Journal of Computational Mathematics, Vol.12, No.2, 1994, 138–146.

THE INTERSECTION OF A TRIANGULAR BÉZIER PATCH AND A PLANE*

Chen Fa-lai[1)]
(*University of Science and Technology of China, Hefei, Anhui, China*)

Jernej Kozak[2)]
(*Department of Mathematics University of Ljubljana, Slovenija*)

Abstract

In this paper, the problem of finding the intersection of a triangular Bézier patch and a plane is studied. For the degree that one frequently encounters in practice, i.e. $n = 2, 3$, an efficient and reliable algorithm is obtained, and computational steps are presented.

1. Introduction

In this paper, the problem of finding the intersection of a triangular Bézier patch and a plane is considered. Such a problem quite frequently comes up in practical CAGD computations. Of course, in practice one has to work with pp-surface rather than with a single patch. However, the basic algorithm has to deal with a single patch on its own. Valuable information from the neighbouring patches can be available only in the simplest case, i.e. when a plane intersects the boundary of the patch.

The intersection problem in a particular form arises often when one works with algebraically rather than parametrically represented planar curves.

Similar problems were considered in [1] and [4], where the intersection of a bicubic Bézier patch and a plane was considered.

Let T be a given triangle. The most natural way to express the parametric Bézier surface S on T is to write it in the barycentric form

$$S := S(F) := S^n(F) := \sum_{i+j+k=n} F_{ijk} B^n_{ijk}, \tag{1.1}$$

where

$$F_{ijk} := (x_{ijk}, y_{ijk}, z_{ijk})$$

* Received April 22, 1993.

1) Supported by NSF and SF of State Education Commission of China.

2) Supported by the Ministry of Science and Technology of Slovenija.

are given control points. The Bernstein basic functions B^n_{ijk} are defined as

$$B^n_{ijk}(q) := B^n_{ijk}(u,v,w) := \frac{n!}{i!j!k!}u^i v^j w^k$$

with (u, v, w),

$$0 \le u, v, w \le 1, \quad u + v + w = 1$$

being the barycentric coordinates of a point $q \in T$. Let $\wp$ be a plane, given by the equation

$$\wp : \quad ax + by + cz + d = 0. \tag{1.2}$$

Let us denote $S_\wp := S \bigcap \wp$. Take (1.1) componentwise into (1.2). This shows that $S^n(F)(q) = S^n(F)(u,v,w) \in S_\wp$ iff the Bernstein polynomial

$$B_n(f) := \sum_{i+j+k=n} f_{ijk}B^n_{ijk}$$

has a zero at $q = (u, v, w)$. Here the coefficients f_{ijk} are given as

$$f_{ijk} := ax_{ijk} + by_{ijk} + cz_{ijk} + d.$$

There is a very natural way of searching for zeros of a Bernstein polynomial defined on a triangle. Let T_1, T_2, T_3 denote the vertices of T, and $T_4 = (1 - s)T_2 + sT_3$ for some fixed s, $0 \le s \le 1$. Let us denote

$$Q_s := B_n(f)|_{T_1T_4}.$$

Then by [2] Q_s is a Bernstein polynomial of one variable, with coefficients being polynomials in s. To be precise, recall $B^n_i(t) := \binom{n}{i}t^i(1-t)^{n-i}$. Then

$$Q_s(t) = \sum_{i=0}^{n} a_i(s)B^n_i(t) \tag{1.3}$$

with

$$a_i(s) := \sum_{j=0}^{i} f_{n-i,i-j,j}B^i_j(s). \tag{1.4}$$

The idea of a general algorithm is quite clear. Move s from 0 to 1, and at each step find the zeros of Q_s. The information from the previous step can be taken as good starting approximation. Of course, at each step some zeros might disappear, and some others might be introduced.

If one is looking for an efficient and reliable algorithm, it is of crucial importance to know in advance if $S_\wp$ is actually not empty. If it is not, some information on positions of zeros is also necessary.

It is easy to verify numerically the following sufficient condition: $S_\wp$ is not empty if $B_n(f)$ has a zero on the border of T. This condition is also necessary for $n = 1$. The main result of this paper gives necessary and sufficient conditions also for $n = 2, 3$: