Beyond Worst-Case Analysis for Root Isolation Algorithms

Alperen A. Ergür, Josué Tonelli-Cueto, Elias Tsigaridas

To cite this version:

Alperen A. Ergür, Josué Tonelli-Cueto, Elias Tsigaridas. Beyond Worst-Case Analysis for Root Isolation Algorithms. ISSAC ’22 - International Symposium on Symbolic and Algebraic Computation, Jul 2022, Villeneuve-d’Ascq, France. hal-03575449

HAL Id: hal-03575449
https://inria.hal.science/hal-03575449v1
Submitted on 25 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Beyond Worst-Case Analysis for Root Isolation Algorithms

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

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

Elias Tsigaridas
elias.tsigaridas@inria.fr
Inria Paris and Sorbonne Université
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 $O_B(d^2 + d\tau)$, where $d$ is the degree of the polynomial and $\tau$ the bitsize of the coefficients.

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

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 [45]. In contrast, other algorithms with inferior worst-case complexity estimates have excellent practical performance, e.g., [21, 25, 50]. 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, [50], that provides (conditional) expected case bounds for the continued fraction algorithm, 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 [43]. 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 $O_B(d^2\tau)$ for a degree $d$ polynomial with maximum coefficient bitsize $\tau$ [34]; see also [3, 24, 29]. Besides the algorithms already mentioned, there are also

\(^1\)According to Wilkinson [53], Turing [51] introduced condition numbers to explain the practical success of Gaussian elimination despite the existing worst-case analyses.
We develop a general model of randomness that provides a
more efficient framework for uni-vatopolynomials with inter-
coefficients. The main complexity parameters for univariate polynomials with
integer coefficients. The first natural candidate is the following: fix an
integer (or rational) coefficients is the degree 1.2 Warm-up: A simple form of the main result
applies several improvements to the modified Weyl algorithm by
Pan’s algorithm.

Notice that the expected time complexity of Descartes’
solver has excellent practical performance and it can routinely solve
though its worst case bound is similar to
Descartes’ based. These algorithms mimic binary search. Given an initial in-
terval 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 poly-
nomial is not square-free. Its complexity is \(O_B(d^3 \tau^2)\) [12, 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, 49]. There are also other algorithms based on the continued fraction expansion of
the real numbers [44, 50] and on point-wise evaluation [7, 54].

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
d, 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 \(O_B(d^3 + d^2 \tau)\).

**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 \(\tau\); 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 \(\tau\), let \(t_0, t_1, \ldots, t_d\) be independent copies of the uniformly distributed integer in \([-2^\tau, 2^\tau]\) \(\cap \mathbb{Z}\), and let \(f = \sum_{i=0}^d t_i X^i\) which we call the uniform random bit polynomial with bitsize \(\tau\). Recall that \(O\), resp. \(\tilde{O}\), denote the arithmetic, resp. bit, complexity and that we use \(O\), resp. \(\tilde{O}\), 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 \(f\) with bit size \(\tau(f)\), Descartes solver isolates the real roots of \(f\) in expected time \(O_B(d \tau + d^2)\).

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 a
smoothed analysis framework on polynomials with integer co-
coefficients.

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

1. the bitsize of \(f\), \(\tau(f)\), is the minimum integer \(\tau\) so that for all \(i \in \{0, 1, 2, \ldots, d\}, P(\{t_i\} \leq 2^\tau) = 1\).
2. the weight of \(f\), \(w(f)\), is the maximum probability that \(t_0, t_1, \ldots, t_d\) and \(\tau\) can take a value, i.e.,

\[w(f) := \max \{P(t_i = k) \mid i \in \{0, 1, d - 1, d\}, k \in \mathbb{R}\}\]

**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 \(\tau\) we introduced is the main example of a random bit polynomial \(f\). Note that in this case we have \(w(f) = \frac{1}{2^{\tau+1}}\) and \(r(f) = \tau\).

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 \(w(f) = 1\) which would make our randomness model equivalent to the worst-case model. To control the effect of large \(w(f)\) 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 \(f\) is

\[u(f) := \ln \left(w(f)^{(1 + 2^{\tau f (1 + 1)})}\right)\]

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

The following three examples illustrate the flexibility of our 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 \(f := \sum_{i \in A} t_i X^i\), where the \(t_i\)’s are independent and uniformly distributed in \([-2^\tau, 2^\tau]\) is a random bit polynomial with \(u(f) = 0\) and \(r(f) = \tau\).

**Example 1.8 (Sign of the coefficients).** Let \(s \in \{-1, +1\}^{d+1}\). The random polynomial \(f := \sum_{i=0}^d s_i X^i\), where the \(s_i\)’s are independent and uniformly distributed in \(s_i \in \{1, 2\} \cap \mathbb{Z}\), is a random bit polynomial with \(u(f) = \ln(3)\) and \(r(f) = \tau\).

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

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 models includes smoothed analysis
over integer coefficients as a special case.
To estimate the number of roots in the Obreshkoff areas we use this fact follows from the work of Obreshkoff [31], see also [26].

bounds and their probabilistic estimates in Section 2, we deal with mostly restricted to continuous random variables, with few exceptions [8]; see also [16, 36, 50]. 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 deal with 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 $\tilde{O}$, resp. $\tilde{O_B}$, 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 $\mathcal{P}_d^{\mathbb{C}}$, the subset of integer polynomial.

Remark 1.12. Note that if $f$ 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 $u(f)$.

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 [48]. 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 Obreschkoff [31], see also [26]. To estimate the number of roots in the Obreschkoff 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 subdivision tree of DESCARTES and so its bit complexity.

Finally, we perform the expected/smoothed analysis of 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, 50]. 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].

Example 1.10 (Smoothed analysis). Let $f \in \mathcal{P}_d$ be a fixed integer polynomial with coefficients in $[-2^\sigma, 2^\sigma]$, $\sigma \in \mathbb{Z} \setminus \{0\}$ and $f \in \mathcal{P}_d$ a random bit polynomial. Then $f_{\sigma} := f + \sigma f$ is a random bit-polynomial with bitsize $\tau(f_{\sigma}) \leq \max\{\tau(f), \tau(\sigma)\} + 1$, where $\tau(a)$ denotes the bitsize of $a$, and uniformity $u(f_{\sigma}) \leq 1 + \max\{\tau - \tau(f), \tau(\sigma)\} + u(f)$. 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 $f$ be random bit polynomial, of degree $d$, bitsize $\tau(f)$, and uniformity parameter $u(f)$, such that $\tau(f) = \Omega(\log d + u(f))$, then DESCARTES solver isolates the real roots of $f$ in expected time \( \tilde{O}_B(d \tau(1 + u(f))^3 + d^2(1 + u(f))^4) \).

**Remark 1.12.** Note that if $f$ 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 $u(f)$.

2 CONDITION NUMBERS, SEPARATION BOUNDS, AND RANDOMNESS

We use various condition numbers for univariate polynomials from [48], cf. [47], to control the separation bound of random polynomials. However, our probabilistic analysis differs from [48] because we consider discrete random coefficients.

2.1 Condition numbers for univ. polynomials

The local condition number of $f \in \mathcal{P}_d$ at $z \in \mathbb{D}$ [48] is

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

where $||f||_1 := \sum |a_i|$ 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

$$C_R(f) := \max_{x \in I} C(f, x).$$

We note that as $C_R(f)$ 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 [48, Theorem 4.4]). There are many interesting properties of $C_R(f)$, but let us state the only one we will use—see [48, Theorem 4.2] for more.

**Theorem 2.1 (2nd Lipschitz property).** [48] Let $f \in \mathcal{P}_d$. The map $\mathbb{D} \ni z \mapsto 1/C(f, z)$ is $[0,1]$ 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 \( \varepsilon \in \left[ 0, \frac{1}{2} \right] \) we set \( I_\varepsilon := \{ z \in C \mid \text{dist}(z, I) \leq \varepsilon \} \). If \( f \in \mathcal{P}_d \), then the \( \varepsilon \)-real separation of \( f \), \( \Lambda^\varepsilon(f) \), is
\[
\Lambda^\varepsilon(f) := \min \left\{ |\zeta - \zeta'| \mid \zeta, \zeta' \in I_\varepsilon, \ f(\zeta) = f(\zeta') = 0 \right\},
\]
if \( f \) has no double roots in \( I_\varepsilon \), and \( \Lambda^\varepsilon(f) := 0 \) otherwise.

**Theorem 2.3 ([48, Theorem 6.3]).** Let \( f \in \mathcal{P}_d \) and assume \( \varepsilon \in \left[ 0, \frac{1}{\text{ed} C_5(f)} \right) \), then \( \Lambda^\varepsilon(f) \geq \frac{1}{\text{ed} C_5(f)} \).

### 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 \( f \in \mathcal{P}^+_{d} \) be a random bit polynomial and \( x \in I \).
Then, for \( t \leq 2^{\gamma(t)} \), \( \mathbb{P}(C(f, x) \geq t) \leq 16 d^2 e^{2u(t)} \frac{1}{t} \).

**Theorem 2.5.** Let \( f \in \mathcal{P}^+_{d} \) be a random bit polynomial. Then, for \( t \leq 2^{\gamma(t)+1} \),
\[
\mathbb{P}(C(f, x) \geq t) \leq 32 d^4 e^{2u(t)} \frac{1}{t},
\]

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 \( f \in \mathcal{P}^+_{d} \) be a random bit polynomial, \( \ell \in \mathbb{N} \) and \( c \geq 1 \). If \( \tau(f) \geq 4 \log_2(c) + 2u(f) \), then
\[
\left( \mathbb{E}_x \left( \min \{ \log_2(C(f, x), c) \} \right) \right)^\frac{1}{2} \leq \Theta \left( \left( \log_2 + u(f) \right) \right).
\]

We would like to understand the limitations of the two theorems and the corollary above. First, note that Theorem 2.4 is meaningful when \( \tau(f) \geq 2 + \frac{1}{2} \log_2(d) + 2u(f) \) and Theorem 2.5 is meaningful when \( \tau(f) \geq 5 + 4 \log_2(2) + 3u(f) \). Intuitively, the randomness model needs some wigglng room to differ from the worst-case analysis. In our case this translates to assume that the bit-size \( \log_2(f) \) is bigger than (roughly) \( \log_2(d) + u(f) \). This is a reasonable assumption because for most cases of interest, \( u(f) \) is bounded above by a constant. Thus, the second condition in Corollary 2.6 becomes
\[
\tau(f) = \Omega \left( \log_2(d) + \log_2(c) \right).
\]

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 \( \tau(f) \) is lower bounded by (roughly) \( \log_2(d) \), which is not uncommon in practice.

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

**Proposition 2.7.** Let \( x \in \mathbb{Z}^N \) be a random vector with independent coordinates. Assume that there is \( w > 0 \) so that for all \( i \) and \( x \in \mathbb{Z}, \mathbb{P}(x_i = x) \leq w. \) Then for every linear map \( A \in \mathbb{R}^{k \times N}, b \in \mathbb{R}^k \) and \( \varepsilon \in \left( \| A \|_{\infty, \infty}, \infty \right), \)
\[
\mathbb{P}(\| A x + b \|_{\infty} \leq \varepsilon) \leq \frac{(2\sqrt{2we})^k}{\sqrt{\text{det} AA^T}}.
\]

**Proof of Theorem 2.4.** \( \mathbb{P}(C(f, x) \geq t) \) equals
\[
\sum_{a_2, \ldots, a_{d-2}} \mathbb{P}(C(f, x) \geq t | c_2 = a_2, \ldots, c_{d-2} = a_{d-2}) \frac{d-2}{i=2} \mathbb{P}(c_i = a_i).
\]

where \( t = \sum_k c_k X^k \). So it is enough to prove the bound for a random bit polynomial \( f \) of the form
\[
f = c_0 + c_1 x + \sum_{k=2}^{d-2} a_k X^k + c_{d-1} X^{d-1} + c_d X^d,
\]

where \( a_2, \ldots, a_{d-2} \in \mathbb{Z} \cap [-2^t, 2^t] \) are arbitrary fixed integers.

Let \( \mathcal{P}_d(a_2, \ldots, a_{d-2}) \) be the affine subspace of \( \mathcal{P}_d \) given by \( f_k = a_k \) for \( k \in \{2, \ldots, d-2\} \). And let \( f \mapsto Af + b \) be the affine mapping that maps \( f \in \mathcal{P}_d(a_2, \ldots, a_{d-2}) \) to \( (f(x), f'(x)/d) \in \mathbb{R}^2 \). In the coordinates we are working on (those of the base \( (1, X, X^{d-1}, X^d) \)), \( A \) has the form
\[
\begin{pmatrix}
1 & x & x^{d-1} \\
0 & 1 & (1 - 1/d) x^{d-2} \\
1 & 0 & x^{d-1}
\end{pmatrix}
\]

So, by an elementary estimation we have \( \| A \|_{\infty, \infty} \leq d + 1 \), and as a direct result of Cauchy-Binet formula we have \( \text{det} AA^T \geq 1/d \).

Now, since \( \| f \|_1 \leq (d + 1) 2^{\gamma(t)} \), we have that \( \mathbb{P}(C(f, x) \geq t) = \mathbb{P}(\|Af + b\|_\infty \leq \|f\|_1 / t) \leq \mathbb{P}(\| Af + b \|_\infty \leq (d + 1) 2^{\gamma(t)}) / t \).

To be able to use Proposition 2.7, we need to assume \( \frac{d+1}{d} 2^{\gamma(t)} \geq 1 \). Then, for \( t \leq 2^{\gamma(t)} \), Proposition 2.7 implies
\[
\mathbb{P}(C(f, x) \geq t) \leq 16 d(d + 1)^2 (2^{2\gamma(t)}) / t^2.
\]

Thus the proof is completed by the definition of \( u(f) \).

**Proof of Theorem 2.5.** We will use a covering/union bound argument. For any finite set \( G \subseteq [-1, 1] \) such that \( \{x - \delta, x + \delta \mid x \in G\} \) covers \([-1, 1]\), using the 2nd Lipschitz property (Theorem 2.1), we have \( 1/\max_{x \in G} C(f, x) \leq 1/C(f) + \delta \). Let \( \delta = 1/dt \), then \( \mathbb{P}(C(f, x) \geq t) \leq \mathbb{P}(\max_{x \in G} C(f, x) \geq t/2) \geq \mathbb{P}(\max_{x \in [-1, 1]} C(f, x) \geq t/2) \). We can construct such \( G \) such that \( \#G \leq 2dt \). Hence the claim follows from Theorem 2.4.

**Proof of Corollary 2.6.** Let
\[
U := \ln(32 d^4 e^{2u(t)}) \leq 4 \ln(\text{ed}) + 2u(f) \text{ and } V := \ln(2^{\gamma(t)+1}).
\]

By assumption, \( U \leq V \leq U + 1 > 1, \) since \( u(f) \geq 0. \) So without loss of generality, we assume \( 0 < U < V \). If \( c \leq U \), then similar arguments imply that the claimed upper bound still holds. Thus
\[
\mathbb{E}_x \left( \min \{ \log_2(C(f, x), c) \} \right) = \int_0^c x e^{-x} \mathbb{P}(\min \{ \log_2(C(f, x), c) \geq s \}) \, ds.
\]

We divide the integral into three summands using the intervals \([0, U], [U, V] \) and \([V, c] \).
In $[0, U]$, we have that $P(\min(\ln C_k(f), c) \geq s) \leq 1$, and so
\[
\int_0^U t_1^{\ell_1-1} P(\min(\ln C_k(f), c) \geq s) \, dt \leq U^{\ell_1}.
\]
In $[U, V]$, by Theorem 2.5 we have that
\[
P(\min(\ln C_k(f), c) \geq s) \leq P(\ln C_k(f) \geq s) \leq e^{-s},
\]
and so $\int_U^V t_1^{\ell_1-1} P(\min(\ln C_k(f), c) \geq s) \, dt$ is bounded by
\[
\int_U^V t_1^{\ell_1-1} e^{-s} \, dt.
\]
By performing a change of variables and extending the domain, we get $\int_0^\infty (s+U)^{\ell_1-1} e^{-s} \, ds$. The latter, expanding the binomial $(s+U)^{\ell_1-1}$ and using that $\Gamma(k + 1) = k!$, is bounded by $\ell_1 \sum_{k=0}^{\ell_1-1} (\ell_1-1)! k! U^{\ell_1-1-k}$. Hence, as $(\ell_1-1)! \leq t_1^{\ell_1-1}$, we get
\[
\int_0^V t_1^{\ell_1-1} P(\min(\ln C_k(f), c) \geq s) \, dt = t_1^{\ell_1-1}.
\]
In $[V, \infty)$, we have that
\[
P(\min(\ln C_k(f), c) \geq s) \leq P(\ln C_k(f) \geq s) \leq e^{-s}.
\]
Therefore, since $e^{V-U} \int_C t_1^{\ell_1-1} \, dt \leq e^{V-U} \int_0^U t_1^{\ell_1-1} \, dt,
\int_U^V t_1^{\ell_1-1} P(\min(\ln C_k(f), c) \geq s) \, dt \leq e^{V-U} e^{c}.
\]
To obtain the final estimate, we add the three upper bounds obtaining the upper bound $U^{\ell_1} + e^{V-U} e^{c}$. After substituting the values of $U$ and $V$ and some elementary estimates, we conclude. □

Proof of Proposition 2.7. Let $\eta \in R^N$ be such that the $\eta_i$ are independent and uniformly distributed in $(-1/2, 1/2)$. Now, a simple computation shows that $x + \eta$ is absolutely continuous and each component has density given by
\[
\delta_{x+\eta}(t) = \sum_{s \in \mathbb{Z}} P(x_i = s) \delta_{\eta_i}(t-s).
\]
Thus each component of $x + \eta$ has density bounded by $w$. We have
\[
P(\|Ax + b\|_{\infty} \leq x) \leq P(\|Ax + b\|_{\infty} \leq 2x)/P(\|A\eta\|_{\infty} \leq x),
\]
since $x$ and $\eta$ are independent, and by the triangle inequality.

On the one hand, we apply [47, 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 $\eta \in R^N$ with independent coordinates with density bounded by $\rho$ and $A \in R^k \times N$, we have that $A\eta$ has density bounded by $(\sqrt{2}\rho)^k/\sqrt{\det AA^T}$. Thus
\[
P(\|Ax + b\|_{\infty} \leq 2x) \leq (2\sqrt{2}\rho x)^k/\sqrt{\det AA^T}.
\]
On the other hand,
\[
P(\|A\eta\|_{\infty} \leq 1) = 1 - P(\|A\eta\|_{\infty} \geq 1) \geq 1 - \mathbb{E}[\|A\eta\|_{\infty}]/\epsilon.
\]
by Markov's inequality. Now, by our assumption on $\epsilon$, we only need to show that $\mathbb{E}[\|A\eta\|_{\infty}] \leq \mathbb{E}[\|A\eta\|_{\infty}]/2$.

By Jensen's inequality,
\[
\mathbb{E}[\|A\eta\|_{\infty}] = \mathbb{E} \left( \lim_{\epsilon \to 0} \|A\eta\|_{2\epsilon} \right) \leq \lim_{\epsilon \to 0} \left( \mathbb{E}[\|A\eta\|]^{2\epsilon} \right)^{1/2\epsilon}.
\]
Expanding the interior and computing the moments of $\eta$, we obtain
\[
\mathbb{E}[\|A\eta\|_{\infty}] \leq \lim_{\epsilon \to 0} \left( \sum_{i=1}^k \sum_{|a|=\epsilon} \left( 2\epsilon \right)^n \prod_{j=1}^n \left( A_{ij} \left( 1/2 \right)^{2\epsilon} / (2\epsilon + 1) \right) \right)^{1/2\epsilon}.
\]
since the odd moments disappear. Thus
\[
\mathbb{E}[\|A\eta\|_{\infty}] \leq \frac{1}{2} \lim_{\epsilon \to 0} \left( \sum_{i=1}^k \sum_{|a|=\epsilon} \left( 2\epsilon \right)^n \prod_{j=1}^n \left( A_{ij} \left( 1/2 \right)^{2\epsilon} / (2\epsilon + 1) \right) \right)^{1/2\epsilon} = \frac{\|A\eta\|_{\infty}}{2},
\]
where we obtained the bound of $\|A\eta\|_{\infty}/2$ 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 $D$, because the symmetry on our randomness model forces any bound on the number of roots in $D$ to be of the form $O(d)$. For this, we consider a family of disks $(\mathbb{D}: \xi_n, \rho_n)$ inspired by the one in [30], where we will specify $N$ in the sequel. In particular,
\[
\xi_n, N = \left\{ \frac{\operatorname{sgn}(n)}{\left( 1 + \frac{1}{\sqrt{2\pi}} \right)}, \quad \text{if } n = N - 1 \right\}
\]
\[
\xi_n, N = \left\{ \frac{\operatorname{sgn}(n)}{\left( 1 - \frac{1}{\sqrt{2\pi}} \right)}, \quad \text{if } n = N \right\}.
\]
We will abuse notation and write $\xi_n$ and $\rho_n$ 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 $N = |\log d|$,}
\[
\psi(f) := \# \left\{ z \in \mathbb{D} : \mathbb{D}(\xi_n, \rho_n) \cap f(z) = 0 \right\}.
\]
where $f \in P_d$. We use these bounds to estimate the number of steps of DESCARTES($f$).

3.1 Deterministic bound

Theorem 3.1. Let $f \in P_d$. Then
\[
\psi(f) \leq \sum_{n=-\left|\log d\right|}^{\left|\log d\right|} \log |f(\xi_n)|/|f(\xi_n)|.
\]

Lemma 3.2. Let $f \in P_d$, $\xi \in \mathbb{D}$, and $\rho > 0$. If $|\xi| + 2\rho < 1 + 1/d$, then $\#(\mathbb{D}(\xi, \rho) \cap f = 0) \leq \log |f(\xi)|/|f(\xi)|$.

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 $\mathbb{D}(\xi_n, \rho_n) \in \Omega_d$.

Proof of Lemma 3.2. We use a classic result of Ritchmarsh [46, p. 171] that bounds the number of roots in a disk. For $\delta \in (0, 1)$, we have that $\#(\mathbb{D}(\xi, \rho) \cap f = 0) \leq (\ln(1/\delta))^{-1} \ln(\max_{z \in \mathbb{D}} |f(\xi + \rho z/\delta)|/|f(\xi)|)$.

Take $\delta = 1/2$. By our assumption, $\xi + 2\rho \mathbb{D} \in (1 + 1/d)\mathbb{D}$, so $\max_{z \in \mathbb{D}} |f(\xi + \rho z/\delta)| \leq \max_{z \in \mathbb{D}} |f(z)| \leq \min f(\xi), \min f(\xi)|, \quad \text{since } |f(\xi)| \leq |f(\xi)|$, for $z \in (1 + 1/d)\mathbb{D}$ [48, Proposition 3.9]. □
3.2 Probabilistic bound

Theorem 3.3. Let \( f \in P_d \) be a random bit polynomial. Then for all \( t \leq \tau(f)(2[lg d] + 1) \),
\[
\Pr (q(f) \geq t) \leq 4d^2 (2[lg d] + 1) e^{4t} c^{-\frac{\tau(f)}{2}}.
\]

Corollary 3.4. Let \( f \in P_d \) be a random bit polynomial and \( k \in \mathbb{N} \). Suppose that \( \tau(f) \geq 10 ln(ed) + 2u(f) \). Then
\[
\left( \mathbb{E}_q(f)^t \right)^\frac{1}{t} \leq 2(1 + e) (6 ln(ed) + u(f)) ln(ed) + \left( \frac{4d^2 e^{4t} c^{-\frac{\tau(f)}{2}}}{\tau(f)^t} \right) \frac{1}{t}.
\]

In particular, if \( \tau(f) \geq (9 + 3e) ln(ed) + 2u(f) \), then
\[
\left( \mathbb{E}_q(f) \right)^t \leq O \left( (\tau ln(d + u(f)) ln(d) \right).
\]

Proof of Theorem 3.3. If \( \# (Z(f) \cap \Omega_d) \geq t \), then, by Theorem 3.1, there is an \( n \) such that \( \log(e^{|f|}/\|f(\xi_n)\|) \geq t/2[lg d] + 1) \). Hence
\[
\Pr (q(f) \geq t) \leq \sum_{n=0}^{[lg d]} \left( \Pr (\|f\|/\|f(\xi_n)\|) \geq t/2[lg d] + 1) \right).
\]
Now, fix \( x \in I \). We argue as in the proof of Theorem 2.4, but we consider that map \( f \rightarrow f(x) \) instead of the map mapping \( f \rightarrow (f(x), f'(x)/d) \), so that our matrix \( A \) takes the form
\[
\begin{pmatrix}
1 & x & x^d & 1
\end{pmatrix}.
\]
Note that this \( A \) has \( \|A\|_{\infty, \infty} \leq d + 1 \). So, we can apply Proposition 2.7 to show that for any \( s \leq e^{t(f)} \),
\[
\Pr (\|f\|/\|f(x)\|) \geq s \leq 4d^2 e^{4t(f)} / s.
\]
if \( s = e^{t(f)} \), with \( N = 2[lg d] + 1 \), then the bound follows. \( \square \)

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 \( U e^{-t} \) 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 \( x \in [0, \infty) \) a random variable. If \( \Pr (x \geq t) \leq e^{t(x)} \) for \( s \leq V \), then \( \exists (\min(x,c)) t' \leq U \), \( c' e^{t(x)} \).

By Theorem 3.3, the random variable \( g(f)/(2[lg d] + 1) \) satisfies the conditions to be a random variable \( x \) with \( U = ln(4d^2 (2[lg d] + 1) e^{4t(f)}) \leq 4 ln(ed) + ln(2[lg d] + 1) + u(f) \), \( V = ln(2e^t (2[lg d] + 1)) \), and \( c = e^t e^{4t} \), since the roots are at most \( d \). By our assumptions \( U \leq V \), that concludes the proof. \( \square \)

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 \( f = \sum_{i=0}^{d} f_i X^i \in \mathbb{P}_d \) equals the number of positive real roots (counting multiplicities) of \( f \), say \( r \), plus an even number; that is \( r \equiv \text{VAR}(f) \mod 2 \).

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 \( x \rightarrow \frac{x^d}{d} \) that maps \( J \) to \( (0, \infty) \). Then
\[
\text{VAR}(f, J) := \text{VAR}((X + 1)^d (f(X + 1)^d))
\]

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 \( V = \text{VAR}(f, J_0) \). 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 \( J_1 \) and \( J_2 \) and repeat the process. The pseudocode 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 \( J_L = (a, \inf(d J)) \) and \( J_R = (\inf(d J), b) \) respectively. The internal nodes of the tree correspond to intervals \( J \), such that \( \text{VAR}(f, J) \geq 2 \). The leaves 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 the 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 \( f = \sum_{i=0}^{d} f_i X^i \in \mathbb{P}_d \) of bit-size \( \tau \).

• The reciprocal transformation is \( R(f) = X^d f(X) \).

• The homothetic transformation of \( f \) by \( x^k \), for a positive integer \( k \), is \( H_k(f) = 2^k x^k f(\frac{x}{2^k}) = \sum_{i=0}^{d} 2^k(d-i) f_i X^i \). It costs \( O_d(\mu(t + dk)) \).

• The Taylor shift of \( f \) by in integer \( c \) is \( T_c(f) = f(x + c) = \sum_{i=0}^{d} a_i X^i \), where \( a_i = \sum_{j=0}^{d} \binom{j}{i} f_j c^{j-i} \) for \( 0 \leq i \leq d \). It costs \( O_d(\mu^{2d}(\sigma + d) \log d) \).

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 (−∞, −1) and (1, ∞).

4.1 Bounds on the number of sign variations

For this subsection we consider \( f = \sum_{i=0}^{d} a_i x^i \in \mathbb{P}_d \) 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 \( 0 \leq q \leq d \) and a real open interval \( J = (a, b) \). The Obreshkoff discs \( D_Q \) and \( D_Q^\varnothing \) are discs the boundaries of which go through the endpoints of \( J \). Their centers are above, respectively below, \( J \) and they form an angle \( \varrho = \frac{\pi}{1 + \frac{1}{2}N} \) with the endpoints of \( I \). Its diameter is \( \text{wid}(f) / \sin(\frac{\pi}{1 + \frac{1}{2}N}) \).

The Obreshkoff area is \( A_Q(J) = \text{interior}(D_Q \cup D_Q^\varnothing) \); it appears with grey color in Fig. 1. The Obreshkoff lens is \( L_Q(J) = \text{interior}(D_Q \cap D_Q^\varnothing) \); it appears in light-grey color in Fig. 1. If it is clear from the context, then we omit \( I \) and we write \( A_Q \) and \( L_Q \), instead of \( A_Q(J) \) and \( L_Q(J) \). It holds that \( L_d \subset L_d-1 \subset \cdots \subset L_1 \subset L_0 \) and \( A_d \subset A_{d-1} \subset \cdots \subset A_0 \subset A_d \).

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 \( f \in \mathbb{P}_d \) and real open interval \( J = (a, b) \). If the Obreshkoff lens \( L_d-k \) contains at least \( k \) roots (counted with multiplicity) of \( f \), then \( k \leq \text{var}(f, J) \). If the Obreshkoff area \( A_k \) contains at most \( k \) roots (counted with multiplicity) of \( f \), then \( \text{var}(f, J) \leq k \). Especially

\[ \# \{ \text{roots of } f \text{ in } L_d \} \leq \text{var}(f, J) \leq \# \{ \text{roots of } f \text{ in } A_d \} . \]

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 area controls the width of the subdivision tree of DESCARTES.

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

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 \( \varepsilon \)-real separation of \( f \), \( \Delta^\varepsilon(f) \) (Definition 2.2).

---

**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 \( D_{n+1} \setminus D_n \).

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

\[ 2\text{wid}(f, J) \leq \min\{\Delta^\varepsilon(f, \varepsilon)\}, \]

then \( \text{var}(f, J) = 0 \) (and \( J \) does not contain any real root), or \( \text{var}(f, J) = 1 \) (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 [48, 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 4.3 and 4.6. To bound the width of the DESCARTES’ tree we use the Obreshkoff 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 \( f \in \mathbb{P}_d \), then, using DESCARTES, the number of subdivision steps to isolate the real roots in \( I = (−1, 1) \) is

\[ \widetilde{O}(d \varrho(f))^2 \lg C(f) + d \varrho(f)^2 \lg^2 C(f). \]

Recall that \( C(f) \) 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 \( \Omega_d \). Recall that \( \Omega_d \) is the union of the discs \( D_n := \mathbb{D}(\xi_n, \rho_n) := \xi_n + \rho_n \mathbb{D} \), where \( n | \leq N \); 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 \( n \leq 0 \)). Note that \( J_n \) is the union of 3 intervals of size \( 1/2^n \varepsilon^2 \). 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 \( w(n) \), 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 \( n \geq 0 \); similar arguments apply for \( n \leq 0 \). We consider two cases: \( n < N \) and \( n = N \).
Case $n < N$. It holds $\text{wid}(j_n) = \rho_n = 3/2^{n+3}$. For each $j_n$, assume that we perform a number of subdivision steps to obtain intervals, say $J_n, \ell$, with $\text{wid}(J_n, \ell) = 2^{-\ell}$. We choose $\ell$ so that the corresponding Obershokoff areas, $\mathcal{A}_\ell(j_n, \ell)$, are inside $\Omega_d$. In particular, we ensure that the Obershokoff areas related to $J_n, \ell$ lie in $D_{n+1}$.

The diameter of the Obershokoff discs, $\mathcal{D}_\ell(j_n, \ell)$ and $\mathcal{D}_\ell(j_n, \ell)$, is $\text{wid}(j_n, \ell)/\sin \frac{\pi}{\theta}$. For every $\mathcal{A}_\ell(j_n, \ell)$ to be in $D_{n+1}$ and inside inside $\Omega_d$, it suffices that a disc with diameter $2 \text{wid}(j_n, \ell)/\sin \frac{\pi}{\theta}$, that has its center in the interval $\langle \delta, \varepsilon \rangle$ 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 $\mathcal{A}_\ell(j_n, \ell)$ and lies $D_{n+1}$. This auxiliary disc is the dotted (red) disc in Fig. 2 (left). It should be that

$$2 \text{wid}(j_n, \ell)/\sin \frac{\pi}{\theta} \leq 2 \rho_{n+1} = 3/2^{n+3}.$$ 

Taking into account that $\text{wid}(j_n, \ell) = 2^{-\ell}$ and

$$\sin \frac{\pi}{\theta} > \sin \frac{1}{2} \geq \frac{1}{2} \sqrt{1 + \frac{1}{4}} \geq \frac{1}{\sqrt{2} \theta},$$

we deduce $2^{-\ell+1} \rho_2 \leq 3/2^{n+3}$ and so $\ell \geq \lg \frac{2^{n+3}}{\theta}$. Hence, $\text{wid}(j_n, \ell) = 3/2^{n+3}$ and so $J_n$ is partitioned at most $\text{wid}(J_n)/\text{wid}(j_n, \ell) = 4\theta$ (sub)intervals. So, during the subdivision process, starting from each $J_n$, we obtain the intervals $J_{n, \ell}$ after performing at most $8\theta$ subdivision steps (this is the size of the complete binary tree starting from $J_0$). To say it differently, the subdivision tree that has $J_n$ as its root and the intervals $J_{n, \ell}$ as leaves has depth $\ell = \lfloor \lg(4\theta) \rfloor$. The same hold for $J_{n-1}$ because $\rho_n \leq \rho_n$, for all $0 \leq n \leq N - 1$.

Thus, the width of the tree starting at $J_n$ is at most $O(\theta \ell^2)$, because we have $O(\theta)$ subintervals $J_{n, \ell}$ for each $\text{VAR}(f; J_n)$ $\leq \theta$.

Case $n = N$. Now $\text{wid}(J_N) = 3/2^{N+1}$. 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 $\text{wid}(J_N) = 1/2^N$; see Figure 2.

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

$$\text{diam}(j_{n, \ell})/\sin \frac{\pi}{\theta} \leq \rho_{n+1} = 1/2^{N+1}.$$ 

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

By combining all the previous estimates, we conclude that the subdivision tree of descartes has width $O(\theta \ell^2 d + \ell \Theta \ell^2 \log \ell d)$.

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

$$1/2^{\ell-1} \leq \min(\Delta_{\ell}^R(f), \epsilon).$$

Fix $\epsilon = 1/(\text{od} C_{\mathcal{C}}(f))$. Then, by Theorem 2.3, it suffices to hold

$$\ell \geq 1 + \lg(12d C_{\mathcal{C}}(f)).$$

Hence, the depth of the subdivision tree is at most $O(\lg(d C_{\mathcal{C}}(f)))$.

Therefore, since the subdivision tree of descartes has width $O(\theta \ell^2 \log d + \log \ell^2 d)$ and depth $O(\lg(\log(\ell^2 d + d \log \ell)))$, 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 $\tilde{O}(d + d^2 \delta)$, where $\delta$ is the size of the highest bitsize that we compute with, or equivalently the depth of the subdivision tree. In our case, $\delta = O(\lg(d C_{\mathcal{C}}(f)))$. 

4.2.2 Expected complexity estimates.

Theorem 4.8. Let $f \in \mathcal{P}^C_{\mathcal{D}}$ be a random bit polynomial with $\tau(f) \geq \Omega(\lg d + u(f))$. Then, using descartes, the expected number of subdivision steps to isolate the real roots in $I = (-1, 1)$ is

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

The expected bit complexity of descartes is

$$\tilde{O}(d(\ell^3 + (d + u(f))^3 + d^2(1 + u(f))^4)).$$

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

$$\tilde{O}(d + d^3).$$

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 $\tilde{O}(d^2 \ell^2)$ for descartes [14], the bit complexity of descartes at $f$ is at most

$$\tilde{O}
\left(
\begin{array}{c}
\min(\lg C_{\mathcal{C}}(f), d^3 \tau(f))
\end{array}
\right) + \min(\lg C_{\mathcal{C}}(f), d^2 \tau(f)^3).$$

In turn we can bound by

$$\tilde{O}
\left(
\begin{array}{c}
\min(\lg C_{\mathcal{C}}(f), d^3 \tau(f))
\end{array}
\right) + \min(\lg C_{\mathcal{C}}(f), d^2 \tau(f)^3).$$

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

$$\mathbb{E}(\ell^3 \min(\lg C_{\mathcal{C}}(f), d^3 \tau(f)))$$

and

$$\mathbb{E}(\ell)(\ell^2 \min(\lg C_{\mathcal{C}}(f), d^2 \tau(f)^2)).$$

Let us show how to bound the first, because the second one is the same. By the Cauchy-Bunyakovskiy-Schwarz inequality, $\mathbb{E}(\ell)^2 \min(\lg C_{\mathcal{C}}(f), d^3 \tau(f))$ is bounded by

$$\sqrt{\mathbb{E}(\ell)^4 \mathbb{E}(\min(\lg C_{\mathcal{C}}(f), d^3 \tau(f))^2)}.$$ 

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

\[\square\]
REFERENCES

[1] A. Alesina and M. Galuzzi. A new proof of Vincent’s theorem. *L’Enseignement Mathématique*, 44:219–256, 1998.

[2] S. Arora and B. Babai. *Computational complexity: a modern approach*. Cambridge University Press, Cambridge, 2009.

[3] R. Becker, M. Sagraloff, V. Sharma, and C. Yap. A near-optimal subdivision algorithm for complex root isolation based on the Pellet test and Newton iteration. *Journal of Symbolic Computation*, 86:51–96, 2018.

[4] A. Bini and G. Fiorentino. Design, analysis, and implementation of a multi-precision polynomial rootfinder. *Numerical Algorithms*, 23(2):127–173, 2000.

[5] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and real computation*. Springer-Verlag, New York, 1998. With a foreword by Richard M. Karp.

[6] P. Bürgisser and F. Cucker. *Condition: The geometry of numerical algorithms*, volume 349 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer, Heidelberg, 2013.

[7] J. M. McNamee and V. Pan. *Numerical Methods for Roots of Polynomials—Part II*. Chapman & Hall/CRC, Boca Raton, Florida, 3nd edition, 2012.

[8] D. Castro, J. L. Montaña, L. M. Pardo, and J. San Martín. The distribution of Gröbner bases. *Journal of Symbolic Computation*, 47(2):153–166, 2012.

[9] J. R. Johnson, W. Krandick, K. Lynch, D. Richardson, and A. Ruslanov. High-performance implementations of the Descartes method. In *Proc. Annual ACM ISSAC*, pages 138–149, Springer, 2005.

[10] I. Z. Emiris, A. Galligo, and E. P. Tsigaridas. Random polynomials and expected complexity of bisection methods for real solving. In S. Watt, editor, *Proc. 35th ACM Int’l Symp. on Symbolic & Algebraic Comput. (ISSAC)*, pages 235–242, Munich, Germany, July 2010. ACM.

[11] J. Tonelli-Cueto and E. Tsigaridas. Condition Numbers for the Cube. I: Univariate Polynomials and Hypersurfaces. In *Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pages 235–242, Munich, Germany, July 2010. ACM.

[12] J. Tonelli-Cueto and E. Tsigaridas. Condition Numbers for the Cube. II: High-Dimensional Polynomials and Hypersurfaces. In *Proceedings of the 46th International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pages 235–242, Munich, Germany, July 2010. ACM.

[13] E. C. Titchmarsh. *The theory of functions*. Oxford University Press, Oxford, 2nd edition, 1939.

[14] V. Y. Pan. *Univariate polynomials: Nearly optimal algorithms for numerical arithmetic*. Journal of Symbolic Computation, 33(5):701–733, 2002.

[15] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, Cambridge, UK, 2nd edition, 2003.

[16] D. W. Gillies. *The geometry of numerical algorithms*. Springer-Verlag, New York, 1998. With a foreword by Richard M. Karp.

[17] V. T. Potts. *The theory of functions*. Oxford University Press, Oxford, 1935.

[18] T. Roughgarden. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2014.

[19] M. Rudelson and R. Vershynin. The Littlewood-Offord problem and invertibility of random matrices. *Adv. Math.*, 238(2):600–633, 2010.

[20] M. Sagraloff and K. Mehlhorn. Computing real roots of real polynomials. *Journal of Symbolic Computation*, 73:46–86, 2016.

[21] G. Moroz. New data structure for univariate polynomial approximation and applications to root isolation, numerical multipoint evaluation, and other problems, 6 2021. arXiv:2106.02505.

[22] N. Obreshkov. *Zeros of polynomials*. Marin Drinov Academic Publishing House, 2003. Translation from the Bulgarian.

[23] V. Y. Pan. Solving a polynomial equation: some history and recent progress. *SIAM review*, 39(2):187–220, 1997.

[24] V. Y. Pan. Approximating complex polynomial zeros: modified Weyl’s quadrature construction and improved newton’s iteration. *Journal of Complexity*, 16(1):213–264, 2000.

[25] V. Y. Pan. Univariate polynomials:Nearly optimal algorithms for numerical factorization and rootfinding. *Journal of Symbolic Computation*, 33(5):701–733, 2002.

[26] V. Y. Pan. New Progress in Polynomial Root-finding. arXiv:1805.12042 [cs], Feb. 2021. arXiv: 1805.12042.

[27] V. Y. Pan and E. P. Tsigaridas. On the boolean complexity of real root refinement. In Proc. of the 38th International Symposium on Symbolic and Algebraic Computation (ISSAC). ACM Press, 2013.

[28] S. Teichholz and J. Noak. Efficient全面提升al of polynomial’s real roots. *J. Comput. & Applied Math.*, 162(1):33–50, 2004.

[29] A. Alesina and M. Galuzzi. *A new proof of Vincent’s theorem*. L’Enseignement Mathématique, 44:219–256, 1998.

[30] J. P. Dedieu. Points fixes, zéros et la méthode de Newton, volume 54 of *Mathématiques & Applications (Berlin) [Mathematics & Applications]*. Springer, Berlin, 2006.

[31] R. G. Downey and M. R. Fellows. *Fundamentals of parameterized complexity*. Texts in Computer Science. Springer, London, 2013.

[32] D. Castro, J. L. Montaña, L. M. Pardo, and J. San Martín. The distribution of Gröbner bases. *Journal of Symbolic Computation*, 47(2):153–166, 2012.

[33] I. Z. Emiris, B. Mourrain, and E. P. Tsigaridas. Real Algebraic Numbers: Complex-Principles of Mathematical Sciences]. Springer, Heidelberg, 2013.

[34] I. Z. Emiris, V. Y. Pan, and E. P. Tsigaridas. Algebraic algorithms. In T. Gonzalez, editor, *Computing Handbook Set - Computer Science*, volume I, chapter 10. CRC Press Inc., Boca Raton, Florida, 2nd edition, 2012.

[35] A. Eigenwillig, L. Kettner, W. Krandick, K. Mehlhorn, S. Schmitt, and N. Wernert. A descartes algorithm for polynomials with bit-stream coefficients. In *International Workshop on Computer Algebra in Scientific Computing (CASC)*, pages 138–149, Springer, 2005.

[36] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, Cambridge, U.K., 2nd edition, 2003.

[37] J. Tonelli-Cueto and E. Tsigaridas. Condition Numbers for the Cube. I: Univariate Polynomials and Hypersurfaces. In *Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pages 235–242, Munich, Germany, July 2010. ACM.

[38] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge Univ. Press, Cambridge, UK., 2nd edition, 2003.

[39] J. H. Wilkinson. Some comments from a numerical analyst. *J. Assoc. Comput. Mach.*, 18:137–147, 1971.

[40] C. K. Yap and M. Sagraloff. A simple but exact and efficient algorithm for complex root isolation. In *Proc. 36th International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pages 355–360, 2011.