Heuristic Acceleration of Near-Optimal Polynomial Root-Finders

Soo Go, Won Geun Kim, and Victor Y. Pan

Lehman College and
the Graduate Center of The City University of New York and
Touro University
sgo@gradcenter.cuny.edu
victor.pan@lehman.cuny.edu
http://comet.lehman.cuny.edu/vpan
won-geun.kim2@touro.edu

Abstract. We seek all roots of a polynomial given with its real or complex coefficients or all its roots lying in a fixed convex domain of the complex plane. Some recentl algorithms accelerating classical subdivision iterations solve that problem at near-optimal bit-operation cost for worst-case inputs and can be applied to black box polynomials, represented by an oracle (black box subroutine) for their evaluation rather than their coefficients. Nevertheless, we present dramatic semi-heuristic acceleration of these algorithms. Our root-finders, running at the cost below the known lower bound, must fail for worst-case inputs, but we have never encountered them in our numerical tests.

Key Words: symbolic-numeric computing, polynomial root-finding, subdivision

1 Introduction

1.1 State of the Art

The 4-millennia old problem of univariate polynomial root-finding is still a major research subject of Computer Algebra and Symbolic-Numeric Computing (see Sec. 1. 3 for brief history and related works).

Problem 0. Given $d+1$ real or complex coefficients of a dth degree polynomial

$$p := p(x) := \sum_{i=0}^{d} p_i x^i := p_d \prod_{j=1}^{d} (x - z_j),\ p_d \neq 0, \tag{1}$$

a convex domain $\mathbb{D}$ in the complex plane, and a real b, approximate within $1/2^b$ all $m_{\mathbb{D}} \leq d$ roots[1] of p lying in $\mathbb{D}$.

Much more limited was the study of **Problem 1**, which is Problem 0 for a *black box polynomial* p, represented by an oracle (a black box subroutine) for its evaluation rather than its coefficients.

It is harder to devise black box polynomial root-finders, but they run faster where a polynomial can be evaluated faster, e.g., is sparse, is the sum of a small number of shifted monomials such as $(x - c_i)^d$, or is Mandelbrot-like, defined by a recurrence, such as $p_1(x) := x$, $p_{i+1}(x) := xp_i^2(x) + 1$, $i = 1, 2, \ldots, k$ (see some recent empirical progress in root-finding for Mandelbrot polynomials in $[1, 20]$). Moreover, black box polynomial root-finding avoids the well-known heavy penalty of the coefficients swell for root-squaring and shifting and scaling the variable, which are basic operations for some popular polynomial root-finders.

By combining polynomial evaluation and interpolation, we can transform Problems 0 and 1 into one another but at a much higher cost than the cost of root-finding itself in various important applications where $m_{\mathbb{D}} \ll d$ (see Appendix A).

[1] We always count roots with their multiplicities and consider a root of multiplicity μ numerically - as a root cluster of μ simple roots infinitely close to each other.

[25–27] solve Problems 0 and 1 at the cost of numerical evaluation of p at $O(m^2 \log(d) \log(b))$ points with $O(b \log(d))$ bit precision provided that at most $m := m_{\mathbb{D},\delta}$ roots lie in the δ-neighborhood of a domain $\mathbb{D}$ for a fixed small $\delta > 0$. With incorporation of the breakthrough of [18, 9] for multipoint polynomial evaluation,[2] this enables solution of Problem 0 for worst-case inputs by using $O((m^2 + d)b)$ bit-operations up to a polylog factor in $b + d$.

Up to such a factor, that upper bound matches the lower bound $0.25b \max(d+1, (m+1)m)$ of [25, Cor. B.3]. In words, the root-finders of [25–27] run nearly as fast as one reads coefficients with a precision required for the task.

1.2 Technical background

Subdivision root-finders, traced back to Herman Weyl's paper [30] of of 1924, have been accelerated in [8, 28, 22] and recently - to near-optimal level - in the papers [25–27], which reduce root-finding to *σ-soft exclusion/inclusion (e/i) tests* of $O(m)$ complex discs for any fixed σ between 1 and $\sqrt{2}$ and to at most $2m-2$ *disc compression steps*, in turn reduced in [27, Cor. 4.1] to order of $m \log(b)$ soft e/i tests overall.

Definition 1 *A σ-soft e/i test of a disc $D(c, \rho) = \{x : |x - c| \le \rho\}$ outputs 0 and stops if it detects that $m_{D(c,\rho)} = 0$ (exclusion); it outputs 1 and stops if it detects that $m_{D(c,\sigma\rho)} > 0$ (σ-soft inclusion). The test terminates as soon as it verifies one of the two criteria, $m_{D(c,\sigma\rho)} > 0$ or $m_{D(c,\rho)} = 0$, without checking if the other criterion also holds.*

For a disc D with $m = m_D$ roots and any polynomial p, [25–27] reduce a σ-soft e/i test essentially to the evaluation of *Newton's inverse ratio (NIR)* $\frac{p'}{p}$ at $(m+1)\lceil \log_\sigma(1+6m\sqrt{m+1})\rceil$ points x (see [25, Prop. 3.4]). We further outline this previous work in Appendix C.

1.3. Our contribution. Our semi-heuristic soft e/i test of Sec. 2 (Alg. 1) evaluates NIR at much fewer points, which implies dramatic acceleration of the near-optimal root-finders of [25–27]. In addition, we accelerate the compression steps of [25–27] by a factor of 33.5, based on extension of our soft e/i test of Sec. 2 (Alg. 1) from a disc to a ring (annulus) in the complex plane (see Sec. 2.6).

Since our Alg. 1 enables root-finding at the cost below lower bounds, it must fail for some polynomials p, but we prove some constraints on the roots of such polynomials and have never encountered them in our numerical experiments of Sec. 5 (the first version of the github repository for our codes is publicly available via the following URL link:
https://github.com/wongunkim/heuristic_acc_root_finder).

In Sec. 3 we cover our two other soft e/i tests for discs (Algs. 2 and 3) as well as their dual variants (Algs. 2d and 3d). These algorithms are technically different from Alg. 1 and only apply to discs (not to rings); for discs they nearly match Alg. 1 in computational cost and in output accuracy according to our initial numerical experiments, currently in progress.

In Sec. 4 we recall [27, Thm. A.1], based on which at a dominated cost we can count the roots approximated by a root-finder at the end of the computations. Then, if the target number m of roots is available, e.g., if $m = d$, or if it can be computed at the dominated cost (e.g., if we seek roots in a domain well isolated from external roots), then we can detect at a dominated cost whether any of our e/i tests of Alg. 1, 2, 3, 2d or 3d has failed.

1.3 Brief history and related works

Hundreds of efficient root-finders have appeared [17, 19] and keep appearing. In 1982 Schönhage [29] proposed to compare their efficiency by estimating bit-operations involved, reaching $\tilde{O}(bd^3)$ in his record-fast solution of Problem 0 (we write $\tilde{O}$ for O defined up to polylog factors in $b+d$). Subsequent extensive research has culminated in 1995 (see [21, 23]) with the bound $\tilde{O}((b+d))d^2)$ - near-optimal for $d = \tilde{O}(b)$. The next near-optimal algorithms for Problems 0 appeared in [25–27]. They also solved Problem 1, and [10, 11] implemented and tested their initial variants from [24]. The advanced papers [4, 18, 9] deduced similar cost bounds for Problem 0 simplified with imposing some lower bounds $\delta > 0$ on the pairwise distances between roots of p (see a

[2] [18, 9] combine their novel piece-wise approximation of p by low degree polynomials with nontrivial [15, Alg. 5.3] of 1998, rediscovered in 2013 by Kobel and Sagraloff.

discussion in [29, Secs. 19 and 20] on such simplification). This was a major restriction because the input and rounding errors turn multiple roots into tiny root clusters. The estimates for the bit operation complexity of the efficient root-finders of [4] and [18, 9] include the terms $d\log(\frac{1}{\delta})$ and $d\log(\kappa(\delta))$, respectively, where $\kappa(\delta)$ denotes the maximal condition of the roots; both of these terms grow to ∞ as δ decreases to 0. At the end of Sec. 1.1 we cited Incorporation of fast multi-point polynomial evaluation of [18, 9] into black box root-finder of [25–27] for its extension to the solution of Problems 1 and $1_{\mathbb{D}}$. Otherwise [25–27] are technically independent of [4, 18, 9]. [3]

The known polynomial root-finders preceding [24–27] involve coefficients of p, e.g., user's choice MPSolve [2] needs them for the initialization of its root-finding iterations. The only exception is the advanced pioneering root-finder of [16], although it is restricted to a polynomials having only real roots and only approximates its single absolutely largest root.

2 Background and soft e/i test 1

2.1 Definitions

With evaluation oracle for p we can readily compute the degree d and approximate the coefficient p_d because $\lim_{|x|\mapsto\infty}\frac{p(x)}{x^g}$ is 0 for $g>d$ and p_d for $g=d$, while $\lim_{|x|\mapsto\infty}|\frac{p(x)}{x^g}|=\infty$ for $g<d$. [25–27] used evaluation oracle to approximate $p'(x)$ by the divided difference at points $y\approx x$ and $z\approx x$ and proved near-optimality of the resulting bit-complexity estimates. Next write $\mathrm{NIR}_1(x):=\mathrm{NIR}(x)$ and obtain from factorization in (1) that

$$\mathrm{NIR}_1(x):=\mathrm{NIR}(x)=\sum_{j=1}^{d}\frac{1}{x-z_j}. \tag{2}$$

Extend root-squaring from $p(x)$ (cf. [7]) to $\mathrm{NIR}(x)$ and further to root-lifting:

$$2^i x^{2^i-1}\mathrm{NIR}_{2^{i+1}}(x^{2^{i+1}}):=\mathrm{NIR}_{2^i}(x^{2^i})-\mathrm{NIR}_{2^i}(-x^{2^i}),\ i=0,1,\dots. \tag{3}$$

$$p_h(x^h):=p_d\prod_{j=1}^{d}(x-z_j^h)=(-1)^{dh+d}\prod_{g=0}^{h-1}p(\zeta_h^g x),\ hx^{h-1}\mathrm{NIR}_h(x^h):=\sum_{g=0}^{h-1}\zeta_h^g\mathrm{NIR}(\zeta_h^g x), \tag{4}$$

for a primitive hth root of unity $\zeta_h:=\exp\left(\frac{2\pi\sqrt{-1}}{h}\right)$ and $h=2,3,\dots$.

Extend these definitions and expressions to the *reverse polynomial*

$$p_{\mathrm{rev}}:=x^d p\Big(\frac{1}{x}\Big)\text{ for }x\neq 0,\quad p_{\mathrm{rev}}=p_0\prod_{j=1}^{d}\Big(x-\frac{1}{z_j}\Big)\text{ if }p_0=p(0)\neq 0, \tag{5}$$

$\mathrm{NIR}_{\mathrm{rev}}(x):=\frac{p'_{\mathrm{rev}}(x)}{p_{\mathrm{rev}}(x)}=\frac{d}{x}-\frac{1}{x^2}\mathrm{NIR}\Big(\frac{1}{x}\Big)$, and $\mathrm{NIR}_{h,\mathrm{rev}}(x^h)$, $h=0,1,\dots$.
Define *extremal root radii* $r_1(p):=\max_{j=1}^{d}|z_j|=\frac{1}{r_d(p_{\mathrm{rev}})}$, $r_d(p):=\min_{j=1}^{d}|z_j|=\frac{1}{r_1(p_{\mathrm{rev}})}$.

2.2 The power sums of the roots

Definition 2 *(i)* $s_h(\mathbb{D}):=\sum_{z_j^h\in\mathbb{D}}z_j^h$, *for a domain* $\mathbb{D}$ *and* $h=0,\pm1,\pm2,\dots$,
(ii) $s_h:=\sum_{j=1}^{d}z_j^h$ *for* $h=0,\pm1,\pm2,\dots$,
(iii) $s_{h,q}:=\frac{1}{q}\sum_{g=0}^{q-1}\zeta_q^{(h+1)g}\,\mathrm{NIR}(\zeta_q^g)$, $s_{-h,q}:=\frac{1}{q}\sum_{g=0}^{q-1}\zeta_q^{(h+1)g}\,\mathrm{NIR}_{\mathrm{rev}}(\zeta_q^g)$ *for* $h=0,1,\dots,q-1$,
(iii) $\delta_{h,q}:=|s_{h,q}-s_h(\mathbb{D})|$, *for* $q>1$ *and* $h=0,\pm1,\dots,\pm(q-1)$.

[3] The announcement in [3] of a new near-optimal solution of Problem 0 (with no restriction on δ) stirred confusion because for all proofs [3] referred to [4]. Already the implementations in [10, 11] of some root-finders of [24] greatly superseded the best implementation of the algorithm of [3] in [12].

Readily verify the following lemma and relationships,

$$s_h = -\mathrm{NIR}_{h,\mathrm{rev}}(0),\ s_{-h} = \mathrm{NIR}_h(0)\ (\text{cf. } (2)), \tag{6}$$

$$r_d^h \le \frac{|s_h|}{d} \le r_1^h,\ \frac{1}{r_1^h} \le \frac{|s_{-h}|}{d} \le \frac{1}{r_d^h}, \tag{7}$$

Lemma 1. *For a domain $\mathbb{D}$, it holds that $s_0(\mathbb{D}) = m_{\mathbb{D}}$ and that $s_h(\mathbb{D}) = 0$ for all h if $s_0(\mathbb{D}) = 0$.*

Lemma 2. *Let $|s_{-h}| > \frac{d}{\sigma^h}$ for some $h > 0$. Then $m_{D(0,\sigma)} > 0$.*

Proof. Combined the assumed lower bound on $|s_{-h}|$ with its upper bound of (7) and obtain that $\frac{1}{r_d^h} > \frac{1}{\sigma^h}$, and so $r_d > \sigma$, that is, $m_{D(0,\sigma)} > 0$.

Definition 1. *For a polynomial p and $\sigma \ge 1$ the circle $C(c,\rho) = \{x :\ |x-c| \le \rho\}$ is* σ-isolated *if $m_{D(c,\rho/\sigma)} = m_{D(c,\rho\sigma)}$.*

Lemma 3. *[See [29, Eqn. (12.10)] for $h \ne 0$, [24, 27].] Let $m := m_{D(0,1)}$ and let the circle $C(0,1)$ be σ-isolated for $\sigma \ge 1$. Then $\delta_{h,q} \le \frac{m/\sigma^h + (d-m)\sigma^h}{\sigma^q - 1}$ and $\delta_{-h,q} \le \frac{(d-m)/\sigma^h + m\sigma^h}{\sigma^q - 1}$, for $h = 0, 1, \dots, q-1$.*

Example 1. Apply the lemma to 2-isolated unit circle $C(0,1)$ and obtain $\delta_{0,q} < 1/2$ for $d \le 1,000$ if $q = 10$ and for $d \le 1,000,000$ if $q = 20$.

Remark 1. For a fixed h the error bound on $\delta_{h,q}$ of Lemma 3 decreases proportionally to $\frac{1}{\sigma^q}$. Root-squaring of (3) squares σ, but doubling q makes the same impact on $\frac{1}{\sigma^q}$ and hence on the estimates of Lemma 3 for fixed h, which are quite sharp empirically [10, 11].

2.3. Algorithm 1: σ-soft e/i test 1 of the unit disc $D(0,1)$.
INPUT: a black box polynomial p of a degree d and $\sigma > 1$.
INITIALIZATION: fix two positive integers u_- and $u_+ \ge u_-$ (see Sec. 2.4).
COMPUTATIONS: Recursively, for $q = 2^u$, $u = u_-, u_- + 1, \dots, u_+$, compute
(i) $\mathrm{NIR}(\zeta_q^g)$ for $g = 0, 1, \dots, q-1$ and (ii) $s_{h,q}$ for $h = 0, \pm 1, \dots, \pm(q-1)$.
OUTPUT 1 and stop if (i) $\frac{|s_{h,q}|}{d} > \frac{\sigma^h}{\sigma^q - 1}$ or (ii) $\frac{|s_{-h,q}|}{d} > \frac{1}{\sigma^h} + \frac{1}{(\sigma^q - 1)\sigma^h}$ for some pair of $h \ge 0$ and $u = \bar{u} \le u_+$.
OUTPUT 0, write $\bar{u} := u_+$, and stop otherwise.

2.4. Complexity of Alg. 1. At stage (i) we evaluate $\mathrm{NIR}(x)$ at all qth roots of unity for every u and $q = 2^u$. At stage (ii) we apply discrete Fourier transform at the qth roots of unity twice to compute $2q-1$ linear combinations of these computed values at the arithmetic cost $3u2^u$. As u increases, we reuse all computed values of NIR and their linear combinations. Hence the *overall complexity* of Alg. 1 is dominated by the cost of computing NIR at $2^{\bar{u}}$ points. [$\bar{u} < u_+$ only if Ag. 1 outputs 1, while $\bar{u} = u_+$ is compatible with any of outputs 0 and 1.]

2.5. Correctness of Alg. 1. Suppose that $\frac{|s_{h,q}|}{d} > \frac{\sigma^h}{\sigma^q - 1}$. Then $\frac{|s_{h,q}|}{d} - \delta_{h.q} > 0$ by definition of $\delta_{h,q}$. Therefore, $s_h(D(0,1)) \ge \frac{|s_{h,q}|}{d} - \delta_{h,q} > 0$ by virtue of Lemma 3; hence $m_{D(0,1)} > 0$ by virtue of Lemma 1. Likewise, let $\frac{|s_{-h,q}|}{d} > \frac{1}{\sigma^h} + \frac{1}{(\sigma^q - 1)\sigma^h}$. Then $\frac{|s_{-h,q}|}{d} - \frac{1}{\sigma^h} > \delta_{-h,q}$ by definition of $\delta_{-h,q}$. Therefore, $s_{-h}(D(0,1)) \ge \frac{|s_{-h,q}|}{d} - \delta_{-h,q} > 0$ by virtue of Lemma 3; hence $m_{D(0,1)} > 0$ by virtue of Lemma 2.

Hence Alg. 1 fails only if it outputs 0 while $m_{D(0,1)} = 0$. In that case the roots are restricted by the inequalities

$$\frac{|s_{h,q}|}{d} \le \frac{\sigma^h}{\sigma^q - 1}, \quad \frac{|s_{-h,q}|}{d} \le \frac{1}{\sigma^h} + \frac{1}{(\sigma^q - 1)\sigma^h}, \quad \text{for all } h. \tag{8}$$

For $0 < h < q$ they converge to the equations $s_{h,q} = 0$ as $\sigma^q \mapsto \infty$ and $s_{-h,q} = 0$ as $\sigma^h \mapsto \infty$. In our numerical tests the output 0 of Alg. 1 was 100% correct already for $q \ll d$.

2.6. Two extensions.

Lemma 4. Reduction of e/i tests to the unit disc. *For a complex c and positive ρ and σ, the map $x \mapsto \frac{(x-c)\sigma}{\rho}$, combining* shift *and* scaling *of the variable x, moves the discs $D(c,\rho) \mapsto D(0,\frac{1}{\sigma})$ and moves the zeros z_j of $p(x)$ into the zeros $w_j = \frac{\rho(z_j-c}{\sigma}$ of $t(w) := p(\frac{(w-c)\sigma}{\rho})$ for $j = 1,\dots,d$.*

Now extend the e/i test of Alg. 1 and its analysis to a disc $D(c,\rho)$ and the values

$$s_{h,q}(D(c,\rho)) := \frac{\rho}{q}\sum_{g=0}^{q-1}\zeta_q^{g(h+1)}\frac{p'(c+\rho\zeta^g)}{p(c+\rho\zeta^g)} \text{ for } h = 0,1,\dots,q-1$$

and to a ring (annulus) $A(c,\rho_-,\rho_+) := \{x:\ \rho_- \le |x-c| \le \rho_+\}$ and the values

$$s_{h,q}(A(c,\rho_-,\rho_+)) := \frac{\rho_+}{q}\sum_{g=0}^{q-1}\zeta_q^{g(h+1)}\frac{p'(c+\rho_+\zeta^g)}{p(c+\rho_+\zeta^g)} - \frac{\rho_-}{q}\sum_{g=0}^{q-1}\zeta_q^{g(h+1)}\frac{p'(c+\rho_-\zeta^g)}{p(c+\rho_-\zeta^g)}$$

for $h = 0,1,\dots,q-1$. The complexity of this test of a ring doubles that of a disc and by a factor of 33.5 is less than in [11, 25, 26] (cf. [11, Cor. 13]). This speed up is immediately extended to the disc compression algorithm, reduced to order of $\log(b)$ soft e/i tests in [11, 25, 26].

3 Other new soft e/i tests

3.1 Low-cost estimates for extremal root radii

We begin with the following observation:

Lemma 5. *If $m_{D(0,\delta)} = 0$ for $\delta > 0$, in particular if a σ-soft e/i test for the disc $D(0,\delta)$ outputs 0, then $\delta \le r_d(p)$ and $\frac{1}{\delta} \ge r_1(p_{\rm rev}))$ (cf. (5)).*

Apply this lemma and a σ-soft e/i test to a disc $D(0,\delta)$ for fast decreasing values $\delta = 1/\sigma > 0$ until the test outputs 0. At a cost of performing these σ-soft e/i tests, obtain a lower bound $\delta \le r_d$ and similarly obtain an upper bound $\sigma \ge r_1$ by applying this recipe to $p_{\rm rev}$. [σ-soft e/i tests for large σ are much less costly than for $1 < \sigma < \sqrt{2}$, involved in subdivision iterations.]

3.2 Approximation of the power sums s_{-h}

Schröder's iterations, $x_{i+1} = x_i - \mu\frac{f(x_i)}{f'(x_i)}$, $i = 0,1,\dots$, converge fast to a root z of a functional equation $f(x) = 0$ having multiplicity μ and isolated from the other roots. The next theorem, proved in Appendix B, covers Schröder's single iteration for $\mu = d$, $f = p_h(x)$, and $h \ge 1$.

Theorem 3. *For an integer $h \ge 1$, $r \ge r_1$, and a complex x such that $|x| > 2r$, write*

$$y = y_h(x) := x^h - \frac{d}{\mathrm{NIR}_h(x^h)} \text{ and } \eta := \frac{|x|}{r^2}, \tag{9}$$

and let $m_{D(0,r)} = d$. Then $|y - \frac{s_h}{d}| \le \frac{2}{\eta^h - 2/r^h}$.

We apply Thm. 3 for p replaced by $p_{\rm rev}$ and in all its applications use an upper bound r for the extremal root radius $r_1(p)_{\rm rev} = \frac{1}{r_d(p)}$ computed once and for all according to Sec. 3.1.

Lemma 2 ensures correctness of inclusion (output 1) if $\frac{|s_h|}{d} > \frac{1}{\sigma_h}$, while Alg. 1 estimates $\frac{|s_h|}{d}$ based on Lemma 3. In our next Alg. 2 we apply Lemma 2 again but estimate $\frac{|s_h|}{d}$ by applying Thm. 3 for p replaced by $p_{\rm rev}$.

3.3 Algorithm 2 (soft e/i test 2)

INPUT as in Alg. 1 and $r > \frac{1}{r_d} = \max_{j=1}^{d} \frac{1}{|z_j|}$.

INITIALIZATION: fix two integers $i_- \geq 0$ and $i_+ \geq i_-$ and a complex x such that $\eta = \frac{|x|}{r^2}$ of (9) is large enough.

COMPUTATIONS: Recursively, for $h = 2^i$ and $i = i_-, i_- + 1, \dots$, compute (i) y of Eqn. (9) for $p(x) := p_{h,\mathrm{rev}}(x)$ and (ii) $\delta_h := \frac{2}{\eta^h - 2/r^h}$.

(OUTPUT 1 (inclusion) and stop if $|y| > \frac{1}{\sigma^h} + \delta_h$ for $h = 2^i$ and some $i = \bar{i} \leq i_+$.

OUTPUT 0 (exclusion) and stop otherwise.

Correctness of Alg. 2. If $|y| > \frac{1}{\sigma^h} + \delta_h$, then $\frac{|s_{-h}|}{d} > \frac{1}{\sigma^h}$ by virtue of Thm. 3, and hence output 1 is correct by virtue of Lemma 2. The roots of the polynomials p for which Alg. 2 wrongly outputs 0 satisfy the system of inequalities $\frac{|s_{-h}|}{d} \leq \frac{1}{\sigma^h}$, for $h = 2^i$ and $i = i_-, i_- + 1, \dots, i_+$. The inequalities are quite restrictive for large h, particularly if also η is large; they converge to the equations $s_{-h} = 0$ as $h \mapsto \infty$; furthermore, $\delta_h \mapsto 0$ as $\eta^h \mapsto \infty$.

Complexity analysis: While computing $\mathrm{NIR}_{2^i,\mathrm{rev}}(x^{2^i})$ for $i > 0$ based on (3) or (4), reuse all previously computed values of NIR to evaluate NIR just at $2^{\bar{i}}$ points overall. The cost of that computation dominates the overall *complexity* of Alg. 2. Based on the previous study we can choose a narrow range $[u_-, u_+]$ to decrease the overall complexity of the algorithm.

Remark 2. To devise an alternatve dual algorithm (said to be **Alg. 2d**), approximate $s_{-h} = -\mathrm{NIR}_{-h}(0)$ (cf. (6)) with $-\mathrm{NIR}_{-h}(x)$ for $|x| \ll r_d$ and then approximate $\delta_h = |\mathrm{NIR}_{-h}(x) - \mathrm{NIR}_{-h}(0)|$ with $0.5|\mathrm{NIR}_h(x) - \mathrm{NIR}_h(-x)|$. We expect that this approximation tends to be close, although we cannot extend to it our formal support of Alg. 2 by Thm. 3.

3.4 Algorithm 3 (soft e/i test 3)

It combines Algs. 1 and 2 and outputs 1 if this is implied by the criteria of any of the two algorithms. Otherwise it outputs 0. Likewise, **Alg. 3d** combines Alg. 1 and Alg 2d. It may seem that the complexity of performing these algorithms doubles that of Alg. 1, but with careful reuse of the computed values of NIR one can increase the complexity just by a factor of 1.5.

4 Detection of output errors

The root-finders of [25–27] output a set $\mathcal{S}$ of discs of radii at most $1/2^b$ that together contain all $m_{\mathbb{D}}$ roots lying in an input domain $\mathbb{D}$ except for those lost due to erroneous exclusion claims. We can detect such losses if know $m_{\mathbb{D}}$ (e.g., if we seek all d roots) or can compute $m_{\mathbb{D}}$ at the dominated cost (e.g., based on Lemma 3 if $\mathbb{D}$ is a $\bar{\sigma}$-isolated disc for $\bar{\sigma} - 1$ exceeding a positive constant). Then we complement the root-finders of [25–27] with a root-counter for the union of all discs in the set $\mathcal{S}$. We can do this by first performing pairwise subtractions of $k = O(m_{\mathbb{D}})$ computed values and comparisons of the $0.5(k-1)k$ differences with $1/2^b$ (by using $\tilde{O}(k^2)b$) bit operations) - to cover all roots in $\mathbb{D}$ with σ-isolated discs, namely, with $\bar{k} \leq m_{\mathbb{D}}$ discs for $\sigma := \frac{3k+1}{3k-1}$ (see [27, Thm. A.1]). Then we can count all output roots by applying our Lemma 3 for $h = 0$ to all these discs for $q = \lfloor \log_\theta(4d+2) \rfloor$.

5 Numerical experiments for Alg. 1

[See the first version of the github repository for our codes publicly available via the following URL link: https://github.com/wongunkim/heuristic_acc_root_finder.] For each test polynomial of degree d, we have chosen its m roots uniformly at random in the disc $D(0, 1/1.00001)$ and its other $d - m$ roots uniformly at random in the ring $A(0, 1, 10^{14})$. Tiny isolation of the sets of m internal and $d - m$ external roots from one another ensured that the roots in these sets were counted correctly in our experiments. [25–27] supported near-optimal root-finding by fixing computational precision of order of $b \log(d)$ bits, and this can be extended to Alg. 1, but in our experiments we only dealt with soft e/i tests for polynomials p having roots in specified domains, and double complex precision (15 decimal digits per component - the default complex

type [128bits] in `NumPy`) was sufficient to ensure correctness of our computations. We implemented the algorithms in Python, by using the `NumPy` library.

Implementation and Parameters. We performed our tests for $d = 6400, 12800, 25600$, $\sigma = 1.4$, and $\mu = 0, 1, d/4, d/2, 3d/4, d-1, d$.

Remark 3. In our similar tests for $\sigma = 1.1$ we consistently observed increase of q by a factor of about $\log(1.4)/\log(1.1) \approx 3.5$ in good accordance with our observations in Remark 1.

For each of the 21 pairs (d, ν) we run 100 independent trials.
We set $u_- = 0$ and $u_+ = \lceil \log_2(\log_\sigma(d)+1) \rceil$ equal to 5 in our tests. We displayed the test results in Table 1b. In all 2,100 trials, Alg. 1 succeeded in 100% tests, by correctly outputting 0 for $\nu = 0$ and 1 otherwise. We displayed the maximal number of points at which NIRs were evaluated and the maximal integer h used. The dominant cost of computing NIR remained stable as d scaled but decreased significantly as m increased. We are still testing Algs. 2 and 3 (limited to discs, while Alg.1 also test rings). For discs they perform slightly worse than Alg. 1 in our initial tests.

Table 1: Performance of Algorithm 1

Input Parameters						**Output**
d	σ	θ	**Number of roots m in $D(0,1/\theta)$**	**Max #NIR Evals**	**Max h**	**incorrect**
6400	1.4	1.00001	$m = 0$	32	31	0
			$m \in \{1,2,3,4,5,10\}$	32	0	0
			$m \in \{50, 100, 400\}$	16	0	0
			$m \in \{800, 1600\}$	8	0	0
			$m \in \{3200, 4800, 6399, 6400\}$	4	0	0
12800	1.4	1.00001	$m = 0$	32	31	0
			$m \in \{1,2,3,4,5,10,50\}$	32	0	0
			$m \in \{100, 800\}$	16	0	0
			$m \in \{1600, 3200\}$	8	0	0
			$m \in \{6400, 9600, 12799, 12800\}$	4	0	0
25600	1.4	1.00001	$m = 0$	32	31	0
			$m \in \{1,2,3,4,5,10,50,100\}$	32	0	0
			$m = 1600$	16	0	0
			$m \in \{3200, 6400\}$	8	0	0
			$m \in \{12800, 19200, 25599, 25600\}$	4	0	0

(a) Tests with Various Numbers m of Internal Roots

Root Configuration	**Max #NIR Evals**	**Max h**
All roots external	32	31
Single internal root	32	0
25% internal roots	8	0
$\geq$50% internal roots	4	0

(b) Test results were invariant for $d \in \{6400, 12800, 25600\}$ (with all 2,100 outputs 100% correct)

Acknowledgement: Reviewers' thoughtful comments helped improve greatly our presentation.

References

1. Bini, Dario A.: Numerical computation of the roots of Mandelbrot polynomials: an experimental analysis, *ETNA - Electronic Transs. on Numerical Analysis*, pp. 1-27, 2024/03/06. doi: 10.1553/etna_vol61s1
2. Bini, D.A., Robol, L.: Solving secular and polynomial equations: a multiprecision algorithm. *J. Comput. Appl. Math.* **272** 276–292 (2014) doi: 10.1016/j.cam.2013.04.037

3. Becker, R., Sagraloff, M., Sharma, V., Xu, J., Yap, C.: Complexity analysis of root clustering for a complex polynomial. In ISSAC 2016, pp. 71–78. ACM Press, NY (2016) doi: 10.1145/2930889.2930939
4. Becker, R., Sagraloff, M., Sharma, V., Yap, C.: A near-optimal subdivision algorithm for complex root isolation based on the Pellet test and Newton iteration. J. Symb. Comput. **86** 51–96 (2018) doi: 10.1016/j.jsc.2017.03.009
5. Cox, David A.; Little, John; O'Shea, Donald: Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra. Springer (1997). ISBN 0-387-94680-2
6. Edelman, A., Kostlan, E.: How many zeros of a random polynomial are real? *Bull. Amer. Math. Soc.* **32**, 1–37 (1995)
7. Householder, A.S.: Dandelin, Lobachevskii, or Graeffe? *Amer. Math. Monthly* **66**, 464–466 (1959) doi: 10.2307/2310626
8. Henrici, P.: Applied and Computational Complex Analysis. Vol. 1: Power Series, Integration, Conformal Mapping, Location of Zeros. Wiley, NY (1974)
9. Imbach, R., Moroz, G.: Fast evaluation and root finding for polynomials with floating-point coefficients, In: Procs. ISSAC'23, 325-334 (2023) ACM Prterss, NY,
10. Imbach, R., Pan, V.Y.: New progress in univariate polynomial root-finding. In: Procs. ACM-SIGSAM ISSAC 2020, 249–256 (2020) ACM ISBN 978-1-4503-7100-1/20/07
11. Imbach, R., Pan, V.Y.: Accelerated subdivision for clustering roots of polynomials given by evaluation oracles. In CASC 2022, Springer's LNCS, **13366**, 143–164 (2022) also arXiv:2206.08622 (2022)
12. Imbach, R., Pan, V.Y., Yap, C.: Implementation of a near-optimal complex root clustering algorithm. In ICMS 2018, Springer's LNCS **10931**, 235–244 (2018) doi: 10.1007/978-3-319-96418-8 28
13. Imbach, R.; Pouget, M.; Yap, C.: Clustering Complex Zeros of Triangular Systems of Polynomials. *Math. Comput. Sci.* **15 (2)**, 271–292 (2021)
14. Kac, M.: On the average number of real roots of a random algebraic equation. *Bull. Amer. Math. Soc.* **49**, 314–320 and 938 (1943)
15. Kirrinnis, P.: Polynomial factorization and partial fraction decomposition by simultaneous Newton's iteration. J. Complex. **14**, 378–444 (1998) doi: 10.1006/jcom.1998.0481
16. Louis, A., Vempala, S. S.: Accelerated Newton iteration: roots of black box polynomials and matrix eigenvalues. IEEE FOCS 2016, **1**, 732-740 (2016)
17. McNamee, J.M.: Numerical Methods for Roots of Polynomials, Part I, XIX+354 pages. Elsevier (2007) ISBN: 044452729X; ISBN13: 9780444527295
18. Moroz, G.: New data structure for univariate polynomial approximation and applications to root isolation, numerical multipoint evaluation, and other problems. In: IEEE FOCS 2021, 1090–1099 (2021).
19. McNamee, J.M., Pan, V. Y.: Numerical Methods for Roots of Polynomials, Part 2 (XXII + 718 pages), Elsevier (2013).
20. Mihalach, N., Vigneron, F.: How to split a tera-polynomial, arXiv:2402.06083 (2024)
21. Pan, V.Y.: Optimal (up to polylog factors) sequential and parallel algorithms for approximating complex polynomial zeros. ACM STOC'95, 741–750 (1995)
22. Pan, V.Y.: Approximation of complex polynomial zeros: modified quadtree (Weyl's) construction. and improved Newton's iteration. J. Complexity **16**(1), 213–264 (2000) doi: 10.1006/jcom.1999.
23. Pan, V.Y.: Univariate polynomials: nearly optimal algorithms for factorization and rootfinding. J. Symb. Comput. **33**(5), 701–733 (2002) Proc. in ISSAC'01 (2001)
24. Pan, V.Y.: New progress in polynomial root-finding. arXiv 1805.12042 (2018). Revised (2024)
25. Pan, V.Y.: Near-optimal black box polynomial root-finders, Proc. ACM-SIAM Symp. on Discrete Algorithms (SODA24), ACM Press and SIAM, 2024.
26. Pan, V.Y.: Subdivision and Schröder's Polynomial Root-Finders Combined, In: Procs. SYNASC, 25-33, 2025.
27. Pan, V.Y.; Go, Soo; Luan, Qi; Zhao, Liang: A New Fast Root-Finder for Black Box Polynomials. *Theor. Comp. Science,* **1027** (2025) 115022 (Sec. A: Algorithms, automata, complexity and games, Edited by Paul Spirakis), February https://doi.org/10.1016/j.tcs.2024.115022.
28. Renegar, J.: On the worst-case arithmetic complexity of approximating zeros of polynomials. J. Complex. **3**(2), 90–113 (1987) doi: 10.1016/0885-064X(87)90022-7
29. Schönhage, A.: The fundamental theorem of algebra in terms of computational complexity. Math. Dept., University of Tübingen, Tübingen, Germany (1982)
30. Weyl, H.: Randbemerkungen zu Hauptproblemen der Mathematik. II. Fundamentalsatzder Algebra und Grundlagen der Mathematik. MathematischeZeitschrift **20**, 131–151 (1924)

APPENDIX

A. Variations and Extensions. (i) Solution of Problems 0 and 1 can be extended to root-finding for a triangular multivariate polynomial system of equations [5, 13]. The overall number of solutions of such a system can grow exponentially with its total degree, while only a small number of them lie in the Region of User's Interest.[4] (ii) *Real root-finding* (in a real line segment) is highly important because in applications, e.g., to optimization in algebraic geometry and geometric modeling, only real roots of a polynomial are of interest, and typically they are much less numerous than all d complex roots [14, 6]. (iii) Our black box algorithms applied to a narrow rectangle about the real line segment approximate *nearly real roots*, lying near the real axis, e.g., perturbations of real roots caused by input and rounding errors.

B. Proof of Thm. 3. First let $r = h = 1$. Substitute (2) into (9) and obtain

$$y = x - \frac{d}{\mathrm{NIR}(x)} = x - \frac{d}{\sum_{j=1}^{d} \frac{1}{x - z_j}} = x - \frac{xd}{\sum_{j=1}^{d} \frac{1}{1 - z_j/x}}.$$

Recall that $|\frac{z_j}{x}| < 1$ for all j, by theorem's assumptions for $r = 1$, substitute Newman's expansion $\frac{1}{1-v} = \sum_{g=0}^{\infty} v^g$ for $v = \frac{z_j}{x}$ into the above equation, and deduce that

$$\frac{y}{x} = 1 - \frac{d}{\sum_{j=1}^{d} \sum_{g=0}^{\infty} z_j^g / x^g} = 1 - \frac{d}{\sum_{g=0}^{\infty} \sum_{j=1}^{d} z_j^g / x^g}.$$

Substitute equations $s_g = \sum_{j=1}^{d} z_j^g$, $g = 1, 2, \ldots$, and obtain

$$\frac{y}{x} = 1 - \frac{d}{d + \sum_{g=1}^{\infty} s_g / x^g} = 1 - \frac{1}{1+u} \text{ for } u = \sum_{g=1}^{\infty} \frac{s_g}{d x^g}.$$

Notice that $1 - \frac{1}{1+u} = u - \frac{u^2}{1+u}$, and so $y = xu - \frac{xu^2}{1+u}$. Furthermore, $xu = \frac{s_1}{d} + \Gamma$ for $\Gamma := \sum_{g=2}^{\infty} \frac{s_g}{d} x^{1-g}$. Therefore,

$$y - \frac{s_1}{d} = \Gamma - \frac{xu^2}{1+u}, \text{ and so } \left| y - \frac{s_1}{d} \right| \le |\Gamma| + \left| \frac{xu^2}{1+u} \right|. \tag{10}$$

Notice that $\frac{|s_h|}{d} \le 1$ for all h because $|z_j| \le 1$ for all j and deduce that $|u| \le \frac{1}{|x|} \sum_{g=0}^{\infty} \frac{1}{|x|^g} \le \frac{1}{|x|-1}$ and hence

$$|xu^2| \le \frac{|x|}{(|x|-1)^2}, \quad \frac{1}{|1+u|} \le \frac{1}{1 - \frac{1}{|x|-1}} = \frac{|x|-1}{|x|-2}.$$

Furthermore, $|\Gamma| \le \sum_{g=2}^{\infty} |x|^{1-g} = \frac{1}{|x|-1}$. Combine these bounds with Eqn. (10) and obtain the theorem in the case of $r = h = 1$:

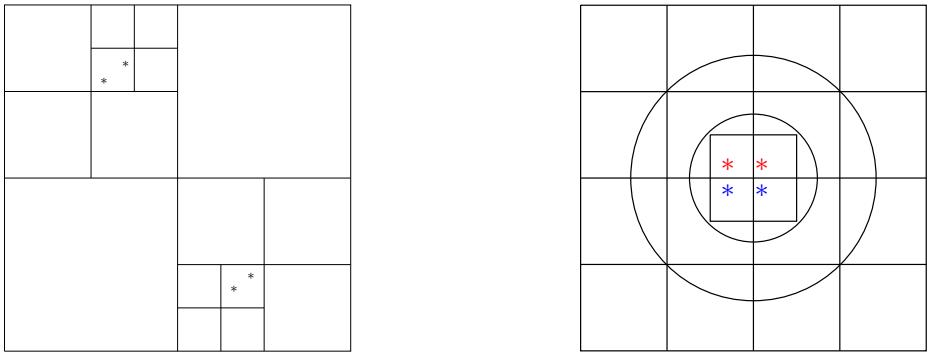
$$\left| y - \frac{s_1}{d} \right| \le \frac{1}{|x|-1} + \frac{|x|}{(|x|-1)(|x|-2)} = \frac{2}{|x|-2}.$$

Next scale the variable $x \mapsto x/r$ implying that $D(0, r) \mapsto D(0, 1)$, $y \mapsto y/r$ (cf. (2)), and $s_1 \mapsto s_1/r$, apply Thm. 3 for $r = h = 1$, and extend it to the case of any $r > 0$ and $h = 1$. To complete the proof for any $h \ge 1$, apply that result to $p_h(x)$ rather than $p(x)$, implying the maps $x \mapsto x^h$, $r \mapsto r^h$, and $s_1 \mapsto s_h$.

C. Reduction of subdivision iterations to soft e/i tests. Subdivision iterations are applied for root-finding in a square in the complex plane, called a *suspect square* and can be readily

[4] In computer aided design a user might want to design a 3D object under some fixed geometric constraints translated into a system of polynomials equations. This system can have an enormous number of solutions, mostly spurious, and the user can greatly benefit from restriction of the computations to a fixed Region of Interest.

extended to the union of suspect squares covering a fixed Region of Interest $\mathbb{D}$. An iteration subdivides, that is, partitions, every suspect square into four congruent sub-squares and to each applies an e/i test – to decide whether the sub-square contains any root lying in the Region $\mathbb{D}$. If it does not, then the sub-square is discarded; otherwise it is called a suspect square and undergoes subdivision (see Fig. 1a).
Efficient e/i tests are known for discs rather than squares. Moreover, an e/i test is hard if a root lies on or near the boundary circle. *σ-soft e/i tests for discs and $\sigma > 1$* address these problems.



(a) Four roots marked by asterisks lie in suspect sub-squares. Empty sub-squares are discarded in subdivision steps.

(b) A subdivision iteration fast decreases the diameter of a component but does not decrease its distance to external roots.

Fig. 1: Overview of the subdivision iteration process and root localization.

Every subdivision iteration keeps every square containing a root and hence approximates every root in $\mathbb{S}$ by the center of a suspect square within its half-diameter, itself halved in every iteration and hence decreased by a factor of 2^u in u iterations. A root can lie in at most 4 suspect squares. If an initial suspect square has diameter 2^h, then in at most $4(b+h)m$ soft e/i tests the centers of at most $4m$ suspect squares approximate within $1/2^b$ all roots in the Region $\mathbb{S}$.

To see, how the factor of $b+h$ is further decreased in [25, 26], partition the union of the suspect squares processed at a subdivision iteration into connected components. [25, 26] prove that progress in root-finding is slow if and only if a component, said too be *compact*, stays unbroken in many consecutive iterations, but this can only occur if the component (i) is made up of at most four suspect squares and (ii) becomes d^ν-isolated for a fixed constant $\nu > 0$ in $O(\log(m))$ iterations (see why so in Fig. 1b) and. [(i) A component with more than four suspect squares contains some roots (a) separated by at least the length λ of a current suspect square and (b) connected by a chain of m suspect squares; in $O(\log(m))$ iterations, however, the overall length of such a chain decreases by a factor of m and is exceeded by λ. (ii)A component is separated from it external roots by at least the length of its square. Subdivision steps do not decrease this separation while fast decreasing the diameter of a compact component.]

[25, 26] compress the minimal covering disc (MCD) of such a strongly isolated compact component (with m roots) into a sub-disc containing the same root set and with diameter minimal up to a bounded factor. [25, 26] test concentric sub-discs for containing m roots and approximate the minimal radius of such discs by means of bisection of their exponents. They verify if a sub-disc contains m roots by using 67 soft e/i tests for discs (see, e.g., [25, Sec 4.2 and Figs. 5 and 6]); our single soft e/i test of Sec. 2.6 for the ring (sharing its boundary circles with the MCD and the tested sub-disc) verifies this at the cost of two e/i tests for discs.