On the Complexity of Reliable Root Approximation

Michael Kerber
Max-Planck-Institut für Informatik
66123 Saarbrücken, Germany
mkerber@mpi-inf.mpg.de

This is an author-prepared version of the article. The original publication is available at www.springerlink.com

Abstract

This work addresses the problem of computing a certified ϵ-approximation of all real roots of a square-free integer polynomial. We proof an upper bound for its bit complexity, by analyzing an algorithm that first computes isolating intervals for the roots, and subsequently refines them using Abbott's Quadratic Interval Refinement method. We exploit the eventual quadratic convergence of the method. The threshold for an interval width with guaranteed quadratic convergence speed is bounded by relating it to well-known algebraic quantities.

1 Introduction

Computing the roots of a univariate polynomial is one of the most prominent problems in Computer Algebra. For the case that only real roots are of interest, several subdivision approaches, based on Descartes' rule of sign or on Sturm's Theorem have been introduced [6, 14]. Their output consists of a set of disjoint intervals, each containing exactly one root of the polynomial, and vice versa, each root is contained in one of the intervals; they are also called *isolating intervals*. These subdivision solvers constitute a popular method for root finding, primarily as they return a certified output (no root is lost, no interval contains several roots). Also, they are relatively easy to implement, and have shown good practical performance. Real root solving is a cornerstone, for instance, for the computation of Cylindrical Algebraic Decomposition [4], for related problems such as topology computation [11, 8] and arrangement computation [10], and many more.

In this work, we will investigate the cost of computing isolating intervals, and subsequently refining them until their width falls below ϵ. An equivalent description is to approximate all roots to a precision of ϵ. It should not be surprising that this problem frequently appears in concrete applications – for instance, when comparing the roots of two polynomials, or when evaluating the sign of an algebraic expression that depends on a root of a polynomial.

While the (worst-case) complexity of the root isolation process has been studied extensively for various isolation methods [9, 12, 16], similar results seem not to be available yet for the subsequent refinement process. Our work will provide a complexity analysis with the following main result. Let $f := \sum_{i=0}^{p} a_i x^i \in \mathbb{Z}[x]$ be a polynomial of degree p, with simple roots and $|a_i| < 2^\sigma$ for each coefficient a_i. For $\epsilon > 0$, computing isolating intervals of width at most ϵ for all roots requires in the worst-case

$$\tilde{O}(p^4\sigma^2 + p^3 \log \epsilon^{-1}), \tag{1}$$

bit operations, where $\tilde{O}$ means that logarithmic factors in p and σ are neglected.

We achieve our bound by analyzing the *Quadratic Interval Refinement (qir)* method to refine isolating intervals, introduced by Abbott [1]. This method can be considered as a hybrid of bisection and (an interval version of) the secant method. We will discuss the algorithm in detail in Section 3. As Abbott has already pointed out, the method initially behaves like naive bisection (linear convergence), but once the interval falls below a certain width, the number of newly obtained bits is doubled in every step (which basically means quadratic convergence). In our analysis, we split the sequence of qir steps into an initial sequence where we assume bisections, and a quadratic sequence where the root is rapidly approximated. We will show that the sum of the cost of all initial sequences is bounded by the first summand of (1) (which also bounds the cost of the root isolation), and that the second summand is caused by the cost of the quadratic sequence. It is remarkable that our analysis profits from considering all (real) roots of f; when restricting to a single root of f, we are able to decrease only the second summand by a factor of p, even if the root is already given by an isolating interval.

The reader might wonder at this point why not using a more prominent algorithm like the famous *Newton iteration* instead of the qir method. A problem in Newton's method lies in the choice of a starting value – an unfortunate one leads to a diverging sequence. A solution is to perform bisections initially to produce an interval where convergence of Newton's method is guaranteed, and then to switch to Newton iteration manually. However, this manual switch depends on theoretical worst-case bounds for valid starting values of Newton's method, thus more bisections than actually necessary are performed in the average case. The qir method, in contrast, switches adaptively as soon as possible, independently of the worst-case bounds that are introduced only for the analysis.

Dekker [7] presented a method which, similarly to the qir, combines bisections and the secant method. Brent [3] combines Dekker's method with inverse quadratic interpolation. Superlinear convergence can also be guaranteed for this method. However, a problem in Dekker's approach is the growth in the bitsize of the iteration values – it appears unclear to the author how to choose a suitable working precision in each substep to avoid a too big coefficient swell-up while still guaranteeing fast convergence. The same holds true for Brent's method, and additionally, an analysis seems to be even more involved as it even adds more ingredients to Dekker's method. The qir method guarantees a minimal growth in the bitsizes, since all intervals are of the form $[\frac{a}{2^\ell}, \frac{a+1}{2^\ell}]$ (with $a, \ell \in \mathbb{Z}$), thus the bitsize of the boundaries is proportional to the interval width, what is the best one can hope for.

The simpleness of the qir method also make this approach attractive for concrete implementation. It is used both in the COCOA library[1] [1] and the (experimental) algebraic kernel of the CGAL library[2] (used, for instance, in [11, 10]). Its application is also attested in [8]. In this work, however, we focus on the complexity analysis, and do not address its practical performance.

This paper is structured as follows: In Section 2, we give a rough overview about real root isolation algorithms, and their complexity. Section 3 revises the qir method. Our complexity bound (1) is proved in Section 4. We conclude in Section 5.

[1] http://http://cocoa.dima.unige.it/
[2] http://www.cgal.org

Notation

It will be convenient to fix some notation. Throughout this article, let $f = \sum_{i=0}^{p} a_i x^i$ be a square-free polynomial (i.e., without multiple roots) of degree p, with integer coefficients a_i of bitsize σ, that means, $|a_i| \leq 2^\sigma$. The complex roots of f are denoted by $\alpha_1, \ldots, \alpha_p$, and we assume exactly the first s roots $\alpha_1, \ldots, \alpha_s$ to be real.

Also, let $0 < \epsilon < 1$ be fixed, and set $L := \log \frac{1}{\epsilon}$. We write $M(n)$ for the cost of multiplying two integers of bitsize n, and assume that $M(n) = O(n \log n \log \log n)$, according to the fast multiplication algorithm by Schönhage and Strassen [15]. To keep the complexity bound handleable, we will often neglect logarithmic factors in p and σ and denote such complexity bounds by $\tilde{O}(\cdot)$. As an example, $M(n) = \tilde{O}(n)$. Finally, for $I = (c, d)$, we denote by $w(I) := d - c$ its width.

2 Root Isolation

Several approaches have been investigated for the root isolation problem. They all accept the square-free polynomial f as input, and produce a list of s isolating intervals for $\alpha_1, \ldots, \alpha_s$. A considerable body of literature has appeared about this problem (a small subset is [5, 6, 14, 16]); it is not the scope of this work to discuss them in detail – still, their worst-case bound is of importance.

Theorem 2.1. Computing isolating intervals for the real roots of f requires at most $\tilde{O}(n^4\sigma^2)$ bit operations in the worst-case (using fast arithmetic). Moreover, each isolating interval is of the form $(\frac{a}{2^\ell}, \frac{a+1}{2^\ell})$ with $a, \ell \in \mathbb{Z}$ and $\log \left|\frac{a}{2^\ell}\right| = O(\sigma)$.

The complexity bounds have been proved for root isolation based on *Sturm sequences* [9], and based on *Descartes' rule of signs* [12]. The special form of the isolating intervals is a consequence of the subdivision that is initially started with an interval $[-2^{O(\sigma)}, 2^{O(\sigma)}]$ that covers all real roots of f (compare [2, §10.1]).

We remark that the *Continued fraction algorithm* (introduced in [5]) usually perform best in practice among the available modern root solvers, although the best known bound in the literature seems to be $\tilde{O}(n^5\tau^2)$ [16]. See [13] for a recent experimental comparison on various modern root solvers.

3 Abbott's Quadratic Interval Refinement

Everybody knows about the most naive method for refining isolating intervals – the *bisection method*. Given an isolating interval (c, d), evaluate f at the midpoint $m = \frac{c+d}{2}$. If $f(m) = 0$, the root is found exactly. Otherwise, either (c, m) or (m, d) is chosen as refined isolating interval, depending on where the sign change takes place. Clearly, the isolating interval is halved in every step which means that one bit of precision is added per bisection.

The analysis of the complexity for the bisection is also straight-forward. The crucial operation is to evaluate f at m, the number of arithmetic operations is linear. If τ denotes the bitsize of c and d, then the bisection has to deal with bitsizes up to $O(\sigma + p\tau)$ during the evaluation, and thus the number of bit operations is bounded by

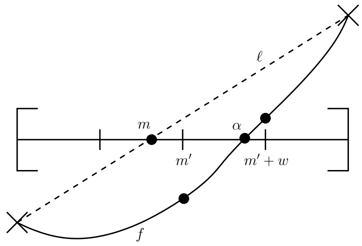
$$O(pM(\sigma + p\tau)).$$

Algorithm 1 Quadratic interval refinement

procedure QIR($f, I = (c, d), N$) $\triangleright$ Returns a pair (J, N_{new}), with J the refined interval
  if $N = 2$, **return** (BISECTION(f, I),4).
  $w \leftarrow \frac{d-c}{N}$
  $m' \leftarrow c + \mathrm{round}(N \frac{f(c)}{f(c)-f(d)})w$ $\triangleright$ $m = c + \frac{f(c)}{f(c)-f(d)}(d - c)$
  $s \leftarrow \mathrm{sgn}(f(m'))$
  if $s = 0$, **return** $([m', m'], \infty)$
  if $s = \mathrm{sgn}(f(c))$ **and** $\mathrm{sgn}(f(m' + w)) = \mathrm{sgn}(f(d))$, **return** $((m', m' + w), N^2)$
  if $s = \mathrm{sgn}(f(d))$ **and** $\mathrm{sgn}(f(m' - w)) = \mathrm{sgn}(f(c))$, **return** $((m' - w, m'), N^2)$
  Otherwise, return $(I,\sqrt{N})$.
end procedure

What if we did bisection until the interval gets smaller then ϵ? We would have to perform up to $\sigma + L$ bisection steps (the initial σ bisections to make its width smaller than one), and the interval boundaries would grow to bitsize $\sigma + L$. Thus, one would arrive at a total complexity of $O(p(\sigma + L)M(p(\sigma + L))) = \tilde{O}(p^2(\sigma + L)^2)$, with an additional factor of p when doing this for each root. Not surprisingly, this is inferior to (1) since L appears quadratically.

A more efficient way of refining the isolating interval has been presented with Abbott's *Quadratic Interval Refinement method* [1]; we call it *qir* from now. Consider an isolating interval $I = (c, d)$ for a root α, and let ℓ be the secant through the points $(c, f(c))$ and $(d, f(d)) \in \mathbb{R}^2$. If I is small enough, f should almost look like the line ℓ over I, and thus, the intersection point m of ℓ with the x-axis should be close to α.



Successful qir instance for $N = 4$

This idea leads to the following algorithm: Having an additional integer N as input, subdivide I (conceptually) by $N + 1$ equidistant grid points (with distance $w := \frac{w(I)}{N}$). Then, compute m', the closest grid point to m, and evaluate $f(m')$. Depending on its sign, evaluate the sign of either the left or right neighboring grid point. If the sign changes from m' to $m' \pm w$, choose it as new isolating interval (this refines by a factor of N) and set N to N^2 for the next qir call. Otherwise, keep I as isolating interval and set N to $\sqrt{N}$ for the next call. If $N = 2$, perform one bisection step. See also Algorithm 1 for a pseudo-code description.

We assume that N is initially set to 4 for an isolating interval returned by a root isolation algorithm, and that the method QIR is always called with the parameter N that has been returned in the previous call for the given interval.

Different from Abbott's original formulation, a call of QIR does not necessarily refine the isolating interval. However, in this case, N is decreased as a side effect, and at the latest when $N = 2$, the method will refine the interval eventually.

Definition 3.1. A qir call $(J, N_2) \leftarrow$QIR(f,I,N_1) *succeeds*, if $J \subsetneq I$, and it *fails*, if $J = I$. Equivalently, the qir calls succeeds, if and only if $N_2 > N_1$.

For one qir call (successful or not), one has to perform only $O(p)$ arithmetic op-

Algorithm 2 Root isolation with refinement

1: **procedure** ISOLATE_AND_REFINE(f, ϵ)
2: $\quad I_1, \ldots, I_s \leftarrow \text{ISOLATE}(f)$ $\qquad \triangleright$ see Section 2
3: $\quad$ **for** $k \in \{1, \ldots, s\}$ **do**
4: $\quad\quad N \leftarrow 4$
5: $\quad\quad$ **while** $\mathrm{width}(I_i) > \epsilon$ **do** $(I_i, N) \leftarrow \text{QIR}(f, I_i, N)$
6: $\quad$ **end for**
7: $\quad$ **return** $I_1, \ldots, I_s$
8: **end procedure**

erations to evaluate f at m' and $m' \pm w$, and perform another constant number of arithmetic operations. The bitsize of m' and $m' \pm w$ is bounded by $O(\log N + \tau)$ where τ is the maximal bitsize of c and d.

It is easy to see that $\log N \in O(\tau)$, assuming that the qir is initially started with $N = 4$: if a qir call with $N > 4$ subintervals is started, there must have been a successful qir call for $\sqrt{N}$. Thus, the width of the interval is at most $\frac{1}{\sqrt{N}}$, and the bitsize of either c or d must be at least $\log \sqrt{N} = \frac{1}{2} \log N$.

After all, the cost of one qir call is thus bounded by

$$O(pM(\sigma + p\tau)),$$

which is equal to the cost of one bisection step. We remark that one successful qir step yields exactly the same result as $\log N$ bisections, so that the isolating interval remains of the form $(\frac{a}{2^\ell}, \frac{a+1}{2^\ell})$ if the initial interval was of this type.

4 Analysis of Root Refinement

We prove the bound given in (1). For that, we analyze the complexity of this straightforward algorithm: Apply QIR to each isolating interval until its width falls below ϵ (Algorithm 2).

Definition 4.1. Let α be a root of f for which Step 2 of Algorithm 2 returned the isolating interval I_0. The *qir sequence* $(s_0, \ldots, s_n)$ for α, is defined as

$$s_0 := (I_0, 4) \qquad s_i := (I_i, N_i) := \text{QIR}(f, I_{i-1}, N_{i-1}) \quad \text{for } i \geq 1$$

where I_n is the first index such that $w(I_n) \leq \epsilon$. We say that $s_{i-1} \overset{\text{QIR}}{\rightarrow} s_i$ *succeeds* if $\text{QIR}(f, I_{i-1}, N_{i-1})$ succeeds, and that $s_{i-1} \overset{\text{QIR}}{\rightarrow} s_i$ *fails* otherwise.

The qir sequence for α is split into two subsequences, according to the value M_α defined in the next lemma. M_α will turn out to be an upper bound for the width of the isolating interval of α that ensures quadratic convergence. We will prove this in Section 4.2, but we already show two simple properties of M_α.

Lemma 4.2. Let $\alpha \in \mathbb{C}$ be a root α of f. We define

$$M_\alpha := \frac{|f'(\alpha)|}{2ep^3 2^\sigma \max\{|\alpha|, 1\}^{p-1}}$$

with $e \approx 2.718$. It holds that

1. $0 < M_\alpha < \frac{1}{p}$
2. Let $\mu \in \mathbb{C}$ be such that $|\alpha - \mu| < M_\alpha$. Then

$$M_\alpha < \frac{|f'(\alpha)|}{2|f''(\mu)|}.$$

Proof. We bound $|f'(\alpha)|$ from above by the following

$$|f'(\alpha)| = |\sum_{i=1}^{p} i a_i \alpha^{i-1}| \leq p 2^\sigma \sum_{i=0}^{p-1} \max\{|\alpha|, 1\}^i \leq p 2^\sigma p \max\{|\alpha|, 1\}^{p-1}$$

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

$$|f''(\mu)| = |\sum_{i=2}^{p} i(i-1) a_i \mu^{i-2}| \leq p^2 2^\sigma \sum_{i=0}^{p-2} \max\{|\mu|, 1\}^i \leq p^2 2^\sigma \sum_{i=0}^{p-2} ((1 + M_\alpha) \max\{|\alpha|, 1\})^i$$

$$\leq p^3 2^\sigma (1 + M_\alpha)^{p-2} \max\{|\alpha|, 1\}^{p-2} < p^3 2^\sigma \underbrace{(1 + \frac{1}{p})^p}_{<e} \max\{|\alpha|, 1\}^{p-1}$$

This shows that

$$\frac{|f'(\alpha)|}{2|f''(\mu)|} > \frac{|f'(\alpha)|}{2e \cdot p^3 2^\sigma \max\{|\alpha|, 1\}^{p-1}} = M_\alpha \qquad \square$$

Definition 4.3. Let $(s_0, \ldots, s_n)$ be the qir sequence for α. Let k be the minimal index such that $s_k = (I_k, N_k) \overset{\text{QIR}}{\rightarrow} s_{k+1}$ succeeds, and $w(I_k) \leq M_\alpha$. We call the sequence $(s_0, \ldots, s_k)$ the *initial sequence*, and $(s_k, \ldots, s_n)$ the *quadratic sequence*.

In other words, the quadratic sequence is the maximal qir sequence that only contains intervals of size at most M_α, and starts with a successful qir step. In the next two subsections, we will bound the cost of the initial sequence and the quadratic sequence separately.

4.1 Cost of the initial sequence

Lemma 4.4. Let I be an isolating interval for α. The cost of the initial sequence of α is bounded by

$$\tilde{O}(p^2(\sigma + \log \frac{1}{M_\alpha})^2)$$

Proof. Let n_q be the number of qir calls until I is refined such that $w(I) < M_\alpha$. Likewise, let n_b be the number of bisections that would be needed to refine I to size M_α. Note that $n_b = O(\sigma + \log \frac{1}{M_\alpha})$.

A successful qir call for some $N = 2^{2^i}$ yields the same accuracy as 2^i bisections, and can only cause up to $i + 1$ subsequent failing qir calls before the next successful qir call. With that argument, it follows that $n_q \leq 2n_b$, so the number of qir calls is in $O(\sigma + \log \frac{1}{M_\alpha})$.

To bound the bitsizes, let N_e be the value of N in the last successful qir call of the initial sequence. It holds that $\log N_e \leq 2n_b$, since otherwise, the preceding qir call would have yielded as much accuracy as $\log\sqrt{N_e} > n_b$ bisections, and the initial sequence would have stopped earlier. Hence, the width of the final interval is at least $\frac{M_\alpha}{N_e}$, and the interval boundaries have bit complexity

$$\log\frac{N_e}{M_\alpha} \leq 2n_b + \log\frac{1}{M_\alpha} = O(\sigma + \log\frac{1}{M_\alpha})$$

Therefore, the bitsizes of the qir calls are bounded by $O(p(\sigma+\log\frac{1}{M_\alpha}))$, which proves the claim. $\square$

It remains to bound the quantity $\log\frac{1}{M_\alpha}$. We do this simultaneously for all real roots of the polynomial, according to the following theorem.

Theorem 4.5. Let $\alpha_1,\ldots,\alpha_s$ be the real roots of f. Then,

$$\sum_{i=1}^{s}\log\frac{1}{M_{\alpha_i}} = O(p(\sigma+\log p)).$$

Proof. Recall that $0 < M_\alpha < 1$ for each (complex) root α, so $\log\frac{1}{M_\alpha} > 0$, and we can bound:

$$\begin{aligned}
\sum_{i=1}^{s}\log\frac{1}{M_{\alpha_i}} \leq \sum_{i=1}^{p}\log\frac{1}{M_{\alpha_i}} &= \log\frac{\prod_{i=1}^{p} 2e\cdot p^3 2^\sigma \max\{|\alpha_i|,1\}^{p-1}}{|\prod_{i=1}^{p} f'(\alpha_i)|} \\
&= p\log(2e) + 3p\log p + p\sigma + (p-1)\log\prod_{i=1}^{p}\max\{|\alpha_i|,1\} - \log|\prod_{i=1}^{p} f'(\alpha_i)|
\end{aligned}$$

For both occurring products, we can apply well-known bounds from Algebra. For the first one, note that

$$\mathrm{Mea}(f) := |a_p|\prod_{i=1}^{p}\max\{|\alpha_i|,1\}$$

is the *Mahler measure* of f, and it holds that ([17, Lemma 4.14], [2, Prop.10.9])

$$\mathrm{Mea}(f) \leq \|f\|_2 \leq \sqrt{p+1}\cdot\|f\|_\infty \leq \sqrt{p+1}\cdot 2^\sigma.$$

So, $\log\prod_{i=1}^{p}\max\{|\alpha_i|,1\} = \log\left(\frac{1}{|a_p|}\mathrm{Mea}(f)\right) \leq \log\mathrm{Mea}(f) = O(\log p + \sigma)$

The second product is related to the resultant of f and f' by the following identity [2, Thm.4.16], [17, Thm.6.15]

$$\mathrm{res}(f,f') = a_p^{p-1}\prod_{i=1}^{p} f'(\alpha_i).$$

In particular, the right hand side yields an integer. It follows

$$-\log|\prod_{i=1}^{p} f'(\alpha_i)| = \log|a_p^{p-1}| - \log|\underbrace{\mathrm{res}(f,f')}_{\geq 1}| < (p-1)\log|a_p| = O(p\sigma).$$

Finally, we can estimate

$$\begin{aligned}
\sum_{i=1}^{s}\log\frac{1}{M_{\alpha_i}} &\leq O(p(\sigma+\log p)) + (p-1)\underbrace{\log\prod_{i=1}^{p}\max\{|\alpha_i|,1\}}_{=O(\log p+\sigma)}\underbrace{-\log|\prod_{i=1}^{p} f'(\alpha_i)|}_{=O(p\sigma)} \\
&= O(p(\sigma+\log p)).
\end{aligned}$$

$\square$

Corollary 4.6. The total computation cost for all initial sequences is $\tilde{O}(p^4\sigma^2)$.

Proof. Combining Lemma 4.4 and Theorem 4.5, we get total costs of

$$\tilde{O}(\sum_{i=1}^{s} p^2(\sigma + \log \frac{1}{M_{\alpha_i}})^2) = \tilde{O}(p^3\sigma^2 + p^2(\sum_{i=1}^{s} \log \frac{1}{M_{\alpha_i}})^2) = \tilde{O}(p^4\sigma^2) \qquad \square$$

Corollary 4.6 shows that refining all isolating intervals to width M_α does not increase the complexity bound to the initial root isolation.

4.2 Cost of the quadratic sequence

In the initial sequence, we have assumed that the qir sequence behaves roughly as the bisection method. As soon as the isolating interval becomes smaller than M_α, we can prove that N is squared in (almost) every step, which leads to quadratic convergence of the interval width. We start with a simple criterion that guarantees a successful qir call.

Lemma 4.7. Let $I = (c, d)$ be an isolating interval of α, with $w(I) = \delta$, and consider the qir call QIR(f,I,N) for some N. Let $m := c + \frac{f(c)}{f(c)-f(d)}(d - c)$ be defined as in the qir method (Algorithm 1). If $|m - \alpha| < \frac{\delta}{2N}$, the qir call succeeds.

Proof. Recall that I is conceptually subdivided into N subintervals of same width, and that m' is chosen as the grid point closest to m. Let J be the subinterval that contains α, and J' be the subinterval that contains m. If $J = J'$, then one of the endpoints of J' is chosen as m', so the qir call succeeds. If $J \neq J'$, they must be adjacent, since otherwise, $|m - \alpha| > \frac{\delta}{N}$. W.l.o.g., assume that $m < \alpha$, then m must be in the right half of J, because otherwise $|m - \alpha| > \frac{\delta}{2N}$. Thus, m' is chosen as the right endpoint of J' which is the left endpoint of J. Therefore, the qir call succeeds. $\square$

We need to investigate the distance between the interpolation point m and the root α. The next theorem shows that this distance depends quadratically on the width of the isolating interval, once it is smaller than M_α. This is basically analogous to Newton's iteration, for which a similar theorem is shown.

Theorem 4.8. Let (c, d) be an isolating interval for α of width $\delta < M_\alpha$. Then $|m - \alpha| < \frac{\delta^2}{2M_\alpha}$.

Proof. We consider the Taylor expansion of f at α. For a given $x \in [c, d]$, we have

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

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

$$\begin{aligned}
|m - \alpha| &= \left|\frac{f(d)(c - \alpha) - f(c)(d - \alpha)}{f(d) - f(c)}\right| \\
&= \left|\frac{\frac{1}{2}(f''(\tilde{\alpha}_1)(d - \alpha)^2(c - \alpha) - f''(\tilde{\alpha}_2)(c - \alpha)^2(d - \alpha))}{f(d) - f(c)}\right| \\
&\leq \frac{1}{2}|d - \alpha||c - \alpha| \cdot \frac{|f''(\tilde{\alpha}_1)|(d - \alpha) + |f''(\tilde{\alpha}_2)|(\alpha - c)}{|f(d) - f(c)|}
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{1}{2}\delta^2 \max\{|f''(\tilde{\alpha}_1)|, |f''(\tilde{\alpha}_2)|\} \frac{(d-\alpha)+(\alpha-c)}{|f(d)-f(c)|} \\
&= \frac{\delta^2 \max\{|f''(\tilde{\alpha}_1)|, |f''(\tilde{\alpha}_2)|\}}{2|f'(\nu)|}
\end{aligned}
$$

for some $\nu \in (c, d)$. The Taylor expansion of f' yields $f'(\nu) = f'(\alpha) + f''(\tilde{\nu})(\nu - \alpha)$ with $\tilde{\nu} \in (c, d)$. Since $\delta \leq M_\alpha$, it follows with Lemma 4.2

$$
|f''(\tilde{\nu})(\nu - \alpha)| \leq |f''(\tilde{\nu})| M_\alpha \leq \frac{1}{2}|f'(\alpha)|.
$$

Therefore $|f'(\nu)| > \frac{1}{2}|f'(\alpha)|$, and it follows again with Lemma 4.2 that

$$
|m - \alpha| \leq \frac{\delta^2 \max\{|f''(\tilde{\alpha}_1)|, |f''(\tilde{\alpha}_2)|\}}{|f'(\alpha)|} = \frac{\delta^2}{2\frac{|f'(\alpha)|}{2\max\{|f''(\tilde{\alpha}_1)|, |f''(\tilde{\alpha}_2)|\}}} < \frac{\delta^2}{2M_\alpha} \qquad \square
$$

We apply this theorem on the quadratic sequence.

Corollary 4.9. Let I_j be an isolating interval for α of width $\delta_j \leq \frac{1}{N_j} M_\alpha$. Then, each call of the qir sequence $(I_j, N_j) \overset{\text{QIR}}{\to} (I_{j+1}, N_{j+1}) \overset{\text{QIR}}{\to} \ldots$ succeeds.

Proof. We do induction on i. Assume (for $i \geq 0$) that the first i calls succeeded. Then, it is easily shown that $\delta_{j+i} := w(I_{j+i}) = \frac{N_j \delta_j}{N_{j+i}} < \frac{M_\alpha}{N_{j+i}}$ (by another induction, and exploiting that $N_{j+i}^2 = N_{j+i+1}$). Using Theorem 4.8, we have that

$$
|m - \alpha| \leq \delta_{j+i}^2 \frac{1}{2M_\alpha} \leq \delta_{j+i} \frac{M_\alpha}{N_{j+i}} \frac{1}{2M_\alpha} = \frac{1}{2} \frac{\delta_{j+i}}{N_{j+i}}
$$

By Proposition 4.7, this is enough to guarantee success for the qir method. $\square$

Corollary 4.10. In the quadratic sequence, there is at most one failing qir call.

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

$$
(I_{i-1}, N_{i-1}) \overset{\substack{\text{Sucess}\\ \text{QIR}}}{\to} (I_i, N_i) \overset{\substack{\text{Fail}\\ \text{QIR}}}{\to} (I_{i+1}, N_{i+1}) \overset{\text{QIR}}{\to} \ldots.
$$

One observes that $w(I_{i+1}) = w(I_i) = \frac{w(I_{i-1})}{N_{i-1}} \leq \frac{M_\alpha}{N_{i-1}}$, and $N_{i+1} = \sqrt{N_i} = \sqrt{N_{i-1}^2} = N_{i-1}$. By Corollary 4.9, all further qir calls succeed. $\square$

If the quadratic sequence starts with a bisection (i.e., $N = 2$ initially), no failing qir call occurs. Otherwise, the single failing step is due to the fact that the quadratic sequence might start with a too big value of N, just because the algorithm was “too lucky” during the initial sequence.

Let $(I_{i-1}, N_{i-1}) \overset{\text{QIR}}{\to} (I_i, N_i)$ be the failing qir call in the quadratic sequence. Since $w(I_{i+k}) = \frac{N_i w(I_i)}{N_{i+k}}$ by the proof of Corollary 4.10, it follows that

$$
w(I_{i+k+1}) = \frac{w(I_{i+k})^2}{N_k \cdot w(I_k)}
$$

for any $k \geq 0$. That means, the interval width decreases quadratically in each step (up to the constant $N_k \cdot w(I_k)$) which ultimately justifies the term "quadratic" in the Quadratic Interval Refinement method (the idea of our exposition was already sketched in Abbott's original work [1]).

Lemma 4.11. The number of bit operations in the quadratic sequence of a root α is bounded by

$$\tilde{O}(p^2 \log L(\sigma + \log \frac{1}{M_\alpha}) + p^2 L).$$

Proof. By Corollary 4.10, the quadratic sequence consists of at most $\log L+1$ qir calls, since N is doubled in each step, except the possible failing step. The bitsize in the first qir call of the sequence is $O(p(\sigma + \log \frac{1}{M_\alpha}))$, and increases by at most 2^i after the i-th iteration. Therefore, the complexity of the quadratic sequence is given by

$$O\left(\sum_{i=1}^{\log L+1} p \cdot M(p(\sigma + \log \frac{1}{M_\alpha} + 2^i))\right) = \tilde{O}\left(p^2 \sum_{i=1}^{\log L+1} \sigma + \log \frac{1}{M_\alpha} + 2^i\right)$$

$$= \tilde{O}\left(p^2 \log L(\sigma + \log \frac{1}{M_\alpha}) + p^2 \sum_{i=1}^{\log L+1} 2^i\right) = \tilde{O}(p^2 \log L(\sigma + \log \frac{1}{M_\alpha}) + p^2 L)$$

□

Corollary 4.12. The total cost of all quadratic sequences for the real roots $\alpha_1, \ldots, \alpha_s$ of f is bounded by

$$\tilde{O}(p^3 \sigma \log L + p^3 L).$$

Proof. We combine Lemma 4.11 and Theorem 4.5 to obtain

$$\sum_{i=1}^{s} \tilde{O}(p^2 \log L(\sigma + \log \frac{1}{M_\alpha}) + p^2 L) = \tilde{O}(p^3 \sigma \log L + p^2 \log L \underbrace{\sum_{i=1}^{s} \log \frac{1}{M_\alpha}}_{=O(p(\sigma + \log p))} + p^3 L)$$

□

Combining the cost for root isolation (Theorem 2.1) with the cost of the initial sequences (Corollary 4.6) and the cost of the quadratic sequences (Corollary 4.12) proves the main result:

Theorem 4.13. Isolating the real roots of f, and computing an isolating interval of width at most ϵ for each root using Algorithm 2 requires

$$\tilde{O}(p^4 \sigma^2 + p^3(L + \sigma \log L)) = \tilde{O}(p^4 \sigma^2 + p^3 L)$$

bit operations.

Proof. We only have to argue why the summand $p^3 \sigma \log L$ can never dominate the other two. If $p^4 \sigma^2$ was dominated by $p^3 \sigma \log L$, $\log L$ would dominated $p\sigma$, and in particular L would dominate 2^σ. If also $p^3 L$ was dominated by $p^3 \sigma \log L$, then $\frac{L}{\log L}$ is dominated by σ, so L is dominated by $\sigma^{1+\gamma}$ for any $\gamma > 0$. Contradiction. □

If only one isolating interval is refined, our method shows a complexity of $\tilde{O}(p^4\sigma^2 + p^2L)$, and thus only a partial improvement (even without considering the initial root isolation step). The reason is that we are not aware of an improved bound for the initial sequence of a single α compared to what we prove in Theorem 4.5.

From a theoretical point of view, we do not expect significant improvements when using any other quadratic convergent method than qir: the first summand $p^4\sigma^2$ of Theorem 4.13 appears due to root isolation, and the summand p^3L seems to be unavoidable as well, since for each root, one has to perform at least one evaluation of f for a rational number of bitsize $O(L)$, which leads to $O(n)$ arithmetic operations with integer of bitsize up to $\tilde{O}(nL)$.

5 Conclusions and Further Work

Theorem 4.13 shows that refining all real roots of a polynomial to width ϵ is as complex as just isolating the roots, provided that $\log \epsilon^{-1} = \tilde{O}(p\sigma^2)$. We believe this result to be of general interest for algorithm dealing with real algebraic numbers. For instance, the usage of qir instead of naive bisection removes the asymptotic bottleneck in the topology computation algorithm presented in [11]; this is currently work in progress.

On the practical side, we have argued that qir has a more adaptive behavior than a combination of bisection and Newton's method, since the switch from linear to quadratic convergence happens without a "manual" control from outside. Abbott's work [1] has already shown that qir is competitive to Newton's method in a different context. However, a comparison to other hybrid approaches like Brent's method is still missing.

Acknowledgements: The author would like to thank Tobias Gärtner, Kurt Mehlhorn, Michael Sagraloff and Vikram Sharma for valuable discussions,

References

[1] J. Abbott: "Quadratic Interval Refinement for Real Roots". URL `http://www.dima.unige.it/~abbott/`. Poster presented at the 2006 Int. Symp. on Symb. and Alg. Comp. (ISSAC 2006).

[2] S. Basu, R. Pollack, M.-F. Roy: *Algorithms in Real Algebraic Geometry*, *Algorithms and Computation in Mathematics*, vol. 10. Springer, 2nd edn., 2006.

[3] R. Brent: *Algorithms for Minimization without Derivatives*, chap. 4. Prentice-Hall, 1973.

[4] B. F. Caviness, J. R. Johnson (eds.): *Quantifier Elimination and Cylindrical Algebraic Decomposition, Texts and Monographs in Symbolic Computation*. Springer, 1998.

[5] G. E. Collins, A. G. Akritas: "Polynomial Real Root Isolation Using Descartes' Rule of Signs". In: R. D. Jenks (ed.) *SYMSAC*. ACM Press, Yorktown Heights, NY, 1976 272–275.

[6] G. E. Collins, R. Loos: "Real zeroes of polynomials". In: *Computer algebra: symbolic and algebraic computation (2nd ed.)*, 83–94. Springer, New York, NY, USA, 1983.

[7] T. Dekker: "Finding a zero by means of successive linear interpolation". In: B. Dejon, P. Henrici (eds.) *Constructive Aspects of the Fundamental Theorem of Algebra*, 1969 .

[8] D. I. Diochnos, I. Z. Emiris, E. P. Tsigaridas: "On the complexity of real solving bivariate systems". In: *ISSAC '07: Proceedings of the 2007 international symposium on Symbolic and algebraic computation*. ACM, New York, NY, USA, 2007 127–134.

[9] Z. Du, V. Sharma, C. Yap: "Amortized Bound for Root Isolation via Sturm Sequences". In: *Proceedings of the International Workshop on Symbolic-Numeric Computation (SNC 2007)*, 2007 .

[10] A. Eigenwillig, M. Kerber: "Exact and efficient 2D-arrangements of arbitrary algebraic curves". In: *Proc. of the nineteenth annual ACM-SIAM Symposium on Discrete Algorithms (SODA'08)*, 2008 122–131.

[11] A. Eigenwillig, M. Kerber, N. Wolpert: "Fast and Exact Geometric Analysis of Real Algebraic Plane Curves". In: C. W. Brown (ed.) *Proocedings of the 2007 International Symposium on Symbolic and Algebraic Compuation (ISSAC 2007)*, 2007 .

[12] A. Eigenwillig, V. Sharma, C. Yap: "Almost Tight Recursion Tree Bounds for the Descartes Method". In: *Proceedings of the 2006 International Symposium on Symbolic and Algebraic Computation*, 2006 71–78.

[13] I. Z. Emiris, M. Hemmer, M. Karavelas, B. Mourrain, E. P. Tsigaridas, Z. Zafeirakopoulos: *Experimental evaluation and cross-benchmarking of univariate real solvers*. Rapport de recherche EMIRIS:2008:INRIA-00340887:1, INRIA, Sophia Antipolis, France, 2008.

[14] J. R. Johnson: "Algorithms for Real Root Isolation". In: Caviness and Johnson [4], 1998.

[15] A. Schönhage, V. Strassen: "Schnelle Multiplikation grosser Zahlen". *Computing* **7** (1971) 281–292.

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

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