Journal of Computational Mathematics, Vol.14, No.3, 1996, 237–248.

ON COMPUTING ZEROS OF A BIVARIATE BERNSTEIN POLYNOMIAL*

F.L. Chen[1)]
(*Department of Mathematics, University of Science and Technology of China, Hefei, China*)
J. Kozak[2)]
(*Department of Mathematics, University of Ljubljana, 1000 Ljubljana, Slovenija*)

Abstract

In this paper, the problem of computing zeros of a general degree bivariate Bernstein polynomial is considered. An efficient and robust algorithm is presented that takes into full account particular properties of the function considered. The algorithm works for rectangular as well as triangular domains. The outlined procedure can also be applied for the computation of the intersection of a Bézier patch and a plane as well as in the determination of an algebraic curve restricted to a compact domain. In particular, singular points of the algebraic curve are reliably detected.

1. Introduction

In [6] and [4], the problem of finding the intersection of a cubic Bézier patch and a plane was considered. [6] considered a rectangular, and [4] a triangular patch. Since the Bernstein operator $B_n : f \mapsto B_n(f)$ preserves linear functions, the problem was simplified to the computation of zeros of a bivariate Bernstein polynomial $B_n(f)$. Both papers produced simple and efficient computational algorithms. It is based upon the following idea: determine the points where inside the support the topology of zeros of $B_n(f)$ changes. This was done by restricting the bivariate polynomial to a particular line direction, and determine these points from the fact that this restriction is a cubic polynomial. The zero branches were then separately computed between each pair of exceptional points.

A similar problem can be traced to [7] in a slightly different context, this time for general n. Let

$$p(x,y) = 0, \;\; p(x,y) := \sum_{i=0}^{n} \sum_{j=0}^{n-i} p_{ij} x^i y^j \tag{1.1}$$

be the equation that defines a given planar algebraic curve. Suppose that one is interested in computing set of points $\{(x,y)\}$ that satisfies (1.1) in a given triangle $T \in \mathbb{R}^2$. The recipe in [7] suggests to rewrite p as a Bézier patch over triangle, and look for

* Received July 8, 1994.

1) The Project Supported by National Natural Science Fundation of China and Science Fundation of National Educational Committee of China.

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

its zeros. Though no special algorithm for this particular problem was suggested, some help was given by the theorem 1 ([7]): if the corresponding Bézier net is strictly increasing along mesh lines in one direction, any line in this direction will cross the algebraic curve at most once.

In this paper, we extend the algorithm for computing zeros of the Bernstein polynomial presented in [4] to the general degree case. As it turns out rather unexpectedly, the general algorithm is as simple as its cubic counterpart though general degree algebraic equations admit no radical solutions. As already pointed out, the outlined procedure solves also the Bézier patch-plane intersection problem as well as the problem of computation of algebraic curves restricted to compact domains $\Omega \in \mathbb{R}^2$. It should also be mentioned that the given algorithm is computationally superior to the methods for computing algebraic functions that one encounters in standard mathematical packages. In particular, the comparison with the ImplicitPlot procedure used in the Mathematica package ([1 , p. 127]) was tested.

For the sake of simplicity, we shall consider only the triangle case. It is easy to see that the basic steps work out for both the rectangular and triangular support. Thus it is obvious that the algorithm can be simply transformed to handle the rectangular case. We shall demonstrate this just by computational examples.

Let us recall some notation and basic facts. Let T be a given nondegenerate triangle. The most natural way to express the Bernstein polynomial on a triangle is to write it in the barycentric form. The Bernstein basic functions B^n_{ijk} in this case are defined as

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

with

$$\beta := \beta_T : \mathbb{R}^3 \to \mathbb{R}^2$$

being the (invertible) barycentric map. The Bernstein polynomial of a function $f : \mathbb{R}^2 \to \mathbb{R}$ reads as

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

where $f_{ijk} := f(\beta_T^{-1}(\frac{i}{n}, \frac{j}{n}, \frac{k}{n}))$ are given coefficients.

The key step of the algorithm is to reduce two-dimensional problem to one dimension. Let T_1, T_2, T_3 denote the vertices of T. Choose fixed s, $0 \le s \le 1$, and somewhat arbitrary $T_4 = (1-s)T_2 + sT_3$. Then by [3]

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

is a Bernstein polynomial of one variable, with coefficients being polynomials in s. As already computed in [4]

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

$$a_i(s) := \sum_{j=0}^{i} f_{n-i,i-j,j} B^i_j(s) \qquad (1.2)$$