Computing at Least One of Two Roots of a Polynomial is, in General, not Algorithmic

Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso, El Paso, Texas 79968, USA
vladik@utep.edu

1. Overview

- In our previous work, we provided a theoretical explanation for an empirical fact that:
  - it is easier to find a unique root
  - than to find multiple roots.
- In this paper, we strengthen that explanation.
- Namely, we show that finding one of many roots is also difficult.

2. Background

- For each class of problems:
  - before we start designing an algorithm for solving all problems from this class,
  - it is desirable to know whether such a general algorithm is indeed possible.
- Among these problems are:
  - solving systems of equations (in particular, equations),
  - finding optima of a given function on a given box, etc.
- It is known that there exists an algorithm which is applicable:
  - to every system $f_1(x_1, \ldots, x_n) = 0, \ldots, f_m(x_1, \ldots, x_n) = 0$ with computable functions f_i
  - which has exactly one solution on a given box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$.

3. Background (cont-d)

- This known algorithm computes the corresponding solution. solution.
- To be more precise, it produces, for every $\varepsilon > 0$, an ε-approximation to the solution.
- It is also known that no algorithm is possible:
  - which is applicable to every computable system which has exactly two solutions and
  - which would return *both* solutions.
- The proof shows that such an algorithm is not possible even for computable polynomial equations.

4. Mathematical comment

- This algorithmic impossibility is due to the fact that we allow computable polynomials.
- For polynomials with rational (or even algebraic) coefficients, solution problems are algorithmically decidable.

5. Practical comment

- This result is in good accordance with the empirical fact that in general:
  - it is easier to find a point $(x_1, \ldots, x_n)$, in which a given system of equations has a unique solution
  - than when this system has several solutions.

6. Formulation of the problem

- A natural question is:
  - since in the general two-roots case, we cannot return *both* roots,
  - maybe we can return at least one of them?

7. What was known and what we plan to do

- It is known that no such algorithm is possible for *general* computable functions.
- This construction requires a computable function which is more complex than a polynomial.
- In this paper, we show that already for computable *polynomial* equations, it is impossible to compute even one of the roots.
- In this proof, we will use the polynomials:
  - with the smallest possible number of variables and
  - of the smallest possible degree.
- We also prove a similar result for optimization problems.

8. Definitions

- We want to formulate our result in precise terms.
- For this purpose, we need to recall the definition of a computable number.
- Crudely speaking, a real number is computable if it can be computed with an arbitrary accuracy.
- Formally, a real number x is called *computable* if there exists an algorithm that:
  - given an integer k,
  - returns a rational number r for which $|x - r_k| \le 2^{-k}$.
- By a *computable polynomial*, we mean a polynomial with computable coefficients.
- In line with our promise, we will prove this result for the case of the smallest possible number of variables: one.

9. First result

- *No algorithm is possible:*
  - *that is applicable to any computable polynomial function $f(x)$ with exactly two roots,*
  - *and that returns one of these roots.*
- Our proof will use the known fact that no algorithm is possible for detecting:
  - whether a given constructive real number α is non-negative
  - or whether it is non-positive.
- For every computable real number α, we can form a polynomial

$$f_\alpha(x) = [(x-1)^2 + \alpha] \cdot [(x+1)^2 - \alpha].$$

- This polynomial is equal to 0 if one of the two factors is equal to 0.
- When $\alpha = 0$, this polynomial $f_\alpha(x)$ has exactly two roots: 1 and -1.

10. First result (cont-d)

- When $\alpha > 0$, the first factor is positive, so $f_\alpha(x) = 0$ if and only if $(x+1)^2 = \alpha$, hence $x + 1 = \pm\sqrt{\alpha}$.
- So, for such α, the polynomial $f_\alpha(x)$ has exactly two roots: $x = -1 \pm \sqrt{\alpha}$.
- Similarly, when $\alpha < 0$, the polynomial $f_\alpha(x)$ has exactly two roots: $x = 1 \pm \sqrt{|\alpha|}$.
- If we could compute one of roots, then:
  - by computing this root with enough accuracy and comparing it with 1,
  - we could tell whether this root is close to 1 or to -1.
- According to our description of the roots:
  - if this root is close to 1, then $\alpha \le 0$;
  - if this root is close to -1, then $\alpha \le 0$.

11. First result (cont-d)

- We have mentioned that we cannot check whether $\alpha \geq 0$ or $\alpha \leq 0$.
- We thus cannot return one of the roots.
- The proposition is proven.

12. Comment

- In the proof, we used a 4th degree polynomial.
- Let us give reasons why we cannot use a polynomial of a lower degree.
- Since we need polynomials with two roots, we must use polynomials of degree at least 2.
- If a quadratic polynomial has exactly 2 roots, then we can find these rules by using a standard formula.
- So these roots are easy to compute.
- For a cubic polynomial $f(x)$, the only way to have exactly two real roots is to have one double root.
- At this root, the derivative $f'(x)$ is equal to 0.
- And the solution to the quadratic equation $f'(x) = 0$ can be easily found.

13. Second result

- *No algorithm is possible:*
  - *which is applicable to any computable polynomial* $f(x)$ *that attains its minimum at exactly two points,*
  - *and which returns one of these points.*
- *Proof:* it is sufficient to consider $f_\alpha^2(x)$, where $f_\alpha(x)$ is the polynomial from the previous proof.
- This new polynomial is always non-negative, and it attains its minimum 0 if and only $f_\alpha(x) = 0$.

14. Acknowledgments

This work was supported in part by:

- National Science Foundation grants 1623190, HRD-1834620, HRD-2034030, and EAR-2225395;
- AT&T Fellowship in Information Technology;
- program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478, and
- a grant from the Hungarian National Research, Development and Innovation Office (NRDI).