# Non-manifold Boundary Representation with Parametric Geometric Elements

by

Y. Wang, L. Gursoz, J. Chen,
F. Prinz, N. Patrikalakis

EDRC 24-37-90

# Non-manifold Boundary Representation with Parametric Geometric Elements

**Yu Wang, E. Levent Gursoz,**
**Jyun-Ming Chen, Friedrich B. Prinz**
**Engineering Design Research Center**
**Carnegie Mellon University**
**Pittsburgh, PA 15213**
**and**
**N. M. Patrikalakis**
**Department of Ocean Engineering**
**Massachusetts Institute of Technology**
**Cambridge, MA 02139**

**October 8,1990**

**Abstract**

This paper describes a development of non-manifold boundary representation with parametric surfaces for solid modeling. We introduce a geometric modeler currently restricted to linear geometric elements, and discuss an investigation to extend its geometric domain to integrate parametric curves and surfaces. Emphasis is given on efficiently maintaining the topological and geometric information of non-manifold boundaries, and the algorithms for robust and efficient Boolean operations of parametric surfaces. The algorithms are demonstrated with examples of non-uniform rational B-spline surfaces.

# 1 Non-manifold Boundaries with Non-linear Geometries

The next generation computer aided geometric modeling systems are likely to combine the wire frame, surface, and solid modeling paradigm. A designer will be able to transform a wire frame sketch into a true solid step by step. Such a procedure requires that the underlying modeling system be capable of classifying or reclassifying the modeling space as the new geometric entities are being created or old ones modified. *Classifying* refers to recognizing the dimensionality, and the genus of all objects involved. Automatically detecting intersections between the entities in the modeling space is another required feature. Systems with such capabilities have been developed recently [1, 2, 3, 4]. However, the implementation has been restricted to the linear domain, i.e., vertices, straight line segments and planar surface facets are the only geometric entities allowed. These so-called *non-manifold* modelers [1, 5] received increasing attention in the literature. The extension of non-manifold modelers into the nonlinear domain is the next logical step.

The geometric modeling system called NOODLES [3] under development uses a collection of disjoint point sets of varying dimensionality to provide uniform geometric modeling in Euclidean space. These point sets are related by their topological adjacency information and are augmented by their geometric representation. A use-based data structure is employed to include a set of topological support elements which represent the non-manifold topologies completely and unambiguously. This structure accommodates multiple geometric interpretations of the topological elements. For example, an edge describing an intersection of two curved surfaces has descriptions in the parameter domains of both surfaces. This design provides a unique capability to logically and physically separate the topological management system from the three-dimensional surface geometry system. There are two important reasons for this, especially in the case of curved surfaces. First, this would enable us to support geometric operation algorithms made independent of the specific details of the geometric representation and allow us to cover and explore alternative geometric representations with minimum programming effort. Second, the geometric entity can be defined by multiple representations, which is very useful in operating (e.g., intersecting, interrogating, and displaying) the geometric entities, since different representations offer different advantages. Currently one of the key barriers in achieving this is the availability of robust and fast intersection algorithms between nonlinear geometric objects.

# 2 Intersections of Parametric Surfaces

As in any geometric modeling system, the computation of intersections between the surfaces of geometric objects is a fundamental problem. There are three basic requirements for this problem: accuracy, efficiency, and reliability. Accuracy specifies the tolerance of the derived intersection

curve with respect to the actual curve lying on the participating surfaces. Reliability requires the solution to be topologically consistent. Each distinct open segment and closed loop that comprise the intersection should be precisely identified. Robust algorithms are required such that the problem can be solved automatically without human intervention. This property is also essential to ensure the global consistency of the geometric data, especially in the case of non-manifold topology, where the co-existence of objects of mixed dimensionality demands more explicit evaluation of their adjacency relationships. Furthermore, numerical errors in finite precision computations complicate the problem. For a future system of a truly unified computer environment to support design, analysis, and manufacturing, it is highly desirable that the robust numerical computations be carried out efficiently and accurately.

Much research has been carried out on the subject of surface to surface intersections, including analytic, subdivision, lattice evaluation, and marching methods. A recent review of the state of the art on intersection algorithms may be found in [6]. Analytic approaches seek analytic expressions for the intersection of two algebraic surfaces [7]. If the surfaces are in low degree rational parametric forms, this type of approach is realistic. Although it can reliably provide a close form representation of the intersection in terms of the implicit polynomial, the complexity of the representation causes it to be impractical for the application of relatively high order parametric polynomial patches such as bicubic patches.

Subdivision methods [8] decompose the surfaces into polygons or triangles and an approximation of the intersection is calculated from intersections of the linear facets. These methods may fail to compute all loops and branches of the intersection, due to the finite level of subdivision. Moreover, topological inconsistencies may also occur because of the linear approximation. Further increase of the subdivision level does not assure the correct topology of intersection, and can rather result in excessive computation. One advantage of these methods is that bounding boxes for subpatches can be built on the basis of control polyhedra. Further subdivision of subpatches can be eliminated if the bounding boxes do not intersect [9]. In lattice evaluation methods, a set of spatial curves lying on one of the surfaces are defined and their intersection points with the other surface are computed. Then these points are sorted and connected to form an approximation to the intersection curve [10]. This method is usually a preparatory step for marching methods. Because of the discrete nature of the method, it is difficult to recognize the existence of small loops and singularities. Marching methods utilize the numerical methods for solving nonlinear equations to obtain the intersection curve. They require starting points to be provided for each branch of the intersection curve and an algorithm to determine the marching step on the curve. These methods are usually very sensitive to singularities and near singularities..

Recently, there has been a strong interest in developing techniques that are capable of handling intersections of critical features such as self-crossing, tangen: contact, small loops, and closely

spaced curves [11,12,13,14]. These techniques include a priori computation of some characteristic points of the intersecting surfaces to guarantee and to guide a reliable and complete computation of all loops and branches of the intersection. The most recent development by [14] applies the orthogonal projection method of [15] to describe the intersection curve by a set of ordinary differential equations and then to trace it by numerical integration. The technique of tracing is shown to be very efficient. The computation of the characteristic points are carried out by some level of initial subdivision of the surfaces, which provides initial guesses for subsequent direct numerical solutions of the characteristic points and the intersection curve. A posteriori verification is performed to ensure that all these points are found, which guarantees the completion of intersection curve. Cones or pyramids bounding normal vectors, or rotational indices of a vector field for each subpatch are tested for the verification. While the tracing method substantially matures the intersection curve computation, however the verification of the characteristic points is not sufficient to exclude the possibility of small loops. Subdividing the surface patch further could increase the level of the confidence, but it would also significantly increase the computational cost.

In the following, we describe a method which has firmer theoretical grounds to ensure that no branches and closed loops of the intersection are missed. The method is based on the techniques of the orthogonal projection and the numerical integration of ordinary differential equations. The method uses the same technique to trace the intersection curve and the characteristic curves. Subsequently, all the characteristic points needed are discovered. Trading for surface subdivision, linear approximation of intersection, and characteristic point verification, we show that this method is more efficient and reliable for a large class of surfaces. This alternative would also provide a unified algorithm which performs both the global structure analysis (i.e., characteristic points calculation and verification) and the local detail tracing (i.e., intersection curve computation).

# 3 Characteristic Points of Intersection Curves

## 3.1 Curves of Surface Intersection

Given two parametric surfaces r(w, v) and q(s, r), the goal of surface intersection is to compute the solutions to the following equation

$$r(ii,v)-q(5,f)»0 \qquad (1)$$

describing the intersection curve in the three-dimensional space. The parametric surfaces are assumed to be defined on some finite domains of the parametric variables (w, v) and (s, *t).* Without loss of generality, we take these domains to be the unit squares w, v, *s, t* € [0, 1]. Figure 1 shows an example of two intersecting bicubic non-uniform rational B-spline (NURBS) surfaces.

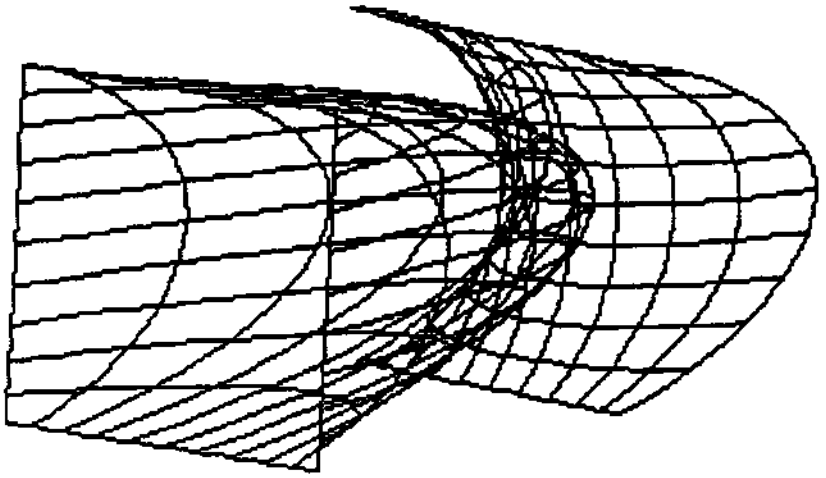

**Figure 1: Two intersecting bicubic NURBS surfaces.**

In general, the intersection curve may be composed of entities of different topology: arcs, closed loops, and tangential contact points. The curve may also have local singular features such as cusps and self-intersections (Figure 2). The Boolean operation requires complete characterization of the intersection curve and accurate computation of discrete points on the curve. These are the very difficulties encountered in the conventional methods discussed in the previous section.

For the problem of intersection of an algebraic surface and a parametric surface, a new algorithm has been proposed [16]. A set of *characteristic points* are defined to partition the curve into a number of *regular* and *monotonic* branches, which are then traversed and connected to complete the curve. The characteristic points are defined by the solution of simultaneous algebraic equations and may be determined by numerical methods such as those using the Bernstein representation and adaptive subdivision [17, 18].

## 3.2 The Distance Function of Orthogonal Projection

An alternative formulation of the intersection problem has been proposed in terms of a distance function of the two surfaces [12,13,14]. The distance function <*f*> between the parametric surfaces r(w, v) and *q(s, t)* is defined as [14]

$$<t>\{u, v) = n, \bullet [r(w, v) - Q(r(w, v))] \quad (2)$$

where Q is the orthogonal projection of the point r(w, v) onto the surface q, and n^ is the unit normal vector at Q on the surface q. The value of the function at a given point on surface r is the

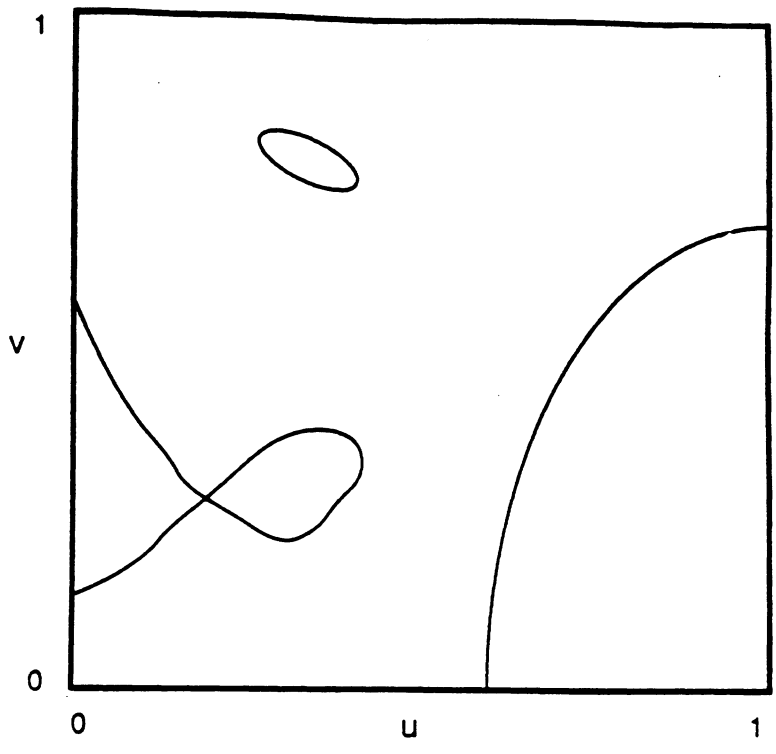


Figure 2: Examples of topological forms of intersection curves in the parameter space.

distance from this point to the corresponding projection **Q**. The orthogonal projection is defined by the following equations[15]

$$(\mathbf{r}(u,v) - \mathbf{q}(s,t)) \bullet \mathbf{q}_s = 0 \quad (3)$$

$$(\mathbf{r}(u,v) - \mathbf{q}(s,t)) \bullet \mathbf{q}_t = 0 \quad (4)$$

The definition of the orthogonal projection requires that the two partial derivatives $\mathbf{q}_s$ and $\mathbf{q}_t$ are linearly independent everywhere, defining the surface normal. It also requires that the surface to be second order continuous. When the orthogonal projection is properly defined, the vector $\mathbf{r} - \mathbf{q}$ is normal to the surface $\mathbf{q}$ at the point **Q**. The distance function $\phi(u,v)$ is then said to be an oriented distance function and to be regular.

In this formulation, the intersection curve between surfaces $\mathbf{r}(u,v)$ and $\mathbf{q}(s,t)$ can be described by the implicit equation

$$\phi(u,v) = 0 \quad (5)$$

in the parameter space $u,\ v \in [0,\ 1]$. The above expression is not explicit, but rather a procedural representation, since the distance function and the orthogonal projection are not explicit. Procedures may be developed to evaluate the distance function and to determine a point on the curve [14].

## 3.3 The Characteristic Points

The alternative formulation of the intersection (5) provides an advantage of direct application of the techniques used for the problem of algebraic and parametric surface intersection discussed above. A key aspect of the techniques includes a set of characteristic points which partition the intersection curve defined on the unit parameter square [0, 1] x [0, 1] into distinct segments and provide the necessary information about the global structure of the curve. [16] defines the set of characteristic points to include:

1. *Border points.* These are the intersection points of a boundary curve of one parametric surface with the other surface. For example,

$$r(w, v) - q(s, t) = 0 \text{ at } u = 0 \tag{6}$$

2. *Turning points.* The w- and v- turning points of the curve are defined as the points which satisfy

$$\phi(u, v) = \langle \pounds_v(w, v) = 0, \quad ftu, \quad v) = \langle t \rangle_u(u, v) = 0 \tag{7}$$

   respectively. These are the points where the curve tangent, when mapped to parameter space, is parallel to the axes $u = 0$ and $v = 0$, respectively.

3. *Singular points.* If a turning point is both a $u$- and a v- turning point, then it is identified as a singular point, satisfying

$$\langle f \rangle(u, v) = \langle f \rangle_u(u_y v) = 4 \rangle v(u, v) = 0 \tag{8}$$

   A singular point can be further classified as a double point, a cusp point, or an isolated point, depending on the Gaussian curvature of the surface $\langle f \rangle(u, v)$ at that point. It may also be of higher multiplicity [16].

These characteristic points are basic points needed to provide minimum information to partition the intersection curve in the parametric space. Algebraic and numerical methods to determine these points are discussed in [16]. Numerical methods based on the convexity property of the Bernstein representation of the algebraic curve intersection to determine these points are discussed in [18]. In addition, another set of characteristic points, *critical points,* are also classified, as related to the existence of closed loops of the curve.

A closed loop of intersection curve or a tangent contact point of intersection is the closed contour curve of the distance function corresponding to the zero distance value, i.e., $\pounds(M, V) = 0$. It has been shown that this contour encircles at least one extrenum of the distance function within the parameter domain of the contour, provided that the function is well defined [13, 14]. The

extremum belongs to a set of critical points where the gradient of the distance function vanishes, i.e., V<£(M, V) = 0. Therefore, a critical point is defined to satisfy

$$<£U(W,V) = <£_V(M,V) = 0 \quad (9)$$

It is observed that if all the critical points are detected, then at least one point on each closed loop of the intersection curve can be determined [19]. If a critical point lies on the intersection curve, then it is also a singular point. These points, in addition tD other characteristic points, would provide sufficient information to start and continue to trace each distinct segments of the curve and to correctly connect them to form the complete intersection curve.

Detection of all characteristic points is the major task to guarantee the robustness of intersection algorithms. Combined with subdivision, the interval method suggested by [11] to check intersections of cones bounding surface subpatch normals does not seem to be very efficient. [14] has further tightened the bound and substantially reduced the computation time. However, the detection and verification of the characteristic points still takes a significant portion of the entire computation, as shown in the examples presented in [14]. It is of great interest to further enhance and improve the efficiency of the solutions of the task. The method of vector field outlined in [12] promises an alternative. However, no technical details and numerical experiences needed to judge its performance were given. [14] suggested to use the Poincar6 index theorem of the vector field to check the existence of a critical point. Both surfaces are subdivided, and then the vector field index is computed for each subpatch corresponding to the distance function. Unfortunately, the condition of zero index is necessary but not sufficient to exclude the existence of a critical point within the subpatch. In the following, we shall examine the global properties of the distance function, and then propose an alternative method which has firmer theoretical grounds to guarantee that no critical points are missed, provided that the critical points are not degenerate.

# 4 Properties of the Distance Function

In this section, we discuss the local and global properties of the distance function, which provide the mathematical foundations for the proposed method.

## 4.1 Continuities of the Distance Function

As we discussed earlier, the distance function </> between the parametric surfaces r(w, v) and q(,s, *t)* is defined as [14]

$$<f>(u, v) = n, \bullet [r(n, v) - Q(r(ii, v))] \quad (10)$$

where Q is the orthogonal projection of the point r(w, v) onto the second surface q, provided that q is of class $C^2$. The local properties of the distance function include the following:

1. $<t>$ is defined almost everywhere [13]. There are two special cases where the function is ill-defined First, for a given point r(w, v) there may be more than one point on the surface q($, *t)* satisfying **the** projection conditions defined by Equations (3) and (4). In this case, the point r(w, v) is on the so called cut-locus of the surface *q(s>* **r)[14].** The second special case arises when the parameter domain of the surface q(,s, *t)* is finite, which is the case considered in this paper, and there does not exist a point on the surface to satisfy the projection conditions. We will discuss the implications of the special cases in Section 6.2.

2. If both surfaces are second order continuous, then $<£(w, v)$ is also second order continuous everywhere except where it is ill-defined. This property is evident if one examines the analytical expressions for the second order derivatives of the distance function given in Appendix A.I. In this paper, we require both surfaces to be of class $C^2$, assuring the second order continuity of the distance function.

## 4.2 Distributions of the Critical Points

Critical points represent global features of the distance function and provide a topological characterization of the shape description of the function. They also have a direct link to the structure of the intersection curve.

Recall that a point is a critical point of the distance function $<£(M,V)$ if $<j>_u = d<f>/du = 0$ and $<f>_v = d<j>/dv = 0$. A critical point is said to be *nondegenerate* if the determinant of the Hessian matrix $H = [<f>ij\ (ij = w, v)$ is nonzero. Otherwise, it is *degenerate.* We consider the case of nondegenerate critical points only. [14] described an approach for a case of degenerate intersection. A nondegenerate critical point is a local maximum or minimum if $det(H) > 0$, and a saddle point if $det(H) < 0$. In the case of nondegenerate critical points, a closed loop of intersection curve encircles at least one local maximum or minimum. A double point on the intersection curve is the case that the curve precisely crosses a saddle point. Under exceptional circumstances for higher order surfaces, more complex critical points may also occur.

In [20], a systematic scheme was presented for describing the distribution of the nondegenerate and isolated critical points on a surface. The surface is characterized by all its critical points and a set of lines called ridge lines and course lines that connect a minimum point or a maximum point with a saddle point and partition the surface into simple regions of slope district. Additional set of points of intersection between a ridge or course line and the boundary of the surface are also included to completely comprise the characterization. It was shown that each ridge or course line must either reach a critical point or intersect the boundary of the surface. It cannot be periodical. For detailed description of the scheme, the reader is referred to [21] and [20]. Such a property seems to be useful to locate all critical points of the surface in the finite domain where it is defined.

If one starts from the points of all ridge or course line lying on the boundary of the surface and traverses the ridge or course lines, one would encounter all the critical points.

In fact, such a scheme would be similar to a method suggested by [12] for obtaining all the intersection curves of two general surfaces. [12] defined a curve between a saddle point and a local maximum or minimum point to be a connecting curve if it goes across both points and also the following ordinary differential equations are also satisfied

$$\dot{u}(t) = \langle f \rangle_u(u,v) \quad (11)$$

$$\dot{v}(r) = \langle f \rangle_v(u,v) \quad (t \geq 0) \quad (12)$$

These are the identical conditions required for ridge and course lines defined in [21]. Such a line between a saddle point and a local extremum point is unique. The connecting lines were suggested in [12] to be used to lead to all critical points.

However, there exist two difficulties in traversing all the ridge and course lines to find all the critical points. First, it is not stable to numerically integrate the ordinary differential equations to approach a saddle point, since it is not a stable critical point and any small perturbation will cause divergence from its precise course, resulting in the integrated trajectories to move away from the saddle point. The second difficulty is to locate the boundary points. [20] have described a necessary but not sufficient condition for a point on the boundary of the surface to be simultaneously on a ridge or course line. Using that condition, fake ridge or course lines may be introduced where none belongs.

# 5 Characteristic Curves of the Distance Function

We propose an alternative to compute all the characteristic points, including all the critical points and the starting points needed for tracing intersection loops. Note that at a critical point $\langle t \rangle_u(u, v) = 0$ and $\langle £_v(w, v) = 0$. Either equation, in fact, describes curves that lie on the surface described by the distance function $\langle f \rangle(u,v)$. These curves are called the *characteristic curves,* and they also have their projection in the parameter space (w, v) and *(s, t).* The equation

$$\wedge \ , \ v \ ) = 0 \quad (13)$$

describes *uncharacteristic curves,* and

$$\langle Mw,vj = 0 \quad (14)$$

describes *v-characteristic curves.* Any intersection of a w-characteristic curve with a v-characteristic curve defines a critical point. Figure 3 illustrates the characteristic curves and the critical point in the parameter space (w,v) for the case of two bicubic NURBS surfaces shown in Figure 1.

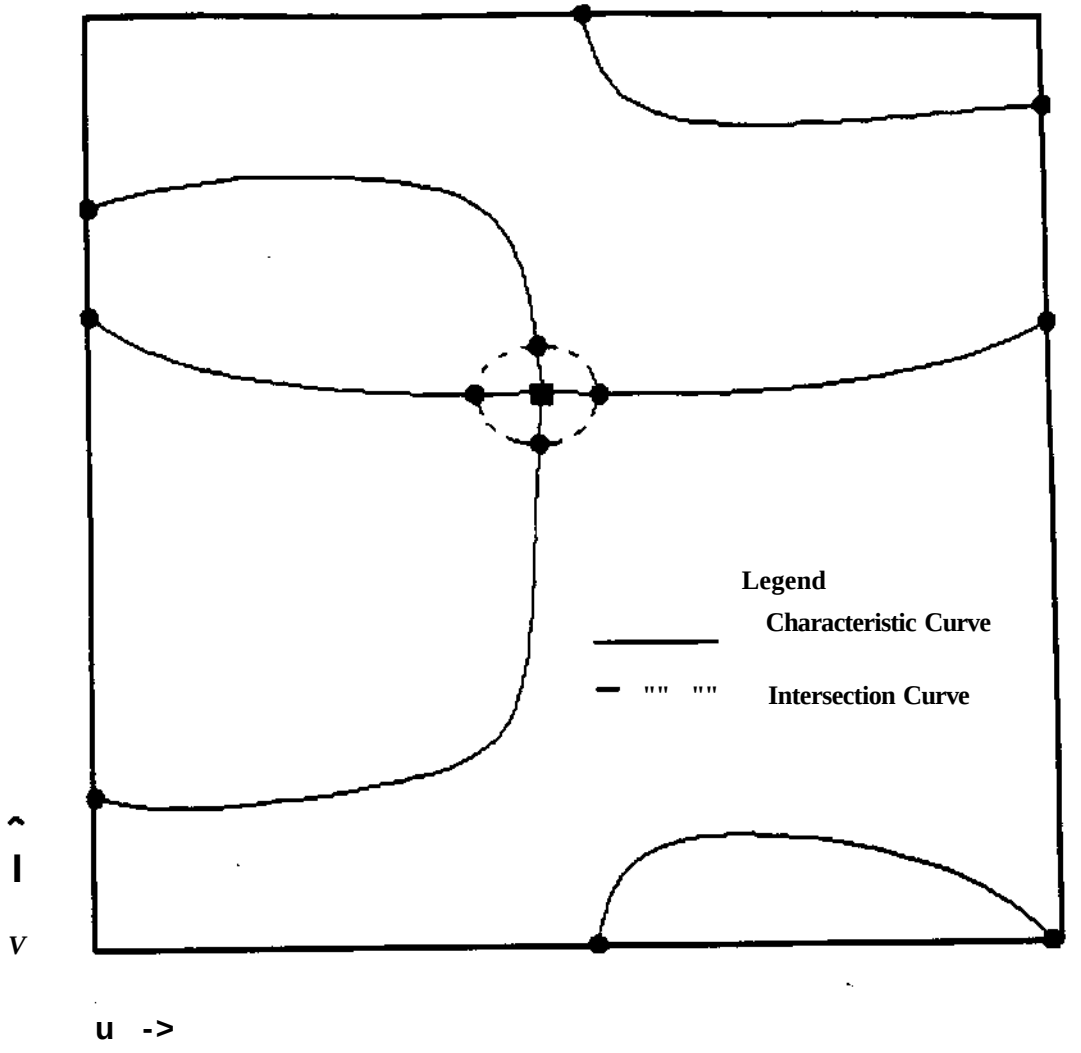


**Figure 3: The characteristic curves and the intersection loop in the parameter domain for two bicubic NURBS surfaces.**

The concept of characteristic curve was introduced in [22] for the generation of contour lines of parametric surfaces.

Unlike the ridge and course lines, the characteristic curves are coordinate-system dependent. However, they offer two important advantages for identifying closed loops of the intersection curve. First, they are described by implicit expressions. This fact allows us to apply various numerical schemes to trace them without the numerical stability problem. This is the subject of next section. Second, critical points are also interconnected by the characteristic curves. If we start from a given critical point to follow its *u*- or v- characteristic curve, there are three possibilities to end the tracing. (1) The characteristic curve reaches another critical point. (2) It intersects the boundary of the surface of the distance function. (3) It approaches a critical point of its own, which is a higher order singularity of the distance function (See next section).

We could use these properties for the purpose of locating all the critical points. Firstly, if all the intersection points of the characteristic curves with the boundary of the distance function surface are identified, then the characteristic curves that start from these intersections can be traced to expose all critical points lying on their paths. Secondly, for every available critical point we traverse all characteristic curves emanating from it to approach other critical points that it connects to. Unfortunately, it might be possible that neither such a boundary point exists nor a critical point is known to exist in the beginning of the process. We do not know in this case if a critical point can exist in the interior of the the distance function parameter domain. This is still under research. In our experience we have not come across such a case where a critical point exists but no boundary point are found. However, one remedy to such a possibility is to find one critical point first using the algorithm suggested in [12] and then to trace its characteristic curves.

Finally, since a characteristic curve must intersect a closed loop of the intersection curve encircling a critical point, the oriented distance function must change sign at the intersection. While marching along a characteristic curve, at least one point on each closed intersection loop will be detected. In fact, one will find all the turning points described in Section 3 of the intersection curve by detecting the sign change while tracing the characteristic curves. A *u*- (v-) characteristic curve will reveal the v- (w-) turning points.

# 6 Computation of Characteristic Curves

## 6.1 Tracing a Characteristic Curve

Since the characteristic curves are described by implicit equations, there are two different methods to compute and trace them. First, we may apply the curve tracing techniques developed for homotopy continuation methods [23]. Second, if we parameterize the characteristic curves by a

parameter $w$, for example, arc length, these curves can be described by the following differential equations

$$\frac{d}{dw}\phi_u(u,v) = 0 \tag{15}$$

for $u$-characteristic curves, and

$$\frac{d}{dw}\phi_v(u,v) = 0 \tag{16}$$

for $v$-characteristic curves.

These curves are similar to the intersection curves defined by $\phi(u,v) = 0$, and they also have two images of the orthogonal projection: one in $(u,v)$ space and the other in $(s,t)$ space. Using the technique of orthogonal projection presented in [15], we can derive the tensorial differential equations describing these images. Therefore, the well-developed methods for integrating ordinary differential equations can be directly applied to trace the characteristic curves. Since this technique has been experimented by [14] for computing the intersection curves with promising performance, we propose to use the same technique for the characteristic curve computation. In fact, a numerical integration technique with adaptive step size control for ordinary differential equations often diminishes the need for the Newton-Raphson corrector for a homotopy continuation method, since the predictor used for integration rarely results in inaccurate numerical solutions. In other words, the Newton-Raphson corrector often converges at once.

The characteristic curves are expressed as the solutions of the following tensorial differential equations:

$$\frac{d\phi_u}{dw} = \phi_{uu}\frac{du}{dw} + \phi_{uv}\frac{dv}{dw} = 0 \tag{17}$$

and

$$\frac{d\phi_v}{dw} = \phi_{vu}\frac{du}{dw} + \phi_{vv}\frac{dv}{dw} = 0 \tag{18}$$

respectively for $u$- and $v$- characteristic curves. Their general solutions are given, respectively,

$$\left[\frac{du}{dw}\ \frac{dv}{dw}\right]^T = \pm\alpha\left[-\phi_{uv}\ \phi_{uu}\right]^T \tag{19}$$

and

$$\left[\frac{du}{dw}\ \frac{dv}{dw}\right]^T = \pm\beta\left[\phi_{vv}\ -\phi_{uv}\right]^T \tag{20}$$

where parameter $w$ corresponds to arc length in the parameter space $(u,v)$, and

$$\alpha = \frac{1}{\sqrt{\phi_{uv}^2 + \phi_{uu}^2}}$$

and

$$\beta = \frac{1}{\sqrt{\phi_{vv}^2 + \phi_{uv}^2}}$$

The orthogonal projection of these curves in the parameter domain of surface q(5, *t)* is given by [14]

$$\begin{bmatrix} ds & dt \end{bmatrix}^T \wedge -- _x - \begin{bmatrix} du & dv \end{bmatrix}^T \tag{21}$$

where K = *[Kg]* and G = $[G\$]_f$ and

$$*i/ = q\,i * q/ - (r - q) \# q// \quad \text{with} \quad ij = s,t \tag{22}$$

and

$$Gij = {}^{r}j \bullet Q< \quad {}^{with} \quad i = s>t \text{ and } ;^{\cdot} = M,V \tag{23}$$

Tracing characteristic curves requires numerical integration of Equations (19) or (20) and (21). There exist a variety of methods for integrating ordinary differential equations with step size control, including adaptive Runge-Kutta methods, multistep Adams methods, and the Bulirsch-Stoer method. We choose the Adams-Bashforth method of multistep solutions. For analysis and implementation of numerical algorithms for theses methods, the reader is referred to a typical numerical analysis textbook, for example, [24], Some numerical experiments are reported in Section 7.

## 6.2 Special Cases of Singularity

As we discussed above, the distance function may not be well-defined at some points in the parameter domain and the characteristic curves may have their own singularities. We discuss remedies for these special cases of singularities.

The first kind of singularity is related to the orthogonal projection used to define the distance function <*j*>. For a given point on the first surface r(w,v), there are three possibilities about its orthogonal projection on the second surface qCs,O- (1) a projection exists and is unique, (2) multiple projections exists, and (3) no projection exists. The first case is said to be regular, and cases (2) and (3) are singular.

The singularity in case (2) will cause the matrix K in Equation (21) to become singular, yielding non-existence of K~*. In such case, one needs to determine every projection of the point r onto the surface q, and computations can be done with each of the projections for tracing the intersection or the characteristic curves. In order not to miss such a singularity, [15] has suggested to check sign change in *det(K)* along numerical curve tracing.

For the third case where a projection does not exist, the problem may be circumvented by introducing either of the following remedies. Any $C^2$ extension of the second surface q may be taken. [13] has shown that such an attempt is feasible when the projection is slightly beyond the boundaries of surface q. [14] used the minimum distance projection [15] as a means of extension. In the present context of tracing a characteristic curve, we shall explicitly compute the points where

the curve intersects the boundary of the surface r or where the image of the curve on the second surface q intersects the boundary of the second surface. These points are the initial points needed to compute the characteristic curves. In this case, it is not necessary to extend the surfaces, since two surfaces do not intersect at points where the orthogonal projection of one surface onto the other does not exist. In Figure 4, we show the boundary of the regular projection on the two surfaces and, correspondingly, in their parameter spaces.

It is also possible that the tensorial differential equations become singular, i.e., the right hand side of the system of differential equations (19) or (20) for a *u*- (v-) characteristic curve becomes zero. In this case, a linearization of the system of differential equations at the singular point can be used to approximate the system solutions, yielding tangent directions of the characteristic curve at the singular point. Each branch of the curve emanating along each tangent is then traced by the numerical integration. This is similar to the technique presented in [14] to trace an intersection curve from singularities, although one order higher expansions of the distance function $\langle j \rangle$ are needed in this case.

## 6.3 Description of the Algorithm

Having emphasized the detection of characteristic points through tracing the characteristic curves, we now briefly describe a numerical method presented in [14] to compute the intersection curve itself and outline the suggested algorithm. In the formulation of the oriented distance function, the intersection curve between the two parametric surfaces r and q is represented by $\langle £(u, v) = 0$. Tensorial differential equations have been derived for the purpose of numerical computation of the curve (See Appendix A.2). Our method for the detection of characteristic points including critical points, in fact, uses the same technique. Based on this technique, we have provided a unified algorithm which performs both topological structure analysis and local detail computation of the intersection curve. The following is a brief description of the suggested algorithm.

Step 1. Compute the border points where one of the boundary curves on either of the surfaces intersects the other surface.

Step 2. Search along the boundaries of the parameter spaces (w, v) and *(s, i)* to find all initial points where either $\langle f \rangle_u(u, v) = 0$ or $0_v(w, v) = 0$, for either u- or v- characteristic curve. This involves root finding in one-dimensional space.

Step 3. For each initial characteristic point found above, integrate the corresponding tensorial differential equations to trace each characteristic curve.

1. If a v- (M-) characteristic point is found along a w- (v-) characteristic curve, then a critical point is found.

2. If $\langle j \rangle = 0$ or change of sign in the value of $\langle f \rangle$ is detected, then a starting point on the intersection curve is found-

Step 4. For each starting point for intersection curve found in Steps 1 and 3, use the method of numerical integration to compute the intersection curve.

# 7 Numerical Experiments and Discussions

In an experimental implementation of our algorithm, we used a computer program developed by NAG [1] for numerical integration of ordinary differential equations with variable step size control. For the intersection curve, the tolerance for controlling the error in the numerical integration of equations describing the curve was specified as $10^{-12}$. We present three example to demonstrate our algorithm. [2]

Figure 4 presents a case of two parametric surfaces, each of degree six in each parameter direction. The intersection of the surfaces consists of three closed loops. A v-characteristic curve was traced to find two starting points on each closed loop and the critical points. These starting points were used as initial points for numerical integration of differential equations describing these closed loops. The boundary of regular projection is also shown in the parameter domains of the two surfaces.

For the cases presented in Figures 5 and 6, both *u*- and v- characteristic curves were traced to initiate the intersection curve computation. In Figure 5, a bicubic NURBS patch intersects a plane surface, while in Figure 6 two nearly coincident quadratic patches intersect.

Table 1 provides the CPU time for the computation of the intersections for the examples. They are performed on a Silicon Graphics Personal IRIS workstation. In these cases, the computational time for tracing the characteristic curves is about half of or the same order as that for tracing the intersection curves. In some other cases that we tested, the computation time for characteristic curve varies from 10% to 200% of that for actual intersection curve, depending on the complexity of two intersecting surfaces.

# 8 Conclusion

Allowing mixed dimensions, the geometric modeling system under development brings together the elements of points, lines, surfaces, and solids by permitting these representations to co-exist in the data structure. This provides a basis of uniform geometry for a wide range of applications

[1] Mark 13, Numerical Algorithm Group, Oxford, England, 1990

[2] The NURBS surfaces of these examples are developed by G. A. Kriezis of MIT.

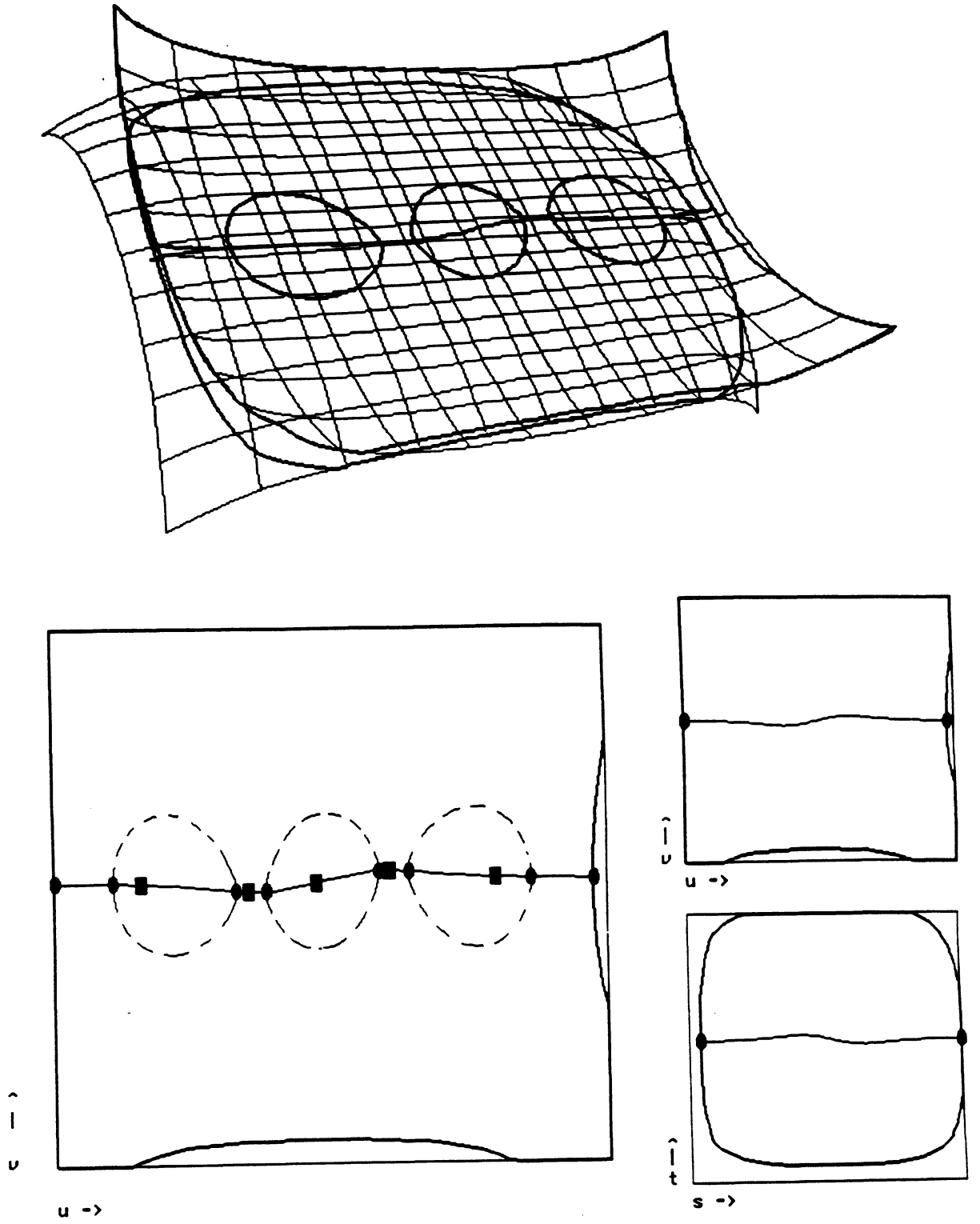


Figure 4: Intersection of two NURBS surfaces of degree six.

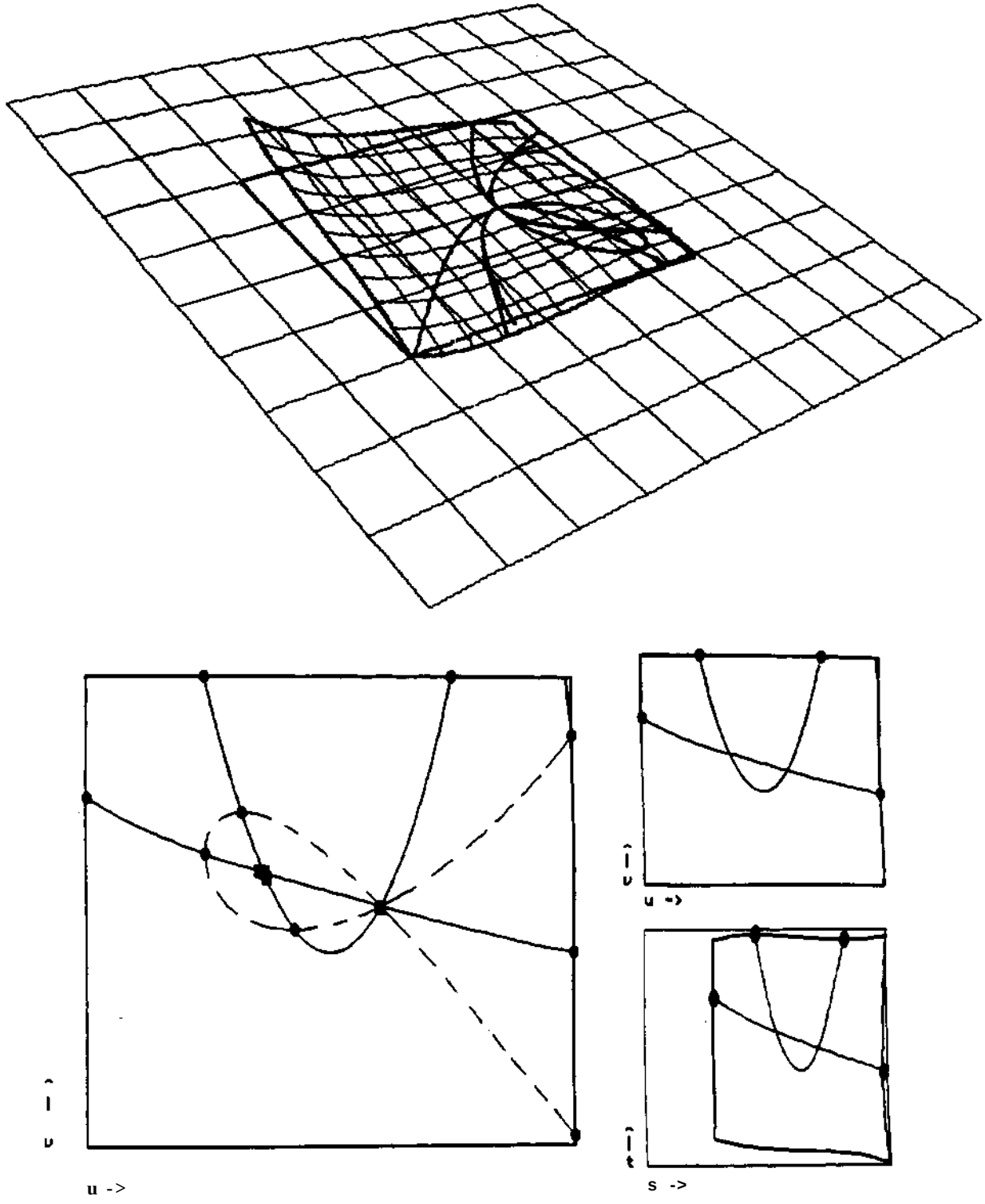


**Figure 5: Intersection of a bicubic patch with a plane.**

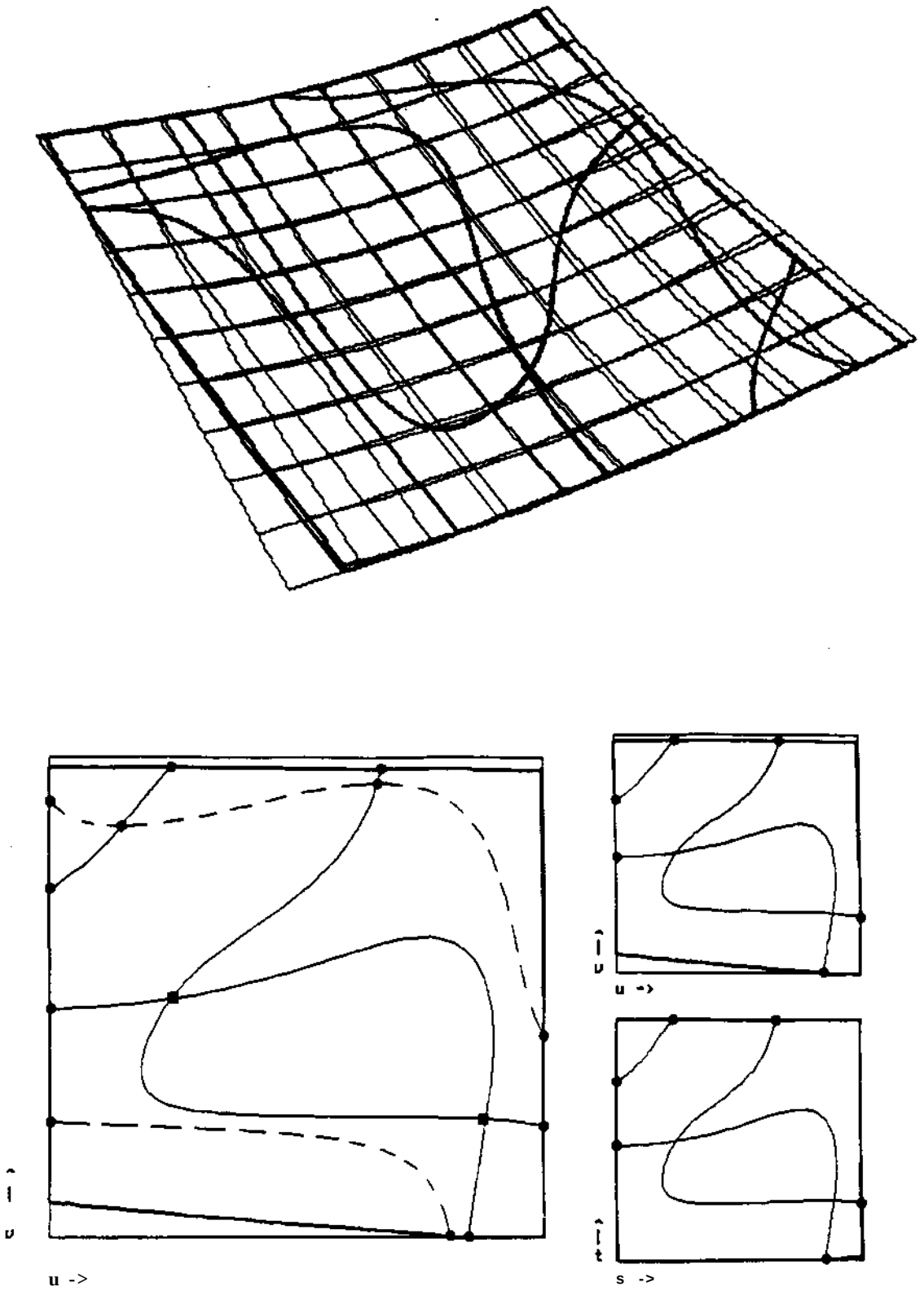


**Figure 6: Intersection of two closely spaced quadratic patches.**

| Figure Number | Characteristic Curve (CPU sec.) | Intersection Curve (CPU sec.) |
|---|---|---|
| 4 | 34.92 | 151.71 |
| 5 | 31.45 | 63.31 |
| 6 | 26.75 | 27.98 |

**Table 1: Computational time for the intersection algorithm.**

in solid modeling and beyond. By introducing curved geometric elements into the non-manifold boundary representation system, we will enhance its generality and enable it to cover a broader range of applications for CAD and CAM. The surface intersection algorithm presented here is aimed to provide the level of robustness and efficiency that are needed in the environment of non-manifold modeling involving arbitrary parametric surfaces or curves. Based upon firmer theoretical grounds, the a priori computation of the characteristic curves is capable of revealing the complete global structure and local singularities of the intersection curve. Upon detecting the topological structure and obtaining a sufficient number of starting points for each segment of the curve, all the segments and closed loops of the curve are numerically traced by integrating the tensorial differential equations describing the curve. Unlike the subdivision-based methods, this algorithm guarantees that no intersection branches and small loops are missed, provided that the intersection is not degenerate.

The algorithm does not require subdivision of the surfaces into subpatches, provided sufficient continuities of the parametric surfaces. This nature provides a possibility to eliminate the need for subdivision in the entire process of Boolean operations, and therefore, to increase the efficiency of such operation. Complexity of traditional algorithms is governed by the efficiency of the schemes used to subdivide the surfaces and by the number of resulting subpatches after eliminating non-intersecting subpatches. Complexity of the presented algorithm depends primarily on the complexity of the participating surfaces and the intersection curve. The characteristic curves of the distance function describe the topological information of the spatial relationship of the two surfaces, but their relative "distance" is irrelevant. In the case that the two surfaces are closely spaced, the unified algorithm without subdivision is more efficient. Otherwise, a hybrid method of coarse subdivision and characteristic curve computation would provide a better balance between efficiency and reliability.

# Acknowledgement

We wish to express our appreciation to Dr. G. A. Kriezis of MIT for his assistance in this work. This work has been supported by the Engineering Design Research Center of Carnegie Mellon University, and by the Concurrent Engineering Research Center of West Virginia University. The MIT part of the work is supported in part by the MIT Sea Grant College Program, the National Science Foundation and the Office of Naval Research under grant numbers NA86AA-D-SG0089, DMC-8720720 and N00014-87-K-0462, respectively.


# A Appendices

## A.1 Derivatives of the Distance Function

The derivatives up to second order of the oriented distance function, defined as

$$\phi(u,v) = \mathbf{n}_q \bullet [\mathbf{r}(u,v) - \mathbf{q}(\mathbf{Q}(\mathbf{r}(u,v)))]$$

where $(\mathbf{r} - \mathbf{q}) \bullet \mathbf{q}_s = 0$ and $(\mathbf{r} - \mathbf{q}) \bullet \mathbf{q}_t = 0$, are given as: [14]

$$\phi_u = \mathbf{n}_q \bullet \mathbf{r}_u \tag{24}$$

$$\phi_v = \mathbf{n}_q \bullet \mathbf{r}_v \tag{25}$$

$$\phi_{uu} = \mathbf{n}_{q,u} \bullet \mathbf{r}_u + \mathbf{n}_q \bullet \mathbf{r}_{uu} \tag{26}$$

$$\phi_{uv} = \mathbf{n}_{q,v} \bullet \mathbf{r}_u + \mathbf{n}_q \bullet \mathbf{r}_{uv} \tag{27}$$

$$\phi_{vu} = \mathbf{n}_{q,u} \bullet \mathbf{r}_v + \mathbf{n}_q \bullet \mathbf{r}_{vu} \tag{28}$$

$$\phi_{vv} = \mathbf{n}_{q,v} \bullet \mathbf{r}_v + \mathbf{n}_q \bullet \mathbf{r}_{vv} \tag{29}$$

where

$$\mathbf{n}_{q,u} = [\mathbf{n}_{q,s}\, \mathbf{n}_{q,t}] \bullet \{\mathbf{K}^{-1}[\mathbf{r}_u \bullet \mathbf{q}_s\, \mathbf{r}_u \bullet \mathbf{q}_t]^T\} \tag{30}$$

$$\mathbf{n}_{q,v} = [\mathbf{n}_{q,s}\, \mathbf{n}_{q,t}] \bullet \{\mathbf{K}^{-1}[\mathbf{r}_v \bullet \mathbf{q}_s\, \mathbf{r}_v \bullet \mathbf{q}_t]^T\} \tag{31}$$

where $\mathbf{K}$ is given in Equation (22).

## A.2 Differential Equations for Intersection Curves

Tensorial differential equations for an intersection curve described by $\phi(u,v) = 0$ are defined by differentiating the equation with respect to a parameter $w$ [14]:

$$\frac{d\phi}{dw} = \phi_u \frac{du}{dw} + \phi_v \frac{dv}{dw} = 0 \tag{32}$$

**Its general solutions are given as**

$$\left[\frac{du}{dw}\ \frac{dv}{dw}\right]^T = \left[\frac{\phi_v}{\sqrt{\phi_u^2+\phi_v^2}}\ \frac{-\phi_u}{\sqrt{\phi_u^2+\phi_v^2}}\right]^T \tag{33}$$

**where parameter *w* corresponds to arc length in the parameter space (w,v). The orthogonal projection of the curve in the parameter domain of surface q(s,r) is given by Equation (21). Together, a system of four dimensional first order differential equations is defined.**

# References


[1] K. J. Wetter, "Edge based data structures for solid modeling in curved-surface environments," *IEEE Computer Graphics and Applications,* vol. 5, pp. 21-40, January 1985.

[2] H. Masuda et al., "A mathematical theory and application of non-manifold geometric modeling," in *Advanced Geometric Modelling for Engineering Applications* (F. -L. Krause and H. Jansen, eds.), pp. 89 - 103, New York: North-Holland, 1989.

[3] E. L. Gursoz, Y. Choi, and F. B. Prinz, "Vertex-based representation of non-manifold boundaries," in *Geometric Modeling for Product Engineering* (M. J. Wozny, J. U. Turner, and K. Preiss, eds.), pp. 107 - 130, New York: North-Holland, 1990.

[4] Y. Yamaguchi and F. Kimura, "Boundary Neighborhood Representation for Non-manifold Topology," in *Proc.forlFIP WG 52 on Geometric Modeling,* Rensselaerville, 1990.

[5] J. R. Rossignac and M. A. O'Connor, "Sgc: A dimension-independent model for pointsets with internal structures and incomplete boundaries," Tech. Rep. RC 14340, IBM, 1989.

[6] N. M. Patrikalakis, "Interrogation of Surface Intersection," in *Geometry Processing* (R. E. Barnhill, ed.), SIAM, 1990. To Appear.

[7] T. W. Sederberg, D. C. Anderson, and R. N. Goldman, "Implicit representation of parametric curves and surfaces," *Computer Vision, Graphics, and Image Processing,* vol. 28, pp. 72 - 84, 1984.

[8] E. G. Houghton *et al.,* "Implementation of a divide-and-conquer method for intersection of parametric surfaces," *Computer Aided Geometric Design,* vol. 2, pp. 173 - 183,1985.

[9] Q. S. Peng, "An algorithm for finding the intersection lines between two b-spline surfaces," *Computer Aided Design,* vol. 16, no. 4, pp. 191 - 196,1984.

[10] K.-Y. Wang, "Intersections of general parametric surfaces," Tech. Rep. 89CRD003, Automation Systems Laboratory, GE Research and Development Center, January 1989.

[11] T. W. Sederberg and R. J. Meyers, "Loop detection in surface patch intersections," *Computer Aided Geometric Design,* vol. 5, pp. 161 - 171,1988.

[12] K.-R Cheng, "Using plane vector fields to obtain all the intersection curves of two general surfaces," in *Theory and Practice of Geometric Modeling* (W. Strasser and H.-P. Seidel, eds.), pp. 187-204, New York: Springer-Verlag, 1989.

[13] R. P. Markot and R. L. Magedson, "Solutions of tangential surface and curve intersections," *Computer-Aided Design,* vol. 21, no. 7, pp. 421 - 429,1989.

[14] G. A. Kriezis, *Algorithms for Rational Spline Surface Intersections.* PhD thesis, Depanment of Ocean Engineering, Massachusetts Institute of Technology, March 1990.

[15] J. Pegna and F.-E. Wolter, "Designing and mapping trimming curves on surfaces using orthogonal projection," in *ASME 16th Design Automation Conference,* (Chicago, IL), ASME, vol. 1, pp. 235 - 245, September, 1990.

[16] R. T. Farouki, "The characterization of parametric surface sections," *Computer Vision, Graphics, and Image Processing,* vol. 33, pp. 209 - 236,1986.

[17] R. B. Lee and R. F. Fredericks, "Intersection of parametric surfaces and a plane," *IEEE Computer Graphics and Applications,* vol. 4, no. 8, pp. 48 - 51,1984.

[18] N. M. Patrikalakis and P. V. Prakash, "Surface intersections for geometric modeling," *Journal of Mechanical Design, ASME Transactions,* vol. 112, pp. 100- 107,1990.

[19] P. Sinha, E. Klassen, and K. K. Wang, "Exploiting topological and geometric properties for selective subdivision," in *Proceedings of Symposium on Computational Geometry,* (Baltimore, Maryland), pp. 39 - 45, ACM SIGGRAPH, June 1985.

[20] L. R. Nackman and S. M. Pizer, "Three-dimensional shape description using the symmetric axis transform. I: Theory," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-7, pp. 187 - 202, March 1985.

[21] L. R. Nackman, "Three-dimensional critical point configuration graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-6, pp. 442 - 450, March 1984.

[22] Y. J. Chen and B. Ravani, "Offset surface generation and contouring in computer-aided design," *Journal of Mechanisms, Transmissions, and Automation in Design,* vol. 109, pp. 133 -142, March 1987.

[23] W. I. Zangwill and C. B. Garcia, *Pathways to Solutions, Fixed Points, and Equilibria.* Englewood Cliffs, NJ: Prentice-Hall, 1981.

[24] G. Hall and J. M. Watt, (eds.), *Modern Numerical Methodsjor Ordinary Differential Equations.* Oxford: Clarendon Press, 1976.