ON SOLVING UNIVARIATE SPARSE POLYNOMIALS IN LOGARITHMIC TIME

J. MAURICE ROJAS* AND YINYU YE†

ABSTRACT. Let f be a degree D univariate polynomial with real coefficients and exactly m monomial terms. We show that in the special case $m=3$ we can approximate within ε all the roots of f in the interval $[0,R]$ using just $O\left(\log(D)\log\left(D\log\frac{R}{\varepsilon}\right)\right)$ arithmetic operations. In particular, we can count the number of roots in any bounded interval using just $O(\log^2 D)$ arithmetic operations. Our speed-ups are significant and near-optimal: The asymptotically sharpest previous complexity upper bounds for both problems were super-linear in D, while our algorithm has complexity close to the respective complexity lower bounds. We also discuss conditions under which our algorithms can be extended to general m, and a connection to a real analogue of Smale's $17^{\underline{\text{th}}}$ Problem.

1. INTRODUCTION

Real-solving — the study of solving systems of polynomial equations over the real numbers — occupies a curious position within computational algebraic geometry. From the point of view of computational complexity, classical algebraic geometry has left real-solving almost completely untouched. For example, rigorous lower bounds for the arithmetic complexity of finding approximations to the roots of polynomial systems didn't appear until the work of Renegar in the late 1980's [Ren87, Ren89], and finding optimal bounds continues to be the subject of much active research [BMST97, BCSS98, MP99, Roj99b, GLS01, MPR03].

As for counting real roots, there is a beautiful result that one can bound their number independently of the degrees of the underlying polynomials: Askold Khovanski proved an explicit upper bound singly exponential in the number of variables and the total number of monomial terms [Kho91]. While this **fewnomial** bound is far from optimal in higher dimensions [LRW03], it is significantly smaller than the number of complex roots when the underlying polynomials have sufficiently high degree. (Sparse polynomials are sometimes also known as lacunary polynomials and, over $\mathbb{R}$, are a special case of fewnomials — a more general class of analytic functions of parametrized complexity [Kho91].)

We are then naturally lead to suspect that a similar improvement is possible for the harder problem of approximating the real roots. So can one solve **sparse** polynomial systems over the real numbers significantly faster than via the usual algorithms based on complex algebraic geometry? The existence of general speed-ups of this nature is still an open problem, even in the univariate case: For example, until the present paper, it was still unknown whether the real roots of a univariate trinomial of degree D could be approximated within a number of arithmetic operations sub-linear in D [MP99, Roj99b, GLS01, MPR03].

We now answer this last question affirmatively as follows: Let $I \subseteq \mathbb{C}$ be any subset, $f \in \mathbb{C}[x_1]$ a degree D polynomial, and suppose $\zeta_1, \ldots, \zeta_m$ are all the distinct roots of f in the region I. By **ε-approximating the roots of f in I** we will mean finding complex numbers $z_1, \ldots, z_m \in I$ such that for any $i \in \{1, \ldots, m\}$, we have $|\zeta_i - z_j| < \varepsilon$ for some $j \in \{1, \ldots, m\}$.

Date: May 18, 2004.

⋆ Partially funded by Hong Kong/France PROCORE Grant #9050140-730, a Texas A&M University Faculty of Science Grant, and NSF Grants DMS-0138446 and DMS-0211458.

† Partially funded by NSF Grant DMS-9703490 and the City University of Hong Kong. The second author's work was done while visiting the Department of Mathematics at the City University of Hong Kong.

(So ε-approximating in I implies a correct root count in I as well.) When $I \subseteq \mathbb{R}$ and $f \in \mathbb{R}[x_1]$, we will further stipulate that the z_i all lie in $\mathbb{R}$.

Theorem 1. *Let $R, \varepsilon > 0$ and suppose $f \in \mathbb{R}[x_1] \setminus \{0\}$ has degree D and at most 3 monomial terms. Then we can ε-approximate all the roots of f in the closed interval $[0, R]$ using just $O\big(\log(D) \log\big(D \log \frac{R}{\varepsilon}\big)\big)$ arithmetic operations. In particular, we can count exactly the number of roots of f in any bounded interval using just $O(\log^2 D)$ arithmetic operations.*

As is standard, we count arithmetic operations as field operations in the field over $\mathbb{Q}$ generated by the coefficients of f. Note also that our underlying algorithm handles degenerate roots (i.e., roots of multiplicity > 1) with no difficulty.

Remark 1. *Throughout this paper, all O-constants and Ω-constants are absolute and effectively computable.* $\diamond$

Remark 2. *Our speed-ups are significant: The current asymptotically sharpest (sequential worst-case) arithmetic complexity upper bound for ε-approximating the roots of a general degree D univariate polynomial in the open disc $\{z \mid |z| \leq R\}$ is $O\big(D \log^5(D) \log\log \frac{R}{\varepsilon}\big)$* [BP94, NR96]. *The analogous upper bound for counting real roots in an interval is $O\big(D \log^2(D) \log\log D\big)$, via the technique of* **Sylvester-Habicht sequences** [Roy96, LM01]. *In particular, no sharper upper bounds were known before for the problems considered in Theorem 1 above.* $\diamond$

Remark 3. *Our speed-ups are also near-optimal: For any fixed $D, R \geq 2$, our ε-approximation algorithm matches (up to an asymptotically constant multiple) the $\Omega\big(\log\log \frac{1}{\varepsilon}\big)$ arithmetic complexity lower bound known for ε-approximating the roots of $x^2 - N$ where $1 \leq N \leq 2$* [BMST97]. *As for counting the roots, there do not appear to be any explicit lower bounds known for the arithmetic complexity of counting the real roots of m-nomials. However, one should note that the arithmetic complexity of just evaluating a degree D monomial is $\Omega(\log D)$ in the worst case* [dMS96, Mor97]. $\diamond$

Our algorithms are based on an earlier hybrid algorithm of the second author which combines bisection and Newton iteration, a new observation on the **Sturm sequences** of trinomials (Theorem 3 of Section 1.2), and some analytic estimates (Theorem 5 of Section 2). We in fact give a more general algorithm that applies to certain univariate m-nomials and implies a deeper complexity result in Theorem 6 of Section 2. However, for $m \geq 4$, there are two main obstructions to showing that our more general algorithm has complexity sub-linear in D. We refer to these obstructions as Problems A and B, and describe the first now.

Problem A. *Is there an absolute contant κ such that one can count exactly the number of real roots (in any input interval) of arbitrary m-nomials of degree D using, say, just $O(\log^{\kappa m} D)$ arithmetic operations?* $\diamond$

One may note that we have hedged our bets in Problem A by letting the complexity bound increase exponentially in m. This is motivated by the **NP**-hardness of the multivariate analogue of Problem A.

Proposition 1. *Suppose $f \in \mathbb{Z}[x_1, \ldots, x_n]$ is an m-nomial and we measure the* **size** *of f as the total number of bits necessary to write the binary expansions of all the coefficients and exponents of f. Then it is* **NP***-hard (in the classical Turing sense, using the preceding notion of size) to decide whether f has a real root or not.* ■

The proof follows immediately from a standard reduction of the special case of n-variate polynomials of degree 6 to **3-SAT** (in conjunctive normal form) [GJ79]. So **NP**-hardness starts at $m \geq \Omega(n^6)$, if not earlier. Via a more intricate argument, one can show that **NP**-hardness starts at $m \geq 6n+6$, if not earlier [RS04]. To obtain the coarser lower bound, one respectively substitutes 0, 1, $1-x$, $x+y-xy$, and a new equation, for each `False`, `True`, $\neg x$, $x \vee y$, and conjunction in a conjunctive normal form. By adding equations of the form $x(1-x)=0$ to force the variables to be 0-1, one can thus change any **3-SAT** instance into a real system of equations of degree ≤ 3. Summing the squares of the equations, one then obtains a single degree 6 polynomial whose feasibility is equivalent to the existence of a satisfying assignment for one's original **3-SAT** instance.

So root counting is subtle for n-variate m-nomials, and this persists even for fixed n: We present some examples in Section 1.2 revealing that the classical approach of Sturm sequences will most likely need to be abandonded for $n=1$ and $m>3$. Nevertheless, speed-ups for m **fixed** still appear possible and are of considerable practical interest for large D.

Noting that ε-approximation extends naturally to polynomial systems (by ε-approximating each coordinate separately), a consequence of our univariate trinomial algorithm is the following result which may be of use for solving general pairs of bivariate trinomials.

Theorem 2. *Suppose $f \in \mathbb{R}[x_1, x_2]$ has exactly 3 monomial terms, D is the degree of f, and $\delta > 0$ is any constant. Then, using just $O\left(\log(D) \log\left(D \log \frac{R}{\varepsilon}\right)\right)$ arithmetic operations and $O\left(\log^{2+\delta} D\right)$ bit operations, we can ε-approximate all isolated inflection points, vertical tangents, and singular points of the curve $\{(x_1, x_2) \in [0, R]^2 \mid f(x_1, x_2) = 0\}$.*

The asymptotically sharpest (sequential worst-case) arithmetic complexity upper bound for ε-approximating the roots of a system of n polynomial equations in n variables are at best polynomial in $\log\log \frac{R}{\varepsilon}$ and a quantity sometimes attaining D^n (see, e.g., [MP99, Roj99b, GLS01, MPR03]). The importance of Theorem 2 lies in that it is a first step toward a practical two-dimensional analogue of bisection. The latter in turn is a first step toward generalizing our algorithms here to higher dimensions.

We leave the bit complexity of our algorithms for a future paper. In particular, unless stated otherwise, our underlying model of computation here is the BSS model (with inequality) over $\mathbb{R}$ [BCSS98]. Those unfamiliar with this model can simply think of such a machine as one's favorite lap-top computer, augmented with unlimited memory, a flawless operating system, and additional registers that allow arithmetic and inequality checking with real numbers as well as bits. (Strictly speaking, a BSS machine is different — but still polynomial-time equivalent to — this simplified model.) However, let us touch upon a deeper question: The connection of our work to Smale's notion of **approximate roots** (see, e.g., [Sma86, BCSS98] and Section 2 below).

Definition 1. *Suppose $f_1, \ldots, f_n \in \mathbb{R}[x_1, \ldots, x_n]$, $F := (f_1, \ldots, f_n)$, and the total number of distinct exponent vectors in the monomial term expansions of $f_1, \ldots, f_n$ is m. We then call F a* **real m-sparse $n \times n$ polynomial system**. $\diamond$

Definition 2. *Let $B \subseteq \mathbb{C}^n$ be any open ball and $F : B \longrightarrow \mathbb{C}^n$ an analytic function. Also, given any $z_0 \in B$, let us define the sequence $(z_i)_{i=0}^{\infty}$ by $z_i := z_{i-1} - \frac{dF}{dz}\big|_{z_{i-1}}^{-1} F(z_{i-1})$ for all $i \in \mathbb{N}$. Then, if there is a root $\zeta \in B$ of F such that $|z_{i+1} - \zeta| \leq 8(\frac{1}{2})^{2^i} |z_0 - \zeta|$ for all i (i.e., if Newton iteration for F, starting at z_0, converges quadratically to a root of F), we call z_0 an* **approximate root** *of F with* **associated root** ζ. $\diamond$

Real Analogue of Smale's 17$^{\underline{\text{th}}}$ Problem. *For fixed n, can all real roots of an m-sparse $n \times n$ polynomial system be found approximately, on the average, in polynomial time with a uniform algorithm? More precisely, let F be an m-sparse $n \times n$ polynomial system with maximal exponent D and coefficients that are, say, independent standard real Gaussian random variables. Is there a uniform algorithm that finds a set of approximate roots close to all the real roots of F, with average-case arithmetic complexity $O((m \log D)^\nu)$ for some constant ν depending only on n?* $\diamond$

The polynomial system $((x_1-1)(x_1-2), \ldots, (x_n-1)(x_n-2))$ clearly shows that fixing n is necessary in our real analogue above. The appellation **uniform** merely emphasizes that there be a single algorithm, with all steps explicit and constructive, which works for all inputs [BCSS98].

The original statement of Smale's 17$^{\underline{\text{th}}}$ Problem [Sma98] (see also [Sma00, Pg. 287]) differs from our analogue above as follows: (1) n is allowed to vary (so one seeks an **absolute** constant ν), (2) one averages over choices of complex coefficients, and (3) one instead asks for a **single** complex approximate root. Smale also left the underlying probability distribution unspecified. We observe that a positive answer to the following variant of Problem A would be quite useful in the direction of our real analogue:

Stochastic Version of Problem A: *Suppose f is a univariate m-nomial of degree $\leq D$ with coefficients that are real standard Gaussian random variables. Is there a uniform algorithm that counts exactly the number of roots of f (in any input interval), with* **average-case** *arithmetic complexity $O((m \log D)^\kappa)$ for some absolute constant κ?* $\diamond$

Smale's original 17$^{\underline{\text{th}}}$ Problem remains unsolved, although a partial affirmative answer (an algorithm containing a non-constructive step which may be called many times) was found by Shub and Smale in the mid-1990's [SS94, BCSS98].

1.1. Earlier Work on Approximating Roots.

When can we solve a polynomial system in time polylogarithmic in the degree of the underlying complex algebraic set? An affirmative answer is trivial for the special case of a single polynomial with ≤ 1 monomial term, and was known at least since the mid-1970's for the (univariate) binomial case, e.g., [Bre76] and [Ye94, Sec. 4]. On the other hand, little seems to be known about the case of 3 or more monomial terms: To the best of the authors' knowledge, the only result close in spirit to Theorem 1 is a result of Daniel Richardson [Ric93] implying that the arithmetic complexity of counting the number of real roots of $c_1 + c_2 x^d + c_3 x^D$ (with $0 < d < D$) is polynomial in d and $\log D$.

Not much else seems to be known about the intrinsic complexity of real solving, or even real root counting, for univariate polynomials with 4 or more monomial terms. One known speed-up over the usual univariate algorithms over $\mathbb{C}$ is [Roj00a, Main Theorem 1.2] which gives an arithmetic complexity bound of $O\left(m \log(D) \log\left(D \log \frac{R}{\varepsilon}\right)\right)$ for ε-approximating the roots of

$$c_1 x^{a_1} + \cdots + c_k x^{a_k} - c_{k+1} x^{a_{k+1}} - \cdots - c_m x^{a_m},$$

when $0 \leq a_1 \leq \cdots \leq a_m = D$ are integers and all the c_i are positive real numbers.

It is also not difficult via the results above to construct various systems of multivariate trinomials which admit super-fast solving in the sense of Theorem 1 as well. For example, if one has $f_1 \in \mathbb{R}[x_1]$, $f_2 \in \mathbb{R}[x_1, x_2], \ldots f_n \in \mathbb{R}[x_1, \ldots, x_n]$, and all the f_i are trinomials of degree $\leq D$, then to solve $F := (f_1, \ldots, f_n)$ one can simply solve f_1 first and then recursively solve the resulting smaller system. Letting $\mathcal{N}_F$ denote the number of roots in the nonnegative

orthant of such an F, it is easily checked that $\mathcal{N}_F \leq 2^n$ (see, e.g., [LRW03, Thm. 3, Part (c)]). We can then easily derive, via Theorem 1, an arithmetic complexity upper bound of $O\left(\mathcal{N}_F \log^2(D) \log\log \frac{R}{\varepsilon}\right)$ for the more general problem where we instead ε-approximate all the roots of F in the **orthant-wedge**

$$W_R^n := \left\{(x_1, \ldots, x_n) \in \mathbb{R}^n \;\middle|\; x_1^2 + \cdots + x_n^2 \leq R^2 \;,\; x_i \geq 0 \;\text{ for all }\; i\right\}.$$

Extending these results to general trinomial systems, not to mention general sparse systems, remains an open problem. Nevertheless, for a general **bi**nomial system F with exactly $\mathcal{D} < \infty$ roots in $\mathbb{C}^n$ (counting multiplicities), one can ε-approximate all its roots in $W_{\mathbb{R}}^n$ using just $O\left((\log \mathcal{D})\left(B^3 \log^2(n) + \log\log \frac{R}{\varepsilon}\right)\right)$ arithmetic operations and $O(n^3 B^{2+\delta} \log^{2+\delta} n)$ bit operations, where B is the total number of bits needed to write down the exponents of F and $\delta > 0$ is an arbitrary constant [Roj00a, Main Theorem 1.3]. (The bound in [Roj00a] is written in a slightly different manner and we have here taken the liberty of improving the bit complexity portion by employing a recent result of van der Kallen on computing the Hermite normal form of an integral matrix [vdK00].)

1.2. **A New Algebraic Observation and Earlier Work on Counting Roots.**
As for merely counting the real roots, there are still surprising gaps in our knowledge. For instance, the main general algebraic techniques for real root counting — Sturm-Habicht sequences [Stu35, Hab48, Roy96, LM01] and Hermite's Method [Her56, PRS93, Roy96] — originated in the $19^{\underline{\text{th}}}$ century and are still hard to improve upon.

Definition 3. *For any sequence of real numbers $s := (s_1, \ldots, s_k)$, the* **number of sign alternations of s**, $\mathbf{N_s}$, *is the number of pairs (j, j') with $1 \leq j < j' \leq k$, $s_j s_{j'} < 0$, and $s_i = 0$ for all $i \in \{j+1, \ldots, j'-1\}$. Also, for any polynomial $f \in \mathbb{R}[x_1]$, define the sequence $(p_i)_{i=0}^{\infty}$ where $p_0 := f$, $p_1 := f'$, $p_{i+2} := q_i p_{i+1} - p_i$, q_i is the quotient of $\frac{p_i}{p_{i+1}}$, and p_K is the last element of the sequence not identically equal to 0. We then call $(p_0, \ldots, p_K)$ the* **Sturm sequence** *of f.* $\diamond$

Lemma 1. [Stu35, Roy96] *Following the notation of Definition 3, let $a, b \in \mathbb{R}$ with $a \leq b$, $A := (p_0(a), \ldots, p_K(a))$, and $B := (p_0(b), \ldots, p_K(b))$. Then the number of distinct roots of f in the open interval (a, b) is exactly $N_A - N_B$. In particular, $[a_1 < \cdots < a_m$ and $c_1, \ldots, c_m \in \mathbb{R}] \Longrightarrow c_1 x^{a_1} + \cdots + c_m x^{a_m}$ has no more than $N_{(c_1, \ldots, c_m)}$ $(\leq m-1)$ positive roots.* ■

Remark 4. *The latter part of the above lemma is* **Descartes' Rule of Signs** *which dates back to the famous philosopher's 1637 book La Geometrie (see also* [SL54, Pg. 160]*).* $\diamond$

The following observation on the Sturm sequence of a trinomial, which we use in proving Theorem 1, may be of independent interest.

Theorem 3. *Following the notation above, suppose f has at most 3 monomial terms. Then $K \leq 3\lceil \log_2 D \rceil + 2$ and $(p_1, \ldots, p_K)$ consists solely of binomials, monomials, and/or constants. In particular, the entire Sturm sequence of f can be evaluated at any real number using just $O(\log^2 D)$ arithmetic operations.*

Extending our last theorem to polynomials with more monomials appears unlikely: First, note that the quotient of the division of two binomials can be quite non-sparse, e.g., $\frac{x^{N+1}-1}{x-1} = x^N + \cdots + x + 1$. So the expansion of any intermediate quotients must be avoided. Furthermore, the **tetra**nomial case already gives some indication that Sturm sequences may in fact have to be completely abandoned: the following example shows that, for all $D > 2$, the fourth element of the Sturm sequence of a degree $2D$ tetranomial can already be a $(D+1)$-nomial.

Example 1. *Consider the tetranomial* $p_0(x) := x^{2D} + x^{D+1} + x^D + 1$ *and its derivative* $p_1 := p_0' = 2Dx^{2D-1} + (D+1)x^D + Dx^{D-1}$. *The resulting Sturm sequence then continues with* $p_2 = -\frac{D-1}{2D}x^{D+1} - \frac{1}{2}x^D - 1$ *and from here it is easy to see (by a writing a simple recursion for the resulting long division) that the quotient* q_3 *of* $\frac{p_1}{p_2}$ *is a polynomial of degree* $D-2$ *with exactly* $D-1$ *monomial terms. Thus,* $p_3 := q_3 p_2 - p_1$ *has degree* D *and at least* $D+1$ *monomial terms.* $\diamond$

We are also willing to conjecture that the maximal length of the Sturm sequence of a degree $2D$ tetranomial is $\Omega(D)$. (`Maple` experiments have verified this up to $D = 150$.) Nevertheless, while the behavior of tetranomials is thus more complicated, this example need not rule out a more clever method to circumvent these difficulties.

1.3. **A Connection to Discriminants.**

If one insists on relying on Sturm sequences then one is naturally lead to the univariate sparse discriminant. Briefly, given an n-variate m-nomial f with indeterminate coefficient vector C and exponents contained in $A \subset \mathbb{Z}^n$, its **sparse discriminant** (or ***A*-discriminant**), $\Delta_A(f)$, is the unique (up to sign) irreducible polynomial in $\mathbb{Z}[C]\setminus\{0\}$ of lowest degree which vanishes whenever C is specialized so that f has a root in $(\mathbb{C}\setminus\{0\})^n$ in common with $\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}$ [GKZ94]. It is not hard to see that if (a) $n = 1$, (b) the coefficients of f are all constants, and (c) A is exactly the set of exponent vectors of f, then we have $\Delta_A(f) = 0 \iff K = 1$ in the Sturm sequence $(p_0, \ldots, p_K)$ of f. So the following is clear.

Proposition 2. *Suppose* $f \in \mathbb{R}[x_1]$ *has set of exponent vectors* A *and one can compute the signs of the Sturm sequence of* f *evaluated at any* $r \in \mathbb{R}$ *using just* $T(A)$ *arithmetic operations. Then one can decide the vanishing of* $\Delta_A(f)$ *using just* $T(A)$ *arithmetic operations.* ■

It would be quite enlightening to understand the converse of Proposition 2 and its possible obstructions.

Perhaps more than coincidentally — recalling Theorem 1 — $\Delta_A(f)$ **can** be computed in polynomial time when f is a trinomial: up to sign, the formula is simply

$$\Delta_{\{0,a_2,a_3\}}(c_1 + c_2 x^{a_2} + c_3 x^{a_3}) = a_3^{a_3} c_3^{a_2} c_1^{a_3-a_2} + (-1)^{a_3-1} a_2^{a_2} (a_3 - a_2)^{a_3-a_2} c_2^{a_3}$$

when a_2 and a_3 are relative prime and $0 < a_2 < a_3$ (see [GKZ94, Pg. 406] and Remark 7 of Section 3 below). Whether polynomial time complexity persists for general m-nomials is already non-trivial for the tetranomial case since the underlying discriminants become much more complicated. For example, up to sign, $\Delta_{\{0,6,10,31\}}(c_1 + c_2 x^6 + c_3 x^{10} + c_4 x^{31})$ is the following homogeneous 21-nomial of degree 35:

$$-2^{10}3^{21}5^{10}7^{21}c_1^4c_3^{31} - 2^{13}3^{24}5^57^{21}c_1^2c_2^5c_3^{28} - 2^{14}3^{27}7^{21}c_2^{10}c_3^{25} + 2^83^{15}5^67^{16}\cdot 19\cdot 31^6 11731 c_1^7c_2^3c_3^{23}c_4^2$$

$$-2^83^{13}\cdot 5\cdot 7^{13}\cdot 19\cdot 31^4\cdot 4931\cdot 11924843839 c_1^5c_2^8c_3^{20}c_4^2 - 2^63^{13}5^67^{13}31^{13}c_1^{12}c_2c_3^{18}c_4^4 - 2^83^{16}5^{12}7^{13}\cdot 17\cdot 31^2\cdot 3629537 c_1^3c_2^{13}c_3^{17}c_4^2$$

$$-2^63^9\cdot 5\cdot 7^{11}\cdot 29\cdot 31^{11}\cdot 6361\cdot 7477163 c_1^{10}c_2^6c_3^{15}c_4^4 - 2^{10}3^{19}5^{22}7^{13}c_1c_2^{18}c_3^14c_4^2 - 2^63^85^{12}7^731^9\cdot 857\cdot 295269701 c_1^8c_2^{11}c_3^{12}c_4^4$$

$$-2^73^8\cdot 5\cdot 7^731^{18}\cdot 3327253 c_1^{15}c_2^4c_3^{10}c_4^6 - 2^63^45^{22}7^431^7160730667473 c_1^6c_2^{16}c_3^9c_4^4 + 2^43^35^{12}7^4\cdot 11\cdot 29\cdot 31^{16}\cdot 233\cdot 1559 c_1^{13}c_2^9c_3^7c_4^6$$

$$-2^65^{32}31^5\cdot 248552699041 c_1^4c_2^{21}c_3^6c_4^4 - 2^23^457^4\cdot 11\cdot 29\cdot 31^{25}c_1^{20}c_2^2c_3^5c_4^8 - 2^5\cdot 35^{22}31^{14}\cdot 113\cdot 317\cdot 461 c_1^{11}c_2^{14}c_3^4c_4^6$$

$$-2^73^35^{42}31^3\cdot 41\cdot 367 c_1^2c_2^{26}c_3^3c_4^4 - 2^6\cdot 3\cdot 5^{12}\cdot 19\cdot 31^{23}c_1^{18}c_2^7c_3^2c_4^8 + 2^43^45^{32}31^{12}c_1^9c_2^{19}c_3c_4^6 - 31^{31}c_1^{25}c_4^{10} - 2^63^65^{50}c_2^{31}c_4^4$$

whose largest coefficient has 47 digits. So we pose the following problem:

Univariate Discriminant Complexity Problem. *Given any finite subset* $A \subset \mathbb{Z}$, *define* $\sigma(A)$ *to be the total number of bits needed to write the binary expansions of all the points of* A. *Can one decide the vanishing of* A*-discriminants (for, say, specializations of the coefficients in* $\mathbb{C}$*) using a number of arithmetic operations polynomial in* $\sigma(A)$*?* $\diamond$

This highlights an embarassing gap in what is known about the complexity of discriminant computation: it is known that deciding the vanishing of sparse discriminants for **bi**variate polynomials can be done in polynomial time on a Turing machine (resp. BSS machine over $\mathbb{C}$) only if $\mathbf{NP} \subseteq \mathbf{P}$ (resp. $\mathbf{NP} \subseteq \mathbf{BPP}$) [Roj00a, Main Theorem 1.4], but no such hardness result is known for the univariate case. The preceding inclusions (see [Pap95] for a beautiful introduction to complexity theory) are currently considered quite unlikely. On the positive side, it is known that A-discriminants (for arbitrary finite $A \subset \mathbb{Z}^n$) can be evaluated using a number of arithmetic operations polynomial in $\sigma(A)$ when A has $n+1$ or fewer points [GKZ94, Prop. 1.8, pp. 274–275]. Also, the vanishing of A-discriminants can be decided within the complexity class $\mathbf{P}^{\mathbf{NP}^{\mathbf{NP}}}$ under a number-theoretic hypothesis strictly weaker than the Generalized Riemann Hypothesis [Roj04a].

2. Speed-Ups Through a Variant of α-theory

Checking whether a given point is an approximate root of a given polynomial can be done quite efficiently, thanks to the seminal work of Smale in [Sma86]. Let us now formalize a refined version of this fact.

Definition 4. *For any analytic function $f : \mathbb{C} \longrightarrow \mathbb{C}$, let $\gamma(f,x) := \sup_{k \geq 2} \left| \frac{f^{(k)}(x)}{k!f'(x)} \right|^{\frac{1}{k-1}}$.* $\diamond$

Remark 5. *It is worth noting that $1/\gamma(f,x_0)$ is a lower bound for the radius of convergence of the Taylor series of f about x_0, so $\gamma(f,x_0)$ is finite whenever f is nonsingular at x_0* [BCSS98, Prop. 6, Pg. 167]. $\diamond$

Recall that a function $g : U \longrightarrow \mathbb{R}$, defined on some connected domain $U \subseteq \mathbb{R}$ is **convex** iff $g(\lambda x + (1-\lambda)y) \leq \lambda g(x) + (1-\lambda)g(y)$ for all $x, y \in U$ and $\lambda \in [0,1]$. $\diamond$

Theorem 4. [Ye94, Thm. 2] *Suppose f is convex and analytic on the open interval $(0,R)$. Also let $r \in (0,R)$ be any root of f and suppose that there is an $\bar{\alpha} > 0$ such that $x\gamma(f,x) \leq \bar{\alpha}$, for all x in the closed interval $\left[\frac{r}{1+\frac{1}{8\bar{\alpha}}}, \frac{r}{1-\frac{1}{8\bar{\alpha}}}\right]$. Then for all $z \in (0,R)$:*

1. *If f is monotonically decreasing in $[z,r]$ and $r \in \left[z, (1+\frac{1}{8\bar{\alpha}})z\right]$ then z is an approximate root of f.*
2. *If f is monotonically increasing in $[r,z]$ and $r \in \left[(1-\frac{1}{8\bar{\alpha}})z, z\right]$ then z is an approximate root of f.* ■

Proposition 3. [Ye94, Prop. 1] *Suppose $f_1, f_2 : U \longrightarrow \mathbb{R}$ satisfy the hypotheses of Theorem 4, with $\bar{\alpha}_1$ and $\bar{\alpha}_2$ respectively playing the role of $\bar{\alpha}$, and that f_1 and f_2 are either both increasing or both decreasing. Then $f_1 + f_2$ also satisfies the hypotheses of Theorem 4, with $\bar{\alpha} = \max\{\bar{\alpha}_1, \bar{\alpha}_2\}$.* ■

Proposition 4. *For any $D \in \mathbb{R} \setminus \{0\}$ and $c \in \mathbb{R}$, the function $f : [0,\infty) \longrightarrow \mathbb{R}$ defined by $x \mapsto x^D - c$ satisfies $x\gamma(f,x) = \left|\frac{D-1}{2}\right|$ for all $x > 0$.* ■

The second author observed the special case $D \in \mathbb{Z} \setminus \{0\}$ earlier in [Ye94, Example 1]. The proof their applies to aribtrary real D as well.

An additional technical result we will need is an analytic estimate which globalizes Theorem 4. But first let us observe an oscillation property we will need for our main algorithm.

Definition 5. *For any $D, m \in \mathbb{N}$ define $\mathcal{F}(D,m)$ to be the family of real t-nomials f of degree $\leq D$ with $t \leq m$. For any $f \in \mathcal{F}(D,m) \setminus \mathbb{R}$, let $u_0(f) < \cdots < u_{N(f)}(f)$ be the ordered sequence*

comprised of 0, *the roots of* f' *in* $(0,\infty)$, *and* ∞. *We then say that* f *is* **dampened** *iff for all* $i\in\{0,\ldots,N(f)-1\}$, f'' *has at most one root in* $(u_i(f),u_{i+1}(f))$. $\diamond$

We of course have $\mathcal{F}(D_1,m_1)\subseteq\mathcal{F}(D_2,m_2)$ whenever $D_1\leq D_2$ and $m_1\leq m_2$. Rolle's Theorem applied to f' tells us that $(u_i(f),u_{i+1}(f))$ (for $i\in\{1,\ldots,N(f)-2\}$) always contains at least one root of f''. It is then easy to show via Descartes' Rule and a routine calculation that m-nomials are always dampened when $m\leq 4$.

The invariant defined below generalizes the local quantity maximized in Theorem 4 above, and thus helps enforce the accelerated convergence of Newton's method in certain cases of interest.

Definition 6. *For any* $D,m\geq 2$, *any dampened* $f\in\mathcal{F}(D,m)$, *and* $i\in\{1,\ldots,N(f)-2\}$, *define* $v_i(f)$ *to be the unique root of* f'' *in* $(u_i(f),u_{i+1}(f))$, $v_{-1}(f):=0$, *and* $v_{N(f)}(f):=\infty$. *Also define* $v_0(f)$ *to be* $u_1(f)$ *or the unique root of* f'' *in* $(u_0(f),u_1(f))$, *according as* f'' *is non-vanishing in* $(u_0(f),u_1(f))$ *or not. Also define* $v_{N(f)-1}(f)$ *to be* ∞ *or the unique root of* f'' *in* $(u_{N(f)-1}(f),u_{N(f)}(f))$, *according as* f'' *is non-vanishing in* $(u_{N(f)-1}(f),u_{N(f)}(f))$ *or not. Finally, define* $\bar{\alpha}(D,m)$ *to be*

$$\sup_{\substack{f\in\mathcal{F}(D,m)\\ f\text{ dampened}\\ f(0)\neq 0\text{ and }f'(0)=0}}\ \sup_{i\in\{0,\ldots,N(f)-1\}}\ \sup_{\substack{x\in(v_{i-1}(f),v_i(f))\setminus\{u_i(f)\}\\ (v_{i-1}(f),v_i(f))\setminus\{u_i(f)\}\text{ contains a root of }f}}\{|x-u_i(f)|\gamma(f,x)\}\ .\ \diamond$$

While $\bar{\alpha}(D,m)$ may be hard to compute exactly, we can at least bound it above and below explicitly in some cases of interest.

Theorem 5. *For any* $D,m\geq 2$ *we have* $\bar{\alpha}(D,m)\leq\bar{\alpha}(D,m+1)$. *In particular,*

$$\frac{D-1}{2}=\bar{\alpha}(D,2)\leq\bar{\alpha}(D,3)\leq(D-1)(D-2)/2.$$

One application of our $\bar{\alpha}$-invariant is the following generalization of Theorem 1.

Definition 7. *For any* m*-nomial* f, *let* $\delta(f)$ *be the smallest exponent appearing in* f. *Let us then define the operators* $S:\mathcal{F}(D,m)\longrightarrow\mathcal{F}(D,m)$, $L_1:\mathcal{F}(D,m)\longrightarrow\mathcal{F}(D-1,m-1)$, *and* $L_2:\mathcal{F}(D,m)\longrightarrow\mathcal{F}(D-2,m-1)$ *by* $S(f):=x^{-\delta(f)}f$, $L_1(f):=\frac{d}{dx}S(f)$, *and* $L_2(f):=\frac{d^2}{dx^2}S(f)$.

Theorem 6. *Following the notation of Definition 6, suppose* $f(x):=c_1x^{a_1}+c_2x^{a_2}+\cdots+c_mx^{a_m}$ *is a real* m*-nomial such that*

- **(a)** $m\geq 2$ *and* $0\leq a_1<\cdots<a_m$.
- **(b)** $a_{i+1}=a_i+1$ *and* $a_{j+1}=a_j+1$ *for some* $i\neq j\Longrightarrow a_m-a_1=2$.
- **(c)** *All the polynomials* $\{(S\circ L_{e_1}\circ\cdots\circ L_{e_k})(f)\mid k\in\{0,\ldots,m-1\},\ e_i\in\{1,2\}\}$ *are dampened.*

Then one can find ε*-approximations to all the roots of* f *in* $[0,R]$ *using just*

$$O\left(2^m\left\{\log(D)\log\left(\bar{\alpha}(D,m)\log\frac{R}{\varepsilon}\right)+m^2\mathcal{K}(D,m)\right\}\right)$$

arithmetic operations, where $\mathcal{K}(D,m)$ *is any upper bound on the worst-case arithmetic complexity of counting the roots of an arbitrary* $f\in\mathcal{F}(D,m)$ *in an arbitrary interval. In particular, if* $f\in\mathcal{F}(D,4)$ *then the polynomials*

$$\{S(f),S(L_{e_1}(f)),S(L_{e_1}(L_{e_2}(f))),S(L_{e_1}(L_{e_2}(L_{e_3}(f))))\mid e_i\in\{1,2\}\}$$

are all dampened, for any $D\in\mathbb{N}$.

It is easily checked (recalling our observations after Definition 5) that all binomials, all trinomials, and all tetranomials save those of the form $c_1x^d+c_2x^{d+1}+c_3x^{D+d}+c_4x^{D+d+1}$ satisfy the hypotheses of our complexity bound above. Theorem 6 thus opens up the possibility of super-fast m-nomial solving when $m>3$ is fixed. In particular, good upper bounds on $\mathcal{K}(D,m)$ and $\bar{\alpha}(D,m)$ appear to be a fundamental first step. Our aforementioned Problem A addresses $\mathcal{K}(D,m)$ so consider the following new problem.

Problem B. *Is there an absolute contant κ' such that $\bar{\alpha}(D,m)=O\left(\log^{\kappa' m} D\right)$?* $\diamond$

An interesting related problem, in the spirit of [MR04], is whether one can bound from below the probability that a random m-nomial of degree D (with $m\geq 4$ fixed) is dampened.

We state our underlying algorithms in the next section. We then prove Theorems 1, 6, and 3 in Section 4. We conclude with the proofs of Theorems 2 and 5 in Sections 5 and 6, respectively.

3. The Algorithm and Subroutines

The central algorithm we use to prove Theorems 1 and 6, `MNOMIALSOLVE`, is detailed below. Succinctly, the key idea behind `MNOMIALSOLVE` is to subdivide the input interval into sub-intervals on which $\pm f$ is convex and monotonic, along with some additional sub-intervals on which f is less well-behaved.

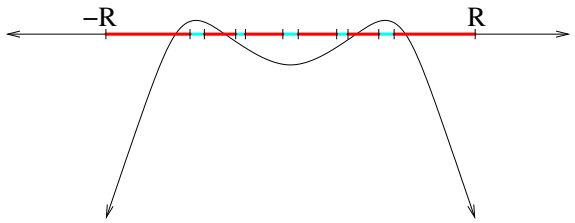

Figure 1 $[-R,R]$ can be expressed as a union of 6 intervals on which $\pm f$ is convex and monotonic, and 5 additional intervals of width $<2\varepsilon$ containing the roots of $f'f''$. $\diamond$

Then, using special properties of sparse polynomials, a suitable combination of bisection and Newton iteration (detailed in the subroutine `HYBRID`) yields ε-approximations to all the roots in $(0,R)$. (Checking whether 0 and/or R are roots can be done simply by evaluating there.) Each approximation is guaranteed to correspond to its own unique root, and the roots in intervals on which f behaves badly are certified via a subroutine called `FASTERCOUNT`. The aforementioned subroutines are described shortly after our main algorithm.

ALGORITHM `MNOMIALSOLVE`

INPUT Real numbers R and ε with $0<\varepsilon<R$, a real m-nomial $f(x):=c_1x^{a_1}+\cdots+c_mx^{a_m}$ satisfying the hypotheses of Theorem 6, and an upper bound $\bar{\alpha}^*$ on $\bar{\alpha}(D,m)$.

OUTPUT A (possibly empty) multiset $Z\subseteq\mathbb{R}$, such that every root of f in $[0,R]$ is ε-approximated by a unique $z_i\in Z$, and $\#Z$ is exactly the number of roots of f in $[0,R]$.

DESCRIPTION

Step 0 Set $Z:=\emptyset$.
Step 1 If $m=0$ then set $Z:=\mathbb{R}$ and STOP.
Step 2 If $m=1$ and $a_1>0$ then set $Z:=\{0\}$ and STOP.
Step 3 If $m=1$ and $a_1=0$ then STOP.
Step 4 If $a_1>0$ then set $Z:=\{0\}$, $f:=f/x^{a_1}$, $a_i:=a_i-a_1$ for all $i\geq 2$, and then $a_1:=0$.

Step 5 If $a_m = 1$ then set $Z := \{-c_1/c_2\}$ or $\emptyset$, according as $c_1 c_2 \leq 0$ or not, then STOP.

Step 6 If $a_m = 2$ then make the change of variables $x := x - \frac{c_2}{2c_3}$.

Step 7 If $f(R) = 0$ then append R to Z.

Step 8 If $m = 2$ and $f(0)f(R) \geq 0$, then STOP.

Step 9 If $m = 2$ and $f(0)f(R) < 0$, find an ε-approximation z to the unique root of f in $(0, R)$ via subroutine `HYBRID` (with input $(\varepsilon, R, \pm f, \bar{\alpha}^*)$, where the sign is chosen so that f is convex on $(0, R)$), set $Z := Z \cup \{z\}$, and STOP.

Step 10 Otherwise, if $a_2 > 1$, find via algorithm `MNOMIALSOLVE` an ordered sequence $u_0 < \cdots < u_{k_1}$ comprised of 0, a set of ε-approximations (of the correct cardinality) in $(0, R)$ for the roots of f' in $(0, R)$, and R. Then GOTO Step 12.

Step 11 If $a_2 = 1$ then redefine $f(x) := x^{a_3} f(1/x)$ and $\varepsilon := \min\left\{1, \frac{\varepsilon^3}{2}, \frac{1}{2R^2}\right\}$ and make the change of variables $x := 1/x$. GOTO Step 10.

Step 12 Define $v_{k_1} := R$.

Step 13 Using algorithm `FASTERCOUNT`, decide if f'' has a root in $[u_0 + \varepsilon, u_1 - \varepsilon]$. If so, let $L := u_1 - u_0 - 2\varepsilon$. Then, evaluating f'' at $u_0 + \varepsilon$ and $u_1 - \varepsilon$, and using algorithm `HYBRID` with input $(\varepsilon, L, \pm f, \alpha^*)$ (the sign chosen so that $\pm f$ is convex on $(u_0 + \varepsilon, u_1 - \varepsilon)$), define v_0 to be the ε-approximation found for the unique root of f'' in $[u_0 + \varepsilon, u_1 - \varepsilon]$. Otherwise, define v_0 to be u_1.

Step 14 Using algorithm `FASTERCOUNT`, decide if f'' has a root in $[u_{k_1-1} + \varepsilon, u_{k_1} - \varepsilon]$. If so, let $L := u_{k_1} - u_{k_1-1} - 2\varepsilon$, $g(x) := f(x + u_{k_1-1})$. Then, evaluating f'' at $u_{k_1-1} + \varepsilon$ and $u_{k_1} - \varepsilon$, and using algorithm `HYBRID` with input $(\varepsilon, L, \pm g, \alpha^*)$ (the sign chosen so that $\pm g$ is convex on $(u_{k_1-1} + \varepsilon, u_{k_1} - \varepsilon)$), define v_{k_1-1} to be the ε-approximation found for the unique root of f'' in $[u_{k_1-1} + \varepsilon, u_{k_1} - \varepsilon]$. Otherwise, define v_{k_1-1} to be R.

Step 15 For all $i \in \{1, \ldots, k_1 - 1\}$, count via algorithm `FASTERCOUNT` the number m_i of roots of f in $(u_i - \varepsilon, u_i + \varepsilon) \cap (0, R)$ closest to u_i and no other root of $f'f''$, and append m_i copies of u_i to Z.

Step 16 For all $i \in \{0, \ldots, k_1 - 1\}$, count via algorithm `FASTERCOUNT` the number n_i of roots of f in $(v_i - \varepsilon, v_i + \varepsilon) \cap (0, R)$ closest to v_i and no other root of $f'f''$, and append n_i copies of v_i to Z.

Step 17 FOR $i \in \{0, \ldots, k_1 - 1\}$ DO

Step 17(a) Evaluate f at $u_i + \varepsilon$ and $v_i - \varepsilon$. If $f(u_i + \varepsilon) = 0$ (resp. $f(v_i - \varepsilon) = 0$) then append $u_i + \varepsilon$ (resp. $v_i - \varepsilon$) to Z and GOTO Step 17(c).

Step 17(b) If $f(u_i + \varepsilon)f(v_i - \varepsilon) < 0$ then let $L_i := v_i - u_i - 2\varepsilon$ and $g_i(x) := f(x + u_i + \varepsilon)$. Then, using subroutine `HYBRID` with input $(\varepsilon, L_i, \pm g_i, \bar{\alpha}^*)$ (the sign chosen so that g_i is convex on $(0, v_i - u_i - 2\varepsilon)$), find an ε-approximation in $(u_i + \varepsilon, v_i - \varepsilon)$ to the unique root of f in $(u_i + \varepsilon, v_i - \varepsilon)$, and append this approximation to Z.

Step 17(c) Evaluate f at $v_i + \varepsilon$ and $u_{i+1} - \varepsilon$. If $f(v_i + \varepsilon) = 0$ (resp. $f(u_{i+1} - \varepsilon) = 0$) then append $v_i + \varepsilon$ (resp. $u_{i+1} - \varepsilon$) to Z and GOTO Step 18.

Step 17(d) If $f(v_i + \varepsilon)f(u_{i+1} - \varepsilon) < 0$ then let $L_i := u_{i+1} - v_i - 2\varepsilon$ and $g_i(x) := f(u_{i+1} - \varepsilon - x)$. Then, using subroutine `HYBRID` with input $(\varepsilon, L_i, \pm g_i, \bar{\alpha}^*)$ (the sign chosen so that g_i is convex on $(0, u_{i+1} - v_i - 2\varepsilon)$), find an ε-approximation in $(v_i + \varepsilon, u_{i+1} - \varepsilon)$ to the unique root of f in $(v_i + \varepsilon, u_{i+1} - \varepsilon)$, and append this approximation to Z.

Step 18 END FOR

Step 19 STOP

Remark 6. *In reality, one should replace the use of subroutine* `FASTERCOUNT` *above by the best algorithm attaining the complexity bound of* $\mathcal{K}(D, m)$ *alluded to in Theorem 6, whenever* $m \geq 4$. *We have omitted this detail above simply to keep the algorithm definite.* $\diamond$

SUBROUTINE HYBRID **(Compare** [Ye94, Pg. 277]**)**

INPUT $\varepsilon, R \in \mathbb{R}$ (with $0 < \varepsilon < R$), a monotonic analytic function $\phi : (0,R) \longrightarrow \mathbb{R}$ (with $\phi(\varepsilon)\phi(R) < 0$ and ϕ convex on $(0,R)$), and a positive upper bound $\bar{\alpha}^*$ on $z\gamma(\phi,z)$ valid for all $z \in (0,R)$.

OUTPUT An ε-approximation of the unique root of ϕ in $(0,R)$, using no more than $O\left(\log\left(\alpha \log \frac{R}{\varepsilon}\right)\right)$ evaluations of ϕ and ϕ', and $O\left(\log\left(\alpha \log \frac{R}{\varepsilon}\right)\right)$ additional arithmetic operations.

DESCRIPTION

Step 0 Define c_0 to be $1 + \frac{1}{8\bar{\alpha}^*}$ or $\frac{1}{1 - \frac{1}{8\bar{\alpha}^*}}$, according as ϕ is decreasing or increasing on $(0,R)$. Then define $c_i := c_{i-1}^2$ for all $i > 0$, and compute $c_1, \ldots, c_M$ where M is the first positive integer with $c_M = c_0^{2^M} \geq \frac{R}{\varepsilon}$. Finally, set $\hat{x} := \varepsilon$ and $\hat{k} := M$.

Step 1 If $[(\phi$ is decreasing on $(0,R)$ and $\phi(c_{\hat{k}-1}\hat{x}) > 0)$ or $(\phi$ is increasing on $(0,R)$ and $\phi(c_{\hat{k}-1}\hat{x}) < 0)]$ then set $\hat{x} := c_{\hat{k}-1}\hat{x}$ and $\hat{k} := \hat{k} - 1$ and GOTO Step 1.

Step 2 Otherwise, if $k > 0$ then set $\hat{k} := \hat{k} - 1$ and GOTO Step 1.

Step 3 Perform $\log_2\left(3 + \log_2 \frac{R}{\varepsilon}\right)$ iterations of Newton's method (with $\hat{x}$ as the starting point), then OUTPUT the very last iterate.

Remark 7. *A well-known trick we will use frequently and implicitly is the computation of x^d, where $x \in \mathbb{R}$ and $d \in \mathbb{Z}$, within $2\lceil \log_2(|d|+1) \rceil$ multiplications using $\leq \lceil \log_2(|d|+1) \rceil$ intermediate real numbers. Briefly, assuming the binary expansion of d is $(a_k \cdots a_0)_2$ (so a_0 is the 1's bit), we simply compute x^{2^i}, via recursive squaring, for all i with a_i nonzero, and then multiply the resulting numbers together. For instance,* $363 = 2^8 + 2^6 + 2^5 + 2^3 + 2 + 1 = (101101011)_2$ *and thus* $x^{363} = (\cdots(\underbrace{x^2)\cdots)^2}_{8} \cdot (\cdots(\underbrace{x^2)\cdots)^2}_{6} \cdot (\cdots(\underbrace{x^2)\cdots)^2}_{5} \cdot ((x^2)^2)^2 \cdot x^2 \cdot x$, *which only requires* 17 *multiplications and enough memory for* 9 *intermediate real numbers — much faster than the naive* 362 *multiplications.* $\diamond$

SUBROUTINE FASTERCOUNT

INPUT A polynomial $p \in \mathbb{R}[x_1]$ with exactly $m \geq 3$ monomial terms, and $a, b \in \mathbb{R}$ with $a < b$.

OUTPUT The number of roots of p in the open interval (a,b).

DESCRIPTION

Step 0 If $m > 3$ then use Sturm-Habicht sequences (as in, say, [LM01]) to count the number of roots in (a,b) and STOP.

Step 1 Otherwise, let $p_0 := p$, $p_1 := p'$, let $-p_2$ be the remainder of p_0/p_1, and set $i := 2$.

Step 2 If $p_i \in \mathbb{R}$ then set $K := i$ and GOTO Step 5.

Step 3 Write $p_i(x) := u_1 x^{a_1} - u_0 x^{a_0}$, $p_{i-1}(x) := v_1 x^{b_1} - v_0 x^{b_0}$ (where $a_1 > a_0$ and $b_1 > b_0$), and let $c_j := \left\lceil \frac{b_j - a_1}{a_j - a_0} \right\rceil$ for $j \in \{0,1\}$.

Step 4 Replace i by $i+1$, define $p_i(x) := -v_1(\frac{u_1}{u_0})^{c_1} x^{b_1 - c_1(a_1 - a_0)} + v_0(\frac{u_1}{u_0})^{c_0} x^{b_0 - c_0(a_1 - a_0)}$, and GOTO Step 2.

Step 5 Using recursive squaring (cf. Remark 7), evaluate and RETURN $N_A - N_B$ (cf. Definition 3) where A (resp. B) is $(p_0(a), \ldots, p_K(a))$ (resp. $(p_0(b), \ldots, p_K(b))$).

4. Correctness and Complexity: Proving Theorems 1, 6, and 3

Proof of Theorem 1:
For open intervals, the second assertion follows immediately from Lemma 1 and Theorem 3. The case of closed intervals follows almost identically, save for an additional evaluation of f at the end-points of the interval which increases the complexity bound by (a negligible) $O(\log D)$.

The first assertion follows immediately from Theorems 6 and 5, and the second assertion of Theorem 1 which we've just proved. ■

Proof of Theorem 6:
Henceforth, for convenience, we will say "time" in place of "arithmetic complexity." Clearly,

it suffices to show that MNOMIALSOLVE is correct and satisfies the stated complexity bound when we set $\bar{\alpha}^* = \bar{\alpha}(D,m)$. Toward this end, let us first observe that $L_1(f)$ has exactly $m-1$ monomial terms and $L_2(f)$ has $\leq m-1$ monomial terms. Descartes' Rule then implies that $L_1(f)$ and $L_2(f)$ each have no more than $m-2$ positive roots. Note also that f, f', and f'' can all be evaluated using just $O(m \log D)$ arithmetic operations, thanks to Remark 7. These observations will be used implicitly throughout our proof.

The correctness of MNOMIALSOLVE is then straightforward for $m \leq 2$. Also, for $m=2$ and $D \geq 3$, the complexity of MNOMIALSOLVE is clearly the same (asymptotically) as that of subroutine HYBRID with $\bar{\alpha}^* := \frac{D-1}{2}$, thanks to Proposition 4. So we can assume $m \geq 3$. Furthermore, by Step 4 of MNOMIALSOLVE, we can clearly assume that $a_1 = 0$.

In the special case where $D=2$ and $m=3$, note that Step 6 is but an implementation of completing the square. So, provided one shifts one's ε-approximations by $\frac{c_2}{2c_3}$ while running the remainder of MNOMIALSOLVE, we can clearly attain the stated complexity bound for $D=2$. So we can assume additionally that $D \geq 3$ and now focus on Steps 10–18.

In the special case $a_2 = 1$, note that the change of variables $x \mapsto 1/x$ maps the interval $(r-\delta, r+\delta)$ to $\frac{1}{r^2-\delta^2}(r-\delta, r+\delta)$ for $r > \delta > 0$. An elementary calculation then reveals that $\delta \leq \max\{1, \varepsilon^3/2\} \Longrightarrow$ for any δ-approximation y of $x \geq \varepsilon$, $1/y$ is an ε-approximation of $1/x$. Note also that $x^{a_3} f(1/x) = c_3 + c_2 x^{a_3 - a_2} + c_1 x^{a_3}$ (since $a_1 = 0$ after Step 4), and $a_3 - a_2 > 1$ since $a_2 = 1$ and $a_3 = D \geq 3$. So, provided one takes a reciprocal after running MNOMIALSOLVE, we see that the worst-case time complexity of any instance with $a_2 = 1$ is asymptotically no worse than that of the instances with $a_2 > 1$. So we can assume additionally that $a_2 > 1$.

Now, by assumption, $S(f)$ is dampened, as are all the inputs going into recursive calls of MNOMIALSOLVE. Note also that by our dampening assumption (and Rolle's Theorem), the intervals from Steps 17(b) and 17(d) indeed contain exactly one root of f''. So our algorithm is well-defined and indeed finds ε-approximations to all the roots of f in $[0,R]$, assuming subroutines FASTERCOUNT and HYBRID are correct. The correctness of the latter two subroutines is proved below, so let us now concentrate on proving our main complexity bound.

So let $\mathcal{C}(m)$ denote the time needed for MNOMIALSOLVE to execute completely for an input consisting of an $f \in \mathcal{F}(D,m)$ (satisfying the dampening assumptions of our current theorem), an interval length of $\leq R$, and a precision ε. Clearly, assuming subroutine HYBRID runs in time $O\left(m \log(D) \log\left(\bar{\alpha}^* \log \frac{R}{\varepsilon}\right)\right)$ (which is covered in a separate proof below), we then have that $\mathcal{C}(m)$ must satisfy the following recurrence relation:

$$\mathcal{C}(m) \leq 2\mathcal{C}(m-1) + O\left(m(m \log D) \log\left(\bar{\alpha}^* \log \frac{R}{\varepsilon}\right)\right) + (2m-4)\mathcal{K}(D,m),$$

where the first term corresponds to finding the ε-approximations of the roots of f' and f'' (Steps 10, 13, and 14), the last term corresponds to the application of algorithm FASTERCOUNT to the intervals about these roots (Steps 13–16), and the O term corresponds to the application of algorithm HYBRID to the intervals between these roots (Steps 17–18). (Note that we have implicitly used the fact that $\mathcal{K}(D,m)$ and $\alpha(D,m)$ are non-decreasing functions of D and m.) In particular, for $m \geq 4$, the last term of our recurrence is justified by Remark 6.

Noting that we can regard R and ε as constants, it is then clear that to find the asymptotics of our recurrence, it suffices to find the asymptotics of the recurrence

$$c_m \leq 2c_{m-1} + m^2 A + (2m-4)B(m).$$

A simple `Maple` calculation (`rsolve(c(m)=2*c(m-1)+a*m*(m-1)+b(m)*(2*m-4),c(m));`) then shows that any such c_m must satisfy $c_m = O(2^m(c_2 + A) + \sum_{j=1}^{m} 2^{m-j} jB(j))$. So, knowing that $\mathcal{C}(2) = O\left(\log(D) \log\log \frac{R}{\varepsilon}\right)$ [Ye94, Pg. 279], we must then have that

$$\mathcal{C}(m) = O\left(2^m \left\{ \log(D) \log\left(\bar{\alpha}^* \log \frac{R}{\varepsilon}\right) + m^2 \mathcal{K}(D,m) \right\}\right). \blacksquare$$

Proof of Theorem 3:
The complexity bound follows immediately from the upper bound on K, assuming algorithm `FASTERCOUNT` is correct. The bound on K and the correctness of algorithm `FASTERCOUNT` are proved below. ■

At this point, we need only show that the subroutines `HYBRID` and `FASTERCOUNT` are correct, and that `FASTERCOUNT` satisfies a sufficiently good complexity bound.

Proof of Correctness and Complexity Analysis of `HYBRID`:
First note that by assumption, there is a unique $j \in \{0, \ldots, 2^{K-1}\}$ such that $(\beta^j \varepsilon, \beta^{j+1}\varepsilon]$ contains the sole root of ϕ in $(0, R)$. It then becomes clear that Steps 0–2 are merely an implementation of bisection that finds this j: In particular, Steps 0–2 essentially find the binary expansion of j (from most significant bit to least significant bit), and $\hat{x} = \beta^j$ once Step 3 is reached. By Theorem 4, $\hat{x}$ is then an approximate root of ϕ, and Step 3 indeed finds an ε-approximation of our root.

As for the complexity of our algorithm, it is clear that $M = \left\lceil \log_2 \left(\log_2 \frac{R}{\varepsilon}\right) - \log_2 \log_2 \beta \right\rceil$ and we thus need only $O\left(\log\left(\log \frac{R}{\varepsilon}\right) - \log\log\beta\right)$ evaluations of ϕ (and inequality checks) until Step 3. Step 3 then clearly takes just $O\left(\log\log \frac{R}{\varepsilon}\right)$ arithmetic operations and evaluations of ϕ and ϕ'.

To conclude, recall the elementary inequalities $\frac{\pm 1}{\log(1 \pm x)} \leq x$, valid for all $x \geq 0$. Observe then that $-\log\log\beta = \log \frac{1}{\log\beta} \leq \log \bar{\alpha}^*$. Since ϕ an m-nomial of degree D implies that ϕ and ϕ' can be evaluated within $O(m \log D)$ arithmetic operations (cf. Remark 7), we are done. ■

Proof of Correctness and Complexity Analysis of `FASTERCOUNT`:
The correctness of `FASTERCOUNT` when $m > 3$ is clear from any of the standard references [Roy96, LM01] so there is nothing to prove.

For $m = 3$, correctness follows easily from Lemma 1 upon noting a simple fact: The remainder of $\frac{v_1 x^{b_1} - v_0 x^{b_0}}{u_1 x^{a_1} - u_0 x^{a_0}}$ is nothing more than the reduction of $v_1 x^{b_1} - v_0 x^{b_0}$ modulo $u_1 x^{a_1} - u_0 x^{a_0}$. In particular, Step 2 of `FASTERCOUNT` is just a compact representation of a sufficient number of applications of the identity $x^{a_1} = \frac{u_0}{u_1} x^{a_0}$ to reduce $-(v_1 x^{b_1} - v_0 x^{b_0})$ modulo $u_1 x^{a_1} - u_0 x^{a_0}$ to a polynomial of degree $< a_1$. So correctness follows immediately.

As for the complexity bound, we need only observe that (by recursive squaring) every execution of Step 2 and Step 3 takes only $O(\log D)$ arithmetic operations, and that $K = O(\log D)$ at the termination of the algorithm, i.e., there are only $O(\log D)$ remainders in our Sturm sequence. The first assertion is clear, so to prove the latter assertion we will need to prove a technical bound on the exponents which occur in our remainder sequence.

In particular, let ℓ_i be the absolute value of the difference exponents of p_i (we set $\ell_i := 0$ if p_i is monomial). Note that p_i a monomial $\Longrightarrow p_{i+1}$ is a monomial, and thus p_{i+2} is constant. Note also that, via the definition of long division, we have $\ell_{i+2} \leq |\ell_{i+1} - \ell_i|$, with equality

occurring iff p_{i+2} is a binomial. So if we can show

$$(\star) \qquad \min\{\ell_{i+2}, \ell_{i+3}\} \leq \ell_i/2$$

for all i, then we will easily obtain our bound on K (since $\ell_0, \ell_1 \leq D$ and all the ℓ_i are integers).

To prove $(\star)$ observe that $\ell_{i+1} \leq \frac{1}{2}\ell_i \implies \ell_{i+2} \geq \frac{1}{2}\ell_{i+1}$ (if $\ell_{i+2} \neq 0$) and thus $\ell_{i+3} \leq \frac{1}{2}\ell_i$. Similarly, $\ell_{i+1} \geq \frac{1}{2}\ell_i \implies \ell_{i+2} \leq \frac{1}{2}\ell_i$. So $K \leq 3\lceil \log_2 D \rceil + 2 = O(\log D)$ and we are done. ■

5. The Proof of Theorem 2

Dividing by a suitable monomial term, we can clearly assume without loss of generality that f has a constant term . Note then that C is diffeomorphic to a line iff f is not the square of binomial, via [LRW03, Prop. 2 and Lemma 1]. So there are actually no isolated singularities for trinomial curves.

It is then clear that x is a vertical tangent of $C \implies x$ is a root of $F_v := (f, \frac{\partial f}{\partial x_2})$. The roots of F_v with x or y coordinate equal to 0 or R can be found easily by suitably fixing one of the variables and applying Theorem 1, so let us concentrate on the roots of F_v in $(0, R)^2$.

By [Roj00a, Lemma 4.2] and [vdK00] we can then find (for any constant $\delta > 0$) a monomial change of variables, using just $O(\log^{2+\delta} D)$ bit operations, such that the resulting sparse polynomial system $G := (g_1, g_2)$ has $g_1 \in \mathbb{R}[x_1, x_2]$ a trinomial, $g_2 \in \mathbb{R}[x_1]$ a binomial, and all underlying exponents have bit-length $O(\log^2 D)$. We can then solve g_2 to accuracy $\min\left\{\varepsilon^{O(D^2)}, 1\right\}$ via Theorem 1, back-substitute the roots into g_1, solve the univariate specialized g_1 to accuracy $\min\left\{\varepsilon^{O(D^2)}, 1\right\}$, then invert our monomial map via [Roj00a, Main Theorem 1.3] and [vdK00] to recover the roots of F_v, using just $O\left(\log(D) \log\left(D \log \frac{R}{\varepsilon}\right)\right)$ additional arithmetic operations and $O(\log^{2+\delta} D)$ additional bit operations. So we can indeed find isolated vertical tangents within the stated complexity bound.

To conclude, note that [LRW03, Lemma 9] tells us that if x is an isolated inflection point of C then $[\partial_1^2 f \cdot (\partial_2 f)^2 - 2\partial_1 \partial_2 f \cdot \partial_1 f \cdot \partial_2 f + \partial_2^2 f \cdot (\partial_1 f)^2] = 0$, where $\partial_i := \frac{\partial}{\partial x_i}$. In the special case of a trinomial, say $f(x_1, x_2) := 1 + Ax_1^a x_2^b + Bx_1^c x_2^d$, the preceding polynomial in derivatives is exactly:

$$-ab\,(a+b)\,R^3 + \left(b^2c^2 + a^2d^2 - b^2c - 2\,abd - a^2d - 2\,abc - 2\,abcd\right) R^2 S$$

$$+ \left(b^2c^2 - 2\,abcd - ad^2 - 2\,acd - 2\,bcd - bc^2 + a^2d^2\right) RS^2 - cd\,(c+d)\,S^3,$$

where $R := Ax_1^a x_2^b$ and $S := Bx_1^c x_2^d$, thanks to a `Maple` calculation. (Note that we must also have $1 + R + S = 0$ since we're assuming $f = 0$.) So we can efficiently solve for x as follows: (1) solve the above cubic for $\frac{R}{S}$ up to accuracy $\min\left\{\varepsilon^{O(D)}, 1\right\}$ via the main algorithm of [NR96] or [BP94], (2) solve for (R, S) via the additional relation $1 + R + S = 0$, (3) solve the resulting binomial systems in (R, S, x_1, x_2) for x via [Roj00a, Main Theorem 1.3] and [vdK00]. So we can ε-approximate the inflection points of C within the stated asymptotic complexity bound as well, and we are done. ■

6. The Proof of Theorem 5

Definition 8. *For any $a \in \mathbb{R}$ and $k \in \mathbb{N} \cup \{0\}$, let $(a)_k := a(a-1)\cdots(a-k+1)$. (So $(a)_0 = 1$, $(a)_1 = a$, $a \in \mathbb{N} \implies (a)_a = a!$, and $a, k \in \mathbb{N}$ with $k > a \implies (a)_k = 0$.)* ⋄

Proof of Theorem 5:
The first bound is immediate since $\mathcal{F}(D,m)\subseteq\mathcal{F}(D,m+1)$.

The formula for $\bar{\alpha}(D,2)$ follows easily since $m=2$ and $f(0)\neq 0$ implies that f' has no positive roots. This in turn implies that $N(f)=1$, so the only interval we (may) need to majorize over is $(u_0(f),u_1(f))=(0,\infty)$. In particular, the quantity we majorize is just $x\gamma(f,x)$, and the formula for $\bar{\alpha}(D,2)$ then follows immediately from Proposition 4. So we can assume $m=3$. Since $\gamma(f,x)=\gamma(cf,x)$ for any nonzero constant c, we can also clearly assume that $f(x)=x^{a_3}-Ax^{a_2}+B$ where A and B are nonzero real constants and $a_3=D>a_2\geq 1$. Since we assume $f'(0)=0$ in the definition of $\bar{\alpha}(D,m)$, we must also have $D\geq 3$ and $a_2>1$.

Next, note that $A<0\implies f'$ has no positive roots. So, similar to the $m=2$ case, one either majorizes over the empty set or $(0,\infty)$. In the former case there is no contribution to $\bar{\alpha}(D,m)$, while in the second case, we see that we are again majorizing $x\gamma(f,x)$ over $(0,\infty)$. Since the latter supremum is no greater than $\frac{D-1}{2}$ by Propositions 3 and 4, we can can now assume additionally that $A>0$.

Now note that

$$\begin{aligned} f'(x) &= a_3x^{a_3-1}-a_2Ax^{a_2-1} \\ &= a_3x^{a_2-1}\left(x^{a_3-a_2}-\frac{a_2A}{a_3}\right), \end{aligned}$$

and

$$\begin{aligned} f''(x) &= a_3(a_3-1)x^{a_3-2}-a_2A(a_2-1)x^{a_2-2} \\ &= a_3(a_3-1)x^{a_2-2}\left(x^{a_3-a_2}-\frac{a_2A(a_2-1)}{a_3(a_3-1)}\right). \end{aligned}$$

Clearly then, $x_1:=\left(\frac{a_2A}{a_3}\right)^{1/(a_3-a_2)}$ is the unique positive root of f' and $x_2:=\left(\frac{a_2(a_2-1)A}{a_3(a_3-1)}\right)^{1/(a_3-a_2)}$ is the unique positive root of f''. Furthermore, since $1<a_2<a_3$, it is clear that $0<x_2<x_1$. In particular, we see that we are left with a collection of 3 intervals to consider in the definition of $\bar{\alpha}(D,m)$:

$$\{(0,x_2),(x_2,x_1),(x_1,\infty)\}$$

and the quantities we majorize on these intervals are, respectively:

$$\{x\gamma(f,x),(x_1-x)\gamma(f,x),(x-x_1)\gamma(f,x)\}.$$

The intervals (x_2,x_1) and (x_1,∞) can be handled via a unified calculation, so let us complete our analysis by examining the intervals (x_2,∞) and $(0,x_2)$.

The Interval (x_2,∞):
Note that $f(y+x_1)=(y+x_1)^{a_3}-A(y+x_1)^{a_2}+B$ and thus

$$f^{(k)}(y+x_1) = (a_3)_k(y+x_1)^{a_3-k}-(a_2)_kA(y+x_1)^{a_2-k}.$$

So, letting $\phi(y) := f(y+x_1)$, we have for all $k \geq 2$:

$$\left|\frac{\phi^{(k)}(y)}{k!\phi'(y)}\right|^{1/(k-1)} = \left|\frac{(a_3)_k(y+x_1)^{a_3-k} - (a_2)_k A(y+x_1)^{a_2-k}}{k!(a_3(y+x_1)^{a_3-1} - a_2A(y+x_1)^{a_2-1})}\right|^{1/(k-1)} \tag{1}$$

$$= \left|\frac{(a_3)_k(y+x_1)^{a_3-a_2} - (a_2)_k A}{k!a_3(y+x_1)^{k-1}\left((y+x_1)^{a_3-a_2} - \frac{a_2A}{a_3}\right)}\right|^{1/(k-1)} \tag{2}$$

$$= \frac{1}{|y|}\left|\frac{(a_3)_k}{k!a_3}\right|^{1/(k-1)}\left|\frac{y^{k-1}\left((y+x_1)^{a_3-a_2} - \frac{(a_2)_k A}{(a_3)_k}\right)}{(y+x_1)^{k-1}\left((y+x_1)^{a_3-a_2} - x_1^{a_3-a_2}\right)}\right|^{1/(k-1)}, \tag{3}$$

which in turn is bounded above by

$$\frac{1}{|y|}\cdot\frac{a_3-1}{2}\left|\frac{y^{k-1}}{(y+x_1)^{k-1}}\cdot\frac{(y+x_1)^{a_3-a_2} - \frac{(a_2)_k A}{(a_3)_k}}{(y+x_1)^{a_3-a_2} - x_1^{a_3-a_2}}\right|^{1/(k-1)} \tag{4}$$

since $\left|\frac{(a_3)_k}{k!a_3}\right|^{1/(k-1)} \leq \frac{a_3-1}{2}$.

Now let $z := y/x_1$ and let ρ be $\left(\frac{(a_2)_k A}{(a_3)_k}\right)^{1/(a_3-a_2)} \Big/ x_1$ or 0 according as $k \leq a_2$ or not. Then for $y \neq -x_1$ (or $z \neq -1$, equivalently) we have:

$$\left|\frac{y^{k-1}}{(y+x_1)^{k-1}}\cdot\frac{(y+x_1)^{a_3-a_2} - \frac{(a_2)_k A}{(a_3)_k}}{(y+x_1)^{a_3-a_2} - x_1^{a_3-a_2}}\right|^{1/(k-1)} = \left|\frac{z^{k-1}}{(z+1)^{k-1}}\cdot\frac{(z+1)^{a_3-a_2} - \rho^{a_3-a_2}}{(z+1)^{a_3-a_2} - 1}\right|^{1/(k-1)} \tag{5}$$

$$= \left|\frac{z^{k-2}}{(z+1)^{k-1}}\cdot\frac{(z+1)^{a_3-a_2} - \rho^{a_3-a_2}}{1 + (z+1) + \cdots + (z+1)^{a_3-a_2-1}}\right|^{1/(k-1)} \tag{6}$$

$$= \left|\frac{z}{z+1}\right|^{(k-2)/(k-1)}\left|\frac{(z+1)^{a_3-a_2} - \rho^{a_3-a_2}}{(z+1) + (z+1)^2 \cdots + (z+1)^{a_3-a_2}}\right|^{1/(k-1)}, \tag{7}$$

where the last equality follows from the identity $\frac{z}{(z+1)^N-1} = \frac{z+1-1}{(z+1)^N-1} = \frac{1}{1+(z+1)+\cdots+(z+1)^{N-1}}$.

Since $\frac{z}{z+1} = 1 - \frac{1}{z+1}$ is clearly an increasing function of z for all $z > -1$ (or $y > -x_1$, equivalently), and since $0 > -\frac{x_2}{x_1} > -1$, we must have for all $y \geq x_2 - x_1$ (or $z \geq \frac{x_2}{x_1} - 1$, equivalently) that

$$\begin{aligned}\left|\frac{z}{z+1}\right|^{(k-2)/(k-1)} &\leq \max\left\{\left(\frac{x_1}{x_2} - 1\right)^{(k-1)/(k-2)}, 1\right\} \\ &= \max\left\{\left(\left(\frac{a_3-1}{a_2-1}\right)^{1/(a_3-a_2)} - 1\right)^{(k-1)/(k-2)}, 1\right\} \\ (\star) \qquad &\leq \max\left\{\frac{a_3-1}{a_2-1} - 1, 1\right\}.\end{aligned}$$

Furthermore, since $\frac{(a_2-2)_k}{(a_3-2)_k} \leq 1$ for $2 \leq k \leq a_2 - 2$, we must have $\frac{x_2}{x_1} \geq \rho$ and thus the last factor of Equality (7) is bounded above by 1. So by Equalities (1)–(7) and $(\star)$ we obtain that

$$|y|\left|\frac{\phi^{(k)}(y)}{k!\phi'(y)}\right|^{1/(k-1)} \leq \frac{1}{2}\max\{(a_3-1)(a_3-2), a_3-1\} \leq (D-1)(D-2)/2,$$

since $D \geq 3$. Thus, the contribution of the interval (x_2, ∞) to $\bar{\alpha}(D,m)$ is no more than $(D-1)(D-2)/2$. ■

The Interval $(0, x_2)$: Similar to the last case, we have for all $x \in (0, x_2)$ that

$$\begin{aligned}
\left|\frac{f^{(k)}(x)}{k!f'(x)}\right|^{1/(k-1)} &= \left|\frac{(a_3)_k x^{a_3-k} - (a_2)_k A x^{a_2-k}}{k!(a_3 x^{a_3-1} - a_2 A x^{a_2-1})}\right|^{1/(k-1)} \\
&= \left|\frac{(a_3)_k x^{a_3-a_2} - (a_2)_k A}{k! a_3 x^{k-1}\left(x^{a_3-a_2} - \frac{a_2 A}{a_3}\right)}\right|^{1/(k-1)} \\
&= \frac{1}{x}\left|\frac{(a_3)_k}{k! a_3}\right|^{1/(k-1)} \left|\frac{x^{a_3-a_2} - \frac{(a_2)_k A}{(a_3)_k}}{x^{a_3-a_2} - x_1^{a_3-a_2}}\right|^{1/(k-1)} \\
&\leq \frac{1}{x}\cdot\frac{a_3-1}{2}\left|\frac{x^{a_3-a_2} - \frac{(a_2)_k A}{(a_3)_k}}{x^{a_3-a_2} - x_1^{a_3-a_2}}\right|^{1/(k-1)}.
\end{aligned}$$

So, for $0 < z < x_2/x_1 = (\frac{a_2-1}{a_3-1})^{1/(a_3-a_2)} < 1$, we have

$$\begin{aligned}
\left|\frac{x^{a_3-a_2} - \frac{(a_2 A)_k}{(a_3)_k}}{x^{a_3-a_2} - x_1^{a_3-a_2}}\right|^{1/(k-1)} &= \left|\frac{z^{a_3-a_2} - \rho^{a_3-a_2}}{z^{a_3-a_2} - 1}\right|^{1/(k-1)} \\
&= \left|1 - \frac{1-\rho^{a_3-a_2}}{1-z^{a_3-a_2}}\right|^{1/(k-1)} \\
&\leq \max\left\{1,\ \left|\frac{1-\rho^{a_3-a_2}}{1-z^{a_3-a_2}}\right|^{1/(k-1)}\right\} \quad \text{(recalling that } \rho<1\text{)} \\
&\leq \max\left\{1,\ \left|\frac{1}{1-z^{a_3-a_2}}\right|^{1/(k-1)}\right\} \quad \text{(recalling that } \rho<1\text{)} \\
&\leq \left|\frac{a_2-1}{a_3-a_2}\right|^{1/(k-1)} \\
&\leq a_2 - 1 \\
&\leq a_3 - 2.
\end{aligned}$$

Thus

$$\left|\frac{f^{(k)}(x)}{k!f'(x)}\right|^{1/(k-1)} \leq \frac{1}{y}\cdot\frac{(a_3-1)(a_2-1)}{2},$$

and the quantity we need to majorize on this interval is no more than $(D-1)(D-2)/2$. ■

Putting together the bounds we've found over our two preceding intervals, we see that our upper bound for $\bar{\alpha}(D, 3)$ holds and we are done. ■

7. Acknowledgements

The authors thank Felipe Cucker, John McDonald, Tom Michiels, and Steve Smale for some delightful conversations. We also thank two anonymous referees for useful suggestions.

References

[BP94] Bini, Dario and Pan, Victor Y., *Polynomial and Matrix Computations, Volume 1: Fundamental Algorithms,* Progress in Theoretical Computer Science, Birkhäuser, 1994.

[BSS89] Blum, Lenore; Shub, Mike; and Smale, Steve; *"On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines,"* Bull. Amer. Math. Soc. (N.S.) 21 (1989), no. 1, pp. 1–46.

[BCSS98] Blum, Lenore; Cucker, Felipe; Shub, Mike; and Smale, Steve; *Complexity and Real Computation,* Springer-Verlag, 1998.

[Bre76] Brent, Richard P., *"Multiple-Precision Zero-Finding Methods and the Complexity of Elementary Function Evaluation,"* Analytic Computational Complexity (Proc. Sympos., Carnegie-Mellon Univ., Pittsburgh, Pa., 1975), pp. 151–176, Academic Press, New York, 1976.

[BMST97] Bshouty, Nader H.; Mansour, Yishay; Schieber, Baruch; and Tiwari, Prasoon, *"A tight bound for approximating the square root,"* Inform. Process. Lett. 63 (1997), no. 4, pp. 211–213.

[GJ79] Garey, Michael R. and Johnson, David S., *Computers and Intractability: A Guide to the Theory of NP-Completeness,* A Series of Books in the Mathematical Sciences, W. H. Freeman and Co., San Francisco, Calif., 1979.

[GKZ94] Gel'fand, Israel Moseyevitch; Kapranov, Misha M.; and Zelevinsky, Andrei V.; *Discriminants, Resultants and Multidimensional Determinants,* Birkhäuser, Boston, 1994.

[GLS01] Giusti, Marc; Lecerf, Gregoire; and Salvy, Bruno, *"A Gröbner-Free Alternative to Polynomial System Solving,"* Journal of Complexity, vol. 17, no. 1, March 2001, pp. 154–211.

[Hab48] Habicht, Walter, *"Eine Verallgemeinerung des Sturmschen Wurzelzhlverfahrens,"* Comment. Math. Helv. **21** (1948), pp. 99–116.

[Her56] Hermite, Charles, *"Sur le nombres des racines d'une équation algébrique comprisé entre des limites donn'es,"* J. Reine Angew. Math. 52 (1856) 39–51; also: Oeuvres, Vol. I (Gauthier-Villars, Paris, 1905) pp. 397–414; English translation: P. C. Parks, Internat. J. Control 26 (1977), pp. 183–195.

[Kho91] Khovanskii, Askold, *Fewnomials,* AMS Press, Providence, Rhode Island, 1991.

[LRW03] Li, Tien-Yien; Rojas, J. Maurice; and Wang, Xiaoshen, *"Counting Real Connected Components of Trinomial Curve Intersections and m-nomial Hypersurfaces,"* Discrete and Computational Geometry, 30 (2003), no. 3, pp. 379–414.

[LM01] Lickteig, Thomas and Roy, Marie-Francoise, *"Sylvester-Habicht Sequences and Fast Cauchy Index Computation,"* J. Symbolic Computation (2001) **31**, pp. 315–341.

[MR04] Malajovich, Gregorio and Rojas, J. Maurice, *"High Probability Analysis of the Condition Number of Sparse Polynomial Systems,"* Theoretical Computer Science, special issue on algebraic and numerical algorithms, to appear.

[dMS96] de Melo, Wellington and Svaiter, Benar Fux, *"The Cost of Computing Integers,"* Proceedings of the American Mathematical Society, Vol. 124, No. 5, May 1996.

[Mor97] de A. Moreira, Carlos Gustavo T., *"On Asymptotic Estimates for Arithmetic Cost Functions,"* Proceedings of the American Mathematical Society, Vol. 125, No. 2, February 1997, pp. 347–353.

[MP99] Mourrain, Bernard and Pan, Victor Y. *"Asymptotic Acceleration of Solving Multivariate Polynomial Systems of Equations,"* Proc. ACM STOC 1998 (Dallas, TX), pp. 488–496, ACM, New York, 1999.

[MPR03] Mourrain, Bernard; Pan, Victor Y.; and Ruatta, Olivier, *"Accelerated solution of multivariate polynomial systems of equations,"* SIAM J. Comput. 32 (2003), no. 2, pp. 435–454

[NR96] Neff, C. Andrew and Reif, John, *"An Efficient Algorithm for the Complex Roots Problem,"* Journal of Complexity **12** (1996), no. 2, pp. 81–115.

[Pap95] Papadimitriou, Christos H., *Computational Complexity,* Addison-Wesley, 1995.

[PRS93] Pedersen, P., Roy, M.-F., Szpirglas, A., *"Counting Real Zeroes in the Multivariate Case,"* Computational Algebraic Geometry (Nice, 1992), pp. 203–224, Progr. Math. **109**, Birkhäuser Boston, Boston, MA, 1993.

[Ren87] Renegar, James, *"On the Worst-Case Arithmetic Complexity of Approximating Zeros of Polynomials,"* J. Complexity 3 (1987), no. 2, pp. 90–113.

[Ren89] _____, *"On the Worst Case Arithmetic Complexity of Approximating Zeros of Systems of Polynomials,"* SIAM J. Comput. **18** (1989), no. 2, pp. 350–370.

[Ric93] Richardson, Daniel, *"Finding the Number of Distinct Real Roots of Sparse Polynomials of the Form $p(x, x^n)$,"* Computational Algebraic Geometry (Nice, 1992), pp. 225–233, Progr. Math., 109, Birkhäuser Boston, Boston, MA, 1993.

[Roj98] _____, *"Intrinsic Near Quadratic Complexity Bounds for Real Multivariate Root Counting,"* Proceedings of the Sixth Annual European Symposium on Algorithms, Lecture Notes in Computer Science 1461, Springer-Verlag (1998).

[Roj99b] _____, *"Solving Degenerate Sparse Polynomial Systems Faster,"* Journal of Symbolic Computation, special issue on elimination theory, vol. 27, 1999.

[Roj00a] _____, *"Some Speed-Ups and Speed Limits for Real Algebraic Geometry,"* Journal of Complexity, FoCM 1999 special issue, vol. 16, no. 3 (sept. 2000), pp. 552–571.

[Roj04a] _____, *"Dedekind Zeta Functions and the Complexity of Hilbert's Nullstellensatz,"* submitted for publication, also available at `http://xxx.arxiv.org/math.NT/030111` .

[RS04] Rojas, J. Maurice Rojas and Stella, Casey, *"A Complexity Threshold for Real Fewnomials,"* preprint.

[Roy96] Roy, Marie-Françoise, *"Basic Algorithms in Real Algebraic Geometry and their Complexity: from Sturm's Theorem to the Existential Theory of Reals,"* Lectures in Real Geometry (Madrid, 1994), pp. 1–67, de Gruyter Exp. Math., 23, de Gruyter, Berlin, 1996.

[SS94] Shub, Mike and Smale, Steve, *Complexity of Bezout's theorem V: Polynomial time,* Theoret. Comput. Sci. **133** (1994), no. 1, pp. 141–164, Selected papers of the Workshop on Continuous Algorithms and Complexity (Barcelona, 1993).

[Sma86] Smale, Steve, *"Newton's Method Estimates from Data at One Point,"* The Merging of Disciplines: New Directions in Pure, Applied, and Computational Mathematics (Laramie, Wyo., 1985), pp. 185–196, Springer, New York, 1986.

[Sma98] _____, *"Mathematical Problems for the Next Century,"* Math. Intelligencer 20 (1998), no. 2, pp. 7–15.

[Sma00] _____, *"Mathematical Problems for the Next Century,"* Mathematics: Frontiers and Perspectives, pp. 271–294, Amer. Math. Soc., Providence, RI, 2000.

[SL54] Smith, David Eugene and Latham, Marcia L., *The Geometry of René Descartes,* translated from the French and Latin (with a facsimile of Descartes' 1637 French edition), Dover Publications Inc., New York (1954).

[Stu35] Sturm, C., *"Mémoire sur la résolution des équations numériques,"* Inst. France Sc. Math. Phys., **6** (1835).

[vdK00] van der Kallen, Wilberd, *"Complexity of the Havas, Majewski, Matthews LLL Hermite normal form algorithm,"* J. Symbolic Comput. 30 (2000), no. 3, pp. 329–337.

[Ye94] Ye, Yinyu, *"Combining Binary Search and Newton's Method to Compute Real Roots for a Class of Real Functions,"* J. Complexity 10 (1994), no. 3, 271–280.

Department of Mathematics, Texas A&M University, TAMU 3368, College Station, TX 77843, USA

E-mail address: `rojas@math.tamu.edu`, *Web-Page:* `http://www.math.tamu.edu/~rojas`

Department of Management Science and Engineering, School of Engineering, Stanford University, Stanford, CA 94305, USA

E-mail address: `yinyu-ye@stanford.edu`, *Web-Page:* `http://www.stanford.edu/~yyye`