Blending Basic Shapes By C-Type Splines and Subdivision Scheme

Xiang Li[1], Mei-E Fang[1, 2, *] and Qian Qi[2]

Abstract: In this article, we adopt the C-type spline of degree 2 to model and blend basic shapes including conics and circle arcs. The C-type spline belongs to the ωB-spline category of splines that are capable of blending polynomial, trigonometric and hyperbolic functions. Commonly used basic shapes can be exactly represented by these types of splines. We derive explicit formulas for the convenience of modeling the basic curves. The entire blending curve is C^1-continuous. In comparison with the existing best blending method by rational G^2 splines, which are rational splines of degree 3, the proposed method allows simpler representation and blending of the basic curves, and it can represent numerous basic shapes including the hyperbolic types. We also design a subdivision method to generate blending curves; this method is precise for the basic curves and approximate for the blending sections. The subdivision process is efficient for modeling and rendering. It has also proven to be C^1-continuous by the asymptotically equivalent theory and the continuity of stationary subdivision method. In addition, we extend the proposed methods to cases involving the modeling and blending of basic surfaces. We provide many examples that illustrate the merits of our methods.

Keywords: Basic shapes, blending, C-type splines, subdivision, C^1-continuous.

1 Introduction

The modeling and blending problems of basic shapes are classical problems in computer-aided geometry design (CAGD) and very important in many other application domains [Karciauskas and Peters (2011); Shen, Hu and Fan (2017); Wang, Lin and Fang (2017)]. Many types of methods have been continuously proposed in the past decade [Shen, Hu and Fan (2017); Hui (1999); Erich (2001); Kouibia, Pasadas, Sbibihb et al. (2013); Kiciak (2013); Lin, Xiong and Liao (2014)]. The Rational G^2 spline proposed in Karciauskas et al. [Karciauskas and Peters (2011)] allows the modeling of multiple basic shapes, including segments of conics and circle arcs in one structure; This is important in conceptual design to avoid the posteriori need to stitch the conceptual pieces together. This method was also extended to cases that require blending of the basic surfaces for more background information [Karciauskas and Peters (2011)]. The traditional

[1] School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, China.

[2] School of Computer, Hangzhou Dianzi University, Hangzhou, China.

[*] Corresponding Author: Mei-E Fang. Email: fme@gzhu.edu.cn.

CMES. doi:10.32604/cmes.2019.05659

approaches of representing and blending basic curves include modeling them as projections of C^2-continuous curves in a homogeneous space and of individually representing each piece of the basic curve by a rational Bernstein-Bezier form [Boehm (1987); Bangert and Prautzsch (1997); Farin (2006)]. The first approach leads to a high degree of parameterizations and the second approach lacks smooth transitions between the two adjacent pieces of the basic curves. Actually, the structure of rational G^2 splines is similar to the structure of cubic rational B-splines, which can reproduce basic curves and implement G^2-continuity between blended basic shapes. However, the weight coefficients for B-splines need to be computed by very complicated formulas. In addition, sampling on basic curves before blending them is also troublesome and subjective. In these blending methods, technologies relative to splines, parameterizations and subdivisions are often used; please refer to Salomon et al. [Salomon (2006); Floater and Hormann (2005); Andersson and Stewart (2010); Ma (2005)] to acquire more knowledge of these methods. Actually, the ωB-splines proposed in Fang et al. [Fang and Wang (2008)] can easily be used to construct many types of basic curves, which are B-like-splines spanned over the blending space of polynomial space, trigonometric space and hyperbolic space. The control points and weight coefficients need to be computed by very complicated formulas. In addition, sampling on basic curves before blending them is also troublesome and subjective. To blend basic shapes, we adopt the C-type ωB-splines of degree 2 (C-type splines for short), which are C^1 and able to exactly represent basic shapes. In this paper, we derive their representation formula and develop a blending method using C-type splines. A C^1 subdivision scheme of blending basic curves is also proposed. The subdivision scheme can generate exact basic curves and approximate blending segments, which converge to the corresponding C-type spline curves. In comparison with the blending method by rational G^2 splines, which are rational splines of degree 3, the proposed method makes it simpler to represent and blend basic curves.

This article is organized as follows. In Section 2, C-type spline bases and curves are defined. Section 3 describes the detailed algorithms for representing and blending basic shapes, including curves and surfaces. Section 4 proposes a subdivision scheme. Section 5 concludes the paper.

2 C-type spline curve of degree 2

The ωB-splines proposed in Fang et al. [Fang and Wang (2008)] used a frequency sequence to determine their types and corresponding coefficients. Therefore, blending different types of curves is permitted when different frequencies are included in the frequency sequence. Moreover, the ωB-splines space is blended by polynomial, trigonometric and hyperbolic spaces, thus many analytical curves are exactly reproduced by ωB-splines. For conics and circular arcs, C-type splines of degree 2 are sufficient, and the lower degree is beneficial for modeling suitable blending segments. Therefore, we select C-type splines of degree 2 to represent and blend basic curves. The definitions for the C-type spline basic functions and curves of degree 2 are given below.

Definition 2.1 (C-type spline bases) Let T be a given knot sequence $\{t_i\}_{-\infty}^{+\infty}$ with $t_i \le t_{i+1}$, $\omega = \{\omega_i\}_{-\infty}^{+\infty}$ be a given frequency sequence. $C_{i,2}(t)$ constructed by the following formulas are called C-type spline bases in the span of $\{\cos\omega_i t, \sin\omega_i t, 1, t, t^2\}$ for $t \in [t_i, t_{i+3}]$.

$$C_{i,2}(t) = \int_{-\infty}^{t} (\delta_i C_{i,1}(s) - \delta_{i+1} C_{i+1,1}(s)) ds \tag{2.1}$$

where

$$C_{i,2}(t) = \begin{cases} \dfrac{\int_{t_i}^{t} \cos\omega_i (s - t_i) ds}{\int_{t_{i+1}}^{t_i} \cos\omega_i (s - t_i) ds} & t_i \le t < t_{i+1} \\ \dfrac{\int_{t_{i+1}}^{t_{i+2}+t_{i+1}-t} \cos\omega_{i+1} (s - t_{i+1}) ds}{\int_{t_{i+1}}^{t_{i+2}} \cos\omega_{i+1} (s - t_{i+1}) ds} & t_{i+1} \le t < t_{i+2} \\ 0 & otherwise \end{cases} \tag{2.2}$$

$$\delta_i = \left(\int_{-\infty}^{+\infty} C_{i,1}(t) dt \right)^{-1}, i = 0, \pm 1, \cdots.$$

As for *ω*, we set $\omega_i = \alpha_i \beta_i$, where $\alpha_i \in \Re^+, \beta_i \in \{0, 1, i\}$, where *i* denotes the unit of imaginary numbers and $\alpha_i (t_{i+1} - t_i) \le \pi$ when $\beta_i = 1$.Therefore, ω_i can be zero (when $\beta_i = 0$), a positive real number (when $\beta_i = 1$) or a pure imaginary number (when $\beta_i = i$). β_i determines the type of basis function on its corresponding interval and affects the shape together with the value of α_i.

Definition 2.2 Let $\{P_i\}_{i=0}^{n} \in \Re^3, C_{i,2}(t)$ be C-type spline bases corresponding to the partition $T = \{t_j\}_{j=-\infty}^{+\infty}$ of the parameter axis t. Then $P(t) = \sum_{i=0}^{n} C_{i,2}(t) P_i$, where $t_2 \le t \le t_{n+1}, n \ge 2$, is called a C-type spline curve corresponding to the knot vector *T*. $P_i (i = 0, 1, \cdots, n)$ are control points.

Because different frequencies can be set for different knot intervals, C-type splines can represent numerous types of curves. Corresponding to the value of *β*, 0, 1 or *i*, C-type spline curves are polynomial, trigonometric polynomial and hyperbolic polynomial types respectively. For the basic surfaces, they can be modeled by revolving around the basic curves. For more details please refer to Fang et al. [Fang and Wang (2008)].

3 Blending basic shapes with C-type splines of degree 2

C-type splines span the space $\{\sin\omega_i t, \cos\omega_i t, 1, t, t^2\}$, which can exactly reproduce basic curves. Their representation structure is similar to polynomial B-splines, without rational forms. According to the standard equations and the corresponding parameters of the basic curves, we can use C-type splines to derive the explicit formula for computing control points that are necessary for representing basic curves. Corollary 3.1, 3.2, and 3.3 show these formulas for parabolas, hyperbolas and Lissajous curves respectively. Please note that ellipses are special cases of Lissajous curves. We can see that these formulas are very simple and regular.

Corollary 3.1 The sufficient conditions that a piece of parabola $(x(t), y(t)) = (t, at^2 + bt + c), t \in [t_1, t_2)$ can be exactly reproduced by C-type curve $P(t) = \sum_{i=0}^{2} C_{i,2}(t)P_i$.

1. In $C_{i,2}(t), \beta_i \equiv 0, \alpha_i \equiv 1$;
2. $\ell = t_2 - t_1$;
3. Control points $P_i(x_i, y_i)(i = 0,1,2)$ satisfy

$$\begin{cases} x_i = \left(i - \dfrac{1}{2}\right)l, \\ y_i = a \cdot i(i-1) \cdot \ell^2 + b \cdot \left(i - \dfrac{1}{2}\right) \cdot l + c. \end{cases}$$

Corollary 3.2 The sufficient conditions that a piece of hyperbola $(x(t), y(t)) = (a\cosh(\lambda t), b\sinh(\lambda t)), t \in [\theta_1/\lambda, \theta_2/\lambda)$ can be exactly reproduced by C-type curve $P(t) = \sum_{i=0}^{2} C_{i,2}(t)P_i$.

1. In $C_{i,2}(t), \beta_i \equiv i, \alpha_i \equiv \cosh\dfrac{\lambda l}{2}$;
2. $\ell = (\theta_2 - \theta_1)/\lambda$;
3. Let $\Delta = \dfrac{\lambda l \sinh\dfrac{\lambda l}{2}}{\sinh\lambda\ell - \cosh\lambda\ell}$,then $P_i(x_i, y_i)(i = 0,1,2)$ satisfy

$$\begin{cases} x_i = \Delta \cdot a\cosh\left(\lambda\left(i - \dfrac{1}{2}\right)\ell\right), \\ y_i = \Delta \cdot b\sinh\left(\lambda\left(i - \dfrac{1}{2}\right)\ell\right). \end{cases}$$

Corollary 3.3 The sufficient conditions that Lissajous curve $(x(t), y(t)) = (a\sinh(\lambda(t+\delta)), b\sinh(\lambda t)), t \in [\theta_1/\lambda, \theta_2/\lambda)$ can be exactly reproduced by C-type curve $P(t) = \sum_{i=0}^{2} C_{i,2}(t)P_i$.

1. In $C_{i,2}(t), \beta_i \equiv 1, \alpha_i \equiv \cosh\frac{\lambda l}{2}$;

2. $\ell = (\theta_2 - \theta_1)/\lambda$;

3. Let $\Delta = \dfrac{2\sin\frac{\lambda l}{2}}{\lambda l \cos^2\left(\frac{\lambda l}{2}\right)}$,then $P_i(x_i, y_i)(i = 0,1,2)$ satisfy

$$\begin{cases} x_i = a\cdot\Delta\cdot\sin\left(\lambda\left(\left(i-\frac{1}{2}\right)\ell+\delta\right)\right), \\ y_i = b\cdot\Delta\cdot\sin\left(\lambda\left(i-\frac{1}{2}\right)\ell\right). \end{cases}$$

All these formulas are derived from the standard equations of basic curves and there are no quality issues. We know all basic curves with the same type can be obtained through executing coordinate transformations to the standard equations mentioned above. Their control points can also be obtained from the corresponding results through corresponding transformations.

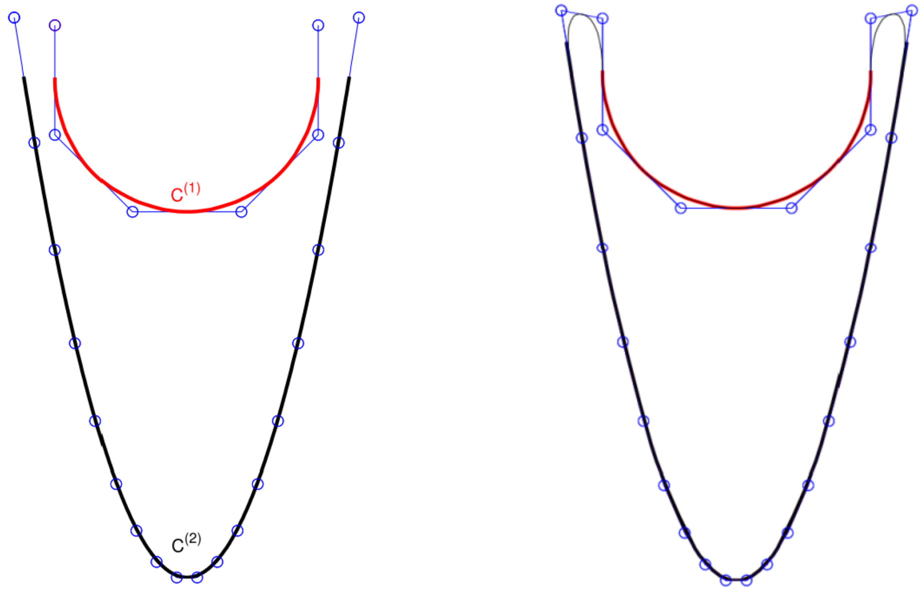
Based on above representation of basic curves, the algorithm of blending basic curves by C-type spline curves are described as Algorithm 3.1.

Algorithm 3.1 Giving the general equation representations *(x(t), y(t))* of basic curves $C^{(k)}, k = 1,2,\cdots,K$ waited to be blended. We need to blend them into a whole curve represented by C-type splines. The control polygon P and the frequency sequence ω are determined by the following procedures:

• For each piece of basic curve $C^{(k)}$, we transform it into the standard form $C'^{(k)} : (x'(t), y'(t))$ through a total coordinate transformation $T^{(k)}$, including rotation, translate and scale transformations.

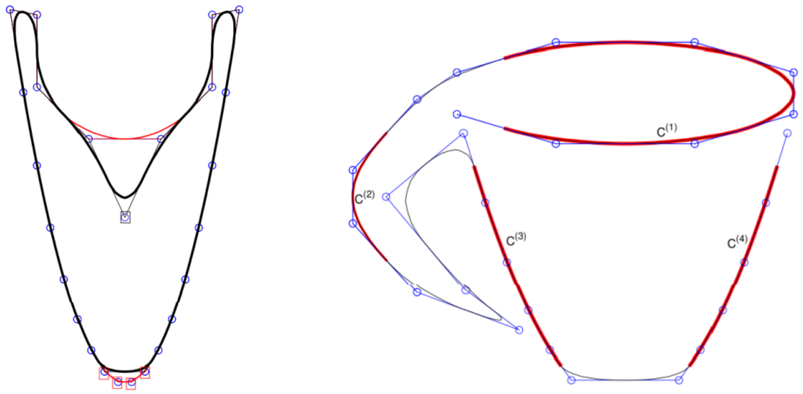
• For each piece of basic curve $C'^{(k)}$, we determine a suitable number $num^{(k)}$ of segments according to its parameter interval. Then the frequency parameter $\omega^{(k)} = \alpha^{(k)}\cdot\beta^{(k)}$ can be determined by the formulas listed in the first condition of the above corollaries. We extend it to the parameter sequence as $\omega^{(k)} = \{\alpha^{k,j_k}\beta^{k,j_k}\}, j_k = 1,\cdots,num^{(k)}+1, \alpha^{k,j_k} \equiv \alpha^{(k)}, \beta^{k,j_k} \equiv \beta^{(k)}$;

- Computing the control point sequence according to its type and the formulas listed in the latter two conditions of those corollaries for each $C'^{(k)}$;
- Implementing converse transformation $T^{(k)^{-1}}$ to control points matrices computed in the above step. Then we obtain the control sequence $P(k)=\left\{P^{k,j_k}, j_k=1,\cdots,num^{(k)}+2\right\}$;
- In order to blend these $C^{(k)}, k=1,2,\cdots,K$, we combine their parameter sequences control points sequences as $\omega'=\left\{\omega^{(1)},\omega_{insert_1},\omega^{(2)},\omega_{insert_2},\cdots,\omega^{(K-1)},\omega_{insert_{(K-1)},}\omega^{(K)}\right\}$, $P'=\left\{P^{(1)},P^{(2)},\cdots,P^{(K)}\right\}$, where ω_{insert} is new frequency parameters for those edges generated because of joining adjacent two basic curves. These new parameters can be set by users.
- As for the shape of each blending segment between two adjacent blended basic shapes, the number of control points can be set by users to frequently control this shape.
- Let the final frequency sequence and control points sequence through above procedures be $\omega=\left\{\omega_q\right\}_{q=1}^{Q_2}, P=\left\{P_q\right\}_{q=1}^{Q_2}$, where $Q_1=Q_2$ for close blending curves and $Q_1=Q_2-1$ for open ones. Then a C-type spline curve $P(t)=\sum_{i=0}^{Q_2} C_{i,2}(t)P_i$ generates, which blends *i*=0 those basic curves $C^{(k)}, k=1,2,\cdots,K$.



(a)

(b)



(c) (d)

Figure 1: some examples of representing and blending basic curves by C-type splines

Given a piece of circular arc $C^{(1)}:(x(t),y(t))=(\cos t,\sin t),t\in[0,\pi)$ and a piece of parabola $C^{(2)}:(x(t),y(t))=\left(t,\frac{4}{1.69}t^2-4\right),t\in[-1.3,1.3)$, we use Corollary 3 and Corollary 1, respectively, to determine their C-type spline representations. As illustrated in Fig. 1(upper-left), $C^{(1)}$ is represented by four segments of C-type spline curves (red) with $\omega_i=\cos(\pi/8)$, and $C^{(2)}$ is represented by 16 segments of C-type spline curves (black) with $\omega_i=0$. By directly connecting their control polygon (blue), we obtain a blending C-type spline curve (Fig. 1(upper-right)). Certainly, one can also insert, delete or adjust some control points to locally change the shape of the blending curve. Please look at Fig. 1 (bottom-left); a new control point marked with a black square is inserted and the four original control points marked with red squares are deleted. For the entire blending curve, only the shapes of two segments of the circular arcs and four segments of the parabolas are locally changed. In Fig. 1(bottom-right), the simple drawing of a cup is blended with a piece of ellipse $C^{(1)}$, a piece of hyperbola $C^{(2)}$ and two pieces of parabolas $C^{(3)}$ and $C^{(4)}$ by C-type splines. Additionally, several new control points are inserted between each two adjacent blended basic curves to frequently control the shapes of the blending segments.

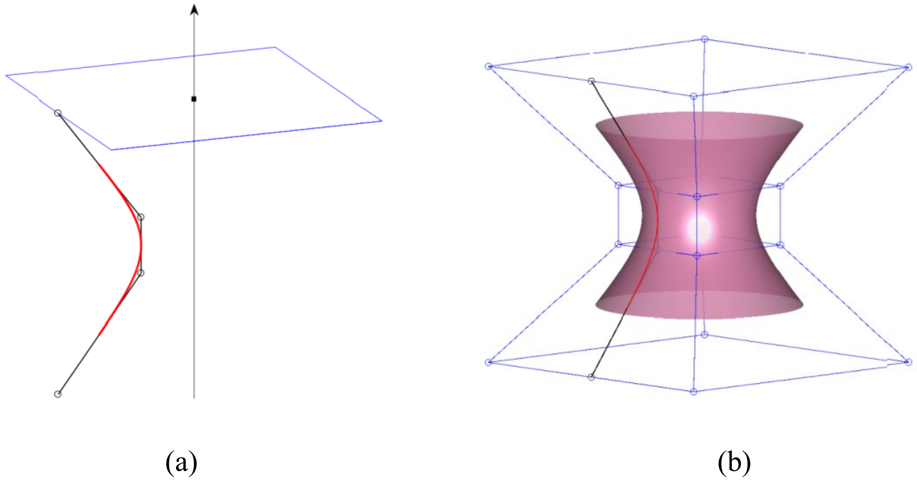
The experiment in Fig. 1 illustrated in Karciauskas et al. [Karciauskas and Peters (2011)] is like the first experiment here. Also, two basic curves are blended where parabola is common. Another basic curve is with circular, not trigonometric form. We can compare the proposed method and G^2-spline method through these two experiments. Please refer to the following Tab. 1.

Table 1: Experimental comparison

	Representing basic shapes	Representation form	Degree	Continuity order
G^2-spline method	Exactly for conics, While approximately for trigonometric shapes	Rational	3	G^2
Ours	Exactly for all	Non-rational	2	C^1

From Tab. 1, we can see that C-type splines of degree two are non-rational, one degree lower than them, but persist C^1-continuous. In fact, C-type splines of degree two can be easily elevated to C^2-continuous (naturally G^2-continuous) C-type splines of degree three. Further, our method is more convenient to model and blend basic curves. All computation formulas are simple and direct because C-type splines have similar representation forms like B-splines. While by G^2 rational splines method, one needs to sample on the basic curves and compute B-spline-like control points and weights according to very complicated transformation formulas.

As we all know, by taking a piece of basic curve as a generatrix we can easily model different types of basic surfaces. Fig. 2 shows the process of generating a hyperbolic surface from its generatrix, a piece of hyperbolic curve. Similarly, the blending problem between basic surfaces can be solved by an extended method from the cases of various curves discussed above. Fig. 3 illustrates some examples. Compared to the blending method of basic surfaces by rational bicubic G^2 splines proposed in Karciauskas et al. [Karciauskas and Peters (2011)], our method is much simpler.


(a)

(b)

Figure 2: (a) A piece of red hyperbolic curve modeled by C-type splines with a black control polygon; (b) Revolving the hyperbolic curve to generate a pink hyperbolic surface

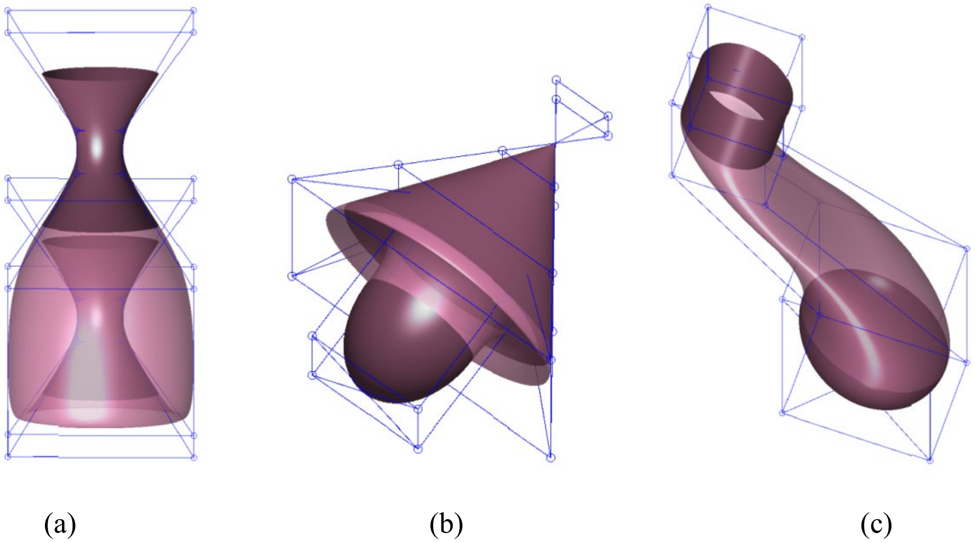


Figure 3: 2-way blending between (a) hyperbola and hyperbola, (b) semisphere and conic, (c) semisphere and cylinder, which are all modeled by C-type splines of degree 2. Here, the basic surfaces are opaque, and the blending surfaces are transparent

4 Blending basic curves by subdivision

In this section, we propose a subdivision method to blend the basic curves. In addition, the subdivision blending curve also reproduces the basic curves to be blended. However, in this section the blending segments are different from the C-type spline blending curve proposed in the last section, and the subdivision blending curve uses a common control polygon and parameter sequence. In fact, the blending segments are also very approximate. In this section, we derive the subdivision scheme and prove its C^1-continuity. To introduce the subdivision scheme, we first discuss its special case of using the constant frequency parameter. In this case, the subdivision curve converges to the corresponding C-type spline curve.

D-type spline curves belong to B-like spline curves. Therefore, it has the optimal properties of B-spline curves including control polygon refinement. In Proposition 4.1, we derive its refinement formula when using the constant frequency parameter.

Proposition 4.1 (The case of using the constant frequency parameter ω_0) Let $C_{i,2}(\ell,t)$ be C-type bases defined on knot vector $\vec{T}=\{t_i=il\}, i=0,\pm1,\cdots$, $C_{i,2}(l/2,t)$ defined on $\vec{T}'=\{t_i'=il/2\}, i=0,\pm1,\cdots$. A C-type curve which is defined over the original set of knots $\{t_i\}$ by

$$P(t)=\sum_{i=1}^{n} P_iC_{i,2}(\ell,t), t\in[3l,(n+1)\ell],$$

can also be defined over the refined knots $\{t_i'\}$ as

$$P(t)=\sum_{i=1}^{2n-2} P_i^2 C_{i,2}(\ell/2,t), t\in[3l,(n+1)\ell],$$

where

$$P_i^2=\begin{cases}\dfrac{\left(4\cos\left(\dfrac{\omega_0\ell}{2}\right)-1\right)\cdot P_{\frac{(i+1)}{2}}+P_{\frac{(i+3)}{2}}}{\dfrac{4\cos\omega_0\ell}{2}}, & i \quad odd\\ \dfrac{P_{i/2}+\left(4\cos\left(\dfrac{\omega_0\ell}{2}\right)-1\right)\cdot P_{\frac{(i+2)}{2}}}{4\cos\omega_0\,\ell/2}, & i \quad even\end{cases} \tag{4.1}$$

Proof According to the definition of $C_{i,1}(\ell,t)$ and $C_{i,1}(\ell/2,t)$, we can easily get their representation formula by Eq. (1). In addition, there exists the following relationship between the two sets of bases:

$$C_{i,1}(\ell,t)=\frac{1}{2\cos(\omega_0\ell/2)}\cdot C_{2i-3,1}(\ell/2,t)+C_{2i-2,1}(\ell/2,t)+\frac{1}{2\cos(\omega_0\ell/2)}\cdot C_{2i-1,1}(\ell/2,t)$$

Then, according to Definition 2.1, we have

$$C_{i,2}(\ell,t)=\frac{1}{\ell}\int_{t-l}^{t} C_{i,2}(\ell,\wp)ds$$

$$=\frac{1}{2}\cdot\frac{1}{\frac{\ell}{2}}\cdot\left(\int_{\frac{t-l}{2}}^{t}+\int_{t-l}^{t-\frac{\ell}{2}}\right)(C_{i,2}(\ell,\wp))ds$$

$$=\frac{1}{4\cos\left(\dfrac{\omega l}{2}\right)}\cdot C_{2i-3,2}\left(\frac{\ell}{2},t\right)+\frac{4\cos\left(\dfrac{\omega_0\ell}{2}\right)-1}{4\cos\left(\dfrac{\omega_0\ell}{2}\right)}\cdot C_{2i-2,2}\left(\frac{\ell}{2},t\right)$$

$$+\frac{4\cos\left(\dfrac{\omega_0\ell}{2}\right)-1}{4\cos\left(\dfrac{\omega_0\ell}{2}\right)}\cdot C_{2i-1,2}\left(\frac{\ell}{2},t\right)+\frac{1}{4\cos\left(\dfrac{\omega_0 l}{2}\right)}\cdot C_{2i,2}\left(\frac{\ell}{2},t\right).$$

When $t\in[3l,(n+1)\ell]$, we know $C_{j,2}(\ell/2,t)=0, j=-1,0,2n-1,2n.$

Then

$$\sum_{i=1}^{n} P_i C_{i,2}(\ell,t) = \sum_{i=1}^{n-1} (\frac{\left(4\cos\left(\frac{\omega_0\ell}{2}\right)-1\right)P_i + P_{i+1}}{4\cos\left(\frac{\omega_0\ell}{2}\right)} \cdot C_{2i-1,2}\left(\frac{\ell}{2},t\right)$$

$$+ \frac{P_i + \left(4\cos\left(\frac{\omega_0\ell}{2}\right)-1\right)P_{i+1}}{4\cos\left(\frac{\omega_0\ell}{2}\right)} \cdot C_{2i,2}\left(\frac{\ell}{2},t\right)$$

$$= \sum_{i=1}^{2n-2} P_i^3 C_{i,2}\left(\frac{\ell}{2},t\right),$$

$$P_i^2 = \begin{cases} \dfrac{\left(4\cos\left(\frac{\omega_0\ell}{2}\right)-1\right)\cdot P_{\frac{(i+1)}{2}} + P_{\frac{(i+3)}{2}}}{\frac{4\cos\omega_0\ell}{2}}, & i \quad odd \\ \dfrac{P_{i/2} + \left(4\cos\left(\frac{\omega_0\ell}{2}\right)-1\right)\cdot P_{\frac{(i+2)}{2}}}{4\cos\omega_0\,\ell/2}, & i \quad even \end{cases}$$

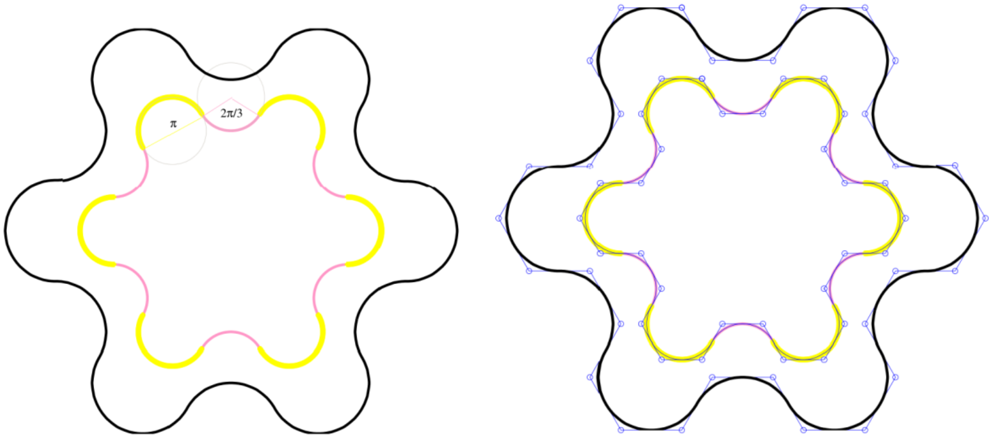
Because of the above, we derive the subdivision scheme of C-type splines with the constant frequency parameter as follows:

Algorithm 4.1 (the subdivision scheme for constant parameter curves) In l-th level of the subdivision, we assume that the current control polygon is $P^l = \{P_1^l, \cdots, P_n^l\}$ and the current parameter sequence is $U^l = \{u_i^l \mid u_i^l \equiv u_l\}_{i=1}^{n-1}$, where $u_1 = \cos(\omega_0\ell/2)$. Then, the l-th level of C-type spline subdivision for the case of the constant frequency parameter can be described as (taking open curves as examples).

$$P_i^{l+1} = \begin{cases} \dfrac{(4u_l-1)\cdot P_{(i+1)/2}^l + P_{(i+3)/2}^l}{4u_l}, & i \quad odd \\ \dfrac{P_{\frac{i}{2}}^l + (4u_l-1)\cdot P_{\frac{(i+2)}{2}}}{4u_l}, & i \quad even \end{cases} ,i = 1,2,\cdots,2n-2,$$

end

where $u_{l+1} = \sqrt{\dfrac{1+u_l}{2}}$.



(a) (b)

Figure 4: The tooling paths of a mold represented by circular splines can also be represented by C-type splines and can be generated by our subdivision scheme

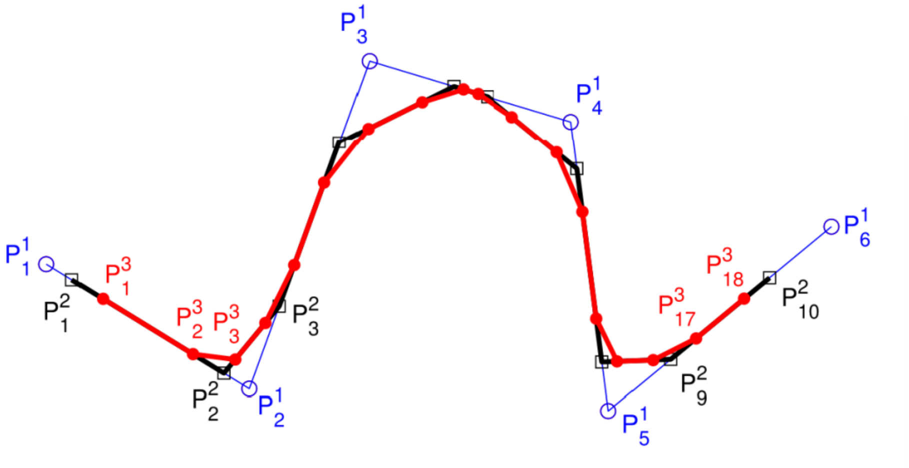
Circular splines are very popular in the domain of NC machining, while control-point curves such as B-splines, NURBS and B-like splines are more commonly used during the modeling period. The control-point curves usually have to be approximated by circular splines before machining. It would be ideal if these two forms could be precisely transformed and generated by subdivision. However, this ideal has never been implemented. Generating circular splines by subdivision has only been researched and has never been implemented, for example Ahmad et al. [Ahmad, Nasri, Overveld et al. (2001)]. As an application of Algorithm 4.1, Fig. 4 illustrates such an example. In Fig. 4(left), we can see two similar tooling paths represented by circular splines. Taking the inside one as an example, it consists of 12 pieces of circular arcs, where each of the 6 yellow pieces spans a π angle and each of the 6 pink pieces spans a $\frac{2\pi}{3}$ angle. First, we split them into three segments and two segments of circular arcs, each segment spanning a $\frac{\pi}{3}$ angle. Then, according to Corollary 3.3, we derive their representations by C-type splines with $\omega_i \equiv \frac{\sqrt{3}}{2}$ and the control points marked with blue circles in Fig. 4(right). Based on this, both tooling paths can be generated by the subdivision scheme described in Algorithm 4.1 and are marked with black curves in Fig. 4(right).

In the following section we further derive the subdivision formula for general C-type spline curves. However, it is very complicated. Under the premise of reproducing basic curves, we adopt a simpler method to generate blending segments by subdivision. In addition, the entire blending curve is very approximate to the corresponding C-type spline curve. This new subdivision scheme is described in the following algorithm:

Algorithm 4.2 (the subdivision scheme for blending curves) In l-th level of the subdivision, we assume that the current control polygon is $P^l = \left\{P_1^l, \cdots P_n^l\right\}$ and the current parameter sequence is $U^l = \{u_i^l\}_{i=1}^{n-1}$. Then, the l-th level of the C-type spline subdivision for the case of the constant frequency parameter can be described with the following pseudo-code (taking open curves as examples).

```
for $i = 1,2,\cdots,n-1$
    $P_{2i-1}^{l+1} = \begin{cases} \frac{4u_{i-1}^l - 1}{4u_{i-1}^l} \cdot P_i^l + \frac{1}{4u_{i-1}^l} \cdot P_{i+1}^l & \text{if} u_i^l = 0 \\ \frac{4u_i^l - 1}{4u_i^l} \cdot P_i^l + \frac{1}{4u_i^l} \cdot P_{i+1}^l & \text{otherwise} \end{cases};$
    $P_{2i}^{l+1} = \begin{cases} \frac{1}{4u_{i+1}^l} \cdot P_i^l + \frac{4u_{i+1}^l + 1}{4u_{i+1}^l} \cdot P_{i+1}^l & \text{if} u_i^l = 0 \\ \frac{1}{4u_i^l} \cdot P_i^l + \frac{4u_i^l \text{ -1}}{4u_i^l} \cdot P_{i+1}^l & \text{otherwise} \end{cases};$
end
for $i = 1,2,\cdots,n-1$
    $u_{2i-1}^{\prime l+1} = u_i^l$;
end
for $i = 1,2,\cdots,n-1$
    $u_{2i}^{\prime l+1} = \begin{cases} -1 & \text{if} u_i^l \neq u_{i+1}^l & \text{and} & u_i^l \neq 0 \\ u_{i-1}^l & \text{if} u_i^l \neq u_{i+1}^l, u_i^l = 0 & \text{and} & i & \text{odd} \\ u_{i+1}^l & \text{if} u_i^l \neq u_{i+1}^l, u_i^l = 0 & \text{and} & i & \text{even} \\ u_i^l & \text{otherwise} \end{cases}$
end
$P^{l+1} = \{P_i^l\}_{i=1}^{2n-2}$;
$U^{l+1} = \left\{ u_i^{l+1} \middle| u_i^{l+1} = \sqrt{\frac{1 + u_i^{\prime l+1}}{2}} \right\}_{i=1}^{2n-3}$.
```

Fig. 5 illustrates the front two levels of subdividing $P = \{P_i^1\}_{i=1}^{6}$ with $U^1 = \{1.88,1,0.62,1.42,1\}$ by the subdivision scheme for blending curves. Both the corresponding control points and the parameter sequences computed by Algorithm 4.2 are shown in this figure.

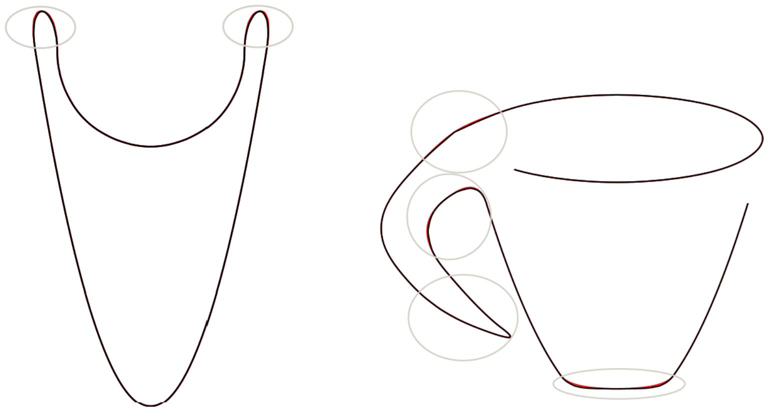


$$U^1 = \{1.88,1,0.62,1.42,1\}$$

$$U^2 = \{1.2,0,1,0,0.9,0,1.1,0,1\}$$

$$U^3 = \{\sqrt{1.1},\sqrt{1.1},0,1,1,1,0,\sqrt{0.95},\sqrt{0.95},\sqrt{0.95},0,\sqrt{1.05},\sqrt{1.05},\sqrt{1.05},0,1,1\}$$

Figure 5: The first two levels of subdividing $P = \{P^1_i\}^6_{i=1}$ with $U^1 = \{1.88,1,0.62,1.42,1\}$ by the subdivision scheme for blending curves



(a) (b)

Figure 6: Compare the blending curves respectively generated by C-type splines (red) and our subdivision scheme for blending curves (black). Those blending segments existing a little deviation between them are circled by gray ellipses

Algorithm 4.2 simplifies the refinement rule for C-type spline curves. When using it to blend basic curves, we set the subdivision parameter u^l_i to zero for blending segments. In fact, it is just a special mark. We can see that u^l_{i-1} is used to compute the left new control point and u_{i+1}^l is used to compute the right point in the formulas for computing control points in Algorithm 4.2. This ensures that each piece of the basic curve is generated by the same rule of Algorithm 4.1, which converges to those basic curves exactly represented by C-type splines. Actually, the entire blending curve obtained by Algorithm 4.2 is approximate to the corresponding C-type spline curve. In Fig. 6, the red curves and the black curves, respectively, are generated by C-type splines and our proposed subdivision scheme for blending curves with common control polygons and corresponding parameter sequences. We can see that those basic curves exactly coincide and only those blending segments in the gray ellipses have a small deviation. Furthermore, according to the theorem proposed in Dyn et al. [Dyn and Levin (1995)] and similar proofs in Tang et al. [Tang, Fang and Wang (2018)], we prove that the proposed subdivision scheme for blending curves is also C^1-continuous.

Let $m^{2,l}$ be the mask of the subdivision scheme described in Algorithm 4.2 of the l-th level. Obviously, it is a nonstationary subdivision scheme. To prove its C^1-continuity, we propose corresponding theorems between it and its relative stationary scheme.

Theorem 4.1 Let $m_s^2 = \left(\cdots,\frac{1}{4},\frac{3}{4},\frac{3}{4},\frac{1}{4},\cdots\right)$ be the subdivision mask of the stationary scheme. Then, for an arbitrary initial parameter sequence U^1, $m^{2,j}$ is asymptotically equivalent to m_s^2.

Proof Because

$$m^{2,l} = \left(\cdots,\frac{1}{4\left(u_i^l\right)},\frac{4u_i^l-1}{4\left(u_i^l\right)},\frac{4u_{i+1}^l-1}{4\left(u_{i+1}^l\right)},\frac{1}{4\left(u_{i+1}^l\right)},\cdots\right). \tag{4.2}$$

Obviously, $\lim_{l\to+\infty} u_i^l = 1$ for all u_i, then

$$m^{2,+\infty} = \lim_{l\to+\infty} m^{2,l} = \left(\cdots,\frac{1}{4},\frac{3}{4},\frac{3}{4},\frac{1}{4},\cdots\right) = m_s^2. \tag{4.3}$$

According to asymptotical equivalence theory [4], if

$$\sum\nolimits_{i=1}^{+\infty}\left\|m^{2,l}-m_s^2\right\|_\infty < +\infty, \tag{4.4}$$

then $\left\{m^{2,l}\right\} \approx \left\{m_s^2\right\}$.The detailed proof of formula (4.4) refers to that of formula (4.5) in the following Theorem 4.2.

Theorem 4.2 The subdivision scheme for blending curves in Algorithm 4.2 generates C^1-continuous limit curves.

Proof From Theorem 1, we know $\left\{m^{2,l}\right\} \approx \left\{m_s^2\right\}$. The stationary subdivision scheme with mask m_s^2 obviously converges to C^1-continuous B-spline curves of degree 2. According to Theorem 8a proposed in Dyn et al. [Dyn and Levin (1995)], Theorem 4.2 holds if we can further prove that

$$\sum\nolimits_{i=1}^{+\infty} 2^l \left\|m^{2,l}-m_s^2\right\|_\infty < +\infty \tag{4.5}$$

In the following section, we prove this conclusion.

First, from formula (4.2) and (4.3), we get

$$\left\|m^{2,l}-m_s^2\right\|_\infty=\max_i\left\{\frac{1}{4}\left|1-\frac{1}{u_i^l}\right|\right\}.$$

Therefore, we only need to prove the convergence of the series $\sum_{i=1}^{+\infty}2^l\left\|m^{2,l}-m_s^2\right\|_\infty$.

When i takes its value i_0, we assume that $\left\|m^{2,l}-m_s^2\right\|_\infty=\frac{1}{4}\left|1-\frac{1}{u_i^l}\right|$.

Then

$$\sum_{i=1}^{+\infty}2^l\left\|m^{2,l}-m_s^2\right\|_\infty=\sum_{i=1}^{+\infty}2^l\cdot\frac{1}{4}\left|1-\frac{1}{u_i^l}\right|.$$

Because $u_{i_0}^l=2\left(u_{i_0}^{l+1}\right)^2-1$ and $\lim_{l\to+\infty}u_{i_0}^{l+1}=1$. Let $W_l=2^{l\text{-}2}\left|1-\frac{1}{u_{i_0}^l}\right|$,

then

$$\lim_{l\to+\infty}\frac{W_{l+1}}{W_l}=\lim_{l\to+\infty}2\cdot\left|\frac{1-u_{i_0}^{l+1}}{u_{i_0}^{l+1}}\right|\cdot\left|\frac{1-u_{i_0}^l}{u_{i_0}^l}\right|$$

$$=\lim_{l\to+\infty}2\cdot\left|\frac{1-u_{i_0}^{l+1}}{u_{i_0}^{l+1}}\right|\cdot\left|\frac{2\left(u_{i_0}^{l+1}\right)^2-1}{2\left(1-\left(u_{i_0}^{l+1}\right)^2\right)}\right|$$

$$=\lim_{l\to+\infty}\frac{2\left(u_{i_0}^{l+1}\right)^2-1}{u_{i_0}^{l+1}\left(1+u_{i_0}^{l+1}\right)}=\frac{1}{2}<1$$

Based on the D'Alembert criteria for convergence of positive series, the series $\sum_{i=1}^{+\infty}2^l\cdot\frac{1}{4}\left|1-\frac{1}{u_{i_0}^l}\right|$ is convergent. That is $\sum_{i=1}^{+\infty}2^l\left\|m^{2,l}-m_s^2\right\|_\infty<+\infty$

So $\left\{m^{2,l}\right\}$ is C^1, Theorem 4.2 is proved.

Please note, the proposed subdivision method and corresponding theories can be naturally extended to the case of tensor product surfaces.

5 Conclusions

Compared with rational G^2 splines used to blend basic shapes, the C-type splines adopted in this paper have shown many advantages such as a nonrational form, one degree lower, a B-spline-like representation, C^1-continuity, and simpler blending procedures and computational formulas. Furthermore, the subdivision scheme for blending shapes, which also reproduces basic shapes and wholly approximates the C-type spline shapes is proposed. Both blending methods proposed in this paper are C^1-continuous. However, rational G^2 splines can be reparameterized to obtain its C^2 transitions. C-type splines of degree 2 are only C^1-continuous at most. Certainly, C^2-continuity can do this if we elevate to C-type splines of degree 3. Because of the simplicity of the blending method and formulas used to reproduce basic shapes, we choose C-type splines of degree 2. In the future, we will develop more applications, with the exception of applications for blending problems of C-type splines. For example, the C-type spline is also very suitable for an iso-geometric analysis (IGA) and the boundary element method (BEM), which have become hot topics in the recent ten years [Austin, Thomas and Yuri (2009); Nguyen, Anitescu, Bordas et al. (2015); Zhang, Sabin and Cirak (2018); Beer and Duenser (2019)]. Basic shapes are commonly used, especially for IGAs relative to CAD/CAM. If we adopt C-type basic functions both for modeling and analysis, there is some merit for modeling geometry, interpolating physical items and making refinements.

Acknowledgement: This work described in this article was supported by the National Science Foundation of China (61772164, 61272032), Provincial Key Platforms and Major Scientific Research Projects in Universities and Colleges of Guangdong (2017KTSCX143) and the Natural Science Foundation of Zhejiang Province (LY17F020025).

References

Andersson, L. E.; Stewart, N. F. (2010): *Introduction to the Mathematics of Subdivision Surfaces*. SIAM.

Austin, C. J.; Thomas J. R. H.; Yuri, B. (2009): *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley & Sons.

Bangert, C.; Prautzsch, H. (1997): Circle and sphere as rational splines. *Neural, Parallel and Scientific Computations*, vol. 5, pp. 153-162.

Beer, D.; Duenser, C. (2019): Isogeometric boundary element analysis of problems in potential flow. *Computer Methods in Applied Mechanics and Engineering*, vol. 347, pp. 517-532.

Boehm, W. (1987): Rational geometric splines. *Computer Aided Geometric Design*, vol. 4, pp. 67-77.

Dyn, N.; Levin, D. (1995): Analysis of asymptotically equivalent binary subdivision schemes. *Mathematical Analysis and Application*, vol. 193, pp. 594-621.

Fang, M. E.; Wang, G. Z. (2008): ωB-splines. *Science in China Series F: Information Sciences*, vol. 51, pp. 1167-1176.

Farin, G. E. (2006): Rational quadratic circles are parametrized by chord length. *Computer Aided Geometric Design*, vol. 23, pp. 722-724.

Floater, M. S.; Hormann, K. (2005): Surface parameterization: a tutorial and survey. advances in multiresolution for geometric modeling. *Mathematics and Visualization Book Series (MATHVISUAL)*, pp. 157-186.

Hartmann, E. (2001): Paraetric G^n blending of curves and surfaces. *The Visual Computer*, vol. 17, pp. 1-13.

Hui, K. C. (1999): Shape blending of curves and surfaces with geometric continuity. *Computer-Aided Design*, vol. 31, pp. 819-828.

Karciauskas, K.; Peters, J. (2011): Rational bi-cubic G2 splines for design with basic shapes. *Eurographics Symposium on Geometry Processing*, vol. 30, pp. 1389-1395.

Karciauskas, K.; Peters, J. (2011): Rational G^2 splines. *Graphical Models*, vol. 73, pp. 286-295.

Kiciak, P. (2013): Spline surfaces of arbitrary topology with continuous curvature and optimized shape. *Computer-Aided Design*, vol. 45, pp. 154-167.

Kouibia, A.; Pasadas, M.; Sbibihb, D.; Zidna, A.; Belkhatir, B. (2013): Geometric continuity C^1G^2 of blending surfaces. *Computer-Aided Design*, vol. 45, pp. 733-738.

Lin, H. W.; Xiong, Y. Y.; Liao, H. W. (2014): Semi-structured B-spline for blending two B-spline surfaces. *Computers and Mathematics with Applications*, vol. 68, pp. 706-718.

Ma, W. Y. (2005): Subdivision surfaces for CAD-an overview, *Computer-Aided Design*, vol. 37, no. 7, pp. 693-709.

Nasri, A. H.; Van Overveld, C. W.; Wyvill, B. (2001): A recursive subdivision algorithm for piecewise circular spline. *Computer Graphics Forum*, vol. 20, pp. 33-45.

Nguyen, V. P.; Anitescu, C.; Bordas, S. P.; Rabczuk, T. (2015): Isogeometric analysis: an overview and computer implementation aspects. *Mathematics and Computers in Simulation*, vol. 117, pp. 89-116.

Salomon, D. (2006): Curves and surfaces for computer graphics. *Chapter 7. B-Spline Approximation*. Springer.

Shen, X. L.; Hu, W. T.; Fan, J. (2017): Automatic blade blend modeling and hexahedral mesh regeneration for aircraft engine optimization. S*tructural and Multidisciplinary Optimization*, vol. 3, pp. 1-11.

Tang, J.; Fang, M. E.; Wang, G. Z. (2018): Subdivision of uniform ωB-spline curves and two proofs of its C^{k-2}-continuity. *Computer Modeling in Engineering & Sciences*, vol. 115, pp. 263-280.

Wang, Z. J.; Lin, X.; Fang, M. E.; Yao, B.; Peng, Y. et al. (2017): Re2l: an efficient output-sensitive algorithm for computing Boolean operations on circular-arc polygons and its applications. *Computer-Aided Design*, vol. 83, pp. 1-14.

Zhang, Q. L.; Sabin, M.; Cirak, F. (2018): Subdivision surfaces with isogeometric analysis adapted refinement weights. *Computer-Aided Design*, vol. 102, pp. 104-114.