TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics, Mechanics and Engineering
Vol. 61, Issue II, June, 2018

SPLINE INTERPOLATION WITH THIRD-DEGREE BÉZIER FUNCTIONS

Nicolae URSU-FISCHER, Diana Ioana POPESCU, Ioan RADU

***Abstract**: The Bézier, B-spline and NURBS curves and surfaces were extensively studied in the literature and used in the shapes design of different products, initially for cars. From the mathematical point of view, in most cases the problems linked to interpolations and approximations with curves and surfaces have been solved.*
Our research team has studied and solved the spline interpolation problem using third-degree Bézier curves between the interpolation points. An efficient algorithm has been set up following the mathematical model which solves the problem. It has been programmed in the C language and used to solve different numerical examples, with results illustrated by diagrams.
It can be observed that there exist multiple variants of interpolation curves due to the imposed interpolation problem specificity: there exist more unknowns than equations (possible conditions to be imposed), that's why one has to start with some initial values for a pair of selected unknowns.
***Key words:** interpolation, Bernstein polynomials, Bézier curves, smoothness conditions, spline functions*

1. INTRODUCTION

For more than two hundred years many mathematicians have focused on the interpolation problem using polynomials, a problem of great theoretical and practical interest [5], [19], [25].

Lagrange, Newton, Gauss, Stirling, Bessel discovered different numerical procedures to establish the interpolation polynomials formulas, the interpolating points being distinct and having abscissas arranged in ascending order.

If the points are disposed anyhow in the plan and the interpolating curve has to pass from a point to another in a precise order, one has to use parametric defined curves - the best known being Bézier, B-spline and NURBS curves. Nowadays these curves are widely used in the programs for the shape design of different object, products, especially in the car industry [1], [6], [7], [8], [9], [10], [13], [20].

When we deal with polynomials to perform interpolation, the great number of points implies a big value of polynomial degrees: if the number of points is **n+1,** the degree is **n**; involving much time to perform the polynomial values computing. To avoid this problem, the solution is to use spline functions, of low degree, usually of third degree, in each interval between two successive points. Also, we have to mention that a condition is necessary: in each common point of two spline functions the values of the first two derivatives have to be equal.

In this paper our aim is to study interpolation with spline functions and to find an algorithm that allows us to construct the mathematical model for interpolating points in plan with spline type Bézier functions of third degree, so that equality is achieved for the first two derivatives in common points.

As will be shown as follows, this algorithm has been found and transposed into a program written in the C programming language. Using this original computer program, several examples have been obtained that illustrate the issue.

In this case, it is not a single solution to a specified problem because the number of conditions (the number of equations) is less than the number of unknowns.

2. BERNSTEIN POLYNOMIALS, BÉZIER CURVES AND SMOOTHNESS CONDITIONS

Bernstein polynomials of degree **n** have the following expression, similar with probability density function in the case of binomial distribution:

$$\Phi_{k,n}(t) = C_n^k\,(1-t)^{n-k}\,t^k\,,\quad k=0,1,2,...,n,\quad t\in[0;1] \tag{1}$$

For the first time, in 1913 [3], Serghei N. Bernstein (1880-1968) has used these polynomials, now known as Bernstein polynomials, to prove famous Weierstrass theorem about the approximation of functions with polynomials.

We have the following known properties and relations regarding these polynomials [2], [4], [16], [19], [25]:

$$\Phi_{k,n}(t)\geq 0\,,\ \sum_{k=0}^{n}\Phi_{k,n}(t) = 1,$$

$$\Phi_{k,n}(t) = (1-t)\,\Phi_{k,n-1}(t) + t\,\Phi_{k-1,n-1}(t)$$

(recurrence formula)

$$\Phi_{k,n}(0)=\delta_{k,0}\ ,\ \Phi_{k,n}(1)=\delta_{k,n} \tag{2}$$

The Bernstein polynomial may be written as a linear combination of power functions:

$$\Phi_{k,n}(t) = \sum_{j=k}^{n}(-1)^{j-k}\,C_n^j\,C_j^k\,t^j$$

also existing the relation that links the power function with Bernstein polynomials, as follows (see [7] pp. 64, [20] for more details):

$$t^k = \frac{1}{C_n^k}\sum_{j=k}^{n}C_j^k\,\Phi_{j,n}(t)$$

Based on Bernstein polynomials, during 1960-1970, Pierre Bézier (1910-1999,) working at Renault company, has established the expression of functions that were used in computer aided design of cars shapes [4]. He was the first researcher that tried and succeeded to establish links between the shape of an object and the mathematical model of some curves or/and surfaces that fit to this shape.

A number of **n+1** points (named knots or control points) are considered in the Oxy plan, with known coordinates: (x_{0P}, y_{0P}), (x_{1P}, y_{1P}), …, (x_{kP}, y_{kP}), …, $(x_{n-1,P}, y_{n-1,P})$, (x_{nP}, y_{nP}) . The polygonal line linking these points is called a control polygon.

The column vectors containing the control points coordinates are denoted by:

$$P_0 = \begin{bmatrix} x_{0P} \\ y_{0P} \end{bmatrix},\ P_1 = \begin{bmatrix} x_{1P} \\ y_{1P} \end{bmatrix},\ ....\quad P_k = \begin{bmatrix} x_{kP} \\ y_{kP} \end{bmatrix},\ .....,\ P_n = \begin{bmatrix} x_{nP} \\ y_{nP} \end{bmatrix} \tag{3}$$

The Bézier curve coordinates are given by:

$$B(t)=\sum_{k=0}^{n}\Phi_{k,n}(t)\ P_k\,,\quad t\in[0,1] \tag{4}$$

or

$$\begin{bmatrix} x_B(t) \\ y_B(t) \end{bmatrix} = \sum_{k=0}^{n} C_n^k\,t^k\,(1-t)^{n-k}\begin{bmatrix} x_{kP} \\ y_{kP} \end{bmatrix}, \tag{5}$$

the coordinates values for each point of the Bézier curve depending on values of the **n+1** Bernstein polynomials of degree **n** multiplied with the values of control points coordinates.

Because the Bernstein polynomials $\mathbf{\Phi_{0,n}}(t)$ and $\mathbf{\Phi_{n,n}}(t)$ satisfy the conditions $\Phi_{0,n}(0) = \Phi_{n,n}(1) = 1$, it results:

$$\begin{bmatrix} x_B(0) \\ y_B(0) \end{bmatrix} = \begin{bmatrix} x_{0P} \\ y_{0P} \end{bmatrix},\ \begin{bmatrix} x_B(1) \\ y_B(1) \end{bmatrix} = \begin{bmatrix} x_{nP} \\ y_nP \end{bmatrix}$$

therefore the Bézier curve starts in the first point $\mathbf{P_0}$ of the control polygon and ends in the last point $\mathbf{P_n}$, not necessarily containing other points of the control polygon.

Formula (4) may be written as follows:

$$\begin{aligned} B(t)&=\sum_{k=0}^{n}\Phi_{k,n}(t)\ P_k = \\ &= C_n^0\,(1-t)^n\,P_0 + C_n^1\,(1-t)^{n-1}\,t\,P_1 + ... + \\ &+ C_n^{n-1}\,(1-t)\,t^{n-1}\,P_{n-1} + C_n^n\,t^n\,P_n \end{aligned} \tag{6}$$

By calculating the first two derivatives in points that correspond to values **t=0** and **t=1** of the parameter **t** , it results:

$$B'(0) = n\,(P_1 - P_0\ ,\quad B'(1) = n\,(P_n - P_{n-1}) \tag{7}$$

$$B''(0)=n(n-1)P_0-2n(n-1)P_1+n(n-1)P_2$$
$$B''(1)=n(n-1)P_{n-2}-2n(n-1)P_{n-1}+n(n-1)P_n \qquad (8)$$

Expressions (7) may be interpreted in the following manner: the tangents to the Bézier curves in the points defined for t=0 and t=1 overlap with the first and the last control polygon sides.

If the interpolation is performed with a sequence of Bézier curves, each curve having the last point in the same position as the first point of the next curve, we can impose some smoothness conditions, demanding the equality of first two derivatives of the functions in the common points.

In this case, the following conditions have to be accomplished:

$$m(U_m-U_{m-1})=n(V_1-V_0) \qquad (9)$$

$$m(m-1)U_{m-2}-2m(m-1)U_{m-1}+m(m-1)U_m=$$
$$=n(n-1)V_0-2n(n-1)V_1+n(n-1)V_2 \qquad (10)$$

the desired interpolation being spline type. Here are some pointers to the literature: [8], [10], [22], [25], [26] a. o.

3. INTERPOLATION WITH SUCCESSIVE BÉZIER FUNCTIONS OF THIRD DEGREE, WITH EQUAL FIRST TWO DERIVATIVES IN COMMON POINTS

According to relation (3), if the number of control points is **n+1**, the degree of polynomials composing the Bézier function is **n**, like in the case of Lagrange **n** degree interpolation polynomial that passes through **n+1** points.

We can avoid the use of high degree interpolation polynomials, considering in each interval a third degree polynomial, so called spline polynomial, [14], [15], [21], two such adjacent polynomials having same values for the first two derivatives in common points.

If the degree of Bézier spline functions is three, it results that each control polygon must have four control points and obviously the first and the last points belong to the interpolation points. In the common points are located the fourth points of one control polygon and the first point of the next control polygon.

Points of control polygon	**Vertexes of auxiliary control polygons**	**Bézier functions of interpolation**
$P_0 \sim P_1$,	$Q_0 \equiv P_0$, Q_1, Q_2, Q_3	$S_{0,1}(t)$
$P_1 \sim P_2$,	Q_3, Q_4, Q_5, Q_6	$S_{1,2}(t)$
. . . . .	. . . .	. . . .
$P_{k-1} \sim P_k$,	Q_{3k-3}, Q_{3k-2}, Q_{3k-1}, Q_{3k}	$S_{k-1,k}(t)$
$P_k \sim P_{k+1}$,	Q_{3k}, Q_{3k+1}, Q_{3k+2}, Q_{3k+3}	$S_{k,k+1}(t)$
. . . . .	. . . .	. . . . . .
$P_{n-1} \sim P_n$,	Q_{3n-3}, Q_{3n-2}, Q_{3n-1}, $Q_{3n} \equiv P_n$	$S_{n-1,n}(t)$

The expressions of third-degree Bézier interpolation functions are:

$$S_{k,k+1}(t)=(1-t)^3Q_{3k}+3(1-t)^2tQ_{3k+1}+$$
$$+3(1-t)t^2Q_{3k+2}+t^3Q_{3k+3}, \quad k=\overline{0,n-1};$$
$$t\in[0;1]$$

and the following conditions must imposed:

$$S_{k-1,k}(1)=S_{k,k+1}(0);$$

$$\left.\frac{dS_{k-1,k}(t)}{dt}\right|_{(t=1)}=\left.\frac{dS_{k,k+1}(t)}{dt}\right|_{(t=0)};$$

$$\left.\frac{d^2S_{k-1,k}(t)}{dt^2}\right|_{(t=1)}=\left.\frac{d^2S_{k,k+1}(t)}{dt^2}\right|_{(t=0)}$$

in order that those functions to be of spline type.

Starting from expression (9) and based on the equality of the first order derivatives, one may write:

$$3(Q_{3k}-Q_{3k-1})=3(Q_{3k+1}-Q_{3k});$$
$$Q_{3k}=\tfrac{1}{2}(Q_{3k-1}+Q_{3k+1}), \quad k=1,2,...,n-1$$

therefore it results that the points belong to the same line and the point Q_{3k} is in the middle of segment $Q_{3k-1}\,Q_{3k+1}$.

Equalizing the values of the second derivatives in both sides of point Q_{3k}, according to expression (10):

$$Q_{3k-2} - 2Q_{3k-1} + Q_{3k} = Q_{3k} - 2Q_{3K+1} + Q_{3k+2},$$
$$k = 1, 2, \ldots, n-1$$

In this case the interpolation points (with known coordinates) are Q_0, Q_3, Q_6, . . . , Q_{3k}, Q_{3k+1}, . . . , Q_{3n-3}, Q_{3n}, for each of them the smoothness conditions have to be accomplished. These points are the first and the last vertices of four sided control polygons.

The main step in solving the imposed interpolation problem is the computing of the intermediate control points Q_1, Q_2, Q_4, Q_5, . . . , Q_{3k+1}, Q_{3k+2}, . . ., Q_{3n-2}, Q_{3n-1} coordinates, of each four sided polygon.

By imposing two conditions (equality of first two derivatives values) for n-1 times (for each of the n-1 points) it will result 2n-2 relations with 2n unknowns.

Hence, it is necessary to impose two supplementary conditions about two points of the control polygons, e. g. for points Q_1 şi Q_{3n-1} considered with known coordinates.

The following simultaneous equations will result:

$$\begin{aligned}
&-2Q_2 + 2Q_4 - Q_5 = -Q_1\\
&Q_2 + Q_4 = 2Q_3\\
&Q_4 - 2Q_5 + 2Q_7 - Q_8 = 0\\
&Q_5 + Q_7 = 2Q_6\\
&Q_7 - 2Q_8 + 2Q_{10} - Q_{11} = 0\\
&Q_8 + Q_{10} = 2Q_9\\
&\ldots\ldots\ldots\ldots\ldots\ldots\\
&Q_{3k-2} - 2Q_{3k-1} + 2Q_{3k+1} - Q_{3k+2} = 0\\
&Q_{3k-1} + Q_{3k+1} = 2Q_{3k}\\
&\ldots\ldots\ldots\ldots\ldots\ldots\\
&Q_{3n-8} - 2Q_{3n-7} + 2Q_{3n-5} - Q_{3n-4} = 0\\
&Q_{3n-7} + Q_{3n-5} = 2Q_{3n-6}\\
&Q_{3n-5} - 2Q_{3n-4} + 2Q_{3n-2} = Q_{3n-1}\\
&Q_{3n-4} + Q_{3n-2} = 2Q_{3n-3}
\end{aligned} \tag{11}$$

This system of equations may be written in the matrix form: **AX=B.**

$$\begin{bmatrix}
-2 & 2 & -1 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0\\
1 & 1 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0\\
0 & 1 & -2 & 2 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0\\
0 & 0 & 1 & 1 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0\\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots\\
0 & 0 & 0 & 0 & \ldots & -2 & 2 & -1 & \ldots & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & \ldots & 1 & 1 & 0 & \ldots & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & \ldots & 0 & 1 & -2 & \ldots & 0 & 0 & 0 & 0\\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots\\
0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & -2 & 2 & -1 & 0\\
0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 1 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 1 & -2 & 2\\
0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & 1 & 1
\end{bmatrix}_{(2n-2)x(2n-2)}
\begin{bmatrix}
Q_2\\ Q_4\\ Q_5\\ Q_7\\ \ldots\\ Q_{3k-1}\\ Q_{3k+1}\\ Q_{3k+2}\\ \ldots\\ Q_{3n-7}\\ Q_{3n-5}\\ Q_{3n-4}\\ Q_{3n-2}
\end{bmatrix}_{(2n-2)}
=
\begin{bmatrix}
-Q_1\\ 2Q_3\\ 0\\ 2Q_6\\ \ldots\\ 0\\ 2Q_{3k}\\ 0\\ \ldots\\ 0\\ 2Q_{3n-6}\\ Q_{3n-1}\\ 2Q_{3n-3}
\end{bmatrix}_{(2n-2)}$$

Noticing the values of matrix elements, we may write the following relations:

$$\begin{aligned}
&a_{i+1,i} = 1, \quad i = 2, 3, 4, \ldots, 2n-2\\
&a_{j,j} = -2, \quad a_{j+1,j+1} = 1, \quad a_{j,j+1} = 2,\\
&a_{j,j+2} = -1, \quad j = 1, 3, 5, \ldots, 2n-1
\end{aligned} \tag{12}$$

$$X_k = Q_{3k-1}\,, \quad X_{k+1} = Q_{3k+1}\,, \qquad k = 1, 2, 3, \ldots, n-1 \tag{13}$$

$$B_1 = -Q_1\,, \quad B_{2n-1} = Q_{3n-1}\,, \qquad B_k = 2Q_{3k/2}\,, \quad k = 2, 4, 6, \ldots, 2n-2 \tag{14}$$

After the numerical solving of the system AX=B (the partial pivoting method was used), the unknown values will result, as elements of the column vector B.

The coordinates of the control polynomial vertices are obtained in the following way:

$$Q_{3k/2-1} = B_{k-1}\,, \quad Q_{3k/2+1} = B_k\,, \qquad k = 2, 4, 6, \ldots, 2n-2 \tag{15}$$

The coordinates of all intermediate points of auxiliary polygons being known at this moment, it is possible to calculate the coordinates of all points of the third-degree Bézier interpolating curves.

Based on this procedure, a C language computer program was written, [17], [24] used for the computation of the examples that follows.

4. NUMERICAL EXAMPLES

Figures 1 to 6 contain six diagrams obtained on the base of the previous presented relations. In each figure the points denoted with 3, 6 and 9 are common for two successive third-degree Bézier spline functions. The total number of interpolation point is five.

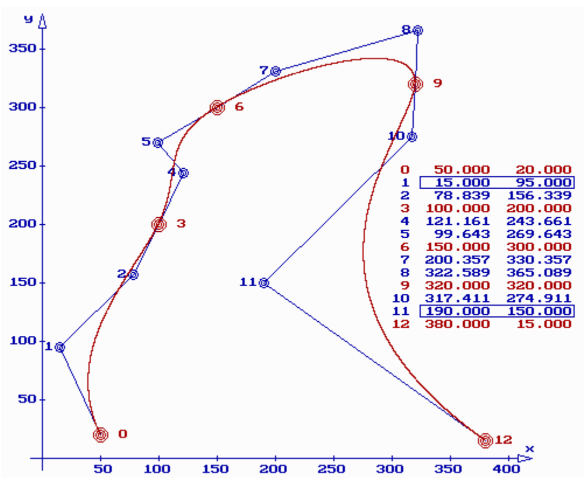

Fig. 1. Coordinates of P_1 [15;95] and P_{11} [190;150]

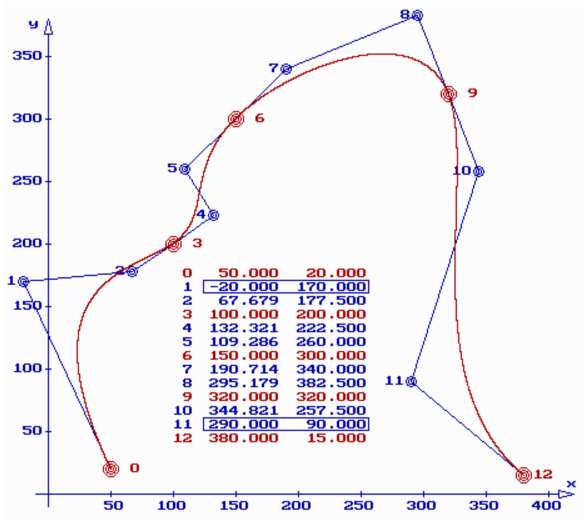

Fig. 2. Coordinates of P_1 [-20;170] and P_{11} [290;90]

The shapes of the diagrams differ from figure to figure because the coordinates of points no. 1 and 11 have different values. These different values are specified for each diagram, being marked by a rectangle.

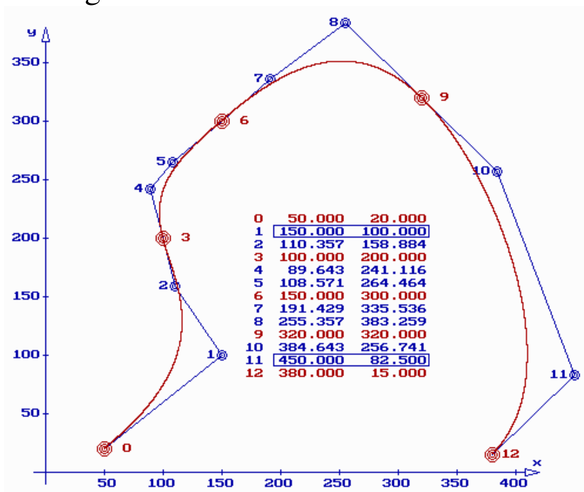

Fig. 3. Coordinates of P_1 [15 ;100] and P_{11} [450;82.5]

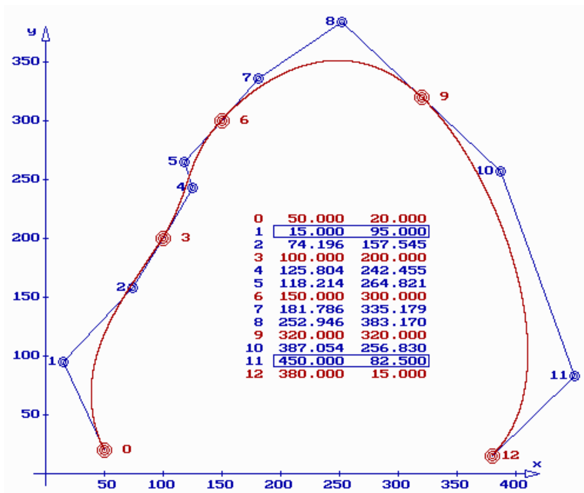

Fig. 4. Coordinates of P_1 [15;95] and P_{11} [450;82.5]

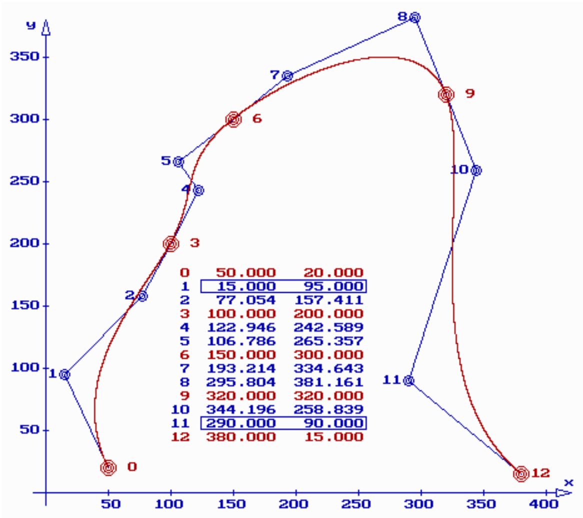

Fig. 5. Coordinates of P_1 [15;95] and P_{11} [290;90]

Figure 7 presents five superposed diagrams. All diagrams fulfil the imposed conditions for being Bézier spline interpolation functions of third degree - contain the five interpolation points and in three of them (points 3, 6 and 9) the condition of smoothness is accomplished.

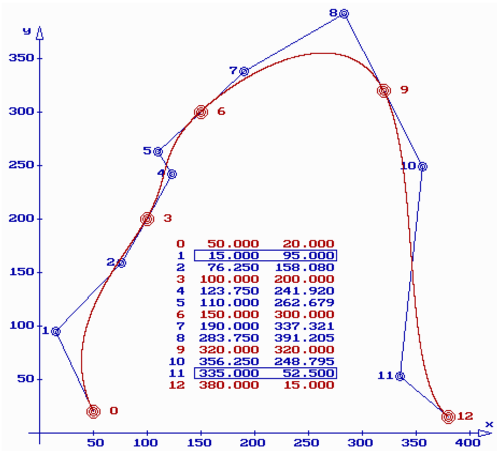

Fig. 6. Coordinates of P_1 [15;95] and P_{11} [335;52.5]

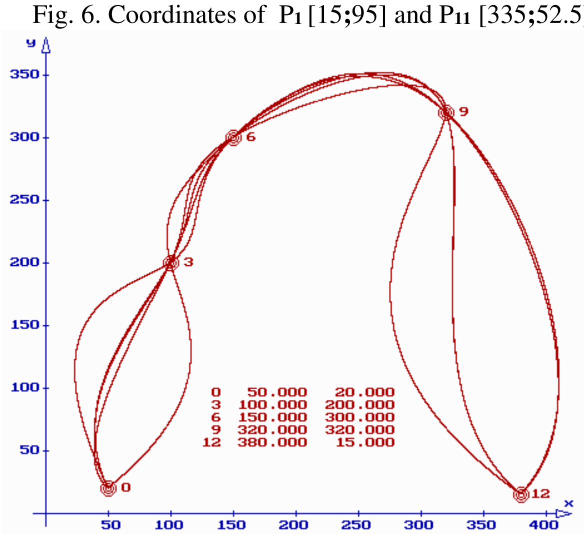

Fig. 7. Five superposed diagrams

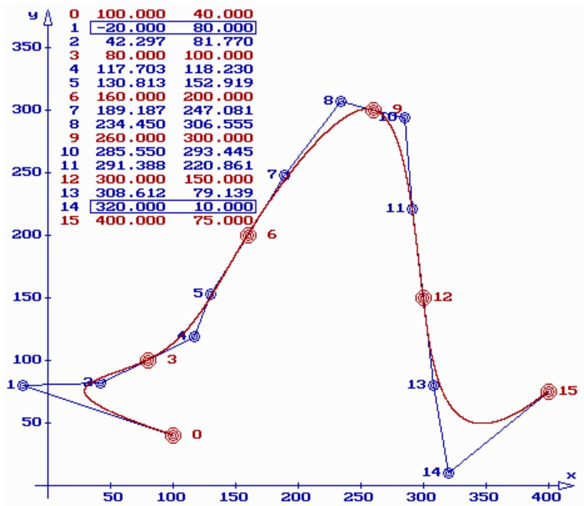

Fig. 8. Coordinates of P_1 [-20;80] and P_{14} [320;10]

The diagrams in figures 8 and 9 were also obtained with our original calculation algorithm transposed in a program by using the C programming language. They contain Bézier spline interpolating third-degree functions passing through six interpolating points.

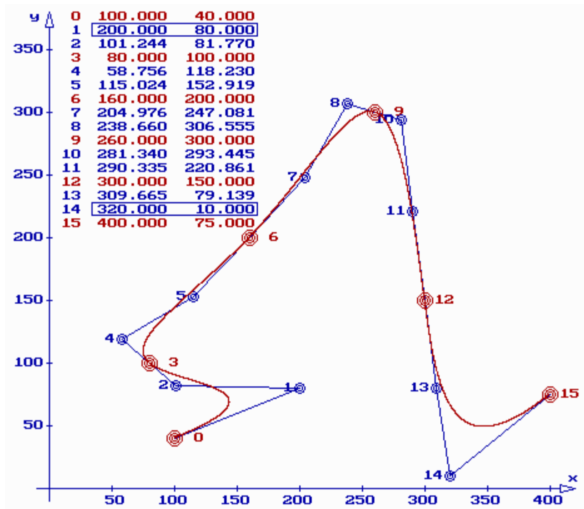

Fig. 9. Coordinates of P_1 [200;80] and P_{14} [320;10]

5. CONCLUSIONS

The focus of our investigations lies in the solving of a useful problem regarding the spline type interpolation with third-degree Bézier functions. The equality of the first two derivatives in the common points of interpolation is ensured.

It is stated that in these cases we may obtain multiple solutions that accomplished the imposed conditions.

The developed algorithm is general, very efficient, and it was tested for many situations, being transposed in a C language computer program.

The multitude of numerical examples with which we have worked, some of them presented in this paper, allowed us to test the correctness and efficiency of the calculation procedure.

6. REFERENCES

[1] Barnhill, R., Riesenfeld, R. (ed.), *Computer Aided Geometric Design*, Academic Press, 1974

[2] Bender, M., Brill, M., *Computergrafik. Ein anwendungsorientiertes Lehrbuch*, Carl Hansen Verlag, München, 2003, 516 s., ISBN 3-446-22150-6.

[3] Bernstein, S. N., *Démonstration du théorème de Weierstrass fondée sur le calcul des probabilités*, Сообщенія Харьковскаго Математическаго Общества, Вторая серія, Томь XIII, 1913, стр. 1-2.

[4] Bézier, P., *Mathématiques et CAO. Courbes et surfaces,* volume 4, Ed. Hermes, Paris, 1987.

[5] Engeln-Müllges, Gisela, Uhlig, F., *Numerical Algorithms with C*, Springer, New York, 1996, 596 pp., ISBN 3-540-60530-4.

[6] Farin, Gerald, *From conics to NURBS*, IEEE Computer Graphics and Applications, vol. 12, no. 5, 1992, pp. 78-86, (NU)

[7] Farin, G. E., *Curves and Surfaces for Computer Aided Geometric Design. A Practical Guide*, Academic Press, San Diego, sec. ed., 1990, 444 pp., ISBN 0-12-249051-7.

[8] Farin, G., Hoschek, J., Kim, M.-S. (eds.), *Handbook of Computer Aided Geometric Design*, Elsevier, Amsterdam, 2002, 820 pp., ISBN 0-444-51104-0 .

[9] Faux, I. D., Pratt, M. J., *Computational Geometry for Design and Manufacture*, John Wiley, New York, 1985, 331 pp., ISBN 0-470-27069-1.

[10] Foley, J. D., van Dam, A., Hughes, J. F., *Computer Graphics, Principles and Practice*, 2/e in C, Addison-Wesley, 1996, 1200 pp.

[11] Forrest, A. R., *Interactive interpolation and approximation by Bézier polynomials*, Computer Aided Design, 1990, vol. 22, no. 9, 1990, pp. 527-537

[12] Lu, L., *Sample-based polynomial approximation of rational Bézier curves*, Journal of Computational and Applied Mathematics, 2011, Vol. 235, pp. 1557-1563

[13] Lyche, T., Schumacher, L. L. (ed.), *Mathematical Methods in Computer Aided Geometric Design*, Academic Press, San Diego, 1989, 611 pp., ISBN 0-12-460515-X

[14] Micula Gh., *Functii spline şi aplicaţii*, (*Spline Functions and Applications – in Romanian*), Editura Tehnică, Bucureşti, 1978, 336 pp.

[15] Micula, Gh., Micula Sanda, *Handbook of Splines, Springer Science & Business Media, 2012, 606 pp., ISBN 978-94-011-5338-6*

[16] Nischwitz, A., Haberäcker, P., *Masterkurs Computergrafik und Bildverarbeitung,* Friedrich Viewig & Sohn Verlag, Wiesbaden, 2004, 860 s., ISBN 3-528-05874-9.

[17] Popescu, Diana Ioana, *Programare în limbajul C (Programming in C- in Romanian)*, Ed. "DSG Press", Dej, 1999, 288 pp., ISBN 973-98621-4-4.

[18] Popescu, Diana Ioana, *Aplicaţii cu SolidWorks. CAD în ingineria mecanică (Applications with SolidWorks. CAD in Mechanical Engineering – in Romanian),* Editura Dacia, Cluj-Napoca, 2003, 191 pp., ISBN 973-35-1728-3.

[19] Pozrikidis, C., *Numerical Computation in Science and Engineering*, University Press, Oxford, 1998, 640 pp., ISBN 0-19-511253-9

[20] Rogers, D. F., Adams, J. A., *Mathematical Elements for Computer Graphics*, McGraw Hill, Boston, sec. ed., 1990, 611 pp., ISBN 0-07-053530-2.

[21] Стечкин, С. Б., Субботин, Ю. Н., *Сплайны в вычислительной математики* (*Splines in Numerical Computing – in Russian*), Наука, Москва, 1976, 248 стр.

[22] Staerk, E., *Mehrfach differentzierbare Bézierkurven und Bézierflächen*, PhD Thesis, Technische Universität Braunschweig, 1976.

[23] Stelia O., Potapenko, L., Sirenko, I., *On the one method of a third-degree Bézier type spline curve construction*, 8 pp., https://arxiv.org/pdf/1712.07485.

[24] Ursu-Fischer, N., Ursu, M., *Programare cu C în inginerie (Programming with C in Engineering – in Romanian*), Casa Cărţii de Ştiinţă, Cluj-Napoca, 2001, 405 pp., ISBN 973-686-227-5.

[25] Ursu-Fischer, N., Ursu, M., *Metode numerice în tehnică şi programe în C/C++* (*Numerical Methods in Engineering and Programs in C/C++ - in Romanian*), vol. II, Casa Cărţii de Ştiinţă, Cluj-Napoca, 2003, 288 pp., ISBN 973-686-464-2.

[26] Ursu-Fischer, N., Ursu, M., *Metode numerice în tehnică (Numerical Methods in Engineering – in Romanian),* Casa Cărţii de Ştiinţă, Cluj-Napoca, 2018, approx. 900 pp. (to appear)

[27] Xiao, G., Xu, X., *Study on Bézier curve variable step-length algorithm*, Physics Procedia, 2012, Vol. 25, pp. 1781-1786.

Interpolare spline cu funcţii Bézier de gradul trei

Rezumat: Curbele Bézier , B-spline şi NURBS utilizate în cadrul proiectării asistate a formelor sunt în atenţia unui număr mare de cercetători, din punct de vedere matematic fiind utilizate atât în probleme de interpolare cât şi de aproximare. Colectivul nostru a studiat şi soluţionat problema interpolării de tip spline utilizând curbe Bézier de gradul trei între punctele de interpolare. A fost stabilit un algoritm pentru formarea elementelor modelului matematic de rezolvare, foarte eficient, care a fost programat în cadrul unui program C, cu care s-au realizat o serie de exemplificări ale procedeului găsit. Diagramele prezentate arată că există mai multe variante de curbe de interpolare care îndeplinesc condiţiile impuse, deoarece numărul acestora este mai mic decât cel al necunoscutelor.

Nicolae URSU-FISCHER, Prof. dr. eng .math., Technical University of Cluj-Napoca, Department of Mechanical Systems Engineering, E-mail: nic_ursu@yahoo.com, Phone: 0264-401659

Diana Ioana POPESCU, Prof. dr. eng., Technical University of Cluj-Napoca, Department of Mechanical Systems Engineering, E-mail: diana.popescu@mep.utcluj.ro, Phone: 0264-401783

Ioan RADU, Prof. dr. eng., “Aurel Vlaicu” University Arad, Faculty of Engineering, Department of Automation, Industrial engineering, Textile production and Transport, E-mail: raduioanuav@gmail.com, Phone: 0257-283010