Zbornik radova Filozofskog fakulteta u Nišu
Serija Matematika 6 (1992), 19–25
FILOMAT–20, Niš, September 26–28, 1991

ON LEVEL SETS OF BERNSTEIN-BÉZIER OPERATORS

LJUBIŠA M. KOCIĆ AND DUŠAN M. MILOŠEVIĆ

ABSTRACT. *The problem of visualization of Bernstein-Bézier polynomial operators defined on the set of continuous functions $f : T \to R$ where T is a triangular domain from R^2, is considered. A method of level-lines is suggested as a satisfactory solution. The corresponding numerical algorithm is described and illustrated with several examples.*

1. Introduction. Let the operator $B : F \to \Phi$ map an arbitrary nonempty set F in the set Φ of functions $\varphi : D(\subset R^d) \to R$. Then, the collection of subsets $\{L_j\}_{j\in J}$ from D, such that $n \in L_j$ implies $\varphi(u) = C_j$, where $\{C_j\}$ is an increasing real sequence, is called the level-sets of the operator B on the domain D. Tipically, the index set J is the set of natural numbers.

The problem of determining the level-sets of polynomial operators, i.e. when F is a polynomial of dimension d is one of easier problems. In this case, the Bernstein-Bézier operator on the simplex domain, seems to be a natural approach. It turns that even such simple operator may have the level-sets which is hard to describe exactly. In this case, even for $d = 2$ or 3 the numerical solution is inavoidable. In the same time, two particular cases of two- and tree-dimensional operators are the most important in applications, so the problem of making an efficient algorithm for extracting level-lines ($d = 2$) or level-surfaces ($d = 3$), from the control data, attracts attention and becomes very current [1], [2], [3], [10].

The subject of this paper is to construct the level-lines (also called contour lines) of Bernstein-Bézier operator defined on an arbitrary triangular domain $T = (T_1, T_2, T_3)$ by

$$B_n(f,t) = \sum_{i+j+k=n} b^n_{ijk}(t) f_{ijk}, \qquad t = (u,v,w) \in T, \tag{1}$$

where $b^n_{ijk}(t) = \binom{n}{ijk} u^i v^j w^k$ are Bernstein-Bézier basis polynomials of barycentric coordinates u, v, w of the point t with respect to the triangular domain T and $f_{ijk} = f(\frac{i}{n}, \frac{j}{n}, \frac{k}{n})$, $i + j + k = n$, are $\frac{1}{2}(n+1)(n+2)$ values of the function f, defined on T, that control the polynomial B_n. The role of the operator B_n, in theoretical as well as in practical considerations is well known, [3]. As far as the applications is concerning, it is enough to mention Computer Aided Geometric Design and Data Visualization. In both topics, the operator B_n is applied for construction the Bernstein-Bézier triangular patches and their compositions. These

AMS MSC (1980) : 41A36, 65D15
Supported by the Science Fund of Serbia, grant number 0401A, through Matematički institut.

applications assume very precise representation of the particular patch at the given graphical medium.

According to the current literature, there are tree basic methods for patch representation: 1. Colored model, 2. Wire-frame model and 3. Map of level-lines.

The wire-frame method uses the visualization of the perspective look of a representative set of lines on the surface being displayed. The most frequently, these lines are images of some regular mesh in the domain of the function. The colored model obtains by the wire-frame model, during the process called rendering. It looks very nice, but the value of such a model, from the point of view of the scientific data conveying, is not very high. According to the authors' opinion, the third model, level-line model is at least of equal value as the first one and, in some cases it is even superior. For example, the monotonicity, convexity, continuity, differentiability, the existence of saddle points, locations of extrema and gradient intensity are some elements that can be read off from the level-line map more easier then from a wire-frame or colored model. The advantage of the first two models is that they can represent a surface in its "natural" viewing and the visual information obtained from these models is the syntetic one. On the other hand, level-lines give mainly an analytic information. Besides, the method of level-lines can be used as a specific wire-frame model, by simple showing them in perspective projection. An additional importance of the level-line model hides in the possibility of using the similar technique for visualization of $4D$ surfaces [1], [2]. Level-lines are broadly applied in the cartography, meteorology, crystollography, geology, medicine (computer tomography) etc. For an application in the optimization theory see [5].

2. Algorithm. The algorithm is based on the improved "implicit" method [4], [9] combined with a special method of searching for an initial starting values. In fact, we are solving the equation

$$B_n(f,t) = C, \qquad t \in T, \tag{2}$$

where B_n, defined by (1), is a polynomial in barycentric coordinates

$$u = \frac{1}{A(T)} A(t, T_1, T_3), \quad v = \frac{1}{A(T)} A(t, T_1, T_2), \quad w = 1 - u - v,$$

where $A(T) = A(T_1, T_2, T_3)$ is the area of the triangle with vertices at T_1, T_2 and T_3. Every nondegenerate triangle T can be affinely transformed into the unit triangle $T_0 = ((0,0),(0,1),(1,0))$, which simplifies the mathematics without lost of generality. In this case, the Descartes coordinates are given by the linear system

$$u = 1 - x - y, \quad v = x, \quad w = y,$$

so that the basis polynomials at the triangular domain T_0 are given by

$$b^n_{ijk}(x,y) = \binom{n}{ijk}(1-x-y)^i x^j y^k, \quad (x,y) \in T_0, \tag{3}$$

where $\binom{n}{ijk} = \frac{n!}{i!j!k!}$, $i+j+k=n$. So, (2) gets the form

$$\sum_{i+j+k=n} \binom{n}{ijk}(1-x-y)^i x^j y^k f_{i,j,k} = C, \quad (x,y) \in T_0, \tag{4}$$

or

$$B(x,y) = C, \quad (x,y) \in T_0. \tag{5}$$

Equation (5) gives an implicit relationship between x and y from T_0. The problem is construction of the locus of point (x,y) that satisfies (5) for a given C. The nature of this problem is much more particular then of the problem considered in [4] and [9]. First of all, the function $B(x,y)$ in (5), is a Bernstein-Bézier polynomial which is known to satisfy the simple recursive relation. Secondly, the domain that contains the level-lines is known. It is the triangular domain T_0. Finally, using the convex hull property

$$B(x,y) \in \operatorname{conv}\{f_{ijk}\}_{i+j+k=n},$$

it follows that (5) has no solutions for $C \notin [m_1, m_2]$, where $m_1 = \min\{f_{ijk}\}$, $m_2 = \max\{f_{ijk}\}$. Tipically, $C \in \{C_r\}$, $C_r = m_1 + \frac{r}{M}(m_2 - m_1)$, $r = 0, \cdots, M$, where M is a natural number. The bigger the m, the more solutions the equation (5) will have, and the number of level-lines will be greater.

The used method is an improved version of the method from [4]. It is based on reduction of (5) on the initial value problem

$$y' = -\frac{B_x(x,y)}{B_y(x,y)}, \quad (x_0,y_0) \in T_0, \tag{6}$$

B_x and B_y being partial derivatives of the function B. If $B_y(x,y) = 0$, for some (x,y), the reciprocal problem

$$x' = -\frac{B_y(x,y)}{B_x(x,y)}, \quad (x_0,y_0) \in T_0, \tag{7}$$

is considered. Note that the polynomial has no singular points, i.e. $B_x(x,y)$ and $B_y(x,y)$ can not vanish at the same time. This makes use of the known numerical algorithms for solving differential equation (6) and (7) possible. Our solution involves the Runge-Kutta method of the fourth order.

During the realization, two problems arose: 1. Evaluation of derivatives B_x and B_y in efficient and numerically stable way; 2. Estimation of the initial value (x_0, y_0).

The first problem is solved by the using of de Casteljau algorithm

$$\begin{aligned} P^0_{ijk}(t) &= f_{ijk}, \quad i+j+k = n \\ P^r_{ijk}(t) &= uP^{r-1}_{i+1,j,k}(t) + vP^{r-1}_{i,j+1,k}(t) + wP^{r-1}_{i,j,k+1}(t), \\ r &= 1, \cdots, n, \qquad i+j+k = n-r, \end{aligned}$$

and the formula for $m-$th directional derivative, in the direction $l = (l_1, l_2, l_3)$

$$\frac{\partial^m}{\partial l^m} B_n(f,t) = \frac{n!}{(n-m)!} \sum_{i+j+k=m} P^{n-r}_{ijk}(t) b^n_{ijk}(l), \ t \in T,$$

which in the case of the first order, reduces on [3]

$$\frac{\partial}{\partial l} B_n(f,t) = n(l_1 P_{100}^{n-1} + l_2 P_{010}^{n} + l_3 P_{001}^{n-1}).$$

Specifically, if the directions are parallel with $x-$axes and $y-$axes, one gets

$$B_x(x,y) = n(P_{001}^{n-1} - P_{100}^{n-1}); \quad B_y(x,y) = n(P_{010}^{n-1} - P_{100}^{n-1}).$$

As far as the estimation of the initial point $t_0 = (x_0, y_0)$ is concerning, two cases occur: a) $t_0 \in \partial T_0$ – the initial point lies on the contour of the triangle domain T_0; b) $t_0 \in \mathrm{int}(T_0)$ – the initial point is inside the triangle.

Determination of the point t_0 performs in these two cases in two different ways. In the first one, the roots of the algebraic equation

$$B_n(f,t) - C = B_n(f - C, t) = 0, \quad t = (u,v,w) \in T_0,$$

are determined for $u = 0$, $v = 0$ *and* $w = 0$, which means that each side of the triangle T_0 yields one equation. So, the problem reduces on the location of the zeroes of the Bernstein polynomial

$$\beta_n(x) = \sum_{i=0}^{n} P_i \binom{n}{i} x^i (1-x)^{n-i}, \quad x \in [0,1],$$

which performs in two stages. Firstly, the zeroes are isolated using the subdivision algorithm with the middle subdivision point, which applies recursively as far as the root is located within the given tolerance ϵ. Here, the variation diminishing property and the convex hull property are used. In the second stage, the root is corrected by the modified regula falsi algorithm [8]. In this case, the contour of the triangle T_0 is intersected by the level-line twice at least.

In the case that the initial point t_0 is inside T_0, the whole level-line lies inside the triangle T_0 too, and the method of scanning along the directions parallel to the sides of the triangle is applied. For example, one sets $x = a$, and then solves the equations

$$B_n(a,y) - C = 0, \quad (a,y) \in T_0,$$

numerically. Suppose that $y = y_0$ is the satisfactory approximate solution. Then the point $(x_0, y_0) = (a, y_0)$ is taken to be an initial point.

3. Examples. The algorithm is tested through many examples and three of them will be presented here. The arrangement of the control points of $n-$th order Bernstein-Bézier polynomial, is accepted to be

$$\begin{matrix} P_{00n} & & \\ \vdots & \ddots & \\ P_{n00} & \cdots & P_{0n0} \end{matrix}$$

The step in the Runge-Kutta method in all examples is $h = 0.01$, which ensures the level-lines to be smooth enough width the relatively short computer execution time

Example 1. For the control points

$$\begin{matrix} 60 & & & & \\ 20 & 56 & & & \\ 62 & 31 & 82 & & \\ 89 & 23 & 43 & 45 & \\ 12 & 37 & 26 & 63 & 34 \end{matrix},$$

the perspective projection of the graph of the corresponding 4-th degree Bernstein-Bézier polynomial is given in Figure 1.A, while the corresponding level-line map is given in Figure 1.B. As it can be seen, the level-line map faithfully reflects the global properties of the triangular patch from the Figure 1.A such as its domains of monotonicity, extremal and saddle points. The complete running time (including displaying) at 20 MHz PC-386 computer is 37.51 sec.

Example 2. The control points

$$\begin{matrix} 0 & & & & & \\ 0 & 0 & & & & \\ 0 & 30 & 0 & & & \\ 0 & 30 & 30 & 0 & & \\ 0 & 30 & 30 & 30 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix},$$

defines the polynomial of order $n = 5$, width global maximum in the triangular domain T (Figure 2.A). The information about maxima as well as the behaviour of the graph on the sides of the triangle T, and even the gradient can be easily read off, from the map given in Figure 2.B. The running time is shown to be 156.87 sec.

Example 3. In this example we follow the behaviour of the level-line map in depending of altering one control point. Figure 3.A shows the map of the level-lines of the polynomial patch given by the control points that are given in the left matrix below, while Figure 3.B displays the level-lines of the polynomial defined by the control points in the right matrix

$$\begin{matrix} 0 & & & & & \\ 50 & 0 & & & & \\ 0 & 0 & 0 & & & \\ 0 & 0 & 100 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & \end{matrix} \qquad \begin{matrix} 0 & & & & & \\ 100 & 0 & & & & \\ 0 & 0 & 0 & & & \\ 0 & 0 & 100 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & \end{matrix}.$$

At the Figure 3.A one of the level-lines has a cusp (a non-differentiobility point) which happens to exist for the 4-th degree patch. In Figure 3.B, an isolated maxima occurs, being circumscribed by an oval curve in the middle of the triangle T. The executing time is 32.19 sec, for Figure 3.A, and 39.88 sec, for Figure 3.B.

4. Conclusion and open questions. The algorithm being explained is rather different from other known algorithms [3], [6], [7], [10]. Algorithms from [6] and [7] uses proximations of the initial patch of n-th order, by the patch of $(n-1)$-th order, using the degree reduction and subdivision algorithm. It is not quite clear if the degree reduction always payed off.

In [3], an efficient algorithm is proposed, but only for quadratic polynomials. This method is further developed in [10], now based on the rational Bezićr segments.

The algorithm presented here, constructs the level-lines fastly and efficiently, with an error which can be made arbitrarily small by adjusting the step in the Runge-Kutta method. Then, it can be applied on the polynomial of an arbitrary order as well as on the surface obtained by the composition of triangular fragments.

Questions, like error analysis and adjusting the length of the step in the Runge-Kutta method are the subjects of the next paper, which is in preparation.

References

[1] R.E.Barnhill, *A Survey of the Representation and Design of Surfaces*, IEEE Computer Graphics and applications, 3 (1983), no. 7, 9–16.

[2] R.E.Barnhill, *Surfaces in computer aided geometric design: A survey with new results*, 2 (1985), no. 1–3, 1–17.

[3] G. Farin, *Triangular Bernstein-Bézier patches*, CAGD 3 (1986), 83–127.

[4] Lj.M. Kocić, *Korišćenje diferencijalnih jednačina za nalaženje grafika implicitnih funkcija*, Diplomski rad, Elektronski fakultet Niš 1975.

[5] Lj.M. Kocić, *A graphical method for separating extrema of implicit function*, Wiss. Z. TH Ilmenau 35 (1989), no.6, 161–164.

[6] C. Petersen, *Adaptive contouring of three dimensional surfaces*, Computer Aided Geometric Design 1 (1984), 61–74.

[7] C. Petersen, B. Piper, A. Worsey, *Adaptive contouring of a trivariate interpolant*, G.Farin. ed., Geometric Modeling, SIAM, Philadelphia, PA (1986).

[8] T. W. Sederberg and S. R. Parry, *Comparison of three curve intersection algorithms*, Computer–aided design, 18 (1986), no. 1, 58–63.

[9] D.DJ.Tošić, Lj.M. Kocić, *O jednom metodu za crtanje ekvipotencijalnih linija u ravni*, Elektrostatika '84, Niš, 1984, 25.1–25.5.

[10] A.J.Worsey, G. Farin, *Conturing a bivariate quadratic polynomial over triangle*, CAGD, 7 (1990) no. 1–4, 337–352.

Departement of Mathematics
Faculty of Electronic Engineering
p.o.Box 73, 18000 Niš
Yugoslavia

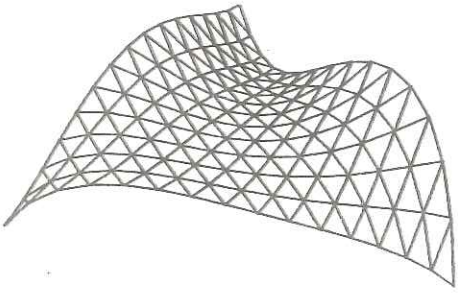

Fig. 1.A

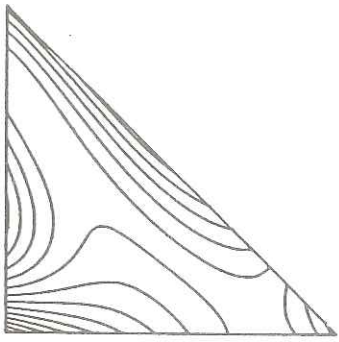

Fig. 1.B

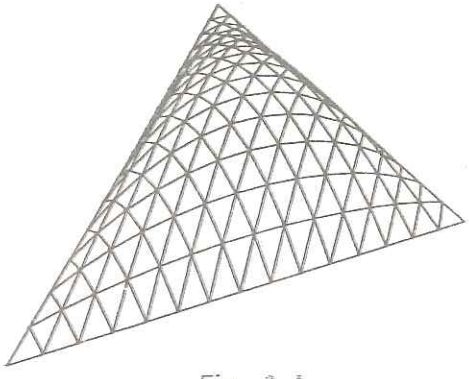

Fig. 2.A

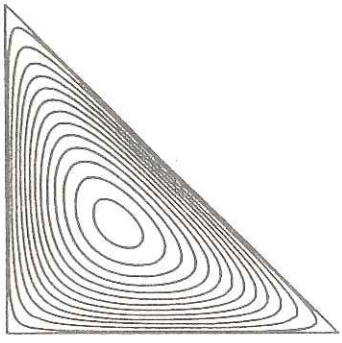

Fig. 2.B

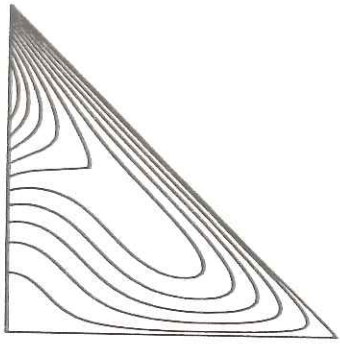

Fig. 3.A

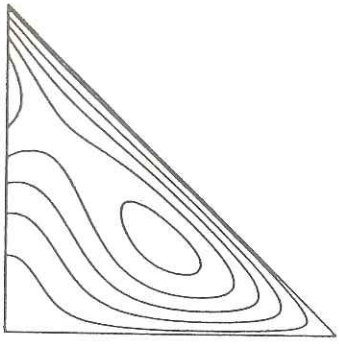

Fig. 3.B