

PROJECTING A HYDROGRAPHIC MAP OF THE REPUBLIC OF MACEDONIA

Vesna Andova, Sanja Atanasova, Elena Jovčevska, Viktorija Jordanova, Ilin Tolovski, Martin Rizov

*Faculty of Electrical Engineering and Information Technologies,
"Ss. Cyril and Methodius" University in Skopje,
Rugjer Bošković bb, P.O. box 574, 1001 Skopje, Republic of Macedonia
vesnaa@feit.ukim.edu.mk*

Abstract: Bezier and B-spline curves are among the most powerful tools used for complex graphical approximation. In this paper, we will use them to recreate the hydrographic map of the Republic of Macedonia. The results obtained in this paper show that cubic spline curves have smaller average deviation in respect to B-spline curves, in contrast to Bezier curves. The results and the images are obtained by using the software package Wolfram Mathematica.

Key words: Bezier curves; B-spline; cubic spline

МОДЕЛИРАЊЕ НА ХИДРОГРАФСКА МАПА НА РЕПУБЛИКА МАКЕДОНИЈА

Апстракт: Безиеовите и Б-кривите спаѓаат во групата најмоќни алатки кои се користат за апроксимација на комплексни графички прикази. Во трудот се користени овие криви за моделирање на хидрографска мапа на Р. Македонија. Резултатите во овој труд покажуваат дека кривите добиени со кубна крива имаат најмало просечно отстапување во однос на Б крива и Безиеовите криви. Резултатите и сликите се добиени со помош на софтверскиот пакет Математика.

Клучни зборови: Безиеови криви, Б-крива, кубна крива

1. INTRODUCTION

Since their introduction in the 1950's and 1960's, Bezier and B-spline curves have quickly become one of the most powerful and sought after tools in many fields where surface approximation is an imperative such as computer graphics, molecular and physical models, 3D animations and many more [7], [11].

This is the case, mostly because the idea behind these mathematical models is quite simple. They both rely on choosing set of control points which the curves will follow. Although the idea looks simple, the mathematical background is not. After choosing the control points, there are number of other properties that need to be set in order to get the desired outcome. Some of these properties are: degree of the curve, weight of the points, control knots etc.

The concept of Bezier curves was introduced in the 1910's by Charles Hermite and Sergei Bernstein. However, it was not until the 1950's when Pierre Bezier and Paul de Casteljaou brought this concept to life and introduced it to graphic specialists. These curves are based on the Bernstein polynomials, and can be calculated recursively either with developing the Bernstein polynomials or with the Casteljaou algorithm [8].

B-spline curve, on the other hand, represents a generalization of a Bezier curve. They are divided into multiple polynomial pieces and depending on the uniformity of these pieces there are two cases, uniform B-splines and NURBS (Non-uniform rational B-spline) [14], [10]. We will later consider both types.

In Section 2 we will explain the equations and all the steps of calculating both B-spline and Bezier curves. In Section 3 we use the B-spline,

Bezier and cubic spline curves to construct a hydrographic map of the Republic of Macedonia. Furthermore, we compare the curves, measure the percentage difference in lengths of the water areas (rivers, springs, not lakes and closed waters), in order to find out which curve will give the best approximation for this problem.

For the purposes of this project we use Matlab for obtaining the coordinates and Wolfram Mathematica for all of the modeling, approximations and calculations. Mathematica's built-in libraries and methods make it extremely powerful and pleasant to use for this sort of problems.

2. DEFINITIONS AND EQUATIONS

In this section we give the basic definitions and relations for Bezier and B-spline curves. For all the terms not defined here consult [15] or any other book from this field.

Bezier curve

As we mentioned, Bezier curves are based on Bernstein polynomials [12] of degree n , which are represented with

$$B_i^n(t) = n_i t^i (1-t)^{n-i}, \quad i = 0, \dots, n. \quad (1)$$

The computation of the Bernstein polynomials can be also displayed in triangular scheme, shown with equation (2):

$$\begin{array}{cccccc}
 B_0^n & B_1^n & B_2^n & \dots & B_n^n \\
 & B_1^n & B_2^n & \dots & B_n^n \\
 1 = & & B_2^n & \dots & B_n^n \\
 & & & \ddots & \vdots \\
 & & & & B_n^n
 \end{array} \quad (2)$$

These polynomials are defined for all $t \in \mathbb{R}$, but in practice they are usually restricted to $t \in [0,1]$. One of the most important properties of Bernstein polynomials is that they are linearly independent [1]. From relation [1] we can see that Bernstein polynomials are also symmetrical, i.e.

$$B_i^n(t) = B_{n-i}^n(1-t). \quad (3)$$

Moreover, the Bernstein polynomials can be calculated with a recursive formula, shown in equation (4).

$$B_i^n(t) = t B_{i-1}^{n-1}(t) + (1-t) B_i^{n-1}(t). \quad (4)$$

The main purpose of Bezier curves is modeling a whole curve or some part of it. When this is done in a Euclidean space, it is represented as a parametric polynomial defined in equation (5):

$$p(t) = \sum_{i=0}^n a_i t^i, \quad a_i \in \mathbb{R}^d. \quad (5)$$

Equation (5) also has a Bernstein representation:

$$p(t) = \sum_{i=0}^n c_i B_i^n(t), \quad c_i \in \mathbb{R}^d. \quad (6)$$

Equation (6) defines a Bezier curve with control points c_i . This curve is restricted to the parameter domain $[0,1]$ which makes p a parametric curve $p: [0,1] \rightarrow \mathbb{R}^d$. The control points c_i when connected are forming the control polygon of the curve. The shape of this polygon represents a shape that the Bezier curve will have when constructed. This property can easily be manipulated which is one of the reasons why Bezier curves are suitable for dynamic change of shape. Bezier curves inherit several properties of Bernstein polynomials, such as the symmetry. From the triangular scheme of the polynomials we have that they sum to 1, which means that every point $p(t)$ is an affine combination of the control points $c_i, i \in \{0, 1, \dots, n\}$ [3], [4]. Because the polynomials are also non-negative in the interval $[0,1]$, every point $p(t)$ is a convex combination of the control points $c_i, i \in \{0, 1, \dots, n\}$. Consequently, the Bezier curve p lies in the convex hull of its control points [6].

B-spline curves

B-spline curve $p(t)$ is analytically defined by

$$P(t) = \sum_{i=0}^n P_i N_{i,k}(t),$$

where $t \in [t_{k-1}, t_n]$, P_i are the control points, k is the order of the polynomial segments of the B-spline curve and $N_{i,k}(t)$ are normalized B-spline blending functions [5]. Order k means that the curve is made up of piecewise polynomial segments of degree $k - 1$. The $N_{i,k}$ functions are described by the order k and by a non-decreasing sequence of real numbers t_i , where $i \in \{0, 1, \dots, n + k\}$ which is referred as knot sequence [9]. The complete equations for $N_{i,k}$ are shown with (7) if $k = 1$ and with (8) if $k > 1$,

$$N_{i,1}(t) = \begin{cases} 1, & t \in [t_i, t_{i+1}) \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

$$N_{i,k}(t) = \frac{t-t_i}{t_{i+k-1}-t_i} N_{i,k-1}(t) + \frac{t_{i+k}-t}{t_{i+k}-t_{i+1}} N_{i+1,k-1}(t) \quad (8)$$

A B-spline curve is geometrically defined by the relation (9), where the set of control points is $P_0, P_1, P_2, \dots, P_n$. Then the B-spline curve $P(t)$ is defined by

$$P(t) = P_l^{k-1}(t), \quad t \in [t_l, t_{l+1}) \quad (9)$$

where

$$P_i^{(j)}(t) = \begin{cases} (1-\tau_i^j)P_{i-1}^{j-1}(t) + \tau_i^j P_i^{j-1}(t), & j > 0 \\ P_i, & j = 0 \end{cases}$$

and

$$\tau_i^j = \frac{t-t_i}{t_{i+k-j}-t_i}.$$

3. COMPARISON BETWEEN THE CURVES

a) Bezier or B-spline curves?

In this section we state some of the main differences between Bezier and B-spline curves.

- 1) Bezier curves can be viewed as a parametric curve consisting of basic functions which are Bernstein polynomials, and on the other hand, B-splines are based on piecewise Bezier curves.
- 2) Bezier curves lacked in local control. This means that if we add or remove one control point, it will effect the hole curve. With B-splines we have more freedom and they provide more control flexibility.
- 3) Bezier curves are more suitable and give better approximation if the control points are nearly collinear. This means that when modeling some curve, we can make combination of both Bezier and B-spline curve depending on the choice of control points.
- 4) Bezier curves are special cases of B-spline curves. If $n = p$ (i.e. the degree of a B-spline curve is equal to n , the number of control points minus 1), and there are $2(p+1) = 2(n+1)$ knots with $p+1$ of them clamped at each end, this B-spline curve reduces to a Bezier curve. To simplify, any Bezier curve of arbitrary degree can be converted into a B-spline and

any B-spline can be converted in to one or more Bezier curves [13].

- 5) B-spline curves require more information and a more complex theory than Bezier curves, which means more time for running the code when modeling.
- 6) B-spline curves satisfy all important properties that Bezier curves have (affine invariance, positivity, variation Diminishing Property, the convex hull property any many others).
- 7) B-splines do not interpolate any of its control points, while the Bezier curve automatically clamps its end points. However, B-splines can be forced to interpolate any of its n control points without repeating it, which is not possible with the Bezier curve [2], [16].

b) Obtaining coordinates from the original hydrographic map

In order to be able to construct a hydrographic map with B-spline, Bezier and cubic spline curves, we needed to successfully select all of the water areas inside the Republic of Macedonia. We did just that by using an official 2600×1960 pixel hydrographic map (shown on Figure 1) designed by Prof. Dr. Ivica Milevski from the Institute of Geography at the Ss. Cyril and Methodius University in Skopje. For obtaining the coordinates, we used a custom Matlab script based on the *ginput* function. Even though bigger number of points would result in more precisely constructed map, we concentrated on selecting the right points, rather than every point. This way, the results would represent genuine differentiation between the B-spline, Bezier and cubic spline curves where we can clearly see the difference in the ways these curves are built. After obtaining the coordinates, all of the following analyses will be done in Wolfram Mathematica.

c) Constructing the model plots in Mathematica

Mathematica is among the most powerful software packages when it comes to solving geometrical modeling and calculation problems. It contains many built-in libraries and functions which make the coding cleaner, yet the results precise [17]. For constructing the models we used Mathematica's built-in methods BSplineCurve, BezierCurve and SplineFit. Figure 2 shows the plot constructed with B-spline curve.

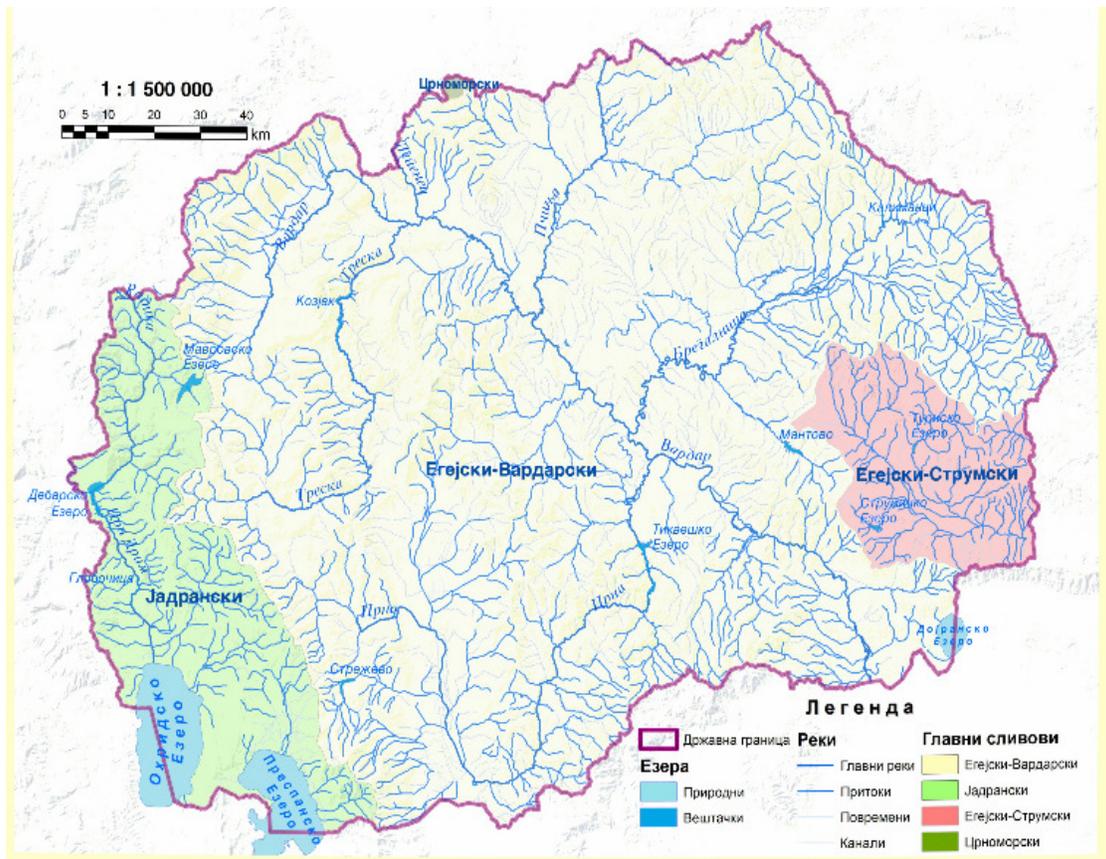


Fig. 1. Hydrographic map of Macedonia

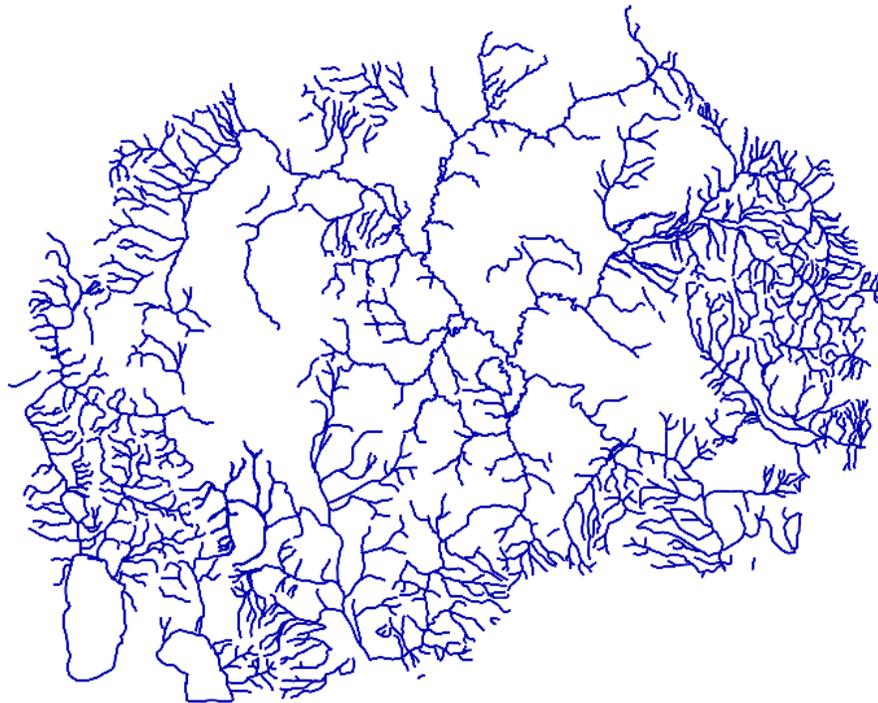


Fig. 2. Hydrographic map of Macedonia plotted with B-spline curves

On Fig. 3 and Fig. 4 you can see the maps constructed with Bezier and cubic spline curves respectively. Alongside these methods we also used some other methods along the course of this project, all of them that are relevant to the geometrical modeling are displayed in Table 1.

The outcome of these functions can be modified even more given the fact that they have many properties that can be tuned in a way that will give the best result for any given problem [18]. In this

case, by modifying the spline degree in the BSplineCurve method it was evident that B-spline with spline degree 2 had the smallest variation from the coordinates of the original hydrographic map compared to the maps plotted with Bezier-curve and spline fit methods, so B-spline of the second degree was chosen as a benchmark to which the other curves will be compared and measured.



Fig. 3. Hydrographic map of Macedonia plotted with Bezier curves

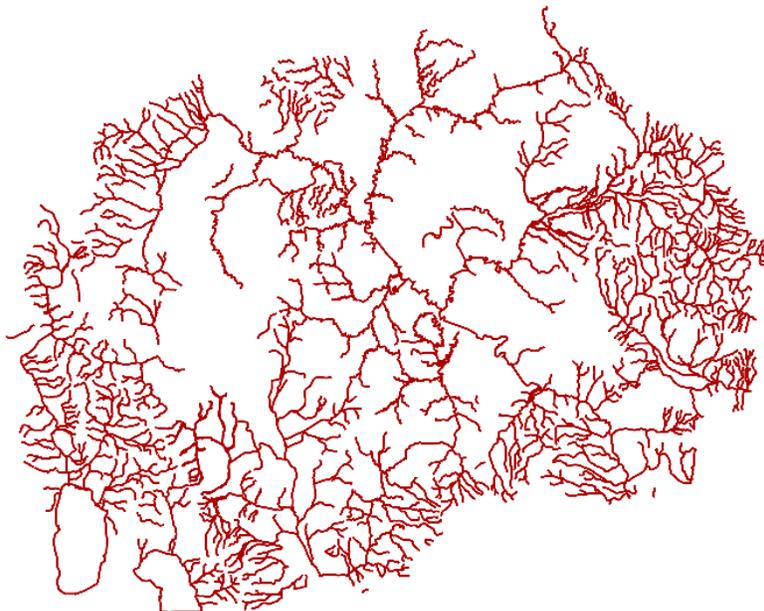


Fig. 4. Hydrographic map of Macedonia plotted with cubic spline curves

Table 1.

Methods used for modeling of the hydrographic map

Method	Description
BSplineCurve [$\{pt_1, pt_2, \dots\}$]	BSplineCurve is a graphics primitive that represents a non-uniform rational B-spline curve with control points p_i, \dots
BSplineFunction [$\{pt_1, pt_2, \dots\}$]	BSplineFunction represents a B-spline function for a curve defined by the control points p_i, \dots
BezierCurve [$\{pt_1, pt_2, \dots\}$]	BezierCurve is a graphics primitive that represents a Bezier curve with control points p_i, \dots
BezierFunction [$\{pt_1, pt_2, \dots\}$]	BezierFunction represents a Bezier function for a curve defined by the control points p_i, \dots
SplineFit [$\{pt_1, pt_2, \dots\}$, type]	SplineFit generates a SplineFunction object of the specified type from the points p_{t1}, p_{t2}, \dots
Spline [$\{pt_1, pt_2, \dots\}$, type]	Spline is a two-dimensional graphics primitive which represents a spline of type <i>type</i> through (or controlled by) points p_{t1}, p_{t2}, \dots
Discretize Graphics [g]	Discretizes a 2D or 3D graphic <i>g</i> into a MeshRegion.
ArcLength[reg]	ArcLength gives the length of the one-dimensional region <i>reg</i> .
ParametricPlot [$\{f_x, f_y\}$, u , u_{\max} , u_{\min}]	ParametricPlot generates a parametric plot of a curve with x and y coordinates f_x and f_y as a function of u .

d) Result measuring and comparison

In order to make a successful comparison with the original hydrographic map, or in this case our reference map which is the one that we construct with B-spline curves, we will measure the deviation in the lengths of all of the water areas that we constructed with B-spline, Bezier and cubic spline curves. First of all, for measuring the length of a single river or water area we use the method ArcLength that was mentioned and described in section *b*. After calculating the lengths of every water area, we calculate the deviation in percentages between the lengths of B-spline curve on one side and lengths of Bezier and cubic spline curves on the other. To do this we use the following formulas:

$$SRD_{CS}(\%) = \frac{|Len_{BS} - Len_{CS}|}{Len_{BS}} \cdot 100 \quad (10)$$

$$SRD_{BZ}(\%) = \frac{|Len_{BS} - Len_{BZ}|}{Len_{BS}} \cdot 100 \quad (11)$$

where Len_{BS} , Len_{CS} and Len_{BZ} represent the lengths of a single river constructed with B-spline, cubic spline and Bezier curves respectively. The result of (10) and (11) gives the percentage deviation for a single river. The smaller the percentage, the bigger the precision of the curve. Because there were big variations in the results for separate rivers, we also calculate the average deviation for all of the rivers. We do that by summing the absolute value of all of the differences between the rivers. Afterwards, we divide that with the sum of the lengths constructed with B-spline. The previous explanation is depicted with the following formulas:

$$ARD_{CS}(\%) = \frac{\sum_{i=1}^{870} |Len_{BS,i} - Len_{CS,i}|}{\sum_{i=1}^{870} Len_{BS}} \cdot 100 = 6.655\% \quad (12)$$

$$ARD_{BZ}(\%) = \frac{\sum_{i=1}^{870} |Len_{BS,i} - Len_{BZ,i}|}{\sum_{i=1}^{870} Len_{BS}} \cdot 100 = 11.027\% \quad (13)$$

Equations (12) and (13) are showing the average deviation when the calculation is done on all 870 water areas that exist in the Republic of Macedonia. From the results we can see that cubic spline curves have smaller average deviation in respect to B-spline curves, in contrast to Bezier curves. Given the fact that B-spline curves represent generalized Bezier curves and additionally cubic spline curves are special case of the B-spline curves this result is expected and logical.

This result was also confirmed by the data obtained when modeling a hydrographical map of Macedonia, for vast number of rivers the cubic spline method gave better results than Bezier curves. One such river is shown on Figure 5. In this case the deviation of the cubic spline is only 5%, while the Bezier curve's deviation is 16.92%.

These two methods are "equally" good for modeling short and smooth rivers. One such river is shown on Figure 6. As the river is short we used small number of points for the modeling. Our result show that the deviation of the Bezier curves is 2.06% and for the cubic spline 1.89% for the river on Figure 7.

When projecting a long river, we have to use a large number of points. At the same time, most of the "long" rivers have a lot of curves, and therefore

the deviation is big for the both methods. One such river is shown on Figure 7. For this river the deviation for the Bezier curves method and for the cubic spline is 12.98% and 11.91%, respectively.

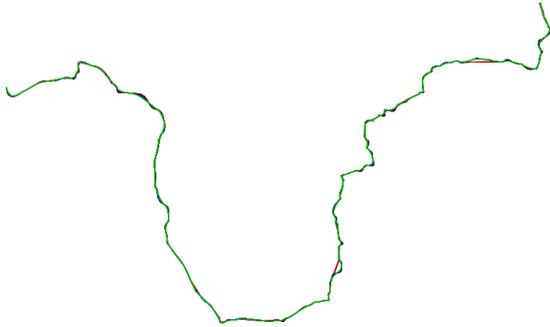


Fig. 5. A river projected by B-spline (blue), cubic spline (red) and Bezier curves (green). The deviation of the Bezier curve is much bigger than the deviation of the cubic spline



Fig. 6. A river projected by B-spline (blue), cubic spline (red) and Bezier curves (green). Bezier curves and cubic spline are "equally" good

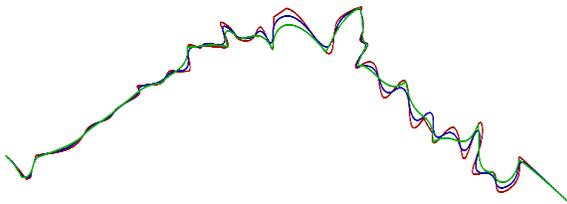


Fig. 7. A river projected by B-spline (blue), cubic spline (red) and Bezier curves (green). Both method have big deviation

Bezier curves gave better results only for "small" rivers (i.e. curves). These rivers are short, do not have many curves, and are modeled with several points which is the main reason for the obtained results. In total, cubic spline curves gave better approximation on 72.068% of the rivers, while Bezier curves gave better approximation on only 27.931% of all 870 measured water areas. The main reason for this gap is that most of the water areas measured are rather long and contain many curves, which given the analyses made for the

rivers shown on Figures 5, 6 and 7 it is intuitive that cubic spline curves will give more precise approximation for the water areas overall.

Despite the bigger precision, cubic spline curves take more time to be calculated, due to the fact that they have more properties and are more advanced than the Bezier curves.

CONCLUSION

When the first approaching this problem, it was straight-forward that it is an approximation problem. Since Bezier and B-spline curves are the cornerstones of geometrical approximation and modeling, we used them as building blocks on which we would construct a hydrographic map of the Republic of Macedonia. While the image manipulation and obtaining of coordinates was done in Matlab, all of the geometrical modeling and calculations were done in Wolfram Mathematica, due to its powerful and robust kernel tools. After plotting the map with B-spline, Bezier and cubic spline curves, we compared the coordinates of each curve with the original hydrographic map, and B-spline was the one whose coordinates were the closest. Therefore, it was taken as a reference curve. After measuring the percentage deviation length of each water area, and then also the average deviation on all of the areas, we concluded that cubic spline curves gave more precise length approximation of the water areas in the Republic of Macedonia compared to Bezier curves. This result was logical from the beginning, given the fact that cubic spline curves are much more flexible due to the bigger number of properties that can be modified in favor of getting a better solution. However, this benefits come with the price of speed, because the computer takes more time to process and calculate cubic spline curves rather than the ones obtained with the Bezier method.

Acknowledgment: The authors would like to thank Prof. Dr. Ivica Milevski from the Institute of Geography at the Ss. Cyril and Methodius University in Skopje for providing the map.

REFERENCES

- [1] Vera B. Anand: *Computer Graphics and Geometric Modeling for Engineers*, John Wiley & Sons Inc. 1996.
- [2] V. Andova, S. Atanasova, K. Bacev, Gj. Peev, G. Kostov: *Approximation of map borders using Mathematica*, *ETAI* (2015).

- [3] R. H. Bartels, J. C. Beatty, B. A. Barsky: *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, Morgan Kaufmann Publishers Inc. 1987.
- [4] R. C. Beach: *An Introduction to the Curves and Surfaces of Computer-Aided Design*, Van Nostrand Reinhold, 1991.
- [5] E. Cohen: *Discrete B-splines and Subdivision Techniques in Computer Aided Geometric Design and Computer Graphics*, International Association for Bridge and Structural Engineering, 1979.
- [6] C. De Boor: *A Practical Guide to Splines*, Revised Edition, Springer, 1978.
- [7] G. Farin: *Curves and Surfaces for CAGD. A Practical Guide*, fifth edition, Academic Press, 2002.
- [8] G. Farin, J. Hoschek, M.-S. Kim: *Handbook of Computer Aided Geometric Design*, Elsevier, 2002.
- [9] G. Farin: *NURBS: From Projective Geometry to Practical Use*, 2nd edition, AK Peters, 1999.
- [10] J. Fiorot, P. Jeannin: *Rational Curves and Surfaces*, Wiley, 1992.
- [11] J. Gallier: Curves and Surfaces. In: *Geometric Modeling: Theory and Algorithms*, Morgan Kofman Publishers, 2000.
- [12] G. G. Lorentz: *Bernstein Polynomials*, University of Toronto Press, 1953.
- [13] R. F. Riesenfeld: *Applications of B-spline Approximation to Geometric Problems of Computer-Aided Design*, Syracuse University, 1973.
- [14] D. Rogers: *An Introduction to NURBS with Historical Perspective*, Elsevier, 2001.
- [15] W. Rudin: *Real and Complex Analysis*, McGraw-Hill Higher Education, 1986.
- [16] D. Salomon: *Curves and Surfaces for Computer Graphic*, Springer, 2006.
- [17] S. Wagon: *Mathematica in Action Problem Solving Through Visualization and Computation*, Springer, 2010.
- [18] S. Wolfram: *The Mathematica Book*, Fifth Edition, Wolfram Media, 2003.