Simplification of Algebraic Equations OR A Smooth Introduction to Gröbner Bases

Nikolay N. Vassiliev
St. Petersburg Department of Steklov Mathematical Institute
Russian Academy of Sciences
St. Petersburg, Russia
vasiliev@pdmi.ras.ru

Translated and adapted from Russian by

Alkiviadis G. Akritas
Department of Electrical and Computer Engineering
University of Thessaly
Volos, Greece
akritas@uth.gr

1 To simplify equations reduce their degree

Quite early in school we begin dealing with equations in one or more variables. Some of them, for example equations in one variable of first and second degree, are easily solved. Many may remember how complex are the formulae to find the roots of equations in one variable of degree three and four; they were discovered by the Italian mathematicians Gerolamo Cardano and Niccolò Fontana Tatraglia. As for the roots of equations of degree five or higher, they cannot be expressed as radicals of any degree. This was first proved by Paolo Ruffini, for the fifth degree equations, and was later generalized to equations of any degree by Niels Henrik Abel and Évariste Galois.

Since there are no simple formulae for the roots of equations, it might seem at first that it is hopeless to search for solutions of systems of algebraic equations. Fortunately, this is just the first impression. In fact, in the case of equations in one variable, a lot is known about the roots of an arbitrary algebraic equation. In practice there is no need to express the roots as radicals; sometimes only the number of the real roots is needed and at other times the roots need to be computed to a certain degree of accuracy. For an arbitrary equation of the form $p(x) = 0$, all these problems can be successfully solved.

According to the fundamental theorem of Algebra, a polynomial of degree n, has exactly n roots, multiplicities counted. If only the real roots are needed,

then there exist algorithms to effectively compute them to any degree of accuracy. There also exist exact formulae for all the roots of an arbitrary equation in one variable of arbitrary degree; these formulae do not use radicals, but rather the more complex non-elementary Weierstrass' θ-functions.

In the sequel a method is presented for dealing with arbitrary systems of polynomial equations in several variables; however, what is needed first is a way to simplify these polynomials.

2 Division of a polynomial by another polynomial

Below will be examined systems of polynomial equations of the form

$$\begin{cases} p_1(x_1, x_2, \ldots, x_n) = 0 \\ p_2(x_1, x_2, \ldots, x_n) = 0 \\ \qquad\qquad \vdots \\ p_m(x_1, x_2, \ldots, x_n) = 0 \end{cases}.$$

The number of equations can be arbitrary and is in no way related with the number of variables; in principle, the system can have an infinite(!) number of equations.

To see if the problem is simpler in the case of polynomials in one variable, the following system is simplified first:

$$\begin{cases} p = x^5 - 2x^4 + 5x^3 + 2x^2 - 4x + 10 = 0 \\ q = x^3 - x^2 + 3x + 5 = 0 \end{cases}.$$

It is not difficult to see that, in this example, the first polynomial, p, can be simplified if the second one, q, multiplied times x^2, is subtracted from it; in other words,

$$p - x^2 q = -x^4 + 2x^3 - 3x^2 - 4x + 10 = x^4 - 2x^3 + 3x^2 + 4x - 10 = 0.$$

The resulting equation of fourth degree (with positive coefficient of x^4) can be also simplified in the same way — subtracting from it the second equation, q, multiplied times x. Applying the same method a third time the resulting second degree equation is:

$$x^2 - 2x + 5 = 0.$$

The simplification procedure described above is continued with the following new system of polynomials — where p is replaced by q and q is replaced by $x^2 - 2x + 5 = 0$:

$$\begin{cases} p = x^3 - x^2 + 3x + 5 = 0 \\ q = x^2 - 2x + 5 = 0 \end{cases}.$$

However, in this case the result is unexpectedly $0 = 0$. That means that the polynomial $x^3 - x^2 + 3x + 5$ disappeared and there remained only the second degree polynomial $x^2 - 2x + 5$, whose roots can be easily computed.

Paying closer attention to the above example it is obvious that two polynomial divisions took place; the first time the polynomial $p(x)$ was divided by $q(x)$ and the remainder was $x^2 - 2x + 5$, whereas the second time the polynomial $q(x)$ was divided by $x^2 - 2x + 5$ and the remainder was 0.

Polynomial division can be performed in a way analogous to the (European style) long division of integers as shown below (see the Wikipedia article on *Long division* for the USA style of long division):

$$
\begin{array}{rrrrrr|l}
x^5 & -2x^4 & +5x^3 & +2x^2 & -4x & +10 & x^3 - x^2 + 3x + 5 \\
\cline{7-7}
x^5 & -x^4 & +3x^3 & +5x^2 & & & x^2 - x + 1 \\
\cline{1-4}
 & -x^4 & +2x^3 & -3x^2 & -4x & & \\
 & -x^4 & +x^3 & -3x^2 & -5x & & \\
\cline{2-5}
 & & x^3 & & +x & +10 & \\
 & & x^3 & -x^2 & +3x & +5 & \\
\cline{3-6}
 & & & x^2 & -2x & +5 &
\end{array}
$$

The results of these hand computations can always be verified using the function `quorem(p,q,x)` of the free computer algebra system **Xcas**.

In a certain sense, polynomial division is simpler than integer division. This is because the leading term of the first polynomial is divided by the leading term of the second polynomial, the resulting quotient is multiplied times the whole divisor and the product is subtracted from the dividend; this process is repeated until the remainder polynomial is of degree smaller than that of the divisor.

It is obvious that any two polynomial equations in one variable can be replaced by one. From this it follows that if the above simplification method is applied to a system of polynomials equations in one variable it will always lead the system to a single equation.

The simplification method described above is called the *Euclidean algorithm* and computes the *greatest common divisor (gcd)* of two polynomials.

In summary, it has been shown that a system of polynomial equations in one variable is always equivalent to a single equation. Replacing the whole system by one equation, all its solutions can be easily found. More precisely, it is the exact number of the complex roots (of the system) that has been computed — which is equal to the degree of the (equivalent) equation that is obtained as a result of the simplification process. It is also quite possible that the gcd of a system of polynomials in one variable is 1, that is, a nonzero constant. Then the system under examination is equivalent to the equation $1 = 0$, which means it does not have any solutions. In this case the system is called *inconsistent*.

3 The case of two or more variables: The result is not unique

In what follows the general case is dealt with — that is systems of equations in several variables. It turns out that the same simplification method can be applied here as well, even though certain complications arise. To examine a concrete example, consider the system of equations:

$$\begin{cases} p = x^3y + y^2 - 1 = 0 \\ q = x^3 + 3y - 1 = 0 \end{cases}.$$

It is obvious that the term x^3y can be easily eliminated

$$p - y * q = -2y^2 + y - 1 = 0.$$

Replacing p by q and q by $-2y^2 + y - 1 = 0$ results in the equivalent system

$$\begin{cases} p = x^3 + 3y - 1 = 0 \\ q = -2y^2 + y - 1 = 0 \end{cases},$$

which is easily solved since q is a second degree polynomial equation *only* in y.

In the case of systems of polynomials in one variable, the simplification consists of reducing their degree. However, in the general case of two or more variables, the situation is a bit more complicated: reducing the degree in variable x may increase the degree in variable y, and vice-versa. A way out of this situation could be found if it was known — as in the case of univariate polynomials — which of the terms of a polynomial is the leading one. This leading term could then be eliminated using the leading term of another polynomial. In the sequel the term of a polynomial *without its numerical coefficient* will be called *monomial*.

To order monomials what is needed first is an ordering among variables. There are $n!$ ways to order a set of n variables; for the example that follows below, in the variables x and y, the ordering chosen is according to the following *ranking* (the symbol $\prec$ means *precedes*), $1 \prec a \prec b, \ldots, \prec x \prec y \prec z$, in which case, variable y has higher ranking than variable x. This ordering of the variables is called *lexicographical* because it follows the way in which words are ordered in a dictionary (lexico, in Greek) or encyclopedia; namely, letters are written in alphabetical order and words are ordered first according to the first letter, then according to the second letter etc.

Having ordered the variables, to determine which of two monomials — in a given polynomial — is the leading one, compare the degrees of the variable with the highest ranking (*leading variable*): If the degree of the leading variable in the first monomial is greater, then this is the leading monomial; and if the degrees of the leading variable in the two monomials are the same, then compare the degrees of the variable with ranking one less than the highest, etc.

The advantage of ordering monomials is that it is now possible to define the leading term in every polynomial of a given system of equations. Having defined the leading term of each polynomial in the system one can test whether the leading term of a given polynomial p is divided by one of the leading terms of the other polynomials in the system and if the answer is yes, to simplify the polynomial p. Obviously, with this simplification the leading term of p is reduced and eventually a system of equations is obtained where (pairwise) the leading terms of the polynomials do not divide each other. In this case, is it possible that the system of equations cannot be further simplified? A simple example will show that this is not so. Consider the system of equations

$$\begin{cases} y^5 + 5x^3y^3 + 2 = 0 \\ x^2y + 3x + y + 1 = 0 \end{cases} .$$

Keeping in mind that y is the variable with the higher ranking, the leading terms of the two polynomials can be found either by simple inspection (after some practice) or with the help of the function `lcoeff(poly,[y,x])`[1] of the free computer algebra system **Xcas** — notice that the variables are written as `[y,x]` to reflect the chosen ordering, where variable y has higher ranking than variable x; either way, the leading terms (monomials) are y^5 and x^2y and none divides the other. However, in the first equation there is the term $5x^3y^3$ which can be eliminated — even though it is not the leading term — since it is divisible by the leading monomial x^2y of the second equation. Indeed, subtracting the second polynomial multiplied times $5xy^2$, from the first polynomial, leads to the system of equations

$$\begin{cases} y^5 - 5xy^3 - 15x^2y^2 - 5xy^2 + 2 = 0 \\ x^2y + 3x + y + 1 = 0 \end{cases} .$$

Here again — in the first equation — there is the term $-15x^2y^2$ which can be eliminated since it is divisible by the leading monomial x^2y of the second equation. Indeed, subtracting — from the first polynomial — the second polynomial multiplied times $-15y$ leads to the system of equations

$$\begin{cases} y^5 - 5xy^3 - 5xy^2 + 15y^2 + 45xy + 15y + 2 = 0 \\ x^2y + 3x + y + 1 = 0 \end{cases} ,$$

which cannot be further simplified in this way.

It is now time to introduce new terminology. Having fixed an ordering on the monomials, one says that a polynomial p can be *reduced* with the help of

[1] The function `lcoeff(poly,[y,x])` has a third argument to indicate the monomial ordering; by default it is the lexicographical order. The monomial orderings available in **Xcas** are discussed in Section 6.

(or modulo) the set of polynomials $q_1, q_2, \ldots, q_n$ if the leading term of p can be divided by the leading term of one of the polynomials q_i. The process of simplifying polynomials by eliminating the leading term, as described above, is called *reduction*. If p_1 is the result of the reduction of polynomial p modulo the set of polynomials $q_1, q_2, \ldots, q_n$ then one says that the polynomial p reduces to p_1 with the help of (or modulo) the set of polynomials $q_1, q_2, \ldots, q_n$.

It follows easily that since the process of reducing the leading terms cannot go on for ever, one can always use the reduction process to reduce any system of equations to an equivalent one where the leading terms, pairwise, do not divide one another. This whole process can be considered a generalization of the Euclidean division algorithm for polynomials in one variable. The only difference is that now the polynomial p is divided not by a single polynomial but by a whole set of polynomials $q_1, q_2, \ldots, q_n$ simultaneously.

The process of reduction - division can be done exactly like the long division. In the European style of division, the divisors are all written on the right, and are separated by comma. Each subtraction in this scheme corresponds to a reduction with a certain degree of freedom. Namely, as mentioned above, since the dividend (the polynomial p) is to be reduced with the help of the leading term of any divisor (polynomial) $q_1, q_2, \ldots, q_n$, it may well turn out that p can be reduced modulo the leading terms of *several* polynomials q_i. In cases like this, an arbitrary choice is made as to which of the polynomials $q_1, q_2, \ldots, q_n$ should be used to reduce p. Due to this freedom of choice, the result of these reductions depends on the polynomials $q_1, q_2, \ldots, q_n$ used. The only thing that is guaranteed is that, at the end of the process, the resulting polynomial cannot be further reduced with the help of the polynomials $q_1, q_2, \ldots, q_n$. By analogy with the division of polynomials in one variable, the resulting polynomial could be called the *remainder* of the division of p by the set of polynomials $q_1, q_2, \ldots, q_n$.

In **Xcas** the process of reduction - division can be computed with the help either of the function `greduce(p,[q1,...,qk],[x1,...,xl])`, or of the function `rem(p,[q1,...,qk],[x1,...,xl])`[2], where in both cases the variables in the third argument are written in the chosen ordering. The set of polynomials q_i, in the second argument, can contain a single element, in which case a single reduction is performed.

Example. To demonstrate the reduction - division process consider the polynomial $p = x^2y - y$ which is to be divided by the polynomials $q_1 = xy - x$ and

[2] For **Xcas** version 0.9.8 or higher. For operations related to Gröbner bases (defined in Sections 5 and 9) **Xcas** uses either built-in functions or functions from the library of the computer algebra system **CoCoA**, which specializes in **Co**mputations in **Co**mmutative **A**lgebra. Inserting to the corresponding commands the optional argument `with_cocoa=true` or `with_cocoa=false` determines which functions will be used. At the time of writing this article (summer 2012), functions from the **CoCoA** library are used by default; this might change in the future.

$q_2 = x^2 - y$; here again $x \prec y$, that is, variable y has higher ranking than variable x. If p is reduced by q_1 the remainder is $-y + x^2$, which, in turn, can be reduced by q_2 to obtain the final remainder 0. In **Xcas** the corresponding command is `greduce(p,[q1,q2],[y,x])` or `rem(p,[q1,q2],[y,x])`; notice that the order of the polynomials is $[q_1, q_2]$, which means divide p by q_1 first, if that is possible. However, if p is reduced by q_2 the remainder is $x^4 - y$, which, in turn, can be reduced again by q_2 to obtain the final remainder $x^4 - x^2$, which cannot be further reduced neither by q_1 nor by q_2. In **Xcas** the corresponding command is `greduce(p,[q2,q1],[y,x])` or `rem(p,[q2,q1],[y,x])`; notice the order of the polynomials is now $[q_2, q_1]$.

4 Ideals and their relation to systems of polynomials

As was indicated above, a system of the form

$$\begin{cases} p_1(x_1, x_2, \ldots, x_n) = 0 \\ p_2(x_1, x_2, \ldots, x_n) = 0 \\ \qquad\qquad \vdots \\ p_m(x_1, x_2, \ldots, x_n) = 0 \end{cases}$$

may be brought into such a form that no further reduction of leading terms is possible. Is this the simplest form of an arbitrary system? Unfortunately, things are again a bit more complicated. Consider the system

$$\begin{cases} x^2y - xy + 1 = 0 \\ x^3 - y^2 + 1 = 0 \end{cases},$$

where the variables have been ordered as before, with y having higher ranking than x.

It is easily seen that in this example no reduction is possible because the leading terms are x^2y and $-y^2$ and none divides the other. However, this does not mean that it is impossible, in principle, to simplify this system after making certain other transformations. Indeed, multiply the first equation times y, the second times x^2 and subtract one from the other to get $-x^5 + 2x^2y^2 - x^2 - xy^2 + y = 0$ with leading term $2x^2y^2$. This last equation may be reduced with the help of the first equation of the system; application of the reduction method yields the equation $-x^5 - x^2 + xy^2 - y = 0$ with leading term xy^2, and which can be also directly derived from the original system of equations. Again, the last equation may be reduced with the help of the second equation of the system; application of the reduction method yields the equation $x^5 - x^4 + x^2 - x + y = 0$ with leading term y, which cannot be further reduced.

What actually happened here? A system of polynomial equations was given that could not be reduced with the help of mutual reductions of the equations. Next, the equation $-x^5+2x^2y^2-x^2-xy^2+y=0$ was derived from the original system, which equation was reduced to $x^5-x^4+x^2-x+y=0$ with the help of the system equations.

In other words, together with the original system of equations

$$\begin{cases} p_1(x_1,x_2,\ldots,x_n)=0 \\ p_2(x_1,x_2,\ldots,x_n)=0 \\ \qquad\qquad \vdots \\ p_m(x_1,x_2,\ldots,x_n)=0 \end{cases} \tag{1}$$

it was considered necessary to examine all the equations that could possibly be derived from it. Thus, any equation $p_i(x_1,x_2,\ldots,x_n)=0$ can be multiplied times an arbitrary polynomial $q(x_1,x_2,\ldots,x_n)=0$ to yield the equation $q\cdot p_i=0$ as a result. In an analogous way, adding any two equations results in an equation which is derived from the two equations that were added. Examine now all the equations that can be derived from an arbitrary system of equations of type (1).

Define by J the set of all possible polynomials p derived from a system of equations of type (1). Then it is easily seen that together with any two polynomials p_1 and p_2, the set J also contains their sum, and together with any polynomial p it also contains all its multiples $q\cdot p$. Such sets are called *polynomial ideals*.

It is now obvious that every system of polynomial equations generates a polynomial ideal. The polynomial ideal, itself, can be considered a system with an infinite number of polynomial equations. It turns out that one can easily learn to work with such infinite systems, which are all equivalent to systems with a finite number of equations. However, this is the essence of an important theorem by Hilbert and will be proved in the sequel.

At this point it can be stated that the basic idea of polynomial simplification is the foundation of the theory on Gröbner bases.

5 Reduction to the simplest, or canonical form

Take another look at the system of algebraic equations

$$\begin{cases} p_1(x_1,x_2,\ldots,x_n)=0 \\ p_2(x_1,x_2,\ldots,x_n)=0 \\ \qquad\qquad \vdots \\ p_m(x_1,x_2,\ldots,x_n)=0 \end{cases}$$

and along with it consider the whole ideal J of polynomials, which are generated by the polynomials $p_1,p_2,\ldots,p_m$.

Suppose that somehow the original system of equations can be brought to the form

$$\begin{cases} g_1(x_1, x_2, \ldots, x_n) = 0 \\ g_2(x_1, x_2, \ldots, x_n) = 0 \\ \qquad\qquad \vdots \\ g_r(x_1, x_2, \ldots, x_n) = 0 \end{cases},$$

which is considered simpler. The only requirement of such a system is that any equation derived from the original system, that is any polynomial belonging to the ideal J, may be simplified with the help of the set of polynomials $g_1, g_2, \ldots, g_r$.

Definition. The set of polynomials g_i is a *Gröbner basis* of the polynomial ideal J if the leading term of any polynomial belonging to the ideal J can be divided by one of the leading terms of the polynomials $g_1, g_2, \ldots, g_r$.

The definition is definitely non-constructive, since it is impossible to check this condition for the infinite number of polynomials in J. Despite this fact, it is possible to prove some very remarkable properties of these bases.

6 Properties of Gröbner bases

Suppose that somehow a Gröbner basis $g_1, g_2, \ldots, g_r$ has been found (using, for example, the function `gbasis([`p_1`,`p_2`,...,`p_k`],[`x_1`,`x_2`,...,`x_l`])`[3] of **Xcas**) for the polynomial ideal J. It will be shown that it is now possible to easily check whether a given polynomial p belongs to the ideal J or not. In other words, for any arbitrary equation it is possible to check whether it is derived from a given system of equations. This is known as the *ideal membership problem*.

As stated previously, there exists an algorithm for dividing an arbitrary polynomial p by a set of polynomials, an operation which is equivalent to a sequence of reductions (`greduce(p,[`q_1`,`q_2`,...,`q_k`],[`x_1`,`x_2`,...,`x_l`])` or `rem(p,[`q_1`,`q_2`,...,`q_k`],[`x_1`,`x_2`,...,`x_l`])` in **Xcas**; see also Section 3 for comments regarding this function). Apply this algorithm to the polynomial p and the set of polynomials $g_1, g_2, \ldots, g_r$ and denote by res the resulting polynomial. Clearly, res cannot be further reduced (simplified) with the help of the polynomials $g_1, g_2, \ldots, g_r$.

Assume now that p belongs to the ideal J. The result of each reduction also belongs to the ideal since the polynomial that was subtracted each time was a multiple of some g_i. Therefore, the final result res also belongs to the ideal J. However, by the definition of Gröbner bases, the leading term of every

[3] This is equivalent to `gbasis([`p_1`,`p_2`,...,`p_k`],[`x_1`,`x_2`,...,`x_l`],with_cocoa=true)`; that is, it uses the library of the **CoCoA** system.

polynomial in the ideal J is divided by one of the leading terms of $g_1, g_2, \ldots, g_r$, which implies that the polynomial res has to be reducible. Therefore, res is equal to zero, and the following has been proved:

Theorem. A polynomial p belongs to an ideal J if and only if on dividing p by the set of polynomials $g_1, g_2, \ldots, g_r$ of the Gröbner basis of J the remainder is equal to zero.

Notice that whereas the definition of Gröbner bases is non-constructive, the algorithm for solving the *ideal membership problem* is constructive — provided the polynomials g_i form a Gröbner basis of the ideal J.

Another remarkable property of the set of polynomials which are a Gröbner basis is that, on dividing any polynomial by the set of polynomials in the basis, the remainder is uniquely defined — despite the fact that in the division process the reductions may be executed in any order.

To see this, suppose that the polynomial p is reduced in two different ways to the polynomials res_1 and res_2, which cannot be further reduced with the help of the polynomials g_i. Since in the reduction process multiples of the polynomials g_i are subtracted from p, the following equalities hold: $p - res_1 = \sum_{k=1}^{r} f_i \cdot g_i$ and $p - res_2 = \sum_{k=1}^{r} h_i \cdot g_i$.

Therefore, the difference $res_1 - res_2$ belongs to the ideal J and, if $res_1 - res_2 \neq 0$, then the monomial of the leading term of this difference will be reducible with the help of the polynomials g_i — by the definition of Gröbner bases; moreover, this monomial must exist either in res_1 or in res_2, which implies that either res_1 or res_2 is reducible. Hence, $res_1 - res_2 = 0$.

Notice that so far it has not not been required that one should not be able to simplify the system $g_1, g_2, \ldots, g_r$ itself. Such a stronger condition leads to the concept of the *reduced* Gröbner basis.

Definition. The Gröbner basis $g_1, g_2, \ldots, g_r$ of the ideal J is called *minimal*, if all its leading coefficients are 1 and its leading terms pairwise cannot divide one another.

Definition. The Gröbner basis $g_1, g_2, \ldots, g_r$ of the ideal J is called *reduced*, if all its leading coefficients are 1 and none of the monomials of the polynomial g_i can be divided by any of the leading terms of the other polynomials in the basis.

It turns out that for any polynomial ideal there exists a *unique* reduced Gröbner basis. That is, any system of polynomials, even an infinite one, may be reduced to a unique canonical form.

This uniqueness depends on the monomial ordering that has been established;

and there is an infinite number of such orderings. Even in the lexicographical order, one can change the ranking of the variables, considering that variable x has higher ranking than y; that is, if the symbol $\succ$ is read as *is followed by*, then the ordering is $1 \succ a \succ b, \ldots, \succ x \succ y \succ z$. Another monomial ordering would be to consider not the degree of the leading variable but the *total degree* (the sum of the degrees of the variables in the monomial). Changing the monomial ordering may change the Gröbner basis.

In **Xcas** the monomial orderings that one can chose from are: `plex`, the default lexicographical order, `tdeg`, the total degree and `revlex`, first total degree and then reverse lexicographical order. Reverse lexicographical order is not what its name implies; what it means is that if x is the leading variable (i.e. the ordering is $x \succ y \succ z$) xy^2z^2 is the leading term of the polynomial $x^2y^2z + xy^2z^2$, whereas if z is the leading variable (i.e. the ordering is $x \prec y \prec z$) the leading term of the polynomial is x^2y^2z.

The Gröbner basis for a system of polynomial equations gives most of the answers to the questions arising in the study of such systems. For example, to find out if a system of polynomial equations is consistent, it suffices to compute any Gröbner basis and check whether it contains a non-zero constant. If the Gröbner basis contains the constant 1, then the whole system is equivalent to the equation $1 = 0$, which means it is inconsistent. Additionally, in the case of a finite number of solutions, one can easily compute them whereas, in the case of an infinite number of solutions, one can determine whether they lie on a curve, a surface or a set of higher dimensions.

Using Gröbner bases one can effectively eliminate a variable from a system of equations. For example, in the previously stated example

$$\begin{cases} x^2y - xy + 1 = 0 \\ x^3 - y^2 + 1 = 0 \end{cases},$$

one can easily eliminate the variable y by computing the Gröbner basis of the system — provided of course that y is the leading variable (i.e. the variable ordering is $x \prec y$). Lo and behold, the basis contains the polynomial $x^7 - 2x^6 + x^5 + x^4 - 2x^3 + x^2 - 1$ of only one variable, x.

In **Xcas** elimination is achieved by computing the Gröbner basis of the system in lexicographical order `gbasis([`$x^2 * y - x * y + 1, x^3 - y^2 + 1$`],[y,x])`[4] and looking in it for a polynomial in the single variable x[5]. Since there are only two variables, in this case the elimination of y can be also achieved with the command `resultant(`$x^2 * y - x * y + 1, x^3 - y^2 + 1, y$`)`.

[4] This is equivalent to `gbasis([`$x^2 * y - x * y + 1, x^3 - y^2 + 1$`],[y,x],plex)`; that is, it computes the basis in lexicographical order using functions from the **CoCoA** library.

[5] In **CoCoA** the polynomial in the single variable is the first polynomial of the basis.

7 Monomial ideals and Dickson's lemma

An ideal J is called a monomial ideal if it is generated by monomials. Equivalently, one can say that if $f = \sum_{\alpha \in \mathbb{N}^n} k_\alpha x^\alpha$ belongs to J, then $x^\alpha \in J$ whenever $k_\alpha \neq 0$.

Below is a proof of Dickson's famous lemma, which states that any monomial ideal has a finite basis; this lemma is used in the proof of the existence and termination of the minimal Gröbner basis of any arbitrary ideal J.

By definition, the leading monomials of the polynomials of a minimal Gröbner basis of an ideal J do not, pairwise, divide each other. Dickson's lemma guarantees that all these sets of monomials are finite.

Three equivalent versions of this lemma are presented along with the proof of the third version which is the least demanding.

Dickson's Lemma — version 1. Any monomial ideal has a finite basis.

Dickson's Lemma — version 2. For every infinite sequence $\{f_k : k \in \mathbb{N}\}$ of n-tuples of natural numbers there exist indices $i < j$ such that $f_i \leq_n f_j$. The relation $\leq_n$ on $\mathbb{N}^n$ is defined as $\{a_1, \ldots, a_n\} \leq_n \{b_1, \ldots, b_n\}$ if and only if $a_i \leq b_i$ for all $i = 1, \ldots, n$.

Dickson's Lemma — version 3. Let $n \in \mathbb{N}$ and let $T = \{t_k, k \in \mathbb{N}\}$ be any infinite set of monomials in the variables $\{x_1, x_2, \ldots, x_n\}$. Then in the set T there exist two monomials t_i and t_j such that t_i divides t_j.

The proof of the third version of Dickson's Lemma is by induction on the number of variables.

For monomials in one variable, say x, the proof is obvious: for any two monomials x^n and x^m, the one with the smaller power divides the other.

Assume now that that the lemma is true for arbitrary infinite sets in l variables. Take the infinite set of monomials $T = \{t_k, k \in \mathbb{N}\}$ in $l+1$ variables and assume that pairwise they do not divide each other. Take the first monomial $t_1 = x_1^{e_1} x_2^{e_2} \cdots x_{l+1}^{e_{l+1}}$ in T. By assumption, any other monomial t in T does not divide t_1, which means that the degree of one of the variables $\{x_1, x_2, \ldots, x_l, x_{l+1}\}$ in t_1 is strictly less than the corresponding degree in the monomial t. Subdivide now the set T in $l+1$ subsets $T_1, T_2, \ldots, T_{l+1}$. In subset T_1 belong all the monomials whose degree in the variable x_1 is less than the corresponding degree of the monomial t_1, etc. According to this rule, if a monomial may belong simultaneously to several subsets T_i, then pick any one of the possible candidates. Since the set T is infinite, one of the subsets, say T_i, will also be infinite. In this set of monomials the degree of the variable x_i is strictly less than e_i. Now, break up the set T_i into e_i subsets. In the first of them the degree of the monomials

in the variable x_i is 0, in the second 1 and in the e_i-th set the degree of the monomials in the variable x_i is $e_i - 1$. Again one of these sets is infinite and, in all the monomials of this infinite set, the degree in the variable x_i is the same; hence, these monomials may be distinguished only by the degrees in the other variables. Dividing out the variable x_i one obtains an infinite set of monomials in l variables, which pairwise do not divide each other. However, this contradicts the induction hypothesis.

With the help of Dickson's lemma one can now easily prove Hilbert's basis theorem, one of the most important theorems on polynomial ideals.

8 Hilbert's basis theorem and Gröbner bases

This theorem was discovered by Hilbert long before the appearance of Gröbner bases, but it is equivalent to the existence and finiteness of such bases for any polynomial ideal.

Theorem. In the algebra of polynomials with a finite number of variables, every ideal is generated by a finite set of polynomials $\{g_i\}_{i=1}^r$.

In this section a stronger statement is proven, namely:

Theorem. In the algebra of polynomials with a finite number of variables, every ideal J has a finite Gröbner basis $G = \{g_1, g_2, \ldots, g_r\}$.

Proof. For the ideal J consider the infinite set of monomials $Mon(J)$, which consists of all the leading terms of polynomials in the ideal J. This set must contain *minimal* elements, which cannot be divided by any other monomial in the set $Mon(J)$. Indeed, take any monomial $t_1 \in Mon(J)$. If it is not minimal, then it can be divided by some other monomial t_2, which in turn can be divided by some other monomial t_3, etc. Such a chain cannot be infinite, and hence it ends with a minimal monomial. All these minimal elements do not divide each other and hence, by Dickson's lemma, their number is finite. Let this set of minimal elements be $\{t_1, t_2, \ldots, t_r\}$, where each one of them is the leading term of some polynomial, say g_i, of J. It turns out these polynomials $\{g_i\}_{i=1}^r$ form the Gröbner basis of the ideal J. Indeed, the leading term of every polynomial in J belongs to the set $Mon(J)$ which implies that it can be divided by one of the minimal monomials t_i. However, these monomials t_i are the leading terms of the polynomials $\{g_i\}_{i=1}^r$, which means that these polynomials satisfy the definition of a Gröbner basis. Therefore, the theorem has been proven.

The basis computed in the theorem does not have to be reduced. However, one can apply to this set the reduction algorithm discussed earlier in order to

obtain a reduced basis. As has been shown above, this reduced basis is unique provided an ordering of the monomials has been fixed.

So far it has been shown that the Gröbner bases exist, but alas, the proof of the theorem is not constructive and no algorithm for the computation of these bases has been presented. The definition of Gröbner bases, according to which the leading term of any polynomial of the ideal J is divided by the leading term of one of the polynomials in the basis, is of little help, since no one can check this property on all the polynomials of J.

9 Computation of Gröbner bases

The first algorithm for the construction of Gröbner bases was proposed by the Austrian mathematician Bruno Buchberger, who also introduced this term in mathematics — to honor his Ph.D. Thesis advisor Wolfgang Gröbner.

Before discussing this algorithm for computing the Gröbner bases, here is a more modest task. Suppose an arbitrary polynomial ideal is given along with a set of polynomials $\{p_1, p_2, \ldots, p_k\}$ belonging to this ideal. How can it be tested if these polynomials form a Gröbner basis for this ideal?

Recall that, according to the definition of Gröbner bases, what is required is to test whether the leading term of any polynomial of the ideal J is divided by the leading term of one of the polynomials in the basis. However, since there is an infinite number of polynomials in the ideal J it is impossible to test this condition. Therefore, the question is now expressed algorithmically and in a somewhat different form.

The fact that the leading term of any polynomial of the ideal J is divided by the leading term of one of the polynomials in the basis means that one can apply to any polynomial of the ideal J the reduction-division algorithm with the help of (modulo) the set of polynomials $\{p_1, p_2, \ldots, p_k\}$. The result also belongs to the ideal J and the reduction algorithm can be applied to it as well. This process either will terminate (with 0 if the set $\{p_1, p_2, \ldots, p_k\}$ forms a Gröbner basis — as was shown in Section 6) or it will go on forever. However, the whole purpose of the reduction process is to *reduce* the leading term of the polynomial to which it is applied. Hence it follows — even though some reasoning is required — that the process of reduction of the leading term cannot go on for ever. This imposes certain conditions on the monomial ordering, but all the orderings discussed so far satisfy them.

To test if the set of polynomials $\{p_1, p_2, \ldots, p_k\}$ is a Gröbner basis — that is, to test if any polynomial in the ideal J is reduced to 0 modulo the set of polynomials $\{p_1, p_2, \ldots, p_k\}$ — one has to clearly identify a *finite* set of polynomials — on which to check this condition — such that their reducibility to zero would imply the reducibility to zero of any polynomial of the ideal. It turns

out that it is possible to construct such a set of polynomials. However, whereas the construction of such a set of polynomials is not complicated, the proof of Buchberger's theorem, which states that their reducibility to zero implies the reducibility to zero of all elements of the ideal, is somewhat complicated and it is omitted from this discussion. But what are these polynomials?

Take any pair (p_i, p_j) of elements from the set $\{p_1, p_2, \ldots, p_k\}$. Let their leading terms be, respectively, t_i and t_j, and their respective (leading) coefficients c_i and c_j. The monomials t_i and t_j may have a greatest common divisor, which will be denoted by *gcd(*t_i*,*t_j*)* — and which is computed by the function `gcd(`t_i`,`t_j`)` in **Xcas**. Form now the new polynomial $S(p_i, p_j)$ which is defined by

$$S(p_i, p_j) = \frac{c_j t_j p_i - c_i t_i p_j}{gcd(t_i, t_j)}.$$

These polynomials are called *S-polynomials*. And here is now Buchberger's amazing theorem:

Theorem. Assume a given monomial ordering. The finite set of polynomials $\{g_1, g_2, \ldots, g_r\}$ from the ideal J forms a Gröbner basis for this ideal and for this ordering if and only if all S-polynomials $S(g_i, g_j)$ are reduced to zero with the help of the set $\{g_1, g_2, \ldots, g_r\}$.

Such a finite condition may be effectively checked in a finite number of steps, even though this is labor-intensive process.

This theorem is the basis of the algorithm for the construction of a Gröbner basis. With the help of this algorithm any system of equations can, in principle, be transformed to the simplest canonical form.

10 Buchberger's algorithm

Suppose an ideal is given by the finite set of polynomials

$$\begin{cases} p_1(x_1, x_2, \ldots, x_n) \\ p_2(x_1, x_2, \ldots, x_n) \\ \qquad\qquad \vdots \\ p_m(x_1, x_2, \ldots, x_n) \end{cases}.$$

At first the polynomials p_i are reduced with respect to each other and this results in the reduced set $G = \{g_1, g_2, \ldots, g_k\}$. Now check, whether this is a Gröbner basis. If the answer is yes, then the process terminates. If not, then one of the S-polynomials $S(g_i, g_j)$ does not reduce to zero. Add now this non-zero reduction to the set G as the polynomial g_{k+1} and again execute all possible

reductions. The fact that this process cannot go on for ever may also be deduced from Dickson's lemma. At the end the result is the needed Gröbner basis.

It should be noted, that the result of the process does not depend on the order one chooses the S-polynomials or on the order all these possible reductions are executed. The Gröbner basis depends only on the ideal and the chosen monomial ordering. However, the speed of the algorithm does depends on all these factors, and it turns out that there are many ways to speed up the process.

At first, it turns out, one need not examine *all* possible S-polynomials, but only those that satisfy certain additional conditions. These conditions are usually called *criteria*, and the most powerful of them were developed by Buchberger — the so called *Buchberger's criteria*. In this case a sharper version of Buchberger's version needs to be proven, where it is stated that the reducibility of only those S-polynomials that satisfy the criteria, needs to be tested to see if a set is a Gröbner basis.

Another possibility to speed up the algorithm is to search for the best strategy for choosing the S-polynomial at every step. Moreover, computing all the S-polynomials that reduce to zero is, in principle, useless and one would like to foretell such an outcome. In the past thirty years many different strategies were developed for improving the performance of the algorithm, and active research in this area still continues.

11 Information obtained from Gröbner bases

At this point one should know what a Gröbner basis is and how the algorithm works for computing this sort of bases with the help of a computer. Unfortunately, the complexity of this algorithm is quite high and it is *not* recommended that one should try and solve a system of polynomial equations by finding the Gröbner basis of the system with pencil and paper. This can be done only in very simple cases. For the general case one should use a computer algebra system and **Xcas** is recommended as a wonderful tool.

The advantages of reducing a system of polynomial equations to its corresponding Gröbner basis are listed below.

In a nutshell, having computed the Gröbner basis of a system of equations one can practically find out everything about this system and its solutions.

It has already been stated — in Sections 2 and 6 — how to tell whether a system of equations is consistent or not: the Gröbner basis of an inconsistent system will consist of only the constant 1. That is, the system is equivalent to the single equation $1 = 0$.

Next, if one is interested in the number of complex roots of the system, this can be easily computed from the Gröbner basis of the system. The process is demonstrated in the following example, where for clarity only two variables are used — however, it can be also used in the general case.

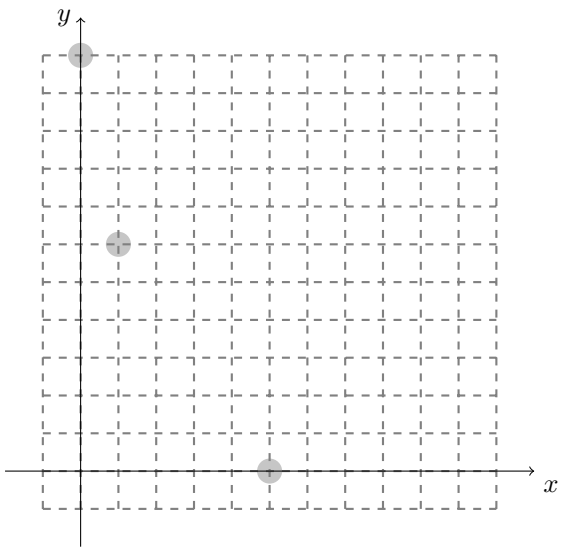

Figure 1: The points (0,11), (1,6) and (5,0) corresponding to the terms y^{11}, xy^6 and x^5, respectively.

Suppose the following system of equations is given:

$$\begin{cases} 3x^3y + 5xy^6 + 2 = 0 \\ x^5 + y^5 = 1 \end{cases}, \tag{2}$$

with x being the leading variable, i.e. $x \succ y$. Compute the Gröbner basis of the system, using total degree as the monomial ordering (see Section 6). In **Xcas** the command is `gbasis([`$3x^3y + 5xy^6 + 2, x^5 + y^5 - 1$`],[`$x, y$`],tdeg)`, and the answer is

$$\begin{cases} x^5 + y^5 - 1 \\ 5xy^6 + 3x^3y + 2 \\ 25y^{11} - 25y^6 - 9x^4y - 10x^4 - 15x^2y - 6x \end{cases}.$$

Consider the leading monomial (the leading term without the coefficient) of each of the three polynomials in the basis. These monomial are x^5, xy^6 and y^{11}. On a cross-section paper, plot now three points with integer coordinates corresponding to these monomials; that is, plot the points $(5,0), (1,6)$ and $(0,11)$ as shown in Figure 1.

Count now all the points with integer coordinates ≥ 0 that are to the *left* and *below* the three plotted points. It is easily seen that there are exactly 35 points, which implies that the system has exactly 35 complex solutions. If another monomial ordering had been chosen, then that would probably result in another

basis with different leading monomials but, nonetheless, the number of points with integer coefficients lying to the *left* and *below* the points corresponding to the leading monomials would still be 35.

If one needs to compute these 35 solutions, then the Gröbner basis of the system has to be computed again using lexicographical order[6]. Try this out on **Xcas** with the command `gbasis([`$3x^3y + 5xy^6 + 2, x^5 + y^5 - 1$`],[`$x, y$`])`. The resulting basis has two polynomials, $\{g_1, g_2\}$ and, as can be seen from equation (3), the first polynomial, g_1, has degree 35 and is independent of the variable x.

$$\begin{aligned} g_1 = &-3125y^{35} + 3125y^{30} - 243y^{20} - 1350y^{19} - 1500y^{18} \\ &+ 729y^{15} + 2700y^{14} + 1500y^{13} - 729y^{10} - 1350y^9 + 243y^5 + 32 \end{aligned} \tag{3}$$

The 35 roots of this polynomial can be found using the function `solve(`g_1`=0,y)` of **Xcas**, where one should first make sure that the appropriate box — in the `CAS Configuration` submenu — has been checked for the operations to be done over the complex numbers (by default operations are done over the reals). To find the corresponding values of the variable x, one has to replace each one of the 35 values of y in the second polynomial, g_2, of the Gröbner basis and solve it for x. This can be done because the leading term of g_2 is x and all other terms depend only on y — but, alas, the coefficients are horrendous. To save space, the case of real roots is presented here since it is quite simple.

In most applications one is mainly interested only in the real solutions of the system. To compute them, make sure that the operations in **Xcas** are done over the reals and call `solve(`g_1`=0,y)` to obtain the root $y = 1.0013$. Using the command `subst(`g_2`,y=1.0013)` in **Xcas** replace this value of y in the second polynomial, g_2, of the Gröbner basis and solve the resulting equation for x to obtain $x = -0.36728$. Hence, the system of equations (2) has one real root $x = -0.36728, y = 1.0013$ — and this can be easily verified with the command `solve([`$3x^3y + 5xy^6 + 2, x^5 + y^5 - 1$`],[`$x, y$`])`.

Looking at the second polynomial, g_2, of the Gröbner basis one cannot help but be impressed.

[6] Alternatively — since, in general, computing lexicographically ordered Gröbner bases takes a lot of time — one can use the so called *FGLM*-algorithm to compute the lexicographically ordered Gröbner basis from the already computed one in total degree.

$$
\begin{aligned}
g_2 = &- 3679446269344147064217867529068050667256490019026831920x \\
&- 33450260014643756092564956631851672976217498299687500000y^{34} \\
&+ 47024812313616802389767609859229308982587080095605468750y^{33} \\
&- 49085561713650227534231198594637060876793531875000000y^{32} \\
&+ 41367427873559221267273370836755268508579578125000000y^{31} \\
&- 45924897547451313152572990933831233467174716544062500y^{30} \\
&- 77192352575853432579939767123074110320432194853125000y^{29} \\
&- 48731087566302218372082755827497949932169541083803468750y^{28} \\
&+ 84975391632162356321237590474010958286747177341900000y^{27} \\
&- 40565034524305905496809134557282010862751189665000000y^{26} \\
&- 14175445089509252683063366649320851083854898213531250000y^{25} \\
&+ 39433187466796941516357463195370332377617751293125000000y^{24} \\
&+ 26934329158370558631825780416412269588680127531250000y^{23} \\
&+ 16630193270089603279367336601880573616368560779400000y^{22} \\
&- 10818614373511865569800188709901461438997419183000000000y^{21} \\
&+ 15402649342122677973508407015627912845355979773468750000y^{20} \\
&- 12528160916265674953249975363239710068975695862212000y^{19} \\
&- 84400103026007815198616547443930977137830717391057187 5y^{18} \\
&- 18271998277684407897619972999208648185439330709684375 0y^{17} \\
&+ 24843450380186927051798545755967844014291327331390625 00y^{16} \\
&+ 14667755184908612857192728037355928365792709656221200 0y^{15} \\
&+ 62076013284456764922457367291179154343707973436860000 0y^{14} \\
&- 39835308896014538642040822554471807252472337932512023y^{13} \\
&- 44878546529231272537985026695513072980221312690784046 60y^{12} \\
&- 12296839907172061053207077148037292099357981360390625 00y^{11} \\
&+ 78503888530272387662844650601705298978729671423339000y^{10} \\
&- 44278299348148869317430305859406277836754194128069000 0y^{9} \\
&+ 23658098595121505864269324834553197058487175864801396 71y^{8} \\
&+ 41353085543424887443521943621538852392295771872552484 10y^{7} \\
&- 14750162932496429763774227796486315706905153300000000 0y^{6} \\
&+ 11871272616878489329081786996280153342258317571310000 0y^{5} \\
&+ 47837339917872061932616501235049260719631537494080000y^{4} \\
&- 12751358825097709801298375042029842829873939303188557 73y^{3} \\
&+ 22952927869128406871708699571670698771803704758000000y^{2} \\
&- 18457980566150441031462963228820288042681185600000000 * y \\
&+ 19121961277149639001581974898205471805982052725888000
\end{aligned}
$$

No wonder one had to use a computer algebra system to replace y by the value 1.0013 in g_2. Nonetheless, the structure of g_2 is quite simple.

In the next section Gröbner bases are used for the solution of Geometrical problems.

12 Discovering new formulae

The problem at hand is to discover a formula expressing the area of a triangle with sides a, b, c through its three heights h_a, h_b, h_c. Recall the well-known formulae to compute the area of a triangle:

$$A = \frac{1}{2}ah_a, \quad A = \frac{1}{2}bh_b, \quad A = \frac{1}{2}ch_c$$

and the not so well-known Heron's formula (named after Heron of Alexandria):

$$A = \sqrt{p(p-a)(p-b)(p-c)},$$

where p is the semiperimeter

$$2p = a + b + c.$$

Changing Heron's formula to

$$A^2 = p(p-a)(p-b)(p-c)$$

one obtains a system of five equations in eight variables. Compute the Gröbner basis of this system in lexicographical order, with the variables h_a, h_b, h_c, A having the lowest ranking. Then, one of the basis polynomials is

$$h_a^4h_b^4h_c^4A^2+h_a^4h_b^4A^4-2h_a^4h_b^2h_c^2A^4+h_a^4h_c^4A^4-2h_a^2h_b^4h_c^2A^4-2h_a^2h_b^2h_c^4A^4+h_b^4h_c^4A^4.$$

Dividing out A^2, and solving for A^2 one obtains the required answer

$$A^2 = \frac{h_a^4h_b^4h_c^4}{2h_a^4h_b^2h_c^2+2h_a^2h_b^4h_c^2+2h_a^2h_b^2h_c^4-h_a^4h_b^4-h_a^4h_c^4-h_b^4h_c^4}.$$

Notice that the above formula was obtained by eliminating the variables a, b, c, and p in the process of computing the Gröbner basis. In **Xcas** this formula can be obtained as follows:

- define the system of five equations
  `I:=[2A-a*ha,2A-b*hb,2A-c*hc,A^2-p*(p-a)*(p-b)*(p-c),2p-(a+b+c)];`
- in lexicographical order (`plex`) compute its Gröbner basis, which consists of twenty seven polynomials
  `gb:=gbasis(I,[a,b,c,p,ha,hb,hc,A],plex);`

– the four variables a, b, c, and p have been eliminated in $gb[0]$, the first polynomial of the Gröbner basis; $gb[0]$, has seven terms, where A^2 appears once and A^4 appears six times; replace A^2 by x using the function `algsubs()`; this way A^4 will be replaced by x^2; the resulting `expression` is a second degree polynomial in x
`expression:=algsubs(`A^2`=x,gb[0]);`

– solve the second degree equation `expression=0` for x and pick the second root, which is $\neq 0$
`solve(expression=0,x)[1];`