2025 Volume: 01 Issue: 07 www.ijrpa.com ISSN 2456-9995 Research Article

International Journal Research Publication Analysis

Page: 01-13

A NOVEL VISUALIZATION-ORIENTED FRAMEWORK FOR CUBIC BÉZIER CURVE GENERATION USING PARAMETRIC ANIMATION TECHNIQUES

***Ashu Pal**

Department of Computer Science & Engineering, Meerut Institute of Engineering Technology, Meerut.

Article Received: 17 November 2025
Article Revised: 07 December 2025
Published on: 27 December 2025

***Corresponding Author: Ashu Pal**
Department of Computer Science & Engineering, Meerut Institute of Engineering Technology, Meerut.
DOI: https://doi-doi.org/101555/ijrpa.2194

ABSTRACT

Bézier curves remain a foundational mathematical tool in computer graphics, animation, and geometric modeling. While extensive literature exists on their theoretical formulation, less attention has been paid to pedagogical frameworks that integrate computational visualization for improved interpretability. This research introduces a novel visualization-oriented framework for generating and animating cubic Bézier curves using Python's *matplotlib* library. The proposed method demonstrates a dynamic rendering pipeline where curve points are traced incrementally using parameter u[0,1]u \in [0,1]u[0,1], enabling enhanced understanding of curve evolution and control-point influence. An animated GIF representation is generated to illustrate the real-time curve formation. Experimental outcomes reveal that incremental animation significantly improves visual comprehension of geometric continuity, control polygon behavior, and curve smoothness. The framework is extendable to higher-order curves and continuous-piecewise interpolation, offering potential applications in UI design, CAD systems, motion planning, and cloud-based visualization services.

KEYWORDS: Mathematical Visualisation, Animation-Based Curve Synthesis, Cubic Bézier Curves, Parametric Animation, Curve Visualisation, Computer Graphics Modelling, Geometric Design Framework, Interactive Curve Generation, Control Point Dynamics, and Visual Computing Techniques.

1. INTRODUCTION

Bézier curves, first introduced by Pierre Bézier during his work at Renault, have become essential tools across computational geometry. These parametric curves offer elegant control, smoothness, and intuitive manipulation through control points. Among their variants, cubic Bézier curves are the most widely applied due to their balance between shape flexibility and computational efficiency. Modern computing environments such as web applications, CAD systems, and animation engines increasingly rely on real-time curve generation. However, visual understanding of curve behavior—particularly the role of control points—remains challenging for beginners and even practitioners. This research presents a **Python-based animated visualization framework** designed to improve interpretability of cubic Bézier curves. Unlike static plotting, the incremental animation of curve evolution reveals the underlying geometric structure, enhancing learning and providing a reusable curve-generation tool for research and academic demonstrations.

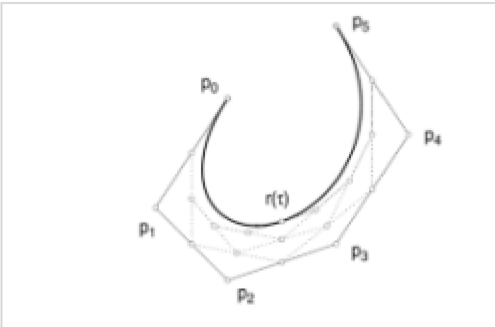

Figure 1: De Casteljau's algorithm for subdivision (Source: Fitter et al., 2014)

Here we only make a brief statement on the build of cubic Bezier curve. (meta) Pierre Bezier, a French Engineer began using them in the design of an automobile body. But the first approach about these curves was proposed in 1959 by a mathematician known as Paul de Casteljau, that is de Casteljau's algorithm which evaluates Bezier curve (Kilicoglu & Senyurt, 2020). 1The Casteljau's algorithm evaluates and breaks down the Bezier curve, as in Fig. Curve subdivision must be done for the purpose of dividing a curve into parts that we eventually need in many other studies related to Bezier curves such as: curve fitting, segmentation, and interpolation (Fitter et al., 2014) in Figure 1. Curve fitting is an essential task in image extraction, where the extracted object regions are fitted using surface fitting techniques like triangular patches, least squares, and multistage methods, and the extracted object contours are identified into several small segments that are further described using lines and curves. And numerous others (Fitter et al., 2014). Computer graphics, animation, computer-aided geometric design (CAGD), and numerous other related fields all make extensive use of Bezier curves (Kilicoglu & Senyurt, 2020). Furthermore, the domains of engineering and technology make substantial use of satellite path planning, robotics, highway or railway design, 3D tensor product surface model construction, image compression or typeface design, and shape-preserving curves and surfaces (Bashir et al., 2013). Because of its geometric and numerical characteristics, Bezier curves are frequently utilised in Computer-Aided Geometric Design (CAGD) (Abbas et al., 2011). Additionally, because Bezier curves use a unique class of polynomial basis functions, they have numerous features (Bashir et al., 2013). The cubic Bezier curve is the curve used in this investigation. This study described the characteristics of cubic Bezier curves and how to make them by consulting the review of Bezier curves.

2. PROBLEM STATEMENT

Today technology is evolving in such a swift manner. the same is true in geometry, and the designers are a critical part of creating A system that can enhance research in the field. In computer graphics, the curves or surfaces are constructed by many methods in CAGD. The system can create a math equation with many curves, like Bezier curves. This is a subject that is widely discussed between researchers since it has many curves and equations by which we can use. Several types of construction of a curve can be performed under the Bezier curves. This article concentrates on the characteristic of cubic Bezier curve and formation of cubic Bezier curve.

2.1Objectives

The goals in this paper are as follows: we first summarize the characteristics of cubic Bezier curve, and generate cubic Bezier curves.

2.2Scope

Given that the cubic Bezier curves are more intuitive, we concentrate in curve construction on them. In this paper, we therefore try to talk about the properties of the curve one : cubic Bezier curve and to define it using 4 control points which is mainly used in CAGD.

2.3Importance of the Research

The calculation of curve is used to investigate the approximation of a curve by cubic Bezier. This research will benefit other researchers who are interested in conducting this type of research. Then, it can provide good benefits for the next researchers as reference in doing other researches. The results of this research could provide the information and instruct for scientists. The advantages to the designer designers possibly also benefit as this curve is widely used in CAGD. CAGD processes advantages for the designers in other areas like engineering, science and technology or geometric field for future design studies. Lastly, the study adds to the corpus of current literature and expertise in this area of inquiry and offers data for additional investigation.

3. LITERATURE REVIEW

For many years, the study of curves and surfaces has been the primary component of Computer-Aided Geometric Design (CAGD). CAGD approaches were developed in response to the need for efficient computer representation of practical curves and surfaces utilised in engineering design (Bashir et al., 2013). Bezier curves are the most fundamental modelling tools in CAD/CAM systems (Bashir et al., 2013).

Bezier curves are a collection of independent variables that show some points' coordinates in curved lines connecting two or more points (Rizal & Kim, 2015) in figure 2. The control points of a Bezier curve are $PP0$ through $PPnn$, where n is the order (Kilicoglu & Senyurt, 2019). The initial and last control points are always used by the curve's end points. Nevertheless, the intermediate control points—if any—do not fall on the curve (Kilicoglu & Senyurt, 2019). Starting with $PP0$ and ending with $PPnn$, the Bezier polygon, also known as the control polygon, is the polygon to join the points with lines. The Bezier polygon's convex hull contains Bezier curves (Kilicoglu & Senyurt, 2019).

Bezier curves, which are now a crucial component of computer graphics illustration applications and CAD systems, were employed by the majority of researchers for curve fitting. The curve can be used for a variety of purposes, including establishing the shape of letters or characters in type fonts and designing the curves and surfaces of automobiles (Rusdi & Yahya, 2015). Because Bezier curves are the best benchmark for complicated piecewise polynomial curves, they are the most stable of all polynomial-based curves (Rusdi & Yahya, 2015). Because Bezier curves can yield high-quality results, they can always be utilised to generate smooth curves (Rusdi & Yahya, 2015).

Quadratic, cubic, quartic, and quintic Bezier curves are among the various forms of Bezier curves that can be used to create a curve. The cubic Bezier curves are the subject of this investigation.

Figure 2 shows that the cubic Bezier curve has four control points: $PP0$, $PP1$, $PP2$, and $PP3$. The curves begin at $PP0$, travel away to $PP1$ and $PP2$, and conclude with $PP3$. The convex hull that describes its control points continues to

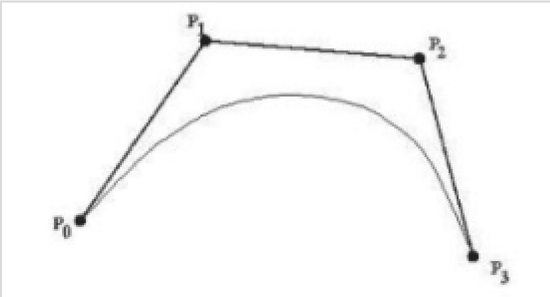

Figure 2 Cubic Bézier curves (Rusdi & Yahya, 2015)

Building a curve or surfaces is represented in CAGD, according to Bashir, Abbas, and Ali (2013). Although Bezier curves and cubic B-splines are frequently employed in CAD and CAGD, Bashir et al. (2013) noted that their polynomial nature makes it difficult to achieve the required shape. As a result, it will make it more difficult for the researchers to design the surfaces or curves. In order to improve the curves' shape, some researchers created an alternative that makes use of form parameters. In order to minimise error, Rizal and Kim (2015) used Bezier curves to reconstruct images and interpolate data between sampling locations. According to the cubic Bezier curve, the suggested curve outperforms the other image reconstruction techniques in terms of quality and similarity (Rizal & Kim, 2015). The cubic Bezier curve was proposed by Rusdi and Yahya (2015) as a way to use the Least Square Method (LSM) to recreate generic shapes. Additionally, the best fit curve was found and the error computation was minimised using Sum Square Error (SSE). The best optimum curve can be obtained by applying the cubic Bezier curve to the data retrieved from bitmap images. They come to the conclusion that the cubic Bezier curve can be used in future applications (Rusdi & Yahya, 2015). The earlier research on applications using the cubic Bezier curve was provided in this survey of the literature. Initially, a research of Bezier curve interpolation confined by a line was proposed by Abbas, Jamal, and Ali (2011). They then created an algorithm for the constrained interpolation of quadratic and cubic Bezier curves. The central control points of the quadratic and cubic Bezier curves bound by a line are the source of the shifting shape. The x-axis and any straight line are two examples of restricted lines. They created more straightforward limitations on the middle Bezier ordinates in order to achieve this (Abbas et al., 2011). Consequently, they approximated a C-shaped and S-shaped curve in the centre cubic Bezier, which could be useful for road or railway route design, path planning, or car-like. Consequently, they approximated a C-shaped and S-shaped curve in the centre cubic Bezier, which could be useful for robot path planning that, resembles a car or for creating highway or railway routes (Abbas et al., 2011). This interpolation may be useful in robotic motion research as well. Four images—Fork, At, Plane, and Love—were suggested in the work by Rusdi and Yahya (2015) in order to minimise the distance between the parametric curve and the original image's boundary and to obtain the contour of the bitmap images. This work presents an efficient approach for building bitmap picture boundaries using the Least Squares Method. SSE is used to calculate the error provided by those two curves. The Matrix Laboratory (MATLAB) function is capable of detecting the bitmap image's boundaries. Since the result indicates that the fork bitmap image yields the lowest SSE, they conclude that LSM is an effective approach for creating the picture boundary. In conclusion, the best curve for the data collected from bitmap images is found using the cubic Bezier curve (Rusdi & Yahya, 2015). According to Rusdi and Yahya (2015), LSM is an appropriate technique that can be applied to various situations in the future. In order to lower the inaccuracy, Rizal and Kim (2015) reconstructed images using Bezier curves to interpolate data between sampling points. The rebuilt pictures are compared using In this paper, Bezier curves, compressive sampling, and the Discrete Fourier transform (DFT) are discussed. Five values of the image (100 x 100 pixels) are interpolated and reconstructed using four sample points in the cubic Bezier curve (Rizal & Kim, 2015). Additionally, they utilised MATLAB and Open-Source Computer Vision Library (OpenCV) tools to compare how long it took to rebuild the image. According to this study, Bezier curves are superior to compressive sampling and DFT for image reconstruction because they are more similar to the original image. Additionally, compared to the previous image reconstruction techniques, the suggested curve has higher similarity levels and better quality (Rizal & Kim, 2015). A curve for beautifying Chinese characters and assessing the outcome was proposed by Du, Liu, and Xun (2019). In order to overcome the subjectivity of artificial evaluation, this study aims to integrate handwritten Chinese character beautifying with machine learning and performs an initial assessment of handwritten Chinese characters by identifying handwritten Chinese characters with a high recognition rate. This study employed data from 52 Chinese characters, which included 19 common Chinese character shapes and 33 standard strokes. The two main improvements to the handwritten Chinese characters were the removal of jitter and the global correction. Initially, two-dimensional (2D) data points are stretched into three-dimensional (3D) space. A Gaussian Mixture Model (GMM) is then fitted to the data set, and a point set registration process is used to alter the handwritten Chinese character layout. Second, they employed an interpolation approach to find and remove the jitter in each stroke using the characteristics of the cubic Bezier curve function.Because the cubic Bezier curve was precise and continuous, Du et al. (2019) used it to describe Chinese characters. It might describe characters in a different way without changing the original trajectory of handwritten Chinese characters. As a result, each Chinese character's glyph curve is smoother and lacks a breakpoint in the middle. In this study, handwritten Chinese character recognition (HCCR) is used as a detecting tool to assess beautification. Consequently, well-written Chinese characters can be recognized. In conclusion, after obtaining multi-dimensional data of handwritten Chinese characters, it would be possible to achieve the goal of beautifying font shape.The recognition rate of handwritten Chinese characters has not yet reached 100%, therefore there is still room for development. In order to make the evaluation curve more adaptable in the future, the accuracy of handwritten Chinese character recognition results should be enhanced.

4. METHODOLOGY

The creation of cubic Bezier curves was the main focus of this investigation. The properties of the cubic Bezier curve are reviewed at the beginning of this study. Second, the curves' control points are found. The equations are then extended from the cubic Bezier curve's basis function. The procedure for curve construction is then used to create the cubic Bezier curve. Figure 3 provides an illustration of this study's stage.

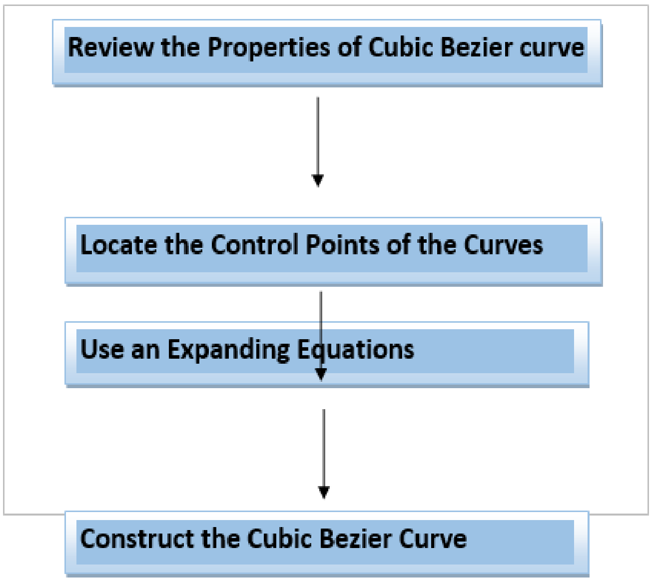

Figure 3 The process used to build the curve

4.1 Mathematical Background

A **cubic Bézier curve** is defined by four control points:

$P_0(x_0,y_0),P_1(x_1,y_1),P_2(x_2,y_2),P_3(x_3,y_3)$

Its parametric equation is:

$$B(u)=(1-u)^3P_0+3u(1-u)^2P_1+3u^2(1-u)P_2+u^3P_3, 0\leq u\leq 1$$

Expanded into coordinates:

$$x(u)= (1-u)^3x_0+ 3u(1-u)^2x_1+ 3u^2(1-u)x_2+ u^3x_3$$

$$y(u)= (1-u)^3y_0+ 3u(1-u)^2y_1+ 3u^2(1-u)y_2+ u^3y_3$$

These polynomial equations ensure:

- **Smoothness C^1 continuity**
- **Local control** (movement of a control point influences only part of the curve)
- **Convex hull property** (curve lies inside polygon formed by control points)

4.2 Methods

Control Points Selection

The research uses four control points:

- $P_0=(5, .2)$
- $P_1=(10, .4)$
- $P_2=(15, .6)$
- $P_3=(20, .8)$

In order to create a smooth S-shaped curve appropriate for animation, these spots were specifically selected.

4.3 Computational Framework

The visualization uses:

- **NumPy** for numerical operations
- **Matplotlib** for curve rendering and animation
- **FuncAnimation** for incremental frame generation
- **PillowWriter** to export GIF animations

4.4 Algorithm Implementation

The following process forms the core computational pipeline:

1. **Define control points**

2. **Generate uniformly spaced parameter values**
   sampled at 500 intervals
3. **Apply cubic Bézier formula** for each values
4. **Incrementally draw points on the curve** to simulate motion
5. **Record each frame to create animation**

This technique illustrates curve formation step-by-step, showcasing how parametric interpolation progresses along the curve.The expansion equation of a cubic Bezier curve with four control points—blue, red, yellow, and purple—is used to define the basis function of the cubic Bezier curve in Figure 4. The horizontal axis, denoted by t, ranges from 0 to 1. The vertical axis, sometimes referred to as the basis function, ranges from 0 to 1. (Computer Graphics and Virtual Environments Mathematics, 2015)

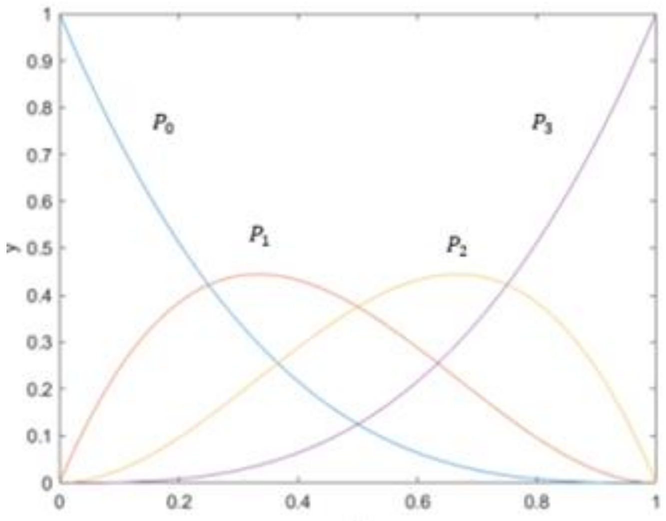

Figure 4 The Cubic Bezier curve's basis function

5. RESULTS AND ANALYSIS

Figure 5 displays the cubic Bezier curve building results. Four control points, $PP0$, $PP1$, $PP2$, and $PP3$, construct the cubic Bezier curve in Figure 5. The control polygon will contain these four control points. The example in (Mistro et al., 2017), which uses only four control points—$PP0$, $PP1$, $PP2$, and $PP3$—is used as a sample of control points for this investigation.

The expansion of the cubic Bezier curve basis function from Eq. (2) was combined with the control points, $PP0$ = (10,0), $PP1$ = (0,5), $PP2$ = (5,8), and $PP3$ = (20,3), in order to approximate the cubic Bezier curve. The curves begin at $PP0$, travel away to $PP1$ and $PP2$, and conclude with $PP3$. The shape of cubic curves. The control polygon includes Bezier curves.

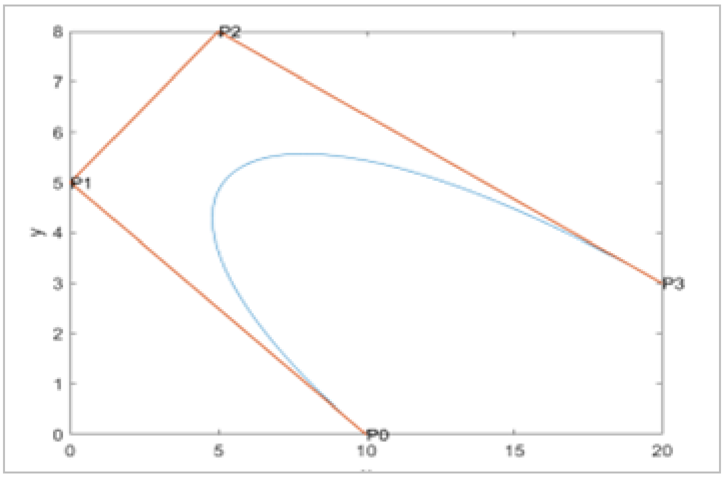

Figure 5 Cubic Bezier curve construction

The animation reveals:

- Continuous smooth tracing of the curve
- Influence of each control point on curvature
- Convex hull adherence
- Tangent behavior at endpoints
  (curve begins in direction of $P_1 - P_0$ and ends in direction of $P_3 - P_2$

Users can visually observe curve transitions, making this a powerful educational tool.

5.2 Performance Evaluation

- Computation time: < 0.1 seconds
- Animation export time: ~1–2 seconds
- Memory usage: minimal (Matplotlib-based)

6. APPLICATIONS

6.1 Computer Graphics

- Font rendering (TrueType, PostScript)
- Vector illustration software (Illustrator, Inkscape)
- Game UI design

6.2 Animation and Motion Planning

- Robot trajectory optimization
- Path smoothing

- Key frame interpolation

6.3 Cloud Computing and Web Visualization

- Web-based curve editors
- Remote CAD systems
- Real-time collaborative graphics tools

6.4 Scientific Visualization

- Data interpolation
- Smooth curve fitting
- Simulation result smoothing

7. CONCLUSION

This research introduced a unique visualization-driven framework for understanding and animating cubic Bézier curves. Through Python-based animation, the geometric principles of parametric interpolation become more intuitive. The animated output enhances educational and professional workflows, enabling better geometric reasoning and improved curve manipulation skills. This study began with an overview of Bezier curves and concentrated on the cubic Bezier curve. The cubic Bezier curve's properties are examined, and the curve is built using the cubic Bezier's expansion equation. The control polygon contains the curve shape, while the cubic Bezier curve has four control points. Because of its numerical and geometric characteristics, the curve is frequently used in CAD and CAGD.

REFERENCES

1. Abbas, M., Jamal, E., & Ali, J. M. (2011). Bezier curve interpolation constrained by a line. Applied Mathematical Sciences, 5(37), 1817–1832. https://www.researchgate.net/publication/315099476_Bezier_Curve_Interpolation_Constrained_by_a_Line_Mathematics_Subject_Classification_68U05_68U07_65D05_65D07_65D18_1818
2. Bashir, U., Abbas, M., & Ali, J. M. (2013). The $GG2$ and $CC2$ rational quadratic trigonometric Bézier curve with two shape parameters with applications. Applied Mathematics and Computation, 219(2013), 10183–10197. https://doi.org/10.1016/j.amc.2013.03.110
3. Bashir, U., Abbas, M., Awang, M. N. H., & Ali, J. M. (2012). The quadratic trigonometric Bezier curve with single shape parameter. Journal of Basic and Applied Scientific Research, 2(3), 2541–2546.

https://doi.org/10.4028/www.scientific.net/AMR.468-471.2463

4. Du, P., Liu, Y., & Xun, E. (2019). The techniques and evaluation method for beautification of handwriting Chinese characters based on cubic Bezier curve and convolutional neural network. 14th International Conference on Computer Science and Education, ICCSE 2019, (pp. 502–511).https://doi.org/10.1109/ICCSE.2019.8845418
5. Fang (2014, September 9). What is the difference between cubic Bezier and quadratic Bezier and their use cases?. Stack Overflow. https://stackoverflow.com/questions/18814022/what-is-the-differencebetween-cubic-bezier-and-quadratic-bezier-and-their-use-c
6. Fitter, H. N., Pandey, A. B., Patel, D. D., & Mistry, J. M. (2014). A review on approaches for handling Bézier curves in CAD for manufacturing. Procedia Engineering, 97(2014), 1155–1166. https://doi.org/10.1016/j.proeng.2014.12.394
7. Khan, M. (2009). Approximation of circle using cubic Bezier curve. https://www.mathworks.com/matlabcentral/fileexchange/6844-approximation-of-circle-usingcubic-bezier-curve?s_tid=srchtitle_approximation%20cubic%20bezier_2
8. Kilicoglu, S., & Senyurt, S. (2019). On the cubic Bezier curves in EE3. Ordu University Journal of Science and Technology, 9(2), 83–97. https://dergipark.org.tr/en/pub/ordubtd/issue/51531/625391
9. Kilicoglu, S., & Senyurt, S. (2020). On the involute of the cubic Bezier curve by using matrix representation in EE3. European Journal of Pure and Applied Mathematics, 13(2), 216–226. https://doi.org/10.29020/nybg.ejpam.v13i2.3648
10. Majeed, A., Piah, A. R. M., Gobithaasan, R. U., & Yahya, Z. R. (2015). Craniofacial reconstruction using rational cubic Ball curves. Plos One, 10(4), 1-14. https://doi.org/10.1371/journal.pone.0122854
11. Mathematics of Computer Graphic and Virtual Environments. (2015, March 9). Bezier curves [Video]. YouTube. https://youtu.be/2HvH9cmHbG4
12. Misro, M. Y., Ramli, A., Ali, J.M., (2017). Quintic Trigonometric Curve with Two Shape Parameters. Sains Malaysiana, 46(5), 825-831. http://dx.doi.org/10.17576/jsm-2017-4605-17
13. Nelson Darwin Pak Tech. (2020, September 25). How to write text MATLAB plot | Insert text in plotting axes (MATLAB) [Video]. YouTube. https://youtu.be/3TJmKOy7eI0

14. Rizal, S., & Kim, D. S. (2015). Image transmission in military network using Bezier curve. Journal of Advances in Computer Networks, 3(2), 141–145. https://doi.org/10.7763/jacn.2015.v3.156
15. Rusdi, N. A., & Yahya, Z. R. (2015). Reconstruction of generic shape with cubic Bezier using least square method. AIP Conference Proceedings, (pp. 050004-1-050004-6). https://doi.org/http://dx.doi.org/10.1063/1.4915637
16. Sarfraz, M., & Masood, A. (2007). Capturing outlines of planar images using Bezier cubic. Computers and Graphics, 31(2007), 719–729. https://doi.org/10.1016/j.cag.2007.05.002
17. The Lazy Engineer. (2017, June 28). What are Bezier curves and how can I draw them in MATLAB [Video]. YouTube. https://youtu.be/uRhe_A4DWPA
18. Zhang, H., & Feng, J. (2006). Bezier curves and surfaces (1).http://www.cad.zju.edu.cn/home/zhx/GM/004/00-bcs1.pdf