International Journal of Engineering and Science Invention (IJESI)
ISSN (Online): 2319-6734, ISSN (Print): 2319-6726
www.ijesi.org ||Volume 14 Issue 5 May 2025 || PP 15-20

A Short Survey on B-spline and Bezier Methods

Shubhashree Bebarta[1*], Bibhakar Kodamasingh[2], Ranjan Kumar Jati[3]
[1]*Department of Basic Science and Humanities, Raajdhani Engineering College, Bhubaneswar, India*
[2]*Department of Mathematics, ITER, Shiksha O Anusandhan University, Bhubaneswar, India*
[3]*Department of Mathematics, DRIEMS University, Tangi, Cuttack, India*

Abstract: Two effective techniques for approximating data in all areas of engineering problems are the B-spline and Bezier methods. They can be used for data interpolation with a few tweaks and enhancements. Although there are various methods for formulating B-splines in the literature, the B-spline equation is defined by a set of independent functions. The number of pairs of data or control points is equal to the number of coefficients in the B-spline equation. The B-spline method's benefit is that most set functions decrease at specific control points. The governing parametric equations can be drawn to produce Bezier curves. Bernstein polynomials are the parametric equations. A Bezier curve passes through the end points but not all of the data points. It is typically used in approximation methods. The Bezier curve can be used for interpolation since it can be made to pass through every control point by taking into account a few complimentary points between the original locations. This procedure produces curves that are smoother and less sinuous. These two approaches handle a number of challenges for comparing the interpolation qualities. The findings demonstrate the strength and resilience of both approaches for interpolating a highly irregular data set. In terms of accuracy, smoothness, and less computations, the Bezier model outperforms B-splines.

Keywords: Bezier curve, B-spline curve, Interpolation, Approximation.

Date of Submission: 15-05-2025 Date of Acceptance: 26-05-2025

I. Introduction

Geometric modeling is crucial to fill the gap between computer science research and industry. It is important for both design and function in many fields, like aircraft, digital devices, cars, scientific and medical instruments, and home goods. This paper concentrates on Computer Aided Geometric Design (CAGD), a subset of Geometry, Numerical Analysis, Computer Graphics, and Approximation Theory. Curves and surfaces are important parts of CAGD. In computational graphics, representations of large data sets are also included in CAGD. B-splines, trigonometric splines, and Bezier methods are all techniques used to create smooth curves and surfaces in computer graphics, geometric modeling, and other fields. B-splines are piecewise polynomials that provide local control and smoothness, while trigonometric splines offer advantages for representing periodic shapes and curves. Bezier curves are a specific type of parametric curve defined by control points and Bernstein polynomials, and they can be used to create smooth curves and surfaces. The B-spline curves, particularly trigonometric B-spline curves, have attained remarkable significance in the field of Computer Aided Geometric Design (CAGD). Different researchers have developed different interpolants for shape designing using Ball, Bezier, and ordinary B-splines. Preserving geometrical shapes of a given data set is essential for geometric design. It has great significance in different areas such as curve design, approximation, interpolation and reverse engineering. The definition of interpolating curves or surfaces that may be efficiently employed in the manufacturing business is a significant difficulty in the CAD/CAM industry. For many other applications such as the trajectories of cars, the design of aeroplanes, ships, and machine components, particles etc. One might desire to design an interpolating curve that would represent the information accurately. In this way, shape preserving interpolation techniques are important. Interpolation does not preserve the shape inherently. But under certain constraints the shape is preserved.

Piecewise polynomial functions have several important features. Specifically, due to their simplicity in representation, they provide a convenient way to define geometrical shapes. Further, the mathematical analysis of a geometrical shape is well understood by piecewise polynomial functions. Below a brief introduction to a special class of piecewise polynomial functions called classes of spline functions is given.

II. Splines

In mathematics, functions serve as important tools to describe and analyze physical processes and phenomena in nature. However, the explicit representation of the functions modeling these phenomena is known only for a few cases. Most of the time, we need to deal with approximations based on known information. Therefore, one of the objectives of applied mathematics is to find such approximations. For example, fitting data problems arise in all areas of the natural sciences. Approximation theory studies various classes of approximating functions, while numerical analysis chooses and analyzes the actual algorithms of approximation. The use of classes of spline functions in recent years has proven to be very beneficial in both approximation theory and numerical analysis. Originally, the word spline refers to a flexible ruler that is used to draw curves, mainly in the aircraft and shipbuilding industries. Classes of spline functions are easy to assess. They possess structural properties as well as excellent approximation powers.

Geometric modeling developed its roots in the era of Euclid and Descartes [1]. The manufacturing of curves dates to Roman times. It started with the purpose of shipbuilding, such as the ship's rib. Later, in between the 13th to 16th centuries, the Venetians modernized this shipbuilding technique. The ship hull was constructed by changing the ribs along with the keel. Earlier, there was no definite sketch of a ship hull. The splines are wooden beams used to make smooth curves, known as classical splines. According to [1], the earliest work on splines is the nineteenth-century work of Lo- vachevsky, where the splines are constructed using convolution. In 1944, Roy A. Liming proposed drafting methods and analysis with computational techniques in his book entitled Practical Analytic Geometry with aircraft applications. These methods helped in the designing and manufacturing of aircraft in World War II. Soon after, in 1946, Schoenberg [2] published a paper that marked as the beginning of the modern theory for spline approximations. Around 1960, De Castlejau and Bezier developed the subdivision algorithms for the Bernstein form of curves and surfaces. Particularly, the Bezier curve is a special parametric curve, based on a control polygon, and uses the Bernstein polynomial as a blending function. Since then, Bezier curves have formed the basis of CAGD. In 1972, De Boor, Cox and Mansfield discovered the recurrence relations for B-splines [3,4]. Lane and Riesenfield provided an important algorithm for B-splines in 1980 [5]. The enormous literature published in the last few decades shows the significant application of the spline theory in large areas of modern numerical mathematics, such as CAGD, Numerical integration and differentiation, Data fitting, Interpolation and Approximation, Wavelets and Fractals.

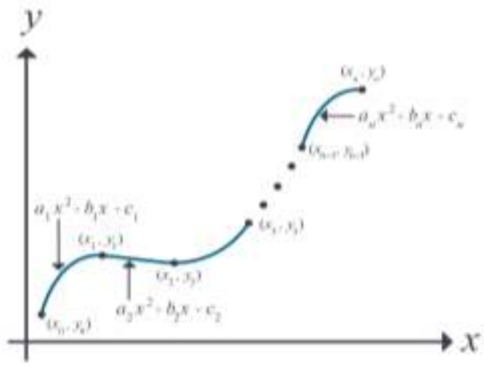

Fig 1: Spline Method Interpolation

III. Interpolation And Approximation

In computer graphics, we are concerned primarily with how the curve appears to a human observer and not with its mathematical properties. Our goal 4 is not to replace one curve by another, worrying about error estimates and other matters, as is done in mathematics. Rather, we want to create a curve that has a smooth look and satisfies our visual sense. To this end, the interpolation and approximation methods developed mainly in numerical analysis have proven to be a good tool.

In interpolation, the curve goes precisely through the points, as shown in Figure 1. In approximation the curve does not necessarily go through the points; it goes near to, or approximates them, as in Figure 2.

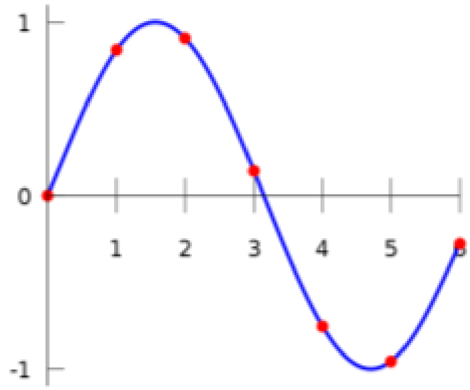

Fig 2: Interpolating Curve

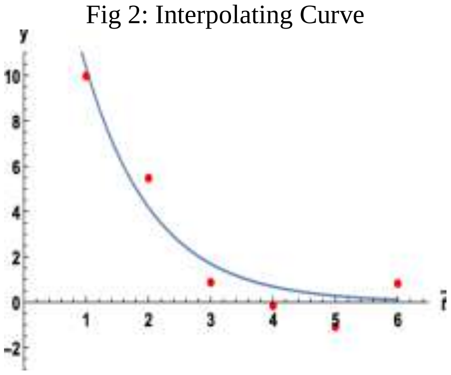

Fig 3: Approximating Curve

IV. Properties For Designing Curves

The requirements of the user and the computer combine to suggest a number of properties that our representations must have. We list below some of the important properties for designing curves. The requirements for designing surfaces are very similar as we know that surfaces can be represented using simple extensions of techniques for representing curves.

1- Spatial uniqueness

Two objects cannot occupy the same space at the same time. Objects shapes are expressed in a computer by data structures using numerical shape data and functions, but spatial uniqueness is very difficult to express. Therefore, a program is needed to test for overlap between positions of two bodies.

2- Control points

A common way to control the shape of a curve interactively is to locate points through which the curve must pass or points that control the curve's shape in a predictable way. These points are called control points. A curve is said to interpolate the control points if it passes through them.

3- Multiple values

In general, a curve is not a graph of a single-valued function of a coordinate, irrespective of the choice of coordinate system. So a curve can be multivalued with respect to all coordinate systems.

4- Axis independence

The shape of an object must not change when the control points are measured in a different coordinate system. If for example, the control points are rotated 90 degrees, the curve should rotate 90 degrees but not change shape. Some mathematical formulations will cause a curve’s shape to change if the reference coordinate system is changed.

5- Global or local control

As a designer manipulates a control point, a curve may change shape only in the region near the control point, or it may change shape throughout. This last behavior, called global control, may be annoying to the designer trying to make fine adjustments to just one portion of the curve.

6- Variation-diminishing property

Some mathematical representations have an annoying tendency to amplify, rather than smooth, any small irregularities in the shape outlined by control points. Others, possessing a variation diminishing property, always smooth the designer's control points.

7- Versatility

A curve representation that allows only a limited variety of shapes may frustrate a designer. A framework that provides only arcs of circles, for example, lacks sufficient versatility to model most designs. More flexible techniques allow the designer to control the versatility of a curve representation, often by adding or removing control points. For example, a curve specified by two control points might be a straight line connecting the points, introducing a third control point allows the curve to take on a large number of additional shapes, depending on the location of the control point.

V. B-Spline

The term spline comes from engineering drawing, where a spline is a piece of flexible wood to draw smooth curves. A spline is a parametric curve defined by control points. The control points are adjusted by the user to control the shape of the curve. Two features of the Bezier or Bernstein basis function limit the flexibility of the resulting curves. Firstly, the number of control points defines the order of the resultant curve. Secondly, any point on the curve is a result of blending the values of all defining vertices, and hence any change of one vertex is felt throughout the entire curve. B-spline curves overcome these two important limitations of Bezier curves. Here (i) the order or degree of the curve is independent of the number of control points, and (ii) the control points have only local control over the shape of the spline or surface.

The letter B in B-splines represents base or fundamental. B-splines are piecewise functions that exhibit global smoothness. Knots are real numbers that represent the components' meeting places. Define a partition or a non-decreasing series of knots as follows:

$$t := \left\{t_j\right\}_{j=1}^{m} = \left\{t_1 \le t_2 \le \dots \le t_m\right\}, m \in \mathrm{N} \quad (1)$$

The B-spline of order 1 for t is the characteristic function of this partition as follows

$$B_{i,1}(x) := \begin{cases} 1, & if \ t_i \le x < t_{i+1} \\ 0, & otherwise \end{cases} \quad i \in \mathrm{Z}. \quad \dots(2)$$

The B-spline basic function is constructed by the summation or linear combination of a series of B-spline basic functions that are linearly independent in the interval [a,b], as

$$S(x) = \sum_{i=0}^{n} c_i B_i\left[u_i(x)\right] = c_0 B_0(u_0) + c_1 B_1(u_1) + \dots + c_n B_n(u_n) \quad \dots(3)$$

Where $u_i(x) = \dfrac{x - x_i}{h} \quad h = \dfrac{b-a}{n}$

B-spline basis is used to describe a B-spline curve, several additional properties are immediately known:

1- Each basis function is positive or zero for all parameter values, i.e., $N_{i,k}$>0.
2- Except for k=l each basis function has precisely one maximum value.
3- The maximum order of the curve is equal to the number of defining polygon vertices.
4- The maximum order of the curve is equal to the number of defining polygon vertices.
5- The curve exhibits the variation diminishing property. Thus the curve does not oscillate about any straight line more than its defining polygon.
6- The curve generally follows the shape of the defining polygon. 6- Any affine transformation can be applied to the curve by applying it to the defining polygon vertices (i.e., the curve is transformed by transforming the defining polygon vertices). 7- The curve lies within the convex hull of its defining polygon.

Examples: We demonstrate different B-spline functions using MATHEMATICA.

Example-1

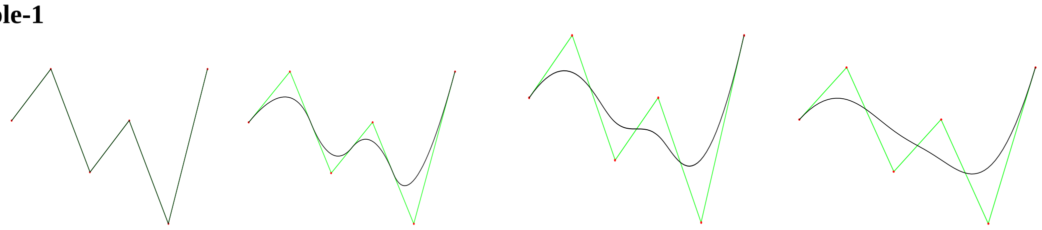

Fig:4 Different types of B-spline function

Example 2:
This transforms an image using a smooth vector field defined by a B-spline function

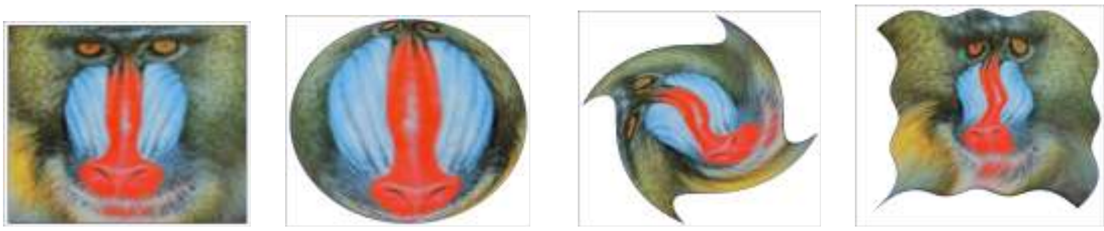

Fig:5 Transform Images Using B-Spline Functions

VI. Bezier Curve

Bezier curves can be defined by a recursive algorithm, which is how de Casteljau first developed them. It is also necessary, however, to have an explicit representation for them, i.e., to express a Bezier curve in terms of a non-recursive formula rather than an algorithm. Bezier curves have been employed in CAGD and many other fields, including [1,6,7,8]. The control points for each Bezier curve are unique, according to [9]. According to [10], the most prevalent topic of study in CAGD is the use of shape control parameters to generate Bezier curves. The Bezier curves are generalized into Generalized Bernstein-Bezier (GBB) curves in [11], which are also used for design purposes. Bernstein polynomials satisfy the recursion:

$B_i^n(t) = (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t)$ with $B_0^0(t) \equiv 1$ and $B_j(t) \equiv 0$ for $j \notin \{0,1,\dots,n\}$.

VII. The De Casteljau Algorithm

Paul de Faget de Casteljau, born in 1830, is a French physicist and mathematician [13]. While working at Citroen in 1959, he discovered a recursive scheme for determining points on a certain set of curves, which was eventually codified and popularized by engineer Pierre Bezier. Within the last 20 years de Casteljau's algorithms became a fundamental tool in CAGD [12]. The de Casteljau algorithm may be used to recursively compute points on Bezier curves of any rank. It makes use of consecutive linear interpolations to arrive at a location on the target curve. For example, to assess a cubic Bezier curve at a given value, four control points must be defined, and the method must be repeated 3 times. De Casteljau’s algorithms have been a fundamental tool in CAGD during the last twenty years. His concept of control points provided an easy understanding of why control points are so powerful in vehicle and ship design, the aerospace industry, and medical and geological representations.

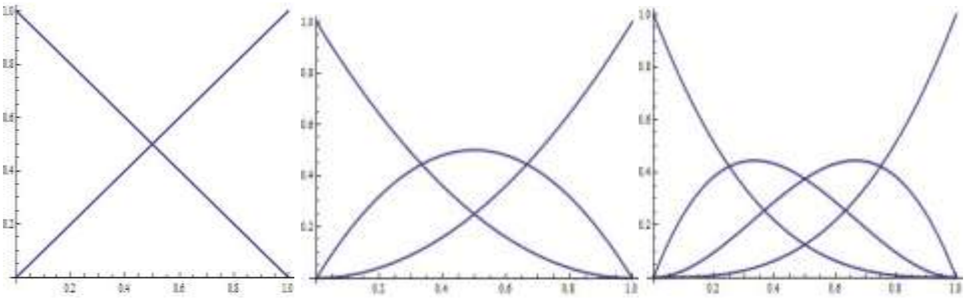

Fig 6 Bernstein Polynomials $B_i^1(t), B_i^2(t), B_i^3(t)$

VIII. Conclusion

The Bezier and B-spline methods are effective for interpolation and approximation of data, particularly for data with uneven or nonuniform distribution. Both approaches can be used with varied element sizes. The B-spline approach is more sensitive to abrupt changes in element size and unexpected shifts in data trends. As the element size deviates from uniformity, B-spline curves lose their smoother features.

References

[1] Farin, Gerald. Curves and surfaces for CAGD: a practical guide. Elsevier, 2001.
[2] Schoenberg, Isaac Jacob. "Contributions to the problem of approximation of equidistant data by analytic functions. Part B. On the problem of osculatory interpolation. A second class of analytic approximation formulae." Quarterly of Applied Mathematics 4.2 (1946): 112-141.
[3] Cox, Maurice G. "The numerical evaluation of B-splines." IMA Journal of Applied mathematics 10.2 (1972): 134-149.
[4] De Boor, Carl. "On calculating with B-splines." Journal of Approximation theory 6.1 (1972): 50-62.

[5] Lane, Jeffrey M., and Richard F. Riesenfeld. "A theoretical development for the computer generation and display of piecewise polynomial surfaces." IEEE Transactions on Pattern Analysis and Machine Intelligence 1 (1980): 35-46.
[6] Bézier, Pierre. The mathematical basis of the UNIURF CAD system. Butterworth-Heinemann, 2014.
[7] Bezier, Pierre. "Mathematical and practical possibilities of UNISURF." Computer aided geometric design. Academic Press, 1974. 127-152.
[8] Boehm, Wolfgang, and Andreas Müller. "On de Casteljau's algorithm." Computer Aided Geometric Design 16.7 (1999): 587-605.
[9] Berry, Thomas G., and Richard R. Patterson. "The uniqueness of Bézier control points." Computer Aided Geometric Design 14.9 (1997): 877-879.
[10] Qin, Xinqiang, et al. "A novel extension to the polynomial basis functions describing Bézier curves and surfaces of degree n with multiple shape parameters." Applied Mathematics and Computation 223 (2013): 1-16.
[11] Jena, Mahendra Kumar, P. Shunmugaraj, and P. C. Das. "A subdivision algorithm for generalized Bernstein–Bézier curves." Computer aided geometric design 18.7 (2001): 673-698.
[12] Boehm, Wolfgang, and Andreas Müller. "On de Casteljau's algorithm." Computer Aided Geometric Design 16.7 (1999): 587-605.
[13] De Casteljau, Paul de Faget. "De Casteljau's autobiography: My time at Citroën." Computer Aided Geometric Design 16.7 (1999): 583-586.