Systems of polynomial equations

Johan Richter

Robotics

Suppose you want to model a robot arm. You will probably model it as a sequence of line segments connected by joints, rather like our own arms. If (x_i, y_i) is the coordinates of some joint, (x_{i+1}, y_{i+1}) is the coordinates of the next joint (or the hand) and the segment has length L_i then we get the equation

$$(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 = L_i^2.$$

Note that the angle at the i-th joint can be computed from the equations

$$\begin{aligned} x_{i+1} - x_i &= L_i \cos(\theta_i) \\ y_{i+1} - y_i &= L_i \sin(\theta_i). \end{aligned}$$

It is easy to see that we get a system of polynomial equations that describe possible positions of the robot arm.

Geometry

Another application of system of polynomial equations is in theorem proving in geometry. Many statements in Euclidean geometry can be formulated as result about systems of polynomial equations.

Example. A well-known statement in Euclidean geometry says that *the bisectors of the three sides of a triangle meet in one point.* We are going to show how this can be formulated as a statement about polynomials.

One vertex, A, of the triangle can be assumed to be the origin and another, B, to have coordinates $(c, 0)$, without loss of generality. The third vertex, C, has coordinates (a, b).

The bisectors of AB and BC meet in some point, with coordinates (x_1, y_1). We get the following equations

$$\begin{aligned} x_1 - \frac{c}{2} &= 0 \\ \frac{c-a}{b} \cdot x_1 - y_1 + \frac{a^2 + b^2 - c^2}{2b} &= 0. \end{aligned}$$

Similarly, if (x_2, y_2) are the coordinates of the intersection of the bisectors of AB and AC we have

$$\begin{aligned} x_2 - \frac{c}{2} &= 0 \\ \frac{a}{b} \cdot x_2 + y_2 - \frac{b^2 + a^2}{2b} &= 0. \end{aligned}$$

We thus want to prove that these four equations imply that $(x_1, y_1) = (x_2, y_2)$.

Sudoko

Sudoko is a form of number puzzle where you enter mark squares of a 9x9 grid with an integer subject to certain restrictions.

We will show how a Sudoko puzzle can be encoded as a system of equations. Introduce 81 variables, x_i. For square i we have an equation,

$$\prod_{k=1}^{9}(x_i - k) = 0,$$

to encode that x_i must take an integer value between 1 and 9.

For the first row we have the equation

$$\prod_{k=1}^{9}(x_k - 1) = 0,$$

to encode the fact that one of the squares in the first row must be marked with 1. Continuing in this way we get a system of equations (exercise for the reader: how many equations do we get) that is equivalent to conditions of a sudoko board. By fixing some of the variables x_i we can create a sudoko puzzle.

Gröbner basis

We will discuss results relevant to solving systems of polynomial equations but we will begin by adopting a slightly different point of view. The methods we introduce will eventually allow us to compute whether a system of equations has a solution, and if so how many. Sometimes they will also allow us to find the solutions explicitly.

We will be working over the complex numbers, for technical reasons. We will often write the variables as x_1, x_2 and so on. Set $\bar{x} = (x_1, x_2, \ldots, x_m)$. So we will write $f(\bar{x})$ instead of $f(x_1, \ldots, x_m)$ for convenience's sake.

Note that if

$$g(\bar{x}) = h_1(\bar{x})f_1(\bar{x}) + \ldots h_n(\bar{x})f_n(\bar{x}),$$

then if a is a common zero to all the f_i, then it is a zero of g as well.

This leads us to introduce a piece of terminology.

Definition. An *ideal* is a set of polynomials, I, such that, for $f, g \in I$ and $h \in K[\bar{x}]$, we have $f + g \in I$ and $hf \in I$. A finite set of polynomials, $\{f_1, \ldots, f_n\} \subseteq I$, such that any element, g, in I can be written

$$g(\bar{x}) = h_1(\bar{x})f_1(\bar{x}) + \ldots h_n(\bar{x})f_n(\bar{x}),$$

for some h_i, is called a *basis* for I and we say that the f_i *generate* I.

Theorem (Hilbert Basis Theorem)**.** *Every ideal in $K[\bar{x}]$ has a finite basis.*

If we have a system of equations,

$$\begin{cases} f_1 = 0 \\ \ldots \\ f_n = 0, \end{cases}$$

the polynomials on the LHS generate some ideal. We will find a better basis for this ideal which gives us an equivalent system of equations which is easier to work with.

Before we define this basis we need to take a detour by trying to generalize two concepts from univariate polynomials to multivariate polynomials.

We would like to speak of the *leading term* of a multivariable polynomial. To do this we want to introduce an order on the set of monomials. There are several possible good ways of doing this but we insist that the ordering should satisfy two properties:

1. 1 is less than any other monomial.

2. If $u < v$ and w is any other monomial, then $uw < uw$.

One possible order is the *lexiographic* order (dictionary order) where one has

$$x_1 < x_2 < x_2x_1 < x_2^2 < x_2^2x_1 < x_2^2x_1^3 < x_2^3.$$

The leading term of a polynomial, f, for a given monomial ordering, is simply the largest monomial that occurs with non-zero coefficient in the polynomial. We denote it $\mathrm{lt}(f)$ and its coefficent by $\mathrm{lc}(f)$.

If at least one of the monomials of a polynomial f is a multiple of the leading monomial of a polynomial g, then we say that f is reducible by g and define the *reduction* of f by g as follows: if m is the largest monomial in f that is divisible by $\mathrm{lt}(g)$, it has coefficient c and $m = q\,\mathrm{lt}(g)$ then we form

$$\mathrm{red}(f, g) = f - \frac{c}{\mathrm{lc}(g)}qg.$$

Theorem. *If $f_1, \ldots f_n$ is any finite set of polynomials, and g is a polynomial then we can write*

$$g = \sum a_i f_i + r,$$

where a_i and r are polynomials and r is not reducible by any of of the f_i.

Finally we define the important concept of a Gröbner basis.

Definition. Let I be an ideal. A basis for I, $G = \{g_i\}$, is a *Gröbner basis* for I if the remainder in mutivariate division by G is unique for any polynomial. It is a *reduced Gröbner basis* if the leading coefficient of every element in G is 1 and there is no subset of G that is also a Gröbner basis.

Theorem. *Given any monomial ordering, any ideal has a unique reduced Gröbner basis.*

In practice, this Gröbner basis can be computed by most computer algebra systems. If you are unlucky this may take a lot of time, because the Gröbner basis can be very large.

Applications of Gröbner basis

Suppose as before that we have a system of equations, $f_i = 0$. If we compute the Gröbner basis for the ideal generated by the f_i we get a set of polynomials, $G = \{g_i\}$. The system of equations, $g_i = 0$, is a system of equations that is equivalent to the system we started with. Hopefully it is a simpler system. In general it will still not be trivial to solve, but there are certain conclusions that can be drawn from it.

Theorem. *If $G = \{g_i\}$ is a Gröbner basis, then the system of equations $g_i = 0$, $i = 1, \ldots, n$ has a solution unless $c \in G$ for some constant c.*

We can generalize the preceding theorem to give us a count of how many solutions there are.

Theorem. *Suppose $G = \{g_i\}$ is a Gröbner basis. Then the system of equations $g_i = 0$ has finitely many solutions if and only if for each variable x there is an element of G with leading term that is a power of x. The number of solutions, counted with multiplicity, is equal to the number of monomials that are not a multiple of any leading term in G.*

Note that this allows us to check if our Sudoko systems have unique solutions.

It is easy to determine whether a polynomial lies in the ideal generated by a certain Gröbner basis.

Theorem. *Let G be a Gröbner basis. A polynomial f lies in the ideal generated by G if and only if the remainder of f upon division by G is zero.*

Proof. By definition, f lies in the ideal generated by G if and only if we can write $f = \sum a_i g_i$. This is equivalent it having a zero remainder when you divide by G in some way. Since G is a Gröbner basis the remainder is unique, which implies the theorem. □

We can use this to solve our geometry problem on the intersection of bisectors. We check that the polynomials $x_1 - x_2$ and $y_1 - y_2$ lie in the ideal generated by the polynomials on the LHS of the system of equations.

If we want to try and solve a system of polynomial equations explicitly, it is especially convenient to use some lexicographic order. Consider some ideal $I \subseteq \mathbb{C}[x, y, z]$. We can consider its intersection with $\mathbb{C}[z]$ and with $\mathbb{C}[y, z]$. These intersections will be ideals in the respective polynomial subrings.

Theorem. *Let I be an ideal in $\mathbb{C}[x, y, z]$ If G is a Gröbner basis of I with respect to the lexicographic order with $x > y > z$ then the ideal $I \cap \mathbb{C}[z]$ is generated by $G \cap \mathbb{C}[z]$. Similarly the ideal $I \cap \mathbb{C}[y, z]$ is generated by $G \cap \mathbb{C}[y, z]$.*

The applications to solving systems of equations are obvious.

Computing Gröbner bases

I will briefly describe an algorithm for computing Gröbner bases. It is known as Buchberger's algorithm, after its inventor. (Who also invented Gröbner bases. Gröbner was his PhD advisor.)

A preliminary concept we will use is something called the S-polynomial. Suppose we have fixed some monomial ordering. If we are given two polynomials f and g we can form

$$S(f,g) = \frac{\operatorname{lcm}(\operatorname{lt}(f),\operatorname{lt}(f))}{\operatorname{lt}(f)} f - \frac{\operatorname{lcm}(\operatorname{lt}(f),\operatorname{lt}(f))}{\operatorname{lt}(g)} g.$$

Here lcm is the least common multiple of two monomials.

We have the following theorem.

Theorem. *Let I be an ideal and $G = \{g_1, \dots, g_t\} \subset I$ be a finite set of polynomials in I. Then G is a Gröbner basis for I if and only if for all i, j it is true that the remainder of $S(g_i, g_j)$ on division by G is* 0.

We can now describe the algorithm, which we note does not initially give us a reduced Gröbner basis. We start with some set of polynomials F that generate an ideal and transform it into a Gröbner basis for the same ideal. We assume that we are working with some fixed monomial ordering. (In practice, this ordering would be part of the input to the algorithm.) We assume that $\operatorname{red}(f, F)$, where f is a polynomial and F is a set of polynomials, give some particular remainder that you can get when you perform division of f by F.

The algorithm forms a sequence of sets, F_i. We have $F_1 = F$ and

$$F_{n+1} = F_n \cup \{\operatorname{red}(S(f,g), F_n) \mid \operatorname{red}(S(f,g), F_n) \neq 0, f, g \in F_n\}.$$

Theorem. *Using the notation above one finds that the sequence (F_n) is eventually constant, ie $F_N = F_{N+1} = F_{N+2} = \dots$ for some N. Then F_N is a Gröbner basis for the ideal generated by F.*

We note that if $F_k = F_{k+1}$ then $F_i = F_k$ for all $i > k$. So we can easily check when the algorithm has terminated.

References

A good reference for all this material is *Ideals, Varieties and Algorithms* by Cox and O'Shea.