CONTINUOUS AMORTIZATION: A NON-PROBABILISTIC ADAPTIVE ANALYSIS TECHNIQUE*

MICHAEL BURR†, FELIX KRAHMER‡, AND CHEE YAP†

Abstract.
Let f be a univariate polynomial with real coefficients, $f \in \mathbb{R}[X]$. Subdivision algorithms based on algebraic techniques (e.g., Sturm or Descartes methods) are widely used for isolating the roots of f in a given interval. In this paper, we consider subdivision algorithms based on purely numerical primitives such as function evaluation. Such methods have adaptive complexity, are local, and are also applicable when f is transcendental. The complexity analysis of adaptive algorithms is a new challenge for computer science. In this paper, we introduce a form of continuous amortization for adaptive complexity.

Our analysis is applied to an evaluation-based root isolation algorithm called EVAL. EVAL is based on an algorithm of Mitchell and can also be seen as a 1-dimensional analogue of algorithms by Plantinga and Vegter for meshing curves and surfaces. The algorithm itself is simple, but its complexity analysis is not. Our main result is an $O(d^3(\log d + L))$ bound on the subdivision-tree size of EVAL for the benchmark problem of isolating all real roots of a square-free integer polynomial f of degree d and logarithmic height L.

Our proof introduces several novel techniques: First, we provide an adaptive upper bound on the complexity of EVAL using an integral, analogous to integral bounds provided by Ruppert in a different context. Such integrals can be viewed as a form of continuous amortization. In addition, we use two algebraic amortization techniques: one is based on the standard Mahler-Davenport root bounds, but the other, based on evaluation bounds, is new.

Key words. Continuous Amortization, Adaptive Analysis, Subdivision Algorithms, Integral Analysis, Amortization, Root Isolation

AMS subject classifications. 65Y20, 68W40

1. Introduction.

The analysis of algorithms is a highly developed area of theoretical computer science. Current analysis techniques are mostly aimed at *combinatorial* and *discrete* algorithms. But in Computational Science & Engineering (CS&E) applications, *numerical* and *continuous* algorithms predominate. An important paradigm for these continuous and numerical algorithms is iteration (e.g., Newton's method). Iteration can be combined with decomposition of the computational domain; this technique is known as the **subdivision method** in the computational literature[1] for curves and surfaces. A famous example is the Marching Cube algorithm [22]. Subdivision methods typically use termination criteria based on local, a posteriori ε-parameters; in this sense, these algorithms are described as **adaptive**. The standard worst-case analysis techniques are inappropriate for adaptive algorithms. A serious challenge to theoretical computer science is to provide new complexity analyses that can account for the adaptive behavior of numerical algorithms. This would open up the vast territory of algorithms in CS&E for theoretical algorithms develop-

* This work is supported in part by NSF Grants CCF-0728977 and CCF-0917093. Yap's work is also partly supported by Korea Institute of Advanced Study.

†Courant Institute, New York University, 251 Mercer St., New York, NY 10012 ({burr,yap}@cims.nyu.edu)

‡Hausdorff Center for Mathematics, Endenicher Allee 60, 53115 Bonn, Germany (felix.krahmer@hcm.uni-bonn.de)

[1]The subdivision terminology has several related but distinct uses. For instance, there are subdivision surfaces from mesh refinement. One could perform subdivision in parameter space (e.g., Bezier subdivision), but we focus on "domain subdivision", a multidimensional generalization of binary search. The multidimensional search literature often focuses on the data structures (e.g., quadtree, k-d tree), but we focus on the algorithmic paradigm of subdivision.

ment. Such a challenge arose in the 1980s, after Klee and Minty showed that it is impossible to account for the generally good performance of the simplex algorithm by using a worst-case analysis. Starting from the work of Borgwardt, Smale and others [4], a variety of probabilistic analyses were able to provide polynomial-time bounds. Nevertheless, the probabilistic assumptions in such results are considered unsatisfactory and arbitrary. The acclaimed smoothed analysis of Spielman and Teng [40] tries to overcome such objections by "localizing" the probabilistic assumptions to the input instance. As far as the authors are aware, there has been no previous non-probabilistic methods for analyzing adaptive algorithms. The present paper proposes such a non-probabilistic method.

The main contribution of this paper is the development of a complexity analysis technique that can be applied to a large class of subdivision algorithms. We illustrate its application on a practical real root isolation algorithm. The algorithm is chosen for its simplicity and its structural similarities to other subdivision algorithms. Our analysis of this algorithm, however, is nontrivial and exhibits most of the necessary elements needed to extend it to other subdivision algorithms of this type. Similar subdivision algorithms are prevalent in graphics, geometric modeling applications, and scientific computation (e.g., [10]). They are easy to implement because they typically use numerical primitives and simple data structures, such as quadtrees. Unfortunately, this power makes such algorithms difficult to analyze.

In this paper, the complexity of a subdivision algorithm refers to the number of subdivisions performed by the algorithm, also called the **recursion tree size**. Kearfott [18] has a general complexity analysis of subdivision methods for finding zeros of multidimensional systems. His recursion tree size bound is based only on the maximum depth of subdivision. Such bounds, however, may be exponential in the true recursion tree size because they cannot account for adaptivity: subdivision algorithms will spend more time subdividing near more difficult features, generating a few deep paths in the tree, while the overall tree size remains modest in size. Indeed, in this paper, we show that the overall tree size is always polynomial in the depth.

The recursion tree size should be bounded in terms of intrinsic parameters of the input. A central question is: *How should we quantify such parameters?* The most common approach is based on the condition number of the problem, and is used extensively in the Smale school [3]. A second approach is based on precision sensitivity [36, 2]: the bit-version of output sensitivity, which is well-known in computational geometry [14]. The third approach, which is expanded in this paper, is based on defining an adaptivity measure on the input instance via an integral. In computational geometry, such an integral was introduced by Ruppert [34] via a local feature size function.

1.1. Continuous Amortization. Amortization is a well-known computational paradigm and analysis technique in discrete algorithms [11]. We can view the integral approach as a continuous form of amortization: in fact, if the number of subdivisions in a region R is bounded above by an integral $I = \int_{x \in R} \phi(x)dx$, then the value I can be interpreted as an amortized complexity, where the complexity charge $\phi(x)$ is distributed over the region R. Intuitively, integral approaches represent a kind of averaging, while condition number approaches are a worst-case measure. For instance, the condition number is infinite for singular inputs, while it is possible to have a finite bound on an improper integral. Although Ruppert's integral approach is well-known in Computational Geometry, its application has been restricted to proving that the work done by certain algorithms achieves the bound of this integral. An explicit

bound on his integral is not computed in such results. In this paper, our challenge is to convert an integral bound into an explicit *a priori* complexity bound.

1.2. Algebraic Amortization. In order to ultimately bound the above integral for our problem, we must use amortization in yet another form: algebraically rather than continuously. Algebraic amortization originated with Davenport [13]: the idea is to replace individual root separation bounds by a bound for a product of such root separations. In our previous work, we had extended such arguments to other Sturm methods [15] and to the Descartes method [16]. These arguments are based on the Mahler-Davenport bound [13, 42]. In this paper, we need the Mahler-Davenport bound in a form that was stronger than was previously used. Moreover, we develop another algebraic amortized bound based on evaluation bounds. Recently, Cheng et al. [7] also used evaluation bounds, but in a non-amortized, multivariate setting.

1.3. The EVAL algorithm. We illustrate our technique by providing an analysis for EVAL, a simple but practical evaluation-based subdivision algorithm for real root isolation of a polynomial $f \in \mathbb{R}[X]$. Our version is exact and based on bigfloat computations; the fixed-precision form of this algorithm was first formulated by Mitchell [24] who, in turn, based it on an algorithm of Moore [25, p. 62]. We first came to the EVAL algorithm by way of specializing the 2- and 3-D meshing algorithm of Plantinga-Vegter [29] to 1-D, in our effort to extend their work to treat singularities. We achieved such extensions in [6] (for 1-D) and in [5] (for 2-D). There are many well-known subdivision algorithms for real root isolation, e.g., the Sturm method [31, 21, 15] or the Descartes method [8, 16, 20, 9]. These methods use sophisticated algebraic primitives, thereby restricting f to polynomials. On the other hand, the numerical primitives in EVAL allow us in [6] to extend its reach beyond polynomials (e.g., to hypergeometric functions).

The underlying principle of EVAL is the Bolzano theorem (a special case of the Intermediate Value Theorem): if f is continuous, $f(a)f(b) < 0$ and $a < b$ then there is some $c \in (a, b)$ such that $f(c) = 0$. So we may regard EVAL as an instance of the Bolzano method (in analogy to the Sturm or Descartes methods). The computational model for the Bolzano method is purely numerical: the primitives are the evaluation of a function and interval evaluations of the function and its derivatives. In terms of the complexity of the primitives, we see the progression $STURM > DESCARTES > BOLZANO$. This suggests that the Bolzano method would be more adaptive than Descartes, just as Descartes is more adaptive than Sturm [17].

It is standard to judge the complexity of root isolation algorithms using the benchmark problem of isolating all the real roots of an integer polynomial of degree d and logarithmic height L. Over 20 years ago, Davenport [13] proved that the tree size is $O(d(\log d + L))$ for the Sturm method. Only recently has it been shown that the Descartes method achieves the same bound [16]. Both methods are optimal for $L \geq \log d$ [16]. In this paper, we prove that EVAL has a complexity of $O(d^3(\log d + L))$. While this bound is far from optimal, the achievement is that we provide a polynomial bound in d, L whereas the only previously known bound for the complexity of this algorithm is a trivial exponential bound.

1.4. Related Work. There is a vast literature on the complexity of root isolation (e.g., Pan [27]). For the benchmark problem, the bit-complexity of $O(d^3(\log d + L))$ for complex roots was first achieved by Schönhage [35]. There are improved bounds when we count arithmetic operations, but they do not surpass Schönhage's bound when converted to bit-complexity. These algorithms aim at finding all roots of

a polynomial (so they are optimized for the benchmark problem). In contrast, subdivision methods have the important *local property* (i.e., they can isolate roots in any given region). In the algebraic computing community, the Descartes method appears to be most practical [9, 17, 33, 26, 33].

The importance of our result is that it points the way to the amortized analysis of subdivision algorithms for curves and surfaces. This is an area for which there are many practical adaptive algorithms but virtually no suitable analysis. We hope that the analysis in this paper will serve as an example and provide a framework of the analysis for these practical adaptive algorithms. We draw an analogy in this situation with the field of linear programming: the simplex algorithm is highly successful in practice but traditional complexity analysis is unable to account for this. This led to the fruitful work of Smale, Borgwardt and others, including the smoothed analysis of Spielman and Teng [40]. Our work follows this tradition except that, for the first time, we break out of the probabilistic framework.

1.5. Overview of Paper. In §2, we describe our computational model and the EVAL algorithm. In §3, we introduce the main result and the integral bound on the number of subdivisions of the EVAL algorithm. In §4, we prove an amortized evaluation bound that is used in the rest of the paper. In §5 and §6, we provide the two bounds needed to complete the proof of our main result. We conclude in §7. An appendix provides all missing proofs.

2. An Evaluation-based Algorithm. Fix f to be a square-free polynomial in $\mathbb{R}[X]$ of degree d. In the Plantinga-Vegter computational model, we use the box (i.e., interval) version of f and its derivatives.

2.1. Box Functions. For any set $S \subseteq \mathbb{R}$, define $\Box S$ to be the set of closed intervals whose endpoints lie in S. For interval $I = [a, b]$, let $m(I) := (a + b)/2$ and $w(I) := b - a$ be (resp.) the midpoint and width of I. A partition of I is a finite subset $P \subseteq \Box I$ of non-overlapping intervals whose union is I. The size $\#(P)$ of P is the number of intervals in P. Our partitions mostly come from repeated bisections: Let P be a partition and $X \in P$, then to **bisect** X **in** P means to form a new partition in which $X = [c, d]$ is replaced by its two children, i.e., the two intervals $[c, m(X)], [m(X), d]$. As a result, $\#(P)$ increases by 1. A partition of I resulting only from repeated bisections of $\{I\}$ is called a **subdivision** of I.

It is important to clarify the underlying computational model for our algorithms: We assume all computation is reduced to interval arithmetic over **bigfloats**, i.e., the set $\mathbb{F} := \{m2^n : m, n \in \mathbb{Z}\} = \mathbb{Z}\left[\frac{1}{2}\right]$. Thus, the operations $(\pm, \times)$ and bisection are all exact.

A box function for f on I is a function $\Box f : \Box(I \cap \mathbb{F}) \to \Box\mathbb{F}$ such that for all $X \in \Box(I \cap \mathbb{F})$, $f(X) \subseteq \Box f(X)$. Here, $f(X) := \{f(a) : a \in X\}$ is the set extension of f, and $\Box f$ is a function applied to intervals and not $\Box$ applied to the set $f(X)$. To ensure the termination of the EVAL algorithm, it is assumed that $\Box f$ is continuous, i.e., if $X_1, X_2, \cdots$ is a strictly decreasing sequence of intervals whose limit is a point p, then the limit of the images $\Box f(X_i)$ is $f(p)$.

2.2. The Evaluation Algorithm. We now present the Evaluation Algorithm, EVAL, also discussed in [6]. Given an interval $I = [a, b]$, EVAL isolates all the real roots of $f(x)$ in the interval (a, b). Specifically, it outputs a sequence of pairwise-disjoint isolating intervals, one for each real root of f in the interval. The isolating intervals are either of the form $[c, c]$, implying that $f(c) = 0$, or $[c, d]$, implying that

there is a root in (c, d). The idea is to maintain a subdivision P of I. Initially, $P = \{I\}$. The algorithm operates in two phases.

PHASE 1: Repeatedly bisect each X in P until each interval in P is EVAL-terminal. An interval is **EVAL-terminal** if one of the following two conditions hold for X:

$$C_0(X): \quad 0 \notin \square f(X)$$
$$C_1(X): \quad 0 \notin \square f'(X)$$

If, when subdividing, $f(m(X)) = 0$, then output $[m(X), m(X)]$.

PHASE 2: Let P_{EVAL} be the subdivision of I at the end of Phase 1. For each $X \in P_{EVAL}$, take one of two actions: If $C_0(X)$ holds, discard X. If $C_1(X)$ holds, evaluate the sign of f at the two end points of $X = [c, d]$. If $f(c)f(d) < 0$ output $[c, d]$, else discard X.

This algorithm terminates because f has simple roots in I (as f is square-free): For if the algorithm did not terminate, then there would be a path of infinite length in the subdivision tree. The intervals along this path would form a decreasing sequence of intervals converging to a point p; therefore, $\square f$ and $\square f'$ applied to these intervals would converge to $f(p)$ and $f'(p)$ (resp.) at least one of which is nonzero. Its correctness then follows from the definition of $\square f$ and the Bolzano theorem (cf. §1.3).

2.3. Centered Form Interval Functions. For our complexity results, we need stronger convergence properties for $\square f$; we use the centered form box function [30], defined as follows:

$$\square f(X) := f(m(X)) + \sum_{i=1}^{d} \frac{|f^{(i)}(m(X))|}{i!} \left(\frac{w(X)}{2}[-1, 1]\right)^i. \tag{2.1}$$

where $d = \deg f$ and $\left(\frac{w(X)}{2}[-1, 1]\right)^i$ is an interval arithmetic expression. By consistently multiplying through by $d!$, we have the same results and can keep all calculations in $\mathbb{F}$. For experimental comparison of centered forms with other box functions, see [23, 41]. The centered form box function satisfies the following conditions:

PROPOSITION 2.1 ([30]). *Let $Y \subseteq X$ be intervals, then there exists a positive number K_X, the* **Lipschitz constant** *such that:*
(Inclusion isotone) $\square f(Y) \subseteq \square f(X)$.
(Lipschitz continuous) $w(\square f(Y)) \leq K_X \cdot w(Y)$.
(Quadratic convergent) $w(\square f(Y)) - w(f(Y)) \leq K_X \cdot w(Y)^2$.

A Lipschitz constant for the centered form can be computed in the following way:

$$K_X = K_X(f) := \max_{a \in X} \sum_{i=1}^{d} \frac{|f^{(i)}(a)|}{i!} (w(X))^{i-1}. \tag{2.2}$$

The definitions of $\square f'$ and $K'_X = K_X(f')$ are obtained by replacing f by f' in (2.1) and (2.2).

Our goal is to find an upper bound for the size $\#(P_{EVAL})$, which is one more than the number of bisection steps. We begin our analysis with a simple observation:

LEMMA 2.2. *Let $Y \subseteq X$ be intervals. If $a \in Y$ and $0 \in \square f(Y)$ then $w(Y) \geq |f(a)|/K_X$.*

Proof. Since $\{0, f(a)\} \subseteq \square f(Y)$, we have $w(\square f(Y)) \geq |f(a)|$. By the Lipschitz property, $w(Y) \geq w(\square f(Y))/K_X$ and hence $w(Y) \geq |f(a)|/K_X$. □

3. An Integral Bound. In the remainder of this paper, we will analyze the complexity of the EVAL algorithm for the benchmark problem where $f \in \mathbb{Z}[X]$ is square-free, of height $\|f\| < 2^L$, and the endpoints of $I = [a, b]$ are integers. The height $\|f\|$ is the maximum absolute value of the coefficients of f and the logarithmic height is $L = \log_2 \|f\|$. We assume that $a, b \leq 2^L$ since all real zeros of f lie in this range [42]. We assume for simplicity that f' is square free; the removal of this assumption does not affect the final result, but requires a delicate construction, a sketch of which appears in §6.1

THEOREM 3.1 (Main Result). *The number of bisections performed by EVAL on input f and interval I is $O(d^3(\log d + L))$.*

Under the mild assumption of $L \geq \log d$, this bound becomes $O(d^3 L)$. This should be compared to the optimal bound of $O(dL)$ known for the Sturm and Descartes methods [16]. Our proof exploits the *gamma function* that is central in Smale's theory of point estimates [3, 37], defined as:

$$\gamma(x) = \gamma_f(x) := \max_{i \geq 2} \left(\frac{|f^{(i)}(x)|}{i!|f'(x)|} \right)^{1/(i-1)}.$$

Intuitively, $\gamma(x)^{-1}$ is the radius of Newton convergence of f at x. We write $\gamma'(x)$ for $\gamma_{f'}(x)$; in the literature, $\gamma'(x)$ is also written as $\gamma_2(x)$. $\gamma'(x)$ should not be confused with the derivative of $\gamma(x)$, which is not used in this paper.

LEMMA 3.2. *Suppose that there exists a $b \in J$ such that $w(J) \leq \frac{1}{2\gamma(b)}$. Then $K_J \leq 2d|f'(b)|$.*

This is proved in the Appendix by replacing each $f^{(i)}(a)$ in the definition (2.2) of K_J by its Taylor expansion at b and then applying the triangle inequality. We now introduce the function that we use in our analysis:

$$G(a) := \min \left\{ \frac{1}{2\gamma(a)}, \frac{|f(a)|}{2d|f'(a)|} \right\}. \tag{3.1}$$

LEMMA 3.3. *Let J be an interval. If $\exists b \in J$ such that $w(J) < G(b)$, then J is EVAL-terminal. In fact, $C_0(J)$ holds.*

Proof. Since $w(J) < \frac{1}{2\gamma(b)}$, Lemma 3.2 implies that $K_J \leq 2d|f'(b)|$. This inequality combined with $w(J) < \frac{|f(b)|}{2d|f'(b)|}$ implies that $w(J) < \frac{|f(b)|}{K_J}$. By Lemma 2.2, $C_0(J)$ holds. □

Consider a conceptual algorithm, GEN. This algorithm is conceptual because it is not meant to be implemented. It is introduced as a tool for analysis, for comparison to the EVAL algorithm. The GEN algorithm is similar to PHASE 1 of the EVAL algorithm, and we describe it analogously. Given an interval $I = [a, b]$, GEN subdivides I until the condition of Lemma 3.3 holds for all intervals. We maintain a subdivision P of I. Initially $P = \{I\}$.

PHASE: Repeatedly bisect each X in P until each interval X in P is **GEN-terminal**. By this we mean that the following condition holds:

$$G_0(X) : \exists b \in X \text{ such that } w(X) \leq G(b)$$

This algorithm is called GEN for generic as it is a member of a class of generic subdivision algorithms that include PHASE 1 of the EVAL algorithm. In the next section, we explore this class of algorithms in more depth and show how they can be used to bound other subdivision based algorithms. Technically, GEN is not an

algorithm because it does not terminate near roots of f since $G(a)$ cannot be used to ensure that $C_1(J)$ holds. We address the termination problem §3.2. If the algorithm terminates, then let P_{GEN} be the partition at the end of the GEN algorithm.

THEOREM 3.4. *If the GEN algorithm terminates on I, then the following inequalities hold:*

$$\#P_{EVAL} \overset{(i)}{\leq} \#P_{GEN} \overset{(ii)}{\leq} \max\left\{1, \int_I \frac{2da}{G(a)}\right\}$$

If GEN does not terminate, then the integral on the right is infinite.

Proof. (i) If $X \in P_{GEN}$, then from Lemma 3.3, X is EVAL-terminal. Therefore EVAL must terminate at some node X' along the path from I to X. Thus the subdivision tree from EVAL is a subtree of the subdivision tree from GEN, implying the first inequality. (ii) The proof of this inequality follows from Theorem 3.5 (in the next section) and Lemma 3.3. □

3.1. Framework of Stopping Functions. In this section, we provide a generalization of the above technique that promises to be an important tool for bounding the complexity of a large class of subdivision algorithms. Algorithms in this class include EVAL, Plantinga-Vegter's Algorithm and Snyder's Algorithm for isotopic approximation [29, 28, 39]. We now formulate an abstract algorithm called GENERIC which is intended to be a prototype of this class of algorithms in the one dimensional case. It can be generalized easily to higher dimensions.

Let B_0 be a fixed predicate (i.e., Boolean function) on $\square\mathbb{F}$. GENERIC, which depends on B_0, is the following algorithm: Given an interval $I = [a, b]$, GENERIC maintains a partition P of I. Initially, $P = \{I\}$. Then, repeatedly bisect any interval X in P for which $B_0(X)$ is false. The algorithm terminates when all intervals X in P satisfy B_0. If the algorithm terminates, let $P_{GENERIC}(I)$ denotes the final partition.

By a **stopping function** for GENERIC we mean a function $F : \square\mathbb{F} \to \mathbb{R}$ such that, for any interval X, if there exists $b \in X$ such that $w(X) < F(b)$, then $B_0(X)$ holds. For example, suppose $B_0(X)$ is the predicate that holds when $C_0(X)$ or $C_1(X)$ holds, i.e., X is EVAL-terminal. Then Lemma 3.3 shows that the function $G(a)$ in (3.1) is a stopping function.

We have the following bound on $\#P_{GENERIC}(I)$ in terms of any stopping function:

THEOREM 3.5. *If F is a stopping function for GENERIC then*

$$\#P_{GENERIC}(I) \leq \max\left\{1, \int_I \frac{2da}{F(a)}\right\}.$$

If GENERIC does not terminate, the integral is infinite.

Proof. If $\#P_{GENERIC} = 1$, then the bound is immediate. If $\#P_{GENERIC} > 1$, then an examination of the GENERIC algorithm shows that for $X \in P_{GENERIC}$, since GENERIC did not terminate at the parent of X, the following must hold:

$$\forall c \in X, w(X) \geq \frac{1}{2}F(c).$$

In addition, $\int_I \frac{2da}{F(a)} = \sum_{X \in P_{GENERIC}} \int_X \frac{2da}{F(a)}$ and so it suffices to show that for every $X \in P_{GENERIC}$, $\int_X \frac{2da}{F(a)} \geq 1$. Let $d \in X$ be such that $F(d)$ is maximal. Then

$$\int_X \frac{2da}{F(a)} \geq \int_X \frac{2da}{F(d)} = \frac{2}{F(d)}w(X) \geq \frac{2}{F(d)} \cdot \frac{F(d)}{2} = 1.$$

In the case when GENERIC does not terminate, we can still look at the partition P at any moment in time. The above argument shows that $\#P$ is still bounded by the integral $\int_I 2da/F(a)$. Since $\#P$ can be chosen arbitrarily large, this shows the integral to be unbounded. □

The setup of GENERIC and the existence of F is a very common situation among subdivision algorithms, and this type of continuous amortization argument has significant promise to provide bounds for other subdivision algorithms. We now return to the analysis of our example algorithm, EVAL.

3.2. Avoiding Zeros of ff'. The GEN algorithm has the useful property of bounding the number of subdivisions of EVAL, Theorem 3.4. On most inputs, however, GEN does not terminate. In this section, we first isolate the regions where GEN fails to terminate and perform GEN on the regions where it does terminate. This will make the bound on the time complexity finite, while also using the inequality of Theorem 3.4.

By definition, $G(a) \geq 0$ for all a and $G(a) = 0$ iff $f(a) = 0$ or $f'(a) = 0$. Thus, if the integral $\int \frac{2da}{G(a)}$ of Theorem 3.4 is taken over a union of intervals $I' \subseteq I$ that avoid the zeros of f and f', then the integral will be finite. We now construct such an I'.

For each zero $\alpha \in \text{ZERO}(f)$, let $\rho(\alpha)$ be the distance from α to the nearest zero of f different from α. Similarly, if $\beta \in \text{ZERO}(f')$, let $\rho'(\beta)$ be the corresponding function for f'. Since f and f' have no roots in common, we can merge these two ρ functions into one, $\overline{\rho} : \text{ZERO}(ff') \to \mathbb{R}_{>0}$ where $\overline{\rho}(\alpha) = \rho(\alpha)$ when $f(\alpha) = 0$ and $\overline{\rho}(\alpha) = \rho'(\alpha)$ when $f'(\alpha) = 0$.

We now provide another conceptual algorithm GEN$'$, viewed as a two-staged refinement of the GEN algorithm. GEN$'$ will subdivide I until the roots of ff' are sufficiently isolated and then perform the GEN algorithm on the intervals without roots (on these intervals, GEN terminates). Once again, we maintain a subdivision P of I. Initially, $P = \{I\}$. The algorithm operates in two phases:

PHASE 1: Repeatedly bisect each X in P, until each interval in P is GEN$'$-terminal. An interval X is GEN$'$-terminal if one of the following two conditions hold for X:

$$\#(X \cap \text{ZERO}(ff')) = 0$$
$$\#(X \cap \text{ZERO}(ff')) = 1 \text{ and } w(X) < \min\left\{B(\alpha), \frac{\overline{\rho}(\alpha)}{8d(d-1)}\right\}$$

where α is the unique element in $(X \cap \text{ZERO}(ff'))$ and $B(\alpha)$ is a technical bound presented below.

PHASE 2: For each $X \in P$ with $\#(X \cap \text{ZERO}(ff')) = 0$, partition X using the GEN algorithm.

We consider two partitions of I: let P_1 be the partition at the end of Phase 1, and P_2 the partition at the end of Phase 2. An interval $X \in P_1$ is defined to be special if $\#(X \cap \text{ZERO}(ff')) = 1$ and non-special otherwise. Let $P_1' \subseteq P_1$ be the set of non-special intervals of P_1 and $I' = \cup_{X \in P_1'} X$ be the union of all non-special intervals. This algorithm terminates because the technical bound is always positive and when there are no zeros of ff' in X, the contrapositive of Theorem 3.4 implies that the GEN algorithm terminates on these intervals.

LEMMA 3.6. *If $X \in P_1$ is special, then it is EVAL-terminal.*

Proof. For each special interval, there is a unique $\alpha \in X \cap \text{ZERO}(ff')$. The two bounds in the GEN$'$ algorithm, $\frac{\overline{\rho}(\alpha)}{8d(d-1)}$ and $B(\alpha)$, are used to ensure that either

$C_0(X)$ or $C_1(X)$ holds in each special interval. We now present the technical bound $B(\alpha)$ in the GEN$'$ algorithm. Define

$$B(\alpha) = \begin{cases} \infty & \text{if } \alpha \text{ is zero of } f \\ \sqrt{\frac{|f(\alpha)|}{3|f''(\alpha)|}} & \text{if } \alpha \text{ is zero of } f' \end{cases}.$$

The bound $\frac{\overline{\rho}(\alpha)}{8d(d-1)}$ is designed to bound $w(X)$ by $\frac{1}{8\gamma(\alpha)}$ or $\frac{1}{8\gamma'(\alpha)}$. We next appeal to the following result from [38] which is reproduced in Proposition A.2 in the appendix:

PROPOSITION 3.7. *If f is square free, then $\frac{1}{\gamma(\alpha)} > \frac{\rho(\alpha)}{d(d-1)}$.*

First, assume that α is a root of f. In this case, we show the bound $K'_X < \frac{7}{9}\frac{|f'(\alpha)|}{w(X)}$ by replacing each $f^{(i)}(a)$ in the definition of K'_X (2.2) by its Taylor expansion at α and then applying the triangle inequality (see Lemma A.4 in the Appendix). Then $C_1(X)$ holds from Lemma 2.2 which implies that $w(X) < \frac{|f'(\alpha)|}{K'_X}$.

Now, assume that α is a root of f'. In this case, we show that $K_X < 3|f''(\alpha)|w(X)$ using the same technique as above (see Lemma A.5 in the Appendix). Then $C_0(X)$ holds from Lemma 2.2, because the relationship that $B(\alpha)$ gives between $f(\alpha)$ and $f''(\alpha)$ and the bound of K_X imply that $w(X) < \frac{|f(\alpha)|}{K_X}$. □

COROLLARY 3.8. $\#(P_{EVAL}) \overset{(i)}{\leq} \#(P_2) \overset{(ii)}{\leq} \#(P_1) + \int_{I'} \frac{2da}{G(a)}$

Proof. (i) By Theorem 3.4 and Lemma 3.6, we know that for all $X \in P_2$, X is EVAL-terminal. By applying the same argument as in part (i) of Theorem 3.4, we achieve the desired result. (ii) Let the number of special intervals be s_I. Then, we can bound $\#(P_2)$ by:

$$\begin{aligned} \#(P_2) &\leq s_I + \sum_{X \in P'_1} \max\left\{1, \int_X \frac{2da}{G(a)}\right\} \\ &\leq s_I + \sum_{X \in P'_1} 1 + \sum_{X \in P'_1} \int_X \frac{2da}{G(a)} \leq \#(P_1) + 2\int_{I'} \frac{da}{G(a)}. \end{aligned} \tag{3.2}$$

Note that since I' does not include any zeros of ff', this integral is finite. □

The remainder of the paper will bound $\#(P_1)$ and $\int_{I'} \frac{da}{G(a)}$. We show the former is $O(d(\log d + L))$ (§5) and the latter is $O(d^3(\log d + L))$ (§6). This will complete the proof of our main theorem. First, however, we present an evaluation bound (§4) that is critical to these results.

4. An Amortized Evaluation Bound. Our main complexity result is based on two distinct kinds of bounds. The first are the usual Mahler-Davenport bounds that involve root separation bounds (cf. [16]). As in [7], we also need another kind of bound that we call evaluation bounds: They refer to upper and lower bounds on $|f(\alpha)|$ where $f \in \mathbb{C}[X]$ and $\alpha \in \mathbb{C}$. Lower bounds are only possible with the additional assumption that α and the coefficients of f are algebraic numbers. Our bounds are described as *amortized bounds* because they bound a product of $|f(\alpha)|$'s. The evaluation bound here is distinct from the multivariate version used in [7]. The evaluation bound below is also of independent interest.

Let $f = \sum_{i=0}^{d} c_i X^i \in \mathbb{C}[X]$ and $\mathrm{lc}(f) := |c_d|$ be the magnitude of the leading coefficient of f. Define $\mathrm{tc}(f)$ to be the magnitude of the tail coefficient of f, i.e., let t be the smallest index where $c_i \neq 0$, then $\mathrm{tc}(f) = |c_t|$. Let $\mathtt{res}(f, g)$ denote the resultant

of two polynomials f, g. In addition to heights, we use the Mahler measure $M(f)$ of f: If $\alpha_1, \ldots, \alpha_d$ are the complex roots of f, then $M(f) := \mathrm{lc}(f) \prod_{i=1}^{d} \max\{1, |\alpha_i|\}$.

THEOREM 4.1. *Let $\phi(X), \eta(X) \in \mathbb{C}[X]$ be complex polynomials of degrees m and n respectively. Let $\beta_1, \ldots, \beta_n$ be all the zeros of $\eta(X)$.*
(a)

$$\prod_{i=1}^{n} |\phi(\beta_i)| \leq ((m+1)\|\phi\|)^n \left(\frac{M(\eta)}{\mathrm{lc}(\eta)}\right)^m. \tag{4.1a}$$

(b) Let $F, H \in \mathbb{Z}[X]$ be relatively prime such that $F = \phi\widetilde{\phi}$, $H = \eta\widetilde{\eta}$ for some $\widetilde{\phi}, \widetilde{\eta} \in \mathbb{C}[X]$. If the degrees of $\widetilde{\phi}$ and $\widetilde{\eta}$ are $\widetilde{m}$ and $\widetilde{n}$, respectively, then

$$\prod_{i=1}^{n} |\phi(\beta_i)| \geq \frac{1}{\mathrm{lc}(\eta)^m \left((m+1)\|\phi\|\right)^{\widetilde{n}} M(\widetilde{\eta})^m \left((\widetilde{m}+1)\|\widetilde{\phi}\|\right)^{n+\widetilde{n}} M(H)^{\widetilde{m}}}. \tag{4.1b}$$

(c) As an alternative to (b), it is also true that:

$$\prod_{i=1}^{n} |\phi(\beta_i)| \geq \frac{1}{\mathrm{lc}(\eta)^m \left((m+\widetilde{m}+1)\|F\|\right)^{\widetilde{n}} M(\widetilde{\eta})^{m+\widetilde{m}} \left((\widetilde{m}+1)\|\widetilde{\phi}\|\right)^{n} M(\eta)^{\widetilde{m}}}. \tag{4.1c}$$

Proof. (a) Index the β_i's such that for some $n' \in \mathbb{N}$, $|\beta_i| \geq 1$ iff $i > n'$. For $i = 1, \ldots, n'$, $|\phi(\beta_i)| < \|\phi\|(m+1)$ and hence

$$\prod_{i=1}^{n'} |\phi(\beta_i)| \leq (\|\phi\|(m+1))^{n'}. \tag{4.2}$$

For $i = (n'+1), \ldots, n$, $|\phi(\beta_i)| \leq \|\phi\|(m+1)|\beta_i|^m$, and hence

$$\prod_{i=n'+1}^{n} |\phi(\beta_i)| \leq (\|\phi\|(m+1))^{n-n'} \left(\prod_{i=n'+1}^{n} |\beta_i|\right)^m = (\|\phi\|(m+1))^{n-n'} \left(\frac{M(\eta)}{\mathrm{lc}(\eta)}\right)^m \tag{4.3}$$

The product of (4.2) and (4.3) finishes the proof of (a).

(b) Let $\beta_1, \ldots, \beta_n, \beta_{n+1}, \ldots, \beta_{n+\widetilde{n}}$ be the roots of H, then from ([42, p. 167]), $\mathtt{res}(F, H) = \mathrm{lc}(H)^{m+\widetilde{m}} \prod_{i=1}^{n+\widetilde{n}} F(\beta_i)$. Since F and H are integer and relatively prime, the magnitude of their resultant is at least one. Thus,

$$1 \leq |\mathtt{res}(F, H)| = \mathrm{lc}(H)^{m+\widetilde{m}} \cdot \prod_{i=1}^{n} |\phi(\beta_i)| \left(\prod_{i=n+1}^{n+\widetilde{n}} |\phi(\beta_i)| \prod_{i=1}^{n+\widetilde{n}} |\widetilde{\phi}(\beta_i)|\right) \tag{4.4}$$

$$\prod_{i=1}^{n} |\phi(\beta_i)| \geq \frac{1}{\mathrm{lc}(H)^{m+\widetilde{m}} \prod_{i=n+1}^{n+\widetilde{n}} |\phi(\beta_i)| \prod_{i=1}^{n+\widetilde{n}} |\widetilde{\phi}(\beta_i)|}$$

$$\geq \frac{1}{\mathrm{lc}(H)^{m+\widetilde{m}} \left((m+1)\|\phi\|\right)^{\widetilde{n}} \left(\frac{M(\widetilde{\eta})}{\mathrm{lc}(\widetilde{\eta})}\right)^m \left((\widetilde{m}+1)\|\widetilde{\phi}\|\right)^{n+\widetilde{n}} \left(\frac{M(H)}{\mathrm{lc}(H)}\right)^{\widetilde{m}}},$$

where the last inequality follows from part (a). Since $\mathrm{lc}(H) = \mathrm{lc}(\eta)\,\mathrm{lc}(\widetilde{\eta})$, the last expression simplifies to the bound in (4.1b).

(c) From the estimate in (4.4), we alternately proceed as follows:

$$\prod_{i=1}^{n} |\phi(\beta_i)| \geq \frac{1}{\mathrm{lc}(H)^{m+\widetilde{m}} \prod_{i=n+1}^{n+\widetilde{n}} |F(\beta_i)| \prod_{i=1}^{n} |\widetilde{\phi}(\beta_i)|}$$
$$\geq \frac{1}{\mathrm{lc}(H)^{m+\widetilde{m}} \left((m+\widetilde{m}+1)\|F\|\right)^{\widetilde{n}} \left(\frac{M(\widetilde{\eta})}{\mathrm{lc}(\widetilde{\eta})}\right)^{m+\widetilde{m}} \left((\widetilde{m}+1)\|\widetilde{\phi}\|\right)^{n} \left(\frac{M(\eta)}{\mathrm{lc}(\eta)}\right)^{\widetilde{m}}},$$

which simplifies to (4.1c). □

In addition to our evaluation bound, we will need the following lower bound on the product of the roots of a polynomial.

LEMMA 4.2. *If $S \subseteq \{\alpha_{t+1}, \ldots, \alpha_d\}$ is a subset of the non-zero roots of f then*

$$\prod_{\alpha \in S} |\alpha| \geq \frac{\mathrm{tc}(f)}{M(f)}.$$

In particular, if f is an integer polynomial, $\prod_{\alpha\in S} |\alpha| \geq \frac{1}{M(f)}$.
Proof.

$$\prod_{\alpha \in S} |\alpha| \geq \prod_{i=t+1}^{d} \min\{1, |\alpha_i|\} = \prod_{i=t+1}^{d} \frac{|\alpha_i|}{\max\{1, |\alpha_i|\}} = \frac{\mathrm{lc}(f) \prod_{i=t+1}^{d} |\alpha_i|}{M(f)} = \frac{\mathrm{tc}(f)}{M(f)}.$$ □

5. Bounding the Size of P_1. This section bounds the size of P_1 appearing in (3.2). We focus on the at most $2(2d-1)$ special intervals in P_1 (there are $\leq 2d-1$ roots of ff', but each root may lie on the boundary of two special intervals).

Consider the subdivision tree T_1 whose leaves are labeled by P_1. Clearly, $\#(T_1) = 2\#(P_1) - 1$. Call a leaf *special* if it represents a special interval. Let T_2 be the result of pruning all non-special leaves from T_1 (if the tree is only the root, $T_2 = T_1$). Since each non-special leaf of T_1 has a sibling which is either special or an interior node and the root has no sibling, $\#(T_2) \geq \frac{\#(T_1)+1}{2} = \#(P_1)$. The external path length of a tree T, written as $EPL(T)$, is the sum of lengths of paths from the root to each leaf of T ([19, p. 399]). Then, $\#(T_2) \leq EPL(T_2) + 1$. We bound $\#(P_1)$ via the following much stronger result, a bound on $EPL(T_2)$, which is proved in this section:

LEMMA 5.1. $\#(P_1) \leq EPL(T_2) + 1 = O(d(\log d + L))$.

This bound on $EPL(T_2)$ is also needed for the results in §7.

5.1. Bounding the External Path Length of T_2. This section will prove Lemma 5.1. Each leaf s of T_2 is associated with a unique $\alpha_s \in \textsc{Zero}(ff') \cap I$ and an inverval I_s in P_1. Let $S = \{s | \alpha_s \in \textsc{Zero}(f)\}$ and $S' = \{s | \alpha_s \in \textsc{Zero}(f')\}$. Let T_3 (resp. T_3') be the subtree of T_2 comprising all paths from the root of T_2 to a leaf s where $s \in S$ (resp. $s \in S'$). Clearly $EPL(T_2) = EPL(T_3) + EPL(T_3')$. Moreover,

$$EPL(T_3) = \sum_{s \in S} \lg(w(I)/w(I_s)), \qquad EPL(T_3') = \sum_{s \in S'} \lg(w(I)/w(I_s)),$$

where $\lg = \log_2$. Recall our assumption that $w(I) \leq 2^{L+1}$. Hence

$$EPL(T_3) \leq 2d(L+1) - \sum_{s \in S} \lg w(I_s) \qquad EPL(T_3') \leq 2(d-1)(L+1) - \sum_{s \in S'} \lg w(I_s) \tag{5.1}$$

In Phase I of the GEN′ algorithm, there are three conditions that need to hold for interval I_s to be GEN′-terminal. Two of these are the numerical conditions that appear above and the third is the restriction that there is only one root of ff' in I_s. In the case that all three conditions hold on $w(I)$, both GEN′ and EVAL perform no subdivisions and the necessary results are trivial. Thus, we assume that $w(I)$ is sufficiently large; in this case, for each α_s the most restrictive of these three conditions will provide a lower bound on $w(I_s)$. In fact,

$$w(I_s) \geq \frac{1}{2} \min \left\{ \rho_{ff'}(\alpha_s), \frac{\overline{\rho}(\alpha_s)}{8d(d-1)}, B(\alpha_s) \right\}. \tag{5.2}$$

Here $\rho_{ff'}$ is the corresponding ρ for ff' (cf. §3.2).

First, consider the case when $s \in S$. In this case, $B(\alpha_s) = \infty$ and Renegar [32] shows that $\rho_{ff'}(\alpha_s) \geq \rho(\alpha_s)/d$. Therefore, the minimum in (5.2) is $\frac{\overline{\rho}(\alpha_s)}{8d(d-1)}$. Then, by a direct application of a result in [15], the following bound holds:

$$-\lg \prod_{s \in S} w(I_s) \leq -\lg \prod_{s \in S} \frac{\rho(\alpha_s)}{16d(d-1)} = O(d \log d + dL). \tag{5.3}$$

Combining (5.1,5.3), the result $EPL(T_3) = O(d \log d + dL)$ follows.

Next, consider the case where $s \in S'$. Now, we split S' into S'_0, S'_1 and S'_2 depending on which of the above bounds is minimal in (5.2). S'_0 corresponds to $\rho_{ff'}(\alpha_s)$, S'_1 corresponds to $\frac{\overline{\rho}(\alpha_s)}{8d(d-1)} = \frac{\rho'(\alpha_s)}{8d(d-1)}$, and S'_2 corresponds to $B(\alpha_s)$. The desired expression is split into three portions:

$$\prod_{s \in S'} w(I_s) \geq \prod_{s \in S'_0} \frac{1}{2} \rho_{ff'}(\alpha_s) \prod_{s \in S'_1} \frac{\rho'(\alpha_s)}{16d(d-1)} \prod_{s \in S'_2} \frac{1}{2} B(\alpha_s).$$

Applying the same result from [15] as above, we may bound the S'_0- and S'_1-portions as

$$-\lg \prod_{s \in S'_0} \frac{1}{2} \rho_{ff'}(\alpha_s) = O(d \log d + dL) \qquad -\lg \prod_{s \in S'_1} \frac{\rho'(\alpha_s)}{16d(d-1)} = O(d \log d + dL). \tag{5.4}$$

Finally, we bound the S'_2-portion:

$$-\lg \prod_{s \in S'_2} \sqrt{\frac{|f(\alpha_s)|}{3|f''(\alpha_s)|}} = O(d \log d + dL). \tag{5.5}$$

It suffices to show $-\lg \prod_{s \in S'_2} \frac{|f(\alpha_s)|}{|f''(\alpha_s)|} = O(d(\log d + L))$ because the difference from (5.5) is a lower order term (involving $|S'_2| \log 3$ combined with the bound $|S'_2| \leq 2(d-1)$). This in turn reduces to individually bounding the product of the numerators and the product of the denominators.

For both bounds, we let $\eta(X) = \prod_{s \in S'_2} (X - \alpha_s)$ and $\widetilde{\eta}(X) = f'(X)/\eta(X)$. Then, it easily follows that $M(\widetilde{\eta}), M(\eta) \leq M(f') \leq \sqrt{d}\|f'\| \leq d^{3/2} 2^L$ (e.g., [42, p. 117]). To prove the bound on $\prod_{s \in S'_2} |f(\alpha_s)|$, we substitute into Theorem 4.1(b) η and $\widetilde{\eta}$ as above, so that $H(X) = f'(X)$, and $\phi(X) = F(X) = f(X)$ with $\widetilde{\phi}(X) = 1$. Then the evaluation bound (4.1b) gives:

$$-\lg \prod_{s \in S'_2} |f(\alpha_s)| \leq \lg \left(\mathrm{lc}(\eta)^m \cdot ((m+1)\|\phi\|)^{\widetilde{n}} M(\widetilde{\eta})^m \cdot \left((\widetilde{m}+1)\|\widetilde{\phi}\| \right)^{n+\widetilde{n}} M(H)^{\widetilde{m}} \right)$$

$$\leq \lg\left(((d+1)\|f\|)^{\widetilde{n}}(d^{3/2}2^L)^d\right) = O(d(\log d + L)).$$

Similarly, we obtain the bound on $\prod_{\alpha\in S_2'}|f''(\alpha)|$ from the upper bound in (4.1a) by setting $\eta(X)$ as above and $\phi(X) = f''(X)$. The bound (4.1a) gives:

$$\lg\prod_{s\in S_2'}|f''(\alpha_s)| \leq \lg\left(((m+1)\|\phi\|)^n\left(\frac{M(\eta)}{\mathrm{lc}(\eta)}\right)^m\right)$$
$$\leq \lg\left((d\|\phi\|)^d M(\eta)^{d-1}\right) = O(d(\log d + L)).$$

Combining (5.1,5.4,5.5), the result $EPL(T_3') = O(d\log d + dL)$ follows.

6. Bounding the Integral $\int_{I'}\frac{da}{G(a)}$. This section bounds the integral from (3.2), needed for the main result (Theorem 3.1):

THEOREM 6.1. $\int_{I'}\frac{dx}{G(x)} = O(d^3(\log d + L))$.

We bound this integral by a sum of two integrals:

$$\frac{1}{2}\int_{I'}\frac{dx}{G(x)} = \int_{I'}\max\left\{\gamma(x), \frac{d|f'(x)|}{|f(x)|}\right\}dx \leq \int_{I'}\gamma(x)dx + d\int_{I'}\left|\frac{f'(x)}{f(x)}\right|dx$$

We first show that these two integrals are closely related and the same proof and bound can be applied to both integrals. First, $\left|\frac{f'(x)}{f(x)}\right| = \left|\sum_{\alpha\in\mathrm{ZERO}(f)}\frac{1}{x-\alpha}\right| \leq \sum_{\alpha\in\mathrm{ZERO}(f)}\left|\frac{1}{x-\alpha}\right|$. The following Lemma gives a similar result for the other integral.

LEMMA 6.2. $\gamma(x) \leq \sum_{\beta\in\mathrm{ZERO}(f')}\frac{1}{2|x-\beta|}$

The proof uses the fact $f^{(i)}(x)/f'(x) = \sum_{(j_1,\ldots,j_{i-1})}\prod_{\ell=1}^{i-1}\frac{1}{x-\beta_{j_\ell}}$, where j_ℓ's are $i-1$ distinct elements from the set $\{1,\ldots,d-1\}$. The complete proof of this Lemma is given in the Appendix (Lemma A.1).

Thus, both integrands reduce to the same form $\sum_\alpha\frac{1}{|x-\alpha|}$. We may restrict attention to the case $\alpha\in\mathrm{ZERO}(f)$. The case of $\alpha\in\mathrm{ZERO}(f')$ can be obtained from the first case by replacing L by $L+\log d$ in our bounds (since $\|f'\| \leq d\|f\| \leq d2^L = 2^{L+\log d}$). The result we will prove is in the following Lemma:

LEMMA 6.3. $\sum_{\alpha\in\mathrm{ZERO}(f)}\int_{I'}\left|\frac{1}{x-\alpha}\right|dx = O(d^2(\log d + L))$.

Let $\alpha_1,\ldots,\alpha_d$ be the roots of f and write $\alpha_i = r_i + is_i$ where $r_i = \Re(\alpha_i)$ and $s_i = \Im(\alpha_i)$ are the real and imaginary parts of α_i. We re-index the α_i's so that $s_i = 0$ iff $i \leq k$; thus the real roots of f are $r_1,\ldots,r_k$. We split the sum in Lemma 6.3 into the cases where α is real or complex.

LEMMA 6.4 (Real Part). $\sum_{i=1}^{k}\int_{I'}\frac{dx}{|x-r_i|} = O(d^2(\log d + L))$.

Proof. Let $1 \leq i \leq k$ and assume first that r_i does not lie on the boundary of two special intervals. Then let $X_i = [a_i, b_i]$ be the special interval that contains r_i, $n_i = \lg(w(I)/w(X_i))$ be the depth of X_i in the subdivision tree and I_i be $I\setminus X_i$. If r_i is on the boundary of two special intervals, then let X_i be the union of these intervals, n_i be the sum of the depths of the two intervals, and I_i be $I\setminus X_i$. This choice of n_i accounts for both special intervals containing r_i and ensures that $N = \sum n_i$ is $EPL(T_3)$. For each i,

$$\int_{I'}\frac{1}{|x-r_i|}dx \leq \int_{I_i}\frac{1}{|x-r_i|}dx = -\int_a^{a_i}\frac{1}{x-r_i}dx + \int_{b_i}^{b}\frac{1}{x-r_i}dx$$

This evaluates to $\log|a-r_i| + \log|b-r_i| - \log|b_i-r_i| - \log|a_i-r_i|$. The terms $\log|a-r_i|+\log|b-r_i|$ are bounded by $O(L)$ since $w(I) \le 2^{L+1}$. The terms $-\log|b_i-r_i| - \log|a_i-r_i|$ are bounded using the evaluation bound, as follows:

Let $\phi_i(X) = X - r_i$, $\widetilde{\phi}_i(X) = f(X)/\phi_i(X)$, $\eta_i(X) = (X-a_i)(X-b_i)$, and $\widetilde{\eta}_i(X) = 4^{n_i}$. It is fairly easy to see that $\eta_i\widetilde{\eta}_i \in \mathbb{Z}[X]$ since at every subdivision, we perform at most one division by 2 to get the new endpoints. Thus, $\|\phi_i\| \le 2^L$ since $r_i \in I$, $\|\widetilde{\phi}_i\| \le 2^{d-1}(d+1)\|f\| \le 2^{d-1}(d+1)2^L$ by [42], and $M(\eta_i\widetilde{\eta}_i) = \max\{4^{n_i}, 4^{n_i}a_i, 4^{n_i}b_i, 4^{n_i}a_ib_i\} \le 4^{n_i}4^L$. The evaluation bound of Theorem 4.1(b) now gives that

$$-\log|a_i-r_i| - \log|b_i-r_i| = O(n_i + dn_i + d + \log d + L + dL).$$

By summing over all k terms of the summation, noting that each n_j occurs only once and using that $N = EPL(T_3) = O(d(\log d + L))$, the sum becomes $O(N + dN + dk + k\log d + kL + kdL) = O(d^2(\log d + L))$ □

For the complex part, we obtain a better bound:

LEMMA 6.5 (Complex Part). $\displaystyle\sum_{i=k+1}^{d} \int_{I'} \frac{dx}{2|x-\beta_i|} = O(d(d+L))$.

Proof. In the Appendix, we construct two integer polynomials $r(X)$ (Lemma B.1) and $s(X)$ (Lemma B.2) whose zero sets include r_i and s_i, respectively. After this construction, the proof of the Lemma becomes very similar to the one for the real part.

First, consider the case where $a + |s_i| \le r_i \le b - |s_i|$ with $I = [a,b]$. Then

$$\begin{aligned}\int_{I'} \frac{dx}{|x-\alpha_i|} &\le \int_a^b \frac{dx}{|x-\alpha_i|} \le \int_a^b \frac{dx}{\max\{|x-r_i|, |s_i|\}} \\ &= \int_a^{r_i-|s_i|} \frac{dx}{r_i - x} + \int_{r_i-|s_i|}^{r_i+|s_i|} \frac{dx}{|s_i|} + \int_{r_i+|s_i|}^{b} \frac{dx}{x-r_i} \qquad (*) \\ &= \ln\left(\frac{r_i-a}{|s_i|}\right) + 2 + \ln\left(\frac{b-r_i}{|s_i|}\right).\end{aligned}$$

where $(*)$ is valid since $\max\{|x-r_i|, |s_i|\} = |s_i|$ iff $x \in [r_i - |s_i|, r_i + |s_i|]$. Next, suppose that $r_i - |s_i| \le a$. Then a similar bound leads to the above expression with the term $\ln\left(\frac{r_i-a}{|s_i|}\right)$ dropped. Finally, if $r_i + |s_i| \ge b$ then the term $\ln\left(\frac{b-r_i}{|s_i|}\right)$ is dropped. Combining all these cases, we obtain:

LEMMA 6.6.

$$\int_{I'} \frac{dx}{|x-\alpha_i|} \le \ln\max\left\{1, \left(\frac{r_i-a}{|s_i|}\right)\right\} + 2 + \ln\max\left\{1, \left(\frac{b-r_i}{|s_i|}\right)\right\}.$$

Lemma 6.6 implies

$$\begin{aligned}&\int_{I'} \sum_{i=k+1}^{d} \frac{dx}{|x-\beta_i|} \\ &\qquad \le \ln \prod_{i=k+1}^{d} \max\left\{1, \left(\frac{r_i-a}{|s_i|}\right)\right\} + 2(d-k) + \ln \prod_{i=k+1}^{d} \max\left\{1, \left(\frac{b-r_i}{|s_i|}\right)\right\}.\end{aligned}$$

Next, we bound $\ln \prod_{i=k+1}^{d} \max\left\{1, \left(\frac{r_i - a}{|s_i|}\right)\right\}$. Let R_0 be the set of α_i such that $r - |s_i| \geq a$, with this notation, the above expression becomes $\ln \prod_{\alpha_i \in R_0} \left(\frac{r_i - a}{|s_i|}\right)$. As usual, we bound the numerator and denominator separately.

First, we bound $-\ln \prod_{\alpha_i \in R_0} |s_i|$. Since the s_i's are roots of $s(Y)$, it follows from Lemma 4.2 that

$$\prod_{\alpha_i \in R_0} |s_i| \geq \frac{1}{M(s)},$$

Thus by Lemma B.2,

$$-\ln \prod_{\alpha_i \in R_0} |s_i| \leq \ln M(s) \leq \ln\left(((d+1)2^{d+L})^{2d-1}\right) = O(d(d+L))$$

Now, we bound $\ln \prod_{\alpha_i \in R_0} (r_i - a)$ using the evaluation bound. Let $\phi(X) = X - a$ and $\eta(X) = \prod_{\alpha_i \in R_0}(X - r_i)$ be the polynomials in Theorem 4.1(a). It follows that $\|\phi\| \leq 2^L$, and $M(\eta) \leq M(r) \leq ((d+1)2^{d+L})^{2d-1}$ since the r_i's are the roots of $r(X)$. Then since $|R_0| \leq d$, the evaluation bound gives,

$$\ln \prod_{\alpha_i \in R_0} (r_i - a) \leq \ln\left((2 \cdot 2^L)^{|R_0|}((d+1)2^{d+L})^{2d-1}\right) = O(d(d+L))$$

Adding these two bounds together gives the desired bound of $O(d(d+L))$ on this integral.

The bound for the other integral is completely analogous. In this case, instead of R_0, we use R_1, the set of all α_i such that $r_i + |s_i| \leq b$. This gives the same bound for the other integral. Adding these two bounds together completes the proof. □

6.1. Removing the Assumption on f'. In this section, we sketch a proof of our result that does not require f' to be square-free. There are three steps that must be taken to get this result: The first is to adapt the termination conditions in the GEN′ algorithm. When α is a root of order m, we must replace the bound $\frac{\overline{\rho}(\alpha)}{8d(d-1)}$ by $\frac{\rho_{f^{(m)}}(\alpha)}{8d(d-1)}$, the corresponding bounds for the mth derivative of f, and when $m \geq 1$ the bound $B(\alpha)$ should be $\sqrt[m+1]{\frac{|f(\alpha)|}{3 \cdot 2^{3m-3}|f^{(m+1)}(\alpha)|}}$. By substituting these bounds into the lemmas in the Appendix, we derive analogous results to the results appearing there.

The next change appears in the bound $EPL(T_3')$. To bound the size of this tree, we break it up into pieces according to the multiplicity of the roots. In other words, $EPL(T_3') = \sum_m EPL(T_{3,m}')$ where $T_{3,m}'$ corresponds to the tree whose leaves are the special intervals for roots of order m. Then the same proofs as in the text, using the appropriate derivative of f, give that $EPL(T_{3,m}') = O(d \log d + dL)$.

The third change occurs in the calculation of the real part of the integral of $\int \gamma(x)dx$. This is the only integral affected because the computation of the other integral focuses on the roots of f. In particular, we apply Lemma 6.4 to each of the trees $T_{3,m}'$ to get that the sum of the integrals for the real roots of order m is $O(N_m + dN_m + dk_m + k_m \log d + k_m L + k_m dL)$ where $N_m = EPL(T_{3,m}')$ and k_m is the number of roots of order m. Summing over all possible m gives a bound on the real roots as $O(d^3(\log d + L))$.

7. Conclusion. The analysis of adaptive subdivision algorithms is virgin territory for complexity theory. We have introduced some novel techniques for such analysis: a form of continuous amortization (integral bounds) and algebraic amortization techniques. In this paper we illustrated the use of these techniques by analyzing the EVAL algorithm. We pose the problem to extend this continuous amortization technique to other similar subdivision algorithms for which no suitable complexity bounds are known.

Appendix A. Bounds Using Gamma.

In this of this appendix, we provide several omitted proofs. If a theorem or lemma is restated from the text, it will reference the original numbering (but may appear in a different order below).

A.1. Bounding Gamma. LEMMA A.1 (Lemma 6.2). *Let $\beta_1, \ldots, \beta_{d-1}$ be all the zeros of f', then*

$$\gamma(x) \leq \sum_{j=1}^{d-1} \frac{1}{2|x-\beta_j|}$$

Proof. We have

$$\frac{f^{(i)}(x)}{f'(x)} = \sum\nolimits'_{(j_1,\ldots,j_{i-1})} \prod_{\ell=1}^{i-1} \frac{1}{x-\beta_{j_\ell}}$$

where the summation ranges over all ordered $(i-1)$-tuples of distinct elements taken from $\{1,\ldots,d-1\}$, The prime in the summation symbol, $\sum'$, indicates the that the j_k's are distinct. When we omit the prime in the summation, it means that the tuples could have duplicated components. Thus

$$\begin{aligned}
\left|\frac{f^{(i)}(x)}{f'(x)}\right|^{1/(i-1)} &= \left|\sum_{(j_1,\ldots,j_{i-1})}' \prod_{\ell=1}^{i-1} \frac{1}{x-\beta_{j_\ell}}\right|^{1/(i-1)} \leq \left(\sum_{(j_1,\ldots,j_{i-1})}' \prod_{\ell=1}^{i-1} \frac{1}{|x-\beta_{j_\ell}|}\right)^{1/(i-1)} \\
&\leq \left(\sum_{(j_1,\ldots,j_{i-1})} \prod_{\ell=1}^{i-1} \frac{1}{|x-\beta_{j_\ell}|}\right)^{1/(i-1)} \qquad \text{(unprimed summation)} \\
&= \left(\left(\sum_{j=1}^{d-1} \frac{1}{|x-\beta_j|}\right)^{i-1}\right)^{1/(i-1)} = \sum_{j=1}^{d-1} \frac{1}{|x-\beta_j|}.
\end{aligned}$$

For $i \geq 2$, we have $i! \geq 2^{i-1}$, and hence

$$\left|\frac{f^{(i)}(x)}{i!f'(x)}\right|^{1/(i-1)} \leq \frac{1}{2}\left|\frac{f^{(i)}(x)}{f'(x)}\right|^{1/(i-1)} \leq \frac{1}{2}\sum_{j=1}^{d-1} \frac{1}{|x-\beta_j|}. \qquad \square$$

PROPOSITION A.2 (Proposition 3.7,[38]). *If f is square free, then $\frac{1}{\gamma(\alpha)} > \frac{\rho(\alpha)}{d(d-1)}$.*

Proof. We begin just as above. Let $\beta_1, \ldots, \beta_{d-1}$ be all the zeros of f', then we have

$$\frac{f^{(i)}(x)}{f'(x)} = \sum\nolimits'_{(j_1,\ldots,j_{i-1})} \prod_{\ell=1}^{i-1} \frac{1}{x-\beta_{j_\ell}} \tag{A.1}$$

where the summation ranges over all ordered $(i-1)$-tuples of distinct elements taken from $\{1,\ldots,d-1\}$, The prime in the summation symbol, $\sum'$, indicates the that the j_k's are distinct. Now, let α be a root of f, therefore $|\beta_j-\alpha| \geq \rho_{ff'}(\alpha)$ for all j, where $\rho_{ff'}(\alpha)$ is the distance from α to the nearest distinct root of ff'. In fact, $\rho_{ff'}(\alpha)$ is the distance from α to the nearest root of f'. By taking the absolute value of both sides of (A.1), we find that

$$\left|\frac{f^{(i)}(\alpha)}{f'(\alpha)}\right| \leq \frac{(d-1)!}{(d-i)!}\frac{1}{\rho_{ff'}(\alpha)^{i-1}}$$

Then, introducing a $1/i!$ to both sides, we get the inequality

$$\left|\frac{f^{(i)}(\alpha)}{i!f'(\alpha)}\right| = \frac{1}{i}\binom{d-1}{i-1}\frac{1}{\rho_{ff'}(\alpha)^{i-1}} < \left(\frac{d-1}{\rho_{ff'}(\alpha)}\right)^{i-1}$$

Therefore,

$$\gamma_f(\alpha) := \max_{i\geq 2}\left(\frac{|f^{(i)}(x)|}{i!|f'(x)|}\right)^{1/(i-1)} < \frac{d-1}{\rho_{ff'}(\alpha)}$$

Finally, Renegar [32] has shown that $\rho_{ff'}(\alpha) \geq \rho(\alpha)/d$. Substituting this result into the above expression gives

$$\gamma_f(\alpha) < \frac{d(d-1)}{\rho(\alpha)} \qquad \square$$

A.2. Bounding K_X Using Gamma. When using the GEN algorithm, intervals are subdivided until there is some point $a \in J$ such that $w(J) < G(a)$. Often, however, it is necessary to use the information based at a to conclude facts about another point of J. The following three lemmas provide this type of result.

LEMMA A.3 (Lemma 3.2). *Let $b \in J$ such that $w(J) \leq \frac{1}{2\gamma(b)}$. Then $K_J \leq 2d|f'(b)|$.*

Proof.

$$\begin{aligned}
K_J &= \max_{a\in J}\sum_{i=1}^{d}\frac{|f^{(i)}(a)|}{i!}w(J)^{i-1} = \max_{a\in J}\sum_{i=1}^{d}\left|\sum_{j=i}^{d}\frac{f^{(j)}(b)(b-a)^{j-i}}{i!(j-i)!}\right|w(J)^{i-1} \\
&\leq \sum_{i=1}^{d}\sum_{j=i}^{d}\frac{|f^{(j)}(b)|}{i!(j-i)!}w(J)^{j-1} \qquad (|a-b|\leq w(J)) \\
&= \sum_{j=1}^{d}\frac{|f^{(j)}(b)|}{j!}w(J)^{j-1}\sum_{i=1}^{j}\binom{j}{i} \leq \sum_{j=1}^{d}\frac{|f^{(j)}(b)|}{j!}w(J)^{j-1}2^j \\
&\leq \sum_{j=1}^{d}\frac{|f^{(j)}(b)|}{j!}\frac{2^j}{2^{j-1}\gamma(b)^{j-1}} \qquad w(J)\leq\frac{1}{2\gamma(b)} \\
&\leq 2|f'(b)| + 2\sum_{j=2}^{d}\frac{|f^{(j)}(b)|}{j!}\frac{j!|f'(b)|}{|f^{(j)}(b)|} \qquad \gamma(b)\geq\left(\frac{|f^{(j)}(b)|}{j!|f'(b)|}\right)^{\frac{1}{j-1}} \\
&= 2d|f'(b)| \qquad \square
\end{aligned}$$

The conclusion of the next two lemmas were used in the proof of Lemma 3.6:

LEMMA A.4. *Let J be a special interval containing α with $\alpha \in \text{ZERO}(f)$ and $w(J) < \frac{\rho(\alpha)}{8d(d-1)}$. Then $w(J) < \frac{|f'(\alpha)|}{K'_J}$.*

Proof. From Proposition 3.7, we know that the condition on $w(J)$ implies that $w(J) < \frac{1}{8\gamma(\alpha)}$. Now, by computing an upper bound on K'_J, we show the desired result.

$$\begin{aligned}
K'_J &= \max_{a\in J} \sum_{i=1}^{d-1} \frac{|(f')^{(i)}(a)|}{i!} w(J)^{i-1} = \max_{a\in J} \sum_{i=2}^{d} \frac{|f^{(i)}(a)|}{(i-1)!} w(J)^{i-2} \\
&= \max_{a\in J} \sum_{i=2}^{d} \left| \sum_{j=i}^{d} \frac{f^{(j)}(\alpha)(a-\alpha)^{j-i}}{(j-i)!(i-1)!} \right| w(J)^{i-2} \\
&\le \sum_{i=2}^{d} \sum_{j=i}^{d} \frac{|f^{(j)}(\alpha)|}{(j-i)!(i-1)!} w(J)^{j-2} && |a-\alpha| \le w(J) \\
&\le \sum_{i=2}^{d} \sum_{j=i}^{d} \frac{|f^{(j)}(\alpha)|}{(j-1)!} \binom{j-1}{i-1} w(J)^{j-2} = \sum_{j=2}^{d} \frac{|f^{(j)}(\alpha)|}{(j-1)!} w(J)^{j-2} \sum_{i=2}^{j} \binom{j-1}{i-1} \\
&\le \sum_{j=2}^{d} \frac{|f^{(j)}(\alpha)|}{(j-1)!} w(J)^{j-2} 2^{j-1} \\
&\le \frac{1}{w(J)} \sum_{j=2}^{d} \frac{|f^{(j)}(\alpha)|}{(j-1)!} \frac{2^{j-1}}{8^{j-1}\gamma(\alpha)^{j-1}} && w(J) \le \frac{1}{8\gamma(\alpha)} \\
&\le \frac{1}{w(J)} \sum_{j=2}^{d} \frac{|f^{(j)}(\alpha)|}{(j-1)!} \frac{j!|f'(\alpha)|}{|f^{(j)}(\alpha)|} 2^{-2j+2} && \gamma(\alpha) \ge \left(\frac{|f^{(j)}(\alpha)|}{j!|f'(\alpha)|} \right)^{\frac{1}{j-1}} \\
&= \frac{|f'(\alpha)|}{w(J)} \sum_{j=2}^{d} j 2^{-2j+2} < \frac{7}{9} \frac{|f'(\alpha)|}{w(J)}
\end{aligned}$$

Then, by rearranging $w(J)$ and K'_J, we find that $w(J) < \frac{7}{9}\frac{|f'(\alpha)|}{K'_J} < \frac{|f'(\alpha)|}{K'_J}$, as desired. ∎

LEMMA A.5. *Let J be a special interval containing α with $\alpha \in \text{ZERO}(f')$ and $w(J) < \min\left\{ \frac{\rho'(\alpha)}{8d(d-1)}, \sqrt{\frac{|f(\alpha)|}{3|f''(\alpha)|}} \right\}$. Then $w(J) < \frac{|f(\alpha)|}{K_J}$.*

Proof. From Proposition 3.7 we know that the condition on $w(J)$ implies that $w(J) < \frac{1}{8\gamma'(\alpha)}$. By computing an upper bound on K_J, we can show the desired result.

$$\begin{aligned}
K_J &= \max_{a\in J} \sum_{i=1}^{d} \frac{|f^{(i)}(a)|}{i!} w(J)^{i-1} = \max_{a\in J} \sum_{i=1}^{d} \left| \sum_{j=i}^{d} \frac{f^{(j)}(\alpha)(a-\alpha)^{j-i}}{(j-i)!i!} \right| w(J)^{i-1} \\
&\le \sum_{i=1}^{d} \sum_{j=i}^{d} \frac{|f^{(j)}(\alpha)|}{(j-i)!i!} w(J)^{i-1} && |a-\alpha| \le w(J) \\
&= \sum_{j=1}^{d} \frac{|f^{(j)}(\alpha)|}{j!} w(J)^{j-1} \sum_{i=1}^{j} \binom{j}{i} \le \sum_{j=1}^{d} \frac{|f^{(j)}(\alpha)|}{j!} w(J)^{j-1} 2^{j}
\end{aligned}$$

$$= \sum_{j=2}^{d} \frac{|f^{(j)}(\alpha)|}{j!} w(J)^{j-1} 2^j \qquad \alpha \in \text{ZERO}(f')$$

$$= w(J) \sum_{j=2}^{d} \frac{|f^{(j)}(\alpha)|}{j!} w(J)^{j-2} 2^j \leq w(J) \sum_{j=2}^{d} \frac{|f^{(j)}(\alpha)|}{j!} \frac{2^j}{8^{j-2}\gamma'(\alpha)^{j-2}}$$

$$\leq 2w(J)|f''(\alpha)| + w(J) \sum_{j=3}^{d} \frac{|f^{(j)}(\alpha)|}{j!} \frac{(j-1)!|f''(\alpha)|}{|f^{(j)}(\alpha)|} 2^{-2j+6} \qquad (*)$$

$$= |f''(\alpha)|w(J) \sum_{j=2}^{d} \frac{2^{-2j+6}}{j} < 64|f''(\alpha)|w(J) \left(\ln\left(\frac{4}{3}\right) - \frac{1}{4} \right) < 3|f''(\alpha)|w(J)$$

Where $*$ is valid because $\gamma(\alpha) \geq \left(\frac{|(f')^{(j-1)}(\alpha)|}{(j-1)!|(f')'(\alpha)|} \right)^{\frac{1}{j-2}}$. Therefore, it follows that $\frac{|f(\alpha)|}{K_J} > \frac{|f(\alpha)|}{3|f''(\alpha)|w(J)} \left(= \frac{B(\alpha)^2}{w(J)} \right) \geq \frac{(w(J))^2}{w(J)} = w(J)$, completing the result. □

Appendix B. On the Real and Imaginary Part of Zeros.

Let $f \in \mathbb{Z}[X]$ be a real polynomial of degree $d \geq 1$. Suppose its complex zeros are $\alpha_1, \ldots, \alpha_d$ and let $r_i = \Re(\alpha_i)$ and $s_i = \Im(\alpha_i)$ for each i. We re-index the α_i's so that $s_i = 0$ iff $i \leq k$ and so the nonreal roots of f are given by $\alpha_{k+1}, \ldots, \alpha_d$. Our goal is to construct two integer polynomials $r(X), s(X)$ whose roots contain the r_i's and the s_i's, respectively. We also want to bound the Mahler measures of $r(X)$ and $s(X)$.

To analyze the real and complex parts of the roots independently, we consider $f(X + \mathbf{i}Y)$, regarded as a polynomial in $\mathbb{Z}[X, \mathbf{i}Y]$. We also rewrite $f(X + \mathbf{i}Y)$ as $p(X, Y) + \mathbf{i}q(X, Y)$, with $p(X, Y), q(X, Y) \in \mathbb{Z}[X, Y]$. Clearly, $p(X, Y)$ (resp., $\mathbf{i}q(X, Y)$) is the sum of the monomials of $f(X + \mathbf{i}Y) \in \mathbb{Z}[X, \mathbf{i}Y]$ whose degree in Y is even (resp., odd).

B.1. Real Part. We first construct a polynomial $r(X)$ whose roots include all the r_i's (cf. [42, p. 202]). If d is even, then $\deg_Y(p(X, Y)) = d$ and $\deg_Y(q(X, Y)) = d-1$, and if d is odd, then $\deg_Y(p(X, Y)) = d-1$ and $\deg_Y(q(X, Y)) = d$. It easily follows that the r_i's, for all i, are real roots of the resultant $r(X) := \mathtt{res}_Y(p(X, Y), q(X, Y))$. In the following we use $\mathtt{lead}(f)$ which is signed version of $\mathrm{lc}(f)$, i.e., $\mathtt{lead}(f)$ is defined in the same was as $\mathrm{lc}(f)$ (see §4), but the absolute value is dropped.

LEMMA B.1. *$r(X) = \mathtt{res}_Y(p(X, Y), q(X, Y))$ is not the zero polynomial and has Mahler measure $M(r) \leq ((d+1)2^{d+L})^{2d-1}$.*

Proof. We first show that $r(X)$ is not the zero polynomial. Let $D(X, Y) := \gcd_Y(p(X, Y), q(X, Y))$ where p, q are viewed as polynomials in Y with coefficients in $\mathbb{Z}[X]$. By [12], $\mathtt{res}_Y(p(X, Y), q(X, Y)) = 0$ if and only if $D(X, Y)$ has positive degree in Y. In addition, $D(X, Y) \in \mathbb{Q}[X, Y]$ since both $p(X, Y)$ and $q(X, Y) \in \mathbb{Q}[X, Y]$. On the other hand, since $D(X, Y)$ divides both $p(X, Y)$ and $q(X, Y)$, we conclude that $D(X, Y)$ divides $f(X + iY)$. Since $f(Z) = \mathtt{lead}(f) \prod(Z - \alpha_i) = \mathtt{lead}(f) \prod(X + \mathbf{i}Y - \alpha_i)$ and all these factors are irreducible polynomials in $\mathbb{C}[Z]$ or $\mathbb{C}[X, Y]$, it follows that $D(X, Y) = \prod_j (X + \mathbf{i}Y - \alpha_{i_j}) = \prod_j (Z - \alpha_{i_j}) = D(Z)$, for some subset $\{\alpha_{i_j} : j \in J\}$ of the roots of f. $D(Z)$ is entire because it is a polynomial in Z. For any $Z_0 = X_0 + \mathbf{i}Y_0 \in \mathbb{C}$, $D(Z_0) = D(X_0, Y_0)$; since $X_0, Y_0 \in \mathbb{R}$ and $D(X, Y) \in \mathbb{Q}[X, Y]$, it follows that $D(X_0, Y_0) \in \mathbb{R}$. Therefore, $D(Z)$ is a real valued entire fucntion. By a standard result (e.g., [1]) we conclude that D is a constant function. This implies $D(X, Y)$ does not have positive degree in Y. It follows $r(X)$ is not the zero polynomial.

To bound the measure, we first compute the coefficient of Y^k in $f(X + \mathbf{i}Y)$. All monomials of $f(X + \mathbf{i}Y)$ come from products of the form $a_n(X + \mathbf{i}Y)^n$, where $|a_n| < 2^L$. The coefficient of Y^k here is 0 if $n < k$ and $a_n\binom{n}{k}(\mathbf{i})^k X^{n-k}$ otherwise. The X-degree is maximized when $n = d$ and therefore, the X-degree of the coefficient of Y^k is exactly $d - k$ since $a_d \neq 0$.

Next, we bound the Mahler measure of $r(X)$ by $\|r(X)\|_2$ ([42]), and then use the Goldstein-Graham bound ([42, p. 173]) to complete the result. Recall that the 1- or 2-norm is the norm applied to the coefficients of the polynomial. To use this bound, we first need to compute a bound on the 1-norm of the coefficients of Y^k. The coefficient of Y^k is a sum of terms of the form $a_n\binom{n}{k}(\mathbf{i})^k X^{n-k}$. The absolute value of $a_n\binom{n}{k}(\mathbf{i})^k$ is bounded by $\binom{d}{k}2^L$ since $\|f\| < 2^L$. The number of terms in the coefficient of Y^k is $d - k + 1$, which is bounded by $d + 1$. Therefore, the 1-norm of the coefficient of Y^k is bounded by $(d+1)\binom{d}{k}2^L$. Let $\overline{p}$ be the 2-norm of the vector of the 1-norms of the coefficients of Y^k corresponding to k even. $\overline{p}$ is bounded by $\sum_{j=0}^{\lfloor d/2 \rfloor}(d+1)\binom{d}{j}2^L \leq (d+1)2^{d+L}$. Similarly, let $\overline{q}$ be the 2-norm of the vector of the 1-norms of the coefficients of Y^k corresponding to k odd. Then the same bound holds for $\overline{q}$, i.e., $\overline{q} \leq (d+1)2^{d+L}$. Now, by examining the Goldstein-Graham bound, we find that $\|r(X)\|_2 \leq \overline{p}^{d-1}\overline{q}^d$ if d is even, and $\|r(X)\|_2 \leq \overline{p}^d\overline{q}^{d-1}$ if d is odd. Both of these simplify to the desired result. □

In fact, the degree of $r(X)$ is exactly d^2, which is found by comparing the leading coefficient of $r(X)$ to the leading coefficient of $f(Z) = Z^d$. As this result is not essential for our main result, we omit its longer and tedious proof.

B.2. Imaginary Part. A similar procedure can be used to construct a polynomial $s(Y)$ whose roots include all the s_i's. Some details are slightly different, and we derive them here. Note that $\deg_X(p(X,Y)) = d$ and $\deg_X(q(X,Y)) = d - 1$, independent of the parity of d. It easily follows that the s_i's for all i are real roots of the resultant $s(Y) = \mathtt{res}_X(p(X,Y), q(X,Y))$.

LEMMA B.2. *$s(Y) = \mathtt{res}_X(p(X,Y), q(X,Y))$ is not the zero polynomial and has Mahler measure $M(s) \leq ((d+1)2^{d+L})^{2d-1}$.*

Proof. The proof is very similar to the proof in the real case. We first show that $s(X)$ is not the zero polynomial. The same proof as in the real case applies since the roles of X and Y are symmeteric, and hence $s(X)$ is not the zero polynomial.

To bound the measure, we compute the coefficient of X^k in $f(X + \mathbf{i}Y)$. Using similar arguments to the real case, we get that the coefficient of X^k in $a_n(X + \mathbf{i}Y)^n$ is 0 if $n < k$ and $a_n\binom{n}{k}(\mathbf{i})^{n-k}Y^{n-k}$. The Y-degree of this is maximized when $n = d$. However, if we look for monomials with real coefficients, then if $d - k$ is even, the Y-degree is maximized when $n = d$, and if $d - k$ is odd, the Y-degree is maximized with the largest $n \leq d - 1$ such that $a_n \neq 0$ and $n - k$ is even. On the other hand, if we look for monomials with imaginary coefficients, then if $d - k$ is even, the Y-degree is maximized with the largest $n \leq d - 1$ such that $a_n \neq 0$ and $n - k$ is odd, and if $d - k$ is odd, the Y-degree is maximized when $n = d$.

The bound on the Mahler measure of s is very similar to the real case. We bound the Mahler measure of $s(X)$ by $\|s(X)\|_2$ ([42]), and then use the Goldstein-Graham bound ([42, p. 173]) to complete the result. The proof proceeds analogously with the real case and provides the bound $\|s(X)\|_2 \leq \overline{p}^{d-1}\overline{q}^d$, which simplifies to the desired result. $\overline{p}$ and $\overline{q}$ are defined similarly to the real case and the same upper bound applies. □

As in the real case, one can actually show that $s(Y)$ has degree d^2.

REFERENCES

[1] L. V. Ahlfors. *Complex Analysis*. McGraw-Hill, New York, 1953.

[2] T. Asano, D. Kirkpatrick, and C. Yap. Pseudo approximation algorithms, with applications to optimal motion planning. *DCG*, 31(1):139–171, 2004.

[3] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer-Verlag, New York, 1998.

[4] K. H. Borgwardt. Probabilistic analysis of the simplex method. In J. Lagarias and M. Todds, editors, *Mathematical Developments Arising from Linear Programmng*, volume 114, pages 21–34. AMS, 1990. This volume also has papers by Karmarkar, Megiddo.

[5] M. Burr, S. Choi, B. Galehouse, and C. Yap. Complete subdivision algorithms, II: Isotopic meshing of singular algebraic curves. In *Proc. Int'l Symp. Symbolic and Algebraic Computation (ISSAC'08)*, pages 87–94, 2008. Hagenberg, Austria. Jul 20-23, 2008.

[6] M. Burr, V. Sharma, and C. Yap. Evaluation-based root isolation, Feb. 2009. In preparation.

[7] J.-S. Cheng, X.-S. Gao, and C.-K. Yap. Complete numerical isolation of real zeros in zero-dimensional triangular systems. *J. of Symbolic Computation*, 44(7):768–785, 2009. Special Issue of JSC based on ISSAC 2007. Available online at JSC.

[8] 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.

[9] G. E. Collins, J. R. Johnson, and W. Krandick. Interval arithmetic in cylindrical algebraic decomposition. *J. of Symbolic Computation*, 34:145–157, 2002.

[10] M. Corazza, F. Corazza, L. C. Filho, and C. Daviva. A subdivision algorithm for phase equilibrium calculations at high pressures. *Brazilian J. Chemical Engineering*, 24(4):611–622, 2007.

[11] 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.

[12] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer-Verlag, New York, 1992.

[13] 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.

[14] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 1997.

[15] 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.

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

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

[18] R. B. Kearfott. Abstract generalized bisection with a cost bound. *Math.Comp.*, 49:187–202, 1987.

[19] D. E. Knuth. *The Art of Computer Programming: Fundamental Algorithms*, volume 1. Addison-Wesley, Boston, 2nd edition edition, 1975.

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

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

[22] 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.

[23] R. Martin, H. Shou, I. Voiculescu, A. Bowyer, and G. Wang. Comparison of interval methods for plotting algebraic curves. *Computer Aided Geometric Design*, 19(7):553–587, 2002.

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

pages 68–74, 1990.
[25] R. E. Moore. *Interval Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1966.
[26] B. Mourrain, F. Rouillier, and M.-F. Roy. The Bernstein basis and real root isolation. In J. E. Goodman, J. Pach, and E. Welzl, editors, *Combinatorial and Computational Geometry*, number 52 in MSRI Publications, pages 459–478. Cambridge University Press, 2005.
[27] V. Y. Pan. Solving a polynomial equation: some history and recent progress. *SIAM Review*, 39(2):187–220, 1997.
[28] S. Plantinga. *Certified Algorithms for Implicit Surfaces*. Ph.D. thesis, Groningen University, Institute for Mathematics and Computing Science, Groningen, Netherlands, Dec. 2006.
[29] 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.
[30] H. Ratschek and J. Rokne. *Computer Methods for the Range of Functions*. Horwood Publishing Limited, Chichester, West Sussex, UK, 1984.
[31] D. Reischert. Asymptotically fast computation of subresultants. In *ISSAC 97*, pages 233–240, 1997. Maui, Hawaii.
[32] J. Renegar. On the worst-case arithmetic complexity of approximating zeros of polynomials. *Journal of Complexity*, 3:90–113, 1987.
[33] F. Rouillier and P. Zimmermann. Efficient isolation of [a] polynomial's real roots. *J. Computational and Applied Mathematics*, 162:33–50, 2004.
[34] J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. of Algorithms*, 18(3):548–585, 1995.
[35] A. Schönhage. The fundamental theorem of algebra in terms of computational complexity, 1982. Manuscript, Department of Mathematics, University of Tübingen. Updated 2004.
[36] J. Sellen, J. Choi, and C. Yap. Precision-sensitive Euclidean shortest path in 3-Space. *SIAM J. Computing*, 29(5):1577–1595, 2000. Also: 11th ACM Symp. on Comp. Geom., (1995)350–359.
[37] V. Sharma, Z. Du, and C. Yap. Robust approximate zeros. In G. S. Brodal and S. Leonardi, editors, *Proc. 13th European Symp. on Algorithms (ESA)*, volume 3669 of *Lecture Notes in Computer Science*, pages 874–887. Springer-Verlag, Apr. 2005. Palma de Mallorca, Spain, Oct 3-6, 2005.
[38] V. Sharma and C. Yap. Complexity of strong root isolation, 2007. In preparation.
[39] J. M. Snyder. Interval analysis for computer graphics. *SIGGRAPH Comput.Graphics*, 26(2):121–130, 1992.
[40] D. A. Spielman and S.-H. Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. of the ACM*, 51(3):385–463, 2004.
[41] V. Stahl. *Interval Methods for Bounding the Range of Polynomials and Solving Systems of Nonlinear Equations*. Ph.D. thesis, Johannes Kepler University, Linz, 1995.
[42] C. K. Yap. *Fundamental Problems of Algorithmic Algebra*. Oxford University Press, 2000.