B-spline Based on Vector Extension Improved CST Parameterization Algorithm

Bowen Yan[1], Yuanyuan Si[2], Zhaoguo Zhou[1], Wei Guo[1], Hongwu Wen[1], and Yaobin Wang[3]

[1]Military Transportation University
[2]Characteristic Medical Center of Chinese People's Armed Police Force
[3]Southwest University of Science and Technology

July 4, 2023

Abstract

In this paper, the vector extension operation is proposed to replace the de Boor-Cox formula for a fast algorithm to B-spline basis functions. This B-spline basis function based on vector extending operation is implemented in the class and shape transformation (CST) parameterization method in place of the traditional Bezier polynomials to enhance the local ability of control and accuracy to represent an airfoil shape. To calculate the k-degree B-spline function's nonzero values, the algorithm can improve the computing efficiency by 2k+1 times.

Hosted file

`Manuscript.tex` available at https://authorea.com/users/635542/articles/652860-b-spline-based-on-vector-extension-improved-cst-parameterization-algorithm

Highlights

B-spline Based on Vector Extension Improved CST Parameterization Algorithm

Bowen Yan,Yuanyuan Si,Zhaoguo Zhou,Wei Guo,Hongwu Wen,Yaobin Wang

- Using B-spline to improve the CST parameterization algorithm increases the parameterization design space and improves the local control ability of the CST algorithm.
- The vector expansion method is used to realize the fast evaluation of B-spline, and the calculation efficiency is increased by 2k+1 times.

B-spline Based on Vector Extension Improved CST Parameterization Algorithm

Bowen Yan[a], Yuanyuan Si[b,<], Zhaoguo Zhou[a], Wei Guo[a], Hongwu Wen[a] and Yaobin Wang[c,<]

[a]Army Military Transportation University, Tianjin, China
[b]Characteristic Medical Center of Chinese People's Armed Police Force, Tianjin, China
[c]Southwest University of Science and Technology, Mianyang, China

ARTICLE INFO

Keywords
Vector Extension
de Boor-Cox Formula
B-spline Functions
CST
Bezier Polynomials

ABSTRACT

In this paper, the vector extension operation is proposed to replace the de Boor-Cox formula for a fast algorithm to B-spline basis functions. This B-spline basis function based on vector extending operation is implemented in the class and shape transformation (CST) parameterization method in place of the traditional Bezier polynomials to enhance the local ability of control and accuracy to represent an airfoil shape. To calculate the k-degree B-spline function's nonzero values, the algorithm can improve the computing e□ciency by 2k+1 times.

1. Introduction

The aerospace achievement of a country has become an ever-important symbol of whether the country is a strong and big power under the current complex international situation. The exploration and development of technologies in the aviation field have a profound influence on a country's economy and national defense, among other aspects. One important indicator of whether a country has strong air superiority and mature and advanced air transport, among other aspects, is the aircraft. Undoubtedly, a country that has more advanced, higher-speed, and safer aircraft will be endowed with more advantages [24, 8, 27, 3, 36]. Designing outstanding aircraft has, therefore, become the objective of many scholars and the development of national defense science and technology.

The most basic, as well as the most important step of aircraft design, is to design its shape. Civil aircraft can only survive the fierce competition in the market when it is economical, safe, and reduces its manufacturing cost to the highest possible extent [6, 13, 14, 34, 15]. However, the development of the fighter has been transferred from the past when the high-altitude, high-speed performance was emphasized to today when the high maneuverability of the high subsonic speed and transonic speed in low- and medium-altitude, among others, is emphasized [30, 17, 21, 35, 40].

It is well known that the quality of the shape design of the aircraft is one of the important factors that can determine whether the aircraft is enabled su□cient lift to keep it flying, whether the aircraft has good maneuverability and stealth performance, and whether the aircraft can have good aerodynamic performance [25, 2, 1, 43, 33]. The first step of the shape design of the aircraft is to parameterize the shape, which has a critical influence on the success of the aircraft design and directly a□ects the overall project progress. A good parameterization methodology should have the following features — firstly, it should have a large optimization space; secondly, it has good local control ability and a relatively small number of design variables; finally, it needs to be ensured that the geometric shape of the aircraft in the optimization process is smooth [12, 9, 18, 47, 28]. The shape design of the aircraft is also a crucial step in the aerodynamic optimization of the aircraft that determines the overall aerodynamic performance of the aircraft. Hence the first step of the shape design of the aircraft is to employ a proper parameterization method to generate a basic aircraft shape [5, 46, 42]. Because the shape of the aircraft needs constant modification in the overall design phase, it has become a must-be-solved problem as to how to quickly generate the three-dimensional shape model of the aircraft and realize it in a precise and complete way[32, 45, 11, 41, 19].

The Class and Shape Transformation (CST) method put forward by Kulfan [22, 7] has gained a wide application due to its features such as excellent robustness and smooth geometric description capacity. The main idea of the CST method is to control the shape using the Class Function and the Shape Function [23]. The Class Function will generate the basic shape of the geometric figures, and the Shape Function constructed by Bernstein Polynomial will rectify the basic geometric figure so that the features of the shape changes can be described [10]. The German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt; DLR) has verified the application of the method in the design of the flying-wing layout aircraft; Liu C Z et al. [29] achieved the aerodynamic layout optimization design of the supersonic aircraft with the method. The global features of the basic function of the Bernstein Polynomial decide that the method can generate geometric shapes with relatively high smoothness, and at the same time decides that the method has the shortcoming of inadequate descriptive capacity of the partial geometric characteristics. Wang X et al. [50] proposed an improved method where the B-spline basis function is used to replace the Bernstein Polynomial, to

[<]Corresponding author
(B. Yan); (Y. Si);
(Z. Zhou); (W. Guo);
(H. Wen); (Y. Wang)
ORCID(s): (B. Yan)

improve the partial descriptive capacity of the shape. The B-spline method shows a good effect in data fitting, smoothing, and interpolation, so, the method does reach an optimization when the order is relatively small. However, the B-spline basis function [49] is a piecewise-defined function that cannot be expressed by a unified analytical expression. Most of the functions will be evaluated through an iterative method where multiple repetitive calculations occur; the calculation amount dramatically increases as the order increases. Wang X et al. only used a 3-order B-spline basis function to prove this[50]. The paper, therefore, employs the vector extension operation that can achieve the quick evaluation of the B-spline basis function to further optimize the CST method, improve the calculation efficiency of the modified B-spline basis function CST method, and maximize the value of the modified B-spline basis function CST method.

2. B-spline substitute Bezier polynomial modeling

2.1. B-spline basis function definition

The basic function of the B-spline can be represented by many definitions, and the most widely recognized of them is the de Boor-Cox recursive definition method. In this paper, we choose the de Boor-Cox formula as the standard definition of b-spline basis functions. It not only demonstrates the properties but provides a clear recursive solution algorithm of the B spline basis function [4, 38, 39, 44].

B-spline basis functions of k order can be defined as:

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1 & u \in [u_i, u_{i+1}) \\ 0 & \text{others} \end{cases} \\ N_{i,k}(u) = \frac{u-u_i}{u_{i+k}-u_i} N_{i,k-1}(u) + \frac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}} N_{i+1,k-1}(u) \end{cases} \tag{1}$$

Where:

u_i: The value of the ith nodes;

$\vec{U} = [u_0, u_1, \ldots, u_n]$: The sequence of nodes represented by vectors, n being an integer;

$u_{i,k}(u)$: The value of the ith basis function of k-order B-spline when the parameter is u.

2.2. B-spline basis function definition

Defining an airfoil shape function and specifying its geometry class is equivalent to defining the actual airfoil coordinates which are obtained from the shape function and class function as:

$$\zeta(\psi) = C^{N1}_{N2}(\psi)\, S(\psi) + \psi \Delta\zeta_T \tag{2}$$

Where:

ψ = x_c and ζ = y_c.

The term $S(\psi)$ is the shape function.

The term $C^{N1}_{N2}(\psi)$ is the class function.

The term $\psi\Delta\zeta_T$ provides control of the trailing edge thickness.

After replacing the shape function with the B-spline basis function, the CST method can be expressed as:

$$\zeta(\psi) = C^{N1}_{N2}(\psi) \sum_{i=0}^{n} y_i N_{i,k}(\psi) + \psi \Delta\zeta_T \tag{3}$$

The CST method using the B-spline basis function can improve the ill-conditioned equation solution and enhance the local ability of control.

Examples of decompositions of the unit shape function using various orders of Bernstein polynomials and B-spline basis function are shown in figure 1.

3. The Quick Evaluation Method of B-spline Basis Function

3.1. The Disadvantages of de Boor-Cox Formula

In the actual calculation of B-spline [26, 16, 20, 31], it is not necessary to calculate the whole B-spline value, sometimes only the B-spline basis function of a parameter value is calculated. As the B-spline basis function has local support property, we just need to figure out $N_{m,k}(u), m = i-k, i-k+1, \ldots, i$ when $u \in [u_i, u_{i+1})$. Under this occasion, the value of the other K-order B-spline would be 0. This property explicitly states the calculation range of the solution of B-spline basis function, which is helpful to improve the efficiency of the algorithm.

Figure 2 shows that the calculation path of the k-order B-spline basis function is marked with a dotted triangle. It can be intuitively seen that each of the base functions depends not only on the two adjacent basis functions of the next layer but also on the two adjacent basis functions on the upper layer. The calculation method of equation 1 does not make full use of this grid structure, and the overlapped parts of the two dashed triangles have a large number of repeated computations.

3.2. Preparations

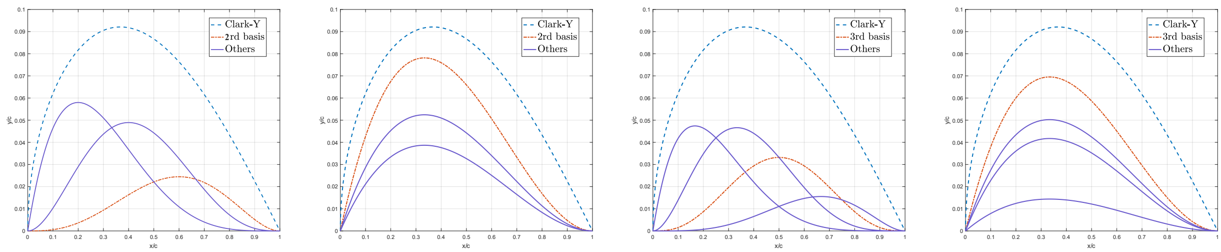
Before we elaborate on the quick evaluation method of the B-spline basis function, we shall first define an algorithm for vector extension [48, 37].

Definition 1

Given n+1 dimensional vector $\vec{N} = [N_0, N_1, \ldots, N_{n-1}, N_n]$, N_j represents the element of vector $\vec{N}$, when $j < 0$ or $j < n$, we shall refer to vector $\vec{N}$ as subscript overflow.

Definition 2

Given n+1 dimensional vector $\vec{N} = [N_0, N_1, \ldots, N_{n-1}, N_n]$ and n+1 dimensional vector $\vec{A} = [a_0, a_1, \ldots, a_{n-1}, a_n]$; $N_j, a_j \in R$, R being the real number field, $j = 0, 1, \ldots, n$; n being an integer. After multiply operation of the elements of vector $\vec{N}$, $\vec{A}$ — N_j, a_j according to corresponding sequence numbers, we obtain the n+1 dimensional middle vector $\vec{Q} = [Q_0, Q_1, \ldots, Q_{n-1}, Q_n]$, and $Q_j = a_j N_j$, $j = 0, 1, \ldots, n$; moreover, we prescribe that when subscript overflow occurs on vectors $\vec{Q}$, $\vec{N}$, their corresponding elements would be 0. Therefore, the n+2 dimensional vector generated according to equation 4 $\vec{Z} = [Z_0, Z_1, \ldots, Z_{n-1}, Z_n, Z_{n+1}]$ shall be referred to as the extended vector of vector $\vec{N}$ and vector $\vec{A}$,



(a) 2-order Bernstein Polynomial (b) 2-order B-spline (c) 3-order Bernstein Polynomial (d) 3-order B-spline

Figure 1: Algorithm Comparison

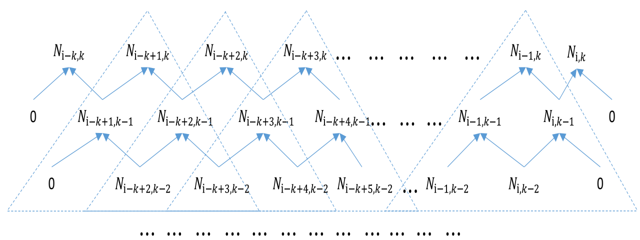


Figure 2: The Calculation Path of the k-order B-spline Basis Function

of which $\tilde{N}$ shall be referred to as primary vector and $\tilde{A}$ as auxiliary vector.

$$Z_j = N_j + Q_{j-1} - Q_j \tag{4}$$

The equation provides an operational rule for the combination of 2 m+1 dimensional vectors into 1 m+2 dimensional vector, which is referred to as vector extension operation. It is a mapping from vector to vector. Here we introduce the operational sign for vector extension $<\tilde{N}|\tilde{A}>$, operator $<\ |\ >$ being the mapping from $R^n \to R^{n+1}$, R being the real number field.

Definition 3

Given vector $\tilde{N} \in R^n$, vector cluster $\{\tilde{C}^i\}_{i=n}^{n+m-i}$, $\tilde{C}^i \in R^i$, $i = n, n+1, \cdots, n+m-1$, R being the real number field, m,n,i being integers. Vector $\tilde{N}$ is the primary vector while $\{\tilde{C}^i\}_{i=n}^{n+m-i}$ is the auxiliary vector cluster. First, from vector extension operation of $\tilde{N}$, $\tilde{C}^n$, we have vector $<\tilde{N}|\tilde{C}^n>$, which is still the primary vector; then, have vector extension operation between $<\tilde{N}|\tilde{C}^n>$ and $\tilde{C}^{n+1}$, after m times of extension, obtaining n+m dimensional new vector $\tilde{M}$, which shall be referred to as the m-time extended vector of primary vector $\tilde{N}$ and auxiliary vector cluster $\{\tilde{C}^i\}_{i=n}^{n+m-i}$, expressed as:

$\tilde{M} = << \cdots <<< \tilde{N}|\tilde{C}^n>|\tilde{C}^{n+1}>|\tilde{C}^{n+2}>|\tilde{C}^{n+3}>|\cdots>\tilde{C}^{n+m-1}>$

This recursive operation is favorable for computer calculation. We refer to the operation given by Definition 3 as an operation for recursive extension of a vector.

3.3. Proofs to the Fast Algorithm

We shall begin with description of symbols to be used:

$\tilde{U}$: The sequence of nodes represented by vectors, $\tilde{U} = [u_0, u_1, \cdots, u_n]$;

$N_{j,m}(u)$: The jth spline basis function of m-order B-spline, where $m = 0, 1, 2, \cdots, k$;

N_i^m: The value of the ith basis function of m-order B-spline $N_{i,m}(u)$ when $u \in [u_i, u_{i+1})$, where $m = 0, 1, 2, \cdots, k$;

$\tilde{N}^m$: The vector consists of the sequence of m-order B-spline non-zero spline basis function value $[N^m_{i-m}, N^m_{i-m+1}, \cdots, N^m_i]$ when $u \in [u_i, u_{i+1})$, containing m+1 elements, where $m = 0, 1, 2, \cdots, k$;

a_j^m: Coefficient of node when $u \in [u_i, u_{i+1})$;

$$a_j^m = \frac{u - u_j}{u_{j+m} - u_j} \tag{5}$$

where $1 \le m \le k; i - m + 1 \le j \le i$; prescribe $\frac{0}{0} = 0$

$\tilde{A}^m$: The vector consists of the sequence of code coefficients when $u \in [u_i, u_{i+1})$, $[a^m_{i-m+1}, a^m_{i-m+2}, \cdots, a^m_i]$, containing m elements, where $m = 0, 1, 2, \cdots, k$.

Before proving the quick evaluation of the B-spline basis function, we need to first prove Lemma 1.

Lemma 1

For m-order B-spline, when $u \in [u_i, u_{i+1})$, vector $\tilde{N}^m$ is the extended vector with $\tilde{N}^{m-1}$ as the primary vector and $\tilde{A}^m$ as the auxiliary vector, symbolized as $\tilde{N}^m = <\tilde{N}^{m-1}|\tilde{A}^m>$.

Proof

According to the local support property of B-spline, $\tilde{N}^m$ is a m+1 dimensional vector, N^m_{i-m+j} is an element of vector $\tilde{N}^m$ with the sequence number j, $j = 0, 1, 2, \cdots, m$; both vector $\tilde{A}^m$ and vector $\tilde{N}^{m-1}$ are m dimensional vectors, $N^{m-1}_{i-m+1+r}$, $a^m_{i-m+1+r}$ being the elements of them respectively with the sequence number r, $r = 0, 1, 2, \cdots, m-1$. Since the vector extending operation is the mapping of $R^n \to R^{n+1}$, we have a m+1 dimensional vector $\tilde{M}$ as:

$\tilde{M} = <\tilde{N}^{m-1}|\tilde{A}^m> = [M_0, M_1, \cdots, M_m]$.

Pursuant to the vector extending algorithm under Definition 2, we have:

$$M_j = N^{m-1}_{i-m+1+j} + a^m_{i-m+j} N^{m-1}_{i-m+j} - a^m_{i-m+1+j} N^{m-1}_{i-m+1+j} \tag{6}$$

where: $0 \le j \le m$, with the prescription that the corresponding element shall be reset 0 upon subscript overflow of the vector.

According to the equations 5 and 6, we have:

$$M_j = \frac{u - u_{i-m+j}}{u_{i+j} - u_{i-m+j}} N^{m-1}_{i-m+j} + \frac{u_{i+1+j} - u}{u_{i+1+j} - u_{i-m+1+j}} N^{m-1}_{i-m+1+j} \quad (7)$$

where: $0 \le j \le m$.

Comparing equation 1 with equation 7, we notice that when $j = 0, 1, 2, \cdots, m$ in equation 7, $N^{m}_{i-m+j} = M_j$, i.e. the elements of vector $\tilde{M}$ and vector $\tilde{N}^m$ present one-to-one equivalence, namely, $\tilde{M} = \tilde{N}^m$.

Therefore, $\tilde{N}^m = < \tilde{N}^{m-1} | \tilde{A}^m >$ is established.

Theorem 1

For the k-order B-spline, when $u \in [u_i, u_{i+1})$, $\tilde{N}^m$ is the m-time expanded vector of 1 - dimensional unit primary vector $\tilde{N}^0 = [1]$ and auxiliary vector cluster $\{\tilde{A}^1\}^m_{i=l}$, where $l = 1, 2, 3, \cdots, m$.

symbolized as:

$\tilde{N}^m = << \cdots <<<< [1] | \tilde{A}^1 > | \tilde{A}^2 > | \tilde{A}^3 > | \tilde{A}^4 > | \cdots > \tilde{A}^m >$

Proof

According to Lemma 1, with the inductive method, we only need to prove the establishment of $\tilde{N}^1 = < [1] | \tilde{A}^1 >$.

Given that:

$\tilde{A}^1 = [a^1_i] = [\frac{u - u_i}{u_{i+1} - u_i}]$, $\tilde{N}^1 = [N^1_{i-1}, N^1_i]$

And we know from Definition 1 that:

$N^1_{i-1} = \frac{u_{i+1} - u}{u_{i+1} - u_i}$, $N^1_i = \frac{u - u_i}{u_{i+1} - u_i}$

i.e., $\tilde{N}^1 = [\frac{u_{i+1} - u}{u_{i+1} - u_i}, \frac{u - u_i}{u_{i+1} - u_i}]$

and: $< [1] | \tilde{A}^1 > = [\frac{u_{i+1} - u}{u_{i+1} - u_i}, \frac{u - u_i}{u_{i+1} - u_i}]$

Therefore, $\tilde{N}^1 = < [1] | \tilde{A}^1 >$ is established. The proposition is proved.

Theorem 1 provides us with a recursive vector extending method for the realization of B-spline basis function evaluation, whose calculation path is as shown in Figure 3. We learn from Figure 3 that the quick evaluation method of vector extension for the B-spline basis function is just a reticular computing mechanism. With 1 - dimensional unit vector[1] as the primary vector, through recursive extension, the method realizes the synchronous computation of the value of B-spline basis functions with the same times (the rectangle with dotted lines in Figure 3). The recursive vector extension method can effectively avoid repeated calculation and thus improve computational efficiency, which is fully demonstrated by the absence of overlap in the calculation paths shown in Figure 3. To sum up, the evaluation method of vector extension for the B-spline presents higher algorithmic efficiency than the de Boor-Cox algorithm in equation 1 concerning the calculation path.

3.4. Algorithm Flow

The quick evaluation method of B-spline basis function is based on a recursive vector operation which is favorable for computer calculation. In vector extension operation, the main operation is the calculation of auxiliary vector cluster. Pursuant to equation 5, we know the calculation of vector cluster $\{\tilde{A}^m\}^k_{m=1}$ is only related to node vector $\tilde{U}$ and parameter u.

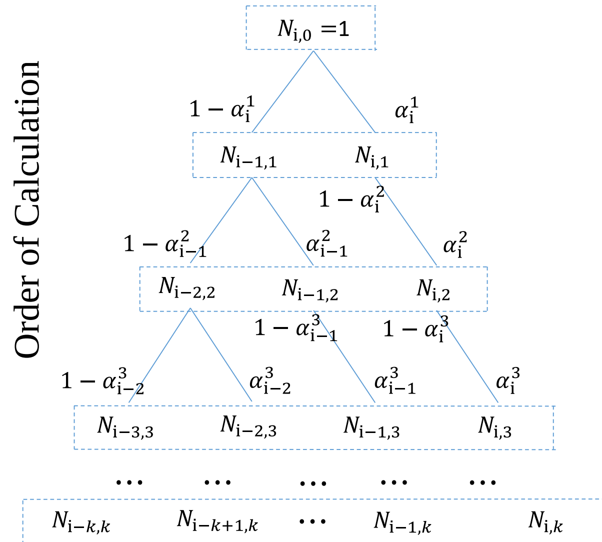


Figure 3: The Calculation Path of the Quick Evaluation Method of Vector Extension for B-spline Basis Function

For node vector $\tilde{U} = [u_0, u_1, u_2, \cdots, u_n]$, the parameter value is the k-order B-spline of u, The computer algorithm flow of the quick evaluation method of B-spline basis function of vector extension is as follows:

Step 1: Initialize the vector member variable: B-spline order number variables m=0, the interval of the parameter values of B-spline Current_i=0, primary vector $\tilde{N}^m = \tilde{N}^0 = [1]$;

Step 2: According to the parameter u, determine its range $u \in [u_i, u_{i+1})$, find the current node interval value Current_i=i.

Step 3: Update parameter m=m+1, Computing auxiliary vector $\tilde{A}^m = [a^m_{i-m+1}, a^m_{i-m+2}, \cdots, a^m_i]$;

Step 4: According to vector $\tilde{N}^{m-1}$ and $\tilde{A}^m$, Computing vector $\tilde{Q} = [a^m_{i-m+1} N^{m-1}_{i-m+1}, a^m_{i-m+2} N^{m-1}_{i-m+2}, \cdots, a^m_i N^{m-1}_i]$;

Step 5: According to equation 4, we get the new Vector $\tilde{N}^m = [N^m_{i-m}, N^m_{i-m+1}, \cdots, N^m_i]$;

Step 6: If m>k ,we choose step 7, otherwise we choose step 3.

Step 7: The result of the calculation is Vector $\tilde{N}^m$.

3.5. Efficiency of Algorithm

Compared with the de Boor-Cox algorithm, the evaluation method of vector extension for B-spline basis functions presents higher algorithmic efficiency. In comparison between equation 1 and equation 4, the calculation quantity of non-zero values of k-order B-spline basis functions, we have the recursive vector extension method is 2k+1 times as efficient as the de Boor-Cox algorithm.

Figure 5 shows the comparison of efficiency between B-spline and de Boor-Cox algorithm. Clark-Y(de Boor-Cox) represents the Clark-Y airfoil generated using the CST algorithm based on the de Boor-Cox formula. Clark-Y(B-spline) represents the Clark-Y airfoil generated by the modified CST algorithm using B-spline basis function based on vector extending operation. The remaining two curves represent the

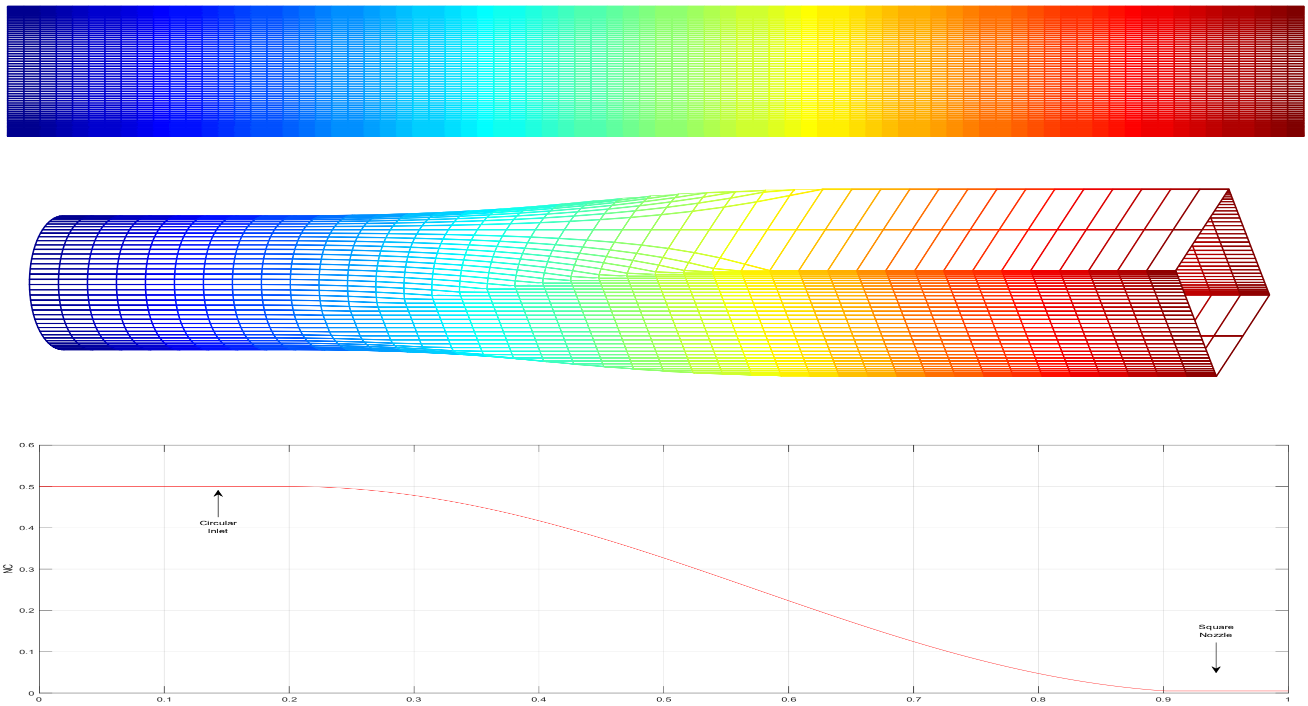


Figure 4: B-spline Local Control Ability

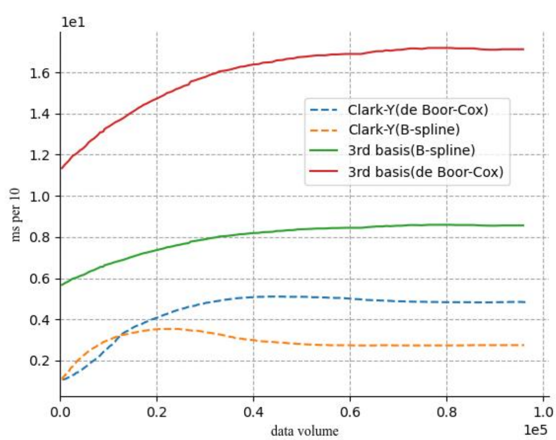


Figure 5: Eﬃciency Comparison

Table 1
Comparison of the calculation quantity of the (k+1)th non-zero basis function of m-order B-spline

	Multiplication	Division
Vector Extension	k.k + 1/_2	k.k + 1/_2
de Boor-Cox	k.k + 1/^{2}	k.k + 1/^{2}

eﬃciency comparison between the B-spline improved CST algorithm and the de Boor-Cox formula under 3rd basis. With the increasing amount of data, B-spline basis function based on vector extending operation has shown better performance advantages. The physical hardware configuration used in the above experiments is shown in Table 2.

Table 2
Hardware Configuration

Name	Properties
CPU	12th Gen Intel(R) Core(TM) i7-1260P
Memory	16G
Hard Disk	450G
Operating System	CentOS 7.9.2009

4. Application of B-spline Based on Vector Extension

4.1. Local Control Ability of B-spline Based on Vector Extension

Figure 4 shows the control of the cross-section from the circular to square using the CST method improved by B-spline based on vector extension. The initial geometry shape at the inlet is a circular duct defined with a cross-section class function with exponents equal to 0.5. The duct geometry, in this example, retains a constant cross-section from 0 to 20% of the length. The last 5% length of the duct has a square cross-section which has class function exponents equal to 0.005. In between 20% and 95% of the length, the cross-section which has class function exponents was decreased from 0.5 at 20% to 0.001 at 95% by a cubic variation with zero slopes at both ends.

4.2. Two-Dimensional Model Using the Improved CST Method

Figure 6 shows we use a B-spline based on vector extension to improve the CST method to generate 2d wings.

Figure 6: The Clark-Y Wing Based on B-spline Vector Extension Method

4.3. Three-Dimensional Wing Using the Improved CST Method

The three-dimensional wing is essentially an extension of two-dimensional wings, which requires attention to control the change of appearance. Figure 7 shows we use a B-spline based on vector extension to improve the CST method to generate 3d wings.

5. Conclusion

B-spline based on vector extension can make a quick evaluation which will improve the computational e ciency of the B-spline CST method, increase the design space, improve the controllability, and nally lay the foundation for a good aerodynamic shape by improving the accuracy of subsequent airfoil aerodynamic optimization.

Acknowledgment

This work is supported nancially by the Sichuan Natural Science Foundation for Distinguished Young Scholar (2023NSFSC1966), National Natural Science Foundation of China (61672438).

Con ict of Interest

Authors have no con ict of interest relevant to this article.

References

[1] Abdessemed, C., Bouferrouk, A., Yao, Y., 2022. E ects of an unsteady morphing wing with seamless side-edge transition on aerodynamic performance. Energies 15, 1093.

Figure 7: 3D Wing Using B-spline Vector Extension CST Method.

[2] Auteri, F., Savino, A., Zanotti, A., Gibertini, G., Zagaglia, D., Tekap, Y.B., Braza, M., 2022. Experimental evaluation of the aerodynamic performance of a large-scale high-lift morphing wing. Aerospace Science and Technology 124, 107515.

[3] Barnhart, C., Belobaba, P., Odoni, A.R., 2003. Applications of operations research in the air transport industry. Transportation Science 37, 368 391.

[4] Barrera, D., Eddargani, S., Lamnii, A., 2022. A novel construction of b-spline-like bases for a family of many knot spline spaces and their application to quasi-interpolation. Journal of Computational and Applied Mathematics 404, 113761.

[5] Bennehard, Q., 2022. Aerodynamic shape optimization of a short medium range blended wing body aircraft, in: 56th 3AF International Conference AERO2022.

[6] Bravo-Mosquera, P.D., Catalano, F.M., Zingg, D.W., 2022. Unconventional aircraft for civil aviation: A review of concepts and design methodologies. Progress in Aerospace Sciences 131, 100813.

[7] BRENDA, M.K., 2007. A universal parametric geometry representation method- CST . Technical Report. AIAA-2007-62.

[8] Chau, T., Kenway, G., Kiris, C.C., 2022. Conceptual exploration of aircraft con gurations for the susan electrofan, in: AIAA SciTech 2022 Forum, p. 2181.