From Approximate Factorization to Root Isolation

Kurt Mehlhorn
Max Planck Institute for Informatics
mehlhorn@mpi-inf.mpg.de

Michael Sagraloff
Max Planck Institute for Informatics
msagralo@mpi-inf.mpg.de

Pengming Wang
Max Planck Institute for Informatics
s9pewang@stud.uni-saarland.de

ABSTRACT

We present an algorithm for isolating all roots of an arbitrary complex polynomial p which also works in the presence of multiple roots provided that arbitrary good approximations of the coefficients of p and the number of distinct roots are given. Its output consists of pairwise disjoint disks each containing one of the distinct roots of p, and its multiplicity. The algorithm uses approximate factorization as a subroutine. For the case, where Pan's algorithm [16] is used for the factorization, we derive complexity bounds for the problems of isolating and refining all roots which are stated in terms of the geometric locations of the roots only. Specializing the latter bounds to a polynomial of degree d and with integer coefficients of bitsize less than τ, we show that $\tilde{O}(d^3 + d^2\tau + d\kappa)$ bit operations are sufficient to compute isolating disks of size less than $2^{-\kappa}$ for all roots of p, where κ is an arbitrary positive integer.

Our new algorithm has an interesting consequence on the complexity of computing the topology of a real algebraic curve specified as the zero set of a bivariate integer polynomial and for isolating the real solutions of a bivariate system. For input polynomials of degree n and bitsize τ, the currently best running time improves from $\tilde{O}(n^9\tau + n^8\tau^2)$ (deterministic) to $\tilde{O}(n^6 + n^5\tau)$ (randomized) for topology computation and from $\tilde{O}(n^8 + n^7\tau)$ (deterministic) to $\tilde{O}(n^6 + n^5\tau)$ (randomized) for solving bivariate systems.

Categories and Subject Descriptors

G.1.5 [**Roots of Nonlinear Equations**]: Polynomials, methods for

Keywords

Root Isolation, Root Refinement, Bit Complexity, Numerical Algorithms, Certified Algorithms, Cylindrical Algebraic Decomposition

1. INTRODUCTION

Root isolation is a fundamental problem of computational algebra. Given a univariate polynomial p with complex coefficients and possibly multiple roots, the goal is to compute disjoint disks in the complex plane each containing exactly one root. We assume the existence of an oracle that can be asked for rational approximations of the coefficients of arbitrary precision. In particular, non-rational coefficients can never be learned exactly in finite time.

In this generality, the problem is unsolvable. Consider $p(x) = (x-\sqrt{2})^2(x+1) = x^3 + (-2\sqrt{2}-1)x^2 + (2+2\sqrt{2})x + 2$ and assume that an algorithm terminates after having received approximations 1, α, β, and 2 for the coefficients, where α and β are rational and the polynomial $x^3 + \alpha x^2 + \beta x + 2$ has three distinct roots. If the algorithm outputs two disks, the adversary commits to this α and β as the two middle coefficients. If the algorithm outputs three disks, the adversary commits to p.

The example shows that the problem needs to be restricted. In addition to our assumption that the coefficients of our input polynomial p are provided by coefficient oracles, we further assume that *the number k of distinct roots* is also given. Root isolation is a key ingredient in the computation of a CAD (cylindrical algebraic decomposition) for a given set of multivariate polynomials and, in particular, for computing the topology of algebraic curves and surfaces. In these applications, one has to deal with polynomials with multiple roots and algebraic coefficients which can be approximated to an arbitrary precision. In addition, the number of distinct roots is readily available from an algebraic decomposition of the input. At this point, we refer to some recent symbolic-numeric algorithms [20, 6, 3, 2] which combine structural information derived from symbolic computation and the use of numerical root finding methods to isolate the roots of polynomials with algebraic coefficients for which only approximations are given.

We now give a short overview of our algorithm and our results. Let $p(x) = \sum_{i=0}^{n} p_i x^i$ be a polynomial with k distinct roots $z_1, \ldots, z_k$. For $i = 1, \ldots, k$, let $m_i := \mathrm{mult}(z_i, p)$ be the *multiplicity* of z_i, and let $\sigma_i := \sigma(z_i, p) := \min_{j \neq i} |z_i - z_j|$ be the *separation* of z_i from the other roots of p. Then, our algorithm outputs isolating disks $\Delta_i = \Delta(\tilde{z}_i, R_i)$ for the roots z_i and the corresponding multiplicities m_i. The radii satisfy $R_i < \frac{\sigma_i}{64n}$, thus the center $\tilde{z}_i$ of Δ_i approximates z_i to an error of less than $\frac{\sigma_i}{64n}$. If the number of distinct roots of p differs from k, we make no claims about termination and output.

The coefficients of p are provided by oracles. That is, on input L, the oracle essentially returns binary fraction approximations $\tilde{p}_i$ of the coefficients p_i such that $\left\| p - \sum_{i=0}^{n} \tilde{p}_i x^i \right\| \leq 2^{-L} \|p\|$. Here, $\|p\| := \|p\|_1 = |p_0| + \ldots + |p_n|$ denotes the *one-norm* of p. The details are given in Section 2.

Our algorithm has a simple structure. We first use any algorithm (e.g. [4, 19, 16, 22]) for approximately factorizing the input polynomial. It is required that it can be run with different levels of precision, and that, for any given integer b, it returns approximations $\hat{z}_1$ to $\hat{z}_n$ for the roots of p such that

$$\left\| p - p_n \prod_{1 \leq j \leq n} (x - \hat{z}_j) \right\| \leq 2^{-b} \|p\| . \qquad (1)$$

In a second step, we partition the root approximations $\hat{z}_1$ to $\hat{z}_n$ into

ISSAC'13, June 26–29, 2013, Boston, Massachusetts, USA.

k clusters $C_1,\ldots,C_k$ based on geometric vicinity. We enclose each cluster C_i in a disk $D_i = \Delta(\tilde{z}_i, r_i)$ and make sure that the disks are pairwise disjoint and that the radii r_i are not "too small" compared to the pairwise distances of the centers $\tilde{z}_i$.[1] In a third step, we verify that the n-times enlarged disks $\Delta_i = \Delta(\tilde{z}_i, R_i) = \Delta(\tilde{z}_i, n \cdot r_i)$ are disjoint and that each of them contains exactly the same number of approximations as roots of p counted with multiplicity. If the clustering and the verification succeed, we return the disks $\Delta_1,\ldots,\Delta_k$ and the number of approximations $\hat{z} \in \{\hat{z}_1,\ldots,\hat{z}_n\}$ in the disk as the multiplicity of the root isolated by the disk. If either clustering or verification does not succeed, we repeat with a higher precision. Strzebonski [20] has previously described a similar approach. The main difference is that he used a heuristic for the clustering step and hence could neither prove completeness of his approach nor analyze its complexity. He reports that his algorithm does very well in the context of CAD computation.

In the example above, we would have the additional information that p has exactly two distinct roots. We ask the oracle for an L-approximation of p for sufficiently large L and approximately factor it. Assume, we obtain approximations -1.01, 1.4, and 1.5 of the roots, and let $\hat{p} = (x+1.01)(x-1.4)(x-1.5)$. The clustering step may then put the first approximation into a singleton cluster and the other two approximations into a cluster of size two. It also computes disjoint enclosing disks. The verification step tries to certify that p and $\hat{p}$ contain the same number of roots in both disks. If L and b are sufficiently large, clustering and verification succeed.

If Pan's algorithm [16] is used for the approximate factorization step, then the overall algorithm has bit complexity[2]

$$\tilde{O}\left(n^3 + n^2 \sum_{i=1}^{k} \log M(z_i) + n \sum_{i=1}^{k} \log\left(M(\sigma_i^{-m_i}) \cdot M(P_i^{-1})\right)\right) \quad (2)$$

where $P_i := \prod_{j\neq i}(z_i - z_j)^{m_j} = \frac{p^{(m_i)}(z_i)}{m_i! p_n}$, and $M(x) := \max(1,|x|)$. Observe that our algorithm is adaptive in a very strong sense, namely, the above bound exclusively depends on the actual multiplicities and the geometry (i.e. the actual modulus of the roots and their distances to each other) of the roots.

Our algorithm can also be used to further refine the isolating disks to a size of $2^{-\kappa}$ or less, where κ is a given integer. The bit complexity for the refinement is given by the bound in (2) plus an additional term $\tilde{O}(n \cdot \kappa \cdot \max_i m_i)$. In particular for square-free polynomials, the amortized cost per root and bit of precision is one showing that the method is optimal up to polylogarithmic factors.

For the benchmark problem of isolating all roots of a polynomial p with *integer* coefficients of absolute value bounded by 2^τ, the bound in (2) becomes $\tilde{O}(n^3 + n^2\tau)$. The bound for the refinement becomes $\tilde{O}(n^3 + n^2\tau + n\kappa)$, even if there exist multiple roots.

For a square-free integer polynomial p, we are aware of only one method [8, Theorem 3.1] that achieves a comparable complexity bound for the benchmark problem. That is, based on the gap theorem from Mahler, one can compute a theoretical worst case bound b_0 of size $\Theta(n\tau)$ with the property that if n points $\hat{z}_j \in \mathbb{C}$ fulfill the inequality (1) for a $b \geq b_0$, then they approximate the corresponding roots z_j to an error less than $\sigma_j/(2n)$; cf. Lemma 4 for an adaptive version. Hence, for $b \geq b_0$, Pan's factorization algorithm also yields isolating disks for the roots of p using $\tilde{O}(n^2\tau)$ bit operations. Note that this approach achieves a good worst case complexity, however, for the price of running the factorization algorithm with $b = \Theta(n\tau)$, even if the roots are well conditioned. In contrast, our algorithm turns Pan's factorization algorithm into a highly adaptive method for isolating and approximating the roots of a general polynomial. Also, for general polynomials, there exist theoretical worst case bounds [19, Section 19] for the distance between the roots of p and corresponding approximations fulfilling (1). They are optimal for roots of multiplicity $\Omega(n)$ but they constitute strong overestimations if all roots have considerably smaller multiplicities. For the task of root refinement, the bit complexity of our method (i.e. $\tilde{O}(n \max_i m_i \cdot \kappa)$ for κ dominating) adapts to the highest occurring multiplicity, whereas this is not given for the currently best methods [12, 16, 18] which achieve the bound $\tilde{O}(n^2\kappa)$.

We would also like to remark that we are aware of only two previous root isolation algorithms [20, 14] that can cope with multiple roots. The latter algorithm can cope with at most one multiple root and needs to know the number of distinct complex roots as well as the number of distinct real roots. The former algorithm has the same applicability as our algorithm, but it has heuristic steps.

Our new root isolation algorithms has an interesting consequence on the complexity of computing the topology of a real planar algebraic curve specified as the zero set of an integer polynomial and for isolating the real solutions of a bivariate polynomial system. Both problems are well-studied [1, 10, 11, 20, 6, 5, 2, 7, 13]. More specifically, in an extended version [15] of this paper, we apply our root isolation method to a recent randomized algorithm [2] for computing the topology of a planar algebraic curve. This yields bounds on the *expected* number of bit operations which improve the currently best (which are both deterministic) bounds [7, 13] from $\tilde{O}(n^9\tau + n^8\tau^2)$ to $\tilde{O}(n^6 + n^5\tau)$ for topology computation and from $\tilde{O}(n^8 + n^7\tau)$ to $\tilde{O}(n^6 + n^5\tau)$ for solving bivariate systems, where n and τ are upper bounds on the degree and the bitsize of the input polynomials, respectively.

2. BASIC PROPERTIES

We consider a polynomial $p(x) = p_n x^n + \ldots + p_0 \in \mathbb{C}[x]$ of degree $n \geq 2$, where $p_n \neq 0$. In addition to our notations from the introduction, we fix the following definitions:

- τ_p denotes the minimal non-negative integer with $\frac{|p_i|}{|p_n|} \leq 2^{\tau_p}$ for all $i = 0,\ldots,n-1$,
- $\Gamma_p := M(\max_i \log|z_i|)$ the *logarithmic root bound* of p,
- $\mathrm{Mea}(p) := |p_n| \cdot \prod_{i=1}^{k} M(z_i)^{m_i}$ the *Mahler Measure* of p.

A straight forward argument shows that the quantities τ_p, Γ_p, $|p_n|$ and $\mathrm{Mea}(p)$ are closely related; see [15] for details.

LEMMA 1. *$\Gamma_p \leq 1 + \tau_p$ and $\tau_p - n - 1 \leq \log \frac{\mathrm{Mea}(p)}{|p_n|} \leq n\Gamma_p$.*

We assume the existence of an oracle which provides arbitrary good approximations of the polynomial p. Let $L \geq 1$ be an integer. We call a polynomial $\tilde{p} = \tilde{p}_n x^n + \ldots + \tilde{p}_0$, with $\tilde{p}_i = s_i \cdot 2^{-\ell}$ and s_i, $\ell \in \mathbb{Z}$, an *approximation of precision L* of p if $|\tilde{p}_i - p_i| \leq 2^{-L-\log(n+1)}\|p\|$, $\ell \leq L + \lceil\log(n+1)\rceil - \lfloor\log\|p\|\rfloor$, and $\log|s_i| \leq L + \lceil\log(n+1)\rceil + 1$ for all i. When considering p_i as infinite bitstring $p_i = \mathrm{sgn}(p_i) \cdot \sum_{k=-\infty}^{+\infty} b_k 2^k$, $b_k \in \{0,1\}$, then we can obtain $\tilde{p}_i$ from the partial string which starts at index $k_1 = \lfloor\log\|p\|\rfloor$ and ends at index $k_2 = \lfloor\log\|p\|\rfloor - L - \lceil\log(n+1)\rceil$, that is, $s_i := 2^l \cdot \mathrm{sgn}(p_i) \cdot \sum_{k=k_2}^{k_1} b_k 2^k$, and $l = L + \lceil\log(n+1)\rceil - \lfloor\log\|p\|\rfloor$. *We assume that we can ask for an approximation of precision L of p at cost $O(n(L+\log n)) = \tilde{O}(nL)$.* This is the cost of reading the approximation of precision L. The next Lemma summarizes some elementary properties of approximations of precision L. Again, we refer to the extended version [15] of this paper for its simple proof.

[1] This will turn out to be crucial to control the cost for the final verification step. For details, we refer to Sections 3.2 and 3.3.

[2] $\tilde{O}$ indicates that we omit logarithmic factors.

LEMMA 2. *If $\tilde{p}$ is an approximation of precision L of p, then*

- $\|\tilde{p}\|/2 \le \|p\| \le 2\|\tilde{p}\|$.
- *If $L \ge \tau_p + 4$, then $2^{-L-\log(n+1)}\|\tilde{p}\| \le |\tilde{p}_n|/4$.*
- *If $2^{-L-\log(n+1)}\|\tilde{p}\| \le |\tilde{p}_n|/4$, then $|\tilde{p}_n|/2 \le |p_n| \le 2|\tilde{p}_n|$.*

The above lemma suggests an efficient method for estimating p_n. We ask for approximations $\tilde{p}$ of precision L of p for $L = 1,2,4,\ldots$ until $2^{-L-\log(n+1)}\|\tilde{p}\| \le |\tilde{p}_n|/4$. Then, $|\tilde{p}_n|/2 \le |p_n| \le 2|\tilde{p}_n|$ by part 3 of the Lemma. Also $L \le 2(\tau_p+4)$ by part 2 of the above Lemma. The cost is $\tilde{O}(n\tau_p) = \tilde{O}(n^2\Gamma_p)$ bit operations, where we used the upper bound for τ_p from Lemma 1. Observe that this bound depends only on the geometry of the roots (i.e. the actual root bound Γ_p) and the degree but not (directly) on the size of the coefficients of p. We remark that a "good" integer approximation Γ of Γ_p can also be computed with $\tilde{O}(n^2\Gamma_p)$ bit operations. The proof (see [15, Theorem 1]) of the latter fact is almost identical to the one given in [17, Section 6.1], however, a small modification (essentially, we replaced linear search by exponential binary search) yields an improvement from $\tilde{O}(n^2\Gamma^2)$ to $\tilde{O}(n^2\Gamma)$.

THEOREM 1. *An integer $\Gamma \in \mathbb{N}$ with*

$$\Gamma_p \le \Gamma < 8\log n + \Gamma_p \tag{3}$$

can be computed with $\tilde{O}(n^2\Gamma_p)$ bit operations. The computation uses an approximation of precision L of p with $L = O(n\Gamma_p)$.

3. ALGORITHM

We present an algorithm for isolating the roots of a polynomial $p(x) = \sum_{i=0}^n p_i x^i = p_n \prod_{i=1}^k (x-z_i)^{m_i}$, where the coefficients p_i are given as described in the previous section. The algorithm uses an arbitrary polynomial factorization algorithm to produce approximations for the roots $z_1,\ldots,z_k$, and then performs a clustering and certification step to verify that the candidates are of sufficient quality. For concreteness, we pick Pan's factorization algorithm [16] for the factorization step, which also currently offers the best worst case bit complexity.[3] If the candidates do not pass the verification step, we reapply the factorization algorithm with a higher precision. For a given positive integer b denoting the desired precision, the factorization algorithm computes n root approximations $\hat{z}_1,\ldots,\hat{z}_n$. The quality of approximation and the bit complexity are as follows:

THEOREM 2. *For an arbitrary integer $b \ge n\log n$, complex numbers $\hat{z}_1,\ldots,\hat{z}_n$ can be computed such that*

$$\left\|p - p_n\prod_{i=1}^n (x-\hat{z}_i)\right\| \le 2^{-b}\|p\|$$

using $\tilde{O}(n^2\Gamma + bn)$ bit-operations. We write $\hat{p} := p_n\prod_{i=1}^n(x-\hat{z}_i)$. The algorithm returns the real and imaginary part of the $\hat{z}_i$'s as dyadic fractions of the form $A\cdot 2^{-B}$ with $A \in \mathbb{Z}$, $B \in \mathbb{N}$ and $B = O(b+n\Gamma_p)$. All fractions have the same denominator.

PROOF. If all roots of p have absolute value less than 1, then we can use Pan's Algorithm to obtain the above result; see [16, Theorem 2.1.1]. For general polynomials, we first scale p such that the roots of the scaled polynomial are contained in the unit disk $\Delta(0,1)$. For this purpose, we compute a Γ as in Theorem 1, and then consider the polynomial $f(x) := p(s\cdot x) = \sum_{i=0}^n f_i x^i$ with $s := 2^\Gamma$. Then, $f(x)$ has roots $\xi_i = z_i/s \in \Delta(0,1)$, and thus we can use Pan's algorithm with $b' := n\Gamma + b$ to compute an approximate factorization $\hat{f}(x) := \sum_{i=0}^n \hat{f}_i x^i := f_n\prod_{i=1}^n(x-\hat{\xi}_i)$ such that $\left\|f-\hat{f}\right\| < 2^{-b'}\|f\|$. According to Theorem 1, the cost for computing Γ is bounded by $\tilde{O}(n^2\Gamma)$ bit operations. The cost for running Pan's Algorithm is bounded by $\tilde{O}(n^2\Gamma)+\tilde{O}(nb') = \tilde{O}(n^2\Gamma+nb)$ bit operation. Furthermore, we need an approximation of precision b' of f, and thus an approximation of precision L of p with $L = O(ns+b) = \tilde{O}(n\Gamma_p + b)$. Again, the cost is bounded by $\tilde{O}(n^2\Gamma+nb)$.

Let $\hat{z}_i := s\cdot\hat{\xi}_i$ for all i and $\hat{p}(x) := p_n\cdot\prod_{i=1}^n(x-\hat{z}_i) = \hat{f}(x/s) = \sum_{i=0}^n \hat{f}_i/s^i x^i$. Then, $\hat{p}$ has the desired property. Namely, $\|\hat{p}-p\| \le \sum_{i=0}^n |f_i-\hat{f}_i| \le 2^{-b'}\sum_{i=0}^n |f_i| \le 2^{-b'}s^n\sum_{i=0}^n |f_i/s^i| = 2^{-b}\|p\|$. □

We now examine how far the approximations $\hat{z}_1,\ldots,\hat{z}_n$ can deviate from the actual roots for a given value of b. Let $\Delta(z,r)$ be the disk with center z and radius r and let $\mathrm{bd}\,\Delta(z,r)$ be its boundary. We further define $P_i := \prod_{j\ne i}(z_i-z_j)^{m_j}$. Then, $p^{(m_i)}(z_i) = m_i!\,p_n P_i$.

LEMMA 3. *If $r \le \frac{\sigma_i}{n}$, then $|p(x)| > r^{m_i}\cdot\frac{|p_nP_i|}{4}$ for all $x \in \Delta(z_i,r)$.*

PROOF. We have

$$\begin{aligned}
|p(x)| &= |p_n|\,|x-z_i|^{m_i}\prod_{j\ne i}|x-z_j|^{m_j}\\
&\ge |p_n|\,|x-z_i|^{m_i}\prod_{j\ne i}|z_i-z_j|^{m_j}\cdot(1-|x-z_i|/|z_i-z_j|)^{m_j}\\
&\ge r^{m_i}(1-1/n)^{n-m_i}\,|p_n|\prod_{j\ne i}|z_i-z_j|^{m_j} > r^{m_i}|p_nP_i|/4.
\end{aligned}$$

□

Based on the above Lemma, we can now use Rouché's theorem to show that, for sufficiently large b, the disk $\Delta(z_i,2^{-b/(2m_i)})$ contains exactly m_i root approximations.

LEMMA 4. *Let $\hat{p}$ be such that $\|p-\hat{p}\| \le 2^{-b}\|p\|$. If*

$$b \ge \max(8n, n\log(n)),\ \textit{and } b \textit{ is a power of two} \tag{4}$$

$$2^{-b/(2m_i)} \le \min(1/(2n^2), \sigma_i/(2n)),\ \textit{and} \tag{5}$$

$$2^{-b/2} \le \frac{|P_i|}{16(n+1)2^{\tau_p}M(z_i)^n} \tag{6}$$

for all i, the disk $\Delta(z_i,2^{-b/(2m_i)})$ contains exactly m_i root approximations. For $i \ne j$, let $\hat{z}_i$ and $\hat{z}_j$ be arbitrary approximations in the disks $\Delta(z_i,2^{-b/(2m_i)})$ and $\Delta(z_j,2^{-b/(2m_j)})$, respectively. Then,

$$(1-1/n)\,|z_i-z_j| \le |\hat{z}_i-\hat{z}_j| \le (1+1/n)\,|z_i-z_j|.$$

PROOF. Let $\delta_i := \left(16\cdot(n+1)\cdot 2^{-b}2^{\tau_p}|P_i|^{-1}M(z_i)^n\right)^{1/m_i}$. It is easy to verify that $\delta_i \le 2^{-b/(2m_i)} \le \min(1,\sigma_i)/(2n)$ from (6) and (5). Note that to show that $\Delta(z_i,\delta_i)$ contains m_i approximations, it suffices to show that $|(p-\hat{p})(x)| < |p(x)|$ for all x on the boundary of $\Delta(z_i,\delta_i)$. Then, Rouché's theorem guarantees that $\Delta(z_i,\delta_i)$ contains the same number of roots of p and $\hat{p}$ counted with multiplicity. Since z_i is of multiplicity m_i and $\delta_i < \sigma_i/n$, the disk contains exactly m_i roots of p counted with multiplicity. We have (note that $M(x) \le (1+1/(2n^2))\cdot M(z_i)$ for $x \in \mathrm{bd}\,\Delta(z_i,\delta_i)$)

$$\begin{aligned}
|(p-\hat{p})(x)| &\le \|p-\hat{p}\|\cdot M(x)^n < 2^{-b}\|p\|M(x)^n\\
&\le 2^{-b}\|p\|\cdot(1+1/(2n^2))^n\cdot M(z_i)^n\\
&\le 4\cdot 2^{-b}\cdot 2^{\tau_p}|p_n|\cdot(n+1)\cdot M(z_i)^n\\
&\le \delta_i^{m_i}|p_nP_i|/4 < |p(x)|,
\end{aligned}$$

where the inequality in line three follows from $\|p\| \le (n+1)|p_n|2^{\tau_p}$, the first one in line four follows from the definition of δ_i, and the

[3] In practice, one might consider a numerical root finder [4] based on the Aberth-Ehrlich method instead. There is empirical evidence that such methods achieve comparable complexity bounds.

last inequality follows from Lemma 3. It follows that $\Delta(z_i, 2^{-b/(2m_i)})$ contains exactly m_i approximations. Furthermore, since $\delta_i \le \sigma_i/(2n)$ for all i, the disks $\Delta(z_i,\delta_i)$, $1 \le i \le k$, are pairwise disjoint.

For the second claim, we observe that $|\hat{z}_\ell - z_\ell| \le 2^{-b/(2m_\ell)} \le \sigma_\ell/(2n) \le |z_i - z_j|/(2n)$ for $\ell = i, j$ and hence $|\hat{z}_i - z_i| + |\hat{z}_j - z_j| \le |z_i - z_j|/n$. The claim now follows from the triangle inequality. □

We have now established that the disks $\Delta(z_i, 2^{-b/(2m_i)})$, $1 \le i \le k$, are pairwise disjoint and that the i-th disk contains exactly m_i root approximations provided that b satisfies (4) to (6). We want to stress that the radii $2^{-b/(2m_i)}$, $1 \le i \le k$, are vastly different. Assume $b = 40$. For a one-fold root ($m = 1$), the radius is 2^{-20}, for a double root ($m = 2$) the radius is 2^{-10}, for a four-fold root ($m = 4$) the radius is 2^{-5}, and for a twenty-fold root ($m = 20$), the radius is as large as $1/2$; see Figure 1 for an illustration.

Unfortunately, the conditions on b are stated in terms of the unknown quantities m_i, σ_i and $|P_i|$, as well as the center z_i. In the remainder of the section, we will show how to cluster root approximations and to certify them. We will need the following more stringent properties for the clustering and certification step.

$$2^{-b/(2m_i)} < \min\left(\left(\frac{\sigma_i}{4n}\right)^8, \frac{\sigma_i}{1024n^2}\right) \tag{7}$$

$$2^{-b/8} < \min(1/16, |P_i|/((n+1)\cdot 2^{2n\Gamma_p + 8n})) \tag{8}$$

Let b_0 be the smallest integer satisfying (4) to (8) for all i. Then,

$$b_0 = O(n\log n + n\Gamma_p + \max_i(m_i \log M(\sigma_i^{-1})) + \max_i \log M(P_i^{-1})).$$

3.1 Overview of the Algorithm

On input p and the number k of distinct roots, the algorithm outputs isolating disks $\Delta_i = \Delta(\tilde{z}_i, R_i)$ for the roots of p as well as the corresponding multiplicities m_i. The radii satisfy $R_i < \sigma_i/(64n)$.

The algorithm uses the factorization step with an increasing precision until the result can be certified. If either the clustering step or the certification step fails, we simply double the precision. There are a couple of technical safeguards to ensure that we do not waste time on iterations with an insufficiently large precision (Steps 2, 5, and 6); also recall that we need to scale our initial polynomial.

1. Compute an integer bound Γ for Γ_p that fulfills inequality (3).
2. Compute a 2-approximation $\lambda = 2^{l_\lambda}$, $l_\lambda \in \mathbb{Z}$, of $\|p\|/|p_n|$.
3. Scale p, that is, $f(x) := p(s\cdot x)$, with $s := 2^\Gamma$, to ensure that the roots $\xi_i = z_i/S$, $i = 1,\ldots,k$, of f are contained in the unit disk. Let b be the smallest integer satisfying (4)
4. Run Pan's algorithm on input f with parameter $b' := b + n\Gamma$ to produce approximations $\hat{\xi}_1,\ldots,\hat{\xi}_n$ for the roots of f. Then, $\hat{z}_i := s\cdot\hat{\xi}_i$ are approximations of the roots of p, and $\|\hat{p} - p\| < 2^{-b}\|p\|$, where $\hat{p}(x) := p_n\prod_{i=1}^n (x - \hat{z}_i)$.
5. If there is a $\hat{z}_i$ with $\hat{z}_i \ge 2^{\Gamma+1}$, return to Step 4 with $b := 2b$.
6. If $\prod_{i=1}^n M(\hat{z}_i) > 8\lambda$, return to Step 4 with $b := 2b$.
7. Partition $\hat{z}_1,\ldots,\hat{z}_n$ into k clusters $C_1,\ldots,C_k$. Compute (well separated) enclosing disks $D_1,\ldots,D_k$ for the clusters. If the clustering fails to find k clusters and corresponding disks, return to Step 4 with $b := 2b$.
8. For each i, let Δ_i denote the disk with the same center as D_i but with an n-times larger radius. We now verify the existence of $|C_i|$ roots (counted with multiplicity) of p in Δ_i. If the verification fails, return to Step 4 with $b := 2b$.
9. If the verification succeeds, output the disks Δ_i and report the number $|C_i|$ of root approximations $\hat{z} \in \{\hat{z}_1,\ldots,\hat{z}_n\}$ contained in the disks as the corresponding multiplicities.

Note that Steps 5 and 6 ensure that $\log M(\hat{z}_i) = O(\Gamma_p + \log n)$ for all i, and $\log\prod_{i=1}^n M(\hat{z}_i) = O(\log(\|p\|/|p_n|)) = O(\log n + \tau_p) = \tilde{O}(n\Gamma_p)$. The following Lemma guarantees that the algorithm passes these steps if $b \ge b_0$.

LEMMA 5. *For any $b \ge b_0$, it holds that $|\hat{z}_i| < 2^{\Gamma+1}$ for all i, and $\prod_{i=1}^n M(\hat{z}_i) < 8\lambda$.*

PROOF. In the proof of Lemma 4, we have already shown that $|\hat{z}_i| \le (1 + 1/(2n^2))\cdot M(z_i)$ for all i. Hence, it follows that $|\hat{z}_i| \le (1+1/(2n^2))\cdot 2^{\Gamma_p} < 2\cdot 2^{\Gamma_p} \le 2^{\Gamma_p+1}$, and (note that $(1+\frac{1}{2n^2})^n < 4$)

$$\prod_{i=1}^n M(\hat{z}_i) \le 4\cdot\prod_{i=1}^k M(z_i)^{m_i} < \frac{4\,\mathrm{Mea}(p)}{|p_n|} \le \frac{4\|p\|_2}{|p_n|} \le \frac{4\|p\|}{|p_n|} < 8\lambda.$$

□

3.2 Clustering

After candidate approximations $\hat{z}_1,\ldots,\hat{z}_n$ are computed using a fixed precision parameter b, we perform a partitioning of these approximations into k clusters $C_1,\ldots,C_k$. Our clustering algorithm works in phases. In each phase, it attempts to form a cluster based on some unclustered approximation as seed. After all approximations have been clustered, ideally, each of the clusters now corresponds to a distinct root of p. The algorithm satisfies the following properties: (1) For $b < b_0$, the algorithm may or may not succeed in finding k clusters. (2) For $b \ge b_0$, the clustering always succeeds. Whenever the clustering succeeds, the cluster C_i with seed $\tilde{z}_i$ is contained in the disk $D_i := \Delta(\tilde{z}_i, r_i)$, where $r_i \approx \min(\frac{1}{n^2}, \frac{\tilde{\sigma}_i}{256n^2})$, and $\tilde{\sigma}_i = \min_{j\ne i}|\tilde{z}_i - \tilde{z}_j|$. Furthermore, for $b \ge b_0$, D_i contains the root z_i (under suitable numbering) and exactly m_i approximations.

Before we describe our clustering method, we discuss two evident approaches that do not work for any b of size comparable to b_0 or smaller. A clustering with a fixed grid does not work as root approximations coming from roots with different multiplicities may move by vastly distinct amounts. As a consequence, we can only succeed if $b > (\max_i m_i)\cdot\log(\min_i \sigma_i)^{-1}$ which can be considerably larger than b_0, see Figure 1. A clustering based on Gershgorin disks does not work as well because very good approximations of a multiple root lead to large disks which then fail to separate approximations of distinct roots. In particular, if approximations are identical, the corresponding Gershgorin disks have infinite radius.

For our clustering, we use the fact that the factorization algorithm provides approximations $\hat{z}$ of the root z_i with distance less than $2^{-b/(2m_i)}$ (for $b \ge b_0$). Thus, we aim to determine clusters C of maximal size such that the pairwise distance between two elements in the same cluster is less than $2\cdot 2^{-b/(2|C|)}$. We give details.

1. Initialize $\mathscr{C}$ to the empty set (of clusters).
2. Initialize C to the set of all unclustered approximations and choose $\hat{z} \in C$ arbitrarily. Let $a := 2^{\lfloor\log n\rfloor+2}$ and $\delta := 2^{-b/4}$.
3. Update C to the set of points $q \in C$ satisfying $|\hat{z} - q| \le 2\sqrt[a/2]{\delta}$.
4. If $|C| \ge a/2$, add C to $\mathscr{C}$. Otherwise, set $a := a/2$ and continue with step 3.
5. If there are still unclustered approximations, continue with step 2.
6. If the number of clusters in $\mathscr{C}$ is different from k, report failure, and return to the factorization step with $b := 2b$.

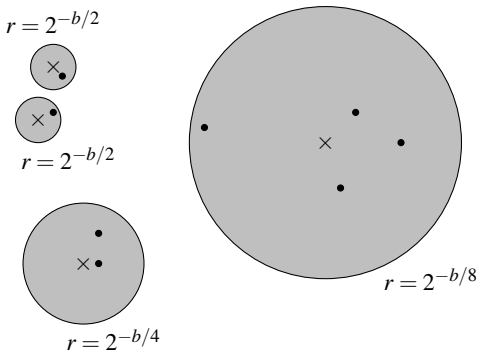

Figure 1: Example of a polynomial with four distinct roots with multiplicities 1, 1, 2, and 4. Crosses are roots of the polynomial, dots represent the approximations. The disk around a root shows the potential locations of its approximations. Note that the straightforward approach to cluster with a fixed distance threshold fails for all b with $b < (\max_i m_i) \cdot \log(\min_i \sigma_i)^{-1}$: For each such b, one can not choose any threshold that allows detecting the simple roots without splitting the four-fold root.

Note that, for $b \geq b_0$, the disks $\Delta(z_i, 2^{-b/(2m_i)})$ are disjoint. Let Z_i denote the set of root approximations in $\Delta(z_i, 2^{-b/(2m_i)})$. Then, $|Z_i| = m_i$ according to Lemma 4. We show that, for $b \geq b_0$, the clustering algorithm terminates with $C = Z_i$ if called with an approximation $\hat{z} \in Z_i$.

LEMMA 6. *Assume $b \geq b_0$, $\hat{z}_i \in Z_i$, $\hat{z}_j \in Z_j$, and $i \neq j$. Then,*

$$\left|\hat{z}_i - \hat{z}_j\right| \geq 2 \cdot (2^{-b/(16m_i)} + 2^{-b/(16m_j)}).$$

PROOF. Since $b \geq b_0$, we have $2^{-b/(2m_\ell)} \leq \sigma_\ell$ for $\ell = i, j$ by (5) and $2^{-b/(16m_\ell)} = (2^{-b/(2m_\ell)})^{1/8} \leq \sigma_\ell/(4n) \leq \sigma_\ell/8$ by (7). Thus,

$$\begin{aligned}\left|\hat{z}_i - \hat{z}_j\right| &\geq \max(\sigma_i, \sigma_j) - 2^{-\frac{b}{2m_i}} - 2^{-\frac{b}{2m_j}} \geq \frac{\sigma_i}{2} + \frac{\sigma_j}{2} - \frac{\sigma_i}{4} - \frac{\sigma_j}{4} \\ &\geq 2 \cdot (2^{-b/(16m_i)} + 2^{-b/(16m_j)}).\end{aligned}$$

□

LEMMA 7. *If $b \geq b_0$, the clustering algorithm produces clusters C_1 to C_k such that $C_i = Z_i$ for all i (under suitable numbering). Let $\tilde{z}_i$ be the seed of C_i and let $\tilde{\sigma}_i = \min_{j \neq i} \left|\tilde{z}_i - \tilde{z}_j\right|$. Then, $(1 - 1/n)\sigma_i \leq \tilde{\sigma}_i \leq (1 + 1/n)\sigma_i$ and C_i as well as the root z_i is contained in $\Delta(\tilde{z}_i, \min(\frac{1}{n^2}, \frac{\tilde{\sigma}_i}{256n^2}))$.*

PROOF. Assume that the algorithm has already produced Z_1 to Z_{i-1} and is now run with a seed $\hat{z} \in Z_i$. We prove that it terminates with $C = Z_i$. Let ℓ be a power of two such that $\ell \leq m_i < 2\ell$. The proof consists of two parts. We first assume that steps 2 and 3 are executed for $a = 2\ell$. For this case, we show that the algorithm will terminate with $C = Z_i$. In the second part of the proof, we show that the algorithm does not terminate as long as $a > 2\ell$.

Assume the algorithm reaches steps 2 and 3 with $a/2 = \ell$, i.e. $a/2 \leq m_i < a$. For any approximation $q \in Z_i$, we have $|\hat{z} - q| \leq 2 \cdot 2^{-b/(2m_i)} = 2\sqrt[m_i/2]{\delta} \leq 2\sqrt[a/2]{\delta}$. Thus, $Z_i \subseteq C$. Conversely, consider any approximation $q \notin Z_i$. Then, $|\hat{z} - q| \geq 2 \cdot 2^{-b/(16m_i)} > 2\sqrt[4m_i]{\delta} \geq 2\sqrt[2a]{\delta}$, and thus no such approximation is contained in C. This shows that $C = Z_i$. Since $|C| \geq a/2$, the algorithm terminates and returns Z_i.

It is left to argue that the algorithm does not terminate before $a/2 = \ell$. Since ℓ and a are powers of two, assume we terminate with $a/2 \geq 2\ell$, and let C be the cluster returned. Then, $m_i < a/2 \leq |C| < a$ and Z_i is a proper subset of C. Consider any approximation $q \in C \setminus Z_i$, say $q \in Z_j$ with $j \neq i$. Since $q \notin Z_i$, we have $|q - \hat{z}| \geq 2 \cdot (2^{-b/(16m_i)} + 2^{-b/(16m_j)}) > 2 \cdot 2^{-b/(16m_i)} > 2\sqrt[4m_i]{\delta}$. And since $q \in C$, we have $|q - \hat{z}| \leq 2\sqrt[a/2]{\delta}$. Thus, $4m_i \leq a/2$ and, hence, there are at least $3a/8$ many approximations in $C \setminus Z_i$. Furthermore, $\left|z_i - z_j\right| \leq |z_i - \hat{z}| + |\hat{z} - q| + \left|q - z_j\right| \leq 2^{-b/(2m_i)} + 2\sqrt[a/2]{\delta} + 2^{-b/(2m_j)} \leq 2^{-b/(16m_i)} + 2\sqrt[a/2]{\delta} + 2^{-b/(16m_j)} \leq 3\sqrt[a/2]{\delta}$. Consequently, there are at least $3a/8$ roots $z_j \neq z_i$ counted with multiplicity within distance $3\sqrt[a/2]{\delta}$ to z_i. This observation allows us to upper bound the value of $|P_i|$, namely

$$\begin{aligned}|P_i| = \prod_{j \neq i} |z_i - z_j|^{m_j} &\leq (3\sqrt[a/2]{\delta})^{3a/8} 2^{(n - m_i - 3a/8)\Gamma_p} \\ &< 3^n \delta^{3/4} 2^{n\Gamma_p} \leq 3^n 2^{-3b/16} 2^{n\Gamma_p} < 3^n 2^{-b/8} \cdot 2^{n\Gamma_p},\end{aligned}$$

a contradiction to (8).

We now come to the claims about $\tilde{\sigma}_i$ and the disks defined in terms of it. The relation between σ_i and $\tilde{\sigma}_i$ follows from the second part of Lemma 4. All points in $C_i = Z_i$ have distance at most $2 \cdot 2^{-b/(2m_i)}$ from $\tilde{z}_i$. Also, by (5) and (7),

$$2 \cdot 2^{-b/(2m_i)} < \min(1/n^2, \sigma_i/(512n^2)) \leq \min(1/n^2, \tilde{\sigma}_i/(256n^2))$$

Hence, C_i as well as z_i is contained in $\Delta(\tilde{z}_i, \min(1/n^2, \tilde{\sigma}_i/(256n^2)))$. □

LEMMA 8. *For a fixed precision b, computing a complete clustering needs $\tilde{O}(nb + n^2\Gamma_p)$ bit operations.*

PROOF. For each approximation, we examine the number of distance computations we need to perform. Recall that b (property (4)) and a are powers of two, $a \leq 4n$ by definition, and $b \geq 8n \geq 2a$ by property (4). Then, $\sqrt[a/2]{\delta} = 2^{-b/(2a)} \in 2^{-\mathbb{N}}$. Thus, the number $\sqrt[a/2]{\delta}$ has a very simple format in binary notation. There is a single one, and this one is $b/(2a)$ positions after the binary point. In addition, all approximations $\hat{z}$ have absolute value less than $2 \cdot 2^{\Gamma}$ due to Step 5 in the overall algorithm. Thus, each evaluation of the form $|\hat{z} - q| \leq 2\sqrt[a/2]{\delta}$ can be done with

$$O(\Gamma + \log \delta^{-2/a}) = O((b/a) + \Gamma) = O((b/a) + \Gamma_p + \log n)$$

bit operations.

For a fixed seed $\hat{z}$, in the i-th iteration of step 2, we have at most $a \leq n/2^{i-2}$ many unclustered approximations left in C, since otherwise we would have terminated in an earlier iteration. Hence, we perform at most a evaluations of the form $|\hat{z} - q| \leq 2\sqrt[a/2]{\delta}$, resulting in an overall number of bit operations of $a \cdot O((b/a) + \Gamma) = O(b + a\Gamma)$ for a fixed iteration. As we halve a in each iteration, we have at most $\log n + 2$ iterations for a fixed $\hat{z}$, leading to a bit complexity of $O(b \log n + n\Gamma) = \tilde{O}(b + n\Gamma) = \tilde{O}(b + n\Gamma_p)$.

In total, performing a complete clustering has a bit complexity of at most $\tilde{O}(nb + n^2\Gamma_p)$. □

When the clustering succeeds, we have k clusters C_1 to C_k and corresponding seeds $\tilde{z}_1, \ldots, \tilde{z}_k \subseteq \{\hat{z}_1, \ldots, \hat{z}_n\}$. For $i = 1, \ldots, k$, we define $D_i := \Delta(\tilde{z}_i, r_i)$, where $\tilde{z}_i$ is the seed for the cluster C_i and

$$r_i := \min(2^{-\lceil 2 \log n \rceil}, 2^{\lceil \log \tilde{\sigma}_i/(256n^2) \rceil}) \geq \min(\frac{1}{2n^2}, \frac{\tilde{\sigma}_i}{256n^2}). \quad (9)$$

In particular, r_i is a 2-approximation of $\min(1/n^2, \tilde{\sigma}_i/(256n^2))$. Notice that the cost for computing the separations $\tilde{\sigma}_i$ is bounded by $\tilde{O}(nb + n^2\Gamma_p)$ bit operations since we can compute the nearest neighbor graph of the points $\tilde{z}_i$ (and thus the values $\tilde{\sigma}_i$) in $O(n \log n)$ steps [9] with a precision of $O(b + n\Gamma)$.

Now, suppose that $b \geq b_0$, Then, according to Lemma 7, the cluster C_i is contained in the disk D_i. Furthermore, D_i contains exactly

one root z_i of p (under suitable numbering of the roots), and it holds that $m_i = \mathrm{mult}(z_i, p) = |C_i|$ and $\min(1/(2n^2), \sigma_i/(512n^2)) \le r_i \le \min(1/n^2, \sigma_i/(64n^2))$. Hence, before we proceed, we verify that each disk D_i actually contains the cluster C_i. If this is not the case, we report a failure, and return to the factorization with $b := 2b$.

In the next step, we aim to show that each of the enlarged disks $\Delta_i := \Delta(\tilde{z}_i, R_i) := \Delta(\tilde{z}_i, nr_i)$, $i = 1, \ldots, k$, contains exactly one root z_i of p, and that the number of elements in $C_i \subseteq \Delta_i$ equals the multiplicity of z_i. Notice that, from the definition of r_i and Δ_i, it obvious that the disks Δ_i are pairwise disjoint and that $C_i \subseteq D_i \subseteq \Delta_i$.

3.3 Certification

In order to show that Δ_i contains exactly one root of p with multiplicity $|C_i|$, we show that each Δ_i contains the same number of roots of p and $\hat{p}$ counted with multiplicity. For the latter, we compute a lower bound for $|\hat{p}(z)|$ on the boundary $\mathrm{bd}\,\Delta_i$ of Δ_i, and check whether this bound is larger than $|(\hat{p}-p)(z)|$ for all points $z \in \mathrm{bd}\,\Delta_i$. If this is the case, then we are done according to Rouché's theorem. Otherwise, we start over the factorization algorithm with $b = 2b$. We now come to the details:

1. Let $\lambda = 2^{l_\lambda}$ be the 2-approximation of $\|p\|/|p_n|$ as defined in step 2 of the overall algorithm.
2. For $i = 1, \ldots, k$, let $z_i^* := \tilde{z}_i + n \cdot r_i \in \Delta_i$. Note that $|z_i^*| \le (1 + 1/n) \cdot M(\tilde{z}_i)$ since $nr_i \le 1/n$.
3. We try to establish the inequality
$$|\hat{p}(z_i^*)/p_n| > E_i := 32 \cdot 2^{-b} \lambda M(\tilde{z}_i)^n \qquad (10)$$
for all i. If this check is satisfied, we know that each disk Δ_i contains exactly one root z_i of p and that its multiplicity equals the number $|C_i|$ of approximations in Δ_i (Lemma 10). In order to establish the inequality, we consider $\rho = 1, 2, 4, \ldots$ and compute $\left|\hat{p}(z_i^*)/p_n\right|$ to an absolute error less than $2^{-\rho}$. If, for all $\rho \le b$, we fail to show that $\left|p(z_i^*)/p_n\right| > E_i$, we again return to the factorization step with $b := 2b$. Otherwise, let ρ_i be the smallest ρ for which we are successful.
4. If, at any stage of the algorithm, $\sum_i \rho_i > b$, we also report a failure and go back to the factorization step with $b := 2b$.
5. If we can verify that $|\hat{p}(\tilde{z}_i + nr_i)/p_n| > E_i$ for all i, we return the disks Δ_i and the multiplicities $m_i = |C_i|$.

LEMMA 9. *For any i, we can compute $|\hat{p}(z_i^*)/p_n|$ to an absolute error less than $2^{-\rho}$ with a number of bit operations less than*
$$\tilde{O}(n(n + \rho + n \log M(\tilde{z}_i) + \tau_p)).$$
For a fixed b, the total cost for all evaluations in the above certification step is bounded by $\tilde{O}(nb + n^2\tau_p + n^3)$.

PROOF. Consider an arbitrary subset $S \subseteq \{\hat{z}_1, \ldots, \hat{z}_n\}$. We first derive an upper bound for $\prod_{\hat{z} \in S} |z_i^* - \hat{z}|$. For that, consider the polynomial $\hat{p}_S(x) := \prod_{\hat{z} \in S}(x - \hat{z})$. The i-th coefficient of $\hat{p}_S$ is bounded by $\binom{|S|}{i} \cdot \prod_{\hat{z} \in S} M(\hat{z}) \le 2^n \prod_{i=1}^n M(\hat{z}_i) \le 8\lambda \cdot 2^n$ due to Step 6 in the overall algorithm. It follows that
$$\prod_{\hat{z} \in S} |z_i^* - \hat{z}| = |\hat{p}_S(z_i^*)| \le (n+1) M(z_i^*)^n \cdot 8\lambda \cdot 2^n < 2^{4n + \tau_p + 6} M(\tilde{z}_i)^n,$$
where we used that $M(z_i^*) < (1 + 1/n) \cdot M(\tilde{z}_i)$ and $\lambda < 2\|p\|/|p_n| < 2^{\tau_p + 1}(n+1)$. In order to evaluate $|\hat{p}(z_i^*)/p_n| = \prod_{j=1}^n |z_i^* - \hat{z}_j|$, we use approximate interval evaluation with an absolute precision $K = 1, 2, 4, \ldots$. More precisely, we compute the distance of z_i^* to each of the points $\hat{z}_j$, $j = 1, \ldots, n$, up to an absolute error of 2^{-K}, and then take the product over all distances using a fixed point precision of K bits after the binary point.[4] We stop when the resulting interval has size less than $2^{-\rho}$. The above consideration shows that all intermediate results have at most $O(n + \tau_p + n \log M(\tilde{z}_i))$ bits before the binary point. Thus, we eventually succeed for a $K = O(\rho + \tau_p + n + n \log M(\tilde{z}_i))$. Since we perform n subtractions and n multiplications, the cost is bounded by $\tilde{O}(nK)$ bit operations for each K. Hence, the bound for evaluating $|\hat{p}(z_i^*)/p_n|$ follows.

We come to the second claim. Since we double ρ in each iteration and consider at most $\log b$ iterations, the cost for the evaluation of $|\hat{p}(z_i^*)/p_n|$ are bounded by $\tilde{O}(n(n + \rho_i + n \log M(\tilde{z}_i) + \tau_p))$. Since we ensure that $\sum_i \rho_i \le b$, it follows that the total cost is bounded by $\tilde{O}(nb + n^2\tau_p + n^3 + n^2 \log(\prod_{i=1}^k M(\tilde{z}_i)))$. The last summand is smaller than $n^2 \cdot 8\lambda$ according to Step 6 in the overall algorithm, and $\lambda < 2\|p\|/|p_n| < 2(n+1)2^{\tau_p}$. This shows the claim. □

We now show that inequality (10) implies that the disk Δ_i contains the same number of roots of the polynomials $\hat{p}$ and p.

LEMMA 10. *1. If inequality (10) holds for all i, then Δ_i isolates a root of z_i of p of multiplicity $m_i = \mathrm{mult}(z_i, p) = |C_i|$.*

2. If $b \ge b_0$, then
$$\frac{|\hat{p}(z_i^*)|}{|p_n|} > \left(\frac{\min(256, \sigma_i)}{1024n}\right)^{m_i} \cdot \frac{|P_i|}{8} \ge 64 \cdot 2^{-b} \lambda M(\tilde{z}_i)^n$$

PROOF. We first show that $|\hat{p}(x)| \ge \frac{1}{4}|\hat{p}(z_i^*)|$ for all $x \in \mathrm{bd}\,\Delta_i$. We fix an approximation $\hat{z}$ in some disk D_j. Suppose that x is the farthest point on $\mathrm{bd}\,\Delta_i$ from $\hat{z}$, and y the nearest. Then, for $i \ne j$, we have $|x - \hat{z}| \le |x - \tilde{z}_i| + |\tilde{z}_i - \tilde{z}_j| + |\tilde{z}_j - \hat{z}| \le (1 + 1/n)|\tilde{z}_i - \tilde{z}_j|$, and $|y - \hat{z}| \ge |\tilde{z}_i - \tilde{z}_j| - |y - \tilde{z}_i| - |\tilde{z}_j - \hat{z}| \ge (1 - 1/n)|\tilde{z}_i - \tilde{z}_j|$. Similarly, for $i = j$, it holds that $|x - \hat{z}| \le |x - \tilde{z}_i| + |\tilde{z}_i - \hat{z}| \le (1 + 1/n) nr_i$ and $|y - \hat{z}| \ge |y - \tilde{z}_i| - |\tilde{z}_i - \hat{z}| \ge (1 - 1/n) nr_i$. Hence, for arbitrary points $x, y \in \mathrm{bd}\,\Delta_i$ and an arbitrary approximation $\hat{z}$, it follows that
$$(1 - 1/n)|y - \hat{z}| \le |x - \hat{z}| \le (1 + 1/n)|y - \hat{z}| \,.$$
We conclude that $(1 - 1/n)^n |\hat{p}(z_i^*)| \le |\hat{p}(x)| \le (1 + 1/n)^n |\hat{p}(z_i^*)|$ for all $x \in \mathrm{bd}\,\Delta_i$. This shows the above claim.

We can now prove Part 1 of the lemma. We have $|x| < (1 + 1/n) M(\tilde{z}_i)$ for all $x \in \mathrm{bd}\,\Delta_i$ since $nr_i < 1/n$,. Now, if $|\hat{p}(z_i^*)|/|p_n| > 32 \cdot 2^{-b} \lambda M(\tilde{z}_i)^n$, then
$$\begin{aligned} |\hat{p}(x)| &> |\hat{p}(z_i^*)|/4 > 8|p_n| \lambda 2^{-b} (1 - 1/n)^n M(x)^n \\ &> \|p\| 2^{-b} M(x)^n \ge \|\hat{p} - p\| M(x)^n \ge |\hat{p}(x) - p(x)|. \end{aligned}$$
Hence, according to Rouché's theorem. Δ_i contains the same number (namely, $|C_i|$) of roots of p and $\hat{p}$. If this holds for all disks Δ_i, then each of the disks must contain exactly one root since p has k distinct roots. In addition, the multiplicity of each root equals the number $|C_i|$ of approximations within Δ_i.

It remains to show the third claim. Since $b \ge b_0$, it follows that $\min(1/(2n^2), \sigma_i/(512n^2)) \le r_i \le \min(1/n^2, \sigma_i/(64n^2))$ and $|\tilde{z}_i - z_i| < r_i$; cf. the remark following the definition of r_i in (9). Thus,
$$\begin{aligned} |\hat{p}(z_i^*)| &\ge |p(z_i^*)| - 2^{-b} \|p\| \cdot M(z_i^*)^n \\ &= |p(z_i + (\tilde{z}_i - z_i + nr_i))| - 2^{-b} \|p\| \cdot M(z_i^*)^n \\ &\ge ((n-1) r_i)^{m_i} |p_n P_i|/4 - 4 \cdot 2^{-b} \|p\| M(z_i)^n \\ &\ge \left(\frac{(n-1)\min(256, \sigma_i)}{512n^2}\right)^{m_i} |p_n P_i|/4 - 4 \cdot 2^{-b} \|p\| M(z_i)^n \\ &\ge \left(\frac{\min(256, \sigma_i)}{1024n}\right)^{m_i} |p_n P_i|/4 - 4 \cdot 2^{-b} \|p\| M(z_i)^n, \end{aligned}$$

[4] In fact, we compute an interval I_j of size less than 2^{-K} such that $|z_i^* - \hat{z}_j| \in I_j$, and then consider the product $\prod_j I_j$.

where the first inequality is due to $|(p-\hat{p})(x)| < 2^{-b}\|p\| \cdot M(x)^n$, the second inequality follows from $|\tilde{z}_i - z_i + nr_i| \le (n+1)r_i \le \sigma_i/n$, Lemma 3 and $M(z_i^*) < (1+1/n)\cdot M(z_i)$, and the third inequality follows from $r_i \ge \min(\frac{1}{2n^2}, \frac{\sigma_i}{512n^2})$. In addition, we have

$$2^{-b}\|p\| M(z_i)^n \le \left(\frac{\min(256,\sigma_i)}{1024n}\right)^{m_i} \cdot \frac{|p_n P_i|}{4096}, \qquad (11)$$

since

$$\begin{aligned} &2^{-b}\|p\| M(z_i)^n \\ &\quad \le 2^{-b/8}\cdot 2^{-b/2}\cdot 2^{\tau_p}|p_n|\cdot(n+1)\cdot M(z_i)^n \\ &\quad \le \frac{|P_i|}{(n+1)2^{2n\Gamma_p+8n}}\left(\frac{\min(256,\sigma_i)}{1024n}\right)^{m_i} 2^{\tau_p}|p_n|(n+1)M(z_i)^n \\ &\quad \le \left(\frac{\min(256,\sigma_i)}{1024n}\right)^{m_i}\cdot\frac{|p_n P_i|}{2^{7n-1}} \le \left(\frac{\min(256,\sigma_i)}{1024n}\right)^{m_i}\cdot\frac{|p_n P_i|}{4096}, \end{aligned}$$

where the second inequality follows from (8), (7), and (5) [5], and the third inequality follows from $\tau_p \le n\Gamma_p + n + 1$ (Lemma 1) and $M(z_i)^n \le 2^{n\Gamma_p}$. Finally,

$$\begin{aligned} \frac{|\hat{p}(z_i^*)|}{|p_n|} &> \left(\frac{\min(256,\sigma_i)}{1024n}\right)^{m_i}\cdot\frac{|P_i|}{8} \ge 512\cdot 2^{-b}\frac{\|p\|}{|p_n|}M(z_i)^n \\ &\ge 64\cdot 2^{-b}\lambda M(\hat{z}_i)^n, \end{aligned}$$

where the first and the second inequality follow from (11) and the third inequality holds since λ is a 2-approximation of $\|p\|/|p_n|$ and $|z_i|^n \le (1+1/n)^n|\tilde{z}_i|^n \le 4|\tilde{z}_i|^n$. □

LEMMA 11. *There exists a $b^* \ge b_0$ bounded by*

$$O\left(n\log n + n\Gamma_p + \sum_{i=1}^k\left(\log M(P_i^{-1}) + m_i \log M(\sigma_i^{-1})\right)\right)$$

such that the certification step succeeds for any $b > b^*$. The total cost in the certification algorithm (i.e. for all iterations until we eventually succeed) is bounded by

$$\tilde{O}\left(n^3 + n^2\tau_p + n\cdot\sum_{i=1}^k\left(\log M(P_i^{-1}) + m_i\log M(\sigma_i^{-1})\right)\right)$$

bit operations.

PROOF. Due to Lemma 10, $|\hat{p}(z_i^*)/p_n| > \left(\frac{\min(256,\sigma_i)}{1024n}\right)^{m_i}\cdot\frac{|P_i|}{8} > 64\cdot 2^{-b_0}\lambda M(\hat{z}_i)^n$. Thus, in order to verify inequality (10), it suffices to evaluate $|\hat{p}(z_i^*)/p_n|$ to an error of less than $|\hat{p}(z_i^*)/2p_n|$. It follows that we succeed for some ρ_i with

$$\rho_i = O(m_i\log n + m_i\max(1,\log\sigma_i^{-1}) + \log\max(1,|P_i|^{-1})).$$

In Step 3 of the certification algorithm, we require that the sum over all ρ_i does not exceed b. Hence, we eventually succeed in verifying the inequality (10) for all i if b is larger than some b^* with

$$\begin{aligned} b^* &= O(b_0 + \textstyle\sum_i m_i\log n + \sum_i(\log M(P_i^{-1}) + m_i\log M(\sigma_i^{-1}))) \\ &= O(n\log n + n\Gamma_p + \textstyle\sum_i(\log M(P_i^{-1}) + m_i\log M(\sigma_i^{-1}))). \end{aligned}$$

For the bound for the overall cost, we remark that, for each b, the certification algorithm needs $\tilde{O}(n^3 + nb + n^2\tau_p)$ bit operations due to Lemma 9. Thus, the above bound follows from the fact that that we double b in each step and that the certification algorithm succeeds under guarantee for all $b > b^*$. □

[5] Observe $2^{-b/(2m_i)} \le \min(\frac{1}{2n^2}, \frac{\sigma_i}{1024n^2}) \le \frac{\min(256,\sigma_i)}{1024n}$.

4. COMPLEXITY ANALYSIS

We now turn to the complexity analysis of the root isolation algorithm. In the first step, we provide a bound for general polynomials p with complex coefficients. In the second step, we give a simplified bound for the special case, where p has integer coefficients. We also give bounds for the number of bit operations that is needed to refine the isolating disks to a size less than $2^{-\kappa}$, with κ an arbitrary positive integer.

THEOREM 3. *Let $p(x) \in \mathbb{C}[x]$ be a polynomial as defined in Section 2. We assume that the number k of distinct roots of p is given. Then, for all $i = 1,\ldots,k$, the algorithm from Section 3 returns an isolating disk $\Delta(\tilde{z}_i, R_i)$ for the root z_i and the corresponding multiplicity m_i, and it holds that $R_i < \frac{\sigma_i}{64n}$.*

For that, it uses a number of bit operations bounded by

$$\tilde{O}\left(n^3 + n^2\tau_p + n\cdot\sum_{i=1}^k\left(\log M(P_i^{-1}) + m_i\log M(\sigma_i^{-1})\right)\right) \qquad (12)$$

The algorithm needs an approximation of precision L of p, with

$$L = O\left(n\Gamma_p + \sum_{i=1}^k\left(\log M(P_i^{-1}) + m_i\log M(\sigma_i^{-1})\right)\right). \qquad (13)$$

PROOF. For a fixed b, let us consider the cost for each of the steps in the algorithm: Steps 1-3, 5 and 6 do not use more than $\tilde{O}(n^2\Gamma_p + nb)$ bit operations (Theorem 1 and Lemma 2). The Steps 4 and 7 do not use more than $\tilde{O}(n^2\Gamma_p + nb)$ bit operations (Corollary 2 and Lemma 8), and the Steps 8 and 9 use a number of bit operations bounded by (12) (Lemma 11).

Furthermore, the oracle must provide an approximation of precision $O(n\Gamma_p + b)$ of p in order to compute the bound Γ for Γ_p, to compute the 2-approximation λ of $\|p\|/|p_n|$, and to run Pan's algorithm. The algorithm succeeds in computing isolating disks if $b > b^*$ with a b^* as in Lemma 11. Since we double b in each step, we need at most $\log b^*$ iterations and the total cost for each iteration is bounded by (12). This shows the complexity result.

It remains to prove the bound for R_i. When the clustering succeeds, it returns disks $D_i = \Delta(\tilde{z}_i, r_i)$ with $\min(\frac{1}{2n^2}, \frac{\tilde{\sigma}_i}{256n^2}) \le r_i \le \min(\frac{1}{n^2}, \frac{\tilde{\sigma}_i}{128n^2})$ for all $i = 1,\ldots,m$. It follows that $R_i = n\cdot r_i \le \frac{\tilde{\sigma}_i}{128n}$, and thus $|z_i - z_j| \ge |\tilde{z}_i - \tilde{z}_j| - |z_i - \tilde{z}_i| - |z_j - \tilde{z}_j| > |\tilde{z}_i - \tilde{z}_j|\cdot(1 - 1/(64n)) > |\tilde{z}_i - \tilde{z}_j|/2$ for all i, j with $i \ne j$. We conclude that $\sigma_i > \tilde{\sigma}_i/2 \ge 64nR_i$. □

We remark that the bound (12) can be reformulated in terms of values that exclusively depend on the degree n and the geometry of the roots (i.e. their absolute values and their separations). Namely, due to Lemma 1, $\tau_p \le n + 1 + \log\frac{\mathrm{Mea}(p)}{|p_n|}$, and the latter expression only involves the degree and the absolute values of the roots of p. This yields the bound (2) that we stated in the introduction.

Next, we show that combining our algorithm with Pan's factorization algorithm also yields a very efficient method to further refine the isolating disks.

THEOREM 4. *Let $p(x)$ be a polynomial as in Theorem 3, and κ be a given positive integer. We can compute isolating disks $\Delta_i(\tilde{z}_i, R_i)$ with radius $R_i < 2^{-\kappa}$ in a number of bit operations bounded by*

$$\mathscr{B} + \tilde{O}(n\kappa\cdot\max_{1\le i\le k} m_i), \qquad (14)$$

where $\mathscr{B}$ is bounded by (12). For that, we need an approximation of precision L of p with $L = \mathscr{L} + \tilde{O}(n\kappa\cdot\max_{1\le i\le k} m_i)$, where $\mathscr{L}$ is bounded by (13).

PROOF. As a first step, we use the algorithm from Section 3 to compute isolating disks $\Delta_i = \Delta(\tilde{z}_i, R_i)$ with $R_i \le \sigma_i/(64n)$. Each

disk Δ_i contains the root z_i, $m_i = \mathrm{mult}(z_i, p)$ approximations $\hat{z} \in \{\hat{z}_1, \ldots, \hat{z}_n\}$ of z_i, and it holds that $\sigma_i/2 < \tilde{\sigma}_i < 2\sigma_i$. We further define $\hat{P}_i := \prod_{j:\hat{z}_j \notin \Delta_i} (\tilde{z}_i - \hat{z}_j)$. Since $|\tilde{z}_i - z_i| < \sigma_i/(64n)$ for all i, we have $(1 - \frac{1}{64n})|z_i - z_j| \leq |\tilde{z}_i - \hat{z}| \leq (1 + \frac{1}{64n})|z_i - z_j|$ for all $j \neq i$ and $\hat{z} \in \Delta_j$. Thus, $|\hat{P}_i|$ is a 2-approximation of $|P_i|$, that is, $1/2|P_i| < |\hat{P}_i| < 2|P_i|$. Similar as in the certification step, we now use approximate interval arithmetic to compute a 2-approximation μ_i of $|\hat{P}_i|$, and thus a 4-approximation of $|P_i|$. A completely similar argument as in the proofs of Lemma 9 and Lemma 11 then shows that we can compute such μ_i's with less than $\tilde{O}(n^3 + n^2\tau_p + n\sum_i \log M(P_i^{-1}))$ bit operations. Now, from the 2- and 4-approximations of σ_i and $|P_i|$, we can determine a b_κ such that (A) the properties (4) to (6) are fulfilled, and, in addition, (B) the inequality $2^{-b/(2m_i)} < 2^{-\kappa}$ holds. Then, from Corollary 2 and Lemma 4, we conclude that Pan's factorization algorithm (if run with $b \geq b_\kappa$) returns, for all i, m_i approximations $\hat{z}$ of z_i with $|\hat{z} - z_i| < 2^{-b/(2m_i)} < 2^{-\kappa}$. Thus, for each i, we can simply choose an arbitrary approximation $\hat{z} \in \Delta_i$ and return the disk $\Delta(\hat{z}, 2^{-\kappa})$ which isolates z_i. The total cost splits into the cost for the initial root isolation and the cost for running Pan's Algorithm with $b = b_\kappa$. Since the latter cost is bounded by $\tilde{O}(nb_\kappa + n^2\Gamma_p)$, the bound (14) follows. □

Finally, we apply the above results to the important special case, where p is a polynomial with integer coefficients.

THEOREM 5. *Let $p(x) \in \mathbb{Z}[x]$ be a polynomial of degree n with integer coefficients of size less than 2^τ. Then, we can compute isolating disks $\Delta(\tilde{z}_i, R_i)$, with $R_i < \frac{\sigma_i}{64n}$, for all roots z_i together with the corresponding multiplicities m_i using*

$$\tilde{O}(n^3 + n^2\tau) \tag{15}$$

bit operations. For a given positive integer κ, we can further refine all disks Δ_i to a size of less than $2^{-\kappa}$ with a number of bit operations bounded by $\tilde{O}(n^3 + n^2\tau + n\kappa)$.

PROOF. In a first step, we compute the square-free part p^* of p. According to [21, §11.2], we need $\tilde{O}(n^2\tau)$ bit operations for this step, and p^* has integer coefficients of bitsize $O(n + \tau)$. The degree of p^* yields the number k of distinct roots of p. Thus, we can directly apply our algorithm from Section 3 to the polynomial p. In order to derive the bound in (15), we have to reformulate the bound from (12) in terms of the degree n and the bitsize τ of p. From [7, Theorem 2], we conclude that $\sum_i m_i \log\max(1, \sigma_i^{-1}) = \tilde{O}(n^2 + n\tau)$. Furthermore, we have $\tau_p \leq \tau$. Finally, we can bound $\sum_{i=1}^k \log M(P_i^{-1})$ by $\tilde{O}(n^3 + n^2\tau)$. For that, we use a similar approach as in the proof of [7, Theorem 2]. That is, we consider a square-free factorization $p(x) = \prod_{l=1}^n (Q_l(x))^l$ of p and write each partial product $\prod_{i:Q_l(z_i)=0} P_i$ in terms of the leading coefficients of Q_l and the resultant $\mathrm{res}(Q_l, Q_l') \in \mathbb{Z}$ of Q_l and its derivative Q_l'. For more details, we refer to [15] and [7].

We turn to the proof of the bound for the cost of refining the isolating disks $\Delta_i(\tilde{z}_i, R_i)$ to a size of less than $2^{-\kappa}$. For the refinement, we consider the square-free part p^*. Note that the disks Δ_i obtained in the first step are also isolating for the roots of p^* (p and p^* have exactly the same distinct roots) and that $R_i < \sigma(z_i, p)/(64n) = \sigma(z_i, p^*)/(64n) \leq \sigma(z_i, p^*)/(64 \deg p^*)$. Thus, proceeding in completely analogous manner as in the proof of Theorem 4 (with the square-free part p^* instead of p) shows that we need $\tilde{O}(n^3 + n^2\tau + n\kappa)$ bit operations for the refinement. □

5. REFERENCES

[1] D. S. Arnon, G. E. Collins, and S. McCallum. Cylindrical Algebraic Decomposition I. *SIAM Journal of Computing*, 13(4):865–889, 1984.

[2] E. Berberich, P. Emeliyanenko, A. Kobel, and M. Sagraloff. Exact Symbolic-Numeric Computation of Planar Algebraic Curves. *CoRR*, abs/1201.1548, 2012.

[3] E. Berberich, M. Kerber, and M. Sagraloff. An efficient algorithm for the stratification and triangulation of an algebraic surface. *Comput. Geom.*, 43(3):257–278, 2010.

[4] D. Bini and G. Fiorentino. Design, Analysis, and Implementation of a Multiprecision Polynomial Rootfinder. *Numerical Algorithms*, 23:127–173, 2000.

[5] D. I. Diochnos, I. Z. Emiris, and E. P. Tsigaridas. On the Asymptotic and Practical Complexity of Solving Bivariate Systems Over the Reals. *J. Symb. Comput.*, 44(7):818–835, 2009.

[6] A. Eigenwillig, M. Kerber, and N. Wolpert. Fast and Exact Analysis of Real Algebraic Plane Curves. In *ISSAC*, pages 151–158, New York, NY, USA, 2007. ACM.

[7] P. Emeliyanenko and M. Sagraloff. On the Complexity of Solving a Bivariate Polynomial System. In *ISSAC*, pages 154–161, New York, NY, USA, 2012. ACM.

[8] I. Z. Emiris, V. Y. Pan, and E. P. Tsigaridas. Algebraic Algorithms. available at `tr.cs.gc.cuny.edu/tr/files/TR-2012001.pdf`, 2012.

[9] D. Eppstein, M. Paterson, and F. Yao. On Nearest-Neighbor Graphs. *Discrete & Comput. Geometry*, 17:263–282, 1997.

[10] L. Gonzalez-Vega and M. E. Kahoui. An Improved Upper Complexity Bound for the Topology Computation of a Real Algebraic Plane Curve. *J. Complexity*, 12(4):527–544, 1996.

[11] H. Hong. An Efficient Method for Analyzing the Topology of Plane Real Algebraic Curves. *Mathematics and Computers in Simulation*, 42(4-6):571–582, 1996.

[12] M. Kerber and M. Sagraloff. Efficient Real Root Approximation. In *ISSAC*, pages 209–216, New York, NY, USA, 2011. ACM.

[13] M. Kerber and M. Sagraloff. A Worst-case Bound for Topology Computation of Algebraic Curves. *J. Symb. Comput.*, 47(3):239–258, 2012.

[14] K. Mehlhorn and M. Sagraloff. A Deterministic Descartes Algorithm for Real Polynomials. *J. Symb. Comput.*, 46(1):70 – 90, 2011. A preliminary version appeared in ISSAC 2009.

[15] K. Mehlhorn, M. Sagraloff, and P. Wang. From Approximate Factorization to Root Isolation with Application to Cylindrical Algebraic Decomposition. *CoRR*, abs:1301.4870, 2013. `http://arxiv.org/abs/1301.4870`.

[16] V. Pan. Univariate Polynomials: Nearly Optimal Algorithms for Numerical Factorization and Root Finding. *J. Symb. Comput.*, 33(5):701–733, 2002.

[17] M. Sagraloff. On the Complexity of Real Root Isolation. *CoRR*, abs/1011.0344, 2010.

[18] M. Sagraloff. When newton meets descartes: a simple and fast algorithm to isolate the real roots of a polynomial. In *ISSAC*, pages 297–304, 2012.

[19] A. Schönhage. The Fundamental Theorem of Algebra in Terms of Computational Complexity. Technical report, Math. Inst. Univ. Tübingen, 1982.

[20] A. Strzebonski. Cylindrical Algebraic Decomposition Using Validated Numerics. *J. Symb. Comp*, 41:1021–1038, 2006.

[21] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, Cambridge, 1999.

[22] C. K. Yap and M. Sagraloff. A Simple but Exact and Efficient Algorithm for Complex Root Isolation. In *ISSAC*, pages 353–360, New York, NY, USA, 2011. ACM.