Chapter 7
Gröbner Bases

7.1 Statement of the Problem

In this lecture we shall consider (in a slightly vulgarized form, without rigorous mathematical terms) an important mathematical achievement of the second half of the last century—Gröbner bases, the Buchberger algorithm (which constructs them), and their applications (see [12, 13] for an introduction).

Suppose we have n variables $x_1, \ldots, x_n$. They are not independent, but satisfy some polynomial equations $p_1 = 0, \ldots, p_m = 0$ (p_j are polynomials of x_i). Let's consider some polynomial q of the same variables. It is natural to ask if this polynomial is equal to 0 due to the constraints on our variables or not. If there is another polynomial q_2, there is the question of their equality.

These questions would become very easy if we had an algorithm reducing polynomials of dependent variables to a canonical form. Two equal polynomials reduce to the same canonical form; a polynomial equal to 0 reduces to the canonical form 0.

We can try to use the equations $p_j = 0$ for simplifying the polynomial q, i.e., for replacing its more complicated terms by combinations of simpler ones. But to do so we first have to accept some convention in which terms are more complicated and which are more simple.

7.2 Monomial Orders

We need a total order of monomials (i.e., products of powers of the variables $x_1^{n_1} \cdots x_n^{n_n}$). An order is total if for any monomials s and t either $s < t$ or $s > t$ or $s = t$ is true. An order is admissible if two properties are satisfied:

- $1 \leqslant s$ for any monomial s.
- If $s < t$ then $s\,u < t\,u$ for any monomial u.

Three admissible orders are most popular.

A. Grozin, *Introduction to Mathematica® for Physicists*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-00894-3_7, © Springer International Publishing Switzerland 2014

Lexicographic

Anybody who has ever seen a dictionary knows what is lexicographic order. We are comparing two monomials: $s = x_1^{n_1} x_2^{n_2} \cdots x_n^{n_n}$ and $t = x_1^{m_1} x_2^{m_2} \cdots x_n^{m_n}$. If the degree of the main variable x_1 in s is larger than in t ($n_1 > m_1$), then $s > t$. If it is smaller ($n_1 < m_1$), then $s < t$. If $n_1 = m_1$, we compare the degrees of the next variable x_2: if $n_2 > m_2$, then $s > t$; if $n_2 < m_2$, then $s < t$; if $n_2 = m_2$, we compare the degrees of x_3; and so on.

By Total Degree than Lexicographic

First we compare the total degree $n = n_1 + n_2 + \cdots + n_n$ of the monomial s and the total degree $m = m_1 + m_2 + \cdots + m_n$ of the monomial t. If $n > m$ then $s > t$; if $n < m$ then $s < t$; if the total degrees are equal, we compare s and t lexicographically.

By Total Degree than Reverse Lexicographic

First we compare the total degrees. If they are equal, then we begin from the junior variable x_n: if its degree in s is larger than in t ($n_n > m_n$), then $s < t$; if it is smaller ($n_n < m_n$), then $s > t$; if $n_n = m_n$, we compare the degrees of the previous variable x_{n-1}; and so on, that is, this is (within some total degree) the reverse lexicographic order with respect to the reverse list of variables.

7.3 Reduction of Polynomials

Let's fix some admissible monomial order. We'll write polynomials in descending order: the leading term first, followed by the rest ones. We'll normalize all polynomials p_i in such a way that the coefficient of the leading term is 1. Now they can be used as substitutions which replace the leading term by minus sum of the remaining ones, that is, if some term of a polynomial q is divisible by the leading term of some polynomial p_i, we remove this leading term and insert minus sum of the remainder terms of p_i instead. This is called reduction of the polynomial q with respect to the set of polynomials p_i; if none of the substitutions is applicable, the polynomial q is called reduced. For example, let's consider a set of polynomials

In[1] := p1 = x^2 + y^2 − 1; p2 = x ∗ y − 1/4;

Let's try to reduce the polynomial

In[2] := q = x^2 ∗ y;

(we use the lexicographic order with $x > y$). This can be done in different ways.

Let's first reduce q with respect to p_1:

In[3] := PolynomialReduce[q, {p1}, {x,y}]

Out[3] = $\{\{y\}, y - y^3\}$

The result means that if we subtract the polynomial p_1 multiplied by y from q, then the reduced polynomial $y - y^3$ is obtained. This is what we are interested in.

In[4] := q1 = %[[2]]

Out[4] = $y - y^3$

Now let's reduce q with respect to p_2:

In[5] := PolynomialReduce[q, {p2}, {x,y}]

Out[5] = $\left\{\{x\}, \frac{x}{4}\right\}$

In[6] := q2 = %[[2]]

Out[6] = $\frac{x}{4}$

So, we have obtained two different results, q_1 and q_2. In fact they are equal due to $p_1 = 0$ and $p_2 = 0$, but this is not evident. Every time when more than one substitution can be applied to a term of a polynomial q (in this particular case, we can replace either x^2 or xy in $x^2 y$), a fork appears; maybe, its branches join later, but maybe, they don't (as in this case).

A set of polynomials $p_1, \ldots, p_n$ is called a **Gröbner basis** (for a given monomial order) if reduction of any polynomial q with respect to this set is unique.

This definition is not constructive: it does not say how to check if a given set of polynomials forms a Gröbner basis. Presently we shall formulate Buchberger algorithm which transforms a set of polynomials (constraints on variables) into an equivalent system of constraints which is a Gröbner basis.

7.4 S-Polynomials

In our example, the constraints $p_1 = 0$ and $p_2 = 0$ allow us to simplify the monomials x^2 and xy. Do these constraints contain an extra information usable for simplification but not obvious? Yes, they do! Let's multiply p_1 and p_2 by monomials (i.e., products of powers of variables) in such a way that their leading terms become identical (equal to the least common multiple of the leading terms of p_1 and p_2). Then we subtract the second polynomial from the first one. The leading terms cancel, and we get a new polynomial with a new leading term which can be used for simplifying terms in q (because this new polynomial also vanishes). This polynomial is called the S-polynomial $S[p_1, p_2]$ (from the word subtraction). In our example

In[7] := S = Expand[y * p1 − x * p2]

Out[7] = $\frac{x}{4} - y + y^3$

This polynomial can be added to the system of constraints $p_1 = 0$, $p_2 = 0$. Let's normalize its leading coefficient to 1:

In[8] := p3 = Expand[4 * S]

Out[8] = $x - 4y + 4y^3$

In[9] := Clear[S]

Now we have a new possibility for reduction:

In[10] := PolynomialReduce[q2, {p3}, {x, y}]

$$\text{Out}[10] = \left\{\left\{\frac{1}{4}\right\}, y - y^3\right\}$$

Now we've got the same result q_1. The polynomials $\{p_1, p_2, p_3\}$ form a Gröbner basis. This set can be simplified by reducing them with respect to each other:

In[11] := PolynomialReduce[p1, {p3}, {x, y}]

$$\text{Out}[11] = \left\{\left\{x + 4y - 4y^3\right\}, -1 + 17y^2 - 32y^4 + 16y^6\right\}$$

In[12] := p1a = Expand[%[[2]]/16]

$$\text{Out}[12] = -\frac{1}{16} + \frac{17y^2}{16} - 2y^4 + y^6$$

In[13] := PolynomialReduce[p2, {p3}, {x, y}]

$$\text{Out}[13] = \left\{\{y\}, \frac{1}{4}\left(-1 + 16y^2 - 16y^4\right)\right\}$$

In[14] := p2a = Expand[-%[[2]]/4]

$$\text{Out}[14] = \frac{1}{16} - y^2 + y^4$$

In[15] := PolynomialReduce[p1a, p2a, {x, y}]

$$\text{Out}[15] = \left\{\left\{-1 + y^2\right\}, 0\right\}$$

The polynomial p_{1a} reduces to 0, and hence it can be excluded from the system of constraints on our variables x, y. The polynomials p_{2a} and p_3 form a reduced Gröbner basis (with respect to the lexicographic order with $x > y$). Reduced Gröbner basis is unique (for a given monomial order), if we accept the convention that the coefficients of the leading terms are 1.

7.5 Buchberger Algorithm

Generalizing this example, we can formulate an algorithm for construction of the Gröbner basis of a set of n polynomials $P = \{p_i\}$:

1. $S = \{$the set of pairs (p_i, p_j) of these polynomials with $i < j \leqslant n\}$
2. **while** S is not empty
3. choose and remove some pair (p_i, p_j) from S;
4. calculate the S-polynomial $S[p_i, p_j]$;
5. reduce it with respect to P;
6. if the result is not 0, add this polynomial to P,
   and the corresponding pairs to S.

The set of pairs S alternatingly shrinks and grows. But it can be proved that this process terminates after a finite number of steps and produces a Gröbner basis P. Reducing these polynomials with respect to each other and throwing zeros away, one can get the reduced Gröbner basis. Some variations can improve the efficiency of the algorithm. For example, when adding a new polynomial to the set P, we can reduce all polynomials already in P with respect to the new one; if some of

them changes, reduce other ones with respect to them, and so on (throwing zeros away while doing so). The order in which pairs are selected from the set S is very important—a good choice can reduce the amount of computations drastically.

Let's ask *Mathematica* to construct the Gröbner basis for the system $\{p_1, p_2\}$ with respect to the lexicographic order with $x > y$:

In[16] := B = GroebnerBasis[{p1,p2},{x,y}]

Out[16] = $\{1 - 16y^2 + 16y^4, x - 4y + 4y^3\}$

Let's reduce the polynomial q to the canonical form, i.e., reduce it with respect to the Gröbner basis (the result is unique).

In[17] := PolynomialReduce[q,B,{x,y}]

Out[17] = $\left\{\left\{-\frac{x}{4}, \frac{1}{4} + xy\right\}, y - y^3\right\}$

It is difficult to predict the complexity of the Buchberger algorithm. In worst cases it can be very high, i.e., constructing the Gröbner basis of a moderately large system can require a huge amount of calculations. The complexity strongly depends on the monomial order being used. In the case of ordering by the total degree (and then something) reduction tries to lower the total degree of a polynomial. The number of possible terms in a polynomial of a low total degree is small. In the case of the lexicographic order, a polynomial of y of an arbitrarily large degree is considered simpler than x to the first power. Therefore reduction does not lower the number of terms in a polynomial as strongly as in the case of total-degree orders, and the complexity of Gröbner basis calculations is higher. On the other hand, a reduced Gröbner basis with respect to a lexicographic order provides more information useful for solving the system, as we shall see soon. *Mathematica* knows how to construct Gröbner bases with respect to monomial orders we discussed.

In[18] := B = GroebnerBasis[{p1,p2},{x,y},
MonomialOrder → DegreeLexicographic]

Out[18] = $\{-1 + 4xy, -1 + x^2 + y^2, x - 4y + 4y^3\}$

In[19] := PolynomialReduce[q,B,MonomialOrder → DegreeLexicographic]

Out[19] = $\left\{\left\{\frac{x}{4}, 0, 0\right\}, \frac{x}{4}\right\}$

In[20] := Clear[p1,p2,p3,p1a,p2a,q,q1,q2,B]

7.6 Is the System Compatible?

Consider the system

In[21] := p1 = x^2 * y + 4 * y^2 − 17; p2 = 2 * x * y − 3 * y^3 + 8;
p3 = x * y^2 − 5 * x * y + 1;

Let's construct its Gröbner basis—an equivalent system of equations.

In[22] := GroebnerBasis[{p1,p2,p3},{x,y,z}]

Out[22] = $\{1\}$

This system contains the equation $1 = 0$. This means that it has no solutions. If the Gröbner basis contains 1, the system is incompatible. The inverse statement can be

also proved—the Gröbner basis of an incompatible system always contains 1 (if we normalize all leading coefficients to 1; otherwise, just some nonzero constant).

In[23] := Clear[p1, p2, p3]

7.7 Gröbner Bases with Respect to Lexicographic Order

Reduction with respect to the lexicographic order first of all tries to lower the degree of the main variable (x in our examples), and if possible, down to 0. Therefore usually there is a subset of polynomials in a reduced Gröbner bases which don't contain x. When x is absent, reduction tries to lower the degree of y, and if possible, down to 0. Therefore usually among these polynomials there are those which don't contain y, and so on. In other words, a lexicographic Gröbner bases has a triangular structure. For example,

In[24] := B = GroebnerBasis[{x^2 + y^2 + z^2, x + y − z, y + z^2}, {x, y, z}]

Out[24] = $\{z^2 + z^3 + z^4, y + z^2, x - z - z^2\}$

The polynomial

In[25] := p1 = B[[1]]

Out[25] = $z^2 + z^3 + z^4$

depends only on the most junior variable z. This means that projections of all solutions of our system on the z axis form a finite set of points—roots of this equation. In our example, they are $z = 0$ and

In[26] := p1 = Expand[p1/z^2]; s = Solve[p1 == 0, z]

Out[26] = $\left\{\left\{z \to -(-1)^{1/3}\right\}, \left\{z \to (-1)^{2/3}\right\}\right\}$

In[27] := z1 = ComplexExpand[z/.s[[1]]]

Out[27] = $-\frac{1}{2} - \frac{i\sqrt{3}}{2}$

In[28] := z2 = ComplexExpand[z/.s[[2]]]

Out[28] = $-\frac{1}{2} + \frac{i\sqrt{3}}{2}$

Substituting any of these z values to

In[29] := p2 = B[[2]]

Out[29] = $y + z^2$

we find the corresponding y value. Substituting these z and y into

In[30] := p3 = B[[3]]

Out[30] = $x - z - z^2$

we find the corresponding x value. Thus solving any system of polynomial equations with several unknowns reduces to solving single-variable polynomial equations sequentially, thanks to lexicographic Gröbner bases. Even when some of them cannot be solved in radicals, it is easy to solve them numerically to any desired precision.

In[31] := Clear[B, p1, p2, p3, z1, z2]

And here is another example.

In[32] := B = GroebnerBasis[{x^2 − 2 ∗ x ∗ y + 2 ∗ y^2 − 1, x ∗ y − y ∗ z + z^2 − 1, x ∗ z + y^2 − y ∗ z − 1}, {x, y, z}]

Out[32] $= \{1 - y^2 - 2z^2 + y^2z^2 + z^4, -y + y^3 + z - y^2z + yz^2 - z^3, x - y - 2z + y^2z + z^3\}$

Now we have no equations with a single variable z; there are 2 equations containing z and y:

In[33] := p1 = Factor[B[[1]]]

Out[33] $= (-1 + z)(1 + z)\left(-1 + y^2 + z^2\right)$

In[34] := p2 = Factor[B[[2]]]

Out[34] $= (y - z)\left(-1 + y^2 + z^2\right)$

The common set of their solutions is the circle $y^2 + z^2 = 1$. Substituting a point on this circle into

In[35] := p3 = B[[3]]

Out[35] $= x - y - 2z + y^2z + z^3$

we find the corresponding x value, that is, the set of solutions of this system is one-dimensional.

In[36] := Clear[B, p1, p2, p3]

For solving a system of polynomial equations it is useful to construct its Gröbner basis and then to factorize its elements.

7.8 Is the Number of Solutions Finite?

Gröbner bases with respect to other monomial orders don't have such simple triangular structure. But any Gröbner basis can tell us not only if the system is compatible but also if the number of its solutions is finite. Let's consider the same examples.

In[37] := GroebnerBasis[{x^2 + y^2 + z^2, x + y − z, y + z^2}, {x, y, z}, MonomialOrder → DegreeLexicographic]

Out[37] $= \{x + y - z, y + z^2, -y + y^2 - yz\}$

The leading terms of the polynomials forming this basis are x, z^2, and y^2. What is the dimensionality of the space of polynomials which cannot be reduced with respect to this basis? Only monomials which are not divisible by these leading terms cannot be reduced, namely, 1, y, and z. So the space of polynomials reduced to the canonical form is three-dimensional for our system of constraints on the variables. Therefore our system has 3 solutions (there explicit form can be obtained more easily from the lexicographic Gröbner basis, as we have seen).

If each variable raised to some power is the leading term of some element of a Gröbner basis, then any monomials with this (or higher) degree of this variable are reducible. Irreducible monomials are inside the parallelepiped bounded by these powers, and their number is finite. Therefore the space of polynomials reduced to the canonical form is finite-dimensional, and the system has a finite number of solutions.

And here is our second example:

In[38] := GroebnerBasis[{x^2 − 2 ∗ x ∗ y + 2 ∗ y^2 − 1, x ∗ y − y ∗ z + z^2 − 1, x ∗ z + y^2 − y ∗ z − 1}, {x, y, z}, MonomialOrder → DegreeLexicographic]

Out[38] $= \{-1 + y^2 + xz - yz, -1 + xy - yz + z^2, -3 + x^2 + 2y^2 - 2yz + 2z^2, x - y - 2z + y^2 z + z^3, x - 2y + y^3 - z + yz^2\}$

The leading terms are xz, xy, x^2, $y^2 z$, and y^3. Among them there are powers of x and of y, but not of z. Therefore the space of polynomials in the canonical form (i.e., reduced with respect to this basis) is infinite-dimensional. This space contains, e.g., the directions 1, z, z^2, $z^3 \ldots$ (and not only them). This means that the set of solutions of our system is infinite.

So, the criterion works in the opposite direction, too. If there exists a variable no power of which appears as the leading term of some element of the Gröbner basis (not being multiplied by some other variable), then all powers of this variable are irreducible, and the space of polynomials in the canonical form is infinite-dimensional. And hence the set of solutions of the equation system is infinite.

Knowing the reduced Gröbner basis (for any monomial order) one can also find the dimensionality of the set of solutions [14]. Consider sets of variables satisfying the following condition: none of the leading terms of the elements of the basis is a product of powers of these variables. The number of variables in the longest set gives the dimensionality of the set of solutions. In our example there is just one such set—$\{z\}$. Therefore the set of solutions of this system is one-dimensional.