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

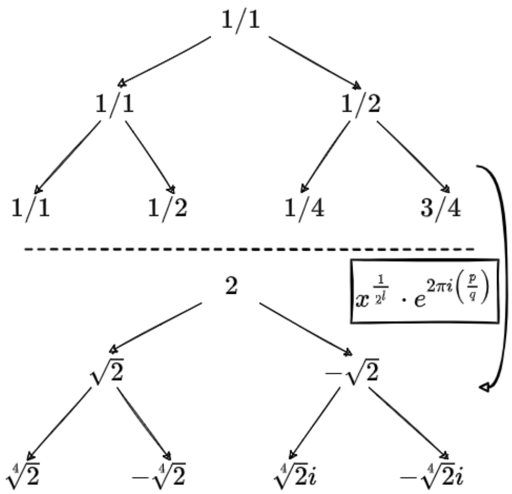

Figure 1: The upper tree depicts the steps of Circle_Roots_Rational_Form(p,q,l) in Alg.1 for $l = 2$, $p = 1$, and $q = 1$. The lower tree depicts the steps of Roots(r, t, u, l) in Alg.2 for $r = 2$, $l = 2$, $p = 1$, and $q = 1$

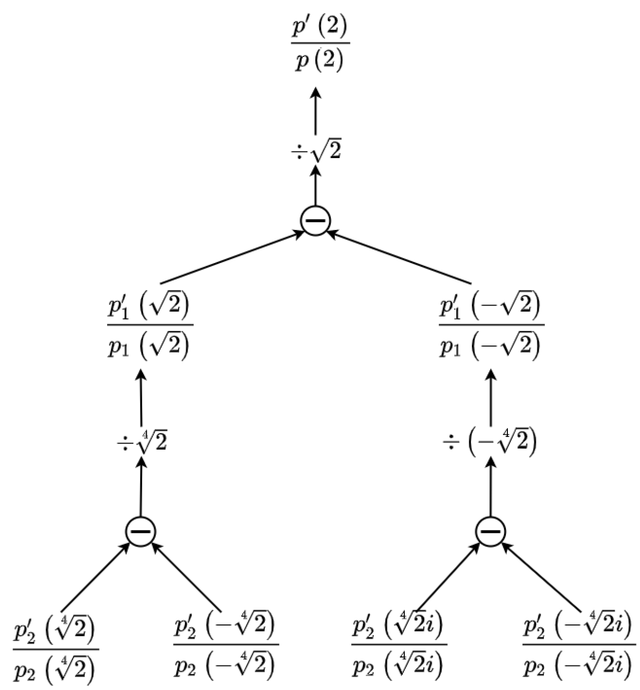

Figure 2: The steps of DLG_Rational_Rorm(p, p', r, t, u, l) in Alg.3 for $r = 2$, $l = 2$, $t = 1$, and $u = 1$.

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 $\left(2 \cdot \pi \cdot i \cdot \frac{r}{s}\right)$ for r, s in root_tree]
roots =[$\sqrt[2^l]{r}$·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.XXXXXXX

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 derivs[l][0]
```

Algorithm 4 DLG(p, p', l, x, ϵ)

```
angle := (1/(2πi)) log(x)
Set t, u := angle_as_integer_ratio(angle, ε)
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

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

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