Analytic Root Clustering: A Complete Algorithm using Soft Zero Tests*

Chee Yap**, Michael Sagraloff, and Vikram Sharma

[1] Courant Institute of Mathematical Sciences
New York University
New York, NY 10012, USA
yap@cs.nyu.edu
[2] Max-Planck Institute of Computer Science
Saarbrücken, Saarland, Germany
msagralo@mpi-inf.mpg.de
[3] Institute of Mathematical Sciences
Chennai, India
vikram@imsc.res.in

Abstract. A challenge to current theories of computing in the continua is the proper treatment of the **zero test**. Such tests are critical for extracting geometric information. Zero tests are expensive and may be uncomputable. So we seek geometric algorithms based on a weak form of such tests, called **soft zero tests**. Typically, algorithms with such tests can only determine the geometry for "nice" (e.g., non-degenerate, non-singular, smooth, Morse, etc) inputs. Algorithms that avoid such niceness assumptions are said to be **complete**. Can we design complete algorithms with soft zero tests? We address the basic problem of determining the geometry of the roots of a complex analytic function. This is formalized as the **root clustering problem**, and we provide a complete (δ, ϵ)**-exact** algorithm based on soft zero tests.

1 Introduction: Soft Zero Tests

Almost a century ago, mathematicians and logicians began to develop a theory of computation. It led to the highly successful theory of recursive functions and its higher analogues [15]. Subsequently, in the hands of computer scientists, the lower analogues (at the subrecursive levels) were developed. This is Complexity Theory as we know it today [8]. The lower analogues turn out to have a richer and harder theory: thus, the P versus NP is easily resolved at the higher level. The main line of this development, especially in computer science, is largely about computing over a discrete universe like strings or natural numbers. The issues of computing in the continua, or its surrogate, the real line ($\mathbb{R}$) is side-stepped by this development. One approach to the continua is to use abstract

* This paper was presented at an invited Special Session on "Computational Complexity in the Continuous World" at **Computability in Europe** (CiE2013), July 1-5, Milan, Italy.
** This work is supported by an National Science Foundation Grant #CCF-0917093.

computational models that have operations on continua data, given as primitives. Examples include Theoretical Computer Science under the Real RAM Model, the Algebraic School [2], and also Information-Based Complexity (IBC) [18]. But a more foundational approach is to consider computational models which (at least in principle) truly operate at the bit level, like Turing machines. The analytic school of real computation [19,10] is the main representative.

It is apparent that computing over a discrete universe is vastly different than computing over the continua. For instance, the fall-back method of "brute force search" in discrete computation is not an option in the continua. Indeed, brute force searches in the continua typically do not halt. From the perspective of Exact Geometric Computation (EGC), current models of continua computing are lacking [22]. The touchstone is the **Zero Problem**, deciding if a real constant is zero. Current models lead to one of two conclusions about the Zero Problem: (A) the problem is undecidable, or (B) the problem is trivial by fiat (zero test is a primitive in the model). Our approach in [22] allows the zero problems to have a range of complexity, consistent with what is observed in practice.

The EGC viewpoint is motivated by practical and correct implementation of continua algorithms. It is the most successful approach in computational geometry, and implemented in libraries such as `LEDA`, `CGAL`, `Core Library` (see references in [7,22]). Nevertheless, there are barriers when we address non-linear and/or non-algebraic problems. We are therefore motivated to study weaker notions of exactness in geometric computation. In particular, we explore models of real computation in which only *the non-zero sign* of real constants can be decided: given a numerical constant x (represented implicitly in some way) we can only to ask whether $x > 0$ or $x < 0$, but not $x = 0$. See [21, Section VI]. In terms of programming constructs, we allow guarded statements of the form "if $x > 0$ then do ..." (but there is no immediate else-clause because the failure of "$x > 0$" does not allow us to conclude that $x \leq 0$). The test $x > 0$ is implemented by iterative approximation of x, a paradigm is nicely captured in the subdivision framework (e.g., [21]). We call these **soft zero tests** (see Section 6), and they embody the well-known dictum in numerical computation: *never compare a quantity to zero.* A realistic theoretical model for such computation is the numerical pointer machine [22] based on Schönhage's pointer machines.

What kind of geometric information can we compute using "soft algorithms", i.e., with soft zero tests? Clearly, in practice most computational scientists use such algorithms. But we are interested **exact algorithms** that guarantee the correct geometry. A striking example is Plantinga and Vegter's soft algorithm [13] for computing isotopic approximations of curves and surfaces. We recently [21] gave a soft algorithm for the Voronoi diagram of polygonal objects. Both these examples had to assume "nice" inputs: the curves and surfaces must be non-singular [13], the Voronoi diagram must be non-degenerate [21]. Algorithms that avoid niceness assumptions on inputs are said to be **complete**. So the main challenge of this paper is design soft algorithms that are also complete. One way to get obtain soft-and-complete algorithms is to exploit algebraic zero bounds. For analytic problems, such bounds are not readily available and we must weaken the exact geometry criteria using the backwards error idea from numerical analysis. Informally, we propose to compute "an ϵ-correct output for some δ-perturbation of the input". The precise usage of these δ, ϵ parameters will depend on the problem, but generally they lead to the concept of **(δ, ϵ)-exactness**. In summary, our specific goal is to construct (δ, ϵ)-exact algorithms that uses only soft zero tests, and are complete.

In this paper, we achieve this goal for one of the simplest geometric problems in the continua: determining the geometry of zeros of a complex analytic function f [11]. One formulation of this classical problem is called **root isolation**, defined as follows: given an input function f and a region of interest $B_0 \subseteq \mathbb{C}$, to compute a maximal set $\mathcal{D} = \{D_i : i = 1, \ldots, n\}$ of pairwise disjoint disks, each containing exactly one distinct root of f in B_0. For algebraic polynomials, algebraic techniques such as Sturm, Descartes, Continued Fraction methods are available [5]. With soft zero tests, our analytic techniques cannot distinguish between a root of multiplicity k and a cluster of k roots. Hence we normally require f to be "nice", namely, has simple roots only. With our completeness goal, we must allow multiple roots. So we now associate a multiplicity $\mu_i \geq 1$ with each output disk D_i, meaning that D_i contains a "cluster" of μ_i roots (counted with multiplicity). Thus the exact root isolation problem is transformed into the **root clustering problem**.

All proofs are provided in an Appendix.

¶*1. Related Work.* A classic reference for the geometry of roots is Marden [11]. Rahman and Schmeisser [14] is a comprehensive modern account. There is a large literature on exact root isolation for polynomials and its complexity (see [5] and references therein). For analytic functions, Giusti et al [6] noted that "in contrast to polynomials, few algorithms are known for locating and approximating clusters of zeros of analytic functions". Their paper [6] contains a review of what is known, and they provided an analysis of Newton iteration (generalized to multiple roots with Schröder's iteration) using a generalization of Smale's α-theory. Like Rump [16], many papers (e.g., [12]) focus on predicates for confirming analytic root clusters; they do not necessarily synthesize these predicates into a global method for locating root clusters. Yakoubsohn [20] uses only exclusion methods (but without root confirmation) and ϵ cut-offs for analytic zeros; he further provided complexity analysis. Another approach to analytic zeros is to use subdivision combined with the argument principle (e.g., [9,3]). Algorithms for roots of polynomials using argument principle are also known, but their complexity are suboptimal in this case. Intuitively, they are suboptimal because of unnecessary exact root determination in *each* subdivision box.

2 Conditions for Root Clustering

We address two basic questions. First, when does the set of roots in a disk D form a meaningful cluster? Second, what computational properties of the input function f allow us to construct effective and exact root clustering algorithms?

¶*2. What is a root cluster?* For a disk $D \subseteq \mathbb{C}$, let $r(D)$ and $m(D)$ denote its radius and center, and for $\alpha > 0$, let αD denote the disk centered at $m(D)$ with radius $\alpha r(D)$. Suppose $f : \mathbb{C} \to \mathbb{C}$ is an analytic function. Define $\tau(\mu) := \min\{1+\mu, 3\}$. A disk $D \subseteq \mathbb{C}$ is **isolating** for $f(z)$ if there is an $\mu \geq 0$ such that both D and $\tau(\mu)D$ contain exactly μ roots of f (counted with multiplicity). If $\mu = 0$, then D is called an **exclusion disk**. If $\mu \geq 1$, the non-empty set of roots in D is called a (root) **cluster**. The following shows that our clusters are natural, and are determined only by the "geometry of the roots".

Lemma 1. *Let C_0 be a root cluster of f. Then there is a unique unordered tree $T(C_0)$ rooted at C_0 whose set of nodes are the root clusters contained in C_0. Parent child relation in $T(C_0)$ is defined using the relation: $C \subseteq C' \subseteq C_0$ iff C is a descendent of C'.*

A collection $\mathcal{D} = \{D_1, \ldots, D_n\}$ of pairwise disjoint isolating disks is called an **isolating system** for f in B_0 if (1) each D_i has at least one root and $m(D_i) \in B_0$, and (2) each root of f in B_0 is in some D_i. Call $\mathcal{D}$ an **ϵ-isolating system** in case each $D_i \in \mathcal{D}$ has radius at most ϵ. Note that roots outside B_0 but within distance ϵ from the boundary of B_0 are allowed to appear in $\mathcal{D}$.

We now formalize the **root clustering problem**: given an analytic function $f : \mathbb{C} \to \mathbb{C}$, a closed square box $B_0 \subseteq \mathbb{C}$ and $\epsilon > 0$, to compute an ϵ-isolating system for f in B_0. We may omit the ϵ parameter if $\epsilon = \infty$.

¶*3. On Box Functions and (δ, ϵ)-Approximations.* Unlike algebraic polynomials, it is a non-trivial issue to specify an input analytic function f. In practice, functions are parametrized by numerical parameters. E.g., polynomials are parametrized by coefficients, and hypergeometric functions by their hypergeometric parameters. Such functions may be composed using standard operations. These parameters may be arbitrarily approximated (e.g., the coefficients are algebraic numbers). *Based on these parameters, we assume that f and all its higher derivatives are effectively approximated by (a) box functions and (b) (δ, ϵ)-approximations*, as explained next.

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function. Write $|x|$ for the ∞-norm of $x \in \mathbb{R}^d$. Following [22], real numbers are approximated by elements of the set $\mathbb{F} = \{m2^n : m, n \in \mathbb{Z}\}$ of dyadic numbers; also, let $\square\mathbb{F}$ denote the set of closed intervals with endpoints in $\mathbb{F}$. (a) A **box function** for f, usually denoted $\square f$, is $\square f : \square\mathbb{F}^d \to \square\mathbb{F}$ such that, for any sequence of boxes $B_i \subseteq \square\mathbb{F}^d$ $(i = 1, 2, \ldots)$ that strictly converges to a point $\alpha \in \mathbb{R}^d$ as $i \to \infty$, then $\square f(B_i) \to f(\alpha)$. Box functions are easy to construct using interval arithmetic. (b) A **(δ, ϵ)-approximation** of f is

$$\widehat{f} : \mathbb{F}^{d+1} \to \mathbb{F}^2 \tag{1}$$

such that,[4] for all $x \in \mathbb{F}^d$ and $p \in \mathbb{F}$, if $x' = x \pm 2^{-\widehat{f}_0(x;p)}$, then

$$f(x') = \widehat{f}_1(x;p) \pm 2^{-p}.$$

[4] We write "$a = b \pm \epsilon$" to mean that $|b - a| \leq \epsilon$, and write "$[a \pm \epsilon]$" for the interval $[a - \epsilon, a + \epsilon]$.

Here, $\widehat{f}(x;p)$ is written as the pair $(\widehat{f}_0(x;p), \widehat{f}_1(x;p)) \in \mathbb{F}^2$. By Heine-Borel, the existence of $\widehat{f}$ implies the existence of box functions $\Box f$. We can view $\delta := 2^{-\widehat{f}_0(x;p)}$ and $\epsilon := 2^{-p}$ as the input and output perturbation bounds. The function f is clearly continuous if it has a (δ, ϵ)-approximation, corresponding to the standard definition of continuity: for all $\epsilon > 0$, there exists $\delta > 0$, such that if $x' = x \pm \delta$ then $f(x') = f(x) \pm \epsilon$.

These definitions extend to a complex function $f : \mathbb{C} \to \mathbb{C}$ provided we view it as the function $f = (f_x, f_y) : \mathbb{R}^2 \to \mathbb{R}^2$. Then a (δ, ϵ)-approximation of f is just a pair $(\widehat{f}_x, \widehat{f}_y)$ where each $\widehat{f}_i$ $(i = x, y)$ is a (δ, ϵ)-approximation of f_i. But we can combine the δ and ϵ parameters of the individual f_i's and obtain $\widehat{f} = (\widehat{f}_x, \widehat{f}_y) : \mathbb{F}^3 \to \mathbb{F}^3$.

What are examples of parametrized family of function with such properties? Most elementary functions can be viewed as hypergeometric functions with rational parameters; for this class, we have shown (δ, ϵ)-algorithms ([4]), and moreover, the derivatives of a hypergeometric function is effectively derived from its parameters. Suppose we view $f(z)$ as the function $F(\mathbf{a}; z)$ where $\mathbf{a}$ are the parameters that specify f, and F is continuous in these parameters. Our notion of (δ, ϵ)-approximation can now be applied to F, leading to algorithms in which f itself is perturbed. Our algorithm below could be viewed this way.

3 Predicates for Root Clusters

To provide a complete method for localizing roots, we need a predicate $C_k(D)$ to confirm that a given disk $D \subseteq \mathbb{C}$ contains k roots of f, counted with multiplicity. Rump [16] reviewed this problem, giving 10 different predicates. We will be focusing on one of these predicates, from Pellet [11,14].

Fix an analytic function $f : \mathbb{C} \to \mathbb{C}$. For integer $k \geq 0$ and reals $r, K \geq 1$, define the predicate

$$C_k(m, r, K) : \quad |f_k(m)| r^k > K \sum_{i \neq k} |f_i(m)|\, r^i \tag{2}$$

where $f_i(m) := \frac{f^{(i)}(m)}{i!}$ (coefficients of z^i in the Taylor expansion of $f(z)$ at m). The constant K will be important later when discussing soft versions of these tests. When $K = 1$, just write "$C_k(m, r)$" for $C_k(m, r, K)$. Note that $C_0(m, r)$ (i.e., $k = 0$) is exclusion predicate of [17].

Lemma 2. *If $C_k(m, r)$ holds then the complex disk $D_m(r) \subseteq \mathbb{C}$ contains exactly k roots of f.*

When f is a polynomial, we obtain Pellet's theorem [11]:

Theorem 1 (Pellet (1881)). *If $Q(z) = \sum_{i=0}^{n} q_i z^i$ with $q_n q_0 \neq 0$ and $|q_k| z^k - \sum_{i \neq k} |q_i| z^i$ has two real positive roots $r < R$, then Q has exactly k roots in $D_0(r)$ and there are no roots in the annulus $D_0(R) \setminus D_0(r)$.*

Rump observed that Pellet's method is among the best of his 10 methods; the main limitation is that the size of its coefficients tend to overflow machine precision (his experimental setup is limited to machine precision).

¶*4. Effective Analytic Version C_k Test.* For an analytic function f, the C_k test is not effective. For this, we need the complex form of Taylor's Theorem with Remainder. This seems to be a little known result[5] due to Darboux (1876). A more general statement with proof is conveniently provided by Batra [1, Appendix].

Theorem 2 (Darboux). *Let $f : D_0 \to \mathbb{C}$, analytic in an open disk D_0 be given, and let $a, b \in D_0$. Then there exists $0 \le \Theta \le 1$ and $\omega \in \mathbb{C}, |\omega| \le 1$ such that for $h := b - a$ and $\xi := a + \Theta(b - a)$ it holds true that*

$$f(b) = \sum_{\nu=0}^{k} f_\nu(a) h^\nu + \omega h^{k+1} f_{k+1}(\xi).$$

Now we introduce the interval version of the C_k test of (2) above:

$$\square C_k(m, r, K): \quad |f_k(m)| r^k > K \left(\sum_{i=0}^{k-1} |f_i(m)|\, r^i + |\square f_{k+1}(D_m(r))|\, r^{k+1} \right). \quad (3)$$

Again, "$\square C_k(m, r)$" refers to $\square C_k(m, r, 1)$. Here, $\square f_{k+1}(D)$ is some box function for $f_{k+1}(z)$. Note: we use the absolute value $|\square f_{k+1}(\cdots)|$ of the output box. The analogue of Lemma 2 can be shown using Darboux's theorem:

Lemma 3. *If $\square C_k(m, r)$ holds, then $D_m(r)$ contains exactly k roots counted with multiplicities.*

4 Exact Algorithm for Root Clustering

We give a simple version of our root clustering algorithm, assuming the exact evaluation of the predicates C_k and $\square C_k$ and ignoring fine tuning that may be important in practice. Our algorithm uses the classic subdivision paradigm (e.g., [21]). This may be viewed as the repeated subdivision of an initial box $B_0 \subseteq \mathbb{C}$, each box being subdivided ("split") into four congruent subboxes, until all the boxes satisfy some predicate. If X is a box or disk with center m_X and radius r_X, then we write "$C_k(X)$" instead of $C_k(m_X, r_X)$.

Define the **function firstC**(B, N) to return the smallest $k = 0, \ldots, N$ such that $D(2k \cdot B)$ is isolating and contains k roots; otherwise, firstC(B, N) returns -1. To verify that $D(2k \cdot B)$ is isolating, we can check the predicates $\square C_k(2k \cdot B)$ and $\square C_k(\tau(k) 2k \cdot B)$. Alternatively, in case f is a polynomial, we can check that $C_k(2k \cdot B)$ and $C_k(\tau(k) 2k \cdot B)$ holds.

Our algorithm's input has the form (f, B_0, N) where f is analytic and B_0 is a closed square box such that $D(B_0)$ has at most N roots. For instance, if f is a polynomial, we can choose N to be its degree. For general analytic functions, this N may be first estimated by numerical integration. Our algorithm has three queues Q_0, Q_1 and $\mathcal{D}$. Queue Q_0 contains boxes in arbitrary order, Q_1 is a max-priority queue containing box-integer pairs (B, k), with k as the priority.

[5] Thanks to Prashant Batra for bringing this to our attention.

Queue $\mathcal{D}$ is the output, and contains (B,k) pairs in arbitrary order. Each (B,k) represents an isolating disk $2k \cdot D(B)$ containing k roots. A pair (B,k) and (B',k') is said to be **in conflict** if their isolating disks intersect.

```
EXACT ROOT CLUSTERING ALGORITHM

      Input: f : ℂ → ℂ, B_0 ⊆ ℂ, N ≥ 1, as described.
      Output: An isolating system 𝒟 for f in B_0.

      Q_0 ← {B_0}, Q_1 ← ∅, 𝒟 ← ∅      ◁ Initialize Queues
0.    while (Q_0 is non-empty)
          B ← Q_0.pop()
          k ← firstC(B, N)
1.        If k < 0, split B and push its 4 children into Q_0.
2.        elif 1 ≤ k ≤ N, Q_1.push(B, k)
3.    while (Q_1 is non-empty)
          (B, k) ← Q_1.pop()
4.        If (B, k) does not conflict with any pair in 𝒟,
5.            𝒟.push(B, k)
      Return 𝒟
```

Theorem 3. *The Exact Root Clustering Algorithm halts, and produces an isolating system for the roots of f in B_0.*

We easily modify this algorithm to compute an ϵ-isolating systems: Let the precision $p \in \mathbb{F}$ is given as input, $p := \lg(1/\epsilon)$. Replace $\mathrm{firstC}(B,N)$ by $\mathrm{firstC}(B,N,p)$ which returns -1 if $r(B) > 2^{-p} = \epsilon$. Otherwise, it return $\mathrm{firstC}(B,N)$ as before. If when ϵ is small enough, we isolate only the roots in B_0.

5 Applications of Soft Zero Tests

The preceding algorithm is exact but not effective as it assumes the exact evaluation of C_k or $\square C_k$ in firstC. Such algorithms are often deemed sufficient (cf. the papers in ¶1, Related Work). It is assumed that a numerical implementation of the algorithm can invoke error analysis to tell us the circumstances under which the output is correct. Unfortunately, this falls short of the usual standard for algorithms in theoretical computer science. The solution we will now provide is to replace the above predicates by their soft versions, denoted $\widetilde{C}_k$ and $\square\widetilde{C}_k$, respectively.

¶5. *Soft Zero Test.* First consider the following **soft zero test**: given two numerical expressions A and B, both non-negative and at least one positive, determine either the non-zero sign of $A-B$, or that A, B are **relatively equal** in the sense that $\frac{1}{2}A < B < 2A$. Observe that if A, B are relative equal but $A \neq B$, then the output is non-deterministic: both the (correct) non-zero sign of $A-B$ or relative equality are possible outputs. Write $(A)_p$ to mean any p-bit approximation of A, i.e., $(A)_p = A \pm 2^{-p}$. We are allowed to compute any p-bit approximation of A and B for this problem. Here is our Soft Zero Test procedure: start with $p = 1$. We halt if one of the following two conditions hold:

(I) $|(A)_p - (B)_p| > 2^{1-p}$.
(II) $|(A)_p - (B)_p| \le 2^{1-p}$ and $\max\{(A)_p, (B)_p\} \ge 7 \cdot 2^{-p}$.

If (I) holds, output the sign of $(A)_p-(B)_p$, and if (II) holds, output "RELATIVE EQUALITY". Otherwise, we double p and repeat.

Theorem 4. *The Soft Zero Test procedure halts and is correct.*

We apply the soft zero test to implement soft predicate $\square\widetilde{C}_k(m,r)$ (the case of $\widetilde{C}_k(m,r)$ is similar). Recall that we know (δ,ϵ)-approximations $\widehat{f}_i : \mathbb{F}^3 \to \mathbb{F}^3$ of each Taylor coefficient function $f_i(z)$, $i \ge 0$, (see (1)). To decide $\square\widetilde{C}_k(m,r)$, let us write the predicate (3) as the inequality $A > B$ where $A := |f_k(m)|r^k$, and $B = E+F$, with $E := \sum_{i=0}^{k-1} |f_i(m)|r^i$ and $F := |\square f_{k+1}(D_m(r))|\, r^{k+1}$. It is easy to compute $(A)_p, (E)_{p+1}$ and $(F)_{p+1}$ using the $\widehat{f}_i$'s. Note that F is an interval, say $[a,b]$, and our approximation amounts to widening the output interval by at most 2^{-p}, $(F)_{1+p} \subseteq [a-2^{-1-p}, b+2^{-1-p}]$. So $(B)_p = (E)_{p+1} + (F)_{p+1}$. Therefore, we could apply our soft zero test to determine the non-zero sign $A-B$, or determine the "RELATIVE EQUALITY" of A, B. If $A - B$ is positive, we output success for our soft predicate $\square\widetilde{C}_k(m,r)$, and otherwise failure.

Lemma 4.
(a) If the soft $\square\widetilde{C}_k(m,r)$ *succeeds, then exact* $\square C_k(m,r)$ *succeeds.*
(b) If exact $\square C_k(m,r,2)$ *succeeds, then soft* $\square\widetilde{C}_k(m,r)$ *succeeds.*

We now describe our **Soft Root Clustering Algorithm**. Basically, we use the soft $\square\widetilde{C}_k$ instead of the exact $\square C_k$ in the Exact Root Clustering Algorithm. These predicates are used within the function firstC(B,N). But there is an important twist: we must now test if the disks $D(4k \cdot B)$, not $D(2k \cdot B)$, are isolating for $k = 0, 1, \ldots, N$. With this modification, Thm. 6 (below) and Lemma 4(b) implies halting. Finally, by exploiting our (δ,ϵ)-approximations $\widehat{f}_i : \mathbb{F}^3 \to \mathbb{F}^3$ of the Taylor coefficients, we can turn this into a (δ,ϵ)-algorithm in the sense that we also compute a $\delta^* > 0$ such that for all δ^*-perturbations of the input, our ϵ-output remains correct. Recall that the ϵ input parameter is not explicitly described, but it is easy to take this into account. This yields:

Theorem 5. *The Soft Root Clustering Algorithm is a complete (δ,ϵ)-algorithm for the root clustering problem that is based on soft zero tests.*

6 Analysis of C_k Test

Suppose the analytic function $f(z)$ has a root α of multiplicity $k \ge 0$. So $f^{(k)}(\alpha) \ne 0$ and $f^{(j)}(\alpha) = 0$ for $j = 0, \ldots, k-1$. Then

$$f(z) = \sum_{i \ge 0} f_i(\alpha)(z-\alpha)^i = \sum_{i \ge k} f_i(\alpha)(z-\alpha)^i.$$

Notation: In the analysis of this section, we let m denote a point near α, and let $r := |m - \alpha|$ (the "radius"). If E, F are numerical expressions that depend on r, we shall write "$E \simeq F$" to mean that, as $r \to 0$, we have $E = F(1 \pm o(1))$. Also "$E \lesssim F$" means $E < F$ as $r \to 0$. Likewise, "$E = O(F)$" means there is a constant $K > 0$ such that $E \leq K \cdot F$ for all r small enough. These notations are illustrated in the statement of the next lemma.

Lemma 5. *For $j \geq 0$:*

$$|f_j(m)|r^j \simeq \begin{cases} |f_k(\alpha)|r^k \binom{k}{j} & \text{if } \ j \leq k \\ O(r^j) & \text{if } \ j > k \end{cases}$$

In our application, instead of using radius $r = |m - \alpha|$, we need to consider cr for some constant $c > 0$:

Lemma 6.

$$\sum_{j=0}^{k} \left| \frac{f_j(m)(cr)^j}{f_k(m)(cr)^k} \right| \simeq \left(1 + \frac{1}{c}\right)^k .$$

This follows from the previous lemma by summation.

By separating out the f_k term in the previous lemma, we get:

Lemma 7. *If $c \geq k$, then*

$$\sum_{j=0}^{k-1} |f_j(m)| \, (cr)^j \lesssim |f_k(m)| r^k \left(k c^{k-1}(e-1)\right).$$

Theorem 6. *Let $D_i = D_{m_i}(r_i)$ $(i \geq 0)$ be a sequence of disks, $D_{i+1} \subseteq D_i$, that converges to a point α. Let α have multiplicity $k \geq 0$, and c be any constant greater than $(e-1)kK$.*
(1) The test $\square C_k(m_i, cr_i, K)$ succeeds for i large enough.
(2) If f is a polynomial, the test $C_k(m_i, cr_i, K)$ succeeds for i large enough.

7 Conclusion

There is increasing interest in numerical, evaluation-based approaches to exact geometric algorithms: from root isolation to topology of curve and surfaces. Such algorithms are realistic, practical, and have adaptive complexity. It is part of the trend towards symbolic-numeric computation. Until now, the evaluation algorithms for isolating the roots of a function f have two limitations: (1) they require f to have simple roots, and (2) they assume that f is a polynomial. In this paper, we have produced a evaluation-based algorithm for its generalization to root clustering. Our algorithm (1') allows f to have multiple roots and (2') applies to analytic functions. In the future, we plan to produce complexity analysis as well as implementation of our algorithms. We pose as a general challenge to produce similar soft-but-complete algorithms for other geometric problems.

References

1. P. Batra. Globally convergent, iterative path-following for algebraic equations. *Math. in Computer Sci.*, 4(4):507–537, 2010. Special Issue.
2. L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer-Verlag, New York, 1998.
3. M. Dellnitz, O. Schütze, and Q. Zheng. Locating all the zeros of an analytic function in one complex variable. *J. Comput. Appl. Math.*, 138(2):325–333, 2002.
4. Z. Du and C. Yap. Absolute approximation of the general hypergeometric functions. In *Proc. 7th Asian Symp. on Computer Math. (ASCM)*, pp. 246–249, 2005. KIAS, Seoul, Dec 8–10, 2005.
5. I. Z. Emiris, V. Y. Pan, and E. P. Tsigaridas. Algebraic and numerical algorithms. In M. J. Atallah and M. Blanton, eds., *Algorithms and Theory of Computation Handbook*, vol. I, chap. 17. CRC Press Inc., Boca Raton, Florida, 3nd edn., 2012.
6. M. Giusti, G. Lecerf, B. Salvy, and J.-C. Yakoubsohn. On location and approximation of clusters of zeros of analytic functions. *Found. Comp. Math.*, 5(3):257–311, 2005.
7. D. Halperin, E. Fogel, and R. Wein. *CGAL Arrangements and Their Applications*. Springer-Verlag, Berlin and Heidelberg, 2012.
8. L. A. Hemaspaandra and M. Ogihara. *The Complexity Theory Companion*. Springer-Verlag, 2002.
9. T. Johnson and W. Tucker. Enclosing all zeros of an analytic function - a rigorous approach. *J. Comput. Appl. Math.*, 228(1):418–423, June 2009.
10. K.-I. Ko. *Complexity Theory of Real Functions*. Progress in Theoretical Computer Science. Birkhäuser, Boston, 1991.
11. M. Marden. *The Geometry of Zeros of a Polynomial in a Complex Variable*. Math. Surveys. American Math. Soc., New York, 1949.
12. X.-M. Niu, T. Sakurai, and H. Sugiura. A verified method for bounding clusters of zeros of analytic functions. *J. Comput. Appl. Math.*, 199(2):263–270, Feb. 2007.
13. S. Plantinga and G. Vegter. Isotopic approximation of implicit curves and surfaces. In *Proc. Symp. on Geometry Processing*, pp. 245–254, New York, 2004. ACM Press.
14. Q. I. Rahman and G. Schmeisser. *Analytic Theory of Polynomials*. Oxford University Press, 2002.
15. H. Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, New York, 1967.
16. S. M. Rump. Ten methods to bound multiple roots of polynomials. *J. Computational and Applied Mathematics*, 156:403–432, 2003.
17. M. Sagraloff and C. K. Yap. A simple but exact and efficient algorithm for complex root isolation. *36th ISSAC*, pp. 353–360, 2011. June 8-11, San Jose, California.
18. J. Traub, G. Wasilkowski, and H. Woźniakowski. *Information-Based Complexity*. Academic Press, Inc, 1988.
19. K. Weihrauch. *Computable Analysis*. Springer, Berlin, 2000.
20. J.-C. Yakoubsohn. Numerical analysis of a bisection-exclusion method to find zeros of univariate analytic functions. *J. of Complexity*, 21(5):652–690, 2005.
21. C. Yap, V. Sharma, and J.-M. Lien. Towards Exact Numerical Voronoi diagrams. In *9th Int. Symp. of Voronoi Diagrams in Sci. and Eng. (ISVD).*, pp. 2–16. IEEE, 2012. Invited Talk. June 27-29, 2012, Rutgers U., NJ.
22. C. K. Yap. Theory of real computation according to EGC. In *Reliable Implementation of Real Number Algorithms: Theory and Practice*, No. 5045 in LNCS, pp. 193–237. Springer, 2008.

APPENDIX: Proofs

The proofs of the theorems of Sections 4 and 5 will make forward references to the lemmas of Section 6.

Section 2: Conditions for Root Clustering

Lemma 1. *Let C_0 be a root cluster of f. Then there is a unique unordered tree $T(C_0)$ rooted at C_0 whose set of nodes are the root clusters contained in C_0. Parent child relation in $T(C_0)$ is defined using the relation: $C \subseteq C' \subseteq C_0$ iff C is a descendent of C'.*

Proof. We show that if C, C' are distinct root clusters, then they are either disjoint or one is included in the other. In proof, suppose $C \cap C'$ contains $k \geq 1$ roots (counted with multiplicity). We must show that C is contained in C' or vice-versa. By way of contradiction, assuming there is no containment relationship. Then C and C' each have at least $k+1$ roots. By definition, C (C') is contained in an isolating disk D (D'). Wlog, let $r(D) \geq r(D')$ and C has $\mu \geq k+1 \geq 2$ roots. Thus $\tau(\mu) = 3$. Clearly $D \cap D'$ is non-empty, and so $D' \subseteq 3D = \tau(\mu)D$. But there are no roots in $\tau(\mu)D \setminus D$, and so all the roots of D' are contained in D. Thus $C' \subseteq C$, contradiction. Note: this also shows that the set of nodes of $T(C_0)$ is a natural set of clusters. **Q.E.D.**

Section 3: Predicates for Root Clusters

Lemma 2. *If $C_k(m, r)$ holds then the complex disk $D_m(r) \subseteq \mathbb{C}$ contains exactly k roots of f.*

Proof.

Define $g(z) := f_k(m)(z-m)^k$. Then $C_k(m, r)$ implies that for ζ on the boundary of $D_m(r)$, we have

$$|g(\zeta)| = |f_k(m)|r^k > |f(\zeta) - g(\zeta)|.$$

Therefore, by Rouché's theorem $g(z)$, and $f(z)$ have the same number of roots in $D_m(r)$. Clearly, $g(z)$ has m as a root with multiplicity k, which implies that $f(z)$ also has exactly k roots in $D_m(r)$.

Q.E.D.

Lemma 3. *If $\square C_k(m, r)$ holds, then $D_m(r)$ contains exactly k roots counted with multiplicities.*

Proof.

Define $g(z) := f_k(m)(z - m)^k$. So for ζ on the boundary of $D_m(r)$, we have

$$|g(\zeta)| = |f_k(m)|r^k > \sum_{i=0}^{k-1} |f_i(m)|r^i + \max |f_{k+1}(D_m(r))|r^{k+1}.$$

But by triangular inequality the RHS of the inequality above is greater than

$$|f(\zeta) - g(\zeta)| = |\sum_{i=0}^{k-1} f_i(m)(z-m)^i + \omega f_{k+1}(\xi)(z-m)^{k+1}|,$$

where ω and ξ are as in Darboux's theorem. Thus, $|g(\zeta)| > |f(\zeta) - g(\zeta)|$ and by Rouché's theorem g and f have the same number of zeros in $D_m(r)$, namely k.

Q.E.D.

Section 4: Exact Algorithm for Root Clustering

Theorem 3. *The Exact Root Clustering Algorithm halts, and produces an isolating system or the roots of f in B_0.*

Proof. Upon termination, the queue Q_1 contains a collection of pairs (B_i, k_i), representing isolating disks $2k_i D(B_i)$. Why is this an isolating system? Note that a box B is discarded if $C_0(B)$ holds. This is implicit in Lines 1–2, since we only process a box if $k \neq 0$. It follows that at the start Line 3, every root in B_0 is in some cluster of the queue Q_1 (by Thm. 6). Lines 4–5 discards a box (B, k) if it conflicts with some (B', k') that is already in $\mathcal{D}$. Since $k' \geq k$ (as Q_1 is a priority queue), it means the roots in $2k \cdot D(B)$ are contained in $2k' \cdot D(B')$. Thus the final $\mathcal{D}$ represents an isolating system for f in B_0. It remains to prove termination. Only the first while-loop (Line 0) has an issue. Suppose there is an infinite path $(B_i : i \geq 0)$ in the (implicit) subdivision tree. This means that $\mathrm{firstC}(B_i, N) < 0$ for all i. Suppose $B_i \to \alpha \in \mathbb{C}$ as $i \to \infty$. If α is a k-fold root of f, Thm. 6 implies that the tests $C_k(2k \cdot B_i)$ and $C_k(\tau(k)2k \cdot B_i)$ succeeds for i large enough. This contradict the assumption that $\mathrm{firstC}(B_i, N) < 0$ for all i.

Q.E.D.

Section 5: Applications of Soft Zero Tests

Theorem 4. *The Soft Zero Test procedure halts and is correct.*

Proof. We show that (II) implies $\frac{1}{2}A < B < 2A$. Let $\epsilon := 2^{-p}$. Then $|(A)_p - (B)_p| \leq 2\epsilon$ implies $|A - B| \leq 4\epsilon$. Also $\max\{(A)_p, (B)_p\} \geq 7\epsilon$ implies $\min\{(A)_p, (B)_p\} \geq 5\epsilon$, and so $\min\{A, B\} \geq 4\epsilon$. Wlog, $A \leq B$. Then $A \leq B < B + 4\epsilon \leq 2B$. Hence $\frac{1}{2}A < B < 2B$. It remains to prove termination. There are two possibilities: (1) $AB \neq 0$ and (2) $AB = 0$. First consider $AB \neq 0$: if $A = B$, then (II) will hold when $p \geq 3 - \lg A$. Otherwise $A \neq B$, then (I) will hold when $p > 1 - \lg |A - B|$. For possibility (2), assume WLOG that $A = 0$. Then we see that (I) will hold when $p > 2 - \lg B$.

Q.E.D.

Lemma 4.
(a) If $\square\widetilde{C}_k(m, r)$ succeeds, then $\square C_k(m, r)$ succeeds.
(b) If $\square C_k(m, r, 2)$ succeeds, then $\square\widetilde{C}_k(m, r)$ succeeds.

Proof. (a) The test $\square C_k(m, r)$ succeeds iff $A > B$ (for the appropriate A, B). . In our soft version, when we output success, it means we have verified $A > B$.

(b) The exact tests $\square C_k(m, r, 2)$ and $\square C_k(m, r)$ can be written as $A > 2B$ and $A > B$ (respectively). Suppose $A > 2B$ holds. In the soft version, the test $A > B$ can only fail under one of two ways: either (i) $A < B$ or (ii) A, B are relatively equal. But if $A > 2B$, both (i) and (ii) cannot hold. Hence the soft version must succeed, as claimed. **Q.E.D.**

Theorem 5. *The Soft Root Clustering Algorithm is complete (δ, ϵ)-algorithm for the root clustering problem that is based on soft zero tests.*

Proof. Two details remain:

(1) Why does the soft algorithm halt? According to Thm. 6 says that the test $\square C_k(m, cr, K)$ will eventually succeed if we choose $c \geq (e-1)kK$. For $K = 2$, we can choose $c = 4k$. Next, Lemma 4(b) tells us that if $\square C_k(m, 4r, 2)$ succeeds, then eventually $\square \widetilde{C}_k(m, 4r)$ will succeed.

(2) We want an (δ, ϵ)-algorithm in the sense that we also compute a $\delta^* > 0$ such that for all δ^*-perturbations of the input, our ϵ-output remains correct. Our goal is to compute q^* where $\delta^* = 2^{-q^*}$. Recall that we use the (δ, ϵ)-approximations $\widehat{f_i} : \mathbb{F}^3 \to \mathbb{F}^2$ to evaluate the Taylor coefficients $f_i(m)$ for $i \geq 0$. In each call to $\widehat{f_i}$, we obtain a $q \in \mathbb{F}$, corresponding to the precision of the parameters of the function f_i and the argument m. I.e., 2^{-q} is the perturbation of f_i We update q^* to be the minimum of all the q's obtained in this way. **Q.E.D.**

Section 6: Analysis of C_k Test

Lemma 5. *For $j \geq 0$:*

$$|f_j(m)|r^j \simeq \begin{cases} |f_k(\alpha)|r^k \binom{k}{j} & \text{if } j \leq k \\ O(r^j) & \text{if } j > k \end{cases}$$

Proof. By differentiating the Taylor expansion $f(m) = \sum_{i \geq 0} f_i(\alpha)(m-\alpha)^i$, we get

$$f_j(m) = \frac{f^{(j)}(m)}{j!} = \begin{cases} \sum_{i \geq k} f_i(\alpha)\binom{i}{j}(m-\alpha)^{i-j} & \text{if } j \leq k, \\ \sum_{i \geq j} f_i(\alpha)\binom{i}{j}(m-\alpha)^{i-j} & \text{if } j > k. \end{cases}$$

$$f_j(m)(m-\alpha)^j = \begin{cases} \sum_{i \geq k} f_i(\alpha)\binom{i}{j}(m-\alpha)^{i} & \text{if } j \leq k, \\ \sum_{i \geq j} f_i(\alpha)\binom{i}{j}(m-\alpha)^{i} & \text{if } j > k. \end{cases}$$

$$|f_j(m)|r^j = \begin{cases} r^k \left| f_k(\alpha)\binom{k}{j} + \sum_{i > k} f_i(\alpha)\binom{i}{j}(m-z)^{i-k} \right| & \text{if } j \leq k, \\ r^j \left| \sum_{i \geq j} f_i(\alpha)\binom{i}{j}(m-z)^{i-j} \right| & \text{if } j > k \end{cases}$$

$$\begin{cases} \simeq r^k \left| f_k(\alpha) \right| \binom{k}{j} & \text{if } j \leq k, \\ = O(r^j) & \text{if } j > k. \end{cases}$$

Q.E.D.

Lemma 6.

$$\sum_{j=0}^{k} \left| \frac{f_j(m)(cr)^j}{f_k(m)(cr)^k} \right| \simeq \left(1+\frac{1}{c}\right)^k.$$

Proof. For $j = 0, \ldots, k$,

$$\begin{aligned}
\left|\tfrac{f_j(m)(cr)^j}{f_k(m)(cr)^k}\right| &= c^{j-k}\left|\tfrac{f_j(m)r^j}{f_k(m)r^k}\right| \\
&\simeq c^{j-k}\left(\tfrac{r^k|f_k(\alpha)|\binom{k}{j}}{r^k|f_k(\alpha)|}\right) \text{ (by previous lemma)} \\
&= \tbinom{k}{j}c^{j-k}. \\
\textstyle\sum_{j=0}^k \left|\tfrac{f_j(m)(cr)^j}{f_k(m)(cr)^k}\right| &\simeq \textstyle\sum_{j=0}^k \binom{k}{j}c^{k-j} \\
&= \left(1+\tfrac{1}{c}\right)^k.
\end{aligned}$$

Q.E.D.

Lemma 7. *If $c \geq k$, then*

$$\sum_{j=0}^{k-1} |f_j(m)|\,(cr)^j \lesssim |f_k(m)|r^k\left(kc^{k-1}(e-1)\right).$$

Proof. Rewriting the previous lemma,

$$\begin{aligned}
\sum_{j=0}^{k} |f_j(m)|\,(cr)^j &\simeq |f_k(m)|(cr)^k\left(1+c^{-1}\right)^k \\
&\simeq |f_k(m)|r^k(c+1)^k. \\
\sum_{j=0}^{k-1} |f_j(m)|\,(cr)^j &\simeq |f_k(m)|r^k\left((c+1)^k - c^k\right) \\
&\lesssim |f_k(m)|r^k\left(kc^{k-1}\right)\left(\sum_{i\geq 1} 1/i!\right) \\
&= |f_k(m)|r^k\left(kc^{k-1}\right)(e-1).
\end{aligned}$$

Q.E.D.

Theorem 6. *Let $D_i = D_{m_i}(r_i)$ $(i \geq 0)$ be a sequence of disks, $D_{i+1} \subseteq D_i$, that converges to a point α. Let α have multiplicity $k \geq 0$, and c be any constant greater than $(e-1)kK$.*
(1) The test $\square C_k(m_i, cr_i, K)$ succeeds for i large enough.
(2) If f is a polynomial, the test $C_k(m_i, cr_i, K)$ succeeds for i large enough.
Proof. (1) Note that $r_i \geq |m_i - \alpha|$, and all our asymptotic estimates in our analysis for $r = |m-\alpha|$ can be applied here. We must show that $\square C_k(m_i, cr_i, K)$

holds (ev. i). Here we write "ev. i" (read "eventually i") to mean "for i large enough". Write $\square C_k(m_i, cr_i, K)$ as the predicate

$$|f_k(m)|(cr)^k > A_i + B_i$$

where $A_i := K \sum_{j=0}^{k-1} |f_j(m_i)|(cr_i)^j$ and $B_i := K \ \left|\square f_{k+1}(D_{m_i}(r_i))\right| (cr_i)^{k+1}$. But

$$\begin{aligned} |f_k(m_i)|(cr_i)^k &= |f_k(m_i)|(cr_i)^k \left(\frac{(e-1)kK + (c-(e-1)kK)}{c} \right) \\ &= \underbrace{K|f_k(m_i)|(cr_i)^k \left(\frac{k(e-1)}{c} \right)}_{A_i'} + \underbrace{|f_k(m_i)|(cr_i)^k \left(\frac{c-(e-1)kK}{c} \right)}_{B_i'} \end{aligned}$$

By Lemma 7, the $A_i' > A_i$ (ev. i). To compare B_i and B_i', since $|f_k(m_i)|$ converges to $|f_k(\alpha)| > 0$, there exist constants $\kappa, \kappa' > 0$ such that

$$B_i \le \kappa (cr_i)^{k+1} \quad \text{and} \quad B_i' \ge \kappa' (cr_i)^k \quad \text{(ev. } i\text{)}.$$

This implies that $B_i' > B_i$ for i large enough.

(2) When f is a polynomial, again we must show (ev. i)

$$|f_k(m)|(cr)^k > A_i + B_i$$

with A_i as before, but $B_i := K \sum_{j \ge k+1} |f_j(m_i)| r_i^i$. Since the sum in B_i is finite, Lemma 5 implies $B_i = O(r^{k+1})$ as $r_i \to 0$. The result follows as before. **Q.E.D.**