Interpolation Algorithm using Catmull-Rom Spline Function and Operator

Oana-Ancuta Buzura, Rodica Holonec, Petru Marius Maier

Faculty of Electric Engineering, Technical University of Cluj-Napoca, Romania

Abstract *- This article presents the spline interpolation using an operation applied to an array of values and then, with an operator, a smooth curve is drawn through data points. In the first section, it is described in few words why the spline function is a very efficient tool in interpolation. The next section is a brief review of spline interpolation. The third section presents the mathematical model for Catmull-Rom spline function and reveals the algorithm for development of a Catmull-Rom spline operator structure. Last section presents an application of Catmull-Rom spline interpolation, using the operator built using virtual instrumentation. The Catmull-Rom spline is a particular case of cubic splines family and makes it possible to achieve a smooth close curve (isoline) for many practical applications.*

Keywords –*spline function, interpolation, Catmull-Rom spline, spline operator, isolines.*

1. INTRODUCTION

The word “spline” actually refers to a thin strip of wood or metal and several heavy pieces which anchor the flexible strip, so the original spline was not mathematical. The strip curved depends on where these heavy pieces were placed. Later, spline function received a mathematical expression.

When is should to model different phenomena (physical, economic, social), must to manipulate unknown functions often defined by experimentally obtained data [1]. It is often needed to estimate some values that are between two know data. For their determination, it is necessary to find an analytic form, for those functions, which must be simple and easy to integrate in an algorithm [2].

The estimation of some values, between two known values, can be made by interpolation or approximation. The difference between those two operations is represented by the criterions. If the chosen criterion requires that the function has to pass through all known points, the operation is called interpolation and if the chosen criterion requires that the function has to approximate as well is possible, the operation is called approximation [3].

One of the most used classes of interpolation functions is the class of polynomials, because it can have the integral and derivative calculated easily. Polynomials have played a central role in interpolation theory and numerical analysis.

Designing algorithms for complicated electrical engineering problems, such as transient analysis circuits, or problems of electromagnetic field, requires understanding of numerical algorithms for interpolation of functions and their derivatives [4].

A major drawback in terms of a single polynomial interpolation for entire interval, in other words applying a global form of interpolation, is that it is not convergent. Moreover, even if interpolation function converges, increasing the number of knots requires increasing the accuracy of approximation and leads to significant increases in the volume of calculations, which, secondarily, by propagation of errors results in increased approximation errors [2].

This inconvenience can be eliminated by using polynomial interpolation on piecewise (subdivision) of entire interval. Those functions are characterized by the same shape on each piecewise and by the connection conditions in knots. An important class of functions that often occurs in numerical analysis is the class of polynomial functions on piecewise interval. Spline functions are a class of polynomial functions on intervals and the most popular and most commonly used in practice spline function is the cubic spline [5]. Those functions have its first and second derivatives continuous on the larger interval.

2. SPLINE INTERPOLATION

The definition of spline function is presented in [6], so it is a piecewise polynomial function satisfying continuity conditions between the knots. In same paper, it is demonstrated that the interpolation function, that has the particularity to converge towards continuous function that is interpolated, is the spline function. The degree of the spline function is given by the degree of the polynomial.

2.1. Mathematical definition of spline function

If is considered a function *s(x)* defined in the interval [a,b], with real value:

$$s(x): [a, b] \to \mathcal{R} \quad (1)$$

and the knots defined as x_i, i=$\overline{0,n}$ with specification that first and last knots will be the endpoint of interval, then can be written:

$$a = x_0 < x_1 < x_2 < \cdots < x_i < x_{i+1} < \cdots < x_{n-1} < x_n = b \quad (2)$$

Between those knots are (n-1) interpolation intervals as

$I_i = [x_i, x_{i+1}]$.

Polynomial expression of spline function is:

$$s_n(x) = \sum_{k=0}^{n} a_i (x - x_i)^k \quad , \quad x \epsilon [x_i, x_{i+1}], \quad i = \overline{0, n-1} \quad (3)$$

and the expression on each interval is:

$$s(x) = \begin{cases} s_0(x) & if \quad x_0 < x < x_1 \\ s_1(x) & if \quad x_1 < x < x_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots & \\ s_n(x) & if \quad x_{n-1} < x < x_n \end{cases} \quad (4)$$

The function s:[a,b] → R is called spline function of k order if there are satisfied the following properties:

(a) It's restrictions on subintervals I_i are polynomials of the same k order;

(b) Spline function of k order can be derived by (k-1) times in interval [a,b], which means the function is in $C^{(k-1)}$ on [a,b].

$$\begin{cases} s_{k,i}(x_{i+1}) = s_{k,i+1}(x_{i+1}) \\ s'_{k,i}(x_{i+1}) = s'_{k,i+1}(x_{i+1}) \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots \\ s_{k,i}^{(n)}(x_{i+1}) = s_{k,i+1}^{(n)}(x_{i+1}) \end{cases} \quad for\ any\ i = 0,1,\dots\dots,(n-1) \quad (5)$$

A cubic spline function is expressed by a third degree polynomial.

To express the interpolation spline function it is necessary to find 4n coefficients, as (a_i, b_i, c_i, d_i). First property ensures (n+1) conditions, other 3(n-1) conditions are ensured by the continuity property of first and second derivatives. So, in order to generate a unique spline, two other conditions must be imposed upon the system. Usually, those conditions are set on endpoints.

2.2. The spline interpolation method

Interpolation is a process of finding new data points within the range of a known data points. Referring to this process, the author stated in reference [7]: "The spline interpolation is based on the following principle. The interpolation interval is divided into small subintervals. Each of these subintervals is interpolated by using the third-degree polynomial. The polynomial coefficients are chosen to satisfy certain conditions (these conditions depend on the interpolation method)."

The spline interpolation has the main advantages: stability of the results and accuracy and simplicity of calculations.

The most widespread spline interpolation function is the spline function of third order called cubic spline interpolation. In Fig. 1 is given the graphical representation of cubic spline function:

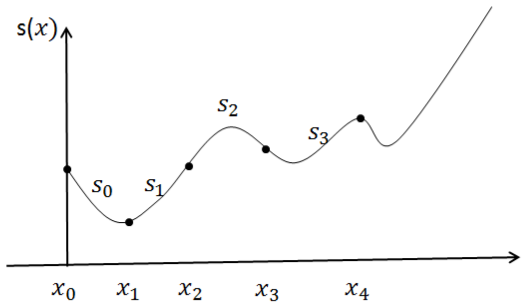

Fig. 1 Graphical representation of cubic spline function.

The Catmull-Rom Spline is part of cubic interpolating spline family, used initially in computer graphics, in curves and surfaces design. Recently, the area of Catmull-Rom Spline applications has been extended in artificial intelligence, image and signal processing. Our propose, in this paper, is to present the algorithm of Catmull-Rom Spline operator which can help to draw a contour.

3. THE ALGORITHM OF CATMULL-ROM SPLINE INTERPOLATION

3.1. The mathematical model of Catmull-Rom spline function

Catmull-Rom spline function is formulated by an essential condition: the tangent of each control point is calculated by using the previous and next control point.

Definition of Catmull-Rom spline is: given (n+1) control points {P0, P1, ..., Pn}, the aim is to find a curve that interpolates these control points with following conditions:

- the curve will pass through all of them;
- if one of the control points is moved, it only affects the curve locally;
- the curve will be defined on each segment by using two control points and specifying the tangent to the curve at each control point.

As mentioned in [8], the expression of Catmull-Rom Spline is obtained by using Ferguson method:

$$p = [1 \quad h \quad h^2 \quad h^3] \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\tau & 0 & \tau & 0 \\ 2\tau & \tau - 3 & 3 - 2\tau & -\tau \\ -\tau & 2 - \tau & \tau - 2 & \tau \end{bmatrix} \quad (6)$$

This expression can be demonstrated as follows:

If it is considered 4 control points p_{i-2}, p_{i-1}, p_i si p_{i+1} , as presented in Fig. 2:

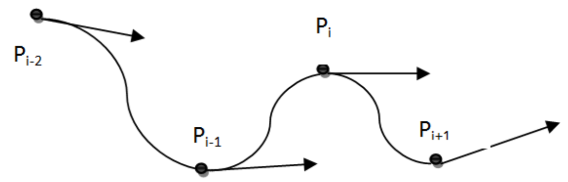

Fig. 2 A Catmull-Rom spline which pass through four control points.

The cubic spline function expressed by the polynomial form is:

$$p = \sum_{i=0}^{3} a_i h^i = a_0 + a_1 h + a_2 h^2 + a_3 h^3 \qquad (7)$$

Considering the following constraints, as in [10]:

$$\begin{aligned} p(0) &= p_{i-1} \\ p(1) &= p_i \\ p'(0) &= \tau(p_i - p_{i-2}) \\ p'(1) &= \tau(p_{i+1} - p_{i-1}) \end{aligned} \qquad (8)$$

where τ is a parameter, named „tension", set to different values 0,2, 0,5, 0,75 etc., it can be substituted those constraints in the polynomial equation (7) and it will obtain the system:

$$\begin{aligned} p_{i-1} &= a_0 \\ p_1 &= a_0 + a_1 + a_2 + a_3{}^{**} \\ \tau(p_i - p_{i-2}) &= a_1 \\ \tau(p_{i+1} - p_{i-1}) &= a_1 + 2a_2 + 3a_3 \ {}^{***} \end{aligned} \qquad (9)$$

If it will be replace a_0 and a_1 in second and fourth equations of system (3.1.4), it will obtain the system:

$$\begin{aligned} p_1 &= p_{i-1} + \tau(p_i - p_{i-2}) + a_2 + a_3 \\ \tau(p_{i+1} - p_{i-1}) &= \tau(p_i - p_{i-2}) + 2a_2 + 3a_3 \end{aligned} \qquad (10)$$

By solving this system of two equations with two unknowns, the solutions will be:

$$\begin{aligned} a_0 &= p_{i-1} \\ a_1 &= -\tau p_{i-2} + \tau p_i \\ a_2 &= 2\tau p_{i-2} + (\tau - 3)p_{i-1} + (3 - 2\tau)p_i \\ &\quad - \tau p_{i+1} \\ a_3 &= -\tau p_{i-2} + (2 - \tau)p_{i-1} + (\tau - 2)p_i \\ &\quad + \tau p_{i+1} \end{aligned} \qquad (11)$$

The matrix equation of this system is exactly the Catmull-Rom Spline interpolation function.

The simple result of matrix algebra makes this mathematical model to be easily implemented and applied in different areas.

The "tension" affects the curve in interpolation (control) points, defining the tangent amplitude to each control point and not the direction of the curve, so it can be chosen any reasonable value in order to obtain the desired curvature.

3.2 The Catmull-Rom spline operator algorithm.

The algorithm of Catmull-Rom Spline interpolation operator SI_{CR} is based on mathematical model described in the previous section and on the other operators developed in [14], [15] and [16]. To implement those operators, we used the virtual instrumentations VI, available in LabView library [11-13]. Graphical symbol is illustrated in Fig. 3.

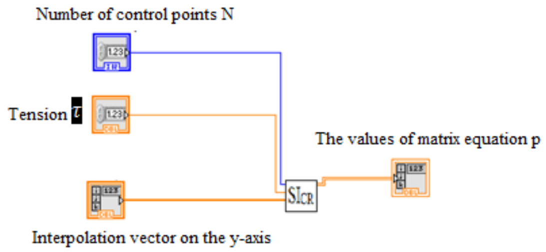

Fig. 3 Graphical symbol of SI_{CR}

The graphical structure of Catmull-Rom Spline interpolation operator SI_{CR} is developed in Fig. 4, and the inputs and output are presented in Table 1.

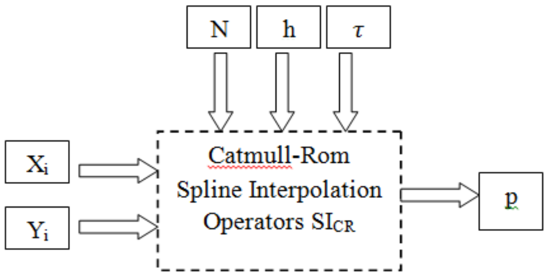

Fig. 4 The graphical structure of Catmull-Rom spline interpolation operator SI_{CR}

Table 1 – Inputs and output of operator SI_{CR}

INPUTS	
Interpolation vector on the x-axis	$X_i = \{x_i\}$
Interpolation vector on the y-axis	$Y_i = \{y_i\}$
Interpolation step h	
Number of control points N	
Tension τ	
OUTPUT	
The values of matrix equation p	$p_i = \{p_i\}$

4. APLICATIONS OF CATMULL-ROM SPLINE INTERPOLATION OPERATOR

Catmull-Rom Spline interpolation operator SI_{CR} was developed due to the need to achieve contours or isolines by interpolation, when the experimental data is in the form of a grid (table, matrix).

A major disadvantage of cubic spline interpolation operators, not presented here, is that it can only be used for values strictly increasing on x-axis. The use of cubic spline interpolation operators for plotting isolines or contours requires many sequences which makes the implementation difficult. In this situation is appropriate to use another type of operator. Thus, it was developed a Catmull-Rom Spline interpolation operator SI_{CR}, which can be easily used for plotting contours, isolines of temperature, pressure, humidity, keraunic levels, etc. which, subsequently, allows to build regional or global maps, with applications in many areas.

The goal of this study is to validate the operator by using it in an application that aims to plot an isoline (keraunic level) of experimental data given in a grid form (in our case we will consider the numbers of storm-days per year).

The aim is to plot the isolines which correspond to keraunic level 3, equivalent of an average number of 40 to 45 storm-days / year.

Even if keraunic map of Romania is based on measurements done between 1966 and 1980 and it is not updated considering the dynamic changes of climatology, this paper will not plot the keraunic levels map, but proposes a possible application of Catmull-Rom Spline interpolation operator.

To obtain the desired outcome of proposed application, it is necessary to follow several stages:

STAGE I: Selection from the grid of only the values which correspond to the desired isoline.

In general, the experimental data obtained in the form of a grid are measurements between equidistant points, with variable values, as shown in Table 2.

Table 2 - Measurements on a small area.

38	40	42	40	37	32	25	36
37	40	43	43	41	39	43	44
35	37	43	44	37	37	41	35
38	41	42	44	36	42	45	41
34	35	41	36	38	38	36	41
29	38	37	38	42	41	36	38
31	40	40	37	35	42	44	37
43	44	44	42	32	43	45	43
38	43	42	40	35	42	45	44

For our application, it will be selected only the values which are in the range 40-45 days of storm / year.

To further simplify the Table 2, it is proposed to assign value 1 for all measurements in the considered interval and to the others will be assigned the value 0.

Table 3 - Values 0 or 1 for the measurements

0	1	1	1	0	0	0	0
0	1	1	1	1	0	1	1
0	0	1	1	0	0	1	0
0	1	1	1	0	1	1	1
0	0	1	0	0	0	0	1
0	0	0	0	1	1	0	0
0	1	1	0	0	1	1	0
1	1	1	1	0	1	1	1
0	1	1	1	0	1	1	1

Once selected those values, the question is how to join all the points of value 1, in a closed loop, to obtain the isoline. This selection method will be described in stage II.

STAGE II: establishing the order of interpolation points to plot the isoline. In this stage is used a technique similar with a technique applied in image processing, described in [9], that extracts the boundaries of an image.

This algorithm is described in the following steps:

Step 1: The first step, and the most important, is choosing the start point and direction of movement. Once this was decided, it must take into account the points neighboring this point (it is also the central point), defined in all directions, according to the Table 4. It is considered the central point with the coordinates (x_n, y_n), that point which is moving to each new selection.

Table 4 - Neighboring points of the central point $P(x_n, y_n)$

$P(x_{n-1}, y_{n+1})$	$P(x_n, y_{n+1})$	$P(x_{n+1}, y_{n+1})$
$P(x_{n-1}, y_n)$	$P(x_n, y_n)$	$P(x_{n+1}, y_n)$
$P(x_{n-1}, y_{n-1})$	$P(x_n, y_{n-1})$	$P(x_{n+1}, y_{n-1})$

Step 2. Selecting the direction of movement to the right and the start point as $P(x_n, y_n)$, it must be compared with the three neighboring points of start point, which are:

$P(x_{n+1}, y_{n+1})$; $P(x_{n+1}, y_n)$; $P(x_{n+1}, y_{n-1})$.

Condition 1: if $P(x_{n+1}, y_{n+1})$ is equal with 1 then this point becomes the next central (start) point. The program will select this point and the algorithm will start again.

Condition 2: if $P(x_{n+1}, y_{n+1})$ is equal with 0, the algorithm will pass to the next neighboring point of selected point.

Condition 3: if $P(x_{n+1}, y_n)$ is equal with 1 then this point becomes the next central (start) point. The program will select this point and the algorithm will start again.

Condition 4: if $P(x_{n+1}, y_n)$ is equal with 0, the algorithm will pass to the next neighboring point of selected point, which is $P(x_{n+1}, y_{n-1})$

Condition 5: if $P(x_{n+1}, y_{n-1})$ is equal with 1 then this point becomes the next central (start) point. The program will select this point and the algorithm will start again.

Condition 6: if $P(x_{n+1}, y_{n-1})$ is equal with 0, then the searching must rotate 90 degrees in clockwise direction.

All the conditions, described above, are repeated in the new direction of search, thus search follows all the steps for other three neighboring points of central point, but selected on the horizontal direction:

$P(x_{n-1}, y_{n-1})$; $P(x_n, y_{n-1})$; $P(x_{n+1}, y_{n-1})$.

It is important to mention that in this algorithm one must keep the same direction in selection of neighboring points. Thus:

- on horizontal movement, from left to right, the neighborhood points are vertical and searching starts from top to bottom,

- on horizontal movement, from right to left, the searching will start from the bottom to the top,

- Similarly, on vertical movement, from top to bottom, the neighborhood points are horizontal and searching starts from right to left,
- For vertical movement, from the bottom to the top, the searching starts from left to right.

For our application, the start point is the first value equal with 1 from the first line of grid. The searching will be in clockwise direction, as can be seen in Fig 5.

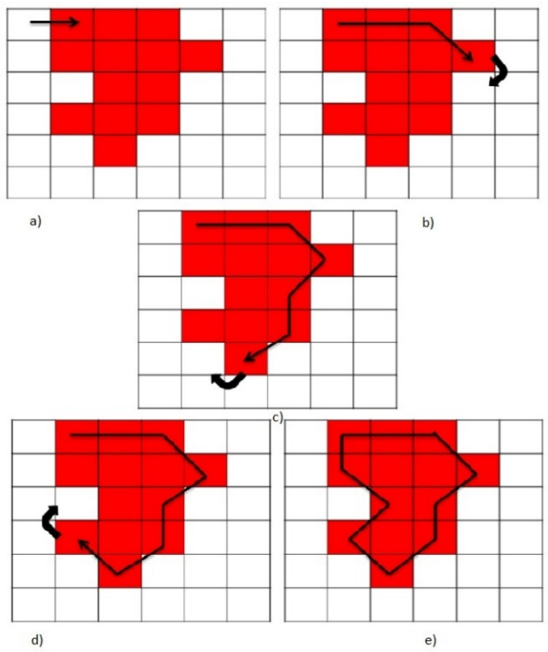

Fig. 5 Graphic representation of searching algorithm.
a) Start point is the first value equal with 1 from the first line of grid and the searching is in clockwise direction
b) When all the right neighboring points of selected points are 0, the searching must rotate 90 degrees in clockwise direction.
c) When all the bottom neighboring points of selected points are 0, the searching must rotate 90 degrees in clockwise direction.
d) When all the left neighboring points of selected points are 0, the searching must rotate 90 degree in clockwise direction.
e) The selection of the values is finished when the searching arrives at the start point.

This algorithm was built in Labview/Mathscript.

STAGE III: Appling the Catmull-Rom Spline interpolation operator.

The inputs are:

Interpolation vector on the x-axis $X_i = \{x_i\}$ – all the column index:

2	3	4	5	4	4	3	2	3	2	2

Interpolation vector on the y-axis $Y_i = \{y_i\}$ –all the row index:

1	1	1	2	3	4	5	4	3	2	1

Interpolation step h =10

Number of control points N=10

Tension $\tau = 1.7$

With this operator, one can easily find the output which is the values of matrix equation p $p_i = \{p_i\}$ as in Table 5.

Table 5 - Values of matrix equation p

1	0,86	0,78	0,75	0,75	0,78	0,83	0,89	0,94	0,98
1	0,98	0,94	0,89	0,83	0,78	0,75	0,75	0,78	0,86
1	1,13	1,21	1,25	1,27	1,28	1,32	1,39	1,51	1,71
2	2,27	2,43	2,50	2,51	2,50	2,48	2,49	2,56	2,72
3	3,27	3,43	3,50	3,51	3,50	3,48	3,49	3,56	3,72
4	4,30	4,53	4,71	4,84	4,92	4,97	4,99	5,01	5,00
5	5,00	5,00	4,99	4,97	4,92	4,84	4,71	4,53	4,30
4	3,72	3,56	3,49	3,48	3,50	3,51	3,50	3,43	3,27
3	2,72	2,56	2,49	2,48	2,50	2,51	2,50	2,43	2,27
2	1,71	1,51	1,39	1,32	1,28	1,27	1,25	1,21	1,13
1									

In Fig. 6 was plotted the isoline of keraunic level 3, based on data from Table 5.

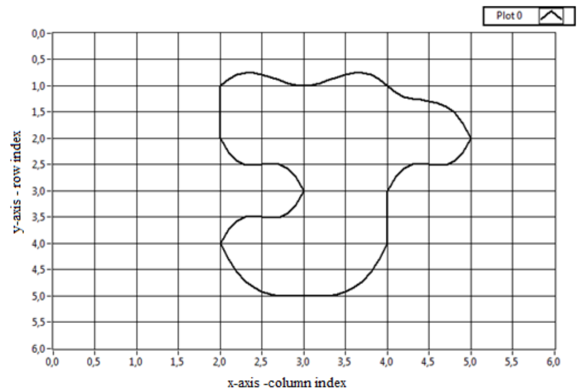

Fig. 6 Isoline of keraunic level 3.

In Fig. 7 are four such isolines obtained by running the program for more cases like the one shown above.

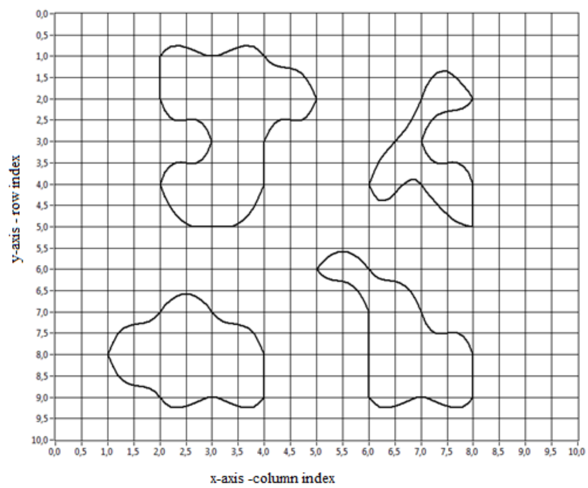

Fig. 7 Four isolines of keraunic level 3.

5. CONCLUSION

Cubic spline interpolation is a special case of spline interpolation that is used very often to avoid the problem of Runge’s phenomenon. This method gives an interpolating polynomial that is smoother and has smaller errors than some other interpolating polynomials such as Lagrange polynomial or Newton polynomial. So the cubic is the optimal degree for splines.

Analyzing the results of the entire algorithm, the conclusion is that Caltmull-Rom spline interpolation is a fast, efficient and stable method of function interpolation.

The above example can be extrapolated to other parameters that must be represented in isolines maps. This operator can be automated to generate as many isolines are necessary. Also, this algorithm is very efficient and the volume of calculus is minimized.

ACKNOWLEDGMENT

This paper was supported by the project "Improvement of the doctoral studies quality in engineering science for development of the knowledge based society - QDOC" contract no. POSDRU/107/1.5/S/78534, project co-funded by the European Social Fund through the Sectorial Operational Program Human Resources 2007-2013.

REFERENCES

1. Schumaker, Larry L: *Spline Function: Basic Theory*, Editia a treia, Cambridge University Press, 2007.
2. Cicone, T.: Metode numerice în ingineria economică, http://www.omtr.pub.ro/t_cicone/didactic/mn_files/MN-cap_2_Aproximare.pdf
3. Atkinson, K., Han, W.: *Teaching numerical analysis using elementary numerical analysis* http://www.cs.uiowa.edu/~atkinson/ena_master.html
4. Ciuprina, Gabriela: *Algoritmi numerici pentru calcule stiintifice în ingineria electrică*, Editura MatrixROM, 2013.
5. Roșca,I. : Calcul numeric. Elemente de teoria aproximarii, București 2011.
6. Micula, Gh.: *Funcții spline și aplicații*, Editura Tehnică, Bucureștii, 1978.
7. http://www.alglib.net/interpolation/spline3.php
8. Kenneth, I. Joy: Catmull-Rom Splines, Geometric Modeling On-Line, Notes, 1999.
9. Pavlidis,Th.: *Algorithms for Graphics and Image Processing*, Berlin: Springer and Rockville: Computer Science Press, 1982.
10. Twigg,Ch: *Catmull-Rom splines*, https://www.cs.cmu.edu/~462/projects/assn2/assn2/catmullRom.pdf
11. “LabVIEW–Graphical Programming for Instrumentation”, User Manual, National Instruments, Corporation, 1996
12. ”LabVIEW”, User Manual National Instruments, January 2000.
13. ”LabVIEW–Function and VI Reference Manual”, National Instruments, 2000.
14. Todoran, Gh., Holonec, R.: ”Spline operators with aplications in the field of circuit analysis”, in ”Revue roumain des sciences techniques”, seria Electrotehnique et energetique, Romanian Academy Publishing, tome 46, nr. 4, October-December 2001.
15. Todoran, Gh., Holonec, R.:” The modeling of the Analogue Low Pass Filters Using the Spline Operators”, Acta Electrotehnica, vol. 45, no. 2, 2004.
16. Todoran, Gh., Holonec, R.:” The modeling of the Analogue High Pass, Band-Pass and Band-Stop-Type Filters by Using Spline Operators”, Acta Electrotehnica, vol. 45, no. 2, 2004.

Oana Ancuta Buzura
Faculty of Electrical Engineering, Technical University of Cluj-Napoca, 26-28, G. Bariţiu St., Cluj-Napoca, Romania
Oana.BUZURA@ethm.utcluj.ro