1992

On Projections in Geometric Design

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

Report Number:

92-003

ON PROJECTIONS IN GEOMETRIC DESIGN

Christoph M. Hoffmann

CSD-TR 92-003
January 1992

On Projections in Geometric Design

Christoph M. Hoffmann*
Department of Computer Science
Purdue University
West Lafayette, Indiana 47906
USA

January 6, 1992

Abstract

We discuss using projected manifolds in geometric design and processing. Curves and surfaces are defined in n-space as algebraic sets of the appropriate dimensionality, and their projection into a three-dimensional subspace is considered. In this approach, the additional variables serve to express simply geometric constraints, and so complex surfaces such as offsets, equi-distance surfaces, rounds and fillets can be represented exactly. We describe the definitional method and discuss some of the algorithmic infrastructure available to manipulate and interrogate curves and surfaces so represented. This representation generalizes both implicit and parametric representations, and can deal with surfaces that otherwise could be obtained exactly only through elimination computations of forbidding complexity.

1 Introduction

There are two well-established paradigms for representing curves and surfaces that are based on rational parametric or implicit algebraic forms. A large literature has accumulated many efficient and ingenious algorithms for working with these representations. Yet despite this extensive work, there are tasks of interest that appear to be difficult to implement based on these traditional representations. For instance, given a surface, how can we derive a new surface that is the offset of the old one, by a fixed distance? The mathematical difficulty of giving the offset of a general, curved surface is in strong contrast to the simplicity with

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

which offsets can be defined and communicated between humans. Such difficulties suggest that we seek new curve and surface representations that might facilitate such operations and, at the same time, admit convenient algorithms for working with them.

The majority of work in *Computer-Aided Geometric Design* (CAGD) concentrates on *parametric surfaces*, i.e., surfaces that are composed from *patches* defined by coordinate functions

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

where the parameters u and v range over a finite domain, such as the unit square $[0,1] \times [0,1]$, and the coordinate functions $h_i(u,v)$ are polynomials or ratios of polynomials in u and v. When the coordinate functions are represented in a special form, for instance in the Bernstein-Bézier basis, a strong correlation between a net of *control points* in 3-space and the coefficients of the coordinate functions exists that provides an intuitive control of the shape of the patch and lends itself well to interactive free-form surface design. Furthermore, the Bernstein-Bézier basis affords many ingenious and efficient techniques for evaluating surface points, or ensuring continuity degrees between adjacent patches. For an entry into the literature on this subject we refer to books such as [4, 12, 18] or to survey articles such as [5]. Parametric surfaces are not closed under geometric operations such as offsetting.

Implicit algebraic surfaces provide a more general alternative to the parametric surface representation. An implicit algebraic surface is defined by an equation

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

where f is a polynomial in x, y and z. Algebraic surfaces are closed under most geometric operations of interest, including offsetting, but they lack a strong correlation between intuitive shape and the form of the polynomial $f(x,y,z)$. Nevertheless, implicit surfaces play a firm role in geometric modeling, and [13] gives many details of their properties and techniques for interrogating them, from a geometric modeling perspective.

Implicit algebraic surfaces include in principle all parametric surfaces, but not vice versa. Current techniques for converting from parametric to implicit form, and methods for parameterizing implicit algebraic surfaces that possess a rational parametric form, can be demanding symbolic computations [14]. These computations are becoming more and more efficient thanks to recent advances in symbolic computation. However, it is unlikely that the geometric operations we consider here can be approached through elimination.

One would like to construct curved surfaces that satisfy prescribed constraints. Such surfaces may be readily explained in intuitive geometric terms:

For instance, given two surfaces f and g, consider all points in space that have equal minimum distance from the given surfaces. Such points form the *equal-distance* or *Voronoi surface* of f and g. These and other examples are given in [8, 9, 10, 11, 14, 15]. Such surface definitions have the basic property that a new surface is expressed in terms of one or more *base surfaces* and a number of geometric constraints. Despite the conceptual simplicity, it is by no means trivial to represent such surfaces implicitly or approximate them parametrically. In fact, an implicit form cannot be derived in many cases because the elimination computation exceeds all reasonable bounds in space and time requirements.

2 The Dimensionality Paradigm

We define a constrained surface as the natural projection of a 2-manifold in higher-dimensional space. The manifold is specified by a system of nonlinear equations in n variables, $n > 3$. Usually, the equations are algebraic, although this would not be strictly necessary. The extra dimensions may represent point coordinates on the base surface(s), distances, or other quantities used to express the constraints that must be obeyed. The surface we want is then the natural projection of this manifold into a three-dimensional subspace.

In the case of algebraic base surfaces, the resulting system of equations could be processed by a number of symbolic computation algorithms that eliminate all additional variables to derive a single implicit equation for the surface. This is normally intractable. So, we will work directly with the system of equations. We demonstrate the definitional approach with an example.

2.1 An Example Definition

We consider the definition of an equal-distance surface as an example of this approach, following the presentation of [16]. Assume that we are given two implicit surfaces $f(x, y, z) = 0$ and $g(x, y, z) = 0$. 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.

Note that Assertion (1) declares the names of nine variables, the coordinates of three points, whereas Assertions (2–5) simply state the geometric relationships that these points must satisfy.

In order to obtain an equational representation of the equal-distance surface, we translate the Assertions (2–5), using the variable names of (1). We obtain in sequence:

$$
\begin{aligned}
f(u_1, v_1, w_1) &= 0 \quad (1)\\
g(u_2, v_2, w_2) &= 0 \quad (2)\\
(x-u_1)^2 + (y-v_1)^2 + (z-w_1)^2 - (x-u_2)^2 + (y-v_2)^2 + (z-w_2)^2 &= 0 \quad (3)\\
[x-u_1, y-v_1, z-w_1] \cdot [-f_{v_1}, f_{u_1}, 0] &= 0 \quad (4)\\
[x-u_1, y-v_1, z-w_1] \cdot [f_{w_1}, 0, -f_{u_1}] &= 0 \quad (5)\\
[x-u_1, y-v_1, z-w_1] \cdot [0, -f_{w_1}, f_{v_1}] &= 0 \quad (6)\\
[x-u_2, y-v_2, z-w_2] \cdot [-g_{v_2}, g_{u_2}, 0] &= 0 \quad (7)\\
[x-u_2, y-v_2, z-w_2] \cdot [g_{w_2}, 0, -g_{u_2}] &= 0 \quad (8)\\
[x-u_2, y-v_2, z-w_2] \cdot [0, -g_{w_2}, g_{v_2}] &= 0 \quad (9)
\end{aligned}
$$

Subscripting, as in f_{u_1}, denotes partial differentiation.

Equations (1–3) express Assertions (2) and (3). Equations (4–6) together express Assertion (4), since the three vectors

$$
\begin{gathered}
[-f_{v_1}, f_{u_1}, 0]\\
[f_{w_1}, 0, -f_{u_1}]\\
[0, -f_{w_1}, f_{v_1}]
\end{gathered}
$$

are tangent to f and span the tangent space as long as p_f is not a singular point on the surface. Similarly, Eqs. (7–9) together express Assertion (5).

Note that if f is given in parametric form, then Eq. (1) and Eqs. (4–6) have to be adjusted accordingly. This is routine and shows that the methodology is independent of whether the base surfaces are in implicit or in parametric form.

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 describe other examples of surface definitions with the dimensionality paradigm:

- offset surfaces, [15, 14];
- constant-radius blends, [14];
- variable-radius blends, [14, 8];
- ruled surfaces in parametric blending, [10];

- trimming surfaces in skeleton computations, [11].

We do not repeat these definitions here.

2.2 Faithful Definition Systems

The above equation system also defines certain points that are unwanted because they do not reflect the geometric intent. The points come from two sources:

1. Distance constraints are expressed by local extremal conditions. Thus, global minimum distance is not expressed.

2. At certain points some of the equations may become dependent. For example, if p_f is a singular point, then Eqs. (4–6) vanish. In consequence, the system also defines a submanifold that projects to the equal-distance surface of g and the singular point.

It is not possible to express global minimum distance without introducing inequalities. This type of extraneous solution is excluded algorithmically, by the surface interrogation algorithms. The possible local interdependence of individual equations can be excluded, however.

The unwanted solutions are eliminated by adding more equations, [17], that encode unequalities. These new equations use one or more additional variables. The idea is familiar from the refutational approaches in automated geometry theorem proving, [19, 20]. Details are reported in [17].

It is not always simple to define precisely what is meant by "extraneous" solution. In [17], the extraneous solutions in the case of the equi-distance surface, Eqs. (1–9), is defined as follows. Let $p = (x, y, z)$ be a point of the equal-distance surface, $p_f = (u_1, v_1, w_1)$ a point on f at minimum distance from p, and $p_g = (u_2, v_2, w_2)$ a point on g also at minimum distance from p. The points p_f and p_g are *footpoints* of p on f and g, respectively. Footpoints and the associated surface point(s) are said to *correspond*. Then a solution is *extraneous* if it corresponds to a footpoint that, in turn, corresponds to infinitely many solutions. Using this definition, it can be shown that all real extraneous solutions to Eqs. (1–9) must arise as follows, [17]:

1. Footpoints p_f or p_g are singular. In this case, we obtain as extraneous solutions points at equal distance from the singular point and the other surface. In case both footpoints are singular but not coincident, there is an additional plane. If both footpoints are singular and coincident, then every point in $\mathbf{R}^3$ is an extraneous solution.

2. The footpoints coincide, are regular, and the base surfaces intersect tangentially. In this case, the common surface normal is extraneous.

In the proof, [17] assumes that f and g are algebraic surfaces, because Bezout's Theorem is used to show that all other footpoints correspond to finitely many points of the equal-distance surface.

We modify the system as follows: All singular footpoints on the base surfaces are excluded by adjoining

$$\begin{aligned}(\alpha f_{u_1} - 1)(\alpha f_{v_1} - 1)(\alpha f_{w_1} - 1) &= 0\\ (\beta g_{u_2} - 1)(\beta g_{v_2} - 1)(\beta g_{w_2} - 1) &= 0\end{aligned}$$

where α and β are new variables. The two equations express that not all partial derivatives of f and of g vanish simultaneously at footpoints. Moreover, adjoining

$$(\gamma U - 1)(\gamma V - 1)(\gamma W - 1)(\gamma N - 1) = 0$$

where

$$\begin{aligned}U &= u_1 - u_2 & V &= v_1 - v_2\\ W &= w_1 - w_2 & N &= \|\nabla f \times \nabla g\|\end{aligned}$$

expresses that the footpoints p_f and p_g are distinct, or else that the surface normals through them do not coincide. The resulting system has no extraneous real solutions. In the same way, [17] excludes extraneous solutions from offsets, constant-radius blends and variable-radius blends.

3 Interrogating Higher-Dimensional Surfaces

A sizable body of algorithmic infrastructure has been developed that deals with surfaces defined with the dimensionality paradigm. The following algorithms are now available:

1. Given two surfaces and an initial point on both, evaluate their intersection; see [3, 13, 15]. 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, [10], and give a local parametric or local explicit surface approximant of arbitrary contact order, [10, 14].

3. Given a surface and an initial point, globally approximate the surface; [10].

These algorithms do not require the system to consist of algebraic equations, but assume that the equations are continuously differentiable.

There are several techniques for finding initial points. When nothing is known about the geometry of the surface, then generic techniques such as [2, 1, 6, 7, 21] can be applied. However, the geometric intent of the system is usually known and should provide good initial estimates for starting points that can be refined using Newton iteration. At this time, there is no systematic work that applies geometric reasoning to finding initial points algorithmically.

3.1 Local Parametric Approximation

Let $\mathcal{S}$ be a manifold defined by

$$\begin{aligned} f_1(x_1, x_2, ..., x_n) &= 0 \\ f_2(x_1, x_2, ..., x_n) &= 0 \\ &\vdots \\ f_m(x_1, x_2, ..., x_n) &= 0 \end{aligned} \tag{10}$$

and let $p = (u_1, u_2, ..., u_n)$ be a solution of the system. Assume that every hypersurface f_i is regular and twice continuously differentiable at p, and that the Jacobian matrix

$$\left(\frac{\partial f_i}{\partial x_j}\right)_{i,j}$$

has rank $n-2$. Then $\mathcal{S}$ defined by Eq. (10) has dimension 2 in the neighborhood of p. Its natural projection into the (x_1, x_2, x_3)-subspace will be denoted by $\pi(\mathcal{S})$. We compute a local approximant to the manifold at p, using the approach of [10, 14]. More precisely, we compute coordinate functions

$$\begin{aligned} x_1 &= h_1(s,t) \\ x_2 &= h_2(s,t) \\ &\vdots \\ x_n &= h_n(s,t) \end{aligned} \tag{11}$$

such that

$$p = (h_1(0,0), ..., h_n(0,0))$$

and

$$f_i(h_1(s,t),\ h_2(s,t),\ ...,\ h_n(s,t)) \equiv 0$$

for $i = 1, ..., m$. The projection $\pi(\mathcal{S})$ of the manifold $\mathcal{S}$ into the (x_1, x_2, x_3)-subspace is locally approximated by

$$\begin{aligned} x_1 &= h_1(s,t) \\ x_2 &= h_2(s,t) \\ x_3 &= h_3(s,t) \end{aligned}$$

in the vicinity of the projection $\pi(p)$ of p.

The Taylor expansion of f_i at $p = (u_1, ..., u_n)$ is

$$\begin{aligned} f_i(x_1, ..., x_n) &= f_i(u_1 + \delta_1, u_2 + \delta_2, ..., u_n + \delta_n) \\ &= f_i(u_1, ..., u_n) \\ &\quad + f_i^{(1)}\delta_1 + \cdots + f_i^{(n)}\delta_n \\ &\quad + [f_i^{(1,1)}\delta_1^2 + \cdots + f_i^{(n,n)}\delta_n^2]/2 \\ &\quad + f_i^{(1,2)}\delta_1\delta_2 + \cdots + f_i^{(1,n)}\delta_1\delta_n \\ &\quad + f_i^{(2,3)}\delta_2\delta_3 + \cdots + f_i^{(n-1,n)}\delta_{n-1}\delta_n \\ &\quad + \text{higher order terms} \end{aligned} \tag{12}$$

where $f_i^{(j,k)}$ denotes the partial derivative of f_i by x_j and x_k.

We choose s and t such that $p = (h_1(0,0), h_2(0,0), ..., h_n(0,0))$. Then the Taylor expansion of the h_i is

$$\begin{aligned} h_k(s,t) &= h_k(0,0) \\ &\quad + h_k^{(s)}s + h_k^{(t)}t \\ &\quad + [h_k^{(s,s)}s^2 + 2h_k^{(s,t)}st + h_k^{(t,t)}t^2]/2 \\ &\quad + \text{higher order terms} \end{aligned}$$

where $h_k^{(s)}$ denotes the partial derivative of h_k by s, and so on. By assumption, there is a neighborhood of p in which

$$f_i(h_1(s,t), h_2(s,t), ..., h_n(s,t)) \equiv 0$$

We set

$$\begin{aligned} \delta_1 &= h_1^{(s)}s + h_1^{(t)}t + [h_1^{(s,s)}s^2 + 2h_1^{(s,t)}st + h_1^{(t,t)}t^2]/2 + \cdots \\ \delta_2 &= h_2^{(s)}s + h_2^{(t)}t + [h_2^{(s,s)}s^2 + 2h_2^{(s,t)}st + h_2^{(t,t)}t^2]/2 + \cdots \\ &\vdots \\ \delta_n &= h_n^{(s)}s + h_n^{(t)}t + [h_n^{(s,s)}s^2 + 2h_n^{(s,t)}st + h_n^{(t,t)}t^2]/2 + \cdots \end{aligned}$$

and note that

$$\begin{aligned} \delta_k^2 &= (h_k^{(s)})^2 s^2 + 2h_k^{(s)}h_k^{(t)}st + (h_k^{(t)})^2 t^2 + \cdots \\ \delta_k\delta_j &= h_k^{(s)}h_j^{(s)}s^2 + (h_k^{(s)}h_j^{(t)} + h_k^{(t)}h_j^{(s)})st + h_k^{(t)}h_j^{(t)}t^2 + \cdots \end{aligned}$$

By substitution

$$f_i^{(1)}h_1^{(s)} + f_i^{(2)}h_2^{(s)} + \cdots + f_i^{(n)}h_n^{(s)} = 0 \tag{13}$$

$$f_i^{(1)}h_1^{(t)} + f_i^{(2)}h_2^{(t)} + \cdots + f_i^{(n)}h_n^{(t)} = 0 \tag{14}$$

$$f_i^{(1)}h_1^{(s,s)} + f_i^{(2)}h_2^{(s,s)} + \cdots + f_i^{(n)}h_n^{(s,s)} = -c_i \quad (15)$$

$$f_i^{(1)}h_1^{(s,t)} + f_i^{(2)}h_2^{(s,t)} + \cdots + f_i^{(n)}h_n^{(s,t)} = -d_i \quad (16)$$

$$f_i^{(1)}h_1^{(t,t)} + f_i^{(2)}h_2^{(t,t)} + \cdots + f_i^{(n)}h_n^{(t,t)} = -e_i \quad (17)$$

where $i = 1, ..., m$. The right hand sides are, respectively,

$$\begin{aligned} c_i &= f_i^{(1,1)}(h_1^{(s)})^2 + \cdots + f_i^{(n,n)}(h_n^{(s)})^2 \\ &\quad +2[f_i^{(1,2)}h_1^{(s)}h_2^{(s)} + \cdots + f_i^{(1,n)}h_1^{(s)}h_n^{(s)} \\ &\qquad \cdots + f_i^{(n-1,n)}h_{n-1}^{(s)}h_n^{(s)}] \end{aligned}$$

$$\begin{aligned} d_i &= f_i^{(1,1)}h_1^{(s)}h_1^{(t)} + \cdots + f_i^{(n,n)}h_n^{(s)}h_n^{(t)} \\ &\quad +f_i^{(1,2)}h_1^{(s)}h_2^{(t)} + \cdots + f_i^{(1,n)}h_1^{(s)}h_n^{(t)} \\ &\qquad \cdots + f_i^{(n-1,n)}h_{n-1}^{(s)}h_n^{(t)} \\ &\quad +f_i^{(1,2)}h_1^{(t)}h_2^{(s)} + \cdots + f_i^{(1,n)}h_1^{(t)}h_n^{(s)} \\ &\qquad \cdots + f_i^{(n-1,n)}h_{n-1}^{(t)}h_n^{(s)} \end{aligned}$$

$$\begin{aligned} e_i &= f_i^{(1,1)}(h_1^{(t)})^2 + \cdots + f_i^{(n,n)}(h_n^{(t)})^2 \\ &\quad +2[f_i^{(1,2)}h_1^{(t)}h_2^{(t)} + \cdots + f_i^{(1,n)}h_1^{(t)}h_n^{(t)} \\ &\qquad \cdots + f_i^{(n-1,n)}h_{n-1}^{(t)}h_n^{(t)}] \end{aligned}$$

The partial derivatives of the coordinate functions h_i are computed from these systems of linear equations, and define an approximate local parameterization of the manifold given by Eq. (10).

The linear systems of Eqs. (13–17) have rank $n-2$. Their solutions, therefore, have the form

$$\alpha_1 \nabla f_1 + \cdots + \alpha_{n-2}\nabla f_{n-2} + \beta \mathbf{t}_1 + \gamma \mathbf{t}_2$$

where $\mathbf{t}_1$ and $\mathbf{t}_2$ are two linearly independent tangent directions to the surface at the point p. These tangent directions are determined by the method chosen to solve the linear systems. See also [9].

3.2 Local Curvature

As shown in [10], it is possible to determine the curvature of the surface $\pi(S)$ at p. The main result can be summarized as follows:

Theorem Let $\mathbf{n}_i$ be the normal vector to the hypersurface f_i at the point p, for $1 \le i \le m$. Let α_i be such that the last $n-3$ components of $\mathbf{n}_0 = \sum_{i=1}^m \alpha_i \mathbf{n}_i = (a, b, c, 0, ..., 0)$ are zero, and such that $a^2 + b^2 + c^2 = 1$. Let $\mathbf{v} = (v_1, v_2, ..., v_n)$

be a tangent vector to $\mathcal{S}$ at p where $v_1^2 + v_2^2 + v_3^2 = 1$. Let H_i be the Hessian of f_i. Then the normal curvature of $\pi(\mathcal{S})$ at p in the (projected) direction $\pi(\mathbf{v})$ is given by

$$\kappa = -\mathbf{v}^T(\sum_{i=1}^{m} \alpha_i H_i)\mathbf{v}$$

It is well-known that the principal curvatures and their directions can be recovered from the normal curvatures in three different directions; e.g., [22]. Thus, the principal curvatures, mean curvature, and Gauss curvature of $\pi(\mathcal{S})$ can all be determined with help of the theorem.

3.3 Surface Intersection

The derivation of local approximants can be used to construct local approximants to surface intersections as well. For details, see [3, 13].

3.4 Global Approximation

We want to use the local approximant determination as part of a scheme for globally approximating curves and surfaces defined with the dimensionality paradigm. For curves such as surface intersections it is easy to derive a marching scheme [3, 13]. In order to obtain a good scheme for approximating surfaces we need a way to organize the surface exploration such that the same neighborhood is not explored several times. Since we are ultimately interested in the projection $\pi(\mathcal{S})$, it is advantageous to approximate the projection only, and Chuang's algorithm [9] does this as follows, using a grid in 3-space to detect whether a volume of space has already been explored.

1. At p, a local approximant to $\mathcal{S}$ is constructed.

2. The projected approximant, $(h_1(s,t), h_2(s,t), h_3(s,t))$, is intersected with the faces of the cube, as a function of s and t.

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

4. The estimated point $h_1(s_1, t_1), ..., h_n(s_1, t_1)$ is refined using Newton iteration.

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.

The main advantage of this approach is that the dimension of the meshed space does not depend on the number of variables used to define $\mathcal{S}$. Thus, increasing the number of variables does not raise the complexity of the mesh. 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.

We illustrate the method for linear approximants. Assume that we are at a point $p=(u_1,u_2,...,u_n)$ on $\mathcal{S}$ that projects to $\pi(p)$ in a cube, and we have constructed the linear approximant

$$\begin{aligned} L: \ x_1 &= u_1 + v_1 s + w_1 t \\ x_2 &= u_2 + v_2 s + w_2 t \\ &\vdots \\ x_n &= u_n + v_n s + w_n t \end{aligned}$$

Then L intersects the faces of the cube containing $\pi(p)$, perhaps as shown in Figure 1. Each face intersection is determined. If the face plane is $x_1 = a$, then the line in (s,t)-space corresponding to the intersection with L is simply

$$v_1 s + w_1 t = a - u_1$$

Intersections with the planes $y=b$ and $z=c$ are analogous. So, the intersections define a polygon in (s,t)-space that corresponds to the area of L contained in the cube, as shown in Figure 2. Possibly with help of additional lines corresponding to the intersection of L with the faces of adjacent cubes, we can now find good estimates (s_1,t_1) for new points in neighboring cells, as illustrated in Figure 3.

Implementation shows that a variation of this algorithm is more convenient: Determine the intersection of L with the *edges* of the cube in which L was constructed. These intersections are then refined with Newton iteration, and the deformed polygon so defined in each cube is triangulated, yielding a faceted approximation that is continuous across the facet edges.

4 Summary

The main motivation of this work has been the experience that many geometric operations are simple to express by formulating a system of nonlinear equations, but that a subsequent elimination computation to derive an equivalent implicit equation cannot be carried out in practice, in most cases. Advances in symbolic computation will certainly shift the boundary of when the derivation of an implicit form is practical, but these advances are not likely to displace our approach to complex surface definition.

There is a practical advantage of the higher-dimensional formulation that has not been mentioned yet: Experimental evidence suggests that numerical

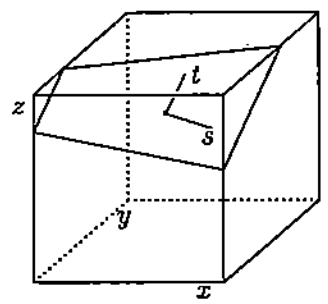

Figure 1: Linear Approximant in a Space Element and Local Frame

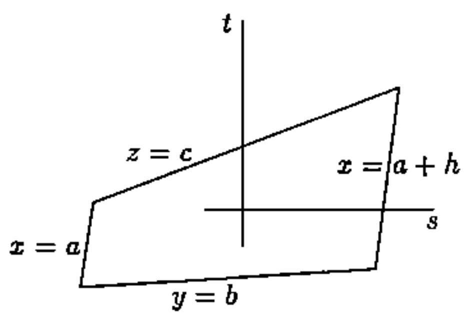

Figure 2: Corresponding Face Intersection Lines in (s, t)-Space

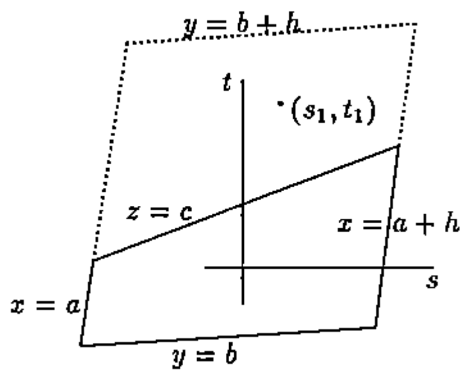

Figure 3: Finding a Point (s_1, t_1) in Adjacent Cubes

geometric computations are more robust in the higher-dimensional form of a surface than in the implicit form. Since the degree of the implicit form can be extremely high even when all equations of the nonlinear system have degree no higher than 2, this observation could be simply a consequence of working with lower degree polynomials.

We may think of integers as representable by a number, or by a product. Then composite integers can be represented in a number of ways, with the full prime factorization at one end of the spectrum, and the number at the other end. In analogy, we think of the implicit form as the number, and of the nonlinear system as a factored form. Variable elimination is then analogous to multiplying factors, but for factoring there seems to be no analogous systematic process. It would be interesting to find such a process.

Acknowledgements

Much of the material summarized here was done in collaboration with collegues and students, and I thank them all. The technical results presented have been published in a number of research articles in books and journals.

References

[1] E. Allgower, K. Georg, and R. Miranda. Computing real solutions of polynomial systems. Technical report, Colorado State University, Mathematics Department, 1990.

[2] E. Allgower and S. Gnutzmann. An algorithm for piecewise linear approximation of implicitly defined two-dimensional surfaces. *SIAM Journal of Numerical Analysis*, 24:452–469, 1987.

[3] C. Bajaj, C. M. Hoffmann, J. Hopcroft, and R. Lynch. Tracing surface intersections. *CAGD*, 5:285-307, 1988.

[4] R. Bartels, J. Beatty, and B. Barsky. *Splines for Use in Computer Graphics and Geometric Modeling.* Morgan Kaufmann, Los Altos, Cal., 1987.

[5] W. Boehm, G. Farin, and J. Kahmann. A survey of curve and surface methods in CAGD. *CAGD*, 1:1–60, 1984.

[6] 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.

[7] B. Buchberger, G. Collins, and B. Kutzler. Algebraic methods for geometric reasoning. *Annual Reviews in Computer Science*, 3:85–120, 1988.

[8] V. Chandru, D. Dutta, and C. Hoffmann. Variable radius blending with cyclides. In K. Preiss, J. Turner, M. Wozny, editor, *Geometric Modeling for Product Engineering*, pages 39–57. North Holland, 1990.

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

[10] J.-H. Chuang and C. Hoffmann. Curvature computations on surfaces in n-space. Technical Report CER-90-34, Purdue University, Computer Science, 1990.

[11] D. Dutta and C. Hoffmann. A geometric investigation of the skeleton of CSG objects. In *Proc. ASME Conf. Design Automation*, Chicago, 1990.

[12] G. Farin. *Curves and Surfaces for Computer-Aided Geometric Design.* Academic Press, 1988.

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

[14] 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.

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

[16] C. M. Hoffmann and G. Vaněček. Fundamental techniques for geometric and solid modeling. In C. T. Leondes, editor, *Advances in Control and Dynamics, 48*, pages 101–165. Academic Press, 1991.

[17] C. M. Hoffmann and P. J. Vermeer. Eliminating extraneous solutions in curve and surface operations. *IJCGA*, 1:47–66, 1991.

[18] J. Hoschek. *Grundlagen der Geometrischen Datenverarbeitung.* Teubner Verlag, Stuttgart, 1989.

[19] D. Kapur. Geometry theorem proving using Hilbert's nullstellensatz. In *SYMSAC 86*, pages 202–208, Waterloo, Ont., 1986.

[20] D. Kapur. A refutational approach to geometry theorem proving. In D. Kapur and J. Mundy, editors, *Geometric Reasoning*, pages 61–93. M.I.T. Press, 1989.

[21] A. Morgan. *Solving Polynomial Systems Using Continuation for Scientific and Engineering Problems.* Prentice-Hall, Englewood Cliffs, N.J., 1987.

[22] J. Pegna and F.-E. Wolter. A simple practical criterion to guarantee second order smoothnes blend surfaces. manuscript, 1989.