Root-Squaring for Root-Finding

Pedro Soto
The Graduate Center, CUNY
New York, New York, USA
psoto@gradcenter.cuny.edu

Soo Go
The Graduate Center, CUNY
New York, New York, USA
sgo@gradcenter.cuny.edu

ABSTRACT

We revisit the classical root-squaring formula of Dandelin-Lobachevsky-Graeffe for polynomials and find new interesting applications to root-finding.

CCS CONCEPTS

• **Computing methodologies → Hybrid symbolic-numeric methods**.

KEYWORDS

symbolic-numeric computing, root finding, polynomial algorithms, computer algebra

1 INTRODUCTION

We revisit the famous methods simaltanously discovered by Dandelin, Lobachevsky, and Graeffe (See [12] for a history of this problem) and and make furhter progress by applying this indentity to the problem of approximating the root radius of a polynomial. This is a key step in many sub-division based root finding algorithms. One of the novelties of our algorithm is the Rational Root Tree Algorithm, *i.e.,* Alg. 1, which allows one to compute the angles of the iterated 2th roots in Eq. 3 with exact percision over the rationals; thus, greatly reducing the numerical stability Eq. 3 since, as we will see in Sec. 6 (as well as Thm. 5.1), the most numerically unstable steps in Eq. 3 is the computation of the roots $x^{\frac{1}{2^m}}$. Another advantage of our algorithm is that it assumes the blackbox polynomial model; *i.e.,,* our algorithm works under the stronger assumption that one only compute $p(x)$ via some oracle. We give both theoritical gaurentess in Sec. 6 and empirical evidence in Sec. 7 that our algorithm performs well.

2 RELATED WORKS

The main method for rootfinding by root-squaring was simaltanously discovered by Dandelin, Lobachevsky, and Graeffe in the 19th century (See [12]). Two of the first works to consider algorithms based on the formulae for modern computers were [9] and [10]; the later of which gave explicit pseudocode for the a rootfinding algorithm that uses Eq. 1. The work [15] also considers rootfinding using Eq. 1 and further state that the DLG method becomes very unstable for $\ell > 2$; by contrast, as we will see in Sec. 7, our algorithm performs reasonably well for ℓ as large as 12. The same authors went on design a variation of the DLG rootfinding algorithm that makes use of different renomalizations and other preprocessing tranformations for DLG root finding in [16], the work also proves convergence results for their DLG iterative algorithm. The authors of [3] consider DLG based algorithms for the solutions of fractional-order polynomials, *i.e.,* generalized polynomials that have rational exponents. In [20] the authors use apply the DLG formulae to root counting. The authors of [11] apply the DLG formulae to solving polynomials over finite fields. The authors of [2] apply DLG iterations to the benchmark problem (*i.e.,,* the isolsation of the roots of a polynomials, See Sec.3) to improve upon the record bound in [19] which at the time was unborken for 14 years. This short survey illustrates that the the application of the DLG to root finding (while more than a centuary old) is still an interesting research topic. One of the novelties and improvements of our algorithms is to consider the identies given by Eq. 2, instead of the usual Eq.1, to approximating the root radius. Furthermore, we show that, contrary to intuition, our algorithm is numerical stable and well behaved when taking the limit of Eq. 2 to 0.

3 BACKGROUND AND MOTIVATION

3.1 Dandelin, Lobachevsky, and Graeffe's Formaulae

The common procedure for root radii approximation, based on the classical technique of recursive root-squaring, is to first make an input polynomial $p(x)$ monic by scaling it and/or the variable x and then apply k DLG (that is, Dandelin's aka Lobachevsky's or Gräffe's) root-squaring iterations (cf. [12]),

$$p_0(x) = \frac{1}{p_d}p(x), \ p_{i+1}(x) = (-1)^d p_i(\sqrt{x})p_i(-\sqrt{x}), \ i = 0, 1, \ldots .\ell \tag{1}$$

for a fixed positive integer ℓ (see Remark 1 below). The ith iteration squares the roots of $p_i(x)$ and consequently the root radii from the origin, as well as the isolation of the unit disc $D(0, 1)$. Then one approximates the ratio ρ_+/ρ_-, the new scaled ratio of the root radii, for the polynomial $p_\ell(x)$ within a factor of γ and readily recovers the approximation of this ratio for $p_0(x)$ and $p(x)$ within a factor of $\gamma^{1/2^\ell}$.

Given the coefficients of $p_i(x)$ we can reduce the ith root-squaring iteration, that is, the computation of the coefficients of $p_{i+1}(x)$, to polynomial multiplication and perform it in $O(d \log(d))$ arithmetic operations. Unless the positive integer ℓ is small, the absolute values of the coefficients of $p_\ell(x)$ vary dramatically, and realistically one should either stop because of severe problems of numerical stability or apply the stable algorithm by Gregorio Malajovich and Jorge P. Zubelli [15], which performs a single root-squaring at arithmetic cost of order d^2.

For black box polynomials p, however, we apply DLG iterations without computing the coefficients, and the algorithm turns out to be quite efficient: for ℓ iterations evaluate $p(x)$ at 2^ℓ equally spaced points on a circle and obtain the values of the polynomial $p_\ell(x) = \prod \left(x - x_j^{2^\ell}\right)$ at these 2^ℓ points.

Furthermore, evaluate the ratio $p'(x)/p(x) = p_0'(x)/p_0(x)$ at these points by applying the recurrence

$$\frac{p_{i+1}'(x)}{p_{i+1}(x)} = \frac{1}{2\sqrt{x}}\left(\frac{p_i'(\sqrt{x})}{p_i(\sqrt{x})} - \frac{p_i'(-\sqrt{x})}{p_i(-\sqrt{x})}\right), \ i = 0, 1, \ldots \quad (2)$$

Recurrences (1) and (2) reduce evaluation of $p_\ell(c)$ to the evaluation of $p(c)$ at $q = 2^\ell$ points $c^{1/q}$ and for $c \neq 0$ reduce evaluation of the ratio $p_\ell'(x)/p_\ell(x)$ at $x = c$ to the evaluation of the ratio $p'(x)/p(x)$ at the latter $q = 2^\ell$ points $x = c^{1/q}$. We will see that we can apply recurrence (2) to support fast convergence to the convex hull of the roots.

For $x = 0$ recurrence (2) can be specialized as follows:

$$\frac{p_\ell'(0)}{p_\ell(0)} = \left(\frac{p_{\ell-1}'(x)}{p_{\ell-1}(x)}\right)'_{x=0} = \left(\frac{p'(x)}{p(x)}\right)^{(\ell)}_{x=0}, \ \ell = 1, 2, \ldots \quad (3)$$

Notice an immediate extension:

$$\frac{p_\ell^{(h)}(0)}{p_\ell(0)} = \prod_{g=1}^{h}\left(\frac{p^{(g)}(x)}{p^{(g-1)}(x)}\right)^{(\ell)}_{x=0}, \ h = 1, 2, \ldots. \quad (4)$$

Equations (3) and more generally (4) enable us to strengthen upper estimates in Eq. 5 & Eq. 6 and more generally Eq. 15 for root-radii $r_j(0, p)$ at the origin because $r_j(0, p_\ell) = r_j(0, p)^{2^\ell}$ for $j = 1, \ldots, d$ (see Eq. 5 & Eq. 6); we can approximate the higher order derivatives $\left(\frac{p^{(g)}(x)}{p^{(g-1)}(x)}\right)^{(\ell)}$ at $x = 0$ by following Remark. 2. Besides the listed applications of root-squaring, one can apply DLG to randomized exclusion tests for sparse polynomial. One can apply root-squaring $p(x) \mapsto p_\ell(x)$ to improve the error bound for the approximation of the power sums of the roots of $p(x)$ in the unit disc $D(0, 1)$ by Cauchy sums, but the improvement is about as much and at the same additional cost as by increasing the number q of points of evaluation of the ratio $\frac{p'}{p}$.

REMARK 1. *One can approximate the leading coefficient p_d of a black box polynomial $p(x)$. This coefficient is not involved in recurrence (2), and one can apply recurrence (1) by using a crude approximation to p_d and if needed can scale polynomials $p_i(x)$ for some i.*

3.2 Extremal root radii

We cover some known estimates for extremal root radii in Sec. 3.3. The most used estimates (Eq. 15 for $i = 1$ and $c = 0$) readily follow from straightforward algebraic manipulations of $p_{\text{rev}} := p_d + p_{d-1}x^1 + \ldots + p_0x^d$:

$$r_d(c, p) \leq d\left|\frac{p(c)}{p'(c)}\right| \quad (5)$$

and

$$r_1(c, p) \geq \left|\frac{p_{\text{rev}}'(c)}{dp_{\text{rev}}(c)}\right|. \quad (6)$$

We strengthen these estimates in the case of $c = 0$ by recalling DLG rootsquaring iterations Eq. 2 of Sec. 3.1 and Equation 3. This immediately implies the following extension of Eq. 5 & Eq. 6 for $c = 0$ and all non-negative integers ℓ :

$$r_d(0, p)^{2^\ell} \leq d\left|\left(\frac{p'(x)}{p(x)}\right)^{(\ell)}_{x=0}\right|^{-1} \quad (7)$$

and

$$r_1(0, p)^{2^\ell} \geq d^{-1}\left|\left(\frac{p_{\text{rev}}'(x)}{p_{\text{rev}}(x)}\right)^{(\ell)}_{x=0}\right| \quad (8)$$

REMARK 2. *Given a complex c and a positive integer ℓ one can approximate the values at $x = c$ of $(p(x)/p'(x))^{(\ell)}$ for a black box polynomial $p(x)$ by using divided differences, by extending expressions $p^{(i+1)}(c) = \lim_{x \to c} \frac{p^{(i)}(x) - p^{(i)}(c)}{x - c}$ for $i = 0, 1, \ldots, \ell - 1$ and based on the mean value theorem [7].*

For $\ell = 1$ one can instead apply the algorithm supporting the following elegant theorem and implicit in its constructive proof; the complexity of straightforward recursive extension to $\ell = 2, 3, \ldots$ increases exponentially in ℓ.

THEOREM 3.1. *Given an algorithm that evaluates a black box polynomial $p(x)$ at a point x over a field $\mathcal{K}$ of constants by using A additions and subtractions, S scalar multiplications (that is, multiplications by elements of the field $\mathcal{K}$), and M other multiplications and divisions, one can extend this algorithm to the evaluation at x of both $p(x)$ and $p'(x)$ by using $2A + M$ additions and subtractions, $2S$ scalar multiplications, and $3M$ other multiplications and divisions.*

PROOF. [14], [1] proves the theorem for any function $f(x_1, \ldots, x_s)$ that has partial derivatives in all its s variables $x_1, \ldots, x_s$. □

Next we prove that estimates 7 and 8 are extremely poor for worst case inputs.

THEOREM 3.2. *The ratios $\left|\frac{p(0)}{p'(0)}\right|$ and $\left|\frac{p_{\text{rev}}(0)}{p_{\text{rev}}'(0)}\right|$ are infinite for $p(x) = x^d - h^d$, while $r_d(c, p) = r_1(c, p) = r_d(c, p_{rev}) = r_1(c, p_{rev}) = |h|$. Proof. Observe that the roots $x_j = h\exp\left(\frac{(j-1)\mathbf{i}}{2\pi d}\right)$ of $p(x) = x^d - h^d$ for $j = 1, 2, \ldots, d$ are the d th roots of unity up to scaling by h.*

Clearly, the problem persists at the points x where $p'(x)$ and $p_{\text{rev}}'(x)$ vanish; rotation of the variable $\mathcal{R}_a : p(x) \mapsto t(x) = p(ax)$ for $|a| = 1$ does not fix it but shifts $\mathcal{T}_c : p(x) \mapsto t(x) = p(x - c)$ for $c \neq 0$ can fix it, thus enhancing the power of estimates 7 and 8. Furthermore,

$$\frac{1}{r_d(c, p)} \leq \frac{1}{d}\left|\frac{p'(c)}{p(c)}\right| = \frac{1}{d}\left|\sum_{j=1}^{d}\frac{1}{c - x_j}\right|$$

by virtue of 5 & 6, and so the approximation to the root radius $r_d(c, p)$ is poor if and only if severe cancellation occurs in the summation of the d roots, and similarly for the approximation of $r_1(c, p)$. Such cancellations only occur for a narrow class of polynomials $p(x)$, with a low probability under random root models, although it occurs for the polynomials $p(x) = x^d - h^d$ of Thm. 5 and with high probability for random polynomials of [8].

3.3 Classical estimates for extremal root radii

Next we recall some non-costly estimates known for the extremal root radii $r_1 = r_1(0, p)$ and $r_d = r_d(0, p)$ in terms of the coefficients of p (cf. [13], [17], and [21]) and the two parameters

$$\tilde{r}_- := \min_{i \geq 1} \left| \frac{p_0}{p_i} \right|^{\frac{1}{i}}, \tilde{r}_+ := \max_{i \geq 1} \left| \frac{p_{d-i}}{p_d} \right|^{\frac{1}{i}} \tag{9}$$

These bounds on r_1 and r_d hold in dual pairs since $r_1(0, p) r_d(0, p_{\text{rev}}) = 1$. Furthermore, we have that

$$\frac{1}{d} \tilde{r}_+ \leq r_1 < 2\tilde{r}_+, \frac{1}{2} \tilde{r}_- \leq r_d \leq d\tilde{r}_-, \tag{10}$$

$$\tilde{r}_+ \sqrt{\frac{2}{d}} \leq r_1 \leq \frac{1 + \sqrt{5}}{2} \tilde{r}_+ < 1.62\tilde{r}_+ \text{ if } p_{d-1} = 0, \tag{11}$$

$$0.618\tilde{r}_- < \frac{2}{1 + \sqrt{5}} \tilde{r}_- \leq r_d \leq \sqrt{\frac{d}{2}} \tilde{r}_- \text{ if } p_1 = 0, \tag{12}$$

$$r_1 \leq 1 + \sum_{i=0}^{d-1} \left| \frac{p_i}{p_d} \right|, \frac{1}{r_d} \leq 1 + \sum_{i=1}^{d} \left| \frac{p_i}{p_0} \right|. \tag{13}$$

$M(p) := |p_d| \max_{j=1}^{d} \{1, |x_j|\}$ is said to be the Mahler measure of p, and so $M(p_{\text{rev}}) := |p_0| \max_{j=1}^{d} \{1, \frac{1}{|x_j|}\}$. It holds that

$$r_1^2 \leq \frac{M(p)^2}{|p_d|} \leq \max_{i=0}^{d-1} \left| \frac{p_i}{p_d} \right|^2, \frac{1}{r_d^2} \leq \frac{M(p_{\text{rev}})^2}{|p_0|^2} \leq \max_{i=1}^{d} \left| \frac{p_i}{p_0} \right|^2 \tag{14}$$

Theorem 167 supports very fast approximation of all root radii of p at the origin at a very low cost, which complements estimates 9, 10 11, 12, 13, and 14.

One can extend all these bounds to the estimates for the root radii $r_j(c, p)$ for any fixed complex c and all j by observing that $r_j(c, p) = r_j(0, t)$ for the polynomial $t(x) = p(x - c)$ and applying Taylor's shift; i.e., applying the mapping $\mathcal{S}_{c,\rho} : p(x) \mapsto p\left(\frac{x-c}{\rho}\right)$.

The algorithms in Sec. 5 closely approximate root radii $r_j(c, p)$ for a black box polynomial p and a complex point c at reasonably low cost, but the next well-known upper bounds on r_d and lower bounds on r_1 (cf. [13],[6],[18],[4], and [5]) are computed at even a lower cost, defined by a single fraction $\frac{p_0}{p_i}$ or $\frac{p_{d-i}}{p_d}$ for any i, albeit these bounds are excessively large for the worst case input.

Finally, we have that $r_d \leq \rho_{i,-} := \left(\binom{d}{i} \left| \frac{p_0}{p_i} \right| \right)^{\frac{1}{i}}, \frac{1}{r_1} \leq \frac{1}{\rho_{i,+}} :=$

$\left(\binom{d}{i} \left| \frac{p_d}{p_{d-i}} \right| \right)^{\frac{1}{i}}$ and therefore, since $p^{(i)}(0) = i! p_i$ for all $i > 0$, that

$$r_d \leq \rho_{i,-} = \left(i! \binom{d}{i} \left| \frac{p(0)}{p^{(i)}(0)} \right| \right)^{\frac{1}{i}}, \frac{1}{r_1} \leq \frac{1}{\rho_{i,+}} = \left(i! \binom{d}{i} \left| \frac{p_{\text{rev}}(0)}{p_{\text{rev}}^{(i)}(0)} \right| \right)^{\frac{1}{i}} \tag{15}$$

for all i; from which we obtain relations 5 and 6 for $i = 1$.
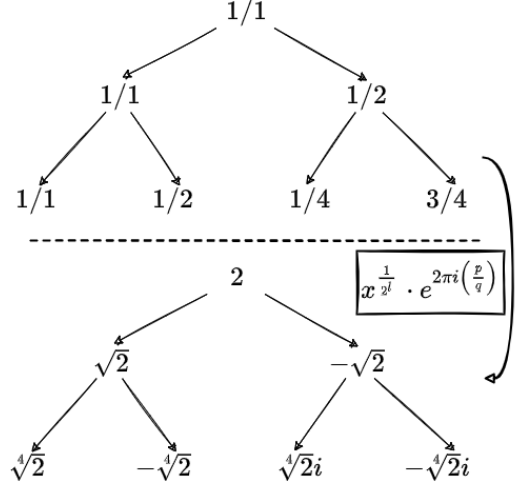
4 MOTIVATING EXAMPLE

Figure 1: The upper tree depicts the steps of Cir-cle_Roots_Rational_Form(p, q, l) in Alg.1 for $l = 2$, $p = 1$, and $q = 1$. The lower tree depicts the steps of Roots(r, t, u, l) in Alg.2 for $r = 2$, $l = 2$, $p = 1$, and $q = 1$
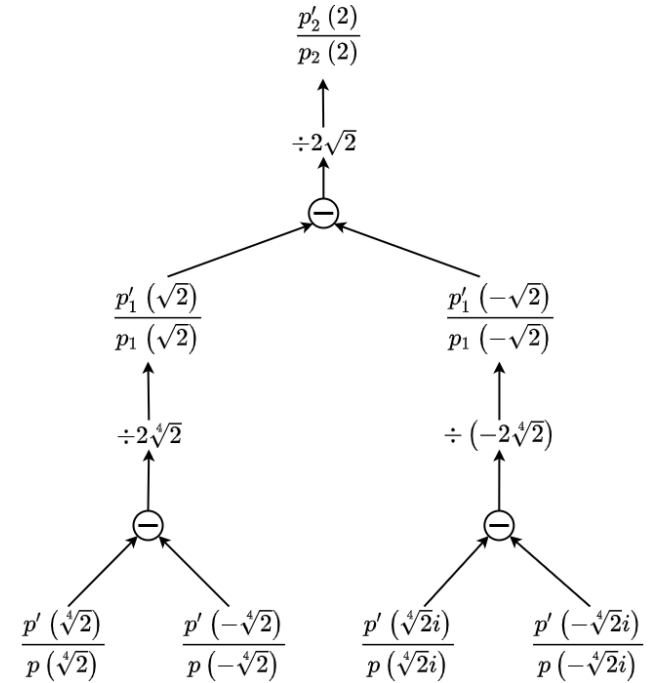

Figure 2: The steps of DLG_Rational_Rorm(p, p', r, t, u, l) in Alg.3 for $r = 2$, $l = 2$, $t = 1$, and $u = 1$.

5 ALGORITHM DESIGN

In this section give the design of the general algorithm for approximating the root radius given by Eq. 3. There are two main steps: 1) going down the rational root tree, *i.e.*, performing Alg. 1, and 2) going up the rational root tree, *i.e.*, performing Alg. 3. The going-down is depicted in Fig. 1 and the going-up is depicted in Fig. 2. The other procedures are simple bookkeeping/preprocessing steps in between the two main going-down/going-up steps. In particular we 1) first convert a complex number x into polar coordinates r, θ where $\theta \approx \frac{p}{q} = \frac{p}{2^\epsilon}$, *i.e.*, perform Alg. 4, 2) we then perform the first going down pass which gives the rational angles for the roots of x in fraction form, *i.e.*, perform Alg. 1, 3) we then perform the second pass of the going down algorithm where we compute the values $|x|^{\frac{1}{2m}} \exp(2\pi i \frac{p}{q})$ at the mth level, and 4) finally, we then compute the values given by Eq. 2 going back up the rational root tree.

Algorithm 1 Circle_Roots_Rational_Form(p, q, l)

---

  if $p\%q == 0$ **then**
    $r, s := (1,1)$
  else
    $r, s := (p, 2q)$
  end if
  if $r\%s == 0$ **then**
    $t, u := (1,2)$
  else
    $t, u := (2r + s, 2s)$
  end if
  if $l == 1$ **then**
    return $[(r, s), (t, u)]$
  else if $l\ != 0$ **then**
    left := Circle_Roots_Rational_Form$(r, s, l - 1)$
    right := Circle_Roots_Rational_Form$(t, u, l - 1)$
    return left $\cup$ right
  else
    return $[(p, q)]$
  end if

The intuition behind Algorithm. 1 is that the square root operation satisfies

$$p\%q \neq 0 \implies \sqrt{\exp\left(2\pi i \frac{p}{q}\right)} = \exp\left(2\pi i \frac{p}{2q}\right) \quad (16)$$

and

$$p\%q = 0 \implies \sqrt{\exp\left(2\pi i \frac{1}{1}\right)} = \exp\left(2\pi i \frac{1}{1}\right) = 1, \quad (17)$$

and the negation operation satisfies

$$p\%q \neq 0 \implies -\exp\left(2\pi i \frac{r}{s}\right) = \exp\left(2\pi i \frac{2r + s}{2s}\right) \quad (18)$$

and

$$p\%q = 0 \implies \sqrt{\exp\left(2\pi i \frac{1}{1}\right)} = \exp\left(2\pi i \frac{1}{2}\right) = -1; \quad (19)$$

thus, the first four lines of Algorithm. 1 computes the (angle of the) positive square root, $\sqrt{x}$, of complex number on the unit circle

and the next four lines Alg. 1 computes the (angle of the) negative square root, $-\sqrt{x}$. Therefore, we have:

Theorem 5.1. *For a complex number, x, with a rational angle,* i.e., $x = |x| \exp\left(2\pi i \frac{p}{q}\right)$, *Alg. 1 correctly computes the roots in Eq. 2.*

Proof. Equation 16, 17, 18, and 19 give the base case and the theorem follows by a straightforward induction. □

Algorithm 2 Roots(r, t, u, l)

---

  root_tree = Circle_Roots_Rational_Form(p, q, l)
  circ_root = $[\exp\left(2 \cdot \pi \cdot i \cdot \frac{r}{s}\right)$ for r, s in root_tree$]$
  roots = $[\sqrt[2^l]{r} \cdot$ root for root in circ_root$]$
  return roots

Algorithm 3 DLG_Rational_Form(p, p', r, t, u, l)

---

  root := Roots(r, t, u, l)
  for $r_i \in$ root **do**
    base_step$[i] := \frac{p'(r_i)}{p(r_i)}$
  end for
  diff$[0]$:= base_step
  for $i \leq l$ **do**
    for $j \leq 2^{l-i-1}$ **do**
      diff$[i + 1][j] := \frac{1}{2} \frac{\text{diff}[i][2j] - \text{diff}[i][2j+1]}{\text{root}[2j]}$
      root = roots$(r, t, u, l - 1 - i)$
    end for
  end for
  return diff$[l][0]$

Algorithm 4 DLG(p, p', l, x, ϵ)

---

  angle := $\frac{1}{2\pi i} \log(x)$
  $u := 2^\epsilon$
  $t := ($angle$\cdot u)\%1$
  $r := |x|$
  return DLG_Rational_Form(p, p', r, t, u, l)

---

6 THEORETICAL ANALYSIS

Theorem 6.1. *Algorithm. 3 performs $q \log q$ floating point subtractions, divisions, and multiplications and $q \log q$ applications of sin and cos, where $q = 2^l$; furthermore, Algorithm. 3 performs at most $Cq \log q$ integer additions, "multiplications-by-2", and $\%2^\epsilon$ (i.e., mod 2^ϵ) operations, where $C = 1, 3, 2$ respectively.*

Theorem 6.2. *The $q \log q$ integer additions, "multiplications-by-2", and $\%2^\epsilon$ (i.e., mod 2^ϵ) operations in Algorithm. 3 have negligble overhead; more precisely, integer addtions are always additions of $2\ \epsilon$-bit integers and "multiplications-by-2" and $\%2^\epsilon$ (i.e., mod 2^ϵ) operations have constant time overhead.*

Table 1: Experimental Data for Chebyshev.

DEGREE	ℓ	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
20	4	616	332	0.15	0.09	0.22	$[0.0785, 0.997]$
40	5	617	332	0.09	0.05	1.54	$[0.0393, 0.999]$
80	6	617	334	0.05	0.03	5.33	$[0.0196, 1.0]$
160	7	617	334	2.71	0.02	23.54	$[0.00982, 1.0]$
320	8	617	334	37.64	0.46	111.70	$[0.00491, 1.0]$

Table 2: Experimental Data for Chromatic polynomial (zero root removed) kind a

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
22	4	616	332	0.21	0.13	0.24	$[1.0, 3.17]$
86	6	617	334	0.03	0.04	5.83	$[1.0, 3.25]$
342	8	617	334	0.71	0.54	107.09	$[0.884, 3.41]$

Table 3: Experimental Data for Chromatic polynomial deflated: a

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
20	4	616	332	0.15	0.17	0.23	$[1.3, 3.01]$
84	6	617	334	0.05	0.00	10.33	$[1.1, 3.06]$
340	8	617	334	0.65	0.69	96.91	$[0.741, 3.11]$

Table 4: Experimental Data for Chromatic polynomial (zero root removed) kind c

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
23	4	616	332	0.21	0.13	0.27	$[1.0, 3.03]$
343	8	617	334	0.71	0.27	103.02	$[0.897, 4.13]$

7 EXPERIMENTAL RESULTS

8 CONCLUSION

REFERENCES

[1] Walter Baur and Volker Strassen. 1983. The complexity of partial derivatives. *Theoretical computer science* 22, 3 (1983), 317–330.

[2] Ruben Becker, Michael Sagraloff, Vikram Sharma, and Chee-Keng Yap. 2018. A near-optimal subdivision algorithm for complex root isolation based on the Pellet test and Newton iteration. *J. Symb. Comput.* 86 (2018), 51–96.

[3] Stanislaw Bialas and Henryk Górecki. 2010. Generalization of Vieta's formulae to the fractional polynomials, and generalizations the method of Graeffe-Lobachevsky. *Bulletin of The Polish Academy of Sciences-technical Sciences* 58 (2010), 624–629.

[4] Dario Andrea Bini and Giuseppe Fiorentino. 2000. Design, analysis, and implementation of a multiprecision polynomial rootfinder. *Numerical Algorithms* 23, 2 (2000), 127–173.

[5] Dario A Bini and Leonardo Robol. 2014. Solving secular and polynomial equations: A multiprecision algorithm. *J. Comput. Appl. Math.* 272 (2014), 276–292.

[6] Carsten Carstensen. 1991. Inclusion of the roots of a polynomial based on Gerschgorin's theorem. *Numer. Math.* 59, 1 (1991), 349–360.

[7] Carl de Boor. 1995. Divided Differences.

[8] Paul Erdos and Pál Turán. 1950. On the distribution of roots of polynomials. *Annals of mathematics* (1950), 105–119.

[9] A. A. Grau. 1963. On the Reduction of Number Range in the Use of the Graeffe Process. *J. ACM* 10, 4 (oct 1963), 538–544. https://doi.org/10.1145/321186.321198

[10] A. A. Grau. 1965. Algorithm 256: Modified Graeffe Method [C2]. *Commun. ACM* 8, 6 (jun 1965), 379–380. https://doi.org/10.1145/364955.364974

[11] Bruno Grenet, Joris van der Hoeven, and Grégoire Lecerf. 2015. Deterministic root finding over finite fields using Graeffe transforms. *Applicable Algebra in Engineering, Communication and Computing* 27 (2015), 237–257.

[12] Alston S. Householder. 1959. Dandelin, Lobacevskii, or Graeffe. *The American Mathematical Monthly* 66, 6 (1959), 464–466. http://www.jstor.org/stable/2310626

[13] Movlud Kerimovich Kerimov. 1977. Applied and computational complex analysis. Vol. 1. Power series, integration, conformal mapping, location of zeros: Henrici P. xv+ 682 pp., John Wiley and Sons, Inc., New York–London, 1974. Book review. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki* 17, 5 (1977), 1330–1331.

[14] Seppo Linnainmaa. 1976. Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics 1976 16:2* 16 (1976), 146–160. Issue 2. https://doi.org/10.1007/BF01931367

[15] Gregorio Malajovich and Jorge P. Zubelli. 2001. On the Geometry of Graeffe Iteration. *J. Complex.* 17 (2001), 541–573.

[16] Gregorio Malajovich and Jorge P. Zubelli. 2001. Tangent Graeffe iteration. *Numer. Math.* 89 (2001), 749–782.

[17] Maurice Mignotte and Doru Stefanescu. 1999. *Polynomials: An algorithmic approach.* Springer.

[18] Victor Y Pan. 2000. Approximating complex polynomial zeros: modified Weyl's quadtree construction and improved Newton's iteration. *journal of complexity* 16, 1 (2000), 213–264.

[19] Victor Y Pan. 2002. Univariate polynomials: nearly optimal algorithms for numerical factorization and root-finding. *Journal of Symbolic Computation* 33, 5

Table 5: Experimental Data for Chromatic polynomial deflated: d

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
11	3	616	332	0.25	0.24	0.06	$[1.27, 2.8]$
43	5	616	332	0.10	0.09	2.55	$[1.02, 2.97]$
171	7	617	334	0.26	1.13	26.25	$[0.715, 3.07]$
683	9	619	338	0.16	0.25	410.18	$[0.519, 3.1]$

Table 6: Experimental Data for Curz

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
20	4	616	332	0.15	0.85	0.24	$[0.452, 1.15]$
40	5	616	332	0.19	0.77	1.40	$[0.379, 1.26]$
80	6	617	334	0.08	0.22	6.12	$[0.318, 1.34]$
160	7	617	334	0.03	0.50	23.06	$[0.271, 1.38]$

Table 7: Experimental Data for Easy

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
100	6	617	334	0.12	0.05	21.45	$[0.949, 0.98]$
200	7	617	334	0.06	0.03	63.90	$[0.971, 0.99]$
400	8	617	334	0.03	0.01	155.17	$[0.983, 0.995]$
800	9	619	338	0.02	0.01	529.08	$[0.991, 0.991]$
1600	10	619	338	0.01	0.00	2228.22	$[0.995, 0.999]$
3200	11	619	338	0.00	?	?	$[0.997, 0.999]$

Table 8: Experimental Data for Exp

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
50	5	616	332	0.14	0.94	1.62	$[14.9, 39.4]$
100	6	617	334	0.36	0.05	6.60	$[28.9, 83.9]$
200	7	617	334	0.71	0.83	26.70	$[56.8, 176.0]$
400	8	617	334	0.96	0.98	107.56	$[113.0, 365.0]$

(2002), 701–733.

[20] Joris van der Hoeven. 2011. Efficient root counting for analytic functions on a disk.

[21] Chee-Keng Yap et al. 2000. *Fundamental problems of algorithmic algebra*. Vol. 49. Oxford University Press Oxford.

Table 9: Experimental Data for Geom1

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
10	3	616	332	0.33	0.25	0.05	$[1.0, 10^{18}]$
15	3	616	332	0.40	0.99	0.7	$[1.0, 10^{28}]$
20	4	616	332	0.20	0.99	0.23	$[1.0, 10^{38}]$
40	5	616	332	0.12	0.99	1.16	$[1.0, 10^{78}]$

Table 10: Experimental Data for Geom2

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
10	3	616	332	0.33	0.25	0.6	$[10^{-18}, 1.0]$
15	3	616	332	80920.06	0.28	0.08	$[10^{-28}, 1.0]$
20	4	616	332	2.63×10^{26}	0.17	0.23	$[10^{-38}1.0]$
40	5	616	332	1.61×10^{72}	0.10	1.13	$[10^{-78}, 1.0]$

Table 11: Experimental Data for Geom3

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
10	3	616	332	0.33	0.25	0.05	$[9.54 \times 10^{-7}, 0.25]$
20	4	616	332	2.41	0.17	0.26	$[9.09 \times 10^{-13}, 0.25]$
40	5	616	332	1.95×10^{18}	0.10	1.17	$[8.27 \times 10^{-25}, 0.25]$
80	6	617	334	1.83×10^{45}	0.06	5.21	$[6.84 \times 10^{-49}, 0.25]$

Table 12: Experimental Data for Geom4

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
10	3	616	332	0.33	0.25	0.04	$[4.0, 1.05 \times 10^6]$
20	4	616	332	0.20	0.70	0.22	$[4.0, 1.05 \times 10^12]$
40	5	616	332	0.12	0.99	1.17	$[4.0, 1.05 \times 10^24]$
80	6	617	334	0.07	0.99	5.11	$[4.0, 1.05 \times 10^48]$

Table 13: Experimental Data for Hermite

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
20	4	616	332	0.15	0.12	0.23	$[0.245, 5.39]$
40	5	616	332	0.09	0.08	1.13	$[0.175, 8.1]$
80	6	617	334	0.05	0.05	5.01	$[0.124, 11.9]$
160	7	617	334	0.03	0.03	20.61	$[0.0877, 17.2]$
320	8	617	334	2.08	0,78	86.41	$[0.062, 24.7]$

Table 14: Experimental Data for Kam1

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
7	2	615	331	0.36	0.99	0.01	$[3.0 \times 10^{-12}, 15.8]$
7	2	615	331	0.36	0.99	0.01	$[3.0 \times 10^{-40}, 10^4]$
7	2	615	331	0.10	0.99	0.01	$[3.0 \times 10^{-140}, 10^{14}]$

Table 15: Experimental Data for Kam2

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
9	3	616	332	0.10	0.99	0.04	$[1.73 \times 10^{-6}, 251.0]$
9	3	616	332	0.10	0.99	0.03	$[1.73 \times 10^{-70}, 10^8]$
9	3	616	332	4.23×10^{46}	0.99	0.03	$[1.73 \times 10^{-20}, 10^{28}]$

Table 16: Experimental Data for Kam3

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
9	3	616	332	0.10	0.99	0.03	$[1.73 \times 10^{-6}, 251.0]$
9	3	616	332	0.10	0.99	0.03	$[1.73 \times 10^{-20}, 10^8]$
9	3	616	332	4.23×10^{46}	0.99	0.03	$[1.73 \times 10^{-70}, 10^{28}]$

Table 17: Experimental Data for Kir1

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
10	5	616	332	0	0	1.36	$[0.5, 0.5]$
20	6	617	334	0	0	2.63	$[0.5, 0.5]$
40	7	617	334	0	0	10.73	$[0.5, 0.5]$

Table 18: Experimental Data for Kir1 mod

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
10	5	616	332	0	0	1.44	$[0.5, 0.5]$
20	6	617	334	0	0	2.84	$[0.498, 0.502]$
40	7	617	334	0	0	9.84	$[0.496, 0.504]$

Table 19: Experimental Data for Lagurerre

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
20	4	616	332	0.20	0.16	0.21	$[0.0705, 66.5]$
40	5	616	332	0.12	0.10	1.16	$[0.0357, 142.0]$
80	6	617	334	0.07	0.06	5.17	$[0.018, 297.0]$
160	7	617	334	3.08	0.95	20.24	$[0.00901, 610.0]$
320	8	617	334	41.38	0.99	85.15	$[0.00451, 1.24 \times 10^3]$

Table 20: Experimental Data for Lar

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
20	4	616	332	7.05×10^9	0.99	0.19	$[3.73 \times 10^{-22}, 10^{50}]$
200	7	617	332	9.69×10^{19}	0.31	0.92	$[3.73 \times 10^{-22}, 41.0]$
20	4	616	334	-	-	-	$[0.0, 6.16e+15]$
20	4	616	332	-	-	-	$[1.0, +inf]$
23	4	616	332	-	-	-	$[6.16 \times 10^{-105}, +inf]$
20	4	616	332	-	-	-	$[1.0, +inf]$

Table 21: Experimental Data for Legendre

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
20	4	616	332	0.15	0.09	0.22	$[0.0765, 0.993]$
40	5	616	332	0.09	0.06	1.16	$[0.0388, 0.998]$
80	6	617	334	0.05	0.03	5.17	$[0.0195, 0.998]$
160	7	617	334	2.72	0.06	20.99	$[0.00979, 1.0]$
320	8	617	334	37.70	0.45	83.82	$[0.0049, 1.0]$

Table 22: Experimental Data for LSR

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
500	8	617	334	1918.33	0.99	5.67	$[0.0001, 2.0 \times 10^4]$
500	8	617	334	5766.11	0.99	2.94	$[3.33 \times 10^{-5}, 10^3]$
500	8	617	334	1.05	0.99	2.90	$[0.0916, 10^{200}]$
52	5	616	332	1.81×10^{14}	0.99	0.25	$[10^{-20}, \times 10^{10}]$
52	5	616	332	1.81×10^{34}	0.99	0.20	$[10^{-40}, 10^{20}]$
52	5	616	332	1.81×10^{74}	0.99	0.22	$[10^{-80}, 10^{40}]$
200	7	617	334	3.62×10^{18}	0.99	9.49	$[10^{-20}, 10^{20}]$
24	4	616	332	2.88×10^8	0.99	0.37	$[10^{-20}, 10^{20}]$

Table 23: Experimental Data for Mandelbrot

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
31	4	616	332	0.29	0.14	0.57	$[0.445, 2.0]$
63	5	616	332	0.12	0.08	2.01	$[0.403, 2.0]$
127	6	617	334	0.08	0.04	8.95	$[0.373, 2.0]$
255	7	617	334	0.06	0.70	45.13	$[0.351, 2.0]$
511	8	617	334	0.02	1.62	143.01	$[0.334, 2.0]$
1023	9	619	338	0.35	0.14	546.54	$[0.321, 2.0]$
2047	10	619	338	1.11	0.24	2278.95	$[0.311, 2.0]$
4095	11	619	338	1.68	-	-	$[0.303, 2.0]$

Table 24: Experimental Data for Mig1

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
20	4	616	332	0.12	0.99	0.08	$[0.01, 2.26]$
50	5	616	332	0.01	0.98	1.75	$[0.00999, 1.83 \times 10^3]$
100	6	617	334	0.05	0.99	0.41	$[0.01, 1.15]$
100	6	617	334	0.01	0.99	3.78	$[0.01, 7.92]$
200	7	617	334	2.65	0.99	0.84	$[0.01, 1.07]$
200	7	617	334	2.59	0.99	7.55	$[0.01, 2.33]$
500	8	617	334	18.19	0.98	1.60	$[0.01, 1.03]$
500	8	617	334	18.01	0.98	16.16	$[0.01, 1.36]$

Table 25: Experimental Data for Mult

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
15	3	616	332	0.29	0.18	0.08	$[0.869, 1.07]$
68	6	617	334	0.05	0.05	4.41	$[0.25, 2.24]$
22	4	616	332	0.21	0.10	0.24	$[1.0, 20.0]$
20	4	616	332	0.07	0.47	0.39	$[0.01, 2.68]$

Table 26: Experimental Data for $x^n - 1$

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
50	5	616	332	4.92×10^{13}	0.99	0.09	$[1.0, 1.0]$
100	6	617	334	7.06×10^6	0.99	0.21	$[1.0, 1.0]$
200	7	617	334	2.69×10^3	0.99	0.40	$[1.0, 1.0]$
400	8	617	334	50.50	0.98	0.81	$[1.0, 1.0]$
800	9	619	338	6.33	0.86	1.67	$[1.0, 1.0]$
1600	10	619	338	1.70	0.63	3.29	$[1.0, 1.0]$
3200	11	619	338	0.64	0.39	6.20	$[1.0, 1.0]$
6400	12	623	346	0.28	0.22	13.22	$[1.0, 1.0]$

Table 27: Experimental Data for $x^n - i$

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
50	5	616	332	5.03×10^{13}	0.99	0.11	$[1.0, 1.0]$
100	6	617	334	7.06×10^6	0.99	0.23	$[1.0, 1.0]$
200	7	617	334	2.69×10^3	0.99	0.43	$[1.0, 1.0]$
400	8	617	334	50.50	0.98	0.82	$[1.0, 1.0]$
800	9	619	338	6.33	0.86	1.82	$[1.0, 1.0]$
1600	10	619	338	1.70	0.63	3.53	$[1.0, 1.0]$
3200	11	619	338	0.64	0.39	6.79	$[1.0, 1.0]$
6400	12	623	346	0.28	0.22	13.82	$[1.0, 1.0]$

Table 28: Experimental Data for Sendra

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
20	4	616	332	0.917	0.158	0.26	$[0.9, 2.05]$
40	5	616	332	0.916	0.101	1.21	$[0.95, 2.02]$
80	6	617	334	2.25	0.573	5.18	$[0.975, 2.01]$
160	7	617	334	5.39	2.14	21.34	$[0.987, 2.01]$
320	8	617	334	4.32	1.64	88.01	$[0.994, 2.0]$

Table 29: Experimental Data for Sparse

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
100	6	617	334	1.0	1.0	0.28	$[0.968, 1.01]$
200	7	617	334	0.995	0.995	0.79	$[0.969, 1.0]$
400	8	617	334	0.927	0.929	2.11	$[0.969, 1.0]$
800	9	619	338	0.73	0.739	5.26	$[0.969, 1.0]$
6400	12	623	346	0.13	0.158	66.58	$[0.969, 1.0]$

Table 30: Experimental Data for Spiral

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
10	3	616	332	2.1E-7	1.21E-6	0.05	$[1.0, 1.0]$
15	3	616	332	1.49E-7	1.15E-6	0.1	$[1.0, 1.0]$
20	4	616	332	3.05E-7	1.3E-6	0.22	$[1.0, 1.0]$
25	4	616	332	2.47E-7	1.25E-6	0.35	$[1.0, 1.0]$
30	4	616	332	2.08E-7	1.21E-6	0.35	$[1.0, 1.0]$

Table 31: Experimental Data for Toeplitz

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
128	7	617	334	20.6	0.562	17.45	$[1.31, 64.4]$
128	7	617	334	31.2	0.0272	17.6	$[0.4, 13.2]$
256	8	617	334	2.97	0.918	72.2	$[1.34, 64.4]$
256	8	617	334	12.8	0.599	69.02	$[0.383, 13.2]$

Table 32: Experimental Data for Wilk

DEGREE	l	$e = -\log(|x|)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
20	4	616	332	16.2	0.141	0.35	$[1.0, 20.0]$
30	4	616	332	4.59	0.982	0.39	$[1.0, 319.0]$
40	5	616	332	35.3	0.0927	1.23	$[1.0, 40.0]$
80	6	617	334	88.7	0.121	5.18	$[1.0, 80.0]$
160	7	617	334	27.2	0.824	31.48	$[1.0, 160.0]$
320	8	617	334	4.31	0.983	93.81	$[1.0, 320.0]$

Table 33: Experimental Data for

DEGREE	l	$e = -\log(|x|)$	MPMATH Precision	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
20							[,]
40							[,]
80							[,]
160							[,]
320							[,]