ON SOLVING FEWNOMIALS OVER INTERVALS IN FEWNOMIAL TIME

J. MAURICE ROJAS⋆ AND YINYU YE†

Abstract. Let f be a degree D univariate polynomial with real coefficients and at most 3 monomial terms. We show that all the roots of f in any closed interval of length R can be approximated within an accuracy of ε using just $\mathcal{O}(\log D \log\log \frac{R}{\varepsilon} + \log^2 D)$ arithmetic steps, i.e., the arithmetic complexity is polylogarithmic in the degree of the underlying complex variety. In particular, of independent interest is an algebraic sub-routine of our algorithm which counts the roots in any given interval using just $\mathcal{O}(\log^2 D)$ arithmetic steps. The best previous arithmetic complexity upper bounds for these respective solving and counting problems were $\mathcal{O}(D \log^5 D \log\log \frac{R}{\varepsilon})$ and $\mathcal{O}(D \log^2 D \log\log D)$.

We also discuss conditions under which our algorithms can be extended to univariate m-nomials.

1. Introduction

Real-solving — the study of solving polynomial systems over the real numbers — occupies a curious position within computational algebraic geometry. From the point of view of computational complexity, classical algebraic geometry left real-solving almost completely untouched. For example, rigorous 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 [Ren89], and finding optimal bounds continues to be the subject of much active research [BCSS98, MP99, Roj99b, GLS01].

Focusing on real roots, there is a beautiful result that one can bound their number independently of the degrees of the underlying polynomials — there is an explicit bound singly exponential in the number of variables and the total number of monomial terms [Kho91]. While the optimality of this "fewnomial" bound is still unknown in the multivariate case, it is significantly smaller than the number of complex roots when the underlying polynomials have sufficiently high degree. This naturally leads one to suspect if a similar phenomenon occurs for the harder problem of approximating the real roots, i.e., can one solve sparse polynomial systems over the real numbers significantly faster than via the usual algorithms based on algebraic geometry over $\mathbb{C}$? 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**[1] of degree D could be approximated within a number of arithmetic steps sub-linear in D [PRS93, Roj98, MP99, Roj99b].

Date: July 2, 2001.

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

† 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.

[1] i.e., a polynomial with exactly 3 monomial terms. More generally, we call a polynomial with exactly m monomial terms an ***m*-nomial**, and when $m=2$ we sometimes say **binomial**.

We now answer this last question affirmatively as follows: Suppose $f \in \mathbb{R}[x_1]$ has degree D and real roots $\zeta_1, \ldots, \zeta_\mu$. By **$\varepsilon$-approximating the roots of[2] f** we will mean finding real numbers $z_1, \ldots, z_m$ such that for any $i \in \{1, \ldots, \mu\}$, $|\zeta_i - z_j| < \varepsilon$ for some $j \in \{1, \ldots, m\}$. (So $m < \mu$ and $m > \mu$ are both possible.)

Theorem 1. *Let $a \in \mathbb{R}$, $R, \varepsilon > 0$, and suppose $f \in \mathbb{R}[x_1]$ has degree D and at most 3 monomial terms. Then we can ε-approximate all the roots of f in the closed interval $I := [a - R, a + R]$ within $\mathcal{O}\left(\log D \log\log \frac{R}{\varepsilon} + \log^2 D\right)$ arithmetic operations.[3] Furthermore, we can count the number of roots of f in I within $\mathcal{O}(\log^2 D)$ arithmetic operations, and we can thus construct our approximations so that they are in bijective (and order-preserving) correspondence with the roots of f in I.*

The main obstructions to extending our algorithm to sparse polynomials with 4 or more terms are (a) efficiently counting the number of roots in an interval and (b) bounding an analytic invariant associated to the accelerated convergence of Newton's method.[4] Precise statements of what can and remains to be done appear in theorems 3 and 4 of section 1.2 below.

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 and D is an upper bound on all the exponents appearing in f. Then we can ε-approximate, within $\mathcal{O}((\log D)(\log^2 D + \log\log \frac{R}{\varepsilon}))$ arithmetic operations and $\mathcal{O}(\log^{6.376} D)$ bit operations, all isolated inflection points, vertical tangents, and singular points of the curve in the positive quadrant defined by $f = 0$.*

We point out that the best[5] current general algorithms for multivariate polynomial system solving, e.g., [MP99, Roj99b, GLS01], would result in an arithmetic complexity bound polynomial in $\log\log \frac{R}{\varepsilon}$ and D instead.

We leave the bit complexity of our algorithms for a future paper. In particular, our primary 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 an ordinary Turing machine augmented with registers that allow arithmetic with real numbers as well as bits.

1.1. **Earlier Work.**

Note that the current, theoretically fastest, general algorithms [BP94, NR96] for univariate polynomial solving over the complex numbers, applied to the real-solving problem in theorem 1, would yield a sequential arithmetic complexity upper bound of $\mathcal{O}(D \log^5 D \log\log \frac{R}{\varepsilon})$.

---

[2]Henceforth, this will be understood to be our definition of "solving."

[3]Here and throughout we count an arithmetic operation as a field operation in the field generated over $\mathbb{Q}$ by the coefficients of f.

[4]Curiously, (a) appears to be the only obstruction to extending our trinomial result to real exponents.

[5]In the sense of having (or obviously implying) the best current explicit complexity bound for ε-approximation, either with or without degeneracy assumptions.

In particular, for just counting the number of roots, it is known that **Sylvester-Habicht sequences** [Roy96, LM01] yield a sequential arithmetic complexity upper bound of $\mathcal{O}(D\log^2 D\log\log D)$ for counting roots in an interval.

"Super-fast" solving for the case of ≤ 1 monomial term is of course trivial, and the binomial case was known at least since the mid-70's, e.g., [Bre76] and [Ye94, sec. 4]. On the other hand, little seems to be known about the case of exactly 3 monomial terms: To the best of the author's 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 $c_0+c_1x^d+c_2x^D$ (with $0<d<D$) is polynomial in $\log D$ and d.

Little 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 $\mathcal{O}(m(\log D\log\log\frac{R}{\varepsilon}+\log^2 D))$ for ε-approximating the roots of

$$c_0+c_1x^{a_1}+\cdots+c_kx^{a_k}-c_{k+1}x^{a_{k+1}}-\cdots-c_mx^{a_m},$$

when $(a_1,\ldots,a_m)$ is a strictly increasing sequence of positive integers, and all the c_i are positive real numbers.

It is 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, then to solve $F:=(f_1,\ldots,f_n)$ one can simply solve f_1 first and then recursively solve the resulting smaller systems. This approach easily yields an arithmetic complexity bound of $\mathcal{O}\left(\mathcal{N}\log^2 D\log\log\frac{R}{\varepsilon}\right)$, where we now instead ε-approximate roots in the **orthant-wedge**

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

$\mathcal{N}$ ($\leq 2^n$ [LRW01, thm. 3, part (c)]) is the maximum number of isolated roots in W_R^n of such an F, and D denotes the maximum total degree of any f_i. However, extending these results to general trinomial systems, let alone general sparse systems, remains an open problem.

On the other hand, there is at least an algorithm, with arithmetic complexity logarithmic in the degree of the underlying complex variety, for approximating the roots of general **bi**nomial systems in W_R^n [Roj00a, Main Theorem 1.3].

1.2. α-theory, Sturm Sequences, Speed-Ups, and Obstructions Thereof.

Our algorithms are based on an earlier hybrid algorithm of the second author which combines bisection and Newton iteration, along with a new observation on the **Sturm sequences** of binomials (theorem 5 below). The latter refinement, along with examples showing how it can not be extended naively to general m-nomials are covered below as well.

The two key tools are an application of Smale's **$\boldsymbol{\alpha}$-theory** [Sma86] and an algebraic observation on trinomials. In particular, we will need the following fundamental results.

Definition 1. *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 $\lambda\in[0,1]$. Also, for any analytic function $f:U\longrightarrow\mathbb{R}$ and $z\in U$, let $z_0:=z$ and $z_i:=z_{i-1}-\frac{f(z_{i-1})}{f'(z_{i-1})}$ for all*

$i\in\mathbb{N}$. Then, if there is a root ζ 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, converges quadratically to a root of f), we call z an **approximate root** *of f with* **associated root** *ζ.* $\diamond$

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

Ye's Theorem. [Ye94, thm. 2] *Following the notation of definition 1, suppose further that f is convex. Also let r be a root of f in the open interval $(0,R)$ and suppose that there is an $\bar{\alpha}>0$ such that $\left|\frac{f^{(k)}(x)}{k!f'(x)}\right|^{1/(k-1)}\leq\frac{\bar{\alpha}}{x}$, for all $k>1$ and for all x in the closed interval $[\frac{r}{1+\frac{1}{8\bar{\alpha}}},\frac{r}{1-\frac{1}{8\bar{\alpha}}}]$. Then*

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

Remark 1. *The supremum in the theorem above is usually referred to as $\alpha(f,x)$ in the work of Smale and his co-authors. Hence our introduction of $\bar{\alpha}$.* $\diamond$

Proposition 1. [Ye94, prop. 1] *Suppose $f_1,f_2:U\longrightarrow\mathbb{R}$ satisfy the hypotheses of Ye's Theorem, 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 Ye's Theorem, but with $\bar{\alpha}=\max\{\bar{\alpha}_1,\bar{\alpha}_2\}$.* ■

Proposition 2. [Ye94, example 1][6] *For any $d\in\mathbb{R}\setminus\{0\}$, the function $x^d:(0,\infty)\longrightarrow\mathbb{R}$ satisfies the hypotheses of Ye's Theorem with $\bar{\alpha}=\left|\frac{d-1}{2}\right|$. Furthermore, f satisfies the hypotheses of Ye's Theorem $\Longrightarrow af+b$ does as well, for any $a,b\in\mathbb{R}$.* ■

An additional technical result we will need is an upper bound on a particular collection of the $\bar{\alpha}$-invariants from Ye's Theorem. Let $\mathbb{R}^+$ denote the positive real numbers and observe that $|x|=|\frac{x^k}{x}|^{1/(k-1)}$.

Theorem 3. *Let $\bar{\boldsymbol{\alpha}}\boldsymbol{(D,m)}:=\sup_{(f,I)}\left\{\sup_{k>1 \text{ and } x\in I\setminus\{c\}}\left\{\left|\frac{(x-c)^k f^{(k)}(x-c)}{k!(x-c)f'(x-c)}\right|^{1/(k-1)}\right\}\right\}$, where the $f\in\mathbb{R}[x_1]$ ranges over all t-nomials with $t\leq m$, I ranges over all the maximal open subsets of $\mathbb{R}^+$ such that $\pm f$ is convex on (and has a root in) I, and c is $\emptyset$ or the unique root of f' in I according as f' lacks or has roots in I. Then $\bar{\alpha}(D,3)\leq\frac{(D-1)^2}{2}$, $\bar{\alpha}(D,2)\leq\frac{D-1}{2}$, and $\bar{\alpha}(D,m+1)\geq\bar{\alpha}(D,m)$.*

One application of our globalized version of Smale's local α-invariant is the following generalization of theorem 1.

Theorem 4. *Following the notation of theorem 1, suppose instead that f is an m-nomial. Then one can find ε-approximations to all the roots of f in I within $\mathcal{O}\left(\log D(\log\log\frac{R}{\varepsilon}+\log\bar{\alpha}(D,m))+\mathcal{K}_{D,m}\right)$*

[6]The statement from [Ye94] applied only to $d\in\mathbb{Z}\setminus\{0\}$ but can be extended to real d without difficulty. The second portion of the proposition (on affine invariance) is elementary.

arithmetic operations, where the underlying $\mathcal{O}$-constant is $\mathcal{O}(m^3 2^m)$, and $\mathcal{K}_{D,m}$ denotes the arithmetic complexity of counting the roots of a general degree D m-nomial in an interval.

Theorem 4 thus opens up the possibility of super-fast m-nomial solving when $m>3$ is fixed, contingent on the solution of open problems (a) and (b) mentioned earlier.

To conclude our introduction, let us recall the classical construction of Sturm sequences and then make some further remarks on the complexity of real root counting.

Definition 2. *For any sequence of real numbers $s := (s_1, \ldots, s_k)$, the* **number of sign alternations of $\boldsymbol{s}$, $\boldsymbol{N_s}$**, *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 $(p_0, p_1, \ldots, p_K)$ 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 unique element of the sequence equal to 0. We call $(p_0, \ldots, p_K)$ the* **Sturm sequence** *of f.* $\diamond$

Lemma 1. [Stu35, Roy96] *Following the notation of definition 2, let $a, b \in \mathbb{R}$ with $\alpha \leq \beta$, $A := (p_0(\alpha), \ldots, p_K(\alpha))$, and $B := (p_0(\beta), \ldots, p_K(\beta))$. Then, provided $f(\alpha)f(\beta) \neq 0$, the number of distinct roots of f in the closed interval $[\alpha, \beta]$ 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.* $\blacksquare$

Remark 2. *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 5. *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 $\mathcal{O}(\log^2 D)$ arithmetic operations.*

Extending our last theorem to polynomials with more monomials is subtle: In particular, 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 $p_0(x) := x^{2D} + x^{D+1} + x^D + 1$ and $p_1 := p_0' = 2Dx^{2D-1} + (D+1)x^D + Dx^{D-1}$. 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 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.

Interestingly, if one insists on remaining with Sturm sequences, then one is naturally lead to the univariate **sparse discriminant**. Briefly, given an m-nomial f with indeterminate coefficient vector c, its sparse discriminant, $\boldsymbol{\Delta(f)}$, is the unique (up to sign) irreducible polynomial in $\mathbb{Z}[c]\setminus\{0\}$, of lowest degree, which is not identically zero but vanishes whenever c is specialized so that f has a nonzero complex root in common with f'. It is not hard to see that when the coefficients of f are all constants then $\Delta(f)$ is just the last entry of the Sturm sequence of f. So the following is clear.

Proposition 3. *Suppose $f \in \mathbb{R}[x_1]$ has degree D and exactly m monomial terms. Then, if one could compute, for all $m, D \in \mathbb{N}$ and $r \in \mathbb{R}$, the signs of the Sturm sequence of f evaluated at r in time polynomial in m and $\log D$, then one could at least as quickly decide the vanishing of $\Delta(f)$.* ■

Perhaps not coincidentally (recalling theorem 1), $\Delta(f)$ **can** be computed in polynomial time when f is a trinomial: up to sign, the formula is simply

$$\Delta(c_0 + c_1x^{a_1} + c_2x^{a_2}) = a_2^{a_2}c_2^{a_1}c_0^{a_2-a_1} + (-1)^{a_2-1}a_1^{a_1}(a_2-a_1)^{a_2-a_1}c_1^{a_2}$$

[GKZ94, pg. 406]. 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, $\Delta(c_0 + c_1x^6 + c_2x^{10} + c_3x^{31})$ is the following 21-nomial:

$$\begin{gathered}
-2^{10}3^{21}5^{10}7^{21}c_2^{31}c_0^{4} - 2^{13}3^{24}5^{5}7^{21}c_2^{28}c_0^{2}c_1^{5} - 2^{14}3^{27}7^{21}c_2^{25}c_1^{10} + 2^{8}3^{15}5^{6}7^{16}(19)31^{6}11731c_2^{23}c_3^{2}c_0^{7}c_1^{3} \\
-2^{8}3^{13}(5)7^{13}(19)31^{4}(4931)(11924843839)c_2^{20}c_3^{2}c_0^{5}c_1^{8} - 2^{6}3^{13}5^{6}7^{13}31^{13}c_2^{18}c_3^{4}c_0^{12}c_1 - 2^{8}3^{16}5^{12}7^{13}(17)31^{2}(3629537)c_2^{17}c_3^{2}c_0^{3}c_1^{13} \\
-2^{6}3^{9}(5)7^{11}(29)31^{11}(6361)(7477163)c_2^{15}c_3^{4}c_0^{10}c_1^{6} - 2^{10}3^{19}5^{22}7^{13}c_2^{1}4c_3^{2}c_0c_1^{18} - 2^{6}3^{8}5^{12}7^{7}31^{9}(857)(295269701)c_2^{12}c_3^{4}c_0^{8}c_1^{11} \\
-2^{7}3^{8}(5)7^{7}31^{18}(3327253)c_2^{10}c_3^{6}c_0^{15}c_1^{4} - 2^{6}3^{4}5^{22}7^{4}31^{7}160730667473c_2^{9}c_3^{4}c_0^{6}c_1^{16} + 2^{4}3^{3}5^{12}7^{4}(11)(29)31^{16}(233)(1559)c_2^{7}c_3^{6}c_0^{13}c_1^{9} \\
-2^{6}5^{32}31^{5}(248552699041)c_2^{6}c_3^{4}c_0^{4}c_1^{21} - 2^{2}3^{4}57^{4}(11)(29)31^{25}c_2^{5}c_3^{8}c_0^{20}c_1^{2} - 2^{5}(3)5^{22}31^{14}(113)(317)(461)c_2^{4}c_3^{6}c_0^{11}c_1^{14} \\
-2^{7}3^{3}5^{42}31^{3}(41)(367)c_2^{3}c_3^{4}c_0^{2}c_1^{26} - 2^{6}(3)5^{12}(19)31^{23}c_2^{2}c_3^{8}c_0^{18}c_1^{7} + 2^{4}3^{4}5^{32}31^{12}c_2c_3^{6}c_0^{9}c_1^{19} - 31^{31}c_3^{10}c_0^{25} - 2^{6}3^{6}5^{50}c_3^{4}c_1^{31}
\end{gathered}$$

which has 47 digits in its largest coefficient.

So proposition 3 thus points to yet another gap in what is known about the complexity of discriminant computation: it is known that sparse discriminants for **bi**variate polynomials can be computed in polynomial time[7] only if $\mathbf{NP} \subseteq \mathbf{BPP}$ [Roj00a, Main Theorem 1.4]. This putative inclusion of two well-known complexity classes (see [Pap95] for a beautiful introduction to complexity theory) is currently considered quite unlikely. The intrinsic complexity of univariate sparse discriminants is even less understood, still lacking even a conditional certification of hardness.

Remark 3. *Another interesting open question is whether there is a randomized* **complex** *analogue of theorem 1, i.e., is there an algorithm for ε-approximating a* **single** *complex root of trinomial, with* **average***-case arithmetic complexity polynomial in* $\log D$*? This would be a first step toward a sparse univariate analogue of* **Smale's 17$^{\underline{\text{th}}}$ Problem** [Sma00].

[7]On a BSS machine over $\mathbb{R}$ with equality test but no inequality tests (or, equivalently, a BSS machine over $\mathbb{C}$).

Smale's 17th Problem asks for a randomized polynomial time algorithm for finding a single **approximate**[8] *root of a general multivariate polynomial system, and has a partial affirmative answer in the special case of "dense" multivariate systems* [BCSS98]. $\diamond$

We state our underlying algorithms in the next section. We then prove theorems 1, 4, and 5 in section 3, and conclude with the proofs of theorems 2 and 3 in sections 4 and 5.

2. The Algorithm and Subroutines

The central algorithm we use to prove theorems 1 and 4, `MNOMIALSOLVE`, is detailed below. We point out that this algorithm makes some mild restrictions on the input, and is thus actually a subroutine of the true algorithms underlying theorems 1 and 4. As for a succinct description, `MNOMIALSOLVE` subdivides 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. Then, using special properties of sparse polynomials, a suitable combination of bisection and Newton iteration (detailed in the subroutine `HYBRID`) yields approximations to all the roots. Each approximation is guaranteed to correspond to its own unique root by a refined Sturm sequence calculation (applied to the intervals where f behaves poorly), done in another subroutine called `FASTERCOUNT`. The aforementioned subroutines are described shortly after our main algorithm.

ALGORITHM MNOMIALSOLVE

INPUT A univariate polynomial $f(x) := c_0 + c_1x^{a_1} + \cdots + c_{m-1}x^{a_{m-1}} \in \mathbb{R}[x]$ with exactly $m \geq 2$ monomial terms (where $0 < a_1 < \cdots < a_{m-1} = D$), real numbers R and ε with $0 < \varepsilon < R$, and an upper bound $\bar{\alpha}^*$ on $\bar{\alpha}(D, m)$.

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

DESCRIPTION

Step 0 Set $Z := \emptyset$ and $I := (0, R)$.

Step 1 If $m = 2$ then, if $f(0)f(R) < 0$, find an ε-approximation z to the unique root of f in I via subroutine HYBRID (with input $(\varepsilon, R, f, \bar{\alpha}^*)$, set $Z := \{z\}$, and STOP. Otherwise, if $m = 2$ and $f(0)f(R) \geq 0$, then set $Z := \{R\}$ and STOP.

Step 2 Recursively applying algorithm MNOMIALSOLVE, find sets of $\frac{\varepsilon}{2}$-approximations (of the correct cardinality) for the roots in I of f'/x^{a_1-1} and f''/x^e, where e is a_2-2 or a_1-2 according as a_1 is 1 or not. Call these sets of approximations Z_1 and Z_2 respectively.

Step 3 Letting $(w_1, \ldots, w_{\tilde{k}})$ be the sorted sequence of points in $Z_1 \cup Z_2$, define $\tilde{I}_i := (w_i - \frac{\varepsilon}{2}, w_i + \frac{\varepsilon}{2})$ for all $i \in \{1, \ldots, \tilde{k}\}$. Then define $\tilde{\mathcal{I}} := \{\tilde{I}_j\}_{i=1}^{\tilde{k}}$ and define $\mathcal{I} := \{I_j\}_{j=1}^{k}$ to be the collection of the closures of the intervals constituting $I \setminus (\tilde{I}_1 \cup \cdots \cup \tilde{I}_{\tilde{k}})$.

Step 4 Using subroutine FASTERCOUNT, count the number, n_j, of roots of f in each $\tilde{I}_j$.

[8] The notion of an approximate root is formalized in the next section. See [Sma86, BCSS98] for further background.

Step 5 `For every` j `with` $n_j > 0$, `append` n_j `copies of the midpoint` w_j `of` $\tilde{I}_j$ `to` Z.
Step 6 `By computing the sign of` f `at the endpoints of each` I_i, `determine which of the remaining intervals contain a root of` f, `and delete every` I_i **not** `containing a root of` f `from` $\mathcal{I}$.
Step 7 `FOR` $j \in \{1, \ldots, k\}$ `DO`
Step 7(a) `Let` c_j `be the unique end-point of` I_j `which is of distance` $< \frac{\varepsilon}{2}$ `of a root of` f'. `Also let` δ `be 1 or` -1 `according as` c_j `is a left or right end-point.`
Step 7(b) `By evaluating` f `at the end-points of` I_j `and using subroutine HYBRID with input` $(\varepsilon, R, f(\delta(x - c_j)), \bar{\alpha}^*)$, `find an` ε`-approximation for every root of` f `contained in` I_j, `and append all these approximations to` Z. `END FOR`
Step 8 `OUTPUT` Z `and STOP.`

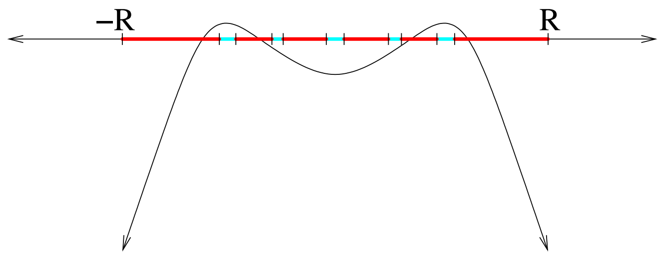

FIGURE 1. $[-R, R]$ can be expressed as a union of 6 intervals on which $\pm f$ is convex and monotonic, and 5 additional intervals of width $< \varepsilon$ containing roots of $f'f''$.

SUBROUTINE HYBRID [Ye94, pg. 277]

INPUT $\varepsilon > 0$, $R \in \mathbb{R}$ (with $0 < \varepsilon < R$), a monotonic analytic function $\phi : (0, R) \longrightarrow \mathbb{R}$ (with $\phi(0)\phi(R) < 0$ and $\pm\phi$ convex on $(0, R)$), and an upper bound $\bar{\alpha}$ on $\sup\limits_{x \in (0,R)\ ,\ k > 1} \left\{ x \left| \frac{\phi^{(k)}(x)}{k!\phi'(x)} \right|^{1/(k-1)} \right\}$.

OUTPUT An ε-approximation of the unique root of ϕ in $(0, R)$.

DESCRIPTION

Step 0 Defining b_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, let $c_i := c_i^2 - 1$ for all $i > 0$, and compute $c_1, \ldots, c_K$ where K is the first positive integer with $c_K = c_0^{2^K} \geq \frac{b}{a}$. Then set $\hat{x} := \varepsilon$, $\hat{k} := K$, and replace ϕ by $-\phi$ if ϕ is increasing.

Step 1 If $\phi(b_{\hat{k}-1}\hat{x}) > 0$ then set $\hat{x} := b_{\hat{k}-1}\hat{x}$ and $\hat{k} := \hat{k} - 1$ and GOTO Step 1.

Step 2 Perform $\log_2(3 + \log_2 \frac{R}{\varepsilon})$ iterations of Newton's method (with $\hat{x}$ as the starting point), then OUTPUT the very last iterate.

Remark 4. *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* $x^{363} = \underbrace{(\cdots(x^2)\cdots)^2}_{8} \cdot \underbrace{(\cdots(x^2)\cdots)^2}_{6} \cdot \underbrace{(\cdots(x^2)\cdots)^2}_{5} \cdot ((x^2)^2)^2 \cdot x^2 \cdot x,$

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 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 simply use the fastest available algorithm, e.g., [LM01] if nothing faster is available for the input m-nomial, and STOP.

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

Step 2 If $p_i\in\mathbb{R}$ then set $K:=i$ and GOTO Step 4.

Step 3 Write $p_i(x):=u_1x^{a_1}-u_0x^{a_0}$, $p_{i-1}(x):=v_1x^{b_1}-v_0x^{b_0}$ (where $a_1>a_0$ and $b_1>b_0$), and let $c_j:=\lceil\frac{b_j-a_1}{a_1-a_0}\rceil$ 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 1.

Step 5 Using recursive squaring (cf. remark 4), evaluate and RETURN N_A-N_B (cf. definition 2) where A (resp. B) is $(p_0(a),\ldots,p_K(a))$ (resp. $(p_0(b),\ldots,p_K(b))$).

3. Correctness and Complexity: Proving Theorems 1, 4, and 5

Proof of Theorem 1:
The second portion follows immediately from lemma 1 and theorem 5. The first assertion follows immediately from theorems 4 and 3 and the second assertion. ■

Proof of Theorem 4:
Henceforth, for convenience, we will say "time" in place of "arithmetic complexity." First note that a quick check of the coefficients, and dividing by a suitable monomial, allows us to reduce to the case where $m\geq 2$ is exactly the number monomial terms of f. (We can clearly output $\emptyset$ or $\mathbb{R}$ for Z according as m is 1 or 0.) Also note that we can evaluate f (and f' and f'') within $\mathcal{O}(m\log D)$ arithmetic operations, invoking repeated squaring (cf. remark 4). Finally, note that we can simply intersect $[a-R,a+R]$ with the positive and negative rays (and substitute $x\mapsto -x$ as necessary), and evaluate f at $\{0,a-R,a+R\}$ in time $\mathcal{O}(m\log D)$, to reduce to the case where the $[a-R,a+R]$ is contained in the positive ray. We will then use algorithm `MNOMIALSOLVE` to prove theorem 4.[9]

Toward this end, since our initial simplifications take only time $\mathcal{O}(m\log D)$, it will clearly suffice to show that `MNOMIALSOLVE` is correct and satisfies the complexity bound stated in theorem 4. (Note in particular that the intervals on which we apply `HYBRID` are strictly contained in the intervals on which $\bar{\alpha}(D,m)$ is defined.) The correctness of `MNOMIALSOLVE` is straightforward, 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 complexity bound.

Note that f'/x^{a_1-1} has exactly $m-1$ monomial terms, f''/x^e has $\leq m-1$ monomial terms, and both polynomials have nonzero constant terms. So the recursion in Step 2 is

[9]The value we use for $\bar{\alpha}^*$ will simply be the best currently available upper bound for $\bar{\alpha}(D,m)$. Clearly, as of this writing, this is practical only for $m\leq 3$ (cf. theorem 3).

indeed well-defined. Descartes' Rule then implies that the number of intervals in $\mathcal{I} \cup \tilde{\mathcal{I}}$ is no more than $4m-7$.

Let $\mathcal{C}(D,m)$ denote the time needed for MNOMIALSOLVE to execute completely. Clearly, assuming subroutine HYBRID runs in time $\mathcal{O}((m\log D)(\log D\log\log\frac{R}{\varepsilon}+\log\bar{\alpha}^*))$ (which is proved below), we then have that $\mathcal{C}(D,m)$ must satisfy the following recurrence relation: $\mathcal{C}(D,m)\leq\mathcal{C}(D-1,m-1)+\mathcal{C}(D-2,m-1)+\mathcal{O}\left(m(m\log D)\left(\log D\log\log\frac{R}{\varepsilon}+\log\bar{\alpha}^*\right)+m\mathcal{K}_{D,m}\right)$, where the first two terms correspond to finding the $\frac{\varepsilon}{2}$-approximations of the roots of f' and f'' , the first term in the $\mathcal{O}$ corresponds to the application of algorithm HYBRID, and the last term corresponds to the application of algorithm FASTERCOUNT.

Knowing that $\mathcal{C}(D,3)=\mathcal{O}(\log D\log\log\frac{R}{\varepsilon}+\log^2 D)$ (thanks to theorems 3 and 5), we can expand $\mathcal{C}(D,m)$ directly to obtain

$$\mathcal{C}(D,m)\leq\sum_{i=0}^{m-2}\mathcal{O}\left(i^2 2^i\left(\log(D-i)\log\log\frac{R}{\varepsilon}+\log(D-i)\log\bar{\alpha}(D-i,m-i)+i\mathcal{K}_{D-i,m-i}\right)\right),$$

with all underlying "$\mathcal{O}$" constants identical. To conclude, the right-hand side is clearly $\mathcal{O}\left(m^2 2^m(m\log^2 D\log\log\frac{R}{\varepsilon}+m\log D\log\bar{\alpha}(D,m)+\mathcal{K}_{D,m})\right)$. ■

Proof of Theorem 5: The complexity bound follows immediately from the upper bound on K, using algorithm FASTERCOUNT. 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:
See [Ye94, sec. 3] for a detailed proof. In particular, the preceding reference implies that HYBRID is indeed correct, and always terminates within $\mathcal{O}(\log\log\frac{R}{\varepsilon}+\log\bar{\alpha})$ steps, where a step is either a field operation in the field over $\mathbb{Q}$ generated by the evaluations of ϕ and ϕ', or a single evaluation of ϕ or ϕ'. Since we are working with m-nomials of degree $\leq D$, recursive squaring (cf. remark 4) easily implies that each evaluation of ϕ or ϕ' requires only $\mathcal{O}(m\log D)$ arithmetic operations. So we are done. ■

Proof of Correctness and Complexity Analysis of FASTERCOUNT:
The correctness of FASTERCOUNT when $m>3$ relies on the available correct algorithm, so there is nothing to prove. For $m\leq 3$, correctness follows easily from lemma 1 upon noting a simple fact: The remainder of $\frac{v_1x^{b_1}-v_0x^{b_0}}{u_1x^{a_1}-u_0x^{a_0}}$ is nothing more than the reduction of $v_1x^{b_1}-v_0x^{b_0}$ modulo $u_1x^{a_1}-u_0x^{a_0}$. 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 $-(v_1x^{b_1}-v_0x^{b_0})$ (modulo $u_1x^{a_1}-u_0x^{a_0}$) 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 $\mathcal{O}(\log D)$ arithmetic operations, and that $K=\mathcal{O}(\log D)$ at the termination of the algorithm, i.e., there are only $\mathcal{O}(\log D)$ 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 $\ell_{i+1} \leq \frac{1}{2}\ell_i \Longrightarrow \ell_{i+2} \geq \frac{1}{2}\ell_{i+1}$ (if $\ell_{i+2} \neq 0$) and thus $\ell_{i+3} \leq \frac{1}{2}\ell_i$. Similarly, $\ell_{i+1} \geq \frac{1}{2}\ell_i \Longrightarrow \ell_{i+2} \leq \frac{1}{2}\ell_i$. So $K \leq 3\lceil \log_2 D \rceil + 2 = \mathcal{O}(\log D)$ and we are done. ■

4. 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 [LRW01, 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 $C \Longrightarrow x$ is a root of $F_v := (f, \frac{\partial f}{\partial x_2})$. By [Roj00a, lemma 4.2] we can then find a monomial change of variables, within $\mathcal{O}(\log^{6.376} D)$ bit operations and $\mathcal{O}(\log D(\log^2 D + \log\log \frac{R}{\varepsilon}))$, 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 $\mathcal{O}(\log^2 D)$. We can then solve g_2 to accuracy $\min\{\varepsilon^{\mathcal{O}(D^2)}, 1\}$ via theorem 1, back-substitute the roots into g_1, solve the univariate specialized g_1 to accuracy $\min\{\varepsilon^{\mathcal{O}(D^2)}, 1\}$, then invert our monomial map via [Roj00a, Main Theorem 1.3] to recover the roots of F_v. So we can indeed find isolated vertical tangents within the stated complexity bound.

To conclude, note that [LRW01, lemma 9] tells us that x is an isolated inflection point of $C \Longrightarrow [\partial_1^2 f \cdot (\partial_2 f)^2 - 2\partial_1\partial_2 f \cdot \partial_1 f \cdot \partial_2 f + \partial_2^2 f \cdot (\partial_1 f)^2] = 0$. 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 efficiently solve for x as follows: (1) solve the above cubic for $\frac{R}{S}$ up to accuracy $\min\{\varepsilon^{\mathcal{O}(D)}, 1\}$ 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]. So we can ε-approximate the inflection points of C within the stated asymptotic complexity bound as well, and we are done. ■

5. The Proof of Theorem 3

Definition 3. *For any $a\in\mathbb{R}$ and $k\in\mathbb{N}\cup\{0\}$, let $(a)_k:=a(a-1)\cdots(a-k+1)$. (So $(a)_1=a$ and $a\in\mathbb{N}\Longrightarrow(a)_a=a!$.)* $\diamond$

The very last bound follows directly from the definition of $\bar{\alpha}(D,m)$.

To prove the first two bounds, first note that Rolle's Theorem implies that every I indeed contains at most one root of f'. The second bound then follows immediately from proposition 2. Note also that propositions 1 and 2 imply the first inequality in the special case of a trinomial of the form $c_0+c_1x^{a_1}+c_2x^{a_1}$ where (c_0,c_1,c_2) has at most one sign alternation.

To prove the remaining cases of the first inequality, it clearly suffices to consider the trinomial $f(x)=x^{a_2}-Ax^{a_1}+B$ where $A,B>0$ and $a_1,a_2\in\mathbb{N}$ with $D=a_2>a_1\geq 2$. (We can assume $a_1>1$ since $a=1$ easily implies that $\pm f$ is always convex and we can then conclude via proposition 1.)

Now note that $f'(x)=a_2x^{a_2-1}-a_1Ax^{a_1-1}=a_2x^{a_1-1}\left(x^{a_2-a_1}-\frac{a_1A}{a_2}\right)$, and $f''(x)=a_2(a_2-1)x^{d-2}-a_1A(a_1$ So to analyze $\bar{\alpha}$, the substitution $x\mapsto -x$ easily shows us that we can restrict to sub-intervals of the positive ray.

Under our assumptions so far, there will always be a unique positive root $x_1:=(\frac{a_1A}{a_2})^{1/(a_2-a_1)}$ of f'. If $f(x_1)>0$ then f has no roots greater than 0 and is not considered in the definition of $\bar{\alpha}(D,m)$. We therefore assume that $f(x_1)<0$ and thus that f has exactly two roots in $(0,\infty)$: one in the interval $(0,x_1)$ and another in (x_1,∞). More to the point, these two intervals are the only ones we need to consider in bounding $\bar{\alpha}(D,m)$, so let us now begin majorizing on these two intervals.

(x_1,∞):
Note that f is a convex function in this interval and let $y:=x-x_1$ be a new variable. Then f becomes $\phi(y)=(y+x_1)^{a_2}-a(y+x_1)^{a_1}+b$ and we have $\phi'(y)=a_2(y+x_1)^{a_2-1}-a_1A(y+x_1)^{a_1-1}$. Also, for $2\leq k\leq a_1$ we have $\phi^{(k)}(y)=(a_2)_k(y+x_1)^{a_2-k}-(a_1)_kA(y+x_1)^{a_1-k}$, and for $a_1<k\leq a_2$ we have $\phi^{(k)}(y)=(a_2)_k(y+x_1)^{a_2-k}$. Furthermore, we also have

$$\left|\frac{\phi^{(k)}(y)}{k!\phi'(y)}\right|^{1/(k-1)}=\left|\frac{(a_2)_k(y+x_1)^{a_2-k}-(a_1)_kA(y+x_1)^{a_1-k}}{k!(a_2(y+x_1)^{a_2-1}-a_1A(y+x_1)^{a_1-1})}\right|^{1/(k-1)}=\cdots$$

$$\cdots=\left|\frac{(a_2)_k(y+x_1)^{a_2-a_1}-(a_1)_kA}{k!a_2(y+x_1)^{k-1}\left((y+x_1)^{a_2-a_1}-\frac{a_1A}{a_2}\right)}\right|^{1/(k-1)}=\frac{1}{|y|}\left|\frac{(a_2)_k}{k!a_2}\right|^{1/(k-1)}\left|\frac{y^{k-1}\left((y+x_1)^{a_2-a_1}-\frac{(a_1)_kA}{(a_2)_k}\right)}{(y+x_1)^{k-1}\left((y+x_1)^{a_2-a_1}-x_1^{a_2-a_1}\right)}\right|^{1/(k-1)}$$

$$\leq\frac{1}{|y|}\cdot\frac{a_2-1}{2}\left|\frac{y^{k-1}}{(y+x_1)^{k-1}}\cdot\frac{(y+x_1)^{a_2-a_1}-\frac{(a_1)_kA}{(a_2)_k}}{(y+x_1)^{a_2-a_1}-x_1^{a_2-a_1}}\right|^{1/(k-1)},$$

since $\left|\frac{(a_2)_k}{k!d}\right|^{1/(k-1)} \leq \frac{a_2-1}{2}$. Now let $z := y/x_1$ and $\rho := \left(\frac{(a_1)_k A}{(a_2)_k}\right)^{1/(a_2-a_1)} /x_1$. Then $\rho = \left(\frac{(a_1-1)_{k-1}}{(a_2-1)_{k-1}}\right)^{1/(a_2-a_1)} < 1$ and, for $z \geq -1$,

$$\left|\frac{y^{k-1}}{(y+x_1)^{k-1}} \cdot \frac{(y+x_1)^{a_2-a_1} - \frac{(a_1)_k A}{(a_2)_k}}{(y+x_1)^{a_2-a_1} - x_1^{a_2-a_1}}\right|^{1/(k-1)} = \left|\frac{z^{k-1}}{(z+1)^{k-1}} \cdot \frac{(z+1)^{a_2-a_1} - \rho^{a_2-a_1}}{(z+1)^{a_2-a_1} - 1}\right|^{1/(k-1)} = \cdots$$

$$\cdots = \left|\frac{z^{k-2}}{(z+1)^{k-1}} \cdot \frac{(z+1)^{a_2-a_1} - \rho^{a_2-a_1}}{1+(z+1)+\cdots+(z+1)^{a_2-a_1-1}}\right|^{1/(k-1)} = \left|\frac{z^{k-2}}{(z+1)^{k-2}} \cdot \frac{(z+1)^{a_2-a_1} - \rho^{a_2-a_1}}{(z+1)+(z+1)^2\cdots+(z+1)^{a_2-a_1}}\right|^{1/(k-1)} = \cdots$$

$$\cdots = \left|\frac{z^{k-2}}{(z+1)^{k-2}}\right|^{1/(k-1)} \cdot \left|\frac{(z+1)^{a_2-a_1} - \rho^{a_2-a_1}}{(z+1)+(z+1)^2\cdots+(z+1)^{a_2-a_1}}\right|^{1/(k-1)}.$$

Obviously, for $z \geq 0$, both quantities of the product are bounded above by 1, which leads to $\left|\frac{\phi^{(k)}(y)}{k!\phi'(y)}\right|^{1/(k-1)} \leq \frac{1}{y} \cdot \frac{a_2-1}{2}$. Thus, on this interval, the quantity we need to bound is no more than $\frac{a_2-1}{2} = \frac{D-1}{2}$.

Intervals Within $(0, x_1)$: First consider the case where $f(x_2) > 0$ — that is, the root of f in $(0, x_1)$ is actually in (x_2, x_1) (and thus, via the definition of $\bar{\alpha}(D,m)$, we can ignore the interval $(0, x_2)$). Recall that $x_1 = \left(\frac{a_1 A}{a_2}\right)^{1/(a_2-a_1)}$ and, similarly, $x_2 = \left(\frac{a_1(a_1-1)A}{a_2(a_2-1)}\right)^{1/(a_2-a_1)} < x_1$. Again, f is a convex function in this interval. Let $y = x - x_1$ as before. Then $f(x)$ can be replaced by $\phi(y) := (y+x_1)^{a_2} - a(y+x_1)^{a_1} + b$, and it suffices to find the the root of ϕ in $(x_2 - x_1, 0)$. Note that we still have all the inequalities we had before. The only thing we need to prove is that

$$\left|\frac{z^{k-2}}{(z+1)^{k-2}}\right|^{1/(k-1)} \left|\frac{(z+1)^{a_2-a_1} - \rho^{a_2-a_1}}{(z+1)+(z+1)^2+\cdots+(z+1)^{a_2-a_1}}\right|^{1/(k-1)}$$

is bounded in $(x_2/x_1 - 1, 0)$ or $((\frac{a_1-1}{a_2-1})^{1/(a_2-a_1)} - 1, 0)$. Since $x_2/x_1 = \left(\frac{a_1-1}{a_2-1}\right)^{1/(a_2-a_1)} \geq \rho$, we have $\left|\frac{(z+1)^{a_2-a_1}-\rho^{a_2-a_1}}{(z+1)+(z+1)^2+\cdots+(z+1)^{a_2-a_1}}\right|^{1/(k-1)} \leq 1$. Also in this interval, $\left|\frac{z^{k-2}}{(z+1)^{k-2}}\right|^{1/(k-1)} \leq \left|\frac{1}{(z+1)^{k-2}}\right|^{1/(k-1)} \leq \left(\frac{a_2-1}{a_1-1}\right)^{1/(a_2-a_1)} \leq a_2 - 1$. Hence, $\left|\frac{\phi^{(k)}(y)}{k!\phi'(y)}\right|^{1/(k-1)} \leq \frac{1}{-y} \cdot \frac{(a_2-1)^2}{2}$. Thus, on this interval, the quantity we need to maximize is no more than $\frac{(D-1)^2}{2}$.

To conclude, consider the case $f(x_2) < 0$ — that is, the root is actually in $(0, x_2)$ (and again, via the definition . Now $f(x)$ is a concave function in $(0, x_2)$. For this case, we search the root in x-space. Recall that $f'(x) = a_2 x^{a_2-1} - a_1 A x^{a_1-1}$ and, for $2 \leq k \leq a_1$, we have $f^{(k)}(x) = (a_2)_k x^{a_2-k} - (a_1)_k A x^{a_1-k}$, and for $a_1 < k \leq d$ we have $f^{(k)}(x) = (a_2)_k x^{a_2-k}$. Now $\left|\frac{f^{(k)}(x)}{k!f'(x)}\right|^{1/(k-1)} = \left|\frac{(a_2)_k x^{a_2-k} - (a_1)_k A x^{a_1-k}}{k!(a_2 x^{a_2-1} - a_1 A x^{a_1-1})}\right|^{1/(k-1)} = \left|\frac{(a_2)_k x^{a_2-a_1} - (a_1)_k A}{k! a_2 x^{k-1}\left(x^{a_2-a_1} - \frac{a_1 A}{a_2}\right)}\right|^{1/(k-1)} =$

$\frac{1}{x}\left|\frac{(a_2)_k}{k!d}\right|^{1/(k-1)}\left|\frac{x^{a_2-a_1}-\frac{(a_1)_kA}{(a_2)_k}}{x^{a_2-a_1}-x_1^{a_2-a_1}}\right|^{1/(k-1)} \leq \frac{1}{x}\cdot\frac{a_2-1}{2}\left|\frac{x^{a_2-a_1}-\frac{(a_1)_kA}{(a_2)_k}}{x^{a_2-a_1}-x_1^{a_2-a_1}}\right|^{1/(k-1)}$. So, for $0<z<x_2/x_1=(\frac{a_1-1}{a_2-1})^{1/(a_2-a_1)}<1$, we have $\left|\frac{x^{a_2-a_1}-\frac{(a_1A)_k}{(a_2)_k}}{x^{a_2-a_1}-x_1^{a_2-a_1}}\right|^{1/(k-1)}=\left|\frac{z^{a_2-a_1}-\rho^{a_2-a_1}}{z^{a_2-a_1}-1}\right|^{1/(k-1)}=\left|1-\frac{1-\rho^{a_2-a_1}}{1-z^{a_2-a_1}}\right|^{1/(k-1)}\leq \max\left\{1,\ \left|\frac{1-\rho^{a_2-a_1}}{1-z^{a_2-a_1}}\right|^{1/(k-1)}\right\}\leq\max\left\{1,\ \left|\frac{1}{1-z^{a_2-a_1}}\right|^{1/(k-1)}\right\}\leq\left|\frac{a_1-1}{a_2-a_1}\right|^{1/(k-1)}\leq a_1-1$. Hence, $\left|\frac{f^{(k)}(x)}{k!f'(x)}\right|^{1/(k-1)}\leq\frac{1}{y}\cdot\frac{(a_2-1)(a_1-1)}{2}$. Thus the quantity we need to maximize on this interval is no more than $\frac{(D-1)^2}{2}$.

Putting together the bounds we've found over the preceding intervals, we see that our first inequality holds and we are done. ■

6. Acknowledgments

The authors thank Felipe Cucker, John McDonald, Tom Michiels, and Steve Smale for some delightful conversations.

References

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

[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.

[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.

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

[LRW01] LI, Tien-Yien; Rojas, J. Maurice; and WANG, Xiaoshen, *"Counting Isolated Roots of Trinomials in the Plane and Beyond,"* submitted for publication, also available as Math ArXiV preprint `math.CO/0008069`.

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

[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.

[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.

[Ren89] Renegar, Jim, *"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] Rojas, J. Maurice, *"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.

[Roj00b] ________, *"Algebraic Geometry Over Four Rings and the Frontier to Tractability,"* invited paper, Contemporary Mathematics, vol. 270, Proceedings of a Conference on Hilbert's Tenth Problem and Related Subjects (University of Gent, November 1–5, 1999), edited by Jan Denef, Leonard Lipschitz, Thanses Pheidas, and Jan Van Geel, pp. 275–321, AMS Press (2000).

[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.

[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.

[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).

[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, College Station, TX 77843, USA
E-mail address: `rojas@math.tamu.edu`, *Web-Page:* `http://www.math.tamu.edu/~rojas`

Department of Management Sciences, College of Business Administration, University of Iowa, Iowa City, IA 52242, USA
E-mail address: `yinyu-ye@uiowa.edu`, *Web-Page:* `http://dollar.biz.uiowa.edu/col/ye`