

Tracing along an intersection closed loop, when should one stop?

WU, SHIN - TING AND OSMAR ALÉSSIO

Electrical and Computer Engineering Faculty, State University of Campinas -
P.O.Box 6101, 13083-970 - Campinas, SP, Brazil
ting,osmar@dca.fee.unicamp.br

Abstract. In this paper we present an algorithm for estimating the rotation index of a closed loop from the number of singular points on it. On the basis of this index we also devise a procedure that guarantees to have an arbitrary loop being traced only once. Combining this procedure with a higher-order stepping algorithm, a robust intersection technique is resulted.

1 Introduction

The surface-surface intersection is a fundamental problem in computational geometry and geometric modeling of complex shapes. For general parametric surface intersections, the most commonly used methods include marching and subdivision. Subdivision-based algorithms [3, 10] characteristically tessellate the surfaces into piecewise linear approximations and intersect the facets. The accuracy of these subdivision algorithms depends on how the flatness of the subpatches is defined. Small intersection loops or isolated points are very difficult, if not impossible, to find. Marching-based algorithms [1, 2] begin by finding a starting point on the intersection curve, and proceed to march along the curve. Because of the inherent geometric complexity of high degree algebraic curves that could yield from the intersection of two regular surfaces, marching along a branch or a closed loop is also a difficult problem. Such a curve may have singular points where the normal vectors of the intersecting surfaces are parallel and the numerical solution scheme may fail. Moreover, when a closed loop is traced, no sufficient condition is known for stopping the marching.

To our knowledge, a classical solution for handling closed loops on the parametric domain is to reduce them into open curves, by detecting them previously [12, 13] and subdividing the surface domain such that they are split into several branches belonging to distinct subpatches [6–8]. After then, these subpatches are dealt with individually. The loop detection algorithms are generally performed in two steps: closed loop existence test and splitting point determination. Normal criterion [13, 14] and distance criterion [6–9] are two most referenced criteria. The first one is based on the fact that when two surfaces intersect in closed loops, there are points in the both surfaces whose normal vectors are parallel. This way for loop detection is efficient and robust. Unfortunately, this is not a sufficient condition, since it is also fulfilled by contact areas. The second criterion explores the fact that at least one point inside a closed loop is a critical point of the oriented distance function of the

two intersecting surfaces. Detection on the basis of distance criterion, however, involves heavy computation and its robustness depends on how closely the critical points are located.

Wu and Andrade [17] propose an algorithm for estimating the curvature of the intersection curve. This curvature is applied in the computation of tracing step direction and size. In their work they show that one can trace along a curve with insignificant deviations, even when it contains singular points. The procedure only stops when a border of the parametric domain is reached or a singular point is met. To avoid “infinite iterations” as they go along a closed loop, they fixed in 400 a maximal number of traced points.

Later, Wu et al. [16] present a simple and robust algorithm for estimating the other local geometric properties of intersection curves, namely the binormal and torsion vectors at any point. In addition, two new marching directions that make use of these properties are given. When a singular point is met, they switched to the formulas derived by Ye and Maekawa [18] for computing them in an exact way. With regard to parametric domains, any branch or closed loop may be therefore traced out in one shot. This result motivates us to pursue a sufficient condition for stopping marching along a closed loop.

In this paper we present, without a rigorous proof, a set of conditions for stopping a “cyclic” traverse. Besides a proximity and a contact-order condition, we introduce rotation index condition to ensure a complete and non-overlapping traverse. Several examples are given to illustrate its efficiency and robustness. Section 2 provides the theoretical foundation. In Section 3 we present our proposal and in Section 4, an algorithm. Representative experimental results are included in Section 5 to demonstrate its validity. Finally, some concluding remarks are drawn in Section 6.

2 Background

Let $\alpha: [0, l] \rightarrow R^2$ be a plane closed curve given by $\alpha(s) = (x(s), y(s))$ where s is the arc length. Since s is the arc

length, the tangent vector $t(s) = (\dot{x}(s), \dot{y}(s))$ has unit length. The map $t : [0, l] \rightarrow R^2$ is a differentiable curve and is called *tangent indicatrix*. Its trace is contained in a circle of radius 1 (Figure 1).

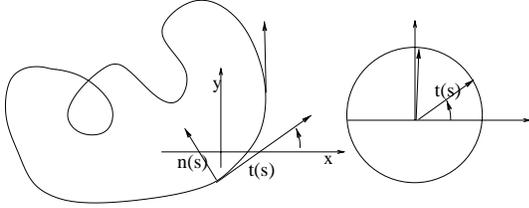


Figure 1: Tangent indicatrix.

Let $\theta(s)$, $0 \leq \theta(s) \leq 2\pi$, be the angle that $t(s)$ makes with the x axis; that is, $\dot{x}(s) = \cos \theta(s)$ and $\dot{y}(s) = \sin \theta(s)$. $\theta(l)$ measures the total angle described by the point $t(s)$ on the tangent indicatrix, as we run the curve α from 0 to l (Figure 2). Since α is closed, this angle is an integer multiple of 2π , that is $\theta(l) - \theta(0) = 2\pi I$.

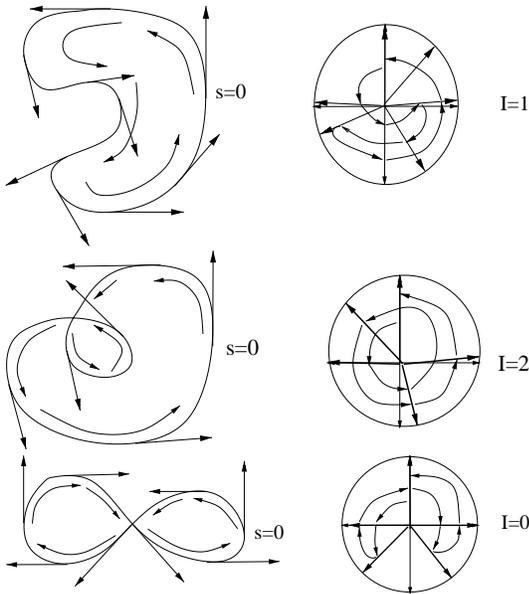


Figure 2: The rotation index.

The integer I is called the *rotation index* or *turning number* of the curve α . Intuitively, this number tells how many times the (oriented) tangent vector to a curve turns around as we follow it along the curve [4, 5, 15].

It is also proved that an arbitrary closed curve in R^2 is homotopic to one of the following:

1. a simple curve, if $I = \pm 1$ (Fig 3.a),
2. an eight-shaped curve, if $I = 0$ (Fig 3.b), and

3. a m -self-looped curve, if $|I| = m$ (Fig 3.c).

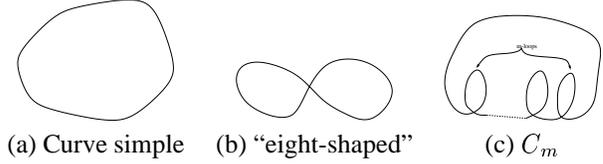


Figure 3: Three classes of curves.

Observe that the “eight-shaped” and the “ m -self-looped” curves contain *branch points*. To distinguish them, we classify these branch points into *eight-shaped (topological branch) points* and *self-looped (topological branch) points* to refer, respectively, the ones on the eight-shaped curves and the ones on the “self-looped” curves. The eight-shaped points separate “loops” with opposite orientation (Figure 4a); while the self-looped points connect “loops” with the same orientation [4] (Figure 4b).

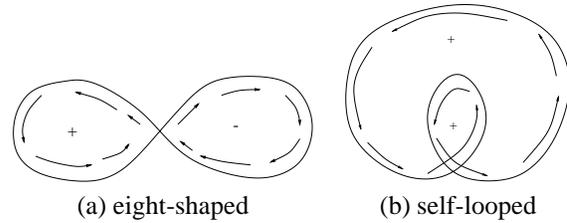


Figure 4: Orientation of loops.

Sederberg et al. [12] presented four equations that must be satisfied by a singular point on the intersection of two regular surfaces $F(u, v)$ e $G(s, w)$:

$$\begin{aligned}
 F_u \times F_v \cdot G_s &= 0 \\
 F_u \times F_v \cdot G_w &= 0 \\
 (F - G) \cdot F_u &= 0 \\
 (F - G) \cdot F_v &= 0
 \end{aligned} \tag{1}$$

Numerical Newton methods [11] may be applied for solving this equation system to obtain a singular point.

Ye and Maekawa [18] proposed an algorithm for computing the local properties of the intersection curve at non-transversal intersection points. Given two regular surfaces, $F(u, v)$ and $G(s, w)$, and their second fundamental form coefficients (L^F, M^F, N^F) and (L^G, M^G, N^G) , respectively. They showed that the tangent vector of an intersection point may be a function of $w = \frac{u'}{v'}$, if $b_{11} \neq 0$, or $w = \frac{v'}{u'}$, when $b_{11} = 0$ and $b_{22} \neq 0$. The unknowns u' and v' must satisfy the expression

$$b_{11}(u')^2 + 2b_{12}(u')(v') + b_{22}(v')^2 = 0, \tag{2}$$

where

$$\begin{aligned}
b_{11} &= a_{11}^2 L^G + 2a_{11}a_{21}M^G + a_{21}^2 N^G - L^F, \\
b_{12} &= a_{11}a_{12}L^G + 2(a_{11}a_{22} + a_{21}a_{12})M^G + \\
&\quad a_{21}a_{22}N^G - M^F, \\
b_{22} &= a_{12}^2 L^G + 2a_{21}a_{22}M^G + a_{22}^2 N^G - N^F. \\
a_{11} &= \frac{(F_u \times G_w) \cdot N}{(G_s \times G_w) \cdot N} \\
a_{12} &= \frac{(F_v \times G_w) \cdot N}{(G_s \times G_w) \cdot N} \\
a_{21} &= \frac{(G_s \times F_u) \cdot N}{(G_s \times G_w) \cdot N} \\
a_{22} &= \frac{(G_s \times F_v) \cdot N}{(G_s \times G_w) \cdot N}
\end{aligned} \tag{3}$$

Moreover, they noted that there are four distinct cases to the solution of Eq. (2) depending on the discriminant $b_{12}^2 - b_{11}b_{22}$:

1. $b_{12}^2 - b_{11}b_{22} < 0$: the point is an isolated tangential contact point of F and G .
2. $b_{12}^2 - b_{11}b_{22} = 0$: the point is a tangential point, in the sense that F and G intersect at the point and at its neighborhood.
3. $b_{12}^2 - b_{11}b_{22} > 0$: the point is a branch point of the intersection curve – the curve crosses on this point.
4. $b_{11} = b_{12} = b_{22} = 0$: the point is a higher-order contact point.

3 Our Proposal

By inspection, we set the following three conditions for ensuring that any closed curve is traced once as we run from the point P_0 to the point P_n :

Proximity condition: P_0 and P_n must be “sufficiently” close.

Contact-order condition: The differential geometric properties of the intersecting surfaces must be “almost” coincident at P_0 and P_n .

Rotation index condition: The rotation index of a (plane) closed loop is an integer n , that is, the sum of its tangent vector changes must be “almost” $2n\pi$.

The first condition is a trivial necessary condition for arbitrary closed curve. However, there is a variety of closed curves containing points that satisfy this condition but are not the extreme points of the sequence (Figure 5). To put aside these points, we introduced the contact-order condition, which intuitively tells us that no abrupt change in the

differential geometric properties may occur when we run along an intersection curve. Even so, there are curves, such as self-intersecting curves, which possess points that fulfill both conditions but should not be considered the extreme points of the tracing trajectory (Figure 5b).

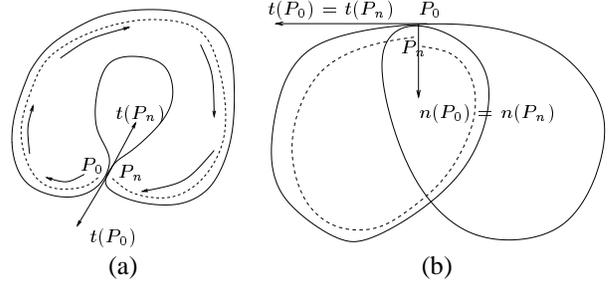


Figure 5: Curves for which (a) the proximity condition and (b) the proximity and contact conditions are fulfilled.

The three conditions are, unfortunately, interlocked. The rotation index n is important for evaluating whether the curve is a candidate for having closed loops – the exact sum of its tangent vector changes must be $2n\pi$. However, n is a global topological property, in the sense that one can only derive n when the curve is completely traversed. Theoretically, if we do not know when we return to the first point of a closed curve during traversing, it is hard to estimate the number n . And consequently, we do not have enough information for evaluating whether the marching process should be stopped or not.

Fortunately, we know that any curve is homotopic to one of the curves in Figure 3 and the rotation index of each one is directly related with the number of *branch point* [4]. Hence, if we know the homotopic curve of an arbitrary curve, we may easily estimate its rotation index by counting the number of eight-shaped and self-looped points.

In the practice, homotopic curves may differ widely each other from the geometric point-of-view. Figure 6 illustrates three homotopic curves with the rotation index equal to 3. It is worth remarking that only two of the six branch points in the curve shown in Figure 6b are topological branch points. The others may disappear after deforming “smoothly” and conveniently the curve.

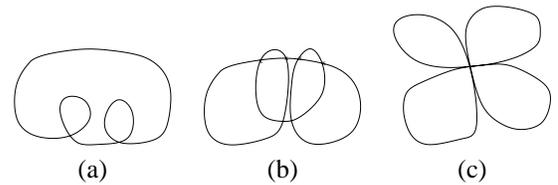


Figure 6: Homotopic curves.

In addition, Figure 7a shows how we may deform “smoothly” a curve with one singular point and one branch point into a curve with only one (topological) branch point. Figure 7b shows how we may deform “smoothly” a curve with one singular point into a simple curve. A solution we adopted is to “classify” the singular points with the use of the discriminant $b_{12}^2 - b_{11}b_{22}$ of Eq. (2) to discard trivially the tangential points.

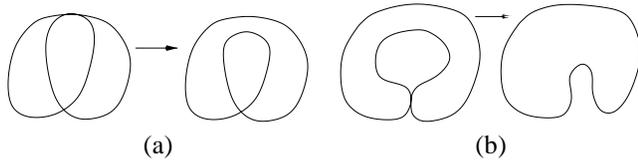


Figure 7: Singular and branch points.

To make the problem harder, we may have curves that are visually similar, but not homotopic each to other (Figure 8). Among the branch points we must, besides distinguishing the topological branch points from the non-topological ones, tell two kinds of topological branch points: the ones in whose neighborhood the tangent vectors of the intersection curve changes their sign (Figure 8a) and the ones in whose neighborhood the sign of the tangent vectors is preserved.

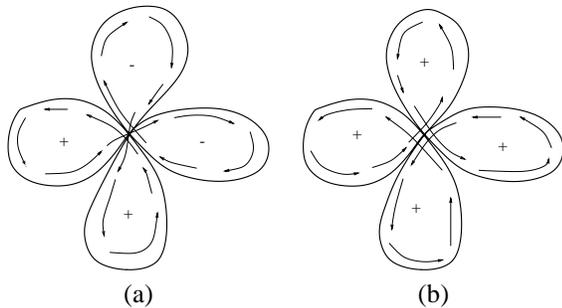


Figure 8: Curves homotopic to (a) a 1-eight-shaped curve and (b) a 2-self-looped curve.

In the next section we present an algorithm that may deliver the correct rotation index for any closed curve to be traced and decide correctly the instant for stopping the tracing.

4 Algorithm

With the marching algorithm presented by Wu et al. [16] one may traverse an arbitrary intersection curve in one shot until the border of the parametric domain of intersecting surfaces is reached. On the one hand, the procedure does not suffer from the “singularity” restrictions; on the other hand, when it runs along a closed loop, it will fall in infinite

iterations. Our goal is to devise a procedure that may avoid it.

According to the three conditions established in Section 3, the proximity condition for any closed loop α is easily expressed by

$$\|P_0 P_n\| < \epsilon$$

and the contact order conditions may be translated into a set of equations, such as

$$\cos \theta = t(P_0) \cdot t(P_n) < 1 - \epsilon \quad (4)$$

$$\cos \beta = n(P_0) \cdot n(P_n) < 1 - \epsilon \quad (5)$$

$$\cos \gamma = b(P_0) \cdot b(P_n) < 1 - \epsilon,$$

where t , n , and b denote, respectively, the normalized tangent, normal, and binormal vector of the intersection curve at a point. From our exhaustive tests, we realize that, except at some tangential points (Figure 9), only the conditions expressed by Eqs. (4) and (5) are sufficient. Moreover, knowing that the tangent vectors at P_n must be almost collinear to the tangent vector at P_0 , we simplified the computation by taking the vector $P_i - P_0$ at each traced point i and compared it with $t(P_0)$.

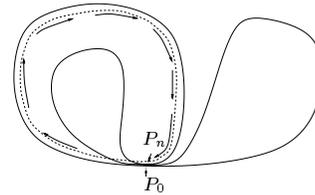


Figure 9: A tangential point where higher contact-order conditions must be verified.

The main problem that we faced was to estimate correctly for an arbitrary curve its rotation index.

As already mentioned, we may easily estimate the rotation index by counting the number of eight-shaped and self-looped points of its homotopic curve – one of the curves depicted in Figure 3. We also know that each branch point corresponds necessarily to a singular point, but a singular point does not imply a topological branch point.

Considering that the clockwise orientation of a closed curve is positive, we implemented the following procedure for tracing and registering the crossed singular points, from which we believe that one may easily estimate the rotation index of the intersection closed loop.

- Set the rotation index $m = 0$.
- Initialize three stacks: one stack for piling the branch points that the tracing algorithm meets along its trajectory for the first time; the second for piling the “unpaired” branch points that were popped out from the

first pile; and the third for piling the “pairwise” branch points that were popped out from the first one.

- For each traced point, we test whether the normal vectors of the two intersecting surfaces are almost parallel. If it is the case, we apply the condition expressed by Eq. (1) to obtain the singular point. We used the last traced point P_{i+1} in the region where those normal vectors are almost parallel or where the vector $N_F(P_i) \times N_G(P_i)$ is not in the same direction $N_F(P_{i+1}) \times N_G(P_{i+1})$ (Figure 10).

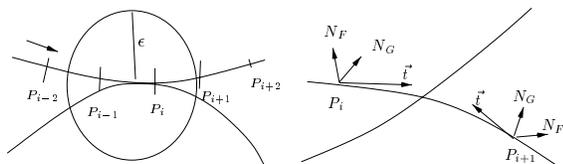


Figure 10: The guess point for obtaining a singular point.

- For each singular point, we applied the technique proposed by Ye and Maekawa [18] for identifying this point as a tangential or a branch one.
- Whenever a branch point is met, we check whether it is
 1. in the first stack: we move it to the third stack and every branch point piled over it are transferred to the second stack. We also compute the orientation of the closing loop and increment m by 1, if it is positive; otherwise, we decrement m by 1.
 2. in the second stack: the point is ignored.
 3. in the third stack: we verify whether the sign of the tangent vector changes. If it changes, we compute the orientation of the closing loop and increment m by 1, if it is positive; otherwise, we decrement it by 1. If it does not change, we only increment m by 1 when the point is crossed more than twice.
 4. is not found in the three stacks: we push it into the first stack.
- Whenever the proximity and tangent vector ($P_i \vec{P}_0$) conditions are fulfilled by a point P_i , we distinguish two situations:
 - 1) P_0 is not a tangential point: we compute the orientation of the closing loop and increment m by 1, if the orientation is positive; otherwise, we decrement m by 1.
 - 2) P_0 is a tangential point: we must check whether higher contact-order conditions are also satisfied. If it is the case, the same procedure for non-tangential point is applied. Otherwise, we keep on tracing.

Knowing how to estimate the rotation index of a closed loop from the branch points that we met as we run along a curve, we can decide for stopping our marching at a point P as follows. Consider that we register the tangent vector changes at each traced point, we only need to compare the sum of tangent vector angle changes with $2\pi m$, where m is the estimated rotation index. We consider that the closed curve is completely traced, when these values are approximately equal.

5 Experiments

We test exhaustively our algorithm on a variety of pairs of parametric surfaces in order to validate it experimentally. In this section we present some of them. To show numerically the behavior of our algorithm we also include for each example a table containing the first and the last traced points and all the singular points that were met with their geometric properties, namely distance, vector to the first point P_0 , and the total variation on the tangent vector angles.

The first pair of surfaces

$$S_{11}(u, v) = (u, v, 0.2 * u^4 + 0.1 * v^4)$$

$$S_{12}(s, w) = (s, w, 0.3 * s^2 * w - 0.1 * w^2 + 0.2 * w^3)$$

results in two singular points, although the intersection curve is homotopic to a 1-self-looped curve (Figure 11).

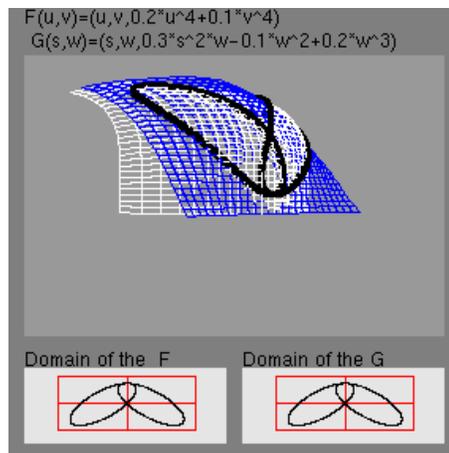


Figure 11: Intersection S_{11} with S_{12} .

It starts at the point $(-0.01, 0.00)$, which is very close to a singular point $(0.0, 0.0)$. The branch point $(0.0, 1.0)$ and the singular point were passed twice while the curve is traced. Note in Table 1 that at the singular point, the proximity and contact-order conditions are satisfied, but the rotation index condition is not. Hence, the tracing is kept on until the point $(-0.01, 0.00)$ is reached, at which the three conditions are satisfied.

The second pair of surfaces

Start Point domain $(-0.01,0.00) \vec{p}=(1.00,-0.02)$				
Point	Distance	$P_n P_0$	Total angle	index
$(-0.01,0.00)$	0.0000	$(1.00,-0.02)$	0.0000	0
...	0
$(0.00,0.00)$	0.0992	$(1.00,-0.01)$	—	0
...	0
$(0.00,1.00)$	0.9973	$(0.01,1.00)$	4.2079	0
...	0
$(0.00,0.00)$	0.0047	$(1.00,-0.01)$	6.2841	0
...	0
$(0.00,1.00)$	0.9985	$(0.01,1.00)$	8.3949	1
...	1
$(-0.01,0.00)$	0.0001	$(1.00,-0.02)$	12.5665	2

Table 1: Representative Points of $S_{11} - S_{12} = 0$.

$$S_{21}(u, v) = (u, v, (u^2 + v^2)^2 + 3 * u^2 * v - v^3)$$

$$S_{22}(s, w) = (s, w, 0)$$

results in a 1-self-looped curve (Figure 12).

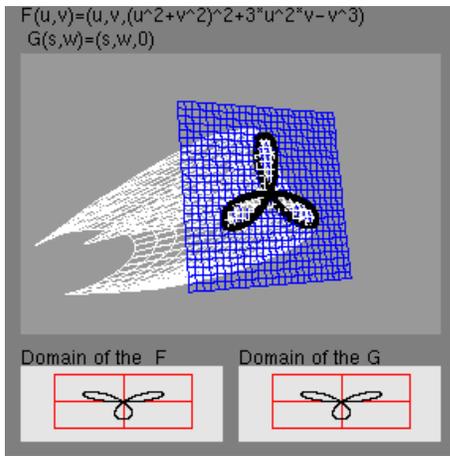


Figure 12: Intersection S_{21} and S_{22} .

Start Point domain $(0.14,0.88) \vec{p}=(-0.30,0.95)$				
Point	Distance	$P_n P_0$	Total angle	index
$(0.14,0.88)$	0.0000	$(-0.30,0.95)$	0.2096	0
...	0
$(0.00,0.00)$	0.7991	$(-0.22,-0.98)$	3.4514	0
...	0
$(0.00,0.00)$	0.8031	$(-0.21,-0.98)$	7.6432	1
...	1
$(0.00,0.00)$	0.8002	$(-0.21,-0.98)$	11.8350	1
...	1
$(0.15,0.86)$	0.0731	$(-0.28,0.96)$	12.7120	2

Table 2: Representative Points of $S_{21} - S_{22} = 0$.

It starts at the point $(0.17, 0.79)$. Although the branch

point $(0.0, 0.0)$ is passed three times while the curve is traced, the estimated rotation index is 2 (Table 2). It is because that the sign of the tangent vectors does not change in the neighborhood of this point. Hence, the tracing is kept on until the point $(0.15, 0.86)$ is reached, at which the three conditions are satisfied.

The third pair of surfaces

$$S_{31}(u, v) = (u, v, (u^2 + v^2)/3 - 9)$$

$$S_{32}(s, w) = (s, w, \sin(w) - 3)$$

results in one intersecting curve, which is homotopic to a simple curve (Figure 13).

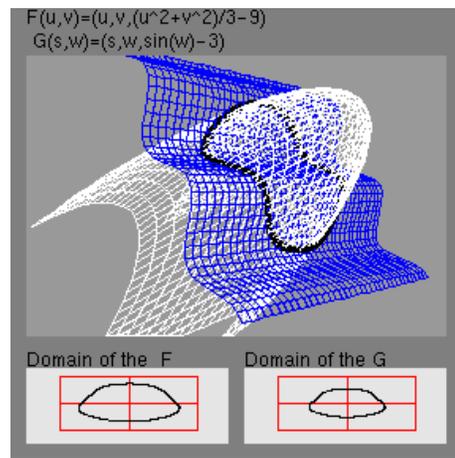


Figure 13: Intersection S_{31} and S_{32} .

As we ran along the curve from the point $(4.42, 0.82)$, we stopped at $(4.42, 0.87)$, when the three conditions are satisfied. Table 3 shows that no singular point was crossed.

Start Point domain $(4.42,0.82) \vec{p}=(0.05,1.00)$				
Point	Distance	$P_n P_0$	Total angle	index
$(4.42,0.82)$	0.0000	$(0.05,1.00)$	0.0000	0
...	0
...	0
$(4.42,0.87)$	0.0570	$(0.03,1.00)$	6.3102	1

Table 3: Representative Points of $S_{31} - S_{32} = 0$.

The fourth pair of surfaces

$$S_{41}(u, v) = (u, v, -u^4 + v^2 - v^4)$$

$$S_{42}(s, w) = (s, w, 0)$$

results in two intersecting curves, each one is homotopic to a simple curve (Figure 14).

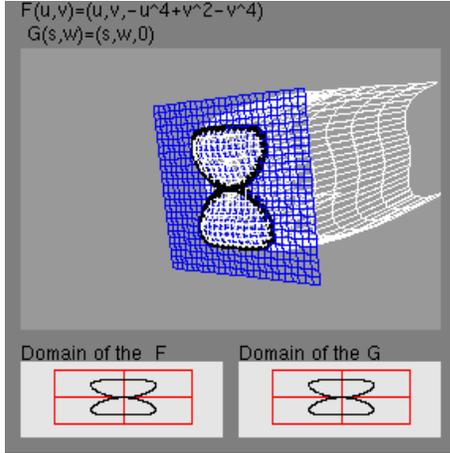


Figure 14: Intersection S_{41} and S_{42} .

Table 4 shows the extreme and singular points that we passed while a curve is traced. Note that the tangential point does not affect the marching behavior.

Start Point domain (0.71,0.74) $\vec{p}=(0.11,-0.99)$				
Point	Distance	$P_n \vec{P}_0$	Total angle	index
(0.71,0.74)	0.0000	(0.11,-0.99)	0.0000	0
...	0
(0.00,0.00)	1.0232	(-0.69,-0.73)	1.6758	0
...	0
(0.71,0.69)	0.0530	(0.03,-1.00)	6.4389	1

Table 4: Representative Points of $S_{41} - S_{42} = 0$.

The fifth pair of surfaces

$$S_{51}(u, v) = (u, v, (1/6) * u^6 - (5/4) * u^4 + 2 * u^2 + v^2)$$

$$S_{52}(s, w) = (s, w, 0.917)$$

results in a 2-eight-shaped curve (Figure 15).

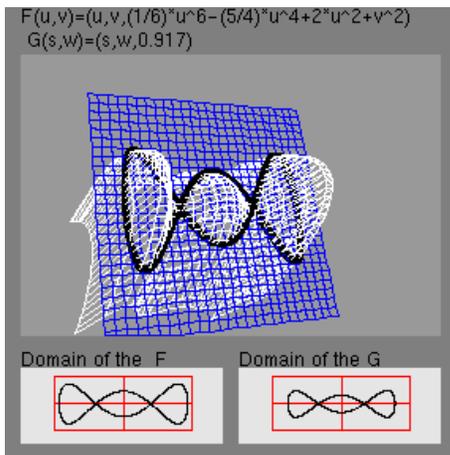


Figure 15: Intersection S_{51} and S_{52} .

Start Point domain (-1.65,1.18) $\vec{p}=(-0.54,0.84)$				
Point	Distance	$P_n \vec{P}_0$	Total angle	index
(-1.65,1.18)	0.0000	(-0.54,0.84)	0.0000	0
...	0
(-1.00,0.00)	1.3478	(0.49,-0.87)	5.0999	0
...	0
(1.00,0.00)	2.9040	(0.91,-0.41)	3.0849	0
...	0
(1.00,0.00)	2.9040	(0.91,-0.41)	8.1669	1
...	1
(-1.00,0.00)	1.3478	(0.49,-0.87)	6.1569	0
...	0
(-1.71,1.26)	0.0991	(-0.56,0.83)	6.1774	1

Table 5: Representative Points of $S_{51} - S_{52} = 0$.

Observe in Table 5 that the estimated rotation index is 1, as expected.

Finally, we present the intersection of

$$S_{61}(u, v) = (u, v, u^4 - u^2 + v^4 - v^2)$$

$$S_{62}(s, w) = (s, w, -0.25)$$

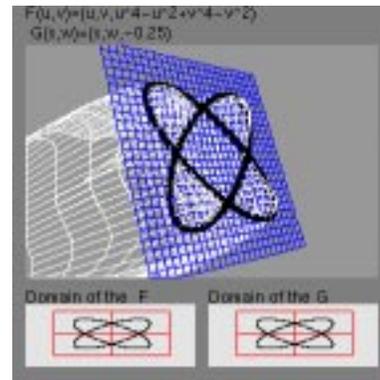


Figure 16: Intersection S_{61} and S_{62} .

Start Point domain (0.85,0.98) $\vec{p}=(-0.92,0.38)$				
Point	Distance	$P_n \vec{P}_0$	Total angle	index
(0.85,0.98)	0.0000	(-0.92,0.38)	0.0000	0
...	0
(0.00,0.71)	0.8863	(-0.95,-0.30)	1.0059	0
...	0
(-0.71,0.00)	1.8335	(-0.85,-0.53)	1.3215	0
...	0
(0.00,-0.71)	1.8830	(-0.45,-0.89)	4.1164	0
...	0
(0.71,0.00)	0.9852	(-0.14,-0.99)	4.4638	0
...	0
(0.82,0.99)	0.0312	(-0.94,0.33)	6.3909	1

Table 6: Representative Points of $S_{61} - S_{62} = 0$.

which results in two intersecting curves, each one is homotopic to a simple curve (Figure 16). As we ran along one of them starting from the point $(0.85, 0.98)$, we stopped at $(0.82, 0.99)$, when the three conditions are satisfied. Table 6 shows that during tracing four branching points were crossed. They, however, did not affect the rotation index, because the last point was reached before we over-traced them.

6 Concluding Remarks

We presented an algorithm for estimating the rotation index of a closed curve on a parametric domain on the basis of the singular points along the two intersecting surfaces. We applied it for validating the last point to be traced in a closed loop. Differently from the differential geometrical properties, the rotation index is a global geometrical property which “holds” the behavior of the traced curve. In this way, unnecessary point duplications are prevented.

Although we do not have rigorous proof that the procedure works for any case, we tested it for several situations and it could handle all of them correctly.

As further work we intend to demonstrate its validity or to establish precisely the scope where it is valid.

7 Acknowledgment

The second author would like to acknowledge the CNPq for financial support under the grant n^o 145149/1998-6.

References

- [1] C. L. Bajaj, C. M. Hoffmann, J.E.Hopcroft, and R.E.Lynch. Tracing surface intersections. *Computer Aided Geometric Design*, 5:285–307, 1988.
- [2] R. E. Barnhill, G. Farin, M. Jordan, and B.R.Piper. Surface/surface intersection. *Computer Aided Geometric Design*, 4(1/2):3–16, July 1987.
- [3] R. E. Barnhill and S. N. Kersey. A marching method for parametric surface/surface intersection. *Computer Aided Geometric Design*, 7(1–4):257–280, 1990.
- [4] M. Berger and B. Gostiaux. *Differential Geometry: Manifolds, Curves, and Surfaces*, volume 1. Springer-Verlag, 1988.
- [5] M. P. Carmo. *Differential geometry of curves and surfaces*. Prentice Hall Inc., New Jersey, 1st. edition, 1976.
- [6] N. M. Patrikalakis G. A. Kriezis and F-E. Wolter. Topological and differential equation methods for surface intersections. *Computer-Aided Design*, 24(1):41–55, 1992.
- [7] Yawei Ma and Yuan-Shian Lee. Detection of loops and singularities of surface intersections. *Computer-Aided Design*, 30(14):1059–1067, 1998.
- [8] Yawei Ma and Ren C. Luo. Topological method for loop detection of surface intersection problems. *Computer-Aided Design*, 27(11):811–820, 1995.
- [9] R. P. Markot and R. L. Magedson. Procedural method for evaluating the intersection curves of two parametric surfaces. *Computer-Aided Design*, 23(6):395–404, 1991.
- [10] G. Müllenheim. On determining start points for a surface/surface intersection algorithm. *Computer Aided Geometric Design*, 8:401–408, 1991.
- [11] William H Press and et al. *Numerical recipes in C: the art of scientific computing*. Cambridge University Press, USA, 2nd edition, 1992.
- [12] T. W. Sederberg, H. N. Christiansen, and S. Katz. Improved test for closed loops in surfaces intersections. *Computer Aided Design*, 21(8):505–508, october 1989.
- [13] T. W. Sederberg and R. J. Meyers. Loop detection in surface patch intersections. *Computer Aided Geometric Design*, 5:161–171, 1988.
- [14] P. Sinha, E. Klassen, and K. K. Wang. Exploiting topological and geometric properties for selective subdivision. In *Proc. ACM Symp. Computational Geometry*, pages 39–45, 1985.
- [15] D. J. Struik. *Lectures on Classical Differential Geometry*. Dover Publications, Inc., 2nd. edition, 1961.
- [16] S. T. Wu, O. Alessio, and S. I. R. Costa. On estimating local geometric propoerties of intersection curves. In *Proceedings of SIBGRAPI 2000*, pages 152–159, 2000.
- [17] S. T. Wu and L. N. de Andrade. Marching along a regular surface/surface intersection with circular steps. *Computer Aided Geometric Design*, 16:249–268, 1999.
- [18] X. Ye and T. Maekawa. Differential geometry of intersection curves of two surfaces. *Computer Aided Geometric Design*, 16:767–788, 1999.