Gröbner Bases and Macaulay Matrices

Bruno Buchberger

Research Institute for Symbolic Computation
Johannes Kepler University, Linz, Austria

In my PhD thesis 1965 and the subsequent publication 1970 in aequationes mathematicae, I introduced the notion of Gröbner bases and gave an algorithm for constructing Gröbner bases. The algorithm is based on proving that a set *F* of multivariate polynomials (over a field) is a Gröbner bases iff all S-polynomials of *F* reduce to 0 w.r.t. *F*. The algorithm for constructing a Gröbner basis G for any given *F*, proceeds by iterating the formation of S-polynomials and adding non-zero reduction results to the basis until no more non-zero results occur. In this approach to Gröbner bases, the notion of S-polynomials plays the crucial role:

> The S-polynomial of two (monic) polynomials *f* and *g* :=
> *u* . *f* - *v* . *g*, where the power products *u* and *v* are such that
> *u* . LPP(*f*) = *v* . LPP(*g*) = the least common multiple of LPP(f) and LPP(g).

Here LPP stands for “leading power product” (w.r.t. to a giving admissible ordering) and the reduction of a polynomial *f* w.r.t. a given set *F* of polynomials is a generalization of the notion of division of univariate polynomials with remainder. The steps in the reduction of polynomials w.r.t. polynomial sets can also be viewed as Gauß’schen steps in a matrix (a “Macaulay matrix”), where the columns are labeled by the (ordered) power products and a polynomial *h* is represented as a row with the coefficient of *h* at power product *p* entered in the respective column. It should be clear that if we take the (infinite) matrix of all multiples *u* . *f* (*u* a power product and *f* in *F*) as a vector space basis over the field of coefficients then the vector space generated by this basis is equal to the ideal generated by *F*.

In the early eighties, I was asking myself whether one could also pursue a completely different strategy for computing Gröbner bases proceeding by the following three steps:

> 1. Produce all multiples *u* . *f* (which I called "shifts") of the polynomials in the initial basis *F* (up to a certain degree *D*) and put them into a matrix (the “Macaulay matrix of degree D).
> 2. Triangularize the resulting matrix.
> 3. Take the "contour" in the diagonal of the matrix, i.e. the set of all those polynomials in the diagonal whose leading power products are not a multiple of the leading power product of any other polynomial in the diagonal.

(See B. Buchberger, Miscellaneous Results on Groebner Bases for Polynomial Ideals II. Technical Report 83/1, University of Delaware, Department of Computer and Information Sciences, 1983.)

It is easy to prove that the above procedure yields a Gröbner basis if one starts with the infinite Macaulay matrix containing *all* shifts *u.f.* However, of course, this is not an algorithm since the initial matrix is infinite. So I posed the question whether one can give an a-priori bound on *D* so that, if one puts all shifts *u* . *f* with degree of *u* smaller or equal *D* to the initial matrix, the resulting matrix will be a Gröbner basis for F. Over the years, several of my PhD students tried to find such a bound and prove it correct but only recently my PhD student Manuela Wiesinger-Widi was able to establish such a bound, namely:

> **Theorem (Manuela Wiesinger-Widi 2014)**: In the above procedure, it suffices to take the power products *u* with degree less or equal to

$$\sum_{j=0}^{n-1}(d\,r)^{2^j} + 2\left(\frac{d^2}{2}+d\right)^{2^{n-1}}.$$

In this talk, we will sketch the proof and make some remarks about the possible applications of this result. The immediate application, namely the computation of Gröbner bases is practically not interesting because, typically, the algorithm using S-polynomials will be much faster. S-polynomials can just be considered as the strategy to avoid generating the complete Sylvester matrix and to stay with the lowest possible degrees in the matrix of shifts. This is analogous to the univariate situation of two polynomials *f* and *g:* The triangularization of the Sylvester matrix (with a number of rows which is equal to the sum of the degrees of *f* and *g*), in principle, yields the greatest common divisor (GCD) of *f* and *g.* However, Euclid's algorithm achieves this with staying always below the maximum degree of *f* and *g.* The above result, however, could be interesting for establishing a generalization of Habicht's (1948) theory of subresultants that explain a-priori which coefficients may occur in the intermediate polynomials of a GCD computation. This could also have relevance for the numeric computation of Gröbner bases. We will discuss some aspects of this in the talk.

In fact, the above "linear algebra" approach has been used over the years in various studies on Gröbner bases, for example in the theoretical work on bounds by E. Mayr and his students, in the theoretical work of D. Grigoriev, and in the practical approach by J.C. Faugère et al. to the computation of Gröbner bases by partially setting up the generalized Sylvester matrix and triangularizing it. In the talk, I will also report on some of my own experiments with the Macaulay matrix approach to Gröbner bases.

Also, in the past years, I was working on formalizing Gröbner bases theory (in the Theorema Project) with the goal of developing the theory semi-automatically by the Theorema automated reasoning tools. This resulted in a completely automated synthesis of my 1965 algorithm (work together with my former PhD student Adrian Craciun) and a complete formalization of the entire S-poloynomials based theory (by my former PhD student Alexander Maletzky). Maletzky is now working on the formalization of the Macaulay matrices based theory. I will also report on this.