Journal of Applied Mathematics and Computation, 2023, 7(2), 298-303
https://www.hillpublisher.com/journals/JAMC/
ISSN Online: 2576-0653
ISSN Print: 2576-0645

Design of Biquintic B-spline Surface Construction Algorithm

Shuhui Zhang*, Xiuping Liu

School of Mathematical Sciences, Dalian University of Technology, Dalian, Liaoning, China.

How to cite this paper: Shuhui Zhang, Xiuping Liu. (2023) Design of Biquintic B-spline Surface Construction Algorithm. *Journal of Applied Mathematics and Computation*, 7(2), 298-303.
DOI: 10.26855/jamc.2023.06.011

Received: June 3, 2023
Accepted: June 29, 2023
Published: July 30, 2023

***Corresponding author:** Shuhui Zhang, School of Mathematical Sciences, Dalian University of Technology, Dalian, Liaoning, China.

Abstract

The tensor product *Bézier* patch and B-spline patch are two of the most popular and widely used parametric surface representations. B-spline surfaces play an important role in the CAD/CAM/CAE. The tensor product B-spline surfaces provide continuity without the imposition of constraints in surface fitting process. This work concentrates on the basic concept, properties of B-spline surfaces. Firstly, we introduce the B-spline basis function and its properties and its calculation algorithm. In addition, B-spline curves and B-spline surfaces are meticulously studied, especially biquintic B-spline surface, and then we present the algorithm of generating biquintic B-spline surface. Finally, based on the algorithm of generating biquintic B-spline surface we proposed, a biquintic B-spline surfaces is generated based on MATLAB, and the running result is given. Besides, we analyze the results of the operation and come to a conclusion: biquintic B-spline surfaces generally don't pass through any vertices of the control mesh (also called convex hull).

Keywords

B-spline, B-spline surface, Surface construction

1. Introduction

With the development of technology, people's production and life are more and more influenced by computer graphics, which continuously affects everyone who uses computers. Computer aided geometry design (CAGD) is a kind of applied subject which rises with the modern manufacturing industry such as ship, automobile and aircraft. The commonly used curved surfaces in computer graphics include interpolating curves and surfaces, *Bézier* curves and surfaces, B-spline curves and surfaces, and non-uniform rational spline curves and surfaces. The tensor product *Bézier* patch and B-spline patch are two of the most popular and widely used parametric surface representations in the fields of computer graphics, geometric modeling, CAD/CAM and reverse engineering because of their elegant geometric properties [1-3]. B-spline surfaces have good local properties, so they are more and more widely used in engineering design. The biquintic is the lowest degree for which there exists a local scheme of G1 smooth B-spline surfaces with interior single knots [5]. It is very necessary to study biquintic B-spline curves and surfaces.

2. B-spline Curves and Surfaces

2.1 Advantages of B-spline Curves

Curves consisting of just polynomial or rational segment are often inadequate. Their shortcomings are:

- a high degree is required in order to satisfy a large number of constraints; e.g. (n-1)-degree is needed to pass a polynomial *Bézier* curve through n data points. However, high degree curves are inefficient to process and are numerically unstable;

- a high degree is required to accurately fit some complex shapes;
- single-segment curves(surfaces) are not well-suited to interactive shape design; although $Bézier$curves can be shaped by means of their control points (and weights), the control is not sufficiently local.

In 1972, Gordon et al. used B-spline basis instead of Bernstein basis function to improve the above shortcomings. First, the B-spline curve is closer to the control polygon than the $Bézier$ curve, and it is smoother, and the polynomial degree of its basis function can be given as needed. Secondly, B-spline curve can modify the curve locally. Since B-spline curve is composed of segments, the control of the vertices of the control polygon on the curve is flexible and intuitive. Modification of a control point will only cause changes in the shape of curves near and near the control point, while the shape of curves far away will not be affected, which makes B-spline widely used in the design of interactive free curve and surface.

2.2 Definition of the Basis Functions of B-spline

Let $U = \{u_0, \dots\dots, u_m\}$ be a nondecreasing sequence of real numbers, i.e., $u_i \le u_{i+1}$, $i = 0, \dots\dots, m-1$. The $u_i$are called knots, and $U$is the knot vector. The i-th basis function of p-degree (order p+1) B-spline, denoted by $N_{i,p}(u)$, is defined as

$$N_{i,0}(u) = \begin{cases} 1 & if\ u_i \le u \le u_{i+1} \\ 0 & otherwise \end{cases} \tag{2.1}$$

$$N_{i,p}(u) = \frac{u-u_i}{u_{i+p}-u_i} N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}(u) \tag{2.2}$$

For brevity we often write $N_{i,p}$ instead of $N_{i,p}(u)$.

Note that

1) $N_{i,0}(u)$ is a step function, equal to zero everywhere except on the half-open interval $u \in [u_i, u_{i+1})$;
2) for $p > 0, N_{i,p}(u)$ is a linear combination of two $(p-1)$-degree basis functions;
3) computation of a set of basis functions requires specification of a knot vector,U, and the degree, p;
4) $N_{i,p}(u)$ can yield the quotient; we define this quotient to be zero;
5) the computation of the p-th degree function generates a truncated triangular table:

$$\begin{array}{ccccc}
N_{0,0} & & & & \\
 & N_{0,1} & & & \\
N_{1,0} & & N_{0,2} & & \\
 & N_{1,1} & & N_{0,3} & \\
N_{2,0} & & N_{1,2} & & \ddots \\
 & N_{2,1} & & N_{1,3} & \cdots \\
N_{3,0} & & N_{2,2} & \vdots & \\
 & N_{3,1} & \vdots & & \\
N_{4,0} & \vdots & & & \\
\vdots & & & &
\end{array}$$

Fig. 1. *p*-th degree function generates a truncated triangular table.

2.3 Definition and Properties of B-spline Curves

A p-th degree B-spline curve is defined by

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u)\boldsymbol{P_i}, \qquad a \le u \le b \tag{2.3}$$

Where the $\{\boldsymbol{P_i}\}$ are the control points, and the $\{N_{i,p}(u)\}$are the p-th degree B-spline basis function (2.1) (2.2) defined on the nonperiodic (and nonuniform) knot vector ($m+1$ knots):

$$U = \{\underbrace{a, \dots\dots, a}_{p+1}, u_{p+1}, \dots\dots, u_{m-p-1}, \underbrace{b, \dots\dots, b}_{p+1}\}$$

Unless stated otherwise, we assume that $a = 0$and $b = 1$. The polygon formed by the $\{\boldsymbol{P_i}\}$ is called the control polygon.

Now, we list a number of properties of B-spline curves.

(1) $C(u)$is a piecewise polynomial curve (since the $N_{i,p}(u)$are piecewise polynomials); the degree, p, number of control points, $n+1$, and number of knots, $m+1$, are related by

$$m = n + p + 1 \tag{2.4}$$

(2) Strong convex hull property: the curve is contained in the convex hull of its control polygon; in fact, if $u \in [u_i, u_{i+1}), p \le i \le m-p-1$, then $C(u)$is in the convex hull of the control points $\boldsymbol{P_{i-p}}, \dots\dots, \boldsymbol{P_i}$.(Fig.6 is an example)

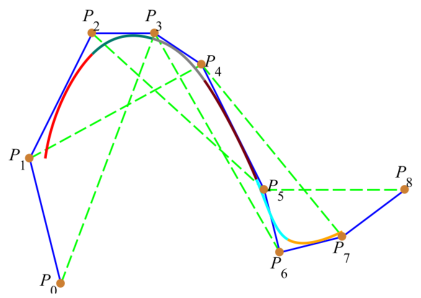

Fig. 2. The picture of Strong convex hull property.

(3) Local modification scheme: moving $\boldsymbol{P_i}$ changes $C(u)$ only in the interval $[u_i, u_{i+p+1})$.

(4) Moving along the curve from $u=0$to $u=1$, the $N_{i,p}(u)$ functions act like switches.

(5) The derivatives of $C(u)$:

$$C^{(k)}(u) = \sum_{i=0}^{n-k} N_{i,p-k}(u)\, P_i^{(k)} \tag{2.5}$$

with $P_i^{(k)} = \begin{cases} P_i & k=0 \\ \frac{p-k+1}{u_{i+p+1}-u_{i+k}}\left(P_{i+1}^{(k-1)} - P_i^{(k-1)}\right) & k>0 \end{cases}$ and $U^{(k)} = \{\underbrace{0,\dots\dots,0}_{p-k+1}, u_{p+1},\dots\dots,u_{m-p-1},\underbrace{1,\dots\dots,1}_{p-k+1}\}$.

A B-spline surface is obtained by taking a bidirectional net of control points, two knot vectors, and the products of the univariate B-spline functions

$$S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v)\, \boldsymbol{P_{i,j}} \tag{2.6}$$

with

$$U = \{\underbrace{0,\dots\dots,0}_{p+1}, u_{p+1},\dots\dots,u_{r-p-1},\underbrace{1,\dots\dots,1}_{p+1}\}$$

$$V = \{\underbrace{0,\dots\dots,0}_{q+1}, v_{q+1},\dots\dots,v_{s-q-1},\underbrace{1,\dots\dots,1}_{q+1}\}$$

U has $r+1$knots and Vhas $s+1$. Besides, $r = n+p+1$and $s = m+q+1$.

In addition, $S(u,v)$also takes the form:

$$S(u,v) = N_{p,k}(u)^T * \boldsymbol{P_{k,l}} * N_{l,q}(v),\ i-p \le k \le i, j-q \le l \le j \tag{2.7}$$

Where $N_{p,k}(u)^T$is a $1\times(p+1)$row vector of scalars, $\boldsymbol{P_{k,l}}$is a $(p+1)\times(q+1)$matrix of control points, and $N_{l,q}(v)$is a $(q+1)\times 1$column vector of scalars.

Now, we list a number of properties of B-spline surfaces.

(1) Nonnegativity: $N_{i,p}(u)N_{j,q}(v) \ge 0$ for all i,j,p,q,u,v.

(2) $N_{i,p}(u)N_{j,q}(v) = 0$if $(u,v)$is outside the rectangle $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$.

(3) Strong convex hull property: if $(u,v) \in [u_{i_0}, u_{i_0+1}) \times [v_{j_0}, v_{j_0+1})$, then $S(u,v)$is in the convex hull of the control points $\boldsymbol{P_{i,j}}, i_0 - p \le i \le i_0$and $j_0 - q \le j \le j_0$.

(4) Local modification scheme: if $\boldsymbol{P_{i,j}}$is moved it affects the surface only in the rectangle $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$

(5) The derivatives of $S(u,v)$:

$$\frac{\partial^{k+l}}{\partial^k u \partial^l v} S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}^{(k)} N_{j,q}^{(l)} \boldsymbol{P}_{\boldsymbol{i},\boldsymbol{j}} = N_{r,p}^{(k)}(u)^T * \boldsymbol{P}_{\boldsymbol{r},\boldsymbol{s}} * N_{s,q}^{(l)}(v) \tag{2.8}$$

3. Construction of a biquintic B-spline Surface

3.1 Algorithm Design

The biquintic B-spline surface is extended from the quintic B-spline curve, and the spatial mesh is constructed from the control points of two sets of orthogonal quintic B-spline curves to generate surfaces. Given 100 control points, a 10*10 mesh is generated through the given control points, the basis function of the B-spline surface is algorithmically designed, and the values are assigned from the $u$direction and $v$direction respectively, so that the control point and the basis function are multiplied and summed, so as to generate the corresponding type value point, connect the type value point, and perform cyclic operation through programming, and then generate the B-spline surface.

3.2 Algorithm implementation

As can be seen from (2.6), the B-spline surface expression is

$$S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} \boldsymbol{P}_{\boldsymbol{i},\boldsymbol{j}} N_{i,p}(u) N_{j,q}(v)$$

with

$$U = \{\underbrace{0, \dots \dots, 0}_{p+1}, u_{p+1}, \dots \dots, u_{r-p-1}, \underbrace{1, \dots \dots, 1}_{p+1}\}$$

$$V = \{\underbrace{0, \dots \dots, 0}_{q+1}, u_{q+1}, \dots \dots, u_{s-q-1}, \underbrace{1, \dots \dots, 1}_{q+1}\}$$

U has $r+1$knots and Vhas $s+1$. Besides,

Given $\boldsymbol{P}_{\boldsymbol{i},\boldsymbol{j}}$,$U$ and V, the spatial mesh formed by two adjacent points in $\boldsymbol{P}_{\boldsymbol{i},\boldsymbol{j}}$ is called the control mesh. And the expression for the biquintic B-spline surface is

$$S(u,v) = \sum_{i=0}^{5} \sum_{j=0}^{5} \boldsymbol{P}_{\boldsymbol{i},\boldsymbol{j}} N_{i,5}(u) N_{j,5}(v), \qquad (u,v) \in [0,1] \times [0,1] \tag{3.1}$$

besides,

$$S(u,v) = \left[N_{0,5}(u) N_{1,5}(u) N_{2,5}(u) N_{3,5}(u) N_{4,5}(u) N_{5,5}(u)\right] * \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} & P_{0,4} & P_{0,5} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} & P_{1,4} & P_{1,5} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} & P_{2,4} & P_{2,5} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} & P_{3,4} & P_{3,5} \\ P_{4,0} & P_{4,1} & P_{4,2} & P_{4,3} & P_{4,4} & P_{4,5} \\ P_{5,0} & P_{5,1} & P_{5,2} & P_{5,3} & P_{5,4} & P_{5,5} \end{bmatrix} * \begin{bmatrix} N_{0,5}(v) \\ N_{1,5}(v) \\ N_{2,5}(v) \\ N_{3,5}(v) \\ N_{4,5}(v) \\ N_{5,5}(v) \end{bmatrix}. \tag{3.2}$$

Given $\boldsymbol{P}_{\boldsymbol{i},\boldsymbol{j}}$, the algorithm of generating biquintic B-spline surface is as follows:

Table 1. The algorithm of generating biquintic B-spline surface

Algorithm: The algorithm of generating biquintic B-spline surface
Input: $\boldsymbol{P}_{\boldsymbol{i},\boldsymbol{j}}$ **Output:** Abiquintic B-spline surface $S(u,v)$ 1: Determine the given knot vectorU, V type that is now required. 2: Calculate the expression of quintic basis function of B-spline:$N_{0,5}(u)N_{1,5}(u)N_{2,5}(u)N_{3,5}(u)N_{4,5}(u)N_{5,5}(u)$. 3: Determine the value of u, v and calculate the value of $N_{0,5}(u)N_{1,5}(u)N_{2,5}(u)N_{3,5}(u)N_{4,5}(u)N_{5,5}(u)$ given the value of u, v. 4: Calculate $S(u,v)$by (3.2). 5: The resulting biquintic B-spline surface is drawn according to the value of $S(u,v)$.

4. Result

In MATLAB R2021a, we program (2.1) (2.2) to get the basis function of B-spline surface, and then (3.1) is used to program to finally obtain the biquintic B-spline surface. Now, we give a set of $\boldsymbol{P_{i,j}}$ with knot vectors$U$ and V:

$$PX = \begin{bmatrix} 1 & 1 & 2 & 1 \\ 4 & 4 & 4 & 4 \\ 7 & 6 & 7 & 8 \\ 10 & 9 & 10 & 9 \end{bmatrix}; PY = \begin{bmatrix} 1 & 3 & 6 & 9 \\ 0 & 3 & 6 & 9 \\ 0 & 3 & 6 & 9 \\ 1 & 4 & 7 & 10 \end{bmatrix}; PZ = \begin{bmatrix} 3 & 5 & 5 & 2 \\ 4 & 6 & 7 & 4 \\ 4 & 7 & 6 & 5 \\ 2 & 4 & 5 & 4 \end{bmatrix};$$

$$U = V = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$$

note that PX is the $x$value of $\boldsymbol{P_{i,j}}$; PY is the $x$value of $\boldsymbol{P_{i,j}}$; PX is the $x$value of $\boldsymbol{P_{i,j}}$; we can get the biquintic B-spline surface generating by the $\boldsymbol{P_{i,j}}$,U and V.

From Fig.4, a biquintic B-spline surface is an interwoven quintic B-spline curve. Besides, biquintic B-spline surfaces generally don't pass through any vertices of the control mesh (also called convex hull).

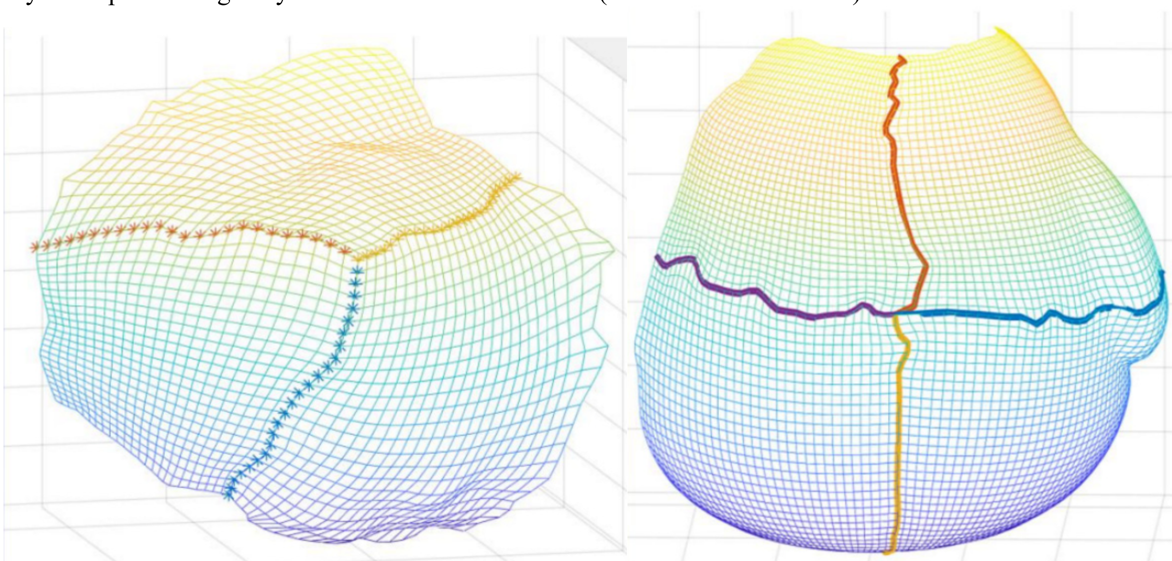

Fig. 4. The picture of biquintic B-spline surface generating by given $\boldsymbol{P_{i,j}}$, $\boldsymbol{U}$ and $\boldsymbol{V}$. The hollow points are $\boldsymbol{P_{i,j}}$.

5. Conclusion

This paper provides a comprehensive and systematic introduction to the definition, properties, and algorithms of the basis function of B-spline,B-spline curves, B-spline surfaces, and other related knowledge. In addition, the article gives expressions of biquintic B-spline surfaces and shows an example of biquintic B-spline surface in MATLAB.

References

[1] Piegl, L., Tiller, W., 1997. The NURBS Book, second ed. Springer.

[2] De Boor C. A practical guide to splines. Berlin: Springer; 1978.

[3] Farin G. Curves and surfaces for computer aided geometric design: a practical guide. New York: Academic Press; 1993.

[4] Eck M, Hoppe H. Automatic reconstruction of B-spline surfaces of arbitrary topological type. ACM Compute Graph SIGGRAPH 1996; 325–34.

[5] Shi X, Wang T. A practical construction of G1 smooth biquintic B-spline surfaces over arbitrary topology. Computer Aided Design 36 (2004) 413-424.

[6] Shi X, Wang T. Reconstruction of convergent G1 smooth B-spline surfaces. Computer Aided Geometric Design 21 (2004) 893-913.

[7] Yang H P, Wang W P, Sun J G. Control point adjustment for B-spline curve approximation [J]. Computer-Aided Design, 2003, 36(7):639-652.

[8] Varady T, Martin RR, Cox J. Reverse engineering of geometric models: an introduction [J]. Computer-Aided Design, 1997, 29(4):255-268.

[9] Sederbergtw, Parrysr.Free-form deformation of solid geometric models [J]. ACM SIGGRAPH Computer Graphics, 1986, 20(4): 151-153.

[10] Gordon. W. J & Ricsenfeld. R. F. Bernstein-Bezier Methods for the Computer-Aided Design of free-Form Curve and Suefaces, J.ACM, 1974, Vol.21, No.2, 193-310.

[11] Woodward C. Skinning techniques for interactive B-spline surface interpolation [J]. Computer-Aided Design, 1988, 20 (8): 441-451.

[12] Wang W K, Zhang H, Park H, et al. Reducing control points in lofted B-spline surface interpolation using common knot vector determination [J]. Computer-Aided Design, 2008, 40 (8): 999-1008.

[13] Park H. Lofted B-spline surface interpolation by linearly constrained energy minimization [J]. Computer-Aided Design, 2003, 35 (14): 1261-1268

[14] PieglL, TillerW.Reducing control points in surface interpolation [J]. IEEE Computer Graphics and Applications, 2000, 20 (5): 70-74.

[15] Huang W J, Wang Z G.B-spline surface smooth splicing method [J]. Mechanical engineering and automation, 2020.

[16] Li B, Wu L J, Han S, et al. Design and implementation of B-spline surface construction algorithm [J]. Henan Science and Technology, 2019, (2):14-16.

[17] Wu L J, Han S, Li B. Design and research of B-spline surface splicing method [J]. Journal of Shenyang Normal University (Natural Science Edition), 2018, 36(6):545-549.

[18] Han S. Design and implementation of B-spline surface stitching algorithm [D]. PhD thesis. Shenyang: Shenyang Normal University, 2019.

[19] Jiang S F, Zhang Y Q. B-spline surface transition/fine-tuning reduction method in reverse reconstruction [J]. Journal of Beijing Institute of Technology, 2018, 38(8):802–807.