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

ACM Reference Format:
Pedro Soto and Soo Go. 2022. Root-Squaring for Root-Finding. In *Proceedings of International Symposium on Symbolic and Algebraic Computation (ISSAC '22).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

We revisit the famous methods simaltanously discovered by Dandelin, Lobachevsky, and Graeffe (See [1] 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.

2 RELATED WORKS

3 BACKGROUND AND MOTIVATION

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. [1]),

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

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 [2], 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

$$\frac{p_{i+1}'(x)}{p_{i+1}(x)} = \frac{1}{2\sqrt{x}}\Big(\frac{p_i'(\sqrt{x}\,)}{p_i(\sqrt{x}\,)} - \frac{p_i'(-\sqrt{x}\,)}{p_i(-\sqrt{x}\,)}\Big),\ i = 0, 1, \dots \tag{2}$$

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:

$$\frac{p_\ell'(0)}{p_\ell(0)} = \Big(\frac{p_{\ell-1}'(x)}{p_{\ell-1}(x)}\Big)'_{x=0} = \Big(\frac{p'(x)}{p(x)}\Big)^{(\ell)}_{x=0},\ \ell = 1, 2, \dots \tag{3}$$

Notice an immediate extension:

$$\frac{p_\ell^{(h)}(0)}{p_\ell(0)} = \prod_{g=1}^{h}\Big(\frac{p^{(g)}(x)}{p^{(g-1)}(x)}\Big)^{(\ell)}_{x=0},\ h = 1, 2, \dots . \tag{4}$$

Equations (3) and more generally (4) enable us to strengthen upper estimates in Eq. 5 & Eq. 6 and more generally Eq. 9 for root-radii $r_j(0, p)$ at the origin because $r_j(0, p_\ell) = r_j(0, p)^{2^\ell}$ for $j = 1, \dots, d$ (see Eq. 5 & Eq. 6); we can approximate the higher order derivatives $\Big(\frac{p^{(g)}(x)}{p^{(g-1)}(x)}\Big)^{(\ell)}$ 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 $\frac{p'}{p}$.

ISSAC '22, July 04–07, 2022, Lille, France

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

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.*

3.1 Extremal root radii

We cover some known estimates for extremal root radii in Sec. 3.2. The most used estimates (Eq. 9 for $i = 1$ and $c = 0$) readily follow from straightforward algbraic manipluations of $p_{\rm rev} := p_d + p_{d-1}x^1 + ... + p_0x^d$:

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

and

$$r_1(c,p) \ge \left|\frac{p'_{\rm rev}(c)}{dp_{\rm rev}(c)}\right|. \tag{6}$$

We strengthen these estimates in the case of $c = 0$ by recalling DLG rootsquaring iterations (??) of Sec. ?? and equation (??). This immediately implies the following extension of (11) for $c = 0$ and all non-negative integers ℓ :

$$r_d(0,p)^{2^\ell} \le d/\left|\left(\frac{p'(x)}{p(x)}\right)^{(\ell)}_{x=0}\right| \tag{7}$$

and

$$r_1(0,p)^{2^\ell} \ge 1/\left|\left(\frac{dp_{\rm rev}\,(x)}{p'_{\rm rev}\,(x)}\right)^{(\ell)}_{x=0}\right| \tag{8}$$

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 (cf. Alg. 117), by extending expressions $p^{(i+1)}(c) = \lim_{x\to c}\frac{p^{(i)}(x)-p^{(i)}(c)}{x-c}$ for $i = 0, 1, \dots, \ell-1$ and based on the mean value theorem* [??].

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 $\ell = 2, 3, \dots$ 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.*

60. , [8, Thm. 2] prove the theorem for any function $f(x_1, \dots, x_s)$ that has partial derivative in all its s variables $x_1, \dots, x_s$. □

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

Theorem 3.2. *The ratios $\left|\frac{p(0)}{p'(0)}\right|$ and $\left|\frac{p_{\rm rev}(0)}{p'_{\rm rev}(0)}\right|$ 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 $j = 1, 2, \dots, d$ are the d th roots of unity up to scaling by h.*

Clearly, the problem persists at the points x where $p'(x)$ and $p'_{\rm rev}\,(x)$ vanish; rotation of the variable $p(x) \leftarrow t(x) = p(ax)$ for $|a| = 1$ does not fix it but shifts $p(x) \leftarrow t(x) = p(x - c)$ for $c \ne 0$ can fix it, thus enhancing the power of estimates (??) and (??). Furthermore,

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

by virtue of (??), 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 a cancellation only occurs 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 whp for random polynomials of [??].

3.2 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. [42, Sec. 6.4], [69], and [117]) and the two parameters

$$\tilde r_- := \min_{i\ge1}\left|\frac{p_0}{p_i}\right|^{\frac{1}{i}}, \tilde r_+ := \max_{i\ge1}\left|\frac{p_{d-i}}{p_d}\right|^{\frac{1}{i}}$$

These bounds on r_1 and r_d hold in dual pairs since $r_1(0,p)r_d\,(0, p_{\rm rev}\,) = 1$ (see equation (8) for $j = 1$).

$$\frac{1}{d}\tilde r_+ \le r_1 < 2\tilde r_+, \frac{1}{2}\tilde r_- \le r_d \le d\tilde r_-,$$

$$\tilde r_+\sqrt{\frac{2}{d}} \le r_1 \le \frac{1+\sqrt5}{2}\tilde r_+ < 1.62\tilde r_+ \text{if } p_{d-1} = 0,$$

$$0.618\tilde r_- < \frac{2}{1+\sqrt5}\tilde r_- \le r_d \le \sqrt{\frac{d}{2}}\tilde r_- \text{if } p_1 = 0,$$

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

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

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

Theorem 167 supports very fast approximation of all root radii of p at the origin at a very low cost, which complements estimates (170) – (175).

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 (cf. Sec. 2.5).

Our algorithms of Secs. 6.7 and 6.2 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. [42, Cor. 6.4 g], [22], [79, Remark 6.4], [15, Thms. 8, 9, 10, 13, and 14], and [19, Secs. 3.1 and 3.2]) 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. $r_d \leq \rho_{i,-} := \left(\left(\begin{array}{c} d \\ i \end{array}\right)\left|\frac{p_0}{p_i}\right|\right)^{\frac{1}{i}}, \frac{1}{r_1} \leq \frac{1}{\rho_{i,+}} := \left(\left(\begin{array}{c} d \\ i \end{array}\right)\left|\frac{p_d}{p_{d-i}}\right|\right)^{\frac{1}{i}}$ for all i

$$r_d \leq \rho_{i,-} = \left(i!\left(\begin{array}{c} d \\ i \end{array}\right)\left|\frac{p(0)}{p^{(i)}(0)}\right|\right)^{\frac{1}{i}}, \frac{1}{r_1} \leq \frac{1}{\rho_{i,+}} = \left(i!\left(\begin{array}{c} d \\ i \end{array}\right)\left|\frac{p_{\text{rev}}(0)}{p_{\text{rev}}^{(i)}(0)}\right|\right)^{\frac{1}{i}} \tag{9}$$

for all i. For $i = 1$ obtain (11).

4 MOTIVATING EXAMPLE

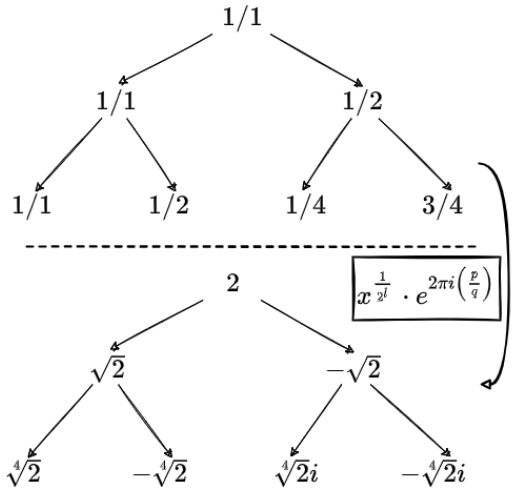

Figure 1: The upper tree depicts the steps of CIRCLE_ROOTS_RATIONAL_FORM(p, q, l) in Alg.1 for $l = 2$, $p = 1$, and $q = 1$. The lower tree depicts the steps of ROOTS(r, t, u, l) in Alg.2 for $r = 2$, $l = 2$, $p = 1$, and $q = 1$

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 $|x|^{\frac{1}{2^m}} \exp(2\pi i \frac{p}{q})$ at the m^{th} level, and 4) finally, we then compute the values given by Eq. 2 going back up the rational root tree.

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

$$p\%q \neq 0 \implies \sqrt{\exp\left(2\pi i \frac{p}{q}\right)} = \exp\left(2\pi i \frac{p}{2q}\right) \tag{10}$$

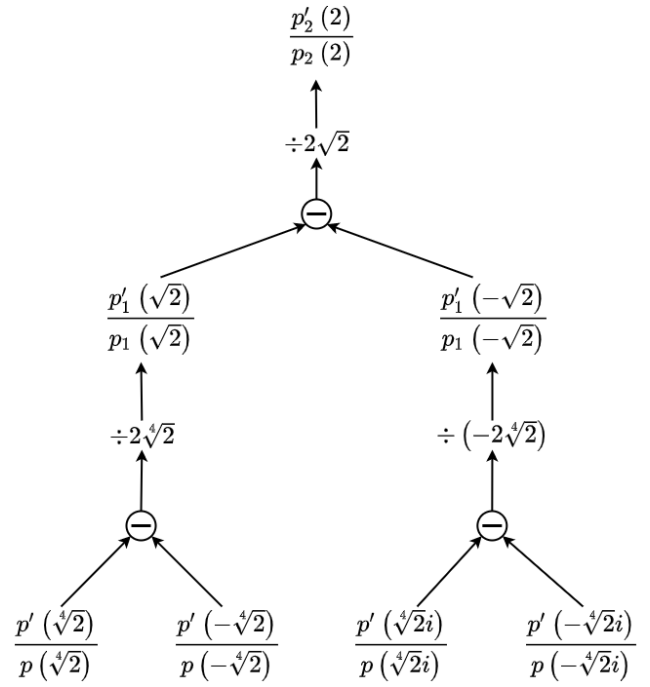

Figure 2: The steps of DLG_RATIONAL_RORM(p, p', r, t, u, l) in Alg.3 for $r = 2$, $l = 2$, $t = 1$, and $u = 1$.

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 ∪ right
else
  return [(p, q)]
end if
```

and

$$p\%q = 0 \implies \sqrt{\exp\left(2\pi i \frac{1}{1}\right)} = \exp\left(2\pi i \frac{1}{1}\right) = 1, \tag{11}$$

and the negation operation satisfies

$$p\%q \neq 0 \implies -\exp\left(2\pi i \frac{r}{s}\right) = \exp\left(2\pi i \frac{2r+s}{2s}\right) \tag{12}$$

and

$$p\%q = 0 \implies \sqrt{\exp\left(2\pi i \frac{1}{1}\right)} = \exp\left(2\pi i \frac{1}{2}\right) = -1; \qquad (13)$$

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 10, 11, 12, and 13 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(2 · π · i · r/s) for r, s in root_tree]
roots =[ ∛[l]r·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 ∈ root do
    base_step[i] := p'(r_i)/p(r_i)
end for
diff[0] := base_step
for i ≤ l do
    for j ≤ 2^(l−i−1) do
        diff[i + 1][j]:= 1/2 (diff[i][2j]−diff[i][2j+1])/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 := 1/(2πi) log(x)
u := 2^ϵ
t := (angle · 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 ϵ-bit integers and "multiplications-by-2" and $\%2^\epsilon$ (*i.e., *mod 2^ϵ) operations have constant time overhead.*

7 EXPERIMENTAL RESULTS

8 CONCLUSION

REFERENCES

[1] Alston S. Householder. 1959. Dandelin, Lobacevskii, or Graeffe. *The American Mathematical Monthly* 66, 6 (1959), 464–466. http://www.jstor.org/stable/2310626
[2] Gregorio Malajovich and Jorge P. Zubelli. 2001. On the Geometry of Graeffe Iteration. *J. Complex.* 17 (2001), 541–573.

Table 1: Experimental Data for Chebyshev.

DEGREE	ℓ	$e = -\log(\lvert x \rvert)$	MPMATH PRECISION	RELATIVE ERROR r_d	RELATIVE ERROR r_1	RUNTIME	MPSOLVE ROOT RADIUS
20	4	616	332	0.15	0.09	0.17	$[0.00982, 1.0]$
40							
80							
160							
320							