BÉZIER FITTING TO ALMOST OVAL GEAR DEVICES

Karol Gajda [a], Piotr Krawiec [b], Adam Marlewski [c]

[a] Institute of Mathematics, Poznan University of Technology, Poland
[b] Chair of Basics of Machine Design, Poznan University of Technology, Poland
[c] Institute of Mathematics, Poznan University of Technology, Poland

[a]karol.gajda@put.poznan.pl, [b] piotr.krawiec@put.poznan.pl, [c] adam.marlewski@put.poznan.pl

ABSTRACT
Bézier curves are widely applied in computer aided design. In the paper first we shortly say what these curves are and how they are constructed, with examples added to. Next we present their application to model almost oval gear devices.

Keywords: Bernstein polynomials, Bézier curves

1. BERNSTEIN POLYNOMIALS AND BÉZIER CURVES

In 1886 Karl Weierstrass proved that any continuous function on a closed and bounded interval can be uniformly approximated on that interval by polynomial to any degree of accuracy. This result, known as Weierstrass first approximation theorem, has been later shown in many ways. In 1912 Sergei Natanovich Bernstein presented the constructive proof where there are introduced polynomials known as Bernstein polynomials. A **Bernstein polynomial**, $B_{f,n}$, of the function $f \in C_{<0,\,1>}$ and of degree n is defined as follows

$$B_{f,n}(t) := \sum_{k=0}^{s} f(\frac{k}{s}) \cdot p_{s,k}(t)\,, \qquad (1)$$

where

$$p_{s,k}(t) := \binom{s}{k} \cdot t^k \cdot (1-t)^{s-k}\,. \qquad (2)$$

The polynomial $p_{s,k}$ is called k-th **basic Bernstein polynomial**, and the set $B = [\, p_{s,0}, p_{s,1}, ..., p_{s,s}\,]$ is called s-th (**standard**) **Bernstein base**. The standard book discussing their properties still remains (Lorentz 1953). Bernstein polynomials and its generalizations are still investigated both in theoretical and applied aspects, and find new applications in various areas, see e.g. (Chen 2000, Ding and Zhang 2003, Hong and Mitchell 2007, John 2007, Kowalski 2006, Madi 2004).

Polynomials $p_{s,k}$ appeared no later than in 1713 when it was issued the book *Ars conjectandi*. Its author, Jacob Bernoulli (1654-1705), described there so-called **binomial distribution**, later known also as **Bernoulli distribution**. This is a discrete probability distribution, which takes values 1 and 0 with the success probability t and value 0 with the failure probability $1-t$, resp. Then the probability to gain k times in s trials is equal $p_{s,k}(t)$. For this reason Bernstein's proof is known also as a probabilistic one.

Let $c = (c_0, c_1, ..., c_s)$ be a sequence of numbers. **Bernstein polynomial generated by** this **sequence** is the polynomial defined by the formula

$$B_c(t) := \sum_{k=0}^{s} c_k \cdot p_{s,k}(t)\,,\; 0 \le t \le 1\,, \qquad (3)$$

so it is the Cauchy product, $B_c = B{\cdot}c$, of the Bernstein base B and the column vector $c = [\, c_0, c_1, ..., c_s\,]^{\mathrm{T}}$ corresponding to c. A number s is called a **potential degree** of this polynomial.

The definition of Bernstein polynomial can be extended by substituting a vector c by $((s+1), n)$-matrix. This matrix denoted by P, $P = [\, P_{j,k}\,]_{\,j=0,\,1,\,...,\,s;\;k=1,\,2,\,...,\,n}$, we set (again applying Cauchy multiplication) $B_P = B{\cdot}P$, so at each point t there holds $B_P(t) = B(t){\cdot}P$. Just defined vector B_P is called a **Bernstein (polynomial) set generated by matrix** P.

With $n = 2$ this set is composed of $n = 2$ polynomials. If $P_{j,1} = t_j = j/(s+1)$ and $P_{j,2} = c_j$, then $B_P(t) = [\, t, B_c(t)]$, so the set B_P is simply a parametric representation of the Bernstein polynomial which is explicitly described by the explicit formula $y = B_c(t)$. This is called a **standard Bernstein approximation**.

For arbitrary values of two-column matrix P the elements of the vector B_P are polynomials, the first one is determined by the first column of P, and the second one by the 2nd column of P, in consequence B_P is a parametric equation of a flat curve. Analogously, for $n = 3$ the set B_P generated by three-column real matrix P is a parametric representation of a curve in the space $\boldsymbol{R}^3$. Usually the parameter is denoted by t, the spaces $\boldsymbol{R}^2$ and $\boldsymbol{R}^3$ are equipped in orthocartesian systems Oxy and $Oxyz$, resp. Lines of the matrix P are the coordinates of the points. The graph of the vector B_P, i.e. the set $\{\, B_P(t) : t \in \boldsymbol{R}\, \}$, is called **Beziér curve** and we say that this curve is generated by the matrix P, or is determined by so-called **control points**, i.e. the points which coordinates are given by lines of P. Referring to this interpretation we can call any line of the matrix P as its point. The polygonal line joining control points as they are listed in P, is called a **Bézier polygon**, or an **initial contour of Bernstein approximation**.

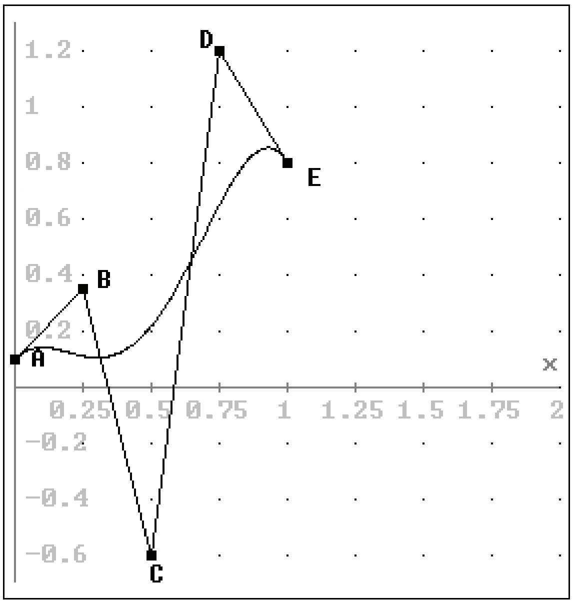

Figure 1: Bézier polygon and curve determined by the sequence $ABCDE$, where $A = (0, 0.1)$, $B = (0.25, 0.35)$, $C = (0.5, –0.6)$, $D = (0.75, 1.2)$ and $E = (1, 0.8)$.

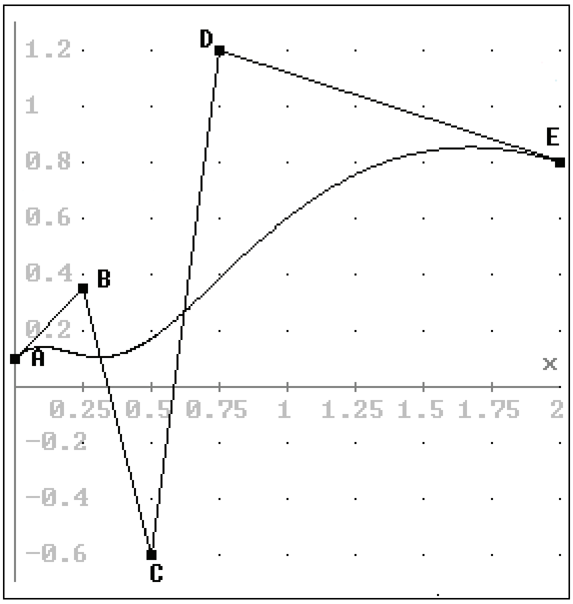

Figure 2: Bézier polygon and curve determined by the sequence $ABCDE$, where $A = (0, 0.1)$, $B = (0.25, 0.35)$, $C = (0.5, –0.6)$, $D =(0.75, 1.2)$ and $E = (2, 0.8)$.

Example 1. The vector $c = [\ 0.1, 0.35, –0.6, 1.2\ 2\ ]^T$, the sequence $ABCDE = (\ A = (0, 0.1),\ \ B = (0.25, 0.35),\ \ C = (0.5, –0.6),\ \ D = (0.75, 1.2),\ \ E = (1, 0.8))$ and the matrix

$$P = \begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} = \begin{bmatrix} 0 & 0.1 \\ 0.25 & 0.35 \\ 0.5 & -0.6 \\ 0.75 & 1.2 \\ 1 & 0.8 \end{bmatrix}, \tag{4}$$

determine the Bernstein set $B_P(t) = B(t)\cdot P = [t, –8.9\cdot t^4 + 15.8\cdot t^3 – 7.2\cdot t^2 + t – 0.1]$, see Figure 1.

Example 2. The vector $c = [0, 0.25, 0.5, 0.75, 2\ ]^T$, as well as the matrix P differening form the above one only at the entry $P_{4,2} = E_2$, now $E_2 = 2$, generate the Bernstein polynomial $y = \mathrm{B_c}(t)$, where $\mathrm{B_c}(t) = B(t)\cdot P = t^4 + t$, see Figure 2.

Example 3. With the points A, B, C, D and E as in Example 2 we see that both the sequence $BACDE$ and the matrix P with lines storing these points in the indicated order generate the Bernstein set $B_P(t) = B(t)\cdot P = [2.25\cdot t^4 – 4\cdot t^3 + 4.5\cdot t^2 – \mathrm{t} + 0.25, –7.65\cdot t^4 + 11.8\cdot t^3 –2.7\cdot t^2 – t + 0.35]$.

The elements of this vector denoted by x and y, we have the relation $[\ x, y\ ] = [2.25\cdot t^4 – 4\cdot t^3 + 4.5\cdot t^2 – \mathrm{t} + 0.25, –7.65\cdot t^4 + 11.8\cdot t^3 –2.7\cdot t^2 – t + 0.35]$.

It is the parametric equation of the curve drawn, as the parameter $t \in <0, 1>$, in Figure 3.

Example 4. With the same points A, B, C, D and E as above, but taken in the different order, namely as the sequence $BACED$, we have Bernstein equation $[\ x, y\ ] =[\ –4\cdot t^4 + t^3 + 4.5\cdot t^2 –t + 0.25, –5.65\cdot t^4 + 10.2\cdot t^3 – 2.7\cdot t^2 –t + 0.35]$. Its graph, as well as the initial contour of Bernstein approximation, is drawn in Figure 4.

In the years 1959-62 Pierre Bézier and Paul de Casteljau started, in Renault and Citroën car enterprises, to approximate shapes by curves governed by Bernstein sets. This approach gained a wide popularity in the mathematical modeling of various shapes and the designers using it started to call the graphs of Bernstein equations as Bézier curves. Within the area at hand they are used only when the parameter t runs from 0 to 1, and therefore a **Bézier curve** is an arc starting at the point produced by $t = 0$ and having its other end for $t = 1$; these points are called a **starting**, or **zero**, point and a **last**, or s-th, **point**, if this curve is determined by s+1 control points. Directly from the definition of the polynomial B_P it is easy to see that $B_P(0) = P_0$ and $B_P(1) = P_s$, if the lines of the generating matrix P are indexed from 0 up to s and, at the same time, j-th line identifies the j-th control point P_j, j=0,1,2,...,s.

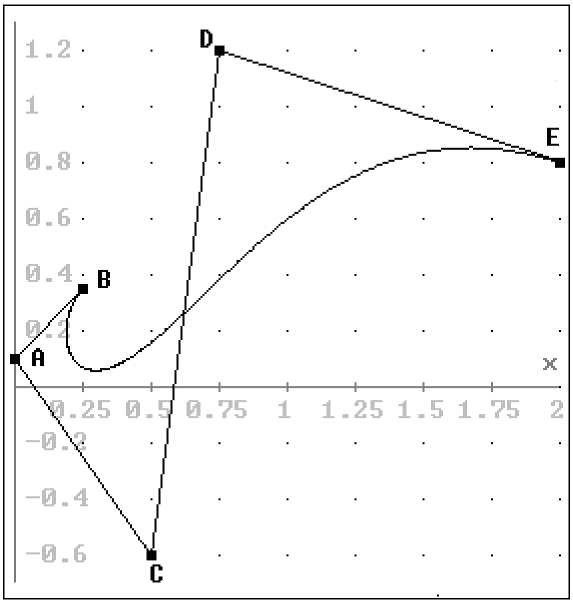

Figure 3: Bézier polygon and curve determined by the sequence *BACDE*, where *A*, *B*, *C*, *D* and *E* are as in Figure 2.

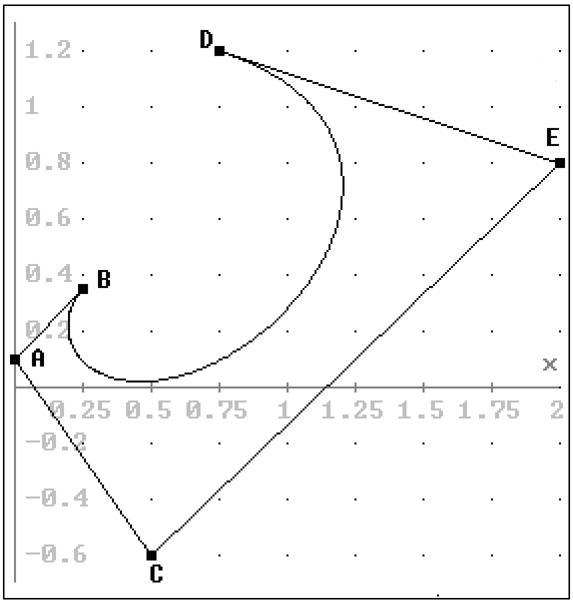

Figure 4: Bézier polygon and curve determined by the sequence *ABCED*, where *A*, *B*, *C*, *D* and *E* are as in Figures 2 and 3.

2. DE CASTELJAU ALGORITHM

Direct computation of any value $B_P(t)$ with $t \in (0, 1)$ needs to know the values of $p_{s,k}(t)$, k=1,2,...,s–1, so there have to be known the binomial coefficients $s!/\{k!\cdot(s-k)\}$. These operations are hard in programming: there are quickly produced really large numbers (e.g. 20! = 2 432 902 008 176 640 000, 10! = 3 628 800, $20!/10!^2$ = 184 756) and the range for integer type variables (this overcome usually is not signalized by computers), in floating-point arithmetic there arise roundings/truncations which may totally destruct results. Fortunately, these disadvantages take no place when we apply de Casteljau algorithm. This is the sequence of operations, and each one of them maps the matrix and a number t (it is, as always, the parameter of the representation) into a matrix having one less line. Lines of the input matrix denoted by $a_1, a_2, ..., a_r$, this transformation, T, works as follows

$$T\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \cdots \\ a_{r-1} \\ a_r \end{bmatrix} = \begin{bmatrix} (1-t)\cdot a_1 & +t\cdot a_2 \\ (1-t)\cdot a_2 & +t\cdot a_3 \\ (1-t)\cdot a_3 & +t\cdot a_4 \\ & \cdots \\ (1-t)\cdot a_{r-1} & +t\cdot a_r \end{bmatrix}. \tag{5}$$

Thus T acts over each column independently. It can be represented via the matrix, it transforms a r-dimensional vector W_r into the (r–1)-dimensional vector W_{r-1}, simply by Cauchy multiplication $W_{r-1} = TW_r := T(W_r) = T_r \cdot W_r$, where T_r is the matrix of operation T, i.e. it is the $(r-1, r)$-matrix

$$T_r = \begin{bmatrix} u & t & & & & \\ & u & t & & & \\ & & \ddots & \ddots & & \\ & & & u & t & \\ & & & & u & t \end{bmatrix}, \tag{6}$$

where $u := 1 - t$, and all elements outside both filled lines are 0.

As above, $P_0, P_1, ..., P_s$ are control points, they are consecutive lines of the generating matrix P, $P = [\, P_j \,]_{j=0..s}$, and they form the sequence $P_0P_1P_2...P_s$,. **de Casteljau algorithm**, realized on the sequence $P_0P_1P_2...P_s$, i.e. on the matrix $W_{s+1} := P$, and with the value t, is the computation of consecutive vectors $W_s := T_{s+1}W_{s+1}$, $W_{s-1} := T_sW_s$, ..., $W_1 := T_2W_2$. One sees that the final value

$W_1 = T_2W_2 = T_2T_3W_3 = ... = T_2T_3...T_{s+1}W_{s+1} = TP = B_P(t)$, where T is the superposition of all transformations T_j, so $T := T_2T_3...T_s$. (and, in matrix approach, the product of all matrices T_j,). The matrix W_i is called t-th **de Casteljau matrix**, and its graph is named **de Casteljau contour**.

Example 5. Let's illustrate the work of de Casteljau algorithm with the initial contour *BACED*, where *A*, *B*, *C*, *D* and *E* are as in Examples 3 and 4, and with t = 0.8. We have

$$W_5 = P = \begin{bmatrix} B \\ A \\ C \\ E \\ D \end{bmatrix} = \begin{bmatrix} 0.25 & 0.35 \\ 0 & 0.1 \\ 0.5 & -0.6 \\ 0.75 & 1.2 \\ 2 & 0.8 \end{bmatrix} \tag{7}$$

and it is the first of de Casteljau matrices. Next ones are

$$W_4 = \begin{bmatrix} 0.05 & 0.15 \\ 0.4 & -0.46 \\ 0.7 & 0.84 \\ 1.75 & 0.88 \end{bmatrix}, \quad (8)$$

$$W_3 = \begin{bmatrix} 0.33 & -0.338 \\ 0.64 & 0.58 \\ 1.54 & 0.872 \end{bmatrix}, \quad (9)$$

$$W_2 = \begin{bmatrix} 0.578 & 0.3964 \\ 1.36 & 0.8136 \end{bmatrix} \quad (10)$$

and, finally, the last one is

$$W_1 = [1.2036, 0.73016] = B_P(0.8). \quad (11)$$

The graph of the input matrix P and graphs of all other de Casteljau matrices are shown in Figure5. The work with $t = 4/5$ produces successive control points sitting on segments of polygonal lines and divided any of them in the proportion 4:1. The last matrix, W_1, has one line only, its graph is the point. In Figure 5 this is the point W. It is the point corresponding to the value $t = 0.8$.

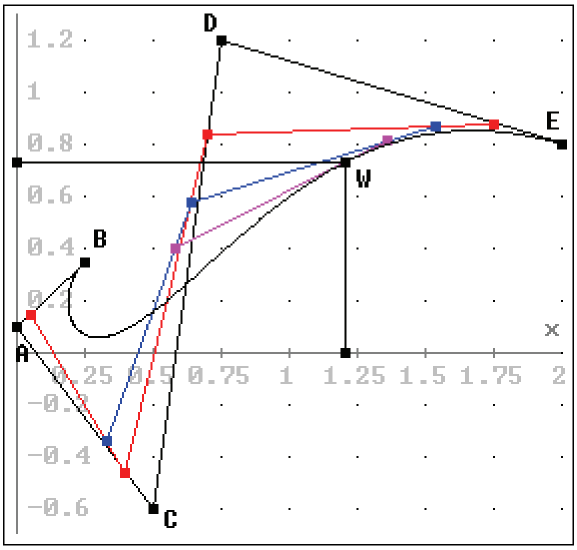

Figure 5: Consecutive de Casteljau contours generated by the points sequenced *BACED* and with $t = 0.8$; the last contour is the point $W = B_P(t = 0.8)$.

3. DEGREE RISING

In modeling a Bézier curve, the basic operations are the change of (coordinates of) control points, the increasing in number of control points, the deletion of a control point. It means the change of elements in the generating matrix P, the increase or decrease of lines of P, respectively. In general, any such change affects the entire curve.

Potential degree rising of Bézier curve is the process resulting in the replacing of a given matrix P by the matrix $\widetilde{P}$, which row degree is greater by 1 than that of P and producing the same curve. It is enough to describe this procedure in a standard case, i.e. when the Bézier curve is defined by the vector $c = [c_0, c_1, ..., c_s]^T$. In view of the identity $c_k = t{\cdot}c_k + (1{-}t){\cdot}c_k$ it is easy to see that

$$B_c(t) = \sum_{k=0}^{s} p_{s,k}(t)\cdot c_k =$$
$$\sum_{k=0}^{s+1} p_{s+1,k}(t)\cdot w_k = B_w(t), \quad (12)$$

where $w_0 = c_0$, $w_{s+1} = c_s$ and

$$w_k = \frac{k}{s+1}\cdot c_{k-1} + \left(1 - \frac{k}{s+1}\right)\cdot c_k \text{ for } k=1,2,...,s. \quad (13)$$

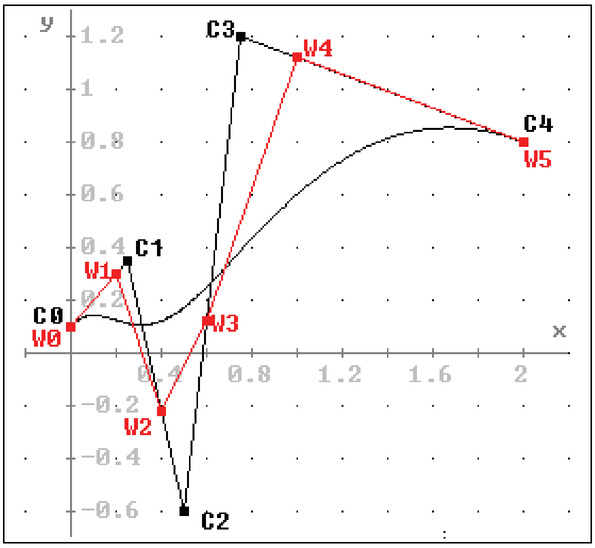

Figure 6: Bézier curve generated by any of vectors: $c = [\,0, 0.25, 0.5, 0.75, 2\,]^T$, $w = [\,0.1, 0.3, -0.22, 0.12, 1.12, 0.8]^T$

Example 6 (see Figure 6) The vectors $c = [0, 0.25, 0.5, 0.75, 2\,]^T$ and $w = [\,0.1, 0.3, -0.22, 0.12, 1.12, 0.8]^T$ determine the same Bézier curve, $y = t^4 + t$. Denoting the augmented matrices of c and w by C and W, respectively, we have

$$C = \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix} = \begin{bmatrix} 0 & 0.1 \\ 0.25 & 0.35 \\ 0.5 & -0.6 \\ 0.75 & 1.2 \\ 2 & 0.8 \end{bmatrix} \quad (14)$$

$$W = \begin{bmatrix} W_0 \\ W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \end{bmatrix} = \begin{bmatrix} 0 & 0.1 \\ 0.2 & 0.3 \\ 0.4 & -0.22 \\ 0.6 & 0.12 \\ 1 & 1.12 \\ 2 & 0.8 \end{bmatrix}. \qquad (15)$$

Degree rising is applied when Bézier curve approximates a designed shape in a degree not good enough and probably the change in the positioning of the control points does not results in a better fitting.

In spite of the name of the considered procedure, the polynomials B_c and B_w are of the same algebraic degree, but, in general, even a very small change in w makes the degree of B_w higher. The inverse process to the rising, the deletion of any point results, in general, in decrease of the algebraic degree, so it produces another Bézier curve. We apply it when a Bézier curve determined by s+1 points is good enough and we want to check whether it is possible to have good approximation with s control points.

4. SMOOTH JOINT OF BÉZIER CURVES

We say that two polynomials, U and W, **smoothly meet** each other at a point τ, if at this point they have the common value and their derivatives do it, too, i.e. if $U(\tau) = W(\tau)$, $U'(\tau) = W'(\tau)$.

Therefore Bernstein polynomials B_c and B_d, determined by vectors $c = [\,c_i\,]_{i=0..\gamma}$ and $d = [\,d_j\,]_{j=0..\delta}$, resp., are smoothly joined at a point A, if

- A is the last point of the curve B_c and the initial point of B_d,
- A is collinear with the point no. γ–1 of B_c and the point no.1 of B_d.

For $n = 2$, so on the plane $\boldsymbol{R}^2$, the smooth joint takes place if $\gamma\cdot\{c_\gamma - c_{\gamma-1}\} = \delta\cdot\{d_1 - d_0\}$. Consequently, Bézier curves generated by matrices C and D meet smoothly at the point $A = C_\gamma = D_0$, if $\gamma\cdot\{A - C_{\gamma-1}\} = \delta\cdot\{D_1 - A\}$,where C_i and D_j denote i-th point of C, i.e. i-th line of the matrix C, and j-th point of D.

The inverse process, the **subdivision** of a given Bézier curve generated by a matrix B into two Bézier curves which meet each other at its arbitrary point, Q, means algebraically to produce two matrices, C and D, generating such curves. Now Q is the last point of the matrix C and the initial point of the matrix D.

5. INVERSE BERNSTEIN APPROXIMATION

The **inverse (standard) Bernstein approximation** consists in determination of the vector c such that Bézier curve generated by c, i.e. $y = B_c(t)$ and $0 \le t \le 1$, coincides with a given curve $y = f(t)$ at s+1 points, see e.g. (Becker 1979). We find the vector $c = [c_0, c_1, c_2, ..., c_{s-2}, c_{s-1}, c_s]^{\mathrm{T}}$ by taking s+1 points $F_j = (t_j, f_j)$ with abscissas $t_j := j\cdot h$, where $h := 1/s$, j=0,1,2,...,s, and ordinates $f_j = f(t_j)$. The defining condition takes form $B_r(t_j) = f_j$ for $j = 0, 1, ..., s$, and its matrix form is $M\cdot c = f$, where, with no reason to confuse by using the same letter f, $f = [f_0, f_1, f_2, \ldots, f_{s-2}, f_{s-1}, f_s]^{\mathrm{T}}$, $M = [\,m_{j,k}\,]_{j,\,k = 0,1,2,...,s}$, $m_{j,k} = p_{s,k}(j\cdot h)$.

The resolving system $M\cdot c = f$ may be at once reduced by 2, because from the equalities $B_r(0) = r_0$, $B_r(1) = r_s$ it follows $c_0 = f_0$, $c_s = f_s$. If we know values $d_0 = f\,'(0)$ and $d_s = f\,'(1)$, then we may reduced it by 2 again, because $s\cdot\{\,c_1 - c_0\,\} = d_0$, $s\cdot\{\,c_s - c_{s-1}\,\} = d_s$.

Example 7. We find the vector c, which generates the Bernstein polynomial B_c assuming at points $j\cdot h$, j=0,1,..,s, h=1/s, s=6, values $f(j\cdot h)$, where $f(t) = \sin(\pi\cdot t)$. Now

$$f = [\,0,\ 1/2,\ \sqrt{3}/2,\ 1,\ \sqrt{3}/2,\ 1/2,\ 0\,]^{\mathrm{T}}, \qquad (16)$$

$$M = \frac{1}{46656}\begin{bmatrix} 46656 & 0 & 0 & 0 & 0 & 0 & 0 \\ 15625 & 18750 & 9375 & 2500 & 375 & 30 & 1 \\ 4096 & 12288 & 15360 & 10240 & 3840 & 768 & 64 \\ 729 & 4374 & 10935 & 14580 & 10935 & 4374 & 729 \\ 64 & 768 & 3840 & 10240 & 15360 & 12288 & 4096 \\ 1 & 30 & 375 & 2500 & 9375 & 18750 & 15625 \\ 0 & 0 & 0 & 0 & 0 & 0 & 46656 \end{bmatrix} \qquad (17)$$

and the solution of the resolving system $M\cdot c = f$ is

$$c = [0,\ 6160 - 3375q,\ 9720q - 16208,\ 22536 - 12555q,\ 9720q - 16208,\ 6160 - 3375q,\ 0]^{\mathrm{T}}/600, \qquad (18)$$

where $q := \sqrt{3}$. In Figure 7 we see points P_j with ordinates equal to these numbers, they sit on the polygon of the Bézier curve determined by them. Moreover, we see the graph of the approximation error (to be visible, it is magnified 10000 times). The Bézier curve we produced practically coincides with the given arc $y = \sin(\pi\cdot x)$.

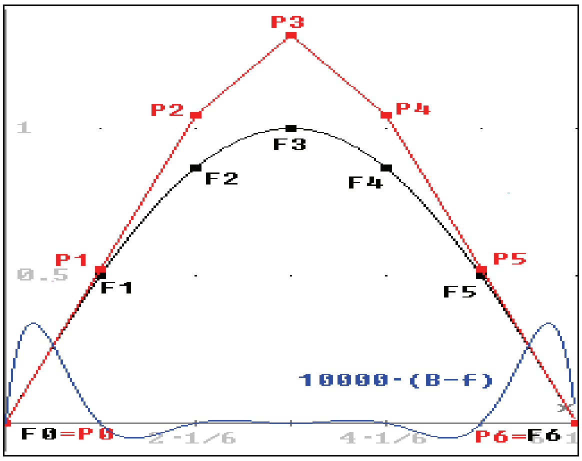

Figure 7: Graphs $y = f(x) = \sin(\pi\cdot x)$, $x \in$ <0, 1>, points $F_j = (j/6, f_j = f(j/6))$, $j = 0, 1, ..., 6$, sitting on this graph, points $P_j = (j/6, c_j)$ generating the Bernstein polynomial B_P and the graph $y = 10^4\cdot\{B_P - f\}(x)$ showing the error of the approximation of the function f by the polynomial B_P.

6. APPROXIMATING NONCIRCULAR WHEEL

Let's consider a variable-speed toothed belt transmission system created by combining the geometric and kinematic characteristics of a non-circular transmission system with a timing belt transmission system.

The required degree of speed variability is obtained by the use of pulleys constructed with wheel rims having shapes of ellipses, ovals or non-circular disks (figure 8).

Figure 8: Wheels of the variable-speed transmission system: a) the noncircular wheel, b) the elliptical wheel

The construction must meet some conditions. Namely, the length of the belt must be equal to the length of system envelope. In order to ensure correct operation of a variable-speed transmission system the active and passive sections of the belt must be tight thanks to constant action of applicable force. Circumferences of the wheels are the product of the pitch of the belt and an integer number. Thus, one is able to determine the average transmission ratio of the system as the relation of wheels' circumferences or their number of teeth. Driving cyclicity can be guaranteed only by toothed belts whose plastic strain will increase during operation only slightly. At the same time the belts must be initially pre-tightened in order to avoid belt's slip or skipping on the teeth of the wheels.

A model of a two-wheel transmission system consisting of an elliptical driving wheel and a noncircular driven wheel is presented in Figure 9.

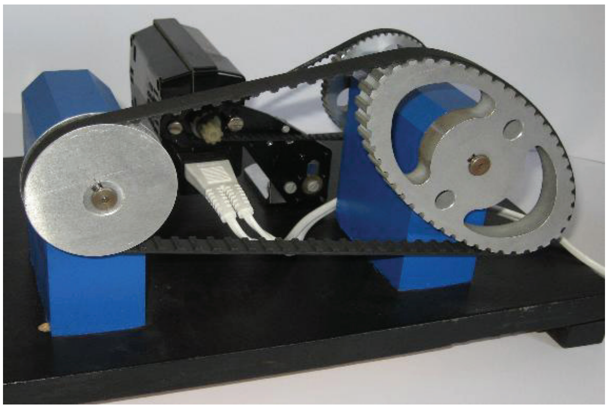

Figure 9: The view of variable-speed toothed belt transmission system with an eccentrically mounted wheel

In the research on effective parts of some devices (see Figure 8 – photo) there appear problems to get their mathematical expressions for the border line in aim to, e.g., to reproduce them on computer-controlled punch or cutting machineries, to project teeth of gears. In considered case the curve is close to, but it is not an ellipse or any other standard shape. Therefore we look for the Bernstein parameterization. There are offered computer programs to find it, e.g. (Zhao and Shene 1999), but we worked in our own program, called BezierFit, elaborated within Delphi system from Code Gear.

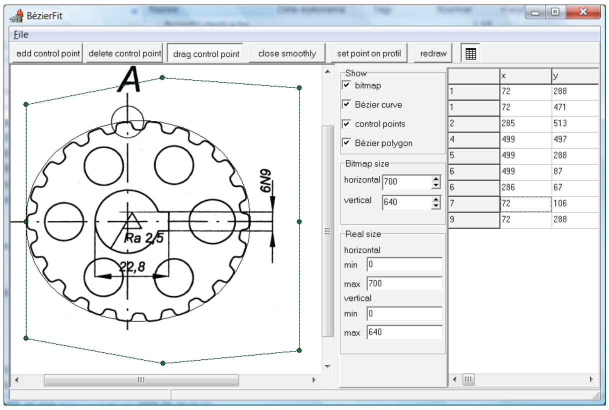

Figure 10: BezierFit program by Karol Gajda

Working in this program we look at the screen, see Figure 10. In the background of its working area there is displayed, from a BMP file, the line to be modeled, and there is the reason we opt for our own program. In this area we may place controlling points (the first placing defines the input matrix *P*), we can move them, we can delete any point, we can add points in the way to keep up-to-now curve unchanged. Every operation is at instance visualized as the Bézier curve. In the operational area of the screen there are buttons activating offered operations. Here are, e.g., the positioning of control points (by mouse or by setting the coordinates), the memorizing of produced coefficients (they are saved as a text file *.BFP), smooth closing (choosing it we close smoothly the profile, i.e., in the way that the last point coincides with the starting point P_0), the positioning of points on a given profile (and then, by the inverse Bernstein approximation, the program produces the appropriate control points).

REFERENCES

Becker, M., 1979. An elementary proof of the inverse theorem for Bernstein polynomials. *Aequationes mathematicae*, 19(1), 145–150.

Chen, Y., 2000. Some thoughts on the polynomial approximations to a given function. *Chinese J.of Physics*, 38(5), 927–938.

Ding, R., Zhang, Y., 2003. The extension of the dual de Casteljau algorithm, *Proceedings of 4th Int.Conf. on Parallel and Distributed Computing, Applications and Technologies*, 68–92, 2003, Chengdu, IEEE Press.

Hong, X., Mitchell, R.J., 2007. Hammerstein model identification algorithm using Bezier-Bernstein approximation. *IET Control Theory Appl.*, 1(4), 49–59.

John, Ch.T., 2007. All Bézier curves are attractors of iterated function systems. *New York J.Math.*, 13, 107–115.

Kowalski, E, 2006. Bernstein polynomials and Brownian motion, *American Mathematical Monthly*, 12.

Lorentz, G.G., 1953. *Bernstein polynomials*, Toronto: University of Toronto Press.

Madi, M., 2004. Closed-form expressions for the approximation of arclength parametrization for Bézier curves, *Int.J.Appl.Math.Comput.Sci.*, 14(1), 33–41.

Zhao, Y., Shene, C.K., 1999. *DesignMentor program*. Department of Computer Science, Michigan Technological University. Available from: http://lumimath.univ-mrs.fr/~jlm/cours/DMmanuals/curve/ [accessed 26 April 2008].