Fast Cauchy Sum Algorithms for Polynomial Zeros and Matrix Eigenvalues

Victor Y. Pan, Soo Go, Qi Luan, Liang Zhao

http://comet.lehman.cuny.edu/vpan
victor.pan@lehman.cuny.edu
sgo@gradcenter.cuny.edu
qi_luan@yahoo.com
Liang.Zhao1@lehman.cuny.edu

Polynomial Root-Finding

Given coefficients $p_0, p_1, ..., p_d$**, approximate the roots** $x_1, x_2, ..., x_d$**:**

$p(x) = p_d \prod_{j=1}^{d}(x - x_j) = p_0 + p_1 x + \ldots + p_d x^d = 0$, $p_d \neq 0$.

Central question for math and computational math for ≈4000 years, well into the 19th century

Intensive study in the 1980s and 1990s

Pan (STOC 1995): divide-and-conquer algorithm approximating all d roots within $\epsilon = 2^{-b}||p||$ in nearly optimal Boolean complexity $\tilde{O}((b + d)d^2)$

What's Next?

Two points of consideration:

1. The algorithm is quite involved and has not been implemented
2. Performs under the classical model involving coefficients

Black Box Polynomial

Polynomials given with an oracle for their evaluations

Examples

Shifted sparse

$$p(x) = t(x - c)$$

Sum of shifted monomials

$$p(x) := \alpha(x-a)^d + \beta(x-b)^d + \gamma(x-c)^d$$

Recursively defined, e.g., Mandelbrot's

$$p_0(x) := 1,\ p_1(x) := x,\ p_{i+1}(x) := xp_i(x)^2 + 1,\ i = 0, 1, \ldots$$

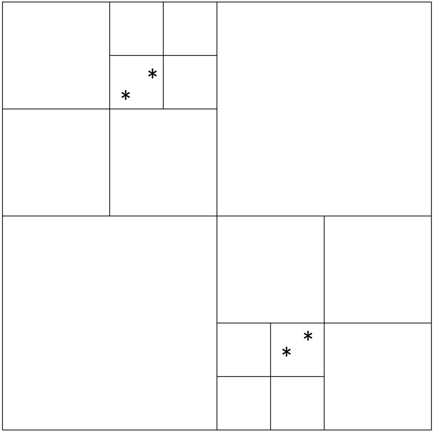
Goal

We want to devise an algorithm that

1. can work with black box polynomial
2. is both optimal and easy to implement

Subdivision Iterations (Exclusion-Inclusion Test)

Weyl 1924, Henrici 1974, Renegar 1987, Pan 2000



Our root-finder accelerates the classical subdivision approach, based on proposing novel exclusion-inclusion (e-i) tests

Subdivision Iterations

Centers of all suspect squares together approximate all roots within half diagonal, decreasing by twice in every iteration

$\Rightarrow$ error decreases by a factor of 2^b after b iterations

A root can lie in up to 4 suspect squares

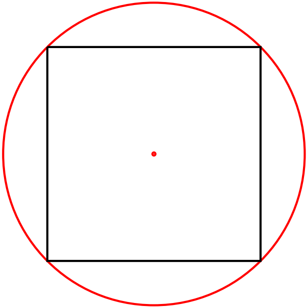
$\Rightarrow \leq 4db$ tests overall to approximate d roots within $R/2^b$, $R =$ half diagonal of the initial suspect square.

Challenges in E-I Testing

1. Known techniques are efficient for discs, not squares

2. We compute numerically; how can we be certain if a root is inside or outside the square?

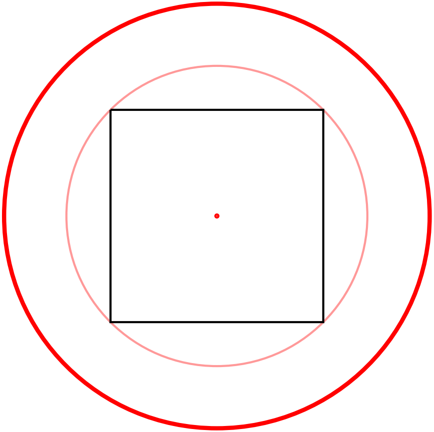
Solution: Firm Exclusion, Soft Inclusion



Exclusion: If no roots in the minimal covering disc, discard

Inclusion: If roots exist in a slightly larger disc, subdivide

Solution: Firm Exclusion, Soft Inclusion

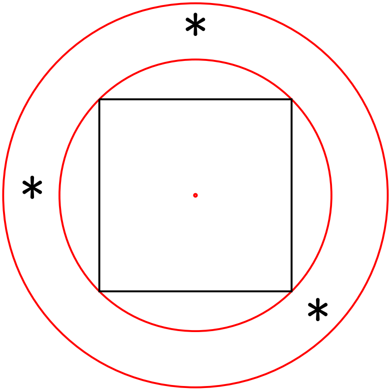


Exclusion: If no roots in the minimal covering disc, discard

Inclusion: If roots exist in a slightly larger disc, subdivide

Firm Exclusion, Soft Inclusion

Both tests can be satisfied at the same time!

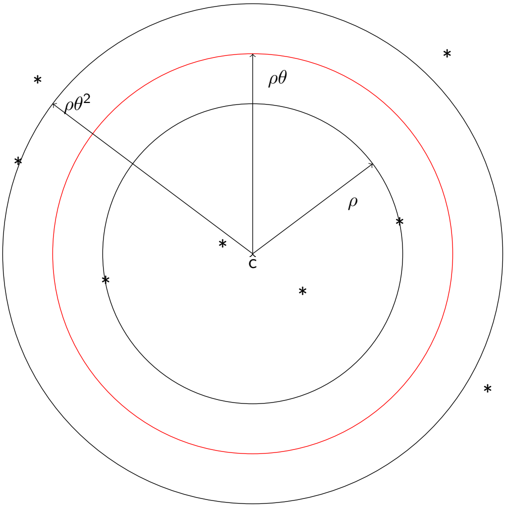


Headache? No. Just take either one of the decisions.

Soft test can be applied to any disc or square containing a fixed number of roots.

See [arXiv 1805.12042] for more.

Isolation (θ)



Cauchy Integral

Use **Cauchy integrals** with subdivision

Approximate the power sums of all m roots, $x_1, \ldots, x_m$, in a disc D with boundary circle Γ by

$$s_h := \sum_{x_j \in \mathcal{D}} x_j^h = \frac{1}{2\pi\sqrt{-1}} \int_\Gamma \frac{p'(x)}{p(x)}\, x^h\, dx,\ \ h = 0, 1, \ldots$$

$s_0 =$ the number of roots in the disc.

Approximating Cauchy Integral

For $D = D(0,1) = \{x : \ |x| \leq 1\}$, approximate the integral by

$$s_{h,q} := \frac{1}{q} \sum_{g=0}^{q-1} \zeta^{(h+1)g} \ \frac{p'(\zeta^g)}{p(\zeta^g)},$$

$$q > 1, \quad h = 0, 1, \ldots, q-1, \quad \zeta = \exp\left(\frac{2\pi\sqrt{-1}}{q}\right)$$

Goal: Obtain $s_0 = m$ by approximating it within $< 1/2$ and rounding to the nearest integer while using minimal q

Recap

Exclusion:

If $\bar{s}_0 = 0$ on the superscribing disc, discard square

Inclusion:

If $\bar{s}_0 > 0$ slightly larger disc, continue subdivision

We want this with q large enough but also minimal.

Choosing q

Theorem (Schönhage 1982)

Let the unit disc $D(0,1)$ be θ-isolated. Then

$$|s_{h,q} - s_h| \leq \frac{d\theta^h}{\theta^q - 1} \text{ for } h = 0, 1, \ldots, q-1. \tag{1}$$

$\Rightarrow$ Error decreases exponentially in q

Generalization

Extend to any disc using linear map

$$x \mapsto \frac{x-c}{\rho};\; D(c,\rho) \mapsto D(0,1);\; p(x) \mapsto t(x) = p\Big(\frac{x-c}{\rho}\Big)$$

This mapping does not change the isolation

Refinement using Randomization

Can't be sure of the isolation of the fixed initial square

$\Rightarrow$ Use **randomization** to modify the softness of our tests

New plan:

Randomly choose a radius of the initial circle from the range $[2^{0.2}, 2^{0.4}]$ and test as before

$q = \lfloor 10 m\gamma \log_2(4d+2) \rfloor$ sufficient for output error $< 1/2$ with probability at least $1 - 1/\gamma$, $\gamma \geq 1$, ; $q = q(m,d) = \tilde{O}(m)$

Error decreases by the same factor as q increases

Refinement using Randomization

Strengthen via repetition:

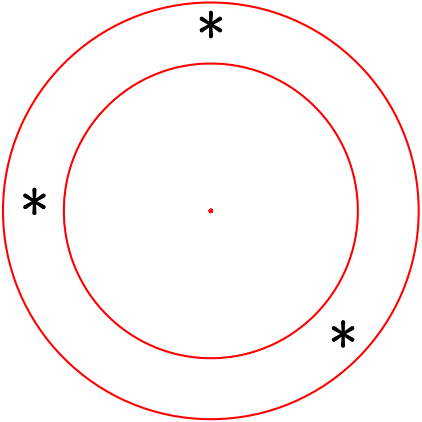
Repeat ν times to reduce the error probability to $\leq 1/\gamma^{\nu}$ at the cost of $q\nu = \lfloor 10m\gamma \log_2(4d+2) \rfloor \nu$ evaluations.

Pick some constant γ, e.g., $\gamma = 2$

Also choose a constant ν sufficiently large, or even something $O(\log(d))$; still $q(m, d) = \tilde{O}(m)$

Why is this going to work?

We only need randomization for the exclusion test.
(Inclusion is deterministic.)



Overall Complexity

Precision of computing is bounded by $\log(d)$, assuming precise polynomial evaluation (Louis and Vempala, FOCS 2016)

Evaluation with error $\in 1/d^{O(1)}$ still enables precision $\in O(\log(d))$

Total number of evaluations is $\tilde{O}(q(m,d)mb)$

Boolean complexity bound:

$$\mathbb{B}_{roots} = \tilde{O}(\mathbb{B}_d mq(m,d)b) = \tilde{O}(\mathbb{B}_d m^2 b),$$

$\mathbb{B}_\mathrm{d}$:= Boolean complexity of degree d polynomial evaluation

Can be decreased to near OPTIMAL $\tilde{O}((b+d)q(m)m) = \tilde{O}((b+d)m^2)$ with additional techniques (See arXiv 1805.12042)

Correctness

Possible, though unlikely, some roots are lost during the subdivisions. But we check the number of roots as we output to detect this.

We can only get error by deleting suspect square with roots.

Cost of detecting errors using Las Vegas randomization is dominated by the cost of computations.

$\Rightarrow$ Don't need additional complexity increase

Deterministic E/I Test (Can we do it?)

Only need to worry about exclusion: could be wrong if there are roots near the boundary

Repeat the test $2m + 1$ times using concentric circles and use majority vote

m roots can only impact m of the tests

Requires a factor of 2m+1 additional increase in the number of tests

Summary

We accelerate classical subdivision algorithm to root-finding in near optimal Boolean time even for general p(x) given with coefficients;

Faster if p(x) can be evaluated fast, e.g., is shifted sparse, Mandelbrot's etc.

Thank you!

arXiv 1805.12042