Chapter 7

Gröbner Bases Techniques

Beginning with Descartes, mathematics has been developing tools to formulate and prove geometric theorems algebraically, and, vice versa, to express geometric facts in algebraic terms in an effort to interpret algebraic theorems geometrically. The resulting discipline of *algebraic geometry* is of interest to us because it delivers a symbolic representation of geometric objects that allows us to compute with geometric objects using symbolic manipulation. We have made periodic use of this fact in the preceding chapters, formulating algorithms that accept algebraic equations as input and deliver, as output, other algebraic equations. Examples have included converting between implicit and parametric forms, approximating surface intersections by parametric expressions, mapping space curves to plane curves, and so on.

In this chapter, we examine these computations in more detail. More specifically, we concentrate on computing with *ideals*. Ideals are sets of polynomials that describe elementary geometric objects symbolically, and are a natural representation of geometric objects. We can often find the solution of a system of linear equations more conveniently by considering linear combinations of the given equations. Likewise, when solving systems of algebraic equations, considering algebraic combinations of them may lead to an easier solution, as long as the solution set is not altered. The set of all such algebraic combinations is an ideal.

The algorithms of this chapter are very general and are capable of solving, in principle, a wide spectrum of difficult and important problems. Such

generality does not come without its price, however, for some of the computations can be very space and time intensive. Therefore, we must weigh carefully where and when to apply these techniques. However, research is appearing that aims at *specializing* the algorithms given here in situations not requiring the full generality they currently embody. This work could result in highly efficient and sophisticated tools for addressing some of the difficult mathematical problems faced by geometric and solid modeling.

7.1 Chapter Overview

The algorithms to be discussed require a working knowledge of algebra and algebraic geometry. As in Chapter 5, we therefore begin with an informal review of the needed vocabulary and concepts, for the benefit of the nonspecialist. The purpose of the informality is to develop the intuition underlying these terms and ideas. So, we relate the mathematical concepts to applications that are already quite familiar. Once armed with the road maps provided here, the reader can consult books on algebra or algebraic geometry for further details.

The central data structure in the chapter is a special set of polynomials defining an ideal. This set is a *Gröbner basis* of the ideal. An ideal has many generating sets defining it, but the advantage of a Gröbner basis is that many algorithmic problems can be solved easily once a Gröbner basis is known.

We are interested in those algorithmic problems that arise from applications in geometric and solid modeling. However, the concept of a Gröbner basis is best grasped by considering first a more abstract problem; namely, the question whether a given polynomial g is in a given ideal I. Assuming that the ideal is given by the polynomials $\{f_1, ..., f_r\}$, we ask whether the polynomial g can be written as an algebraic combination of the f_j; that is, whether $g = h_1 f_1 + \cdots + h_r f_r$ for some polynomials h_j. This problem of *ideal membership* can be answered using a Gröbner basis of the ideal.[1]

With this problem as the focus, we explain in Section 7.3 what a Gröbner basis is, how to construct it, that the basis depends on certain term orderings, and how to use it to decide ideal membership.

Solving systems of algebraic equations is a fundamental activity in geometric and solid modeling. Section 7.4 therefore discusses how to solve systems of algebraic equations using Gröbner bases. We give a general algorithm and discuss as geometric applications how to find all points in which three or more surfaces intersect, and how to find all singular points of an algebraic curve. This section also contains an example discussing some of the subtleties that arise when we use Gröbner bases methods to solve a system of algebraic equations, in which the individual equations contain symbolic parameters. This situation arises, for example, when we wish to find the singularities of a

[1] This problem has a geometric significance: Roughly speaking, g can be so written whenever the surface $g = 0$ contains all points that are the common intersection of the surfaces $f_j = 0$. Section 7.2.6 explains why this interpretation is not exact.

family of curves, or to locate surface intersections for families of intersecting surfaces.

Section 7.5 considers operations on curves and surfaces such as implicitization, inversion, and offsetting. These are mathematically demanding problems for which Gröbner bases methods provide uniform solutions.

Section 7.6 briefly sketches the applicability of Gröbner bases methods to geometric theorem proving. The techniques developed for this problem have bearing on the robustness problem in geometric modeling in that they can provide general reasoning capabilities for the problems discussed in Chapter 4 in Section 4.4, on representations and models.

Constructing a Gröbner basis can be resource intensive, because of both the need for exact arithmetic and the possibility of generating and analyzing many polynomials. This fact hinders using Gröbner bases in practice. For this reason, Section 7.7 reviews known complexity results and discusses our experience with using Gröbner bases in geometric applications. It turns out that recent research on basis conversion has significantly improved the efficiency of this approach. Section 7.8 explains the method and gives a variant that can handle large-scale elimination problems. This material, we believe, is of great practical consequence and is paradigmatic of possible specializations that could be efficient and of widespread applicability.

7.2 Algebraic Concepts

7.2.1 Fields, Rings, and Polynomials

The simplest object we consider is described by a single algebraic equation of the form

$$f(x_1, ..., x_n) = 0$$

where f is a polynomial in n variables. We think of the variables as *coordinates* in an n-dimensional Cartesian space. Depending on the interpretation of the coordinates, the space corresponds to affine or to projective space. When substituting specific values for the x_i satisfying the equation, we obtain certain *points* on the $(n-1)$-dimensional *hypersurface* implicitly defined by f. For example, in 3-space, the equation

$$x^2 + y^2 + z^2 - 1 = 0$$

defines the unit sphere.

So far, we have tacitly assumed that the possible values for the x_i are real numbers. But we can use different sets of values, thereby giving a different meaning to f and to the space containing the hypersurface it defines. Such a set of possible values must be drawn from a *field*; that is, it must have

elements that may be added, subtracted, multiplied, and divided. We usually fix the field of coordinate values, concentrating on the geometry of the Cartesian space, and call it the *ground field*.

A field can be finite, or it can be infinite. Simple examples of a finite field include the integers modulo a fixed prime number p. For our purposes, the field $\mathbf{R}$ of real numbers is of primary interest. However, algebraic geometry has considered mostly the field $\mathbf{C}$ of complex numbers. For instance, when considering the equation

$$x^2 + y^2 = 0$$

over $\mathbf{C}$, this equation describes the two complex lines

$$x + iy = 0 \qquad \text{and} \qquad x - iy = 0$$

intersecting at the origin. Over the reals, the equation would describe just one point; namely, the origin. Therefore, we have to be aware that subtle problems can arise when we try to apply classical algebraic geometry to real spaces. The reason algebraic geometry has been developed primarily for the ground field $\mathbf{C}$ is that then certain fundamental theorems have uniform validity; for example, theorems on the dimensionality of hypersurfaces.

Let us denote the ground field by k, avoiding for the moment a commitment to a specific one. A *univariate polynomial* over k has the form

$$\sum_{i=0}^{m} a_i x^i$$

where x is a variable symbol, and the *coefficients* a_i are numbers in k. The set of all univariate polynomials in x is denoted by $k[x]$. We can add and subtract polynomials from each other and we can multiply them, but we cannot, in general, divide two polynomials. A set in which addition, subtraction, and multiplication are defined is called a *ring*. The set $k[x]$ is a ring.

Whether a polynomial can be factored will depend on the ground field k. The polynomial

$$x^2 + 1$$

does not factor over the reals, but it will factor as $(x - i)(x + i)$ over the complex numbers. A polynomial that factors nontrivially is called *reducible*. One that cannot be factored is *irreducible*. The example $x^2 + 1$ shows that reducibility is a relative notion that depends on the ground field.

7.2.2 Field Extensions and Rational Functions

A field may be considered a subfield of a larger field, provided the arithmetic operations are compatible in both fields. For example, the field $\mathbf{R}$ is a

subfield of $\mathbf{C}$. It can be convenient to think of this relationship as a process of enlarging the smaller field by adding new elements to it. This process is referred to as a field *extension*. A field extension may be done for pragmatic reasons. For instance, complex numbers were "invented" so that the polynomial $x^2 + 1$ would have roots. More generally, the *fundamental theorem of algebra* states that, over the complex numbers, every univariate polynomial can be factored into linear factors.

When extending a field, we *adjoin* the elements of the larger field. To keep things simple, we adjoin only as many elements as are needed. That is, when adjoining a new element u to a field k, we include automatically all elements that must be added as consequence of the field operations — that is, all elements obtained from u and the elements in k by successive additions, subtractions, multiplications, and divisions. Thus, when extending $\mathbf{R}$ to $\mathbf{C}$, we adjoin the imaginary unit i. In fact, it turns out that all other complex numbers then can be expressed as $a + bi$, were a and b are reals. If the field k is extended by some new element u, the new field is denoted $k(u)$.

There are two types of field extensions: *algebraic extensions* and *transcendental extensions*. In an algebraic extension, we adjoin an element that is the root of a specific polynomial $q \in k[x]$. It can be shown that all elements of the extension field so obtained can be expressed in the form $a_0 + a_1u + a_2u^2 + ... + a_{m-1}u^{m-1}$, where the degree of q is m.

For example, an algebraic extension of $\mathbf{R}$ is $\mathbf{C}$, which has the additional property that every polynomial in $\mathbf{R}[x]$ factors into linear components. When extending the reals to the complex numbers, we adjoin the root i of $x^2 + 1$. Therefore, all complex numbers can be expressed as $a + bi$. If we begin with the field $\mathbf{Q}$ of *rational* numbers, we might also adjoin a root of $x^2 - 2$, and obtain the field $\mathbf{Q}(\sqrt{2})$. All elements in this field can be written as $a + b\sqrt{2}$, where a and b are rational numbers. We might then extend the resulting field by adjoining a root of some other polynomial.

By adjoining to $\mathbf{Q}$ the roots of all univariate polynomials with rational coefficients, we get the field of *algebraic numbers*. After that, we would still miss some real numbers, such as π, that are *transcendental*. The second type of field extension, then, is a transcendental extension in which the element adjoined to the field k does not satisfy any algebraic relation — that is, is not the root of any polynomial in $k[x]$. When adjoining a transcendental x to k, new elements are obtained from the transcendental and the elements of k by successive additions, subtractions, multiplications, and divisions. These new elements can be written uniquely as ratios of relatively prime polynomials in the transcendental, $p(x)/q(x)$, where q is not the zero polynomial. The set of all these ratios is denoted $k(x)$. It is a field because there is a natural division operation defined on these ratios. Note that the assumption that x is transcendental (i.e., is not the root of some polynomial) implies that we do not accidentally divide by zero (i.e., by a root of $q(x)$).

Transcendental field extension might seem a rather remote concept. From our perspective, however, transcendental extensions correspond to computing

with symbolic parameters. For example, consider a sphere of radius r. We write

$$x^2 + y^2 + z^2 - r^2 = 0$$

as its equation. We consider this a polynomial over x, y, and z, but conceptualize r as a *parameter*, and treat it differently from the variables x, y, and z. When computing with this equation, perhaps for purposes of defining another surface whose shape depends somehow on the radius of the sphere, we may freely form expressions involving r, such as

$$y^2 + z^2 - \frac{r}{r^2 + 1} = 0$$

defining, say, a cylinder whose radius is a rational function of r. Although we do not think of instantiating these surfaces with transcendental numbers, we compute with r as though it were transcendental.

7.2.3 Multivariate Polynomials and Ideals

We form *multivariate* polynomials with more than one variable symbol. Using the symbols $x_1, ..., x_n$, such a polynomial is written

$$\sum_{j=1}^{m} a_j x_1^{e_{1,j}} x_2^{e_{2,j}} \cdots x_n^{e_{n,j}}$$

where the coefficients a_j are in the ground field k. The exponents $e_{i,j}$ are, of course, nonnegative integers.

Just as in the univariate case, we can add, subtract, and multiply multivariate polynomials, but we cannot, in general, divide two multivariate polynomials. The set of all multivariate polynomials in the variables $x_1, ..., x_n$ is denoted $k[x_1, ..., x_n]$ and forms a ring.

We saw in the case of univariate polynomials that the reducibility of a polynomial (i.e., whether it can be factored) depends on the ground field. This is still the case for multivariate polynomials, but there are also multivariate polynomials that cannot be factored over any ground field. Such polynomials are called *absolutely* irreducible. The polynomial $x^2+y^2+z^2-1$ is absolutely irreducible.

We fix a ground field k, and consider the n-dimensional affine space k^n over k. The points in this space are n-tuples $(x_1, x_2, ..., x_n)$, where the x_i take on values in k. We consider the hypersurface $f = 0$ defined by a multivariate polynomial f. We assume that f is irreducible; that is, that it does not factor. We observe that any *multiple* cf of f defines the same hypersurface, where c is a nonzero field element. Moreover, for any polynomial g, the hypersurface $gf = 0$ certainly includes the hypersurface $f = 0$. This raises the question of

whether there exists a *unique* algebraic representation for the hypersurface $f = 0$. The answer is yes, but the unique representation requires a *set* of polynomials, rather than a single one.

Consider the surface $f = 0$, and let g be any polynomial. All surfaces $gf = 0$ will contain the surface $f = 0$. Moreover, for fixed f in $k[x_1, ..., x_n]$, the intersection of all surfaces $gf = 0$, where g varies over $k[x_1, ..., x_n]$, is precisely the surface $f = 0$. So, for fixed f, we consider the set

$$I\langle f\rangle = \{gf \in k[x_1, ..., x_n] \mid f \text{ fixed}\}$$

as the description of the surface. In Section 7.2.6, we explain that this description is not always unique.

$I\langle f\rangle$ has the property that the sum and difference of any two polynomials in the set is again in $I\langle f\rangle$. Moreover, the product of any polynomial in $k[x_1, ..., x_n]$ with an element of $I\langle f\rangle$ is again in $I\langle f\rangle$. Sets with these properties are called *ideals*.

Now consider a finite set F of polynomials $f_1, f_2, ..., f_r$ in $k[x_1, ..., x_n]$. We form all *algebraic combinations* of the f_i; that is, we form the set of polynomials

$$I\langle F\rangle = \{g_1 f_1 + g_2 f_2 + \cdots + g_r f_r \mid g_i \in k[x_1, ..., x_n]\}$$

Clearly, $I\langle F\rangle$ is an ideal. We say that $I\langle F\rangle$ is the *ideal generated by* F, and that F is a *generating set* of $I\langle F\rangle$. Generating sets are not unique, and a basic theme of this chapter is to find generating sets that have special properties that are useful for solving geometric problems.

The nonuniqueness of generating sets has been used implicitly; for instance, in surface intersection. When determining the intersection of two quadrics f and g, we may proceed as follows. First, replace one of the quadrics with a *ruled* quadric surface $f' = \lambda f + \mu g$, where λ and μ are suitable numbers; that is, with a cylinder, a cone, or a hyperboloid. Then compute $f' \cap g$ instead of $f \cap g$. The same intersection is obtained, but the reformulated problem simplifies the treatment of special cases. Algebraically, we have replaced the generators $\{f, g\}$ of the ideal describing the intersection curve with the generators $\{f', g\}$.

7.2.4 The Residue Class Ring of an Ideal

Given an ideal I in the ring $k[x_1, ..., x_n]$ of multivariate polynomials over the ground field k, we consider the *residue class ring* R_I of I. The elements in R_I are equivalence classes of polynomials in $k[x_1, ..., x_n]$; that is, they are disjoint subsets of $k[x_1, ..., x_n]$, where two polynomials p and q are in the same

equivalence class iff their difference $p - q$ is in the ideal I. Computations in residue class rings will be considered later, in Section 7.8.1.

We denote the elements of R_I with $[p]$, where p is any polynomial in the equivalence class $[p]$. The operations on the equivalence classes are induced in the natural way via

1. $[p] + [q] = [p + q]$
2. $[p][q] = [pq]$

As an example, let I be the ideal generated by $\{x^2, y\}$. The elements in I have the form $ux^2 + vy$, where u and v are polynomials in $k[x, y]$. It is easy to see that $p - q$ is in the ideal I whenever p and q have the same constant term and the same x term; that is, $p = a + bx + ...$ and $q = a + bx + ...$, for numbers a and b in k. Hence two such polynomials are in the same equivalence class.

The residue class ring R_I of an ideal may be considered to be a vector space over the ground field k. Moreover, if I is a zero-dimensional ideal — that is, if there are only finitely many points $(a_1, ..., a_n) \in k^n$ satisfying every polynomial in I — then the residue class ring can be shown to be a finite-dimensional vector space.

7.2.5 Algebraic Sets and Varieties

We consider the ideal $I \subset k[x_1, ..., x_n]$ generated by the set $F = \{f_1, ..., f_r\}$. Let $p = (a_1, ..., a_n)$ be a point in k^n such that $g(p) = 0$ for every $g \in I$. The set of all such points p is the *algebraic set* $V(I)$ of I. Clearly, for p to be in the algebraic set $V(I)$, it suffices that $f_i(p) = 0$ for every generator f_i in F.

In three dimensions, the algebraic surface $f = 0$ is the algebraic set of the ideal $I\langle f\rangle$. The intersection of two algebraic surfaces f and g in 3-space is an algebraic space curve. Hence, such a curve is the algebraic set of the ideal $I\langle\{f, g\}\rangle$.[2] It is not true that every algebraic space curve can be defined as the intersection of two surfaces. Additional surfaces may be required in certain cases. An example is the *twisted cubic*, a curve parametrically defined as

$$\begin{aligned} x &= t \\ y &= t^2 \\ z &= t^3 \end{aligned}$$

See also Figure 7.1. To define it, we need to intersect three algebraic surfaces. For example, we could intersect a paraboloid with two cubic surfaces

$$x^2 - y = 0 \cap y^3 - z^2 = 0 \cap z - x^3 = 0$$

Later, we explain why two surfaces alone do not suffice, which motivates us to consider ideals with generating sets that contain more than two polynomials.

[2] In the following, we will write $I\langle f, g\rangle$ instead of $I\langle\{f, g\}\rangle$.

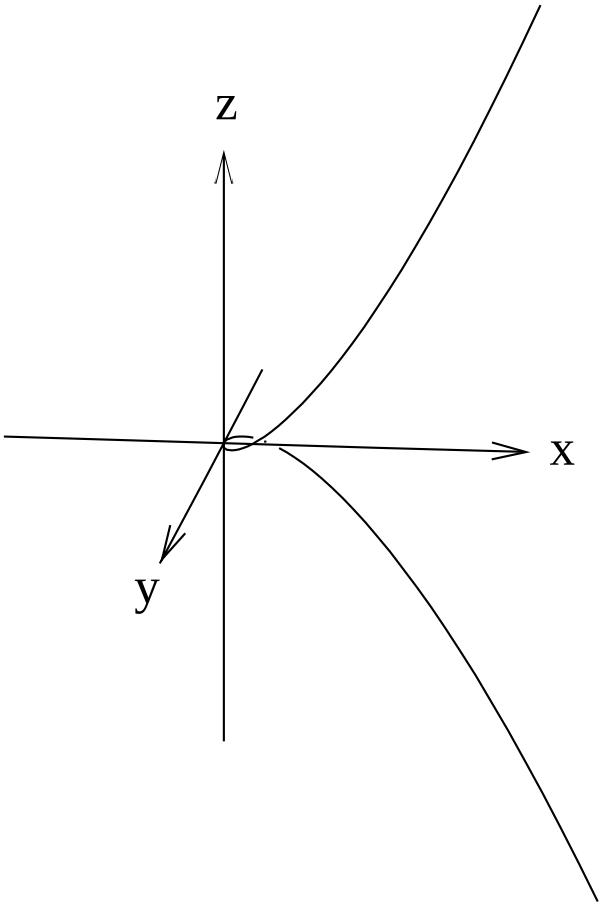

Figure 7.1 Twisted Cubic (t, t^2, t^3)

When we are given a set $F = \{f_1, ..., f_r\}$ of generators, we expect in general that the algebraic set defined by it in k^n has dimension $n-r$. This is an analogy to linear algebra, where a set of r linear equations in n variables defines, in general, a linear subspace of dimension $n - r$. Just as in linear algebra, this requires that the equations $f_i = 0$ be *algebraically independent.* However, the matter becomes more complicated in the algebraic case, in that the algebraic set of the ideal $I\langle F\rangle$ could consist of several components, some of which might have different dimensions.

Let us consider the algebraic set $V(I)$ defined by the ideal I in k^n. It is possible that $V(I)$ is the union of two or more point sets, each of which can be defined separately by an ideal. In this case, we say that the set $V(I)$ is *reducible.* The notion is analogous to polynomial reducibility: A multivariate polynomial f that factors describes a surface consisting of several components. Each component belongs to an irreducible factor of f. In the same spirit, the reducibility of an algebraic set $V(I)$ mirrors the fact that we can *decompose* the ideal I into several components, although this no longer looks like polynomial factorization in general. Each such ideal component

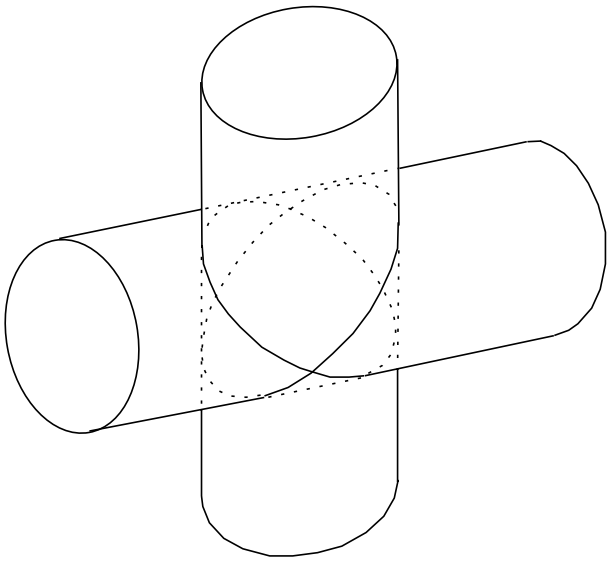

Figure 7.2 Reducible Intersection of Two Cylinders

defines a component of the algebraic set $V(I)$. If an algebraic set $V(I)$ cannot be decomposed, we say that $V(I)$ is a *variety*, or, more simply, that it is *irreducible*.

As an example, consider the intersection curve of the two cylinders

$$\begin{aligned} f_1 : \ x^2 + y^2 - r^2 &= 0 \\ f_2 : \ y^2 + z^2 - r^2 &= 0 \end{aligned}$$

Since the cylinders intersect through their axes and have equal radii, the intersection consists of two ellipses in the planes

$$g_1 : \ x + z = 0 \qquad \text{and} \qquad g_2 : \ x - z = 0$$

as shown in Figure 7.2. Each ellipse can be described separately, as the intersection of one of the cylinders with one of the planes. One of them is the intersection of f_1 with g_1; that is, it is the algebraic set belonging to the ideal generated by $\{f_1, g_1\}$. The other ellipse is the algebraic set of the ideal generated by $\{f_1, g_2\}$. Each ellipse is irreducible. Hence, we can summarize the situation as follows: The ideal $I_1 = I\langle f_1, f_2\rangle$ is reducible, and decomposes into the ideals $I_2 = I\langle f_1, g_1\rangle$ and $I_3 = I\langle f_1, g_2\rangle$; the algebraic set $V(I_1)$ is the union of the two varieties $V(I_2)$ and $V(I_3)$.

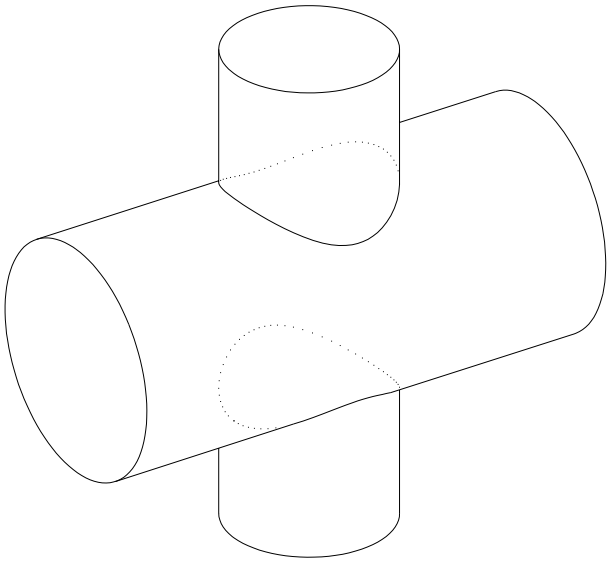

Figure 7.3 Irreducible Intersection of Two Cylinders

Now consider the intersection of two cylinders, one with the radius 1, the other with the radius $\sqrt{2}$.

$$\begin{aligned} f_1 : \; x^2 + y^2 - 1 &= 0 \\ f_2 : \; y^2 + z^2 - 2 &= 0 \end{aligned}$$

Here the intersection curve, shown in Figure 7.3, appears to be reducible. However, it is not reducible, and the two components cannot be defined separately by polynomials. To understand this fact, we recall Bezout's theorem from Section 5.3.3 in Chapter 5. The theorem states that two irreducible surfaces of degree m and n intersect in a curve of degree mn. So, the curve shown in Figure 7.3 has degree 4. The union of two curves of degree m and n is a reducible curve of degree $m + n$. Were the intersection curve shown in Figure 7.3 reducible, the two components would each have to have degree 2, since neither can be of degree 1. But every degree 2 space curve is planar, and the components in the figure evidently are not planar. Hence, the intersection is irreducible.

Recall the earlier assertion that the twisted cubic cannot be defined algebraically as the intersection of two surfaces. It can be shown that this curve has degree 3, since a plane in general position will intersect it in three points. By Bezout's theorem, therefore, the curve would have to be the intersection of a plane and a cubic surface. But the twisted cubic is not a plane curve, so this would be a contradiction.

7.2.6 Prime Ideals and Radicals

Let V be a variety; that is, an irreducible algebraic set in k^n. This means that there is an ideal $I \subset k[x_1, ..., x_n]$ defining V. In fact, there may be several such ideals, for the ideal I may contain redundancies not reflected in V as a set of points. We consider the intersection curve V of the cylinder f_1 above with the plane g_1. We know it is an ellipse, and is a variety. The ideal $J_1 = I\langle f_1, g_1\rangle$ defines V, but so does the ideal $J_2 = I\langle f_1, g_1^2\rangle$. J_1 and J_2 are different ideals, since J_2 does not contain g_1, but contains the higher powers of g_1. On the other hand, every polynomial in J_2 is also in J_1, so J_2 is a proper subset of J_1.

Viewed geometrically, as an algebraic set of J_1, the ellipse V is the intersection of a cylinder with a plane, whereas, understood as an algebraic set of J_2, it is the intersection of a cylinder with the *double* plane $g_1^2 = 0$. In the latter case, we should consider the points of V to count double, once for each of the two planes $g_1 = 0$. Hence, the variety V does not reflect the algebraic *multiplicity* of the ideal elements.

To associate with an algebraic set V a unique ideal, we introduce the notion of *radical ideal*. An ideal I is a radical ideal of V if $V = V(I)$ and I is *maximal*. That is, every other ideal J with $V = V(J)$ is contained in I. Given an algebraic set V, there is a unique radical ideal $I = Rad(V)$ such that $V = V(I)$. If V is a variety, then the radical ideal is a *prime* ideal. A *prime ideal* I has the property that, whenever a reducible polynomial is in I, then at least one of its factors is in I. The ideal J_1 is a prime ideal. The ideal J_2 is not. We can see that J_2 is not prime, because g_1^2 is in J_2, but neither of its two linear factors is.

The distinction between an algebraic set as a set-theoretic object and as an algebraic object is important. Viewed set-theoretically, the points in the set have no multiplicity — hence the concept of radical ideals. Viewed as an algebraic object, the points may have higher multiplicity. The twisted cubic illustrates this distinction. Viewed set-theoretically, we argued that it cannot be defined as the intersection of two algebraic surfaces, because of Bezout's theorem. However, if the curve points are considered as having a higher multiplicity, then the twisted cubic (t, t^2, t^3) is in fact the intersection of

$$y^2 - xz = 0 \cap x^3 - 2xyw + zw^2 = 0$$

The two surfaces are tangent to each other in the curve, so each curve point has multiplicity 2 and Bezout's theorem is satisfied. In general, it is not known whether every algebraic space curve can be defined as the intersection of only two surfaces, in this sense.

Note that the problem should be considered projectively. Again, the twisted cubic illustrates the situation. In affine space, the twisted cubic (t, t^2, t^3) is the intersection of

$$y^3 - z^2 = 0 \cap x^2 - y = 0$$

When embedded into projective space, however, the surfaces intersect at infinity in a triple line.

7.3 Gröbner Bases

An ideal can have many different generating sets. Depending on the use to which we want to put them, some generating sets will be better than others. We consider first in detail the problem of testing whether a given polynomial g is in some ideal I. We consider a class of generating sets that allows conceptually simple algorithms to decide ideal membership. These generating sets are Gröbner bases, and although ideal membership is not a central problem in solid modeling, Gröbner bases are also advantageous for many of the problems that are important to geometric and solid modeling. We seek a solution to the *ideal membership problem.*

> **Problem**
> Given a finite set of polynomials $F = \{f_1, \ldots, f_r\}$ and a polynomial g, decide whether g is in the ideal generated by F; that is, whether g can be written in the form $g = h_1 f_1 + h_2 f_2 + \cdots + h_r f_r$, where the h_i are polynomials.

The difficulty of the problem is to determine the coefficient polynomials h_i. When no special assumptions can be made about the generators, deciding whether g is in the ideal is not easy. For instance, even when the f_i are all quadratic and the polynomial g is quartic, we cannot assume a priori that the coefficient polynomials are of degree 2 or less.

We will solve the ideal membership problem by repeatedly *rewriting* g until g has been simplified to the point where the original question can be answered by inspection. Specifically, we will repeatedly subtract from g multiples of the f_i. Since these multiples are in $I\langle F\rangle$, it is clear that the rewritten g is in the ideal iff g is in the ideal. Moreover, if g is in the ideal, then there must exist some rewriting sequence that reduces g to zero. Whether such a rewriting sequence can be found easily depends on specific properties of the generators.

7.3.1 Lexicographic Term Ordering and Leading Terms

Assume that we are rewriting some polynomial $g \in I\langle F\rangle$ with the goal of reducing it to zero. At each rewriting step, we would like assurance that we are making progress toward this goal. This means that, at each step, we want to obtain a polynomial that is in some sense *simpler* than the preceding one. So, we must define an appropriate notion of "this polynomial is simpler than that one." We develop it from an ordering of individual terms in polynomials. This gives us first a concept for judging whether a single term is more complicated than another one. Moreover, given f and g, we will declare that f is more complicated than g if the most complicated term of g precedes the

most complicated term of f in the term ordering. The term ordering to be introduced now is only one possibility. Later, we will also introduce different orderings.

Assume that all polynomials are in $k[x_1, ..., x_n]$. A product of the form

$$x_1{}^{e_1} x_2{}^{e_2} \cdots x_n{}^{e_n}$$

with $e_i \geq 0$, is called a *power product*. We define a *lexicographic ordering*, written $\prec$, of the power products as follows:

1. $1 \prec x_1 \prec x_2 \prec \ldots \prec x_n$.
2. If $u \prec v$, then $uw \prec vw$ for all power products w.
3. If u and v are not yet ordered by rules 1 and 2, then order them lexicographically as strings.[3]

For instance, with $n = 2$, setting $x_1 = x$ and $x_2 = y$, we have the following ordering of power products:

$$1 \prec x \prec x^2 \prec ... \prec x^k \prec ... \prec y \prec xy \prec x^2y \prec ... \prec y^2 \prec xy^2 \prec ...$$

Every term in a polynomial g consists of a coefficient and a power product. The term whose power product is largest with respect to the ordering $\prec$ is called the *leading term* of g, written $lt(g)$. Among all the terms of g, $lt(g)$ is considered the most complicated term. The leading term consists of the *leading coefficient*, $lcf(g)$, and the *leading power product*, $lpp(g)$.

Definition
The polynomial f is *simpler* than the polynomial g if $lpp(f) \prec lpp(g)$.

Example 7.1: Assuming $x \prec y$, the leading term of $g = 2y^3 - xy^2 + x^2$ is $2y^3$. The leading coefficient of g is 2, and the leading power product is y^3. The leading term of $h = 3xy^3 - x^2 + 1$ is $3xy^3$, with the leading power product xy^3. Since $y^3 \prec xy^3$, we consider h to be more complicated than g. $\diamond$

7.3.2 Rewriting and Normal-Form Algorithms

We are given a polynomial g, and a set of polynomials $F = \{f_1, ..., f_r\}$. We plan to rewrite g using the polynomials in F, simplifying g at each step, until it cannot be further simplified. When g cannot be further simplified, we say

[3]The power product $x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$ precedes $x_1^{b_1} x_2^{b_2} \cdots x_n^{b_n}$ lexicographically as a string if there is $1 \leq r \leq n$ such that $a_r < b_r$ and $a_{r+1} = b_{r+1}, \ldots, a_n = b_n$.

that g is in *normal form* with respect to F. A normal form of g with respect to F is denoted by $NF(g, F)$. The rewriting is done as follows:

Input: A set F of polynomials, and a polynomial g.
Output: A normal form $NF(g, F)$ of g with respect to F.
Method:

1. Set $g_0 = g$ and $i = 0$.
2. For $i = 0, 1, 2, ...$ repeat step 3 until g_i cannot be rewritten; then output g_i and stop.
3. If there is a polynomial f in F such that the leading power product of f divides a power product p in g_i, then rewrite g_i as $g_{i+1} = g_i - buf$, where b is the quotient of the coefficient of p by $lcf(f)$ and $u = p/lpp(f)$.

Note that any term of g_i could be rewritten in step 3.

It can be shown that the rewriting algorithm must terminate. Step 3 eliminates a term in g_i, but it may introduce more new terms, so termination is not immediately obvious. However, observe that, since the cancellation is done with the leading term of f, the newly introduced terms in g_{i+1} must precede the term just eliminated from g_i in the term ordering. Thus, to show termination, we must show that the terms introduced in step 3 cannot form an infinite descending chain in the ordering.

Example 7.2: Let $F = \{y^2 + x^2 - 1, xy - x^2 + 1\}$, let $g = 2y^3 + x^2 - xy^2$, and let $x \prec y$. The leading terms are, respectively, y^2, xy, and $2y^3$. We rewrite g in three steps, obtaining g_3 as a normal form of g with respect to F.

$$\begin{aligned} g = g_0 &= 2y^3 - xy^2 + x^2 \\ \to g_1 &= g - 2y(y^2 + x^2 - 1) &&= -xy^2 - 2x^2y + 2y + x^2 \\ \to g_2 &= g_1 - (-y)(xy - x^2 + 1) &&= -3x^2y + 3y + x^2 \\ \to g_3 &= g_2 - (-3x)(xy - x^2 + 1) &&= 3y - 3x^3 + x^2 + 3x \end{aligned}$$

◇

Note that the normal form is not necessarily *unique*, since there may be more than one $f \in F$ with which to rewrite g in step 2, leading to different sequences of rewriting steps with possibly different outcomes. For example, when using $y^2 + x^2 - 1$ to rewrite g_1, we obtain eventually the normal form $2y - x^3 + x^2 + x$.

If the normal form arrived at by the preceding algorithm is known to be unique, then it can be shown that g is in the ideal precisely when $NF(g, F) =$

0. Therefore, we will look for special generating sets with the property that normal forms are unique.

7.3.3 A Membership Test for Ideals

We would like to use the rewriting method for deciding whether g is in the ideal generated by F. Fortunately, there always exists a set G of polynomials that generates the same ideal as F and has the property that the rewriting algorithm produces unique normal forms. Such a set is called a *Gröbner basis* of the ideal $I\langle F\rangle$. Thus, the ideal membership problem is solved as follows:

Input: A set F of polynomials, and a polynomial g.
Output: "Yes" if g is in the ideal generated by F; "No" otherwise.
Method:
1. Construct a Gröbner basis from F.
2. Compute $h = NF(g, G)$. If $h = 0$, then output "Yes"; otherwise, output "No."

Example 7.3: Consider the ideal generated by

$$F = \{(x-1)^2 + y^2 - 2,\ (x+1)^2 + y^2 - 2\}$$

where $x \prec y$. We ask whether $x - y$ is in the ideal generated by F. So, we first construct the Gröbner basis G for the ideal, as explained later. The basis is

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

Then, we compute the normal form of $x - y$ with respect to G. It is y (i.e., not zero); hence, $x - y$ is not in the ideal generated by F. $\diamond$

Geometrically, F defines two circles, and the algebraic set defined by F consists of the intersection points of these circles. We note that the intersection points are easier to compute from the Gröbner basis G than from F. This will generally be the case for Gröbner bases constructed with the lexicographic term ordering.

7.3.4 Buchberger's Theorem and Construction of Gröbner Bases

There are several algorithms for constructing a Gröbner basis from a given set F based on Buchberger's theorem. Technically, they all rewrite the input polynomials, thereby simplifying them and adding certain polynomials that

are, roughly speaking, least common multiples of the input polynomials. The polynomials added are called *S-polynomials*. Gröbner bases algorithms are not yet widely available, and for this reason we sketch a simple version that can be implemented without great difficulties. The algorithm to be described consists of two conceptual operations:

1. Consider a polynomial, and bring it into normal form with respect to some set of generators G.

2. From certain generator pairs, compute S-polynomials and add their normal forms to the generator set.

Initially, the set G is the input set F of polynomials. Considering each pair of generators, an S-polynomial is constructed for the pair and is rewritten into normal form. If the normal form is not zero, then it is added to G. Eventually, G is transformed into a Gröbner basis in this way.

Note that all coefficient arithmetic must be exact. Floating-point arithmetic would introduce errors that would effectively change the ideal described by the input polynomials. The sensitivity to such arithmetic errors, and the precise consequences of numerical errors on the algebraic sets described, are not understood at this time.

Definition
Let f and g be two polynomials with respective leading power products u_f and u_g. Let w be the least common multiple of these power products, such that $w = v_f u_f = v_g u_g$ for some power products v_f and v_g. Let c_f be the leading coefficient of f, c_g the leading coefficient of g. Then the polynomial

$$S(f,g) = c_g v_f f - c_f v_g g$$

is the *S-polynomial* of f and g, and is denoted $S(f,g)$.

Example 7.4: Let $f = 2x^2y - x + 1$, $g = 3xy^2 - 2y^2 + x$. Then $u_f = x^2y$, $u_g = xy^2$, $v_f = y$, and $v_g = x$. Hence, $S(f,g) = 3yf - 2xg = 4xy^2 - 3xy + 3y - 2x^2$. ◇

The algorithm for computing a Gröbner basis of F is based on *Buchberger's theorem*.

Theorem
Let G be a set of polynomials in $k[x_1, ..., x_n]$. Then the following are equivalent:

1. G is a Gröbner basis.
2. For all $f, g \in G$ we have $NF(S(f,g), G) = 0$.

Thus, the basic idea is to generate S-polynomials from pairs in the set G, and to add their normal forms to G. It can be proved that this process must terminate. The basis computation is now as follows:

Input: A set F of polynomials.
Output: A Gröbner basis G of the ideal generated by F.
Method:

1. Set $G := F$, and let B be the set of all pairs $\{f_1, f_2\}$ of polynomials in G, with $f_1 \neq f_2$.
2. While B is not empty, repeat the following steps. Thereafter stop; G is a Gröbner basis.
3. Delete a pair $\{f_1, f_2\}$ from B, and compute the normal form $h = NF(S(f_1, f_2), G)$.
4. If $h \neq 0$, then add to B all pairs of the form $\{f, h\}$, where $f \in G$, and add h to G.

Example 7.5: We illustrate the algorithm with the set $F = \{f_1, f_2\}$, where

$$\begin{aligned} f_1 &= 2x^2y - x + 1 \\ f_2 &= 3xy^2 - 2y^2 + x \end{aligned}$$

We assume $x \prec y$. Initially, $G = \{f_1, f_2\}$ and $B = \{\{f_1, f_2\}\}$. We begin by removing the pair $\{f_1, f_2\}$ from B, and constructing its S-polynomial

$$S(f_1, f_2) = 4xy^2 - 3xy + 3y - 2x^2$$

Now $S(f_1, f_2)$ is reduced using f_2. After clearing denominators, we obtain a normal form

$$f_3 = 8y^2 - 9xy + 9y - 6x^2 - 4x$$

Then f_3 is added to G, and the pairs $\{f_1, f_3\}$ and $\{f_2, f_3\}$ are added to B. Next, we construct $S(f_1, f_3)$. Two reduction steps using f_1 and clearing of denominators bring $S(f_1, f_3)$ into the normal form:

$$f_4 = -8xy + 8y + 12x^4 + 8x^3 + 9x^2 - 18x + 9$$

We add f_4 to G and add to B the pairs $\{f_1, f_4\}, \{f_2, f_4\}, \{f_3, f_4\}$. Next, we construct $S(f_2, f_3)$. Reduction using f_3, f_1, and f_4 results in the new polynomial

$$f_5 = -8y - 20x^4 - 8x^3 - 15x^2 + 34x - 19$$

The next S-polynomial, $S(f_1, f_4)$ yields the normal form

$$f_6 = 4x^5 + 3x^3 - 8x^2 + 7x - 2$$

At this point in the algorithm, we have a set G consisting of $f_1, ..., f_6$, and a set of pairs B containing 11 unprocessed pairs. Each of these unprocessed pairs will generate an S-polynomial, and each of these S-polynomials can be reduced to zero. Thus, no new pairs are generated, and the set $\{f_1, ..., f_6\}$ is a Gröbner basis. $\diamond$

7.3.5 Improved Basis Construction and Reduced Gröbner Bases

Most of the variants of the algorithm given previously concentrate on eliminating certain pairs from B *before* reducing the S-polynomials constructed from them. A pair can be eliminated if we can show that its S-polynomial must reduce to zero. Other modifications order the pairs in B by various strategies that increase the chances of so eliminating pairs. One such strategy is to remove early those pairs from B whose leading power products have a small least common multiple. These heuristics can result in significant speedups and should be implemented.

We give two criteria for eliminating a pair $\{h_1, h_2\}$ from B. The first criterion is as follows. If there is another polynomial h_3 in G with the property that the leading power product of h_3 divides the least common multiple of the leading power products of h_1 and h_2 in G, and if both pairs $\{h_1, h_3\}$ and $\{h_2, h_3\}$ are not in B, then the pair $\{h_1, h_2\}$ does not need to be considered. Intuitively, the presence of h_3 implies that the S-polynomial of h_1 and h_2 will reduce to zero. For example, consider the situation in the basis construction just after processing the pair $\{f_2, f_4\}$. The next pair $\{f_3, f_4\}$ would generate a least common multiple of xy^2. We try to apply the criterion, using for h_3 the polynomial f_2 whose leading power product divides xy^2. Both $\{f_2, f_3\}$ and $\{f_2, f_4\}$ are not in B, so the criterion applies. In our example, the criterion eliminates four more pairs.

The second criterion to eliminate pairs states that, if the leading power products of f_1 and of f_2 are coprime, then the pair $\{f_1, f_2\}$ is redundant. As an example, consider the pair $\{f_5, f_6\}$ whose leading power products are y and x^5, respectively. Since they are coprime, $S(f_5, f_6)$ must reduce to zero.

So far, the algorithm only adds new polynomials to G. It is possible to remove certain other polynomials during the computation. Briefly, if f can be reduced to zero using the polynomials in $G - \{f\}$, then f is redundant and can be deleted. Moreover, if the normal form of f is not zero, then f can be replaced with its normal form. Here, unprocessed pairs involving f are replaced by pairs involving the normal form of f. When these steps are incorporated, we obtain a *reduced* Gröbner basis that is then *unique*, provided the leading coefficients are scaled by some convention. The reduced Gröbner basis in Example 5, without coefficient scaling, is

$$\begin{aligned} f_5 &= -8y - 20x^4 - 8x^3 - 15x^2 + 34x - 19 \\ f_6 &= 4x^5 + 3x^3 - 8x^2 + 7x - 2 \end{aligned}$$

From now on, we consider only reduced Gröbner bases.

7.3.6 Admissible Term Orderings

We have described the Gröbner basis construction with respect to a lexicographic ordering of terms. Other orderings are possible, and the basis-construction algorithm should be implemented such that it works with every suitable ordering. Most generally, the basis calculation can be based on any admissible term ordering.

Definition
An *admissible* term ordering $\prec_a$ is a total order of power products that satisfies

1. $1 \prec_a x_i$, for all variables x_i.
2. For all power products u, v, and w, $u \prec_a v$ implies $uw \prec_a vw$.

The two major term orderings in current use are the lexicographic and the total degree ordering. Both can be further varied by permuting the variables. For instance, in $k[x, y]$, we can construct either ordering with $x \prec y$ or with $y \prec x$. Moreover, they can be combined in various ways.

The *total degree* ordering, denoted by $\prec_t$, is defined by requiring that all power products of degree n precede the power products of degree $n+1$. Two power products of equal degree are ordered lexicographically. For example, for two variables with $x \prec_t y$, we have

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

The *reverse lexicographic total degree* ordering, denoted by $\prec_r$, is defined by requiring that all power products of degree n precede all power products

of degree $n+1$. Two power products of equal degree are ordered in *reverse* lexicographic order. For example, for two variables with $x \prec_r y$, we have

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

Example 7.6: The Gröbner basis of the set F of Example 7.5 with respect to the total degree ordering is

$$4x^3 - 10xy + 4y + 3x - 3$$
$$8y^2 - 9xy - 6x^2 + 9y - 4x$$

$\diamond$

The ordering used can profoundly influence both the time needed to construct a Gröbner basis and the basis size. In most applications, it appears that using the total degree ordering or the reverse lexicographic total degree ordering is much faster and leads to smaller bases than using the lexicographic ordering. On the other hand, the lexicographic ordering has many useful properties that would make it the ordering of choice in most geometric applications.

One consequence of this situation is current research on *basis conversion*. The idea is to construct a Gröbner basis with the total degree ordering, and then transform this basis to another Gröbner basis with respect to the lexicographic ordering. Algorithms for this conversion are discussed in Section 7.8.

7.4 Solving Algebraic Equations

Many geometric applications require solving a system of algebraic equations. If $F = 0$ is a system of algebraic equations, then constructing a Gröbner basis for the ideal generated by F yields an equivalent system $G = 0$ that has the same solution set but is often easier to solve. In this section, we explore this approach.

Given a system F of algebraic equations, it can be shown that F has no solutions iff 1 is in the Gröbner basis G of the ideal generated by F. This theorem does not require that G be constructed with a special term ordering. However, if we wish to determine actual solutions of the system F, then the term ordering used matters.

7.4.1 Triangularizing Algebraic Equations

A Gröbner basis of I constructed with the lexicographic ordering contains information about the elimination ideals of I, and can be used to solve algebraic equations. Let $I \subset k[x_1, ..., x_n]$ be an ideal. Then the set of polynomials

in I that contain only the variables $x_1, ..., x_r$ is

$$I_r = \{f \in I \mid f \in k[x_1, ..., x_r]\} = I \cap k[x_1, ..., x_r]$$

In the ring $k[x_1, ..., x_r]$, the set I_r is evidently an ideal, and we call it the r^{th} *elimination ideal* of I. As we shall see, these ideals help in solving an algebraic system $F = 0$, and much information about them is implicit in a Gröbner basis for $I\langle F\rangle$, as stated by the following key theorem.

Theorem
Let F be a set of polynomials in the variables $x_1, ..., x_n$, and G be a Gröbner basis for the ideal I generated by F with respect to the lexicographic ordering based on $x_1 \prec ... \prec x_n$. Then, for $1 \leq r < n$, the polynomials $G \cap k[x_1, ..., x_r]$ are a Gröbner basis of the elimination ideal $I_r = I \cap k[x_1, ..., x_r]$.

This theorem implies, roughly, that a lexicographic Gröbner basis is a triangular system of polynomial equations. We use it as follows to solve the system $F = \{f_1 = 0, ..., f_k = 0\}$.

Input: A set $F = \{f_1, ..., f_k\}$ of polynomials in $k[x_1, ..., x_n]$.

Output: All solutions of F in the set X_n if F has finitely many solutions, or a message that F has infinitely many solutions.

Method:

1. Construct a reduced lexicographic Gröbner basis G for $I\langle F\rangle$, with $x_1 \prec x_2 \prec \cdots \prec x_n$.
2. If $1 \in G$, then stop: F does not have any solution.
3. If G does not contain a univariate polynomial g_1 in $k[x_1]$, then stop: The solution to F does not consist of a finite set of points.
4. Let g_1 be a polynomial of lowest degree in $G \cap k[x_1]$, and let $X_1 = \{(\alpha) \mid g_1(\alpha) = 0\}$ be the roots of g_1.
5. Repeat steps 6 and 7 with $i = 2, ..., n$.
6. Initialize X_i to the empty set.
7. For each $(\alpha_1, ..., \alpha_{i-1})$ in X_{i-1}, substitute α_s for x_s in $G \cap k[x_1, ..., x_i]$, where $1 \leq s \leq i-1$. From among the resulting univariate polynomials select one of lowest degree that is not identically zero, say p. Then, let $\beta_1, ..., \beta_r$ be the roots of p. Add to X_i all tuples of the form $(\alpha_1, ..., \alpha_{i-1}, \beta_s)$, where $s = 1, ..., r$.

It can be shown that the polynomial g_1 selected in step 4 is unique, and that the algorithm correctly determines all solutions of F. Note that certain polynomials in $G \cap k[x_1, ..., x_i] - k[x_1, ..., x_{i-1}]$ may vanish for specific values in X_i. The examples that follow illustrate this point.

7.4.2 Finding Surface Intersections

We can use the algorithm to solve nonlinear equations for finding the intersection of algebraic surfaces.

Example 7.7: We compute the intersection of the three cylinders

$$\begin{aligned} x^2 + y^2 - 1 &= 0 \\ x^2 + z^2 - 1 &= 0 \\ y^2 + z^2 - 1 &= 0 \end{aligned}$$

Using the lexicographic ordering with $x \prec y \prec z$, the Gröbner basis G is

$$\{x^2 - 2, y^2 - 2, z^2 - 2\}$$

This system is especially simple since it is diagonal. The roots of the first polynomial are $x = \pm 1/\sqrt{2}$. Substitution into other equations is not necessary, since they do not mention x. The roots of the second and third equation are $y = \pm 1/\sqrt{2}$ and $z = \pm 1/\sqrt{2}$. Thus, we have eight different solutions, given by the eight combinations of solutions to the three equations. $\diamond$

Example 7.8: We compute the intersection of the surfaces

$$\begin{aligned} z^2 + 2yz + 2xz + y^2 + 2xy + x^2 - 1 &= 0 \\ z^2 - 2yz - 2xz + y^2 + 2xy + x^2 - 1 &= 0 \\ z^2 - 2yz + 2xz + y^2 - 2xy + x^2 - 1 &= 0 \\ z^2 + 2yz - 2xz + y^2 - 2xy + x^2 - 1 &= 0 \\ z^2 + y^2 - x - 1 &= 0 \end{aligned}$$

Here, the first four quadratic surfaces are pairs of planes bounding an octahedron. The plane pairs intersect in the six points $(\pm 1, 0, 0)$, $(0, \pm 1, 0)$, $(0, 0, \pm 1)$. The fifth surface is a paraboloid of rotation that passes through five of these points but not through the point $(1, 0, 0)$. The Gröbner basis computed for this set is $G = G_1 \cup G_2 \cup G_3$, where

$$G_1 = \{x^2 + x\}$$

$$\begin{aligned} G_2 &= \{xy,\, y^3 - y\} \\ G_3 &= \{xz,\, yz,\, z^2 + y^2 - x - 1\} \end{aligned}$$

The set X_1 consists of the roots of $x^2 + x$, and is

$$X_1 = \{(-1),\, (0)\}$$

We substitute -1 for x in the basis subset G_2 and obtain $-y$ as the lowest-degree polynomial in y. Its root is 0, so we add the pair $(-1, 0)$ to X_2. Next, we substitute 0 for x and obtain the polynomial $y^3 - y$, with roots 1, -1, and 0, as the lowest-degree nontrivial polynomial. Thus, the final set X_2 is

$$X_2 = \{(-1, 0),\, (0, 1),\, (0, -1),\, (0, 0)\}$$

For the set X_3, we explore the four substitutions for x and y defined by X_2. With three of them, we obtain a linear polynomial in z, with the unique root 0 in each case. The fourth substitution, $(0, 0)$, yields $z^2 - 1$, with roots 1 and -1. Hence, X_3 is

$$X_3 = \{(-1, 0, 0),\, (0, 1, 0),\, (0, -1, 0),\, (0, 0, 1),\, (0, 0, -1)\}$$

These are all the solutions of the original system of equations. $\diamond$

7.4.3 Locating Singularities

In Section 6.5.5 in Chapter 6, we discussed locating singularities of plane algebraic curves, and we described several methods. If the curve coefficients are known precisely, then we can apply the Gröbner basis method to precompute all singularities by solving the system $\{f = 0, f_x = 0, f_y = 0\}$. We give two examples.

Example 7.9: Consider the cubic curve $f = 28y^3 + 26xy^2 + 28y^2 + 7x^2y + 16xy + 7y + x^3/2 + 3x/2$. We considered parameterizing this curve in Section 5.5.3 of Chapter 5. As a rational cubic, it must have a singular point, which we find by solving the system

$$\{f = 0, f_x = 0, f_y = 0\}$$

With the ordering $y \prec x$, we obtain the Gröbner basis $\{2y + 1, x\}$ Hence, f has one singular point, at $(0, -1/2)$. $\diamond$

Example 7.10: Consider the quartic $g = x^4 + x^2y^2 - y^2 - 2x^2 + 1$. Using the lexicographic ordering, the Gröbner basis of $\{g, g_x, g_y\}$ is

$$\{x^4 - 2x^2 + 1, y^2 + 2x^2 - 2, x^2y - y\}$$

Hence, there are two curve singularities, at $(\pm 1, 0)$. $\diamond$

7.4.4 Basis Determination with Symbolic Quantities

The curve g of Example 7.10 is a member of a family of quartics given by

$$f(a) = x^4 + x^2y^2 - y^2 - 2a^2x^2 + a^4$$

with $g = f(1)$. It would be attractive to determine the locus of the singularities of the curve $f(a)$ irrespective of the value of a.

The algorithm presented for solving systems of algebraic equations can be used without difficulties in extension fields. If a is transcendental, then we can compute the Gröbner basis for $\{f, f_x, f_y\}$ over $k(a)$. However, the results of this computation are not necessarily valid when a takes on certain values that are algebraic numbers. The problem is that the necessary coefficient arithmetic may entail computing with polynomials in a that could be zero for certain specific values. For transcendentals, this problem does not arise, since a transcendental cannot be the root of any polynomial.

Consider determining the Gröbner basis for $\{f, f_x, f_y\}$ in $k(a)[x, y]$. The input set to the basis computation is

$$\begin{aligned} f_{1,1} &= x^2y^2 - y^2 + x^4 - 2a^2x^2 + a^4 \\ f_{1,2} &= 2(xy^2 + 2x^3 - 2a^2x) \\ f_{1,3} &= 2(x^2y - y) \end{aligned}$$

We use the lexicographic ordering with $x \prec y$. Before forming any S-polynomials, we simplify $f_{1,1}$, replacing it with $f_{1,1} - yf_{1,3}/2$, and we eliminate the factor 2 from the other two polynomials. We obtain the set

$$\begin{aligned} f_{2,1} &= x^4 - 2a^2x^2 + a^4 \\ f_{2,2} &= xy^2 + 2x^3 - 2a^2x \\ f_{2,3} &= x^2y - y \end{aligned}$$

The S-polynomial of $f_{2,2}$ and $f_{2,3}$ is $f_{2,4} = y^2 + 2a^2x^2 - 2a^2$. After adjoining it, we reduce $f_{2,2}$ by replacing it with $f_{2,2} - xf_{2,4} + 2xf_{2,1}$, and we obtain

$$f_{3,2} = 2(1-a^2)x^3 - 2a^2(1-a^2)x$$

If a is transcendental, then the structure of $f_{2,3}$ is x^3+bx. If $a = \pm 1$, however, then this polynomial is zero. Hence, for $a = 1$, the Gröbner basis could differ structurally from the one obtained for the transcendental a. This is indeed so. When a is transcendental, we obtain the Gröbner basis

$$\{y, x^2 - a^2\}$$

Substitution of $a = 1$ would yield the set $\{y, x^2 - 1\}$. However, as we saw in Example 7.10, the Gröbner basis for $\{f(1), f(1)_x, f(1)_y\}$ is $\{x^4 - 2x^2 + 1, y^2 + 2x^2 - 2, x^2y - y\}$.

So, when considering a family of polynomials with some parameters $a_1, ..., a_m$, the Gröbner basis can be computed over $k(a_1, ..., a_m)$. The results will be valid for transcendental values and for those algebraic values for the a_i for which none of the coefficient polynomials generated during the basis computation vanish. For "exceptional values" for which one or more coefficient polynomials in the a_i vanish, a separate computation is needed.

7.5 Operations on Curves and Surfaces

As we saw, a Gröbner basis for $F \subset k[x_1, ..., x_n]$, constructed with the lexicographic ordering, provides the elimination ideals $I_1, ..., I_{n-1}$ of I at the same time. In the case of zero-dimensional ideals, we used this fact to simplify structurally a set of algebraic equations that we wanted to solve. By triangularizing the system of equations, we reduced the problem to finding the roots of univariate polynomials. We now explore a different aspect of the triangularization procedure using the Gröbner basis construction as a general elimination procedure.

7.5.1 Implicitization and Inversion

If a surface is given parametrically as

$$\begin{aligned} x &= h_1(s,t) \\ y &= h_2(s,t) \\ z &= h_3(s,t) \end{aligned}$$

then its implicit form can be determined by elimination of s and t. In Section 5.6.1 of Chapter 5, we explored using the Sylvester resultant as an elimination tool, and observed that repeated application leads to extraneous factors. These factors are intrinsic because the method is based on projection. For polynomial functions h_i, the Gröbner basis approach achieves simultaneously the elimination of s and t, as well as a surface inversion. We demonstrate the procedure with an example.

Example 7.11: Consider the parametric surface

$$\begin{aligned} x &= st \\ y &= st^2 \\ z &= s^2 \end{aligned}$$

We construct the Gröbner basis with the lexicographic ordering for the corresponding ideal

$$F = \{x - st, y - st^2, z - s^2\}$$

Ordering the variables $z \prec y \prec x \prec t \prec s$, we obtain the Gröbner basis

$$\begin{aligned} G = \{ \quad & x^4 - y^2 z, \\ & tx - y,\ tyz - x^3,\ t^2 z - x^2, \\ & sy - x^2,\ sx - tz,\ st - x,\ s^2 - z \} \end{aligned}$$

In G, the first polynomial, $x^4 - y^2 z$, is the implicit surface form. Note the absence of extraneous factors. Polynomials $tx - y$ and $sy - x^2$ are the first polynomials in the basis that introduce the variables t and s, respectively. They provide an *inversion* of the surface; that is, given a point (x, y, z) on the surface, we can determine its parametric coordinates (s, t) from these polynomials. Moreover, the linearity of these two polynomials in t and s implies that the surface parameterization is *faithful*; that is, to each point (x, y, z) there corresponds only one point (s, t) in parameter space. $\diamond$

7.5.2 Offset Surfaces

In Section 6.3.3 of Chapter 6, we gave a procedure for determining the offsets of an algebraic surface f. The procedure consisted of formulating a family of spheres of radius equal to the offset distance with centers on the surface f, and determining the envelope of this family of spheres. If the algebraic equation of the offset surface is needed, then we must eliminate the generic coordinates of the sphere centers. The elimination can be done using Gröbner bases constructed with the lexicographic ordering.

Example 7.12: We consider offsetting the ellipsoid $2x^2+y^2+z^2-2=0$ by the distance 1. Applying the procedure of Section 6.3.3 in Chapter 6, we obtain the following equations.

$$\begin{aligned}
(x-u_1)^2+(y-u_2)^2+(z-u_3)^2-1 &= 0 && (7.1)\\
2u_1^2+u_2^2+u_3^2-2 &= 0 && (7.2)\\
(x-u_1)u_2-2(y-u_2)u_1 &= 0 && (7.3)\\
(y-u_2)u_3-(z-u_3)u_2 &= 0 && (7.4)
\end{aligned}$$

Equation (7.1) is the sphere defining the offset distance, and equation (7.2) places the sphere's center on the ellipsoid. The other two equations are the derivatives of the sphere in two linearly independent tangent directions. We order the variables x, y, z, u_1, u_2, u_3, and construct the Gröbner basis for these equations, obtaining an implicit form of degree 9 with 32 terms. $\diamond$

Closer inspection of the surface computed in Example 7.12 reveals an extraneous factor y, which is found to be present because of the problem formulation and is not a consequence of applying the Gröbner basis method. For the directional derivatives (7.3) and (7.4), we used the tangent directions

$$\begin{aligned}
t_1 &: \ (-u_2, 2u_1, 0)\\
t_2 &: \ (0, -u_3, u_2)
\end{aligned}$$

They become linearly dependent when $u_2 = 0$. This condition holds on the intersection curve of the ellipsoid with the plane $y = 0$. Consider the spheres centered on that curve. For every point on that curve, the directional derivatives are in the direction $(0, u, 0)$; hence, all points of a sphere that are in the plane $y = 0$ satisfy the differential conditions (7.3) and (7.4), and since these points cover a two-dimensional region in the $y = 0$ plane, we obtain the extraneous factor y. Choosing for the directional derivatives the tangents

$$\begin{aligned}
t_3 &: \ (-u_2, 2u_1, 0)\\
t_4 &: \ (-u_3, 0, 2u_1)
\end{aligned}$$

we obtain the extraneous factor x instead.

We can reformulate the offset problem to avoid the degeneracies. However, since the extraneous factor has such a simple structure, it seems easier just to factor it out.

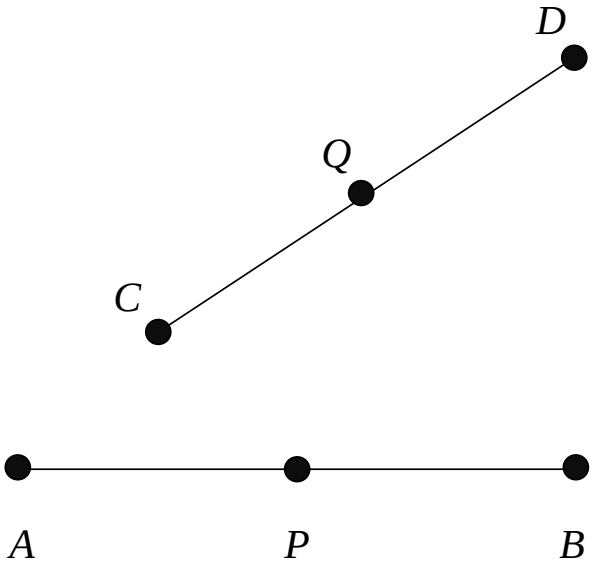

Figure 7.4 Geometric Example Theorem

7.6 Geometric Theorem Proving

Gröbner bases can be used in geometric theorem proving. That is, given a geometric theorem, there are systematic procedures for translating a geometric configuration into an algebraic formula and posing the conclusion of a geometric theorem as a problem of ideal membership whose answer determines whether the theorem holds.

This work is of potential use in robustness problems. As pointed out in Section 4.4 of Chapter 4, coping with numerical uncertainty can be approached as a reasoning problem in that the interpretation of a numerical result is considered to be a logical decision that must be consistent with all other such decisions. Of course, each decision has a geometric meaning — for example, whether two edges intersect, whether a vertex is incident to a face, and so on. If we can account for this geometric meaning, then the symbolic reasoning establishing consistency is related to proving geometric theorems.

In this section, we briefly sketch how the Gröbner basis approach to geometric theorem proving works. We begin with a brief sketch of the method, and illustrate the various technical issues using the following example:

Geometric Example Theorem
If two line segments $\overline{AB}$ and $\overline{CD}$ are congruent, then so are their halves.

Figure 7.4 shows an instance of this theorem.

7.6.1 Outline of the Proof Method

We will translate the geometric theorem into a logical formula of the form

$$(\forall x_1, ..., x_n)((f_1 = 0 \wedge \cdots \wedge f_r = 0) \Rightarrow g = 0) \tag{7.5}$$

where $g = 0$ and the $f_i = 0$ are polynomial equations in the variables $x_1, ..., x_n$. The x_i are point coordinates. The part $(f_1 = 0 \wedge \cdots \wedge f_r = 0)$ expresses the *hypothesis* of the theorem. In our example, the hypothesis asserts that the line segments $\overline{AB}$ and $\overline{CD}$ are congruent. The part $g = 0$ expresses the theorem's *conclusion* — in our example, the congruence of the segment halves.

Formula (7.5) formally states that, for all values $(x_1, ..., x_n)$ for which every $f_j(x_1, ..., x_n)$ is zero, $g(x_1, ..., x_n)$ is also zero. Abstractly, the set of polynomials $F = \{f_1, ..., f_r\}$ generates an ideal $J = I\langle F\rangle$. Associated with J is the algebraic set $V(J)$ consisting of all points $(x_1, ..., x_n)$ for which all polynomials in J vanish. Formula (7.5) states that the hypersurface $g = 0$ contains the algebraic set $V(J)$. Algebraically, therefore, formula (7.5) says that the polynomial g is in the radical ideal of the algebraic set $V(J)$. See also Section 7.2.6.

We prove that g is in the radical ideal $Rad(V(J))$ by assuming that it is not and deriving a contradiction from this assumption. Suppose we can find a point $(x_1, ..., x_n)$ such that all $f_j \in F$ vanish but g does not. Then, we have proved that g is not in the radical ideal $Rad(V(J))$, and the point is a counterexample of the theorem to be proved. That is, we try to satisfy the formula

$$(\exists x_1, ..., x_n)(f_1 = 0 \wedge \cdots \wedge f_r = 0 \wedge g \neq 0)$$

The inequality $g \neq 0$ is transformed into an equality by introducing a new variable z:

$$(\exists x_1, ..., x_n, z)(f_1 = 0 \wedge \cdots \wedge f_r = 0 \wedge gz - 1 = 0) \tag{7.6}$$

Clearly, $gz - 1 = 0$ is possible only for points at which g does not vanish.

In view of the preceding discussion, we consider the ideal generated by the set $\tilde{F} = F \cup \{gz - 1\}$ consisting of all the f_j and the polynomial $gz - 1$. This ideal contains a point $(x_1, ..., x_n, z)$ iff formula (7.5) is false; that is, iff formula (7.6) can be satisfied. In that case, the theorem is not valid.

The decision of whether there are points in the algebraic set of the ideal generated by $\tilde{F}$ is made by investigating whether the system of algebraic equations $\tilde{F} = 0$ has a solution. To find that out, we construct a Gröbner basis G for the ideal $I\langle\tilde{F}\rangle$. There is a solution to the system $\tilde{F} = 0$ iff 1 is not in the basis G. See also Section 7.4.

Proving a geometric theorem with the Gröbner basis approach, therefore, proceeds as follows:

1. Translate the geometric theorem into formula (7.5).

2. By introducing a new variable, change the formula into the format of formula (7.6). Let $\tilde{F}$ be the set of polynomials in this formula.

3. Construct a Gröbner basis for the ideal generated by $\tilde{F}$. The geometric theorem is true iff 1 is not in the basis.

Note that the basis can be constructed with respect to any admissible term ordering.

7.6.2 Translating Geometric Configurations

The basic idea of translating a geometric configuration into a set of algebraic formulae is to assign symbolic coordinates to the points and to express the configuration as algebraic equations in these coordinates.

Let the points of the first segment be A, P, B, and let those of the second segment be C, Q, D, as shown in Figure 7.4 before. The hypothesis of the example theorem is rephrased to make its structure more apparent:

- The points A, P, and B are collinear.
- The points C, Q, and D are collinear.
- The segments $\overline{AB}$ and $\overline{CD}$ are congruent.
- P is the midpoint of $\overline{AB}$.
- Q is the midpoint of $\overline{CD}$.

The conclusion is

- The segments $\overline{AP}$ and $\overline{CQ}$ are congruent.

We explain how each assertion is translated.

Consider three points A, P, and B. With point coordinates $A = (x_A, y_A)$, $P = (x_P, y_P)$, $B = (x_B, y_B)$, we express that the points are collinear:

$$\frac{x_A - x_P}{y_A - y_P} = \frac{x_A - x_B}{y_A - y_B}$$

Clearing the denominator, we have stated collinearity by the polynomial equation

$$(x_A - x_P)(y_A - y_B) - (x_A - x_B)(y_A - y_P) = 0 \tag{7.7}$$

Implicit in the statement "A, P, and B are collinear" is the assumption that at least two of the points A, P, and B are distinct. If this assumption is not expressed by the algebraic formulation, then the translation process is

not faithful, and we risk the possibility of "proving" false geometric theorems. We can express that two points are not coincident as

$$x_A - x_P \neq 0 \vee y_A - y_P \neq 0$$

Therefore, the statement "A, P, and B are collinear" can be expressed by the following conjunction

$$\begin{aligned} & (x_A - x_P)(y_A - y_B) - (x_A - x_B)(y_A - y_P) = 0 \\ \wedge \quad & (x_A - x_P \neq 0 \vee y_A - y_P \neq 0) \\ \wedge \quad & (x_A - x_B \neq 0 \vee y_A - y_B \neq 0) \end{aligned}$$

Note that in this formulation we have expressed that A and B are different points, and that A and P are different points, but we have not expressed that P and B are different points. In the following, other constraints will establish that P and B are not coincident, thereby compensating for the asymmetry of the formula.

To express the congruence of the line segments $\overline{AB}$ and $\overline{CD}$, we state that they have equal length:

$$(x_A - x_B)^2 + (y_A - y_B)^2 - (x_C - x_D)^2 - (y_C - y_D)^2 = 0$$

The fact that P is the midpoint of the segment $\overline{AB}$ is expressed by requiring that the segments $\overline{AP}$ and $\overline{PB}$ have equal length:

$$(x_A - x_P)^2 + (y_A - y_P)^2 - (x_P - x_B)^2 - (y_P - y_B)^2 = 0$$

The entire theorem can now be expressed as the following formula, which is universally quantified in all coordinate variables.

$$
\begin{aligned}
& [(x_A - x_P)(y_A - y_B) - (x_A - x_B)(y_A - y_P) = 0 \\
\wedge \quad & (x_A - x_P \neq 0 \vee y_A - y_P \neq 0) \\
\wedge \quad & (x_A - x_B \neq 0 \vee y_A - y_B \neq 0) \\
\wedge \quad & (x_C - x_Q)(y_C - y_D) - (x_C - x_D)(y_C - y_Q) = 0 \\
\wedge \quad & (x_C - x_Q \neq 0 \vee y_C - y_Q \neq 0) \\
\wedge \quad & (x_C - x_D \neq 0 \vee y_C - y_D \neq 0) \\
\wedge \quad & (x_A - x_B)^2 + (y_A - y_B)^2 - (x_C - x_D)^2 - (y_C - y_D)^2 = 0 \\
\wedge \quad & (x_A - x_P)^2 + (y_A - y_P)^2 - (x_P - x_B)^2 - (y_P - y_B)^2 = 0 \\
\wedge \quad & (x_C - x_Q)^2 + (y_C - y_Q)^2 - (x_Q - x_D)^2 - (y_Q - y_D)^2 = 0] \\
\Longrightarrow & \\
& (x_A - x_P)^2 + (y_A - y_P)^2 - (x_C - x_Q)^2 - (y_C - y_Q)^2 = 0
\end{aligned}
\tag{7.8}
$$

7.6.3 Formula Manipulation

Formula (7.8) is not yet in the format of formula (7.5), because of the inequalities and the disjunctions ($\vee$) in the hypothesis part. We change the formula by first replacing the inequalities with equalities, and then replacing the disjunctions with products.

By introducing the additional variables z_i, $i = 1, ..., 8$, we replace all inequalities in the hypothesis part of formula (7.8) with equalities. For example,

$$x_A - x_P \neq 0 \vee y_A - y_P \neq 0$$

is replaced with

$$(x_A - x_P)z_1 - 1 = 0 \vee (y_A - y_P)z_2 - 1 = 0$$

Each disjunction is next replaced with a product. So, for example,

$$(x_A - x_P)z_1 - 1 = 0 \vee (y_A - y_P)z_2 - 1 = 0$$

is changed into

$$((x_A - x_P)z_1 - 1)((y_A - y_P)z_2 - 1) = 0$$

With these changes, the theorem has been expressed in the required format and can be proved as described.

7.6.4 Choice of Coordinate Axes and Other Heuristics

The process of constructing a Gröbner basis from the set $\tilde{F}$ is facilitated by choosing a suitable coordinate system. For example, the origin of the coordinate system can be placed at the point A, and the x axis laid through the segment $\overline{AB}$. Moreover, after suitable scaling, we can assume that $B = (1, 0)$. The effect of these heuristics is that some variables are eliminated and that some of the polynomials simplify.

Other simplifications are possible. For example, having expressed that $A \neq B$ and P is the midpoint of the segment $\overline{AB}$, it follows that $A \neq P$, so the polynomial expressing this inequality can be deleted. Similarly, $C \neq Q$ is implied. Other heuristics can be formulated based on similar basic geometric observations.

7.7 Complexity

Construction of a Gröbner basis is a potentially time-consuming process. It has been shown that the worst-case complexity is doubly exponential. That is, given a set F in $k[x_1, ..., x_n]$ with highest degree m, the Gröbner basis could contain polynomials of degree proportional to 2^{2^m}. In the case of zero-dimensional ideals, the corresponding bound is 2^m, so this case is more favorable.

When F is in $k[x_1, x_2]$ (i.e., contains bivariate polynomials), more specific complexity bounds are available. If m is the highest degree of any polynomial in F, it can be shown that the degree of any polynomial occurring throughout the basis computation is bounded by m^2 for arbitrary admissible orderings, and by $2m - 1$ for the total degree ordering.

For trivariate polynomials, the following bound is known. Let d be the lowest degree, and m be the highest degree, of any polynomial in the input set F. Then any polynomial generated during the basis computation has a degree of at most $2^d(8m + 1)$, when using the total degree ordering.

Practical experience shows that the running time depends heavily on the variable ordering, and on the choice of the coordinate system. Possible coefficient growth can also influence the running time significantly. All these phenomena are demonstrated now.

7.7.1 Simple Basis Experiments

It is instructive to consider the actual running times for the basis computations done in this chapter. First, we consider Examples 7.3 through 7.11. These are relatively small computations in which variable ordering and term ordering affect the running times only insignificantly. The computations were done on a Symbolics 3650 Lisp machine using the Gröbner basis implementation provided with Macsyma 412.45. Table 7.1 summarizes the results. Times are given in seconds for both the lexicographic and the total degree

Example	Lexicographic	Total Degree
3	0.05	0.05
5, 6	0.16	0.14
7	0.06	0.06
8	0.32	0.37
9	0.23	0.23
10	0.12	0.10
11	0.27	0.21

Table 7.1 Time in Seconds to Construct the Gröbner Basis

orderings. As the values indicate, no significant difference is observed between the lexicographic ordering and the total degree ordering. In many cases, both orderings lead to the same basis, indicating that the ideals with which we deal are structurally extremely simply. Moreover, the polynomials involved are sparse, due to favorably chosen coordinate systems.

7.7.2 Large Basis Computations

The use of basis computations to eliminate variables in curve and surface operations can lead to significantly longer running times. Here, the effect of term orderings becomes very noticeable. It appears that the ideals defined in these problems have a complicated structure, as evidenced by long running times during which many S-polynomials are formed and reduced.

Table 7.2 shows the timings for several runs of Example 7.12. We observe that the running times differ dramatically between basis computations using the lexicographic ordering and basis computations using the total degree

Variable Ordering	Total Degree Ordering	Lexicographic Ordering
x, y, z, u_1, u_2, u_3	1.76	216.54
z, x, y, u_1, u_2, u_3	2.19	787.78
y, z, x, u_1, u_2, u_3	5.81	217.82
x, y, z, u_2, u_3, u_1	2.69	1101.99
x, y, z, u_2, u_1, u_3	3.13	252.20

Table 7.2 Time in Seconds to Construct the Gröbner Basis

ordering. Also, when the variables are arranged differently, the running times can vary significantly.

7.7.3 Coefficient Growth

Gröbner bases algorithms assume exact arithmetic and are therefore implemented using rational arithmetic. Much of the observed time can depend on the size of the rationals manipulated. A simple experiment demonstrates this point.

Consider again the three-cylinder intersection of Example 7.7. We order the variables $x \prec y \prec z$ and construct the basis with the lexicographic ordering in a total time of 0.06 seconds. Next, we rotate the cylinders, first about the z axis by an arc of 1, then about the x axis by an arc of $1/2$, and, finally, again about the z axis by an arc of 1, using single-precision floating-point arithmetic. The coefficients of the three new equations are then converted to rational numbers by Macsyma's RAT function. The three cylinders are now in general position, and their eight points of intersection are expected to have different x coordinates. Due to numerical errors in rotating and in converting to rationals, the cylinders have become slightly elliptic, but there are still eight intersections. We compute the Gröbner basis for these three equations, now consuming 405 seconds, yielding the basis

$$G_1 = \{h_1(x), y - h_2(x), z - h_3(x)\}$$

where h_1 has degree 8, and h_2 and h_3 both have degree 7. That is, the running time has been prolonged by almost four orders of magnitude.

In the longer computation, 231 polynomial pairs are considered. From 30 pairs, S-polynomials are formed and reduced. Using the first criterion of Section 7.3.5, we eliminate 114 pairs. Applying the second criterion, we reject an additional 87 pairs. In large part, therefore, the longer running time is due to the huge rationals involved, which have numerators and denominators with a magnitude of about 10^{700}.

7.8 Basis Conversion

We have seen that a lexicographic Gröbner basis is a very useful data structure that can yield much information about an ideal and its algebraic set. However, experience shows that, in geometric modeling, the known methods for constructing a lexicographic base often demand excessive resources, both in time and in space. This is a serious limitation that should be addressed.

A basic approach to overcoming the inefficiencies of the lexicographic basis construction would be to reformulate the algorithms that use the bases such that they use instead bases constructed with an ordering that leads to better performance. Indeed, constructing a total degree Gröbner basis is an

acceptably efficient computation in many situations. In some cases, we can reformulate the algorithms. For example, there is an algorithm for solving a system of algebraic equations that has finitely many solutions, based on a total degree basis.

Another general idea is to construct first the Gröbner basis using the total degree ordering, and then to *convert* this basis to a Gröbner basis with respect to the lexicographic ordering. If the conversion can be done efficiently, then this method will solve the problem for all algorithms that require a lexicographic basis.

Efficient basis conversion is known to exist for zero-dimensional ideals; that is, for ideals whose algebraic set consists of finitely many points. We explain this method conceptually, and explain a variation of it that can be used for variable elimination.

7.8.1 Computing in the Residue Class Ring

We plan to compute in the residue class ring R_I to reduce certain algebraic computations to linear algebra problems. Recall from Section 7.2.4 that R_I is a vector space over the ground field k. The elements of R_I are equivalence classes; that is, they are sets of polynomials in $k[x_1, ..., x_n]$. Computing with equivalence classes will be reduced to computing with certain *representatives*; that is, in each class, we will identify a unique polynomial and compute with it.

Let p be any polynomial in $k[x_1, ..., x_n]$. The equivalence class of p in R_I is denoted $[p]$. The representative of the class $[p]$ will be denoted by $\bar{p}$.

Given a Gröbner basis G of I, we use $\bar{p} = NF(p, G)$ as the representative of the equivalence class $[p]$ of the polynomial p. It can be shown that

$$NF(p, G) = NF(q, G) \qquad \Longleftrightarrow \qquad [p] = [q]$$

So, for each equivalence class, we have a unique representative. In particular, the representative for the equivalence class of the polynomials in the ideal is zero. We exploited this fact when testing ideal membership in Section 7.3. The following observation is not difficult to prove.

Theorem
Let a and b be numbers in k. Then

$$NF(au + bv, G) = c(aNF(u, G) + bNF(v, G))$$

where c is a nonzero constant.

Applying the theorem repeatedly, we see that, for polynomials u_j and numbers a_j, we have

$$\sum_{j=0}^{m} a_j u_j \in I \Longleftrightarrow \sum_{j=0}^{m} a_j NF(u_j, G) = 0 \tag{7.9}$$

This is true in particular for power products u_i. The importance of formula (7.9) is that it allows us to test ideal membership incrementally using linear dependence tests.

7.8.2 Basis Conversion for Zero-Dimensional Ideals

We sketch an algorithm for the following problem: Given a Gröbner basis G for a zero-dimensional ideal I in $k[x_1, ..., x_n]$ with respect to some admissible order, construct a lexicographic Gröbner basis G' for I. The algorithm proceeds as follows:

1. For $j = 0, 1, 2, ...$, generate power products $u_j = x_1^{e_{1,j}} x_2^{e_{2,j}} \cdots x_n^{e_{n,j}}$ in a suitable order, and compute $\bar{u}_j = NF(u_j, G)$.
2. For each j, test whether there exists a linear dependence

$$\bar{u}_j - \sum_{i=0}^{j-1} a_i \bar{u}_i = 0$$

   If so, add the polynomial p to G', where

$$p = u_j - \sum_{i=0}^{j-1} a_i u_i$$

Two problems must be solved for this algorithm to work. We must generate the power products u_j in such an order that every polynomial p discovered in step 2 is in the lexicographic basis G'. Moreover, we must have a termination criterion. The following theorem addresses the first problem. In conjunction with the fact that, for a zero-dimensional ideal I, the vector space R_I is finite-dimensional, we can then derive a termination criterion.

Theorem
Let U be the set of all power products u that are not a multiple of some leading power product of a polynomial in the Gröbner basis G of I. Then, the equivalence classes $[u]$ of the $u \in U$ are linearly independent and form a basis of R_I.

Note that the theorem does not assume that I is zero-dimensional. However, if I is zero-dimensional, then R_I has finite dimension and the set U is finite.

Conversion Algorithm

We discover the polynomials in G' by generating all power products in increasing lexicographic ordering. Note that this must not be done naively, for then the algorithm would not terminate. For example, there are infinitely many power products x_1^j, $j = 0, 1, 2, ...$, that precede the power product x_2. By the preceding theorem, however, we can skip multiples of leading power products. Let L be the set of leading power products of the polynomials already discovered in the basis G'. Beginning with the power product $x_1^0 \cdots x_n^0 = 1$, we generate the next power product in lexicographic order, but skip all multiples of the power products in L. Deferring the details of power-product generation for the moment, the basis-conversion algorithm is as follows. In it, the function $next(u, L)$ generates the next power product subject to the constraints implied by L.

Input: A Gröbner basis G, with respect to some admissible term ordering, of the zero-dimensional ideal I.

Output: A lexicographic Gröbner basis G'.

Method:

1. Set G', L, U, and U' to empty, and set u to 1.
2. While the basis G' is not complete, do steps 3 through 6. Thereafter, stop; G' is a lexicographic basis for I.
3. Determine $\bar{u} = NF(u, G)$, and test whether $\bar{u}$ is a linear combination of the elements in U'.
4. If $\bar{u}$ is linearly independent, then add u to U, and $\bar{u}$ to U', and skip step 5.
5. Let $\bar{u} = \sum_{j=1}^{m} a_j \bar{u}_j$, for some $\bar{u}_j$ in U'. Add $u - \sum_{j=1}^{m} a_j u_j$ to G', and add u to L.
6. Replace u with $next(u, L)$. The basis G' is complete if $next$ determines that no successor exists.

It can be shown that this algorithm correctly determines the lexicographic basis in a finite number of steps.

Various optimizations should be incorporated in the reduction to normal form and in the determination of linear dependence. With these optimizations, and assuming that G is a reduced basis with respect to the reverse lexicographic total degree ordering, it can be shown that the algorithm requires $O(n^3D^2 + nD^3)$ steps, where n is the number of variables and D is the size of U, assuming that the arithmetic operations on the coefficients require unit cost. Practical experience shows that using basis conversion is much faster than is constructing the lexicographic basis directly.

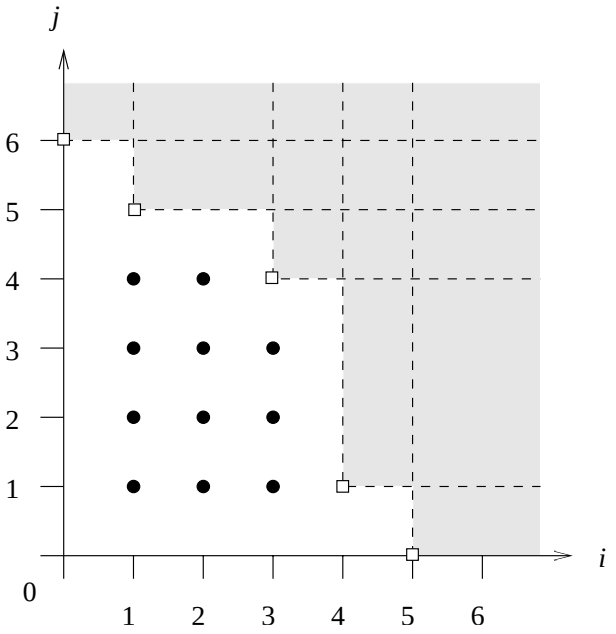

Figure 7.5 Staircase of Leading Power Products

The Staircase

We discuss how to implement the function $next(u, L)$ that generates the next power product in the basis conversion. We restrict our discussion to the bivariate case $n = 2$. The general case is relatively straightforward.

Let G' be a reduced lexicographic Gröbner basis for the ideal I in $k[x, y]$. The leading power products of the polynomials in G' must be relatively prime, and therefore form a *staircase pattern*, as shown in Figure 7.5. In the figure, the point (i, j) represents a power product $x^i y^j$. Five leading power products are shown. Each defines a rectangular area whose points represent power products that are multiples of the power product. These areas are shaded. The set U, therefore, consists of the equivalence classes of all power products belonging to points outside any shaded area. In the figure, the basis is of a zero-dimensional ideal and the set U is finite.

The function $next(u, L)$ has to generate the power products in U in increasing lexicographic ordering. Since the leading power products in G' are not known in advance, *next* will generate them also. However, since a linear dependence is discovered, these power products are then added to the set L instead. Figure 7.6 shows the sequence in which the power products will be generated, assuming that the algorithm discovers the leading power products of Figure 7.5. The function $next(u, L)$ is now as follows:

Input: A power product u, a set of power products L.

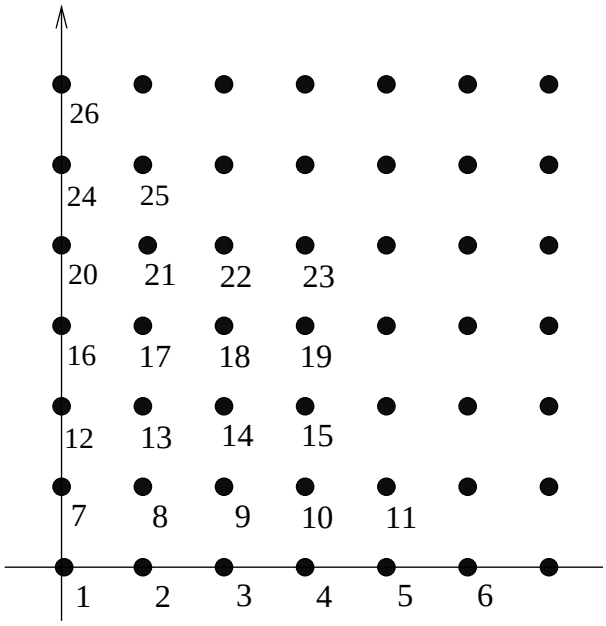

Figure 7.6 Power-Product Generation Sequence

Output: The power product v that is next in lexicographic order and is not a multiple of any $w \in L$.

Method:

1. Set $u_1 = xu$.
2. If u_1 is not a multiple of any power product in L, then return with $v = u_1$.
3. If $u_1 = x^a y^b$ is a multiple of some $w = x^i y^j$ in L and $a = 1$, then stop: No successor of u exists that is not a multiple of some w in L. Otherwise, return with $v = y^{b+1}$.

A Basis-Conversion Example

We illustrate basis conversion with a simple example. Consider the ideal in $k[x, y]$ generated by

$$G = \{x^3 + 2xy - x + 1, y^2 + x - 3\}$$

G is already a Gröbner basis with respect to the total degree ordering, and the ideal $I\langle G\rangle$ can be shown to be zero-dimensional. We generate the monomials $1, x, x^2, \ldots$ with the following normal forms.

$$\begin{aligned} NF(1,G) &= 1 \\ NF(x,G) &= x \\ NF(x^2,G) &= x^2 \\ NF(x^3,G) &= -2xy + x - 1 \\ NF(x^4,G) &= -2x^2y + x^2 - x \\ NF(x^5,G) &= -4xy - 5x^2 + 2y + 13x - 1 \\ NF(x^6,G) &= -4x^2y + 12xy + 13x^2 - 6x + 5 \end{aligned}$$

At this point, we discover a linear dependence among the normal forms:

$$\begin{aligned} NF(1,G) - 2NF(x,G) - 11NF(x^2,G) \\ +6NF(x^3,G) - 2NF(x^4,G) + NF(x^6,G) &= 0 \end{aligned}$$

Hence, we have found a polynomial in G'; namely,

$$1 - 2x - 11x^2 + 6x^3 - 2x^4 + x^6$$

The set U is, at this point, $\{1, x, x^2, x^3, x^4, x^5\}$. We add the leading power product x^6 of p to L. Since x^7 is a multiple of x^6, the next power product generated is y. The monomial y cannot be simplified and is already in normal form. $y = NF(y,G)$ is not linearly independent because

$$\begin{aligned} -NF(1,G) - 11NF(x,G) + 5NF(x^2,G) \\ -2NF(x^3,G) + NF(x^5,G) - 2NF(y,G) = 0 \end{aligned}$$

Therefore, the polynomial

$$-1 - 11x + 5x^2 - 2x^3 + x^5 - 2y$$

is also in G'. So, we add the polynomial to G, and add the leading term, y, to L.

At this point, no lexicographic successor to y can be found that is not a multiple of y or of x^6. In consequence, the algorithm terminates, having found the lexicographic basis

$$G' = \{1 - 2x - 11x^2 + 6x^3 - 2x^4 + x^6,\ -1 - 11x + 5x^2 - 2x^3 + x^5 - 2y\}$$

7.8.3 Variable Elimination

One idea underlying the basis-conversion algorithm is to discover polynomials in I by determining linear dependency relations in R_I. This idea can be applied in a more general way to ideals that are not necessarily zero-dimensional, and it can be used in geometric applications. The general setting is as follows:

> Given an ideal I in $k[x_1, ..., x_n]$ that is known a priori to contain a polynomial p in $k[x_1, ..., x_r]$, for some $r < n$, find such a polynomial p of lowest degree.

Note that we can solve this problem in principle by constructing a lexicographic Gröbner basis. However, constructing a lexicographic basis is often too time consuming, so we seek an alternative. The problem has the following geometric applications:

1. The intersection of two surfaces $f(x, y, z) = 0$ and $g(x, y, z) = 0$ is to be projected onto the (x, y) plane. The ideal I is generated by $\{f, g\}$ in $k[x, y, z]$. The projection p is a polynomial in $k[x, y]$.

2. A parametric surface $x = h_1(s, t)$, $y = h_2(s, t)$, $z = h_3(s, t)$ is to be implicitized. Assuming that the h_i are polynomials, the ideal I is in $k[x, y, z, s, t]$ and is generated by $\{x - h_1, y - h_2, z - h_3\}$. The sought polynomial p is the implicit form of the surface, and is in $k[x, y, z]$.

3. The offset of an implicit surface $f(x, y, z) = 0$ is to be determined. As described in Section 6.3 of Chapter 6, we formulate four polynomial equations defining an ideal I in $k[x, y, z, u, v, w]$. The offset equation is a polynomial p in $k[x, y, z]$.

Other applications are readily formulated in which we are given a system of polynomial equations and seek to eliminate several variables, thus deriving an implied polynomial equation in fewer variables.

The modified algorithm requires a description of the ideal by a Gröbner basis with respect to some admissible term ordering, and is as follows:

Input: A Gröbner basis G for $I \subset k[x_1, ..., x_n]$, and an index $r < n$.

Output: A polynomial $p \in k[x_1, ..., x_r]$ in the ideal, provided such a polynomial exists.

Method:

1. Beginning with 1, generate the power products formed with $x_1, ..., x_r$, in the total degree ordering. For each such power product u, do step 2.
2. Compute $\bar{u} = NF(u, G)$ and test whether there exists a linear dependence between $\bar{u}$ and the normal forms previously generated. If there is a linear dependence, then output the corresponding polynomial defined by this dependence and stop.

Note that this algorithm will not terminate if the ideal does not contain a polynomial p in $k[x_1, ..., x_r]$. Its use is therefore restricted to situations in which p is known to exist. This is the case in many geometric applications.

Example 7.13: Consider the parametric definition of the parabola $x = t$, $y = t^2$. We consider the ideal I generated by $G = \{x-t, y-t^2\}$. With respect to the total degree ordering with $x \prec y \prec t$, G is a Gröbner basis. We seek a polynomial in x and y in I. We generate the following power products and normal forms:

$$\begin{aligned} NF(1, G) &= 1 \\ NF(x, G) &= t \\ NF(y, G) &= t^2 \\ NF(x^2, G) &= t^2 \end{aligned}$$

We discover the linear dependence $NF(x^2, G) - NF(y, G) = 0$, and from it, we obtain the polynomial $y - x^2$ in I. Having found this polynomial, the algorithm stops. $\diamond$

7.9 Notes and References

The algebraic concepts reviewed here are found in most graduate-level texts on algebra and algebraic geometry. However, these texts often present the material in a more concise and abstract form, thereby making it harder for the nonspecialist to access. The algebraic definition of the twisted cubic (t, t^2, t^3) was provided by S. Abhyankar.

The concept of Gröbner bases is due to Buchberger (1965). Buchberger designed and implemented the basis-construction algorithm and investigated many ideal-theoretic applications, including the solvability of systems of algebraic equations, and computations in the residue class ring R_I; see Buchberger (1965 and 1970). For a proof of Buchberger's theorem, see also Buchberger (1976). Buchberger (1985) offers a good survey of Gröbner bases algorithms and their many applications. The chapter contains a very readable introduction to the basis construction and to some of the mathematical applications, including the basis-construction algorithm and the method for solving algebraic equations discussed in this chapter.

Mishra and Yap (1987) analyze the complexity of the basis computation from a computer-science perspective. Buchberger, Collins, and Kutzler (1988) survey applications including geometry theorem proving, and compare Gröbner bases techniques with competing algorithms, such as Wu's method and Collins' quantifier-elimination procedure.

The two criteria given in Section 7.3.5 for avoiding the formation of certain S-polynomials are from Buchberger (1970 and 1976). The theorem on the elimination ideals in Section 7.4.1 is due to Trinks (1978). Kobayashi, Moritsugu, and Hogan (1987) present a modified version to solve systems of algebraic equations with finitely many solutions. Their method determines the next coordinate value from a polynomial of the form $x_i = h(x_1, ..., x_{i-1})$.

In a lecture at Oberwolfach in 1989, V. Weispfenning considered Gröbner bases construction with symbolic quantities and traced the effect of vanishing coefficient polynomials.

Geometric theorem proving is a vigorous research area that is producing a rich spectrum of results. The refutational approach is due to Kapur, and the article in Kapur (1989) gives a good overview of the technique. Our brief presentation is adapted from Kutzler (1988). Kutzler takes great care in devising the translation process such that no implicit assumption is forgotten, and considers the connection to the foundational theories of geometry. Inequalities such as those used to formulate collinearity are known as *nondegeneracy conditions*.

In Wu's approach to geometric theorem proving, presented by Chou (1988), not all such inequalities are formulated. Instead, the method finds certain subsidiary conditions that express these inequalities. In a sense, therefore, the method can make an "approximate" geometric theorem precise, provided the algebraic form of the nondegeneracy conditions is interpreted geometrically. Kapur and Mundy (1988) present a collection of papers from an international workshop on geometric reasoning.

There have been investigations into the effect of floating-point arithmetic on the basis calculations. Errors introduced by floating-point arithmetic affect the basis construction in the sense that new polynomials added to the basis in the course of the computation may have terms different from those that would be present were exact arithmetic used. This implies that certain S-polynomials do not reduce to zero in the floating-point case, and that the ideal therefore will be altered in ways that are difficult to assess geometrically. The presentation by Auzinger and Stetter (1988 and 1989) seems to be the only published work on this topic. In unpublished work, Chuang and Hoffmann found in 1988 that interval arithmetic techniques were not promising, even in cases where the algebraic problem corresponded to well-conditioned transversal surface intersection.

Significant advances in the wider applicability of Gröbner bases should come from the identification of specific subproblems that permit specializing to highly efficient algorithms. The complexity analyses of the bivariate and trivariate cases, in Buchberger (1983) and Winkler (1984), are the first steps in this direction. Basis conversion and its modification are other examples of progress.

The basis-conversion algorithm for zero-dimensional ideals is described in Faugère, Gianni, Lazard, and Mora (1989). It has very good practical performance and permits us to construct much larger lexicographic Gröbner bases than would otherwise be possible.

The modification of the basis-conversion algorithm for eliminating variables in other ideals was conceived in discussions with B. Buchberger and J. Davenport, in December 1988 and January 1989. An experimental version was implemented by J.-H. Chuang and W. Bouma on top of Kapur's Gröbner basis implementation. Preliminary experiments with the algorithm are very

encouraging. Implicit forms of offset surfaces and parametric surfaces have been computed that could not be obtained with the Macsyma implementations of the Sylvester resultant or with the lexicographic Gröbner basis, because of lack of adequate virtual memory.