Second Order Surface Analysis Using Hybrid Symbolic and Numeric Operators*

Gershon Elber† and Elaine Cohen
Computer Science Department,
University of Utah

February 28, 1997

Abstract

Results from analyzing the curvature of a surface can be used to improve the implementation, efficiency, and effectiveness of manufacturing and visualization of sculptured surfaces.

In this paper, we develop a robust method using hybrid symbolic and numeric operators to create trimmed surfaces each of which is solely convex, concave, or saddle and partitions the original surface. The same method is also used to identify regions whose curvature lies within prespecified bounds.

1 Introduction

A critical characteristic for many applications in computer graphics and in CAD is the shape of the model's surface. Second order surface analysis can be used to understand curvature characteristics, and thus shape, and to improve the implementation, efficiency and effectiveness of manufacturing and analysis processes. Fundamental operations, such as adaptive subdivision and refinement, use shape information to decide where and how many knots to add. Algorithms for the creation of tool paths for NC (Numerically Controlled) code generation for freeform surfaces are usually based on ball end cutters with their spherical centers following an (approximate) offset surface of the original surface. Flat end cutters can remove material faster and have a better finish; however, flat end cutters can be used only with 5 axis milling in convex regions (see Figure 1).

*This work was supported in part by DARPA (N00014-88-K-0689). All opinions, findings, conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

†Appreciation is expressed to IBM for partial fellowship support of the first author.

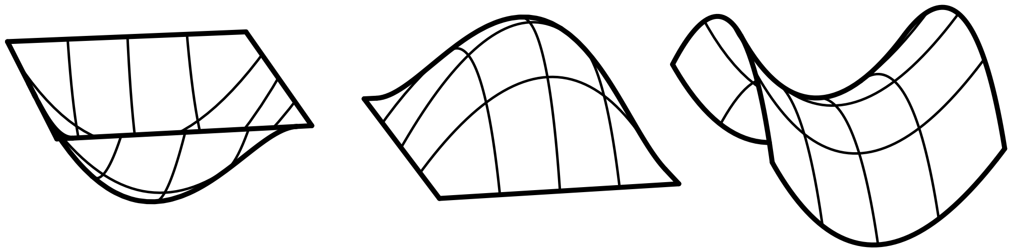

Figure 1: Mainly concave (left), convex (middle) and saddle (right) regions.

Definition 1.1 *A* surface trichotomy *is a partition of a surface into three types of regions: convex, concave and saddle shapes (Figure 1).*

The ability to trichotomize sculptured surfaces into convex, concave or saddle regions (Figure 1) is thus essential to the use of flat end cutters in milling freeform surfaces. Also, regions with small curvature can be accurately milled faster with larger ball end cutters. Since tool changes are time consuming operations they should be minimized. Such minimization can be achieved by subdividing the surface into regions with different curvature bounds, each of which can be milled using tools appropriate to that region.

Methods in use do not support the separation of original surfaces into trimmed surfaces each of which has only one of the three characteristics throughout. That is, either convex everywhere, concave everywhere, or saddle everywhere. Second order surface properties are usually estimated locally by numerically evaluating them at a grid of points or at a finite set of sampled points along the planned milling tool path, in manufacturing. Research into computing curvature has been done in the context of offset operator approximations with cubic B-spline curves [25] and bicubic patches [13].

There have been attempts [1, 2, 8, 17] to understand and compute second order surface properties as well as twist, by evaluation on a predefined grid. The methods use the Gaussian curvature $K(u,v) = \kappa_n^1(u,v)\kappa_n^2(u,v)$ and mean curvature $H(u,v) = \frac{\kappa_n^1(u,v)+\kappa_n^2(u,v)}{2}$, where $\kappa_n^1(u,v)$ and $\kappa_n^2(u,v)$ are the *principal curvatures* at the parameter value (u,v), in an attempt to provide a bound on the surface angularity. However, if the surface is a saddle at (u,v), then κ_n^1 and κ_n^2 have different signs so the magnitude of H is not a useful measure of such a bound. In the extreme condition when the surface is minimal [3], $H \equiv 0$ regardless of the surface angularity. The magnitude of K can also be ineffective. Even if κ_n^1 is large, K may be small because κ_n^2 is small. Therefore, neither K nor H by itself can provide sufficient shape information for subdivision and/or efficient NC applications. This problem has been recognized by some of the authors cited above. These curvature estimation techniques are *local*, since they make use of local surface information only. More surface information might improve an algorithm or change a decision. Local information is inferior to global information in complex settings. Symbolic techniques can be used to help make decisions based upon the entire aspect of a surface rather than a limited number of local samples.

In this paper a hybrid approach using both symbolic and numeric operations for computing curvature properties is developed. We use scalar and vector fields whose definitions are derived from different attributes of the original surface, as auxiliary scalar and vector fields to help analyze the original surface. The zero set of the second fundamental form computed symbolically as a scalar field is used to robustly and completely trichotomize the surface into saddle, convex and concave regions. Bounds on the curvature of the surface are also computed from scalar fields and are analyzed. For example, $\frac{\partial F}{\partial u}(u,v)$ is a vector fields from F, and so is $n(u,v)$, the vector field of unit normals. The two surfaces of principal curvatures, $\kappa_n^1(u,v)$ and $\kappa_n^2(u,v)$ are scalar fields.

Some surfaces have the same domain as the original surface while others do not. If F is a tensor product NURBs surface, then $\frac{\partial F}{\partial u}(u,v)$ is a vector field which is also a tensor product NURBs surface with the same knot vectors, but with lower order and continuity properties. $\frac{\partial F}{\partial u}(u,v) \times \frac{\partial F}{\partial v}(u,v)$ is also a vector field that is a tensor product NURBs surface, but with different knot vectors and order, and lower continuity. However, $n(u,v)$, $\kappa_n^1(u,v)$, and $\kappa_n^2(u,v)$ cannot be represented as piecewise rational parametric functions, as we shall later see, and hence, cannot be represented, in general, as NURBs surfaces.

Contouring techniques [2, 18, 22] developed for freeform surfaces can be applied immediately to bivariate scalar and vector fields once they are represented as NURBs. Since both the original surface and the derived scalar and vector fields share the same parametric domain, one can easily trim the regions in the original surface having certain values. In other cases, the zero sets of the scalar and vector fields might be required. For example, let $\hat{n}_z(u,v)$ be the z component of $\hat{n}(u,v)$, where $\hat{n}(u,v) = \frac{\partial F(u,v)}{\partial u} \times \frac{\partial F(u,v)}{\partial v}$ is an orthogonal vector to the parametric surface $F(u,v)$ at (u,v). Let the orthographic view direction be $+z$. Then the set of zeros of $\hat{n}_z(u,v)$ is simply the parameter values for the silhouettes of the original surface. In other words, the silhouette extraction problem can be mapped to a root finding problem (contouring), which is usually simpler. Trimmed surfaces [4, 19] are the natural way to represent the regions defined by the contouring operator. In fact, the parameter values of the contours of the bivariate vector fields can serve as the parameter values of trimming curves of the original surface.

Throughout this paper examples and properties will be shown using the Bézier and NURBs surface representations. However, any other representation which has analogous subdivision, variation diminishing, and convex hull properties, and supports the operators in Section 3 can be used. All surfaces and images were created and rendered using the Alpha_1 solid modeler developed at the University of Utah.

Section 2 briefly develops the differential geometry used in the analysis. In Section 3 we develop the tools and operators that are required in this analysis while in Section 4 we use these tools to compute second order properties, and use visualization to better understand the shape of a given surface.

2 Differential Geometry

Surface curvature is well understood mathematically and the theory behind it is developed in most introductory differential geometry books [3, 20, 24]. The set of analysis equations that are based on the second fundamental form are used extensively in locally evaluating surface curvature. Because these equations are crucial to our discussion, they are briefly stated here.

Let $F(u,v)$ be a $C^{(2)}$ regular parametric surface. Let the *unnormalized normal* to a surface $F(u,v)$, $\hat{n}(u,v)$, be defined as

$$\hat{n}(u,v) = \frac{\partial F}{\partial u} \times \frac{\partial F}{\partial v}, \tag{1}$$

and define the unit normal, $n(u,v)$, to be

$$n(u,v) = \frac{\frac{\partial F}{\partial u} \times \frac{\partial F}{\partial v}}{\|\frac{\partial F}{\partial u} \times \frac{\partial F}{\partial v}\|}. \tag{2}$$

Since $F(u,v)$ is regular, $\|\hat{n}(u,v)\| \neq 0$ and $n(u,v)$ is well defined.

Let $C(t) = F(u(t), v(t))$ be a regular curve on F, that is $\left\|\frac{dC(t)}{dt}\right\| \neq 0$. The rate of change of the arc length of C with respect to its parameter, t, is $\frac{ds}{dt} = \left\|\frac{dC(t)}{dt}\right\|$ where s is arc length. Since $\frac{dC(t)}{dt} = \left(\frac{\partial F}{\partial u}\frac{du}{dt} + \frac{\partial F}{\partial v}\frac{dv}{dt}\right)$, one can show [16, 20, 24] that

$$\left(\frac{ds}{dt}\right)^2 = \left[\frac{du}{dt} \quad \frac{dv}{dt}\right] G \left[\frac{du}{dt} \quad \frac{dv}{dt}\right]^T = I\left(\frac{du}{dt}, \frac{dv}{dt}\right).$$

I is known as the first fundamental form, with matrix G equal to:

$$G = (g_{ij}) = \begin{bmatrix} \left\langle \frac{\partial F}{\partial u}, \frac{\partial F}{\partial u} \right\rangle & \left\langle \frac{\partial F}{\partial u}, \frac{\partial F}{\partial v} \right\rangle \\ \\ \left\langle \frac{\partial F}{\partial v}, \frac{\partial F}{\partial u} \right\rangle & \left\langle \frac{\partial F}{\partial v}, \frac{\partial F}{\partial v} \right\rangle \end{bmatrix}, \tag{3}$$

where $\langle \cdot, \cdot \rangle$ denotes an inner product.

By considering all such curves, $C(t)$, through a point (u,v) and differentiating twice, one can extract second order properties of the surface F at (u,v). The second order derivatives of $C(t)$ contain terms with $\frac{\partial F}{\partial u}$ and $\frac{\partial F}{\partial v}$ as factors. However, the inner product of these terms with n is always zero since the partials are in the tangent plane of $F(u,v)$. Therefore, $\left\langle n(u,v), \frac{d^2C(t)}{dt^2} \right\rangle$, the component of $\frac{d^2C(t)}{dt^2}$ pointing in the direction perpendicular to the surface is composed of second order derivatives only.

$$\left\langle n(u,v), \frac{d^2C(t)}{dt^2} \right\rangle$$

$$
\begin{aligned}
&= \left\langle n(u,v), \frac{\partial^2 F}{\partial u^2} \right\rangle \left(\frac{du}{dt} \right)^2 + 2 \left\langle n(u,v), \frac{\partial^2 F}{\partial u \partial v} \right\rangle \frac{du}{dt}\frac{dv}{dt} + \left\langle n(u,v), \frac{\partial^2 F}{\partial v^2} \right\rangle \left(\frac{dv}{dt} \right)^2 \\
&= \begin{bmatrix} \frac{du}{dt} & \frac{dv}{dt} \end{bmatrix} L \begin{bmatrix} \frac{du}{dt} & \frac{dv}{dt} \end{bmatrix}^T \\
&= II\left(\frac{du}{dt}, \frac{dv}{dt} \right). \qquad (4)
\end{aligned}
$$

II is known as the second fundamental form, with matrix L equal to:

$$
L = (l_{ij}) = \begin{bmatrix} \left\langle n, \frac{\partial^2 F}{\partial u^2} \right\rangle & \left\langle n, \frac{\partial^2 F}{\partial u \partial v} \right\rangle \\ \left\langle n, \frac{\partial^2 F}{\partial u \partial v} \right\rangle & \left\langle n, \frac{\partial^2 F}{\partial v^2} \right\rangle \end{bmatrix}. \qquad (5)
$$

Let $\hat{l}_{ij}$ denote the inner product with the unnormalized normal $\hat{n}(u,v)$. For example, $\hat{l}_{11} = \left\langle \hat{n}, \frac{\partial^2 F}{\partial u^2} \right\rangle$.

The normal curvature on the surface $F(u,v)$ in some tangent direction Δ, where $\Delta = \left\langle \delta, \left(\frac{dF}{du}, \frac{dF}{dv} \right) \right\rangle$, and $\delta = \left(\frac{du}{dt}, \frac{dv}{dt} \right)$, is defined [3, 16, 20, 24] as:

$$
\kappa_n = \frac{II(\frac{du}{dt}, \frac{dv}{dt})}{I(\frac{du}{dt}, \frac{dv}{dt})} = \frac{\delta L \delta^T}{\delta G \delta^T}. \qquad (6)
$$

The normal curvature depends on the surface tangent direction Δ, and is equal to the curvature of the osculating circle to the intersection curve between $F(u,v)$ and the plane through $n(u,v)$ and Δ at (u,v) (Figure 2). The extremal values of the normal curvature serve as bounds on the components of curvature not contained in the tangent plane.

The normal curvature is an *intrinsic property* [20, 24] of the surface, that is, not dependent on parametrization. By differentiating (6) with respect to δ, the problem of finding extrema of κ_n is transformed [3, 16, 20, 24] into the problem of solving for the roots of

$$
|G|\,\kappa_n^2 + (g_{11}l_{22} + l_{11}g_{22} - 2g_{12}l_{12})\kappa_n + |L| = a\kappa_n^2 + b\kappa_n + c = 0, \qquad (7)
$$

where $|G|$ and $|L|$ denotes the determinants of G and L, respectively.

The Gaussian curvature is a scalar value and is defined as the product of the two roots of (7), κ_n^1 and κ_n^2,

$$
K = \kappa_n^1 \kappa_n^2 = \frac{|L|}{|G|}. \qquad (8)
$$

The mean curvature is define as their arithmetic average,

$$
H = \frac{\kappa_n^1 + \kappa_n^2}{2} = -\frac{(g_{11}l_{22} + l_{11}g_{22} - 2g_{12}l_{12})}{2\,|G|}. \qquad (9)
$$

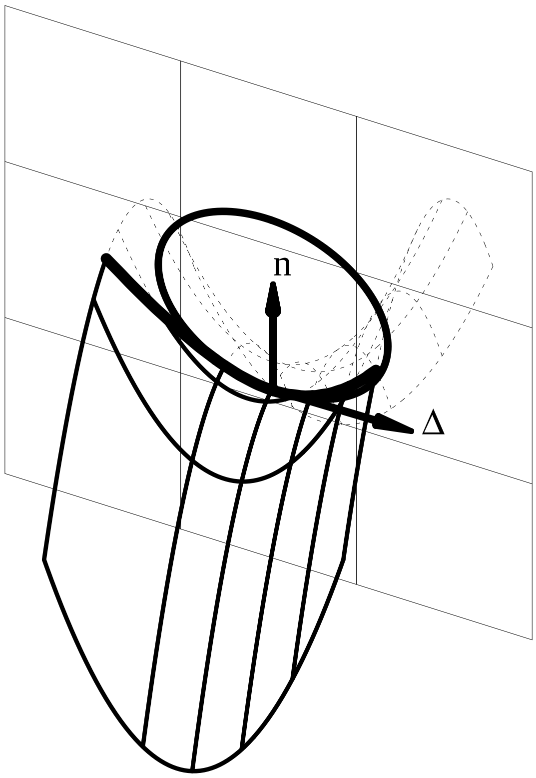

Figure 2: Normal curvature κ_n (circle) of $F(u, v)$ at (u, v) in direction Δ.

3 Tools and Operators

In order to *symbolically* represent curvature properties as bivariate Bézier or NURBs vector fields, one must be able to represent a surface which is the symbolic sum, difference, and product of surfaces, or the derivative of a surface, as a single Bézier or NURBs surface. Methods to represent, as a single Bézier or NURBs surface, the results of the above operators on Bézier or NURBs surfaces, are presented below.

3.1 Symbolic computation

Given a Bézier or NURBs curve, the NURBS representation of the derivative is well known [12] to be,

$$\frac{dC(t)}{dt} = \frac{d\sum_{i=0}^{m-1} P_i B_{i,\tau}^k(t)}{dt} = (k-1)\sum_{i=0}^{m-2} \frac{P_{i+1} - P_i}{t_{i+k} - t_i} B_{i,\tau}^{k-1}(t), \tag{10}$$

where P_i are points of the control polygons of $C(t)$. This result easily extends to tensor product surfaces.

The symbolic computation of sum and/or difference of vector field represented as Bézier or NURBs curves is achieved by computing the sum and/or difference of their respective coefficients [11, 12, 14], once the two curves are in the same space. This requirement can be met by representing them as NURBs vector fields with the same order (using degree raising [5, 6] on the lower order one, if necessary) and the same continuity (using refinement [7] of knot vectors for NURBs).

$$\begin{aligned} C_1(t) \pm C_2(t) &= \sum_{i=0}^{k} p_i B_{i,\tau}^k(t) \pm \sum_{i=0}^{k} q_i B_{i,\tau}^k(t) \\ &= \sum_{i=0}^{k} (p_i \pm q_i)\, B_{i,\tau}^k(u), \end{aligned} \tag{11}$$

where p_i and q_i are vector coefficients of the scalar blending functions. This result easily extends to tensor product surfaces as well [11].

Representation of the product of two scalar fields or one vector fields and one scalar fields is the last symbolic tool required. For Bézier curves [14, 12],

$$\begin{aligned} C_1(t)C_2(t) &= \sum_{i=0}^{m} p_i B_i^m(t) \sum_{j=0}^{n} q_j B_j^n(t) \\ &= \sum_{k=0}^{m+n} r_k B_k^{m+n}(t), \end{aligned} \tag{12}$$

where

$$r_k = \sum_{\substack{i,\, j \\ i+j=k}} p_i q_j \frac{\binom{m}{i}\binom{n}{j}}{\binom{m+n}{k}}.$$

This result can also be extended to tensor product surfaces [11]. It is also necessary to represent scalar products as part of representing sums and differences of rational curves and surfaces, as well as for representing derivatives of rationals.

Finding a representation for the product of NURBs is far more difficult. One might consider subdividing the surfaces into Bézier patches at all the interior knots, computing the product, and merging the results back. However, the continuity information along the interior knots is lost. The NURBs representation can be computed in two different ways. One recently developed method [21] supports symbolic computation of the coefficients of the product after finding the knot vector of the product curve. Since this is computationally expensive and complex to implement, one might choose to exploit the B-spline representation's uniqueness property and compute the coefficients of the product by solving an equivalent interpolation problem [11].

3.2 Contouring operator

We would be interested in finding the zero set of a bivariate scalar or vector field or the constant set of it. These sets, referred to as contours, in the parameter space of the scalar or vector field can be used as trimming curves for the original surface [19], so the trimmed surface will hold all regions of the original surface with values larger (or smaller) than the contouring level.

Computing the contours is closely related to computing surface-surface intersections and ray-surface intersections, problems with inherent numerical complexities.

Let $F(u,v) = (\frac{x(u,v)}{w(u,v)}, \frac{y(u,v)}{w(u,v)}, \frac{z(u,v)}{w(u,v)})$ and $P = Ax + By + Cz + D = 0$ be a bivariate vector field and a contouring plane, respectively. By substituting the components of $F(u,v)$ into P, one can solve for all values of u and v in the parametric domain for which $F(u,v) \cap P$ is not empty.

$$\begin{aligned} S(u,v) &= A\frac{x(u,v)}{w(u,v)} + B\frac{y(u,v)}{w(u,v)} + C\frac{z(u,v)}{w(u,v)} + D \\ &= \frac{Ax(u,v) + By(u,v) + Cz(u,v) + Dw(u,v)}{w(u,v)}. \end{aligned} \tag{13}$$

A single NURBs surface representation for (13) can be found using the operations defined in section 3.1, namely surface addition and surface multiplication. The zero set of the surface $S(u,v)$, in (13), is the set of parametric values for the required intersection, and can be found using a subdivision based approach [11]. Since both $F(u,v)$ and $S(u,v)$ share the same parametric domain, mapping the parametric domain information back to $F(u,v)$ is trivial. $S(u,v)$ is a *scalar* surface, which leads to a simpler and faster computation. Assuming $w(u,v) \neq 0$, the zero set of $S(u,v)$ can be computed using only the numerator of $S(u,v)$. Thus, even if $F(u,v)$ is a rational surface, contouring computations can be performed on scalar polynomial surfaces. If $F(u,v)$ is a scalar field, that is $F(u,v) = (\frac{z(u,v)}{w(u,v)})$, constant z contouring can be performed in a similar way, but with $S(u,v) = C\frac{z(u,v)}{w(u,v)} + D = \frac{Cz(u,v)+Dw(u,v)}{w(u,v)}$. In the following section, the tools discussed above are used. The basic operations for surfaces, addition, subtraction, and multiplication are combined with differentiation to compute or approximate bivariate scalar and vector fields, as necessary. Then, the contouring algorithm will be used to analyze and extract useful information from these surfaces.

4 The approach

The tools defined in Section 3 can now be used to symbolically compute the second order properties discussed in Section 2 of a given surface. Bivariate vector fields represented as NURBs are derived whenever possible so that the method can take advantage of the computational characteristics of NURBs.

4.1 Surface Trichotomy

Use of the curvature trichotomy of a surface can result in a more optimal freeform surface milling process. Only convex regions (see Figure 1) are millable using flat end cutters and five axis milling. Flat end cutters, as oppose to ball end cutters, can mill faster and remove more material per unit time. Furthermore, the surface finish of flat end cutters is usually better. Using the trichotomy operator, convex regions within surfaces can be detected and milled in more efficient way and with a better finish. The Utah teapot in Figure 8 is mostly convex (red regions) and therefore this particular model may be milled mostly using a flat end cutter.

The determinant of L, $|L|$, in (7) is the key to this second order surface analysis. If $|L| = 0$, one of the normal curvature extrema κ_n^i must be zero. Assuming the surface is curvature continuous, adjacent regions for which κ_n^i has a different sign must be separated by a curve, C_s, for which $|L| = 0$, that is, one of the $\kappa_n^i = 0$. Furthermore, if $|L| > 0$ at some point p on the surface F, the surface is either convex or concave at p, while if $|L| < 0$ the surface locally is a saddle. In order to compute a scalar field representing $|L|$ using (5), it is necessary to find a square root to compute $n(u, v)$, which cannot be represented, in general, as a polynomial or as a piecewise rational. However, by reordering the operations to use the unnormalized surface normal $\hat{n}(u, v)$ and noting $n(u, v)$ appears twice as a factor in each term of $|L|$, $|L|$ can be represented exactly as a rational function and with no square roots,

$$|L| = \frac{\hat{l}_{11}\hat{l}_{22} - \hat{l}_{21}\hat{l}_{12}}{\|\hat{n}\|^2}. \tag{14}$$

This equation is representable as a NURBs using only operations from Section 3. $\hat{n}$ is a cross product of two surface partials $\frac{\partial F}{\partial u}$ and $\frac{\partial F}{\partial v}$. The components of L, $\hat{l}_{ij}$, are inner products of $\hat{n}$ with second order partials of F. Since only the zero set is of interest, and F is assumed to be a regular surface, it is necessary to examine only the numerator of (14). Once the zero set of $|L|$ has been computed, trimmed surfaces are created, each of which is completely convex, concave or saddle. The sign of $|L|$ at a single point on each trimmed surface is then used to classify the saddle regions while convex and concave regions are distinguished from each other by simply evaluating the sign of $\hat{l}_{11}$, for example, at that single point. While the saddle region is an intrinsic surface characteristic, the convex/concave classification is parameterization dependent. Flipping the u or v (but not both) surface parameterization direction will flip the normal direction $n(u, v)$ and therefore the sign of $\hat{l}_{11}$.

Figures 3 through 7 show some examples. Figure 3 is a biquadratic B-spline surface with three internal knots in each direction (patches of a B-spline surface are counted as how many Bézier patches would result from subdividing the NURBs surface at each interior knot, so this surface yields 16 polynomial patches), while Figure 4 is a single biquadratic patch. The bicubic surfaces in Figures 5 and 6 have two internal knots in each direction, yielding 9 polynomial patches. Figure 7 top is a bicubic NURBs surface with a single internal knot in each direction, yielding four Bézier patches. All Figures have been colored consistently, with yellow marking the saddle regions, red representing a convex region and green representing a concave region.

The biquadratic surface of Figure 3 is not C^2 along each internal knot, and the surface trichotomy is isoparametric along the internal knots lines.

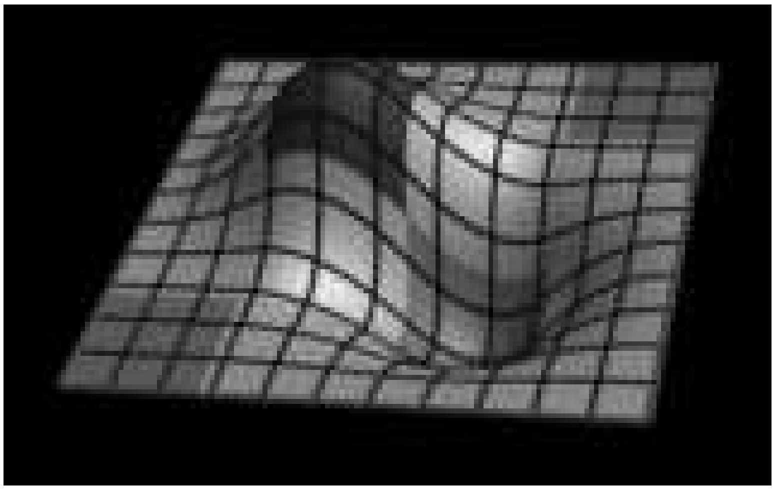

Figure 3: Biquadratic surface tricho¡tomy with 16 polynomial patches.

However, in general, this behavior should not be expected, or even anticipated, for biquadratic surfaces, since even a single biquadratic patch may contain both convex and saddle regions simultaneously as shown in Figure 4.

The surface in Figure 5 uses the same control mesh as the one in Figure 3 but is bicubic. Both surfaces in Figure 3 and Figure 5 uses appropriate uniform open end condition knot vectors. A comparison of these two Figures graphically demonstrates the influence of the *order* of the tensor product spline surface on the shape, as shown by comparing the shapes and locations of the convex and concave regions. This phenomena is somewhat counterintuitive to the common belief that two NURBs surfaces with the same mesh but different order are very similar, except that the one with higher order is a smoother version. The curvature characteristics have actually been changed. Figure 3 has one concave region, one convex region and two flat regions, all of which have isoparametric boundaries. Figure 5, however, has only one concave region and one convex region. The union of the two regions have a figure eight boundary, where convex and concave change at a single point. The curved boundaries of those regions are different than the straight line boundaries in Figure 3.

Figure 6 shows that the combination of symbolic computation (of $|L|$ as a scalar field) with numeric analysis (zero set contouring) can detect widely separated and isolated regions. In addition, it demonstrates the robustness of this methodology by accurately detecting two very shallow concave regions in the middle of the surface. In Figures 5 and 7, another ill conditioned case is shown in which several convex and concave regions meet at a single point. Since trimmed surfaces are formed, it was necessary that the boundaries be completely and correctly defined. The points where the three regions meet are correctly detected and determined and the topology of the regions is correctly maintained, which also demonstrates another type of robustness.

To provide a better sense of the process, the bottom of Figure 7 also shows the scalar field surface representation for the determinant of the second fundamental form, $|L|$, with

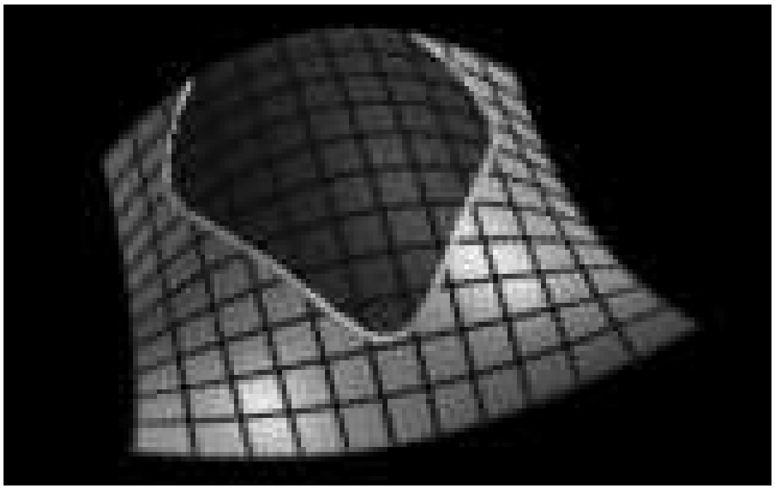

Figure 4: Biquadratic polynomial trichotomy.

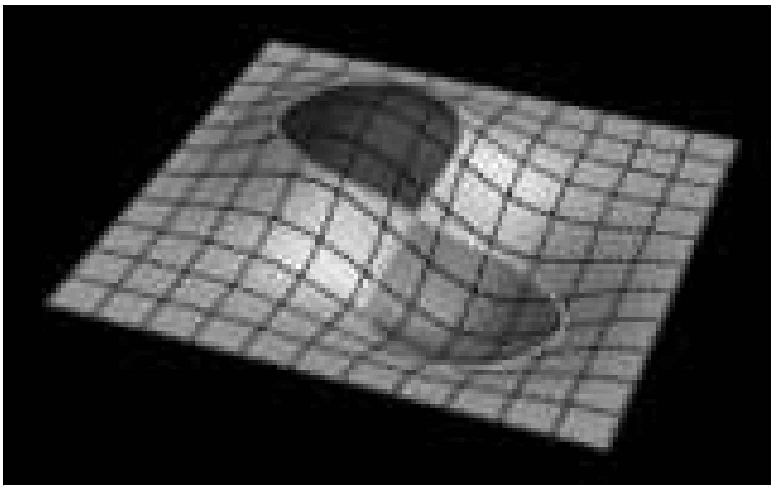

Figure 5: Bicubic surface trichotomy: same control mesh as Figure 3.

its zero set, as a function of u and v.

Finally, Figure 8 demonstrates this method on a more realistic object. The Utah teapot trichotomy degenerated into a dichotomy since no concave regions exist in the teapot model.

4.2 Bounding the Curvature

The extrema of the surface curvature are important for analyzing the curvature of a given surface. Normal curvature extrema occur in the principal directions [16, 20, 24], but the direct application of quadratic equation solution for (7) would require finding a square root. However, since the surface has been subdivided into convex, concave, and saddle regions, each region carries the following property:

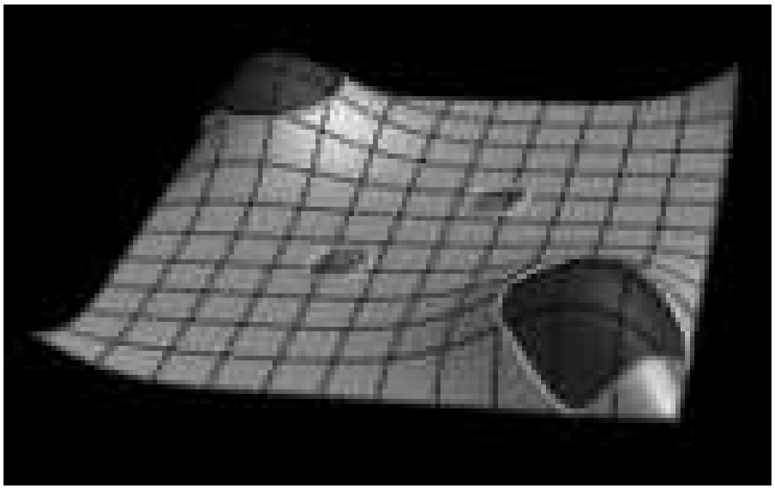

Figure 6: Bicubic with isolated convex and concave regions surrounded by saddle region.

- If the region has a saddle shape, then one of the principal curvatures, κ_n^1, is positive while the other, κ_n^2, is negative.
- If the region is convex both principal curvatures are negative.
- If the region is concave both principal curvatures are positive.

Using quadratic equation properties from (7), it can easily be shown that:

$$\begin{aligned} \psi &= (2H)^2 \\ &= \left(\kappa_n^1 + \kappa_n^2\right)^2 \\ &= \left(-\frac{b}{a}\right)^2 \\ &= \left(-\frac{g_{11}l_{22} + l_{11}g_{22} - 2g_{12}l_{12}}{|G|}\right)^2 \\ &= \frac{\left(g_{11}\hat{l}_{22} + \hat{l}_{11}g_{22} - 2g_{12}\hat{l}_{12}\right)^2}{|G|^2 \, \|\hat{n}\|^2} \end{aligned} \tag{15}$$

and

$$\begin{aligned} \phi &= (\kappa_n^1 - \kappa_n^2)^2 = \frac{b^2 - 4ac}{a^2} \\ &= \frac{(g_{11}l_{22} + l_{11}g_{22} - 2g_{12}l_{12})^2 - 4\,|G|\,|L|}{|G|^2} \\ &= \frac{(g_{11}\hat{l}_{22} + \hat{l}_{11}g_{22} - 2g_{12}\hat{l}_{12})^2 - 4\,|G|\,\left|\hat{L}\right|}{|G|^2 \, \|\hat{n}\|^2}. \end{aligned} \tag{16}$$

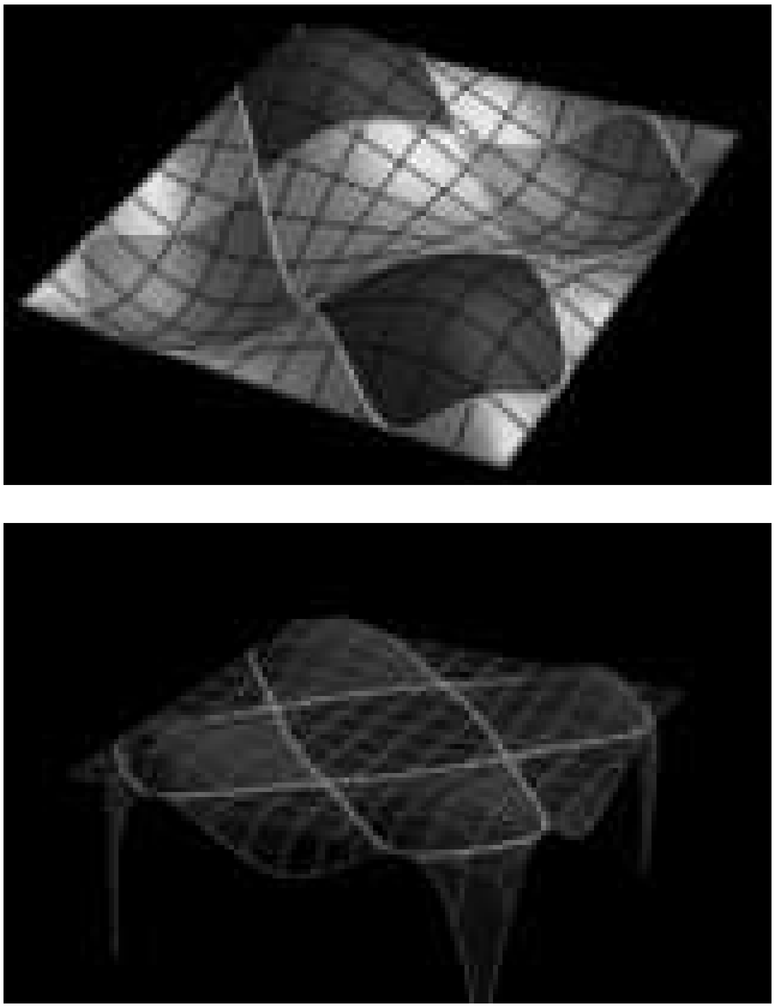

Figure 7: Bicubic surface with convex and concave regions meeting at a single point (top), and the scalar field of its second fundamental form with its zero set (bottom).

Both (15) and (16) can be represented without square roots and are therefore representable as NURBs using the model and tools defined in Section 3.

By characterizing the scalar field $\psi(u,v) = (\kappa_n^1(u,v) + \kappa_n^2(u,v))^2$, curvature estimates for the convex and concave regions can be determined so the computed curvature will be at most twice as large as the real normal curvature in the extreme case where both $\kappa_n^1(u,v)$ and $\kappa_n^2(u,v)$ are equal. One can obtain similar bounds by using $\phi = (\kappa_n^1(u,v) - \kappa_n^2(u,v))^2$ as the curvature estimator for saddle regions.

$\psi(u,v)$ and $\phi(u,v)$ can be used as curvature estimates for the appropriate trimmed regions and can be contoured to isolate regions with curvature larger than some allowable threshold. Furthermore, one can use (the square root of) $\psi(u,v)$ and $\phi(u,v)$ as pseudo color values to render the input surface $F(u,v)$ according to its curvature and provide visual feedback on which regions are highly curved. In other words, make the color of

Figure 8: Teapot trichotomy is degenerated into a ditochomy - no concave regions exist.

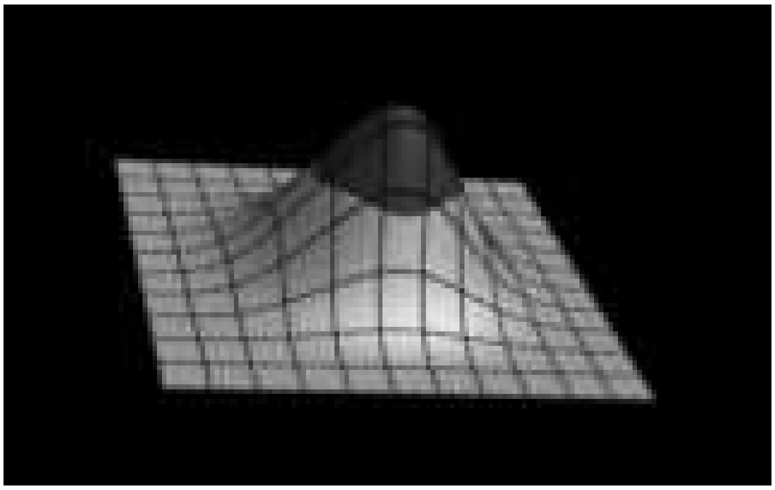

Figure 9: Surface dichotomy - saddle and convex regions.

$F(u, v)$ at the parameter value (u, v) depend on the value of the square root of $\psi(u, v)$ in convex and concave regions, and on the value of the square root of $\phi(u, v)$ in saddle regions. Using this technique, one can enhance the display of regions with high curvature, low curvature, or within certain bands of curvatures. Figures 9 through 11 demonstrate this. In Figure 9, the surface has been first subdivided into a saddle region (yellow) and a convex region (red). $\psi(u, v)$ has been used as the pseudo color in the convex region of the surface while $\phi(u, v)$ has been used for the same purpose in the saddle region, to render the image in Figure 11. Figure 10 shows $\psi(u, v)$ and $\phi(u, v)$. Not surprisingly, $\psi(u, v)$ is wider in the highly curved convex region since the two principal curvatures cancel each other in $\phi(u, v)$.

A different approach can be used to achieve a better bound. By expanding ϕ,

$$\phi \;=\; (\kappa_n^1 - \kappa_n^2)^2$$

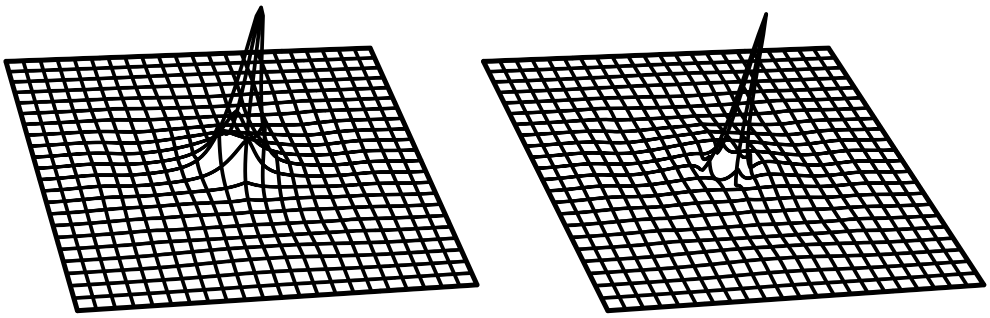

Figure 10: $\psi(u,v)$ (left), $\phi(u,v)$ (right), for the surface in Figure 9.

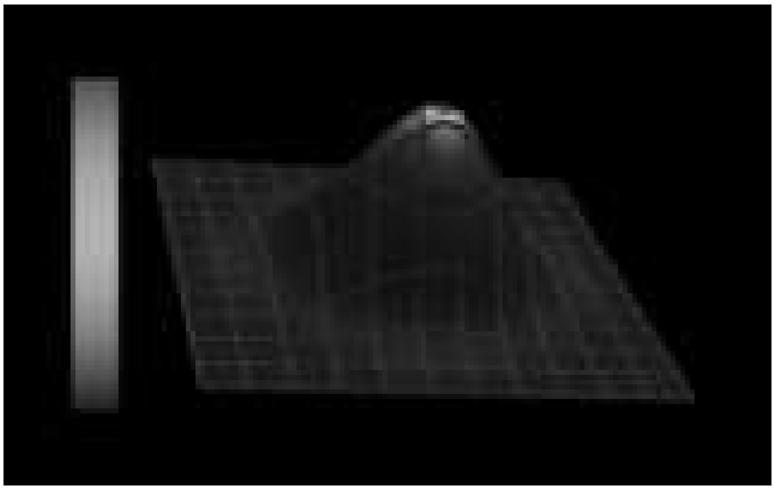

Figure 11: Curvature estimate using surface dichotomy, for the surface in Figure 9.

$$= \left(\kappa_n^1\right)^2 - 2\kappa_n^1\kappa_n^2 + \left(\kappa_n^2\right)^2. \tag{17}$$

Or

$$\begin{aligned}
\xi &= (\kappa_n^1)^2 + (\kappa_n^2)^2 \\
&= \phi + 2\kappa_n^1\kappa_n^2 \\
&= \phi + 2K \\
&= \phi + 2\frac{|L|}{|G|} \\
&= \frac{(g_{11}\hat{l}_{22} + \hat{l}_{11}g_{22} - 2g_{12}\hat{l}_{12})^2 - 4\,|G|\left|\hat{L}\right|}{|G|^2\,\|\hat{n}\|^2} + 2\frac{|L|}{|G|} \\
&= \frac{(g_{11}\hat{l}_{22} + \hat{l}_{11}g_{22} - 2g_{12}\hat{l}_{12})^2 - 4\,|G|\left|\hat{L}\right|}{|G|^2\,\|\hat{n}\|^2} + 2\frac{\left|\hat{L}\right|}{|G|\,\|\hat{n}\|^2}
\end{aligned}$$

Figure 12: Utah teapot curvature estimation.

$$= \frac{(g_{11}\hat{l}_{22} + \hat{l}_{11}g_{22} - 2g_{12}\hat{l}_{12})^2 - 2\,|G|\left|\hat{L}\right|}{|G|^2 \,\|\hat{n}\|^2}. \tag{18}$$

$+\sqrt{\xi}$ is bounded to be at most $\sqrt{2}$ greater than the larger magnitude of the principal curvatures. This worst case occurs when the two principal directions have the same magnitudes. Furthermore, ξ can be represented using the tools described in Section 3. Figure 12 demonstrates this approach applied to the Utah teapot model. The use of ξ may help to isolate regions with low curvature, which can be milled using larger ball end tools in a more optimal way. Figure 13 shows such a surface subdivided in such regions. The curvature bound surface, $\xi(u, v)$, (Figure 14) of the surface in Figure 13 is being contoured and regions with different curvature bounds are formed. It is clear from Figure 13 that the blue regions can be milled using a very large ball end cutter, the green regions with a medium size cutter and only the yellow and red regions, which are less than 5% of the whole surface area, should be milled with a small size tool.

5 Conclusions

A method to partition a surface into three disjoint trimmed surfaces (convex, concave, and saddle) and to determine global bounds on surface curvatures, has been presented which combines symbolic and numeric methods. The hybrid method was found to be robust and fast. The computation involved in the creation of a derived vector field, that is exact to machine accuracy, usually takes less than a second for a Bézier surface on an SGI 240/GTX (25MHz R3000). This symbolic computation creates closed forms with complexity directly bounded by the surface orders and continuity (knot vectors). Given a surface to analyze, the evaluation of the derived scalar fields (equations (14), (15), (16), (18)) is performed using the symbolic operators defined in section 3. Contouring usually

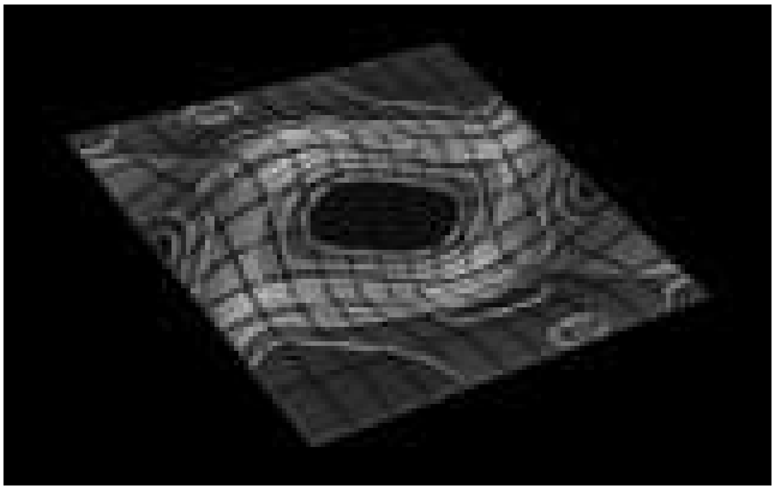

Figure 13: The surface is subdivided into regions with different curvature bounds.

takes an order of magnitude longer than the creation of the scalar fields. A scalar or a vector field is substituted into (13), forming a new nonrational scalar field, which is contoured using a subdivision based approach [11].

Since milling is several magnitudes slower than even the contouring process, and since the same toolpath may be used thousands of times, time is not a major factor in optimizing the milling process. The ability to isolate regions in a surface with specific curvature bounds makes it possible to mill the surface more optimally by using the largest possible tool for each region.

The orders of the resulting scalar and vector fields are high. A second fundamental form determinant scalar field for a bicubic B-spline surface has degree 14. The degree of the scalar fields $\psi(u, v)$, $\phi(u, v)$ and $\xi(u, v)$ is even higher, degree 30. However, because the evaluation of Bézier and B-spline representations is robust, the high order does not introduce any numerical problems [15]. However, the numeric contouring process becomes more time consuming, since the complexity of a single subdivision operation grow at least quadratically with the order of the bivariate scalar field.

The analysis demonstrated in this paper exercises a combination of a symbolic computation in which a derived scalar field is computed, and numerically analyzed by evaluation or contouring. Computation of the scalar fields is robust for Bézier surfaces since closed form formulations exist for all operators. The use of interpolation to compute products of NURBs was found to be unstable for higher orders (> 10) and therefore surfaces in such cases were split into Bézier patches. Since contouring methods for freeform surfaces are well known and are, in general, robust, the whole analysis was found to be very stable.

Derived vector fields can be used for other applications, and other second order surface properties can be similarly computed, such as geodesic, Gaussian, or mean curvatures. Furthermore, this methodology has been successfully used to solve other problems as well. The silhouette extraction algorithm presented in [9] has been enhanced to use the zero set (contour) of the z component of the normal vector field, $n_z(u, v)$, with more reliable,

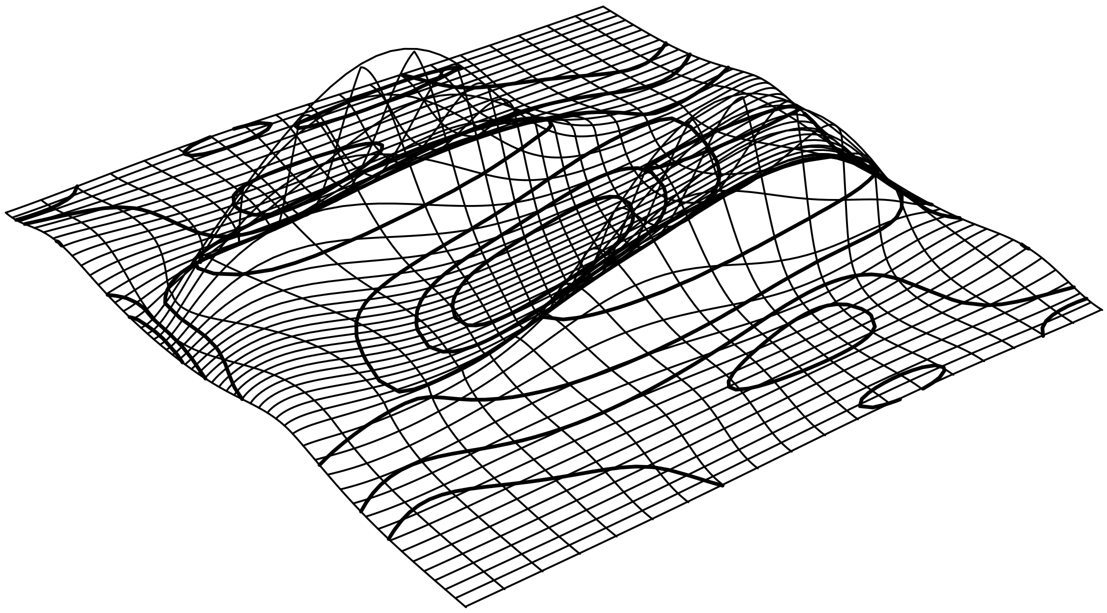

Figure 14: Curvature surface bound, ξ, of the surface in figure 13.

faster results. The error introduced by an offset approximation to a curve or a surface can be globally bounded and represented as an error curve or an error surface. Similarly it has been used to reduced the error of a curve or a surface offset approximation to a required tolerance [10].

The work presented here makes it practical to use second order surface analysis as a tool to support the development of robust, accurate, optimal algorithms for NC toolpath generation and to support alternative criteria for surface subdivision based on the second order properties of the shape. Consideration of Figures 3 and 5 shows another area of use. Users of NURBs are frequently unaware of the implications on the *shape* of the surface from using different orders. Manipulating the same control mesh can give different, unexpected, shapes depending on the order. The ability to accurately visualize second order properties in a reasonable time will enable better inspection and understanding of the effect of order, and potentially knot vector, changes. Furthermore, while NC verifications frequently simulate the tool path moving over the surface geometry, they do not check that a tool path for a convex region is actually cutting a convex region. The work presented here can be used in implementing that larger *visual process validation*. The viewer can use the understanding gained from exhibiting second order properties to take effective action.

6 Acknowledgment

The authors are grateful to the anonymous reviewers of this paper for their valuable remarks on the various drafts of this paper.

References

[1] R. E. Barnhill, G. Farin, L. Fayard and H. Hagen. Twists, Curvatures and Surface Interrogation. Computer Aided Design, vol. 20, no. 6, pp 341-346, July/August 1988.

[2] J. M. Beck, R. T. Farouki, and J. K. Hinds. Surface Analysis Methods. IEEE Computer Graphics and Applications, Vol. 6, No. 12, pp 18-36, December 1986.

[3] M. D. Carmo. Differential Geometry of Curves and Surfaces. Prentice-Hall 1976.

[4] M. S. Casale. Free-Form Solid Modeling with Trimmed Surface Patches. IEEE Computer Graphics and Applications, Vol. 7, No. 1, pp 33-43, January 1987.

[5] E. Cohen, T. Lyche, and L. Schumaker. Degree Raising for Splines. Journal of Approximation Theory, Vol 46, Feb. 1986.

[6] E. Cohen, T. Lyche, and L. Schumaker. Algorithms for Degree Raising for Splines. ACM Transactions on Graphics, Vol 4, No 3, pp.171-181, July 1986.

[7] E. Cohen, T. Lyche, and R. Riesenfeld. Discrete B-splines and subdivision Techniques in Computer-Aided Geometric Design and Computer Graphics. Computer Graphics and Image Processing, 14, 87-111 (1980).

[8] J. C. Dill. An Application of Color Graphics to the Display of Surface Curvature. SIGGRAPH 1981, pp 153-161.

[9] G. Elber and E. Cohen. Hidden Curve Removal for Free Form Surfaces. SIGGRAPH 90, pp 95-104.

[10] G. Elber and E. Cohen. Error Bounded Variable Distance Offset Operator for Free Form Curves and Surfaces. International Journal of Computational Geometry and Applications, Vol. 1., No. 1, pp 67-78, March 1991.

[11] G. Elber. Free Form Surface Analysis using a Hybrid of Symbolic and Numeric Computation. Ph.D. thesis, University of Utah, Computer Science Department, 1992.

[12] G. Farin. Curves and Surfaces for Computer Aided Geometric Design. Academic Press, Inc. Second Edition 1990.

[13] R. T. Farouki. The Approximation of Non-Degenerate Offset Surfaces Computer Aided Geometric Design, Vol. 3, No. 1, pp 15-43, May 1986.

[14] R. T. Farouki and V. T. Rajan. Algorithms For Polynomials In Bernstein Form. Computer Aided Geometric Design 5, pp 1-26, 1988.

[15] R. T. Farouki and V. T. Rajan. On The Numerical Condition of Polynomial in Bernstein Form. Computer Aided Geometric Design 4, pp 191-216, 1987.

[16] I. D. Faux and M. J. Pratt. Computational Geometry for Design and Manufacturing. John Wiley & Sons, 1979.

[17] A. R. Forrest. On the Rendering of Surfaces. SIGGRAPH 1979, pp 253-259.

[18] R. T. Lee and D. A. Fredericks. Intersection of parametric Surfaces and a Plane. IEEE Computer Graphics and Applications, Vol. 4, No. 8, pp 48-51, August 1984.

[19] T. McCollough. Support for Trimmed Surfaces. M.S. thesis, University of Utah, Computer Science Department, 1988.

[20] Millman and Parker. Elements of Differential Geometry. Prentice Hill Inc., 1977.

[21] K. Morken. Some Identities for Products and Degree Raising of Splines. To appear in the journal of Constructive Approximation.

[22] S. G. Satterfield and D. F. Rogers. A Procedure for Generating Contour Lines From a Bspline Surface. IEEE Computer Graphics and Applications, Vol. 5, No. 4, pp 71-75, April 1985.

[23] T. W. Sederberg and A. K. Zundel. Scan Line Display of Algebraic Surfaces. SIGGRAPH 1989, pp 147-156.

[24] J. J. Stoker. Differential Geometry. Wiley-Interscience 1969.

[25] D. J. Walton and D. S. Meek. Curvature Bounds For Planar B-spline Curve Segments. Computer Aided Design, vol. 20, no. 3, pp 146-150, April 1988.