Near Optimal Tree Size Bounds on a Simple Real Root Isolation Algorithm

Vikram Sharma
Institute of Mathematical Sciences
Chennai, India 600113
vikram@imsc.res.in

Chee K. Yap*
Dept. of Computer Science, New York University
New York, NY 10012
yap@cs.nyu.edu

ABSTRACT

The problem of isolating all real roots of a square-free integer polynomial $f(X)$ inside any given interval I_0 is a fundamental problem. EVAL is a simple and practical exact numerical algorithm for this problem: it recursively bisects I_0, and any sub-interval $I \subseteq I_0$, until a certain numerical predicate $C_0(I) \vee C_1(I)$ holds on each I. We prove that the size of the recursion tree is

$$O(d(L + r + \log d))$$

where f has degree d, its coefficients have absolute values $< 2^L$, and I_0 contains r roots of f.

In the range $L \geq d$, our bound is the sharpest known, and provably optimal. Our results are closely paralleled by recent bounds on EVAL by Sagraloff-Yap (ISSAC 2011) and Burr-Krahmer (2012). In the range $L \leq d$, our bound is incomparable with those of Sagraloff-Yap or Burr-Krahmer.

Similar to the Burr-Krahmer proof, we exploit the technique of "continuous amortization" from Burr-Krahmer-Yap (2009), namely to bound the tree size by an integral $\int_{I_0} G(x)dx$ over a suitable "charging function" $G(x)$.

The introduction of the output-size parameter r seems new. We give an application of this feature to the problem of ray-shooting (i.e., finding smallest root in a given interval).

Keywords

Continuous amortization, Subdivision algorithm, Integral analysis, Real Root isolation

1. INTRODUCTION

Given a square-free polynomial $f \in \mathbb{Z}[X]$, the problem is to isolate some or all the roots of f. It is a very classic problem that is treated in many fields, with many variations and known algorithms. We focus on exact methods for isolating real roots that provide global guarantees of correctness. Such methods are traditionally based on algebraic approaches [24, 15, 9, 10, 28]. But there is growing interest in numerical approaches that are exact. We are interested in numerical methods because they are typically easier to implement and have adaptive complexity (see [2] for an extended discussion of this point). Various exact numerical algorithms have been extensively studied in the interval analysis community [19, 20].

There is one major gap in the numerical approaches: they generally lack (non-trivial) complexity analysis. This is because many numerical algorithms are adaptive and iterative, and operate by successive approximations in a continuum (e.g., in $\mathbb{R}, \mathbb{C}, \mathbb{R}^n$). In particular, the class of **subdivision algorithms** uses adaptive iteration to repeatedly subdivide an initial domain until some terminal condition holds. Examples include various marching cube type algorithms [17, 22, 29, 16]. But there are few techniques for analyzing adaptive iteration. One approach that can account for adaptive complexity (e.g., in Linear Programming) is to invoke probabilistic assumptions. In [4], we introduced a general framework for analyzing subdivision algorithms. This is a non-probabilistic approach providing worse-case complexity bounds, and can be interpreted as a kind of "continuous amortization". Note that the concept of amortization is well-known in discrete algorithms [30, 6].

In this paper, we will use the continuous amortization framework to analyze a simple real-root isolation algorithm called EVAL [4, 3]. EVAL is based on well-known numerical predicates called "centered-forms" in the interval literature [23]. Assuming that the polynomial $f(X) \in \mathbb{Z}[X]$ has degree d and maximum absolute coefficient size $< 2^L$, our main result is that the recursion tree for EVAL is $O(d(L + r + \log d))$ where $r \leq d$ is the number of real roots of f in the input interval I_0. There are three recent complexity analysis of EVAL:

- Burr, Krahmer and Yap [4] gave the first polynomial bound of $O(d^3(L+\log d))$ using complicated arguments that invoke several non-trivial algebraic root bound techniques.
- Sagraloff and Yap [27] showed that EVAL has tree size $O(d(L + \log d)(\log L + \log d))$. This proof does not use continuous amortization, but the technique seems rather more robust because it applies also to a complex analogue of EVAL.
- Burr and Krahmer [3] obtained the bound $O(d(L + \log d))$ but under the additional hypothesis that f'

*This authors' work is supported by NSF Grants CCF-0728977 and CCF-0917093.

is square-free. Their breakthrough comes from realizing that the well-known potential function $S(z) = \sum_\alpha \frac{1}{|z-\alpha|}$ (with α ranging over the roots of f) can be used as a "charging function".

Our new result is a refinement and variation of the arguments in [3]. In the range $L \geq d$, our new result improves on [27] by removing the logarithmic factor in $\log L$ and $\log d$, and improves on [3] by removing the need for f' to be square-free. In this range, our bound is the best possible in view of the lower bound from Eigenwillig et al. [10]. An intriguing question raised by our result is the complexity of EVAL in the range is $\log d < L < d$ where we now have 3 mutually incompatible complexity bounds.

§1. Some Related Literature.

We focus only on exact subdivision approaches to real root isolation: the classic method is based on Sturm sequences. In recent years, algorithms based on the Descartes Method [5, 26] have been favored by implementers. The Continued Fraction Method [1] is another approach that is empirically the fastest among these methods. A comprehensive empirical study of these methods on the Benchmark problem of isolating all the roots of algebraic polynomials is given by Hemmer et al [11]. It has been long known from Davenport (1985) [7] that the tree size from Sturm methods is $O(d(L + \log d))$. In Eigenwillig et al (2006) [10], we proved the same bound for Descartes methods. Similar sharp bounds for Continued Fraction method is finally obtained in [28]. Furthermore, we showed these bounds to be optimal for $L \geq \log d$ (see [10]). It was surprising that a simple numerical approach like EVAL can[1] match the worst case complexity bounds of its more algebraic counterparts.

EVAL in the current form, as an exact root isolation algorithm, was introduced by Yap as the 1-D analogue of algorithms by Vegter-Plantinga [22] for curves and surfaces. However, the basic form of this algorithm first appeared in Mitchell's work on ray-shooting [18]. EVAL should be viewed as the simplest algorithm of a family that includes the recently proposed CEVAL for complex root isolation [27, 13, 14]. Higher dimensional analogues appear in [22, 16]. This makes connections to the large marching-cube related literature. Subdivision methods similar to EVAL were proposed by Yakoubsohn and Dedieu [31, 8] and Pan (modifying Weyl's Algorithm) [21]. In contrast to EVAL or CEVAL, these algorithms rely mainly on an exclusion test (corresponding to C_0 in EVAL or CEVAL), but there is no confirmation test to ensure that a particular output box has exactly one root. We refer to [3] for further review of related literature.

§2. The Output Size Parameter r.

The introduction of the parameter r seems new in the context of root isolation. In computational geometry, such a parameter is typically called an output-size parameter. Then we call our complexity bound is **output-sensitive**. Note that L and d are global parameters, but r depends on the "local" argument I_0. We can thus view our bound as a "local bound". It is immediate that for $r \leq \log d$, our bound is optimal (regardless of whether $L \geq d$ or not). There is an important application of root finding that can exploit the parameter r: in computer graphics, the problem of ray shooting amounts to computing the smallest real root in the given interval I_0. We show that a simple modification of EVAL to this problem will achieve optimal tree size bounds for $L \geq \log d$.

[1] When $L \leq d$, some extra qualifications are needed.

2. PRELIMINARIES: ON EVAL ALGORITHM

Let us fix some notations. Throughout the paper, let $f(X) \in \mathbb{Z}[X]$ denote a square-free polynomial of degree d whose integer coefficients have absolute value $< 2^L$. So the main complexity parameters are $d \geq 1$ and $L \geq 1$.

Given an interval $I = [a, b]$ where $a \leq b$, let its midpoint and width be $m(I) := (a+b)/2$ and $w(I) := b - a$. Let P be a set $\{J_1, \ldots, J_m\}$ where each $J_i \subseteq I$. We call P a **partition** of I if $I = \cup_{i=1}^m J_i$, and each J_i is a finite union of intervals, and for $i \neq j$, the interiors of J_i and J_j are disjoint. In particular, $\{I\}$ is a partition of I, the **trivial partition**. Typically, P is just a set of intervals.

Our algorithms work by refining such trivial partitions. Suppose $J \in P$ is an interval. Then to **split** J in P means to replace $J = [a, b]$ by the two sub-intervals $[a, m(J)]$ and $[m(J), b]$. These two sub-intervals are called the **children** of J. Note that after splitting J in P, the size of P increases by 1.

We let $V = V(f) := \{z \in \mathbb{C} : f(z) = 0\}$ denote the set of zeros of f. Similarly, $V' = V(f') := \{z' \in \mathbb{C} : f'(z) = 0\}$ where f' is the derivative of f. The elements of V and V' are also called **roots** and **critical points** of f, respectively. For a general polynomial f, $V(f)$ is a multiset where each $x \in V$ has a multiplicity $\mu_V(x) \geq 1$. By extension, if x does not occur in V, we write $\mu_V(x) = 0$. If $\mu_V(x) = 1$ for all $x \in V$, then we say V is an **ordinary set**. If U, V are multisets, we write $U \subseteq V$ if $\mu_U(x) \leq \mu_V(x)$ for all x. The square-freeness of f implies V is an ordinary set, but it is important to remember that V' may not be an ordinary set.

An interval $J = [a, b]$ is **isolating** for f if one of two conditions hold:
(i) $a = b$ and $f(a) = 0$,
(ii) $a < b$ and $f(a)f(b) < 0$ and $|V \cap J| = 1$.

We are interested in isolating real roots, and write $V_{\mathbb{R}}$ and $V'_{\mathbb{R}}$ (resp.) for $V \cap \mathbb{R}$ and $V' \cap \mathbb{R}$. To isolate the roots of f in I_0 means to compute a set S of isolating intervals for f, such that $|S| = |V \cap I_0|$ and the intervals in S are pairwise disjoint.

Our EVAL algorithm is based on two interval predicates C_0 and C_1 where, for any interval J with width $w = w(J)$ and midpoint $m = m(J)$,

$$C_0(J) \equiv \quad |f(m)| \quad > \sum_{i \geq 1} \frac{|f^{(i)}(m)|}{i!} \left(\frac{w}{2}\right)^i$$

$$C_1(J) \equiv \quad |f'(m)| \quad > \sum_{i \geq 1} \frac{|f^{(i+1)}(m)|}{i!} \left(\frac{w}{2}\right)^i$$

It is easy to see that if $C_0(J)$ holds, then f has no zeros in J, and if $C_1(J)$ holds, then f' is monotone in J and hence has at most one zero in J. If $w(J) > 0$ and $C_1(J)$ holds, then clearly J is isolating iff $f(a)f(b) < 0$.

§3. The Subdivision Process of EVAL.

Given an input interval I_0, consider the simple iterative process:

> SUBDIVIDE(I_0):
> Initialize two queues, $Q \leftarrow \{I_0\}$ and $P \leftarrow \emptyset$.
> $\triangleright$ *Invariant: $P \cup Q$ is a partition of I_0*
> While Q is non-empty
> Remove an interval J from Q
> If $(C_0(J) \vee C_1(J))$ holds
> Push J into P.
> Else
> Push the children of J into Q.
> Return(P)

This "subdivision process" will terminate because f is square-free. Let $P(I_0)$ denote the partition of I_0 at the termination of this process. Our goal is to bound the size, denoted $\#P(I_0)$, of this partition. Alternatively, we view this process as producing a recursion tree $T(I_0)$ rooted at I_0, where each internal node is an interval J whose two children are obtained by bisecting J. Thus $P(I_0)$ is just the set of leaves of $T(I_0)$.

To turn this Subdivision Process into a root isolation algorithm, we just have to take the following additional actions during the while-loop, to detect and to output isolating intervals:

(1) Whenever we bisect an interval J, we check if the midpoint $m(J)$ is a root. If so, we output $[m(J), m(J)]$.

(2) For each interval $J = [a, b]$ that is pushed into queue P, we check if $C_1(J)$ holds and if $f(a)f(b) < 0$. If so, we output J.

Outside of the while-loop, we also check if the two endpoints of I_0 are roots, and output them accordingly. This completes the description of EVAL. For more details of EVAL see [4, 3, 27, 14].

It follows from the nature of binary trees that the size of $P(I_0)$ essentially controls the overall complexity of the EVAL algorithm. We had said that the main complexity parameters are d and L, but evidently the input I_0 has an influence on the size of $P(I_0)$. To remove this influence, we assume the first step of EVAL is to replace I_0 by $I_0 \cap [-2^L, 2^L]$, since all real roots of f lie in $[-2^L, 2^L]$ (e.g., [32]). We may henceforth assume that I_0 is contained in $[2^{-L}, 2^L]$.

3. MAIN RESULT AND APPLICATION TO RAY SHOOTING

Our main result is a bound on the size of the recursion tree $T(I_0)$:

THEOREM 1 (MAIN RESULT). *The subdivision tree $T(I_0)$ of EVAL has size*

$$O(d(L + r + \log d)) \tag{1}$$

where r is the number of real roots of f in I_0.

Most of the paper is devoted to proving this result. But here we give an application of the new bound.

§4. Finding the smallest root in I_0.
Ray-casting is an important primitive for rendering images in computer graphics. In computational geometry, this primitive is known as "ray shooting" and is used as a point sampling primitive in many algorithms (e.g., in computing an isotopic approximation of a manifold surface). Ray shooting can be reduced to the problem of computing the first (or smallest) root of a real function f in a given interval.

Let EVAL1 denote the following simple modification of EVAL to find the smallest root of f in interval I_0. If I_0 has no roots, EVAL1 will return an empty set (or any suitable indicator). We simply modify the Subdivision Process in §3 so that it always extract the leftmost interval in the queue Q in the while-loop. As soon as a root is detected, EVAL1 returns this root and terminates. Otherwise, it terminates when the queue Q is empty, returning the empty set.

Correctness of EVAL1 is clear. We only address the size of the recursion tree $T_1(I_0)$ produced by EVAL1.

THEOREM 2. *The recursion tree size of $EVAL1$ is $O(d(L+\log d))$.*

We will prove Theorem 2 after the proof of the main result. Its arguments depend on a sharp bound of $O(d(L + \log d))$ on the height of the recursion tree produced by EVAL1.

Clearly, $T_1(I_0)$ is a subtree of the recursion tree $T(I_0)$ of EVAL. Therefore, our bound on the height of $T_1(I_0)$ is implied by a similar bound on the height of $T(I_0)$, a fact that has independent interest. Note that a height bound on $T(I_0)$ follows simply from the size bound in [27] or our main theorem, but these are not sharp enough for Theorem 2.

THEOREM 3. *The height of the recursion tree $T(I_0)$ of EVAL (and hence EVAL1) is $O(d(L + \log d))$.*

The proof can be given here because it is independent of the main result.

Proof. Let $\sigma(\alpha)$ denote the distance of $\alpha \in V$ to the nearest root in $V \setminus \{\alpha\}$. Also, $\sigma(f) := \min\{\sigma(\alpha) : \alpha \in V\}$ be the root separation bound of f. It is well known (e.g., [32]) that $-\log \sigma(f) = O(d(L + \log d))$.

Let interval J be a leaf of $T(I_0)$. It suffices to show that

$$w(J) \geq \frac{1}{4d}\left(\frac{\sigma(f)}{2d}\right). \tag{2}$$

This would imply that the depth of J is at most

$$\begin{aligned} \lg\left(\frac{w(I_0)}{w(J)}\right) &\leq \lg\left(\frac{2L}{\sigma(f)/8d^2}\right) \\ &\leq \lg(16Ld^2) - \lg \sigma(f) \\ &= O(d(L + \log d)) \end{aligned}$$

as claimed.

By way of contradiction, assume (2) fails. Let J' be the parent of J in the recursion tree. Note that $C_0(J')$ and $C_1(J')$ must fail. Moreover,

$$w(J') \leq \frac{1}{2d}\left(\frac{\sigma(f)}{2d}\right). \tag{3}$$

Suppose $m = m(J)$ is the midpoint of J and $\alpha \in V$ is the closest root to m. From [27, Lemma 1(ii)], the failure of the predicate $C_0(J')$ implies that

$$|m - \alpha| \leq 2d \cdot w(J') \leq \frac{\sigma(f)}{2d}. \tag{4}$$

Let $\alpha' \in V'$ be the closest critical point to m. The analogue of [27, Lemma 1(ii)] for the failure of $C_1(J')$ implies that

$$|m - \alpha'| \leq 2(d-1) \cdot w(J') < \frac{\sigma(f)}{2d}. \quad (5)$$

Thus (4) and (5) implies $|\alpha - \alpha'| < \frac{\sigma(f)}{d}$. This contradicts a bound of Renegar [25] saying that the distance of α to any critical point is at least $\sigma(\alpha)/d \geq \sigma(f)/d$. **Q.E.D.**

4. THE INTEGRAL BOUND

The idea of "continuous amortization" is to introduce a continuous function $G : \mathbb{R} \to \mathbb{R}_{\geq 0}$ with the property that for any interval I, the subdivision process on I produces a partition of size at most $1 + \int_I G(x)dx$. Then we may call $G(x)$ a "charging function", in analogy to similar ideas in discrete amortization. Using this framework, our main result (1) is achieved in two steps:

(A) First we bound $\#P(I_0)$ by an integral $\int_{I_0} G(x)dx$ where $G(x)$ is an explicit function.

(B) Second we bound the integral $\int_{I_0} G(x)dx$ by $O(d(L + r + \log d))$, where r is the number of real roots of f in I_0.

What are suitable charging functions? Following [4], a function

$$G : \mathbb{R} \to \mathbb{R}_{\geq 0}$$

is called a **stopping function** (for EVAL) if for every interval I, if there is an $x \in I$ such that

$$w(I)G(x) \leq 1, \quad (6)$$

then $C_0(I)$ or $C_1(I)$ holds. Thus stopping functions can "predict" that certain interval J must be terminal for EVAL. The fundamental lemma below shows that stopping functions can serve as charging functions.

Lemma 4 (Integral Bound). *If $G(x)$ is a stopping function, then*

$$\#(P(I_0)) \leq \max\{1, 2\int_{I_0} G(x)dx\}. \quad (7)$$

The simple proof may be found in [4] or [3, Thm. 3.1, p. 160]. Our definition of stopping functions is an inconsequential variation of the original one: the inequality (6) would have been written "$w(J) \leq G(x)$" in [4, 3]. So our stopping functions are just reciprocals of original ones, and thus the bound in (7) comes from integrating stopping functions, not their reciprocals.

We must next provide an explicit stopping functions in order to apply the fundamental lemma. First, let us define the functions

$$S(x) = S_f(x) := \sum_{\alpha \in V} \frac{1}{|x - \alpha|} \quad (8)$$

and

$$S'(x) = S_{f'}(x) := \sum_{\alpha' \in V'} \frac{1}{|x - \alpha'|} \quad (9)$$

The original paper [4] gave a complicated stopping function which was not easy to bound. The key insight of [3] is that a slight modification of $S(x)$ and $S'(x)$ gives rise to stopping functions.

Lemma 5 (Burr-Krahmer). *The functions $3S(x)/2$ and $3S'(x)/2$ are stopping functions for EVAL.*

For completeness, we give a short direct argument of this Burr and Krahmer result, phrased somewhat more generally. Consider stopping functions of the form $G(x) = K \cdot S(x)$ for some constant $K > 0$. In Burr-Krahmer, $K = 3/2$. How small can K be?

Lemma 6. *Let $G(x) = K \cdot S(x)$. If $K > \frac{1+\ln 2}{2\ln 2} > 1.2213475$, then $G(x)$ is a stopping function.*

Proof. We must show that for any interval J with width $w = w(J)$ and midpoint $m = m(J)$, if there exists $x \in J$ such that

$$K \cdot S(x) \cdot w \leq 1 \quad (10)$$

then $C_0(J)$ holds. (An analogous argument for $S'(x)$ will show that $C_1(J)$ holds.) From (10) implies $S(x) = \sum_{\alpha \in V} 1/|x - \alpha| \leq 1/(Kw)$, and so for any $\alpha \in V$, we have $1/|x - \alpha| \leq 1/(Kw)$ or $|x - \alpha| \geq Kw$. Since $K > 1$, we know that $\alpha \notin J$ and hence $2|m - \alpha| > |x - \alpha|$ where $m = m(J)$. It follows that

$$|m-\alpha| \geq |x-\alpha| - |m-\alpha| \geq |x-\alpha| - w/2 \geq |x-\alpha| \left(1 - \frac{1}{2K}\right)$$

Thus $\frac{1}{|m-\alpha|} \leq \frac{2K}{|x-\alpha|(2K-1)}$. So $S(m) \leq S(x)\frac{2K}{2K-1}$. From (10) again, this implies

$$S(m) \leq S(x)\frac{2K}{2K-1} \leq \frac{2}{(2K-1)w}. \quad (11)$$

We now use the inequality that $|f^{(i)}(m)/f(m)| \leq S(m)^i$ for all $i \geq 1$ (e.g., [3, 27]). It follows that

$$\begin{aligned}
\sum_{i\geq 1} \left|\frac{f^{(i)}(m)}{f(m)}\right| \frac{(w/2)^i}{i!} &\leq \sum_{i\geq 1} S(m)^i \frac{(w/2)^i}{i!} \\
&\leq \sum_{i\geq 1} \left(\frac{2}{(2K-1)w}\right)^i \frac{(w/2)^i}{i!} \\
&= \sum_{i\geq 1} \frac{1}{(2K-1)^i i!} < e^{1/(2K-1)} - 1.
\end{aligned}$$

If we choose $K > \frac{1+\ln 2}{2\ln 2}$ then $1/(2K-1) < \ln 2$ and hence

$$\sum_{i\geq 1} \left|\frac{f^{(i)}(m)}{f(m)}\right| \frac{(w/2)^i}{i!} < e^{\ln 2} - 1 = 1.$$

From the definition of $C_0(J)$, we see that this last inequality is equivalent to the truth of the predicate $C_0(J)$. **Q.E.D.**

The constant $(1 + \ln 2)/(2\ln 2)$ in the lemma above could be further reduced by a tighter analysis if desired.

As noted in [4], if $G_0(x)$ and $G_1(x)$ are stopping functions, then so is $G(x) = \min\{G_0(x), G_1(x)\}$. Henceforth, we fix

$$G(x) := \min\{S(x), S'(x)\}. \quad (12)$$

By the previous lemma, $K \cdot G(x)$ is a stopping function for all $K > 1.23$. The minimization in (12) is important because it ensures that $G(x)$ is finite, i.e., $G(x) < \infty$ for all $x \in \mathbb{R}$. To see this, observe that $S(x)$ is infinite iff $x \in V \cap \mathbb{R}$. Similarly, $S'(x)$ is infinite iff $x \in V' \cap \mathbb{R}$. Since $V \cap V'$ is empty, the finiteness of $G(x)$ follows. Step (A) of our proof now follows from Lemmas 4 and 5, and is summarized by the following lemma:

LEMMA A.
EVAL produces a partition $P(I_0)$ whose size is bounded by the integral

$$\#P(I_0) \le \max\{1, 3\int_{I_0} G(x)dx\}$$

where $G(x) = \min\{S(x), S'(x)\}$.

5. BOUNDING THE INTEGRAL

In this section, we accomplish Step (B) which is to give an explicit bound on the integral in Lemma A. More precisely, we will show:

LEMMA B.

$$\int_{I_0} G(x)dx = O(d(L + r + \log d)) \tag{13}$$

where r is the number of real roots in I_0.

Towards proving Lemma B, we first bound the integral on $G(x)$ by the sum of two integrals on $S(x)$ and $S'(x)$, respectively. Suppose $\{I_1, I_1'\}$ is a partition of I_0 into two sets. Clearly, we have the inequality

$$\int_{I_0} G(x)dx \le \int_{I_1} S(x)dx + \int_{I_1'} S'(x)dx. \tag{14}$$

This inequality is trivial if any of the integrals on the right hand side is infinite. Finiteness of the integrals on the right hand side is equivalent to ensuring that $I_1 \cap V$ and $I_I' \cap V'$ are both empty sets. We will ensure this and some additional properties in forming the partition $\{I_1, I_1'\}$.

Assume $I_0 \cap V = \{\alpha_1, \ldots, \alpha_r\}$ and

$$a < \alpha_1 < \alpha_2 < \cdots < \alpha_r < b$$

where $I_0 = [a, b]$. We may also define $\alpha_0 := a$ and $\alpha_{r+1} := b$.

For each root $\alpha \in I_0$, we define the interval I_α as the intersection of real axis with the disc centered at α and radius equal to half the distance from α to the nearest critical point; note that two such intervals do not overlap, since by Rolle's theorem we have a critical point between any two roots in I_0. Finally, we define the sets I_1 and I_1':

$$I_1' := \bigcup_{\alpha \in I_0 \cap V} I_\alpha$$

and I_1 is just the closure of $I_0 \setminus I_1'$. It is easy to see that $I_1' \cap V' = \emptyset$, $I_1 \cap V = \emptyset$ and thus the right hand side of (14) is finite.

§5. Bounds on two basic integrals.

We will reduce our integrals to one of the two forms here:

LEMMA 7. *Let $\alpha \in \mathbb{C}$ and $J = [r, s] \subseteq I_0$. Assume $\alpha \notin J$.*
(Re) *If α is real, then*

$$\int_J \frac{dx}{|\alpha - x|} = \ln\left|\frac{\alpha - s}{\alpha - r}\right|^{\delta(J>\alpha)} \le L+1-\ln\min\{|\alpha-r|, |\alpha-s|\} \tag{15}$$

where $\delta(P) \in \{+1, -1\}$ is the Kronecker symbol: for any predicate P, $\delta(P) = +1$ if P holds, and $\delta(P) = -1$ otherwise.
(Im) *If α is not real, $\alpha = \mathtt{Re}(\alpha) + \mathbf{i}\mathtt{Im}(\alpha)$, then*

$$\begin{aligned}\int_J \frac{dx}{|\alpha - x|} &= \ln\left(\frac{(s - \mathtt{Re}(\alpha)) + |\alpha - s|}{(r - \mathtt{Re}(\alpha)) + |\alpha - r|}\right) \\ &\le \ln 4\left|\frac{(\alpha - s)(\alpha - r)}{\mathtt{Im}(\alpha)^2}\right| \\ &\le 2(2 + L - \ln|\mathtt{Im}(\alpha)|).\end{aligned} \tag{16}$$

Proof. **(Re)** From basic calculus we verify that (see [3, p. 162])

$$\int_r^s \frac{dx}{|\alpha - x|} = \ln\left|\frac{\alpha - s}{\alpha - r}\right|^{\delta(J>\alpha)}.$$

If $J > \alpha$ then $\int_r^s dx/|\alpha - x| = \ln(s-\alpha) - \ln(r-\alpha)$. If $J < \alpha$, we reverse the roles of r and s. But $\ln\max\{|\alpha-s|, |\alpha-r|\} \le 1 + L$, which gives us the desired upper bound in (15).
(Im) Writing $\alpha = \mathtt{Re}(\alpha) + \mathbf{i}\mathtt{Im}(\alpha) = R + \mathbf{i}I$, we have [3, p. 162]:

$$\int_r^s \frac{dx}{|\alpha - x|} = \operatorname{arcsinh}\left(\frac{s - R}{|I|}\right) - \operatorname{arcsinh}\left(\frac{r - R}{|I|}\right).$$

Since $\operatorname{arcsinh}(x) = \ln(x + \sqrt{1 + x^2})$ we conclude that

$$\int_r^s dx/|\alpha - x| = \ln\left(\frac{(s - R) + |\alpha - s|}{(r - R) + |\alpha - r|}\right)$$

where $|\alpha - s| = \sqrt{(R - s)^2 + I^2}$. The numerator $\ln((s - R) + |s - \alpha|) \le \ln(2|\alpha - s|)$, and the denominator

$$(r - R) + |\alpha - r| = \frac{|I|^2}{|\alpha - r| - (r - R)} \ge \frac{|I|^2}{2|\alpha - r|}.$$

Thus

$$\ln\left(\frac{(s - R) + |\alpha - s|}{(r - R) + |\alpha - r|}\right) \le \ln 4\left|\frac{(\alpha - s)(\alpha - r)}{I^2}\right|.$$

Since $|\alpha - s|, |\alpha - r| \le 2^{L+1}$, we obtain

$$\ln\left(\frac{(s - R) + |\alpha - s|}{(r - R) + |\alpha - r|}\right) \le 2((2 + L) - \ln|I|)$$

as claimed in (16). **Q.E.D.**

§6. Bounding the integral over I_1.

We bound the first integral on the RHS of (14) as follows:

$$\int_{I_1} S(x)dx = O(d(L + \log d)). \tag{17}$$

To show this, we express the integral as a sum over all roots α in V:

$$\int_{I_1} S(x)dx = \sum_{\alpha \in V} \int_{I_1} \frac{dx}{|x - \alpha|}. \tag{18}$$

The summand corresponding to a particular α can be bounded using one of the two cases in Lemma 7:
(Re) Suppose $\alpha \in \mathbb{R}$. Let $I_\alpha = [\alpha^-, \alpha^+]$ be the interval associated with α. Thus $(\alpha - \alpha^-) = (\alpha^+ - \alpha) = |\alpha - \alpha^*|/2$

where α^* is a critical point nearest to α. Writing $I_0 = [a, b]$, we can bound the summand with the help of Lemma 7(Re):

$$\begin{aligned}\int_{I_1} \frac{dx}{|x-\alpha|} &\le \int_{I_0\setminus I_\alpha} \frac{dx}{|x-\alpha|} \\ &= \int_a^{\alpha^-} \frac{dx}{\alpha - x} + \int_{\alpha^+}^b \frac{dx}{x-\alpha} \\ &\le (L+1-\ln|\alpha-\alpha^-|) + (L+1-\ln|\alpha-\alpha^+|) \\ &= 2(L+1) - 2\ln|\alpha-\alpha^-| \\ &= 2(L+1) - 2\ln\frac{|\alpha-\alpha^*|}{2}.\end{aligned}$$

Summing over all real roots $\alpha \in V$, yields $2d(L+1) - 2\ln\prod_{\alpha\in V}|\alpha-\alpha^*|/2$, which is equal to $O(d(L+\log d))$ from Mahler-Davenport [7, 12, 9].

(Im) Suppose $\alpha \notin \mathbb{R}$. Then Lemma 7(Im) says

$$\int_{I_1} \frac{dx}{|x-\alpha|} \le \int_{I_0} \frac{dx}{|x-\alpha|} \le 2(2+L-\ln|\mathtt{Im}(\alpha)|).$$

Again, summing over all non-real $\alpha \in V$ and using the Mahler-Davenport bound we get that (18) is bounded by $O(d(L+\log d))$.

Cases **(Re)** and **(Im)** imply the desired bound in (17).

§7. Bounding the integral over I_1'.

It remains to bound the second integral on the RHS of (14) as follows:

$$\int_{I_1'} S'(x)dx = O(dr). \tag{19}$$

This integral is written as a double summation, summing over all critical points $\alpha' \in V'$, and summing over all $\alpha \in V \cap I_0$:

$$\begin{aligned}\int_{I_1'} S'(x)dx &= \sum_{\alpha'\in V'} \int_{I_1'} \frac{1}{|x-\alpha'|}dx \\ &= \sum_{\alpha'\in V'} \sum_{\alpha\in V\cap I_0} \int_{I_\alpha} \frac{1}{|x-\alpha'|}dx.\end{aligned} \tag{20}$$

Fix a particular root α and critical point α'. Write $I_\alpha = [\alpha^-, \alpha^+]$, and let α^* be a critical point nearest to α; since α is equidistant from α^+ and α^-, we express this distance as $|\alpha - \alpha^\pm|$. There are again two cases to consider.

(Re') Suppose α' is real. Then Lemma 7(Re) yields

$$\int_{I_\alpha} \frac{1}{|x-\alpha'|}dx = \ln\left|\frac{\alpha'-\alpha^+}{\alpha'-\alpha^-}\right|^{\delta(I_\alpha > \alpha')}. \tag{21}$$

By the triangular inequality

$$|\alpha'-\alpha^\pm| \ge |\alpha'-\alpha| - |\alpha-\alpha^\pm| = |\alpha'-\alpha| - \frac{|\alpha-\alpha^*|}{2}. \tag{22}$$

Since α^* is a critical point nearest to α it further follows that

$$|\alpha'-\alpha^\pm| \ge |\alpha'-\alpha| - \frac{|\alpha-\alpha'|}{2} = |\alpha'-\alpha|/2. \tag{23}$$

Similarly, we can show $|\alpha'-\alpha^\pm| \le 2|\alpha'-\alpha|$. Thus

$$\frac{|\alpha'-\alpha|}{2} \le |\alpha'-\alpha^\pm| < 2|\alpha'-\alpha|. \tag{24}$$

Note that these inequalities are independent of the fact that $\alpha' \in \mathbb{R}$. Applying these inequalities to the RHS of (21) we obtain that the integral on the LHS is at most $\ln 4$.

(Im') Suppose $\alpha' \notin \mathbb{R}$. Here, we recognize three subcases (i) $I_\alpha < \mathtt{Re}(\alpha')$, (ii) $I_\alpha > \mathtt{Re}(\alpha')$, and (iii) $\mathtt{Re}(\alpha') \in I_\alpha$. For the first two subcases, we know from [3] that

$$\int_{I_\alpha} \frac{dx}{|x-\alpha'|} \le \ln 2\left|\frac{\alpha'-\alpha^+}{\alpha'-\alpha^-}\right|^{\delta(I_\alpha > \mathtt{Re}(\alpha))}.$$

Furthermore, the bounds from (24) imply that the integral above is bounded by $\ln 8$. In the third subcase, Lemma 7(Im) yields

$$\int_{I_\alpha} \frac{dx}{|x-\alpha'|} \le \ln 4\left|\frac{(\alpha'-\alpha^+)(\alpha'-\alpha^-)}{\mathtt{Im}(\alpha')^2}\right|.$$

Applying the upper bound from (24), we further get

$$\int_{I_\alpha} \frac{dx}{|x-\alpha'|} < \ln 16\frac{|\alpha'-\alpha|^2}{|\mathtt{Im}(\alpha')|^2}.$$

From the triangle inequality it follows that $|\mathtt{Im}(\alpha')| \ge |\alpha'-\alpha| - |\alpha - \mathtt{Re}(\alpha')|$. Since $\mathtt{Re}(\alpha') \in I_\alpha$, we further have $|\mathtt{Im}(\alpha')| \ge |\alpha'-\alpha| - |\alpha-\alpha^\pm|$, which we know from (22) and (23) is greater than $|\alpha'-\alpha|/2$. Thus we have $|\mathtt{Im}(\alpha')| \ge |\alpha'-\alpha|/2$.

Therefore the integral $\int_{I_\alpha} dx/|x-\alpha'|$ in subcase (iii), hence in all subcases, is at most $\ln 64$.

Thus cases **(Re')** and **(Im')** imply that each integral in the RHS of (20) is at most $\ln 64$. If r is the number of real roots in I_0 then we have

$$\int_{I_1'} S'(x)dx < (d-1)r\ln 64.$$

This proves (19).

§8. Wrapping up the Proofs.

To complete the proof of the Main Theorem (Theorem 1), we have to wrap up the arguments for Lemma B. This is accomplished by plugging in the bounds (17) and (19) into (14).

It remains to prove Theorem 2. Let $r = |I_0 \cap V|$. If $r = 0$, our theorem follows from the main result. Hence assume $r > 0$ and EVAL1 returns $J = [c, d] \subseteq I_0$ as the isolating interval for the smallest root in I_0. See Figure 1.

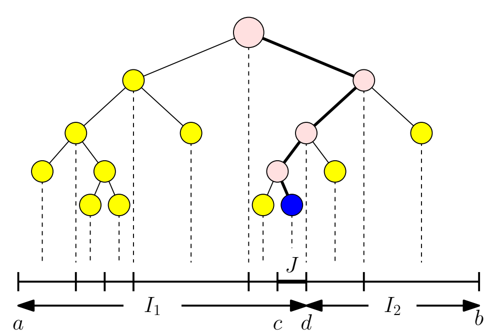

Figure 1: Subdivision of $[a, b]$ by EVAL1

Let $I_0 := [a, b]$. We split I_0 into $I_1 := [a, d]$ and $I_2 := [d, b]$. Consider the partition $P(I_0)$ of I_0 produced by EVAL1. Let P_1 comprise those intervals of $P(I_0)$ that are contained in

$[a,d]$, and $P_2 := P(I_0) \setminus P_1$. Clearly, P_i is a partition of I_i (for $i = 1, 2$). Theorem 2 is a consequence of the following claim:

$$\#P_1, \#P_2 = O(d(L + \log d)).$$

Recall that $T_1(I_0)$ is the recursion tree of EVAL1 on I_0.

We first bound $\#P_1$. Consider any interval $I \in P_1$. At the parent of I, both the predicates C_0 and C_1 failed. Since $G(x) := 3\min\{S(x), S'(x)\}/2$ is a stopping function for EVAL1, it follows from Lemma A that $\#P_1 = O(\max\{1, \int_{I_1} G(x)dx\})$. Then Lemma B implies that $\#P_1 = O(d(L + \log d))$, since there is only one root in I_1.

To bound $\#P_2$, we observe that an interval in P_2 is the right-child of a node in the path (see Figure 1) from I_0 to the leaf $J = [c,d]$ in $T_1(I_0)$. Thus $\#P_2$ is bounded by the depth of J, which we know from Theorem 3 is $O(d(L + \log d))$.

6. CONCLUSION

In this paper, we give a new bound on the complexity of EVAL that is the sharpest in the range $L \geq d$. This result, along with that of Burr-Krahmer [3], has two significance:

- It is contribution to the continuous amortization technique, one of the few non-probabilistic framework for analysis of subdivision algorithms. The development of this and other general techniques for analyzing adaptive iterative algorithms in geometry and algebra is an important but relatively new topic in theoretical computer science.

- Although EVAL has independent interest for root isolation, we view its main significance as a paradigm for a whole class of algorithms: from its extension to complex roots [27], to its higher dimensional analogues[2] for approximating curves and surfaces [27, 22, 16].

The natural open problem is to extend this complexity analysis to algorithms in higher dimensions. Another issue concerns the sharpness of our bound. The interesting range is $\log d < L < d$. Here, we would like to replace the d^2 term in our complexity by $d \log d$. This seems quite challenging given our current understanding of continuous amortization; but there seems to be some possibilities using the alternative approach in [27].

7. REFERENCES

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

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

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

[4] M. Burr, F. Krahmer, and C. Yap. Continuous amortization: A non-probabilistic adaptive analysis technique. *Electronic Colloquium on Computational Complexity (ECCC)*, TR09(136), December 2009.

[2] Historically, our investigation of EVAL came *after* studying the higher dimensional algorithms of Plantinga-Vegter.

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

[6] T. H. Corman, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* The MIT Press and McGraw-Hill Book Company, Cambridge, Massachusetts and New York, second edition, 2001.

[7] J. H. Davenport. Computer algebra for cylindrical algebraic decomposition. Tech. Rep., The Royal Inst. of Technology, Dept. of Numerical Analysis and Computing Science, S-100 44, Stockholm, Sweden, 1985. Reprinted as Tech. Report 88-10 , School of Mathematical Sci., U. of Bath, Claverton Down, Bath BA2 7AY, England. URL http://www.bath.ac.uk/˜masjhd/TRITA.pdf.

[8] J.-P. Dedieu and J.-C. Yakoubsohn. Localization of an algebraic hypersurface by the exclusion algorithm. *Applicable Algbebra in Engineering, Communication and Computing*, 2:239–256, 1992.

[9] Z. Du, V. Sharma, and C. Yap. Amortized bounds for root isolation via Sturm sequences. In D. Wang and L. Zhi, editors, *Symbolic-Numeric Computation*, Trends in Mathematics, pages 113–130. Birkhäuser Verlag AG, Basel, 2007. Proc. Int'l Workshop on Symbolic-Numeric Computation, Xi'an, China, Jul 19–21, 2005.

[10] A. Eigenwillig, V. Sharma, and C. Yap. Almost tight complexity bounds for the Descartes method. In *31st Int'l Symp. Symbolic and Alge. Comp. (ISSAC'06)*, pages 71–78, 2006. Genova, Italy. Jul 9-12, 2006.

[11] M. Hemmer, E. P.Tsigaridas, Z. Zafeirakopoulos, I. Z. Emiris, M. I. Karavelas, and B. Mourrain. Experimental evaluation and cross-benchmarking of univariate real solvers. In *SNC '09: Proc. 2009 Conf. on Symbolic-Numeric Computation*, pages 45–54, New York, NY, USA, 2009. ACM.

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

[13] N. Kamath. Subdivision algorithms for complex root isolation: Empirical comparisons. Master's thesis, Oxford University, Oxford Computing Laboratory, Aug. 2010.

[14] N. Kamath, I. Voiculescu, and C. Yap. Empirical study of an evaluation-based subdivision algorithm for complex root isolation. In *4th Intl. Workshop on Symbolic-Numeric Computation (SNC)*, pages 155–164, 2011.

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

[16] L. Lin and C. Yap. Adaptive isotopic approximation of nonsingular curves: the parameterizability and nonlocal isotopy approach. *Discrete and Comp. Geom.*, 45(4):760–795, 2011.

[17] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In M. C. Stone, editor, *Computer Graphics (SIGGRAPH*

'87 Proceedings), volume 21, pages 163–169, July 1987.

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

[19] R. E. Moore. *Interval Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1966.

[20] A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge, 1990.

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

[22] S. Plantinga and G. Vegter. Isotopic approximation of implicit curves and surfaces. In *Proc. Eurographics Symposium on Geometry Processing*, pages 245–254, New York, 2004. ACM Press.

[23] H. Ratschek and J. Rokne. *Computer Methods for the Range of Functions*. Horwood Publishing Limited, Chichester, West Sussex, UK, 1984.

[24] D. Reischert. Asymptotically fast computation of subresultants. In *ISSAC 97*, pages 233–240, 1997. Maui, Hawaii.

[25] J. Renegar. On the worst-case arithmetic complexity of approximating zeros of polynomials. *Journal of Complexity*, 3:90–113, 1987.

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

[27] M. Sagraloff and C. K. Yap. A simple but exact and efficient algorithm for complex root isolation. In *36th Int'l Symp. Symbolic and Alge. Comp. (ISSAC'11)*, pages 353–360, 2011. June 8-11, San Jose, California.

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

[29] J. M. Snyder. Interval analysis for computer graphics. *SIGGRAPH Comput.Graphics*, 26(2):121–130, 1992.

[30] R. E. Tarjan. Amortized computational complexity. *SIAM J. on Algebraic and Discrete Methods*, 6:306–318, 1985.

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

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