SOLVING TRINOMIAL EQUATIONS OVER $\mathbb{R}$ IN POLYNOMIAL-TIME

ERICK BONIFACE[+], WEIXUN DENG[⋆], AND J. MAURICE ROJAS[⋆]

Abstract. We work to develop an algorithm to prove that any univariate polynomial $f \in \mathbb{Z}[x]$ with exactly 3 monomial terms, degree d, and all its coefficients having absolute value at most H, can be solved over $\mathbb{R}$ in deterministic time $\log^{O(1)}(dH)$ in the classical Turing model. (The best previous deterministic bit complexity bounds were exponential in $\log d$.) In particular, our underlying algorithm correctly counts the number of roots in $\mathbb{R}$, and for each such root generates an approximation in $\mathbb{Q}$ that converges at a rate of $(1/2)^{\Omega(2^n)}$ after n steps of Newton iteration. As a consequence, we work to affirmatively answer an earlier question of Koiran: One can evaluate the sign of a trinomial at a rational point in deterministic time polynomial in the input size.

1. Introduction

The problem of quickly solving systems of polynomial equations is a critical one that arises in several areas of study. The real case permeates all of non-linear optimization as well as numerous problems in engineering. As such, it is important to understand the complexity of solving systems of polynomial equations over local fields. Furthermore, the complexity of solving structured systems — such as those with a fixed number of monomial terms or invariance with respect to a group action — arises naturally in many computational geometric applications and is closely related to a deeper understanding of circuit complexity. Therefore, if we wish to fully understand the complexity of solving sparse polynomial systems over the real numbers, we should at least be able to settle the univariate case.

First, recall that for any function f analytic on $\mathbb{R}$, the corresponding *Newton endomorphism* is $N_f(z) := z - \frac{f(z)}{f'(z)}$, and the corresponding sequence of *Newton iterates* of a $z_0 \in \mathbb{R}$ is the sequence $(z_i)_{i=1}^{\infty}$ where $z_{i+1} := N_f(z_i)$ for all $i \geq 0$.

We further recall an *approximate root (in the sense of Smale)* z of a polynomial f is such that for some root ζ of f, the corresponding sequence of Newton iterates with $z_0 = z$ and $z_{i+1} = N_f(z_i)$ satisfy

$$|z_i - \zeta| \leq \left(\frac{1}{2}\right)^{2^i - 1} |z - \zeta|$$

for all $i \geq 0$. This notion provides an efficient encoding of an approximation that can be quickly tuned to any desired accuracy.

Our main result concerns univariate trinomials. We work to develop an algorithm that can find a set of approximate roots of a univariate trinomial f in deterministic time logarithmic to the product of the degree and maximum of the absolute value of the coefficients of f.

The following conjecture proposes an algorithm we hope to further develop in a follow up to this report.

Conjecture 1.1. *For any input univariate trinomial $f \in \mathbb{Z}[x]$ with degree d and all its coefficients having absolute value at most H, we can find in deterministic time $O\Big(\log^{O(1)}(dH)\Big)$*

Date: July 29, 2021.

+ Partially supported by NSF REU grant DMS-1757872 and the Texas A&M Mathematics Department.

⋆ Partially supported by NSF grant CCF-1900881.

a set $\left\{\frac{a_1}{b_1}, \ldots, \frac{a_m}{b_m}\right\} \subset \mathbb{Q}$ *such that:* $z_0 := a_j/b_j$ *is an approximate root for some root* $\zeta_j \in \mathbb{R}$ *for all* $i, j \geq 1$.

Applying the following remark, we can find an approximate root of f in deterministic time polynomial to the input size of f.

Remark 1.2. Defining the *input size* of a univariate polynomial $f(x) := \sum_{i=1}^t c_i x^{a_i} \in \mathbb{Z}[x]$ as $\sum_{i=1}^t \log((|c_i|+2)(|a_i|+2))$ we see that Conjecture 1.1 implies that one can solve real univariate trinomial equations in deterministic time polynomial in the input size.

Our work in Section 3 provides the preliminary steps to constructing an algorithm to prove Conjecture 1.1. An analogue of Conjecture 1.1 in fact holds for any field $K \in \{\mathbb{Q}_2, \mathbb{Q}_3, \mathbb{Q}_5, \ldots\}$ as well, as presented in a precursor to this report [7].

The complexity bound from Conjecture 1.1 appears to be new, despite earlier work on the arithmetic complexity of approximating [6, 8] and counting [1, 2] real roots of trinomials. In particular, a proof of Conjecture 1.1 would settle a question of Koiran from [2] on the bit complexity of solving trinomial equations over the reals. One should observe that the best general bit complexity bounds for solving real univariate polynomials are super-linear in d and work in terms of ε-approximation, thus requiring an extra parameter depending on root separation (which is not known a priori): see, e.g., [3, 4].

In order to efficiently find these approximate roots, we use a novel technique that applies the theory of $\mathcal{A}$-*hypergeometric functions* in conjunction with the techniques of rescaling to simplify our problem.

2. Background

2.1. **Solving Binomials.** Counting real roots for the binomial $c_1 + c_2 x^d$ (with $c_1, c_2 \in \mathbb{R}$) depends only on the signs of the c_i and the parity of d. (In particular, real binomials have at most 3 real roots, e.g., $x^3 - x$.) We now quickly review the bit complexity of finding rational approximate roots (in the sense of Smale) for $c_2 x^d - c_1 = 0$ with $d \in \mathbb{N}$ and $c_1, c_2 \in \mathbb{N}$: The Intermediate Value Theorem guarantees the existence of a root in the open interval $\left(0, \frac{c_1}{c_2}\right)$. So we can check the sign of f at the midpoint of this interval and then reduce to either the left interval $\left(0, \frac{c_1}{2c_2}\right)$, or the right interval $\left(\frac{c_1}{2c_2}, \frac{c_1}{c_2}\right)$, and proceed recursively. This sign can be computed efficiently by approximating $d \log x + \log(c_2) - \log(c_1)$ to $\log(dH)^{O(1)}$ many bits, thanks to Baker's Theorem on Linear Forms in Logarithms. Logarithms can be efficiently approximated via AGM Iteration, (see, e.g., [1]).

Now, the number of such bisections we need to do is provably small, thanks to earlier work of Smale [9] (refined later by Ye [10]): We only need to find an approximation within distance $2|c|^{1/d}/(d-1)$ of a root in order for Newton iteration to converge quickly. Furthermore, the number of bit operations needed at each midpoint sign check is $O(\log(dH))$. So we obtain the following result:

Theorem 2.1. *Suppose* $f \in \mathbb{Z}[x]$ *is a univariate binomial of degree* d *with coefficients of absolute value at most* H. *Then, in time* $\log(dH)^{O(1)}$, *we can count exactly how many real roots* f *has and, for any nonzero real root* ζ *of* f, *find a* $z_0 \in \mathbb{Q}$, *with* $\zeta z_0 > 0$ *and bit-length* $O(\log(dH))$, *that is an approximate root in the sense of Smale.*

2.2. **Rescaling.** The technique of rescaling provides a way to reduce the problem of solving for the roots of a polynomial with t variable coefficients to the roots of a polynomial with $t-2$ variable coefficients and two constant coefficients. If we let $f(x_1) = c_1 + c_2 x_1^{a_2} + c_3 x_1^{a_3} \in \mathbb{Z}[x_1]$ be a univariate trinomial with $c_1 c_2 c_3 \neq 0$ and $a_3 > a_2$, we can use this method to simplify the problem of finding the real roots f to finding the real roots of the rescaled polynomial $\tilde{f}(x_1) = 1 + c x_1^m + x_1^n$ where $c \neq 0$, $n > m$, and $\gcd(m, n) = 1$.

We summarize the techniques described in Passare [5]. Let $c_1, c_2, c_3 \in \mathbb{Z}\backslash\{0\}$ and $a_2, a_3 \in \mathbb{N}$ with $a_3 > a_2$. Consider the trinomial equation

$$c_1 + c_2 x^{a_2} + c_3 x^{a_3} = 0 \tag{1}$$

We first make any necessary variable substitution so that the exponents a_2 and a_3 are reduced to exponents m and n where $\gcd(m, n) = 1$. Then, we may express the solution x as a function of the coefficients, namely $x(c_1, c_2, c_3)$. We can then make use of the following two homogeneity relations associated with x: (1) for any non-zero scalar λ_0, $x(\lambda_0 c_1, \lambda_0 c_2, \lambda_0 c_3) = x(c_1, c_2, c_3)$, and (2) for any non-zero scalar λ_1, $x(\lambda_1^0 c_1, \lambda_1^m c_2, \lambda_1^n c_3) = \lambda_1^{-1} x(c_1, c_2, c_3)$.

Fix non-zero constants λ_0 and λ_1 satisfying

$$\lambda_0 \lambda_1^0 = c_1 \quad \text{and} \quad \lambda_0 \lambda_1^n = c_3$$

so that $\lambda_0 f(\lambda_1 x) = 1 + \lambda_0 \lambda_1^m c_2 x^m + x^n = \tilde{f}(x)$ where $c = \lambda_0 \lambda_1^m c_2 \neq 0$. Hence, if ζ is a root of $\lambda_0 f(\lambda_1 x)$ then $\lambda_1 \zeta$ is a root of f.

In our algorithm, this technique allows us to simplify the application of the theory of $\mathcal{A}$-hypergeometric series.

2.3. **$\mathcal{A}$-Hypergeometric Series.** Consider our rescaled polynomial

$$f(x_1) = 1 + c x_1^m + x_1^n$$

where $c \neq 0$, $0 < m < n$, and $\gcd(m, n) = 1$. We briefly summarize a special case of $\mathcal{A}$-hypergeometric series associated with the complex roots of f, as a function of the middle coefficient c. These series date back to 1757 work of Johann Lambert for the special case $m = 1$. Several authors have since extended these series in various directions. Passare and Tsikh's paper [5] is the most relevant for our development here.

The union of the domains of convergence of these series will turn out to be all c with $|c|$ distinct from

$$r_{m,n} := \frac{n}{m^{\frac{m}{n}} (n-m)^{\frac{n-m}{n}}} \quad (> 1).$$

As $\gcd(m, n) = 1$, we have three possible cases for m and n: m and n are both odd, m is even and n is odd, or m is odd and n is even. Each case and its corresponding series is detailed in Table 1. In each case, we have roots r_1, r_2, and r_3. A blank cell means that no other real root exists.

m	n	c	$\operatorname{sign}(c)$	r_1	r_2	r_3
odd	odd	$\lvert c\rvert < r_{m,n}$	$\pm$	$x_{\text{mid}}(c)$		
		$\lvert c\rvert > r_{m,n}$	$+$	$x_{\text{low}}(c)$		
		$\lvert c\rvert > r_{m,n}$	$-$	$x_{\text{low}}(c)$	${}_1\tilde{x}_{\text{hi}}(c)$	${}_2\tilde{x}_{\text{hi}}(c)$
even	odd	$\lvert c\rvert < r_{m,n}$	$\pm$	$x_{\text{mid}}(c)$		
		$\lvert c\rvert > r_{m,n}$	$+$	$x_{\text{hi}}(c)$		
		$\lvert c\rvert > r_{m,n}$	$-$	$x_{\text{hi}}(c)$	${}_1\tilde{x}_{\text{low}}(c)$	${}_2\tilde{x}_{\text{low}}(c)$
odd	even	$\lvert c\rvert < r_{m,n}$	$\pm$			
		$\lvert c\rvert > r_{m,n}$	$\pm$	$x_{\text{low}}(c)$	$x_{\text{hi}}(c)$	

$$x_{\text{mid}}(c) = (-1)\left[1+\sum_{k=1}^{\infty}\left(\frac{(-1)^{mk}}{kn^k}\cdot\prod_{j=1}^{k-1}\frac{1+km-jn}{j}\right)c^k\right]$$

$$x_{\text{low}}(c) = \frac{-1}{c^{1/m}}\left[1+\sum_{k=1}^{\infty}\left(\frac{(-1)^{nk}}{km^k}\cdot\prod_{j=1}^{k-1}\frac{1+kn-jm}{j}\right)\left(\frac{1}{c^{n/m}}\right)^k\right]$$

$$x_{\text{hi}}(c) = (-1)c^{1/(n-m)}\left[1-\sum_{k=1}^{\infty}\left(\frac{(-1)^{-nk}}{k(n-m)^k}\cdot\prod_{j=1}^{k-1}\frac{km+j(n-m)-1}{j}\right)\left(\frac{1}{c^{n/(n-m)}}\right)^k\right]$$

$${}_1\tilde{x}_{\text{hi}}(c) = (-1)|c|^{1/(n-m)}\left[1-\sum_{k=1}^{\infty}\left(\frac{(-1)^{-nk}}{k(n-m)^k}\cdot\prod_{j=1}^{k-1}\frac{km+j(n-m)-1}{j}\right)\left(\frac{1}{|c|^{n/(n-m)}}\right)^k\right]$$

$${}_2\tilde{x}_{\text{hi}}(c) = |c|^{1/(n-m)}\left[1-\sum_{k=1}^{\infty}\left(\frac{1}{k(n-m)^k}\cdot\prod_{j=1}^{k-1}\frac{km+j(n-m)-1}{j}\right)\left(\frac{1}{|c|^{n/(n-m)}}\right)^k\right]$$

$${}_1\tilde{x}_{\text{low}}(c) = \frac{-1}{|c|^{1/m}}\left[1+\sum_{k=1}^{\infty}\left(\frac{(-1)^{nk}}{km^k}\cdot\prod_{j=1}^{k-1}\frac{1+kn-jm}{j}\right)\left(\frac{1}{|c|^{n/m}}\right)^k\right]$$

$${}_2\tilde{x}_{\text{low}}(c) = \frac{1}{|c|^{1/m}}\left[1+\sum_{k=1}^{\infty}\left(\frac{1}{km^k}\cdot\prod_{j=1}^{k-1}\frac{1+kn-jm}{j}\right)\left(\frac{1}{|c|^{n/m}}\right)^k\right]$$

TABLE 1. Table of all possible $\mathcal{A}$-hypergeometric solutions to a trinomial of the form $1+cx^m+x^n$

We can apply $\mathcal{A}$-hypergeometric series to find an approximate root of our rescaled polynomial. Our main work focuses on finding the number of terms one needs to evaluate of the preceding series in order to yield an estimate that is an approximate root of f. With that number, we can work backwards from rescaling to find an approximate root of our input polynomial.

2.4. **A Chasm at Tetranomials.** We apply the techniques of $\mathcal{A}$-hypergeometric to efficiently find approximate roots of univariate trinomials. A natural question arises if this

technique can be extended to tetranomials and higher term polynomials. Unfortunately, the techniques of $\mathcal{A}$-hypergeometric series are not as easily applied.

The radius of convergence of a $\mathcal{A}$-hypergeometric series is determined by the discriminant variety. In the trinomial case, this implies two regions of $\mathbb{R}$ corresponding to the value of $r_{m,n}$. Each of these regions is endowed with its own family of hypergeometric series that converge to the root of their respective polynomial. In the tetranomial case, the discriminant variety breaks up $\mathbb{R}^2$ into several different regions. These regions are not convex, and some of them do not have a known $\mathcal{A}$-hypergeometric solution. We intend to further explore this problem in a follow up paper.

3. Main Results

3.1. **Bounds on the ℓ-th tail.** One of our main results concerns an upper bound for the ℓ-th tail of $x_{\text{low}}(c)$ and $x_{\text{hi}}(c)$. This provides us with a way to determine how many terms of our series are necessary to yield an approximate root.

Theorem 3.1. Consider the series

$$x_{\text{low}}(c) = \frac{-1}{c^{1/m}}\left[1+\sum_{k=1}^{\infty}\left(\frac{(-1)^{nk}}{km^k}\cdot\prod_{j=1}^{k-1}\frac{1+kn-jm}{j}\right)\left(\frac{1}{c^{n/m}}\right)^k\right]$$

Then for any integer $\ell \geq 2$,

$$\begin{aligned}
&\left|x_{\text{low}}(c) - \frac{-1}{c^{1/m}}\left[1+\sum_{k=1}^{\ell}\left(\frac{(-1)^{nk}}{km^k}\cdot\prod_{j=1}^{k-1}\frac{1+kn-jm}{j}\right)\left(\frac{1}{c^{n/m}}\right)^k\right]\right| \\
&= \left|\frac{-1}{c^{1/m}}\sum_{k=\ell+1}^{\infty}\left(\frac{(-1)^{nk}}{km^k}\cdot\prod_{j=1}^{k-1}\frac{1+kn-jm}{j}\right)\left(\frac{1}{c^{n/m}}\right)^k\right| \\
&\leq \frac{1}{|c|^{1/m}}\cdot\frac{\left(\frac{n}{n-m}\right)^{\frac{1+n+\ell n}{m}}(n-m)^{\ell}}{\ell\left(m\,|c|^{n/m} - n\left(\frac{n}{n-m}\right)^{\frac{n-m}{m}}\right)\left(m\,|c|^{n/m}\right)^{\ell}}
\end{aligned}$$

Proof. Note that for all integers $k \geq 1$,

$$\begin{aligned}
\prod_{j=1}^{k-1}\frac{1+kn-jm}{j} &= e^{\sum_{j=1}^{k-1}\log(1+kn-jm)-\log(j)} \\
&\leq e^{\log(1+kn-m)+\int_1^{k-1}\log(1+kn-jm)-\log(j)\ dj} \\
&= \left(\frac{1-m+kn}{1+m-km+kn}\right)^{\frac{1+kn}{m}}\left(\frac{1+m-km+kn}{k-1}\right)^{k-1} \\
&\leq \left(\frac{n}{n-m}\right)^{\frac{1+kn}{m}}(n-m)^{k-1}.
\end{aligned}$$

Then

$$\begin{aligned}
&\left|\frac{-1}{c^{1/m}}\sum_{k=\ell+1}^{\infty}\left(\frac{(-1)^{nk}}{km^k}\cdot\prod_{j=1}^{k-1}\frac{1+kn-jm}{j}\right)\left(\frac{1}{c^{n/m}}\right)^k\right|\\
&\leq\left|\frac{-1}{\ell c^{1/m}}\sum_{k=\ell+1}^{\infty}\left(\prod_{j=1}^{k-1}\frac{1+kn-jm}{j}\right)\left(\frac{(-1)^{n}}{mc^{n/m}}\right)^k\right|\\
&=\left|\frac{-1}{\ell c^{1/m}}\sum_{k=\ell+1}^{\infty}\left(\frac{n}{n-m}\right)^{\frac{1+kn}{m}}(n-m)^{k-1}\left(\frac{(-1)^{n}}{mc^{n/m}}\right)^k\right|\\
&\leq\frac{1}{|c|^{1/m}}\cdot\frac{\left(\frac{n}{n-m}\right)^{\frac{1+n+\ell n}{m}}(n-m)^{\ell}}{\ell\left(m|c|^{n/m}-n\left(\frac{n}{n-m}\right)^{\frac{n-m}{m}}\right)\left(m|c|^{n/m}\right)^{\ell}}.
\end{aligned}$$

■

Theorem 3.2. Consider the series

$$x_{\mathrm{hi}}(c)=-c^{1/(n-m)}\left[1-\sum_{k=1}^{\infty}\left(\frac{\nu_{n-m}^{-nk}}{k(n-m)^k}\cdot\prod_{j=1}^{k-1}\frac{km+j(n-m)-1}{j}\right)\left(\frac{1}{c^{n/(n-m)}}\right)^k\right]$$

Then for any integer $\ell\geq 2$,

$$\begin{aligned}
&\left|x_{\mathrm{hi}}(c)-(-1)c^{1/(n-m)}\left[1-\sum_{k=1}^{\ell}\left(\frac{(-1)^{-nk}}{k(n-m)^k}\cdot\prod_{j=1}^{k-1}\frac{km+j(n-m)-1}{j}\right)\left(\frac{1}{c^{n/(n-m)}}\right)^k\right]\right|\\
&=\left|(-1)c^{1/(n-m)}\sum_{k=\ell+1}^{\infty}\left(\frac{(-1)^{-nk}}{k(n-m)^k}\cdot\prod_{j=1}^{k-1}\frac{km+j(n-m)-1}{j}\right)\left(\frac{1}{c^{n/(n-m)}}\right)^k\right|\\
&\leq|c|^{\frac{1}{(m-n)}}\cdot\frac{n^{\ell}\left(\frac{n}{m}\right)^{\left(\frac{-1+m+\ell m}{n-m}\right)}\left(\frac{|c|^{\frac{n}{m-n}}}{n-m}\right)^{\ell}}{\ell\left(n\left(\frac{n}{m}\right)^{\frac{m}{n-m}}+|c|^{\frac{n}{n-m}}(n-m)\right)}.
\end{aligned}$$

Proof. The proof follows similarly from the proof of Theorem 3.1, except we make use of the result that for all $k\geq 1$,

$$\prod_{j=1}^{k-1}\frac{km+j(n-m)-1}{j}\leq\left(\frac{n}{m}\right)^{\frac{km-1}{n-m}}n^{k-1}$$

■

Using the preceding bounds on the ℓ-th tail of the $\mathcal{A}$-hypergeometric series, we can further understand the behavior of these $\mathcal{A}$-hypergeometric series. In the following section, we conjecture the amount of terms of these $\mathcal{A}$-hypergeometric series necessary to yield an approximate root.

3.2. **Proposed Algorithm.** In order to develop our algorithm, we assume the following conjecture holds.

Conjecture 3.3. Let $f(x_1) = c_1 + c_2 x_1^{a_2} + c_3 x_1^{a_3} \in \mathbb{Z}[x_1]$ with $d := a_3 > a_2 \geq 1$ and $|c_i| \leq H$ for all i. Let $g(x_1) = 1 + c x_1^m + x_1^n$ with $0 < m < n$ and $\gcd(m,n) = 1$ be rescaled from f. Then we need to evaluate at most $\log(dH)$ terms of an $\mathcal{A}$-hypergeometric solution and undo our rescaling to yield an approximate root of f.

We have yet to prove this conjecture; however, we suspect a proof will use Theorem 3.1 and 3.2 in conjunction with the results presented in Rojas and Ye [6]. With this assumption, we outline our algorithm.

Algorithm 3.4. (Solving Trinomial Equations Over $\mathbb{R}$)
Input. *$c_1, c_2, c_3, a_2, a_3 \in \mathbb{Z} \setminus \{0\}$ with $|c_i| \leq H$ for all i and $d := a_3 > a_2 \geq 1$.*
Output. *$z_1, \ldots, z_m \in \mathbb{Q}$ such that z_j is an approximate root of f with associated true root $\zeta_j \in \mathbb{R}$ for all j, and the ζ_j are pair-wise distinct.*
Description.
1: Rescale the coefficients and exponents of f so that it is reduced to the form $1 + cx^m + x^n$ where $c \neq 0$, $0 < m < n$ and $\gcd(m,n) = 1$.
2: Compute $r_{m,n} := \frac{n}{m^{\frac{m}{n}}(n-m)^{\frac{n-m}{n}}}$
3: Using Table 1, evaluate $\log(dH)$ terms of $x_{\mathrm{mid}}(c)$, $x_{\mathrm{low}}(c)$, $x_{\mathrm{hi}}(c)$, ${}_1\tilde{x}_{\mathrm{hi}}(c)$, ${}_2\tilde{x}_{\mathrm{hi}}(c)$, ${}_1\tilde{x}_{\mathrm{low}}(c)$, or ${}_2\tilde{x}_{\mathrm{low}}(c)$ depending on the value of m, n, $r_{m,n}$, and c.
4: Undo the rescaling to yield an approximate root of f.

3.3. **Experimental Results.** Through numerical experimentation, we have seen that Conjecture 3.3 holds true for $30,000$ distinct polynomials with $a_2 = 11$, $a_3 = 39$ and $H = 1000$. Thus in every case tested so far, Algorithm 3.4 provided an approximate root of a trinomial in time polynomial to the input size.

We outline an implementation of Algorithm 3.4 in the following example.

Example 3.5. Let $f(x_1) = 470 - 789x_1^{11} + 48x_1^{39} \in \mathbb{Z}[x_1]$. We compute three approximate roots corresponding to the three real roots of f.

1: We compute $\lambda_0 = \frac{1}{470}$ and $\lambda_1 = \left(\frac{470}{48}\right)^{1/39}$ so that

$$g(x) = \lambda_0 f(\lambda_1 x) = 1 - \frac{789}{470}\left(\frac{470}{48}\right)^{11/39} x^{11} + x^{39}$$

Since $\gcd(11, 39) = 1$, we do not need to reduce further.
2: We have that $r_{m,n} = \frac{39}{11^{\frac{11}{39}}(39-11)^{\frac{39-11}{39}}}$.

3: Since m and n are both odd, $c < 0$, $|c| > r_{m,n}$, and $\log(dH) \approx 10$, we consider the following sums

$$r_1 = \frac{-1}{c^{1/11}} \left[1 + \sum_{k=1}^{10} \left(\frac{(-1)^{39k}}{k11^k} \cdot \prod_{j=1}^{k-1} \frac{1+39k-11j}{j} \right) \left(\frac{1}{c^{39/11}} \right)^k \right] \approx 0.9011$$

$$r_2 = -|c|^{1/(39-11)} \left[1 - \sum_{k=1}^{10} \left(\frac{(-1)^{-39k}}{k(39-11)^k} \cdot \prod_{j=1}^{k-1} \frac{11k+j(39-11)-1}{j} \right) \left(\frac{1}{|c|^{39/(39-11)}} \right)^k \right]$$

$$\approx -1.0487$$

$$r_3 = |c|^{1/(39-11)} \left[1 - \sum_{k=1}^{10} \left(\frac{1}{k(39-11)^k} \cdot \prod_{j=1}^{k-1} \frac{11k+j(39-11)-1}{j} \right) \left(\frac{1}{|c|^{39/(39-11)}} \right)^k \right]$$

$$\approx 1.0332$$

4: Undoing the rescaling, we have our three rational approximate roots of f.

$$\lambda_1 z_1 = a_1/b_1 \approx 0.9554$$
$$\lambda_1 z_2 = a_2/b_2 \approx -1.1118$$
$$\lambda_1 z_3 = a_3/b_3 \approx 1.0955$$

Thus, our algorithm provided three approximate roots for a univariate trinomial in deterministic time $\log^{O(1)}(dH)$.

Acknowledgments

I thank all the mentors and participants of the Texas A&M University REU program. I particularly wish to thank Dr. Maurice Rojas, Weixun Deng, and Joshua Goldstein for their help and guidance throughout this project.

References

[1] Frédéric Bihan, J. Maurice Rojas, and Casey E. Stella. Faster real feasibility via circuit discriminants. In *Proceedings of ISSAC 2009*, 2009.

[2] Pascal Koiran. Root separation for trinomials. *J. Symbolic Comput.*, 95:151–161, 2019.

[3] Thomas Lickteig and Marie-Françoise Roy. Sylvester-Habicht sequences and fast Cauchy index computation. *J. Symbolic Comput.*, 31(3):315–341, 2001.

[4] Victor Y. Pan and Elias P. Tsigaridas. On the Boolean complexity of real root refinement. In *ISSAC 2013—Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation*, pages 299–306. ACM, New York, 2013.

[5] Mikael Passare and August Tsikh. Algebraic equations and hypergeometric series. In *The legacy of Niels Henrik Abel*, pages 653–672. Springer, Berlin, 2004.

[6] J. Maurice Rojas and Yinyu Ye. On solving univariate sparse polynomials in logarithmic time. *J. Complexity*, 21(1):87–110, 2005.

[7] J Maurice Rojas and Yuyu Zhu. A complexity chasm for solving univariate sparse polynomial equations over p-adic fields. In *Proceedings of the 2021 International Symposium on Symbolic and Algebraic Computation, ISSAC*, volume 21, 2021.

[8] Michael Sagraloff. A near-optimal algorithm for computing real roots of sparse polynomials. In *ISSAC 2014 (39th International Symposium on Symbolic and Algebraic Computation)*, pages 359–366, 2014.

[9] Steve Smale. Newton's method estimates from data at one point. In *The merging of disciplines: new directions in pure, applied, and computational mathematics (Laramie, Wyo., 1985)*, pages 185–196. Springer, New York, 1986.

[10] Yinyu Ye. Combining binary search and Newton's method to compute real roots for a class of real functions. *J. Complexity*, 10(3):271–280, 1994.

Email address: embonifa@ncsu.edu

North Carolina State University, 2108 SAS Hall, Box 8205, Raleigh, North Carolina 27695.

Email address: deng15521037237@tamu.edu

TAMU 3368, College Station, TX 77843-3368

Email address: rojas@tamu.edu

TAMU 3368, College Station, TX 77843-3368