Gröbner Basis [*]

Graeme Taylor

January 18, 2005

A Gröbner basis for a system of polynomials is an equivalent and hopefully simpler form of that system relative to an ordering, from which information about the roots of the original system can be found.

Finding the common roots of a system is a significant mathematical problem that lends itself to the brute force power of computers. For polynomials in a single variable, this links to finding the greatest common divisor (itself a non-trivial and challenging part of developing computer algebra systems); whilst for a linear system of equations in several variables, the full force of matrix techniques such as gaussian elimination can be employed. However, for systems of arbitrary degree (i.e., not necessarily linear) in multiple variables, the situation rapidly becomes ugly.

Even with a canonical way to describe the polynomials of the system, many systems with different polynomials could share the same set of common roots. What this means in practice is that a complicated system of equations may have a 'simpler' presentation from which the roots are more immediately obvious. The motivation of the Gröbner basis is to have a systematic description of that 'simplest' system to assist in the determination of roots. Whilst such a basis can be described in purely mathematical terms (such as ideals), a more practical formulation requires some additional ideas first.

Definition:Given a set G of polynomials g, define a **reduction of** f **by** g as

$$f \rightarrow_G f' = f - \lambda g$$

where λg annihilates the leading monomial of f. (λ an element of the ideal generated by g in the ring being studied.)

[*] First appeared on Everything2, at *http://www.everything2.com/index.pl?node_id=1693674*

By working with respect to an ordering on the polynomials, the result of a reduction is to give a polynomial equivalent to f (relative to the ideal generated by G), yet simpler with respect to that ordering. This process is unlikely to be unique (many different g might be suitable for performing a reduction), and we can repeat the process several times:

> **Definition**: Denote by $f \to_G^* f'$ the result of a chain of all possible reductions $\to_G$ of f. (That is, repeatedly reduce f by elements of G until no g has a leading monomial that divides the result.) Then f' is known as the **normal form** of f with respect to G, denoted $NF_G(f)$
>
> **Definition**: Let $F = \{f_1, \ldots, f_n\}$ be a collection of polynomials. Then $G = \{g_1, ..., g_k\}$ is a **Gröbner basis** for F if the zeros of $\{f_i\}$ are the same as those for $\{g_j\}$ and NF_G is always unique (a Church-Rosser condition on the reduction).
> Equivalently, if I is the ideal generated by G, then $\forall f \in I,\ NF_G(f) = 0$.
> Also equivalently, the ideal generated by $\{lm(g) \mid g \in G\}$ is equal to the ideal generated by $\{lm(f) \,|\, f \in I\}$, where lm denotes the leading monomial (which requires an ordering to make sense!)

Simply knowing the shape of the Gröbner basis may suffice to describe relevant properties of the solution set. However, as it stands the defintion of the basis is not unique- if any element is also a member of the ideal generated by the others, it is redundant and can be discarded.

> **Definition**:A basis G is **auto-reduced** or **minimal** if $\forall g,\ g {\to_{G\setminus\{g\}}}^* g$ (i.e. no other element of G can reduce g).

If we further stipulate that the basis be monic, we will find a unique (up to an ordering) monic auto-reduced Gröbner basis for any ideal. Then working with such a version we can draw some conclusions:

- $G = \{1\}$ iff there are no solutions.
- Each variable appears in a leading monomial consisting only of a power of that variable iff there are finitely many solutions.
- Therefore, the number of monomials irreducible by $\to_G$ bounds the number of solutions (it is the number of solutions counted with multiplicity).

Given a purely lexicographical Gröbner basis with a finite number of solutions, then we may be able to determine all of those solutions by a method similar to back substitution in linear algebra problems.

If our variables are $x_1, \ldots, x_n$ then there is a polynomial purely in x_n (by our second observation above). This determines the x_n co-ordinates of any solution.

Now there exists at least one polynomial in the basis whose leading monomial is in x_{n-1}. As we are in a purely lexicographical setting, this polynomial can only feature x_{n-1} and x_n. By evaluating for one of the x_n determined above, we get a polynomial in x_{n-1} which can then be solved to find the x_{n-1} co-ordinates for that value of x_n. By trying all the x_n in turn we can thus determine the complete set of x_{n-1}, x_n pairs.

Thus in general knowledge of $x_{i+1}, \ldots, x_n$ allows a determination of x_i. In this way we can build a full solution set. (Assuming, of course, we are equipped to solve any polynomial we are faced with- but we can always at least describe them in (RootOf foo) form even if those roots cannot be evaluated in our chosen ring)

There may be many choices of polynomial at each step, some more restrictive than others. Helpfully, there is a theorem to guide the selection of polynomials.

> **Gianni-Kalkbrener Theorem**: If $x_{i+1}, \ldots, x_n$ have been determined, then the polynomial in x_i of lowest degree such that the leading coefficient in x_i does not vanish for the evaluation determines all possible values of x_i for that evaluation.

The defintion of a Gröbner basis as it stands does not illustrate how to construct one. However, there is a reformulation which suggests a construction, a method known as Buchberger's algorithm.

The situation is complicated somewhat by the representation used for the polynomials. It has been shown that a purely lexicographical version of the basis is the most useful for finding solutions- unfortunately that choice is one of the worst for efficient use of Buchberger's algorithm, which favours a total degree reverse lexicographical rendition. Another algorithm, the FGLM (Faugre-Gianni-Lazard-Mora) algorithm, exists to convert a Gröbner basis from one ordering to another. Thus the general approach to solving systems of polynomials via Gröbner bases proceeds along the following lines-

- Initial system- convert to total degree reverse lexicographical ordering to obtain
- tdeg-rlex system- Apply Buchberger's Algorithm to find
- tdeg-rlex Gröbner basis- Apply FGLM to get
- purelex Gröbner basis- If finite, Solve by use of the Gianni-Kalkbrener theorem to determine
- Solution

All of this (and more) can be handled by the Computer Algebra system Maple, which has an entire package devoted to Gröbner Bases (known as Groebner to avoid the awkward ö when working with an english keyboard layout.)