Mathematical note: Resultants

Warren D. Smith
wds@research.NJ.NEC.COM

date = October 16, 1998

Abstract — **We explain "resultants" and their use in solving polynomial systems of equations. We also discuss "discriminants" and the "Gröbner basis."**

Keywords —

The reader presumably already knows how to solve systems of n independent *linear* equations by "elimination." One finds $n-1$ independent linear combinations of the given equations, in all of which, some one of the n variables has been *eliminated*. One then solves the size $n-1$ system recursively, and then finds the value of the eliminated variable by trivially solving a univariate linear equation.

1 Resultants

It is less well known that systems of *polynomial* equations may also be solved by an elimination idea. However, just taking linear combinations among the given equations is not sufficient. Instead, one replaces an equation by its "resultant" with another equation with respect to some variable x, to eliminate x.

1.1 Definition of the Resultant of two polynomials

Let two polynomials in x be

$$A(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots a_2 x^2 + a_1 x + a_0, \qquad (1)$$

with roots α_j, and

$$B(x) = b_m x^n + b_{m-1} x^{n-1} + \cdots b_2 x^2 + b_1 x + b_0. \qquad (2)$$

with roots β_j. Then

$$\mathrm{Res}_x(A,B) = a_n^m b_m^n \prod_{j=1}^{n} \prod_{k=1}^{m} (\alpha_j - \beta_k) \qquad (3)$$

and equivalently (and more usefully, if one does not know the roots but does know the coefficients of A and B)

$$\mathrm{Res}_x(A,B) = \det \qquad (4)$$

$$\begin{pmatrix} a_n & a_{n-1} & \dots & a_2 & a_1 & a_0 & 0 & \dots & 0 \\ 0 & a_n & a_{n-1} & \dots & a_2 & a_1 & a_0 & 0 & 0 \\ \vdots & & \ddots & & & & \ddots & & \\ 0 & \dots & 0 & a_n & a_{n-1} & \dots & a_2 & a_1 & a_0 \\ b_m & b_{m-1} & \dots & b_2 & b_1 & b_0 & 0 & \dots & 0 \\ 0 & b_m & b_{m-1} & \dots & b_2 & b_1 & b_0 & 0 & 0 \\ \vdots & & \ddots & & & & \ddots & & \\ 0 & \dots & 0 & b_m & b_{m-1} & & b_2 & b_1 & b_0 \end{pmatrix}$$

where the matrix is $(n+m) \times (n+m)$ and has m rows of a's and n rows of b's.

1.2 How to use resultants to do elimination

Two polynomial equations $A(x,y) = 0$ and $B(x,y) = 0$ have a common solution (x,y) if and only if, at that value of y, the polynomials $A(x,y)$ and $B(x,y)$ regarded as univariate polynomials in x, have a common root. In other words, considering (EQ 2) if and only if $\mathrm{Res}_x(A,B) = 0$.

Thus the problem of solving our two polynomial equations in two variables reduces to solving a univariate polynomial (in y) equation. After solving for y, we can then solve for x, again a univariate problem. Warning: spurious solutions can arise: you have to keep checking at every stage that the current partial solution really is compatible with all the equations. But that is just a matter of "plugging and chugging."

Example: $A(x,y) = xy - 1$, $B(x,y) = xy + y$.

$$\mathrm{Res}_x(A,B) = \det \begin{pmatrix} y & -1 \\ y & y \end{pmatrix} = y^2 + y. \qquad (5)$$

This has roots $y = 0$ and $y = -1$. If $y = 0$ then it is clearly impossible to find x so that $A(x,y) = 0$, so this solution is spurious. If $y = -1$, then $x = -1$, which is the unique solution.

To solve any system of multivariate polynomial equations, then, all you need is resultants and a univariate polynomial rootfinder. You just replace the n equations by $n-1$ equations in 1 fewer variable by using resultants of pairs of polynomials to get rid of that variable, and continue on until you are left with a univariate polynomial. Then backsolving (at each step we need to solve a univariate polynomial and discard whatever solutions we can tell are spurious) gives us all solutions.

1.3 Why the determinant and root-product resultant formulas both work

Because: A and B (regarded as univariate polynomials in x) have a common root if x^{n+m-1},..., x^n, ..., x^2, x, 1 is a solution of the *linear* system of equations with the same matrix as the determinant, and the zero vector on the right hand side. A nonzero solution (such as this one) can exist only if the determinant is zero.

This realization is usually attributed to Etienne Bezout. However L. Euler and I. Newton apparently knew it also

(Newton was handicapped by the fact that determinants had not yet been invented, but he had some kind of equivalent process in many low degree cases) and the matrix is often called "Sylvester's matrix" after J.J.Sylvester.

1.4 Discriminants

The discriminant of a polynomial $P(x)$ is zero if and only if it has a multiple root:

$$\mathrm{Discrim}(P) = \mathrm{Res}_x(P, P'). \tag{6}$$

1.5 More efficiency

The fact that the Sylvester resultant only allows one to eliminate *one* variable from *two* polynomial equations in a step, can lead to inefficiency. In fact, the degrees of the polynomials tend to square (or worse) every step, and the coefficients also blow up rapidly[1], causing the whole calculation to consume horribly exponential memory space.

It would be better (to make the exponential blowup less horrible) to be able to make supersteps which eliminate more than one variable from more than 2 equations. Methods for doing that were invented by [5] [6] [8] [9] [13] [14].

In particular, Dixon's method [5] as explained in [?] will allow rephrasing the existence problem for $n+1$ polynomial equations in n variables as a question of whether (assuming the system is in sufficiently "general position") a certain determinant is zero. The determinant is of a matrix $\le n!\prod_i \mathrm{maxdeg}_i$ on a side, where maxdeg_i is the maximum degree to which variable number i occurs in all the polynomial equations.

I think Dixon's method plus some further tricks enables solving the "existential theory of the reals" (ETR) in PSPACE. ETR is known to be PSPACE complete, but the available proofs that ETR$\subseteq$PSPACE seem considerably messier than just directly applying Dixon's formula (although, I admit very nongeneric cases are messier; one needs to apply the resultant backtracking-to-eliminate-spurious-solutions idea in some manner) The "further tricks" are:

1. we need to compute the determinant without ever storing the matrix, and without even storing a vector of the same size as the dimension of the matrix, because that would be super-polynomial space consumption. We are able to generate the jkth matrix entry on demand, though. Fortunately, It was shown by Csanky [?] and also Berkowitz [?] more generally (in arbitrary commutative rings) that the determinant of an $N \times N$ matrix, and indeed all the coefficients of the characteristic polynomial of that matrix (the constant coefficient is the determinant; the linear term gives the trace) may be computed by a parallel computer with polynomial(N) processors, in $O(\log N)^2$ steps. (In[2] computational complexity class lingo, determinant is in the class NC.) These methods depend on the obvious fact that matrix multiplication is in NC. But NC$\subseteq$POLYLOG-SPACE, that is, any computation performable by a parallel computer with polynomial(N) processors, in $O(\log N)^k$ steps where k is a constant and N is the number of bits in the input, is also performable by just *one* processor that only consumes $O(\log N)^{k+1}$ memory bits. Hence, there is a way to evaluate the determinant (or to solve linear systems or do matrix inversion, by "Cramer's formula" expressing these tasks as determinants) even for our huge matrix, in PSPACE.
2. The second trick allows us to avoid having to deal with arithmetic on enormous integers – we do all arithmetic modulo many small primes p and then could reconstruct the answer by chinese remaindering at the end, or alternatively we could just realize that being zero modulo enough primes requires one to actually be zero, or to be enormous (greater than the product of the primes) – and with enough primes we can rule out enormity by considering standard upper bounds (such as Hadamard's) for determinants.

Surveys of resultants include [1] [7] [16].

The best resultant based methods enable solving the "existential theory of the reals" (ETR) in PSPACE [15].

2 Gröbner bases of polynomial Ideals

Unfortunately the approach we just described still seems to rely, ultimately, on a numerical algorithm for univariate polynomial rootfinding. It would be better (for some purposes, anyhow) to be able to prove statements about existence, uniqueness, or counts, of solutions to polynomial systems, without *ever* getting one's hands dirty doing arithmetic with inexact real numbers – while at the same time leaving one's options open to decide to go for such numbers if one later concludes it will be worth the trouble.

A beautiful method to accomplish that is the method of "Gröbner bases."

First, we have to describe the notion of a polynomial "ideal." Suppose you have some system of polynomial equations such as

$$x^2 + y^2 - 1 = 0, \quad xy + yz^3 - 4 = 0, \quad z + y - 2 = 0. \tag{7}$$

Well, we could have described the same algebraic set with a different "basis" (set of generators). For example, we could have set some linear combination of the first two polynomials to 0, rather than the second polynomial. In fact the

[1] Enormous coefficients are best dealt with by using mod p arithmetic for a large number of small primes p, and then reconstructing the answers by chinese remaindering at the end of the calculation.

[2] Csanky's algorithm is very ingenious and can't be explained here. It depends on calculating the sum of kth powers of the eigenvalues of the matrix by matrix powering tricks. Then the conversion between these numbers and the coeffcients of the characteristic polynomial may be done quickly using fast fourier transform tricks. Also, all the arithmetic operations $(+, \times)$ you need with big integers, may be done in NC using FFT tricks for $\times$ and fast "carry lookahead tree" tricks in the case of $+$.

coefficients in the linear combination could themselves have been polynomials.

For algebraists: the multivariate polynomials with integer coeffcients form a "commutative *ring*" (has commutative multiplication and addition, has 1 and 0, has additive inverses). Any set of polynomials generates an "*ideal*" in this ring, i.e., any linear combination among them, with coefficients being any ring elements, is also in the ideal. The simplest example: The integers form a ring, and the ideal generated by 2 is the even integers.

So: among all the possible bases for an ideal, which is the best?

In this particular case, an equivalent basis is

$$z+y=2, \ \ z^7-2z^6+5z^3-6z^2+11z-4x-6=0,$$
$$z^8-4z^7+4z^6+9z^4-24z^3+23z^2-28z+28=0 \quad (8)$$

and this "lexical Gröbner" form has the advantage that the last equation is univariate in z. Then by solving it for z, the other equations (once z is known) also become univariate, allowing a complete solve. This property is analogous to an "upper triangular" linear system.

We'll now describe what exactly a Gröbner basis is, how to find one, and some of the things you can do with it once you've found it. This material is also surveyed in [3] [2] [4]; I recommend the latter.

2.1 Suitable total orderings among multivariate polynomials

A total order "$\prec$" among multivariate monomials is "suitable" if

1. If $a \prec b$ then $ac \prec bc$ for every *monomial* c.
2. If $b \neq 1$ is a monomial then $a \prec ab$.

In other words, higher degree monomials are higher in the ordering, all other things being equal.

Assume some arbitrary total ordering $x_1, x_2, ..., x_n$ of the *n variables* in the polynomial has been fixed beforehand. (It has been suggested that a good one is to make the variables that occur most frequently in the original basis, come earliest.)

Here are three examples of suitable total orderings among monomials $x_1^{e_1}x_2^{e_2}...x_n^{e_n}$.

1. Lexicographic. If $e_1 > e_1'$ then the primed monomial is lexically smaller. If $e_1 < e_1'$, the primed monomial is lexically larger. If $e_1 = e_1'$ then the decision is based on e_2.
2. Total degree then lex. If $e_1+e_2+\cdots+e_n > e_1'+e_2'+\cdots+e_n'$ then the primed monomial is greater. Ties are broken using the lexicographic ordering.
3. Total degree then inverse lex. Same as above, but ties are broken using *reverse* lexicographic order.

Once we have a suitable total ordering on monomials, we always agree to write polynomials as the weighted sum of its monomials in decreasing $\prec$ order. Assume from here on we've agreed to some ordering and we've agreed to do this.

2.2 Reduction

A multivariate polynomial F is *reduced* with[3] respect to some basis set B if no term of f is divisible by the leading term of any element of B.

One can always change F to cause it to be reduced with respect to B (while still B and F generate the same ideal) as follows:

```
Loop through all terms T of F in decreasing order:
   if for some P in B, P's leading term divides T,
      then replace F by a linear combination of
      P and F designed to eliminate that term of F.
```

2.3 S-polynomials

Let the LCM (least common multiple) of the leading terms of two polynomials f and g be called h. If the leading term of f is $f_{\text{lead}} = x_1^{a_1}x_2^{a_2}...x_n^{a_n}$ and the leading term of g is $g_{\text{lead}} = x_1^{b_1}x_2^{b_2}...x_n^{b_n}$, then $h = x_1^{c_1}x_2^{c_2}...x_n^{c_n}$ where $c_j = \max(a_j, b_j)$.

Then the S-polynomial of f and g is

$$S(f,g) = -S(g,f) = \frac{h}{f_{\text{lead}}}f - \frac{h}{g_{\text{lead}}}g. \quad (9)$$

This has been cleverly designed to lie in the same ideal as f and g while cancelling the leading terms of f and g.

2.4 Gröbner basis

Definition: A basis B is *Gröbner* if every polynomial in the ideal generated by B reduces to zero with respect to B.

More useful equivalent definition: A basis B is *Gröbner* if each S-polynomial of any pair of polynomials from B reduces to zero with respect to B.

Buchberger's existence and uniqueness **theorem:** Every ideal has a unique minimal[4] Gröbner basis (with any fixed suitable monomial ordering).

Buchberger's **algorithm** for finding a minimal Gröbner basis: Start with some minimal basis. Choose an S-polynomial of a pair of polynomials from B which reduces to some nonzero polynomial R. (If no such S-polynomial exists, then B is already Gröbner and we are done.) Add R to the basis (i.e. $B \leftarrow B \cup R$) and continue on.

Buchberger's theorem: The algorithm above always terminates.

Of course, there are a large number of tricks to make Buchberger's algorithm run faster. For example, which S-polynomial do you want to do next (assuming there are several choices)? Some choices result in more speed than others! Buchberger suggests that the "greedy heuristic" of always considering $S(f,g)$ where h (that is, the LCM of the leading terms of f and g) has minimal total degree, is a good idea. Also, some monomial orderings result in a lot more speed than others – the ones based mainly on total degree tend to be much faster than mainly on lexicography.

[3] Reduceness is not transitive – it is possible for two polynomials to be reduced with respect to each other, for example y^3+1 and x^2y^2.

[4] Minimal under set inclusion, that is. Without the constraint of minimality, the entire ideal would be a legal Gröbner basis.

2.5 Uses of the Gröbner basis

Corollary of the uniqueness theorem (deciding equivalence): Two ideals are equivalent if and only if they have the same minimal Gröbner basis.

Thus we have our first amazing application of Gröbner bases – we can use them to decide equivalence of two polynomial ideals. One can also use them to solve a polynomial system:

Theorem: For a polynomial system with a finite number of complex solutions, the Gröbner basis in the *lexicographic* term ordering is "upper triangular" in the sense that there is a univariate equation in x_n, and once that is solved there is a a univariate equation in x_{n-1}, and so on.

and to decide if a polynomial system has a finite or infinite number of solutions:

Theorem: A polynomial system has a finite number of complex solutions if and only if each variable appears *alone* (such as x^7 has x appearing alone, but x^6y^4 does not) in some leading term of some polynomial in a minimal Gröbner basis (under any suitable term ordering – it doesn't have to be lexicographic).

Does a system have any solutions?

Theorem: A polynomial system has *no* complex solutions if and only if its Gröbner basis is the single polynomial $\{1\}$.

That was easy! We can also solve the membership problem:

Theorem: f is in the ideal generated by B, where B is Gröbner, if and only if f reduces to zero with respect to B.

Another application (among many): It is also possible to find the univariate polynomial of least degree in the ideal (if there are a finite number of solutions), in some sense leading to the "prettiest" algebraic numbers.

2.6 Computational complexity

In the worst case, it is known [10] [11] [12] that computing a Gröbner basis is EXPSPACE-complete – very nasty – but for small problems it is often feasible to find the Gröbner basis without running out of memory.

Indeed, despite the fact that the best resultant based methods decide existence in PSPACE, while computing a Gröbner basis is EXPSPACE-complete, the elegance of the Gröbner criterion, the rarity of the worst-case behavior, and the fact that reasonably large systems are going to be intractable using any known method, often make Gröbner methods the better choice in practice.

References

[1] S.S. Abhyankar: Historical ramblings in algebraic geometry, Amer. Math. Monthly 83, 6 (1976) 409-448.

[2] T.Becker & V.Weispfenning: Gröbner bases, Springer 1993.

[3] Stuart J. Berkowitz: On computing the determinant in mslal parallel time using a small number of processors, Information Proc. Let. 18 (1984) 147-150

[4] B.Buchberger: Groebner bases: An algorithmic method in polynomial ideal theory, 184-232 (chapter 6) in "Multidimensional systems theory," N.K. Bose ed. D. Reidel 1985.

[5] David Cox, John Little, Donal O'Shea: Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra, Springer.

[6] L. Csanky: Fast parallel matrix inversion algorithms, SIAM J. Computing 5,4 (1976) 618-623.

[7] A.L. Dixon: The eliminant of three quantics in two independent variables, Proc. London Math. Soc. 7 (1908) 50-69; 473-492.

[8] A.L. Dixon: A form of the eliminant of two quantics, Proc. London Math. Soc. 6 (1908) 468-478.

[9] I.M.Gelfand, M.M. Kapranov, A.V. Zelevinsky: Discriminants, resultants, and multidimensional determinants, Birkhauser 1994.

[10] D.Kapur, T.Saxena, L.Yang: Algebraic and geometric reasoning using Dixon resultants, Proc. Int'l Sympos. on Symbolic & Algebraic manipulation (1994) 99-107.

[11] F.S. Macaulay: Some formulae in elimination, Proc. London Math. Soc. 35 (1902) 3-27.

[12] F.S. Macaulay: Note on a resultant of a number of polynomials of the same degree, Proc. London Math. Soc. 21 (1921) 14-21.

[13] E.W.Mayr & A.E.Meyer: The complexity of the word problems for commutative semigroups and polynomial ideals, Adv. Math. 46, 3 (1982) 305-329.

[14] E.W.Mayr: Membership in polynomial ideals over Q is exponential space complete. In proceedings of STACS 1989.

[15] E.W.Mayr: On polynomial ideals, their complexity, and applications, pp. 89-105 in Fundamentals of computation theory 1995, Springer Lecture Notes in Computer Science #965.

[16] F. Morley: The eliminant of a net of curves, Amer. J. Math. 47 (1925) 91-97.

[17] F. Morley & AB Coble: New results in elimination, Amer. J. Math. 49 (1927) 463-488.

[18] J.Renegar: Computational complexity of solving real algebraic formulae. In Proceedings, International Congress of Mathematicians, pp. 1595-1606. Tokyo: Springer-Verlag 1991.

[19] H.S. White: Bezout's theory of resultants and its influence on geometry, Bull. Amer. Math. Soc. 15 (1909) 325-338