Comenius University Bratislava
Faculty of Mathematics, Physics and Informatics

Diploma Thesis

2002 Oľga Filipová

Comenius University Bratislava
Faculty of Mathematics, Physics and Informatics
Department of Computer Graphics
and Image Processing

Diploma Thesis

Intersections and Gröbner Basis

Author: Oľga Filipová
Advisor: RNDr. Pavel Chalmovianský PhD.

March 2002

Vyhlasujem, že predkladanú diplomovú prácu
som vypracovala samostatne s použitím uvedenej literatúry
a na základe konzultácií s diplomovým vedúcim.

V Bratislave
dňa 27. marca 2002

I would like to thank my consultant RNDr. Pavel Chalmovianský
for his time, extensive help and advice.
Similarly, I give thanks to my family and friends
for their patience and encouragement.

Contents

List of Figures

Chapter 1

Introduction

Geometric modeling is a discipline, that deals with description of real objects mainly with respect to their geometric properties. This subject field deals with a wide range of tasks – intersections, parameterization, implicitization, inversion, etc. In this diploma thesis we focused only on one of them, namely intersectionsof algebraic curves and surfaces. Geometric modeling uses methods of geometry and informatics and invents also own methods and algorithms.

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, where possible. The resulting discipline of *algebraic geometry* is of use to geometric modeling because it delivers a symbolic representation of geometric objects that allows us to compute with geometric objects using symbolic manipulation. In fact this means algorithms have to accept algebraic equations as input and deliver, as output, other algebraic equations. The role of mathematics is to interpret the result.

In the last decades the development of computer equipment has enabled the application of modern algebra methods in geometry. Same here we file our second target, the theory of Gröbner basis.

In this diploma thesis we deal with algebraic curves and surfaces. Our goal in this work is to study Gröbner basis and their use by algebraic curve/curve and surface/surface intersection computing. We also aim to examine some curve properties (singularities). We present two new methods – one for piecewise implicit curve intersection, the other for surface/surface intersection.

Evaluating the intersection of parametric and algebraic curves and surfaces is a recurring operation in geometric and solid modeling, as well as computer graphics and computer aided design. An efficient surface intersection algorithm that is numerically reliable, accepts general surface models, and operates without human supervision is critical for every of these fields. Notwithstanding the elegance of the Gröbner basis solution we must remark constructing a Gröbner basis can be computationally expensive, 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. However, research on basis conversion has significantly improved the efficiency of this approach.

The work is organized in the following manner. In Chapter 2 we present our notation and introduce some necessary facts about polynomials, curves and surfaces. In Chapter 3 we briefly review techniques for curve/curve and surface/surface intersection. This topic is very wide and we did not intend to immerse very much. Chapter 4 is the ground of our work. Here we define Gröbner basis and introduce algorithms for curve/curve and surface/surface intersection. Marginally we mention a method for singularities evaluation.

Chapter 2

Essentials

2.1 Polynomials and Ideals

Ideal. Nonempty set I of elements of a ring R is called an *ideal* if and only if:

(1) $\forall x, y \in I : x - y \in I$

(2) $\forall x \in I,\ \forall a \in R : ax \in I \land xa \in I$

Univariate polynomial. An *univariate polynomial* (in fact, a polynomial in one indeterminate) over k has the form

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

where x is an indeterminate symbol, and the coefficients a_i are elements of the field k. We frequently fix the field of coordinate values and call it the ground field. In CAD oriented applications usually $k = \mathbb{R}$ or $k = \mathbb{C}$.

The set of all univariate polynomials in x is denoted by $k[x]$. The set $k[x]$ is a ring with operations of multiplication and addition defined as follows:

If $f(x),\ g(x) \in k[x],\ f(x) = \sum_{i=0}^{n} a_i x^i,\ \ g(x) = \sum_{j=0}^{m} b_i x^i$ we define

$$f(x)g(x) = c_0 + c_1 x + \ldots + c_{n+m} x^{n+m},$$

where

$$c_k = \sum_{i+j=k} a_i b_j \text{ for } k = 0, \ldots, n+m.$$

Similarly, assuming $n \geq m$:

$$f(x) + g(x) = (a_0 + b_0) + (a_1 + b_1)x + \ldots + (a_m + b_m)x^m + a_{m+1}x^{m+1} + \ldots + a_n x^n,$$

$$-f(x) = (-a_0) + (-a_1)x + \ldots + (-a_n)x^n.$$

The reducibility of an univariate polynomial. A polynomial that factors nontrivially is called *reducible*. One that cannot be factored is *irreducible*. The *reducibility*, resp. *irreducibility* of a polynomial depends on the ground field k, e.g. the polynomial $x^2 + 1$ does not factor over the reals, but it will factor as $(x - \mathrm{i})(x + \mathrm{i})$ over the complex numbers.

Multivariate polynomial. Similarly, we talk about *multivariate polynomials* (polynomials in more indeterminates). Using the symbols

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

where the coefficients a_j are elements of the ground field k. The exponents $e_{i,j}$ are, of course, nonnegative integers. The set of all multivariate polynomials in the indeterminates $x_1, \ldots, x_n$ is denoted by $k[x_1, \ldots, x_n]$ and forms a ring together with operations of multiplication and addition[1].

The reducibility of a multivariate polynomial. The reducibility of a multivariate polynomial, as well as the reducibility of univariate polynomials, depends on the ground field. But there are also multivariate polynomials that cannot be factored over any ground field. Such polynomials are called *absolutely irreducible*. For example, the polynomial $x^2 + y^2 + z^2 - 1$ is absolutely irreducible (for explanation see [8]).

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 $(x_1, x_2, \ldots, x_n)$, where the x_i take on the values in k. We consider the hypersurface $f = 0$ defined by a multivariate polynomial f. We assume 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 then a single one.

[1] These operations are defined similarly as in the case of polynomials in one indeterminate.

Consider the surface $f = 0$, and let g be a polynomial. All surfaces $gf = 0$ for all $g, f \in k[x_1, \ldots, x_n]$ (f is arbitrary but fixed) will contain the surface $f = 0$[2]. Moreover, for fixed f in $k[x_1, \ldots, x_n]$, the intersection of all surfaces $gf = 0$, where g varies over $k[x_1, \ldots, x_n]$, is precisely the surface $f = 0$. Thus for fixed f, we consider the set

$$I\langle f\rangle = \{gf \mid g \in k[x_1, \ldots, x_n]\}$$

as the description of the surface. This description is not always unique. The reason is explained in [4, p. 268, Chapter 7], .

$I\langle f\rangle$ is an ideal in $k[x_1, \ldots, x_n]$: $I\langle f\rangle$ 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 $k[x_1, \ldots, x_n]$ with an element of $I\langle f\rangle$ is again in $I\langle f\rangle$. Therefore $I\langle f\rangle$ is an ideal.

Now consider a finite set F of polynomials $f_1, f_2, \ldots, f_r$ ($r \in \mathbb{N}$) in $k[x_1, \ldots, x_n]$. We form all *algebraic combinations* of the f_i; that is, we form the set of polynomials

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

Clearly, $I\langle F\rangle$ is an ideal in $k[x_1, \ldots, x_n]$ (apply the same proof as in the former case of $I\langle f\rangle$). We say that $I\langle F\rangle$ is an *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 diploma thesis is to find generating sets that have special properties useful for solving geometric problems.

Extension field. The field L is termed *simple algebraic extension* of field $F \subseteq L$, if exists an element $u \in L$, u is algebraic over F, that the set $F \cup \{u\}$ generates the field $L = F(u)$. (We say: the element u over F generates L.) If u is transcendental over F, $L = F(u)$ we call *simple transcendental extension* of field F.

Let us make some remarks: the set $X \subseteq L$ generates the field L if and only if every subfield L, containing X, equals L. Element $u \in L$ is *algebraic* over a subfield (subring) $F \subseteq L$, if exists a non-zero polynomial f over F such, that $f(u) = 0$; element u is *transcendental* over F, if $0 \neq f \in F[x]$ implies $f(u) \neq 0$.

2.2 Admissible Term Ordering

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

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

[2]In fact, when we say, surface $gf = 0$ contains the surface $f = 0$, we mean point subsets – this will be better explained in Section 2.4.

with $e_i \geq 0$ for $i = 1, \ldots, n$, is called a *power product* or a *monomial*.

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

a. $1 \prec_a x_i$ for all variables x_i.

b. For all power products u, v, and w, $u \prec_a v$ implies $uw \prec_a vw$.

The major term orderings in current use are the following ones[3]:

Lexicographic ordering. We define a *lexicographic ordering*, written $\prec$, of the power products as follows:

a. $1 \prec x_1 \prec x_2 \prec \ldots \prec x_n$

b. If $u \prec v$, then $uw \prec vw$ for all power products w.

c. If u and v are not yet ordered by rules 1 and 2, then order them lexicographically as strings, i.e. the power product $x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}$ precedes the power product $x_1^{b_1} x_2^{b_2} \ldots x_n^{b_n}$ if there is $1 \leq r \leq n$ such that $a_r < b_r$ and $a_{r+1} = b_{r+1}, \ldots$, $a_n = b_n$.

Total degree ordering. 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.

Reverse lexicographic ordering. We define a *reverse lexicographic ordering*, written $\prec_{rl}$, of the power products as follows:

a. $1 \prec x_1 \prec x_2 \prec \ldots \prec x_n$

b. If $u \prec v$, then $uw \prec vw$ for all power products w.

c. If u and v are not yet ordered by rules 1 and 2, then order them reverse lexicographically as strings, i.e. the power product $x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}$ precedes the power product $x_1^{b_1} x_2^{b_2} \ldots x_n^{b_n}$ if there is $1 \leq r \leq n$ such that $a_r > b_r$ and $a_1 = b_1, \ldots$, $a_{r-1} = b_{r-1}$.

Reverse lexicographic total degree ordering. 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 by reverse lexicographic order.

[3] The following orderings we define according to [4, p. 270], in [2, p. 8] a different definition is used.

All these orderings can be further varied by permuting the variables.
As observed in the given examples (lexicographic ordering, reverse lexicographic ordering, ...), term orders ignore the coefficient of a term, so a term order might more properly be called a monomial order.

Leading term of a polynomial. 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 *leading term* of g, written $lt(g)$. This term consists of the *leading coefficient*, $lcf(g)$, and the *leading power product*, $lpp(g)$.

Example 2.2.1. Assuming $1 \prec x \prec y$, let us have a polynomial $h = 2x^4 + 3y^3 + 4x^2y + xy^2 + xy + 1$ over $\mathbb{R}$. This polynomial contains the following monomials: $x^4,\ y^3,\ x^2y,\ xy^2,\ xy,\ 1$. With the lexicographic ordering, we have the following ordering of the monomials: $1,\ x^4,\ xy,\ x^2y,\ xy^2,\ y^3$. Thus, the leading term of h $lt(h)$ according to the lexicographic ordering is $3y^3$, with the leading coefficient $lc(h) = 3$ and the leading power product $lpp(h) = y^3$.

Compare: Applying the total degree ordering, we have the following ordering of the monomials: $1,\ xy,\ x^2y,\ xy^2,\ y^3,\ x^4$. Thus, the leading term of h $lt(h)$ according to the total degree ordering is $2x^4$, with the leading coefficient $lc(h) = 2$ and the leading power product $lpp(h) = x^4$. $\diamond$

Simpler of two polynomials. The polynomial f is *simpler* than the polynomial g if $lpp(f) \prec lpp(g)$.

2.3 Curves and Surfaces

Algebraic surface. Every algebraic surface S in affine 3-space is determined by an implicit equation $f(x, y, z) = 0$, where $f(x, y, z)$ is a polynomial over k[4] in the indeterminates x, y and z.

Geometric degree of an algebraic surface. The geometric degree of an algebraic surface S is the maximum number of intersections between the surface and a line, counting complex, infinite, and multiple intersections. It is a measure of the "waviness" of the surface. This geometric degree is the same as the degree of the defining polynomial f of the algebraic surface S in the implicit definition.

[4] k is an arbitrary field – in solid modeling we usually consider $k = \mathbb{R}$

Algebraic curve. Curves can be classified, according to equations defining them, as *algebraic curves*, which have algebraic equations, and *transcendental curves*, which have equations containing transcendental functions.

Similarly to 3D-algebraic surfaces, an *algebraic plane curve* is given by an implicit equation $f(x, y) = 0$, where $f(x, y)$ is a polynomial in the indeterminates x and y.

An *algebraic space curve* is the common intersection of two or more algebraic surfaces. Although solid modeling usually restricts attention to those space curves that are the intersection of just two surfaces, one should remember that certain space curves cannot be defined algebraically as the intersection of only two surfaces ([4, p. 163, Chapter 5]). (This motivates us to consider later in this work ideals with generating sets that contain more than two polynomials.)

Example 2.3.1. An example of an algebraic curve that cannot be defined as the intersection of only two surfaces is the *twisted cubic* (see Figure 2.1). The parametric definition of this curve is:

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

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

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

(For explanation why two surfaces do not suffice see [4, p. 267–269, Chapter 7].) $\diamond$

Geometric degree of an algebraic plane curve. Similarly to 3D-algebraic surfaces, the geometric degree of an algebraic plane curve is the maximum number of intersections between this curve and a line, counting complex, infinite, and multiple intersections. This geometric degree is the same as the degree of the defining polynomial f of the algebraic plane curve in the implicit definition.

Example 2.3.2. Consider a curve given by an implicit equation $x^3 - 3x^2 + 3y = 0$. This curve is obviously of degree 3 – it is given by a polynomial of degree 3, and the line $y = 1$ cuts this curve in three points (see Figure 2.2). $\diamond$

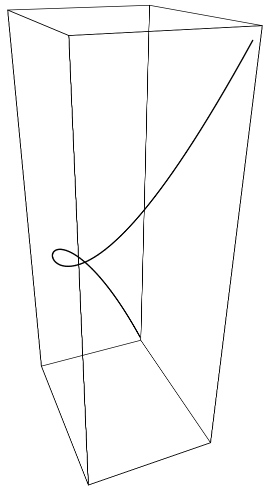

Figure 2.1: Twisted Cubic

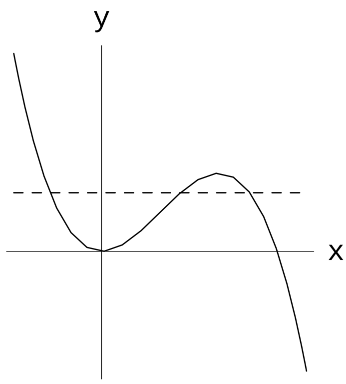

Figure 2.2: Cubic Curve

Typical plane curves of degree 2 are *conic sections*: ellipse, parabola, hyperbola. The general conic is expressed by a general equation of the second degree: $ax^2 + 2hxy + by^2 + 2gx + 2fy + c = 0$.

Geometric degree of an algebraic space curve. The geometric degree of an algebraic space curve is the maximum number of intersections between the specified curve and a plane, counting complex, infinite, and multiple intersections.

Bezout's theorem. Let f and g be two algebraic curves of degree m and n, respectively. If f and g intersect in more than mn points, then they have a common component.

Singularity. The singularity (singular point) of a plane algebraic curve is the simultaneous intersection of $f = 0$, $f_x = 0$, and $f_y = 0$, where f_x, and f_y denote partial derivatives (see [4, p. 280, Chapter 7]).

Examples are *cusps* (see Figure 4.6 on page 35) (cuspidal singularity – a singular point on a curve at which there are two different tangents that coincide), *acnodes* (isolated point – a singular point that does not lie on a given curve but does have coordinates that satisfy the equation of the curve), and *nodes* (a singular point at which the curve intersects itself such that there are two different tangents at the point).

Example 2.3.3. Consider the cubic curve given by an implicit equation $x^3 + y^2 + xy = 0$ (see Figure 2.3). This curve has one singular point—a node—at point $(0, 0)$. $\diamond$

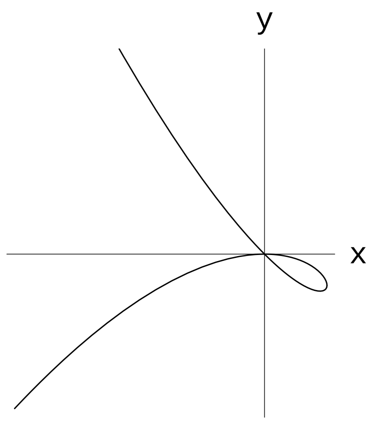

Figure 2.3: Nodal Singularity

2.4 Algebraic Sets and Varieties

Algebraic set. We consider the ideal $I \subset k[x_1, \ldots, x_n]$ generated by the set $F = \{f_1, \ldots, f_r\}$. Let $p = (a_1, \ldots, a_n)$ 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 two dimensions, the algebraic curve $f(x, y) = 0$ is the algebraic set of the ideal $I\langle f\rangle$. Similarly, in three dimensions, the algebraic surface $f(x, y, z) = 0$ is the algebraic set of the ideal $I\langle f\rangle$.

Example 2.4.1. The algebraic set $V(I)$ of $I = I(f)$, where $f(x,y) = x^2 - y$ is the algebraic curve – parabola – $\{(x,y)\colon\ y = x^2\}$ (see Figure 2.4). $\diamond$

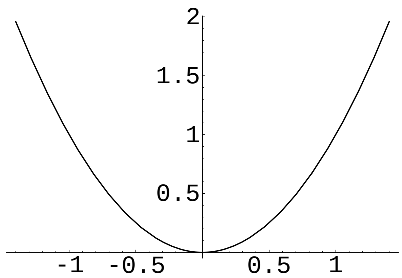

Figure 2.4: Algebraic Set $V(I)$ in Two Dimensions

Example 2.4.2. The algebraic set $V(I)$ of $I = I(f)$, where $f(x,y,z) = x^2 - 0.1y^4 - z$ is the algebraic surface $\{(x,y,z) :\ z = x^2 - 0.1y^4\}$ (see Figure 2.5). $\diamond$

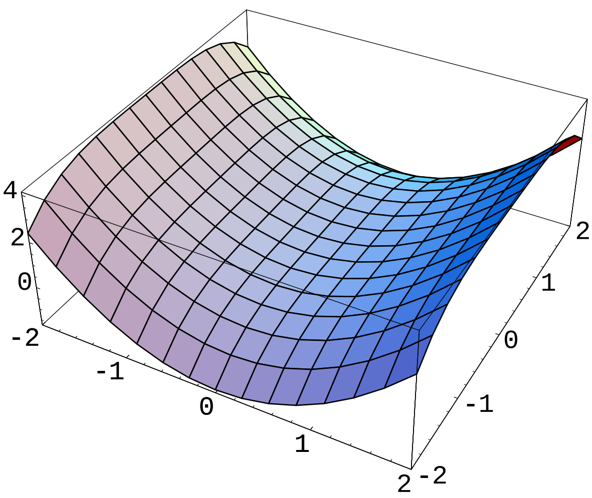

Figure 2.5: Algebraic Set $V(I)$ in Three Dimensions

Dimension of the algebraic set. When we are given a set $F = \{f_1, \ldots, f_r\}$, we expect in general that the algebraic set defined by it in k^n has dimension[5] $n - r$. This is an analogy to linear algebra. This requires that the equations $f_i = 0$ are *algebraically independent*. However, the matter becomes more complicated in the algebraic case: the algebraic set of the ideal $I\langle F\rangle$ could consist of several components, some of which might have different dimensions.

[5]The notion of the dimension is beyond the scope of this work.

Reducibility of the algebraic set. 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 of $V(I)$ 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 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*.

Example 2.4.3. Let us consider an algebraic set $V(I)$ of $I = I(f)$, where

$$f = -x - x^3 + y^2 - xy^2 + x^2y^2 + y^4 - xz + y^2z + z^2 + x^2z^2 + y^2z^2 + z^3$$

is a polynomial over $k = \mathbb{R}$. Since f factors as $(y^2 + z^2 - x)(x^2 + y^2 + z + 1)$, $V(I)$ is the union of two point sets (see Figure 2.6), each given by one of the factors. $\diamond$

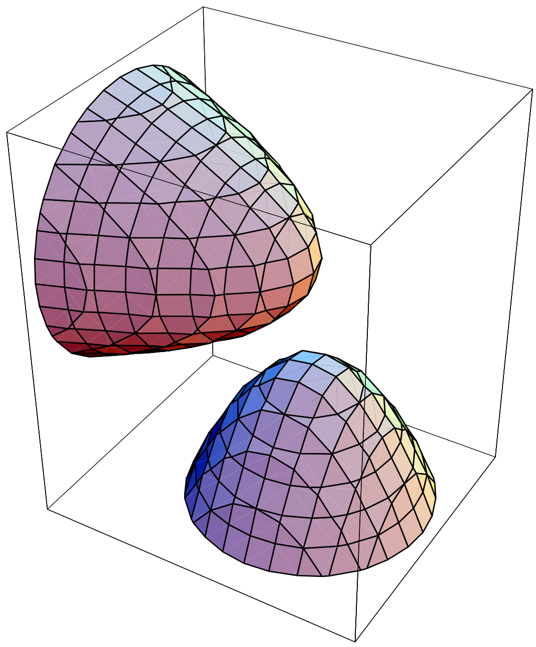

Figure 2.6: Reducible Algebraic Set

2.5 Some Remarks on Implicit Representation

In this work I concern on intersections of implicitly given algebraic curves and surfaces. Thus, I introduce some facts about implicit representation.

- As mentioned above, a real algebraic surface S in $\mathbb{R}^3$ is implicitly defined by a single polynomial equation $\mathcal{F} : f(x, y, z) = 0$, where coefficients of f are over the real numbers $\mathbb{R}$.

- Implicit surfaces pose difficulties in drawing, tessellation, and subdivision, but, because they derive from volumetric functions, they are relatively supple at representing blends. Someway, we notice, techniques to define, represent, process, and display implicit surfaces are increasingly competing with the well-established techniques used for parametric surfaces.

- According to [4, p. 169], while all curves and surfaces with a rational parametric form can be converted to implicit form, at least in principle, only a small subset of these real algebraic curves and surfaces can be expressed in a rational parametric form[6].

- Manipulating polynomials, as opposed to rational functions, is computationally more efficient.

- Algebraic surfaces provide enough generality to accurately model most rigid objects[7].

- An important advantage of the implicit definition $\mathcal{F}$ is its closure property under modeling operations such as intersection, convolution offset, blending, etc. The smaller class of parametrically defined algebraic surfaces is not closed under any of these operations (see [1]). Closure under modeling operations allows cascading repetitions[8] without any need of approximation.

---

[6] A rational parametric definition for a curve is a triple $\mathcal{H}(t) : (x = H_1(t),\ y = H_2(t),\ z = H_3(t))$, where $H_i,\ i = 1, 2, 3$ is a *rational* function in t over $\mathbb{R}$. Similarly for surface, a rational parametric form is a triple $\mathcal{G}(s, t) : (x = G_1(s, t),\ y = G_2(s, t),\ z = G_3(s, t))$, where $G_i,\ i = 1, 2, 3$ is a *rational* function in s and t over $\mathbb{R}^2$.

[7] general topology surfaces

[8] The output of one operation acts as the input to another operation.

Chapter 3

Intersections

The intersection problem in CAGD area can be roughly classified into three major categories:

curve/curve intersection
curve/surface intersection
surface/surface intersection

In the further text we will concentrate on two of these topics – on curve/curve intersections and surface/surface intersections.

The complexity of this topic can be understood partly in some numbers. A generalization of Bezout's theorem states that an algebraic curve of degree m intersects an algebraic surface of degree n in at most mn points (assuming that no part of the curve is common with the surface[1]) and also states that the intersection of a surface of degree m with a surface of degree n is an algebraic curve of degree mn or less [1]. The intersection of two curves of degree m, n respectively are mn or less points. For example, two ellipses (each of degree two) may intersect in at most 4 points (see Figure 3.1).

All intersection algorithms we know face the following complex problems:

- The *robustness* of the algorithm refers to the detection of all curve segments, closed loops, and singularities assuming no numerical errors. The surface intersection problem gets further complicated due to numerical errors present in all finite-precision computation.

- The *accuracy* characterizes numerical stability of the algorithm in the context of floating point arithmetic.

[1] no part of the curve lies entirely on this surface

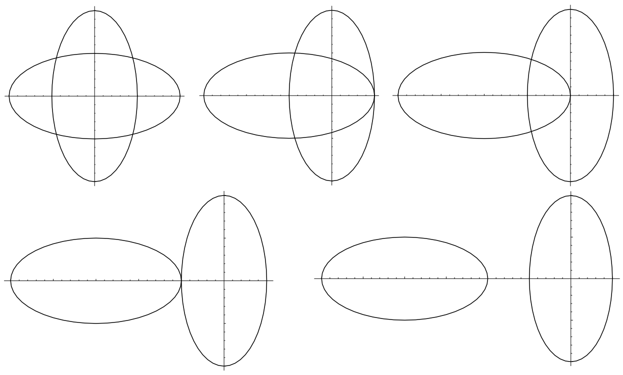

Figure 3.1: Possible Intersections of Two Ellipses

- Intersection algorithms must be *efficient* since they are applied frequently during the design process.

By now, no algorithm meets fully these requirements.

3.1 Curve/Curve Intersection

The problem of computing the intersection of parametric and algebraic curves and surfaces is fundamental to geometric and solid modeling. Previous algorithms are based on techniques from elimination theory or subdivision and iteration.

3.1.1 Previous Work

Parametric Curves. As far as computing the intersection of rational parametric curves is concerned, algorithms based on implicitization, Bézier subdivision and interval arithmetic are well known.

The *implicitization* approach is based on the fact, that every rational parametric plane curve can be implicitized into an algebraic plane curve of the form $F(x, y) = 0$, where $F(x, y)$ is bivariate polynomial. Curve implicitization can be solved using resultant-based method or method based on Gröbner basis (for detailed information on both topics see [4], [12]). Given the implicit representation of one curve, substitute the second parameterization and obtain an univariate polynomial in its parameter. The problem of intersection corresponds to computing the roots of the resulting polynomial. Implicitization based approaches are considered faster than other intersection algorithms for curves of degree up to four [10].

The *Bézier subdivision* is based on the convex hull property of Bézier curves and de Casteljau's algorithm for subdividing Bézier curves. The intersection algorithm compares the convex hulls of the two curves. If they do not overlap, the curves do not intersect. Otherwise the curves are subdivided and the resulting convex hulls are checked for intersection. At each iteration step the algorithm rejects parts of the curve that do not contain intersection points and the new curve segments are better approximated by a straight line. After the two curves segments are approximated by straight lines up to certain tolerance, their intersection points are accepted as the intersection of the curves.

The idea of *interval arithmetic* approach is similar to subdivision. Each curve is preprocessed to determine its vertical and horizontal tangents, and the curve is divided into 'intervals', which have vertical or horizontal tangents only at the endpoints. Thus, we obtain a rectangular bounding box and the subdivision comes to evaluating the coordinate of the midpoint of the interval. The rest is similar to subdivision.

Implicit Curves. The algorithms for algebraic curve intersection are similar to those for parametric curves. Resultants can be used to eliminate one variable from the two equations corresponding to the (plane) curves (see [2], Chapter 3, §1). Then the problem of intersection corresponds to computing roots of the resulting univariate polynomial. This approach causes numerical problems for higher degree curves (greater than four). (The problem of computing real roots of high degree polynomials is ill-conditioned in general.)

In [10] is described an interesting intersection algorithm suitable for both, parametric and algebraic curves, based on the use of resultants – to represent the implicit form of a parametric curve as a matrix determinant – and eigendecomposition. The algorithm for algebraic curves is similar.

3.2 Surface/Surface Intersection

Roughly speaking, the intersection of two surfaces is a curve. In fact, the intersection of two surfaces can be complicated in general, with a number of closed loops, self-intersections and other singularities. A good surface intersection algorithm should, in theory, be able to detect all such features of the intersection curve and trace them correctly in an efficient manner. The different approaches up to now applied to this problem can be categorized according to the main used idea into subdivision, lattice evaluation, analytic methods, and marching methods. However, none of them are able to balance reasonably three conflicting goals of accuracy, robustness, and efficiency.

The components of an intersection curve consist of boundary segments and closed loops. *Start points* on the boundary segments are obtained by curve-surface intersections. Many techniques have appeared over the last few years to detect *closed loops* on the intersection curve (e.g. [11]).

Most algorithms use the local geometry of the curve for *tracing*. These methods do not converge well sometimes and many issues related to choice of step size to prevent component jumping are still open. Therefore, most implementation use very conservative step sizes for tracing and this slows down the algorithm. Overall, current tracing algorithms are not considered robust.

The *singularities* on the intersection curve can be classified in terms of solutions of algebraic equations related to the considered curve equation. Here we suggest to notice a singularity-detection method based on Gröbner basis (for plane curves see Section 4.3.5 on page 34). However, no techniques are known in the literature which can efficiently compute them for high degree surfaces (like bicubic patches) and classify the curve branches in the neighbourhood of the singularity.

3.2.1 Previous Work

There is a significant body of literature addressing the surface/surface intersection problem.

Subdivision method. The basic idea of these methods is to decompose the problem recursively into similar problems which are much simpler. Decomposition continues until a desired level of simplicity is achieved and then the corresponding intersection is obtained directly. The last step is to merge all the individual curves together to get the final solution. (This approach has the flavor of the *divide and conquer* paradigm.) These methods are convergent in the limit but if used for high-precision results lead to data proliferation and are consequently slow. In case subdivision is stopped at some finite steps, it may miss small loops or lead to incorrect connectivity in the presence of singularities. The robustness of this approach can be improved by posing the problem algebraically and using *interval arithmetic* (see [5]).

Lattice evaluation. These techniques decompose surface/surface intersection into series of lower geometric complexity problems like curve-surface intersections. This is followed by connecting the discrete points into curves. Determination of the discrete step size to guarantee robust solution is hard. Further, these technics can be slow and suffer from robustness problem in terms of finding all the small loops and singularities.

Analytic methods. Analytic methods are based on explicit representation of the intersection curve and have been restricted to low degree intersections. Another alternative to the analytic methods are methods using Gröbner basis.

Marching methods. These methods are by far the most widely used because of their generality and ease of implementation. The idea behind marching methods involves analytic formulation of the intersection curve, determination of the start point on each component and the use of local geometry to trace out the curve. The intersection curve can be defined implicitly as an algebraic set based on the surface equations, as a curve of zero distance between the two surfaces, or as a vector field. Tracing can be done on the intersection curve in higher dimensions or on its projection into the plane.

Hybrid Methods. More recently, techniques have been designed that combine features of above different categories. These are generally referred to as hybrid methods.

Chapter 4

Gröbner Basis

Gröbner bases are related to ideals. Ideals are sets of polynomials that describe elementary geometric objects symbolically, and are a natural representation of the geometric objects.

We can often find the solution of a system of linear equations more conveniently by considering linear combinations of the given equations (e.g. this is the principle of the Gaussian elimination). Likewise, when solving systems of algebraic equations, considering algebraic combinations of them may lead to an easier solution. The set of such algebraic combinations is an ideal.

An ideal has many generating sets defining it. Our goal is to study a special set of polynomials defining an ideal – a Gröbner basis. The advantage of Gröbner bases is that many algorithmic problems can be solved easily once a Gröbner basis is known.

4.1 Definition of Gröbner Bases

As mentioned above, an ideal can have many generating sets. Depending on the way of use, some generating sets will be better than others.

Necessary and sufficient condition for $I\langle\{f_1,\ldots,f_n\}\rangle = I\langle\{g_1,\ldots,g_m\}\rangle,\ m,n \in \mathbb{N}$ are $\{f_1,\ldots,f_n\} \in I\langle\{g_1,\ldots,g_m\}\rangle$ and $\{g_1,\ldots,g_m\} \in I\langle\{f_1,\ldots,f_n\}\rangle$.

First we consider 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.

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 rewritten in the form $g = h_1 f_1 + h_2 f_2 + \ldots + h_r f_r$, where the h_i are polynomials.

The difficulty of the problem is to determine the polynomials h_i.

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 if and only if 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.

4.1.1 Rewriting, Normal Form and Membership Test Algorithm

We are given a polynomial g, and a set of polynomials $F = \{f_1, \ldots, f_r\}$. We plan to rewrite g using the polynomials in F, simplifying g at each step, until it cannot be further simplified. Then we say that g is in *normal form* with respect to F, $\mathrm{NF}(g, F)$. The rewriting is done as follows:

Algorithm 4.1.1.
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. i = 0; g_0 = g;
2. For i = 0, 1, 2, ...
3.     If (∃ f ∈ F : lpp(f) divides a power product p in g_i),
4.     then {
5.         u = p/lpp(f);
6.         Denote b the quotient of the coefficient of p by lcf(f).
7.         g_{i+1} = g_i − b u f;
8.     }    //end_then_4
```

It can be shown that the rewriting algorithm must terminate. Intuitively, steps 7 eliminates a term in g_i, but it may introduce more new terms. However, since the cancellation is done with the leading term of f, the newly introduced term in g_{i+1} must precede the term just eliminated from g_i in the term ordering. Thus, to prove termination, we must show that the terms introduced in step 7 cannot form an infinite descending chain in the ordering.

The normal form is not necessarily *unique*, since there may be more than one $f \in F$ with which to rewrite g in step 3, leading to different sequences of rewriting steps with possibly different outcomes.

If the normal form arrived at by the preceding algorithm is known to be unique, than it can be shown that g is in the ideal precisely when $\mathrm{NF}(g, F) = 0$. Therefore, we will look for special generating sets with the property that normal forms are unique.

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 $I\langle F\rangle$. Then the membership problem is solved as follows:

Algorithm 4.1.2.
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. $h =NF(g, G)$;
*3. **If** $(h = 0)$*
*4. **then** output "Yes";*
*5. **else** output "No";*

4.1.2 Buchberger's Theorem and Construction of Gröbner Bases

Definition 4.1.1. 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 $S(f, g)$.

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

Theorem 4.1.3.

Let G be a set of polynomials in $k[x_1, \ldots, x_n]$. 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:

Algorithm 4.1.4.

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

```
 1. G := F;
 2. Denote B the set of all pairs {f1, f2} of polynomials in G, with f1 ≠ f2.
 3. While (B is not empty) {
 4.     Delete a pair {f1, f2} from B.
 5.     h = NF(S(f1, f2), G);
 6.     If (h ≠ 0)
 7.     then {
 8.         For all f ∈ G
 9.             add to B all pairs of the form {f, h};
10.         G = G ∪ {h};
11.     }     // end_then_7
12. }     // end_while_3
13. Output G;     //G is a Gröbner basis.
```

Remark 4.1.1. All coefficient arithmetic must be exact. Floating-point arithmetic would introduce errors that would effectively change the ideal described by the input polynomials.

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. However, the ordering used can substantionaly 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: e.g. a Gröbner basis of I constructed with the lexicographic ordering contains information about the elimination ideals (see Equation 4.1 on page 25) of I, and can be used to solve algebraic equations or, equivalently, find intersection of algebraic surfaces.

4.1.3 Some Properties of Gröbner basis

A Gröbner basis for a system of polynomials is an equivalence system that possesses some useful properties:

- A Gröbner basis of an ideal I is a set of polynomials $\{g_1, \ldots, g_n\}$ such that the leading term of any polynomial in I is divisible by the leading term of at least one of the polynomials $g_1, \ldots, g_n$ (see [12]).

- The determination of a Gröbner basis is very roughly analogous to computing an orthonormal basis from a set of basis vectors and can be described roughly as a combination of Gaussian elimination (for linear functions in any number of variables) and the Euclidean algorithm (it is used to determine the greatest common divisor for univariate polynomials over a field).

4.1.4 Improved Basis Construction and Reduced Gröbner Bases

There are known different modifications of the Algorithm 4.1.4 resulting in significant speedups. Most of the variants of the given algorithm 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.

We give two criteria for eliminating a pair $\{h_1, h_2\}$ from B:

1. 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, 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.

2. If the leading power products of h_1 and h_2 are coprime[1], then the pair $\{h_1, h_2\}$ is redundant.

   It is possible to remove certain other polynomials during the computation:

3. If f can be reduced to zero using the polynomials in $G \setminus \{f\}$, then f is redundant and can be deleted.

4. Moreover, if the normal form of f is not zero, then f can be replaced with its normal form. Also the unprocessed pairs involving f are replaced with pairs involving the normal form of f.

With the algorithm modified in this way, we obtain a *reduced Gröbner basis* that is then *unique*.

Example 4.1.5. We want to determine the reduced Gröbner basis for the set $F = \{f_1, f_2\}$, where

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

We assume $1 \prec x \prec y$ and use the lexicographic ordering. Initially, $G = \{f_1, f_2\}$ and $B = \{\{f_1, f_2\}\}$. We begin with removing the pair $\{f_1, f_2\}$ from B, and constructing its S-polynomial

$$S(f_1, f_2) = -2x^2 - xy - 6y + 2.$$

Since there is no polynomial f in G such that the leading power product of f divides a power product in $S(f_1, f_2)$ ($lpp(f_1) = lpp(f_2) = y^2,\ lpp(S(f_1, f_2)) = xy$),

$$\mathrm{NF}(S(f_1, f_2), G) = S(f_1, f_2) = -2x^2 - xy - 6y + 2 := f_3.$$

Now $G = \{f_1, f_2, f_3\}$, but applying rule 1, the polynomial f_1 is redundant ($f_1 = 3f_2 + f_3$) and can be removed. Thus $G = \{f_2, f_3\}$ and $B = \{\{f_2, f_3\}\}$. We continue removing the pair $\{f_2, f_3\}$ from B, and constructing its S-polynomial

$$S(f_2, f_3) = x - x^3 - 2y - 2xy + 2x^2y + 6y^2.$$

[1]Relatively prime, i.e. describing two monomials that have no divisors in common other than +1 and -1. Thus x^5 and y^4z are relatively prime pairs.

Its normal form looks like

$$\mathrm{NF}(S(f_2, f_3), G) = -22 + 5x + 22x^2 - 5x^3 + 70y := f_4.$$

Thus, $G = \{f_2, f_3, f_4\}$ and $B = \{\{f_2, f_4\}, \{f_3, f_4\}\}$. Applying the rule 4, we want to replace f_2 in G with its normal form $\mathrm{NF}(f_2, G)$:

$$\mathrm{NF}(f_2, G) = 8 + 8x - 3x^2 - 8x^3 - 5x^4 := \tilde{f}_2.$$

Then $G = \{\tilde{f}_2, f_3, f_4\}$ and $B = \{\{\tilde{f}_2, f_4\}, \{f_3, f_4\}\}$. Now we want to replace f_3 in G with its normal form $\mathrm{NF}(f_3, G)$. However,

$$\mathrm{NF}(f_3, G) = 0.$$

According to rule 3, f_3 is redundant and can be deleted. Thus, $G = \{\tilde{f}_2, f_4\}$ and B consists of just one pair: $B = \{\{\tilde{f}_2, f_4\}\}$. Using the rule 2, we can delete the pair $\{\tilde{f}_2, f_4\}$ from B (the leading power products $lpp(\tilde{f}_2) = x^4$ and $lpp(f_4) = y$ are coprime). Since $B = \emptyset$,

$$G = \{\tilde{f}_2, f_4\} = \{8 + 8x - 3x^2 - 8x^3 - 5x^4, -22 + 5x + 22x^2 - 5x^3 + 70y\}$$

is the wanted reduced Gröbner basis. $\diamond$

4.2 Solving Algebraic Equations

The central problem of this chapter, finding the solutions of a system of polynomial equations $\{f_1 = 0, \ldots, f_n = 0\}$, rephrases in fancier language to finding the points of the variety $V(I)$, where I is the ideal generated by $f_1, \ldots, f_n$.

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. It can be shown that F has no solution if and only if 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, to determine actual solutions of the system F, we should use the lexicographic ordering.

Let $I \subset k[x_1, \ldots, x_n]$ be an ideal. Denote by I_r, $r \in \{1, \ldots, n\}$ such a set of polynomials

$$I_r = \{f \in I \mid f \in k[x_1, \ldots, x_r]\} = I \cap k[x_1, \ldots, x_r]. \tag{4.1}$$

In the ring $k[x_1, \ldots, x_r]$, the set I_r is an ideal. We call it the r^{th} *elimination ideal* of I.

Theorem 4.2.1.

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

This theorem is used as follows to solve the system $F = \{f_1, \ldots,\ f_n\} = 0$:

Algorithm 4.2.2.

Input: A set $F = \{f_1, \ldots, f_n\}$ of polynomials in $k[x_1, \ldots, x_n]$.
Output: All solutions of F in the set X_n if F has finitely many solutions, else a message that F has infinitely many solutions.
Method:

1. *Construct a reduced lexicographic Gröbner basis G for $I\langle F\rangle$, with $x_1 \prec \ldots \prec x_n$.*
2. ***If*** $(1 \in G)$
3. ***then*** *stop: F does not have any solution.*
4. ***else*** $\{$
5. ***If*** *(G does not contain an univariate polynomial g_1 in $k[x_1]$)*
6. ***then*** *stop: The solution of F does not consist of a finite set of points.*
7. ***else*** $\{$
8. *Denote g_1 the polynomial of the lowest degree in $G \cap k[x_1]$,*
9. *and $X_1 = \{(\alpha) \mid g_1(\alpha) = 0\}$ // X_1 is the set of the roots of g_1.*
10. ***For*** $i = 2, \ldots,\ n$ $\{$
11. *Initialize X_i to the empty set.*
12. ***For*** *each $(\alpha_1, \ldots, \alpha_{i-1})$ in X_{i-1},*
13. *substitute α_s for x_s in $G \cap k[x_1, \ldots,\ x_i]$, where $1 \leq s \leq i-1$.*
14. ***If*** *this new set does not contain an univariate polynomial g_i in $k[x_i]$)*
15. ***then*** *stop: The solution of F does not consist of a finite set of points.*
16. ***else*** $\{$
    From among the resulting univariate polynomials select one of the lowest degree that is not identically zero, say p.
    Let $\beta_1, \ldots,\ \beta_r$ be the roots of p.
17. $\}$ *// end_else_16*
18. ***For*** $s = 1, \ldots,\ r$
19. $X_i = X_i \cup \{(\alpha_1, \ldots,\ \alpha_{i-1},\ \beta_s)\}$

20. } // end_for_10
21. } // end_else_7
22. } // end_else_4
23. Output X_n; // all solutions of F if F has finitely many solutions

It can be shown that the polynomial g_1 selected in step 8 is unique, and that the algorithm correctly determines all solutions of F.

4.3 Geometric Applications

4.3.1 Finding Intersections of Implicit Curves

In case the curve coefficients are known precisely, we can use the algorithm 4.2.2 to find the intersection of two (or more) algebraic curves. If $f_1 = 0,\ f_2 = 0, \ldots,\ f_n = 0,\ n \in \mathbb{N}$ are the equations of these curves, to find their intersection, we just apply the algorithm in question and solve the respective system $\{f_1 = 0,\ f_2 = 0, \ldots,\ f_n = 0\}$.

Example 4.3.1. We compute the intersection of a circle

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

and an ellipse

$$x^2 - xy + 3y^2 - 1 = 0.$$

We can use the algorithm 4.2.2. In the example 4.1.5 on page 24 we determined the reduced lexicographic Gröbner basis for the respective polynomials:

$$G = \{\tilde{f}_2,\ f_4\} = \{8 + 8x - 3x^2 - 8x^3 - 5x^4, -22 + 5x + 22x^2 - 5x^3 + 70y\}.$$

Obviously, $\tilde{f}_2$ is the polynomial of the lowest degree in $G \cap k[x]$ (it is the only one meeting this requirement). The real roots of this polynomial are $X = \{-1, 1\}$. Substitute these roots respectively for x in polynomial f_4, we solve this polynomial. Thus the set Y of the wanted intersections is:

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

(see Figure 4.1). $\diamond$

This was the case of two dimensions. In three dimensions the situation is only a bit more complicated – instead of one equation per curve, we need (at least) two equations

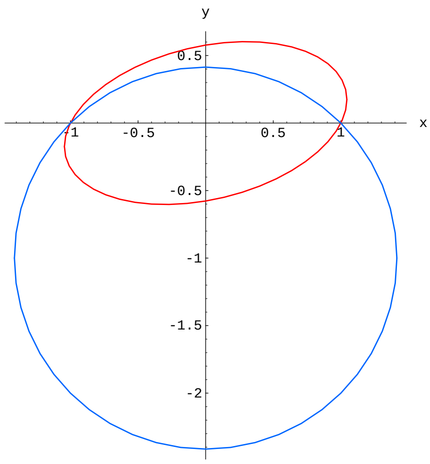

Figure 4.1: Intersection of Circle and Ellipse

to define an algebraic space curve. If we want to find the intersection of curve c_1 given by $f_1 = 0$, $f_2 = 0$ and curve c_2 defined by $g_1 = 0$, $g_2 = 0$, the solvable system is $\{f_1 = 0,\ f_2 = 0,\ g_1 = 0,\ g_2 = 0\}$.

Example 4.3.2. Let us find the intersection of curves c_1, c_2 defined by equations:

$$\begin{aligned} c1 &: x - y = 0,\ x^2 - x^3 - z = 0 \\ c2 &: x^2 - y = 0,\ z - x^3 = 0. \end{aligned}$$

Assuming lexicographic ordering with $1 \prec x \prec y \prec z$, the respective reduced Gröbner basis is $\{x, y, z\}$ and the wanted point of intersection is $(0, 0, 0)$ (see Figure 4.2). $\diamond$

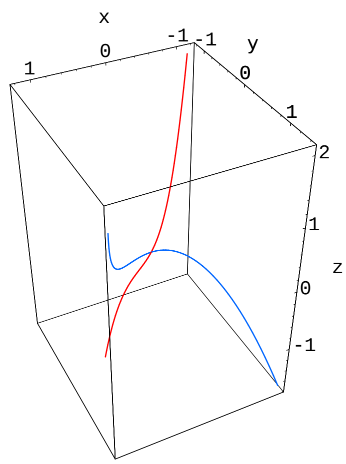

Figure 4.2: Intersection of Space Curves

4.3.2 Implementation of Algorithm 4.2.2

Our results were in practice verified by our package GBInt.m implemented in the application *Mathematica 4* (see Figure 4.3). This package contains 2 functions: GBInt2D, GBInt2Dbbox.

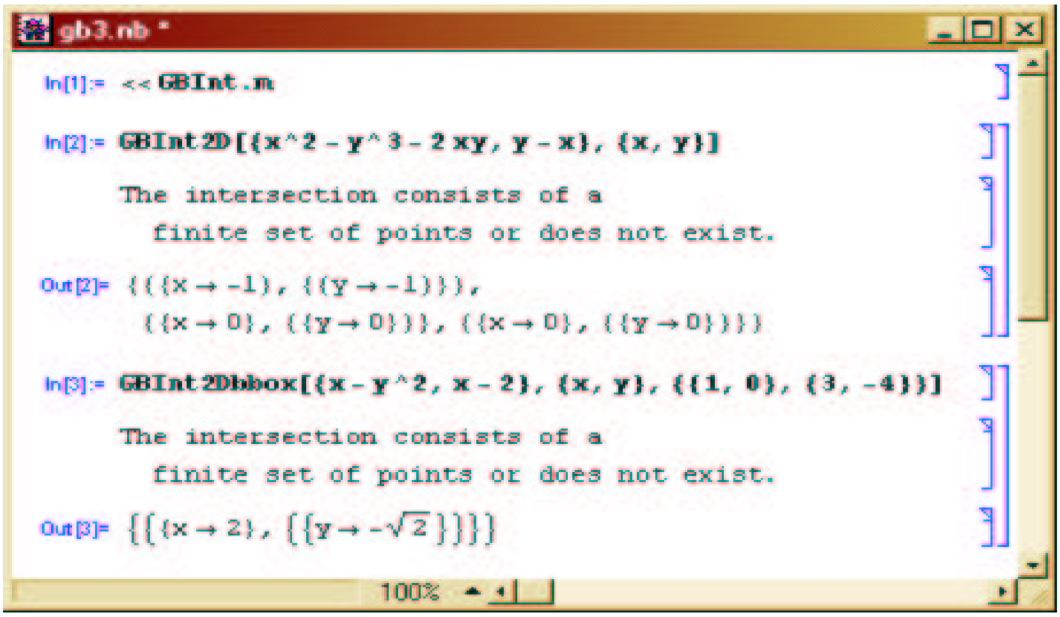

Figure 4.3: Mathematica Package

GBInt2D$[\{poly1, poly2\}, \{x, y\}]$ evaluates intersection points of plane algebraic curves given by the polynomials $poly1$, $poly2$. After entering the input polynomials and the desired order of the indeterminates x, y ($\{x, y\}$ implies $1 \prec x \prec y$), the application reports, whether the curves do intersect or not and whether the intersection consists of finite set.

In case the set of intersection points is finite, the application evaluates the intersection points and outputs the result in the following form: to every x-value assigns the respective set of y-values, e.g. if the set of intersection points is $(0,0)$, $(0,1)$, $(0,2)$, $(3,3)$, the output will be: $\{\{x \to 0\}, \{\{y \to 0\}, \{y \to 1\}, \{y \to 2\}\}, \{x \to 3\}, \{\{y \to 3\}\}\}$.

The following example illustrates the usage of this module:

Example 4.3.3. We compute the intersection of the algebraic curve $x^2 - y^3 - 2xy = 0$ with the line $y = x$ (see Figure 4.4). Assuming lexicographic ordering with $1 \prec x \prec y$, for the set $\{x^2 - y^3 - 2xy,\ x - y\}$, we input

$$\text{GBInt2D}[\{x^2 - y^3 - 2xy, x - y\}, \{x, y\}].$$

This gives the result

$$(-1,-1),\ (0,0),\ (0,0),$$

since the intersection at point $(0,0)$ is double. $\diamond$

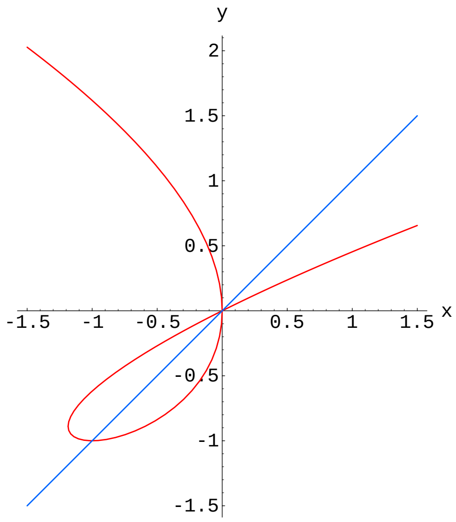

Figure 4.4: Curve/Curve Intersection

The second function GBInt2Dbbox$[\{poly1, poly2\}, \{x,y\}, \{\{p1x, p1y\}, \{p2x, p2y\}\}]$ is similar, it evaluates intersection points of plane algebraic curves given by the polynomials $poly1$, $poly2$ applying the desired order of indeterminates. More over it tests, whether the intersection points are inside the bounding box with sides parallel with the coordinate axes and given by two points $(p1x, p1y)$, $(p2x, p2y)$.

4.3.3 Finding Intersections of Piecewise Implicit Curves

Piecewise implicit curve $\mathcal{C}$ is a curve composed from implicit curves $c_1,\ c_2,\ \ldots,\ c_n,\ n \in \mathbf{N}$, with the endpoints specified for every curve c_i, $(i \in \mathbf{N})$. The "last" point of the curve c_i equals the "initial" point of c_{i+1} for $i \in \{1, \ldots n-1\}$. Endpoints are also called *junction points* or *joints*.

The simplest way how to define endpoints of the curve (or the whole curve) is parametrizing this curve (convert the implicit representation of the curve into parametric one).

However, not all implicit algebraic curves can be expressed in a rational parametric form (confer Section 2.5 on page 13). A complete characterization of this curve property is given by the following *Noether's theorem* [4, p. 169]:

Theorem 4.3.4. *A plane algebraic curve $f(x, y) = 0$ possesses a rational parametric form iff f has genus 0.*

Roughly speaking, the curve genus measures the difference between the actual number of double points (singularities) of f and the maximum number of double points a curve of the same degree as f may have. It can be proved, that a plane curve of degree n can have no more than $(n-1)(n-2)/2$ double points[2]. However, determining the number of double points of f is difficult. Algorithms for determining the genus exist but are nontrivial.

Since our work deals with general implicit representations, we did not want to limit ourselves only to implicit forms that can be parametrized.

Thus, our solution is as follows: every implicit curve c_i ($i \in \{1, \ldots n\}$) of the piecewise implicit curve $\mathcal{C}$ will be defined by an algebraic equation (in the case of plane curves, for space curves c_i will be defined by two or more algebraic equations) and by a bounding box. The bounding box is a rectangle (in 3D-case a box), not necessarily parallel with the coordinate axes, intersecting the segment of the curve it defines in just two intersection points – the endpoints of this segment (see Figure 4.5).

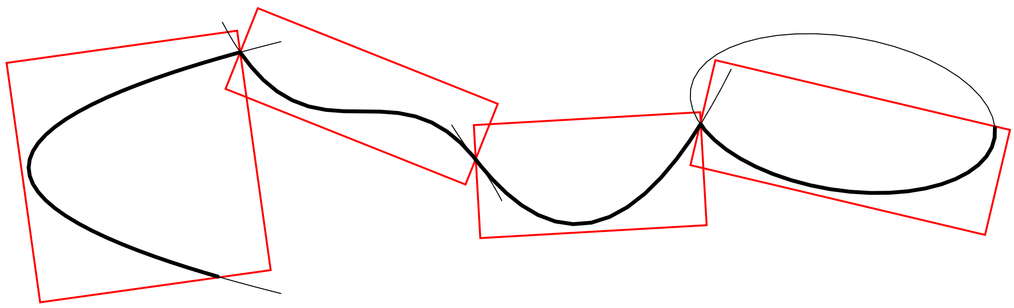

Figure 4.5: Piecewise Implicit Curve

In this point, some problems arise: not every segment of the piecewise implicit curve can be enclosed in one box (in both cases—2D as well as 3D); then we must define several bounding boxes to overcome this.

[2]We see immediately that all plane curves of degree one and two have genus zero and thus can be parametrized using rational polynomial functions. A degree three plane algebraic curve is rational only if it has a double point.

Thus, the intersection of two piecewise implicit curves $\mathcal{C}$ and $\mathcal{K}$ ($\mathcal{C}$ is a curve composed of implicit curves c_1 (denote its bounding box B_1), c_2 (with bounding box B_2), ..., c_n (with B_n), $n \in \mathbf{N}$, $\mathcal{K}$ is a curve composite from implicit curves k_1 (with bounding box H_1), k_2 (with H_2), ..., k_m (with H_m), $m \in \mathbf{N}$) is evaluated as follows:

Algorithm 4.3.5.
Input: Implicit curves $c_1, \ldots, c_n$ with respective bounding boxes $B_1, \ldots, B_n$ and the second set of implicit curves: $k_1, \ldots, k_m$ with respective bounding boxes $H_1, \ldots, H_m$.
Output: Set X of all points of $\mathcal{C} \cap \mathcal{K}$ if this intersection consists of a finite set of points, else a message that $\mathcal{C} \cap \mathcal{K}$ has infinitely many points.
Method:

1. $X = \emptyset$;
2. ***For*** $i = 1, \ldots, n$
3.   ***For*** $j = 1, \ldots, m$ {
4. $X_p = \emptyset$;
5.   *compute the intersection of* c_i, k_j;
6.   ***If*** $(c_i \cap k_j)$ *consists of an infinite set of points*
7.   ***then*** *stop;*
8.   ***else*** {
9. $X_p = c_i \cap k_j \cap B_i \cap H_j$;   // *take only intersection points lying in the respective bounding boxes*
10. $X = X \cup X_p$;
11.   }   // *end_else_8*
12.   }   // *end_for_3*
13. *Output* X;   // *all intersection points of piecewise implicit curves* $\mathcal{C}$, $\mathcal{K}$, *if* $\mathcal{C} \cap \mathcal{K}$ *has finitely many points*

We highlight the main advantage of our solution, that the pure definition of the piecewise implicit curve in our sense evokes the idea of improvement of intersection evaluation – acceleration using bounding boxes. The idea of the acceleration is that we compute the intersection of only those curves, for which intersection of their bounding boxes is unempty. Obviously, counting the intersection of rectangles or boxes is cheaper than using algorithm 4.2.2.

4.3.4 Finding Surface Intersections

Many geometric applications require finding intersections of algebraic surfaces. A sophisticated way to deal with this task is to use Gröbner bases. We need to solve algebraic equations. We can use the algorithm 4.2.2 to solve these generally nonlinear equations.

Algorithm 4.2.2 solves the case of finite intersection (finite set of points) of these (two or more) surfaces. However, the intersection of two surfaces is usually a curve. Here, algorithm 4.2.2 needs our assistance: the following example illustrates our technique.

Example 4.3.6. We want to determine the intersection of two surfaces

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

Assuming lexicographic ordering with $1 \prec x \prec y \prec z$, the reduced Gröbner basis of the respective set of polynomials is:

$$G = \{2x - x^2,\ 2x + y - z\}.$$

According to algorithm 4.2.2, this intersection does not consist of a finite set of points. Thus we have to make an essay of other method.

Let us denote the interrogated polynomials

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

Our method is based on idea of "cutting" the two surfaces $f_1 = 0,\ f_2 = 0$ in an "adequate" way with a series of planes $f_a = 0$, where a is a parameter, $a \in A$, A is a set of parameters. That is, we compute the intersection $f_1 = 0\ \cap\ f_2 = 0\ \cap\ f_a = 0$. "Adequate" way means, the intersection $f_1\ \cap f_2\ \cap\ f_a$ for every $a \in A$ is just a finite set of points (the planes should be chosen in the way, no part of the intersection curve lies in any of the planes f_a). Here we choose the planes $f_a = y - a$. Let us specify the set A of parameters later.

The idea behind is, we want to determine the intersection $f_1\ \cap f_2\ \cap\ f_a$ irrespective of the value of a.

Let us denote: $F := \{f_1, f_2, f_a\}$. Applying the Algorithm 4.2.2, we get "two" results

$$\{x = 0, \quad y = a, \quad z = a\} \tag{4.2}$$

$$\{x = 2, \quad y = a, \quad z = a + 4\}. \tag{4.3}$$

(The respective reduced Gröbner basis for the ideal generated by F is $G = \{2x - x^2, -a + y, a + 2x - z\}$.)

If we look at the equations 4.2, 4.3 a little bit more closely, we realize the first one is a parametric representation of a line in a parameter a, the second one is a parametrization of other line (also in parameter a). Thus, the intersection of surfaces $x^2 + y - z = 0$, $2x + y - z = 0$ are two lines with respective parametric representations 4.2, 4.3 and the set of parameters $A = \mathbf{R}$. $\diamond$

We found a method, that is an improvement of algorithm 4.2.2. Algorithm 4.2.2 was limited to intersections resulting in finite set of points. Our method accepts implicit surfaces as input and yields parametric curves as output. (In general, of course, the intersection of two surfaces is a curve.)

Our algorithm uses idea called *basis determination with symbolic quantities* (see [4, p. 281-282][3]). This means, we do not compute a special Gröbner basis for every value of a parameter a (in our case this is not possible, since the parameter a is a variable over $\mathbf{R}$), but we compute just one Gröbner basis with a as a symbolic quantity.

However, our method faces one problem: It can be nontrivial to find planes satisfying the requirement, that no part of the intersection curve lies in any of the planes f_a, $a \in A$.

4.3.5 Locating Singularities

The singular point of a plane algebraic curve can be found iteratively or by direct methods. If the curve coefficients are known precisely, then we can apply the Gröbner basis method (algorithm 4.2.2) to precompute all singularities by solving the system $\{f = 0,\ f_x = 0,\ f_y = 0\}$ (see the following example).

Example 4.3.7. Consider the cubic curve $f = 0$ (see Figure 4.6), where $f = x^2 - xy^2 - y^3$. We look for a singular point of this curve. We find it by solving the system

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

For f we have $f_x = 2x - y^2$ and $f_y = -2xy - 3y^2$. With the ordering $1 \prec x \prec y$, we obtain the reduced Gröbner basis $\{x,\ y^2\}$. Hence, f has one singular point, at $(0, 0)$. $\diamond$

[3] In [4] is this technique used to determine singularities of every curve of a family of curves – specifically quartics.

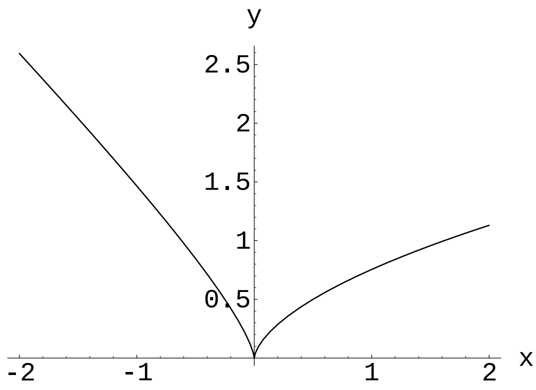

Figure 4.6: Singularity – Cusp

In [4, p. 281-282] is described the way, how to use the formerly mentioned method of basis determination with symbolic quantities (confer Section 4.3.4) to determine singularities of every curve of a family of curves.

Chapter 5

Conclusion and Future Work

In this diploma thesis we have briefly reviewed the main curve/curve and surface/surface intersection techniques. We have presented Gröbner bases and methods for solving algebraic equations, described our package – with two functions for intersection of planar curves – implemented in the application *Mathematica 4* and introduced our algorithms – one for piecewise implicit curves intersection, the other for surface/surface intersection.

It should be mentioned, we did not entirely cover the topic of Gröbner bases. Gröbner basis techniques are capable of solving a wide spectrum of difficult and important problems, like implicitization, inversion and parametrization.

In the future we would like to target on interrogating common tangents and curvatures of parametric curves and the common enhancement of our two algorithms– we want to find algorithm for piecewise implicit surfaces intersection.

References

[1] Bloomenthal, J. — et al.: *Introduction to Implicit Surfaces*, Morgan Kaufmann Publishers, Inc. 1997. ISBN 1-55860-233-X

[2] Cox, D. — Little, J. — O'Shea, D.: *Using Algebraic Geometry*, Springer-Verlag New York, Inc. 1998. ISBN 0-387-98487-9

[3] Filipová, O.: *Gröbner Basis*, Spring Conference on Computer Graphics, Proceedings, 2001.

[4] Hoffmann, C. M.: *Geometric and Solid Modeling*, Morgan Kaufmann Publishers, Inc. 1989. ISBN 1-55860-067-1

[5] Hu, C.-Y. — Maekawa, T. — Patrikalakis, N.M. — Ye, X.: *Robust Interval Algorithm for Surface Intersection*, Computer-Aided Design, Vol.29, No. 9, pp. 617-627, 1997.

[6] Ježek, F. — Tomiczková, S. — Bastl, B.: *Geometrické modelování a jeho aplikace*, Aplimat, preprint. 2002.

[7] Katriňák, T. — et al.: *Algebra a teoretická aritmetika*, Alfa, Bratislava – SNTL, Praha 1985.

[8] Katriňák, T.: *Algebra 2 – lecture notes*, FMPI CU Bratislava, 1998/99.

[9] Krishnan, S. — Manocha, D.: *An Efficient Surface Intersection Algorithm Based on Lower Dimensional Formulation*, ACM Transaction on Graphics, 16(1):74-106, 1997.

[10] Manocha, D. — Demmel, J.: *Algorithms for Intersecting Parametric and Algebraic Curves I: Simple Intersections*, to appear in ACM Transaction on Graphics, 1993.

[11] Krishnan, S. — Manocha, D.: *Symbolic-Numeric Methods of Loop Detection for Curve and Surface Interrogations*, 1996.

[12] Sederberg, T. W. — Zheng, J.: *Algebraic Methods for Computer Aided Geometric Design*