Lecture 1: Historical Perspective and Overview

What is it that comes to our mind when we hear the tile of the course: Algorithms for Solving of Polynomial Equations? The algorithmic part is clear to most of us, I suppose, but what is it that these algorithms are supposed to compute? What do we mean by "Polynomial Equations" and "solving" them? In it most general form, a polynomial equation is of the form

$$P(x) := a_d x^d + a_{d-1} x^{d-1} + \cdots + a_0 = 0$$

where n is the degree of the polynomial, and a_i's are its coefficients from some domain (say integers). A solution x^* is an element in some suitable domain, e.g. $\mathbb{C}$, s.t. $P(x^*) = 0$.

All of us are familiar with the quadratic equation:

$$ax^2 + bx + c = 0$$

that has "solutions" $x = (-b \pm \sqrt{b^2 - 4ac})/2a$. This formula was already known to the Babylonians (around second millenium B.C.), and it is an artefact in the museum of mathematics as it encapsulates the development of the concepts of negative numbers, irrationals, and the imaginary numbers. These concepts were not well-founded and taken for granted as they are today for more than three thousand years (at least in the western world; apparently the Indians treated the negative numbers with usual familiarity). Whether "similar" solutions, i.e. using the four basic operations and radicals, existed for the cubic and the quartic case, was a long outstanding problem that occupied the best minds of the Rennaissance era (roughly more than 3K years), and in fact it was believed that such solutions do not exist (Omar Khayaam, Fibonacci, and as late as Pacioli in 1494 believed so). The first solution to general cubic equation was discovered by Scipione del Ferro, a professor of Mathematics at Bologna, around 1500. But, as was the custom in those times, he did not publish his solutions since discoveries were kept secret and rivals were often challenged to solve the same problem. However, around 1510 he did confide his method to Antonio Maria Fior. Nothing happened until Nicolo Tartaglia, "Stammerer" (an essentially self-taught person who stammerred because of a cut receieved from the sabre of a french soldier), was challenged by Fior to solve thirty cubic equations, which he successfully did. Cardan, a professor of mathematics in Milan, cajoled Tartaglia to reveal his approach, which he revealed in a rather obscure verse form after a pledge from Cardan that he will keep it secret. However, some years later Cardan came to know of the detailed approach from della Nave, son-in-law of Ferro, and realized that Ferro had discovered the same solution as Tartaglia much before Tartaglia. So he broke his pledge and published the solution in *The Ars Magna*, The Great Art, which consequently caused a big fight between him and Tartaglia. The solution for the quartic was given soon by Ludovicio Ferrari, a student of Cardan. The next open problem was solving the quintic equation. Again, some of the best minds, such as Euler, Lagrange, Vandermonde, tried finding solutions for the quintic case but failed. Gauss made significant progress by showing that roots of the equation $x^p = 1$, for a prime p, can be expressed in terms of radicals. The significance of the result is that it shows that some higher-degree equations can be solved using radicals. Based upon Gauss's work, Abel, while still in high school, first thought that he had a general method to solve the quintic case using radicals. However, he soon realized his error and then went to prove that it is not possible to solve quintic equations using radicals in general. But the question still remained to characterize which polynomials can be solved using radicals. This was finally answered by the young Galois, who gave a characterization for solvability using radicals, introducing fundamental new concepts along the way. This negative result implies that there is no way that we can work with "closed-form" solutions to the equations, and hence, for the most part, diverted the interest from developing computational perspectives to handle roots of equations; though it led to discovery of new and interesting fields such as group theory.

Related to the problem of expressing the solutions in closed form is the number of solutions that an equation can have. The answer critically depends on the domain where we want to restrict the solutions too. The fundamental theorem of algebra states that every degree d polynomial over $\mathbb{C}$ has d roots in $\mathbb{C}$. But what if we restrict to the reals, or even to the integers? The problem becomes slightly harder, since nothing equivalent to the fundamental theorem of algebra holds; though we know an equivalent formulation of FTA from Euler that an odd degree polynomial over $\mathbb{R}$ always has at least one real root. In this course we will look at these questions from a computational perspective, as was done till the late 19th century, and add on to that the question of computational complexity. Thus, for a substantial portion of this course we will focus on what may be called the fundamental computational problem of algebra. Most of the topics covered in this section fall under what is traditionally called "Classical Algebra" or the "Theory of Equations". We will give algorithms for finding the complex roots, either all or the roots in some nice input region, of an integer polynomial, i.e. a constructive proof of FTA. We will study the computational complexity and effectiveness of various methods for finding real roots, again either in some input interval or all of them. These are solved problems, but the interesting aspect is to develop not only theoretically tight but also practically efficient algorithms for these problems.

A tentative list of topics to be covered in the course.

1. Polynomials (some fundamental properties, Vieta-Newton's relations); Root Bounds.

2. Euclid's algorithm; resultants; subresultants.

3. Algorithms for root isolation: Sturm's Theory; Descartes's rule of signs, Continued Fractions; Obreschkoff's criteria; Generalizing Descartes's Rule of Signs; Approximating roots; Interval Box-method.

4. Algorithms for computing with Algebraic Numbers. Constructive Root Bounds, and some algorithmic algebraic number theory.

¶1. Higher dimensions In higher dimensions, we will be working with multivariate polynomials $P(x_1, x_2, \dots, x_d)$ in d-dimensions. What is the fundamental theorem of algebra in higher dimensions? What is set of solutions to the equation $P(x_1, \dots, x_d) = 0$? In general the solution set is infinite (e.g. the equation of a circle, ellipse, elliptic curve etc.) so it does not make sense to look at just one polynomial to get an analogous theorem to the FTA. In higher dimensions, we are interested in a system of equations:

$$P_1 = 0, P_2 = 0, \dots, P_m = 0$$

where the P_i's are polynomials in $x_1, \dots, x_d$ and they intersect in finitely many points. In this setting the analogous result to FTA is Bezout's Theorem from Algebraic Geometry. In particular, we will be interested in the case $d = 2$. The solution set of $P(x, y) = 0$ in the plane $\mathbb{R}^2$, when the coefficients of P are integers, is called an algebraic curve (analogous to algebraic numbers in 1-d).

One of the problems that we will be interested is called the **curve-arrangement** problem, which has a nice geometric flavour: given a system of bivariate integer polynomials that intersect in finitely many points output a "topologically faithful" representation of their intersection in $\mathbb{R}^2$. The precise definition of topologically faithful has to wait, but it intuitively means the following: a system of algebraic curves in the plane partition it into connected regions of dimension two, i.e,. the elements in the set $\mathbb{R}^2 \setminus P_1 \cup P_2 \cup \dots \cup P_m$; the vertices and edges of these regions define a combinatorial object, which can be defined using a planar graph, namely a vertex is connected to all edges incident on it, an edge to the two regions bounding it and so on; our aim is to compute a sufficiently good approximation to the curve, say by planar straight line segments, such that the combinatorial structure underlying our construction is the same as the actual combinatorial object. Thus the output has to have the correct combinatorial information as well as the geometric information. We will study various algorithms for the special case of curve-arrangement.

For a system of polynomial in higher dimensions, the structure analogous to curve-arrangements is called the **Cylindrical Algebraic Decomposition** (CAD) of the system. Intuitively it means a partition of $\mathbb{R}^d$ into **cells**, i.e., regions where the signs of the polynomials do not change. This structure played a fundamental role in Tarski's result on the decidability of the first order theory of reals. A Tarski sentence is a boolean formula of polynomial inequalities with all the variables quantified; e.g.,

$$P_1 = 0 \cap P_2 \leq 0 \cup \neg(P_3 \geq 0) \dots$$

or in 1-d $x^2 - 2 = 0 \cap x > 0$. The general decision problem is to decide whether a Tarski sentence is true or false over the reals. This is one of few positive results on decidability; an instance of a negative result very similar to Tarski's is the decidability of the Diophantine equation, i.e., to decide whether a polynomial equation has an integer solution, which was shown to undecidable by Martin Davis, Yuri Matiyasevich, Hilary Putnam and Julia Robinson.

Tarski's original algorithm had an increasing tower as its complexity. Collins, Tarski's student, improved it to a double exponential, which has been further improved subsequently.

A different direction, closer to our original theme, is to solve a system of equations. We will be interested in solutions in $\mathbb{R}^d$, i.e., only the real variety. This field is called real algebraic geometry. In particular, we want to develop the equivalent of Sturm's theory and Descartes's rule of signs in higher dimensions. This would involve developing the algorithmic equivalent of Euclid's algorithm for multivariate polynomials. This was done by Buchberger in his thesis, where he develeoped the notion of Gröbner basis. This result led to a solution of the fundamental constructive problem of polynomial ideal theory: to decide whether a given polynomial P belongs to the ideal generated by $P_1, \dots, P_m$ over the ring $\mathbb{R}[x_1, \dots, x_d]$. We will then study elimination theory, which is a generalization of the concept of resultants to higher dimnesions. These ideas and algorithms will be fundamental in developing algorithms for solving a system of polynomial equations in $\mathbb{R}^d$. In particular, we will study Pedersen's work generalizing Sturm's theory to higher dimensions, and some results in getting a multivariate Descartes's rule of signs.

A tentative list of topics to be covered in the second half.

1. Groups, Rings and Theory of Ideals.

2. Algebraic Curves; Bezout's theorem; Arrangements of Algebraic Curves; Solving Bivariate Systems; Resultants based approach.

3. Elimination Theory; Theory of Ideals (Hilbert's basis theorem and Nullstellensatz).

4. Grobner Basis and solving systems in higher dimensions.

5. Pedersen's thesis – Sturm Theory in Higher Dimensions.

6. Multivariate Descartes's rule of signs.

7. Tarski's – decidability of first oder theory of real closed fields; real closed fields.

8. Applications – Robot Motion Planning, Geometric Modelling.

9. Beyond Polynomials – solving non-linear equations in general.

A note on the complexity model: our complexity model mostly will be the **bit-complexity** model, i.e., we not only want to know the number of algebraic operations but also the cost that each operation takes. Thus in this model our input will mostly be restricted to integer polynomials.

Assessment will be based upon perhaps some assignments, but mostly on presentations of papers.

We will follow various books and papers, as the need arises. "Fundamental Problems of Algorithmic Algebra", by Chee Yap, will be the standard reference though occasionally we will also refer to the following books:

1. Theory of Equations by Burnside, Uspensky and others.

2. Mignotte – Mathematics for Computer Algebra.

3. Basu-Pollack-Roy: Algorithms in Real Algebraic Geometry.

4. Cox, Little, O'Shea: Using Algebraic Geometry.

5. Cohen: Algorithmic Algebraic Number Theory.