Pergamon

Comput. & Graphics, Vol. 21, No. 6, pp. 815–824, 1997

Printed in Great Britain
0097–8493/97 $17.00 + 0.00

PII: S0097–8493(97)00060–5

Technical Section

INCLUSION TEST FOR CURVED-EDGE POLYGONS

J. RUIZ DE MIRAS† and F. R. FEITO

Departamento de Informática, Universidad de Jaén, 23071-Jaén, Spain
e-mail: demiras@ujaen.es

Abstract—In this work we present a new algorithm to study the inclusion of points into polygons whose edges are curve segments. It is valid for closed planar polygons whose edges can be straight line segments, conic arcs or cubic Bézier curves. Also it is valid for manifold and non-manifold closed planar polygons with and without holes. Our algorithm is robust and simple, firstly because it avoids the use of equations systems in the inclusion test, and secondly because it solves special cases effectively and homogeneously.

1. INTRODUCTION

The inclusion of points into polygons is a widely studied problem and there are many algorithms proposed for its solution [1–6]. The best known solution considers an infinite half-line that starts from the point to be studied, and counts the number of intersections of the half-line with the polygon edges. If this number is odd, then the point is inside the polygon. Some proposed algorithms [2, 3] count the intersections by the resolution of equation systems. The main problem of these algorithms is the stability [3] and, moreover, these algorithms must take into account many special cases [2], such as the half-line passing through a vertex (the union of two edges), or the half-line overlapping an edge, *etc.* Other algorithms are based on calculating the so-called winding number [5], whose main problem is that they work with trigonometric functions, introducing, therefore, accuracy problems and high cost of computing time for calculating the operations. Besides, the extension of these algorithms for curved polygons is not easy.

Our work avoids all these problems; in our method the use of equation systems and trigonometric functions is not necessary. Besides, our algorithm avoids special cases, and the most complex operation, in the worst implementation, is calculating a 3×3 determinant, similar to that of the 4×4 determinant method [6]. Therefore, the algorithm obtained is simple and robust.

The problem to solve is: given an arbitrary point and a planar curved-edge polygon (curved polygon to simplify) whose edges can be straight line segments, conic sections or cubic Bézier curves, determine whether the point is inside or outside of the curved polygon [see Fig. 1(a)]. The method that we propose is an extension of a previous method [1] that only works with straight edges. The fundamentals of the method consist in the decomposition of the curved polygons in a set of overlapping triangles and closed curved regions [see Fig. 1(b)]. This reduces the problem to considering the inclusion of the point in these pieces (easy to solve) and to merging the individual results by simple algebraic additions.

The next section introduces basic definitions, notations we will use later, and the inclusion test algorithm for polygons with straight edges. Section 3 introduces the concepts "conic region" and "Bézier region", and reviews the inclusion in these regions. Finally, in the last section we study the inclusion problem of non-manifold closed curved polygons with holes. Conclusions and future perspectives are presented in Section 5.

2. BASIC CONCEPTS

Before discussing the algorithm for curved polygons, we review the concepts in Ref. [1] which are the basis of the new algorithm.

Lemma 1. *For any three different points* P_1, P_2 *and* Q, Q *is on the edge* $\overline{P_1P_2}$ *if and only if*:

(a) $\mathrm{sign}([P_1P_2Q]) = 0$;
(b) $(P_2.x - Q.x)(P_1.x - Q.x) + (P_2.y - Q.y)(P_1.y - Q.y) < 0$

where sign(x) *is equal to* 1, −1 *or* 0 *if* x *is positive, negative or zero, respectively, and* $[P_1P_2Q]$ *is the signed area of triangle* P_1P_2Q, *which can be calculated as*

$$\frac{1}{2}\begin{vmatrix} P_1 \cdot x & P_1 \cdot y & 1 \\ P_2 \cdot x & P_2 \cdot y & 1 \\ Q \cdot x & Q \cdot y & 1 \end{vmatrix}$$

Proof. The first condition means that the points P_1, P_2 and Q are collinear, and the second verifies that

† Author for correspondence.

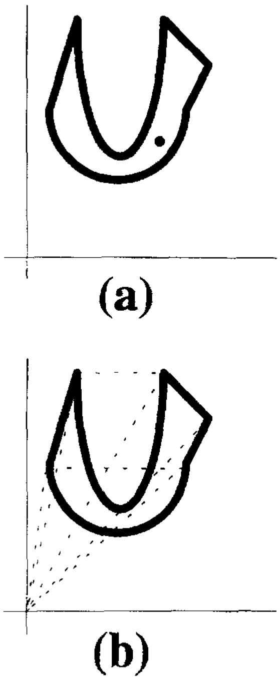


Fig. 1. Problem to study.

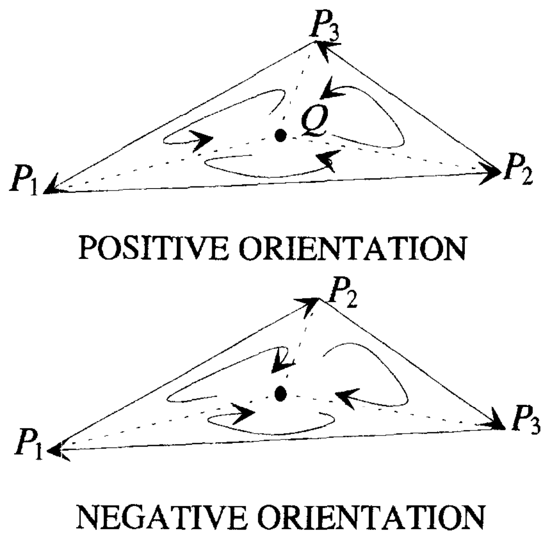


Fig. 2. Triangle inclusion.

the angle formed between the vectors $\overline{QP_1}$ and $\overline{QP_2}$ does not exceed 180°. □

Lemma 2. *Let* $P_1P_2P_3$ *be a triangle and* Q *a point.* Q *is inside* $P_1P_2P_3$ *if, and only if*

$$\mathrm{sign}([P_1P_2P_3]) = \mathrm{sign}([QP_1P_2]) = \mathrm{sign}([QP_2P_3]) = \mathrm{sign}([QP_3P_1])$$

Proof. See Ref. [1]. □

As Fig. 2 shows, since Q is interior to $P_1P_2P_3$, it follows that the orientations (signs of the signed areas) of QP_1P_2, QP_2P_3 and QP_3P_1 are the same as the orientation of $P_1P_2P_3$. If Q is not interior to $P_1P_2P_3$, then there exists at least one triangle with orientation different to $P_1P_2P_3$.

Theorem 1. *Let* $P = e_1, e_2, \ldots, e_n$ *be a general polygon. Let* e_i *denote its ith edge whose vertices are* $e_{i,0}$ *and* $e_{i,1}$. *Let* Q *be any point and let* O *be the origin. Then* Q *is inside* P *if and only if*

$$\sum_{i=1}^{n} \lambda_i = 1,$$

where

$$\lambda_i = \begin{cases} \mathrm{sign}([Oe_i]) & \text{if } Q \in Oe_i, \\ \frac{1}{2}\mathrm{sign}([Oe_i]) & \text{if } Q \in (\overline{Oe_{i,0}} \bigcup \overline{Oe_{i,1}}), \\ 0, & \textit{otherwise} \end{cases}$$

Proof. See Ref. [1]. □

Figure 3 shows four examples of the inclusion test over a non-manifold closed polygon with a hole. The edge e_i is given counter-clockwise orientated if it is an exterior edge of the polygon, and clockwise orientated if it is an interior one, that is, if it is a hole's edge. To determine if a point Q is in a general planar polygon, it is enough to calculate the sum of the signs of the signed areas of all the original triangles that contain Q, determined by the origin and each of the edges e_i of the polygon. If Q is on an original edge common to two (or to an even number if the vertex is non-manifold) original triangles (point Q_1 in Fig. 3), count half a sign for each one. To determine if Q is on the boundary of P we previously check if it is on the edge e_i or not.

3. INCLUSION IN CURVED REGIONS

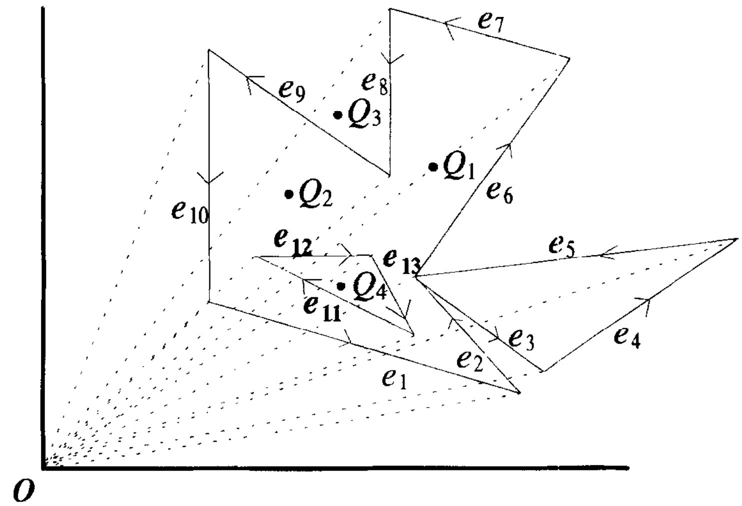
Once the inclusion in polygons has been studied, the next step is to extend this algorithm to deal with curved edges too. The possible curved edges that we consider are conic sections and cubic Bézier curves. But, before studying these particular curves, we provide an intuitive definition about the concept of "curved region".

Definition 1 (curved region). The closed region of the plane delimited both by a curve segment and by the straight line segment joining its ends, and for which it holds that:

(a) the curve segment does not intersect the line segment except for its ends, and
(b) all the points in the region, but the boundary, lie on the same connected component of the curve that contains the curve segment,

is called a *curved region*.

Figures 4 and 5 show various curved regions. While in Fig. 4 all the examples are valid curved



$Q_1 : 0+0+0+0+0+½+½+0 \quad + 0+0+0+0+0 \; = 1$
$Q_2 : 0+0+0+0+0+0 \; +1 \; +(-1)+ \; 1+0+0+0+0 \; = 1$
$Q_3 : 0+0+0+0+0+0 \; +1 \; +(-1)+ \; 0+0+0+0+0 \; = 0$
$Q_4 : 0+0+0+0+0+1 \; +0 \; +0 \quad + \; 0+0+0+0+(-1) = 0$

Fig. 3. Point-in-polygon test.

regions, Fig. 5(e) does not hold the first condition in Definition 1, and Fig. 5(b–d) do not hold the second one. Afterwards, we give methods for dividing these curve segments getting valid curved regions.

Similar to the case of triangles, each curved region has an associate sign of value 1 or -1. This sign gives information about the orientation of the curved region. If a curved region has a sign of value 1 (-1), then going through the curve segment from its initial point to the end point, and coming back through the line segment to the initial point, is done counter-clockwise (clockwise). Figures 4 and 5 show various curved regions with different signs.

The first step in developing the algorithm is to establish the mechanism for deciding the point inclusion in two types of curved regions, "conic regions" and "Bézier regions". After getting this first goal, we will be in a good position to discuss the general algorithm.

3.1. *Inclusion in a "conic region"*

Definition 2 (conic arc). The conic arc A, defined by the tuple (f, V_0, V_1, s), is the collection of points in the plane which belongs to the conic curve defined by $f(x,y)=0$, beginning at the point V_0 and finishing at the point V_1; considering the arc that lies to the right (left) of vector $\overline{V_0V_1}$, going from V_0 to V_1, if s is 1 (-1).

Note: it is supposed that the coefficients in f are given in their general implicit form and in such a way that $f(x,y)<0$ for the points that lie in any concavity of the conic.

Although (f, V_0, V_1) is sufficient to define a conic arc if the coefficients in f define a parabola or a hyperbola, it is not sufficient for ellipses because there are two possible arcs beginning at V_0 and finishing at V_1 [see Fig. 4(c)]. In this case the value of s decides which one is correct.

We can associate to each conic arc $A(f, V_0, V_1, s)$ a curved region defined by the curve segment A and the line segment $\overline{V_0V_1}$. We assign the term *conic region* $\mathbf{R_A}$ to this curved region, and its sign, $\mathbf{R_A}$.sign, is the value of s. Figure 4 shows examples of conic regions. The next lemma gives a method for calculating the inclusion in conic regions.

Lemma 3. *Let* $\mathbf{R_A}$ *be a conic region defined by the conic arc* A(f, V_0, V_1, s), *and let* Q *be any point. Then* Q $\in \mathbf{R_A}$ *if and only if any of the following conditions is verified*:

(a)

$$f(Q)\leqslant 0 \wedge (\operatorname{sign}([V_0QV_1]) = s \vee \operatorname{sign}([V_0QV_1])=0);$$

(b)

$$f=0 \text{ is a hyperbola} \vee \text{(a)} \wedge \left(\operatorname{sign}(d_f(Q)) = \operatorname{sign}(d_f(V_0))\right),$$

where d_f *is the implicit equation of the directrix of the hyperbola* $f=0$.

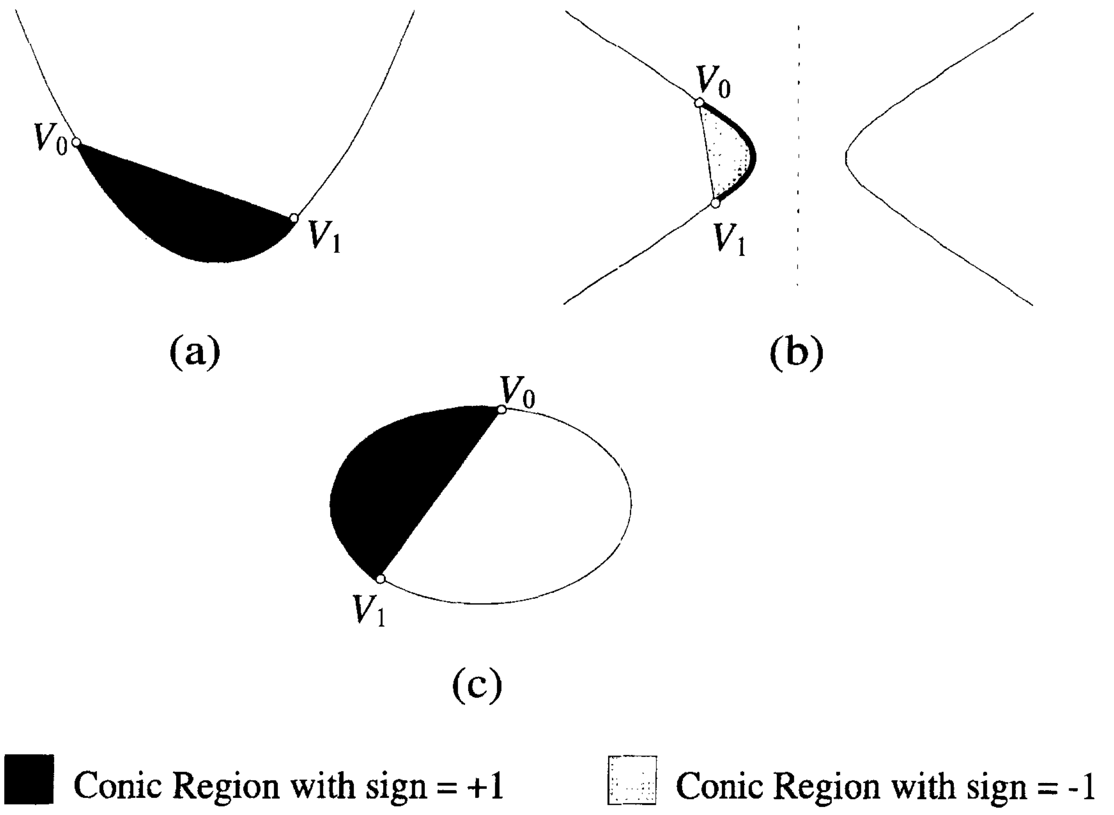


Fig. 4. Conic arcs and conic regions.

Proof. The first term in condition (a) establishes that Q must be on the conic or in any concavity of it, but Q could accomplish this and be located out of R_A. This situation is corrected with the second term in (a) establishing that the orientation of Q with regards to $\overline{V_0V_1}$ must be equal to the orientation of points in conic arc A with regard to $\overline{V_0V_1}$ (given by s); or Q is on the line segment $\overline{V_0V_1}$ ($\text{sign}([V_0QV_1]) = 0$). If $f = 0$ is a hyperbola, besides the condition (a), Q and V_0 (or V_1) must lie in the same side with regard to the hyperbola directrix. This is checked by substituting both points into the directrix implicit equation and verifying that both of them have the same sign. □

3.2. *Inclusion in a "Bézier region"*

In this work, we only consider cubic Bézier curves, *i.e.* curves defined with four control points. Various cubic Bézier curves and their control points are shown in Fig. 5(a–e).

Mathematically, a parametric Bézier curve is defined as follows:

$$B(t) = \sum_{i=0}^{n} C_i J_{n,i}(t), \quad 0 \leqslant t \leqslant 1 \tag{1}$$

where C_i are control points and

$$J_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

Extending Equation (1) with $n = 3$, we obtain the well-known cubic polynomial:

$$B(t) = C_0(1-t)^3 + 3C_1t(1-t)^2 + 3C_2t^2(1-t) + C_3t^3, \quad 0 \leqslant t \leqslant 1 \tag{2}$$

Although Fig. 5(a–e) show basic cubic Bézier curves types, we are going to use only those that verify the two conditions in Definition 1 (valid curve segments). The fundamental property of this type of Bézier curve is that there is not a singular point (self-intersection) contained in the convex hull [see Fig. 5(b–d)], and the interior of line segment $\overline{C_0C_3}$ does not intersect the Bézier curve [see Fig. 5(e)]. To check if a Bézier curve, with no singular point inside its convex hull, intersects $\overline{C_0C_3}$ we can use the following lemma.

Lemma 4. *Let* B(t) *be a Bézier curve defined by Equation (2) that has no singular point inside its convex hull. Then* B(t) *intersects the interior of* $\overline{C_0C_3}$ *if and only if*

$$\text{sign}([C_0C_1C_3]) \neq \text{sign}([C_0C_2C_3])$$

with both signs different from zero.

Proof. Since the derivatives at the initial and final points of the Bézier curve are determined by the

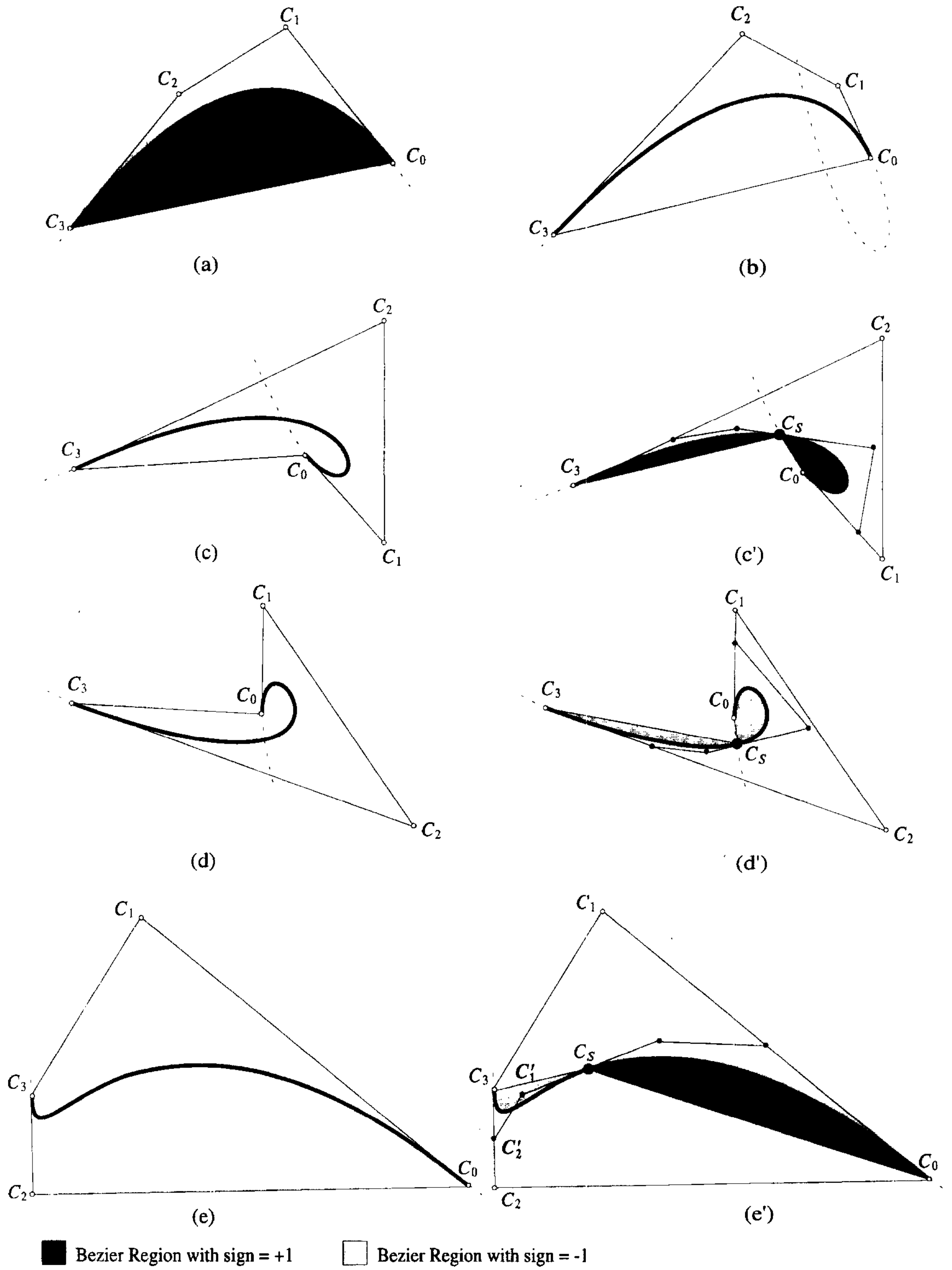


Fig. 5. Cubic Bézier curves and valid Bézier regions.

second and third control points respectively, the only way in which the curve can intersect $\overline{C_0C_3}$ is that C_1 and C_2 be in different sides of the line defined by $\overline{C_0C_3}$, which is equivalent to saying that triangles $C_0C_1C_3$ and $C_0C_2C_3$ have different orientations, which is verified by the lemma condition. □

A Bézier curve similar to the one shown in Fig. 5(e) (without singular point but intersecting $\overline{C_0C_3}$) can be decomposed in valid curve segments by dividing it by its parametric mid-point (using the de Casteljau algorithm [7]) and checking if both obtained curves do not verify the condition of Lemma 4; if any verifies it (rare cases), then the process is repeated recursively [see Fig. 5(e')]. If a Bézier curve contains a singular point inside its convex hull, then it must be decomposed to obtain valid curve segments. We discuss the singular point problem in the following subsection.

3.3. *Locating the singular point in cubic Bézier curves*

It is possible to know if there is a singular point inside the Bézier's convex hull without calculating it. But, since it is often necessary to calculate the singular point to divide the curve, we calculate the singular point as follows:

Substituting in Equation (2) the control points coordinates values, we obtain two equations, $B_x(t)$ and $B_y(t)$, one for each coordinate. If there is a singular point, then there are also two distinct values, t_1 and t_2, such that $B_x(t_1) = B_x(t_2)$ and $B_y(t_1) = B_y(t_2)$. The solution does not exist if $U.y = 0$ or if $U.xT.y - T.xU.y = 0$, where $U = -P_0 + 3(P_1 - P_2) + P_3$ and $T = 3(P_0 - 2P_1 + P_2)$; that is, the curve does not have a singular point. Besides, if t_1 and t_2 are not inside the interval (0,1), we can say that the singular point is not inside the convex hull.

If the singular point is inside the convex hull, then the curve must be divided by its singular point into two curves having no singular point inside their convex hulls [see Fig. 5(c′–d′)]. This operation can be easily done with the de Casteljau recursive algorithm [7].

The next step is to treat Bézier curves as conics, in the sense that we can decide the relative position of a point Q with regards to the Bézier curve. For this purpose, we need to express the Bézier curve in its implicit form, which is discussed in the next section.

Note: We assume that when we refer to a Bézier curve it is a valid curve segment (no singular point into the convex hull and not intersecting the line segment joining its ends).

3.4. *Implicitization of cubic Bézier curves*

There are many studies about the parametric function implicitization problem. We have chosen to solve this problem using Bézout's resultant method [8, 9].

The Bézout method gives the resultant of

$$f(x) = \sum_{i=0}^{m} a_i x^i \quad \text{and} \quad \sum_{j=0}^{n} b_i x^i$$

as a determinant, whose order is $l = \max(m, n)$. In our case:

- $m = n = 3$, $l = 3$;
- $f(t) = at^3 + bt^2 + ct + (d - x)$, Equation (2) coordinate X;
- $g(t) = a't^3 + b't^2 + c't + (d' - y)$, Equation (2) coordinate Y;
- $a_0 = (d - x)$, $a_1 = c$, $a_2 = b$, $a_3 = a$;
- $b_0 = (d' - y)$, $b_1 = c'$, $b_2 = b'$, $b_3 = a'$;
- $x = t$.

Therefore, the Bézout matrix for $f(t)$ and $g(t)$ is

$$\mathbf{M}(f(t), g(t)) = \begin{pmatrix} \begin{vmatrix} a_3 & a_2 \\ b_3 & b_2 \end{vmatrix} & & \begin{vmatrix} a_3 & a_1 \\ b_3 & b_1 \end{vmatrix} & & \begin{vmatrix} a_3 & a_0 \\ b_3 & b_0 \end{vmatrix} \\ \begin{vmatrix} a_3 & a_1 \\ b_3 & b_1 \end{vmatrix} & \begin{vmatrix} a_2 & a_1 \\ b_2 & b_1 \end{vmatrix} & + & \begin{vmatrix} a_3 & a_0 \\ b_3 & b_0 \end{vmatrix} & \begin{vmatrix} a_2 & a_0 \\ b_2 & b_0 \end{vmatrix} \\ \begin{vmatrix} a_3 & a_0 \\ b_3 & b_0 \end{vmatrix} & & \begin{vmatrix} a_2 & a_0 \\ b_2 & b_0 \end{vmatrix} & & \begin{vmatrix} a_1 & a_0 \\ b_1 & b_0 \end{vmatrix} \end{pmatrix} \tag{3}$$

To obtain the implicit form of the Bézier curve, we only have to calculate the determinant of this matrix. But coming back to the beginning of our statement, we are not interested in calculating the implicit Bézier equation, we only want to know the relative position of a point Q with regards to the Bézier curve. Previously and similarly to the conic case, we introduce the Bézier Region concept.

We can associate a curved region to each valid Bézier curve $B(t)$ defined by Equation (2), formed by the points limited by the Bézier curve segment $B(t)$ and the line segment $\overline{C_0C_3}$. This region of the plane is called Bézier Region $\mathbf{R_B}$, and its sign, $\mathbf{R_B}$.sign, is calculated as sign($[C_0C_1C_3]$) if this value is different from zero; else the sign of the Bézier region is the value sign($[C_0C_2C_3]$).

Definition 3 (sign of a point with respect to a Bézier curve). The sign of a point Q with respect to a Bézier curve $B(t)$ (whose parametric equations, expressed in the form (2), are B_x and B_y), denoted $\mathrm{Sign}_\mathbf{B}(Q)$, is calculated as

$$\mathrm{Sign}_\mathbf{B}(Q) = \begin{cases} 1 & \text{if } |\mathbf{M}(B_x - Q.x,\ B_y - Q.y)| > 0, \\ -1 & \text{if } |\mathbf{M}(B_x - Q.x,\ B_y - Q.y)| < 0, \\ 0, & \text{otherwise} \end{cases}$$

$\mathrm{Sign}_\mathbf{B}(Q)$ is equivalent to calculating the sign of the result of substituting the coordinates of Q in the implicit equation of the Bézier curve $B(t)$. Finally, the inclusion of a point Q in the Bézier region $\mathbf{R_B}$ is established with the following lemma:

Lemma 5. *Let $B(t)$ be a valid Bézier curve defined by the control points C_0, C_1, C_2 and C_3, let $\mathbf{R_B}$ be the Bézier region associated with $B(t)$ and let Q be any point. Then*

$$Q \in \mathbf{R}_B \Leftrightarrow Q \in \mathrm{convex}_{\mathrm{hull_B}} \wedge (\mathrm{Sign}_\mathbf{B}(Q) = 0 \vee \mathrm{Sign}_\mathbf{B}(Q) \neq \mathrm{Sign}_\mathbf{B}(C_{\mathrm{out}}))$$

where C_{out} is one of the vertices of the convex hull of $B(t)$ different from C_0 and C_3.

Proof. Obviously, if Q is inside $\mathbf{R_B}$, then it needs to be in the convex hull of $B(t)$. If $\text{Sign}_B(Q)$ is equal to zero, then Q is on the Bézier curve and, therefore, Q is inside $\mathbf{R_B}$. C_{out} is selected in such a way that it is not inside $\mathbf{R_B}$ and since there is not a singular point inside the convex hull, it follows that the Bézier curve splits the convex hull into two zones, one with positive sign and the other with negative one. Therefore, the only possibility for Q to be inside $\mathbf{R_B}$ is that its sign with respect to $B(t)$ be the opposite to that of C_{out}. □

Figure 6 shows various examples of the lemma. C_{out} can be easily selected as the point C_i ($i=1,2$) such that $\text{sign}([C_{i-1}C_iC_{i+1}]) = \mathbf{R_B}.\text{sign}$.

Once the study of conic and Bézier region inclusion is done, we pass to detail the general algorithm for curved polygons.

4. INCLUSION TEST FOR CURVED-EDGE POLYGONS

Now the basic definitions for the development of the new algorithm are introduced.

Definition 4 (curved(-edge) polygon). A curved polygon is a general polygon which has at least one edge defined by conic arcs or valid Bézier curves.

We can represent a curved polygon as follows:

$$\text{CP} = (e_1, e_2, \ldots, e_n) \cup (c_1, c_2, \ldots, c_n)$$

where e_i are the polygon edges whose vertices are $e_{i,0}$ and $e_{i,1}$, and

$$c_i = \begin{cases} \text{null} & \text{if } e_i \text{ has a straight edge,} \\ (f_i, e_{i,0}e_{i,1}s_i) & \text{if } e_i \text{ delimits a conic region,} \\ (C_{i,0}C_{i,1}C_{i,2}, C_{i,3}, s_i) & \text{if } e_i \text{ delimits a valid Bezier region,} \\ & \text{where } C_{i,0} = e_{i,0} \text{ and } C_{i,3} = e_{i,1} \end{cases} \tag{4}$$

where s_i is the value of the sign of the conic or Bézier region defined by the curve segment c_i and the line segment e_i, calculated as shown in previous sections, and $C_{i,j}$ ($j=0,1,2,3$) are the control points of the Bézier curve edge c_i.

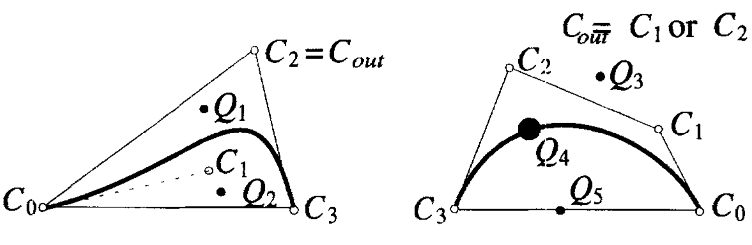
Figure 7 shows an example of a non-manifold closed curved polygon with a hole and its representation. This curved polygon has four curved regions, two conic ones and two Bézier ones (in order to be clear we do not paint the $C_{i,j}$). Similar to polygons in Theorem 1, the exterior edges are given orientated counter-clockwise, and the interior ones (defining holes) are given orientated clockwise. Note that edges e_1, e_2, e_7 and e_{11} do not belong to the curved polygon, but they are necessary in order to define the curved regions in the polygon.

Taking into account that $\text{int}(x)$ represents the interior of the object x in the respective topology to the dimension of x, $\text{sign}(Oe_i) \equiv \text{sign}([Oe_{i,1}e_{i,2}])$ and $\text{sign}(c) \equiv \mathbf{R}_c.\text{sign} \equiv$ (value s) in tuple that defines c, we now enunciate the inclusion in a curved polygons theorem. We briefly give an intuitive approach to the theorem after the proof.

Theorem 2. *Let* $\text{CP}=(\text{e}_1, \text{e}_2, \ldots, \text{e}_n)\cup(\text{c}_1, \text{c}_2, \ldots, \text{c}_n)$ *be a curved polygon and let* Q *be any point different from the coordinate origin, and not on the* CP*'s boundary. Then*

$$Q \in \text{int}(\text{CP}) \Leftrightarrow \sum_{i=1}^{n}(\lambda_i + \beta_i) = 1,$$

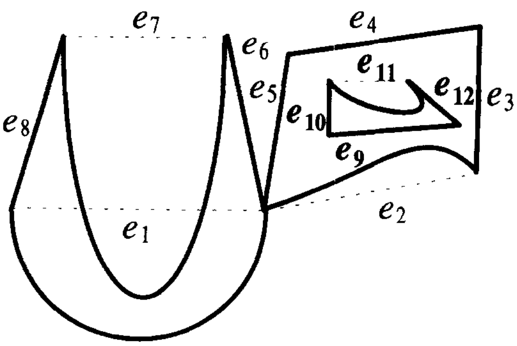
where



$R_B.sign = -1$
$sign([C_0C_1C_2]) = 1$
$sign([C_1C_2C_3]) = -1$
$Sign_B(Q_1) = Sign_B(C_{out})$
$Sign_B(Q_2) \neq Sign_B(C_{out})$

$R_B.sign = 1$
$sign([C_0C_1C_2]) = 1$
$sign([C_1C_2C_3]) = 1$
$Q_3 \notin convex\_hull_B$
$Sign_B(Q_4) = 0$
$Sign_B(Q_5) \neq Sign_B(C_{out})$

Fig. 6. Inclusion test for Bézier regions.



$$c_1 = (f_1, e_{1,0}, e_{1,1}, +1),\ c_2 = (e_{2,0}, C_{2,1}, C_{2,2}, e_{2,1}, -1)$$
$$c_7 = (f_7, e_{7,0}, e_{7,1}, -1),\ c_{11} = (e_{11,0}, C_{11,1}, C_{11,2}, e_{11,1}, +1)$$
$$CP = (e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}) \cup$$
$$(c_1, c_2, \textbf{null}, \textbf{null}, \textbf{null}, \textbf{null}, c_7, \textbf{null}, \textbf{null}, \textbf{null}, c_{11}, \textbf{null})$$

Fig. 7. Curved polygon.

$$\lambda_i = \begin{cases} \mathrm{sign}(Oe_i) & \text{if } Q \in \mathrm{int}(Oe_i), \\ \frac{1}{2}\mathrm{sign}(Oe_i) & \text{if } (Q \in (\mathrm{int}(\overline{Oe_{i,0}}) \cup \mathrm{int}(\overline{Oe_{i,1}}))) \\ & \text{or } (Q \in \mathrm{int}(e_i) \text{ and } c_i \neq \text{null}), \\ 0, & \text{otherwise} \end{cases}$$

and

$$\beta_i = \begin{cases} \mathrm{sign}(c_i) & \text{if } Q \in \mathrm{int}(\mathbf{R}_{c_i}) \text{ and } c_i \neq \text{null}, \\ \frac{1}{2}\mathrm{sign}(c_i) & \text{if } Q \in \mathrm{int}(e_i) \text{ and } c_i \neq \text{null}, \\ 0, & \text{otherwise} \end{cases}$$

If Q is the coordinate origin, we apply a translation to both the curved polygon and Q, and then we can use the theorem. In order to see if Q is on the curved polygon boundary we previously check if Q is on the polygon edge (straight or curved) by using Lemmas 1, 3 and 5 (the last two restricted to the boundary).

Proof. If $Q \in \mathrm{int(CP)}$, then the half-line that starts at the origin, passes through Q and goes to infinity, has, from Q, an odd number of transitions from the inner to the outer of the polygon or *vice versa*. We are going to prove that the theorem expression is equal to one in this case and that it is equal to zero if the number of transitions is even, *i.e.* the term on the right of the expression is equivalent to calculating the parity of the number of transitions from the inner to the outer of the polygon or vice versa.

For each positive original triangle Oe_i (c_i=null) that contains Q, there is an intersection point between the half-line and the edge e_i of this triangle. This necessarily involves that the half-line in this point goes from the inner of CP to its outer. Just the opposite occurs when the triangle has negative orientation, *i.e.* the half-line goes in this case from the outer to the inner of CP. We can, therefore, conclude that calculating the parity of the number of transitions is equivalent to adding the signs of the triangles that contain Q, but if the half-line passes through a vertex, then it is necessary to consider the following situations:

(a) The vertex is common to two triangles of the same sign, in this case $\lambda_i = \frac{1}{2}$ or $\lambda_i = -\frac{1}{2}$ for both triangles, *i.e.* it is equivalent to counting only one transition (with sign 1 or -1 according to the transition direction).
(b) The vertex is common to a positive triangle and to a negative one. In this case in the intersection point there is not an orientation change from inner to outer or *vice versa* and, therefore, it must not be counted as a transition; this is so because the sum of the λ_i values of both triangles is equal to 0 ($\frac{1}{2} + (-\frac{1}{2})$ or $(-\frac{1}{2}) + \frac{1}{2}$).
(c) The vertex is non-manifold; the situation is the same as the one for cases (a) and (b), but the number of triangles that shared that vertex is even (see Fig. 9 point Q_2).

If the half-line overlaps any straight edge, then the edge's vertices are considered as the intersection points, and then, this case is equivalent to a previous one. If the half-line passes through any curved edge (c_i), then there are two cases:

(a) $Q \in \mathrm{int}(\mathbf{R}_{c_i})$; there are two possibilities:
  (a.1) the half-line from Q cuts c_i once [see Fig. 8(a–b)]. This is true if the half-line does not cut the edge e_i, and this means that Q is not inside Oe_i, obtaining $\lambda_i = 0$ and $\beta_i = \mathrm{sign}(c_i)$ (or $\lambda_i = \beta_i = \frac{1}{2}\mathrm{sign}(c_i)$ if Q is on the edge e_i [see point Q' in Fig. 8(a)], *i.e.* it is counted as only one transition (with the proper sign).
  (a.2) the half-line from Q cuts c_i zero or two times [see points Q and Q' in Fig. 8(c)].

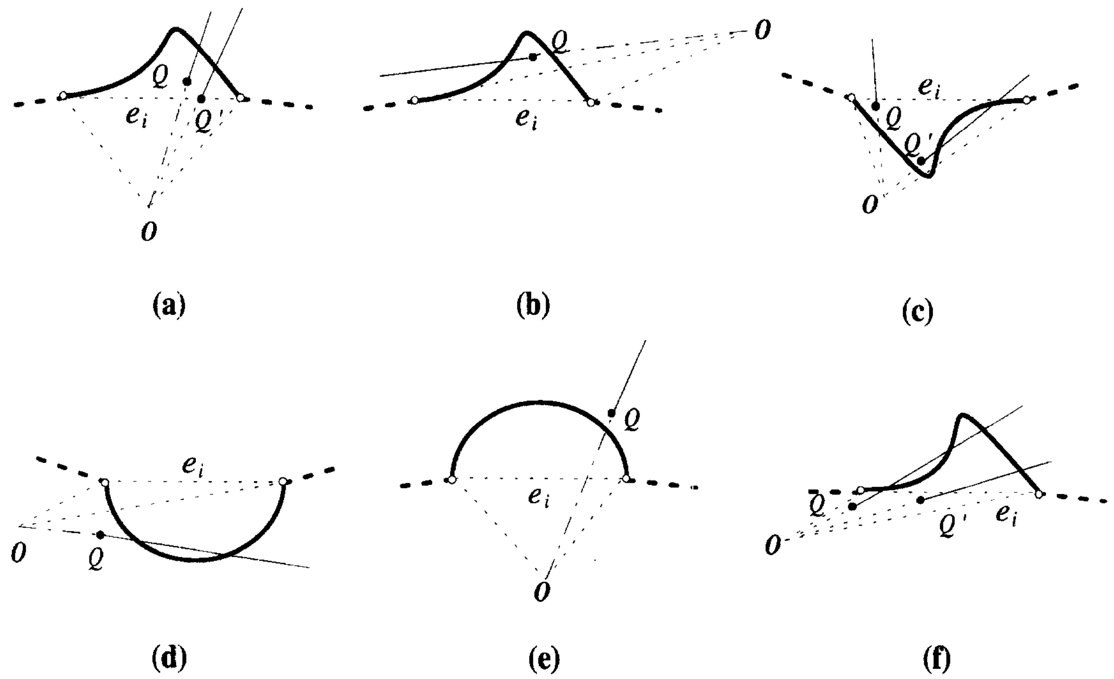
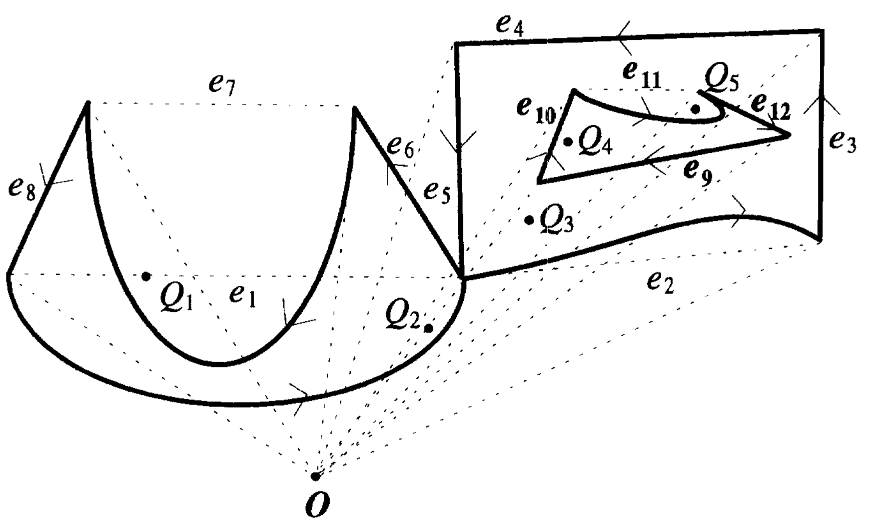


Fig. 8. Theorem 2 proof cases.

This occurs if the half-line cuts the edge e_i, obtaining $\lambda_i = -\beta_i$, which is equivalent to counting an even number (zero or two) of transitions. Note that having two transitions is only possible if c_i is a Bézier curve.

(b) $Q \notin \mathrm{int}(\mathbf{R}_{c_i})$; there are two possibilities:

(b.1) the half-line from Q cuts the curve c_i in zero or two points [see Fig. 8(d–e)]. This necessarily implies that the half-line does not cut the edge e_i and, therefore, Q is not inside Oe_i. Since Q is not inside



Q_1: (-½, ½), (0, 0), 0, 0, 0, 0, (0, -1), 1, 0, 0, (0, 0), 0 = 0
Q_2: (-½, 1), (-½, 0), 0, 1, -½, ½, (0, 0), 0, -½, -½, (1, 0), 0 = 1
Q_3: (0, 0), (0, 0), 0, 1, 0, 0, (0, 0), 0, 1, 0, (-1, 0), 0 = 1
Q_4: (0, 0), (0, 0), 0, 1, 0, 0, (0, 0), 0, 0, 0, (-1, 0), 0 = 0
Q_5: (0, 0), (0, 0), 0, 1, 0, 0, (0, 0), 0, 0, 0, (0, 1), -1 = 1

Fig. 9. Curved polygon inclusion.

```
In:    - Curved polygon CP defined as (4).
       - Point Q.
Out:   - 0 if Q ∉ CP,
       - 2 if Q ∈ boundary(CP),
       - 1 if Q ∈ int(CP).

   tot = 0
   IF (Q ∈ boundary(CP)) THEN RETURN(2)
   FOR i =1 TO n DO
      /** calculating λ_i **/
      IF (Q ∈ int(Oe_i)) THEN
         tot = tot + sign(Oe_i)
      ELSE IF ((Q∈int(Oe_i,0) OR Q∈int(Oe_i,1)) OR
               (c_i≠null AND Q ∈ int(e_i))) THEN
         tot = tot + ½sign(Oe_i)
      /** calculating β_i **/
      IF (c_i ≠ null AND Q ∈ int(R_ci)) THEN
         tot = tot + sign(c_i)
      ELSE IF (c_i ≠ null AND Q ∈ int(e_i)) THEN
         tot = tot + ½sign(c_i)
   END_FOR
   RETURN(tot == 1)
END.
```

Fig. 10. Algorithm.

$\mathbf{R}_{c_i}(\beta_i = 0)$ or inside Oe_i ($\lambda_i = 0$), the contribution is zero, which is equivalent to considering an even number (zero or two) of transitions.

(b.2) the half-line from Q cuts the curve c_i in one or three points [see points Q and Q' in Fig. 8(f)]. In this case we must count only one transition (with the proper sign), which is equivalent to counting an odd number of transitions. This happens because the half-line cuts the edge e_i; therefore Q is inside Oe_i, obtaining $\lambda_i = \text{sign}(Oe_i)$ and $\beta_i = 0$. Similar to (a.2), having three transitions is only possible if c_i is a Bézier curve.

Although the theorem expression seems hard to read, basically it does not differ much from the expression in Theorem 1. Similarly, the process is based on the addition of the signs of the signed areas of the original triangles Oe_i and the signs of the curved regions that contain Q. Now, for each curved region we sum two values, one for the curved region (β_i) and one for the original triangle formed by the origin and the line segment e_i that delimited it (λ_i). Similarly to Theorem 1, if Q is on an original edge common to two (or to a even number if the vertex is non-manifold) original triangles, we must count half a sign for each one. If Q is on an edge e_i that delimited a curved region, as Q belongs both to the curved region $\mathbf{R}_{c_i}$ and to the original triangle Oe_i, we must count half a sign for each one again.

Figure 9 shows the application of the theorem for five different points over the same curved polygon. Each value corresponds to the contribution of an edge e_i (value of λ_i), beginning at e_1, and if e_i has a curve c_i associated to it, we give the pair (λ_i, β_i).

The test can be easily implemented evaluating the formula of the theorem. Figure 10 details, in pseudocode, the implementation of the algorithm.

5. CONCLUSIONS AND FUTURE WORK

A robust and simple algorithm for determining the inclusion of points into curved-edge polygons has been presented. It is robust because it is not necessary to solve equation systems and it is simple because the special cases and the functions it needs are simple. Its validity has been proved formally, and the way it works has been described. We think that the importance of the algorithm is not only these properties, but that it is founded on concepts [10] that can be easily generalised to higher dimensions [11]. In this way, we are working on the extension of the algorithm to 3-D.

Acknowledgements—We want to express our thanks to the referees for their comments and suggestions. We are also indebted to Francisco de Asís Conde for his help and valuable comments.

REFERENCES

1. Feito, F. R., Torres, J. C. and Ureña, A., Orientation, simplicity and inclusion test for planar polygons. *Computers & Graphics*, 1995, **19,** 595–600.
2. Ying, J.-Q. and Sugie, N., A point-inclusion algorithm for a domain with boundary composed of algebraic curve segments. *Systems and Computers in Japan*, 1991, **22,** 1823–1829.
3. Preparata, F. P. and Shamos, M. I., *Computational Geometry: An Introduction.* Springer-Verlag, New York, 1985.
4. Haines, E., Point in polygon strategies. In *Graphics Gems IV*, ed. P. S. Heckbert. Academic Press, New York, 1994.
5. Kalay, Y. E., Determining the spatial containment of a point in general polyhedra. *Computer Vision, Graphics and Image Processing*, 1982, **19,** 303–334.
6. Yamaguchi, F., Niizeki, M. and Fukunaga, H., Two robust, point-in-polygon tests based on the 4×4 determinant method. In *Proceedings of ASME Design Technical Conferences—16th Design Automation conference*, Vol. 23, 1990, pp. 89–95.
7. Farin, G., *Curves and Surfaces for Computer Aided Geometric Design.* Academic Press, San Diego, CA, 1993.
8. Sederberg, T. W., Anderson, D. C. and Goldman, R. N., Implicit representation of parametric curves and surfaces. *Computer Vision, Graphics, and Image Processing*, 1984, **28,** 72–84.
9. Wee, C. E.and Goldman, R. N., Elimination and bivariate resultants. *IEEE Computer Graphics and Applications*, 1995, **15,** 69–77.
10. Feito, F. R. and Torres, J. C., Boundary representation of polyhedral heterogeneous solids in the context of a object algebra. *The Visual Computer*, 1997, **13,** 64–77.
11. Feito, F. R. and Torres, J. C., Inclusion test for general polyhedra. *Computers & Graphics*, 1997, **21,** 23–30.