Mathematical and Computational Applications

Article

A Class of Cubic Trigonometric Automatic Interpolation Curves and Surfaces with Parameters

Juncheng Li

College of Mathematics and Finance, Hunan University of Humanities, Science and Technology, Loudi 417000, China; lijuncheng82@126.com; Tel.: +86-738-8325585

Academic Editor: Mehmet Pakdemirli
Received: 5 March 2016; Accepted: 16 May 2016; Published: 20 May 2016

Abstract: This paper presents the cubic trigonometric interpolation curves with two parameters generated over the space $\{1, \sin t, \cos t, \sin^2 t, \sin^3 t, \cos^3 t\}$. The new curves can not only automatically interpolate the given data points without solving equation systems, but are also C^2 and adjust their shape by altering values of the two parameters. The optimal interpolation curves can be determined by an energy optimization model. The corresponding interpolation surfaces have characteristics similar to the new curves.

Keywords: trigonometric polynomial; interpolation curves and surfaces; C^2 continuity; shape adjustment

---

1. Introduction

It is well known that the trigonometric polynomials have important applications in different areas, such as electronics or medicine [1]. Recently, trigonometric polynomials have received much attention within geometric modeling. For example, Han [2] presented a class of cubic trigonometric polynomial curves with a shape parameter, Zhang [3] constructed uniform B-splines by using trigonometric and hyperbolic basis, Mainar [4] used a class of trigonometric Bernstein basis constructing Bézier-like curve, Wang [5] constructed three types of splines by using trigonometric functions, Han [6] defined the cubic trigonometric Bézier curve with two shape parameters, Han [7] used five trigonometric blending functions defining a class of curve, Bashir [8,9] presented the rational quadratic and cubic trigonometric Bézier curve with two shape parameters, Han [10] constructed the symmetric trigonometric polynomial curves like Bézier curves based on normalized B-basis of the space of trigonometric polynomials of degree n, Yan [11] presented an algebraic-trigonometric blended piecewise curve with two shape parameters, Yan [12] constructed a class of cubic trigonometric non-uniform B-spline curves with local shape parameters, and so on.

In order to construct interpolation curves, Su [13] and Yan [14] constructed the trigonometric curves over the spaces $\{1, t, \sin t, \cos t, \sin 2t, \cos 2t\}$ and $\{1, t, \sin t, \cos t, \sin^2 t, \sin^3 t, \cos^3 t\}$. The two trigonometric curves presented in [13,14] can automatically interpolate the given data points without solving equation systems, which provides a simple and efficient way to construct interpolation curves. However, although the quasi-cubic blended interpolation curves defined by Su [13] could interpolate the given data points automatically, their shapes cannot be adjusted when the data points are fixed. The xyB curves defined by Yan [14] could interpolate the given data points automatically and adjust shape by changing the parameter x when the data points and auxiliary points are fixed, but they are only G^2 and have one degree of freedom in the interpolation curves. The main purpose of this paper is to present a class of trigonometric interpolation curves, who can not only automatically interpolate the given data points without solving equation systems, but are also C^2 and have two degrees of freedom in the interpolation curves.

The rest of this paper is organized as follows. In Section 2, the cubic trigonometric interpolation basis functions with two parameters generated over the space {1, sin$t$, cos$t$, sin^2t, sin^3t, cos^3t} are presented, and some properties of the basis functions are given. In Section 3, the interpolation curves are defined on the base of the basis functions and some properties of the curves are given. Then, determining the optimal interpolation curves is discussed. In Section 4, the corresponding interpolation surfaces are presented. A short conclusion is given in Section 5.

2. The CTI-Basis Functions

The cubic trigonometric interpolation basis functions with two parameters are defined as follows.

Definition 1. *For $0 \leqslant t \leqslant 1$, $\alpha, \beta \in R$, the following four functions about t are called cubic trigonometric interpolation basis functions with parameters α and β (CTI-basis functions for short),*

$$\left\{\begin{array}{l} f_0(t) = \frac{1}{24}\left((-14\alpha - 2\beta + 6) + (9\alpha + 3\beta - 9)S + 24\alpha S^2 + (-19\alpha - \beta + 3)S^3 + (14\alpha + 2\beta - 6)C^3\right) \\ f_1(t) = \frac{1}{24}\left((2\alpha - 10\beta + 6) + (-9\alpha - 3\beta + 9)C + 24\beta S^2 + (-2\alpha - 14\beta - 6)S^3 + (7\alpha + 13\beta + 9)C^3\right) \\ f_2(t) = \frac{1}{24}\left((2\alpha + 14\beta + 6) + (-9\alpha - 3\beta + 9)S - 24\beta S^2 + (7\alpha + 13\beta + 9)S^3 + (-2\alpha - 14\beta - 6)C^3\right) \\ f_3(t) = \frac{1}{24}\left((10\alpha - 2\beta + 6) + (9\alpha + 3\beta - 9)C - 24\alpha S^2 + (14\alpha + 2\beta - 6)S^3 + (-19\alpha - \beta + 3)C^3\right) \end{array}\right. \tag{1}$$

where $S := \sin\frac{\pi}{2}t$, $C := \cos\frac{\pi}{2}t$.

Remark 1. *In order to construct curves interpolating the given data points automatically, we also previously defined two kinds of basis functions. The first kind of basis functions are defined over the space {1, sint, cost, sin2t, cos2t}* [15]. *The corresponding curves can interpolate the given data points automatically, but their shapes cannot be adjusted when the data points and auxiliary points are kept unchanged. Hence, they do not have any degree of freedom in the interpolation curves. The second kind of basis functions are defined over the space {1, t, sint, cost, sin2t, cos2t}* [16]. *Although shapes of these interpolation curves can be adjusted by a control parameter, they are only C^1. If we force them to be C^2, then there is no degree of freedom in the interpolation curves. In order to get interpolation curves with better properties, we change the base space to {1, sint, cost, sin^{2}t, sin^{3}t, cos^{3}t}. Thus, we get CTI-basis functions.*

By simple deduction, the CTI-basis functions defined as Equation (1) have the following properties,

(a) Partition of unity: $f_0(t) + f_1(t) + f_2(t) + f_3(t) \equiv 1$.

(b) Symmetry: $f_i(1-t) = f_{3-i}(t)(i = 0, 1, 2, 3)$.

(c) Properties at the endpoints:

$$\left\{\begin{array}{l} f_0(0) = 0, \quad f_1(0) = 1, \quad f_2(0) = 0, \quad f_3(0) = 0 \\ f_0(1) = 0, \quad f_1(1) = 0, \quad f_2(1) = 1, \quad f_3(1) = 0 \end{array}\right.$$

$$\left\{\begin{array}{l} f_0'(0) = -\frac{\pi}{16}\left(-3\alpha - \beta + 3\right), \quad f_1'(0) = 0, \quad f_2'(0) = \frac{\pi}{16}\left(-3\alpha - \beta + 3\right), \quad f_3'(0) = 0 \\ f_0'(1) = 0, \quad f_1'(1) = -\frac{\pi}{16}\left(-3\alpha - \beta + 3\right), \quad f_2'(1) = 0, \quad f_3'(1) = \frac{\pi}{16}\left(-3\alpha - \beta + 3\right) \end{array}\right.$$

$$\left\{\begin{array}{l} f_0''(0) = \frac{\pi^2}{16}\left(\alpha - \beta + 3\right), \quad f_1''(0) = -\frac{\pi^2}{8}\left(\alpha - \beta + 3\right), \quad f_2''(0) = \frac{\pi^2}{16}\left(\alpha - \beta + 3\right), \quad f_3''(0) = 0 \\ f_0''(1) = 0, \quad f_1''(1) = \frac{\pi^2}{16}\left(\alpha - \beta + 3\right), \quad f_2''(1) = -\frac{\pi^2}{8}\left(\alpha - \beta + 3\right), \quad f_3''(1) = \frac{\pi^2}{16}\left(\alpha - \beta + 3\right) \end{array}\right.$$

3. The CTI-Curves

3.1. Definition and Properties of the CTI-Curves

On the base of the CTI-basis functions, the corresponding curves can be defined as follows.

Definition 2. *Given data points $\boldsymbol{b}_i$ $(i = 0, 1, \cdots, n; n \geqslant 3)$ in $\mathbf{R}^2$ or $\mathbf{R}^3$, the curves*

$$\boldsymbol{p}_i(t) = \sum_{j=0}^{3} f_j(t)\boldsymbol{b}_{i+j} \tag{2}$$

are called cubic trigonometric interpolation curves with parameters α *and* β *(CTI-curves for short), where* $i = 0, 1, \cdots, n-3$, $f_j(t)$ $(j = 0, 1, 2, 3)$ *are the CTI-basis functions defined in Equation* (1).

Theorem 1. *The CTI-curves defined as Equation* (2) *have the following properties,*

(a) *Symmetry: Both* $\boldsymbol{b}_i$ $(i = 0, 1, \cdots, n)$ *and* $\boldsymbol{b}_{n-i}$ $(i = 0, 1, 2, \cdots, n)$ *define the same CTI-curves in a different parameterization for the same shape parameters* α *and* β*, viz.,*

$$\boldsymbol{p}_i(t; \boldsymbol{b}_i, \boldsymbol{b}_{i+1}, \boldsymbol{b}_{i+2}, \boldsymbol{b}_{i+3}) = \boldsymbol{p}_i(1-t; \boldsymbol{b}_{i+3}, \boldsymbol{b}_{i+2}, \boldsymbol{b}_{i+1}, \boldsymbol{b}_i),\ i = 0, 1, \cdots, n-3.$$

(b) *Geometric invariance: Shapes of the CTI-curves are independent of the choice of coordinate system. An affine transformation for the CTI-curves can be performed by carrying out the same affine transformation for the data points.*

(c) C^2 *continuity and automatic interpolation property: For given data points* $\boldsymbol{b}_i$ $(i = 0, 1, \cdots, n)$*, the CTI-curves* $\boldsymbol{p}_i(t)$ $(i = 0, 1, \cdots, n-3)$ *are* C^2 *and automatically interpolate all the given data points expect* $\boldsymbol{b}_0$ *and* $\boldsymbol{b}_n$.

Proof.

(a) From the symmetry of the CTI-basis functions and Equation (2),

$$\begin{aligned}\boldsymbol{p}_i(1-t; \boldsymbol{b}_{i+3}, \boldsymbol{b}_{i+2}, \boldsymbol{b}_{i+1}, \boldsymbol{b}_i) &= f_0(1-t)\boldsymbol{b}_{i+3} + f_1(1-t)\boldsymbol{b}_{i+2} + f_2(1-t)\boldsymbol{b}_{i+1} + f_3(1-t)\boldsymbol{b}_i \\ &= f_3(t)\boldsymbol{b}_{i+3} + f_2(t)\boldsymbol{b}_{i+2} + f_1(t)\boldsymbol{b}_{i+1} + f_0(t)\boldsymbol{b}_i = \boldsymbol{p}_i(t; \boldsymbol{b}_i, \boldsymbol{b}_{i+1}, \boldsymbol{b}_{i+2}, \boldsymbol{b}_{i+3})\end{aligned}$$

(b) Because Equation (2) is an affine combination of the data points, geometric invariance follows.

(c) From the properties at the endpoints of CTI-basis functions and Equation (2),

$$\boldsymbol{p}_i(0) = \boldsymbol{b}_{i+1}, \boldsymbol{p}_i(1) = \boldsymbol{b}_{i+2} \tag{3}$$

$$\boldsymbol{p}_i'(0) = \frac{\pi}{16}(-3\alpha - \beta + 3)(\boldsymbol{b}_{i+2} - \boldsymbol{b}_i),\ \boldsymbol{p}_i'(1) = \frac{\pi}{16}(-3\alpha - \beta + 3)(\boldsymbol{b}_{i+3} - \boldsymbol{b}_{i+1}) \tag{4}$$

$$\boldsymbol{p}_i''(0) = \frac{\pi^2}{16}(\alpha - \beta + 3)(\boldsymbol{b}_i - 2\boldsymbol{b}_{i+1} + \boldsymbol{b}_{i+2}),\ \boldsymbol{p}_i''(1) = \frac{\pi^2}{16}(\alpha - \beta + 3)(\boldsymbol{b}_{i+1} - 2\boldsymbol{b}_{i+2} + \boldsymbol{b}_{i+3}) \tag{5}$$

From Equations (3)–(5), it is follows that

$$\boldsymbol{p}_i^{(k)}(1) = \boldsymbol{p}_{i+1}^{(k)}(0)\ (k = 0, 1, 2) \tag{6}$$

Equation (6) shows that the CTI-curves are C^2. On the other hand, Equation (3) shows that the CTI-curves $\boldsymbol{p}_i(t)$ $(i = 0, 1, \cdots, n-3)$ automatically interpolate all the given data points except $\boldsymbol{b}_0$ and $\boldsymbol{b}_n$.

From **Theorem 1**, if two auxiliary points $\boldsymbol{b}_{-1}$ and $\boldsymbol{b}_{n+1}$ are added to the given data points, the open C^2 CTI-curves $\boldsymbol{p}_i(t)$ $(i = -1, 0, 1, \cdots, n-2)$ interpolating all the data points $\boldsymbol{b}_i$ $(i = 0, 1, \cdots, n)$ would be generated naturally. Generally, $\boldsymbol{b}_{-1}$ and $\boldsymbol{b}_{n+1}$ can be taken as $\boldsymbol{b}_{-1} = 2\boldsymbol{b}_0 - \boldsymbol{b}_1$, $\boldsymbol{b}_{n+1} = 2\boldsymbol{b}_n - \boldsymbol{b}_{n-1}$. If three auxiliary points $\boldsymbol{b}_{-1}$, $\boldsymbol{b}_{n+1}$ and $\boldsymbol{b}_{n+2}$ are taken as $\boldsymbol{b}_{-1} = \boldsymbol{b}_n$, $\boldsymbol{b}_{n+1} = \boldsymbol{b}_0$, $\boldsymbol{b}_{n+2} = \boldsymbol{b}_1$, the closed C^2 CTI-curves $\boldsymbol{p}_i(t)$ $(i = -1, 0, 1, \cdots, n-2)$interpolating all the data points $\boldsymbol{b}_i$ $(i = 0, 1, \cdots, n)$ would be generated naturally.

It is clear that there exist two degrees of freedom in the C^2 CTI-curves even if the data points and auxiliary points are fixed. Different interpolation results can be obtained by altering the parameters α and β of the CTI-curves.

Example 1. Consider the data points $\boldsymbol{b}_0 = (0,0)$, $\boldsymbol{b}_1 = (1,1)$, $\boldsymbol{b}_2 = (2,0)$, $\boldsymbol{b}_3 = (3,1)$, $\boldsymbol{b}_4 = (4,0)$, $\boldsymbol{b}_5 = (5,1)$, $\boldsymbol{b}_6 = (6,0)$, and two auxiliary points $\boldsymbol{b}_{-1} = (-1,-1)$, $\boldsymbol{b}_7 = (7,-1)$. Open C^2 CTI-curves with different values of parameters α and β are shown in Figure 1, where the parameters are taken as $(\alpha, \beta) = (-0.5, 0.5)$ (marked with dotted lines), $(\alpha, \beta) = (0, 0)$ (marked with solid lines), and $(\alpha, \beta) = (0.5, -0.5)$ (marked with dashed lines).

Example 2. Consider the data points $\boldsymbol{b}_0 = (0, 1)$, $\boldsymbol{b}_1 = (1, 2)$, $\boldsymbol{b}_2 = (2, 1)$, $\boldsymbol{b}_3 = (1, 0)$, and three auxiliary points $\boldsymbol{b}_{-1} = (1, 0)$, $\boldsymbol{b}_4 = (0, 1)$, $\boldsymbol{b}_5 = (1, 2)$. Closed C^2 CTI-curves with different values of parameters α and β are shown in Figure 2, where the parameters are taken as $(\alpha, \beta) = (-0.5, 0.5)$ (marked with dotted lines), $(\alpha, \beta) = (0, 0)$ (marked with solid lines), and $(\alpha, \beta) = (0.5, -0.5)$ (marked with dashed lines).

Remark 2. *When using the traditional cubic spline to construct C^2 interpolation curves, the general way is to solve a linear equations system. However, due to the interpolation property and continuity of the CTI-curves, the interpolation curves can be generated naturally without solving an equations system. On the other hand, when the data points and auxiliary points are fixed, the traditional cubic interpolation curves are unique, while the CTI-curves can be adjusted by the parameters α and β.*

Remark 3. *Compared with some similar trigonometric interpolation curves (see in* [13–16]*), the CTI-curves presented in this paper have two outstanding characteristics,*

(a) *Shapes of the CTI-curves can be adjusted by changing the parameters α and β even if data points and auxiliary points are kept unchanged.*
(b) *The CTI-curves are not only C^2 but also have two degrees of freedom in the interpolation curves.*

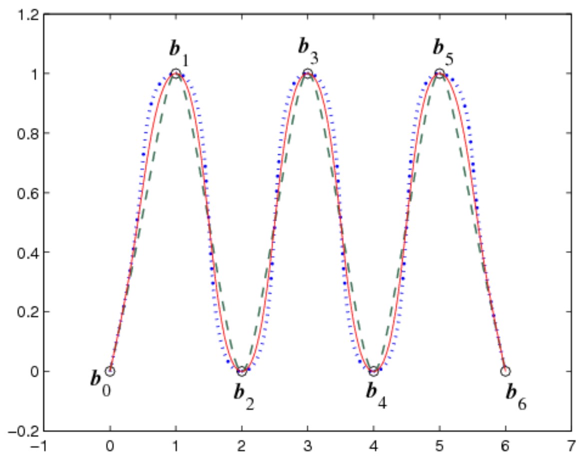

Figure 1. Open C^2 cubic trigonometric interpolation curves (CTI-curves) with different parameters.

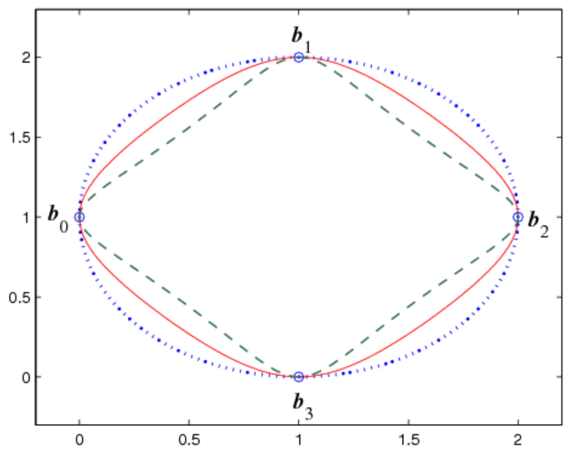

Figure 2. Closed C^2 CTI-curves with different parameters.

3.2. The Optimal Cti-Curves

The CTI-curves have two degrees of freedom. The shapes of the curves are determined by the parameters α and β when the data points and auxiliary points are fixed. Hence, bad interpolation curves would be generated if the parameters α and β are chosen improperly.

Example 3. Consider the data points $\boldsymbol{b}_0 = (0,0)$, $\boldsymbol{b}_1 = (1,0.5)$, $\boldsymbol{b}_2 = (1.5,1)$, $\boldsymbol{b}_3 = (2,2)$, $\boldsymbol{b}_4 = (2.5,2.5)$, $\boldsymbol{b}_5 = (3,2)$, $\boldsymbol{b}_6 = (3.5,1)$, $\boldsymbol{b}_7 = (4,0.5)$, $\boldsymbol{b}_8 = (5,0)$, and two auxiliary points $\boldsymbol{b}_{-1} = (-1,-0.5)$, $\boldsymbol{b}_9 = (6,-0.5)$. Figure 3 shows the CTI-curves with different values of parameters α and β for the same data points and auxiliary points, where the parameters are taken as (a): $(\alpha,\beta) = (-0.1,0.2)$, (b): $(\alpha,\beta) = (3.6,-2.8)$.

It is obvious that the interpolation curves in Figure 3a are more satisfactory than the interpolation curves in Figure 3b. Hence, how to determine proper parameters α and β of the CTI-curves is the key when constructing C^2 interpolation curves when the data points and auxiliary points are fixed. A method for determining the optimal parameters α and β of the CTI-curves is presented as follows.

When the CTI-curves are used to construct C^2 interpolation curves, the interpolation curves are usually required to be smooth. Generally, the smoothness of a curve can be measured by its energy. The lower the energy is, the smoother the curve. According to Reference [17], the energy value of a curve $\boldsymbol{r}(t)$ $(a \leqslant t \leqslant b)$ can be approximately expressed as follows,

$$E_c = \int_a^b (\boldsymbol{r}''(t))^2 \mathrm{d}t \tag{7}$$

From Equation (7), for given data points $\boldsymbol{b}_i$ $(i = 0,1,\cdots,n)$ and auxiliary points $\boldsymbol{b}_{-1}$, $\boldsymbol{b}_{n+1}$, the optimal parameters α and β of the CTI-curves $\boldsymbol{p}_i(t)$ $(i = -1,0,1,\cdots,n-2)$ can be determined by an energy optimization model expressed as follows,

$$\begin{aligned} \min \quad & E_c(\alpha,\beta) = \sum_{i=-1}^{n-2} \int_0^1 \left(\boldsymbol{p}_i''(t)\right)^2 \mathrm{d}t \\ \text{s.t.} \quad & \alpha,\beta \in R \end{aligned} \tag{8}$$

In order to obtain the minimum energy value, there must be

$$\begin{cases} \frac{\partial E_c}{\partial \alpha} = 0 \\ \frac{\partial E_c}{\partial \beta} = 0 \end{cases} \tag{9}$$

Set

$$L_0(t) = \frac{1}{24}\left(-14 + 9S + 24S^2 - 19S^3 + 14C^3\right),\ L_1(t) = \frac{1}{24}\left(2 - 9C - 2S^3 + 7C^3\right),$$

$$L_2(t) = \frac{1}{24}\left(2 - 9S + 7S^3 - 2C^3\right),\ L_3(t) = \frac{1}{24}\left(10 + 9C - 24S^2 + 14S^3 - 19C^3\right),$$

$$M_0(t) = \frac{1}{24}\left(-2 + 3S - S^3 + 2C^3\right),\ M_1(t) = \frac{1}{24}\left(-10 - 3C + 24S^2 - 14S^3 + 13C^3\right),$$

$$M_2(t) = \frac{1}{24}\left(14 - 3S - 24S^2 + 13S^3 - 14C^3\right),\ M_3(t) = \frac{1}{24}\left(-2 + 3C + 2S^3 - C^3\right),$$

$$N_0(t) = \frac{1}{24}\left(6 - 9S + 3S^3 - 6C^3\right),\ N_1(t) = \frac{1}{24}\left(6 + 9C - 6S^3 + 9C^3\right),$$

$$N_2(t) = \frac{1}{24}\left(6 + 9S + 9S^3 - 6C^3\right),\ N_3(t) = \frac{1}{24}\left(6 - 9C - 6S^3 + 3C^3\right),$$

where $S := \sin\frac{\pi t}{2}$, $C := \cos\frac{\pi t}{2}$, $0 \leqslant t \leqslant 1$. Then, Equation (2) can be rewritten as follows,

$$\boldsymbol{p}_i(t) = \boldsymbol{G}_i(t)\alpha + \boldsymbol{H}_i(t)\beta + \boldsymbol{I}_i(t) \tag{10}$$

where

$$\boldsymbol{G}_i(t) = L_0(t)\boldsymbol{b}_i + L_1(t)\boldsymbol{b}_{i+1} + L_2(t)\boldsymbol{b}_{i+2} + L_3(t)\boldsymbol{b}_{i+3},$$
$$\boldsymbol{H}_i(t) = M_0(t)\boldsymbol{b}_i + M_1(t)\boldsymbol{b}_{i+1} + M_2(t)\boldsymbol{b}_{i+2} + M_3(t)\boldsymbol{b}_{i+3},$$
$$\boldsymbol{I}_i(t) = N_0(t)\boldsymbol{b}_i + N_2(t)\boldsymbol{b}_{i+1} + N_3(t)\boldsymbol{b}_{i+2} + N_4(t)\boldsymbol{b}_{i+3}.$$

From Equation (10), Equation (7) can be expressed as follows,

$$E_c = C_1\alpha^2 + C_2\beta^2 + 2C_3\alpha\beta + 2C_4\alpha + 2C_5\beta + C_6 \tag{11}$$

where

$$C_1 = \sum_{i=0}^{n-3}\int_0^1 \left(\boldsymbol{G}_i''(t)\right)^2 \mathrm{d}t,\ C_2 = \sum_{i=0}^{n-3}\int_0^1 \left(\boldsymbol{H}_i''(t)\right)^2 \mathrm{d}t,\ C_3 = \sum_{i=0}^{n-3}\int_0^1 \left(\boldsymbol{G}_i''(t)\cdot\boldsymbol{H}_i''(t)\right)\mathrm{d}t,$$
$$C_4 = \sum_{i=0}^{n-3}\int_0^1 \left(\boldsymbol{G}_i''(t)\cdot\boldsymbol{I}_i''(t)\right)\mathrm{d}t,\ C_5 = \sum_{i=0}^{n-3}\int_0^1 \left(\boldsymbol{H}_i''(t)\cdot\boldsymbol{I}_i''(t)\right)\mathrm{d}t,\ C_6 = \sum_{i=0}^{n-3}\int_0^1 \left(\boldsymbol{I}_i''(t)\right)^2 \mathrm{d}t.$$

By Equation (11), Equation (9) can be rewritten as follows,

$$\begin{cases} C_1\alpha + C_3\beta + C_4 = 0 \\ C_3\alpha + C_2\beta + C_5 = 0 \end{cases} \tag{12}$$

When $C_1C_2 - C_3^2 \neq 0$, from Equation (12), then

$$\begin{cases} \alpha = \frac{C_3C_5 - C_2C_4}{C_1C_2 - C_3^2} \\ \beta = \frac{C_3C_4 - C_1C_5}{C_1C_2 - C_3^2} \end{cases} \tag{13}$$

Remark 4. *If $C_1C_2 - C_3^2 = 0$, there would be no unique solution to Equation* (13). *At this point, the two auxiliary points could be adjusted properly in order to ensure that $C_1C_2 - C_3^2 \neq 0$ holds.*

After the optimal parameters $\alpha = \tilde{\alpha}$ and $\beta = \tilde{\beta}$ are determined by Equation (13), the optimal C^2 CTI-curves $\tilde{\boldsymbol{p}}_i(t)$ $(i = -1, 0, 1, \cdots, n-2)$ interpolating all the given data points $\boldsymbol{b}_i$ $(i = 0, 1, \cdots, n)$ can be obtained.

Example 4. For the same data points and auxiliary points in Example 3, the optimal parameters of the CTI-curves, determined by Equation (13), are $\tilde{\alpha} = -0.0443$ and $\tilde{\beta} = 0.4836$. The optimal C^2 CTI-curves (solid lines) and the interpolation curves constructed by the classical cubic B-spline curves (dashed lines) are shown in Figure 4, where the tangent vectors at the endpoints of the classical cubic B-spline curves are taken as $3(\boldsymbol{b}_0 - \boldsymbol{b}_{-1})$ and $3(\boldsymbol{b}_9 - \boldsymbol{b}_8)$.

For comparing the CTI-curves with the classical cubic B-spline curves to construct interpolation curves, the energy values and time-consuming of the two methods are shown in Table 1.

Table 1. The energy values and time-consuming of the two methods.

Method	Energy Value	Time-Consuming (s)
CTI-curves	4.8584	9.6
Classical cubic B-spline curves	17.6137	16.3

Table 1 shows that the interpolation curves constructed by the CTI-curves are smoother and faster to compute than the classical cubic B-spline curves.

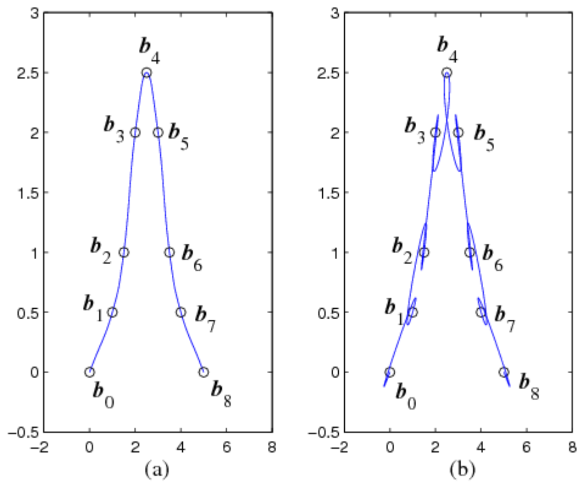

Figure 3. Effects of the parameters on CTI-curves. **(a)** $(\alpha, \beta) = (-0.1, 0.2)$ **(b)** $(\alpha, \beta) = (3.6, -2.8)$.

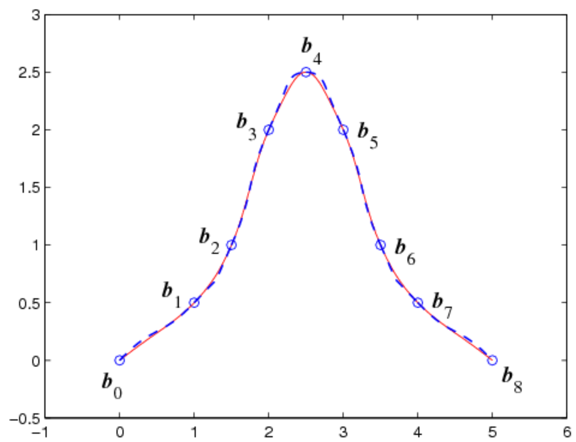

Figure 4. The optimal C^2 CTI-curves.

4. The CTI-Surfaces

Using tensor products, the corresponding CTI-surfaces can be defined as follows.

Definition 3. *Given data points* $\boldsymbol{b}_{k,l}$ $(k = 0, 1, \cdots, m; l = 0, 1, \cdots, n)$ *in* $\mathbf{R}^3$, *the piecewise surfaces*

$$\boldsymbol{p}_{i,j}(u, v) = \sum_{k=0}^{3} \sum_{l=0}^{3} f_k(u) f_l(v) \boldsymbol{b}_{i+k,j+l} \tag{14}$$

are called cubic trigonometric interpolation surfaces with parameters α_1, β_1, α_2 *and* β_2 *(CTI-surfaces for short), where* $i = 0, 1, \cdots, m-3$, $j = 0, 1, \cdots, n-3$, $f_k(u) := f_k(u; \alpha_1, \beta_1)$ *and* $f_l(v) := f_l(v; \alpha_2, \beta_2)$ $(i = 0, 1, 2, 3)$ *are the CTI-basis functions defined according to Equation* (1).

It is not difficult to show that the CTI-surfaces have properties similar to the CTI-curves, which include the following important property.

Theorem 2. *Given data points* $\boldsymbol{b}_{k,l}$ $(k = 0, 1, \cdots, m; l = 0, 1, \cdots, n)$, *the CTI-surfaces* $\boldsymbol{p}_{i,j}(u, v)$ $(i = 0, 1, \cdots, m-3;\ j = 0, 1, \cdots, n-3)$ *automatically interpolate all the given data points except* $\boldsymbol{b}_{0,l}$, $\boldsymbol{b}_{m,l}$ $(l = 0, 1, \cdots, n)$ *and* $b_{k,0}$, $b_{k,n}$ $(k = 0, 1, \cdots, m)$ *and are* C^2.

Proof. From the properties at the endpoints of the CTI-basis functions and Equation (14),

$$\begin{cases} \boldsymbol{p}_{i,j}(0,0) = \boldsymbol{b}_{i+1,j+1} \\ \boldsymbol{p}_{i,j}(0,1) = \boldsymbol{b}_{i+1,j+2} \\ \boldsymbol{p}_{i,j}(1,0) = \boldsymbol{b}_{i+2,j+1} \\ \boldsymbol{p}_{i,j}(1,1) = \boldsymbol{b}_{i+2,j+2} \end{cases} \tag{15}$$

$$\begin{cases} \partial_u \boldsymbol{p}_{i,j}(0,v) = \frac{\pi}{16}\sum_{l=0}^{3} f_l(v)(\boldsymbol{b}_{i+2,j+l} - \boldsymbol{b}_{i,j+l}) \\ \partial_u \boldsymbol{p}_{i,j}(1,v) = \frac{\pi}{16}\sum_{l=0}^{3} f_l(v)(\boldsymbol{b}_{i+3,j+l} - \boldsymbol{b}_{i+1,j+l}) \\ \partial_u \boldsymbol{p}_{i,j}(u,0) = \sum_{k=0}^{3} f'_k(u)\boldsymbol{b}_{i+k,j+1} \\ \partial_u \boldsymbol{p}_{i,j}(u,1) = \sum_{k=0}^{3} f'_k(u)\boldsymbol{b}_{i+k,j+2} \end{cases} \tag{16}$$

$$\begin{cases} \partial_v \boldsymbol{p}_{i,j}(0,v) = \sum_{l=0}^{3} f'_l(v)\boldsymbol{b}_{i+1,j+l} \\ \partial_v \boldsymbol{p}_{i,j}(1,v) = \sum_{l=0}^{3} f'_l(v)\boldsymbol{b}_{i+2,j+l} \\ \partial_v \boldsymbol{p}_{i,j}(u,0) = \frac{\pi}{16}\sum_{k=0}^{3} f_k(u)(\boldsymbol{b}_{i+k,j+2} - \boldsymbol{b}_{i+k,j}) \\ \partial_v \boldsymbol{p}_{i,j}(u,1) = \frac{\pi}{16}\sum_{k=0}^{3} f_k(u)(\boldsymbol{b}_{i+k,j+3} - \boldsymbol{b}_{i+k,j+1}) \end{cases} \tag{17}$$

$$\begin{cases} \partial_{uv} \boldsymbol{p}_{i,j}(0,v) = \frac{\pi}{16}\sum_{l=0}^{3} f'_l(v)(\boldsymbol{b}_{i+2,j+l} - \boldsymbol{b}_{i,j+l}) \\ \partial_{uv} \boldsymbol{p}_{i,j}(1,v) = \frac{\pi}{16}\sum_{l=0}^{3} f'_l(v)(\boldsymbol{b}_{i+3,j+l} - \boldsymbol{b}_{i+1,j+l}) \\ \partial_{uv} \boldsymbol{p}_{i,j}(u,0) = \frac{\pi}{16}\sum_{k=0}^{3} f'_k(u)(\boldsymbol{b}_{i+k,j+2} - \boldsymbol{b}_{i+k,j}) \\ \partial_{uv} \boldsymbol{p}_{i,j}(u,1) = \frac{\pi}{16}\sum_{k=0}^{3} f'_k(u)(\boldsymbol{b}_{i+k,j+3} - \boldsymbol{b}_{i+k,j+1}) \end{cases} \tag{18}$$

$$\begin{cases} \partial_{uu} \boldsymbol{p}_{i,j}(0,v) = \frac{\pi^2}{16}\sum_{l=0}^{3} f_l(v)(\boldsymbol{b}_{i,j+l} - 2\boldsymbol{b}_{i+1,j+l} + \boldsymbol{b}_{i+2,j+l}) \\ \partial_{uu} \boldsymbol{p}_{i,j}(1,v) = \frac{\pi^2}{16}\sum_{l=0}^{3} f_l(v)(\boldsymbol{b}_{i+1,j+l} - 2\boldsymbol{b}_{i+2,j+l} + \boldsymbol{b}_{i+3,j+l}) \\ \partial_{uu} \boldsymbol{p}_{i,j}(u,0) = \sum_{k=0}^{3} f''_k(u)\boldsymbol{b}_{i+k,j+1} \\ \partial_{uu} \boldsymbol{p}_{i,j}(u,1) = \sum_{k=0}^{3} f''_k(u)\boldsymbol{b}_{i+k,j+2} \end{cases} \tag{19}$$

$$\begin{cases} \partial_{vv} \boldsymbol{p}_{i,j}(0,v) = \sum_{l=0}^{3} f''_l(v)\boldsymbol{b}_{i+1,j+l} \\ \partial_{vv} \boldsymbol{p}_{i,j}(1,v) = \sum_{l=0}^{3} f''_l(v)\boldsymbol{b}_{i+2,j+l} \\ \partial_{vv} \boldsymbol{p}_{i,j}(u,0) = \frac{\pi^2}{16}\sum_{k=0}^{3} f_k(u)(\boldsymbol{b}_{i+k,j} - 2\boldsymbol{b}_{i+k,j+1} + \boldsymbol{b}_{i+k,j+2}) \\ \partial_{vv} \boldsymbol{p}_{i,j}(u,1) = \frac{\pi^2}{16}\sum_{k=0}^{3} f_k(u)(\boldsymbol{b}_{i+k,j+1} - 2\boldsymbol{b}_{i+k,j+2} + \boldsymbol{b}_{i+k,j+3}) \end{cases} \tag{20}$$

Equation (15) shows that the CTI-surfaces $\boldsymbol{p}_{i,j}(u,v)$ $(i = 0,1,\cdots,m-3;\ j = 0,1,\cdots,n-3)$ automatically interpolate all the given data points except $\boldsymbol{b}_{0,l}$, $\boldsymbol{b}_{m,l}$ $(l = 0,1,\cdots,n)$ and $\boldsymbol{b}_{k,0}$, $\boldsymbol{b}_{k,n}$ $(k = 0,1,\cdots,m)$. In addition, the following results can also be obtained from Equation (15),

$$\begin{cases} \boldsymbol{p}_{i,j}(1,0) = \boldsymbol{p}_{i+1,j}(0,0) \\ \boldsymbol{p}_{i,j}(1,1) = \boldsymbol{p}_{i+1,j}(0,1) \\ \boldsymbol{p}_{i,j}(0,1) = \boldsymbol{p}_{i,j+1}(0,0) \\ \boldsymbol{p}_{i,j}(1,1) = \boldsymbol{p}_{i,j+1}(1,0) \end{cases} \tag{21}$$

From Equations (16) and (17),

$$\begin{cases} \partial_u \boldsymbol{p}_{i,j}(1,v) = \partial_u \boldsymbol{p}_{i+1,j}(0,v) \\ \partial_u \boldsymbol{p}_{i,j}(u,1) = \partial_u \boldsymbol{p}_{i,j+1}(u,0) \\ \partial_v \boldsymbol{p}_{i,j}(1,v) = \partial_v \boldsymbol{p}_{i+1,j}(0,v) \\ \partial_v \boldsymbol{p}_{i,j}(u,1) = \partial_v \boldsymbol{p}_{i,j+1}(u,0) \end{cases} \tag{22}$$

From Equations (18)–(20),

$$\begin{cases} \partial_{uv} \boldsymbol{p}_{i,j}(1,v) = \partial_{uv} \boldsymbol{p}_{i+1,j}(0,v) \\ \partial_{uv} \boldsymbol{p}_{i,j}(u,1) = \partial_{uv} \boldsymbol{p}_{i,j+1}(u,0) \\ \partial_{uu} \boldsymbol{p}_{i,j}(1,v) = \partial_{uu} \boldsymbol{p}_{i+1,j}(0,v) \\ \partial_{uu} \boldsymbol{p}_{i,j}(u,1) = \partial_{uu} \boldsymbol{p}_{i,j+1}(u,0) \\ \partial_{vv} \boldsymbol{p}_{i,j}(1,v) = \partial_{vv} \boldsymbol{p}_{i+1,j}(0,v) \\ \partial_{vv} \boldsymbol{p}_{i,j}(u,1) = \partial_{vv} \boldsymbol{p}_{i,j+1}(u,0) \end{cases} \tag{23}$$

Equations (21)–(23) show that the CTI-surfaces $\boldsymbol{p}_{i,j}(u,v)$ $(i = 0,1,\cdots,m-3;\ j = 0,1,\cdots,n-3)$ are C^2.

According to Theorem 2, for given data points $\boldsymbol{b}_{k,l}$ $(k = 0,1,\cdots,m; l = 0,1,\cdots,n)$, the CTI-surfaces automatically interpolate all the given data points except $\boldsymbol{b}_{0,l}$, $\boldsymbol{b}_{m,l}$ $(l = 0,1,\cdots,n)$ and $\boldsymbol{b}_{k,0}$, $\boldsymbol{b}_{k,n}$ $(k = 0,1,\cdots,m)$. If auxiliary points $\boldsymbol{b}_{-1,j}$, $\boldsymbol{b}_{m+1,j}$ $(j = 0,1,\cdots,n)$ and $\boldsymbol{b}_{i,-1}$, $\boldsymbol{b}_{i,n+1}$ $(i = 0,1,\cdots,m)$ are added to the given data points, the C^2 CTI-surfaces interpolating all the data points $\boldsymbol{b}_{k,l}$ $(k = 0,1,\cdots,m; l = 0,1,\cdots,n)$ would be generated naturally. Generally, the auxiliary points can be taken as follows,

$$\begin{cases} \boldsymbol{b}_{-1,-1} = 2\boldsymbol{b}_{0,0} - \boldsymbol{b}_{1,1}, \quad \boldsymbol{b}_{-1,n+1} = 2\boldsymbol{b}_{0,n} - \boldsymbol{b}_{1,n-1} \\ \boldsymbol{b}_{m+1,-1} = 2\boldsymbol{b}_{m,0} - \boldsymbol{b}_{m-1,1}, \boldsymbol{b}_{m+1,n+1} = 2\boldsymbol{b}_{m,n} - \boldsymbol{b}_{m-1,n-1} \\ \boldsymbol{b}_{-1,j} = 2\boldsymbol{b}_{0,j} - \boldsymbol{b}_{1,j}, \quad \boldsymbol{b}_{m+1,j} = 2\boldsymbol{b}_{m,j} - \boldsymbol{b}_{m-1,j}, \quad j = 0,1,\cdots,n \\ \boldsymbol{b}_{i,-1} = 2\boldsymbol{b}_{i,0} - \boldsymbol{b}_{i,1}, \quad \boldsymbol{b}_{i,n+1} = 2\boldsymbol{b}_{i,n} - \boldsymbol{b}_{i,n-1}, \quad i = 0,1,\cdots,m \end{cases} \tag{24}$$

It is clear that there exist four degrees of freedom in the C^2 CTI-surfaces, even if the data points and auxiliary points are fixed. Different interpolation results can be obtained by altering the parameters α_i and β_i $(i = 1,2)$ of the CTI-surfaces. Figure 5 shows the C^2 CTI-surfaces with different values of parameters α_i and β_i $(i = 1,2)$, where the data points $\boldsymbol{b}_{k,l}$ $(k = 0,1,2,3; l = 0,1,2,3)$ are fixed and the auxiliary points are added according to Equation (24).

Similar to the optimal CTI-curves, the optimal CTI-surfaces can also be determined by an energy optimization model. According to Ref. [17], the energy value of a surface $\boldsymbol{r}(u,v)$ $(a \leqslant u \leqslant b, c \leqslant v \leqslant d)$ can be approximately expressed as follows,

$$E_s = \int_a^b \int_c^d \left((\boldsymbol{r}_{uu}(u,v))^2 + 2\,(\boldsymbol{r}_{uv}(u,v))^2 + (\boldsymbol{r}_{vv}(u,v))^2 \right) \mathrm{d}u\mathrm{d}v \tag{25}$$

From Equation (25), for given data points $\boldsymbol{b}_{k,l}$ $(k = 0, 1, \cdots, m; l = 0, 1, \cdots, n)$ and auxiliary points $\boldsymbol{b}_{-1,j}$, $\boldsymbol{b}_{m+1,j}$ $(j = 0, 1, \cdots, n)$, $\boldsymbol{b}_{i,-1}$, $\boldsymbol{b}_{i,n+1}$ $(i = 0, 1, \cdots, m)$, the optimal parameters α_i and β_i $(i = 1, 2)$ of the CTI-surfaces $\boldsymbol{p}_{i,j}(u, v)$ $(i = -1, 0, 1, \cdots, m-2; j = -1, 0, 1, \cdots, n-2)$ can be determined by an energy optimization model expressed as follows,

$$\begin{aligned} \min \quad & E_s(\alpha_1, \beta_1, \alpha_2, \beta_2) = \sum_{i=-1}^{m-2} \sum_{j=-1}^{n-2} \int_0^1 \int_0^1 \left(\left(\frac{\partial^2 \boldsymbol{p}_{i,j}(u,v)}{\partial u^2} \right)^2 + 2 \left(\frac{\partial^2 \boldsymbol{p}_{i,j}(u,v)}{\partial u \partial v} \right)^2 + \left(\frac{\partial^2 \boldsymbol{p}_{i,j}(u,v)}{\partial v^2} \right)^2 \right) \mathrm{d}u \mathrm{d}v \\ \text{s.t.} \quad & \alpha_1, \beta_1, \alpha_2, \beta_2 \in R \end{aligned} \tag{26}$$

The particle swarm optimization (PSO) algorithm [18] can be used to solve Equation (26). After the optimal parameters $\alpha_i = \tilde{\alpha}_i$ and $\beta_i = \tilde{\beta}_i$ $(i = 1, 2)$ are determined, the optimal C^2 CTI-surfaces $\tilde{\boldsymbol{p}}_{i,j}(u, v)$ $(i = -1, 0, 1, \cdots, m-2; j = -1, 0, 1, \cdots, n-2)$ interpolating all the given data points $\boldsymbol{b}_{k,l}$ $(k = 0, 1, \cdots, m; l = 0, 1, \cdots, n)$ can be obtained.

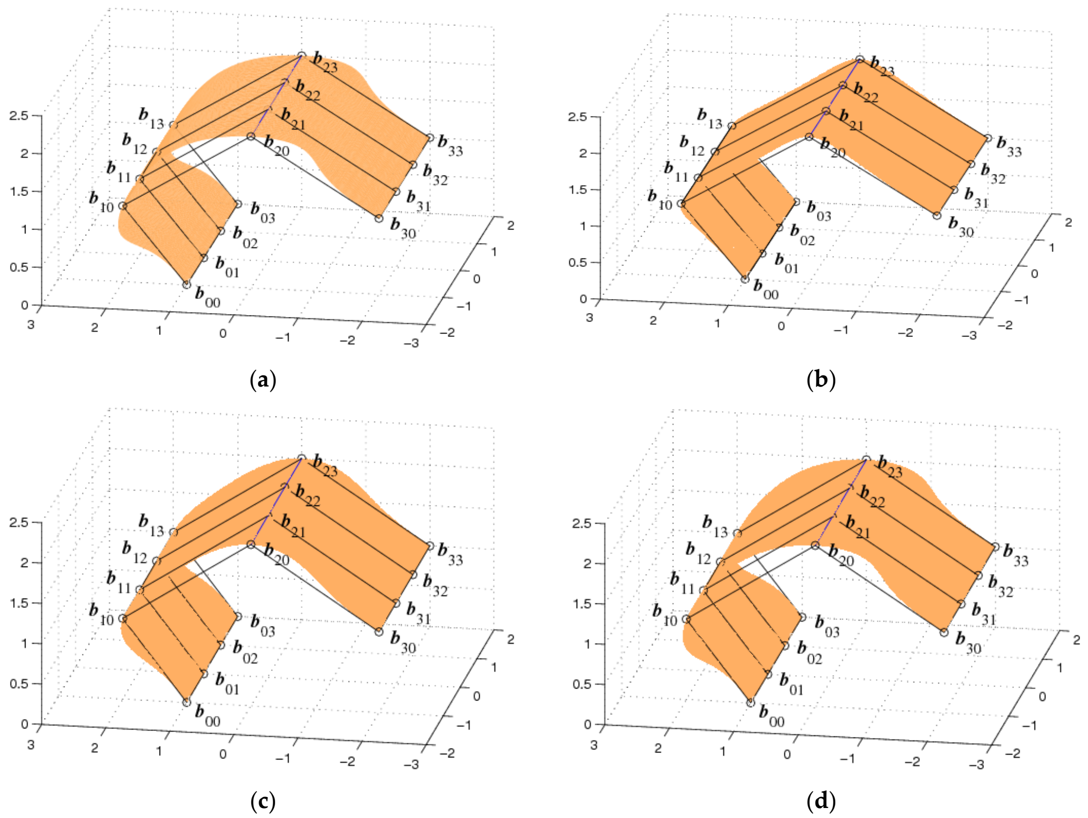

Figure 5. C^2 CTI-surfaces with different parameters. **(a)** $(\alpha_1, \beta_1, \alpha_2, \beta_2) = (-0.5, 0.5, 0.5, -0.5)$; **(b)** $(\alpha_1, \beta_1, \alpha_2, \beta_2) = (0.5, -0.5, -0.5, 0.5)$; **(c)** $(\alpha_1, \beta_1, \alpha_2, \beta_2) = (0.1, -0.5, 0.4, -0.3)$; **(d)** $(\alpha_1, \beta_1, \alpha_2, \beta_2) = (-0.3, 0, 0, 0.4)$.

5. Conclusions

The CTI-curves presented in this paper can not only automatically interpolate the given data points without solving equation systems, but are also C^2 and have two degrees of freedom. The CTI-surfaces also have characteristics similar to the CTI-curves. Therefore, the CTI-curves/surfaces presented in this paper provide a simple and efficient way for constructing interpolation curves and surfaces.

For practical applications of the proposed interpolation curves and surfaces in geometric modeling, it is clear that some special algorithms need to be established. Furthermore, the proposed interpolation curves and surfaces allow only global adjustment. Hence, local adjustment of the proposed interpolation curves and surfaces also need to be investigated. Some interesting results in this area will be presented in a follow-up study.

Acknowledgments: This work was supported by the National Natural Science Foundation of China under the grand number 61472135 and the Scientific Research Fund of Hunan Provincial Education Department of China under the grant number 14B099. The authors are also very grateful to the Hunan provincial key construction discipline "Computer Application Technology" of Hunan University of Humanities, Science and Technology of China.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Hoschek, J.; Lasser, D. *Fundamentals of Computer Aided Geometric Design*; Schumaker, L.L., Translator; A.K. Peters, Ltd.: Natick, MA, USA, 1993.
2. Han, X.L. Cubic trigonometric polynomial curves with a shape parameter. *Comput. Aided Geom. Des.* **2004**, *21*, 535–548. [CrossRef]
3. Zhang, J.W.; Krause, F.L. Extending cubic uniform B-splines by unified trigonometric and hyperbolic basis. *Gr. Model.* **2005**, *67*, 100–119. [CrossRef]
4. Mainar, E. A general class of Bernstein-like bases. *Comput. Math. Appl.* **2007**, *53*, 1686–1703. [CrossRef]
5. Wang, G.Z.; Fang, M.E. Unified and extended form of three types of splines. *J. Comput. Appl. Math.* **2008**, *216*, 498–508. [CrossRef]
6. Han, X.A.; Ma, Y.C.; Huang, X.L. The cubic trigonometric Bézier curve with two shape parameters. *Appl. Math. Lett.* **2009**, *22*, 226–231. [CrossRef]
7. Han, X.L.; Zhu, Y.P. Curve construction based on five trigonometric blending functions. *BIT Numer. Math.* **2012**, *52*, 953–979. [CrossRef]
8. Bashir, U.; Abbsa, M.; Ali, J.M. The G^2 and C^2 rational quadratic trigonometric Bézier curve with two shape parameters with applications. *Appl. Math. Comput.* **2013**, *219*, 10183–10197. [CrossRef]
9. Bashir, U.; Ali, J.M. Rational cubic trigonometrc Bézier curve with two shape parameters. *Comput. Appl. Math.* **2016**, *35*, 285–300. [CrossRef]
10. Han, X.L. Normalized B-basis of the space of trigonometric polynomials and curve design. *Appl. Math. Comput.* **2015**, *251*, 336–348. [CrossRef]
11. Yan, L.L. An algebraic-trigonometric blended piecewise curve. *J. Infor. Comput. Sci.* **2015**, *12*, 6491–6501.
12. Yan, L.L. Cubic trigonometric nonuniform spline curves and surfaces. *Math. Probl. Eng.* **2016**, *2016*, 1–9.
13. Su, B.Y.; Tan, J.Q. A family of quasi-cubic blended splines and applications. *J. Zhejiang Univ. Sci. A* **2006**, *7*, 1550–1560. [CrossRef]
14. Yan, L.L.; Liang, J.F. A class of algebraic-trigonometric blended splines. *J. Comput. Appl. Math.* **2011**, *235*, 1713–1729. [CrossRef]
15. Li, J.C.; Zhao, D.B.; Yang, L. Quasi-cubic trigonometric spline interpolation curves and surfaces. *J. Chin. Comput. Syst.* **2013**, *34*, 680–684. (In Chinese)
16. Yang, L.; Li, J.C.; Kuang, X.L. A family of local adjustable cubic algebraic-trigonometric interpolation spline. *Comput. Eng. Sci.* **2013**, *35*, 130–135. (In Chinese)
17. Poliakoff, J.F. An improved algorithm for automatic fairing of non-uniform parametric cubic splines. *Comput. Aided Des.* **1996**, *28*, 59–66. [CrossRef]
18. Eberhart, R.C.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.