Beyond Worst-Case Analysis for Root Isolation Algorithms

Alperen Ergür
alperen.ergur@utsa.edu
The Univ. of Texas at San Antonio
San Antonio, Texas, USA

Josué Tonelli-Cueto
josue.tonelli.cueto@bizkaia.eu
Inria Paris & IMJ-PRG
Paris, France

Elias Tsigaridas
elias.tsigaridas@inria.fr
Inria Paris & Sorbonne Université
Paris, France

ABSTRACT

Isolating the real roots of univariate polynomials is a fundamental problem in symbolic computation and it is arguably one of the most important problems in computational mathematics. The problem has a long history decorated with numerous ingenious algorithms and furnishes an active area of research. However, the worst-case analysis of root-finding algorithms does not correlate with their practical performance. We develop a smoothed analysis framework for polynomials with integer coefficients to bridge the gap between the complexity estimates and the practical performance. In this setting, we derive that the expected bit complexity of Descartes solver to isolate the real roots of a polynomial, with coefficients uniformly distributed, is $\widetilde{O}_B(d^2 + d\tau)$, where d is the degree of the polynomial and τ the bitsize of the coefficients.

CCS CONCEPTS

• **Theory of computation → Numeric approximation algorithms**; *Randomness, geometry and discrete structures*; *Complexity theory and logic*; • **Computing methodologies → Symbolic and algebraic algorithms**.

KEYWORDS

univariate polynomials, root-finding, Descartes solver, condition-based complexity, average complexity, beyond worst-case analysis

ACM Reference Format:
Alperen Ergür, Josué Tonelli-Cueto, and Elias Tsigaridas. 2022. Beyond Worst-Case Analysis for Root Isolation Algorithms. In *Proceedings of the 2022 Int'l Symposium on Symbolic and Algebraic Computation (ISSAC '22), July 4–7, 2022, Villeneuve-d'Ascq, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3476446.3535475

ACM ISBN 978-1-4503-8688-3/22/07…$15.00
https://doi.org/10.1145/3476446.3535475

1 INTRODUCTION

The interactions between the ways we design and the ways we analyze algorithms are transformative on both ends: Unreasonably effective algorithms transform our complexity analysis frameworks, where else the discovery of essential complexity parameters transforms the ways we design algorithms. In numerical computation, the use of the condition numbers illustrate emphatically this phenomenon: condition numbers are a way of explaining the success of certain numerical algorithms[1], a guiding complexity parameter for the design of new algorithms, and a foundation for average and smoothed analysis of numerical algorithms [5, 6]. In discrete computation, this two-sided interaction between complexity analysis frameworks and algorithms' design forms a dynamic and exciting area of current research [11, 37] with a rich history rooted at the beginnings of complexity theory [2, Ch. 18]. Inspired by these developments, we aim to take a first step for bringing different modalities of algorithmic analysis into symbolic computation. To the best of our knowledge, this large field almost entirely relies on the worst-case analysis.

We consider one of the most basic problems in symbolic computation: computing the roots of univariate polynomials. This is a singularly important problem with applications in the whole range of computer science and engineering. It is extensively studied from theoretical and practical perspectives for decades and keeps attracting plenty of attention [18, 28, 32, 35]. We focus on the *real root isolation* problem: to compute intervals with rational endpoints that contain only one real root of the polynomial and each real root is contained in an interval. Besides its countless direct applications, this problem is omnipresent in symbolic computation; among its numerous uses it stands out as a crucial subroutine for elimination based multivariate polynomial systems solvers, e.g., [18].

Despite the ubiquity of real root isolation in engineering and its relatively long history in theoretical computer science, the state-of-the-art complexity analysis falls short of providing guidance for practical computations. Pan's algorithm [34] has the best worst-case complexity since nearly two decades and is colloquially referred to as the "optimal" algorithm. However, Pan's algorithm is rather sophisticated and has only a prototype implementation in PARI/GP [46]. In contrast, other algorithms with inferior worst-case complexity estimates have excellent practical performance, e.g., [21, 25, 51]. In our view, this lasting discrepancy between theoretical complexity analyses and practical performance is related to the insistence on using the worst-case framework in the symbolic computation community. However, let us mention the exceptions of [16], that provides estimates for the expected complexity of sturm algorithm for real solving, [51], that provides (conditional) expected case

[1] According to Wilkinson [54], Turing [52] introduced condition numbers to explain the practical success of Gaussian elimination despite the existing worst-case analyses.

bounds for the continued fraction algorthm, and [36] that considers the expected number steps for the problem of real root refinement.

We demonstrate how average/smoothed analysis frameworks can help to predict the practical performance of symbolic real root isolation algorithms. In particular, we show that in our data model the DESCARTES solver has a bit complexity quasi-linear in the input size, when we consider the dense representation. This provides an explanation for the excellent practical performance of DESCARTES that even outperforms its numerical alternatives. See §1.2 for a simple statement and §1.3 for the full technical statement.

1.1 Synopsis of real root isolation algorithms

We can (roughly) characterize the various algorithms for (real) root isolation as numerical or symbolic algorithms; the recent years there are also efforts to combine the best of the two worlds.

The numerical algorithms are, in almost all the cases, iterative algorithms that approximate all the roots (real and complex) of a polynomial up to any desired precision. Their main common tool is (a variant of) a Newton operator. The algorithm with the best worst-case complexity due to Pan [34] is based on Schönhage's splitting circle divide-and-conquer technique [44]. It recursively factors the polynomial until we obtain linear factors that approximate, up to any desired precision, all the roots of the polynomial and it has nearly optimal arithmetic complexity. We can turn this algorithm, and also any other numerical algorithm, to an exact one, by approximating the roots up to the separation bound; that is the minimum distance between the roots. In this way Pan obtained the record worst case bit complexity bound $\widetilde{O}_B(d^2\tau)$ for a degree d polynomial with maximum coefficient bitsize τ [34]; see also [3, 24, 29]. Besides the algorithms already mentioned, there are also several seemingly practically efficient numerical algorithms, e.g., MPSOLVE [4] and eigensolve [20], that lack convergence guarantees and/or precise bit complexity bounds.

Regarding symbolic algorithms, the majority is subdivision-based. These algorithms mimic binary search. Given an initial interval that contains all (or some) of the real roots, they repeatedly subdivide it until we obtain intervals containing zero or one real root. Prominent representatives of this approach are STURM and DESCARTES. STURM depends on Sturm sequences to count *exactly* the number of distinct roots in an interval, even when the polynomial is not square-free. Its complexity is $\widetilde{O}_B(d^4\tau^2)$ [9, 12] and it is not so efficient in practice; the bottleneck seems to be the high cost of computing the Sturm sequence. DESCARTES is based on Descartes' rule of signs to bound the number of real roots of a polynomial in an interval. Its worst case complexity is $\widetilde{O}_B(d^4\tau^2)$ [14]. Even though its worst case bound is similar to STURM, the DESCARTES solver has excellent practical performance and it can routinely solve polynomials of degree several thousands [21, 23, 38, 50]. There are also other algorithms based on the continued fraction expansion of the real numbers [45, 51] and on point-wise evaluation [7, 43].

Let us also mention the bitstream version of DESCARTES [13], where we assume that there is an oracle that for each coefficient of the polynomial returns an approximation to any absolute error. This approach, by also incorporating several tools from numerical algorithms, leads to improved variants of DESCARTES [42]. In the end, this variant yields the record worst case complexity bounds and efficient implementation [25] especially when there are clusters of roots. Even more, there is also a subdivision algorithm [3] that applies several improvements to the modified Weyl algorithm by Pan [33] and achieves the (record) complexity bound $\widetilde{O}_B(d^3 + d^2\tau)$.

1.2 Warm-up: A simple form of the main result

The main complexity parameters for univariate polynomials with integer (or rational) coefficients is the degree d and the bitsize τ; the latter refers to the maximum bitsize of the coefficients. We aim for a data model that resembles a "typical" polynomial with exact coefficients. The first natural candidate is the following: fix a bitsize τ, let $\mathfrak{c}_0, \mathfrak{c}_1, \dots, \mathfrak{c}_d$ be independent copies of the uniformly distributed integer in $[-2^\tau, 2^\tau] \cap \mathbb{Z}$, and let $\mathfrak{f} = \sum_{i=0}^{d} \mathfrak{c}_i X^i$ which we call the *uniform random bit polynomial with bitsize* τ. Recall that O, resp. O_B, denote the arithmetic, resp. bit, complexity and that we use $\widetilde{O}$, resp. $\widetilde{O}_B$, to ignore (poly-)logarithmic factors of d. For uniform random bit polynomials, our result has the following form.

THEOREM 1.1. *For a degree d uniform random bit polynomial $\mathfrak{f}$ with bit size $\tau(\mathfrak{f})$, DESCARTES solver isolates the real roots of $\mathfrak{f}$ in expected time $\widetilde{O}_B(d\,\tau + d^2)$.*

Notice that the expected time complexity of DESCARTES solver in this simple model is better by a factor of d than the record worst-case complexity bound of Pan's algorithm.

1.3 Statement of main results in full detail

We develop a general model of randomness that provides the framework of smoothed analysis for polynomials with integer coefficients.

Definition 1.2. Let $d \in \mathbb{N}$. A *random bit polynomial with degree* d is a random polynomial $\mathfrak{f} := \sum_{i=0}^{d} \mathfrak{c}_i X^i$, where the $\mathfrak{c}_i$ are independent discrete random variables with values in $\mathbb{Z}$. Then,

(1) the *bitsize of* $\mathfrak{f}$, $\tau(\mathfrak{f})$, is the minimum integer τ so that for all $i \in \{0, 1, 2, \dots, d\}$, $\mathbb{P}(|\mathfrak{c}_i| \leq 2^\tau) = 1$.
(2) the *weight of* $\mathfrak{f}$, $w(\mathfrak{f})$, is the maximum probability that $\mathfrak{c}_0$, $\mathfrak{c}_1$, $\mathfrak{c}_{d-1}$, and $\mathfrak{c}_d$ can take a value, i.e.,

$$w(\mathfrak{f}) := \max\{\mathbb{P}(\mathfrak{c}_i = k) \mid i \in \{0, 1, d-1, d\},\, k \in \mathbb{R}\}.$$

Remark 1.3. Note that we only impose restrictions on the size of the probabilities of the coefficients of 1, X, X^{d-1} and X^d. This might look odd at the first sight. We set our randomness model this way to be able to consider the most flexible data-model that can be handled by our proof techniques. We provide examples below to justify this technical assumption.

Example 1.4. The uniform random bit polynomial of bitsize τ we introduced is the main example of a random bit polynomial $\mathfrak{f}$. Note that in this case we have $w(\mathfrak{f}) = \frac{1}{1+2^{\tau+1}}$ and $\tau(\mathfrak{f}) = \tau$.

As we will see in the examples below, our randomness model is very flexible. However, this flexibility comes at a cost. In principle, we could have $w(\mathfrak{f}) = 1$ which would make our randomness model equivalent to the worst-case model. To control the effect of large $w(\mathfrak{f})$ we introduce the following quantity, which measures how far we are from a uniform random bit polynomial.

Definition 1.5. The *uniformity* of a random bit polynomial $\mathfrak{f}$ is

$$u(\mathfrak{f}) := \ln\left(w(\mathfrak{f})(1+2^{\tau(\mathfrak{f})+1})\right).$$

Remark 1.6. Note that $u(\mathfrak{f}) = 0$ if and only if the coefficients of 1, X, X^{d-1} and X^d in $\mathfrak{f}$ are uniformly distributed in $[-2^\tau, 2^\tau] \cap \mathbb{Z}$.

The following three examples illustrate the flexibility of our random model by specifying the support, the sign of the coefficients, and their exact bitsize. Although we specify them separately, any combination of specifications is also possible.

Example 1.7 (Support). Let $A \subseteq \{0, 1, \ldots, d-1, d\}$ with $0, 1, d-1, d \in A$. Then $\mathfrak{f} := \sum_{i \in A} \mathfrak{c}_i X^i$, where the $\mathfrak{c}_i$'s are independent and uniformly distributed in $[-2^\tau, 2^\tau]$ is a random bit polynomial with $u(\mathfrak{f}) = 0$ and $\tau(\mathfrak{f}) = \tau$.

Example 1.8 (Sign of the coefficients). Let $s \in \{-1, +1\}^{d+1}$. The random polynomial $\mathfrak{f} := \sum_{i=0}^{d} \mathfrak{c}_i X^i$, where the $\mathfrak{c}_i$'s are independent and uniformly distributed in $s_i([1, 2^\tau] \cap \mathbb{N})$, is a random bit polynomial with $u(\mathfrak{f}) \le \ln(3)$ and $\tau(\mathfrak{f}) = \tau$.

Example 1.9 (Exact bitsize). Let $\mathfrak{f} := \sum_{i=0}^{d} \mathfrak{c}_i X^i$ be the random polynomial, where the $\mathfrak{c}_i$'s are independent random integers of exact bitsize τ, i.e., $\mathfrak{c}_i$ is uniformly distributed in $\mathbb{Z} \cap ([-2^\tau + 1, -2^{\tau-1}] \cup [2^{\tau-1}, 2^\tau - 1])$. Then $\mathfrak{f}$ is a random bit polynomial with $u(\mathfrak{f}) \le \ln(3)$ and $\tau(\mathfrak{f}) = \tau$.

We consider a *smoothed random model* for polynomials, where a deterministic polynomial is perturbed by a random one. In this way our random bit polynomial model includes smoothed analysis over integer coefficients as a special case.

Example 1.10 (Smoothed analysis). Let $f \in \mathcal{P}_d$ be a fixed integer polynomial with coefficients in $[-2^\tau, 2^\tau]$, $\sigma \in \mathbb{Z} \setminus \{0\}$ and $\mathfrak{f} \in \mathcal{P}_d$ a random bit polynomial. Then $\mathfrak{f}_\sigma := f + \sigma\mathfrak{f}$ is a random bit-polynomial with bitsize $\tau(\mathfrak{f}_\sigma) \le \max\{\tau, \tau(\mathfrak{f}) + \tau(\sigma)\} + 1$, where $\tau(a)$ denotes the bitsize of a, and uniformity $u(\mathfrak{f}_\sigma) \le 1 + \max\{\tau - \tau(\mathfrak{f}), \tau(\sigma)\} + u(\mathfrak{f})$. By combining the smoothed random model with the previous examples, we can obtain structured perturbations.

Our main result is the following:

Theorem 1.11. *Let $\mathfrak{f}$ be random bit polynomial, of degree d, bitsize $\tau(\mathfrak{f})$, and uniformity parameter $u(\mathfrak{f})$, such that $\tau(\mathfrak{f}) = \Omega(\lg d + u(\mathfrak{f}))$, then* descartes *solver isolates the real roots of $\mathfrak{f}$ in expected time* $\widetilde{O}_B(d\,\tau\,(1+u(\mathfrak{f}))^3 + d^2\,(1+u(\mathfrak{f}))^4)$.

Remark 1.12. Note that if $\mathfrak{f}$ is not square-free, descartes will compute its square-free part and then proceed as usual. The probabilistic complexity estimate covers this case.

Remark 1.13. One might further optimize the probabilistic estimates present in Section 2.3 by employing strong tools from Littlewood-Offord theory [39]. However, the complexity analysis depends on the random variables in a logarithmic scale and so further improvements on probabilistic estimates will not make any essential improvement on our main result. Therefore, we prefer to use more transparent proofs with slightly less optimal dependency on the uniformity parameter $u(\mathfrak{f})$.

1.4 Overview of main ideas

The important quantities in analyzing descartes are the *separation bound* and the number of complex roots nearby the real axis.

The separation bound is the minimum distance between the distinct (complex) roots of a polynomial. [15]. This quantity controls the depth of the subdivision tree of descartes. To estimate this quantity we use condition numbers [5, 6, 10], following [49]. In short, we use condition numbers to obtain an instance-based estimate for the depth of the subdivision tree of descartes. Even though the descartes algorithm isolates the real roots, complex roots near the real axis control the width of the subdivision tree. This fact follows from the work of Obreshkoff [31], see also [26]. To estimate the number of roots in the Obreshkoff areas we use complex analytic techniques. In short, we bound the number of complex roots in a certain region to obtain an instance-based estimate for the width of the subdivision tree of descartes. Overall, by controlling both the depth—through the condition number—and the width—through the number of complex roots—we estimate the size of the subsdivision tree of descartes and so its bit complexity.

Finally, we perform the expected/smoothed analysis of the algorithm descartes by performing probabilistic analyses of the number of complex roots and the condition number. Expected/smoothed analysis results in computational algebraic geometry are rare and mostly restricted to continuous random variables, with few exceptions [8]; see also [16, 36, 51]. To the best of our knowledge, we present the first known result for the expected complexity of root finding for random polynomials with integer coefficients. Our results rely on the strong toolbox developed by Rudelson, Vershynin, and others in random matrix theory [27, 40].

Organization. We treat in detail condition numbers, separation bounds and their probabilistic estimates in Section 2, we deal with the estimates of the number of complex roots in Section 3, and we show how these quantities control the complexity of descartes obtaining the final complexity estimate in Section 4.

Notation. We denote by O, resp. O_B, the arithmetic, resp. bit, complexity and we use $\widetilde{O}$, resp. $\widetilde{O}_B$, to ignore (poly-)logarithmic factors of d. We denote by $\mathcal{P}_d$ the space of univariate polynomials of degree at most d with real coefficients and by $\mathcal{P}_d^{\mathbb{Z}}$ the subset of integer polynomial. If $f = \sum_{k=0}^{d} f_k X^k \in \mathcal{P}_d^{\mathbb{Z}}$, then the bitsize of f is the maximum bitsize of its coefficients. The set of complex roots of f is $\mathcal{Z}(f)$. We denote by $\textsc{var}(f)$ the number of sign changes in the coefficient list. The *separation bound* of f, $\Delta(f)$ or Δ if f is clear from the context, is the minimum distance between the roots of f, see [9, 15, 19]. We denote by $\mathbb{D}$ the unit disc in the complex plane, by $\mathbb{D}(x, r)$ the disk $x + r\mathbb{D}$, and by I the interval $[-1, 1]$. For a real interval $J = (a, b)$, we consider $\mathtt{mid}(J) := \frac{a+b}{2}$ and $\mathtt{wid}(J) := b - a$. For a $n \in \mathbb{N}$, we use $[n]$ to signify the set $\{1, \ldots, n\}$ and $\mu(n) = O_B(n \lg n)$ for the complexity of multiplying two integers of bitsize n, where lg is the logarithm with base 2.

2 CONDITION NUMBERS, SEPARATION BOUNDS, AND RANDOMNESS

We use various condition numbers for univariate polynomials from [49], cf. [48], to control the separation bound of random polynomials. However, our probabilistic analysis differs from [49] because we consider discrete random coefficients.

2.1 Condition numbers for univ. polynomials

The *local condition number of* $f \in \mathcal{P}_d$ *at* $z \in \mathbb{D}$ [49] is

$$\mathsf{C}(f, z) := \frac{\|f\|_1}{\max\{|f(z)|, |f'(z)|/d\}}, \tag{2.1}$$

where $\|f\|_1 := \sum |f_a|$ is the *1-norm* of f. important for obtaining bit complexity results. The same definition using the ℓ_2-norm is standard in numerical analysis literature, e.g., [22].

We also define the *(real) global condition number of* f as

$$\mathsf{C}_{\mathbb{R}}(f) := \max_{x \in I} \mathsf{C}(f, x). \tag{2.2}$$

We note that as $\mathsf{C}_{\mathbb{R}}(f)$ becomes bigger, f is closer to have a singular real zero inside I. This can be made precise through the so-called condition number theorem (see [49, Theorem 4.4]). There are many interesting properties of $\mathsf{C}_{\mathbb{R}}(f)$, but let us state the only one we will use—see [49, Theorem 4.2] for more.

Theorem 2.1 (2nd Lipschitz property). *[49] Let* $f \in \mathcal{P}_d$*. The map* $\mathbb{D} \ni z \mapsto 1/\mathsf{C}(f, z) \in [0, 1]$ *is well-defined and d-Lipschitz.* □

2.2 Condition-based estimates for separation

The quantity that follows is the separation bound of polynomials and polynomial systems, e.g., [15], suitably adjusted in our setting. This quantity and its condition-based estimate below will play a fundamental role in our complexity estimates.

Definition 2.2. For $\varepsilon \in \left[0, \frac{1}{d}\right)$ we set $I_\varepsilon := \{z \in \mathbb{C} \mid \operatorname{dist}(z, I) \leq \varepsilon\}$. If $f \in \mathcal{P}_d$, then the *ε-real separation of* f, $\Delta_\varepsilon^{\mathbb{R}}(f)$, is

$$\Delta_\varepsilon^{\mathbb{R}}(f) := \min\left\{\left|\zeta - \tilde{\zeta}\right| \mid \zeta, \tilde{\zeta} \in I_\varepsilon,\ f(\zeta) = f(\tilde{\zeta}) = 0\right\},$$

if f has no double roots in I_ε, and $\Delta_\varepsilon^{\mathbb{R}}(f) := 0$ otherwise.

Theorem 2.3 ([49, Theorem 6.3]). *Let* $f \in \mathcal{P}_d$ *and assume* $\varepsilon \in \left[0, \frac{1}{\mathrm{e}d\,\mathsf{C}_{\mathbb{R}}(f)}\right)$*, then* $\Delta_\varepsilon^{\mathbb{R}}(f) \geq \frac{1}{12d\,\mathsf{C}_{\mathbb{R}}(f)}$. □

2.3 Probabilistic bounds for condition numbers

In this section we present our probabilistic framework. The main technical tools are the anti-concentration results by Rudelson and Vershynin [40]. We do not apply these results as a black box, but we develop suitable variants for our setting (Proposition 2.7).

Theorem 2.4. *Let* $\mathfrak{f} \in \mathcal{P}_d^{\mathbb{Z}}$ *be a random bit polynomial and* $x \in I$*. Then, for* $t \leq 2^{\tau(\mathfrak{f})}$, $\mathbb{P}(\mathsf{C}(\mathfrak{f}, x) \geq t) \leq 16\, d^3 \mathrm{e}^{2u(\mathfrak{f})} \frac{1}{t^2}$.

Theorem 2.5. *Let* $\mathfrak{f} \in \mathcal{P}_d^{\mathbb{Z}}$ *be a random bit polynomial. Then, for* $t \leq 2^{\tau(\mathfrak{f})+1}$,

$$\mathbb{P}(\mathsf{C}_{\mathbb{R}}(\mathfrak{f}) \geq t) \leq 32\, d^4 \mathrm{e}^{2u(\mathfrak{f})} \frac{1}{t}.$$

The following corollary looks somewhat different than Thm. 2.4 and Thm. 2.5, but it has the same essence. Unlike the continuous case, in the discrete case we have a worst-case estimate that we can exploit to bound when the condition number is too large.

Corollary 2.6. *Let* $\mathfrak{f} \in \mathcal{P}_d^{\mathbb{Z}}$ *be a random bit polynomial,* $\ell \in \mathbb{N}$ *and* $c \geq 1$*. If* $\tau(\mathfrak{f}) \geq 4\ln(ed) + 2u(\mathfrak{f})$*, then* $\left(\mathbb{E}_{\mathfrak{f}}\left(\min\{\ln \mathsf{C}_{\mathbb{R}}(\mathfrak{f}), c\}\right)^\ell\right)^{\frac{1}{\ell}}$ *is at most*

$$(\ell + 1)\,(4\ln(ed) + 2u(\mathfrak{f})) + \left(\frac{16\, d^4 \mathrm{e}^{u(\mathfrak{f})}}{2^{\tau(\mathfrak{f})}}\right)^{\frac{1}{\ell}} c.$$

In particular, if $\tau(\mathfrak{f}) \geq 4\ln(ed) + 2u(\mathfrak{f}) + 2\ell \ln c$*, then*

$$\left(\mathbb{E}_{\mathfrak{f}}\left(\min\{\ln \mathsf{C}_{\mathbb{R}}(\mathfrak{f}), c\}\right)^\ell\right)^{\frac{1}{\ell}} \leq O(\ell(\ln d + u(\mathfrak{f}))).$$

We would like to understand the limitations of the two theorems and the corollary above. First, note that Theorem 2.4 is meaningful when $\tau(\mathfrak{f}) \geq 2 + \frac{3}{2}\lg(d) + 2u(\mathfrak{f})$ and Theorem 2.5 is meaningful when $\tau(\mathfrak{f}) \geq 5 + 4\lg(2) + 3u(\mathfrak{f})$. Intuitively, the randomness model needs some wiggling room to differ from the worst-case analysis. In our case this translates to assume that the bit-size $\tau(\mathfrak{f})$ is bigger than (roughly) $\lg(d) + u(\mathfrak{f})$. This is a reasonable assumption because for most cases of interest, $u(\mathfrak{f})$ is bounded above by a constant. Thus, the second condition in Corollary 2.6 becomes

$$\tau(\mathfrak{f}) = \Omega(\ell \lg(d) + \lg(c)).$$

Moreover, in the case of application of Corollary 2.6, we will have $c = d^{O(1)}$. Hence we are only imposing that the bit-size $\tau(\mathfrak{f})$ is lower bounded by (roughly) $\ln d$, which is not uncommon in practice.

For proving the above results, we need the following proposition. Recall that for $A \in \mathbb{R}^{k \times N}$, $\|A\|_{\infty,\infty} := \sup_{v \neq 0} \frac{\|Av\|_\infty}{\|v\|_\infty} = \max_{i \in k} \|A^i\|_1$, where A^i is the i-th row of A.

Proposition 2.7. *Let* $\mathfrak{x} \in \mathbb{Z}^N$ *be a random vector with independent coordinates. Assume that there is* $w > 0$ *so that for all* i *and* $x \in \mathbb{Z}$, $\mathbb{P}(\mathfrak{x}_i = x) \leq w$*. Then for every linear map* $A \in \mathbb{R}^{k \times N}$, $b \in \mathbb{R}^k$ *and* $\varepsilon \in [\|A\|_{\infty,\infty}, \infty)$,

$$\mathbb{P}(\|A\mathfrak{x} + b\|_\infty \leq \varepsilon) \leq 2\frac{(2\sqrt{2}w\varepsilon)^k}{\sqrt{\det AA^*}}.$$

Proof of Theorem 2.4. $\mathbb{P}(\mathsf{C}(\mathfrak{f}, x) \geq t)$ equals

$$\sum_{a_2, \dots, a_{d-2}} \mathbb{P}(\mathsf{C}(\mathfrak{f}, x) \geq t \mid \mathfrak{c}_2 = a_2, \dots, \mathfrak{c}_{d-2} = a_{d-2}) \prod_{i=2}^{d-2} \mathbb{P}(\mathfrak{c}_i = a_i).$$

where $\mathfrak{f} = \sum_{k=0}^d \mathfrak{c}_k X^k$. So it is enough to prove the bound for a random bit polynomial $\mathfrak{f}$ of the form

$$\mathfrak{f} = \mathfrak{c}_0 + \mathfrak{c}_1 X + \sum\nolimits_{k=2}^{d-2} a_k X^k + \mathfrak{c}_{d-1} X^{d-1} + \mathfrak{c}_d X^d,$$

where $a_2, \dots, a_{d-2} \in \mathbb{Z} \cap [-2^\tau, 2^\tau]$ are arbitrary fixed integers.

Let $\mathcal{P}_d(a_2, \dots, a_{d-2})$ be the affine subspace of $\mathcal{P}_d$ given by the equations $f_k = a_k$ for $k \in \{2, \dots, d-2\}$. And let

$$f \mapsto Af + b$$

be the affine mapping given by

$$\mathcal{P}_d(a_2, \dots, a_{d-2}) \ni f \mapsto (f(x), f'(x)/d) \in \mathbb{R}^2.$$

In the coordinates we are working on (those of the base $\{1, X, X^{d-1}, X^d\}$), A has the form

$$\begin{pmatrix} 1 & x & x^{d-1} & x^d \\ 0 & 1/d & (1 - 1/d)x^{d-2} & x^{d-1} \end{pmatrix}.$$

So, by an elementary estimation we have $\|A\|_{\infty,\infty} \le d+1$, and as a direct result of Cauchy-Binet formula we have $\sqrt{\det AA^*} \ge 1/d$. Now, since $\|\mathfrak{f}\|_1 \le (d+1)2^{\tau(\mathfrak{f})}$, we have that $\mathbb{P}(\mathsf{C}(\mathfrak{f},x) \ge t) = \mathbb{P}(\|A\mathfrak{f}+b\|_\infty \le \|\mathfrak{f}\|_1/t) \le \mathbb{P}(\|A\mathfrak{f}+b\|_\infty \le (d+1)2^{\tau(\mathfrak{f})}/t)$.

To be able to use Proposition 2.7, we need to assume $\frac{(d+1)2^{\tau(\mathfrak{f})}}{t} \ge d+1 \ge \|A\|_{\infty,\infty}$. Then, for $t \le 2^{\tau(\mathfrak{f})}$, Proposition 2.7 implies

$$\mathbb{P}(\mathsf{C}(\mathfrak{f},x) \ge t) \le 16d(d+1)^2(w2^{\tau(\mathfrak{f})}/t)^2.$$

Thus the proof is completed by the definition of $u(\mathfrak{f})$. □

Proof of Theorem 2.5. We will use a covering/union bound argument. For any finite set $\mathcal{G} \subset [-1,1]$ such that $\{[x-\delta, x+\delta] \mid x \in \mathcal{G}\}$ covers $[-1,1]$, using the 2nd Lipschitz property (Theorem 2.1), we have $1/\max_{x\in\mathcal{G}} \mathsf{C}(f,x) \le 1/\mathsf{C}_{\mathbb{R}}(f) + d\delta$. Let $\delta = 1/dt$, then $\mathbb{P}(\mathsf{C}_{\mathbb{R}}(\mathfrak{f}) \ge t) \le \mathbb{P}\left(\max_{x\in\mathcal{G}} \mathsf{C}_{\mathbb{R}}(\mathfrak{f},x) \ge t/2\right) \le \#\mathcal{G} \max_{x\in[-1,1]} \mathbb{P}(\mathsf{C}(\mathfrak{f},x) \ge t/2)$. We can construct such $\mathcal{G}$ such that $\#\mathcal{G} \le 2dt$. Hence the claim follows from Theorem 2.4. □

Proof of Corollary 2.6. Let

$$U := \ln(32\, d^4 \mathrm{e}^{2u(\mathfrak{f})}) \le 4\ln(ed) + 2u(\mathfrak{f}) \text{ and } V := \ln(2^{\tau(\mathfrak{f})+1}).$$

By assumption, $U \le V$ and $U > 1$, since $u(\mathfrak{f}) \ge 0$. So without loss of generality, we assume $0 < U < V < c$. If $c \le V$, then similar arguments imply that the claimed upper bound still holds. Thus

$$\mathbb{E}_{\mathfrak{f}}\left(\min\{\ln \mathsf{C}_{\mathbb{R}}(\mathfrak{f}), c\}\right)^\ell = \int_0^c \ell s^{\ell-1}\mathbb{P}(\min\{\ln \mathsf{C}_{\mathbb{R}}(\mathfrak{f}), c\} \ge s)\,\mathrm{d}s.$$

We divide the integral into three summands using the intervals $[0,U]$, $[U,V]$ and $[V,c]$.

In $[0,U]$, we have that $\mathbb{P}(\min\{\ln \mathsf{C}_{\mathbb{R}}(\mathfrak{f}), c\} \ge s) \le 1$, and so

$$\int_0^U \ell s^{\ell-1}\mathbb{P}(\min\{\ln \mathsf{C}_{\mathbb{R}}(\mathfrak{f}), c\} \ge s)\,\mathrm{d}s \le U^\ell.$$

In $[U,V]$, by Theorem 2.5 we have that

$$\mathbb{P}(\min\{\ln \mathsf{C}_{\mathbb{R}}(\mathfrak{f}), c\} \ge s) \le \mathbb{P}(\ln \mathsf{C}_{\mathbb{R}}(\mathfrak{f}) \ge s) \le e^{U-s},$$

and so the integral $\int_U^V \ell s^{\ell-1}\mathbb{P}(\min\{\ln \mathsf{C}_{\mathbb{R}}(\mathfrak{f}), c\} \ge s)\,\mathrm{d}s$ is bounded by $\int_U^V \ell s^{\ell-1}e^{U-s}\,\mathrm{d}s$. By performing a change of variables and extending the domain, we get $\int_0^\infty \ell(s+U)^{\ell-1}e^{-s}\,\mathrm{d}s$. The latter, expanding the binomial $(s+U)^{\ell-1}$ and using that $\Gamma(k+1) = k!$, is bounded by $\ell\sum_{k=0}^{\ell-1}\binom{\ell-1}{k}k!U^{\ell-1-k}$. Hence, as $\binom{\ell-1}{k}! \le \ell^{\ell-1}$, we get

$$\int_U^V \ell s^{\ell-1}\mathbb{P}(\min\{\ln \mathsf{C}_{\mathbb{R}}(\mathfrak{f}), c\} \ge s)\,\mathrm{d}s \le \ell^\ell U^{\ell-1}.$$

In $[V,c]$, we have that

$$\mathbb{P}(\min\{\ln \mathsf{C}_{\mathbb{R}}(\mathfrak{f}), c\} \ge s) \le \mathbb{P}(\ln \mathsf{C}_{\mathbb{R}}(\mathfrak{f}) \ge V) \le e^{U-V}.$$

Therefore, since $e^{U-V}\int_V^c \ell s^{\ell-1}\,\mathrm{d}s \le e^{U-V}\int_0^c \ell s^{\ell-1}\,\mathrm{d}s$,

$$\int_V^c \ell s^{\ell-1}\mathbb{P}(\min\{\ln \mathsf{C}_{\mathbb{R}}(\mathfrak{f}), c\} \ge s)\,\mathrm{d}s \le e^{U-V}c^\ell.$$

To obtain the final estimate, we add the three upper bounds obtaining the uper bound $U^\ell + \ell^\ell U^{\ell-1} + e^{U-V}c^\ell$. After substituting the values of U and V and some easy estimations, we conclude. □

Proof of Proposition 2.7. Let $\mathfrak{y} \in \mathbb{R}^N$ be such that the $\mathfrak{y}_i$ are independent and uniformly distributed in $(-1/2, 1/2)$. Now, a simple computation shows that $\mathfrak{x}+\mathfrak{y}$ is absolutely continuous and each component has density given by

$$\delta_{\mathfrak{x}_i+\mathfrak{y}_i}(t) = \sum\nolimits_{s\in\mathbb{Z}} \mathbb{P}(\mathfrak{x}_i = s)\delta_{\mathfrak{y}_i}(t-s).$$

Thus each component of $\mathfrak{x}+\mathfrak{y}$ has density bounded by w. We have

$$\mathbb{P}(\|A\mathfrak{x}+b\|_\infty \le \varepsilon) \le \mathbb{P}(\|A(\mathfrak{x}+\mathfrak{y})+b\|_\infty \le 2\varepsilon)/\mathbb{P}(\|A\mathfrak{y}\|_\infty \le \varepsilon),$$

since $\mathfrak{x}$ and $\mathfrak{y}$ are independent, and by the triangle inequality.

On the one hand, we apply [48, Proposition 5.2] (which is nothing more than [40, Theorem 1.1] with the explicit constants of [27]). The latter states that for a random vector $\mathfrak{z} \in \mathbb{R}^N$ with independent coordinates with density bounded by ρ and $A \in \mathbb{R}^{k\times N}$, we have that $A\mathfrak{z}$ has density bounded by $(\sqrt{2}\rho)^k/\sqrt{\det AA^*}$. Thus

$$\mathbb{P}(\|A(\mathfrak{x}+\mathfrak{y})+b\|_\infty \le 2\varepsilon) \le (2\sqrt{2}w\varepsilon)^k/\sqrt{\det AA^*}.$$

On the other hand,

$$\mathbb{P}(\|A\mathfrak{y}\|_\infty \le \varepsilon) = 1 - \mathbb{P}(\|A\mathfrak{y}\|_\infty \ge \varepsilon) \ge 1 - \mathbb{E}\|A\mathfrak{y}\|_\infty/\varepsilon.$$

by Markov's inequality. Now, by our assumption on ε, we only need to show that $\mathbb{E}\|A\mathfrak{y}\|_\infty \le \|A\|_{\infty,\infty}/2$.

By Jensen's inequality,

$$\mathbb{E}\|A\mathfrak{y}\|_\infty = \mathbb{E}\lim_{\ell\to\infty}\|A\mathfrak{y}\|_{2\ell} \le \lim_{\ell\to\infty}\left(\mathbb{E}\|A\mathfrak{y}\|_{2\ell}^{2\ell}\right)^{\frac{1}{2\ell}}.$$

Expanding the interior and computing the moments of $\mathfrak{y}$, we obtain

$$\mathbb{E}\|A\mathfrak{y}\|_\infty \le \lim_{\ell\to\infty}\left(\sum_{i=1}^k\sum_{|\alpha|=\ell}\binom{2\ell}{2\alpha}\prod_{j=1}^n\left(A_{i,j}^{2\alpha_j}(1/2)^{2\alpha_j}/(2\alpha_j+1)\right)\right)^{\frac{1}{2\ell}},$$

since the odd moments disappear. Thus

$$\mathbb{E}\|A\mathfrak{y}\|_\infty \le \frac{1}{2}\lim_{\ell\to\infty}\left(\sum_{i=1}^k\sum_{|\alpha|=2\ell}\binom{2\ell}{\alpha}\prod_{j=1}^n\left(|A_{i,j}|^{\alpha_j}\right)\right)^{\frac{1}{2\ell}} = \frac{\|A\|_{\infty,\infty}}{2},$$

where we obtained the bound of $\|A\|_{\infty,\infty}/2$ after doing the binomial sum and taking the limit. □

3 NUMBER OF COMPLEX ROOTS

To control the number of complex roots, we will use results from complex analysis and the probabilistic bounds from Section 2. Note that we cannot bound the number of complex roots inside $\mathbb{D}$, because the symmetry on our randomness model forces any bound on the number of roots in $\mathbb{D}$ to be of the form $O(d)$. For of this, we consider a family of disks $\{\mathbb{D}(\xi_{n,N}, \rho_{n,N})\}_{n=-N}^N$, inspired by the one in [30], where we will specify N in the sequel. In particular,

$$\xi_{n,N} = \begin{cases} \mathrm{sgn}(n)\left(1 - \frac{3}{4}\frac{1}{2^{|n|}}\right), & \text{if } |n| \le N-1 \\ \mathrm{sgn}(n)\left(1 - \frac{1}{2^N}\right), & \text{if } |n| = N \end{cases} \tag{3.1}$$

$$\rho_{n,N} = \begin{cases} \frac{3}{8}\frac{1}{2^{|n|}}, & \text{if } |n| \le N-1 \\ \frac{3}{2}\frac{1}{2^N}, & \text{if } |n| = N \end{cases}. \tag{3.2}$$

We will abuse notation and write ξ_n and ρ_n instead of $\xi_{n,N}$ and $\rho_{n,N}$ since we will not be working with different N's at the same time, but only with one N which might not have a prefixed value.

For this family of disks, we will give a deterministic and a probabilistic bound for the number of roots in their union, when $N = \lceil \lg d \rceil$,

$$\varrho(f) := \#\left\{ z \in \Omega_d := \bigcup_{n=-\lceil \log d \rceil}^{\lceil \log d \rceil} \mathbb{D}(\xi_n, \rho_n) \mid f(z) = 0 \right\}, \tag{3.3}$$

where $f \in \mathcal{P}_d$. We use these bounds to estimate the number of steps of DESCARTES(f).

3.1 Deterministic bound

THEOREM 3.1. *Let $f \in \mathcal{P}_d$. Then*

$$\varrho(f) \leq \sum_{n=-\lceil \log d \rceil}^{\lceil \log d \rceil} \log \frac{\mathrm{e}\|f\|_1}{|f(\xi_n)|}.$$

LEMMA 3.2. *Let $f \in \mathcal{P}_d$, $\xi \in \mathbb{D}$, and $\rho > 0$. If $|\xi| + 2\rho < 1 + 1/d$, then $\#(\mathcal{Z}(f) \cap \mathbb{D}(\xi, \rho)) \leq \log(\mathrm{e}\|f\|_1 / |f(\xi)|)$.*

PROOF OF THEOREM 3.1. We only have to apply subadditivity and Lemma 3.2. Note that the condition of the Lemma 3.2 holds for every disk $\mathbb{D}(\xi_n, \rho_n)$ in Ω_d. □

PROOF OF LEMMA 3.2. We use a classic result of Titchmarsh [47, p. 171] that bounds the number of roots in a disk. For $\delta \in (0, 1)$, we have that $\#(\mathcal{Z}(f) \cap \mathbb{D}(\xi, \rho)) \leq (\ln(1/\delta))^{-1} \ln(\max_{z \in \mathbb{D}} |f(\xi + \rho z/\delta)| / |f(\xi)|)$.

Take $\delta = 1/2$. By our assumption, $\xi + 2\rho\mathbb{D} \in (1 + 1/d)\mathbb{D}$, so $\max_{z \in \mathbb{D}} |f(\xi + \rho z/\delta)| \leq \max_{z \in (1+1/d)\mathbb{D}} |f(z)| \leq \mathrm{e}\|f\|_1$, since $|f(z)| \leq \mathrm{e}\|f\|_1$, for $z \in (1 + 1/d)\mathbb{D}$ [49, Proposition 3.9.]. □

3.2 Probabilistic bound

THEOREM 3.3. *Let $\mathfrak{f} \in \mathcal{P}_d^{\mathbb{Z}}$ be a random bit polynomial. Then for all $t \leq \tau(\mathfrak{f})(2\lceil \lg d \rceil + 1)$,*

$$\mathbb{P}(\varrho(\mathfrak{f}) \geq t) \leq 44 d^2 (2\lceil \lg d \rceil + 1) \mathrm{e}^{u(\mathfrak{f})} \mathrm{e}^{-\frac{t}{2\lceil \lg d \rceil + 1}}.$$

COROLLARY 3.4. *Let $\mathfrak{f} \in \mathcal{P}_d^{\mathbb{Z}}$ be a random bit polynomial and $\ell \in \mathbb{N}$. Suppose that $\tau(\mathfrak{f}) \geq 10 \ln(\mathrm{e}d) + 2u(\mathfrak{f})$. Then*

$$\left(\mathbb{E}\varrho(\mathfrak{f})^\ell\right)^{\frac{1}{\ell}} \leq 2(1 + \ell)(6 \ln(\mathrm{e}d) + u(\mathfrak{f})) \ln(\mathrm{e}d) + \left(\frac{44 d^{3+2\ell} \mathrm{e}^{u(\mathfrak{f})}}{2^{\tau(\mathfrak{f})}}\right)^{\frac{1}{\ell}}.$$

In particular, if $\tau(\mathfrak{f}) \geq (9 + 3\ell) \ln(\mathrm{e}d) + 2u(\mathfrak{f})$, then

$$\left(\mathbb{E}\varrho(\mathfrak{f})^\ell\right)^{\frac{1}{\ell}} \leq O\left(\ell(\ln d + u(\mathfrak{f})) \ln d\right).$$

PROOF OF THEOREM 3.3. If $\#(\mathcal{Z}(\mathfrak{f}) \cap \Omega_d) \geq t$, then, by Theorem 3.1, there is an n such that $\log(\mathrm{e}\|f\|_1 / |\mathfrak{f}(\xi_n)|) \geq t/(2\lceil \lg d \rceil + 1)$. Hence

$$\mathbb{P}(\varrho(\mathfrak{f}) \geq t) \leq \sum_{n=-\lceil \lg d \rceil}^{\lceil \lg d \rceil} \mathbb{P}\left(\lg \frac{\mathrm{e}\|f\|_1}{|\mathfrak{f}(\xi_n)|} \geq \frac{t}{2\lceil \lg d \rceil + 1}\right).$$

Now, fix $x \in I$. We argue as in the proof of Theorem 2.4, but we consider that map mapping f to $f(x)$ instead of the map mapping f to $(f(x), f'(x)/d)$, so that our matrix A takes the form

$$\begin{pmatrix} 1 & x & x^{d-1} & x^d \end{pmatrix}.$$

Note that this A has $\|A\|_{\infty,\infty} \leq d + 1$. So, we can apply Proposition 2.7 to show that for any $s \leq 2^{\tau(\mathfrak{f})}$,

$$\mathbb{P}(\mathrm{e}\|\mathfrak{f}\|_1 / |\mathfrak{f}(x)| \geq s) \leq 44 d^2 \mathrm{e}^{u(\mathfrak{f})} / s.$$

Algorithm 1: DESCARTES(f)

Input: A square-free polynomial $f \in \mathcal{P}_d^{\mathbb{Z}}$
Output: A list, S, of isolating intervals for the real roots of f in $J_0 = (-1, 1)$

$J_0 \leftarrow (-1, 1)$, $S \leftarrow \emptyset$, $Q \leftarrow \emptyset$, $Q \leftarrow$ PUSH(J_0)
while $Q \neq \emptyset$ **do**
  $J = (a, b) \leftarrow$ POP(Q) $V \leftarrow$ VAR(f, J)
  switch V **do**
    case $V = 0$ **continue**
    case $V = 1$ $S \leftarrow$ ADD(I)
    case $V > 1$
      $m \leftarrow \frac{a+b}{2}$
      if $f(m) = 0$ **then** $S \leftarrow$ ADD($[m, m]$)
      $J_L \leftarrow [a, m]$; $J_R \leftarrow [m, b]$
      $Q \leftarrow$ PUSH(Q, J_L), $Q \leftarrow$ PUSH(Q, J_R)
RETURN S

If $s = \mathrm{e}^{t/N}$, with $N = 2\lceil \lg(d) \rceil + 1$, then the bound follows. □

PROOF OF COROLLARY 3.4. In the proof of Corollary 2.6 we only used the fact that the tail bound is of the form $U\mathrm{e}^{-t}$ for $t \leq V$ with $U \leq V$. We will use a similar idea in this proof. Let $0 \leq U \leq V$, $c > 0$, and $\mathfrak{x} \in [0, \infty)$ a random variable. If $\mathbb{P}(\mathfrak{x} \geq t) \leq \mathrm{e}^{U-s}$ for $s \leq V$, then $\mathbb{E}(\min\{\mathfrak{x}, c\})^\ell \leq U^\ell + \ell^\ell U^{\ell-1} + \mathrm{e}^{U-V} c^\ell$.

By Theorem 3.3, the random variable $\varrho(\mathfrak{f})/(2\lceil \lg d \rceil + 1)$ satisfies the conditions to be a random variable $\mathfrak{x}$ with $U = \ln(44 d^2 (2\lceil \lg d \rceil + 1)\mathrm{e}^{u(\mathfrak{f})}) \leq 4 \ln(\mathrm{e}d) + \ln(2\lceil \lg d \rceil + 1) + u(\mathfrak{f})$, $V = \ln(2^{\tau(\mathfrak{f})}/(2\lceil \lg d \rceil + 1))$, and $c = \frac{d}{(2\lceil \lg d \rceil + 1)}$; since the roots are at most d. By our assumptions $U \leq V$, that concludes the proof. □

4 THE DESCARTES SOLVER

The DESCARTES solver is an algorithm that is based on Descartes' rule of signs.

THEOREM 4.1 (DESCARTES' RULE OF SIGNS). *The number of sign variations in the coefficients' list of a polynomial $f = \sum_{i=0}^d f_i X^i \in \mathcal{P}_d$ equals the number of positive real roots (counting multiplicities) of f, say r, plus an even number; that is $r \equiv$ VAR(f) mod 2.* □

In general, Theorem 4.1 provides an overestimation on the number of positive real roots. It counts exactly when the number of sign variations is 0 or 1 and if the polynomial is hyperbolic, that is it has *only* real roots. To count the real roots of f in an interval $J = (a, b)$ we use the transformation $x \mapsto \frac{ax+b}{x+1}$ that maps J to $(0, \infty)$. Then

$$\mathrm{VAR}(f, J) := \mathrm{VAR}((X + 1)^d f(\tfrac{aX+b}{X+1}))$$

bounds the number of real roots of f in $I = J$.

Therefore, to isolate the real roots of f in an interval, say $J_0 = (-1, 1)$, we count (actually bound) the number of roots of f in J_0 using $V =$ VAR(f, J_0). If $V = 0$, then we discard the interval. If $V = 1$, then we add J_0 to the list of isolating intervals. If $V > 1$, then we subdivide the interval to two intervals J_L and J_R and we repeat the process. The pseudo-code of DESCARTES appears in Algorithm 1.

The recursive process of the DESCARTES defines a binary tree. Every node of the tree corresponds to an interval. The root corresponds to the initial interval $J_0 = (-1, 1)$. If a node corresponds to an interval $J = (a, b)$, then its children correspond to the open left and right half intervals of J, that is $J_L = (a, \mathtt{mid}(J))$ and $J_R = (\mathtt{mid}(J), b)$ respectively. The internal nodes of the tree correspond to intervals J, such that VAR(f, J) ≥ 2. The leafs correspond to intervals that contain 0 or 1 real roots of f. Overall, the number of nodes of the tree correspond to the number of steps, i.e., subdivisions, that the algorithm performs. We control the number of nodes by controlling the depth of tree and the width of every layer. Hence, to obtain the final complexity estimate it suffices to multiple the number of steps (width times height) with the worst case cost of each step.

The following proposition helps to control the cost of each step. Note that at each step, we do changes of variables to obtain the desired polynomial to perform the sign count.

Proposition 4.2. *Let $f = \sum_{i=0}^{d} f_i X^i \in \mathcal{P}_d^{\mathbb{Z}}$ of bit-size τ.*

- *The reciprocal transformation is $R(f) := X^d f(\frac{1}{X}) = \sum_{k=0}^{d} f_{d-k} X^k$. Its cost is $O_B(1)$ and it does not alter neither the degree nor the bit-size of the polynomial.*
- *The homothetic transformation of f by 2^k, for a positive integer k, is $H_k(f) = 2^{dk} f(\frac{X}{2^k}) = \sum_{i=0}^{d} 2^{k(d-i)} f_i X^i$. It costs $O_B(d\,\mu(\tau + dk)) = \widetilde{O}_B(d\tau + d^2k)$ and the resulting polynomial has bit-size $O(\tau + dk)$. Notice that $H_{-k} = RH_kR$.*
- *The Taylor shift of f by in integer c is $T_c(f) = f(x+c) = \sum_{k=0}^{d} a_k x^k$, where $a_i = \sum_{j=i}^{d} \binom{j}{i} f_j c^{j-i}$ for $0 \le i \le d$. It costs $O_B(\mu(d^2\sigma + d\tau)\lg d) = \widetilde{O}_B(d^2\sigma + d\tau)$ [53, Corollary 2.5], where σ is the bit-size of c. The resulting polynomial has bit-size $O(\tau + d\sigma)$.* □

Remark 4.3. There is no restriction on working with open intervals since we consider an integer polynomial and we can always evaluate it at the endpoints. Also to isolate all the real roots of f it suffices to have a routine to isolate the real roots in $(-1, 1)$. Using the map $x \mapsto 1/x$ we can isolate the roots in $(-\infty, -1)$ and $(1, \infty)$.

4.1 Bounds on the number of sign variations

For this subsection we consider $f = \sum_{i=0}^{d} f_i X^i \in \mathcal{P}_d$ to be a polynomial with real coefficients, not necessarily integers. To establish the termination and estimate the bit complexity of DESCARTES we need to introduce the Obreshkoff area and lens. Our presentation follows closely [17, 26, 42].

Consider $0 \le \varrho \le d$ and a real open interval $J = (a, b)$. The *Obreshkoff discs* $\overline{\mathcal{D}}_\varrho$ and $\underline{\mathcal{D}}_\varrho$ are discs the boundaries of which go through the endpoints of J. Their centers are above, respectively below, J and they form an angle $\varphi = \frac{\pi}{\varrho+2}$ with the endpoints of I. Its diameter is $\mathtt{wid}(J)/\sin(\frac{\pi}{\varrho+2})$.

The *Obreshkoff area* is $\mathcal{A}_\varrho(J) = \mathrm{interior}(\overline{\mathcal{D}}_\varrho \cup \underline{\mathcal{D}}_\varrho)$; it appears with grey color in Fig. 1. The *Obreshkoff lens* is $\mathcal{L}_\varrho(J) = \mathrm{interior}(\overline{\mathcal{D}}_\varrho \cap \underline{\mathcal{D}}_\varrho)$; it appears in light-grey color in Fig. 1. If it is clear from the context, then we omit I and we write $\mathcal{A}_\varrho$ and $\mathcal{L}_\varrho$, instead of $\mathcal{A}_\varrho(J)$ and $\mathcal{L}_\varrho(J)$. It holds that $\mathcal{L}_d \subset \mathcal{L}_{d-1} \subset \cdots \subset \mathcal{L}_1 \subset \mathcal{L}_0$ and $\mathcal{A}_0 \subset \mathcal{A}_1 \subset \cdots \subset \mathcal{A}_{d-1} \subset \mathcal{A}_d$.

The following theorem shows the role of the number of complex roots in the control of the number of variation signs.

Theorem 4.4 ([31]). *Consider $f \in \mathcal{P}_d$ and real open interval $J = (a, b)$. If the Obreshkoff lens $\mathcal{L}_{d-k}$ contains at least k roots (counted with multiplicity) of f, then $k \le \mathrm{VAR}(f, J)$. If the Obreshkoff area $\mathcal{A}_k$ contains at most k roots (counted with multiplicity) of f, then $\mathrm{VAR}(f, J) \le k$. Especially*

$$\#\{\text{roots of } f \text{ in } \mathcal{L}_d\} \le \mathrm{VAR}(f, J) \le \#\{\text{roots of } f \text{ in } \mathcal{A}_d\}. \quad \square$$

This theorem together with the subadditive property of Descartes' rule of signs (Thm. 4.5) shows that the number of complex roots in the Obreshkoff areas controls the width of the subdivision tree of DESCARTES.

Theorem 4.5. *Consider a real polynomial $f \in \mathcal{P}_d$. Let J be a real interval and $J_1, \ldots, J_n$ be disjoint open subintervals of J. Then, it holds $\sum_{i=1}^{n} \mathrm{VAR}(f, J_i) \le \mathrm{VAR}(f, J)$.* □

Finally, to control the depth of the subdivision tree of DESCARTES we use the one and two circle theorem [1, 26]. We present a variant based on the ε-real separation of f, $\Delta_\varepsilon^{\mathbb{R}}(f)$ (Definition 2.2).

Theorem 4.6. *Let $f \in \mathcal{P}_d$, an interval $J \subseteq (-1, 1)$ and $\varepsilon > 0$. If*

$$2\,\mathtt{wid}(J) \le \min\{\Delta_\varepsilon^{\mathbb{R}}(f), \varepsilon\},$$

then either $\mathrm{VAR}(f, J) = 0$ (and J does not contain any real root), or $\mathrm{VAR}(f, J) = 1$ (and J contains exactly one real root).

Proof. The proof follows the same application of the one and two circle theorems as in the proof of [49, Proposition 6.4]. □

4.2 Complexity estimates for DESCARTES

We give a high-level overview of the proof ideas of this section before going into technical details. The process of DESCARTES corresponds to a binary tree and we control its depth using the real condition number and Theorems 2.3 and 4.6. To bound the width of the DESCARTES' tree we use the Obreskoff areas and the number of complex roots in them (Theorem 4.4). By combining these two bounds, we control the size of the tree and so we obtain an instance-based complexity estimate. To turn this instance-based complexity estimate into an expected one, we use Theorems 2.5 and 3.3 (and their Corollaries 2.6 and 3.4).

4.2.1 *Instance-based estimates.*

Theorem 4.7. *If $f \in \mathcal{P}_d^{\mathbb{Z}}$, then, using DESCARTES, the number of subdivision steps to isolate the real roots in $I = (-1, 1)$ is*

$$\widetilde{O}(\varrho(f)^2 \lg(\mathsf{C}_{\mathbb{R}}(f))).$$

The bit complexity of the algorithm is

$$\widetilde{O}_B(d\tau\varrho(f)^2 \lg \mathsf{C}_{\mathbb{R}}(f) + d^2\varrho(f)^2 \lg^2 \mathsf{C}_{\mathbb{R}}(f)).$$

Recall that $\mathsf{C}_{\mathbb{R}}(f)$ appears in (2.2) and $\varrho(f)$ in (3.3).

Proof. We consider the number of steps to isolate the real roots in $I = (-1, 1)$. Let $N = \lceil \log d \rceil$ and $\varrho = \varrho(f)$ the number of complex roots in Ω_d. Recall that Ω_d is the union of the discs $D_n := \mathbb{D}(\xi_n, \rho_n) := \xi_n + \rho_n\mathbb{D}$, where $|n| \le N$; see (3.1) and (3.2) for the concrete formulas, and that it contains the interval I.

The discs partition I into the $2N+1$ subintervals $J_n := [\xi_n, \xi_{n+1}]$ (or $J_n := [\xi_n, \xi_{n-1}]$ if $n \le 0$). Note that J_n is the union of 3 intervals of size $1/2^{n+3}$. Because of this, there is a binary subdivision tree of I of size $O(\lg^2 d)$ such that every of its intervals is contained in some J_n. Thus, if we bound the width of the subdivision tree of DESCARTES starting at each J_n by w, then the width of the subdivision tree of DESCARTES starting at I is bounded by $O(w \lg^2 d + \lg^2 d)$.

We focus on intervals J_n for $n \ge 0$; similar arguments apply for $n \ge 0$. We consider two cases: $n < N$ and $n = N$.

Case $n < N$. It holds $\mathtt{wid}(J_n) = \rho_n = 3/2^{n+3}$. For each J_n, assume that we perform a number of subdivision steps to obtain intervals, say $J_{n,\ell}$, with $\mathtt{wid}(J_{n,\ell}) = 2^{-\ell}$. We choose ℓ so that the corresponding Obreshkoff areas, $\mathcal{A}_\varrho(J_{n,\ell})$, are inside Ω_d. In particular, we ensure that the Obreshkoff areas related to $J_{n,\ell}$ lie in D_{n+1}.

The diameter of the Obreshkoff discs, $\overline{\mathcal{D}}_\varrho(J_{n,\ell})$ and $\underline{\mathcal{D}}_\varrho(J_{n,\ell})$, is $\mathtt{wid}(J_{n,\ell})/\sin\frac{\pi}{\varrho+2}$. For every $\mathcal{A}_\varrho(J_{n,\ell})$ to be in D_{n+1} and hence inside Ω_d, it suffices that a disc with diameter $2\,\mathtt{wid}(J_{n,\ell})/\sin\frac{\pi}{\varrho+2}$, that has its center in the interval $[\xi_n, \xi_{n+1}]$ and touches the right endpoint of J_n, to be inside $D_{n+1} \setminus D_n$. This is the worst case scenario: a disc big enough that contains $\mathcal{A}_\varrho(J_{n,\ell})$ and lies D_{n+1}. This auxiliary disc is the dotted (red) disc in Fig. 2 (left). It should be that

$$2\,\mathtt{wid}(J_{n,\ell})/\sin\tfrac{\pi}{\varrho+2} \le 2\,\rho_{n+1} = 3/2^{n+3}.$$

Taking into account that $\mathtt{wid}(J_{n,\ell}) = 2^{-\ell}$ and

$$\sin\tfrac{\pi}{\varrho+2} > \sin\tfrac{1}{\varrho} \ge \tfrac{1}{\varrho}\Big/\sqrt{1+\tfrac{1}{\varrho^2}} \ge \tfrac{1}{2\varrho},$$

we deduce $2^{-\ell+1}2\varrho \le 3/2^{n+3}$ and so $\ell \ge \lg\frac{2^{n+5}\varrho}{3}$.

Hence, $\mathtt{wid}(J_{n,\ell}) = 3/(2^{n+5}\varrho)$ and so J_n is partitioned to at most $\frac{\mathtt{wid}(J_n)}{\mathtt{wid}(J_{n,\ell})} = 4\varrho$ (sub)intervals. So, during the subdivision process, starting from (each) J_n, we obtain the intervals $J_{n,\ell}$ after performing at most 8ϱ subdivision steps (this is the size of the complete binary tree starting from J_n). To say it differently, the subdivision tree that has J_n as its root and the intervals $J_{n,\ell}$ as leaves has depth $\ell = \lceil\lg(4\varrho)\rceil$. The same hold for J_{N-1} because $\rho_n \le \rho_N$, for all $0 \le n \le N-1$.

Thus, the width of the tree starting at J_n is at most $O(\varrho^2)$, because we have $O(\varrho)$ subintervals $J_{n,\ell}$ and for each $\textsc{var}(f, J_{n,\ell}) \le \varrho$.

Case $n = N$. Now $\mathtt{wid}(J_N) = 3/2^{N+1}$. We need a slightly different argument to account for the number of subdivision steps for the last disc D_N. To this disc we assign the interval $J_N = [1 - 1/2^N, 1]$ with $\mathtt{wid}(J_N) = 1/2^N$; see Figure 2.

We need to obtain small enough intervals $J_{N,\ell}$ of width $1/2^\ell$ so that corresponding Obreskoff areas, $\mathcal{A}_\varrho(J_{N,\ell})$, to be inside D_N. So, we require that an auxiliary disc of diameter $2\,\mathtt{wid}(J_{N,\ell})/\sin\frac{\pi}{\varrho+2}$, that has ts center in the interval $[1, 1/2^{N+1}]$ and touches 1 to be inside D_N; actually inside $D_N \cap \{x \ge 1\}$; see Figure 2. And so

$$2\,\mathtt{wid}(J_{N,\ell})/\sin\tfrac{\pi}{\varrho+2} \le \rho_{n+1} = 1/2^{N+1}.$$

This leads to $\ell \ge \lg(\varrho\, 2^{N+3})$. Working as previously, we estimate that the number of subdivisions we perform to obtain the interval $J_{N,\ell}$ is 8ϱ. Also repeating the previous arguments, the width of the tree of DESCARTES starting at J_N is at most $O(\varrho^2)$.

By combining all the previous estimates, we conclude that the subdivision tree of DESCARTES has width $O(\varrho^2 \lg^2 d + \lg^2 d)$.

To bound the depth of the subdivision tree of DESCARTES, consider an interval J_ℓ of width $1/2^\ell$ obtained after $\ell + 1$ subdivisions. By theorem 4.6, we can guarantee termination if for some $\varepsilon > 0$,

$$1/2^{\ell-1} \le \min\{\Delta_\varepsilon^{\mathbb{R}}(f), \varepsilon\}.$$

Fix $\varepsilon = 1/(\mathrm{e}d\,\mathsf{C}_\mathbb{R}(f))$. Then, by Theorem 2.3, it suffices to hold

$$\ell \ge 1 + \lg(12d\,\mathsf{C}_\mathbb{R}(f)).$$

Hence, the depth of the subdivision tree is at most $O(\lg(d\,\mathsf{C}_\mathbb{R}(f)))$.

Therefore, since the subdivision tree of DESCARTES has width $O(\varrho^2 \log d + \log^2 d)$ and depth $O(\lg(d\,\mathsf{C}_\mathbb{R}(f)))$, the size bound follows. For the bit complexity, by [14], see also [17, 26, 41, 42] and Proposition 4.2, the worst case cost of each step of DESCARTES is $\widetilde{O}_B(d\tau + d^2\delta)$, where δ is the logarithm of the highest bitsize that we compute with, or equivalently the depth of the subdivision tree. In our case, $\delta = O(\lg(d\,\mathsf{C}_\mathbb{R}(f))$. □

4.2.2 Expected complexity estimates.

THEOREM 4.8. *Let $\mathfrak{f} \in \mathcal{P}_d^\mathbb{Z}$ be a random bit polynomial with $\tau(\mathfrak{f}) \ge \Omega(\lg d + u(\mathfrak{f}))$. Then, using DESCARTES, the expected number of subdivision steps to isolate the real roots in $I = (-1, 1)$ is*

$$\widetilde{O}((1 + u(\mathfrak{f}))^3).$$

The expected bit complexity of DESCARTES is

$$\widetilde{O}_B(d\,\tau(\mathfrak{f})(1 + u(\mathfrak{f}))^3 + d^2(1 + u(\mathfrak{f}))^4).$$

If $\mathfrak{f}$ is a uniform random bit polynomial of bitsize τ and $\tau = \Omega(\lg d + u(\mathfrak{f}))$,en the expected number of subdivision steps to isolate the real rin $I = (-1, 1)$ is $\widetilde{O}(1)$ and the expected bit complexity becomes

$$\widetilde{O}_B(d\tau + d^2).$$

PROOF. We only bound the number of bit operations; the bound for the number of steps is analogous. By Theorem 4.7 and the worst-case bound $\widetilde{O}_B(d^4\tau^2)$ for DESCARTES [14], the bit complexity of DESCARTES at $\mathfrak{f}$ is at most

$$\widetilde{O}_B\left(\min\{d\tau(\mathfrak{f})\varrho(\mathfrak{f})^2 \lg \mathsf{C}_\mathbb{R}(\mathfrak{f}) + d^2\varrho(\mathfrak{f})^2 \lg^2 \mathsf{C}_\mathbb{R}(\mathfrak{f})), d^4\tau(\mathfrak{f})^2\}\right),$$

that in turn we can bound by

$$\widetilde{O}_B\Big(d\tau(\mathfrak{f})\varrho(\mathfrak{f})^2 \min\{\lg \mathsf{C}_\mathbb{R}(\mathfrak{f}), d^3\tau(\mathfrak{f})\} + d^2\varrho(\mathfrak{f})^2 \min\{\lg \mathsf{C}_\mathbb{R}(\mathfrak{f}), d^2\tau(\mathfrak{f})^2)\}\}\Big).$$

Now, we take expectations, and, by linearity, we only need to bound

$$\mathbb{E}\,\varrho(\mathfrak{f})^2 \min\{\lg \mathsf{C}_\mathbb{R}(\mathfrak{f}), d^3\tau(\mathfrak{f})\} \text{ and } \mathbb{E}\,\varrho(\mathfrak{f})^2 \left(\min\{\lg \mathsf{C}_\mathbb{R}(\mathfrak{f}), d^2\tau(\mathfrak{f})^2\}\right)^2.$$

Let us show how to bound the first, because the second one is the same. By the Cauchy-Bunyakovsky-Schwarz inequality,

$$\mathbb{E}\,\varrho(\mathfrak{f})^2 \min\{\lg \mathsf{C}_\mathbb{R}(\mathfrak{f}), d^3\tau(\mathfrak{f})\}$$

is bounded by

$$\sqrt{\mathbb{E}\,\varrho(\mathfrak{f})^4}\sqrt{\mathbb{E}\,\left(\min\{\lg \mathsf{C}_\mathbb{R}(\mathfrak{f}), d^3\tau(\mathfrak{f})\}\right)^2}.$$

Finally, Corollaries 2.6 and 3.4 give the estimate. Note that $\tau(\mathfrak{f}) \ge \Omega(\lg d + u(\mathfrak{f}))$ implies $\tau(\mathfrak{f}) \ge \Omega(\lg d + u(\mathfrak{f}) + \ln c)$ (for the worst-case separation bound c [9]) so we can apply Corollary 2.6. □

ACKNOWLEDGEMENTS.

J.T-C. is supported by a postdoctoral fellowship of the 2020 "Interaction" program of the Fondation Sciences Mathématiques de Paris. He is grateful to Evgenia Lagoda for moral support and Gato Suchen for useful suggestions regarding Proposition 2.7. A.E. is supported by NSF CCF 2110075, J.T-C. and E.T. are partially supported by ANR JCJC GALOP (ANR-17-CE40-0009).

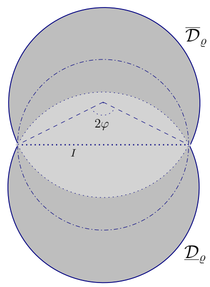

Figure 1 Obreshkoff discs, lens (light grey), and area (light grey and grey) for an interval I.

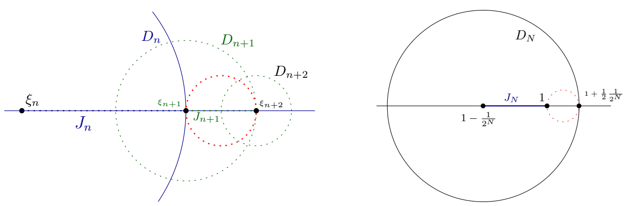

Figure 2 Covering discs of the interval $I = (0, 1)$.
(left) Three covering discs, D_n, D_{n+1}, D_{n+2}.
(right) The (red) dotted circle is the auxiliary disc that we ensure is contained in $D_{n+1} \setminus D_n$.

REFERENCES

[1] Alberto Alesina and Massimo Galuzzi. 1998. A new proof of Vincent's theorem. *Enseign. Math. (2)* 44, 3-4 (1998), 219–256.

[2] S. Arora and B. Barak. 2009. *Computational complexity: a modern approach.* Cambridge University Press, Cambridge. xxiv+579 pages. https://doi.org/10.1017/CBO9780511804090

[3] Ruben Becker, Michael Sagraloff, Vikram Sharma, and Chee Yap. 2018. A near-optimal subdivision algorithm for complex root isolation based on the Pellet test and Newton iteration. *J. Symbolic Comput.* 86 (2018), 51–96. https://doi.org/10.1016/j.jsc.2017.03.009

[4] Dario Andrea Bini and Giuseppe Fiorentino. 2000. Design, analysis, and implementation of a multiprecision polynomial rootfinder. *Numer. Algorithms* 23, 2-3 (2000), 127–173. https://doi.org/10.1023/A:1019199917103

[5] L. Blum, F. Cucker, M. Shub, and S. Smale. 1998. *Complexity and real computation.* Springer-Verlag, New York. xvi+453 pages. https://doi.org/10.1007/978-1-4612-0701-6

[6] Peter Bürgisser and Felipe Cucker. 2013. *Condition: The geometry of numerical algorithms.* Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Vol. 349. Springer, Heidelberg. xxxii+554 pages. https://doi.org/10.1007/978-3-642-38896-5

[7] Michael A. Burr and Felix Krahmer. 2012. SqFreeEVAL: an (almost) optimal real-root isolation algorithm. *J. Symbolic Comput.* 47, 2 (2012), 153–166. https://doi.org/10.1016/j.jsc.2011.08.022

[8] D. Castro, J. L. Montaña, L. M. Pardo, and J. San Martín. 2002. The distribution of condition numbers of rational data of bounded bit length. *Found. Comput. Math.* 2, 1 (2002), 1–52. https://doi.org/10.1007/s002080010017

[9] J. H. Davenport. 1988. *Cylindrical algebraic decomposition.* Technical Report 88–10. University of Bath. http://www.bath.ac.uk/masjhd/

[10] Jean-Pierre Dedieu. 2006. *Points fixes, zéros et la méthode de Newton.* Mathématiques & Applications (Berlin) [Mathematics & Applications], Vol. 54. Springer, Berlin. xii+196 pages.

[11] R. G. Downey and M. R. Fellows. 2013. *Fundamentals of parameterized complexity.* Springer, London. xxx+763 pages. https://doi.org/10.1007/978-1-4471-5559-1

[12] Zilin Du, Vikram Sharma, and Chee K. Yap. 2007. Amortized bound for root isolation via Sturm sequences. In *Symbolic-numeric computation (Trends Math.).* Birkhäuser, Basel, 113–129. https://doi.org/10.1007/978-3-7643-7984-1_8

[13] Arno Eigenwillig, Lutz Kettner, Werner Krandick, Kurt Mehlhorn, Susanne Schmitt, and Nicola Wolpert. 2005. A Descartes algorithm for polynomials with bit-stream coefficients. In *Computer algebra in scientific computing (Lecture Notes in Comput. Sci., Vol. 3718).* Springer, Berlin, 138–149. https://doi.org/10.1007/11555964_12

[14] Arno Eigenwillig, Vikram Sharma, and Chee K. Yap. 2006. Almost tight recursion tree bounds for the Descartes method. In *ISSAC 2006.* ACM, New York, 71–78. https://doi.org/10.1145/1145768.1145786

[15] Ioannis Emiris, Bernard Mourrain, and Elias Tsigaridas. 2020. Separation bounds for polynomial systems. *J. Symbolic Comput.* 101 (2020), 128–151. https://doi.org/10.1016/j.jsc.2019.07.001

[16] Ioannis Z. Emiris, André Galligo, and Elias P. Tsigaridas. 2010. Random polynomials and expected complexity of bisection methods for real solving. In *ISSAC 2010—Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation.* ACM, New York, 235–242. https://doi.org/10.1145/1837934.1837980

[17] I. Z. Emiris, B. Mourrain, and E. P. Tsigaridas. 2008. Real Algebraic Numbers: Complexity Analysis and Experimentation. In *Reliable Implementations of Real Number Algorithms: Theory and Practice (LNCS, Vol. 5045)*, P. Hertling, C. Hoffmann, W. Luther, and N. Revol (Eds.). Springer, Berlin, Heidelberg, 57–82.

[18] Ioannis Z. Emiris, Victor Y. Pan, and Elias P. Tsigaridas. 2012. Algebraic algorithms. In *Computing Handbook Set - Computer Science* (3nd ed.), Teofilo Gonzalez (Ed.). Vol. I. CRC Press Inc., Boca Raton, Florida, Chapter 10, 10–1–10–30.

[19] Paula Escorcielo and Daniel Perrucci. 2017. On the Davenport-Mahler bound. *J. Complexity* 41 (2017), 72–81. https://doi.org/10.1016/j.jco.2016.12.001

[20] Steven Fortune. 2002. An iterated eigenvalue algorithm for approximating roots of univariate polynomials. *J. Symbolic Comput.* 33, 5 (2002), 627–646. https://doi.org/10.1006/jsco.2002.0526 Computer algebra (London, ON, 2001).

[21] Michael Hemmer, Elias P. Tsigaridas, Zafeirakis Zafeirakopoulos, Ioannis Z. Emiris, Menelaos I. Karavelas, and Bernard Mourrain. 2009. Experimental Evaluation and Cross-Benchmarking of Univariate Real Solvers. In *Proceedings of the 2009 Conference on Symbolic Numeric Computation* (Kyoto, Japan) *(SNC '09)*. Association for Computing Machinery, New York, NY, USA, 45–54. https://doi.org/10.1145/1577190.1577202

[22] Nicholas J. Higham. 2002. *Accuracy and stability of numerical algorithms* (second ed.). Society for Industrial and Applied Mathematics (SIAM, Philadelphia, PA. xxx+680 pages. https://doi.org/10.1137/1.9780898718027

[23] Jeremy R. Johnson, Werner Krandick, Kevin Lynch, David G. Richardson, and Anatole D. Ruslanov. 2006. High-performance implementations of the Descartes method. In *ISSAC 2006.* ACM, New York, 154–161. https://doi.org/10.1145/1145768.1145797

[24] Peter Kirrinnis. 1998. Partial fraction decompostion in $\mathbf{C}(\mathbf{z})$ and simultaneous Newton iteration for factorization in $\mathbf{C}[\mathbf{z}]$. *J. Complexity* 14, 3 (1998), 378–444. https://doi.org/10.1006/jcom.1998.0481

[25] Alexander Kobel, Fabrice Rouillier, and Michael Sagraloff. 2016. Computing real roots of real polynomials . . . and now for real!. In *Proceedings of the 2016 ACM International Symposium on Symbolic and Algebraic Computation.* ACM, New York, 303–310. https://doi.org/10.1145/2930889.2930937

[26] Werner Krandick and Kurt Mehlhorn. 2006. New bounds for the Descartes method. *J. Symbolic Comput.* 41, 1 (2006), 49–66. https://doi.org/10.1016/j.jsc.2005.02.004

[27] G. Livshyts, G. Paouris, and P. Pivovarov. 2016. On sharp bounds for marginal densities of product measures. *Israel Journal of Mathematics* 216, 2 (2016), 877–889. https://doi.org/10.1007/s11856-016-1431-5

[28] John M. McNamee and Victor Y. Pan. 2013. *Numerical methods for roots of polynomials. Part II.* Studies in Computational Mathematics, Vol. 16. Elsevier/Academic Press, Amsterdam. xxii+726 pages.

[29] Kurt Mehlhorn, Michael Sagraloff, and Pengming Wang. 2015. From approximate factorization to root isolation with application to cylindrical algebraic decomposition. *J. Symbolic Comput.* 66 (2015), 34–69. https://doi.org/10.1016/j.jsc.2014.02.001

[30] G. Moroz. 2021. New data structure for univariate polynomial approximation and applications to root isolation, numerical multipoint evaluation, and other problems. arXiv:2106.02505.

[31] N. Obreshkoff. 2003. *Zeros of polynomials.* Marin Drinov Academic Publishing House, Sofia, Bulgaria. Translation from the Bulgarian..

[32] Victor Y Pan. 1997. Solving a polynomial equation: some history and recent progress. *SIAM review* 39, 2 (1997), 187–220. https://doi.org/10.1137/S0036144595288554

[33] Victor Y. Pan. 2000. Approximating complex polynomial zeros: modified Weyl's quadtree construction and improved Newton's iteration. *J. Complexity* 16, 1 (2000), 213–264. https://doi.org/10.1006/jcom.1999.0532 Real computation and complexity (Schloss Dagstuhl, 1998).

[34] Victor Y. Pan. 2002. Univariate polynomials: nearly optimal algorithms for numerical factorization and root-finding. *J. Symbolic Comput.* 33, 5 (2002), 701–733. https://doi.org/10.1006/jsco.2002.0531 Computer algebra (London, ON, 2001).

[35] Victor Y. Pan. 2021. New Progress in Polynomial Root-finding. http://arxiv.org/abs/1805.12042 arXiv: 1805.12042.

[36] Victor Y. Pan and Elias P. Tsigaridas. 2013. On the Boolean complexity of real root refinement. In *ISSAC 2013—Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation.* ACM, New York, 299–306. https://doi.org/10.1145/2465506.2465938

[37] T. Roughgarden. 2021. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, Cambridge. https://doi.org/10.1017/9781108637435

[38] Fabrice Rouillier and Paul Zimmermann. 2004. Efficient isolation of polynomial's real roots. *J. Comput. Appl. Math.* 162, 1 (2004), 33–50. https://doi.org/10.1016/j.cam.2003.08.015

[39] M. Rudelson and R. Vershynin. 2008. The Littlewood-Offord problem and invertibility of random matrices. *Adv. Math.* 218, 2 (2008), 600–633. https://doi.org/10.1016/j.aim.2008.01.010

[40] M. Rudelson and R. Vershynin. 2015. Small ball probabilities for linear images of high-dimensional distributions. *Int. Math. Res. Not. IMRN* 19 (2015), 9594–9617. https://doi.org/10.1093/imrn/rnu243

[41] Michael Sagraloff. 2014. On the complexity of the Descartes method when using approximate arithmetic. *J. Symbolic Comput.* 65 (2014), 79–110. https://doi.org/10.1016/j.jsc.2014.01.005

[42] Michael Sagraloff and Kurt Mehlhorn. 2016. Computing real roots of real polynomials. *J. Symbolic Comput.* 73 (2016), 46–86. https://doi.org/10.1016/j.jsc.2015.03.004

[43] Michael Sagraloff and Chee K. Yap. 2011. A simple but exact and efficient algorithm for complex root isolation. In *ISSAC 2011—Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation*. ACM, New York, 353–360. https://doi.org/10.1145/1993886.1993938

[44] Arnold Schönhage. 1982. The Fundamental Theorem of Algebra in Terms of Computational Complexity. Manuscript. Univ. of Tübingen, Germany.

[45] Vikram Sharma. 2008. Complexity of real root isolation using continued fractions. *Theoret. Comput. Sci.* 409, 2 (2008), 292–310. https://doi.org/10.1016/j.tcs.2008.09.017

[46] The PARI Group 2019. *PARI/GP version* 2.11.2. The PARI Group, Univ. Bordeaux. available from http://pari.math.u-bordeaux.fr/.

[47] E. C. Titchmarsh. 1939. *The theory of functions* (second ed.). Oxford University Press, Oxford. x+454 pages.

[48] J. Tonelli-Cueto and E. Tsigaridas. 2020. Condition Numbers for the Cube. I: Univariate Polynomials and Hypersurfaces. In *Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation* (Kalamata, Greece) *(ISSAC '20)*. Association for Computing Machinery, New York, NY, USA, 434–441. https://doi.org/10.1145/3373207.3404054

[49] J. Tonelli-Cueto and E. Tsigaridas. 2021. Condition Numbers for the Cube. I: Univariate Polynomials and Hypersurfaces. To appear in the special issue of the Journal of Symbolic Computation for ISSAC 2020. Available at arXiv:2006.04423.

[50] Elias Tsigaridas. 2016. SLV: a software for real root isolation. *ACM Commun. Comput. Algebra* 50, 3 (2016), 117–120.

[51] Elias P. Tsigaridas and Ioannis Z. Emiris. 2008. On the complexity of real root isolation using continued fractions. *Theoret. Comput. Sci.* 392, 1-3 (2008), 158–173. https://doi.org/10.1016/j.tcs.2007.10.010

[52] A. M. Turing. 1948. Rounding-off errors in matrix processes. *Quart. J. Mech. Appl. Math.* 1 (1948), 287–308. https://doi.org/10.1093/qjmam/1.1.287

[53] Joachim von zur Gathen and Jürgen Gerhard. 2003. *Modern computer algebra* (second ed.). Cambridge University Press, Cambridge. xiv+785 pages.

[54] J. H. Wilkinson. 1971. Some comments from a numerical analyst. *J. Assoc. Comput. Mach.* 18 (1971), 137–147. https://doi.org/10.1145/321637.321638