

Root-Squaring for Root-Finding

Pedro Soto

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

Soo Goo

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

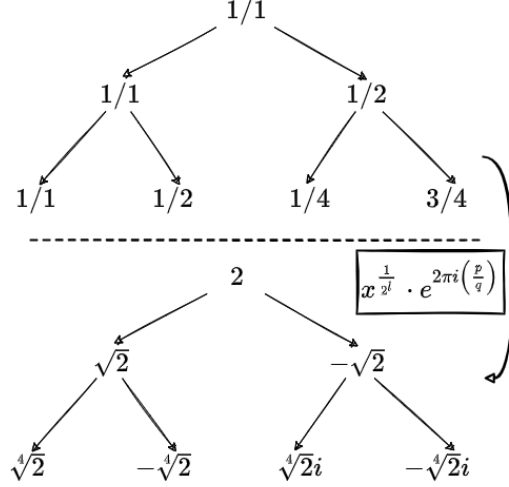


Figure 1: The upper tree depicts the steps of `CIRCLE_ROOTS_RATIONAL_FORM(p,q,l)` in Alg.1 for $l = 2$, $p = 1$, and $q = 1$. The lower tree depicts the steps of `ROOTS(r, t, u, l)` in Alg.2 for $r = 2$, $l = 2$, $p = 1$, and $q = 1$

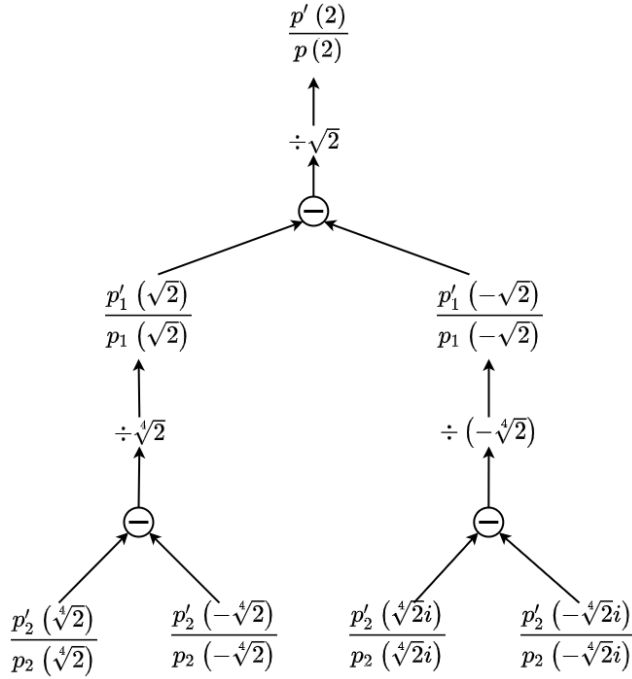


Figure 2: The steps of `DLG_RATIONAL_RORM(p, p', r, t, u, l)` in Alg.3 for $r = 2$, $l = 2$, $t = 1$, and $u = 1$.

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  $\cup$  right
else
    return  $[(p, q)]$ 
end if

```

Algorithm 2 `ROOTS(r, t, u, l)`

```

root_tree = CIRCLE_ROOTS_RATIONAL_FORM( $p, q, l$ )
circ_root =  $[\exp(2 \cdot \pi \cdot i \cdot \frac{r}{s}) \text{ for } r, s \text{ in root\_tree}]$ 
roots =  $[\sqrt[l]{r} \cdot \text{root for root in circ\_root}]$ 

```

ABSTRACT

We revisited the classical root-squaring formula of Dandelin-Lobachevsky-Graeffe for polynomials and found 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 Goo. 2022. Root-Squaring for Root-Finding. In *Proceedings of International Symposium on Symbolic and Algebraic Computation (ISSAC '22)*. ACM, New York, NY, USA, 2 pages. <https://doi.org/XXXXXXX>. XXXXXXXX

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

```

root := ROOTS( $r, t, u, l$ )
for  $r_i \in \text{root}$  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 derivs[ $l$ ][0]

```

Algorithm 4 DLG(p, p', l, x, ϵ)

```

angle :=  $\frac{1}{2\pi i} \log(x)$ 
Set  $t, u$  := angle_as_integer_ratio(angle,  $\epsilon$ )
 $r := |x|$ 
return DLG_RATIONAL_RORM( $p, p', r, t, u, l$ )

```

1 INTRODUCTION

2 RELATED WORKS

3 BACKGROUND

4 MOTIVATING EXAMPLE

5 ALGORITHM DESIGN

6 THEORETICAL ANALYSIS

7 EXPERIMENTAL RESULTS

8 CONCLUSION

REFERENCES

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

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

© 2022 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. .\$.15.00

<https://doi.org/XXXXXXX.XXXXXXX>