189-692B

Geometry and Topology:

Computing Gröbner Bases

Matthew John D. HAYES
William O.J. MOSER

Mechanical Engineering - Mathematics and Statistics
McGill University

1 Introduction

There are many algorithms for the numeric solution of non-linear systems of equations. However, they only approximate the solutions. They ignore the geometric properties of the solution space and do not take possible alternate descriptions of the system into account [1]. The approximations can be very accurate, i.e., to many decimal places. If the equations are well conditioned, accuracy may not even be a relevant issue. Nevertheless, the solutions are still approximate.

Consider a system of linear equations. The computational effort required to solve the system is dramatically reduced by transforming the system to the *reduced row echelon* form by Gauss-Jordan elimination. The reduced system of equations has exactly the same solutions as the original system, but, depending on the original system, may be 'easier' to solve. It would be a good thing if there were an analogous process that could reduce finite systems of non-linear equations such that the reduced system was 'easier' to solve. In the early 1960's, Wolfgang Gröbner wondered if such an algorithm existed.

Gröbner bases were introduced in the Ph.D. thesis of Bruno Buchberger, written in 1965 at the University of Innsbruck, Austria. They were named in honour of Wolfgang Gröbner, Buchberger's supervisor. The essential idea behind the theory is a generalization of the theory of univariate polynomials and finite systems of linear equations. The *Buchberger algorithm* [1, 2], which computes Gröbner bases, is an extension of the *division algorithm* for polynomial long division, the method of determining *least common multiples* (lcm) of certain terms of two polynomials, and the *Euclidean algorithm* for determining the *greatest common divisor* (gcd) of two polynomials. Thus, given a finite set of multivariate polynomials over a field, the Buchberger algorithm computes a new set of polynomials, called Gröbner bases, which are generators of the same ideal as the original. The *minimal Gröbner basis* of a given ideal are thus a set of basis vectors, in that every polynomial in the ideal is generated by a linear combination of of the Gröbner basis. The solution space of the Gröbner basis is identical to the solution space of the ideal. Depending on the given ideal, it may be that the set of polynomials which comprise the Gröbner basis are 'easier' to solve than the given set of the ideal.

The advantage of using Gröbner bases theory over numerical methods, such as the Newton-Raphson or secant methods is that the reduction is algebraic, not numeric. Moreover, the Gröbner bases can always be computed for any ideal and divergence is never a problem.

2 Computational Algebra

A very detailed description of Gröbner bases theory may be found in [1], [2], and [3]. Most of the notation from [1] will be used here so that additional information will be easily accessible from that reference without major notation conflicts. A discussion of the basic theory of Gröbner bases requires a few definitions from abstract algebra.

A *group* consists of a set, $\mathcal{G}$, together with a binary operator, $*$, defined on $\mathcal{G}$ which satisfies the following axioms:

i: [closure]	$x * y \in \mathcal{G}$	$\forall\ x, y \in \mathcal{G}$
ii: [associativity]	$(x * y) * z = x * (y * z)$	$\forall\ x, y, z \in \mathcal{G}$
iii: [identity]	$\exists\ I \in \mathcal{G}:$	$I * x = x * I = x,$ $\forall\ x \in \mathcal{G}$
iv: [inverse]	$\exists\ x^{-1} \in \mathcal{G}:$	$x * x^{-1} = x^{-1} * x = I,$ $\forall\ x \in \mathcal{G}$

If in addition to Axioms 1 through 4, the elements in $\mathcal{G}$ are commutative (i.e., $x*y = y*x$, $\forall x, y \in \mathcal{G}$) then $\mathcal{G}$ is an *Abelian*, or *commutative* group. A *sub-group* $\mathcal{H}$ of group $\mathcal{G}$ is a subset of $\mathcal{G}$ which is a group under the binary operator defined on $\mathcal{G}$.

A *commutative ring* is a set R with two binary operators + (addition) and $\times$ (multiplication) which satisfy the following:

i R is a commutative (Abelian) group with respect to + and $\times$.

ii The operation $\times$ has the closure, associativity, and identity properties.

iii The distributive laws: $\forall\, x, y, z \in R$,

$$\begin{aligned} x \times (y + z) &= (x \times y) + (x \times z), \\ (x + y) \times z &= (x \times z) + (y \times z). \end{aligned}$$

A *field* is a commutative ring in which every element, except 0, has a multiplicative inverse. Let k be a field. A *k-vector space*, V, is an additive commutative group together with an operation called *scalar multiplication* that assigns to each $a \in k$, called a scalar, and to each $\mathbf{v} \in V$, called a vector, an element $a\mathbf{v} \in V$ so that the following hold:

i $a_1(a_2\mathbf{v}) = (a_1a_2)\mathbf{v}$, $\forall\, a_i \in k$ and $\forall\, \mathbf{v} \in V$;

ii $(a_1 + a_2)\mathbf{v} = a_1\mathbf{v} + a_2\mathbf{v}$, $\forall\, a_i \in k$ and $\forall\, \mathbf{v} \in V$;

iii $a(\mathbf{v}_1 + \mathbf{v}_2) = a\mathbf{v}_1 + a\mathbf{v}_2$, $\forall\, a \in k$ and $\forall\, \mathbf{v}_i \in V$;

iv $1\mathbf{v} = \mathbf{v}$, $\forall\, \mathbf{v} \in V$.

If $\mathbf{v}_1, \cdots, \mathbf{v}_n$ are pairwise different elements of a k-vector space V and for all $a \in k$ then any sum of the form

$$\sum_{i=1}^{n} a_i \mathbf{v}_i$$

is also called a *linear combination* of the $\mathbf{v}_i$ with coefficients a_i.

Let B be a subset of V. B is *linearly independent* if for all pairwise different $\mathbf{v}_1, \cdots, \mathbf{v}_n \in B$ and $a_1, \cdots, a_n$ in the field k

$$\sum_{i=1}^{n} a_i \mathbf{v}_i = 0 \quad \text{implies} \quad a_1 = \cdots = a_n = 0.$$

A set that is not linearly independent is called *linearly dependent*.

Let k be any field, $\mathbb{N}$ be the field of *non-negative integers*, i.e., the integers $0, 1, 2, 3, \cdots$, V be any k-vector space, and $B \subset V$. B is a *generating system* of V if $\forall\ \mathbf{v} \in V$, $\exists\ n \in \mathbb{N}$, $\mathbf{v}_1, \cdots, \mathbf{v}_n \in B$, and $a_1, \cdots, a_n \in k$ with

$$\mathbf{v} \quad = \quad \sum_{i=1}^{n} a_i \mathbf{v}_i.$$

B is called a *basis* of V if it is a linearly independent generating system. The following proposition is important to Gröbner bases theory: If V is a k-vector space and B is a subset of V then the following are equivalent

i B is a basis of V.

ii B is a minimal generating system for V.

A polynomial in n variables, $f(x_1, \cdots, x_n)$, with coefficients in k is a finite sum of *terms* of the form

$$a_j x_1^{\beta_1} \cdots x_n^{\beta_n},$$

where $a_j \in k$, $j \in \mathbb{N}$, and $\beta_i \subset \mathbb{N}$ such that, $i = 1, \cdots, n$. The polynomial f may be thought of as a sequence of numbers where $a_j = 0$ for all but finitely many $a \in k$ and $j \in \mathbb{N}$.

Let $k[x_1, \cdots, x_n]$ be the set of all polynomials in n variables[1] with coefficients in the field k. With respect to polynomial addition and multiplication $k[x_1, \cdots, x_n]$ is a commutative polynomial ring. This commutative polynomial ring $k[x_1, \cdots, x_n]$ is also a k-vector space with basis the set $\mathbb{T}^n$ of all power products

$$\mathbb{T}^n \quad = \quad \{x_1^{\beta_1} \cdots x_n^{\beta_n} | \beta_i \subset \mathbb{N}, i = 1, \cdots, n\}.$$

For example, a power product could be $x_1^4 x_2^6 x_5^2 x_8^3$.

All polynomials are uniquely determined by their coefficients. A univariate polynomial may be represented by a sequence of real numbers, the coefficients $a_i \neq 0, i \in \{1, 2, 3, \cdots\}, a \in \mathbb{R}$. Observe that this sequence is a function that maps a positive, non-zero integer to a real number

$$F: \quad \mathbb{N} \to \mathbb{R}.$$

Similarly, a multivariate polynomial can be represented by a function. The real numbers are replaced by an arbitrary ring, $\mathcal{R}$. Further, coefficients are needed not just for powers x^n of x, but for power products of variables as well, i.e., $x_1^{\beta_1} \cdots x_n^{\beta_n}$. The function may be represented as

$$F: \quad \mathbb{N}^n \to \mathcal{R}.$$

This function assigns a coefficient in the ring $\mathcal{R}$ to each n-tuple $(\beta_1, \cdots, \beta_n)$.

[1]From now on, whenever just one, two, or three variable polynomials are considered, variables with subscripts will not be used. Rather, the variables will be denoted x, y, or z as needed.

An affine n-space is defined as

$$k^n = \{(a_1, \cdots, a_n) | a_i \in k, i = 1, \cdots, n\}.$$

The a_i are the basis vectors of the n-space. For example, if $k = \mathbb{R}$, $k^n = \mathbb{R}^n$ is the Euclidean n-space.

A polynomial $f \in k[x_1, \cdots, x_n]$ determines a function $k^n \to k$ defined by

$$(a_1, \cdots, a_n) \to f(a_1, \cdots, a_n), \quad \forall\, (a_1, \cdots, a_n) \in k^n.$$

This function is called *evaluation* [1], and maps the affine n-space k^n to the field k. Hence, there are two distinct ways to regard a polynomial $f \in k[x_1, \cdots, x_n]$: one is as a formal polynomial in $k[x_1, \cdots, x_n]$; the other is as a function that maps $k^n \to k$. In [1], this dual existence of polynomials is considered to be the "...bridge between algebra and geometry".

The *variety* defined by f, $V(f)$, is the set of solutions to the equation $f = 0$, i.e., the zeros, or roots of the polynomial. It is defined mathematically as

$$V(f) = \{(a_1, \cdots, a_n) \in k | f(a_1, \cdots, a_n) = 0, \} \quad \subseteq k^n.$$

For a set of polynomials $f_1, \cdots, f_s \in k[x_1, \cdots, x_n]$ the variety $V(f_1, \cdots, f_s)$ is the set of all solutions to the system

$$f_1 = f_2 = \cdots = f_s = 0, \tag{2.1}$$

or, more formally

$$V(f_1, \cdots, f_s) \;\; = \;\; \{(a_1, \cdots, a_n) \in k | f_i(a_1, \cdots, a_n) = 0, i = 1, 2, \cdots, s\}.$$

Hence, the variety defined by a set of polynomials, or in other words, a system of equations, is the set of all intersections of the system

$$V(f_1, \cdots, f_s) \;\; = \;\; \bigcap_{i=1}^{s} V(f_i).$$

For example, the variety $V(x^2 + y^2 - 4, x - 2y^2) \subseteq \mathbb{R}^2$ is the intersection of the circle $x^2 + y^2 = 4$ and the parabola $x = 2y^2$ in the xy-plane.

Let $\mathcal{R}$ be a commutative ring, and $\{0\} \neq I \subseteq \mathcal{R}$. Then I is called an *ideal* of $\mathcal{R}$ if

i $x + y \in I, \forall\, x, y \in I$, and

ii $ar \in I, \forall\, a \in I$ and $r \in \mathcal{R}$.

I is *trivial* if $I = \{0\}$, and *proper* if $I \neq \mathcal{R}$. The ideal generated by the set of polynomials $f_1, \cdots, f_s$ is denoted by $\langle f_1, \cdots, f_s \rangle$, and is defined mathematically

$$\langle f_1, \cdots, f_s \rangle \;\; = \;\; \left\{ \sum_{i=1}^{s} u_i f_i | u_i \in k[x_1, \cdots, x_n], i = 1, \cdots, s \right\}.$$

This means that if $f_1, f_2 \in I$ then so is $f_1 + f_2$, and if $f_1 \in I$ and u is any polynomial in $k[x_1, \cdots, x_n]$, then $uf \in I$. The set of polynomials $\{f_1, \cdots, f_s\}$ is a *generating set* of the ideal I. An ideal may have many different generating sets with different numbers of elements. For example, in $k[x, y]$, $\langle x + y, x \rangle = \langle x, y \rangle = \langle x + xy, x^2, y^2, y + xy \rangle$.

It is important to note that a variety is determined by an ideal, not by a particular set of equations, or polynomials.

3 The Univariate Case

There exist many algorithms to numerically solve systems like the one in Equation (2.1). They do not take into consideration the geometric properties of the variety, nor consider its possible alternate descriptions. Systems of linear equations can be transformed with Gauss-Jordan elimination to the reduced row echelon form. This is the form of the coefficient matrix where every row has a leading '1' with zeros directly beneath and above it. This system has the same solutions as the original, but requires less computational effort to solve. Gröbner bases theory offers an analogous procedure for non-linear systems. This method involves finding a 'better' representation for the corresponding variety (solution space), meaning that the original non-linear system is now 'easier' to solve. The desired 'better' representation for the variety $V(f_1, \cdots, f_s)$ will be a 'better' generating set for the ideal $I = \langle f_1, \cdots, f_s \rangle$. 'Better', in this case, means the new set of generators give a better understanding of the algebraic structure of $I = \langle f_1, \cdots, f_s \rangle$, and the geometric structure of $V(f_1, \cdots, f_s)$.

To obtain the 'better' generating set the following problem must be addressed: the 'better' generating set of polynomials must be in the same ideal as the original set. This is called the *ideal membership problem*. Suppose the univariate polynomials $f, g_1, \cdots, g_s$ are given over a field and it must be determined if $f \in I\langle g_i \rangle$, i.e., if f is in the ideal generated by the g_i. The *greatest common divisor* (gcd), g, of $I\langle g_i \rangle$ must be determined. Then f is divided by g. The polynomial f will be in the ideal $I\langle g_i \rangle$ if and only if the remainder of this division is zero. If this is the case, then a polynomial q must exist that satisfies $f = qg$.

Gröbner basis theory extends this idea to multivariate polynomials. The criterion for ideal membership is similar. If the *polynomial reduction*, which is also called *generalized division*, of f by $g_1, \cdots, g_s$ has a remainder of zero, then $f \in I\langle g_i \rangle$. The main theorem that makes the theory work is that it is possible to generalize the Euclidean algorithm to a *preprocessing* of the set $g_1, \cdots, g_s$ in such a way that another set is obtained which generates the same ideal and has the desired property that the remainder is zero for every division with a member of the ideal as the dividend. Ideal bases with this property are called *Gröbner bases*. The preprocessing, i.e., the computation of a Gröbner basis from a given set of polynomials is the *Buchberger algorithm*. It is the multivariate analogue to the Euclidean algorithm, and, as well, a generalization of Gauss-Jordan elimination from linear algebra to the non-linear case [2].

3.1 The Euclidean Algorithm

Buchberger's algorithm for computing Gröbner bases is essentially a generalization of the Euclidean algorithm for determining the gcd of two univariate polynomials. It may also be viewed as Gauss-Jordan row reduction for systems of non-linear equations. Before discussing Buchberger's algorithm it would be helpful to review the Euclidean algorithm.

The algorithm attributed to Euclid is for determining the greatest common divisor (gcd) of two positive integers, the largest integer that divides both of them with out leaving a remainder. Suppose a and b are positive integers denoted the *dividend* and *divisor* if $a > b$. Then for some integers q_1 and r_1 (the first *quotient* and *remainder*), $0 \leq r_1 < b$.

$$a \;=\; q_1 b + r_1.$$

Since $r_1 < b$, we also have

$$b = q_2 r_1 + r_2,$$

where q_2 and r_2 are integers, with $0 \leq r_2 < r_1$.

Successive divisions produce the sequence of equations

$$\begin{array}{ll} a = q_1 b + r_1, & 0 \leq r_1 < b, \\ b = q_2 r_1 + r_2, & 0 \leq r_2 < r_1, \\ r_1 = q_3 r_2 + r_3, & 0 \leq r_3 < r_2, \\ \vdots & \vdots \\ r_{n-2} = q_n r_{n-1} + r_n, & 0 = r_n < r_{n-1} < r_{n-2}. \end{array}$$

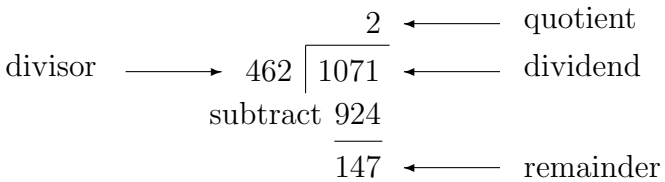
Since the successive remainders are decreasing non-negative integers, the remainder $r_n = 0$ must be obtained after a finite number of divisions. The gcd of a and b is the last positive (i.e. non-zero) remainder in the sequence. This is so because r_{n-1} is a divisor of each divisor and of each remainder. It must, therefore, be a divisor of each dividend, and the gcd of a and b is the same as that of r_{n-2} and r_{n-1}, namely, r_{n-1} [5].

For example, let $a = 1071$ and $b = 462$. Since $a > b$ then a is the dividend and b is the divisor. Applying the Euclidean Algorithm produces the following sequence of equations:

$$\begin{array}{rcl} a = q_1 b + r_1 & \Longrightarrow & 1071 = 2(462) + 147; \\ b = q_2 r_1 + r_2 & \Longrightarrow & 462 = 3(147) + 21; \\ r_1 = q_3 r_2 + r_3 & \Longrightarrow & 147 = 7(21) + 0. \end{array}$$

Since $r_3 = 0$ the algorithm terminates with 21 as the gcd of 1071 and 462.

The operations used in the Euclidean algorithm are addition and division. These operators may also be used on polynomials. Hence, the Euclidean algorithm may be used to determine the gcd of two polynomials. The main tool in the Euclidean algorithm is the division algorithm employed in the long division of real numbers. What follows is the first division and subtraction steps in obtaining the first remainder in the gcd example above. Continued application of the division algorithm reveals that quotient of the division of 1071 by 462 yields $2.31818181818\cdots$, which is an infinitely periodically repeating decimal.



3.2 The Univariate Polynomial Division Algorithm

In this section we consider polynomials in one variable: $0 \neq f \in k[x]$. The *degree* of a polynomial f, denoted by $\deg(f)$, is the largest exponent of x in f. The *leading term* of f, $\mathrm{lt}(f)$, is the highest

Table 1: Polynomial term reference terminology.

Symbol	Meaning
$\deg(f)$	Degree of polynomial f
$\mathrm{lt}(f)$	The leading term of polynomial f
$\mathrm{lc}(f)$	The leading coefficient of polynomial f

degree term of f. The *leading coefficient* of f, $\mathrm{lc}(f)$, is the coefficient of $\mathrm{lt}(f)$. These are summarised in Table 1. So, if

$$f = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0,$$

with $a_0, \cdots, a_n \in k$ and $a_n \neq 0$, then $\deg(f)=n$, $\mathrm{lt}(f)=a_n x^n$, and $\mathrm{lc}(f)=a_n$.

The polynomial f is divisible by the polynomial g if and only if $\deg(g)\leq\deg(f)$. Consider the two polynomials

$$\begin{aligned} f &= a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0, \\ g &= b_m x^m + b_{m-1}x^{m-1} + \cdots + b_1 x + b_0, \end{aligned}$$

with $n = \deg(f) \geq m = \deg(g)$. If this is so, then g *divides* f.

The first step in the division of f by g is to subtract from f the product $\frac{a_n}{b_m}x^{n-m}g$. The factor of g in this product is $\dfrac{\mathrm{lt}(f)}{\mathrm{lt}(g)}$. The remainder after the first division step is denoted as r_1 and is

$$r_1 = f - \frac{\mathrm{lt}(f)}{\mathrm{lt}(g)}g.$$

The first remainder r_1 is called a *reduction* of f by g and the process of computing r_1 is indicated by

$$f \xrightarrow{g} r_1.$$

It is to be observed that $\deg(r_1)$ is necessarily less than $\deg(f)$ due to the subtraction of the factor of g that eliminates $\mathrm{lt}(f)$. If $\deg(r_1) >\deg(g)$ the process continues, reducing r_1 by g to obtain r_2 as

$$r_2 = r_1 - \frac{\mathrm{lt}(r_1)}{\mathrm{lt}(g)}g.$$

The division algorithm continues until the final remainder equals zero, or the degree of the remainder is less than $\deg(g)$. At this point $\mathrm{lt}(g)$ can no longer be used to eliminate $\mathrm{lt}(r)$. If the polynomial division required three steps to obtain the final remainder, the reduction could be represented by

$$f \xrightarrow{g} r_1 \xrightarrow{g} r_2 \xrightarrow{g} r.$$

However, the following shorthand may be used to indicate that repeated reduction steps were used:

$$f \xrightarrow{g}_{+} r.$$

Note that an ordering of the polynomials is implied. That is, for the algorithm to terminate, the final remainder r must be zero, or have a degree less than that of g. This can only occur if the powers of x are ordered with $x^m < x^n$ and $m < n$. The last condition, $m < n$ is equivalent to the statement that x^m divides x^n [1].

It is well established ([4, 5, 6]) that, given a non-zero polynomial $g \in k[x]$, then for any $f \in k[x]$ with $\deg(f) \geq \deg(g)$, $\exists\ q$, the quotient, and the remainder, r, both $\in k[x]$ such that

$$f \;=\; qg + r, \quad \text{with } r = 0 \text{ or } \deg(r) < \deg(g).$$

Moreover, q and r are unique. The single variable division algorithm is listed below.

Algorithm 3.1 The One Variable Division Algorithm.

INPUT: $f, g \in k[x]$ with $g \neq 0$ and $\deg(f) \geq \deg(g)$
OUTPUT: $q, r : f = qg + r$ and $r = 0$ or $\deg(r) < \deg(g)$
INITIALIZATION: $q := 0; r := f$
WHILE $r \neq 0$ **AND** $\deg(g) \leq \deg(r)$ **DO**

$$q := q + \frac{\mathrm{lt}(r)}{\mathrm{lt}(g)}$$
$$r := r - \frac{\mathrm{lt}(r)}{\mathrm{lt}(g)} g$$

CONTINUE
END

Next, consider an ideal $I = \langle f_1, f_2 \rangle \in k[x]$. The gcd of f_1 and f_2 will have a variety identical to $V(f_1, f_2)$ [1]. Hence, it may be that the system $(f_1, \cdots, f_s)$ can be solved with less computational effort if $g = \gcd(f_1, \cdots, f_s)$ is first computed with the Euclidean algorithm. Then all solutions to the system are obtained by solving $g = 0$. The gcd of f_1 and f_2 is a polynomial with the following properties [6]:

1. g divides both f_1 and f_2;
2. if $h \in k[x]$ divides f_1 and f_2, then h divides g;
3. g is monic, i.e., $\mathrm{lc}(g) = 1$.

Furthermore, any other polynomial in $k[x]$ for which the remainder is zero upon division by g is in I. The gcd g is said to *generate* I, and is the 'best' generator for the ideal.

The Euclidean algorithm, discussed earlier, may be expressed as follows:

Algorithm 3.2 The Euclidean Algorithm

INPUT: $f_1, f_2 \in k[x]$, with at least one of f_1, f_2 not zero
OUTPUT: $f = \gcd(f_1, f_2)$
INITIALIZATION: $f := f_1, g := f_2$

WHILE $g \neq 0$ **DO**

$$\begin{aligned} f &\xrightarrow{g}_{+} r \\ f &:= g \\ g &:= r \\ f &:= \frac{1}{\mathrm{lc}(f)} f \end{aligned}$$

CONTINUE
END

4 Term Orders

Employing Gauss-Jordan elimination or the Euclidean algorithm requires a certain ordering of terms. For example, univariate polynomials are ordered by term degree, with the leading term having the highest degree if the division or Euclidean algorithms are to be used. For solving linear systems, the order is unimportant, but it must be specified. For multivariate systems, an analogous order is required.

Recall that the set of power products is denoted by

$$\mathbb{T}^n \;=\; \{x_1^{\beta_1}, \cdots, x_n^{\beta_n} | \beta_i \in \mathbb{N}, i = 1, \cdots, n\}.$$

Let $\boldsymbol{x}^{\boldsymbol{\beta}} = x_1^{\beta_1}, \cdots, x_n^{\beta_n}$, where $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_n) \in \mathbb{N}^n$. Note that in this paper "power product" always refers to a product of the x_i variables, while "term" always refers to the product of a coefficient and a power product. Every power product is a term with coefficient 1, but not every term is a power product. It will be assumed that the different terms in a polynomial have different power products, so $3x^2y$ would never be written as $2x^2y + x^2y$. The terms in a polynomial are arranged in increasing or decreasing order, hence there must be a way to compare any two power products. The order must be a *total order*. That is, given any $\boldsymbol{x}^{\boldsymbol{\alpha}}, \boldsymbol{x}^{\boldsymbol{\beta}} \in \mathbb{T}^n$, exactly one of the following must be true:

$$\boldsymbol{x}^{\boldsymbol{\alpha}} < \boldsymbol{x}^{\boldsymbol{\beta}}; \quad \boldsymbol{x}^{\boldsymbol{\alpha}} = \boldsymbol{x}^{\boldsymbol{\beta}}; \quad \text{or } \boldsymbol{x}^{\boldsymbol{\alpha}} > \boldsymbol{x}^{\boldsymbol{\beta}}.$$

The following three total term orders are used effectively in determining Gröbner bases [1, 2, 3].

Definition 4.1 *Let* **lex** *denote the* **lexicographical** *order on* $\mathbb{T}^n$ *with* $x_1 > x_2 > \cdots > x_n$ *and be defined as follows: If*

$$\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_n), \boldsymbol{\beta} = (\beta_1, \cdots, \beta_n) \in \mathbb{N}^n,$$

then

$$\boldsymbol{x}^{\boldsymbol{\alpha}} < \boldsymbol{x}^{\boldsymbol{\beta}} \iff \begin{cases} \textit{the first coordinates } \alpha_i \textit{ and } \beta_i \textit{ in } \boldsymbol{\alpha} \textit{ and } \boldsymbol{\beta} \\ \textit{from the left which are different satisfy } \alpha_i < \beta_i. \\ \textit{"From the left" means starting with the largest variables.} \end{cases}$$

In the case of two variables x_1 and x_2 with $x_1 > x_2$ using the lexicographical order we have

$$1 < x_2 < x_2^2 < x_2^3 < \cdots < x_1 < x_2x_1 < x_2^2x_1 < \cdots < x_1^2 < \cdots .$$

As mentioned earlier, when only several variables are used we will normally use x, y, and z instead of subscripted ones, but an ordering must be specified nonetheless. For example, in the commutative polynomial ring $k[x, y]$ using the lexicographical order with $x < y$, the following order is implied

$$1 < x < x^2 < x^3 < \cdots < y < xy < x^2y < \cdots < y^2 < \cdots .$$

In the example above the order of x and y was changed from what was likely expected to emphasise the need to impose an order among the variables.

Definition 4.2 *Let* **deglex** *denote the* **degree lexicographical** *order on* $\mathbb{T}^n$ *with* $x_1 > x_2 > \cdots > x_n$ *and be defined as follows: If*

$$\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_n), \boldsymbol{\beta} = (\beta_1, \cdots, \beta_n) \in \mathbb{N}^n,$$

then

$$\boldsymbol{x^\alpha} < \boldsymbol{x^\beta} \iff \begin{cases} \sum_{i=1}^n \boldsymbol{\alpha_i} < \sum_{i=1}^n \boldsymbol{\beta_i} \\ \textit{or} \\ \sum_{i=1}^n \boldsymbol{\alpha_i} = \sum_{i=1}^n \boldsymbol{\beta_i} \textit{ and } \boldsymbol{x^\alpha} < \boldsymbol{x^\beta} \\ \textit{with respect to } \text{lex} \textit{ with } x_1 > \cdots > x_n. \end{cases}$$

Using the degree lexicographical ordering the power products are first ordered by total degree and any ties are broken using the lex order. In the case of two variables with $x_2 < x_1$ we have

$$1 < x_2 < x_1 < x_2^2 < x_1x_2 < x_1^2 < x_2^3 < x_1x_2^2 < x_1^2x_2 < x_1^3 < \cdots .$$

In the commutative polynomial ring $k[x, y]$ the degree lexicographical ordering with $x < y$ is

$$1 < x < y < x^2 < xy < y^2 < x^3 < x^2y < xy^2 < y^3 < \cdots$$

The final term ordering is the *degree reverse lexicographical* order.

Definition 4.3 *Let* **degrevlex** *denote the* **degree reverse lexicographical** *order on* $\mathbb{T}^n$ *with* $x_1 > x_2 > \cdots > x_n$ *and be defined as follows: If*

$$\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_n), \boldsymbol{\beta} = (\beta_1, \cdots, \beta_n) \in \mathbb{N}^n,$$

then

$$\boldsymbol{x^\alpha} < \boldsymbol{x^\beta} \iff \begin{cases} \sum_{i=1}^n \boldsymbol{\alpha_i} < \sum_{i=1}^n \boldsymbol{\beta_i} \\ \textit{or} \\ \sum_{i=1}^n \boldsymbol{\alpha_i} = \sum_{i=1}^n \boldsymbol{\beta_i} \textit{ and the first coordinates } \alpha_i \textit{ and } \beta_i \textit{ in} \\ \boldsymbol{\alpha} \textit{ and } \boldsymbol{\beta} \textit{ from the right, which are different, satisfy } \alpha_i > \beta_i. \end{cases}$$

In this case, "from the right" means that the smallest variables are compared until a set of corresponding exponents are found that have different values.

In the case of two variables, deglex and degrevlex are identical. But, if there are three or more variables in the ring this is no longer the case. This can be seen in the following example:

$$x_1^2 x_2 x_3 > x_1 x_2^3 \quad \text{for deglex with} \quad x_1 > x_2 > x_3$$

but, if the degrevlex order is used the opposite is true:

$$x_1^2 x_2 x_3 < x_1^2 x_2^3 \quad \text{for degrevlex with} \quad x_1 > x_2 > x_3.$$

Using degrevlex the exponents of x_3 are compared because they are the first from the right that are different. That is, on the left hand side the exponent of x_3 is 1, on the right hand side is exponent is 0. The tie is broken because $1 > 0$, hence $\boldsymbol{x}^{\boldsymbol{\alpha}} < \boldsymbol{x}^{\boldsymbol{\beta}}$.

To compare the three term orderings, consider the polynomial in $k[x, y, z]$, described by $f = 4x^2y^2z - 10xy^4 + 2x^4$.

$$\begin{aligned} \text{lex with } x > y > z \implies & \quad xy^4 < x^2y^2z < x^4, \\ \implies & \quad f = 2x^4 + 4x^2y^2z - 10xy^4. \end{aligned}$$

$$\begin{aligned} \text{deglex with } x > y > z \implies & \quad x^4 < xy^4 < x^2y^2z, \\ \implies & \quad f = 4x^2y^2z - 10xy^4 + 2x^4. \end{aligned}$$

$$\begin{aligned} \text{degrevlex with } x > y > z \implies & \quad x^4 < x^2y^2z < xy^4, \\ \implies & \quad f = -10xy^4 + 4x^2y^2z + 2x^4. \end{aligned}$$

Again, note that for the degrevlex ordering, to break the tie the first set of different exponents from the right are those of z. Since $1 > 0$ then $x^2y^2z < xy^4$.

5 Multivariate Polynomial Division Algorithm

Table 2: Polynomial term reference terminology for multiple variables.

Symbol	Meaning
$\deg(f)$	Degree of polynomial f
$\mathrm{lt}(f)$	The leading term of polynomial f
$\mathrm{lc}(f)$	The leading coefficient of polynomial f
$\mathrm{lp}(f)$	The leading power product of polynomial f

Now, consider the case of ideals generated by more than two multivariate polynomials, $I = \langle f_1, \cdots, f_s \rangle$. In order to divide f by $f_1, \cdots, f_s$ requires a reworking of the division and Euclidean algorithms given earlier. The general idea is the same as for linear and univariate polynomials: cancel terms of f using the leading terms of the f_i's, so that new terms are of smaller order than the cancelled terms, and continue the process of subtracting multiples of the f_i's until the remainder

has a degree smaller than any of the f_i's. One complicating factor is that the dividend may have more than one divisor.

Before commencing we will denote the *leading power product* of polynomial f as $\mathrm{lp}(f)$. The lp is listed in Table 2, while deg, lt, and lc are the same as in Table 1. Given $f, g, h \in k[x_1, \cdots, x_n]$ with $g \neq 0$, the reduction symbol given earlier

$$f \xrightarrow{g} h$$

may be thought of as f reducing to h modulo g (in other words the difference of f and h is divisible by g) in a single step, if and only if $\mathrm{lp}(g)$ divides a non-zero term $a_i x_i^{\alpha_i}$ that appears in f, and

$$h = f - \frac{a_i x_i^{\alpha_i}}{\mathrm{lt}(g)} g.$$

In this regard, h is the remainder of a one step division of f by g. This process of subtracting off terms in f that are divisible by $\mathrm{lt}(g)$ continues until $h = 0$, or $\deg(h) <\deg(g)$. This final remainder is denoted by r.

Let f, h, and $f_1, \cdots, f_s$ be polynomials in $k[x_1, \cdots, x_n]$, with $f_i \neq 0 (1 \leq i \leq s)$, and let $F = \{f_1, \cdots, f_s\}$. Then

$$f \xrightarrow{F}_+ h$$

is the notation for f reduces to h modulo F, if and only if there exists a sequence of indices $i_1, i_2, \cdots, i_t \in \{1, \cdots, s\}$ and a sequence of polynomials $h_1, \cdots, h_{t-1} \in k[x_1, \cdots, x_n]$ such that

$$f \xrightarrow{f_{i_1}} h_1 \xrightarrow{f_{i_2}} h_2 \xrightarrow{f_{i_3}} \cdots \xrightarrow{f_{i_{t-1}}} h_{t-1} \xrightarrow{f_{i_t}} h.$$

If $h = 0$ or there is no power product in h that is divisible by any of the $\mathrm{lp}(f_i)$, then h is *reduced* with respect to the set of non-zero polynomials F. Such a reduced polynomial is a *remainder* and is called r. In other words, r can not be reduced modulo F. This reduction process allows for the definition of a multivariate division algorithm, analogous to the univariate case. Given $f, f_1, \cdots, f_s \in k[x_1, \cdots, x_n]$ with $f_i \neq 0$, the algorithm below returns quotients $u_i, \cdots, u_s \in k[x_1, \cdots, x_n]$, and a remainder $r \in k[x_1, \cdots, x_n]$, such that

$$f = u_1 f_1 + \cdots + u_s f_s + r.$$

Algorithm 5.1 Multivariate Polynomial Division Algorithm.

INPUT: $f, f_1 \cdots, f_s \in k[x_1, \cdots, x_n]$ with $f_i \neq 0 (1 \leq i \leq s)$

OUTPUT: $u_1, \cdots, u_s, r$ such that $f = u_1 f_1 + \cdots + u_s f_s + r$ and
r is reduced with respect to $\{f_1, \cdots, f_s\}$ and
$\max(\mathrm{lp}(u_1)\mathrm{lp}(f_1), \cdots, \mathrm{lp}(u_s)\mathrm{lp}(f_s), \mathrm{lp}(r)) = \mathrm{lp}(f)$.

INITIALIZATION: $u_1 := 0, \cdots, u_s := 0, r := 0, h := f$

WHILE $h \neq 0$ **DO**

IF $\exists\ i$ such that $\mathrm{lp}(f_i)$ divides $\mathrm{lp}(h)$ **THEN**
choose the least i such that $\mathrm{lp}(f_i)$ divides $\mathrm{lp}(h)$

$$u_i := u_i + \frac{\mathrm{lt}(h)}{\mathrm{lt}(f_i)}$$

$$h := h - \frac{\mathrm{lt}(h)}{\mathrm{lt}(f_i)} f_i$$

ELSE

$$\begin{aligned} r &:= r + \mathrm{lt}(h) \\ h &:= h - \mathrm{lt}(h) \end{aligned}$$

CONTINUE
END

Note that in this algorithm an ordering is assumed among the polynomials in the set $\{f_1, \cdots, f_s\}$ when i is chosen to be least such that $\mathrm{lp}(f_i)$ divides $\mathrm{lp}(h)$. The univariate polynomial division algorithm starts with $r = f$, then multiples of f_i are subtracted off until $\mathrm{lt}(r)$ is not divisible by $\mathrm{lt}(f_i)$. Because more than one divisors may be involved in the multivariate case, the polynomial h is introduced into the algorithm. It begins with $h = f$ and $r = 0$, and the leading term of h is subtracted off when conditions permit or else the leading term of of h is added to into r, building up the expression for the remainder.

Example 5.1

It is required to reduce f by F, where

$$\begin{aligned} F &= \{f_1, f_2\}, \\ f_1 &= yx - y, \\ f_2 &= y^2 - x, \\ f &= y^2 x. \end{aligned}$$

The order is deglex with $y > x$.

INITIALIZATION: $u_1 := 0, u_2 := 0, r := 0, h := y^2 x$

First pass through the WHILE loop:

$yx = \mathrm{lp}(f_1)$ divides $\mathrm{lp}(h) = y^2 x$

$$\begin{aligned} u_1 &:= u_1 + \frac{\mathrm{lt}(h)}{\mathrm{lt}(f_1)} = y \\ h &:= h - \frac{\mathrm{lt}(h)}{\mathrm{lt}(f_1)} f_1 \\ &= y^2 x - \frac{y^2 x}{yx}(yx - y) \\ &= y^2 \end{aligned}$$

Second pass through the WHILE loop:

$yx = \mathrm{lp}(f_1)$ does not divide $\mathrm{lp}(h) = y^2$

$y^2 = \mathrm{lp}(f_2)$ divides $\mathrm{lp}(h) = y^2$

$$\begin{aligned} u_2 &:= u_2 + \frac{\mathrm{lt}(h)}{\mathrm{lt}(f_2)} = 1 \\ h &:= h - \frac{\mathrm{lt}(h)}{\mathrm{lt}(f_2)} f_2 \\ &= y^2 - \frac{y^2}{y^2}(y^2 - x) \\ &= x \end{aligned}$$

Third pass through the WHILE loop:

$yx = \text{lp}(f_1)$ does not divide $\text{lp}(h) = x$

$y^2 = \text{lp}(f_2)$ does not divide $\text{lp}(h) = x$

$$\begin{aligned} r := r + \text{lt}(h) &= x \\ h := h + \text{lt}(h) &= 0 \end{aligned}$$

The WHILE loop stops and gives the OUTPUT

$$f \xrightarrow{F}_+ x$$

and

$$f = yf_1 + f_2 + x.$$

6 Gröbner Bases

The stage is finally set for the definition of a Gröbner Basis.

Definition 6.1 *A Gröbner Basis for an ideal I is a set of non-zero polynomials $G = \{g_1, \cdots, g_t\}$ contained in I if and only if for all $f \in I$ such that $f \neq 0$, $\exists\ i \in \{1, \cdots, t\}$ such that lp(g_i) divides lp(f).*

That is, if G is a Gröbner basis for I, then all polynomials in I can be reduced with respect to G.

For a subset S of $k[x_1, \cdots x_n]$, the *leading term ideal of* S is defined to be the ideal

$$\text{Lt}(S) = \langle \text{lt}(s) | s \in S \rangle.$$

With this definition in mind, the following statements are equivalent [1]:

1. G is a Gröbner basis for I.
2. $f \in I$ if and only if $f \xrightarrow{G}_+ 0$.
3. Lt(G) =Lt(I).

The proof for the existence of G is given in [1].

6.1 Buchberger's Algorithm

In this section, Buchberger's algorithm for computing Gröbner bases will be presented. But first, one more definition is required.

Definition 6.2 *Let* $0 \neq f, g \in k[x_1, \cdots, x_n]$. *Let the* least common multiple (lcm) *of two power products be denoted* L =*lcm(lp(f),lp(g)). The polynomial*

$$S(f,g) \quad = \quad \frac{L}{\mathrm{lt}(f)} f - \frac{L}{\mathrm{lt}(g)} g$$

is defined to be the S-polynomial *of* f *and* g.

S-polynomials are used for the following reason. In the division of f by $f_1, \cdots, f_s$, it may happen that some term $a_i x_i^{\alpha_i}$ in f is divisible by both $\mathrm{lp}(f_i)$ and $\mathrm{lp}(f_j)$ with $i \neq j$, hence, $a_i x_i^{\alpha_i}$ is divisible by L =lcm($\mathrm{lp}(f_i)$,$\mathrm{lp}(f_j)$). If f is reduced by f_i then

$$h_1 \quad = \quad f - \frac{a_i x_i^{\alpha_i}}{f_i} f_i$$

is obtained. On the other hand, if f is reduced by f_j

$$h_2 \quad = \quad f - \frac{a_i x_i^{\alpha_i}}{f_j} f_j$$

will be obtained. The ambiguity introduced is

$$h_2 - h_1 = \frac{a_i x_i^{\alpha_i}}{f_i} f_i - \frac{a_i x_i^{\alpha_i}}{f_j} f_j = \frac{a_i x_i^{\alpha_i}}{L} S(f_i, f_j).$$

A key theorem concerning S-polynomials is due to Buchberger.

Theorem 6.1 (Buchberger) *Let* $G = \{g_1, \cdots, g_t\}$ *be a set of non-zero polynomials in* $k[x_1, \cdots, x_n]$. G *is a Gröbner basis for the ideal* $I = \langle g_1, \cdots, g_t \rangle$ *if and only if for all* $i \neq j$,

$$S(g_i, g_j) \xrightarrow{G}_+ 0.$$

Buchberger's proof is given in [1].

The Buchberger theorem outlines a strategy for computing Gröbner bases: Reduce the S-polynomials and if a remainder is non-zero, add it to the list of polynomials in the generating set. Continue doing this until there are 'enough' polynomials in the generating set to make all S-polynomials reduce to zero. Buchberger's algorithm will produce a Gröbner basis for the ideal $I = \langle f_1, \cdots, f_s \rangle$, given $F = \{f_1, \cdots, f_s\}$ with $f_i \neq 0 (1 \leq i \leq s)$.

Algorithm 6.1 Buchberger's Algorithm for Computing Gröbner bases.

INPUT: $F = \{f_1, \cdots, f_s\} \subseteq k[x_1, \cdots, x_n]$ with $f_i \neq 0 (1 \leq i \leq s)$
OUTPUT: $G = \{g_1, \cdots, g_s\}$, a Gröbner basis for I
INITIALIZATION: $G := F, \mathcal{G} := \{\{f_i, f_j\} | f_i \neq f_j \in G\}$
WHILE $\mathcal{G} \neq 0$ **DO**
Choose any $\{f, g\} \in \mathcal{G}$
$\mathcal{G} := \mathcal{G} - \{\{f, g\}\}$

$S(f,g) \xrightarrow{G}_{+} h$, where h is reduced with respect to G
IF $h \neq 0$ **THEN**
$\mathcal{G} \cup \{\{u,h\} | \ \forall \ u \in G\}$
$G := G \cup \{h\}$
CONTINUE
END

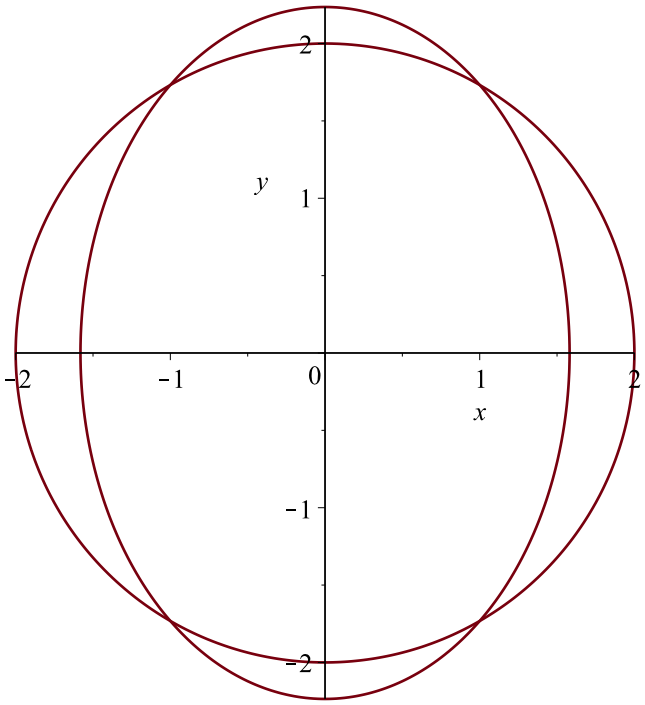


Figure 1: Non-linear equations f_1 & f_2: intersecting circle and ellipse.

Example 6.1

Consider a set of non-linear equations in two variables. The variety of this set contains the real intersections, if any, of these equations. Let the first equation represents a circle with radius 2, centred at the origin of an orthogonal planar Cartesian coordinate system, $x^2 + y^2 = 4$. It is required to determine the set of real intersections (if any) of this circle with the ellipse described by $2x^2 + y^2 = 5$. These two equations may be rearranged as polynomials in two variables, x and y:

$$f_1 : \quad x^2 + y^2 - 4; \tag{6.1}$$

$$f_2 : \quad 2x^2 + y^2 - 5. \tag{6.2}$$

As polynomials they are members of the ideal $I = \langle f_1, f_2 \rangle$. A plot of these geometric entities reveals that they do, indeed, have four intersections. This is shown in Figure 1. Hence, the variety is not the empty set. However, the goal of this example is to illustrate how the Buchberger algorithm computes a Gröbner basis for the ideal I. First, a term ordering is required. We will choose lex with $y < x$, specify the input to the algorithm, and proceed:

INITIALIZATION: $G := \{f_1, f_2\}, \mathcal{G} := \{\{f_1, f_2\}\}$.

Pass one through the WHILE loop:

$\mathcal{G} := \{\{f_1, f_2\}\} - \{\{f_1, f_2\}\} = 0$.

$$\begin{aligned} S(f_1, f_2) &= \frac{L}{\mathrm{lt}(f_1)} f_1 - \frac{L}{\mathrm{lt}(f_2)} f_2 \\ &= \frac{x^2}{x^2}(x^2 + y^2 - 4) - \frac{x^2}{2x^2}(2x^2 + y^2 - 5) \\ &= \frac{1}{2}y^2 - \frac{3}{2}. \end{aligned}$$

$S(f_1, f_2) = \frac{1}{2}y^2 - \frac{3}{2}$ can be reduced by neither f_1 or f_2.
Then $S(f_1, f_2) \xrightarrow{G}_+ = h \neq 0$.
This being the case, let $f_3 := \frac{1}{2}y^2 - \frac{3}{2}$.
Continuing with the first pass:

$\mathcal{G} := \{\{f_1, f_3\}\{f_2, f_3\}\}$,
$G := \{f_1, f_2, f_3\}$.

Pass two through the WHILE loop:

Choose $\{\{f_1, f_3\}\} \in \mathcal{G}$,
$\mathcal{G} := \{\{f_2, f_3\}\}$.

$$\begin{aligned} S(f_1, f_3) &= \frac{x^2y^2}{x^2}(x^2 + y^2 - 4) - \frac{x^2y^2}{(y^2/2)}(\frac{1}{2}y^2 - \frac{3}{2}) \\ &= 3x^2 + y^4 - 4y^2 \\ &= 3f_1 + 4f_3^2 - 2f_3 + 0. \end{aligned}$$

This implies that

$S(f_1, f_3) \xrightarrow{G}_+ 0 = h$.

Pass three through the While loop:

Choose $\{\{f_2, f_3\}\} \in \mathcal{G}$,
$\mathcal{G} := 0$.

$$\begin{aligned} S(f_2, f_3) &= \frac{x^2y^2}{2x^2}(2x^2 + y^2 - 5) - \frac{x^2y^2}{(y^2/2)}(\frac{1}{2}y^2 - \frac{3}{2}) \\ &= 3x^2 + \frac{1}{2}y^4 - \frac{5}{2}y^2 \\ &= 3f_1 + 2f_3^2 - 5f_3 + 0. \end{aligned}$$

This implies that

$S(f_2, f_3) G \longrightarrow_+ 0 = h$.

The WHILE loop stops, since $\mathcal{G} = 0$,

$G := \{f_1, f_2, f_3\}$.

The output of Buchberger's algorithm may depend on the S-polynomials, that is, in each pass of the WHILE loop $\{f, g\} \in \mathcal{G}$ is chosen arbitrarily. In some cases, if the order is changed the output may be a different Gröbner basis. However, the job the Gröbner basis were required to do was to render the system 'easier' to solve. But, the algorithm gave us the original second degree polynomials plus a third univariate second degree polynomial. We are, seemingly, worse off than before! However, our goose is not yet cooked. The computed Gröbner basis can be *minimized*. This leads to the following definition.

Definition 6.3 *A Gröbner basis $G = \{g_1, \cdots, g_t\}$ is called* minimal *if for all i,* $\mathrm{lc}(g_i)=1$ *and for all $i \neq j$,* $\mathrm{lp}(g_i)$ *does not divide* $\mathrm{lp}(g_j)$.

In the previous example, it is readily shown that $\mathrm{lp}(f_2)$ and $\mathrm{lp}(f_1)$ divide each other. However, f_4 can be obtained as a linear combination of f_1 and f_2:

$$\begin{aligned} f_4 &= f_2 - f_1, \\ &= 2x^2 + y^2 - 5 - (x^2 + y^2 - 4), \\ &= x^2 - 1. \end{aligned}$$

Hence, the Gröbner basis is also represented by $G = \{f_4, f_3\}$ But, G is not yet minimal because $\mathrm{lt}(f_3) \neq 1$. This is easily remedied by multiplying through by 2, giving $f_5 = 2f_3 = y^2 - 3$. This gives a minimal Gröbner basis for $I = \langle f_1, f_2 \rangle$ of

$$G = \{x^2 - 1, y^2 - 3\}, \tag{6.3}$$

with all conditions satisfied.

The minimal Gröbner basis are the minimal generating set of the ideal to which equations 6.1 and 6.2 belong. That is, every polynomial in the ideal to which f_1 and f_2 belong can be expressed by a linear combination of the Gröbner basis, f_4 and f_5. The circle is a combination of $f_4 + f_5$ and the ellipse is a combination of $2f_4 + f_5$.

Geometrically, f_4 and f_5 represent a set of two pairs of orthogonal lines, shown in Figure 2a. Clearly then, the points shared by the lines $x = \pm\sqrt{1}$ and $y = \pm\sqrt{3}$ are the same as those shared by $x^2 + y^2 = 4$ and $2x^2 + y^2 = 5$. The variety $V(f_1, f_2)$ is identical to the variety $V(f_4, f_5)$. This is illustrated in Figure 2b. The difference is that it requires less computational effort to solve the system of $\{f_4, f_5\}$ than $\{f_1, f_2\}$.

6.2 Gröbner Basis Computation Using Maple

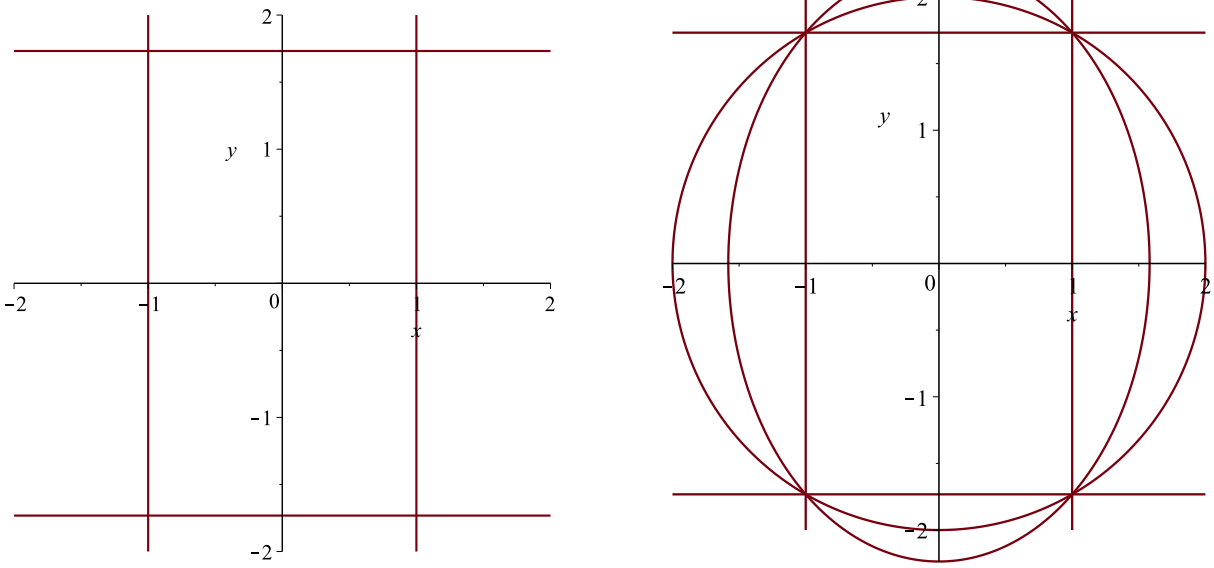
The Gröbner basis for the same ideal I and term ordering from the previous example were determined using the computer algebra software package Maple. They were found to be

$$\{x^2 - 1, y^2 - 3\}. \tag{6.4}$$

They are a set of univariate polynomials

$$\begin{aligned} g_1 &: \ x^2 - 1, \\ g_2 &: \ y^2 - 3. \end{aligned}$$

It is immediately seen that $g_1 = f_4$ and $g_2 = f_5$.



(a) The four orthogonal lines.

(b) $V(f_1, f_2) = V(f_4, f_5)$.

Figure 2: The Gröbner bases f_4 and f_5 generate $I = \langle f_1, f_2 \rangle$

7 Conclusions

A brief introduction to the theory of Gröbner bases has been presented. Because the algorithm for computing these bases reduces systems of non-linear equations algebraically, it offers strong competition to the well established numerical methods for solving such systems. The computational complexity is probably on par with most numerical methods, however divergence is never a problem.

References

[1] Adams, W., Loustaunau, P., 1994, *An Introduction to Gröbner Bases*, American Mathematical Society, Graduate Studies in Mathematics, Vol. 3.

[2] Becker, T., Weispfenning, V., 1993, *A Computational Approach to Commutative Algebra*, Graduate Texts In Mathematics, Springer-Verlag, New York, N.Y..

[3] Cox, D., Little, J., O'Shea, D., 1996, *Ideals, Varieties, and Algorithms, Second Edition*, Springer-Verlag, New York, N.Y..

[4] Clark, A., 1971, *Elements of Abstract Algebra*, Dover Publications Inc., New York, N.Y..

[5] Dolciani, P., Berman, S., Wooton, W., 1963, *Modern Algebra and Trigonometry*, Thomas Nelson & Sons (Canada) Ltd., Don Mills, Ontario, Canada.

[6] Haris, J., 1992, *Algebraic Geometry*, Graduate Texts In Mathematics, Springer-Verlag, New York, N.Y..