

# Marching along a regular surface/surface intersection with circular steps

Wu, Shin-Ting<sup>1</sup> and Lenimar N. Andrade<sup>2</sup>

*UNICAMP - State University of Campinas*  
*Electrical and Computer Engineering Faculty*  
*Dept. Computer Engineering and Industrial Automation*  
*P.O.Box 6101*  
*13083-970 - Campinas, SP, Brazil*  
`ting,lenimar@dca.fee.unicamp.br`

---

## Abstract

This paper presents a simple and elegant algorithm to estimate adaptively the stepping direction and size for tracing a branch of the intersection curve between two regular surfaces without any nonlinear equation system solver [6,1,7]. The step is neither along the tangent vector at the current point [16] nor along a parabola in a vicinity of the current point [19]; it is along a circle at the current point. Although no curvature analysis or power series expansions about each point of the intersection curve were used in its construction, we demonstrate that our circle tends to the exact osculating circle, when the distance between two subsequent sampling points tends to zero. Through numerical examples, we also show that the performance of our algorithm by handling singular points, bifurcations, and points on the closely spaced branches, is equivalent to the ones based on embedding schemes [1,7].

*Keywords:* Surface intersection, regular surfaces, marching method, osculating circle, marching step, geometric modeling.

---

## 1 Introduction

The determination of the intersection between two surfaces is an important problem in geometric modeling.

---

<sup>1</sup> Corresponding author

<sup>2</sup> on leave of Department of Mathematics, Federal University of Paraíba (UFPB), during 1993-1997

Let

$$F(u, v) = (f_1(u, v), f_2(u, v), f_3(u, v))$$

and

$$G(r, s) = (g_1(r, s), g_2(r, s), g_3(r, s))$$

be two parametric surfaces. Then, their intersection corresponds to the solution of the non-linear system with 3 equations and 4 variables

$$\begin{cases} f_1(u, v) = g_1(r, s) \\ f_2(u, v) = g_2(r, s) \\ f_3(u, v) = g_3(r, s) \end{cases},$$

with the restrictions:  $u_{\min} \leq u \leq u_{\max}$ ,  $v_{\min} \leq v \leq v_{\max}$ ,  $r_{\min} \leq r \leq r_{\max}$ , and  $s_{\min} \leq s \leq s_{\max}$ .

This system can have no solution (if the surfaces do not intersect each other), only one solution (when the surfaces are tangent at one point) or even infinite solutions (that can be isolated points, a curve or overlapped region).

There exists a wide variety of methods for surface-surface intersection computation. In [8] they are classified into six categories: algebraic, subdivision, continuation, lattice, marching, and hybrid ones. Algebraic methods rely on the derivation of the equation of the intersection curve by substituting the parameters of a intersecting surface into the implicit form of the other. Subdivision techniques consist in decomposing recursively the surfaces to be intersected into simpler ones, which allow direct solution such as plane/plane intersection [9]. Continuation or homotopy algorithms are based on the idea of finding intersection through a system of differential equations which “embed” the equations of intersecting surfaces [1]. Lattice approaches reduce the dimensionality of surface intersections by discretizing one or both surfaces [16]. Marching schemes generate a sequence of intersection points by stepping from a given point in a direction that depends on the local differential geometry [4]. Finally, several algorithms combine two or more methods to take advantages of them [16,11,7].

To our knowledge, there is a consensus that the marching scheme is of great practical importance for intersection problems. This scheme comprises three primary phases [12,8]: hunting (start point), tracing, and sorting. The *hunting phase* provides starting point for stepping on the intersection curve. It should locate all branches of the intersection curve and prevent multiple copies of the same sequence of points during marching phase. Hodographs [18], subdivision techniques [17,14], and algebraic methods [1,7] have been applied for handling the hunting problem. The *marching phase* computes sequences of

points of an intersection curve branch by tracing out from the starting points. Incorrect step direction or size may lead to erroneous results. Most marching methods make use of curvature analysis or power series expansions about each point of the intersection curve to control the step. Tracing in the tangent direction [12,17], along a circle [2], and along a parabola [19] are some solutions presented in the literature and the curvature dependent [15,17,19] step size is the most used. Differential equation system [6,7] and continuation method [1] are also used to trace out a branch of the intersection curve. The *sorting phase* orders the sequences of points into meaningful branches of the intersection curve. When the points on the intersection curve can be found sequentially, this sorting is trivial.

In this paper we present yet another algorithm for the marching phase. Our algorithm has performance comparable to the tracing procedure presented in [19,1,7] and efficiency comparable to the methods given in [15,17]. Based on the same idea presented in [2], the next point is computed using a circle instead of parabola or tangent vector. We demonstrated that, despite the simplicity in its construction (no second and higher derivatives were employed for its determination), the proposed circle tends to the osculating circle. Consequently, it works well in the presence of strong changes in the curvature, closely spaced features, and bifurcations.

Section 2 the proposed circle and its implementation is presented. In Section 3 we show that the constructed circle is a good approximation to the osculating circle at each point. Next, in Section 4 comparisons with some representative existing algorithms are given. Finally, some concluding remarks are drawn in Section 5.

## 2 A circular step

In this section we present a new stepping algorithm on the basis of the osculating circle concept [13,10]. The osculating circle at a point  $P$  of a curve  $C$  is the best circle that approximates to a curve in a vicinity of  $P$ . Then, keeping on the osculating circle during marching may produce less deviation from the intersection curve. Consequently, less Newton iterations will be required to improve the accuracy of reached points at each step and the procedure may work well even in the presence of bifurcations and closely spaced branches. However, it is not a trivial task to calculate the exact osculating circle at a point  $P$  on the intersection curve because we do not know its equation a priori.

To overcome this problem without abandoning the idea of osculating circle, we propose an efficient algorithm to compute an approximate osculating circle at each current point with the use of previously traced intersection points. We

call the marching step on this approximate osculating circle the *circular step*.

### 2.1 Construction

Given two neighboring intersection points  $P$  and  $Q$ , whose tangent vectors are  $\vec{u}$  and  $\vec{v}$  respectively, an approximate osculating circle at  $Q$  is constructed as follows (Fig 1):

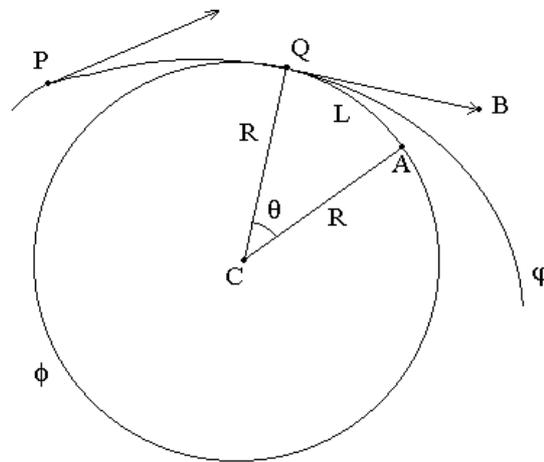


Fig. 1. A circular step

**Center (C):** the intersection of three planes: the plane that contains  $Q$  and has  $\vec{v}$  as normal vector; the plane that contains  $P$  and has  $\vec{u}$  as normal vector; and the plane that contains  $Q$  and has a normal  $\vec{u} \times \vec{v}$ .

**Radius (R):** the distance between  $C$  and  $Q$ .

**Note 1** For non-parallel tangent vectors  $\vec{u}$  and  $\vec{v}$ , the approximate osculating circle can always be constructed. If the tangent vectors are parallel, we may consider that  $P$  and  $Q$  are on the same segment and the radius of the osculating circle at  $Q$  tends to be infinity.

Having been determined the approximated osculating circle, the step size on this circle should be computed. To assure that our marching step adjusts automatically to the changes in the curvature of the intersection curve branch, the next approximated point to intersection curve  $A$  is calculated by incrementing  $L$  radians in central angle  $\widehat{QCA}$ , if the circle radius  $R$  is less than or equal to 1 length unit (Fig. 1). If  $R$  is greater than 1 length unit, then we increment  $L/R$  radians in the central angle. In other words, if the radius  $R$  is less than or equal to 1 length unit, the next point on the circle is chosen by stepping  $L \cdot R$  length units on the arc of circle from  $Q$ ; if the radius is greater than 1

length unit, then we trace  $L$  length units from  $Q$ . In both cases the central angle  $\widehat{QCA}$ ,  $\alpha$ , is equal to the length of the arc  $QA$  divided by the radius  $R$ .

## 2.2 Implementation

Only simple operations, namely product of  $3 \times 3$  matrices and resolution of  $3 \times 3$  linear system, are sufficient for the implementation of the idea presented in Section 2.1.

Let  $P = (p_1, p_2, p_3)$  and  $Q = (q_1, q_2, q_3)$  be two points on the intersection curve with their respective tangent vectors  $\vec{u} = (u_1, u_2, u_3) \neq \vec{0}$  and  $\vec{v} = (v_1, v_2, v_3) \neq \vec{0}$ . Let  $\vec{N} = (N_1, N_2, N_3) = \vec{u} \times \vec{v}$ . Then, the intersection of planes

- $\pi_1$ : plane that passes through point  $P$  and has  $\vec{u}$  as normal vector,
- $\pi_2$ : plane that passes through point  $Q$  and has  $\vec{v}$  as normal vector, and
- $\pi_3$ : plane that passes through point  $Q$  and has  $\vec{u} \times \vec{v}$  as normal vector,

is the center  $C$  of the approximate osculating circle at  $Q$ . Algebraically, this statement may be translated into the problem of solving the following linear system:

$$\begin{cases} u_1x + u_2y + u_3z = p_1u_1 + p_2u_2 + p_3u_3 \\ v_1x + v_2y + v_3z = q_1v_1 + q_2v_2 + q_3v_3 \\ N_1x + N_2y + N_3z = q_1N_1 + q_2N_2 + q_3N_3 \end{cases}, \quad (1)$$

which always has only one solution  $C$  if the vectors  $\vec{u}$  and  $\vec{v}$  are not parallel.

**Note 2** If  $\vec{u}$  and  $\vec{v}$  are parallel,  $\overline{PQ}$  is probably a rectilinear curve (curvature radius =  $\infty$ ). Therefore, the best choice for step vector is the tangent vector at  $Q$ ,  $\vec{u}$  or  $\vec{v}$ .

**Note 3** If  $Q$  is a singular point ( $\vec{v} = \vec{0}$ ), further tracing is not possible. A new initial point should be determined to trace the rest of the curve.

Knowing  $C$ , we can obtain the approximate next point  $A$

- (i) Obtain the normal vector  $n$  to the circumference plane through the expression:

$$\vec{n} = \overrightarrow{CP} \times \overrightarrow{CQ} = (n_1, n_2, n_3).$$

- (ii) Determine the reference system transformation such that after the transformation the osculating circle lies on the plane  $z = 0$  with its center at  $O = (0, 0, 0)$ . It consists of a translation

$$T = [-c_1 \quad -c_2 \quad -c_3 \quad 1]^T$$

followed by a rotation

$$R = \begin{bmatrix} \frac{\lambda}{V} & 0 & \frac{n_1}{V} \\ \frac{-n_1 n_2}{\lambda V} & \frac{n_3}{\lambda} & \frac{n_2}{\lambda} \\ \frac{-n_1 n_3}{\lambda V} & \frac{-n_2}{\lambda} & \frac{n_3}{V} \end{bmatrix},$$

where  $\lambda = \sqrt{n_2^2 + n_3^2}$  and  $V = \sqrt{n_1^2 + n_2^2 + n_3^2}$ .

(iii) Obtain

$$C' = CTR$$

$$P' = PTR$$

$$Q' = QTR$$

(iv) Determine the clockwise or counterclockwise orientation of  $\widehat{P'Q'}$  by observing the sign of the following mix product:

$$\vec{k} \cdot (\overrightarrow{OP'} \times \overrightarrow{OQ'}) = p'_1 q'_2 - p'_2 q'_1.$$

(v) Compute  $A' = (R \cos(\beta), R \sin(\beta), 0)$ , where

$$\beta = \begin{cases} \alpha + L, & \text{if } R \leq 1 \text{ and } p'_1 q'_2 - p'_2 q'_1 > 0 \\ \alpha - L, & \text{if } R \leq 1 \text{ and } p'_1 q'_2 - p'_2 q'_1 < 0 \\ \alpha + L/R, & \text{if } R > 1 \text{ and } p'_1 q'_2 - p'_2 q'_1 > 0 \\ \alpha - L/R, & \text{if } R > 1 \text{ and } p'_1 q'_2 - p'_2 q'_1 < 0 \end{cases}$$

with  $\alpha$  denoting the angle between the vectors  $\vec{i} = (1, 0, 0)$  and the vector  $\overrightarrow{OQ'}$  and  $L$ , the step angle (Section 2) (Fig 2).

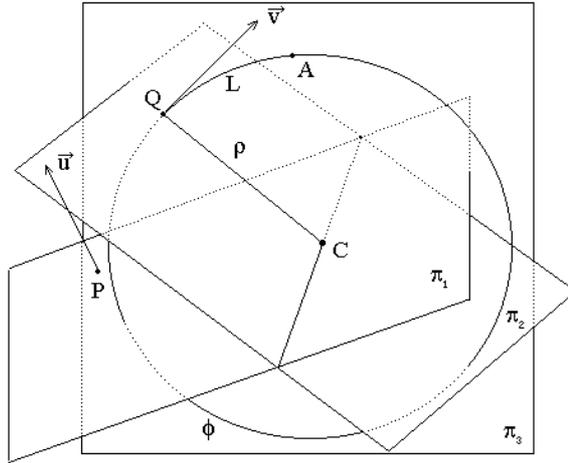


Fig. 2. An efficient implementation

**Note 4** To construct the approximate osculating circle at  $Q$ , one predecessor point is necessary. Hence, it is not possible to give a circular step at the initial point of each branch. In our implementation, we replace this first step by a step  $L_F = \frac{L}{100}$  in the tangential direction.

(vi) Apply the inverse transformations  $R^{-1}$  and  $T^{-1}$  to  $A'$  to obtain the point  $A$ .

**Note 5** *The computation of  $T^{-1}$  and  $R^{-1}$  is trivial, since  $T^{-1} = [c_1 \ c_2 \ c_3 \ 1]^T$  and  $R^{-1}$  is equal to the transpose of  $R$ ,  $[R]^t$ .*

Newton iterations are normally performed to improve  $A$ , such that a closer point to the intersection curve is obtained at each marching step [8].

### 3 Convergence analysis

In this section we show algebraically that our circle, despite having very simple construction, converges to the exact osculating circle when the step size,  $L$ , is sufficiently small.

Given a parametric curve

$$f(t) = (f_1(t), f_2(t), f_3(t)).$$

The center  $EC(t)$  of the exact osculating circle to the intersection curve at point  $t = a$  is given by the expression:

$$EC(a) = f(a) + \left( \frac{|f'(a)|^3}{|f'(a) \times f''(a)|} \right) \frac{\vec{N}(a)}{|\vec{N}(a)|}, \quad (2)$$

where

$$\vec{N}(a) = f''(a) - \left( \frac{f'(a) \cdot f''(a)}{|f'(a)|^2} \right) f'(a). \quad (3)$$

The vector  $\vec{N}(a)$  is called the principal normal vector of the intersection curve at  $t = a$ .

The center  $C = (c_1, c_2, c_3)$  of the approximated osculating circle, defined by the points  $Q = f(b+h)$ ,  $P = f(a)$ , and their respective tangent vectors  $\vec{u} = f'(a)$  and  $\vec{v} = f'(b+h)$ , is the solution of the following  $3 \times 3$  linear system:

$$\begin{cases} C \cdot \vec{u} = P \cdot \vec{u} \\ C \cdot \vec{v} = Q \cdot \vec{v} \\ C \cdot (\vec{u} \times \vec{v}) = Q \cdot (\vec{u} \times \vec{v}) \end{cases} \quad (4)$$

Denoting, respectively, as  $(p_1, p_2, p_3)$ ,  $(q_1, q_2, q_3)$ ,  $(u_1, u_2, u_3)$ , and  $(v_1, v_2, v_3)$  the coordinates of  $P$ , the coordinates of  $Q$ , the coordinates of  $\vec{u}$ , and the coordinates of  $\vec{v}$ , we get with the help of MATHEMATICA [3] the following solution to eq. (4):

$$\begin{aligned}
c_1 = & (p_1 * u_1 * (u_1 * (v_2^2 + v_3^2) - v_1 * (u_2 * v_2 + u_3 * v_3)) \\
& + p_2 * u_2 * (u_1 * (v_2^2 + v_3^2) - v_1 * (u_2 * v_2 + u_3 * v_3)) \\
& + p_3 * u_3 * (u_1 * (v_2^2 + v_3^2) - v_1 * (u_2 * v_2 + u_3 * v_3)) \\
& - q_1 * (u_1 * v_1 * (u_2 * v_2 + u_3 * v_3) - u_2^2 * (v_1^2 + v_3^2) + \\
& u_3 * (2 * u_2 * v_2 * v_3 - u_3 * (v_1^2 + v_2^2))) \\
& + (v_1 * (u_2 * v_2 + u_3 * v_3) - u_1 * (v_2^2 + v_3^2)) * \\
& (q_2 * u_2 + q_3 * u_3) / (u_1^2 * (v_2^2 + v_3^2)) \\
& - 2 * u_1 * v_1 * (u_2 * v_2 + u_3 * v_3) + u_2^2 * (v_1^2 + v_3^2) \\
& - 2 * u_2 * u_3 * v_2 * v_3 + u_3^2 * (v_1^2 + v_2^2))
\end{aligned} \tag{5}$$

$$\begin{aligned}
c_2 = & -(p_1 * u_1 * (u_1 * v_1 * v_2 - u_2 * (v_1^2 + v_3^2) + u_3 * v_2 * v_3) \\
& + p_2 * u_2 * (u_1 * v_1 * v_2 - u_2 * (v_1^2 + v_3^2) + u_3 * v_2 * v_3) \\
& + p_3 * u_3 * (u_1 * v_1 * v_2 - u_2 * (v_1^2 + v_3^2) + u_3 * v_2 * v_3) \\
& - q_1 * u_1 * (u_1 * v_1 * v_2 - u_2 * (v_1^2 + v_3^2) + u_3 * v_2 * v_3) \\
& - q_2 * (u_1^2 * (v_2^2 + v_3^2) - u_1 * v_1 * (u_2 * v_2 + 2 * u_3 * v_3) - \\
& u_2 * u_3 * v_2 * v_3 + u_3^2 * (v_1^2 + v_2^2)) \\
& - q_3 * u_3 * (u_1 * v_1 * v_2 - u_2 * (v_1^2 + v_3^2) + u_3 * v_2 * v_3)) \\
& / (u_1^2 * (v_2^2 + v_3^2) - 2 * u_1 * v_1 * (u_2 * v_2 + u_3 * v_3) \\
& + u_2^2 * (v_1^2 + v_3^2) - 2 * u_2 * u_3 * v_2 * v_3 \\
& + u_3^2 * (v_1^2 + v_2^2))
\end{aligned} \tag{6}$$

$$\begin{aligned}
c_3 = & -(p_1 * u_1 * (u_1 * v_1 * v_3 + u_2 * v_2 * v_3 - u_3 * (v_1^2 + v_2^2)) \\
& + p_2 * u_2 * (u_1 * v_1 * v_3 + u_2 * v_2 * v_3 - u_3 * (v_1^2 + v_2^2)) \\
& + p_3 * u_3 * (u_1 * v_1 * v_3 + u_2 * v_2 * v_3 - u_3 * (v_1^2 + v_2^2)) \\
& - q_1 * u_1 * (u_1 * v_1 * v_3 + u_2 * v_2 * v_3 - u_3 * (v_1^2 + v_2^2)) \\
& - q_2 * u_2 * (u_1 * v_1 * v_3 + u_2 * v_2 * v_3 - u_3 * (v_1^2 + v_2^2)) \\
& - q_3 * (u_1^2 * (v_2^2 + v_3^2) - u_1 * v_1 * (2 * u_2 * v_2 + u_3 * v_3) + \\
& u_2 * (u_2 * (v_1^2 + v_3^2) - u_3 * v_2 * v_3))) \\
& / (u_1^2 * (v_2^2 + v_3^2) - 2 * u_1 * v_1 * (u_2 * v_2 + u_3 * v_3) \\
& + u_2^2 * (v_1^2 + v_3^2) - 2 * u_2 * u_3 * v_2 * v_3 \\
& + u_3^2 * (v_1^2 + v_2^2)).
\end{aligned} \tag{7}$$

**Note 6** Since  $c_1$ ,  $c_2$  and  $c_3$  are continuous functions, there exists a direct correspondence between the step size  $L$  and the increment  $h$ . In other words, we may indistinctly use  $L$  and  $h$  to refer the limit of the convergence of  $C$  (eq. (4)).

To prove algebraically that our circle converges to the exact osculating circle when  $h \rightarrow 0$ , we should show that

- (i) the plane that contains our circle, whose center is  $C$ , tends to coincide with the plane that contains the exact osculating circle, whose center is

$EC$ , and  
(ii) the center  $C$  tends to  $EC$ .

**Proposition 1** *If  $\vec{u}$  and  $\vec{v}$  are non-parallel, then the plane  $\pi_C$  that contains our circle is the plane  $\pi_{EC}$  that contains the exact osculating circle at  $t = a$ , when  $h \rightarrow 0$ .*

**Proof.** If  $h \rightarrow 0$ , then  $Q \rightarrow P$ . Hence, to demonstrate  $\pi_C \rightarrow \pi_{EC}$ , we just need to show that  $\pi_C$  tends to be parallel to  $\pi_{EC}$ , i.e.  $\vec{u}$  and  $\vec{v}$  tend to be perpendicular to  $(N(a) \times \vec{u})$ .

Expanding  $\vec{v} \cdot (N(a) \times \vec{u})$ , we obtain

$$\begin{aligned}\vec{v} \cdot (N(a) \times \vec{u}) &= f'(a) \cdot [f''(a) - ((f'(a) \cdot f''(a))/|f'(a)|^2)f'(a)] \times f'(a-h) \\ &= f'(a) \cdot [f''(a) \times f'(a-h) \\ &\quad - ((f'(a) \cdot f''(a))/|f'(a)|^2)f'(a) \times f'(a-h)] \\ &= f'(a) \cdot [f''(a) \times f'(a-h) - 0] \\ &= f'(a) \cdot [-f'(a-h) \times f''(a)] \\ &= -(f'(a) \times f'(a-h)) \cdot f''(a).\end{aligned}$$

Then, applying the properties  $\vec{v}_1 \times \vec{v}_1 = 0$  and  $\vec{v}_1 \cdot \vec{v}_2 \times \vec{v}_3 = \vec{v}_1 \times \vec{v}_2 \cdot \vec{v}_3$ , we get

$$\lim_{h \rightarrow 0} [\vec{v} \cdot (\vec{N}(a) \times \vec{u})] = -(f'(a) \times f'(a)) \cdot f''(a) = 0$$

Analogously,

$$\lim_{h \rightarrow 0} [\vec{u} \cdot (\vec{N}(a) \times \vec{u})] = \lim_{h \rightarrow 0} f'(a-h) \cdot (\vec{N}(a) \times f'(a-h)) = -(f'(a) \times f'(a)) \cdot f''(a) = 0.$$

□

**Proposition 2** *If the intersection curve  $f(t)$  lies on the plane  $z = 0$ , then  $C$  converges to  $EC$  when  $h \rightarrow 0$ .*

**Proof.** In this case eqs. (5), (6) and (7) are reduced, respectively, to

$$c_1 = \frac{-q_1 * u_2 * v_1 + p_1 * u_1 * v_2 + p_2 * u_2 * v_2 - q_2 * u_2 * v_2}{u_1 * v_2 - u_2 * v_1}, \quad (8)$$

$$c_2 = \frac{-p_1 * u_1 * v_1 + q_1 * u_1 * v_1 - p_2 * u_2 * v_1 + q_2 * u_1 * v_2}{u_1 * v_2 - u_2 * v_1}, \quad (9)$$

$$c_3 = 0. \quad (10)$$

Replacing  $p_1 = f_1(a-h)$ ,  $p_2 = f_2(a-h)$ ,  $q_1 = f_1(a)$ ,  $q_2 = f_2(a)$ ,  $u_1 = f'_1(a-h)$ ,  $u_2 = f'_2(a-h)$ ,  $v_1 = f'_1(a)$ , and  $v_2 = f'_2(a)$  in eqs. (8), (9), and (10), we obtain

$$c_1 = \frac{E1}{E3}$$

$$c_2 = \frac{E2}{E3},$$

where

$$E_1 = -f_1(a)f_2'(a-h)f_1'(a) + f_1(a-h)f_1'(a-h)f_2'(a) + f_2(a-h)f_2'(a-h)f_2'(a) - f_2(a)f_2'(a-h)f_2'(a), \quad (11)$$

$$E_2 = -f_1(a-h)f_1'(a-h)f_1'(a) + f_1(a)f_1'(a-h)f_1'(a) - f_2(a-h)f_2'(a-h)f_1'(a) + f_2(a)f_1'(a-h)f_2'(a), \quad (12)$$

$$E_3 = f_1'(a-h)f_2'(a) - f_2'(a-h)f_1'(a). \quad (13)$$

Adding and subtracting  $f_1'(a)f_2'(a)$  in eq. (13),

$$E_3 = f_1'(a-h)f_2'(a) + f_1'(a)f_2'(a) - f_1'(a)f_2'(a) - f_2'(a-h)f_1'(a) = f_1'(a)(f_2'(a) - f_2'(a-h)) - f_2'(a)(f_1'(a) - f_1'(a-h)),$$

it follows that

$$\lim_{h \rightarrow 0} \frac{E_3}{h} = f_1'(a) \lim_{h \rightarrow 0} \frac{f_2'(a) - f_2'(a-h)}{h} - f_2'(a) \lim_{h \rightarrow 0} \frac{f_1'(a) - f_1'(a-h)}{h} = f_1'(a)f_2''(a) - f_1''(a)f_2'(a).$$

Adding and subtracting the terms  $f_1(a)f_2'(a)f_1'(a-h)$  and  $f_1(a)f_1'(a-h)f_2'(a-h)$  in eq. (11), we obtain

$$\begin{aligned} E_1 &= -f_1(a)f_2'(a-h)f_1'(a) + f_1(a-h)f_1'(a-h)f_2'(a) + f_2(a-h)f_2'(a-h)f_2'(a) \\ &\quad - f_2(a)f_2'(a-h)f_2'(a) + f_1(a)f_2'(a)f_1'(a-h) - f_1(a)f_2'(a)f_1'(a-h) \\ &\quad + f_1(a)f_1'(a-h)f_2'(a-h) - f_1(a)f_1'(a-h)f_2'(a-h) \\ &= -f_2'(a)f_1'(a-h)(f_1(a) - f_1(a-h)) - f_2'(a)f_2'(a-h)(f_2(a) - f_2(a-h)) \\ &\quad - f_1(a)f_2'(a-h)(f_1'(a) - f_1'(a-h)) + f_1(a)f_1'(a-h)(f_2'(a) - f_2'(a-h)). \end{aligned}$$

Hence,

$$\begin{aligned} \lim_{h \rightarrow 0} \frac{E_1}{h} &= -f_2'(a) \lim_{h \rightarrow 0} f_1'(a-h) \left( \lim_{h \rightarrow 0} \frac{f_1(a) - f_1(a-h)}{h} \right) - \\ &\quad f_2'(a) \lim_{h \rightarrow 0} f_2'(a-h) \lim_{h \rightarrow 0} \frac{f_2(a) - f_2(a-h)}{h} - \\ &\quad f_1(a) \left( \lim_{h \rightarrow 0} f_1'(a-h) \lim_{h \rightarrow 0} \frac{f_2'(a) - f_2'(a-h)}{h} \right) + \\ &\quad \lim_{h \rightarrow 0} f_2'(a-h) \lim_{h \rightarrow 0} \frac{f_1'(a) - f_1'(a-h)}{h} = \\ &= -f_2'(a)((f_1'(a))^2 + (f_2'(a))^2) + f_1(a)(f_1'(a)f_2''(a) - f_2'(a)f_1''(a)). \end{aligned}$$

Consequently,

$$\lim_{h \rightarrow 0} c_1 = \lim_{h \rightarrow 0} \frac{E_1}{E_3} = \frac{\lim_{h \rightarrow 0} \frac{E_1}{h}}{\lim_{h \rightarrow 0} \frac{E_3}{h}} = f_1(a) - \frac{f_2'(a)((f_1'(a))^2 + (f_2'(a))^2)}{f_1'(a)f_2''(a) - f_1''(a)f_2'(a)}.$$

Similarly, adding and subtracting the terms  $f_2(a)f_1'(a)f_2'(a-h)$  and  $f_2(a)f_1'(a)f_2'(a)$  in eq.(12),

$$E_2 = f_1'(a-h)f_1'(a-h)(f_1(a) - f_1(a-h)) - f_2(a)f_2'(a)(f_1'(a) - f_1'(a-h)) \\ + f_1'(a)f_2'(a-h)(f_2(a-h) - f_2(a)) + f_2(a)f_1'(a)(f_2'(a-h) - f_2'(a)),$$

$$\lim_{h \rightarrow 0} \frac{E_2}{h} = f_1'(a)((f_1'(a))^2 + (f_2'(a))^2) + f_2(a)(f_1'(a)f_2''(a) - f_2'(a)f_1''(a)),$$

and

$$\lim_{h \rightarrow 0} c_2 = \lim_{h \rightarrow 0} \frac{E_2}{E_3} = \frac{\lim_{h \rightarrow 0} \frac{E_2}{h}}{\lim_{h \rightarrow 0} \frac{E_3}{h}} = f_2(a) + \frac{f_1'(a)((f_1'(a))^2 + (f_2'(a))^2)}{f_1'(a)f_2''(a) - f_1''(a)f_2'(a)}.$$

In this way, we show that in the limit the center of our constructed circle is

$$\left( f_1(a) - \frac{f_2'(a)((f_1'(a))^2 + (f_2'(a))^2)}{f_1'(a)f_2''(a) - f_1''(a)f_2'(a)}, f_2(a) + \frac{f_1'(a)((f_1'(a))^2 + (f_2'(a))^2)}{f_1'(a)f_2''(a) - f_1''(a)f_2'(a)}, 0 \right),$$

which corresponds to the center of the osculating circle of the curve  $f(t) = (f_1(t), f_2(t), 0)$  at  $t = a$ .

□

**Corollary 1** *If the intersection curve  $f(t)$  is a plane curve, then  $C$  converges to  $EC$  when  $h \rightarrow 0$ .*

**Proof.** In this case just a translation  $T$  followed by a rotation  $R$ , as explained in Section 2.2, are sufficient for changing the reference system of  $f(t)$  such that  $f(t)$  lies on the plane  $z = 0$ . Then, from Proposition 2 we have  $C \rightarrow EC$ .

□

**Note 7** *By our construction, the transformation (Section 2.2)*

$$\begin{aligned}
[RT] &= \begin{bmatrix} m_{11} & m_{12} & m_{13} & 0 \\ m_{21} & m_{22} & m_{23} & 0 \\ m_{31} & m_{32} & m_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & m_{14} \\ 0 & 1 & 0 & m_{24} \\ 0 & 0 & 1 & m_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{11}m_{14} + m_{12}m_{24} + m_{13}m_{34} \\ m_{21} & m_{22} & m_{23} & m_{21}m_{24} + m_{22}m_{24} + m_{23}m_{34} \\ m_{31} & m_{32} & m_{33} & m_{31}m_{14} + m_{32}m_{24} + m_{33}m_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

applied to the intersecting surfaces should be such a one that

$$\begin{aligned}
H &= [RT]P = (h_1, h_2, h_3) \\
&= (m_{11}p_1 + m_{12}p_2 + m_{13}p_3 + (m_{11}m_{14} + m_{12}m_{24} + m_{13}m_{34}), \\
&\quad m_{21}p_1 + m_{22}p_2 + m_{23}p_3 + (m_{21}m_{24} + m_{22}m_{24} + m_{23}m_{34}), 0), \quad (14)
\end{aligned}$$

$$\begin{aligned}
K &= [RT]Q = (k_1, k_2, k_3) \\
&= (m_{11}q_1 + m_{12}q_2 + m_{13}q_3 + (m_{11}m_{14} + m_{12}m_{24} + m_{13}m_{34}), \\
&\quad m_{21}q_1 + m_{22}q_2 + m_{23}q_3 + (m_{21}m_{24} + m_{22}m_{24} + m_{23}m_{34}), 0), \quad (15)
\end{aligned}$$

$$\begin{aligned}
\vec{U} &= [R]\vec{u} = (U_1, U_2, U_3) \\
&= (m_{11}u_1 + m_{12}u_2 + m_{13}u_3, m_{21}u_1 + m_{22}u_2 + m_{23}u_3, 0), \quad (16)
\end{aligned}$$

$$\begin{aligned}
\vec{V} &= [R]\vec{v} = (V_1, V_2, V_3) \\
&= (m_{11}v_1 + m_{12}v_2 + m_{13}v_3, m_{21}v_1 + m_{22}v_2 + m_{23}v_3, 0). \quad (17)
\end{aligned}$$

**Proposition 3** *If the intersection curve  $f(t)$  is a twisted curve, then  $C$  converges to  $EC$  when  $h \rightarrow 0$ .*

**Proof.** Applying the transformation  $[RT]$  to the intersecting surfaces and using the notation introduced in Note 7, eqs. (5), (6), and (7) assume, respectively, the following aspect

$$c_1 = \frac{-k_1 * U_2 * V_1 + h_1 * U_1 * V_2 + h_2 * U_2 * V_2 - k_2 * U_2 * V_2}{U_1 * V_2 - U_2 * V_1}, \quad (18)$$

$$c_2 = \frac{-h_1 * U_1 * V_1 + k_1 * U_1 * V_1 - h_2 * U_2 * V_1 + k_2 * U_1 * V_2}{U_1 * V_2 - U_2 * V_1}, \quad (19)$$

$$c_3 = 0. \quad (20)$$

To show that  $C$  is  $EC$  in the limit, it is sufficient to demonstrate that at  $t = a$

$$\lim_{h \rightarrow 0} (c_1, c_2, c_3) = \{\lim_{h \rightarrow 0} [RT]\} EC(a).$$

By construction, a transformation  $[RT]$  is applied on  $f(t)$ , such that our circle at  $t = a$  lies on  $z = 0$ . From Proposition 1 the plane  $\pi_C$  that contains our circle tends to be the plane  $\pi_{EC}$  that contains the exact osculating circle at  $t = a$  in the limit. Then, the center of the exact osculating circle of  $\{\lim_{h \rightarrow 0} [RT]\}f(t)$  at  $t = a$  must have  $z = 0$ .

Replacing  $h_1 = g_1(a - h)$ ,  $h_2 = g_2(a - h)$ ,  $k_1 = g_1(a)$ ,  $k_2 = g_2(a)$ ,  $U_1 = g'_1(a - h)$ ,  $U_2 = g'_2(a - h)$ ,  $V_1 = g'_1(a)$ , and  $V_2 = g'_2(a)$  in eqs. (18) and (19), we may follow the same idea used for the proof of Proposition 2 and come to

$$\begin{aligned} \lim_{h \rightarrow 0} c_1 &= g_1(a) - \frac{g'_2(a)((g'_1(a))^2 + (g'_2(a))^2)}{g'_1(a)g''_2(a) - g''_1(a)g'_2(a)}, \\ \lim_{h \rightarrow 0} c_2 &= g_2(a) + \frac{g'_1(a)((g'_1(a))^2 + (g'_2(a))^2)}{g'_1(a)g''_2(a) - g''_1(a)g'_2(a)}. \end{aligned}$$

Since  $[RT]$  is an isometry and

$$g(a) = \{\lim_{h \rightarrow 0} [RT]\}f(t) = (g_1(a), g_2(a), 0),$$

we conclude that

$$\left( g_1(a) - \frac{g'_2(a)((g'_1(a))^2 + (g'_2(a))^2)}{g'_1(a)g''_2(a) - g''_1(a)g'_2(a)}, g_2(a) + \frac{g'_1(a)((g'_1(a))^2 + (g'_2(a))^2)}{g'_1(a)g''_2(a) - g''_1(a)g'_2(a)}, 0 \right)$$

corresponds to the center of the osculating circle at  $t = a$  of  $\{\lim_{h \rightarrow 0} [RT]\}f(t)$ , also expressed by  $\{\lim_{h \rightarrow 0} [RT]\}EC(a)$ .

□

## 4 Comparisons

Comparisons with some existing tracing techniques, namely along tangent vector, parabola, and by using embedding schemes, are presented in this section.

### 4.1 Tangential step

By tracing an intersection curve with the tangential step, one moves from point to point by a specific step in the direction of tangent vector at each point. This direction corresponds to the first derivative at the point. Although our circle at each point  $P$  was not constructed on the basis of higher derivatives, we demonstrated in Section 3 that it tends to be an osculating circle at  $P$ , which

has contact of at least second order. Because the higher derivatives are used in the computation of the next tracing point, the near lies this point to the curve [8], we may state that our algorithm need less Newton iterations to improve its coordinates in relation to the exact intersection curve.

Despite the improvement in accuracy, our algorithm has the same order of complexity of the one that uses the tangential step. Indeed, it is easy to see from Section 2.2 that, besides  $k$  operations to compute the tangent vector  $\vec{v}$ , we may need less than 100 operations (multiplications and additions) to solve eq. (1) and geometric transformations to compute the circular step at each point.

**Example 1** *For each following pair of surfaces we computed a branch of the intersection curve by two marching techniques - circular and tangential steps - with the same length step  $L$ . The number of distinct traced points (points distant each other less than 0.1 length units were considered identical), and the number of iterations necessary to relax the approximate next point to the intersection curve are summarized in Table 1.*

- (i) *The pair of surfaces  $S_1$  (Fig. 5)*
  - *Cylinder:  $F(u, v) = (v + 4 \sin(u), 1.5v, 5 + v + 4 \cos(u))$ ,  $-\pi \leq u \leq \pi$ ,  $-9 \leq v \leq 9$ .*
  - *Paraboloid:  $G(u, v) = (u, v, 9 - \frac{u^2+v^2}{5})$ ,  $-7.5 \leq u \leq 7.5$ ,  $-7.5 \leq v \leq 7.5$ .*
- (ii) *The pair of tore  $S_2$  [1] (Fig. 3)*
  - *Torus:  $F(u, v) = ((3 + \cos(u)) \sin(v), (3 + \cos(u)) \cos(v), \sin(u))$ ,  $-\pi \leq u, v \leq \pi$ ,*
  - *Torus:  $G(u, v) = (\sin(u), (3 + \cos(u)) \sin(v), (3 + \cos(u)) \cos(v))$ ,  $-\pi \leq u, v \leq \pi$ .*
- (iii) *The pair of surfaces  $S_3$  (Fig. 4)*
  - *Torus:  $F(u, v) = ((10 - 5 \sin(u)) \sin(v), 5 \cos(u), (10 - 5 \sin(u)) \cos(v))$ ,  $0 \leq u \leq 2\pi$ ,  $0 \leq v \leq 2\pi$ .*
  - *Cylinder:  $G(u, v) = (v, 5 \cos(u), 5 \sin(u))$ ,  $0 \leq u \leq 2\pi$ ,  $-20 \leq v \leq 20$ .*

*Observe that, in comparison with tangential steps, our algorithm always traced longer curve segments (much more distinct points) and no more than 2 iterations were necessary to improve the coordinates of the traced points.*

#### 4.2 Parabolic step

Stoyanov [19] proposed to step along a parabola expressed by

$$p(s) = f(s_0) + (s - s_0)f'(s_0) + \frac{1}{2}(s - s_0)^2 f''(s_0), \quad (21)$$

Table 1  
Example 1

Surfaces	$L$	Method	#points	1 it	2 it	3 it
$S_1$	0.05	Circular	556	73.7%	26.3%	0.0%
		Tangent	350	0.3%	99.7%	0.0%
	0.20	Circular	377	2.6%	97.4%	0.0%
		Tangent	233	0.0%	74.3%	25.7%
$S_2$	0.05	Circular	431	23.7%	76.3%	0.0%
		Tangent	314	0.3%	99.7%	0.0%
	0.20	Circular	212	4.2%	95.8%	0.0%
		Tangent	106	0.0%	47.2%	52.8%
$S_3$	0.05	Circular	1259	100.0%	0.0%	0.0%
		Tangent	495	8.0%	92.0%	0.0%
	0.20	Circular	846	38.5%	61.5%	0.0%
		Tangent	240	1.7%	98.3%	0.0%

where  $f(s)$  is the intersection curve parameterized by arc length  $s$  and  $s_0$  corresponds to the contact point of  $f(s)$  to  $p(s)$ . Several linear equation systems must be solved to estimate the first and second partial derivatives,  $f'(s)$  and  $f''(s)$ , that appear in eq. (21). Therefore, it involves much more operations than our algorithm.

Note that eq. (21) is actually the first three terms of the Taylor series expansion at  $s = s_0$ ,

$$f(s) = f(s_0) + (s - s_0)f'(s_0) + \frac{1}{2}(s - s_0)^2 f''(s_0) + \frac{1}{6}f'''(s_0) + \dots$$

Therefore,  $p(s)$  has contact of second order to  $f(s)$  at  $s = s_0$ . And, since  $p(s_0) = f(s_0)$ , it lies, as the osculating circle, on the osculating plane at  $s = s_0$ .

It remains to verify how good is the approximation of  $p(s)$  to  $f(s)$  in the neighborhood of  $s = s_0$  in comparison with the osculating circle

$$g(s) = f(s_0) + \frac{1}{\kappa} \sin(\kappa(s - s_0)) \vec{T}(s_0) + \frac{1}{\kappa} (1 - \cos(\kappa(s - s_0))) \vec{N}(s_0), \quad (22)$$

where  $\vec{T}(s_0)$  and  $\vec{N}(s_0)$  correspond, respectively, to the tangent and normal vector of  $f(s)$  at  $s = s_0$ . Recall that we demonstrated in Section 3 that our constructed circle tends to this circle in the limit.

Rewriting eq. (21) in terms of  $\vec{T}(s_0)$  and  $\vec{N}(s_0)$

$$p(s) = f(s_0) + (s - s_0)\vec{T}(s_0) + \frac{1}{2}(s - s_0)^2\kappa\vec{N}(s_0),$$

we performed several numerical comparisons, such as the ones presented in Example 2, and came to the conjecture that  $g(s)$  is just slightly closer to  $f(s)$  than  $p(s)$  in the neighborhood of  $s = s_0$ . Hence, from our opinion the major advantage of our algorithm over the algorithm proposed by Stoyanov is its simplicity.

**Example 2** *This example illustrates the approximation of  $g(s)$  (osculating circle) and  $p(s)$  (parabola) to a given curve  $\alpha(s)$  at  $s_0 + L$  under the condition that  $p(s_0) = g(s_0) = \alpha(s_0)$ .*

*We computed  $g(s)$  and  $p(s)$  for four distinct values  $s_0$  of*

$$\begin{aligned} F(s) &= \left( \cos\left(\frac{s}{\sqrt{2}}\right), \sin\left(\frac{s}{\sqrt{2}}\right), \frac{s}{\sqrt{2}} \right) \\ G(s) &= \left( \frac{(1+s)^{3/2}}{3}, \frac{(1-s)^{3/2}}{3}, \frac{s}{\sqrt{2}} \right) \\ H(s) &= \left( \frac{s + \sqrt{s^2 + 1}}{2}, \frac{1}{2(s + \sqrt{s^2 + 1})}, \frac{\sqrt{2}\ln(s + \sqrt{s^2 + 1})}{2} \right). \end{aligned}$$

*After then,  $P_p = p(s_0 + L)$ ,  $P_g = g(s_0 + L)$  and their distances to  $\alpha(s)$ ,  $d_p$  and  $d_g$ , were calculated (Table 2). We observed that  $d_g < d_p$  for all tested cases.*

### 4.3 Embedding schemes

In [1] the authors proposed to trace along the tangent direction by using the embedding scheme [8]. To remedy the “imprecision” of tangential steps, they show that by monitoring the behavior of a system of nonlinear equations building up from a set of inequality constraints, abrupt variations on tangent direction may be reliably detected. The drawback of this procedure is the complexity.

Replacing the conventional, initial value problem over an unknown parameter interval by a boundary value problem over a fixed, specified interval is a solution suggested in [7] for determining “unbiasly” a branch of the intersection curve. More specifically, the authors applied spline collocation to a two point boundary value problem for a differential algebraic equation of index two. This scheme requires that the surfaces must be restricted to ones “for which a robust capability for determining all solutions to a nonlinear system of equations”.

Table 2  
Exemplo 2

$\alpha(s)$	$s_0$	$L$	$d_p$	$d_g$
$F(s)$	-0,5	0,05	0,000007	0,000005
	-0,5	0,10	0,000059	0,000042
	0,5	0,05	0,000007	0,000005
	0,5	0,10	0,000059	0,000042
$G(s)$	-0,5	0,05	0,000007	0,000006
	-0,5	0,10	0,000056	0,000049
	0,5	0,05	0,000008	0,000007
	0,5	0,10	0,000065	0,000058
$H(s)$	-0,5	0,05	0,000013	0,000012
	-0,5	0,10	0,000108	0,000093
	0,5	0,05	0,000013	0,000012
	0,5	0,10	0,000105	0,000092

Through our exhaustive tests we may claim that our algorithm is so reliable as the abovementioned solutions, with the advantage that it is much simpler and covers a larger class of surfaces (any regular surfaces). From an initial point, it always traces a branch of intersection curve until one of three cases occurs:

- (i) the domain's boundary is reached,
- (ii) an initial point is reached (when the next reached point lies in the neighborhood of the initial point. In our implementation, an open disk of radius  $\frac{2L}{3}$ ), or
- (iii) singularities of second or higher order.

Let us give some examples.

Fig. 3 depicts the intersection of two tore given in Section 4.1. Observe that since there are six branches, six initial points (white square marks) were necessary to trace them. In this case, only boundary condition (black square marks) was activated to stop tracing. It is worth noting that the bifurcation point in the middle of the both domain spaces,  $F$  and  $G$ , was crossed over without problem.

Another example that validates the behavior of our algorithm in the presence of bifurcation points is the intersection of a torus and a cylinder given in Section 4.1 (Fig. 4). Again the bifurcation points were crossed over without any special handling. Observe that the topology of the intersection curve looks

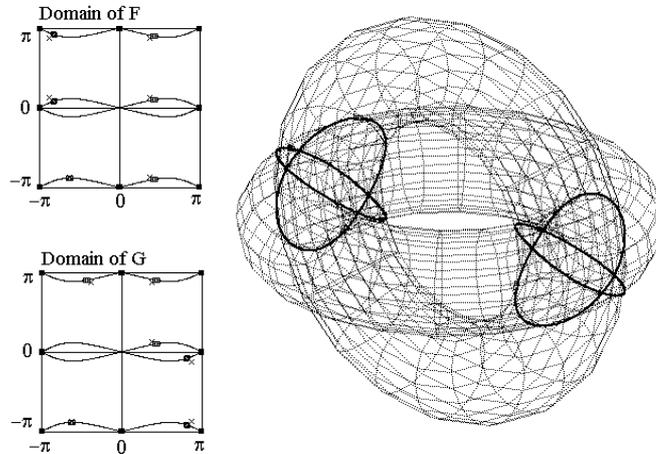


Fig. 3. Intersection torus/torus

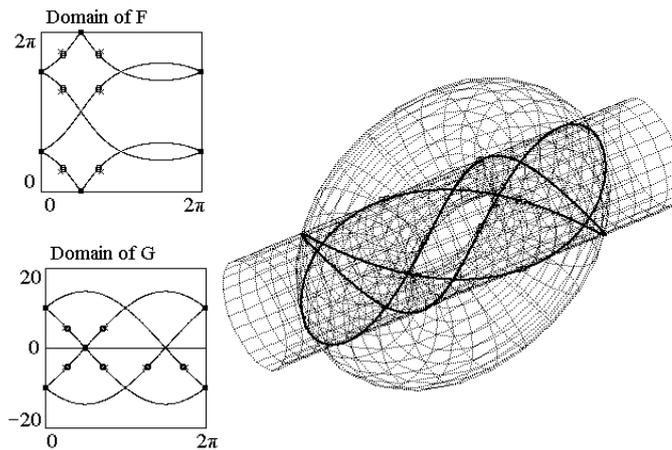


Fig. 4. Intersection torus/cylinder

distinctly in the parameter spaces.

The intersection of a cylinder and paraboloid given in Section 4.1 (Fig. 5) also presents distinct topology in the parameter spaces. This example also illustrates how our algorithm behaves in the presence of turning points. Note that, despite of two turning points, just two initial points were necessary to trace out the whole intersection. This is due to the fact that each step adjusts adaptatively to the curve shape and follows correctly the variation of curvature. Fig. 6 is another example of a curve with several turning points. The involved surfaces are

- $F(u, v) = (u, v, \frac{10(u^2-v^2)}{2+u^4+v^4})$ ,  $u \in [-4, 4]$ ,  $v \in [-4, 4]$ , and
- $G(u, v) = (\frac{u}{v^2+1}, v, \frac{u-1}{2})$ ,  $u \in [-4, 5]$ ,  $v \in [-4, 4]$ .

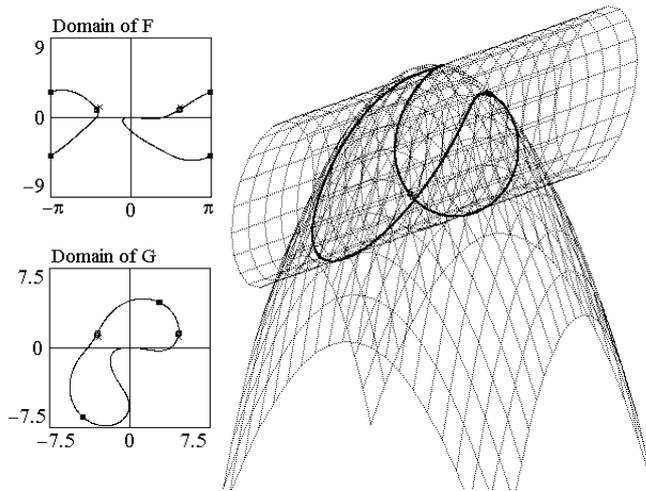


Fig. 5. Intersection cylinder/paraboloid

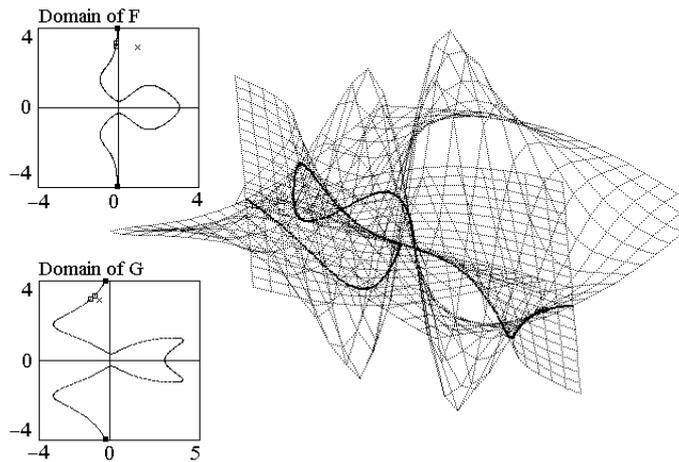


Fig. 6. Intersection of two surfaces with rational parameterization

Our algorithm also handles correctly spatially closed branches, as demonstrated by the intersection of (Fig. 7)

- $F(u, v) = (u, v, 21/100 - 321/100u^2 - 107/100v^2 + 81/5u^4 + 54/5u^2v^2 + 9/5v^4 - 27u^6 - 27u^4v^2 - 9u^2v^4 - v^6)$ ,  $u, v \in [-1, 1]$ , and
- $G(r, s) = (r, s, 0)$ ,  $r, s \in [-1, 1]$ .

In this case, although the minimal distances between the ellipses are 0.03583 (external and middle ones) and 0.03896 (middle and internal ones), the step length  $L = 0.05$  was used to track each branch without failure. Observe also that the tracing stopped when it returned to the initial point.

Finally, in order to illustrate the behavior of our algorithm in the neighborhood of singularities of two non-regular surfaces, we computed the intersection of

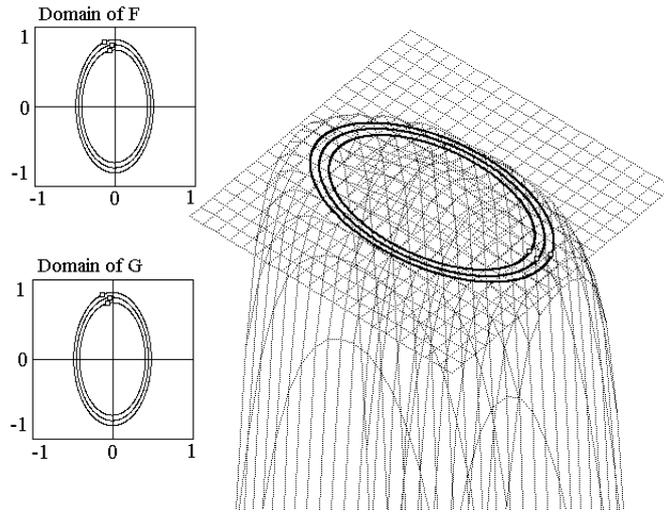


Fig. 7. Intersection plane/surface of degree 6

(Fig. 8) [4]

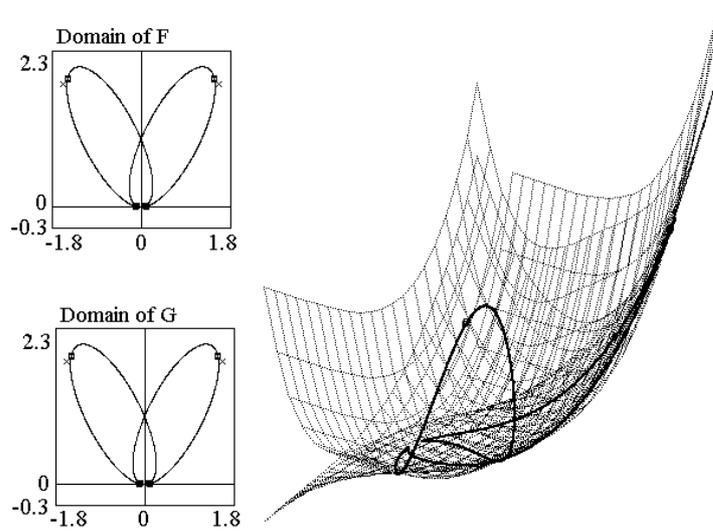


Fig. 8. Intersections surface of degree 3/surface of degree 4

- $F(u, v) = (u, v, (2u^4 + v^4)/10)$ ,  $u \in [-1.8, 1.8]$ ,  $v \in [-0.3, 2.3]$ , and
- $G(r, s) = (r, s, (3r^2s - s^2 + 2s^3)/10)$ ,  $r \in [-1.8, 1.8]$ ,  $s \in [-0.3, 2.3]$ .

Since  $F$  and  $G$  present singularity at  $(0, 0, 0)$ , we provided two initial points to trace out the two crossed ellipses separately.

#### 4.4 Asteasu circular step

In [2] a method for computing the next starting point by means of an osculating circle was presented. It is, however, restricted to the surfaces given in implicit form. Our method is applicable for parametric surfaces.

### 5 Concluding Remarks

We have presented an alternative way to trace out a intersection curve branch from a starting point. According to our experiments, our circular marching algorithm is efficient and robust. In the majority of cases the estimated next point is closer to the intersection curve – normally two Newton iterations are sufficient to improve its accuracy [5].

The major challenge for making viable marching with circular steps was to determine, with low computational cost, an approximate osculating circle at each point. After exhaustive tests we are prone to conclude that our approximate osculating circle is promising in practice. We believe that the combination of our technique with a robust procedure to determine the initial points, e.g [1,7], may result in a powerful intersection algorithm.

### 6 Acknowledgment

The authors are grateful to the reviewers for excellent recommendations. The first author also acknowledge UFPB for their support of his work at Unicamp performed under PICD/CAPES Program.

### References

- [1] K. Abdel-Malek and H.J.Yeh. Determining intersection curves between surfaces of two solids. *Computer Aided Design*, 28(6/7):539–549, 1996.
- [2] C. Asteasu. Intersection of arbitrary surfaces. *Computer-Aided Design*, 20:533–538, 1988.
- [3] N. Blachman. *Matematica: A practical approach*. Prentice Hall Inc., New Jersey, 1992.
- [4] C.L.Bajaj, C.M.Hoffmann, J.E.Hopcroft, and R.E.Lynch. Tracing surface intersections. *Computer Aided Geometric Design*, 5:285–307, 1988.

- [5] Lenimar Nunes de Andrade. *Tracing along regular surface intersection with circular steps*. PhD thesis, FEEC - Unicamp, Campinas, July 1998. (in portuguese).
- [6] G.A.Kriezis and N.M.Patrikalakis. Rational polynomial surface intersections. In Gary A.Gariele, editor, *Advances in Design Automation - 1991*, volume 2, pages 43–52, Miami, Florida, September 1991.
- [7] Thomas A. Grandine and Frederick W. Klein IV. A new approach to the surface intersection problem. *Computer Aided Geometric Design*, 14(2):111–134, February 1997.
- [8] Josef Hoschek and Dieter Lasser. *Fundamentals of Computer Aided Geometric Design*. A K Peters, Ltd., Wellesly, Massachusetts, USA, 1993. Translation of: *Grundlagen der geometrischen Datenverarbeitung*, by Larry L. Schumaker.
- [9] Elizabeth G. Houghton, Robert F. Emmett, James D. Factor, and Chaman L. Sabharwal. Implementation of a divide-and-conquer method for intersection of parametric surfaces. *Computer Aided Geometric Design*, 2:173–183, 1985.
- [10] H.S.M.Coxeter. *Introduction to Geometry*. John Wiley & Sons, Inc., 2nd. ed. edition, 1989.
- [11] Pramod Koparkar. Surface intersection by switching from recursive subdivision to iterative refinement. *The Visual Computer*, 8:47–63, 1991.
- [12] Michael E. Mortenson. *Geometric Modeling*. John Wiley & Sons, USA, 1st. ed. edition, 1985. ISBN 0-471-88279-8.
- [13] M.P.Carmo. *Differential geometry of curves and surfaces*. Prentice Hall Inc., New Jersey, 1st. ed. edition, 1976.
- [14] Gregor Müllenheim. On determining start points for a surface/surface intersection algorithm. *Computer Aided Geometric Design*, 8:401–408, 1991.
- [15] A. Preusser. Computing area filling contours for surfases defined by piecewise polynomials. *Computer Aided Geometric Design*, 3:267–279, 1986.
- [16] 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.
- [17] 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.
- [18] Thomas W. Sederberg and R.J.Meyers. Loop detection in surface patch intersections. *Computer Aided Geometric Design*, 5:161–171, 1988.
- [19] Tz.E.Stoyanov. Marching along surface/surface intersection curves with an adaptative step length. *Computer Aided Geometric Design*, 9:485–489, 1992.