# INCREASING THE FLEXIBILITY AND APPLICATION OF THE B- SPLINE CURVE

**Julijana Citkuseva[1*], Aleksandra Stojanova [2], Elena Gelova[3]**

[1] *"Goce Delcev" University, julijana.citkuseva@ugd.edu.mk*
[2] *"Goce Delcev" University, aleksandra.stojanova@ugd.edu.mk*
[3] *"Goce Delcev" University, elena.gelova@ugd.edu.mk*

[*]Julijana Citkuseva, email: julijana.citkuseva@ugd.edu.mk
[*]Elena Gelova, email: elena.gelova@ugd.edu.mk

**Abstract:** One of the main tasks of geometrical modeling is to find ways of designing and representing freeform curves and surfaces. One of well-known geometric modeling tools in computer aided geometric design are B-spline curves and surfaces. B-spline is smooth in parts polynomial functions with reduced smoothness. In this article we will give ways of increasing the flexibility of the B - spline curve as well as its applications together with associated examples made using the program package Wolfram Mathematica 8.

**Keywords:** b-spline model, knot vector, control polygon, Bezier curve, multiple knot.

# 1  Introduction

A spline curve is a sequence of curve segments that are connected together to form a single continuous curve. B-spline curve addresses problems with the Bezier curve. It provides the most powerful and useful approach to curve design available today.

Freeform curves and surfaces have very broad application. Thus, Bezier-curves are used to draw the path of motion of a point (object). In 3D animations, these curves are used to determine the three-dimensional path and two-dimensional curve for interpretation of key segments. B –splines are used for modeling animation characters for the purposes of modeling animated movies and True Type fonts, as illustrated in this article.

B spline models were first mentioned by Schoenberg [19]. Initially, the practical application of B-spline curves and surfaces was limited by the instability of the algorithms for their calculation. This setback was resolved by Cox and de Boor [6]  who proposed algorithm for numerical computation based on recursive template. Then, Reisenfeld [19] revealed that B-spline is a powerful tool for representing free-form shapes and applied B spline base for curves definitions.

## 1.1 Organization of the paper.

First we will give a brief overview of methods that are used for increasing flexibility of B-spline curves emphasizing two techniques: degree raising and knot inserting. Then, we give two examples of B-spline application.

## 2  Increasing the flexibility of the B- spline curve

If $S_k = \begin{bmatrix} B_0^k(t) & B_1^k(t) & ... & B_n^k(t) \end{bmatrix}^T$ is B spline base defined in vector form, then,

$$S_k(P;t) = S_k^T \cdot P = \sum_{i=0}^{n} B_i^k(t) P_i, \quad t \in [t_{k-1}, t_{n+1})$$

is B spline curve defined with $P = [P_0 \quad \cdots \quad P_n]^T$.

It is clear that the choice of base functions and choice of knot vectors significantly affects the shape of the B-spline curve.

To obtain B-spline curve with greater flexibility two techniques are used:
- Raising the degree and
- Inserting knots (subdivision).

The advantage of raising a degree is that B-spline curve retains smoothness and while applying subdivision decreases differentiability of embedded knots, which depends on multiplicity of elements in the knot vector. If the knot vector doesn't have multiple elements, application of subdivision reduces differentiability to k-2 order.

By raising the degree of B-spline curve, the new curve should be identical to the original curve, i.e.

$$\mathbf{S}_k\,(\mathrm{P};t) = \sum_{i=1}^{n+1} B_i^k(t)\mathrm{P}_i = \sum_{i=1}^{m+1} B_i^{k+1}(t)\mathrm{P}_i^*,$$

where $\mathrm{P}_i^*$ are points of the control polygon for the new curve.

By raising the degree or order of B-spline curve, it becomes smoother compared to the control polygon. Also, the appearance of multiple internal elements in the vector of knots for B spline base functions, means a reduction of continuity in the value for the knot, and drawing the resulting B-spline curve closer to a control polygon. These effects indicate that there are multiple internal elements in the knot vector.

The flexibility of B - spline curve also increases by inserting additional elements in the knot vector. Adding a single element in the knot vector is called knot insertion, and insertion of multiple elements in the knot vector is called a knot refining.

There are two basic methods of inserting knots. The first method contains the so-called Oslo algorithm developed by Cohen [19] and colleagues and Prautzch algorithm, and consists of simultaneously inserting multiple elements in the knot vector. The second method, developed by Boehm [19] consists of repeatedly inserting single element in the knot vector.

Let

$$\mathbf{S}_k\,(\mathrm{P};t) = \sum_{i=1}^{n+1} B_i^k(t)\mathrm{P}_i$$

is an initial curve with the knot vector
$$\tau = [t_1 \quad t_2 \quad \cdots \quad t_{n+k+1}].$$

The new curve, which is obtained after the insertion of knots is defined by
$$\boldsymbol{R}_k\,(\mathrm{P};s) = \sum_{j=1}^{m+1} M_j^k(s)\mathrm{C}_j$$

Where $\tau^* = [y_1 \quad y_2 \quad \cdots \quad y_{m+k+1}],\ m > n$ is the new knot vector and $\boldsymbol{S}_k\,(\mathrm{P};t) = \boldsymbol{R}_k\,(\mathrm{P};s)$.

To determine the new points of the control polygon Oslo algorithm is used.
$$\mathrm{C}_j = \sum_{i=1}^{n+1} \alpha_{i,j}^k \mathrm{P}_i \quad 1 \le i \le n, \quad 1 \le j \le m,$$

where $\alpha_{i,j}^k$ are given with recursive relation
$$\alpha_{i,j}^1 = \begin{cases} 1 & t_i \le y_j < t_{i+1} \\ 0 & \text{otherwise} \end{cases},$$

$$\alpha_{i,j}^k = \frac{y_{j+k-1} - t_i}{t_{i+k-1} - t_i}\,\alpha_{i,j}^{k-1} + \frac{t_{i+k} - y_{j+k-1}}{t_{i+k} - t_i}\,\alpha_{i+1,j}^{k-1},$$

where $\sum_{i=1}^{n+1} \alpha_{i,j}^{k} = 1$.

After inserting the value for the knot, if the original knot vector is uniform, periodic or open, then the end vector nodes are non-uniform. Uniform knot vector can be obtained by inserting multiple values for knot in the middle of each not null interval.

**Example 1.** Consider B spline curve of order k = 3 defined by four points of the control polygon

$$P_1(0,0), \quad P_2(1,4), \quad P_3(3,2), \quad P_4(4,3)$$

and open uniform knot vector $\tau = [0\ 0\ 0\ 1\ 2\ 2\ 2]$.

We will perform subdivision of curve, inserting the value 1 in the knot vector between knots 0 and 1, i.e. the knot vector is $\tau^* = [0\ 0\ 0\ 1\ 1\ 2\ 2\ 2]$. So, starting from uniform, non-uniform open knot vector with one double knot is obtained. Using the Oslo algorithm, the new control points of the polygon are obtained.
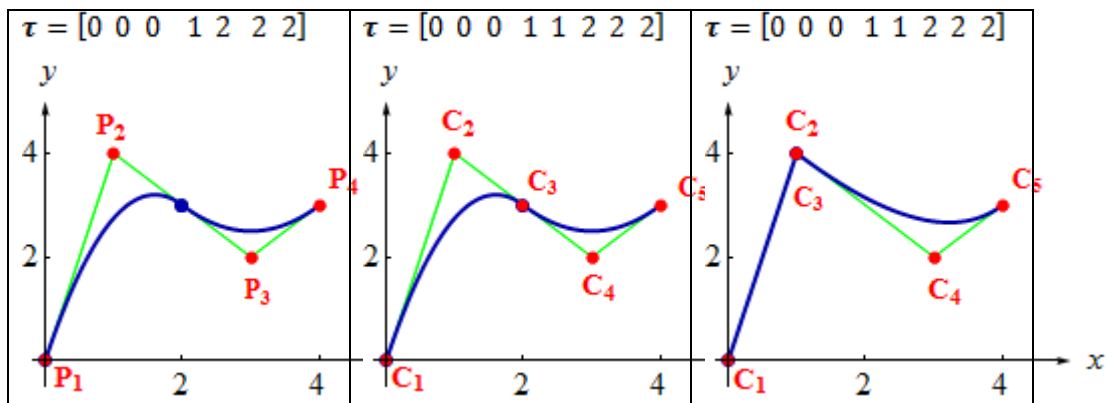
$$C_1 = \alpha_{3,1}^3 P_1 = P_1 = (0,0),$$
$$C_2 = \alpha_{3,2}^3 P_2 = P_2 = (1,4),$$
$$C_3 = \alpha_{2,3}^3 P_2 + \alpha_{3,3}^3 P_3 = \frac{1}{2}(P_2 + P_3) = (2,3),$$
$$C_4 = \alpha_{3,4}^3 P_3 = P_3 = (3,2),$$
$$C_5 = \alpha_{4,5}^3 P_4 = P_4 = (4,3).$$



**Figure 1**. Impact of change in the control point polygon

**Figure 1:** Left, shows the B - spline curve obtained from the initial control polygon, with open uniform vector nodes. The graph in the middle shows the curve after the subdivision. We note that the curve does not change, but the control polygon is changed and now it has five points, and the knot vector is changed too. If points $C_3$ and $C_2$ match, then despite the double element in the knot vector corresponding to $C_2 = C_3$, there is a double point in the

control polygon and the resulting curve will peak in the double knot , as presented in Figure 1 . Right.

## 3  Application of B-spline

B splines are very widely used in computer design. Below we give two examples of their application.

**Example 2.** B splines can be used for drawing animated characters for animated movie's needs. The draw for the various parts of the animated character required different control polygons. In this example the animated character of squirrel is drawn, starting from three control polygons. The first defines the body, the second eye, and the third right ear of squirrel:

**P-body** = {(8,138), (8,138), (22,148), (55,105), (80,132), (52,146), (20,171), (100,210), (153,185), (150,165), (163,168), (180,140), (152,122), (138,144), (120,92), (169,94), (265,151), (311,112), (324,85), (313,74), (342,92), (320,142), (238,161), (225,187), (285,212), (322,212), (326,203), (433,201), (487,159), (487,86), (393,53), (319,64), (316,50), (289,29), (225,27), (265,53), (218,50), (167,18), (95,6), (60,60), (43,105), (8,138), (8,138)};

**P- eye** = {(102,162), (102,162) ,(82,181), (59,170), (77,158), (102,162), (102,162)};

**P- ear** = {(148,183), (148,183), (173,191), (193,169), (171,157), (171,157)}.

On Figure 2, right in the front row is represented figure of squirrel derived from B-spline curve of order k = 2, (control polygons). The remaining graphs can be obtained see B-spline curves of order k = 3,4,5,6 and 7.



$k = 2$  $k = 3$  $k = 4$  $k = 5$
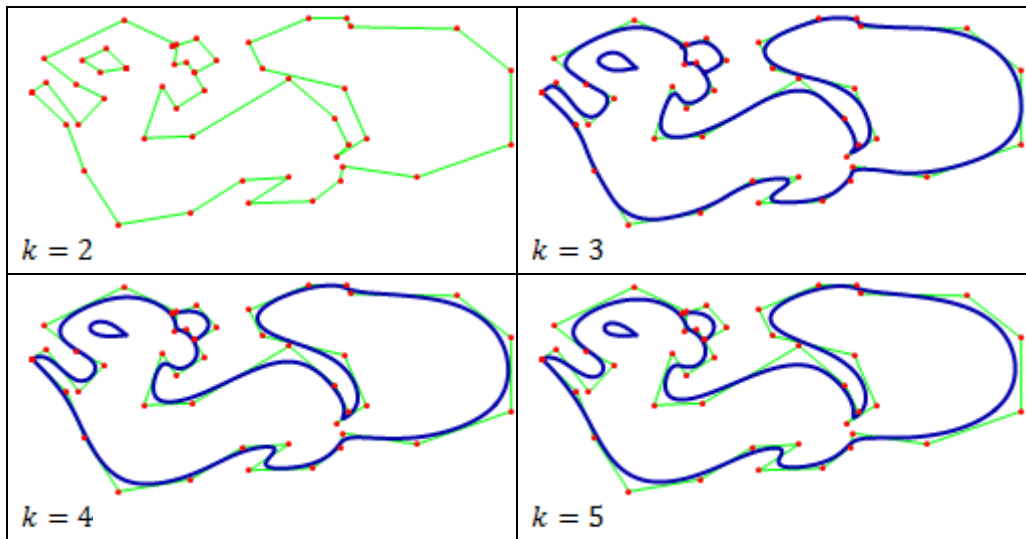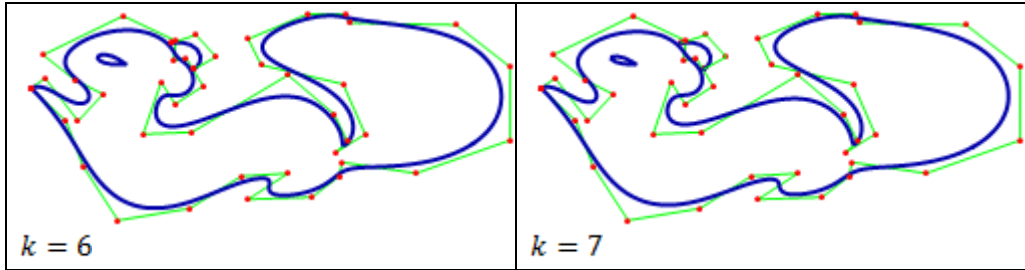
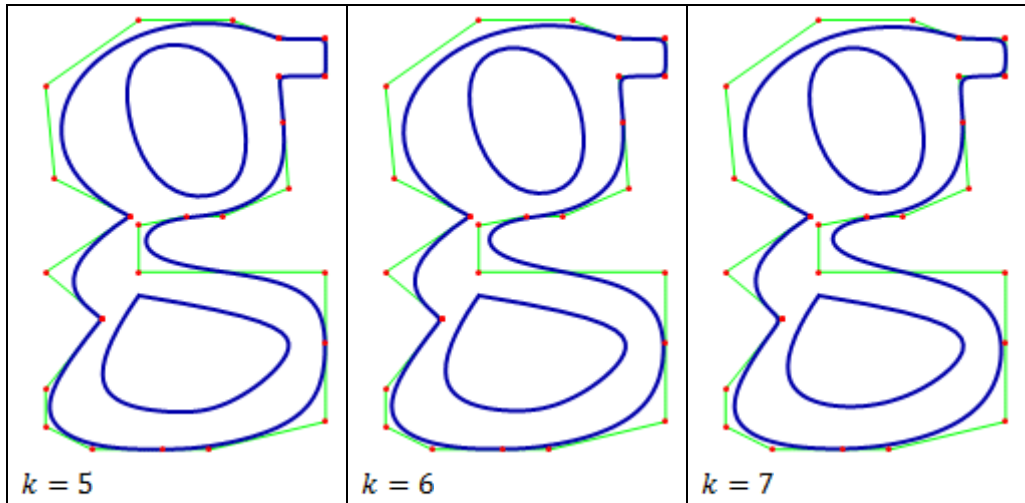**Figure 2**. Application of B-splines for drawing squirrel

**Example 3**. This example will illustrate the application of B-spline for modeling True Type and Post Script fonts, modeling the letter g. First we define the control polygons:

**P1** = {(2.8, 5.2), (1, 4), (2.2, 3), (2.2, 3), (2.2, 3), (1, 1.5), (1, 0.7), (2, 0.2), (3.5, 0.2), (4.5, 0.2), (7, 0.8),  (7, 2.5), (7, 4), (3, 4), (3, 5), (4, 5.2), (4.8, 5.2), (6.2, 5.8), (6.1, 7.2), (6, 8.2), (6, 8.2), (7, 8.2), (7, 8.2), (7, 9), (7, 9), (6, 9), (6, 9), (5, 9.4), (3, 9.4), (1, 8), (1.2, 6), (2.8, 5.2)};

**P2** = {(3, 3.5), (3, 3.5), (1.5, 1.5), (3, 1), (4, 1), (5, 1), (6.8, 2.5), (6, 3), (3, 3.5)};

**P3** = {(5, 5.5), (5.5, 6.5), (5, 9), (3, 9), (2.5, 8), (3.2, 5.5)}.
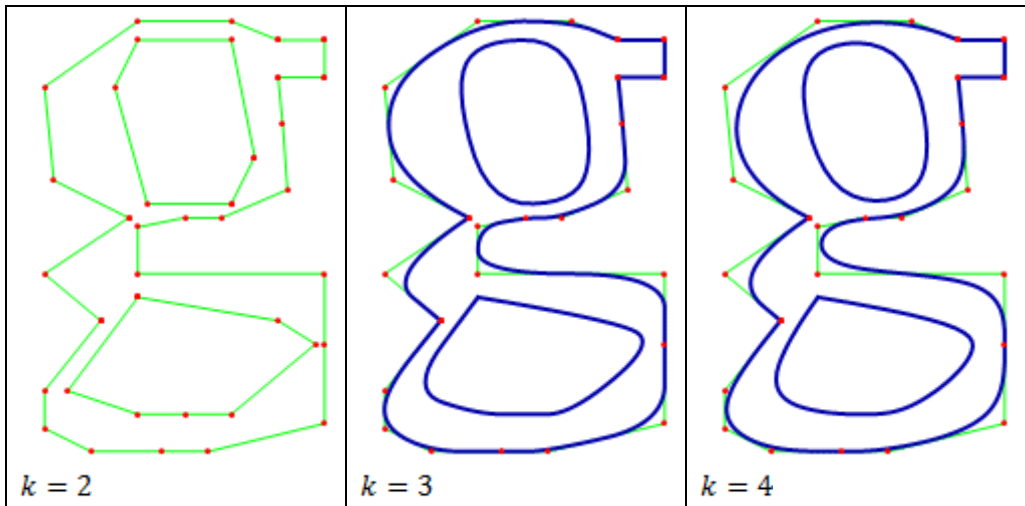
**Figure 4**. Application of B-splines for modeling fonts

**Figure 4:** shows the obtained B-spline curves of order k = 2, 3, 4, 5, 6 and 7.

## 4  Conclusion

B-splines are not used very often in 2D graphics software but are used quite extensively in 3D modeling software. They have an advantage over Bézier curves in that they are smoother and easier to control. B-splines consist entirely of smooth curves, but sharp corners can be introduced by joining two spline curve segments. The continuous curve of a b-spline is defined by control points. While the curve is shaped by the control points, it generally does not pass through them.

Application of the free-form curves is not limited to computer graphics, only. They have an important application in the calculation of optimal orbit and trajectory of aircraft flight. The most commonly used are Bézier and B-spline curves, but there are other splines as Hermit spline, Catmull-Rom spline and T-spline.

## References

[1]  Anant U.*, Shape-preserving Interpolation with Biarcs and NURBS*, Doctoral Thesis, Department of Computer Science University of Manitoba Winnipeg, Manitoba, Canada, 2009.
[2]  Andersson F., *Bezier and B-SplineTechnology*, Umea university Sweden, 2003.
[3]  Casiola G., Morigi S., *Reparametrization of NURBS curves*, International Journal of Shape Modelling, Vol. 2, No. 2&3 (1996), 103-116.
[4]  Casiola G., *A System to Model and Render NURBS curves and Surfaces*, University of Bologna, 2000.
[5]  Coxeter H.S.M., *Introduction to Geometry*, second edition, John Wiley & Sons, INC, 1969.

[6]   de Boor C., B-Spline Basics, web page http://128.105.2.28/debooron/bsplbasic.pdf.
[7]   Dyn N., *Linear Subdivision Schemes for Refinement of Geometric Objects*, http://www.math.tau.ac.il/~niradyn/papers/final.pdf , 2002.
[8]   Dyn N., Levin D., *Subdivision schemes in geometric modeling*, Acta Numerica, 11 , pp 73-144 doi:10.1017/S0962492902000028 (2002).
[9]   Gallier J., *Geometric Methods and Applications for Computer Science and Engineering*, Springer-Verlag, TAM, Vol.38, 2000.
[10]  Hu S. M., Li Y. F., Ju T., Zhu X., , *Modifying the shape of NURBS surfaces with geometric constrains*, Computer Aided Design 33 (2001) 903-912.
[11]  Iglesias      A.,      *B-splines      and      NURBS      Curves      and      Surfaces,* http://etsiso2.macc.unican.es/~cagd.
[12]  Iglesias A., Gutiérrez F., Gálvez A., *A Mathematica Package for CAGD and Computer Graphics*, web page      http://education.siggraph.org/conferences/eurographics/gve-99/proceedings/papers/gve99-a-iglesias.pdf.
[13]  Kocić Lj. M., *Geometrijsko modeliranje*, Univerzitet u Nišu, Elektronski Fakultet, 2010.
[14]  Levin D., Dyn N., *Subdivison Schemes in Geometric Modelling*, Tel-Aviv University, Izrael, 2002.
[15]  Lowther J., Houghton S., *Teaching B-splines Is Not Difficult*, Department of Computer Science Michigan Technological University MI 49931–1295.
[16]  Mortenson M. E., *Geometric Transformations for 3D Modeling*, Industrial Press Inc., 2. Edition, 2007.
[17]  Piegl L., Tiller W., *The NURBS Book*, Springer 1997.
[18]  Piegl L., Tiller W., *Curve and surface constructions using rational B-splines*, Comput. Aided Design **19** (1987), No. 9, 485-498.
[19]  Rogers D. F., *An Introduction to NURBS: with historical perspective*, Academic Press, 2001.
[20]  Sederberg T. W., *Computer Aided Geometric Design Course Notes,* Department of Computer Science, Brigham Young University, 2011.