On Estimating Local Geometric Properties of Intersection Curves

WU, SHIN - TING[1], OSMAR ALÉSSIO[1], SUELI I. R. COSTA[2]

[1]Electrical and Computer Engineering Faculty, State University of Campinas -
P.O.Box 6101, 13083-970 - Campinas, SP, Brazil
ting,osmar@dca.fee.unicamp.br
[2]Institute of Mathematics, State University of Campinas -
P.O.Box 6176, 13083-970, Campinas, SP, Brazil
sueli@ime.unicamp.br

Abstract. This paper presents a simple and robust algorithm for estimating the local geometric properties of intersection curves, namely the tangent, normal, binormal, curvature and torsion vectors at any point. In addition, two new marching directions that make use of these properties are given.

1 Introduction

The numerical marching method is the most widely used method for computing intersection curves. This scheme comprises three primary phases [8, 6]: 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 [10], subdivision techniques [3, 9], and algebraic methods [1, 5] have been applied for handling the hunting problem. The *marching phase* computes sequences of points of an intersection curve branch by tracing out from a starting point. Incorrect step direction or size may lead to erroneous results. The *sorting phase* groups those sequences of points into disjoint branches of the intersection curve. When the points on the intersection curve can be found sequentially, this phase is trivial.

Most marching methods make use of local differential geometry or Taylor series expansions about each point of the intersection curve to control the step. Tracing in the tangent direction [8, 3], along a circle [2, 13], and along a parabola [11] are some solutions presented in the literature and the most used step size is the one dependent on the curvature [3, 11]. Differential equation system [7, 5] and continuation method [1] are also used to trace out a branch of the intersection curve.

When the parametric form of a curve is known its local properties, such as tangent, curvature, normal, binormal, curvature, and torsion, can be derived exactly. However, in the case of marching schemes, these properties are used for determining the unknown curve. Motivated by the applications of differential properties in determination of marching directions, Ye and Maekawa [14] proposed algorithms based on Differential Geometry to compute the local properties of the intersection curve as long as the intersection points are obtained.

Using torsion and the derivative of the curvature, we could test other marching directions inspired in classical Differential Geometry. Instead of using approximative curves that have contact of first and second order, we did some experiments with curves that have a contact of third order with the intersecting curve. However, after several attempts we realize that the formulas proposed by Ye and Maekawa [14], although theoretically correct, are not computationally robust at the intersecting points where the parametric surfaces are not enough transversal. The reason arises from the limitations of computer-based representation for small deviations of the intersecting surface normal vectors.

In this paper we present an alternative way for estimating the properties of the intersection curve of any two regular surfaces. The only requirement is that the normal vectors of the two intersecting surfaces are known. Our algorithm is based on the tangent and normal vectors estimated by Wu and Andrade [13]. Moreover, we used the results to derive more accurate marching directions.

Section 2 introduces some basic concepts of Differential Geometry and Section 3 summarizes some related results. In Section 4 we present our basic idea and its implementation. We also show that the estimated values are good approximations to the theoretical values. Section 5 illustrates the application of our algorithm to intersection problems. Finally, some concluding remarks are drawn in Section 6.

2 Basic Concepts

To be self-contained we summarize in this section some concepts to be used in Section 4 and fix notations for them [12, 4].

A parametric curve $\alpha = \alpha(u)$ in $\Re^3$,

$$\alpha(u) = (x(u), y(u), z(u)), u \in [a, b] \subset \Re,$$

0-7695-0878-2/00 $10.00 © 2000 IEEE

is called regular if its tangent vector

$$\dot{\alpha}(u) = (\dot{x}(u), \dot{y}(u), \dot{z}(u)) \neq 0$$

never vanishes.

If the curve is at least twice continuously differentiable, the deviation of a curve from a straight line at any point u can be measured by its curvature

$$\kappa(u) = \frac{\mid \dot{\alpha}(u) \times \ddot{\alpha}(u) \mid}{\mid \dot{\alpha}(u) \mid^3}, \tag{1}$$

where $\times$ denotes the cross product of two vectors. If $\kappa(u) = 0\ \forall u$, then the curve reduces to a straight line. An isolated point at which the curvature vanishes is an inflection or a flat point. From the curvature, one can derive the radius $r(u)$ of the (osculating) circle ($r(u) = \frac{1}{\kappa(u)}$) whose first and second derivatives agree with those of the curve at the point $\alpha(u)$.

If the curve is at least three times continuously differentiable and $\kappa(u) \neq 0$, one can measure the deviation of a curve from being planar by its torsion τ, given by

$$\tau(u) = \frac{(\dot{\alpha}(u) \times \ddot{\alpha}(u)) \cdot \dddot{\alpha}}{\mid \dot{\alpha}(u) \times \ddot{\alpha}(u) \mid^2}, \tag{2}$$

where $\dddot{\alpha}$ denotes the third-order derivative of α. The torsion $\tau(u)$ vanishes for planar curves. The functions $\kappa(u)$ and $\tau(u)$ are independent of the parameterization and determine uniquely a curve in $\Re^3$. When the parameter s is such one that $\mid \dot{\alpha}(s) \mid = 1\ \forall s$, we say that the parameterization of the curve is by arc length s.

To any point $\alpha(u)$ on the curve α we may introduce a special coordinate system to facilitate the description of local curve properties. This coordinate system with origin at $\alpha(u)$ has axes X, Z, and Y, respectively, in the directions

$$\begin{aligned} t(u) &= \frac{\dot{\alpha}(u)}{\mid \dot{\alpha}(u) \mid}, \\ b(u) &= \frac{\dot{\alpha}(u) \times \ddot{\alpha}(u)}{\mid \dot{\alpha}(u) \times \ddot{\alpha}(u) \mid}, \textit{and} \\ n(u) &= b(u) \times t(u). \end{aligned} \tag{3}$$

The vectors $t(u)$, $n(u)$, and $b(u)$ are called, respectively, (unit) tangent vector, (unit) normal vector, and (unit) binormal vector. This reference system or trihedron is called the Frenet frame. The associate planes are: osculating plane (tn), normal plane (nb), and rectifying plane (tb) (Figure 1).

One can get the so-called Frenet formulas for a curve α parameterized with respect to arc length s

$$\begin{array}{lccc} t'(s) &= & +\kappa(s)n(s) & \\ n'(s) &= -\kappa t(s) & & -\tau b(s) \\ b'(s) &= & -\tau n(s) & \end{array}$$

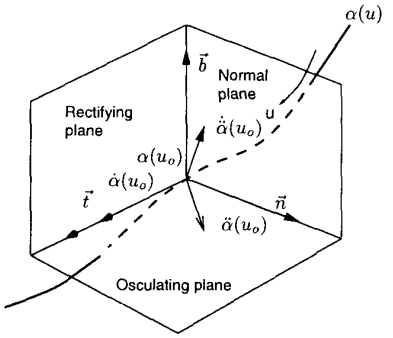

Figure 1: A Frenet frame.

From these equations and Taylor's formulas follows that when $\Delta s \to 0$ the projections of $\alpha(s)$ on the three planes of the Frenet frame behave near the point $\alpha(s_o)$ like the curve

$$\begin{aligned} X(s) &= s - \frac{\kappa(s_o)^2}{6}s^3 \\ Y(s) &= \frac{\kappa(s_o)}{2}s^2 + \frac{\kappa'(s_o)}{6}s^3 \\ Z(s) &= -\frac{\kappa(s_o)\tau(s_o)}{6}s^3 \end{aligned} \tag{4}$$

It is also shown that in the neighborhood of $\alpha(s_o)$ the curve $\alpha(s)$ has third-order contact to a circular helix set at its Frenet frame

$$\beta = \beta(\tilde{s}) = (a\cos\frac{\tilde{s}}{c}, a\sin\frac{\tilde{s}}{c}, -\frac{b\tilde{s}}{c}), \tag{5}$$

where

$$a = \frac{\kappa(s_o)}{\kappa^2(s_o) + \tau^2(s_o)},\ b = \frac{\tau(s_o)}{\kappa^2(s_o) + \tau^2(s_o)},\ c = \sqrt{a^2 + b^2}.$$

In particular, if $\tau(s_o)$ vanishes, the curve behaves like a circle.

As already mentioned, the curvature and the torsion are deviations of a curve from a straight line and from a osculating plane, respectively. Geometrically, with the use of Frenet frame, it is equivalent to say that they are, respectively, the angular velocities of $t(s)$ and $b(s)$ with $\Delta s \to 0$

$$\kappa(s) = \lim_{\Delta s \to 0} \frac{(t(s) - t(s - \Delta s))}{\Delta s} \tag{6}$$

$$\mid \tau(s) \mid = \lim_{\Delta s \to 0} \frac{(b(s) - b(s - \Delta s))}{\Delta s} \tag{7}$$

The torsion $\tau(s)$ may be positive or negative. It is positive when the vector $\alpha(s) - \alpha(s - \Delta s)$ is, with regard to the osculating plane, in the same half-space of the vector $b(s)$, otherwise it is negative.

We also have that

$$| \tau(s) | = \lim_{\Delta s \to 0} \frac{2 \sin \frac{\phi(\Delta s)}{2}}{\alpha(s) - \alpha(s - \Delta s)} \quad (8)$$

$$sign\,[\tau(s)] = sign\,[(\alpha(s) - \alpha(s - \Delta s)) \cdot b], \quad (9)$$

where $\phi(\Delta s)$ is the angle between the binormals at $\alpha(s)$ and $\alpha(s - \Delta s)$.

Consider a parametrized surface $r = r(u, v)$ in $\Re^3$,

$$r = (x(u, v), y(u, v), z(u, v)), \quad u, v \in [a, b] \times [c, d] \subset \Re^2.$$

If x, y, and z are differentiable and

$$r_u \times r_v \neq \vec{0} \text{ for } (u, v) \in [a, b] \times [c, d],$$

we say that $r(u, v)$ is a regular surface. The tangent vector to a surface curve $r(u(t), v(t))$ can be computed from

$$\dot{r} = \frac{\partial r}{\partial u}\dot{u} + \frac{\partial r}{\partial v}\dot{v}$$

In particular, the tangents to the isoparametric curves are given by

$$r_u = \frac{\partial r}{\partial u} \quad r_v = \frac{\partial r}{\partial v}.$$

They are nowhere tangent to each other and determine the surface tangent plane. The normal vector

$$\mathcal{N}(u, v) = \frac{r_u \times r_v}{| r_u \times r_v |} \quad (10)$$

together with the unnormalized vectors r_u and r_v form a local coordinate system at $r(u, v)$. This frame plays the same important role for surfaces as the Frenet frame does for curves.

Suppose that we cut the surface $r(u, v)$ at P with a plane that contains the normal vector at P. This plane intersects $r(u, v)$ along a plane curve $r(u(t), v(t))$ whose curvature is called the normal curvature κ_n of the surface in the direction $\dot{r}(u(t), v(t))$ at P. This curvature is given by

$$\kappa_n = \frac{L(\dot{u})^2 + 2M\dot{u}\dot{v} + N(\dot{v})^2}{E(\dot{u})^2 + 2F\dot{u}\dot{v} + G(\dot{v})^2}, \quad (11)$$

where

$$E = r_u \cdot r_u, \quad F = r_u \cdot r_v, \quad G = r_v \cdot r_v$$
$$L = r_{uu} \cdot \mathcal{N}, \quad M = r_{uv} \cdot \mathcal{N}, \quad N = r_{vv} \cdot \mathcal{N}$$

are, respectively, the first and second fundamental form coefficients of $r(u, v)$.

When two regular surfaces $S_1(u, v)$ and $S_2(s, w)$ intersect, we say that the intersection is transversal at a point $S_1(u, v) = S_2(s, w) = P$ when their normal vectors $\mathcal{N}_{S_1}(u, v)$ and $\mathcal{N}_{S_2}(s, w)$ are not parallel at that point. If an intersection is transversal at P, the tangential direction of the intersection curve at P is given by

$$t = \frac{\mathcal{N}_{S_1}(u, v) \times \mathcal{N}_{S_2}(s, w)}{| \mathcal{N}_{S_1}(u, v) \times \mathcal{N}_{S_2}(s, w) |}. \quad (12)$$

3 Previous Work

Wu and Andrade [13] proposed a robust algorithm for estimating the osculating circle at the intersection point P from the tangent vector at P, t_P, and the tangent vector at the previous intersection point Q, t_Q. It is robust in the sense that as far as the tangent vectors are determinable the method works. An approximate osculating circle at P is constructed as follows (Figure 2):

Center (C): the intersection of three planes: the plane that contains Q and has t_Q as normal vector, the plane that contains P and has t_P as normal vector, and the plane that contains P and has a normal $t_P \times t_Q$.

Radius (R): the distance between C and P.

Then, the curvature and the normal vector at P are, respectively,

$$\kappa = \frac{1}{R}, \quad n = C - P \quad (13)$$

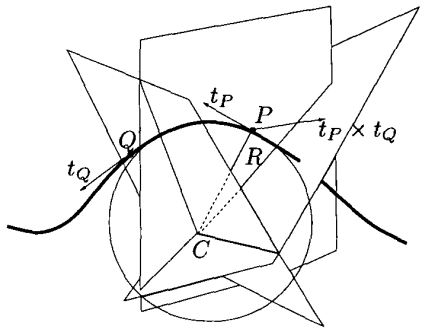

Figure 2: An estimated osculating circle.

Ye and Maekawa [14] derive the local transversal intersection curve properties at a point P from the local properties of the intersecting surfaces, F and G, at that point.

The tangent vector of the intersection curve is given by Eq.(12). If the surfaces are in parametric form, Eq.(10) can be used for computing the surface normal vectors.

The normal vector of the intersection curve is derived as functions of normal curvatures and normal vectors of the two surfaces

$$n = \frac{\kappa_{nF} - \kappa_{nG}\cos\theta}{\sin^2\theta}\mathcal{N}_F + \frac{\kappa_{nG} - \kappa_{nF}\cos\theta}{\sin^2\theta}\mathcal{N}_G, \quad (14)$$

where $\cos\theta = \mathcal{N}_F \cdot \mathcal{N}_G$ and κ_n can be evaluated with help of Eq.(11).

The curvature and its derivative with respect to arc length (notation "$'$") can be computed, respectively, from

$$\kappa = \frac{1}{| \sin\theta |}\sqrt{(\kappa_{nF})^2 + (\kappa_{nG})^2 - 2\kappa_{nF}\kappa_{nG}\cos\theta} \quad (15)$$

$C_1(u) = ((4+\cos(4u))\cos u, (4+\cos(4u))\sin u, \sin(4u)), \Delta u = 0.01$										
u	Point	κ			τ			κ'		
		Exact	Estimated	Ratio	Exact	Estimated	Ratio	Exact	Estimated	Ratio
-9.86	(-3.47,1.61,-0.98)	0.56902	0.56916	0.99975	0.51416	0.53178	0.96687	-0.00685	-0.00371	1.84621
-9.85	(-3.52,1.59,-0.99)	0.56855	0.56878	0.99960	0.49574	0.51295	0.96646	-0.00981	-0.00680	1.44183
-9.84	(-3.57,1.57,-0.99)	0.56793	0.56823	0.99945	0.47778	0.49457	0.96605	-0.01261	-0.00975	1.29293
-9.83	(-3.63,1.55,-0.99)	0.56714	0.56753	0.99931	0.46028	0.47665	0.96566	-0.01525	-0.01255	1.21488
-0.02	(4.99,-0.10,-0.08)	0.51242	0.51255	0.99974	0.12619	0.12716	0.99238	-0.00352	-0.00527	0.66825
-0.01	(4.99,-0.05,-0.04)	0.51225	0.51232	0.99985	0.12562	0.12622	0.99525	-0.00176	-0.00352	0.50076
-0.00	(5.00,-0.00,-0.00)	0.51219	0.51221	0.99996	0.12543	0.12566	0.99817	-0.00000	-0.00176	-
0.01	(4.99,0.05,0.04)	0.51225	0.51221	1.00007	0.12562	0.12548	1.00111	0.00176	-0.00000	-
0.02	(4.99,0.10,0.08)	0.51242	0.51232	1.00018	0.12619	0.12568	1.00405	0.00352	0.00176	1.99821
$C_2(u) = (\cos(u), \sin(u)-1, 2\sin(u/2)), \Delta u = 0.01$										
u	Point	κ			τ			κ'		
		Exact	Estimated	Ratio	Exact	Estimated	Ratio	Exact	Estimated	Ratio
-3.96	(-0.68,-0.27,-1.83)	0.93847	0.93672	1.00186	0.21801	0.22026	0.98981	0.32499	0.32540	0.99874
-3.95	(-0.69,-0.27,-1.83)	0.94197	0.94022	1.00185	0.21593	0.21819	0.98964	0.32475	0.32524	0.99850
-3.94	(-0.69,-0.28,-1.84)	0.94545	0.94371	1.00184	0.21382	0.21610	0.98947	0.32444	0.32500	0.99826
-3.93	(-0.70,-0.29,-1.84)	0.94893	0.94719	1.00183	0.21170	0.21399	0.98930	0.32405	0.32469	0.99802
-0.02	(1.00,-1.02,-0.02)	0.50002	0.50004	0.99996	-0.37499	-0.37498	1.00002	-0.00198	-0.00298	0.66661
-0.01	(1.00,-1.01,-0.01)	0.50000	0.50001	0.99998	-0.37499	-0.37499	1.00001	-0.00099	-0.00198	0.49997
-0.00	(1.00,-1.00,-0.00)	0.50000	0.50000	0.99999	-0.37500	-0.37499	1.00000	-0.00000	-0.00099	-
0.01	(1.00,-0.99,0.01)	0.50000	0.50000	1.00001	-0.37499	-0.37500	0.99999	0.00099	-0.00000	-
0.02	(1.00,-0.98,0.02)	0.50002	0.50001	1.00002	-0.37499	-0.37500	0.99997	0.00198	0.00099	2.00007

Table 1: (1,4) curve and a space curve of degree four.

and

$$\kappa' = c''' \cdot n, \tag{16}$$

where

$$c''' = -\kappa^2 t + \frac{\lambda_{nF} - \lambda_{nG}\cos\theta}{\sin^2\theta}\mathcal{N}_F + \frac{\lambda_{nG} - \lambda_{nF}\cos\theta}{\sin^2\theta}\mathcal{N}_G.$$

λ_n of each surface is a function of its second form coefficients, derivatives, and normal vector $\mathcal{N}$

$$\begin{aligned}\lambda_n &= 3(Lu'v'' + M(u''v' + u'v'') + Nu'v'') + \\ &\quad r_{uuu}\mathcal{N}(u')^3 + 3r_{uuv}\mathcal{N}(u')^2v' + \\ &\quad 3r_{uvv}\mathcal{N}u'(v')^2 + r_{vvv}\mathcal{N}(v')^3.\end{aligned}$$

Note that the derivatives u', u'', v' and v'' can be obtained by solving the following linear systems

$$\begin{cases} Eu' + Fv' &= r_u \cdot t \\ Fu' + Gv' &= r_v \cdot t \end{cases}$$

and

$$\begin{cases} Eu'' + Fv'' &= n \cdot r_u - \frac{E_u}{2}(u')^2 - E_v u'v' - \\ & \quad (F_v - \frac{G_u}{2})(v')^2, \\ Fu'' + Gv'' &= n \cdot r_v - (F_u - \frac{E_v}{2})(u')^2 - G_u u'v' - \\ & \quad \frac{G_v}{2})(u')^2 \end{cases}$$

They also show that the torsion at P can be obtained from

$$\tau = -\frac{1}{\kappa\sin^2\theta}((\lambda_{nF} - \lambda_{nG}\cos\theta)(b\cdot\mathcal{N}_F) + (\lambda_{nG} - \lambda_{nF}\cos\theta)(b\cdot\mathcal{N}_G)), \tag{17}$$

where the curvature κ is given by Eq.(15) and the binormal vector by the expression $t \times n$.

In addition, they derived a set of formulas for computing the differential properties of tangential intersection curves ($\mathcal{N}_\mathcal{F} = \mathcal{N}_\mathcal{G} = \mathcal{N}$). In particular, by setting $\omega = \frac{u'}{v'}$ and $\mu = \frac{v'}{u'}$, the tangent vector may be given by the expressions

$$t = \begin{cases} \frac{\omega F_u + F_v}{|\omega F_u + F_v|}, & \text{if } b_{11} \neq 0 \\ \frac{F_u + \mu F_v}{|F_u + \mu F_v|}, & \text{else if } b_{11} = 0 \text{ and } b_{22} \neq 0, \\ \text{not definable}, & \text{otherwise} \end{cases} \tag{18}$$

where
$b_{11} = a_{11}^2 L_G + 2a_{11}a_{21}M_G + a_{21}^2 N_G - L_F$,
$b_{12} = a_{11}a_{12}L_G + 2(a_{11}a_{22} + a_{21}a_{12})M_G + a_{21}a_{22}^2 N_G - M_F$, and
$b_{22} = a_{12}^2 L_G + 2a_{12}a_{22}M_G + a_{22}^2 N_G - N_F$, with
$a_{11} = \frac{det(F_v, G_w, \mathcal{N})}{\sqrt{E_G G_G - (F_G)^2}}$, $a_{12} = \frac{det(F_v, G_w, \mathcal{N})}{\sqrt{E_G G_G - (F_G)^2}}$,
$a_{21} = \frac{det(G_s, F_u, \mathcal{N})}{\sqrt{E_G G_G - (F_G)^2}}$, and $a_{22} = \frac{det(G_w, F_v, \mathcal{N})}{\sqrt{E_G G_G - (F_G)^2}}$.

$C_1(0.0)$: Exact κ = 0.461539, Exact κ' = -0.00000, Exact torsion= 0.068376						
Δu	κ	$\frac{E_{stimated}}{E_{xact}}(\kappa)$	κ'	$\frac{E_{stimated}}{E_{xact}}(\kappa')$	τ	$\frac{E_{stimated}}{E_{xact}}(\tau)$
0.2	0.466204	0.989992	-0.022913	0.000000	0.119135	0.573938
0.1	0.462718	0.997451	-0.012992	0.000000	0.080786	0.846387
0.01	0.461550	0.999975	-0.001342	0.000000	0.068499	0.998203
0.001	0.461539	1.000000	-0.000134	0.000000	0.068377	0.999982
0.0001	0.461538	1.000000	-0.000013	0.000000	0.068376	1.000000
$C_2(0.0)$: Exact κ = 0.5, Exact κ' = 0.00000, Exact torsion=-0.375000						
Δu	κ	$\frac{E_{stimated}}{E_{xact}}(\kappa)$	κ'	$\frac{E_{stimated}}{E_{xact}}(\kappa')$	τ	$\frac{E_{stimated}}{E_{xact}}(\tau)$
0.2	0.500706	0.998590	-0.020055	0.000000	-0.374626	1.000999
0.1	0.500176	0.999648	-0.009965	0.000000	-0.374907	1.000248
0.01	0.500002	0.999996	-0.000994	0.000000	-0.374999	1.000002
0.001	0.500000	1.000000	-0.000099	0.000000	-0.375000	1.000000
0.0001	0.500000	1.000000	-0.000010	0.000000	-0.375000	1.000000

Table 2: Improvements on problematic points.

4 Our Proposal

Because of the limitation on the computer-based representation, the Eqs.(14–17) only work well when

$$\cos\theta = \mathcal{N}_F \cdot \mathcal{N}_G < 1 - \epsilon. \tag{19}$$

They are numerically unstable when $\sin\theta$ assumes very small values. This means that in the neighborhood of a non-transversal intersection point, wrong values for κ and τ are generated. It leads us to look for a more robust algorithm, even though it delivers approximations instead of exact values.

Similar to the one proposed by Wu and Andrade, our algorithm depends solely on the computability of the tangent vectors of the intersection curve. It is based on the intuitive geometric meaning of the local curve properties expressed by Eqs.(6) and (7).

From Eqs.(12) and (18) one can easily get the tangent vector for transversal as well as tangential intersection point. The normal vector and the curvature can be evaluated by Eq.(13). Knowing the tangent and the normal vectors, the binormal vector is computed from the cross product $b = t \times n$. In this way, a Frenet frame may be defined at any traced intersection point. Eqs.(8) and (9) let us estimate the torsion τ from the deviations of these Frenet frames.

Finally, considering two successive points $\alpha(s - \Delta s)$ and $\alpha(s)$ on the intersection, the derivative of the curvature is simply given by

$$\kappa'(s) = \lim_{\Delta s \to 0} \frac{\kappa(s) - \kappa(s - \Delta s)}{\alpha(s) - \alpha(s - \Delta s)}. \tag{20}$$

We carried out a set of tests in known parametric curves for comparing the estimated values with the exact ones computed from Eqs.(1–3). Table 1 presents the deviations of the estimated values from the exact ones when the Frenet frame varies with u for two curves: a curve of type (1,4) on a torus (it is a helix which turns four times around the torus) and a space curve of degree four.

For simplicity, we considered that Δu is constant, instead of keeping arc length constant. It is worth noting that, except at the points where the sign of the torsion changed, the estimated values are close to the exact ones even with relatively large step sizes. The sign of an estimated torsion also agrees with the sign of the corresponding exact one.

For the problematic points it is expected that smaller step sizes will improve the estimation. Table 2 summarizes the improvements of the estimated values for two problematic points, $C_1(0.0)$ and $C_2(0.0)$, when we reduced the step size Δu. Observe that the smaller is Δu, the closer is the estimated values to the exact ones.

5 Marching Directions

Theoretically we know that the higher the contact order of the intersection curve and a curve along which we step, the more consecutive points they have in common. Hence, one application of our algorithm and the one proposed by Ye and Maekawa [14] is to compute the intersection curve more accurately by stepping in the direction of a curve that has a contact of higher-order with the intersection curve, such as the ones represented by Eqs.(4) and (5). We call the tracing method along Eq.(4) a polynomial one, and along Eq.(5), a helical technique.

The marching direction is used for estimating the next tracing point. Because of the trade-off between efficiency and accuracy, that point is usually not on the intersection curve. Newton iterations are required to improve the accuracy of reached points at each step. It is clear that the near lies the point to the curve [6], the less iterations are necessary to improve its coordinates in relation to the exact intersection curve. Hence, it is interesting to know whether marching along the curves given by Eqs.(4) and (5), which

$F_1(u,v) = (4+\cos(4u))\cos(u) + v(0.00000001 * cos(4u) + 0.99999999 * sin(4u)) * cos(u)$,
$(4+\cos(4u))\sin(u) + v(0.00000001 * cos(4u) + 0.99999999 * sin(4u)) * sin(u)$,
$\sin(4u) + v(0.00000001 * sin(4u) - 0.99999999 * cos(4u))$,
$0 \le u \le 2\pi$ and $-0.5 \le v \le 0.5$
$G_1(s,w) = [(4+\cos(s))\sin(w), (4+\cos(s))\cos(w), \sin(s)]$,
$0 \le s, w \le 2\pi$

Traced Points	Cos(θ)	κ		τ		κ'	
		Ye/Maek.	Wu/Andrade	Ye/Maek.	Our Est.	Ye/Maek.	Our Estimation
(-3.466 1.618 -0.985)	0.9999999	0.544909	0.569099	0.654178	0.518759	-0.0000000001222	-0.0063964385
(-3.475 1.615 -0.986)	0.9999999	0.719336	0.569047	0.747620	0.520539	-0.0000000000593	-0.0052067827
(-3.485 1.611 -0.987)	0.9999999	0.571359	0.568970	0.474029	0.511158	0.0000000000399	-0.0077316070
(-3.494 1.608 -0.988)	0.9999999	infinity	0.568898	-0.00000	0.511021	infinity	-0.0072266777

$F_2(u,v) = (\cos(u), \sin(u) - 1, v)$,
$0 \le u \le 2\pi, -3.0 \le v \le 3.0$,
$G_2(s,w) = (2\cos(s)\cos(w), 2\cos(s)\sin(w), 2\sin(s))$,
$0 \le s \le 2\pi, 0 \le w \le \pi$.

Traced Points	Cos(θ)	κ		τ		κ'	
		Ye/Maek.	Wu/Andrade	Ye/Maek.	Our Est.	Ye/Maek.	Our Estimation
(-0.728 -0.315 1.836)	0.1573029	0.939439	0.93781	0.217445	0.219372	0.316894	0.325097
(-0.722 -0.308 1.840)	0.1539310	0.942687	0.94106	0.215500	0.217445	0.317005	0.324930
(-0.715 -0.301 1.843)	0.1505841	0.945933	0.94431	0.213537	0.215500	0.317039	0.324695
(-0.709 -0.295 1.847)	0.1472627	0.949175	0.94755	0.211556	0.213537	0.316992	0.324390

$F_3(u,v) = (3.5+\cos(3.0u))\cos(u), (3.5+\cos(3.0u))\sin(u), v)$,
$0 \le u \le 2\pi, -1.0 \le v \le 1.0$,
$G_3(s,w) = (3.5+\cos(w))\cos(s), (3.5+\cos(w))\sin(s), \sin(w))$,
$-\pi \le s, w \le 2\pi$.

Traced Points	Cos(θ)	κ		τ		κ'	
		Ye/Maek.	Wu/Andrade	Ye/Maek.	Our Est.	Ye/Maek.	Our Estimation
(1.880 1.988 0.645)	-0.623760	0.485811	0.485389	0.969451	0.975825	0.051255	0.084414
(1.888 1.987 0.651)	-0.618415	0.486647	0.486228	0.963098	0.969450	0.051480	0.083898
(1.896 1.986 0.656)	-0.613077	0.487478	0.487061	0.956767	0.963097	0.051679	0.083360
(1.904 1.985 0.662)	-0.607749	0.488303	0.487889	0.950459	0.956766	0.051851	0.082801

Table 3: Determination of local properties.

require higher-order derivatives, is more efficient than marching along a circle. Moreover, it is worth evaluating the trade-off of our algorithm in relation to the algorithm proposed by Ye and Maekawa [14] in the context of the robust and efficient computation of intersection curves.

In this work we restricted our comparisons to the parametric surfaces. The algorithm proposed by Ye and Maekawa provides an exact computation of the local properties of the intersection curve. However, for the cases where the condition stated by Eq.(19) is not satisfied (i.e. the normal vectors of the intersecting surfaces are almost parallel), the computation of κ, κ', and τ is numerically unstable. One reason comes from the term $\frac{1}{\sin\theta}$ in Eqs.(14–17).

Table 3 presents the local properties of the intersection curve determined by our algorithm and by the algorithm of Ye and Maekawa, as we traced the intersection curve with step size $L = 0.01$.

The first pair of surfaces, $F_1(u,v)$ and $G_1(s,w)$, is almost non-transversal ($\cos\theta \sim 1.0$). In this case our algorithm was able to determine local properties of any point of the intersection curve (Figure 3), whereas the algorithm proposed by Ye and Maekawa could not provide correct data for tracing it (Figure 4). The second pair, $F_2(u,v)$ and $G_2(s,w)$ (Figure 5), and the third pair, $F_3(u,v)$ and $G_3(s,w)$ (Figure 6), have also some non-transversal intersection points. Observe, however, the accuracy of Ye/Maekawa technique in determining local properties for transversal intersection points.

The exact representation of the intersection curve of the first pair of surfaces and the second pair are, respectively, $C_1(u)$ and $C_2(u)$. Note that two lists of points are provided in the Table 1. It is because that we could not be able to get exactly the same sequence of points by applying two different methods for determining local curve properties. We performed our comparisons between the most closest traced points.

Comparing the values in Table 1 with the values in Table 3, we may see that, except in the neighborhood of the

Surfaces	L	Method		#points	1 it	2 it	3 it	4 it	5 it	> 5 it
Ruled Surface/Torus	0.1	Tangential		1371	35.59%	0.00%	0.00%	0.00%	0.00%	64.41%
		Circular		1339	33.46%	0.00%	0.00%	0.00%	0.00%	66.54%
		Polynomial	Our Estimation	1339	33.01%	0.00%	0.00%	0.00%	0.00%	66.99%
			Ye/Maekawa	187	2.54%	0.00%	0.00%	0.00%	0.00%	97.46%
		Helical	Our Estimation	1339	32.49%	0.00%	0.00%	0.00%	0.00%	67.51%
			Ye/Maekawa	655	13.97%	0.00%	0.00%	0.00%	0.00%	86.03%
Torus/Gen. Cylinder	0.1	Tangential		1215	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%
		Circular		1181	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%
		Polynomial	Our Estimation	1181	0.00%	26.08%	73.92%	0.00%	0.00%	0.00%
			Ye/Maekawa	1181	0.00%	94.33%	5.67%	0.00%	0.00%	0.00%
		Helical	Our Estimation	1181	0.00%	32.09%	67.91%	0.00%	0.00%	0.00%
			Ye/Maekawa	1181	0.00%	94.33%	5.67%	0.00%	0.00%	0.00%
Sphere/Cir.Cylinder	0.1	Tangential		1160	0.00%	0.0%	99.83%	0.17%	0.00%	0.00%
		Circular		1098	0.00%	0.00%	99.82%	0.18%	0.00%	0.00%
		Polynomial	Our Estimation	1098	0.00%	1.09%	98.73%	0.18%	0.00%	0.00%
			Ye/Maekawa	1098	0.00%	90.16%	9.66%	0.18%	0.00%	0.00%
		Helical	Our Estimation	1098	0.00%	0.00%	99.82%	0.18%	0.00%	0.00%
			Ye/Maekawa	1098	0.00%	55.74%	44.08%	0.18%	0.00%	0.00%

Table 4: Comparisons of marching directions.

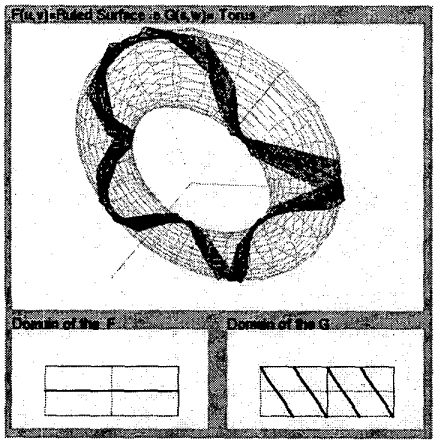

Figure 3: Tracing Ruled Surface/Torus with local properties obtained by our technique .

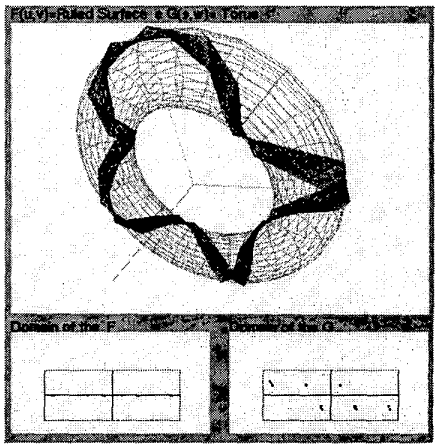

Figure 4: Tracing Ruled Surface/Torus with local properties computed by Ye/Maekawa technique .

non-transversal intersection point, the values computed by the Ye algorithm is more accurate than the ones estimated by our algorithm.

It is expected that, in comparison with methods that trace along curves of degree 2, polynomial and helical techniques require less iterations for improving the coordinates of the the obtained points. For illustrative purpose we present in Table 4 the number of iterations required for tracing the intersection of three given pairs of surfaces.

It is noticeable the superiority of the Ye algorithm in handling transversal intersections that satisfy the condition expressed by Eq.(19). However, when that condition is not fulfilled, marching along curves with a contact of higher order is not possible. Our algorithm does not suffer this shortcoming. Its simplicity favors numerical robustness. At points in the neighborhood of the non-transversal intersection point, it is still able to estimate the local properties with the same precision as at other points.

6 Conclusions

We presented a simple algorithm for estimating some local curve properties and applied them to trace an intersection curve of two parametric surfaces along curves with a contact of third order.

In the cases tested, our algorithm for estimating geo-

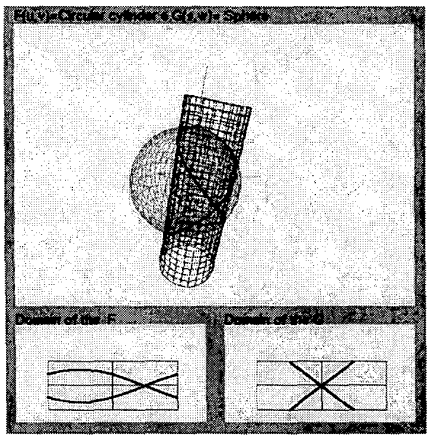

Figure 5: Intersection Circular cylinder/Sphere.

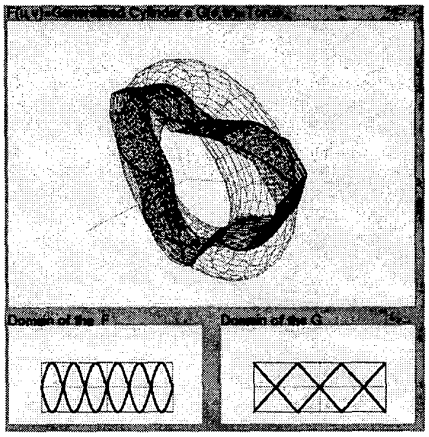

Figure 6: Intersection Generalized Cylinder/Torus.

metric parameters has shown more robust than the Ye algorithm, since it can deal with the non-transversal intersection points that do not satisfy Eq.(19). Some examples were provided to attest this statement.

Additionally, from our numerical experiments we observed that the proposed polynomial and helical marching directions present better performance than the directions proposed in the mentioned work – less iterations were needed to improve the obtained point at each step. However, an accurate analysis of our proposed tracing methods demands a deeper study on topological and geometrical aspects. This discussion will depend on: (1) transversality conditions between the intersecting surfaces and the topological nature of the intersection curves (bifurcation points and their natures, and knotted or unknotted curves for instance), and (2) estimation for total variation of curvature and torsion, including torsion sign discussion.

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] 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. P. Carmo. *Differential geometry of curves and surfaces*. Prentice Hall Inc., New Jersey, 1st. edition, 1976.

[5] T. A. Grandine and F. W. Klein IV. A new approach to the surface intersection problem. *Computer Aided Geometric Design*, 14(2):111–134, February 1997.

[6] J. Hoschek and D. 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.

[7] 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.

[8] M. E. Mortenson. *Geometric Modeling*. John Wiley & Sons, USA, 1st. edition, 1985. ISBN 0-471-88279-8.

[9] G. Müllenheim. On determining start points for a surface/surface intersection algorithm. *Computer Aided Geometric Desgin*, 8:401–408, 1991.

[10] T. W. Sederberg and R. J. Meyers. Loop detection in surface patch intersections. *Computer Aided Geometric Design*, 5:161–171, 1988.

[11] Tz. E. Stoyanov. Marching along surface/surface intersection curves with an adaptative step length. *Computer Aided Geometric Desgin*, 9:485–489, 1992.

[12] D. J. Struik. *Lectures on Classical Differential Geometry*. Dover Publications, Inc., 2nd. edition, 1961.

[13] 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.

[14] X. Ye and T. Maekawa. Differential geometry of intersection curves of two surfaces. *Computer Aided Geometric Desgin*, 16:767–788, 1999.