Cubic B-spline curve approximation by curve unclamping

Xiao-Diao Chen, Weiyin Ma, Jean-Claude Paul

Cubic B-spline curve approximation by curve unclamping

Xiao-Diao Chen [a,b,*], Weiyin Ma [a], Jean-Claude Paul [c,d]

[a] Department of MEEM, City University of Hong Kong, Hong Kong, PR China
[b] Hangzhou Dianzi University, Hangzhou, 310018, PR China
[c] School of Software, Tsinghua University, Beijing 100084, PR China
[d] INRIA, France

ARTICLE INFO

Article history:
Received 1 August 2009
Accepted 22 January 2010

Keywords:
Approximation
Cubic B-spline
Inner point interpolation method
Curve unclamping

ABSTRACT

A new approach for cubic B-spline curve approximation is presented. The method produces an approximation cubic B-spline curve tangent to a given curve at a set of selected positions, called tangent points, in a piecewise manner starting from a seed segment. A heuristic method is provided to select the tangent points. The first segment of the approximation cubic B-spline curve can be obtained using an inner point interpolation method, least-squares method or geometric Hermite method as a seed segment. The approximation curve is further extended to other tangent points one by one by curve unclamping. New tangent points can also be added, if necessary, by using the concept of the minimum shape deformation angle of an inner point for better approximation. Numerical examples show that the new method is effective in approximating a given curve and is efficient in computation.

1. Introduction

B-spline curve approximation is a classic problem in computer aided geometric design [1]. It has many applications in CAD/CAM and computer graphics. For example, non-polynomial parametric curves should be turned into NURBS (non-uniform rational B-splines), the de-facto standard of the CAD/CAM and related graphics industries, before their use in most geometric modeling systems. In the vector font reconstruction problem, the fonts are usually also approximated by smooth B-spline curves.

In practical applications, the distance between the given curve and the approximation curve should be less than a preset tolerance, and the resulting curve is called an error-bounded approximation. The Hausdorff distance is the best metric to measure the corresponding distance between two curves [2,1]. However, it is not an easy task to accurately compute the Hausdorff distance, and other approximate error distances are often used for the error check process. One alternative method is to sample the given curve into a sequence of points and to use the distance between the points and the approximation curve instead. One of the most important goals in B-spline curve approximation is to use as few control points as possible while maintaining the desired accuracy. Usually, it is difficult to obtain the required number of the control points in advance. Most of the methods use an iterative process for adjusting the number of the control points, the parameter values, and the knots to maintain an error bound [3–8]. Conventional B-spline interpolation methods are able to exactly interpolate the selected points. However, there is no guarantee at other positions of the curve and the final approximation is usually improved by the least-squares curve fitting technique with added sample points [1,9].

Recently, Gofuku, Tamura and Maekawa propose a geometric algorithm for interpolating a set of points and their normal directional vectors or tangent directional vectors [10]. In principle, their geometric method is able to meet with any small tolerance. However, the result is an approximate curve and it does not exactly interpolate the positions of the points or the directional vectors of their derivatives. When an appropriate set of tangent points are selected, the method in [10] is able to obtain a good approximation to the given curve.

In this paper, we suppose that the given curve is sampled into a sequence of ordered points with information about their parameter values, derivative vectors and curvatures. An alternative solution to the approximation problem is curve fitting often using positions of the sample points only. We refer to related work on noisy points fitting: [3,8,11,12]. Least-squares B-spline curve fitting technique is frequently used in the previous methods, which minimizes the least-squares error defined as

$$E(\mathbf{b}_0, \ldots, \mathbf{b}_n) = \sum_{i=0}^{m} \|\mathbf{C}(t_i) - \mathbf{p}_i\|^2, \tag{1}$$

where $\mathbf{C}(t)$ is the approximation B-spline curve with control points $\{\mathbf{b}_j\}_{j=0}^{n}$, $\{\mathbf{p}_i\}_{i=0}^{m}$ are the sampled points from the given curve, and t_i is the parameter value of $\mathbf{p}_i$. Before minimizing $E(\mathbf{b}_0, \ldots, \mathbf{b}_n)$, one

* Corresponding author at: Department of MEEM, City University of Hong Kong, Hong Kong, PR China. Tel.: +86 571 86915140.
E-mail addresses: xiaodiao@hdu.edu.cn (X.-D. Chen), mewma@cityu.edu.hk (W. Ma), paul@tsinghua.edu.cn (J.-C. Paul).

should determine the number n of the control points and the corresponding knot vector $\mathbf{U}_n$ of $\mathbf{C}(t)$ as well. The approximation result depends on the selection of n and $\mathbf{U}_n$. There are some notable works based on the optimization approach [13,14], but many challenges still remain to be tackled. When n is equal to m, the curve fitting problem degenerates to a general interpolation problem [15]. A knot removal approach can then be used to reduce the number of control points, which is to progressively remove a selected number of knots that have the least significance to the approximation curve until the error reaches the tolerance level [4,16,17]. A knot increment approach can also be used for the curve fitting problem, which is to use less number of knots at first, and then add more knots to obtain the desired accuracy. Usually, the knot increment approach tends to require less number of control points than the knot removal approach [1]. When the number of the control points is determined, knot placement becomes the most important issue. Recently, Park and Lee provide a dominant point based method for knot placement [1]. Several initial dominant points are selected at local extreme curvatures. More dominant points are then iteratively added to make the approximation curve rapidly converge to the sequence of points. In practical applications, since the number of the control points n is difficult to be obtained in advance according to the preset tolerance, n may be frequently changed and all the control points need to be recalculated. Thus, almost all of the previous computations are wasted.

This paper presents an adaptive method for the B-spline curve approximation problem, which is based on a curve unclamping technique. Firstly, it constructs a cubic Bézier curve by using the inner point interpolation method in [18] for the 2D case or the geometric Hermite method in [19] for the 3D case to approximate the first segment of the given curve (or part of the ordered point sequence) within the tolerance. This cubic Bézier curve is regarded as the seed segment of the final approximation B-spline curve, which is to be extended to other tangent points, one by one, by the curve unclamping technique. The given curve and the approximation curve are tangent with each other at the tangent points. A heuristic method is provided for selecting the locations of the tangent points. It is proved in this paper that such tangent approximation curve exists, and the approximation effect is shown by numerical examples.

The rest of the paper is organized as follows. In Section 2, we briefly review B-spline curve and the conventional B-spline interpolation methods. In Section 3, we introduce methods for constructing the seed cubic Bézier curve. In Section 4, we demonstrate the constructive tangent B-spline curve by curve unclamping. Section 5 discusses the selection of the tangent points. Section 6 shows that the approximation effect can be improved by adding one or more tangent points. Examples and comparisons are shown in Section 7, and conclusions are drawn at the end of this paper.

2. Conventional B-spline interpolation methods

Given a knot vector

$$\mathbf{U} = (u_0, u_1, u_2, \cdots, u_r), \quad u_0 \le u_1 \le \cdots \le u_r,$$

the associated B-spline functions $N_{i,p}$ are defined as follows:

$$N_{i,1}(u) = \begin{cases} 1, & \text{for } u_i \le u < u_{i+1}, \\ 0, & \text{otherwise}, \end{cases}$$

and

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

for $p \ge 2$ and $i = 0, 1, \ldots, r - p$.

A B-spline curve with $n + 1$ control points is then defined as

$$\mathbf{C}(u) = \sum_{i=0}^{n} \mathbf{q}_i N_{i,p}(u), \quad u \in [u_{p-1}, u_{n+1}].$$

With $n + 1$ data points $\mathbf{p}_0, \ldots, \mathbf{p}_n$, one can find an interpolation B-spline curve. In any case, one needs to assign a location parameter τ_i to each of the data points, define a knot vector $\mathbf{U}$, and finally compute the control points [3,5]. The location parameters τ_i can be assigned based on the chord length as

$$\tau_0 = 0, \qquad \tau_i = \tau_{i-1} + \frac{\|\mathbf{p}_i - \mathbf{p}_{i-1}\|}{\sum_{i=1}^{n} \|\mathbf{p}_i - \mathbf{p}_{i-1}\|},$$

or by using a centripetal method as

$$\tau_0 = 0, \qquad \tau_i = \tau_{i-1} + \frac{\sqrt{\|\mathbf{p}_i - \mathbf{p}_{i-1}\|}}{\sum_{i=1}^{n} \sqrt{\|\mathbf{p}_i - \mathbf{p}_{i-1}\|}}.$$

The knot vector $\mathbf{U}$ can be defined as

$$u_0 = \cdots = u_{p-1} = 0, \qquad u_{r-p+1} = \cdots = u_r = 1,$$
$$u_{j+p-1} = \frac{1}{p-1} \sum_{i=j}^{j+p-2} \tau_i, \quad j = 1, \ldots, n - p + 1.$$

A standard interpolation problem is to solve a linear system

$$\mathbf{C}(\tau_i) - \mathbf{p}_i = 0, \quad 0 \le i \le n.$$

When there are $m + 1$ data points, i.e., $\{\bar{\mathbf{p}}_j\}_{j=0}^{m}$, with $m > n$, the corresponding location parameters $\{\bar{\tau}_j\}$ and the knot vector $\bar{\mathbf{U}}$ can also be derived from the data points $\{\bar{\mathbf{p}}_j\}$ in a similar way. Suppose that the new approximation curve corresponding to $\bar{\mathbf{U}}$ is $\bar{\mathbf{C}}(u)$, then the least-squares method is to solve the new control points by minimizing

$$\sum_{j=0}^{m} \|\bar{\mathbf{C}}(\bar{\tau}_j) - \bar{\mathbf{p}}_j\|^2.$$

Usually, the least-squares method produces well-behaved results compared to those of the standard interpolation method, but it cannot ensure that the resulting curve exactly interpolates the data points $\{\bar{\mathbf{p}}_j\}$.

3. Constructing the seed curve

The seed curve is constructed as a cubic Bézier curve and can be written as

$$\mathbf{A}(t) = (1-t)^3\mathbf{q}_0 + 3(1-t)^2 t\mathbf{q}_1 + 3(1-t)t^2\mathbf{q}_2 + t^3\mathbf{q}_3,$$

where $\{\mathbf{q}_i\}_{i=0}^{3}$ are the control points. Suppose that the given curve has two end points $\mathbf{p}_0$ and $\mathbf{p}_1$, and the corresponding tangent vectors at the end points are $\mathbf{t}_0$ and $\mathbf{t}_1$, respectively. From the tangent constraint at the end points, we have

$$\mathbf{q}_0 = \mathbf{p}_0, \qquad \mathbf{q}_1 = \mathbf{p}_0 + \alpha\mathbf{t}_0, \qquad \mathbf{q}_2 = \mathbf{p}_1 - \beta\mathbf{t}_1, \qquad \mathbf{q}_3 = \mathbf{p}_1.$$

When the values of α and β are determined, the corresponding cubic Bézier curve is then defined. The geometric Hermite methods such as the one in [19] can be used for determining the values of of α and β. The least-squares method can also be used, which is to minimize

$$\int_0^1 \|\mathbf{A}(t) - \mathbf{C}(t_0 + (t_1 - t_0)t)\|^2 \mathrm{d}t,$$

where t_0 and t_1 are the parameters of points $\mathbf{p}_0$ and $\mathbf{p}_1$ on the given curve $\mathbf{C}(t)$, respectively.

For the 2D case, we also use the inner point interpolation method in [18], which is to select an inner point where the given

curve and the approximation curve are tangent with each other. Suppose that the inner point of the given curve is $\mathbf{p}^\star = (x^\star, y^\star)$, and $\mathbf{t}^\star = (t_x^\star, t_y^\star)$ is the corresponding tangent vector of the given curve at $\mathbf{p}^\star$. Let $\mathbf{A}(t)$ be $(X(t), Y(t))$. Then we have

$$\begin{cases} X(t) - x^\star = 0, \\ Y(t) - y^\star = 0, \\ X'(t)t_y^\star - Y'(t)t_x^\star = 0. \end{cases} \tag{2}$$

The equation system (2) has three unknown variables; i.e., α, β and t, and three equations as well. The first two equations in the equation system (2) are linear with respect to α and β. The terms α and β can then be directly solved as $\alpha(t)$ and $\beta(t)$. Substituting $\alpha(t)$ and $\beta(t)$ into the third equation of the equation system (2), we obtain a univariate equation in t, which can be simplified into a univariate cubic polynomial equation $H(t)$. A brief overview of related details can be found in Appendix. By solving $H(t) = 0$, we finally obtain the values of t, α and β. Thus, the resulting approximation cubic Bézier curve is also obtained.

4. Constructing the tangent cubic B-spline curve by curve unclamping

In this section, we construct a cubic B-spline curve which is tangent with the given curve $\mathbf{D}(u)$ at positions $\{\mathbf{p}_i\}_{i=0}^k$ with respective parameters $\{\bar{u}_i\}_{i=0}^k$. The derivative vector $\mathbf{t}_i$ of the curve at point $\mathbf{p}_i$ can be directly computed. Firstly, we construct a cubic Bézier curve $\mathbf{C}_1(u)$ discussed in the previous section and the curve is tangent with the given curve at the two end points $\mathbf{p}_0$ and $\mathbf{p}_1$. If k is equal to 1, then the cubic Bézier is the resulting approximation curve. When k is large than 1, we progressively extend the curve $\mathbf{C}_i(u)$ by curve unclamping in [5,20] to obtain $\mathbf{C}_{i+1}(u)$ which is tangent with the given curve at point $\mathbf{p}_{i+1}$, $i = 1, \ldots, k-1$ starting from the seed curve $\mathbf{C}_1(u)$. The curve $\mathbf{C}_k(u)$ is the resulting approximation cubic B-spline curve.

Curve unclamping works as follows. Suppose that the curve $\mathbf{C}_i(u)$ is defined by control points $\{\mathbf{p}_j^i\}_{j=0}^n$ and knot vector

$$\mathbf{U}_i = \{0, 0, 0, 0, u_4, \ldots, u_n, 1, 1, 1, 1\},$$

where $n = i+2$. We want to turn the curve $\mathbf{C}_i(u)$ into a new curve $\mathbf{C}_i^\star(u)$ by curve unclamping, and then obtain the new control points $\{\mathbf{q}_j^i\}_{j=0}^n$ based on a new knot vector

$$\mathbf{U}_i^\star = \{0, 0, 0, 0, u_4, \ldots, u_n, 1, u, u, u\}.$$

It can be verified that

$$\begin{aligned} &\mathbf{q}_j^i = \mathbf{p}_j^i, \quad j = 0, \ldots, n-2, \\ &\mathbf{q}_{n-1}^i = \mathbf{p}_{n-1}^i + \frac{(u-1)}{(1-u_{n-1})}(\mathbf{p}_{n-1}^i - \mathbf{p}_{n-2}^i), \\ &\mathbf{q}_n^i = \frac{(u-u_n)^2}{(1-u_n)^2}\mathbf{p}_n^i + \frac{(u-1)^2}{(1-u_n)(1-u_{n-1})}\mathbf{p}_{n-2}^i \\ &\qquad - \left[\frac{(u-u_n)(u-1)}{(1-u_n)^2} + \frac{(u-1)(u-u_{n-1})}{(1-u_{n-1})(1-u_n)}\right]\mathbf{p}_{n-1}^i \\ &\qquad = \mathbf{Q}_0^i(u-1)^2 + \mathbf{Q}_1^i 2(u-1)(2-u) + \mathbf{Q}_2^i(2-u)^2, \end{aligned} \tag{3}$$

where

$$\begin{aligned} \mathbf{Q}_0^i &= \frac{(2-u_n)^2}{(1-u_n)^2}\mathbf{p}_n^i + \frac{1}{(1-u_n)(1-u_{n-1})}\mathbf{p}_{n-2}^i \\ &\quad - \left[\frac{(2-u_n)}{(1-u_n)^2} + \frac{(2-u_{n-1})}{(1-u_{n-1})(1-u_n)}\right]\mathbf{p}_{n-1}^i, \\ \mathbf{Q}_1^i &= \frac{2-u_n}{1-u_n}\mathbf{p}_n^i - \frac{1}{1-u_n}\mathbf{p}_{n-1}^i, \\ \mathbf{Q}_2^i &= \mathbf{p}_n^i. \end{aligned}$$

The new control points $\{\mathbf{q}_j^i\}_{j=n-1}^n$ are dependent on the parameter u. For the 2D case, u can be simply set as u_t, which is one of the real roots of the quadric polynomial equation

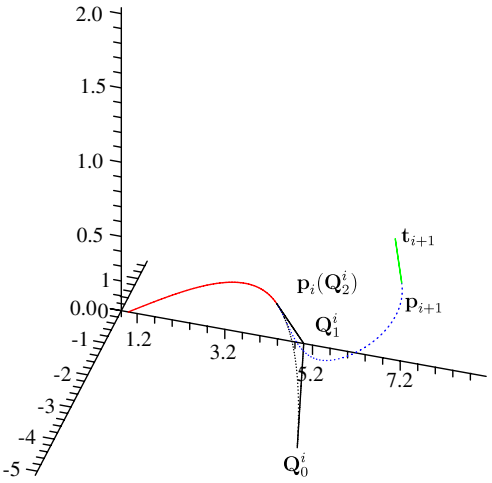

Fig. 1. The locus of $\mathbf{q}_n^i$ has no intersection point with the tangent line in a 3D case. The solid curve in red, the dotted curve in black, the dashed curve in green and the dash-dotted curve in blue are the given curve, the locus of $\mathbf{q}_n^i$, the tangent line at point $\mathbf{p}_{i+1}$ and the extended segment between $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$, respectively.

$$\begin{aligned} &(\mathbf{Q}_0^i(u-1)^2 + \mathbf{Q}_1^i 2(u-1)(2-u) \\ &\quad + \mathbf{Q}_2^i(2-u)^2 - \mathbf{p}_{i+1}) \times \mathbf{t}_{i+1} = 0, \end{aligned} \tag{4}$$

where "$\times$" denotes the cross product. Let $\mathbf{C}_{i+1}(u)$ be defined by control points

$$\mathbf{q}_j^{i+1} = \mathbf{q}_j^i, \quad j = 0, \ldots, n, \quad \text{and} \quad \mathbf{q}_{n+1}^{i+1} = \mathbf{p}_{i+1},$$

and the knot vector

$$\mathbf{U}_{i+1}^\star = \{0, 0, 0, 0, u_4, \ldots, u_n, 1, u_t, u_t, u_t, u_t\}.$$

The curve $\mathbf{C}_{i+1}(u)$ and the given curve are tangent with each other at point $\mathbf{p}_{i+1}$. Usually, the knot vector $\mathbf{U}_{i+1}^\star$ is normalized to be rewritten as

$$\{0, 0, 0, 0, u_4/u_t, \ldots, u_n/u_t, 1/u_t, 1, 1, 1, 1\}.$$

In the 3D case, as shown in Fig. 1, the locus of $\mathbf{q}_n^i$ (in black) may has no intersection point with the tangent line (in green) of the given curve at point $\mathbf{p}_{i+1}$, and the univariate equation system (4) which consists of three quadric polynomial equations may have no such solution. One alternative method is to use two pieces of the approximation curves between each two adjacent points $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$, which ensures that the approximation curve and the given curve are tangent with each other at $\mathbf{p}_{i+1}$. Suppose that the curve $\mathbf{C}_i(u)$ is unclamped to the new knot vector

$$\mathbf{U}_i = \{0, 0, 0, 0, u_4, \ldots, u_n, 1, u_{n+2}, u_t, u_t\},$$

with its new control points $\{\bar{\mathbf{q}}_j^i\}$. Let the curve $\mathbf{C}_{i+1}^\star(u)$ be determined by control points

$$\mathbf{q}_j^{i+1} = \bar{\mathbf{q}}_j^i, \quad j = 0, \ldots, n, \qquad \mathbf{q}_{n+1}^{i+1} = \mathbf{p}_{i+1} - \alpha\mathbf{t}_{i+1} \quad \text{and}$$
$$\mathbf{q}_{n+2}^{i+1} = \mathbf{p}_{i+1},$$

and the knot vector

$$\mathbf{U}_{i+1} = \{0, 0, 0, 0, u_4, \ldots, u_n, 1, u_{n+2}, u_t, u_t, u_t, u_t\}.$$

It can be verified that the curve $\mathbf{C}_{i+1}^\star(u)$ and the given curve are tangent with each other at point $\mathbf{p}_{i+1}$ (also see Fig. 1). The resulting values of α, u_{n+2} and u_t for the 3D case depend on the choice of the objective optimization function which is beyond the scope of this paper, and here we omit the details.

Another solution is to simply set u as one of the real roots of the quadric polynomial equation

$$\begin{aligned} &(\mathbf{Q}_0^i(u-1)^2 + \mathbf{Q}_1^i 2(u-1)(2-u) \\ &\quad + \mathbf{Q}_2^i(2-u)^2 - \mathbf{p}_{i+1}) \cdot \mathbf{v}_{i+1} = 0, \end{aligned}$$

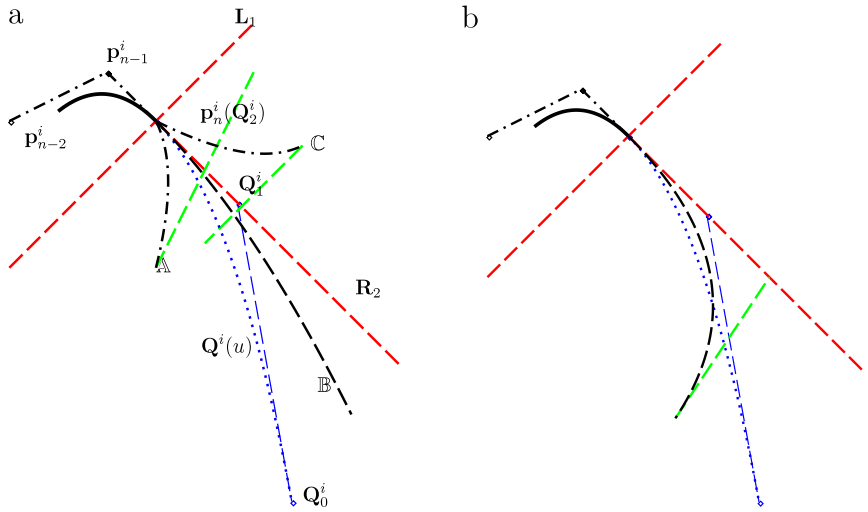

Fig. 2. Illustration for the existence of $u_t \in (1, 2]$: (a) the dashed normal line, the dashed tangent line and the dotted locus of $\mathbf{q}^i_n$ divide the extended region into three parts $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$, respectively, and the existence of u_t is obvious for the cases when the extended curve locates in part $\mathbb{A}$ or $\mathbb{C}$; and (b) the possibility of the existence of u_t in the case $\mathbb{B}$.

where "·" denotes the inner product and $\mathbf{v}_{i+1}$ is the normal directional vector at point $\mathbf{p}_{i+1}$. In this case, the tangent lines of the approximation curve and the given curve at point $\mathbf{p}_{i+1}$ lie in the same tangent plane determined by point $\mathbf{p}_{i+1}$ and the normal vector $\mathbf{v}_{i+1}$, which is similar to that of the method in [10].

5. Selecting the tangent points $\{\mathbf{p}_i\}$

Suppose that the tangent points and their parameters $\{\bar{u}_i\}$ are obtained. The first segment of the approximation cubic B-spline curve can be obtained by the methods mentioned in Section 2, which approximates the given curve $\mathbf{D}(u)$ in the interval $[\bar{u}_0, \bar{u}_1]$. We then extend the approximation B-spline curve by curve unclamping which is tangent with the given curve $\mathbf{D}(u)$ at point $\mathbf{D}(\bar{u}_i)$ in order, $i = 2, \ldots, l+1$. We first discuss the existence of the $u_t \in (1, 2]$ in the extension process, and then provide a constructive method for determining the values of $\{\bar{u}_i\}$.

In principle, u_t can be of an arbitrary value in the interval $(1, +\infty)$. Without loss of generality, we only discuss the existence of $u_t \in (1, 2]$. As shown in Fig. 2(a), the solid curve denotes the given curve $\mathbf{D}(u)$. $\mathbf{p}^i_{n-2}$, $\mathbf{p}^i_{n-1}$ and $\mathbf{p}^i_n$ are the last three control points of the approximation curve $\mathbf{C}_i(u)$. The dotted curve is the quadratic Bézier curve $\mathbf{Q}^i(u)$ with the control points $\mathbf{Q}^i_2$, $\mathbf{Q}^i_1$ and $\mathbf{Q}^i_0$. The line $\mathbf{L}_1$ is passing through point $\mathbf{p}^i_n$ and is perpendicular with the vector $\overrightarrow{\mathbf{p}^i_n\mathbf{p}^i_{n-1}}$, while the ray line $\mathbf{R}_2$ starts at point $\mathbf{p}^i_n$ and passes through point $\mathbf{Q}^i_1$. The line $\mathbf{L}_1$, the ray line $\mathbf{R}_2$ and the quadratic Bézier curve $\mathbf{Q}^i(u)$ locally partition the corresponding region into three sub-regions, marked as $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$. The local segment of $\mathbf{D}(u)$ which is on the right side of the line $\mathbf{L}_1$ may be located in sub-region $\mathbb{A}$, $\mathbb{B}$ or $\mathbb{C}$. If the case of sub-region $\mathbb{A}$ occurs, the curvature of $\mathbf{D}(u)$ at point $\mathbf{p}^i_n$ is larger than that of the quadratic Bézier curve $\mathbf{Q}^i(u)$. If the case of sub-region $\mathbb{B}$ occurs, the curvature of $\mathbf{D}(u)$ at point $\mathbf{p}^i_n$ is smaller than that of the quadratic Bézier curve $\mathbf{Q}^i(u)$. If the case of sub-region $\mathbb{C}$ occurs, the point $\mathbf{p}^i_n$ is likely to be an inflexion point. In cases when sub-region $\mathbb{A}$ or $\mathbb{C}$ occurs, there exists a tangent line of $\mathbf{D}(u)$ at point $\mathbf{D}(u^\star)$ that intersects with the curve $\mathbf{Q}^i(u)$, which means that there exists $u_t \in (1, 2]$. It seems not so good in the case when sub-region $\mathbb{B}$ occurs. However, even in this case, the end tangent line is able to intersect with curve $\mathbf{Q}^i(u)$ as shown in Fig. 2(b). We have the following theorem.

Theorem 1. *If the ratio of the curvatures of the given curve $\mathbf{D}(u)$ and the approximation curve $\mathbf{C}_i(u)$ at the tangent point $\mathbf{p}_i$ is larger than 3/4, then the case of sub-region $\mathbb{B}$ cannot occur (also see Fig. 2 (a)). Thus, it ensures the existence of the tangent approximation curve by curve unclamping.*

Proof. Without loss of generality, suppose that $\mathbf{C}_i(u)$ is just a cubic Bézier curve with control points $\{\mathbf{p}^1_j\}^4_{j=1}$, where $\mathbf{p}^1_4$ coincides with the tangent point $\mathbf{p}_i$. It can be verified that the curvatures of the approximation curve $\mathbf{C}_i(u)$ and the quadratic Bézier curve $\mathbf{Q}^i(u)$ at the tangent point $\mathbf{p}_i$ is

$$c_1 = 18K/27$$

and

$$c_2 = K/2,$$

respectively, where

$$K = \frac{\|(\mathbf{p}^1_4 - \mathbf{p}^1_3) \times (\mathbf{p}^1_4 - 2\mathbf{p}^1_3 + \mathbf{p}^1_2)\|}{\|\mathbf{p}^1_4 - \mathbf{p}^1_3\|^3}.$$

Let the curvature of the given curve $\mathbf{D}(u)$ at point $\mathbf{p}_i$ be c_d. From the assumption, we have $c_d > 3/4c_1$. Note that $c_2 = 3/4c_1$, we have $c_d > c_2$. As Fig. 2(a) shows, if the case of sub-region $\mathbb{B}$ occurs, c_d must be smaller than c_2. Thus, the case of sub-region $\mathbb{B}$ cannot occur, and we have completed the proof. □

We then provide a heuristic method for selecting the points $\{\mathbf{p}_i\}$ or the corresponding parameters $\{\bar{u}_i\}$. Suppose that $\{\hat{u}_i\}^l_{i=0}$ are the parameters where the given curve $\mathbf{D}(u)$ reaches its local extreme signed curvature. Firstly, we construct a cubic Bézier curve tangent with the given curve $\mathbf{D}(u)$ at three points with the parameters $u = 0, \hat{u}_0, \bar{u}_1$, where the initial value of $\bar{u}_1$ is set as $(\hat{u}_0+\hat{u}_1)/2$. The value of $\bar{u}_1$ can be further refined between $\hat{u}_0$ and $(\hat{u}_0 + \hat{u}_1)/2$ according to the approximation effect, including curvature approximation at the point $\mathbf{D}(\bar{u}_1)$. Once the first approximation curve segment $\mathbf{C}_1(u)$ is obtained, we progressively extend the approximation curve to the points $\{\mathbf{D}(\bar{u}_i)\}$ by curve unclamping. Then the initial parameters $\{\bar{u}_i\}^{l+1}_{i=0}$ of the tangent points $\{\mathbf{p}_i\}$ can be simply set as

$$\begin{aligned} &\bar{u}_0 = 0, \\ &\bar{u}_i = (\hat{u}_i + \hat{u}_{i-1})/2, \quad i = 1, \ldots, l, \\ &\bar{u}_{l+1} = 1. \end{aligned} \tag{5}$$

As shown in Fig. 3, this simple selection is able to obtain a good approximation. The selection of $\{\bar{u}_i\}^{l+1}_{i=0}$ can be further refined for the planar case which is based on the following observation. Each extended segment is a Bézier curve and is tangent with the given curve at the two end points. Fig. 4 shows three different cases

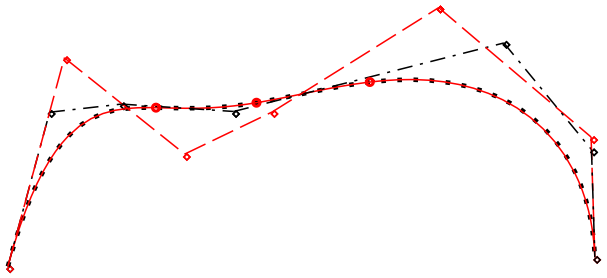

Fig. 3. Selecting tangent points with Eq. (5). The solid curve in red is a quintic B-spline curve and the dotted curve in black is the cubic approximation B-spline curve. The two curves are tangent with each other at both the two end-points and three inner points denoted by solid circles in red, which has a good approximation effect.

according to the number of inner intersection points between the approximation Bézier curve and the given curve; i.e., zero, one and two. The solid curve in red and the dotted curve in black are the given curve and the approximation curve, respectively. In Fig. 4(a) and (c), both of the curvatures of the approximation curve at the end points are smaller or larger than those of the given curve. In Fig. 4(b), one of the curvatures of the approximation curve at the end points is smaller than that of the given curve, and the other is larger than that of the given curve. The inner product between the normal vector at point $\mathbf{D}(\hat{u}_{i-1})$ and the vector $\overrightarrow{\mathbf{C}(\hat{u}_{i-1})\mathbf{D}(\hat{u}_{i-1})}$ will be helpful to further distinguish cases Fig. 4(a) and Fig. 4(c). Similarly as the analysis in [21,19,22], the optimal approximation orders of Fig. 4(a $\sim$ c) are 4, 5 and 6, respectively. As shown in Fig. 4, the higher the optimal approximation order, the better the approximation effect. The approximation effect tends to be poor at places having local maximum curvatures. We sample several points in the interval $[\hat{u}_{i-1}, \hat{u}_i]$, and select the one such that the corresponding optimal approximation order is the highest and the distance between the point $\mathbf{C}(\hat{u}_{i-1})$ and the approximation curve is the smallest. The extended segment which passes through point $\mathbf{C}(\hat{u}_{i-1})$ with local extreme curvature tends to have a better approximation effect in both the position and the curvature. Fig. 5 shows such an example by using the heuristic method. Fig. 5(a) shows the given curve $\mathbf{D}(u)$ in red and the approximation curve $\mathbf{C}(u)$ in black. Fig. 5(b) shows the selected parameters $\{\bar{u}_i\}$ which are denoted by dotted lines, and the corresponding curvature plots of the two curves. As shown in Fig. 5, the new method is able to lead to a good approximation in both the position and the curvature.

6. Adaptive method with a given tolerance

The new method is to firstly construct a cubic Bézier curve for approximating the first segment of the final curve as a seed segment. Then by using the curve unclamping method, the approximation curve is extended to the remainder tangent points $\{\mathbf{p}_i\}$ one by one. We will discuss the existence of the tangent approximation B-spline curve and the corresponding approximation effect after introducing a theorem with which one more tangent point can be used in the extending process.

Fig. 6 shows a C-shape case. As shown in Fig. 6, the solid curve in black is the approximation B-spline curve $\mathbf{C}_i(u)$, the dashed curve in black is the extended segment to point $\mathbf{p}_{i+1}$, the dashed line in green is the tangent line of the given curve at $\mathbf{p}_{i+1}$, which intersects with the dotted quadratic Bézier curve in blue at point $\mathbf{n}$. The point $\mathbf{m}$ is a new tangent point of the given curve which is between point $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$. When the approximation curve is extended to point $\mathbf{m}$, we obtain the new approximation curve $\mathbf{C}_j(u)$, the corresponding quadratic Bézier curve is a dash-dotted one in red with its control points $\mathbf{Q}_0^j$, $\mathbf{Q}_1^j$ and $\mathbf{Q}_2^j$, respectively. We have then the following theorem.

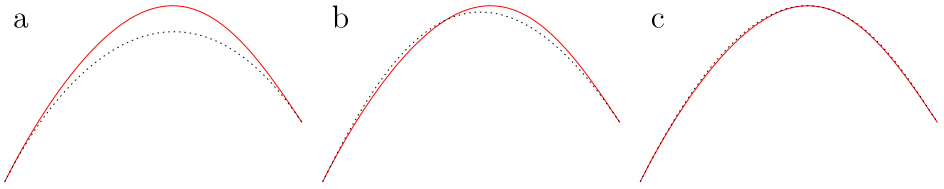

Fig. 4. Three cases of optimal approximation order: (a) 4: without inner intersection point; (b) 5: with one inner intersection point; and (c) 6: with one tangent inner point or two inner intersection points. The solid curve in red and the dotted curve in black are the given curve and the approximation curve, respectively.

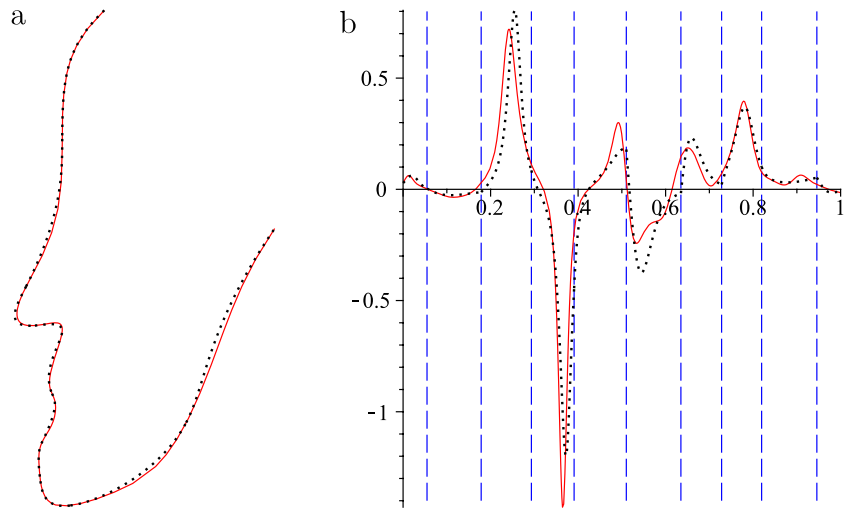

Fig. 5. Illustration for selecting the tangent points: (a) the solid curve in red and the dotted curve in black are the given curve and the approximation curve, respectively; and (b) the parameters of the tangent points which are shown by dotted lines, and the corresponding curvature plots of the two curves. The given curve is bounded by a box with sizes of 80 $\times$30.

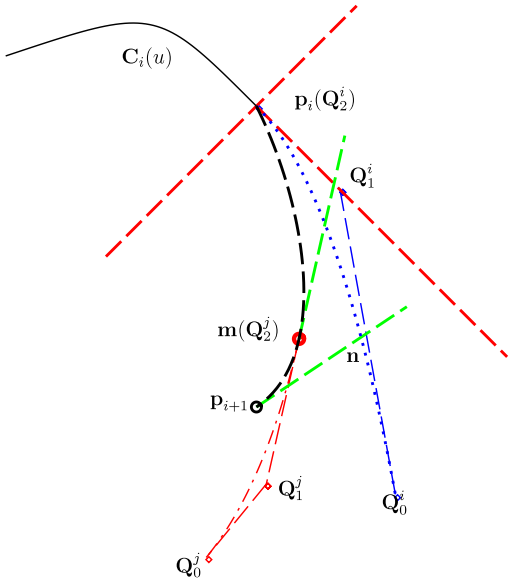

Fig. 6. Illustration of the existence of the tangent approximation B-spline curve while adding one more tangent point.

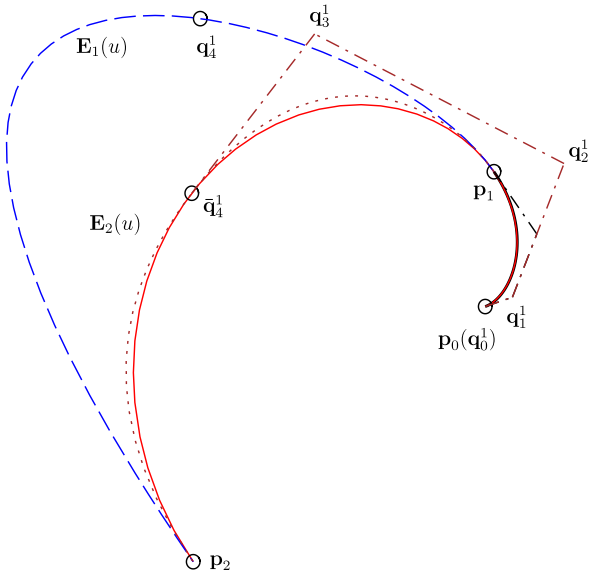

Fig. 7. Improve the approximation effect by adding one more tangent point. The solid curve in red is the given curve. The dashed curve in blue is obtained by directly extending the solid approximation curve in black to point $\mathbf{p}_2$, while the dotted curve in brown is obtained by adding a new inner point $\bar{\mathbf{q}}_4^1$.

Theorem 2. *There exists a new tangent point* $\mathbf{m}$ *of the given curve between* $\mathbf{p}_i$ *and* $\mathbf{p}_{i+1}$ *such that the approximation curve* $\mathbf{C}_i(u)$ *can be extended to points* $\mathbf{m}$ *and* $\mathbf{p}_{i+1}$ *where the approximation curve and the given curve are tangent with each other.*

Proof. Since the segment of the given curve between points $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$ is with C-shape, following the convexity of the given curve and for any tangent point $\mathbf{m}$ of the given curve between $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$, the tangent line of the given curve at $\mathbf{m}$ intersects with the dotted quadratic Bézier curve in blue at a point between $\mathbf{n}$ and $\mathbf{p}_i$, which means that the approximation $\mathbf{C}_i(u)$ can be extended to point $\mathbf{m}$ and is tangent with the given curve at point $\mathbf{m}$. Suppose that point $\mathbf{m}$ is close enough to point $\mathbf{p}_{i+1}$, such that the tangent line at point $\mathbf{p}_{i+1}$ intersects with both the line segments $\mathbf{Q}_2^j\mathbf{Q}_1^j$ and $\mathbf{Q}_2^j\mathbf{Q}_0^j$. Following the convex hull property of a Bézier curve, the tangent line at point $\mathbf{p}_{i+1}$ intersects with the dash-dotted quadratic Bézier curve in red whose control points are $\mathbf{Q}_0^j$, $\mathbf{Q}_1^j$ and $\mathbf{Q}_2^j$, which means that the approximation curve can also be extended to point $\mathbf{p}_{i+1}$ such that the new approximation curve and the given curve are also tangent with each other at point $\mathbf{p}_{i+1}$. Thus, we have completed the proof. □

We now show that the local approximation can be substantially improved by adding one more new tangent point. In Fig. 7, the solid curve in red is the given curve, and the solid segment in black is the approximation curve $\mathbf{C}_1(u)$ between points $\mathbf{p}_0$ and $\mathbf{p}_1$. The dashed segment in blue is obtained by directly extending to point $\mathbf{p}_2$, and the dotted curve in brown is obtained by adding a new inner point $\bar{\mathbf{q}}_4^1$ of the given curve. The dashed segment $\mathbf{E}_1(u)$ in blue, which is between points $\mathbf{p}_0$ and $\mathbf{q}_4^1$, is defined by control points $\mathbf{q}_0^1$, $\mathbf{q}_1^1$, $\mathbf{q}_2^1$, $\mathbf{q}_3^1$ and $\mathbf{q}_4^1$ with knot vector $\hat{\mathbf{U}} = \{0, 0, 0, 0, 1, u_t, u_t, u_t, u_t\}$, while extending the approximation curve $\mathbf{C}_1(u)$ to point $\bar{\mathbf{q}}_4^1$, we obtain the dotted curve $\mathbf{E}_2(u)$ in brown with control points $\mathbf{q}_0^1$, $\mathbf{q}_1^1$, $\mathbf{q}_2^1$, $\mathbf{q}_3^1$ and $\bar{\mathbf{q}}_4^1$ and with the same knot vector $\hat{\mathbf{U}}$. The only difference between segments $\mathbf{E}_1(u)$ and $\mathbf{E}_2(u)$ is just their last control point, which is $\mathbf{q}_4^1$ and $\bar{\mathbf{q}}_4^1$, respectively. As shown in Fig. 7, point $\bar{\mathbf{q}}_4^1$ is on the given curve, and the approximation using $\mathbf{E}_2(u)$ is obviously better than that using curve $\mathbf{E}_1(u)$. In conclusion, the dotted segments in brown is obtained by adding one more tangent point $\bar{\mathbf{q}}_4^1$, while the dashed segments in blue is obtained by directly extending the approximation curve $\mathbf{C}_1(u)$ to point $\mathbf{p}_2$. It is clear that the approximation to the given curve using the dotted segments in brown is much better than that using the dashed segments in blue.

We also provide the concept of the minimum shape deformation angle (*MSDA*) of an inner point for improving the selection of the new added tangent point between $\mathbf{p}_1$ and $\mathbf{p}_2$. As shown in Fig. 8(a), the solid curve in red is the given curve segment, $\mathbf{p}_1$ and $\mathbf{p}_2$ are the two end points of the segment, $\mathbf{m}$ is an inner point of the given curve, $\mathbf{L}_1$, $\mathbf{L}_2$ and $\mathbf{L}_3$ are the tangent line segments of the given curve at points $\mathbf{p}_1$, $\mathbf{m}$ and $\mathbf{p}_2$, respectively. In Fig. 8(b), $\{\alpha_i\}_{i=1}^4$ are four inner angles determined by the tangent line segments $\mathbf{L}_1$, $\mathbf{L}_2$, $\mathbf{L}_3$ and points $\mathbf{p}_1$, $\mathbf{m}$, $\mathbf{p}_2$. The minimum shape deformation angle of $\mathbf{m}$ is determined as

$$MSDA(\mathbf{m}) = \min\{\alpha_i | i = 1, 2, 3, 4\}.$$

To insert an inner point in an interval $[u_s, u_e]$ of the given curve $\mathbf{D}(u)$, we try to find a parameter $\check{u} \in (u_s, u_e)$ such that

$$MSDA(\mathbf{D}(\check{u})) = \max_{u\in[u_s,u_e]} MSDA(\mathbf{D}(u)).$$

In other words, when selecting $\mathbf{m}$ for insertion, we try to maximize the minimum shape deformation angle of $\mathbf{m}$. To simplify the computation, we take an average sample of several points in the interval $[u_s, u_e]$, and select the one with the maximum value of *MSDA* as the resulting approximate solution.

When the given curve $\mathbf{D}(u)$ is a circular arc, for example, the value of

$$MSDA(\mathbf{D}(u))$$

reaches its maximum at the middle of the parameter interval, which is intuitively the best selection. In principle, the *MSDA* takes both the positions and the tangent vectors into account and makes a trade-off selection. Fig. 9 illustrates the effectiveness of the proposed selection method. In Fig. 9, the solid curve in red is the given curve, and the dotted curve in black and the dashed curve in brown are obtained by using inner points $\mathbf{m}_1$ and $\mathbf{m}_2$, respectively. It can be verified that $MSDA(\mathbf{m}_2) > MSDA(\mathbf{m}_1)$, and the corresponding Hausdorff distance by using $\mathbf{m}_2$ is smaller than that of $\mathbf{m}_1$.

7. Two applications of the new method

This section discusses two applications of the new method. The first one is to recover a cubic B-spline curve from a set of sampled points and their normal vectors. The other one is to approximate a curve with straight line segments.

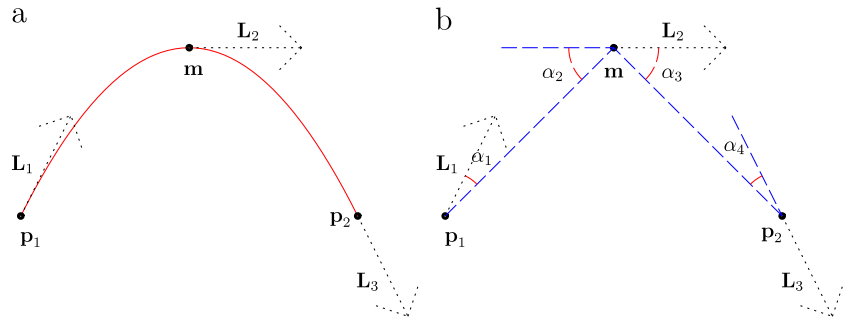

Fig. 8. Illustration of the minimum shape deformation angle of an inner point **m** between $\mathbf{p}_1$ and $\mathbf{p}_2$: (a) three points and their derivative vectors; and (b) the minimum shape deformation angle of the inner point **m**.

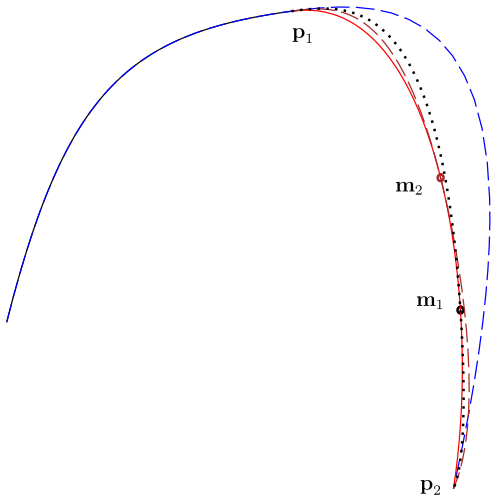

Fig. 9. The dotted segment in black and the dashed segment in brown are obtained by using inner points $\mathbf{m}_1$ and $\mathbf{m}_2$, respectively. It can be verified that $MSDA(\mathbf{m}_2) > MSDA(\mathbf{m}_1)$, and the corresponding Hausdorff distance by using $\mathbf{m}_2$ is smaller.

7.1. Recovering a cubic B-spline curve

Assume a set of ordered points $\{\mathbf{p}_i\}_{i=0}^n$ and their tangent directional vectors $\{\mathbf{t}_i\}$, which are sampled from a cubic B-spline curve. We assume that the set of points include knot points (positions) whose parameters are the knot values. It is also assumed that there are at least two different sampled points between two adjacent knot points. We now prove that the new method can recover the given cubic B-spline curve.

Lemma 1. *The first segment of the given cubic B-spline curve can be recovered by the inner point interpolation method.*

Proof. From the assumption, the first four points $\{\mathbf{p}_i\}_{i=0}^3$ are on the first segment of the cubic spline curve, which is essentially a cubic Bézier curve. We use points $\mathbf{p}_0$ and $\mathbf{p}_3$ as the two end points, and point $\mathbf{p}_1$ as the inner point. From the inner point interpolation method, we obtain a cubic univariate polynomial equation, which has at most three different real roots. Then, we also obtain at most three different cubic Bézier curves, which includes the first segment. Usually, there is a unique cubic Bézier curve interpolating both the positions and the tangent directional vectors of the four points $\{\mathbf{p}_i\}_{i=0}^3$, the one that also interpolates both point $\mathbf{p}_2$ and its tangent directional vector $\mathbf{t}_2$ is denoted as $\bar{\mathbf{C}}_1(u)$, which is the best choice.

Next, we show a simple method to find the first knot point, which is also the end point of the first segment. We extend the cubic Bézier curve $\bar{\mathbf{C}}_1(u)$ by using the curve unclamping technique to obtain two new curves $\mathbf{C}_n^1(t)$ and $\mathbf{C}_n^2(t)$ interpolating both the point $\mathbf{p}_k$ and its tangent directional vector $\mathbf{t}_k$. If there is no knot point between $\mathbf{p}_3$ and $\mathbf{p}_k$, at least one of the two new curves is essentially a Bézier curve and is at least C^4-continuous at $\mathbf{p}_3$. Otherwise, if both $\mathbf{C}_n^1(t)$ and $\mathbf{C}_n^2(t)$ are not C^4-continuous at $\mathbf{p}_3$, it means that there is at least one knot point between $\mathbf{p}_3$ and $\mathbf{p}_k$. The minimum k such that point $\mathbf{p}_k$ is a knot point can be found by the binary search method.

Finally, we extend the curve $\bar{\mathbf{C}}_1(u)$ to the first knot point by using the curve unclamping technique and obtain the resulting cubic Bézier curve, which is just the first segment of the given cubic B-spline curve. We have thus completed the proof. □

Lemma 2. *The second segment of the given cubic B-spline curve can be recovered from the first segment $\bar{\mathbf{C}}_1(u)$ by using the curve unclamping technique.*

Proof. Suppose the first knot point is $\mathbf{p}_k$. From the assumption, both point $\mathbf{p}_{k+1}$ and point $\mathbf{p}_{k+2}$ are the inner points of the second segment. We extend $\bar{\mathbf{C}}_1(u)$ to point $\mathbf{p}_{k+2}$ and obtain two possible new curves $\mathbf{C}_n^1(t)$ and $\mathbf{C}_n^2(t)$. And then, we select the one which interpolates both the point $\mathbf{p}_{k+1}$ and its tangent directional vector $\mathbf{t}_{k+1}$. The segment between point $\mathbf{p}_k$ and $\mathbf{p}_{k+2}$ is essentially a cubic Bézier curve and denoted as $\bar{\mathbf{C}}_2(u)$. Similarly, by using the method mentioned in Lemma 1, we can find the second knot point. By extending $\bar{\mathbf{C}}_2(u)$ to that knot point, we finally recover the second segment. This completes the proof of the lemma. □

Remark 1. Suppose that the curve $\mathbf{C}_n^1(t)$ is obtained by extending $\bar{\mathbf{C}}_1(u)$ to point $\mathbf{p}_{k+1}$ and point $\mathbf{p}_{k+2}$ in order, in which the knot points $\mathbf{p}_k$, $\mathbf{p}_{k+1}$ and $\mathbf{p}_{k+2}$ are of parameters $\bar{u}_0$, $\bar{u}_1$ and $\bar{u}_2$, respectively. Also suppose that the curve $\mathbf{C}_n^2(t)$ is obtained by directly extending $\bar{\mathbf{C}}_1(u)$ to point $\mathbf{p}_{k+2}$, in which the knot points $\mathbf{p}_k$ and $\mathbf{p}_{k+2}$ are of parameters $\hat{u}_0$ and $\hat{u}_1$, respectively. Assume that $\mathbf{C}_n^1(t)$ is essentially the same as $\mathbf{C}_n^2(t)$, then point $\mathbf{p}_{k+1}$ is a point on the curve $\mathbf{C}_n^2(t)$ of parameter

$$\frac{\bar{u}_1 - \bar{u}_0}{\bar{u}_2 - \bar{u}_0}(\hat{u}_1 - \hat{u}_0) + \hat{u}_0.$$

We have now the following theorem.

Theorem 3. *All of the segments of the given cubic B-spline curve can be recovered one by one. And thus, the given cubic B-spline curve is finally recovered.*

Proof. Following Lemmas 1 and 2, the first two segments can be recovered. From Lemma 2, the remainder segments of the given curve can also be recovered in order. From each unclamping process, both the knot vector and the control points are recovered. When all the segments are recovered, we obtain the resulting cubic B-spline curve. In principle, the resulting curve is just the given cubic B-spline curve. This completes the proof of the theorem. □

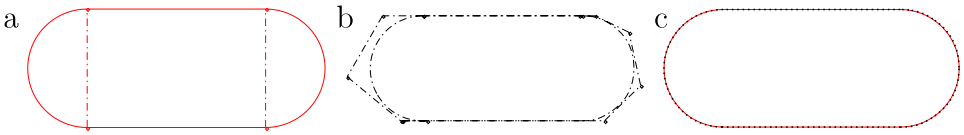

Fig. 10. Example for approximating a curve with straight line segments: (a) the given curve; (b) the resulting approximation curve and all of its 13 control points as well; and (c) the solid given curve in red and the dotted resulting approximation curve in black.

7.2. Approximating a curve with straight line segments

The curve used in this section is G^1-continuous and consists of two line segments and two circular arcs. The final approximation results are shown in Fig. 10(b ∼ c). As shown in Fig. 10(c), the dotted approximation curve in black almost coincides with the solid given curve in red. Suppose that we have obtained a B-spline curve $\mathbf{C}_1(t)$ approximating one of the circular arcs. Then we show how to approximate a line segment. The details are as follows. For any point $\mathbf{p}$ picked on the line segment, the tangent directional vector is always $\mathbf{t} = (1, 0)^T$. Thus, the equation

$$(\mathbf{q}_n^i - \mathbf{p}) \times \mathbf{t} = 0$$

has a unique root $u = 1$, where $\mathbf{q}_n^i$ is determined in Eq. (3). In this case, the curve $\mathbf{C}_1(t)$ is always itself after the unclamping process. Suppose that $\mathbf{C}_1(t)$ has control points $\{\mathbf{q}_i\}_{i=0}^k$ and knot vector

$$\mathbf{U} = \{0, 0, 0, 0, u_4, \ldots, u_k, 1, 1, 1, 1\}.$$

Let

$$\bar{\mathbf{q}}_{k+1} = \mathbf{q}_k, \qquad \bar{\mathbf{q}}_k = \mathbf{q}_k - \varepsilon_1 \mathbf{t}, \qquad \bar{\mathbf{q}}_i = \mathbf{q}_i, \ldots i = 0, \ldots, k-1,$$

where ε_1 is a small positive real number. We modify $\mathbf{C}_1(t)$ to be a new curve $\bar{\mathbf{C}}_1(t)$ with control points $\{\bar{\mathbf{q}}_i\}_{i=0}^{k+1}$ and knot vector

$$\bar{\mathbf{U}} = \{0, 0, 0, 0, u_4, \ldots, u_k, 1 - \varepsilon, 1, 1, 1, 1\},$$

where ε is a small positive real number. It can be verified that the first $k - 3$ segments of curve $\bar{\mathbf{C}}_1(t)$ are the same as those of curve $\mathbf{C}_1(t)$, and $\bar{\mathbf{C}}_1(t)$ is still tangent with the original curve at the end point $\mathbf{q}_k$. The four points $\bar{\mathbf{q}}_{k-1}$,$\bar{\mathbf{q}}_k$,$\bar{\mathbf{q}}_{k+1}$ and $\mathbf{p}$ are on the same line determined by $\mathbf{p}$ and the directional vector $\mathbf{t}$. Thus, by using the unclamping technique with some knot u_t, the $\bar{\mathbf{C}}_1(t)$ can be extended to a new curve interpolating both the point $\mathbf{p}$ and its tangent directional vector $\mathbf{t}$, and the last segment of the new curve is a straight line segment.

Remark 2. The segment with the four control points $\bar{\mathbf{q}}_{k-1}$,$\bar{\mathbf{q}}_k$,$\bar{\mathbf{q}}_{k+1}$ and $\mathbf{p}$ can represent a straight line. However, the distribution of the control points may be uneven, which may cause a saltation of curvature of the next segment at point $\mathbf{p}$. This can be improved by inserting more control points for the straight line segment nearby the point $\mathbf{p}$.

Remark 3. Note that the resulting approximation curve in Fig. 10 (b) is not symmetric. When a given curve is a symmetric one, suppose that $\bar{\mathbf{Q}}$ is an intersection point between the given curve and the symmetric axis. To obtain a symmetric one, one can select symmetric tangent points on both sides of the symmetric axis and the points on the symmetric axis as well, and then constrain the second order derivative vector of the approximation curve at point $\bar{\mathbf{Q}}$ has the same the direction as that of the symmetric axis. Finally, the segments on both sides of the symmetric axis can be merged into a symmetric B-spline curve at point $\bar{\mathbf{Q}}$. Here the detail is omitted.

8. Further examples and comparisons

In some practical applications in CAD/CAM, or even font reconstruction, some key points may be required on the approximation curve. When the data points are determined, the control points can be directly solved by standard interpolation method, while its approximation effect needs to be improved further. Usually, the new method is able to accurately interpolate the data points and has a better approximation effect than that of the standard interpolation method. If it is not required to apply exact interpolation constraints, most of the previous methods use least-squares B-spline curve fitting technique and produce much better approximation than the interpolation method [23,11,1,24,25]. However, for some practical requirements, some key points must be accurately interpolated, which cannot be ensured in the least-squares method (denoted as *LSM*). Note that the least squares method is to minimize the function $E(\mathbf{b}_0, \ldots, \mathbf{b}_n)$ in Eq. (1), the best selection of parameter $\{t_i\}$ should satisfy the claim that point $\mathbf{C}(t_i)$ is the closest point to $\mathbf{p}_i$. Though there are several notable works on the parameterization of $\{t_i\}$ [3,5,4], it still remains a challenge to this problem, especially in the cases when the number of the control points is small. The new approach is easy to ensure that the key points are on the approximation curve. It interpolates the given curve (or the sampled points) in both position and direction of the derivative vector at key positions. Numerical examples of planar curve approximation show the corresponding approximation effect of the new method.

Firstly, we compare the least squares method (*LSM*) with the new method. Each extended segment in the new method interpolates both the positions and the directions of the tangent vectors at two end points, as shown in Fig. 4. It usually has the approximation order 4, 5, or 6, while the *LSM* method usually has the approximation order at most 4. In the optimization opinion, the new method makes a local optimization on the parameter or the knot by using both the positions and the tangents. Numerical examples also show that the new method is able to produce a better approximation than the *LSM* method, even with less control points. Fig. 11 shows a face-shape curve and its resulting approximation curves by using different methods. The solid curve in red is the given curve, the dashed one in green and the dotted one in black are from the least-squares method and the new method, which have 19 and 16 control points, respectively. Fig. 11(b) shows the resulting error plots. In Fig. 11(b), the dashed one in green and the dotted one in black are from the *LSM* method and the new method, respectively. The maximum errors produced by the *LSM* method and the new method are 0.192 and 0.120, respectively. The corresponding average errors produced by the *LSM* method and the new method are 0.0432 and 0.0352. In this case, the approximation using the new method is much better than that using the *LSM* method.

Fig. 12 shows two other examples. Fig. 12(a) and (c) show the original curve and its approximation curves, while Fig. 12(b) and (d) show the corresponding error plots. In these two examples, both of the two larger dimensions of their bounding boxes are 1. The original curve is in red, while the curves in green and in black are the resulting curves from the *LSM* method and the new method, respectively. The two frames at the right which are in blue and in brown provide a local scale-up view shown in the figure. In Fig. 12(a), the numbers of the control points of curves in green and in black are 18 and 13, respectively, while the resulting curves in Fig. 12(c) are of 29 and 20 control points. As shown in Fig. 12(b) and (d), the resulting curve in black which is from the new method has a better approximation effect. Compared with the *LSM* method,

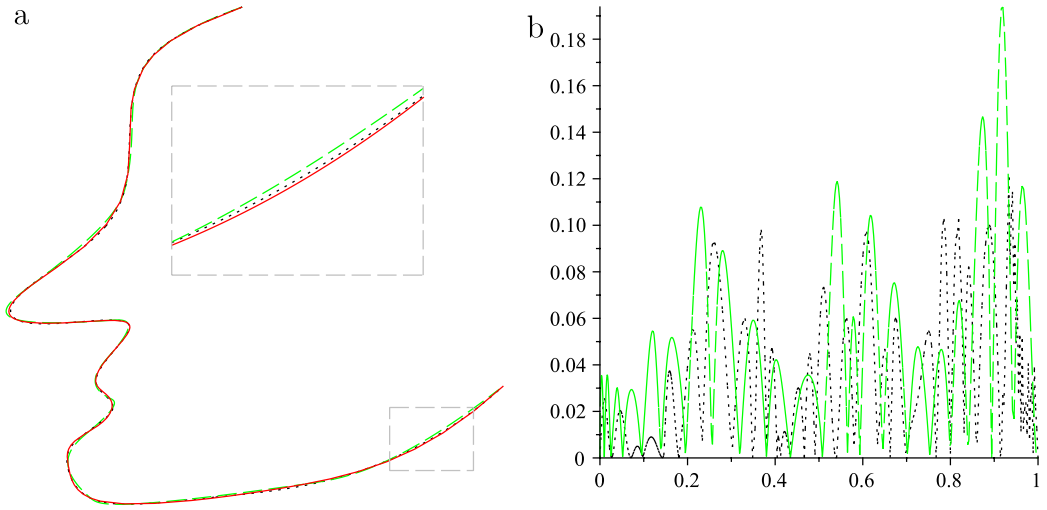

Fig. 11. Results from different methods: (a) the resulting curves with a local magnified view; and (b) the error plots. The solid curve in red is the given curve, while the dashed curve in green and the dotted curve in black are from the *LSM* method and the new method, respectively. The larger dimension of the bounded box of the given curve is 80.

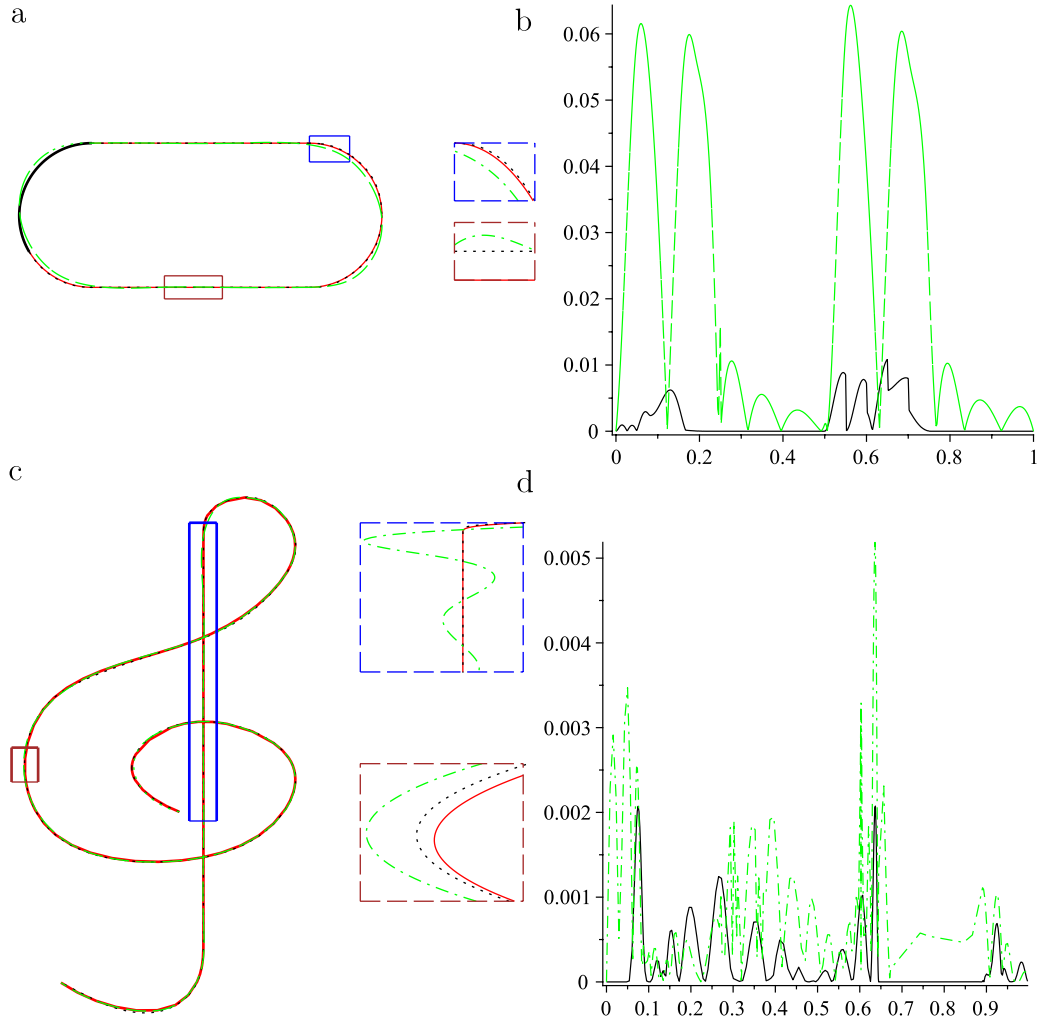

Fig. 12. Examples of curves with a straight line segment: (a) a running path shape curve and its approximation curves with local magnified views; (b) the corresponding error plots; (c) a curve of a stave note shape and its approximation curves with local magnified views; and (d) the corresponding error plots. The solid given curve is in red, while the dashed curve in green and the dotted one in black are the resulting curves from the *LSM* method and the new method, respectively.

the new method may have three advantages: (1) it is able to accurately represent the line segments; (2) the error distance between the given curve and the approximation curve can be much less than that of the *LSM* method, even if it uses a much smaller number of control points; (3) it does accurately interpolate the dominant points.

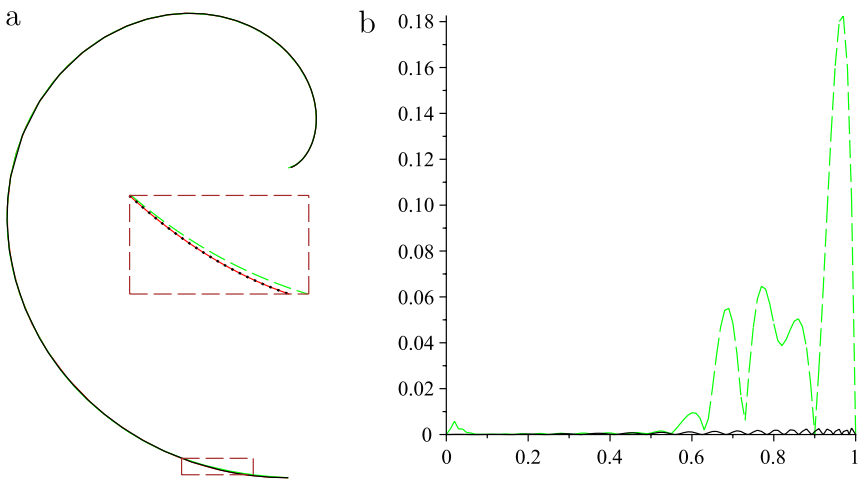

Fig. 13. Example of an involute curve: (a) the resulting curves with a local magnified view; and (b) the corresponding error plots. The solid curve in red is the given curve, while the dashed curve in green and the dotted curve in black are the results from the method in [10] and the new method, respectively.

Secondly, we compare the interpolating method in [10] with the new method. In many cases, there is no curve with n control points accurately interpolating both the positions and the normal vectors of n data points. For example, known from the inner point interpolation method, there are at most three possible cubic Bézier curves interpolating both the positions and the normal vectors of 3 points in many cases, and there is no cubic Bézier curve interpolating both the positions and the normal vectors of 4 points. The method in [10] is usually an approximate interpolation method. The resulting curve of the new method has $n+1$ control points and it is able to exactly interpolate both the positions and the normal vectors of n points. Examples show that the approximation effect to a given curve in these two methods is similar. In principle, the computation time of the method in [10] is $(n-2)\bar{t}_m K$, where n is the number of data points, K is the number of the iterative steps, and $\bar{t}_m$ is the computation time for computing the closest point on the approximation B-spline curve to a data point, which usually needs to solve a quintic polynomial equation several times for the cubic case. With the new method, no iterative step is required. The total computation time is $(n-3)t_m + t_k$, where n is the number of the given points, t_m is the computation time for each unclamping process with one point which needs to solve a quadratic univariate polynomial equation one time, t_k is the computation time of the inner point interpolation method in [18] for solving the first cubic Bézier curve, which needs to solve a cubic univariate polynomial equation one time. Note that the roots of a quadratic or cubic univariate polynomial equation can be explicitly expressed from the formulae and there is no analytic formula for solving a quintic polynomial equation; both the computation times t_m and t_k are usually less than $\bar{t}_m$. Fig. 13 shows an example of an involute curve determined by

$$x(t) = 9(\cos t + t\sin t), \qquad y(t) = 9(\sin t - t\cos t),$$
$$t \in [\pi/10, 2\pi].$$

The data points are sampled at parameters $\{i/10\pi\}_{i=1}^{20}$. The curves in green and in black are obtained by the method in [10] and the new method, and they have 20 and 21 control points, respectively. Fig. 13(a) shows the resulting curves from both the method in [10] and the new method. In principle, the curve in green does not accurately interpolate both of the positions and the normal vectors at the data points, while the new method does in this case. As shown in Fig. 13(b), the maximum errors in the method in [10] and the new method are 0.18 and 0.003, respectively. It is clear that the new method has a better approximation effect than that of the method in [10]. Fig. 14 shows an epitrochoid curve in [26] parameterized as

$$x(t) = (a+b)\cos(2\pi t) - h\cos\left(2\frac{a+b}{b}\pi t\right),$$
$$y(t) = (a+b)\sin(2\pi t) - h\sin\left(2\frac{a+b}{b}\pi t\right),$$

where $a = 10$, $b = 1$ and $h = 2$. The curvature of an epitrochoid curve is more complicated than that of an involute curve. There are 101 points sampled at $\{i/100\}_{i=0}^{100}$ and their normal vectors as well. In this case, the new method can exactly interpolate both the positions and the normal vectors of the data points directly by the curve unclamping technique, and the resulting cubic B-spline curve has 102 control points. The corresponding error plots from both the method in [10] and the new method show, in this case, that the method in [10] has a better approximation effect than that of the new method. In principle, the method in [10] can meet an arbitrary small error at the given points, and the new method can accurately interpolate both the positions and the directional vectors of the tangent vectors of the given points. The new method is able to give the explicit formula for computing the resulting curve and usually requires less computation time than that of the method in [10].

In some cases (e.g., for the 3D case) the new method cannot accurately interpolate all of the positions and the normal vectors of the data points. To avoid this, one can slightly change the positions or the normal vectors of some data points for the curve unclamping process. Another possible way for improving the approximation effect of the resulting curve of the new method is to only interpolate the positions of the points. Once the position of the extended point is chosen (e.g., $\mathbf{D}(\bar{u}_{i+1})$) the extended segment of the approximation curve is essentially a cubic Bézier curve, whose control points depend on the parameter u in Eq. (3). And then, the alternative method leads to a local optimization problem, which can be computed by minimizing a univariate equation

$$H(u) = \int_{u_i}^{\bar{u}_{i+1}} \|\mathbf{C}_u(\phi(t)) - \mathbf{D}(t)\|^2 \mathrm{d}t,$$

where $\mathbf{C}_u(t)$ is the extended curve segment depending on the parameter u, and $\phi(t)$ is a linear function such that $\phi(\bar{u}_i) = 1$ and $\phi(\bar{u}_{i+1}) = u$.

9. Conclusions

This paper discusses the cubic B-spline approximation problem and presents a simple but efficient method based on the curve unclamping technique. It interpolates both the positions and the directions of the derivative vectors of the points, and at the same

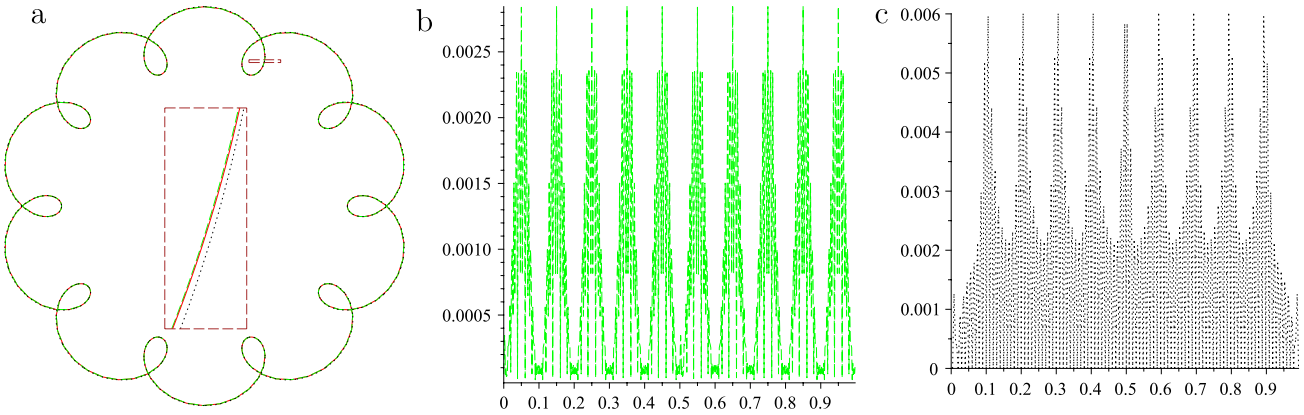

Fig. 14. Example of an epitrochoid curve: (a) the resulting curves with a local magnified view; (b) the resulting error plot from the method in [10]; and (c) the resulting error plot from the new method.

time it tends to approximate the curvature as well. The first segment is solved by the inner point interpolation method in the 2D case or the Geometric Hermite Interpolation method in the 3D case, and the remainder segments are obtained by successive unclamping processes. For each unclamping process, there is a need to solve a quadratic univariate polynomial equation for determining the unclamping parameter u_t such that the resulting curve interpolates both the position and the direction of the tangent vector at the new data point. In principle, the new method is able to recover a cubic B-spline curve. Examples show that it can well approximate a curve with straight line segments and can also accurately represent the line segments.

The new method has a local approximation property. Given an error bound or a tolerance, it can adaptively approximate the data points or the given curve segment by segment. The new method usually has a better approximation than the standard interpolation method. When the number of the control points is small, the new method can also have a better approximation than that of the *LSM* method. Given n suitable data points and their normal vectors, the new method is able to produce a resulting cubic B-spline curve with $n+1$ control points that interpolates both of the positions and the normal vectors of the n points. However, with the new method, there is room to further improve curvature approximation compared with the method of [10]. In some cases, the procedure for knots placement can also be further improved.

As for future work, there is still plenty of room for further development. The current solution for selecting the next data point and its tangent vector as well for the unclamping process is not an optimal solution yet. In some practical cases, there may also be constraints for knots placement. Can the new method produce a cubic B-spline curve such that the error is bounded and the knot vector of the resulting B-spline curve meets the given constraints? With the new method, it should also be possible to use a B-spline curve of any degree, not just the cubic case, for curve approximation, which should be further studied in future work as well.

Acknowledgements

The research was partially supported by the Research Grants Council of Hong Kong Special Administrative Region, China (CityU 1186/07E), the National Science Foundation of China (60803076, 60625202, 60911130368) and the Science Foundation of Zhejiang Province (Y1090004). The authors also wish to thank the anonymous reviewers for their helpful comments and suggestions.

Appendix. The inner point interpolation method

Suppose that the given curve $\mathbf{C}(u)$ has two end points $\mathbf{Q}_0$ and $\mathbf{Q}_1$, and two derivative vectors $\mathbf{T}_0$ and $\mathbf{T}_1$, respectively. Under the tangent constraint, the control points $\{\mathbf{P}_i\}$ of the approximation cubic Bézier curve must satisfy

$$\mathbf{P}_0 = \mathbf{Q}_0, \qquad \mathbf{P}_3 = \mathbf{Q}_1, \qquad \mathbf{P}_1 = \mathbf{P}_0 + \alpha\mathbf{T}_0,$$
$$\mathbf{P}_2 = \mathbf{P}_3 - \beta\mathbf{T}_1. \tag{6}$$

Thus, the remaining work is to determine the values of α and β.

Let $\mathbf{C}(0) = (p_0, q_0)^T$,$\mathbf{C}(1) = (p_1, q_1)^T$, $\mathbf{C}'(0) = (s_0, t_0)^T$, $\mathbf{C}'(1) = (s_1, t_1)^T$, $X_C'(u_1) = m_i$,$Y_C'(u_1) = n_i$, $X_C(u_1) = \delta_i$,$Y_C(u_1) = \theta_i$. We ensure that the approximation Bézier curve and the given curve $\mathbf{C}(u)$ are tangent with each other at the point $\mathbf{C}(u_1)$. For the sake of convenience (if necessary, under a suitable transformation of the coordinate system), suppose that

$$(p_0, q_0) = (0, 0) \quad \text{and} \quad (s_0, t_0) = (0, 1). \tag{7}$$

By selecting one inner point with parameter u_1, we obtain the following system of equations

$$\begin{cases} a_1(v_1)\alpha + b_1(v_1)\beta + c_1(v_1) - \delta_1 = 0, \\ a_2(v_1)\alpha + b_2(v_1)\beta + c_2(v_1) - \theta_1 = 0, \\ (a_1'(v_1)\alpha + b_1'(v_1)\beta + c_1'(v_1))n_1 \\ \quad -(a_2'(v_1)\alpha + b_2'(v_1)\beta + c_2'(v_1))m_1 = 0, \end{cases} \tag{8}$$

where $c_1(v_1) = p_0(B_3^0(v_1)+B_3^1(v_1))+p_1(B_3^2(v_1)+B_3^3(v_1))$, $a_1(v_1) = s_0B_3^1(v_1)$, $b_1(v_1) = -s_1B_3^2(v_1)$, $c_2(v_1) = q_0(B_3^0(v_1) + B_3^1(v_1)) + q_1(B_3^2(v_1) + B_3^3(v_1))$, $a_2(v_1) = t_0B_3^1(v_1)$ and $b_2(v_1) = -t_1B_3^2(v_1)$.

The first two equations in the system of Eq. (8) are linear in α and β. Solving from these two equations, we obtain

$$\begin{cases} \alpha = \dfrac{b_1(v_1)(c_2(v_1) - \theta_1) - b_2(v_1)(c_1(v_1) - \delta_1)}{b_2(v_1)a_1(v_1) - b_1(v_1)a_2(v_1)}, \\ \beta = \dfrac{a_2(v_1)(c_1(v_1) - \delta_1) - a_1(v_1)(c_2(v_1) - \theta_1)}{b_2(v_1)a_1(v_1) - b_1(v_1)a_2(v_1)}. \end{cases} \tag{9}$$

Substituting Eq. (9) into the third equation in the system of Eq. (8), we obtain

$$H(v_1) = (r_3v_1^3 + r_2v_1^2 + r_1v_1 + r_0),$$

where $r_3 = (-t_0s_1q_1 + t_0s_1q_0 - s_0t_1q_1 + s_0t_1q_0 + 2t_1t_0p_1 - 2t_1t_0p_0)m_1 + (-p_1s_1t_0 + s_0t_1p_0 + p_0s_1t_0 - s_0t_1p_1 - 2s_0s_1q_0 + 2s_0s_1q_1)n_1$, $r_2 = (-3t_0(s_1q_0 - s_1q_1 + t_1p_1 - t_1p_0))m_1 + (3s_0(s_1q_0 - s_1q_1 + t_1p_1 - t_1p_0))n_1$, $r_1 = (-3(-q_0 + \theta_1)(t_0s_1 - s_0t_1))m_1 + (3(-p_0 + \delta_1)(t_0s_1 - s_0t_1))n_1$, $r_0 = (t_0s_1\theta_1 - t_1t_0p_0 + t_1t_0\delta_1 - t_0s_1q_0 + 2s_0t_1q_0 - 2s_0t_1\theta_1)m_1 + (s_0t_1\delta_1 - s_0t_1p_0 + s_0s_1\theta_1 - s_0s_1q_0 - 2\delta_1s_1t_0 + 2p_0s_1t_0)n_1$.

Suppose that $v^\star \in [0, 1]$ is the root of $r_3t^3 + r_2t^2 + r_1t + r_0$ which is the closest to parameter u_1. By substituting $v^\star$ instead of v_1 into Eq. (9), we obtain the values of α and β. Thus, we can determine the values of α and β by solving a univariate polynomial $r_3t^3 + r_2t^2 + r_1t + r_0$ whose degree is 3.

References

[1] Park H, Lee J-H. B-spline curve fitting based on adaptive curve refinement using dominant points. Computer-Aided Design 2007;39(6):439–51.
[2] Elber G, Grandine T. Hausdorff and minimal distance between parametric freeforms in $\mathbb{R}^2$ and $\mathbb{R}^3$. In: The 5th international conference of geometric modeling and processing. Lecture notes in computer science, 2008. p. 191–204.
[3] Hoschek J, Lasser D. Fundamentals of computer aided geometric design. London: AK Peters; 1993.
[4] Park H. An error-bounded approximate method for representing planar curves in B-splines. Computer Aided Geometric Design 2004;21(5):479–97.
[5] Piegl L, Tiller W. The NURBS book. second ed. New York: Springer-Verlag; 1997.
[6] Rogers D, Fog N. Constrained B-spline curve and surface fitting. Computer-Aided Design 1989;21(10):641–8.
[7] Sarkar B, Menq C. Parameter optimization in approximating curves and surfaces to measurement data. Computer Aided Geometric Design 1991;8(4):267–90.
[8] Wang W, Pottmann H, Liu Y. Fitting B-spline curves to point clouds by curvature-based squared distance minimization. ACM Transactions on Graphics 2006;25(2):214–38.
[9] Martin A, Zbynek S, Jüttler B. Evolution-based least-squares fitting using pythagorean hodograph spline curves. Computer Aided Geometric Design 2007;24(3):310–22.
[10] Gofuku S-i, Tamura S, Maekawa T. Point-tangent//point-normal B-spline curve interpolation by geometric algorithms. Computer-Aided Design 2009;41(4):412–22.
[11] Park H, Kim K, Lee S. A method for approximate NURBS curve compatibility based on multiple curve refitting. Computer-Aided Design 2000;32(4):237–52.
[12] Simon F, Michael H. Constrained curve fitting on manifolds. Computer-Aided Design 2008;40(1):25–34.
[13] Fang L, Gossard D. Multidimensional curve fitting to unorganized data points by nonlinear minimization. Computer-Aided Design 1995;27(1):48–58.
[14] Laurent-Gengoux P, Mekhilef M. Optimization of a NURBS representation. Computer-Aided Design 1993;25(11):699–710.
[15] Meek D, Ong B, Walton D. Constrained interpolation with rational cubics. Computer Aided Geometric Design 2003;20(3):253–75.
[16] Lyche T, Morken K. A data-reduction strategy for splines with applications to the approximation of functions and data. IMA Journal of Numerical Analysis 1988;8:185–208.
[17] Razdan A. Knot placement for B-spline curve approximation. Report, Arizona State University, 1999. http://citeseer.ist.psu.edu/398077.html.
[18] Chen X, Ma W, Yong J, Paul J. Inner point interpolation method with tangent direction constraint for planar curve approximation. Computer Aided Geometric Design 2010; (under review).
[19] Höllig K, Koch J. Geometric Hermite interpolation. Computer Aided Geometric Design 1995;12(6):567–80.
[20] Hu S, Tai C, Zhang S. An extension algorithm for B-splines by curve unclamping. Computer-Aided Design 2002;34(4):415–9.
[21] de Boor C, Hoillig K, Sabin M. High accuracy geometric Hermite interpolation. Computer Aided Geometric Design 1987;4:269–78.
[22] Höllig K, Koch J. Geometric Hermite interpolation with maximal order and smoothness. Computer Aided Geometric Design 1996;13(8):681–95.
[23] Li W, Xu S, Zhao G, Goh L. Adaptive knot placement in B-spline curve approximation. Computer-Aided Design 2005;37(8):791–7.
[24] Vassilev T. Fair interpolation and approximation of B-splines by energy minimization and points insertion. Computer-Aided Design 1996;28(9):753–60.
[25] Yoshimoto F, Harada T, Yoshimoto Y. Data fitting with a spline using a real-coded genetic algorithm. Computer-Aided Design 2003;35(8):751–60.
[26] Lawrence J. A catalog of special plane curves. New York: Dover Publications Inc; 1972.