




Application of Cubic B-spline Curve in Computer-Aided Animation Design

Liang Li 

Art College, Anshan Normal University Liaoning China, Anshan, Liaoning 114001, China,
shizhonglai@163.com

Corresponding author: Liang Li, shizhonglai@163.com

Abstract. Curves are one of the main research objects of differential geometry, and curved surfaces are one of the important research contents. Curves and surfaces are widely used in real life, such as architectural design, animation design and industrial product appearance design. Curves and surfaces are closely related. With the rapid development of modern science and technology and the high precision requirements of curves and surfaces in manufacturing and industrial production, the traditional interpolation spline and approximation spline algorithms have been unable to meet the needs of actual industrial production. Cubic B-spline curve and surface approximation algorithm. This algorithm is based on cubic B-spline, avoids the disadvantages of traditional interpolation and approximation splines, combines the advantages of both, and studies the approximation fields of curves and surfaces. The development history of free curves and surfaces and the current development status of curves and surfaces are studied. This article briefly describes the cubic B-spline curve and its characteristics, summarizes several common methods for implementing computer animation, points out the control techniques for implementing animation design with cubic B-spline, and gives a method using cubic B-spline, to generate an instance of an animated pattern.

Keywords: Computer-Aided; Cubic B-spline; Animation Design; Graphic Page; Display Memory

DOI: <https://doi.org/10.14733/cadaps.2021.S1.43-52>

1 INTRODUCTION

An important aspect of computer graphics applications is shape design. The so-called shape design refers to the description, calculation, and display of curves and surfaces with a certain algorithm according to the position of a set of value points [1]. Earlier shape designs often used spline curves to fit type value points. It required that the generated curve passed a given number of type value points, such as a cubic parametric spline. However, in practical applications, when designing

the shape of a curve, a common practice for designers is to first draw a contour with a polyline, and then use a curve to approximate the contour of the polyline. Therefore, an overemphasis on generating the curve accurately through the originally inaccurate value points is not only an excessive requirement, but it may also lose the overall aesthetics of the curve. In the early 1970s, Albrecht, G. [2] and Pottmann, H. [3] of the United States and others first proposed the B-spline curve. It controls the B-spline curve through the B-spline control polygon. Local modification, that is, changing the position of one of the vertices will not affect the shape of the entire curve, but only its local part. In addition, it also has features such as closer control polygons and intuitive modification. Therefore, Hu et.al [4] said B-spline curves have received extensive attention in shape design, especially in animation design.

The so-called spline curve refers to a curve given a set of control points, and the approximate shape of the curve is controlled by these points. Spline curves can be generally divided into interpolation splines and approximation splines according to the results of calculations. Although the curves and surfaces finally generated by the interpolation spline can pass all the given data points, when the given data points are unevenly distributed, the generated curve will be inaccurate, and because the interpolation spline is in each sub-interval. Pottmann et al [3] said there is a spline sub-function, so it is very inconvenient to apply, and it also makes the resulting curve not conducive to real-time local modification and post-processing. This makes the interpolation spline only suitable for interpolation occasions such as mathematical lofting or establishing the motion path of objects, but not suitable for shape design. Although the approximation spline can ensure that the resulting curves and surfaces have good smoothness, the curves and surfaces it draws cannot pass all given data points, so they cannot meet specific accuracy requirements.

After systematic research literature [5], we found that the algorithm retains the advantages of B-spline intuitiveness, easy shape control, flexible design, strong local modification ability, simple calculation, and strong adaptability. At the same time, the algorithm has fast convergence speed, high accuracy and easy programming. Therefore, based on this algorithm, this paper extends the quadratic B-spline algorithm to the field of cubic B-splines, and uses the "accelerated" error method to improve the algorithm, and numerical experiments are programmed. The results show that the improved algorithm not only retains the advantages of the original algorithm, but also has faster convergence speed and fewer iterations. At the same time, since the algorithms in and [6] are based on the quadratic B-spline, and the quadratic B-spline is first-order continuously derivable. With higher requirements, the quadratic B-spline algorithm cannot meet the actual needs. In addition, in real life, first-order conductability sometimes fails to meet security requirements. For example, due to the busyness of China's main railway lines, in order to further meet the needs of people's lives and work, it is necessary to appropriately speed up the main railway lines. Wang et al [7] pointed out that China's railways are also capable of speeding up, but the most important issue is to ensure the safety of trains. Operation, and the design of railway curves is one of the key issues to ensure the safe and high-speed operation of trains. As we all know, in order to ensure that the train can run smoothly without generating a big bump, the turning curve of the railway track must be second-order steerable. At this time, the approximation algorithm based on the quadratic B-spline cannot meet the actual needs, and the approximation algorithm based on the cubic B-spline that is generalized and improved in this paper is second-order continuously derivable, which can meet the requirements of railway curve design special requirements. It can be seen that the algorithm in this paper has more practical application value.

2 CUBIC B-SPLINE CURVE AND ITS CHARACTERISTICS

2.1 Problem Description

In the following, the curve approximation algorithm based on quadratic s-splines is extended to the field of cubic s-splines and improved by the "acceleration" error method.

Hypothesis vector:

$$h = (\vec{h}_1 \vec{h}_2 \dots \vec{h}_n) = \begin{bmatrix} h_1^x & h_2^x & \dots & h_n^x \\ h_1^y & h_2^y & \dots & h_n^y \\ h_1^z & h_2^z & \dots & h_n^z \end{bmatrix} \tag{1}$$

For a given data point in three dimensions. Gan et.al [8] hope to draw a smooth curve $s(t)$ through h and makes $s(t)$ satisfy the second-order derivability through the algorithm in this paper. The cubic B-spline curve is shown in Figure 1.

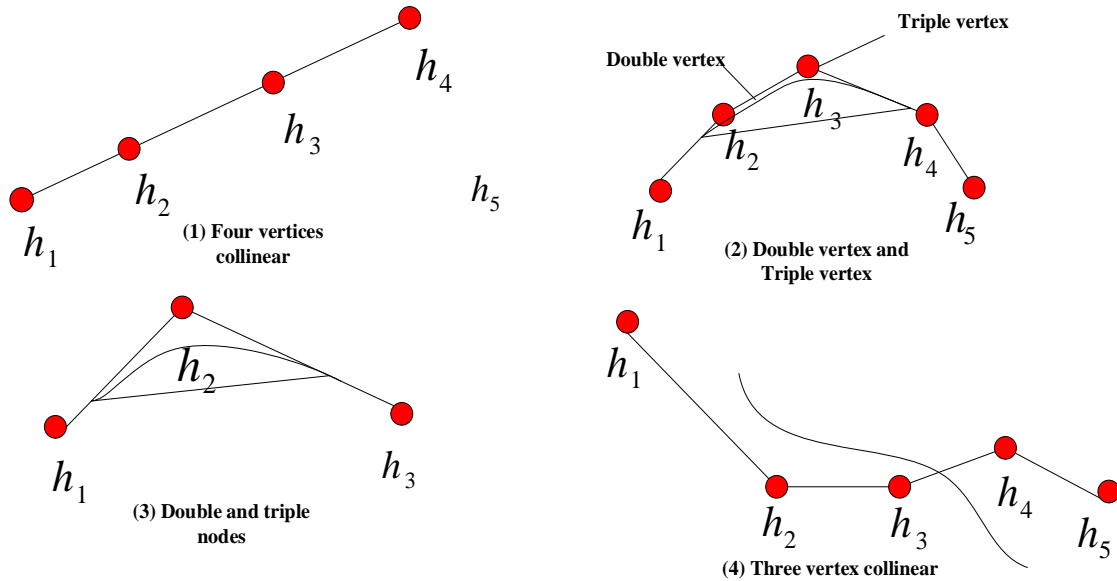


Figure 1: Cubic B-spline curve.

When using a cubic B-spline curve for animation design, the following techniques can be used to achieve the desired effect.

- 1) To make a curve somewhere tangent to the control polygon, use two control vertices to coincide.
- 2) To embed a straight line in the B-spline curve somewhere, you can use the four control vertices to achieve the same line.
- 3) To make the shape of the curve appear a sharp point or pass a certain control vertex point, use the three control vertex points to coincide.

2.2 Boundary Data Processing

In order for $s(t)$ to pass through the boundary point of h , Occelli et al [9] need to properly process the boundary data. In this paper, by supplementing the data $\vec{h}_{-1}, \vec{h}_0, \vec{h}_{n+1}, \vec{h}_{n+2}$ at the head and end points of h , make h after supplementing the boundary data a vector with periodic properties.

$$h' = (\vec{h}_{-1} \vec{h}_0 | \vec{h}_1 \vec{h}_2 \dots \vec{h}_n | \vec{h}_{n+1} \vec{h}_{n+2}) \tag{2}$$

among them $\vec{h}_{n+2} = \vec{h}_2, \vec{h}_{n+1} = \vec{h}_1, \vec{h}_n = \vec{h}_0$.

2.3 The Algorithm in this Paper

In this paper, based on the quadratic B-spline approximation algorithm, a certain generalization is made, and the "acceleration" error method by Xiong et al [10] is used to make the algorithm more efficient, while greatly reducing the number of iterations and iteration time of the algorithm. The steps for drawing a B-spline are shown in Figure 2.

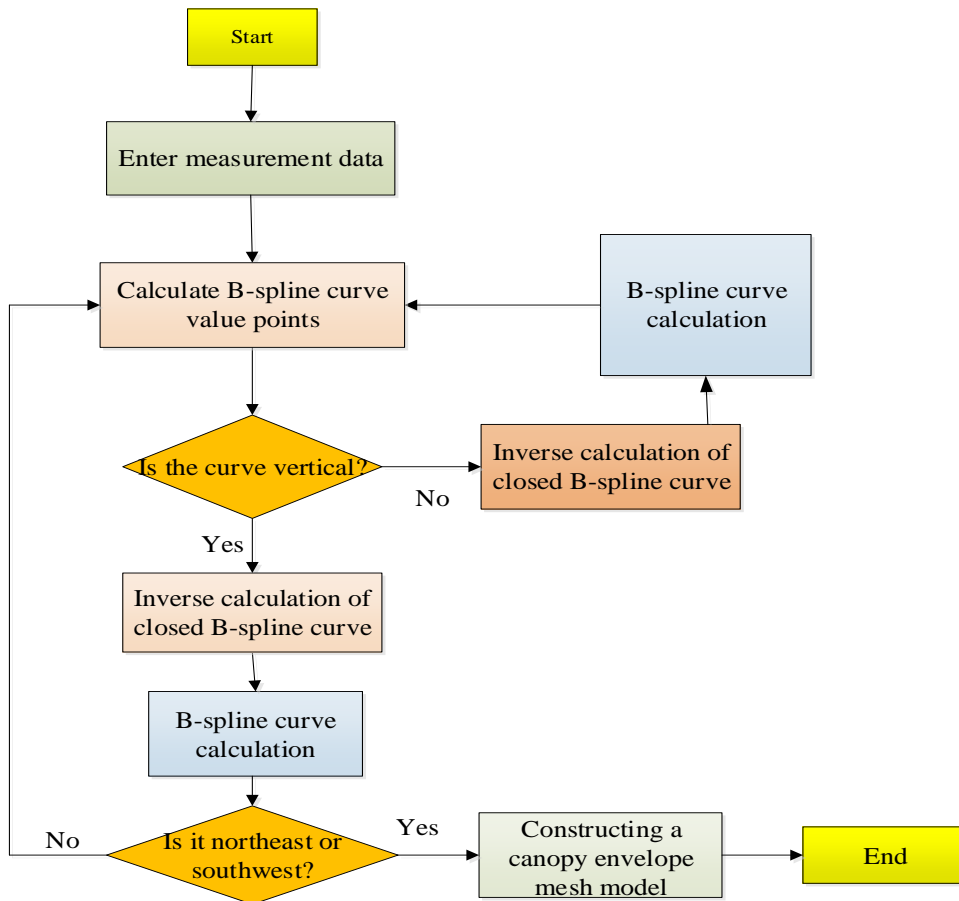


Figure 2: Steps to draw a B-spline.

The specific steps of the algorithm in this paper are given below:

Step 1: Use the vector supplemented \vec{h}' by h as the initial control point, and let $h^{<l>} = \vec{h}'$.

Step 2: According to the definition of cubic B-spline curve, the matrix form of cubic B-spline curve $s(t)$ is expressed as:

$$\vec{s}(t)^{\langle k \rangle} = (\vec{h}_{i-1}^{\langle k \rangle} \vec{h}_i^{\langle k \rangle} \dots \vec{h}_{i+1}^{\langle k \rangle}) \begin{bmatrix} \frac{1}{6} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{6} \\ \frac{2}{3} & 0 & -1 & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{6} \end{bmatrix} \begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \end{bmatrix} \quad (3)$$

$$\vec{s}(t)^{\langle k \rangle} = \begin{bmatrix} x_i(t) \\ y_i(t) \\ z_i(t) \end{bmatrix} \quad (4)$$

$$(\vec{h}_{i-1}^{\langle k \rangle} \vec{h}_i^{\langle k \rangle} \vec{h}_{i+1}^{\langle k \rangle} \vec{h}_{i+2}^{\langle k \rangle}) = \begin{bmatrix} \vec{h}_{i-1}^{x \langle k \rangle} \vec{h}_i^{x \langle k \rangle} \vec{h}_{i+1}^{x \langle k \rangle} \vec{h}_{i+2}^{x \langle k \rangle} \\ \vec{h}_{i-1}^{y \langle k \rangle} \vec{h}_i^{y \langle k \rangle} \vec{h}_{i+1}^{y \langle k \rangle} \vec{h}_{i+2}^{y \langle k \rangle} \\ \vec{h}_{i-1}^{z \langle k \rangle} \vec{h}_i^{z \langle k \rangle} \vec{h}_{i+1}^{z \langle k \rangle} \vec{h}_{i+2}^{z \langle k \rangle} \end{bmatrix} \quad (5)$$

Step 3: Use $\vec{h}^{\langle k \rangle} = (\vec{h}_{-1}^{\langle k \rangle} \vec{h}_0^{\langle k \rangle} | \vec{h}_1^{\langle k \rangle} \vec{h}_2^{\langle k \rangle} \dots \vec{h}_i^{\langle k \rangle} \vec{h}_n^{\langle k \rangle} | \vec{h}_{n+1}^{\langle k \rangle} \vec{h}_{n+2}^{\langle k \rangle})$, to record the corresponding coordinates on the approximate B-spline curve, where

$$\vec{h}^{\langle k \rangle} = \vec{s}(t)^{\langle k \rangle} \quad (i = 1, 2 \dots n)$$

And

$$\begin{aligned} \vec{h}_{-1}^{\langle k \rangle} &= \vec{h}_{n-1}^{\langle k \rangle}, \vec{h}_0^{\langle k \rangle} = \vec{h}_n^{\langle k \rangle}, \\ \vec{h}_{n+1}^{\langle k \rangle} &= \vec{h}_1^{\langle k \rangle}, \vec{h}_{n+2}^{\langle k \rangle} = \vec{h}_2^{\langle k \rangle}. \end{aligned}$$

Step 4: Calculate the error vector production $e = \vec{h}^{\langle k \rangle} - \vec{h}^{\langle k \rangle}$, and judge the error according to the actual accuracy requirements. If the accuracy meets the requirements, the program stops and the output draws a curve, otherwise the program will continue to execute step 5.

Step 5: Make $\vec{h}^{\langle k \rangle} = \vec{h}^{\langle k \rangle} + \frac{3}{2} e^{\langle k \rangle}$, let $\vec{h}^{\langle k \rangle}$ be the loop execute step2, step3, step4 as the new control point.

Among them, the control point required for the k-th cycle is recorded as $\vec{h}^{\langle k \rangle}$, the obtained B-spline approximate curve is recorded as $\vec{s}(t)^{\langle k \rangle}$, and the position of the corresponding sub-control point $\vec{h}^{\langle k \rangle}$ on the curve $\vec{s}(t)^{\langle k \rangle}$ is marked as $\vec{h}^{\langle k \rangle}$, and the error vector is recorded as $e^{\langle k \rangle}$.

3 EXPERIMENTAL RESULTS AND ANALYSIS

Computer animation design is mainly a series of still images displayed at different positions on the display screen. The general process is to display-remove-re-display, that is, to display the image at a certain position on the display screen and then eliminate it, then re-display the original image (or the modified image), and repeat.

Because the speed of switching between images is extremely fast (more than 30 frames / second), it feels as if the images are moving smoothly and continuously, thereby forming an animation effect. Compared with manual animation, computer animation has faster production speed, the advantages of realistic images, convenient modification, easy storage, and low cost are more and more popular. When using cubic B-spline curve for computer animation design, firstly use four control points to form a B-spline curve segment. Multi-segment B-spline connection can form a cubic B-spline curve that meets the requirements. Since the B-spline curve satisfies C2 continuity, it is smooth at the connection point.

3.1 Draw a Curve Based on Discrete Data Points

Let a given set of data be:

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 4 & 6 & 0 & -2 & -4 & -6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

According to the iterative method provided earlier, using MATLAB, we get Figure 3, Figure 4, and Table 1.

Number of iterations	1	2	3	4	5	6
K=1	1.472	4.278	1.897	4.178	3.276	1.084
K=2	4.027	5.68	0.872	5.287	3.764	-3.273
K=3	3.488	4.452	-0.345	4.873	3.928	-4.392
K=4	3.993	3.984	-0.043	5.983	3.029	-3.042
K=5	3.994	4.993	-0.001	5.983	3.002	-3.004
K=6	4.000	4.983	0.001	4.985	3.002	-3.003

Table 1: Table of changes in approach data.

3.2 Numerical Experiments

Surfaces are an ideal design material for 3D solid modeling. In the modeling design of various complex industrial products today, parametric surface is an effective design tool. The shapes of industrial products can be roughly divided into two categories: the first category consists of primary analytical surfaces (such as planes, cylindrical surfaces, conical surfaces, spherical surfaces, torus surfaces, etc.), and most mechanical parts fall into this category. The descriptive geometry and mechanical drawing methods can be used to express and transmit all the shape information completely and clearly; the second type cannot be composed of elementary analytical surfaces, but curved surfaces that change freely in complex ways, so-called free-form curves.

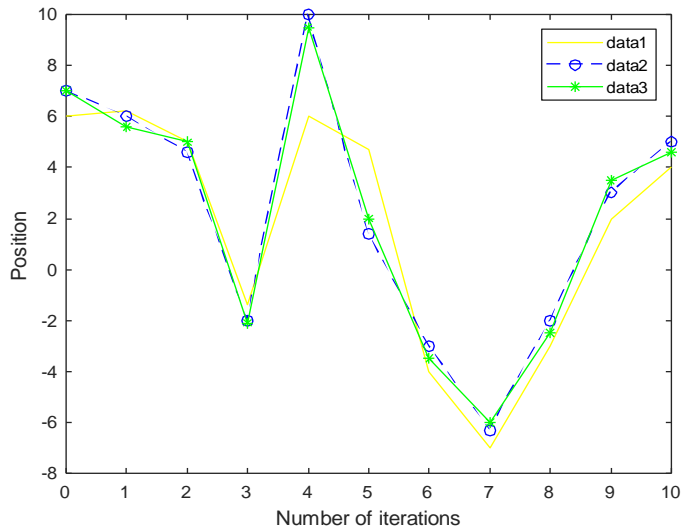


Figure 3: Curve approximation process based on cubic B-spline algorithm.

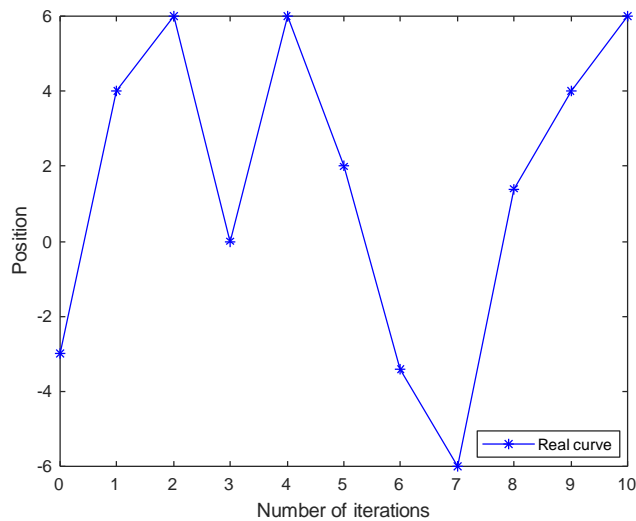


Figure 4: Final Approximation Effect.

In the actual production of industrial products, this type of surface is the most used and most valuable. For example, the curved surface design of a car bumper, the curved surface design of an aircraft wing, and the exterior design of a building. Obviously, free-form curves and surfaces can't meet the requirements of modern industrial products simply relying on traditional mechanical drawing and geometric drawing. In order to realize the application of free-form curves and surfaces, we need to fit the given points by interpolation or approximation, so that the curves and surfaces generated by the fitting can meet the actual design requirements.

In order to avoid the shortcomings of the two methods of interpolation and approximation when constructing free-curve surfaces, and to integrate the advantages of the two, we generalize the curve approximation algorithm based on cubic B-spline to the field of surface approximation. We hope that with this surface approximation algorithm, we can use this algorithm to draw a surface that can pass these type points while ensuring smoothness, given the type points.

(1) Numerical experiment 1

To test the function, a near-surface numerical experiment was performed. Figure 5 and Figure 6 are the comparison between the graphics drawn directly by MATLAB and the approximate images drawn by the algorithm in this paper according to the function analysis formula.

In this digital experiment, we divide a $10 < x < 10$, $10 < y < 10$ into 64 equal parts, respectively, and generate 64 control points. According to the 64 control points of x and Y , according to the algorithm provided in this article, the control is performed. The points are iterated to obtain a near surface image that finally meets the accuracy requirements.

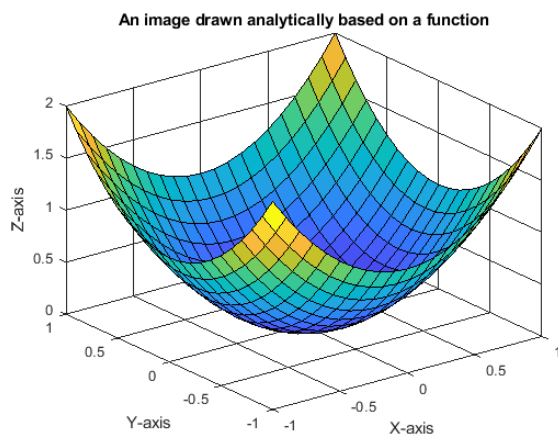


Figure 5: Function original image drawn according to function analysis.

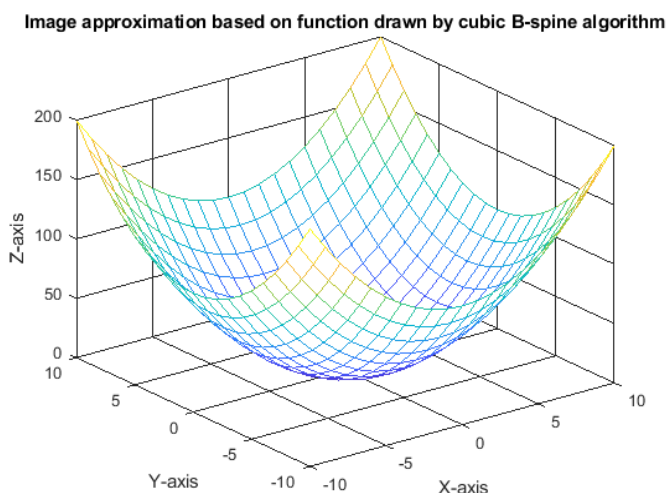


Figure 6: Approximation of a function based on a cubic B-spline algorithm.

(2) Numerical experiment 2

A set of several dimensional data points are given below as experimental data. Surface approximation mapping is performed using the algorithm in this paper. Part of the data is shown in Table 2. Figure 7 shows the experimental results.

x	y	z
1.487	3.058	0.1
2.216	4.041	0.1
1.739	3.497	0.2
2.134	3.472	0.2
3.438	3.980	0.3
4.053	3.078	0.3
2.879	3.048	0.4
3.492	3.792	0.4
3.822	3.012	0.5
3.411	2.019	0.5
3.710	3.048	0.6
3.402	2.672	0.6

Table 2: Part of the data in the numerical experiment.

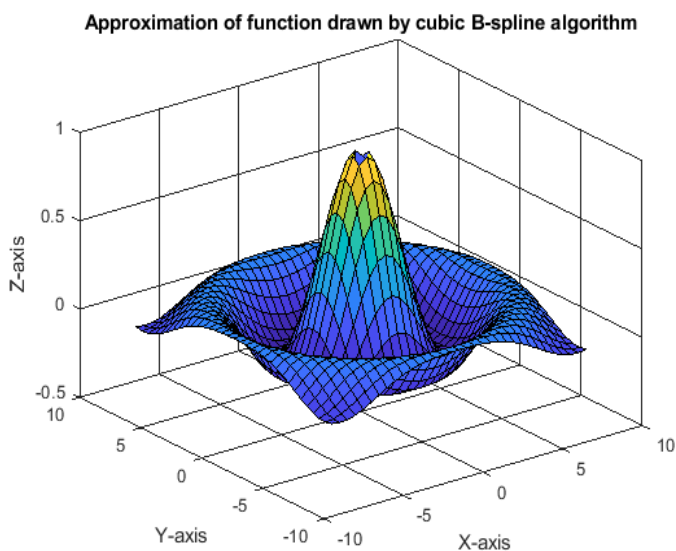


Figure 7: Function approximation based on a cubic B-spline algorithm.

4 CONCLUSIONS

This paper mainly promotes and improves the curve approximation algorithm based on quadratic B-spline, and proposes a curve approximation algorithm based on cubic B-spline. Then, the curve approximation algorithm is extended to the field of surface approximation. The theoretical proof

and numerical experiments show that the approximation algorithm not only retains the advantages of the quadratic B-spline algorithm, but also has faster convergence speed and fewer iterations, which improves the efficiency of the algorithm. The surface drawn using the cubic approximation algorithm based on cubic B-spline in this paper has second-order inducibility, which can obtain a smooth surface, which can better meet the needs of actual industrial production than quadratic B-spline. In addition, the "acceleration" error method adopted in this paper has greatly improved the efficiency of the algorithm. It is of great practical value today in pursuit of efficiency.

This article can be extended in the theoretical proof of convergence. The boundary data is supplemented by the periodic boundary supplement method. However, in practice, different boundary supplement methods may be used according to the different requirements of the industry on products. So the theory of this article proves that there are certain limitations. In proving the convergence of the surface approximation algorithm, we carried out iterative proofs with specific values, without giving general conclusions. At the same time, the method of matrix vectorization in the literature borrowed from this paper makes the iteration matrix very large, which makes the proof process relatively complicated.

Liang Li, <https://orcid.org/0000-0002-0092-6511>

REFERENCES

- [1] Xia, Q.; Chen, C.; Liu, J.; Li, S.; Hao, A.; Qin, H.: Efficient 4D shape completion from sparse samples via cubic spline fitting in linear rotation-invariant space, *Computers & Graphics*, 8(2), 2019, 129-139. <https://doi.org/10.1016/j.cag.2019.05.014>
- [2] Albrecht, G.; Beccari, C.-V.; Canonne, J.-C.; Romani, L.: Planar pythagorean-hodograph B-spline curves, *Computer Aided Geometric Design*, 5(7), 2017, 57-77. <https://doi.org/10.1016/j.cagd.2017.09.001>
- [3] Pottmann, H.; Hofer, M.: A variational approach to spline curves on surfaces, *Computer Aided Geometric Design*, 22(7), 2005, 693-709. <https://doi.org/10.1016/j.cagd.2005.06.006>
- [4] Hu, G.; Qin, X.; Ji, X.; Wei, G.; Zhang, S.: The construction of $\lambda\mu$ -B-spline curves and its application to rotational surfaces, *Applied Mathematics and Computation*, 26(6), 2015, 194-211. <https://doi.org/10.1016/j.amc.2015.05.056>
- [5] Xing, Y.; Xu, R.-Z.; Tan, J.-Q.; Fan, W.; Hong, L.: A class of generalized B-spline quaternion curves, *Applied Mathematics and Computation*, 27(1), 2015, 288-300. <https://doi.org/10.1016/j.amc.2015.09.025>
- [6] Beccari, C.-V.; Casciola, G.: A Cox-de Boor-type recurrence relation for C1 multi-degree splines, *Computer Aided Geometric Design*, 7(5), 2019, 101784. <https://doi.org/10.1016/j.cagd.2019.101784>
- [7] Wang, A.; Zhao, G.: Wavelets based quantitative design of B-spline curves, *Expert systems with applications*, 41(15), 2014, 6871-6875. <https://doi.org/10.1016/j.eswa.2014.04.029>
- [8] Gan, W.-F.; Fu, J.-Z.; Shen, H.-Y.; Chen, Z.-Y.; Lin, Z.-W.: Five-axis tool path generation in CNC machining of T-spline surfaces, *Computer-Aided Design*, 5(2), 2014, 51-63. <https://doi.org/10.1016/j.cad.2014.02.013>
- [9] Occelli, M.; Elguedj, T.; Bouabdallah, S.; Morançay, L.: LR B-Splines implementation in the Altair RadiossTM solver for explicit dynamics IsoGeometric Analysis, *Advances in Engineering Software*, 13(1), 2019, 166-185. <https://doi.org/10.1016/j.advengsoft.2019.01.002>
- [10] Xiong, Y.; Li, G.; Mao, A.: Convergence analysis for B-spline geometric interpolation, *Computers & Graphics*, 36(7), 2012, 884-891. <https://doi.org/10.1016/j.cag.2012.07.002>