w(J_k) \geq \frac{3}{20}\sigma_k$ .

PROOF. Let  $I_k$  denote one of the isolating interval in Algorithm 4 before scaling, i.e., after the first for-loop. It suffices to show that  $w(I_k) \geq \frac{1}{20}\sigma_k$ . Assume for a contradiction that  $I_k$  has a smaller width. Then, an approximate bisection has been performed on a previous version  $I'_k$  of  $I_k$  of width less than  $\frac{1}{5}\sigma_k$  (because one such step can shrink the interval by a factor of at most 4). This bisection can only happen if one of the neighboring intervals, say  $I'_{k+1}$  has smaller width and the distance  $d := \min I'_{k+1} - \max I'_k$  to that interval is smaller than  $3w(I'_k) < \frac{3}{5}\sigma_k$ . On the other hand, the opposite

statement is also true because

$$d \geq |z_k - z_{k+1}| - w(I'_k) - w(I'_{k+1}) \geq \sigma_k - 2w(I'_k) \geq \frac{3}{5}\sigma_k. \quad \square$$

**Lemma 8.** Algorithm 4 is correct, i.e., returns normal intervals.

PROOF. Let  $J_1, \dots, J_m$  denote the returned intervals, and fix some interval  $J_k$  containing the root  $z_k$  of  $f$ . We have to prove the three properties of Definition 1. The first property is clear because the initial interval are assumed to lie in  $(-2^{\tau+1}, 2^{\tau+1})$ , so they are contained in  $(-2^{\tau+3}, 2^{\tau+3})$  after being scaled by 3.

For the second property, fix some  $x_0 \in J_k$ . We have to show that  $|x_0 - z_i| \geq \frac{\sigma_i}{4}$  for every  $i \neq k$ . Let  $z_{k-1} \in J_{k-1}$ ,  $z_{k+1} \in J_{k+1}$  denote the next real roots on the left and right (if existing). Except for  $i \in \{k-1, k, k+1\}$ ,  $|x_0 - z_i| > \sigma_i$  for any real root  $z_i$ , and also,  $|x_0 - z_i| \geq \frac{\sigma_i}{2}$  for any non-real root by the same argument as in Lemma 4. It thus suffices to consider the distances of  $x_0$  to  $z_{k\pm 1}$ .

Let  $I_1, \dots, I_m$  denote the refined isolating intervals in Algorithm 4 before scaling to  $J_1, \dots, J_m$ . We have that

$$3w(I_k) \leq \min I_{k+1} - \max I_k \leq |z_{k+1} - z_k|.$$

Furthermore,  $|x_0 - z_k| \leq 2w(I_k)$  and, by triangle inequality,

$$\begin{aligned} |x_0 - z_{k+1}| &\geq |z_{k+1} - z_k| - 2w(I_k) \geq |z_{k+1} - z_k| - \frac{2}{3}|z_{k+1} - z_k| \\ &= \frac{1}{3}|z_{k+1} - z_k| \geq \frac{\sigma_{k+1}}{3} > \frac{\sigma_{k+1}}{4}. \end{aligned}$$

The same holds for  $|x_0 - z_{k-1}|$ .

For the third property of Definition 1, let  $e$  be one of the endpoints of  $J_k$ . We have just proved that the distance to every root  $z_i$  except  $z_k$  is at least  $\frac{\sigma_i}{3}$ . By Lemma 7, the distance to  $z_k$  is at least  $\frac{1}{20}\sigma_k$  because  $z_k$  is in the central third of  $J_k$ . With an estimation similar as in the proof of Lemma 4, we obtain:

$$|f(e)| \geq \frac{\sigma_k}{20} \prod_{i \neq k} \frac{\sigma_i}{3} \geq \frac{1}{32} \cdot \frac{1}{4^{d-1}} 2^{-\Sigma_f} \geq 2^{-(2d+\Sigma_f+3)},$$

and  $2^{-(2d+\Sigma_f+3)} \geq 2^{-(24d\tau+2\Sigma_f-5\log(b-a))}$  because  $\log(b-a) \leq \tau+3 \leq 2d\tau$  and  $-\Sigma_f \leq d\tau+1 < 2d\tau$ .  $\square$

**Lemma 9.** Algorithm 4 has a complexity of

$$\tilde{O}(d(\tau d + \Sigma_f)^2)$$

PROOF. As a direct consequence of Lemma 7, each interval  $I_k$  is only bisected  $O(\tau + \log(\sigma_k)^{-1})$  many times because each starting interval is assumed to be contained in  $(-2^{\tau+1}, 2^{\tau+1})$ . So the total number of bisections adds up to  $O(d\tau + \Sigma_f)$  considering all roots of  $f$ . Also, the size of the isolating interval  $I_k$  is lower bounded by  $\frac{3}{20} \cdot \sigma_k = 2^{-O(\Sigma_f + d\tau)}$ , so that one approximate bisection step has a complexity of  $\tilde{O}(d(\tau d + \Sigma_f))$  due to Lemma 5.  $\square$

### 5.2 The AQIR sequence

It remains to bound the cost of the calls of AQIR. We mostly follow the argumentation from [12], mostly referring to that article for technical proofs. We introduce the following convenient notation:

*Definition 2.* Let  $I_0 := I$  be a normal isolating interval for some real root  $\xi$  of  $f$ ,  $N_0 := 4$  and  $s := \text{sign}(\min I_0)$ . The AQIR sequence  $(S_0, S_1, \dots, S_{v_\xi})$  is defined by

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

where  $v_\xi$  is the first index such that the interval  $I_{v_\xi}$  has width at most  $2^{-L}$ . We say that  $S_i \xrightarrow{\text{AQIR}} S_{i+1}$  succeeds if  $\text{AQIR}(f, I_i, N_i, s)$  succeeds, and that  $S_i \xrightarrow{\text{AQIR}} S_{i+1}$  fails otherwise.

---

**Algorithm 5** Root Refinement
 

---

INPUT:  $f = \sum a_i x^i \in \mathbb{R}[t]$  a polynomial as in (1), isolating intervals  $I_1, \dots, I_m$  for the real roots of  $f$ ,  $L \in \mathbb{Z}$

OUTPUT: isolating intervals  $J_1, \dots, J_m$  with  $w(J_k) \leq 2^{-L}$

```

1: procedure ROOT_REFINEMENT( $f, L, I_1, \dots, I_m$ )
2:    $s_k := \text{sign}(a_d) \cdot (-1)^{m-k+1} \triangleright s_k = \text{sign}(f(\min I_k))$ 
3:    $J_1, \dots, J_m \leftarrow \text{NORMALIZE}(f, I_1, \dots, I_m)$ 
4:   for  $k=1, \dots, m$  do
5:      $N \leftarrow 4$ 
6:     while  $w(J_k) > 2^{-L}$  do  $(J_k, N) \leftarrow \text{AQIR}(f, J_k, N, s_k)$ 
7:   return  $J_1, \dots, J_m$ 

```

---

As in [12], we divide the QIR sequence into two parts according to the following definition.

*Definition 3.* Let  $\xi$  be a root of  $f$ . Define

$$C_\xi := \frac{|f'(\xi)|}{8ed^3 2^\tau \max\{|\xi|, 1\}^{d-1}},$$

where  $e \approx 2.71\dots$  denotes the Eulerian number. For  $(S_0, \dots, S_{v_\xi})$  the QIR sequence of  $\xi$ , define  $k$  as the minimal index such that  $S_k = (I_k, N_k) \xrightarrow{\text{AQIR}} S_{k+1}$  succeeds and  $w(I_k) \leq C_\xi$ . We call  $(S_0, \dots, S_k)$  *linear sequence* and  $(S_k, \dots, S_{v_\xi})$  *quadratic sequence* of  $\xi$

Note that  $C_\xi = \frac{1}{4} M_\xi$  as defined in [12], and that the linear sequence was called *initial sequence* therein. We renamed it to avoid confusion with the initial normalization phase in our variant.

**Quadratic convergence.** We start by justifying the name ‘‘quadratic sequence’’. Indeed, it turns out that all but one AQIR step in the quadratic sequence are successful, hence,  $N$  is squared in (almost) every step and therefore, the refinement factor of the interval is doubled in (almost) every step. The proof is mostly analogous to [12]. The following bound follows from considering the Taylor expansion of  $f$  at  $\xi$  in the expression for  $m$  (see also Appendix C).

**Lemma 10.** [12, Thm. 4.8] *Let  $(a, b)$  be isolating for  $\xi$  with width  $\delta < C_\xi$  and  $m$  as in Lemma 2. Then,  $|m - \xi| \leq \frac{\delta^2}{8C_\xi}$ .*

**Corollary 1.** *Let  $I_j$  be an isolating interval for  $\xi$  of width  $\delta_j \leq \frac{C_\xi}{N_j}$ . Then, each call of the AQIR sequence*

$$(I_j, N_j) \xrightarrow{\text{AQIR}} (I_{j+1}, N_{j+1}) \xrightarrow{\text{AQIR}} \dots$$

*succeeds.*

PROOF. We use induction on  $i$ . Assume that the first  $i$  AQIR calls succeed. Then, another simple induction shows that  $\delta_{j+i} := w(I_{j+i}) \leq \frac{N_j \delta_j}{N_{j+i}} < \frac{C_\xi}{N_{j+i}}$ , where we use that  $N_{j+i} = N_{j+i-1}^2$ . Then, according to Lemma 10, we have that

$$|m - \xi| \leq \delta_{j+i}^2 \frac{1}{8C_\xi} \leq \delta_{j+i} \frac{C_\xi}{N_{j+i}} \frac{1}{8C_\xi} = \frac{1}{8} \frac{\delta_{j+i}}{N_{j+i}},$$

with  $m$  as above. By Lemma 2, the AQIR call succeeds.  $\square$

**Corollary 2.** [12, Cor. 4.10] *In the quadratic sequence, there is at most one failing AQIR call. (see also Appendix C for a proof)*

**Cost of the linear sequence.** We bound the costs of refining the isolating interval of  $\xi$  to size  $C_\xi$  with AQIR. We first show that, on average, the AQIR sequence refines by a factor two in every second step. This shows in particular that refining using AQIR is at most a factor of two worse than refining using approximate bisection.

**Lemma 11.** *Let  $(S_0, \dots, S_\ell)$  denote an arbitrary prefix of the AQIR sequence for  $\xi$ , starting with the isolating interval  $I_0$  of width  $\delta$ . Then, the width of  $I_\ell$  is not larger than  $\delta 2^{-(\ell-1)/2}$ .*

PROOF. Consider a subsequence  $(S_i, \dots, S_{i+j})$  of  $(S_0, \dots, S_\ell)$  such that  $S_i \xrightarrow{\text{AQIR}} S_{i+1}$  is successful, but any other step in the subsequence fails. Because there are  $j$  steps in total, and thus  $j-1$  consecutive failing steps, the successful step must have used a  $N$  with  $N \geq 2^{2^{j-1}}$ . Because  $2^{j-1} \geq \frac{j}{2}$ , it holds that

$$w(I_{i+j}) \leq \frac{w(I_i)}{N} \leq w(I_{i+j}) 2^{-2^{j-1}} \leq w(I_{i+j}) 2^{-j/2}.$$

Repeating the argument for maximal subsequences of this form, we get that either  $w(I_\ell) \leq w(I_0) 2^{-\ell/2}$  if the sequence starts with a successful step, or  $w(I_\ell) \leq w(I_0) 2^{-(\ell-1)/2}$  otherwise, because the second step must be successful in this case.  $\square$

We want to apply Lemma 6 to bound the bit complexity of a single AQIR step. The following lemma shows that the condition on  $N$  from Lemma 6 is always met in the AQIR sequence.

**Lemma 12.** *Let  $(I_j, N_j) \xrightarrow{\text{AQIR}} (I_{j+1}, N_{j+1})$  be a call in an AQIR sequence and  $I_j := (a, b)$ . Then,  $N_j \leq 2^{2(\tau+5-\log(b-a))}$ .*

PROOF. We do induction on  $j$ . Note that  $I_0 \subset (-2^{\tau+3}, 2^{\tau+3})$  by normality, hence  $b-a \leq 2^{\tau+4}$ . It follows that  $2^{2(\tau+5-\log(b-a))} \geq 4 = N_0$ . Assume that the statement is true for  $j-1$ . If the previous step  $(I_{j-1}, N_{j-1}) \xrightarrow{\text{AQIR}} (I_j, N_j)$  is failing, then  $N_j = \sqrt{N_{j-1}}$  and the isolating interval remains unchanged, so the statement is trivially correct. If the step is successful, then it holds that  $(b-a) \leq \frac{2^{\tau+4}}{\sqrt{N_j}}$ .

By rearranging terms, we get that  $N_j \leq 2^{2(\tau+4-\log(b-a))}$ .  $\square$

It follows inductively that the conditions of Lemma 6 are met for each call in the AQIR sequence because  $I_0$  is normal by construction. Therefore, the linear sequence for a root  $\xi$  of  $f$  is computed with a bit complexity of

$$\tilde{O}((\tau + \log(C_\xi^{-1}))d(\log(C_\xi^{-1}) + d\tau + \Sigma_f)) \quad (6)$$

because  $O(\tau + \log(C_\xi^{-1}))$  steps are necessary to refine the interval to a size smaller than  $C_\xi$  by Lemma 11, and the bit complexity is bounded by  $\tilde{O}(d(\log(C_\xi^{-1}) + d\tau + \Sigma_f))$  with Lemma 6. It remains to bound  $\log(C_\xi^{-1})$ ; we do so by bounding the sum of all  $\log(C_\xi)^{-1}$  with the following lemma.

**Lemma 13.**  $\sum_{i=1}^m \log(C_{z_i})^{-1} = O(d(\tau + \log d) + R)$

PROOF. We can write the sum as

$$\sum_{i=1}^m \log(C_{z_i})^{-1} \leq O(d(\tau + \log d)) + d \cdot \log \text{Mea}(f) - \log \left| \prod_{i=1}^n f'(z_i) \right|$$

where  $\text{Mea}(f)$  is the Mahler measure of  $f$  (see [12, Thm 4.5] for a more detailed calculation). It is known that  $\log \text{Mea}(f) = O(\tau + \log d)$ . For the last summand, we use the relation  $\text{res}(f, f') = a_d^{d-1} \prod_{i=1}^n f'(z_i)$ ; see [2, Thm.4.16] [18, Thm.6.15]. It follows that

$$-\log \left| \prod_{i=1}^n f'(z_i) \right| = \log |a_d^{d-1}| - \log \text{res}(f, f') \leq (d-1)\tau + R. \quad \square$$

When we apply Lemma 13 into (6), we obtain a bound that depends on  $d$ ,  $\tau$ ,  $\Sigma_f$ , and  $R$ . The next result shows that  $\Sigma_f$  is bounded by  $\tilde{O}(d\tau + R)$ . The proof is only sketched for brevity; a complete proof is given in Appendix A.

**Theorem 1.**  $\Sigma_f \in O(d(\tau + \log d) + R)$ .

PROOF. The product of all  $\sigma_i$ 's is a product of root differences, and corresponds to the nearest neighbor graph of the roots of  $f$  [8]. We would like to apply the Davenport-Mahler bound [9] on this product, but the preconditions of it are not satisfied. However, by exploiting simple properties of the nearest neighbor graph, we can define another root product  $P$  such that  $2^{-\Sigma_f} \geq 2^{-5d(\tau+1)}P^6$  and such that the Davenport-Mahler bound is applicable on  $P$ . This yields that  $\log \frac{1}{P} = O(d(\tau + \log d) + R)$ .  $\square$

**Lemma 14.** *The linear sequences for all real roots are computed within a total bit complexity of*

$$\tilde{O}(d(d\tau + R)^2)$$

PROOF. The total cost of all linear sequences is bounded by

$$\tilde{O}\left(\sum_{i=1}^d (\tau + \log(C_{z_i}^{-1}))d(\log(C_{z_i}^{-1}) + d\tau + \Sigma_f)\right).$$

Using Theorem 1 and rearranging terms, we obtain

$$= \tilde{O}(d^2\tau(d\tau + R) + d(d\tau + R)\sum \log(C_{z_i}^{-1}) + d(\sum \log(C_{z_i}^{-1}))^2)$$

which equals  $\tilde{O}(d(d\tau + R)^2)$  with Lemma 13.  $\square$

**Cost of the quadratic sequence.** Let us fix some root  $\xi$  of  $f$ . Its quadratic sequence consists of at most  $1 + \log L$  steps, because  $N$  is squared in every step (except for at most one failing step) and the sequence stops as soon as the interval is smaller than  $2^{-L}$ . Since we ignore logarithmic factors, it is enough to bound the costs of one QIR step in the sequence. Clearly, since the interval is not smaller than  $2^{-L}$  in such a step, we have that  $\log(b-a)^{-1} \leq L$ . Therefore, the required precision is bounded by  $O(L + d\tau + \Sigma_f)$ . It follows that an AQIR step performs up to  $\tilde{O}(d(L + d\tau + \Sigma_f))$  bit operations.

**Lemma 15.** *The quadratic sequences for one real root is computed within a bit complexity of*

$$\tilde{O}(d(L + d\tau + \Sigma_f)).$$

**Total cost.** We have everything together to prove the main result

**Theorem 2.** *Algorithm 5 performs root refinement within*

$$\tilde{O}(d(d\tau + R)^2 + dL)$$

*bit operations for a single real root of  $f$ , and within*

$$\tilde{O}(d(d\tau + R)^2 + d^2L)$$

*for all real roots. The coefficients of  $f$  need to be approximated to  $O(L + d\tau + \Sigma_f)$  bits after the binary point.*

PROOF. We concentrate on the bound on all real roots; the case of a single roots follows easily. By Lemma 9, the normalization requires  $\tilde{O}(d(d\tau + \Sigma_f)^2) = \tilde{O}(d(d\tau + R)^2)$  bit operations. The linear subsequences of the AQIR sequence are computed in the same time by Lemma 14. The quadratic subsequences are computed with  $\tilde{O}(d^2L + d^3\tau + d^2\Sigma_f)$  bit operations by Lemma 15; the latter two terms are both dominated by  $d(d\tau + R)^2$  which yields the complexity bound. The maximal number of required bits follows from Lemma 6 because the maximal required precision in any AQIR step is bounded by  $O(L + d\tau + \Sigma_f)$ .  $\square$

We remark without proof that, with little extra effort, the bound for a single root can be slightly improved to  $\tilde{O}(d(d\tau + \Sigma_f)^2 + dL)$ . For integer polynomials,  $\text{res}(f, f')$  is an integer and consequently  $R < 0$ . This improves the bound from [12] by a factor of  $d$ .

**Corollary 3.** *If  $f$  is a polynomial with integer coefficients, the bit complexity of Algorithm 5 is bounded by*

$$\tilde{O}(d^3\tau^2 + d^2L).$$

## 6. REFERENCES

- [1] J. Abbott. Quadratic Interval Refinement for Real Roots. Poster presented at the 2006 International Symposium on Symbolic and Algebraic Computation (ISSAC 2006), 2006.
- [2] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry*. Springer, 2nd edition, 2006.
- [3] E. Berberich, P. Emeliyanenko, and M. Sagraloff. An elimination method for solving bivariate polynomial systems: Eliminating the usual drawbacks. In *Workshop on Algorithm Engineering & Experiments (ALENEX)*, 2011. to appear.
- [4] E. Berberich, M. Hemmer, and M. Kerber. A generic algebraic kernel for non-linear geometric applications. Research report 7274, INRIA, 2010.
- [5] J. Bus and T.J.Dekker. Two efficient algorithms with guaranteed convergence for finding a zero of a function. *ACM Trans. on Math. Software*, 1(4):330–345, 1975.
- [6] J. Cheng, S. Lazard, L. Peñaranda, M. Pouget, F. Rouillier, and E. Tsigaridas. On the topology of real algebraic plane curves. *Mathematics in Computer Science*, 4:113–137, 2010.
- [7] G. E. Collins and A. G. Akritas. Polynomial Real Root Isolation Using Descartes' Rule of Signs. In *Proc. of the 3rd ACM Symp. on Symbolic and Algebraic Computation (SYMSAC 1976)*, pages 272–275. ACM Press, 1976.
- [8] D.Eppstein, M.S.Paterson, and F.F.Yao. On nearest-neighbor graphs. *Discrete and Computational Geometry*, 17(3):263–282, 1997.
- [9] Z. Du, V. Sharma, and C. Yap. Amortized bound for root isolation via Sturm sequences. In *Symbolic-Numeric Computation*, Trends in Mathematics, pages 113–129. Birkhäuser Basel, 2007.
- [10] A. Eigenwillig, M. Kerber, and N. Wolpert. Fast and exact geometric analysis of real algebraic plane curves. In *Proc. of the 2007 Intern. Symp. on Symbolic and Algebraic Computation (ISSAC 2007)*, pages 151–158, 2007.
- [11] A. Eigenwillig, L. Kettner, W. Krandick, K. Mehlhorn, S. Schmitt, and N. Wolpert. A Descartes algorithm for polynomials with bit-stream coefficients. In *8th International Workshop on Computer Algebra in Scientific Computing (CASC 2005)*, volume 3718 of LNCS, pages 138–149, 2005.
- [12] M. Kerber. On the complexity of reliable root approximation. In *11th International Workshop on Computer Algebra in Scientific Computing (CASC 2009)*, volume 5743 of LNCS, pages 166–167. Springer, 2009.
- [13] K. Mehlhorn, R. Osbild, and M. Sagraloff. A general approach to the analysis of controlled perturbation algorithms. *CGTA*, 2011. to appear; for a draft, see <http://www.mpi-inf.mpg.de/~msagralo/cpgeneral.pdf>.
- [14] V. Y. Pan. Optimal and nearly optimal algorithms for approximating polynomial zeros. *Comput. Math. Appl.*, 31(12):97–138, 1996.
- [15] V. Y. Pan. Solving a polynomial equation: Some history and recent progress. *SIAM Review*, 39(2):187–220, 1997.
- [16] F. Rouillier and P. Zimmermann. Efficient isolation of polynomial's real roots. *J. Comput. Appl. Math.*, 162(1):33–50, 2004.
- [17] M. Sagraloff. On the complexity of real root isolation. arXiv:1011.0344v1, 2010.
- [18] C. K. Yap. *Fundamental Problems in Algorithmic Algebra*. Oxford University Press, 2000.

## APPENDIX

### A. BOUNDING THE SEPARATION

We give a complete proof of Theorem 1. First, we repeat the Davenport-Mahler theorem as given in [9]; see also Eigenwillig's PhD thesis ("Real Root Isolation for Exact and Approximate Polynomials Using Descartes' Rule of Signs", Saarland University 2008) for a more general version.

**Theorem 3 (Davenport-Mahler bound).** *Let  $g \in \mathbb{C}[t]$  square-free of degree  $d$  and let  $G = (V, E)$  be a directed graph on the roots of  $g$  such that:*

- $G$  is acyclic,
- for every edge  $(\alpha, \beta) \in E$ , it holds  $|\alpha| \leq |\beta|$ , and
- the in-degree of any node is at most 1.

In this situation

$$\prod_{(\alpha, \beta) \in E} |\alpha - \beta| \geq \frac{\sqrt{|\text{res}(g, g')|}}{\sqrt{|\text{lcf}(g)|} \text{Mea}(g)^{d-1}} \cdot \left(\frac{\sqrt{3}}{d}\right)^{\#E} \cdot \left(\frac{1}{d}\right)^{d/2}.$$

**Theorem 1.**

$$\Sigma_f \in O(d(\tau + \log d) + R)$$

PROOF. As before, let  $z_1, \dots, z_n$  denote the roots of  $f$ , and let  $B \geq 1$  denote a bound for the maximal absolute value of a root. Observe that, when the  $z$ 's are considered as vertices in the complex plane, each  $\sigma_i$  is given by the length of an edge connecting  $z_i$  to its nearest neighbor. This induces a directed graph on the vertices, which is known as the *nearest neighbor graph* [8] (if a root has more than one nearest neighbor, we pick the one with highest index). Let  $E_0$  denote the edge set of this nearest neighbor graph. We can rewrite:

$$\prod_{i=1}^d \sigma_i = \prod_{(z_i, z_j) \in E_0} |z_j - z_i|$$

Our goal is to apply the Davenport-Mahler bound on this product. However, the nearest-neighbor graph does not satisfy any of the required properties in general. We will transform the edge set  $E_0$  into another edge set  $E_3$  that satisfies the requirements of the Davenport-Mahler theorem, and we will relate the root product of  $E_0$  with the root product of  $E_3$ .

Note that a direct property of nearest neighbor graphs is that all cycles have length 2 [8]. In the first step, we remove one edge of every cycle:

$$E_1 := \{(z_i, z_j) \in E_0 \mid i < j \vee (z_j, z_i) \notin E_0\}$$

This removes at most every second edge, and for every removed edge, there is some edge in  $E_1$  with the same length. Since every root difference is bounded by  $2B$  from above, we can bound

$$(2B)^d \prod_{(z_i, z_j) \in E_0} |z_j - z_i| \geq \prod_{(z_i, z_j) \in E_1} |z_j - z_i|^2.$$

In the next step, we re-direct the edges in  $E_1$  in order to satisfy the second condition of the Davenport-Mahler bound

$$E_2 := \{(z_i, z_j) \mid ((z_i, z_j) \in E_1 \vee (z_j, z_i) \in E_1) \wedge (|z_i| < |z_j| \vee (|z_i| = |z_j| \wedge i < j))\}$$

In simple words, every edge points to the root with greater absolute value. Note that  $E_2$  does not contain any cycles, because the absolute value of a root is non-decreasing on any path, and if it remains

the same, the index increases, thus no vertex can be visited twice on such a path. Since the only difference between  $E_1$  and  $E_2$  is the orientation of edges, we have

$$\prod_{(z_i, z_j) \in E_1} |z_j - z_i| = \prod_{(z_i, z_j) \in E_2} |z_j - z_i|$$

Finally, we need to ensure the last condition of the Davenport-Mahler bound, namely that each vertex has in-degree at most 1. For that, if several edges point to some  $z_j$ , we throw away all of them except the shortest one (in the definition, if the shortest edge is not unique, we keep the one with the maximal index):

$$E_3 := \{(z_i, z_j) \in E_2 \mid \forall (z_k, z_j) \in E_2 : |z_k - z_j| > |z_i - z_j| \vee (|z_k - z_j| = |z_i - z_j| \wedge k \leq i)\}$$

Another basic property of the nearest neighbor graph is that two edges that meet in a vertex must form an angle of at least  $60^\circ$ . It follows that the degree of every vertex is bounded by 6. Since  $E_2$  is a subgraph of the nearest neighbor graph, possibly with some edges flipped, the degree of every vertex is still bounded by 6. Since all edges in  $E_2$  point to the root with greater absolute value, it can be easily seen that the in-degree of  $z_j$  is even bounded by 3. So,  $E_3$  contains at least  $\frac{E_2}{3}$  many edges. Since we always keep a smallest edge pointing to a  $z_j$ , we can bound

$$(2B)^{2d} \prod_{(z_i, z_j) \in E_2} |z_j - z_i| \geq \prod_{(z_i, z_j) \in E_3} |z_j - z_i|^3.$$

Putting everything together, we have that

$$\prod_{(z_i, z_j) \in E_0} |z_j - z_i| \geq (2B)^{-5d} \left( \prod_{(z_i, z_j) \in E_3} |z_j - z_i| \right)^6.$$

$E_3$  meets all prerequisites of the Davenport-Mahler bound and we can thus bound

$$\begin{aligned} \prod_{i=1}^d \sigma_i &= \prod_{(z_i, z_j) \in E_0} |z_j - z_i| \\ &\geq (2B)^{-5d} \left( \prod_{(z_i, z_j) \in E_3} |z_j - z_i| \right)^6 \\ &\geq (2B)^{-5d} \left( \frac{\sqrt{|\text{res}(f, f')|}}{\sqrt{|\text{lcf}(f)|} \text{Mea}(f)^{d-1}} \cdot \left(\frac{\sqrt{3}}{g}\right)^{\#E_3} \cdot \left(\frac{1}{d}\right)^{d/2} \right)^6 \\ &\geq (2B)^{-5d} \left( \frac{\sqrt{|\text{res}(f, f')|}}{\sqrt{|\text{lcf}(f)|} \text{Mea}(f)^{d-1}} \cdot \left(\frac{1}{d}\right)^{2d} \right)^6 \end{aligned}$$

Taking the inverse on both sides and applying the logarithm, we get

$$\begin{aligned} \Sigma_f &\leq 5d \log 2B + 3 \log \text{lcf}(f) + 6(d-1) \log \text{Mea}(f) + 12d \log d + 6 \cdot R \\ &= O(d(\tau + \log d) + R), \end{aligned}$$

exploiting that  $B \in O(2^\tau)$  and  $\log \text{Mea}(f) \in O(\tau + \log n)$ .  $\square$

## B. ERROR ANALYSIS OF INTERVAL ARITHMETIC

We restate Lemma 3 and provide its proof.

**Lemma 3.** Let  $f$  be a polynomial as in (1),  $c \in \mathbb{R}$  with  $|c| \leq 2^\tau$ , and  $\rho \in \mathbb{N}$ . Then,

$$|f(c) - \text{down}(f(c), \rho)| \leq 2^{-\rho+1} (d+1)^2 2^{\tau d} \quad (7)$$

$$|f(c) - \text{up}(f(c), \rho)| \leq 2^{-\rho+1} (d+1)^2 2^{\tau d} \quad (8)$$

In particular,  $\mathfrak{B}(f(c), \rho)$  has a width of at most  $2^{-\rho+2} (d+1)^2 2^{\tau d}$ .

**PROOF.** We do induction on  $d$ . The statement is clearly true for  $d = 0$ . For  $d > 0$ , we write  $f(c) = a_0 + cg(c)$  with  $a_0 \in \mathbb{R}$  the constant coefficient of  $f$  and  $g$  of degree  $d-1$ . Note that, for any integer  $x$ ,  $|\text{down}(x, \rho) - x| < 2^{-\rho}$ , same for up. Therefore, we can bound as follows (again, leaving  $\rho$  out for simplicity):

$$\begin{aligned} |f(c) - \text{down}(f(c))| &= |a_0 + cg(c) - \text{down}(a_0 + cg(c))| \\ &= |a_0 + cg(c) - \text{down}(a_0) - \text{down}(cg(c))| \\ &\leq |cg(c) - \text{down}(cg(c))| + 2^{-\rho} \end{aligned}$$

Note that  $\text{down}(c \cdot g(c)) = \text{down}(H_1(c) \cdot H_2(g(c)))$  where  $H_{1,2} = \text{down}$  or  $H_{1,2} = \text{up}$ . Moreover, we can write  $H_1(c) = c - \varepsilon$  with  $|\varepsilon| < 2^{-\rho}$ . Therefore, we can rearrange

$$\begin{aligned} &|cg(c) - \text{down}(cg(c))| + 2^{-\rho} \\ &\leq |cg(c) - (c - \varepsilon) \cdot H_2(g(c))| + 2^{-\rho+1} \\ &\leq |cg(c) - c \cdot H_2(g(c))| + |\varepsilon| \cdot |H_2(g(c))| + 2^{-\rho+1} \\ &\leq |c| \cdot |g(c) - H_2(g(c))| + 2^{-\rho} |H_2(g(c))| + 2^{-\rho+1} \end{aligned}$$

By a simple inductive proof on the degree, we can show that both  $|\text{up}(g(c))|$  and  $|\text{down}(g(c))|$  are bounded by  $d2^{d\tau}$ . Using that and the induction hypothesis yields

$$\begin{aligned} &|c| \cdot |g(c) - h(g(c))| + 2^{-\rho} |H_2(g(c))| + 2^{-\rho+1} \\ &< 2^\tau 2^{-\rho+1} d^2 2^{\tau(d-1)} + 2^{-\rho} d 2^{\tau d} + 2^{-\rho+1} \\ &\leq 2^{-\rho+1} (d^2 + d + 1) 2^{\tau d} \leq 2^{-\rho+1} (d+1)^2 2^{\tau d} \end{aligned}$$

The bound for  $|f(c) - \text{up}(f(c))|$  follows in the same way.  $\square$

## C. DETAILS ON QUADRATIC CONVERGENCE

For convenience, we repeat some proofs of [12] adapted to our notation. Recall from Definition 3 that

$$C_\xi := \frac{|f'(\xi)|}{8ed^3 2^\tau \max\{|\xi|, 1\}^{d-1}}.$$

We need one additional lemma to prove some properties of  $C_\xi$ .

**Lemma 16.** Let  $\xi \in \mathbb{C}$  be a root of  $f$ .

1.  $0 < C_\xi \leq \frac{1}{d}$
2. Let  $\mu \in \mathbb{C}$  be such that  $|\xi - \mu| < C_\xi$ . Then

$$C_\xi < \frac{|f'(\xi)|}{8|f''(\mu)|}.$$

**PROOF.** By a straight-forward estimation, we can bound  $|f'(\xi)|$  from above by  $d^2 2^\tau \max\{|\xi|, 1\}^{d-1}$ , which proves the first claim.

For the second claim, we bound  $|f''(\mu)|$  from above:

$$\begin{aligned} |f''(\mu)| &= \left| \sum_{i=2}^d i(i-1) a_i \mu^{i-2} \right| \leq d^2 2^\tau \sum_{i=0}^{d-2} \max\{|\mu|, 1\}^i \\ &\leq d^2 2^\tau \sum_{i=0}^{d-2} \left( (1 + C_\xi) \max\{|\xi|, 1\} \right)^i \\ &\leq d^3 2^\tau (1 + C_\xi)^{d-2} \max\{|\xi|, 1\}^{d-2} \\ &< d^3 2^\tau \underbrace{\left(1 + \frac{1}{d}\right)^d}_{< e} \max\{|\xi|, 1\}^{d-1} \quad \square \end{aligned}$$

**Lemma 10.** Let  $(a, b)$  an isolating interval for  $\xi$  of width  $\delta < C_\xi$ . Then  $|m - \xi| < \frac{\delta^2}{8C_\xi}$ .

**PROOF.** We consider the Taylor expansion of  $f$  at  $\xi$ . For a given  $x \in (a, b)$ , we have

$$f(x) = f'(\xi)(x - \xi) + \frac{1}{2} f''(\xi)(x - \xi)^2$$

with some  $\tilde{\xi} \in [x, \xi]$  or  $[\xi, x]$ . Thus, we can simplify

$$\begin{aligned} |m - \xi| &= \left| \frac{f(b)(a - \xi) - f(a)(b - \xi)}{f(b) - f(a)} \right| \\ &= \left| \frac{\frac{1}{2} (f''(\tilde{\xi}_1)(b - \xi)^2 (a - \xi) - f''(\tilde{\xi}_2)(a - \xi)^2 (b - \xi))}{f(b) - f(a)} \right| \\ &\leq \frac{1}{2} |b - \xi| |a - \xi| \cdot \frac{|f''(\tilde{\xi}_1)|(b - \xi) + |f''(\tilde{\xi}_2)|(a - \xi)}{|f(b) - f(a)|} \\ &\leq \frac{\delta^2 \max\{|f''(\tilde{\xi}_1)|, |f''(\tilde{\xi}_2)|\}}{2|f'(v)|} \end{aligned}$$

for some  $v \in (a, b)$ . The Taylor expansion of  $f'$  yields  $f'(v) = f'(\xi) + f''(\tilde{v})(v - \xi)$  with  $\tilde{v} \in (a, b)$ . Since  $\delta \leq C_\xi$ , it follows with Lemma 16

$$|f''(\tilde{v})(v - \xi)| \leq |f''(\tilde{v})| C_\xi \leq \frac{1}{8} |f'(\xi)|.$$

Therefore  $|f'(v)| > \frac{7}{8} |f'(\xi)| > \frac{1}{2} |f'(\xi)|$ , and it follows again with Lemma 16 that

$$\begin{aligned} |m - \xi| &\leq \frac{\delta^2 \max\{|f''(\tilde{\xi}_1)|, |f''(\tilde{\xi}_2)|\}}{|f'(\xi)|} \\ &\leq \frac{\delta^2}{8 \frac{|f'(\xi)|}{8 \max\{|f''(\tilde{\xi}_1)|, |f''(\tilde{\xi}_2)|\}}} < \frac{\delta^2}{8C_\xi} \quad \square \end{aligned}$$

**Corollary 2.** In the quadratic sequence, there is at most one failing AQIR call.

**PROOF.** Let  $(I_i, N_i) \xrightarrow{\text{AQIR}} (I_{i+1}, N_{i+1})$  be the first failing AQIR call in the quadratic sequence. Since the quadratic sequence starts with a successful AQIR call, the predecessor  $(I_{i-1}, N_{i-1}) \xrightarrow{\text{AQIR}} (I_i, N_i)$  is also part of quadratic sequence, and succeeds. Thus we have the sequence

$$(I_{i-1}, N_{i-1}) \xrightarrow{\text{Success AQIR}} (I_i, N_i) \xrightarrow{\text{Fail AQIR}} (I_{i+1}, N_{i+1})$$

One observes easily that  $w(I_{i+1}) = w(I_i) = \frac{w(I_{i-1})}{N_{i-1}} \leq \frac{C_\alpha}{N_{i-1}}$ , and  $N_{i+1} = \sqrt{N_i} = \sqrt{N_{i-1}^2} = N_{i-1}$ . By Corollary 1, all further AQIR calls succeed.  $\square$