1992

Implicit Curves and Surfaces in CAGD

Christoph M. Hoffmann
Purdue University, cmh@cs.purdue.edu

Report Number:
92-002

IMPLICIT CURVES AND SURFACES IN CAGD

Christoph M. Hoffmann

CSD-TR 92-002
January 1992

Implicit Curves and Surfaces in CAGD

Christoph M. Hoffmann*
Computer Science Department
Purdue University

Abstract

We review the role of implicit algebraic curves and surfaces in computer-aided geometric design, and discuss its possible evolution. Implicit curves and surfaces offer certain strengths that complement the strength of parametric curves and surfaces. After reviewing basic facts from algebraic geometry, we explore the problems of converting between implicit and parametric forms. While conversion from parametric to implicit form is always possible in principle, a number of practical problems have forced the field to explore alternatives. We review some of these alternatives, based on two fundamental ideas.

First, we can defer the symbolic computation necessary for the conversion, and map all geometry algorithms to an "unevaluated" implicit form that is a certain determinant. This approach negotiates between symbolic and numerical computation, placing greater stress on the numerical side.

Second, we can sidestep all symbolic computations by not even formulating an implicit form, but rather using a more general system of nonlinear equations. Doing so simplifies a number of otherwise difficult geometric operations, but requires developing a separate algorithmic infrastructure. This second approach generalizes both implicit and parametric forms.

Introduction

By far the most common representation for curves and surfaces in *Computer-Aided Geometric Design* (CAGD) is the *parametric* representation, as is evident from the literature. The reasons are not only historic, but are also rooted in a well-established body of work that elegantly relates intuitive geometric shape with the mathematical representation, and that clarifies approximation and interpolation properties of specific classes of parametric curves and surfaces. Nevertheless, CAGD also studies implicit algebraic curves and surfaces, for this

*Supported in part by ONR Contract N00014-90-J-1599, by NSF Grant CCR 86-19817, and by NSF Grant ECD 88-03017.

larger class of curves and surfaces is closed under many geometric operations of interest, while the class of parametric representations is not; e.g., [4]. For example, given a base curve, its *offset* by a fixed distance is not, in general, a parametric curve, and must therefore be approximated. Moreover, given a point p, it is easy to determine whether p is on an implicit curve or surface, but this determination is not easy if the curve or surface is parametrically represented.

CAGD typically deals with *splines*; that is, with curves or surfaces that consist of individual segments or patches, each belonging to a separate parametric curve or surface. In contrast, this paper considers only individual curves and surfaces.

The study of implicit algebraic curves and surfaces naturally draws on algebraic geometry, a subdiscipline of mathematics that provides some foundational insights into the basic algebraic and geometric properties of algebraic curves and surfaces, *including* parametric curves and surfaces. So, the paper begins with a brief review of selected facts from algebraic geometry, and discusses the problems of converting from parametric to implicit and from implicit to parametric representations. The conversions require substantial symbolic algebraic computations, hindering the wider use of implicits in applications. This situation should change as work in symbolic algebraic computation advances, and recent years have seen impressive progress. But past experience with implicitization algorithms has also motivated other research on implicit curves and surfaces that side-steps this issue altogether. Some of that research is also discussed.

Concepts from Algebraic Geometry

In an effort to eliminate exceptions and special cases from its theorems, algebraic geometry assumes that the curve or surface points under consideration may have complex coordinates, and that there are points "at infinity." Although such generalizations are not necessarily of immediate interest to CAGD, the geometry of a curve or surface at infinity or in the complex part of affine space can influence the details of certain computations, and some cases will be mentioned. Abhyankar [1] presents algebraic geometry material from a foundational view point. Hoffmann [4] presents the material from a geometric modeling perspective, giving both intuition and selected technical details.

An *implicit algebraic curve* is given by an equation of the form

$$f(x,y) = \sum_{i,j} a_{i,j} x^i y^j = 0$$

where f is a polynomial; that is, f has finitely many *terms* of the form $a_{ij}x^iy^j$, where the coefficients a_{ij} are numbers and the exponents i and j are nonnegative integers. The (affine) curve consists of all real or complex points (x,y) that satisfy the equation.

An *implicit algebraic surface* is given by an equation of the form

$$f(x,y) = \sum_{i,j,k} a_{ijk} x^i y^j z^k = 0$$

where f is a polynomial in x, y, and z. The surface consists of all real or complex points (x,y,z) that satisfy the equation.

The *degree* of a term is the sum of the exponents; that is, the degree of $a_{ij}x^iy^j$ is $i+j$, and the degree of $a_{ijk}x^iy^jz^k$ is $i+j+k$. The *degree* of a polynomial is the maximum of the degrees of its terms.

An implicit equation of degree 1 defines a straight line or a plane. Conversely, a straight line in 2-space or a plane in 3-space can always be represented by an implicit equation of degree 1. Degree 2 implicit equations define conics and quadrics. Conversely, every conic or quadric can be defined by a quadratic implicit equation.

A conic or quadric could be *degenerate*. A degenerate conic consists of two lines (parallel, intersecting, or coincident), and a degenerate quadric consists of two planes (parallel, intersecting, or coincident). It is well-known that a conic or quadric is degenerate if, and only if, the polynomial of its implicit equation can be factored into two linear factors, possibly with complex coefficients.

The degree of the implicit equation corresponds to the geometry of the implicit curve as follows: If the equation has degree n, then all but finitely many lines intersect the curve in n points. Similarly, a surface equation of degree n means that all but finitely many lines in space intersect the surface in exactly n points. This requires counting some intersection points as multiple intersections and considering intersections "at infinity." A more general statement is made by Bezout's theorem:

Theorem (Bezout)

Two algebraic curves of degree m and n intersect in either nm points, or else in infinitely many points.

Bezout's theorem for implicit surfaces is not stated here because this would also require discussing algebraic space curves and the question of how they should be represented implicitly.

Implicitization of Parametric Curves and Surfaces

Every integral or rational parametric algebraic curve in the plane can be represented by an implicit algebraic equation, and every integral or rational parametric surface in 3-space can be represented by an implicit algebraic equation. The process of converting a parametric curve or surface to implicit form has been called *implicitization*. Technically, the conversion amounts to eliminating the parametric variable(s) from the equations defining the parametric curve or

surface. For example, consider the parametric curve

$$x = t \qquad y = t^2$$

If t is eliminated from the two equations, the implicit curve equation

$$y - x^2 = 0$$

is obtained. In the case of surfaces, two parameters must be eliminated. For example, let

$$x = \frac{h_1(s,t)}{h_0(s,t)} \qquad y = \frac{h_2(s,t)}{h_0(s,t)} \qquad z = \frac{h_3(s,t)}{h_0(s,t)}$$

be a parametric surface. Then the surface is implicitized by eliminating s and t from the system

$$\begin{aligned} x\,h_0(s,t) - h_1(s,t) &= 0 \\ y\,h_0(s,t) - h_2(s,t) &= 0 \\ z\,h_0(s,t) - h_3(s,t) &= 0 \end{aligned} \tag{1}$$

thereby obtaining a single equation of the form

$$f(x,y,z) = 0$$

Thus, the problem of implicitizing a parametric curve or surface can be reduced to the problem of eliminating one or two variables from a system of nonlinear algebraic equations. The technical issues are how one can do variable elimination algorithmically, and whether the elimination algorithm introduces *extraneous factors*, i.e., whether the resulting polynomial is $f(x,y,z) = f_1(x,y,z)f_2(x,y,z)$ where $f_1(x,y,z) = 0$ is the implicit equation and $f_2(x,y,z)$ is unnecessary.[1]

The process of eliminating variables from systems of linear equations is widely known. There are also well-established techniques for eliminating variables from nonlinear algebraic equations, but they are less well-known and are technically more demanding. The three main algorithmic approaches are resultant-based elimination, the Wu-Ritt method, and the Gröbner bases method. Hoffmann [4] gives an introduction to resultants and Gröbner bases from a geometric modeling perspective. Kapur and Lakshman [7] survey all three approaches from the vantage point of elimination theory.

Resultant-Based Elimination Methods

The oldest, and by now best-known, methodology for eliminating variables is the *resultant method*, whose development began in the last century. The *resultant* of a system of $n+1$ algebraic equations in $n+m$ variables $x_1, \ldots, x_n, y_1, \ldots, y_m$ is

[1] If the surface has base points, the system (1) may entail extraneous solutions that must be excluded with special techniques, as explained later.

an expression in m variables $y_1, ..., y_m$ that vanishes for a specific set of values of the y_i if, and only if, the original system has a solution for the same values of the y_i.

In the simplest case, $n = 1$ and $m = 0$, we are given two polynomials f and g in one variable. The *Sylvester resultant* of the two polynomials is a determinant whose entries are the coefficients of f and g. The resultant is zero iff the two polynomials f and g have a common root.

Implicitizing the rational parametric curve

$$x = \frac{f(t)}{h(t)} \qquad y = \frac{g(t)}{h(t)}$$

is the case $n = 1, m = 2$, for it may be considered as the problem of eliminating t from the system

$$\begin{aligned} xh(t) - f(t) &= 0 \\ yh(t) - g(t) &= 0 \end{aligned} \tag{2}$$

The two polynomials of the system are thought of as polynomials in the variable t with coefficients that are, in turn, polynomials in x and y. Applying the Sylvester resultant then gives the implicit form. Manocha and Canny prove that the algebraic set defined by the polynomial obtained with the resultant is irreducible provided the parameterization has no *base points*, that is, if, for certain parameter values, the denominator and the numerator functions vanish simultaneously. Implicitizing curves without base points that are given by an unfaithful parameterization still may lead to higher multiplicity. For example, as discussed by Recio, the Sylvester resultant for $x = t^2$, $y = t^4$ is $(x^2 - y)^2$. However, higher multiplicity can be determined by GCD computations, a substantially cheaper operation than polynomial factorization.

The Sylvester resultant is of limited use, because it does not apply directly to the problem of implicitizing parametric surfaces. It is possible to use the Sylvester resultant if we consider eliminating the parametric variables s and t successively. As noted in [7], however, this approach is inefficient. Moreover, it may introduce extraneous factors; see [4], Chapter 5.

Many different resultant formulations have been proposed at the beginning of this century, both for curve and for surface implicitization; see, e.g., [9]. For example, the Macaulay resultant eliminates an arbitrary number of variables at once, but requires homogenizing the polynomials and is sensitive to the behavior of the system at infinity. There is also current research on resultant formulations, especially in the presence of base points.

Gröbner Bases Methods

The concept of Gröbner basis is due to Buchberger, and is surveyed in [2]. Let

$$F = \{f_i(x_1, ..., x_n) \mid 1 \leq i \leq m\}$$

be a finite set of polynomials in n variables. The set of common roots of the f_i is called the *algebraic set* defined by F. The set F is *equivalent* to another set G iff the two sets of polynomials define the same algebraic set, accounting for multiplicities. It may be that the set G allows us to better understand the algebraic set it defines than the original set F. For example, if the f_i are linear in the x_j and the set G is equivalent but in triangular form, then the solutions of G are easily found. The idea of Buchberger was to construct, from a given set F of polynomials, an equivalent set G that allows better insight into the structure of the associated algebraic set, and he called such a set a Gröbner basis.

When applied to implicitization, the initial set F consists of two or three polynomials defining the parametric curve or surface, respectively, and the set G will be such that it contains the implicit form of the curve or surface. In the case of rational curves and surfaces, additional polynomials are needed that account for the possibility of base points.

There is a broad literature developing the use of Gröbner bases in symbolic computation and its application. Chapter 7 of [4] gives an introduction into the method and its applications to geometric modeling. Here, only the basic idea of the algorithm is sketched and how it applies to the implicitization problem. It is not possible to describe the technical details of the method in this paper, so we confine the description to an intuitive outline of the computational stages and their motivation.

In order to derive an equivalent set G that makes explicit the properties of the algebraic set, the initial set F of polynomials is rewritten. Two activities are carried out:

1. New polynomials, called *S-polynomials*, are formed from pairs of old polynomials by a certain rule, reminiscent of the way in which the least common multiple of two numbers is formed.
2. The polynomials are reduced by rewriting, a process in which suitable multiples of other polynomials in F are subtracted to cancel complicated terms.

In the course of this computation, the number of polynomials in the set will fluctuate: Some of the new polynomials will reduce to zero and thus do not contribute to the set of polynomials. Some old polynomials may also reduce to zero when certain new polynomials have been added to the set. When all S-polynomials reduce to zero, the set of polynomials is a Gröbner basis.

Rewriting, and the final nature of the Gröbner basis depend on an ordering of polynomials, which in turn depends on an ordering of the terms of the polynomials. Different term orderings can be given. In particular, the *lexicographic* ordering results in a Gröbner basis that has a triangular structure. In the implicitization problem the basis must contain the implicit form without extraneous factors.

For example, assume that we are to implicitize the surface

$$x = st \qquad y = st^2 \qquad z = s^2$$

We construct the Gröbner basis of the input set

$$F = \{x - st, y - st^2, z - s^2\} \tag{3}$$

with respect to the lexicographic ordering induced by the variable ordering $z \prec y \prec x \prec t \prec s$. We could have chosen a different variable ordering as long as the parametric variables s and t are of higher order than x, y, and z, and thus are eliminated first. We obtain the following Gröbner basis:

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

The first polynomial, $x^4 - y^2 z$, is the implicit form of the surface. The second polynomial introduces the variable t and provides a way to find the parametric t coordinate of a surface point: Solve $tx - y = 0$ for the surface point (x, y, z) to obtain t. In the same way, s can be found from, say, $sy - x^2$. Thus we can also solve the *inversion problem* and find for a given point on the parametric surface its (s, t) coordinates.

If the curve or surface is rational, common roots of the numerator and denominator polynomials cause problems. To avoid this, Kalkbrener has suggested adding certain equations to the systems (1) and (2). For example, in the rational curve case (2) we add the equation

$$u\,h(t) - 1 = 0$$

where u is a new variable. The equation states in intuitive terms that $h(t)$ does not vanish. Thus, the roots of $h(t)$ are excluded and with them all base points of the curve.

An important variant of the computation for efficiency considerations can be based on *basis conversion*, a computation due to Faugère, Gianni, Lazard, and Mora. In this variant, only one element of the basis G is determined, namely the implicit form of the parametric curve or surface. See [4] for details.

Wu-Ritt Method

A different method for variable elimination was developed by Wu Wen-Tsün [10] using an idea proposed by Ritt [7]. Wu was interested in automatically proving theorems from geometry. The theorems are translated into an algebraic problem in which the question is investigated whether a particular polynomial

f coding the conclusion of the theorem follows from the hypotheses, encoded as a set F of polynomials. In geometric terms, f follows from F if the algebraic set of F is contained in the algebraic set of f. The method transforms the given system F of polynomials until it has a certain form. The transformation involves rewriting the polynomials in F repeatedly, using pseudo-division, and adding the remainders to the set F.

As in the Gröbner basis case, the description that follows remains intuitive and omits many technical details. See [7] for more details.

The objective of the Wu-Ritt method is to transform F into a triangular system of polynomials. Again, the variables are ordered, but now multivariate polynomials are thought of as polynomials in the highest occurring variable whose coefficients are polynomials in the lower-order variables. In turn, the coefficient polynomials are also so viewed. In the basic loop, a subset of F is identified by selecting polynomials of lowest degree whose highest occurring variable is not yet in the subset. The subset so selected is a *base set*, and all polynomials in F are pseudo-divided by polynomials in the base set. The remainders so obtained are added to F and the process is repeated. Ultimately, no new polynomials are added, and the final base set is triangular.

When applied to sets such as (1) or (2) with a variable ordering $t \prec s \prec \cdots$, the implicit form will be constructed, possibly with some extraneous factors [7]. We illustrate the procedure with the parametric surface (3). The input set F is

$$x - st \tag{5}$$

$$y - st^2 \tag{6}$$

$$z - s^2 \tag{7}$$

Pseudo-division of (6) by (5), with respect to s, produces the remainder

$$y - xt \tag{8}$$

and pseudo-division of (7) by (5) produces the remainder

$$zt^2 - x^2 \tag{9}$$

In both polynomials the variable s has been eliminated, because (5) is linear in s. Now pseudo-division of (9) by (8) produces the remainder

$$y^2 z - x^4$$

which is the implicit form.

Implementation

Careful implementation of a suitable resultant formulation can be the basis of implicitization algorithms that are fairly efficient. Manocha and Canny have

achieved good running times using numerical techniques to augment the symbolic computation. Using a modification of basis conversion, Hoffmann achieved attractive speed-ups for Gröbner based implicitization. Gao and Chou report good results using the Wu-Ritt method for implicitization. Each approach has its proponents who advance the computational machinery by making it more and more efficient. However, even though recent years have seen performance improvements of several orders of magnitude, the earlier, slower algorithms have left many practitioners with the impression that implicitization is necessarily impractical. It is not clear whether this impression remains justified.

Parameterization of Implicits

We stated before that every parametric curve and surface has an implicit representation. The converse is not true, and there are implicit curves and surfaces that are provably not representable in rational parametric form. The problem of finding a parametric representation for a given implicit algebraic curve or surface has therefore two distinct parts:

(a) Determine if the curve or surface can be parameterized, and if so,

(b) find a parameterization.

For nondegenerate quadrics the first part is unnecessary, because every nondegenerate quadric can be parameterized.

There is a geometric interpretation of the parameterization process of curves that is best understood when parameterizing conics:

1. Pick a curve point and consider a pencil of lines through it. Note that the lines in the pencil can be indexed by a parameter, say the slope of the line, and that the lines intersect the conic in two points, the point we picked and one other point.
2. Determine for each line the second intersection with the conic, as a function of the parameter indexing the line in the pencil. The symbolic coordinates of this intersection point are the coordinate functions of the conic.

As example, consider the unit circle

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

Pick as fixed point $(-1, 0)$, say. The pencil of lines through this point is given by

$$y - tx - t = 0$$

Each value of t defines a particular line of the pencil. It is easy to verify that every such line contains $(-1, 0)$ and that t is the slope of the line. Substituting

for y and solving for x gives

$$x = \frac{1 - t^2}{1 + t^2} \tag{10}$$

and so y is given by

$$y = \frac{2t}{1 + t^2} \tag{11}$$

The two equations (10) and (11) constitute a parameterization of the circle. This geometric idea generalizes to other curves in two ways:

1. For certain higher-degree curves a pencil of lines suffices, but the point must be singular of the right multiplicity so that, apart from the fixed point, the lines in the pencil intersect the curve in only one other point.
2. Instead of lines, a family of curves may have to be used, of fixed degree, chosen to contain several fixed points of the curve we wish to parameterize. Again, the fixed points must have the right multiplicity.

Note that by Bezout's theorem a line intersects a curve of degree n in n points. Therefore, if a pencil of lines is to suffice, a curve point of multiplicity $n - 1$ is required. Curves that possess such a point are called *monoids*. Conics are trivially monoids. There are also monoidal surfaces, for instance quadrics, and they are parameterized by an analogous construction.

The following exposition is relatively brief and omits some technicalities. For greater detail see, e.g., [4].

Degree 2 Curves and Surfaces

Degree 2 curves can be parameterized using the pencil-of-lines approach. A curve point is chosen, which may be a point at infinity. For example, if the parabola $y - x^2 = 0$ is to be parameterized, choosing its point at infinity, or choosing the point $(0,0)$ gives the familiar parameterization $x = t$, $y = t^2$. Other points on the parabola could also be chosen, leading to different parameterizations.

A second approach translates the pencil-of-lines approach into an algebraic procedure. Roughly speaking, the implicit curve equation is transformed so that the y^2-term vanishes, using a projective coordinate transformation. Geometrically, this is equivalent to changing the coordinate system so that the curve contains the point at infinity that lies in the direction of the y-axis. The transformed curve can then be parameterized using $x = t$, and from this parameterization a parameterization of the original curve can be obtained by applying the inverse coordinate transformation.

The third method for parameterization applies a numerical iteration to the implicit equation written as a bilinear matrix form. The effect of the iteration

is to diagonalize the matrix. Once diagonalized, a standard parameterization is used that gives the parameterization of the original curve after subjecting it to the inverse of the transformation defined by the iteration.

All three approaches generalize to quadric surfaces. In the first approach, a fixed point is chosen and a bundle of lines through that point is considered. Each line in the bundle is now determined by two parameters instead of one. The computational details are routine. In the second approach, the quadric is transformed such that the conic in which the quadric intersects the plane at infinity goes through a special point. In that case, one of the quadratic terms, say z^2, vanishes from the quadric equation. The transformed quadric is therefore parameterizable with $x = s$, $y = t$. In the third approach, the same numerical iteration is applied, for it does not depend on the size of the matrix. Again, a standard parameterization of the surface, defined by the diagonal matrix, is back transformed to a parameterization of the original surface.

Cubic Curves

Only singular cubics have a parametric form, that is, cubics that have a singular point. The singularity may not be readily apparent: For example, $y - x^3 = 0$ is evidently parameterized by $x = t$, $y = t^3$, but has no finite singular point. Here, the singularity is at infinity.

Choose a pencil of lines through this singular point. Since the fixed point is a double point, a line in the pencil intersects the cubic in one other point, and determining this other point in terms of the line index t gives a parameterization of the cubic. This approach assumes that we know where the singularity is.

There is an algebraic computation that tests whether the cubic is parameterizable and if so, determines a parameterization. Roughly speaking, the cubic curve equation is transformed into an equation of the form

$$\bar{y}^2 = f(x)$$

where f could have degree 4. It can be shown that the cubic is parameterizable iff $f(x)$ has a double root. In that case, a second transformation yields an equation

$$\bar{y}^2 = g(x)$$

where $g(x)$ is at most quadratic. This curve can be parameterized as a conic and the parameterization is back transformed to a parameterization of the original cubic.

Monoids

A *monoid* is a curve or surface of degree n that has an $(n-1)$-fold singular point. All conics and quadrics are monoids, because in this case $n = 2$ and regular

points have multiplicity 1. For cubics a double point is required. Higher-order monoids include the Steiner surfaces. Monoids also are called *dual forms* by some authors because they are so easy to parameterize provided we know an $(n-1)$-fold point.

Briefly, monoids are parameterized using pencils of lines. If the $(n-1)$-fold point is brought to the origin of the coordinate system, the monoid equation becomes especially simple. In the case of curves it is then

$$h_n(x,y) - h_{n-1}(x,y) = 0$$

and in the case of surfaces it is

$$h_n(x,y,z) - h_{n-1}(x,y,z) = 0$$

where, h_n is a polynomial all of whose terms have degree n, and h_{n-1} is a polynomial all of whose terms have degree $n-1$. The curve is parameterized by

$$x(s,t) = \frac{sh_{n-1}(s,t)}{h_n(s,t)} \qquad y(s,t) = \frac{th_{n-1}(s,t)}{h_n(s,t)}$$

Either s or t is set to 1. For monoidal surfaces, the parameterization is

$$\begin{aligned} x(r,s,t) &= \frac{rh_{n-1}(r,s,t)}{h_n(r,s,t)} \\ y(r,s,t) &= \frac{sh_{n-1}(r,s,t)}{h_n(r,s,t)} \\ z(r,s,t) &= \frac{th_{n-1}(r,s,t)}{h_n(r,s,t)} \end{aligned}$$

Again, one of the parameters r, s, or t is set to 1.

For example, consider the unit sphere containing the origin

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

Clearly $h_n = x^2 + y^2 + z^2$ and $h_{n-1} = -2z$. With $r = 1$, the surface is parameterized by

$$\begin{aligned} x(s,t) &= \frac{2t}{1+s^2+t^2} \\ y(s,t) &= \frac{2st}{1+s^2+t^2} \\ z(s,t) &= \frac{2t^2}{1+s^2+t^2} \end{aligned}$$

Note that we could have set $s = 1$ or $t = 1$ instead.

Deferring Implicitization

Univariate and multivariate resultants are essentially determinants, or determinant quotients, whose entries are polynomials and whose value, in the case of implicitization, is the implicit form of a parametric curve or surface. The major cost of implicitization, using resultants, is the evaluation of this determinant because it requires manipulating polynomials with many terms and, possibly, large rational coefficients, assuming exact arithmetic is used.

One of the strengths of the implicit form is the ability to evaluate it for a given point, and to deduce from the value whether the point is on the surface (zero value), or outside (negative value) or inside (positive value). Manocha and Canny [8] observed that this evaluation can be done equivalently by evaluating, in the resultant, each entry, followed by an evaluation of the now numerical determinant. Since the polynomial entries in the determinant are linear, evaluating them numerically is very simple.

The approach requires that the implicit form, as evaluated by the determinant, does not have extraneous factors. In another paper, Manocha and Canny prove that for curves such determinants always exist, although the presence of base points requires special techniques. For surfaces the determinants also exist provided all base points are simple. The case of surfaces with base points of higher multiplicity is open.

In applications such as the intersection of two parametric surfaces, some authors have advocated implicitizing one of the surfaces, and substituting into it the parametric equations of the other surface. This reduces the evaluation of surface intersection to the evaluation of plane curves, as explained in [4]. One may think of this approach to surface intersection evaluation as having a symbolic preprocessing step, here the implicitization and substitution, followed by a numerical computation, the evaluation of the plane curve. Manocha and Canny [8] substitute the parametric equations into every determinant entry, and then evaluate the determinant numerically, for each curve point. Their approach, therefore, reduces the role of symbolic computation in preprocessing. This reduction entails additional numerical computation.

Evaluating a plane curve numerically may require a number operations, such as derivative evaluation. If the plane curve is represented by a determinant whose entries are polynomials, then special algorithms are required. In [8] such algorithms are described.

Constrained Surface Representations

Certain curves and surfaces are naturally described in terms of one or more *base* curve(s) or surface(s) and some geometric constraints. For instance, given two base surfaces f and g, consider all points in space that have equal minimum

distance from the given surfaces. Such points form the *equal-distance* surface of f and g. Other examples include offset curves and surfaces, and blending surfaces obtained as envelope of a rolling ball.

Despite the conceptual simplicity of defining such curves and surfaces intuitively, an exact mathematical representation of them is difficult both in the case of the parametric and the implicit representations. Often, neither representation can be reasonably determined, and so we seek an alternative to these two representation schemata. The *dimensionality paradigm* provides such an alternative [6].

The Dimensionality Paradigm

The definition of a constrained surface often simplifies when we consider it as the natural projection of a manifold in higher-dimensional space. The manifold can be defined simply by a system of nonlinear equations in n variables, where $n > 3$ in general. The extra variables identify certain points on the base surface(s), or specify distances or other geometric data. The surface we want is then the natural projection of this manifold into a three-dimensional subspace.

If the base surfaces are algebraic, then the additional variables could be eliminated from the system of equations, at least in principle, resulting in the implicit surface equation. Such an approach is normally intractable, for the elimination problems are usually well beyond what hardware and software can deliver in the forseeable future. Therefore, one should work *directly* with the system of equations. If the degree of the implicit form is high, and this occurs often, it is also reasonable to expect that subsequent numerical computations are more stable when performed on the system of equations, rather than on the implicit equation.

Example Definition

Assume that two base surfaces f and g are given whose equal-distance surface is sought. The base surfaces could be parametric or implicit. Using a declarative style, we can then describe the equal-distance surface as follows:

1. Let $p = (x, y, z)$ be a point on the equal-distance surface. Moreover, let $p_f = (u_1, v_1, w_1)$ be a point at minimum distance from p on f, and let $p_g = (u_2, v_2, w_2)$ be a point at minimum distance from p on the surface g. Then:

2. The point p_f satisfies the equation of f, and the point p_g satisfies the equation of g.

3. The distance (p, p_f) is equal to the distance (p, p_g).

4. The line $\overline{p, p_f}$ is normal to f at p_f.

5. The line $\overline{p, p_g}$ is normal to g at p_g.

Assertion (1) declares the names of nine variables that comprise the coordinates of the three points p, p_f and p_g. Assertions (2)–(5) express the geometric relationships that these points must satisfy. As shown in [5], the assertions translate very simply into a system of nonlinear equations. In principle, an implicit equation could be derived by eliminating the six variables $\{u_1, v_1, ..., w_2\}$ from the system, but in almost all cases this computation is not tractable.

The entire system of equations defines a manifold in 9-dimensional space. The projection of that manifold into the (x, y, z)-subspace is the equal-distance surface. A number of papers by Chandru, Chiang, Chuang, Dutta, Hoffmann, Lynch, Vermeer, and Zhou discuss other examples of surface definitions using the dimensionality paradigm, including offset surfaces, constant-radius blends, variable-radius blends, ruled surfaces in parametric blending, and trimming surfaces in medial-axis computations.

Surface Interrogation

There is a considerable body of algorithmic infrastructure for surfaces defined using the dimensionality paradigm, including

1. Given two surfaces and an initial point, evaluate their intersection; see [4]. The algorithm is robust and can evaluate very high-degree surface intersections without significant precision problems.

2. Given a surface and an initial point, evaluate locally the curvatures, and give a local parametric or local explicit surface approximant of arbitrary contact order, [3].

3. Given a surface and an initial point, globally approximate the surface; [3]. The algorithm has an adaptive version in which local curvature information determines the number of approximants.

These algorithms can also be used when the system of nonlinear equations is nonalgebraic.

Global Approximation

It is easy to derive a marching scheme for curves such as surface intersections. A similar scheme for evaluating surfaces requires a way to orient the exploration in space such that the same neighborhood is not reevaluated. In [3] this problem has been addressed in the context of the dimensionality paradigm. The technique competes well with other approaches such as Allgower's simplicial continuation method, or the moving-frame method of Rheinboldt. The global approximation is based on the following idea. Given a manifold S by a system

of equations and on it a point p, construct a piecewise approximation of $\mathcal{S}$, beginning at p and extending in all directions.

Rheinboldt's moving-frame method triangulates the tangent space at p and transfers the triangulation to $\mathcal{S}$ using Newton iteration. Each vertex of the triangulation, after projection to a point q on $\mathcal{S}$, becomes the center of a new triangulation, of the tangent space at q. The algorithm resolves local overlap, but cannot resolve global overlap. Thus, when constructing an approximation of a sphere by this method, one would not know when the entire surface has been approximated.

Allgower's method is based on a triangulation of ambient space and only evaluates the system of equations. At each vertex of a simplex, the system of equations is evaluated and from the vector of function values a linear approximant is constructed. Then the simplex faces that intersect the linear approximant are determined, and adjacent simplices are evaluated in the same way. Note that the ambient n-dimensional space is triangulated. Thus, the number of simplices in an elementary volume grows exponentially with the number of variables. However, the method does not require evaluating derivatives of the equations of the system.

In the case of surface definitions with the dimensionality paradigm, the surface that is ultimately of interest is the projection into a three-dimensional subspace. It is therefore advantageous to approximate only the projection, and Chuang's algorithm does this, using a grid in 3-space to detect whether a volume of space has already been explored.

1. At p, construct a local approximant to $\mathcal{S}$. The approximant has the form $x_i = h_i(s,t)$, $1 \leq i \leq n$.

2. Determine how the projected approximant, $(h_1(s,t), h_2(s,t), h_3(s,t))$, intersects the faces of the cube, as a function of s and t.

3. From the intersection curves, determine the coordinates (s_1, t_1) of a point on the approximant that lies in an adjacent cube.

4. Refine the estimated point with Newton iteration.

The advantage of this method is the fact that the dimension of the meshed space does not depend on the number of variables used to define $\mathcal{S}$. Yet, by determining the (s,t) curves, each estimate (s_1, t_1) can be pulled back into the n-space in which $\mathcal{S}$ is given.

There is a tradeoff between the degree of the approximant, the mesh size of the grid, and the difficulty of determining face intersections and adjacent points in Steps 2 and 3. With increasing degree of the approximant, a coarser mesh can be tolerated, so that fewer approximant calculations are needed. However, determining the intersection with the faces of the current cube becomes more difficult.

Implementation

There are pilot implementations of all surface interrogations, and experience indicates that they are both efficient and robust. Varady and his group have implemented rolling-ball blends, defined in the dimensionality paradigm, as part of their solid modeler FFSOLID. Using higher-dimensional surface intersection, the contact curves and cross sections of the blend are determined. Then the blending surface is approximated using double-quadratic patches.

Summary

The role which implicit curves and surfaces could play in CAGD depends on several factors. If it were easy to convert between the implicit and parametric forms, then implicit representations could usefully supplement parametric representations, and one would convert between representations as needed by the geometric operation under consideration. However, algorithmic conversion is not so simple, and ongoing research has to lower the cost further before conversions can become routine operations. Past years have seen significant speed-ups of implicitization algorithms, and the continuing interest of researchers should assure continuing progress.

Two trends for bypassing implicitization offer alternative routes and have been discussed. Manocha and Canny's unevaluated determinants trade preprocessing costs against the cost of subsequent, numerical computations that implement a geometric operation. Hoffmann's dimensionality paradigm provides exact representations of complex constrained curves and surfaces, and the available infrastructure allows interrogating surfaces so defined in an efficient and robust manner without the concern that a derivation of the implicit form might pose intractable problems.

There have been attempts to develop algorithms and design paradigms that are to compete directly with the established paradigms and algorithms of CAGD. Examples include shape modification of implicit surfaces by weighted control point arrangements in space, space deformations, heuristics for deriving piecewise implicit curves and surfaces, and solid modelers working with arbitrary algebraic half spaces. This work is quite recent, and as it seeks to compete with the long and distinguished tradition of CAGD developing parametric representations, it would appear that the long-term importance of these efforts depends much on future research.

Acknowledgements

This report draws on the work of many researchers and they should all be cited. However, editorial policy of this journal limits the number of references.

References

[1] S. Abhyankar. *Algebraic Geometry for Scientists and Engineers.* American Mathematical Society, Mathematical Surveys and Monographs, 35, 1990.

[2] B. Buchberger. Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory. In N. K. Bose, editor, *Multidimensional Systems Theory*, pages 184–232. D. Reidel Publishing Co., 1985.

[3] J.-H. Chuang. *Surface Approximations in Geometric Modeling.* PhD thesis, Purdue University, Computer Science, 1990.

[4] C. M. Hoffmann. *Geometric and Solid Modeling.* Morgan Kaufmann, San Mateo, Cal., 1989.

[5] C. M. Hoffmann. Algebraic and numerical techniques for offsets and blends. In S. Micchelli M. Gasca, W. Dahmen, editor, *Computations of Curves and Surfaces*, pages 499–528. Kluwer Academic, 1990.

[6] C. M. Hoffmann. A dimensionality paradigm for surface interrogation. *CAGD*, 7:517–532, 1990.

[7] D. Kapur and Y. N. Lakshman. Elimination methods: An introduction. In B. Donald, D. Kapur, and J. Mundy, editors, *Symbolic and Numerical Computation: An Integration.* Academic Press, 1992. to appear.

[8] D. Manocha and J. Canny. A new approach for surface intersection. *Intl. Journal Computational Geometry and Applications*, 1, 1991. to appear.

[9] G. Salmon. *Lessons Introductory to the Modern Higher Algebra.* G.E. Stechert & Co., New York, 1885.

[10] Wu Wen-Tsün. Basic principles of mechanical theorem proving in geometries. *J. of Systems Sciences and Mathematical Sciences*, 4:207–235, 1986.