SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

May 2018, Vol. 61 052104:1–052104:15
doi: 10.1007/s11432-017-9134-6

Formula for computing knots with minimum stress and stretching energies

Xuemei LI[1*], Fan ZHANG[2], Guoning CHEN[3] & Caiming ZHANG[1,4]

[1]*School of Computer Science and Technology, Shandong University, Jinan* 250101*, China;*
[2]*School of Computer Science and Technology, Shandong Technology and Business University, Yantai* 264005*, China;*
[3]*Department of Computer Science, University of Houston, Houston* TX77204*, USA;*
[4]*Shandong Provincial Key Laboratory of Digital Media Technology, Jinan* 250014*, China*

Received 9 January 2017/Revised 29 March 2017/Accepted 10 May 2017/Published online 1 September 2017

Abstract Computing knots for a given set of data points in a plane is one of the key steps in the construction of fitting curves with high precision. In this study, a new method is proposed for computing a parameter value (knot) for each data point. With only three adjacent consecutive data points, one may not determine a unique interpolation quadratic polynomial curve, which has one degree of freedom (a variable). To obtain a better curve, the stress and stretching energies are used to optimize this variable so that the quadratic polynomial curve has required properties, which ensure that when the three consecutive points are co-linear, the optimal quadratic polynomial curve constructed is the best. If the position of the mid-point of the three points lies between the first point and the third point, the quadratic polynomial curve becomes a linear polynomial curve. Minimizing the stress and stretching energies is a time-consuming task. To avoid the computation of energy minimization, a new model for simplifying the stress and stretching energies is presented. The new model is an explicit function and is used to compute the knots directly, which greatly reduces the amount of computation. The knots are computed by the new method with minimum stress and stretching energies in the sense that if the knots computed by the new method are used to construct quadratic polynomial, the quadratic polynomial constructed has the minimum stress and stretching energies. Experiments show that the curves constructed using the knots generated by the proposed method result in better interpolation precision than the curves constructed using the knots by the existing methods.

Keywords parameter, stress and stretching energies, polynomial, precision, curve construction

Citation Li X M, Zhang F, Chen G N, et al. Formula for computing knots with minimum stress and stretching energies. Sci China Inf Sci, 2018, 61(5): 052104, doi: 10.1007/s11432-017-9134-6

1 Introduction

One of the fundamental problems faced in computer-aided design, engineering, scientific computing, and computer graphics is the construction of curves with high precision and of an appropriate shape. For different applications, the curves and surfaces require high precision and distinct properties [1–6]. Various energy models, such as stress energy, strain energy, stretching energy, energy or length of a curve, are often used for optimization to ensure that the constructed curves/surfaces have the precision and requisite properties for a specific application. This paper focuses on computing the knots for constructing curves with high precision.

[*] Corresponding author (email: xmli@sdu.edu.cn)

Previous work. Before constructing a parametric curve, the parameter values, i.e., knots, of the data points should be known. We should also know how the choice of a parameterization method makes a significant impact on the resulting interpolated curve. For the same set of data, even with the same interpolation scheme, different parameterization methods will produce different curves. It is well-known that using uniform parameterization to choose knots generally leads to unsatisfactory results. Physically, this indicates particles placed at the same time intervals between any two points regardless of their distance. When the intervals between consecutive data points are uneven, it is difficult for the uniform parameterization method to produce satisfactory results. To overcome this drawback of the uniform parameterization method, several non-uniform parameterization strategies have been proposed. In particular, there are three popular parameterization techniques in use: the chord length method [7], Foley's method [8] and the centripetal method [9]. Among these methods, the chord length method has always been considered as the best method and has been most widely used. However, in terms of the approximation error, our experiments show that none of these methods demonstrate any advantage over the others. Besides, although these methods are widely used to construct parametric curves, in some cases, none of them can produce a satisfactory result.

Based on the work by Lee [9], Jeong et al. [10], parameters were estimated using tabulated parameter and length data. However, our experiments show that the Jeong's method [10] generally results in more errors than the original method proposed in [9]. In addition, among the chord length method, Foley's method [8] and the centripetal method [9], the centripetal method is the only method for parametric curve construction that can ensure that there will be no local self-intersections on the curve [11]. An analysis of the uniform method, chord length method, and centripetal method [9] conducted by Yuksel et al. [11] shows that centripetal parameterization produces more visually appropriate curves as compared to the curves created using uniform and chordal parameterizations. This study, however, does not provide any mathematical explanation for this result. Therefore, though the centripetal method appears to be the preferable one, the interpolating results of these methods do not always capture all the data features. Fang et al. [12] proposed a refined centripetal method to improve the wiggle deviation of the interpolation, especially for abrupt data interpolation.

Note that the interpolation precision of the aforementioned knot selection methods is only linear, which means that for the knots created by these methods, the resulting interpolation curve will be a linearly parameterized straight line if the data points are sampled from a straight line and will not be a quadratic polynomial curve if the data points are sampled from a quadratic polynomial curve. When the data points are sampled from a non-linear curve, e.g., a quadratic polynomial curve, a higher interpolation precision is required to reconstruct the underlying high-order curve [13, 14]. In the work by Zhang et al. [13], a global method for choosing knots was proposed. Using the chosen knots, the constructed interpolants reproduce parametric quadratic curves if the interpolation scheme reproduces quadratic polynomial curves. Recently, by extending the ideas proposed in [13], a method for choosing knots for parametric curve interpolation was introduced by Zhang et al. [14]. Even though this method employs a local knot selection process, it delivers quadratic precision. Therefore, from the perspective of approximation, the method in [14] is better than the chord length method, Foley's method, and the centripetal method. In addition, while the approximation error produced by the method in [14] is relatively small, it is also more practical than other existing methods. Studies by Hartley and Judd [15] and Martin [16] proposed a method of choosing knots through optimization. Some articles also discuss the parameterization problems of spatial data points. In the article by [17–20], the parameterized results are used to construct a parametric surface.

Parameterization for curve and surface construction is still an unresolved problem and has attracted considerable attention. Lü [7] showed that the rational cubic and quartic curves are suitable for G^1 Hermite interpolation. They identified a family of curves that can be parameterized using the chord length method, in which rational chord-length parameterizations are thoroughly investigated. Bastl et al. [21, 22] extended the property of chord length parameterization of quadratic rational curves studied in the work by Lü [7] to surfaces. It is shown that the ratios of the three distances of a point to the patch vertices and the ratios of the distances of the parameter point to the three vertices of the domain triangle are identical. Besides Hermite data, Tsuchie and Okamato [23] also introduced a method for fitting a high-quality planar curve for styling design data by using a G^2 quadratic B-spline curve. In order to attain G^2 continuity of the B-spline curve, a non-uniform knot vector is used, which enables the curve to be composed of fewer segments as compared to a uniform curve. To avoid solving a complicated nonlinear optimization problem, the control points and the knot vector of the B-spline curve are calculated separately. Han [24] also discussed geometric continuous splines in curve design. This study proposed a class of general quartic splines for a non-uniform knot vector. The generated quartic spline curves displayed C^2 continuity with three local adjustable shape parameters. Using a method different from the classical tensor product setting, [25] assigned a different parameter interval to each mesh edge, which allows interpolation of each section polyline with parameter values that can prevent wiggling and generate other interpolation artifacts in the resulting curve, yielding high-quality interpolating surfaces. The splines display C^2 continuity at simple knots and depict the cubic non-uniform B-spline as a special case. Based on the given splines, piecewise quartic spline curves with three local shape parameters are provided. This representation can be used to interpolate sets of points by fixing some values of the curve's parameters. As the shape parameters increase, the quartic spline curves move from the C^2 continuous interpolation curves to the cubic B-spline curves.

Proposed method. In this paper, a new method for computing knots is presented. The new method is derived based on the assumption that the given set of data points are taken from a parametric curve that can be approximated well by piecewise quadratic polynomial curve segments. In particular, the new method assumes that each curve segment between three adjacent points can be approximated by a quadratic polynomial. Because three adjacent consecutive data points are not sufficient for determining a unique interpolation quadratic polynomial curve, the quadratic curve is determined by optimizing its stress energy and stretching energies. This optimization lends appropriate properties to the resulting quadratic curve to ensure that when the three consecutive points are almost co-linear, the first derivative of the quadratic curve constructed is the most appropriate. For three co-linear data points, P_{i-1}, P_i and P_{i+1}, if P_i is between P_{i-1} and P_{i+1}, the quadratic curve $P_i(s) = P_{i-1} + (P_{i+1} - P_{i-1})s$ is the simplest quadratic polynomial passing three points with the first derivative being constant. If P_{i+1} is between P_{i-1} and P_i, the first derivative of the quadratic curve at P_i is zero, which is the most reasonable, the reason is that when a proton moves along a straight line, from P_{i-1} through P_i then back to P_{i-1}, the best case is that its velocity at point P_i is zero. Optimizing the stress and stretching energies is to solve a non-linear problem, which is a time-consuming task. In order to reduce the computation time, a new model is presented to simplify the process of optimizing the energy. The new model is an explicit function and is used to compute the knots directly with less computation. As the knots are determined by optimizing the quadratic curve, they can reflect the distribution of the data points well. Therefore, when used for curve construction, the resulting curve could have higher precision than the methods with linear precision. Our method is a local method, thus, it is easy to modify a curve interactively, consequently making the curve design process more efficient and flexible. Experiments show that approximation precision with our method is better than the ones proposed in [7–9, 12, 14, 22]. Moreover, as the knots are computed by a formula, our method is simpler to implement and easier to compute. Simulation results also show that our method produces smaller contour errors and lower feed-rate fluctuation compared to the other four representative methods and two new methods (Section 4).

2 Basic premise for the study

Let $P_i = (x_i, y_i),\ i = 1, 2, \ldots, n$, be a given set of data points. When constructing a parametric interpolation curve $P(t)$ to pass this set of data points, one needs to compute knot t_i for P_i so that $P(t)$ satisfies the condition $P_i = P(t_i),\ i = 1, 2, \ldots, n$. This paper proposes a method to compute t_i for P_i, $i = 1, 2, \ldots, n$ when the interpolation method is given so that the parametric curve $P(t)$ constructed by the given method delivers a better approximation precision than the curve created using the knots by other methods.

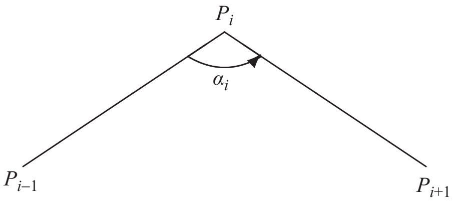

Figure 1 Three consecutive points.

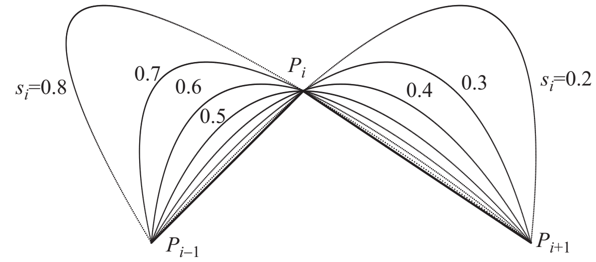

Figure 2 The plots of $P_i(s)$ for $s_i = i/10$, $i = 2, 3, \ldots, 8$.

For a given set of data points, they can be regarded as being sampled from a parametric curve. If the parametric curve is known, the knot for each data point can be computed easily using the parametric curve. Based on the assumption that the given data points are taken from a parametric curve that can be approximated appropriately by piece-wise quadratic polynomial curve segments, we describe the basic premise of our method for computing knots as follows: Without loss of generality, we assume that $P_i = (x_i, y_i)$, $i = 1, 2, \ldots, n$, are taken from a parametric curve $Q(t)$, i.e., $P_i = Q(t_i)$, and the curve segment of $Q(t)$ from P_{i-1} and P_{i+1} can be approximated by a quadratic polynomial $P_i(t)$ passing P_{i-1}, P_i and P_{i+1}, as shown in Figure 1. If $P_i(t)$ is constructed, the knots t_j corresponding to P_j, $j = i-1, i, i+1$ can be computed approximately using $P_i(t)$. Hence, computing t_j, $j = i-1, i, i+1$ becomes a problem of constructing $P_i(t)$. The knots t_i and t_{i+1} can be computed by $P_i(t)$ or $P_{i+1}(t)$. The combination of the two groups of t_i and t_{i+1} is used to determine the end value. The following sections will explain the construction of the quadratic curve $P_i(t)$. Let

$$t = t_{i-1} + (t_{i+1} - t_{i-1})s, \tag{1}$$

then the knots t_{i-1}, t_i and t_{i+1} are transformed into 0, s_i and 1 with one degree of freedom, s_i satisfying

$$s_i = (t_i - t_{i-1})/(t_{i+1} - t_{i-1}). \tag{2}$$

This means that for t_{i-1}, t_i and t_{i+1}, there is only one degree of freedom. The quadratic curve $P_i(s) = (x_i(s), y_i(s))$, $1 < i < n$, that passes P_{i-1}, P_i and P_{i+1} can be defined by the Lagrange formula as follows:

$$P_i(s) = \frac{(s - s_i)(s - 1)}{s_i}(P_{i-1} - P_i) + \frac{s(s - s_i)}{1 - s_i}(P_{i+1} - P_i) + P_i, \tag{3}$$

where s_i is an unknown and satisfies $0 < s_i < 1$.

As s_i is a variable, $P_i(s)$ is a family of curves, as shown in Figure 2, where the curves are plots of $P_i(s)$ in (3) for $s_i = i/10$, $i = 2, 3, \ldots, 8$. The task process for selecting a suitable curve from this family is discussed in the following sections.

2.1 Selecting s_i

Stress energy is widely used to optimize the shape and precision of the curve. One choice is to determine s_i by minimizing the stress energy of $P_i(s)$ in (3), i.e., by minimizing energy as follows:

$$E_s(s_i) = \int_0^1 k(s_i, s)^2 \mathrm{d}s, \tag{4}$$

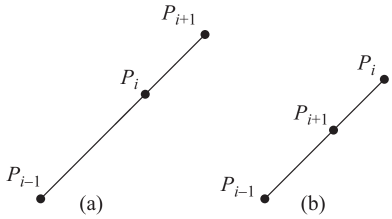

Figure 3 Three co-linear points. (a) $P_i'(s)$ is a constant; (b) $P_i'(s_i) = 0$.

where $k(s_i, s)$ is the curvature of $P_i(s)$ in (3).

There is no guarantee that minimizing the objective function (4) $P_i(s)$ in (3) is a suitable option. For example, if P_{i-1}, P_i and P_{i+1} are co-linear, then for any $0 < s_i < 1$, the minimum value of $E_s(s_i)$ in (4) is zero; that is, in this case, s_i is not unique. In fact, it is difficult to define a set of properties that can be used to characterize whether $P_i(s)$ in (3) meets the requirements or not. However, for some special cases, one can judge whether the curve is suitable or not. For example, if P_{i-1}, P_i and P_{i+1} are co-linear, as shown in Figure 3, one can easily determine whether $P_i(s)$ in (3) is suitable or not. There are two cases to be considered when P_{i-1}, P_i and P_{i+1} are co-linear, as shown by Figure 3(a) and (b). From a physical point of view, the curve $P_i(s)$ can be viewed as a path of a proton moving from point P_{i-1} through P_i to P_{i+1}. Then, $P_i'(s)$ is the proton's velocity. We then discuss how to construct a suitable $P_i(s)$ in (3) from the perspective of proton motion for these special cases. For the case in Figure 3(a), the ideal $P_i(s)$ in (3) is that the proton's velocity of the proton is a constant for $0 \leqslant s \leqslant 1$. It is easy to know that if s_i is defined by the following formula (5), then $P_i'(s)$ is a constant, and $P_i(s)$ in (3) becomes $P_i(s) = P_{i-1} + (P_{i+1} - P_{i-1})s$, which is the optima quadratic polynomial passing P_{i-1}, P_i and P_{i+1},

$$s_i = \frac{d_{i-1}}{d_{i-1} + d_i}, \tag{5}$$

where $d_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$ is the Euclidean distance between P_i and P_{i+1}.

For the case in Figure 3(b), when the proton moves from P_{i-1} to P_i, it should symmetrically move back from P_i to P_{i+1}. The best case scenario would be that the proton's velocity at P_i is zero, i.e., $|P_i'(s_i)| = 0$. Here, the symmetry means that if a proton moves from point P_{i+1} through P_i to P_{i-1}, its velocity at P_i is also zero. Obviously, if s_i is defined by the following formula (6), then $P_i'(s_i)$ is zero.

$$s_i = \frac{\sqrt{d_{i-1}}}{\sqrt{d_{i-1}} + \sqrt{d_i}}, \tag{6}$$

where d_i is defined by (5).

To ensure that $P_i(s)$ in (3) has the properties defined by (5) and (6), s_i in (3) will be determined by minimizing the combination of the stress and stretching energies as explained follows:

$$E(s_i) = \int_0^1 (\beta k(s_i, s)^2 + (1 - \beta)P_i'(s)^2)\mathrm{d}s, \tag{7}$$

where $\beta = \frac{\pi - \alpha_i}{\pi}\frac{\alpha_i}{\pi}$, α_i is the angle between vectors $P_{i-1} - P_i$ and $P_{i+1} - P_i$, as shown in Figure 1, $k(s_i, s)$ is defined in (4). This means that when $\alpha_i = \pi$ or 0, s_i is determined by minimizing $\int_0^1 P_i'(s)^2\mathrm{d}s$.

Theorem 1. For the curve $P_i(s)$ in (3), if s_i is determined by minimizing $E(s_i)$ in (7), then, $P_i(s)$ in (3) has the properties defined by (5) and (6) when P_{i-1}, P_i and P_{i+1} are co-linear.

Proof. When $\alpha_i = \pi$ or 0, then $\beta = 0$, $E(s_i)$ in (7) becomes

$$E(s_i) = \int_0^1 \left(\frac{(r(s) - 1)^2 d_{i-1}^2}{{s_i}^2} + A\frac{2(r(s) - 1)r(s)d_{i-1}d_i}{s_i(1 - s_i)} + \frac{r(s)^2 d_i^2}{(1 - s_i)^2} \right) \mathrm{d}s,$$

with

$$r(s) = 2s - s_i,$$

where $A = 1, -1$ for $\alpha_i = \pi$ and 0, respectively.

Direct computation gives

$$E(s_i) = \frac{1}{3}\left(\frac{d_{i-1}}{s_i} + A\frac{d_i}{1-s_i}\right)^2 + (d_{i-1} - Ad_i)^2.$$

By minimizing $E(s_i)$ we get

$$s_i = \frac{d_{i-1}}{d_{i-1}+d_i}, \qquad \text{for } A = 1, \tag{8}$$

and

$$s_i = \frac{\sqrt{d_{i-1}}}{\sqrt{d_{i-1}}+\sqrt{d_i}}, \qquad \text{for } A = -1. \tag{9}$$

We also have

$$\left|\frac{\partial P_i}{\partial s}(s_i)\right| = \left|\frac{s_i-1}{s_i}d_{i-1} + A\frac{s_i}{1-s_i}d_i\right|. \tag{10}$$

Substituting (8) and (9) into (10), respectively, we get

$$\left|\frac{\partial P_i}{\partial s}(s_i)\right| = d_{i-1} + d_i, \tag{11}$$

and

$$\left|\frac{\partial P_i}{\partial s}(s_i)\right| = 0. \tag{12}$$

When $\alpha_i = \pi$ and s_i is defined by (8), then $P_i(s)$ in (3) becomes

$$P_i(s) = P_{i-1} + (P_{i+1} - P_{i-1})s. \tag{13}$$

This is the ideal quadratic polynomial curve for interpolating P_{i-1}, P_i and P_{i+1}, the velocity of $P_i(s)$ in (13) is a constant defined by (11).

When $\alpha_i = 0$, s_i defined by (9) makes the velocity of $P_i(s)$ in (3) zero at the point P_i. Thus, s_i defined by (9) generates the optimum $P_i(s)$ in (3) which establishes the validity of the proposed method.

2.2 Formula for computing s_i

For every instance of three adjacent data points P_{i-1}, P_i and P_{i+1}, we need to minimize the objective function $E(s_i)$ in (7) to obtain s_i, which is a time consuming calculation. To overcome this drawback, we normalize the data points P_{i-1}, P_i and P_{i+1} as follows, Let $d_{i-1} \geqslant d_i$, then the three points are normalized to the following forms:

$$\begin{aligned} \bar{P}_{i-1} &= P_{i-1}/d_{i-1}, \\ \bar{P}_i &= P_i/d_{i-1}, \\ \bar{P}_{i+1} &= P_{i+1}/d_{i-1}. \end{aligned} \tag{14}$$

Then, the distance from $\bar{P}_{i-1}$ to $\bar{P}_i$ is 1, the one from $\bar{P}_i$ to $\bar{P}_{i+1}$ is less than or equal to 1. If $d_{i-1} < d_i$, the three points are assumed to be in the order from P_{i+1} through P_i to P_{i-1}. We define that for any two sets of the adjacent three data points P_{j-1}, P_j and P_{j+1}, $j = k, m$, after normalization by (14), if the two sets of the adjacent three data points are the same, then, the corresponding s_k and s_m satisfy $s_k = s_m$. With this definition, we can construct the formula for computing s_i in the following way.

Set $(x_{i-1}, y_{i-1}) = (-1, 0)$ and $(x_i, y_i) = (0, 0)$, (x_{i+1}, y_{i+1}) is defined by

$$\begin{aligned} x_{i+1} &= l_j \times \cos(\pi - \alpha_k), \\ y_{i+1} &= l_j \times \sin(\pi - \alpha_k), \end{aligned} \tag{15}$$

where $l_j = j/20$, $j = 2, 3, \ldots, 20$, $\alpha_k = k\pi/40$, $k = 2, 3, \ldots, 40$, as shown in Figure 4.

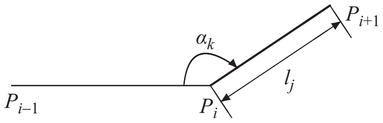

Figure 4 l_j and α_k.

For each set of P_{i-1}, P_i and P_{i+1}, one can get an $s_i = s_{j,k}$ by minimizing $E(s_i)$ in (7), then a point (l_j, α_k, s_i) is formed. There are $M = 19 \times 39$ data points (l_j, α_k, s_i), $j = 2, 3, \ldots, 20$, $k = 2, 3, \ldots, 40$, $s_i = s_{j,k}$ can be regarded as a function of (l_j, α_k). We construct a function $s = s(l, \alpha)$ to approximate these data points. Firstly, we discuss the boundary properties that $s = s(l, \alpha)$ should satisfy. The three boundary properties are as follows.

(1) When $\alpha_k = \pi$, s_i should satisfy (8), i.e., s_i should be defined by

$$s_i = \frac{1}{1 + l_j}. \tag{16}$$

(2) When $\alpha_k = 0$, s_i should satisfy (9), i.e., s_i should be defined by

$$s_i = \frac{1}{1 + \sqrt{l_j}}. \tag{17}$$

(3) When $l_j = 1$, s_i should satisfy

$$s_i = 0.5. \tag{18}$$

To construct $s = s(l, \alpha)$, we define the following four curves $f_r(l)$, $r = 0, 1, 2, 3$, associated with $\alpha_k = 0$, $\pi/3$, $2\pi/3$ and π, respectively, which are defined by

$$\begin{aligned} f_0(l) &= \frac{1}{1 + \sqrt{l}}, \\ f_1(l) &= L_0(l)c_{1,0} + L_1(l)c_{1,1} + L_2(l)c_{1,2} + L_3(l)/2, \\ f_2(l) &= L_0(l)c_{2,0} + L_1(l)c_{2,1} + L_2(l)c_{2,2} + L_3(l)/2, \\ f_3(l) &= \frac{1}{1 + l}, \end{aligned} \tag{19}$$

where $L_0(l)$, $L_1(l)$, $L_2(l)$ and $L_3(l)$ are cubic Langrange basis functions defined on $l = 0, 1/3, 2/3, 1$, $c_{m,g}$, $m = 1, 2$, $g = 0, 1, 2$, are unknowns to be determined. As the cases of $\alpha_k = 0$ and $\alpha_k = \pi$ do not appear generally, $f_0(l)$ and $f_3(l)$ are redefined as follows to provide six degrees of freedom, which are $c_{g,0}$, $c_{g,1}$, and $c_{g,2}$, $g = 0, 3$, respectively.

$$f_k(l) = L_0(l)c_{k,0} + L_1(l)c_{k,1} + L_2(l)c_{k,2} + L_3(l)/2, \quad k = 0, 3. \tag{20}$$

Now $s(l, \alpha)$ is defined by $f_r(l)$, $r = 0, 1, 2, 3$, with cubic polynomial interpolation as follows:

$$s(l, \alpha) = L_0(\alpha)f_0(l) + L_1(\alpha)f_1(l) + L_2(\alpha)f_2(l) + L_3(\alpha)f_3(l), \tag{21}$$

where $L_0(\alpha)$ $L_1(\alpha)$, $L_2(\alpha)$ and $L_3(\alpha)$ are cubic Langrange basis function defined on $\alpha = 0, \pi/3, 2\pi/3, 1$.

Based on the properties of a cubic Langrange basis function, it is easy to detect that $s(l, \alpha)$ in (21) satisfies the boundary properties defined by (16)–(18). The unknowns $c_{m,g}$, $m = 0, 1, 2, 3$, $g = 0, 1, 2$, in $s(l, \alpha)$ in (21) are determined by the least square fitting method, i.e., by minimizing the following objective function:

$$G(c_{0,0}, c_{0,1}, \ldots, c_{3,2}) = \sum_{k=2}^{40} \sum_{j=2}^{20} (s(l_j, \alpha_k) - s_{j,k})^2. \tag{22}$$

Minimization gets

$$\begin{aligned} &c_{0,0} = 0.8136, \quad c_{1,0} = 0.8332, \quad c_{2,0} = 0.9326, \quad c_{3,0} = 0.9903, \\ &c_{0,1} = 0.6441, \quad c_{1,1} = 0.6519, \quad c_{2,1} = 0.7089, \quad c_{3,1} = 0.7524, \\ &c_{0,2} = 0.5542, \quad c_{1,2} = 0.5566, \quad c_{2,2} = 0.5804, \quad c_{3,2} = 0.6003. \end{aligned} \tag{23}$$

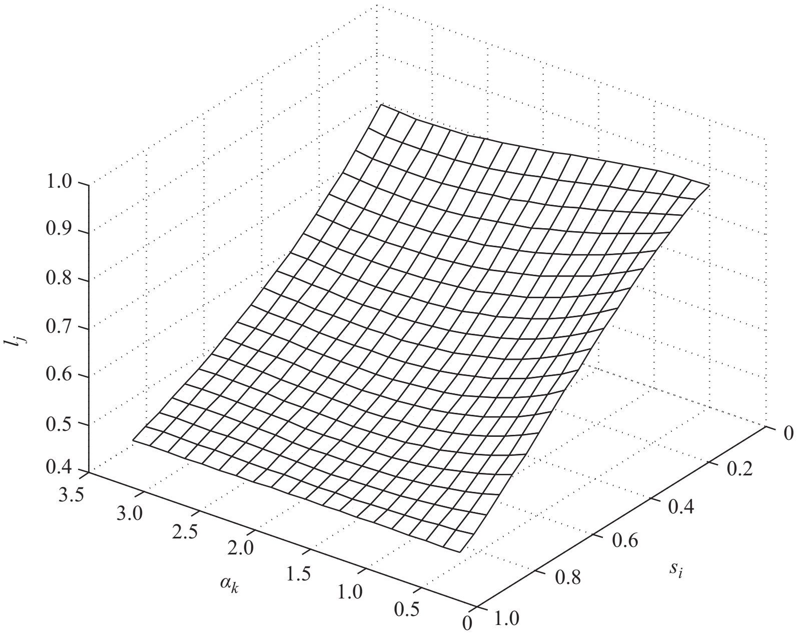

Figure 5 The plot of the formula (21).

The formula (21) with (19), (20) and (23) can approximate the data points (l_j, α_k, s_i), $j = 2, 3, \ldots, 20$, $k = 2, 3, \ldots, 40$, with high precision. Computation showed that the maximum error for the formula (21) approximating (l_j, α_k, s_i), $j = 2, 3, \ldots, 20$, $k = 2, 3, \ldots, 40$, is 0.009857. As the data points (l_j, α_k, s_i), $j = 2, 3, \ldots, 20$, $k = 2, 3, \ldots, 40$, are the solution of the model (7), formula (21) is a better approximation of the model (7). Computing the solution s_i of the model (7) using the formula (21) is an easy task. The plot of the formula (21) is shown in Figure 5, which shows that the solution of the formula (7) is a smooth function, which can be approximated easily.

3 A new local method for computing s_i

With $P_i(s)$ and $P_{i+1}(s)$, there are two knot intervals for P_i and P_{i+1}, i.e., $1 - s_i$ and s_{i+1}, respectively. As $P_i(s)$ and $P_{i+1}(s)$ are defined on different parametric spaces, in general, $1 - s_i$ is not equal to s_{i+1}, even if $P_i(s)$ and $P_{i+1}(s)$ represent the same curve. The reason is that if the knot interval for P_{i-1} and P_{i+1} is set as $[0, 1]$, then, by (2), the knot corresponding to P_{i+2} should be

$$s_{i+2} = (t_{i+2} - t_{i-1})/(t_{i+1} - t_{i-1}). \tag{24}$$

Since P_{i+2} is an arbitrary point and could have any possible position, then in general the knot interval $[s_i, s_{i+2}]$ for P_i and P_{i+2} will not be $[0, 1]$ through a translation transformation defined in (2), i.e., $s_{i+2} - s_i \neq 1$. In this section, we use a normal form of a quadratic curve introduced in the study [14] to merge $1 - s_i$ and s_{i+1} to form the knot interval for P_i and P_{i+1}. All the knot intervals associated with different point pares P_{i-1} and P_i, $i = 2, 3, \ldots, n$, are put together to form a consistent global knot sequence with respect to the same parameterization of a quadratic curve.

For the convenience of discussion, $P_i(s)$ in (3) is written as

$$\begin{aligned} x_i\left(s\right) &= X_{i,2}s^2 + X_{i,1}s + x_{i-1}, \\ y_i\left(s\right) &= Y_{i,2}s^2 + Y_{i,1}s + y_{i-1}, \end{aligned} \tag{25}$$

where

$$\begin{aligned} X_{i,2} &= \frac{x_{i-1} - x_i}{s_i} + \frac{x_{i+1} - x_i}{1 - s_i}, \quad & X_{i,1} &= -X_{i,2}s_i - \frac{x_{i-1} - x_i}{s_i}, \\ Y_{i,2} &= \frac{y_{i-1} - y_i}{s_i} + \frac{y_{i+1} - y_i}{1 - s_i}, \quad & Y_{i,1} &= -Y_{i,2}s_i - \frac{y_{i-1} - y_i}{s_i}. \end{aligned} \tag{26}$$

If $P_i(s)$ and $P_{i+1}(s)$ represent the same curve, they can be transformed into the normal form (29). Based on the normal form, they will have the same knot interval between P_i and P_{i+1}, as they are defined on the same parameter axis [14].

Suppose that in (26), $X_{i,2} \neq 0$ or $Y_{i,2} \neq 0$. By the following transformation:

$$\begin{aligned} \bar{x} &= x\mathrm{cos}\beta_i + y\mathrm{sin}\beta_i, \\ \bar{y} &= -x\mathrm{sin}\beta_i + y\mathrm{cos}\beta_i, \end{aligned} \tag{27}$$

where

$$\cos\beta_i = \frac{X_{i,2}+Y_{i,2}}{\sqrt{X_{i,2}^2+Y_{i,2}^2}}, \quad \sin\beta_i = \frac{Y_{i,2}-X_{i,2}}{\sqrt{X_{i,2}^2+Y_{i,2}^2}},$$

and a linear reparameterization

$$t = \left(X_{i,2}^2+Y_{i,2}^2\right)^{\frac{1}{4}} s. \tag{28}$$

$P_i(s)$ in (25) can be transformed into the following normal form:

$$\begin{aligned} \bar{x}_i\left(t\right) &= t^2 + \bar{X}_1 t + \bar{X}_0, \\ \bar{y}_i\left(t\right) &= t^2 + \bar{Y}_1 t + \bar{Y}_0, \end{aligned} \tag{29}$$

where

$$\begin{aligned} \bar{X}_0 &= \cos\beta_i x_{i-1} + \sin\beta_i y_{i-1}, & \bar{Y}_0 &= -\sin\beta_i x_{i-1} + \cos\beta_i y_{i-1}, \\ \bar{X}_1 &= \frac{\cos\beta_i X_{i,1} + \sin\beta_i Y_{i,1}}{\sqrt{\cos\beta_i X_{i,2} + \sin\beta_i Y_{i,2}}}, & \bar{Y}_1 &= \frac{-\sin\beta_i X_{i,1} + \cos\beta_i Y_{i,1}}{\sqrt{\cos\beta_i X_{i,2} + \sin\beta_i Y_{i,2}}}. \end{aligned} \tag{30}$$

When we convert the quadratic curve $P_i(s)$ in (25) to the normal form in (29), through the reparameterization process (28), the knot intervals s_i and $1-s_i$ associated with P_i become

$$\begin{aligned} \Delta_{i-1}^i &= \left(X_{i,2}^2+Y_{i,2}^2\right)^{\frac{1}{4}} s_i, \\ \Delta_i^i &= \left(X_{i,2}^2+Y_{i,2}^2\right)^{\frac{1}{4}} \left(1-s_i\right), \end{aligned} \tag{31}$$

where $X_{i,2}$ and $Y_{i,2}$ are defined in (26).

By mapping each $P_i(s)$ into the normal form, for each pair of consecutive points P_i and P_{i+1} there are two knot intervals, Δ_i^i and Δ_i^{i+1}, $2 \leqslant i \leqslant n-1$. In general, $\Delta_i^i \neq \Delta_i^{i+1}$. While for the two end data points, there is only one knot interval for each of them, i.e., Δ_1^2 for the pair P_1 and P_2, and Δ_{n-1}^{n-1} for the pair P_{n-1} and P_n. We average the two sequences of knot intervals, $\left\{\Delta_i^i\right\}$ and $\left\{\Delta_i^{i+1}\right\}$, into a single sequence of knot intervals, $\left\{\Delta_i\right\}, i = 1, 2, \ldots, n-1$, using the following formula:

$$\begin{aligned} &\Delta_1 = \Delta_1^2, \\ &\Delta_i = w(i)\Delta_i^i + (1-w(i))\Delta_i^{i+1}, \quad i = 2, 3, \ldots, n-2, \\ &\Delta_{n-1} = \Delta_{n-1}^{n-1}, \end{aligned} \tag{32}$$

where $w(i)$ is the weight function.

We now determine $w(i)$. Consider α_i as shown in Figure 1, if P_{i-1}, P_i and P_{i+1} are co-linear with the case in Figure 3(a), then $\alpha_i = \pi$. In this case, Δ_i should be mainly defined by $\Delta_i^i = d_i$ in (8) to make $P_i(s)$ having a form close to the one in (13). When α_i is larger than α_{i+1}, Δ_i^i should have a bigger impact on the formation of Δ_i than Δ_i^{i+1}, thus $w(i)$ should be proportional to α_i so that it has a larger value. Furthermore, corresponding to P_i and P_{i+1}, there are two knot intervals $1-s_i$ and s_{i+1}. If $s_i(1-s_i) > s_{i+1}(1-s_{i+1})$, then in general $|d_i - d_{i-1}| < |d_{i+1} - d_i|$, which means that s_i has higher precision than s_{i+1}. That said, $w_i(i)$ should have a relationship with the knot intervals $s_i(1-s_i)$ and $s_{i+1}(1-s_{i+1})$, and $w_i(i)$ is inversely proportional to $s_{i+1}(1-s_{i+1})$. We define

$$\begin{aligned} v_1(i) &= \mathrm{e}^{\lambda\alpha_i}/(1+\sigma s_{i+1}(1-s_{i+1})), \\ v_2(i) &= \mathrm{e}^{\lambda\alpha_{i+1}}/(1+\sigma s_i(1-s_i)). \end{aligned} \tag{33}$$

Then, $w(i)$ is defined by

$$w(i) = v_1(i)/(v_1(i) + v_2(i)). \tag{34}$$

The values of λ and σ are selected in a way that when Eqs. (32)–(34) are used to compute the knots, λ and σ should make the computed knots having as small error as possible. Our computation results show that in most cases, setting $\lambda = 1.5$, and $\sigma = 0.3$ makes the knots computed by (32)–(34) having the least errors.

Now, the global knot sequence $\{t_i\}$, $i = 1, 2, \dots, n$, are determined by

$$\begin{aligned} &t_1 = 0; \\ &t_{i+1} = t_i + \Delta_i, \quad i = 1, 2, \dots, n-1. \end{aligned} \tag{35}$$

The discussion above shows that the parameter interval between P_i and P_{i+1} is determined by $P_i(s)$ and $P_{i+1}(s)$, thus, only four points are needed to determine a knot interval, so, our method is local.

4 Experiments

In this section, we compare our method (New) with the chord length method (M1), Foley's method (M2), the centripetal method (M3), the quadratic polynomial precision method (M4) [14], the rational chord length method (M5) [22] and the refined centripetal method (M6) [12]. Three types of primitive curves are used to define the data points for comparison, leading to three types of data points. Among these three different types of data points, two of them are obtained from existing studies [13, 14]. The seven methods are used to compute knots to construct piece-wise cubic Hermite curves which interpolate the data points with C^1 continuity [14]. The tangent vector at each point is computed with the Bessel method [3]. These seven methods are compared via the interpolation precision of the piece-wise cubic Hermite curves constructed. The data points of the first type are taken from a family of elliptic curves, $F_1(k,t) = (x_1(k,t), y_1(k,t))$, defined by

$$\begin{aligned} x_1(k,t) &= (2 + 0.5k)\cos(2\pi t), \\ y_1(k,t) &= 2\sin(2\pi t), \end{aligned} \tag{36}$$

where $k = 0, 1, \dots, 13$.

The data points of the second type are taken from a family of cubic Hermite curves, $F_2(k,t) = (x_2(k,t), y_2(k,t))$, $k = 1, 2, \dots, 14$, which is defined by

$$\begin{aligned} x_2(k,t) &= df_1(t) + 3g_0(t) + dg_1(t), \\ y_2(k,t) &= df_1(t) - dg_1(t), \end{aligned} \tag{37}$$

where $d = 3 + 0.5k$, and

$$\begin{aligned} &f_0(t) = (1-t)^2(1+2t), \quad &&f_1(t) = (1-t)^2 t, \\ &g_0(t) = t^2(3-2t), &&g_1 = -t^2(1-t), \end{aligned}$$

are cubic Hermite basic functions on region $[0, 1]$.

When $k = 0$, $F_2(k,t)$ is a quadratic polynomial, if the data points are taken from a quadratic polynomial curve, then the knots computed by method M4 are exact. Hence the case of $k = 0$ is not considered here. The plots of $F_2(k,t) = (x_2(k,t), y_2(k,t))$, $k = 0, 2, 4, \dots, 14$, are given in Figure 6.

The data points of the third type are taken from a set of special curves, $F_l(t) = (x_l(t), y_l(t))$, $l = 3, 4, 5, 6$, which are defined respectively as follows and the plots are given in Figure 7.

Involute

$$\begin{aligned} x_3(t) &= \cos(t) + t\sin(t), \\ y_3(t) &= \sin(t) - t\cos(t). \end{aligned} \tag{38}$$

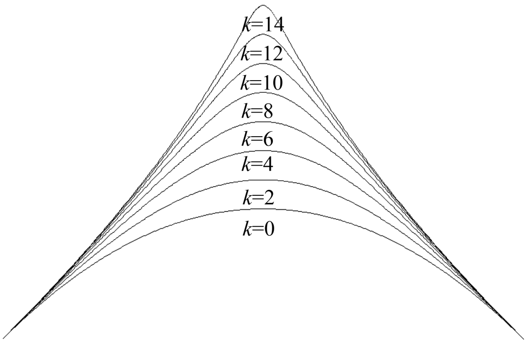

Figure 6 The plots of $F_2(k,t)$.

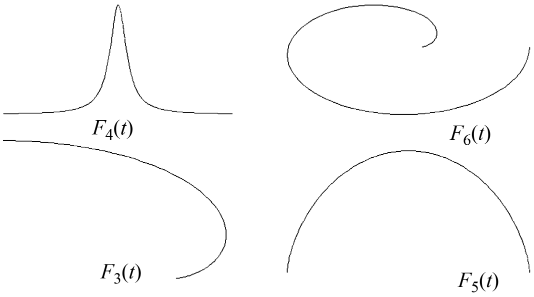

Figure 7 The plots of $F_l(t)$, $l=3,4,5,6$.

Versiera

$$\begin{aligned} &x_4(t)=tg(t),\\ &y_4(t)=\cos(t)^2. \end{aligned} \tag{39}$$

Common cycloid

$$\begin{aligned} &x_5(t)=t-\sin(t),\\ &y_5(t)=1-\cos(t). \end{aligned} \tag{40}$$

Picycloid

$$\begin{aligned} &x_6(t)=3\cos(t)-2\cos(3t/2),\\ &y_6(t)=3\sin(t)-2\sin(3t/2). \end{aligned} \tag{41}$$

The interval [0,1] used in the comparison is divided into 20 sub-intervals to define the data points $P_i=F_j(k,t_i)$ or $F_l(t_i)$, $i=0,1,2,\dots,19$, $j=1,2$, $l=3,4,5,6$, where t_i is defined by

$$t_i=[i+\lambda\sin((20-i)i)]/20,\quad i=0,1,2,\dots,20, \tag{42}$$

where $0<\lambda\leqslant 0.25$ to ensure the data points are non-uniformly distributed [13, 14], and satisfy $\mathrm{Max}\{d_{i-1},d_i\}\leqslant 3\mathrm{Min}\{d_{i-1},d_i\}$.

As $F_2(k,t)$ and $F_l(t)$, $l=3,4,5,6$, are not closed curves, in order to prevent the error at the end points from reaching the maximum value, the tangent vectors of $F_2(k,t)$ and $F_l(t)$, $l=3,4,5,6$, at the end points $t=0$ and $t=1$ are used to construct the cubic Hermite curves. The seven methods are evaluated in terms of the absolute error curves of $F_1(k,t)$, $F_2(k,t)$ and $F_l(t)$, $l=3,4,5,6$, defined below [13,14].

$$\begin{aligned} &E_j(k,t)=|P(s)-F_j(k,t)|=\min\{|P_i(s)-F_j(k,t)|\},\quad j=1,2,\\ &E_l(t)=|P(s)-F_l(t)|=\min\{|P_i(s)-F_l(t)|\},\quad l=3,4,5,\\ &s_i\leqslant s\leqslant s_{i+1},\quad i=0,1,2,\dots,19, \end{aligned} \tag{43}$$

where $P(s)$ denotes one of the cubic Hermite curves constructed by the seven methods. $F_j(k,t)$, $j=1,2$, and $F_l(t)$, $l=3,4,5,6$, are defined by (36)–(41), respectively. $P_i(s)$ denotes the part of $P(s)$ on $[s_i,s_{i+1}]$, $|P(s)-F_j(k,t)|$, $j=1,2$, and $|P(s)-F_l(t)|$, $l=3,4,5,6$, measure the distance from $P(s)$ to $F_j(k,t)$ and $F_l(t)$.

The seven methods are first compared with the first and second types of data points. The comparison results are as follows. For $E_1(k,t)$, $k=0,1,2,\dots,13$, and for $E_2(k,t)$, $k=1,2,3,\dots,14$, when $\lambda=0.15$ in (42), the maximum values of the error curves $E_1(k,t)$ and $E_2(k,t)$ generated by the seven methods are shown in Table 1. For the convenience of comparison, we introduce the Maximum-Minimum error. The Maximum-Minimum error is described as follows: For each given λ and k corresponding to $E_1(k,t)$ or $E_2(k,t)$, each of the seven methods produces one maximum error, hence there are seven maximum errors in total, and the minimum one among the seven maximum errors is known as the Maximum-Minimum error. In Table 1, the Maximum-Minimum errors of $E_1(k,t)$ and $E_2(k,t)$ are marked in bold. When

Table 1 Maximum errors of $E_1(k,t)$ and $E_2(k,t)$ for $\lambda = 0.15$

$E_1(k,t)$	New	M1	M2	M3	M4	M5	M6
k=0	1.64E−3	**1.02E−3**	5.95E−3	1.03E−2	1.11E−3	1.76E−2	8.96E−3
k=1	1.78E−3	2.18E−3	7.67E−3	1.24E−2	**1.24E−3**	2.09E−2	1.09E−2
k=2	2.04E−3	3.49E−3	9.14E−3	1.40E−2	**1.75E−3**	2.36E−2	1.23E−2
k=3	2.39E−3	5.59E−3	1.03E−2	1.51E−2	**2.24E−3**	2.58E−2	1.32E−2
k=4	**2.70E−3**	7.99E−3	1.14E−2	1.57E−2	2.71E−3	2.77E−2	1.36E−2
k=5	**2.96E−3**	1.07E−2	1.27E−2	1.58E−2	3.19E−3	2.93E−2	1.37E−2
k=6	**3.19E−3**	1.35E−2	1.42E−2	1.60E−2	3.63E−3	3.06E−2	1.50E−2
k=7	**3.39E−3**	1.73E−2	1.55E−2	1.76E−2	4.03E−3	3.17E−2	1.66E−2
k=8	**3.58E−3**	2.20E−2	1.66E−2	1.91E−2	4.41E−3	3.26E−2	1.82E−2
k=9	**3.75E−3**	2.71E−2	1.76E−2	2.04E−2	4.75E−3	3.34E−2	1.97E−2
k=10	**3.93E−3**	3.23E−2	1.84E−2	2.17E−2	5.08E−3	3.40E−2	2.10E−2
k=11	**4.22E−3**	3.76E−2	1.91E−2	2.28E−2	5.38E−3	3.45E−2	2.21E−2
k=12	**4.71E−3**	4.30E−2	1.97E−2	2.38E−2	5.66E−3	3.50E−2	2.30E−2
k=13	**5.51E−3**	4.83E−2	2.02E−2	2.46E−2	5.93E−3	3.54E− 2	2.39E−2
$E_2(k,t)$	New	M1	M2	M3	M4	M5	M6
k=1	4.18E−5	7.87E−5	2.24E−4	8.09E−4	**2.15E−5**	1.26E−4	7.60E−4
k=2	5.49E−5	1.01E−4	2.76E−4	8.83E−4	**4.64E−5**	1.57E−4	8.32E−4
k=3	7.69E−5	1.24E−4	3.31E−4	9.53E−4	**7.37E−5**	1.78E−4	9.01E−4
k=4	**1.03E−4**	1.63E−4	3.88E−4	1.02E−3	1.06E−4	1.90E−4	9.64E−4
k=5	**1.33E−4**	2.06E−4	4.45E−4	1.07E−3	1.51E−4	2.30E−4	1.02E−3
k=6	**1.67E−4**	2.47E−4	4.99E−4	1.12E−3	1.85E−4	2.66E−4	1.06E−3
k=7	2.05E−4	2.83E−4	5.49E−4	1.14E−3	**1.93E−4**	2.97E−4	1.09E−3
k=8	**2.45E−4**	3.76E−4	5.90E−4	1.15E−3	2.49E−4	3.24E−4	1.10E−3
k=9	**2.86E−4**	4.92E−4	6.58E−4	1.12E−3	5.03E−4	3.46E−4	1.07E−3
k=10	**3.33E−4**	6.39E−4	7.23E−4	1.04E−3	4.18E−4	3.97E−4	1.01E−3
k=11	**3.97E−4**	9.32E−4	7.76E−4	9.93E−4	5.09E−4	4.70E−4	9.73E−4
k=12	**4.50E−4**	1.35E−3	8.06E−4	1.03E−3	5.78E−4	5.54E−4	1.02E−3
k=13	6.19E−4	1.88E−3	7.97E−4	1.04E−3	**5.42E−4**	6.48E−4	1.04E−3
k=14	8.43E−4	2.50E−3	**7.23E−4**	9.85E−4	8.13E−4	7.48E−4	9.79E−4

Table 2 Maximum-Minimum errors of $E_1(k,t)$ and $E_2(k,t)$

$E_1(k,t)$	New	M1	M2	M3	M4	M5	M6
$\lambda = 0.05$	11	1	0	0	2	0	0
$\lambda = 0.10$	12	1	0	0	1	0	0
$\lambda = 0.15$	10	1	0	0	3	0	0
$\lambda = 0.20$	7	1	0	0	6	0	0
$\lambda = 0.25$	4	1	0	0	9	0	0
Summary	44	5	0	0	21	0	0
$E_2(k,t)$	New	M1	M2	M3	M4	M5	M6
$\lambda = 0.05$	8	0	2	0	4	0	0
$\lambda = 0.10$	9	0	1	0	4	0	0
$\lambda = 0.15$	8	0	1	0	5	0	0
$\lambda = 0.20$	9	0	0	0	4	1	0
$\lambda = 0.25$	10	0	0	0	4	0	0
Summary	44	0	4	0	21	1	0

$\lambda = 0.05i$, $i = 1,2,3,4,5$ in (42), the Maximum-Minimum errors of $E_1(k,t)$ and $E_2(k,t)$ produced by the seven methods are summarized in Table 2. In Table 2, the line of $E_1(k,t)$ with $\lambda = 0.15$ shows that the times of getting the Maximum-Minimum errors by New, M1–M6 methods are 10, 1, 0, 0, 3, 0 and 0 respectively, for $k = 0,1,2,\ldots,13$. The lines with $\lambda = 0.15$ in Table 2 summarize $E_1(k,t)$ and $E_2(k,t)$ in Table 1, respectively. Table 2 shows that for $\lambda = 0.05i$, $i = 1,2,3,4,5$ in (42), the total times of getting the Maximum-Minimum errors of $E_1(k,t)$ and $E_2(k,t)$ by New, M1–M6 methods are 44, 5, 0, 0, 21, 0, 0 and 44, 0, 4, 0, 21, 1, 0, respectively, for $k = 0,1,2,\ldots,13$. In order to make the comparison more intuitive, when $\lambda = 0.15$, the error curves $E_1(k,t)$ and $E_2(k,t)$ for $k = 6$ produced by the six methods are given in Figure 8. As the error curve by M5 is very similar to the one by M3, the error curve by M5 is not given in Figure 8. Based on the results in Tables 1 and 2 and Figure 8, it is clear that the precision of the curves constructed by our method is higher than those by M1–M6 methods, while the precision of the curves obtained with M4 method is higher than the one by M1–M3, M5, M6 methods.

The seven methods are further compared using the third type of data points from the set of special curves, $F_l(t) = (x_l(t), y_l(t))$, $l = 3,4,5,6$. The comparison results are provided in Tables 3 and 4. Table 3 is the Maximum errors for the set of data points taken from $F_3(t)$, when $\lambda = 0.05i$, $i = 1,2,3,4,5$ in (42),

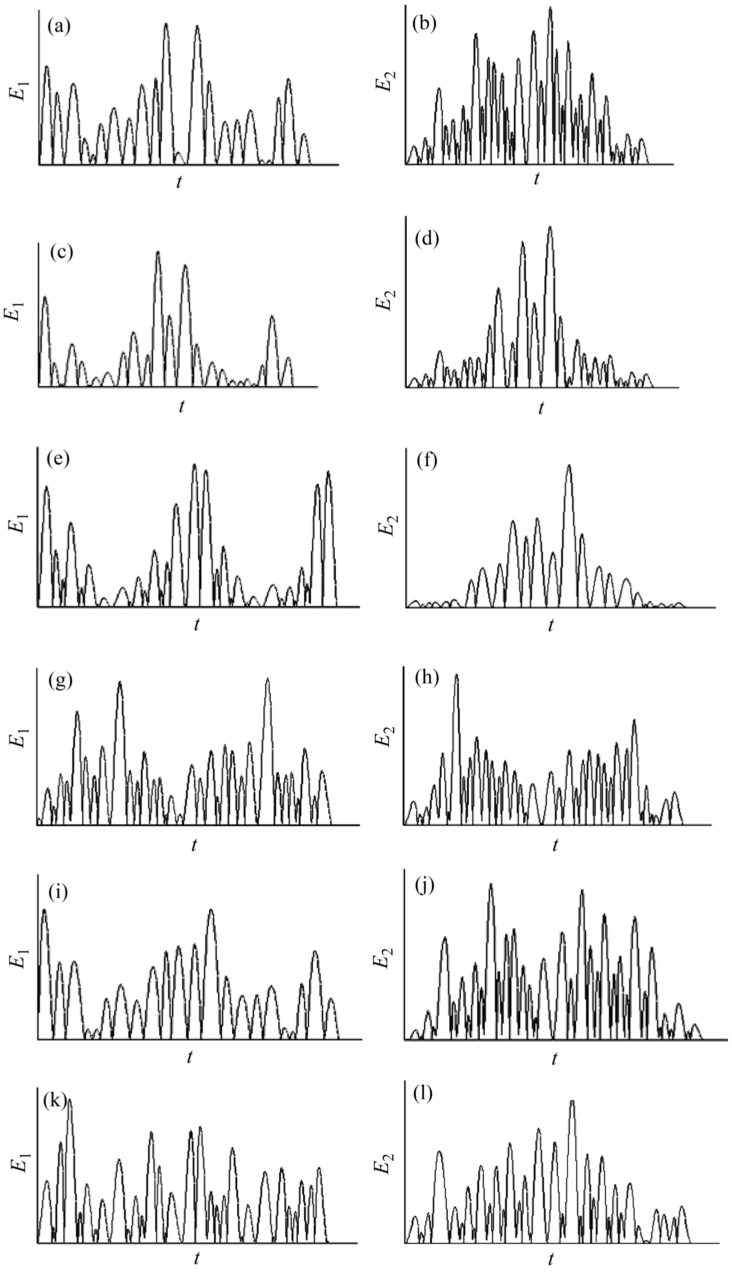

Figure 8 Error curves by six methods. (a) $80E_1(6,t)$ by M1; (b) $700E_2(6,t)$ by M1; (c) $92E_1(6,t)$ by M2; (d) $330E_2(6,t)$ by M2; (e) $92E_1(6,t)$ by M3; (f) $140E_2(6,t)$ by M3; (g) $360E_1(6,t)$ by M4; (h) $800E_2(6,t)$ by M4; (i) $36E_1(6,t)$ by M6; (j) $600E_2(6,t)$ by M6; (k) $396E_1(6,t)$ by New; (l) $1000E_2(6,t)$ by New.

and the Maximum-Minimum errors of $E_3(t)$ are marked in bold. Table 4 shows the total times of getting the Maximum-Minimum errors of $E_l(t)$, $l = 3, 4, 5, 6$, by New, M1–M6 methods are 19, 0, 0, 0, 1, 0 and 0, respectively. Tables 3 and 4 indicate, for $F_3(t)$–$F_6(t)$, that the new method provides the best results among the seven methods. In detail, the test result shows that for the third type of data points, the new method is obviously better than the other six methods when used to construct interpolation curves. Among the rest six methods, for the two sets of data points taken from $F_3(t)$ and $F_6(t)$, M4 has better results than M1–M3 and M5, M6; for the data points taken from $F_4(t)$, M5 provides better results than M1–M4 and M6; for the data points taken from $F_5(t)$, none of these six methods has obvious advantage over the others.

In interactive design, it is often necessary to modify the shape of the curve. Since the new method is

Table 3 Maximum errors of $E_3(t)$

$E_3(t)$	New	M1	M2	M3	M4	M5	M6
$\lambda = 0.05$	**3.75E−6**	6.71E−6	6.23E−6	2.39E−5	5.38E−6	9.13E−6	3.60E−5
$\lambda = 0.10$	**3.82E−6**	6.74E−6	5.87E−6	4.97E−5	4.83E−6	9.22E−6	6.05E−5
$\lambda = 0.15$	**3.88E−6**	6.77E−6	9.06E−5	7.80E−5	4.29E−6	9.30E−6	8.74E−5
$\lambda = 0.20$	**3.94E−6**	6.80E−6	1.31E−5	1.08E−4	4.55E−6	9.38E−6	1.17E−4
$\lambda = 0.25$	**4.00E−6**	6.82E−6	1.78E−5	1.42E−4	4.91E−6	9.45E−6	1.49E−4

Table 4 Maximum-Minimum errors of $E_l(t)$, $l = 3, 4, 5, 6$

$E_l(t)$	New	M1	M2	M3	M4	M5	M6
$\lambda = 0.05$	4	0	0	0	0	0	0
$\lambda = 0.10$	4	0	0	0	0	0	0
$\lambda = 0.15$	4	0	0	0	0	0	0
$\lambda = 0.20$	4	0	0	0	0	0	0
$\lambda = 0.25$	3	0	0	0	1	0	0
Summary	19	0	0	0	1	0	0

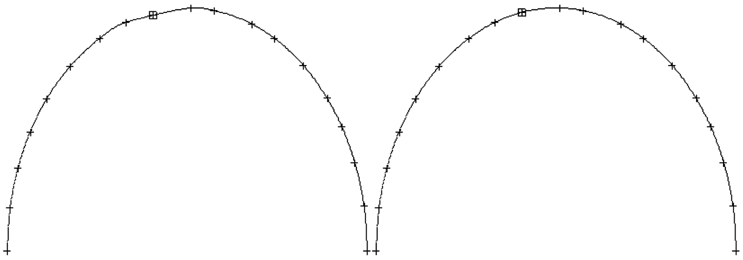

Figure 9 Modify a curve interactively.

local, it is easy to be used to modify a curve interactively. For example, the left C^1 curve in Figure 9 is not appropriate at the local region marked by symbol □, where the symbols + denote the positions of the data points. Changing the position of the data point marked by □ and re-parameterization will only change the parameters of the three adjacent points near the □. The parameters of the other points remain unchanged and consequently the corresponding curve shape is also unchanged. For the left curve in Figure 9, we move the point in symbol □ to a suitable location and re-parameterize the parameters of the three adjacent points. The right C^1 curve in Figure 9 is obtained.

5 Conclusion

The discussion in this paper shows that computing the knots for a given set of data points is equivalent to the problem of constructing the parametric curve. In this study, we propose a new method that assumes that the curve segment between three adjacent points can be approximated by a quadratic polynomial. However, the quadratic polynomial curve passing three consecutive points is not unique. To address this issue, we propose to generate a unique curve by minimizing its stress and stretching energies, which ensures that the quadratic curve has the appropriate properties so that when the three consecutive points are co-linear, the quadratic curve constructed is optimal, i.e., for three co-linear data points, P_{i-1}, P_i, and P_{i+1}, if P_i is between P_{i-1} and P_{i+1}, then the quadratic curve constructed becomes a linear polynomial $P_i(s) = P_{i-1} + (P_{i+1} - P_{i-1})s$, which is the ideal curve in this case. While if P_{i+1} is between P_{i-1} and P_i, then the first derivative of the quadratic curve at P_i is zero, which is also the ideal interpolating quadratic curve in this case.

Minimizing the stress and stretching energies to solve a non-linear problem is time-consuming. To overcome this drawback, we proposed a new model to simplify the minimization process, which approximates the stress and stretching energies with high precision. The new model is an explicit function and is used to compute the knots directly and requires less computation. This means that some classes of non-linear optimization problems can be simplified using an explicit function with high precision. Thus, using approximation techniques and machine learning techniques to simplify the process of solving the time-consuming non-linear optimization problems is feasible and effective.

The least that our method offers is linear precision, which means that from the point of view of approximation, our method is better than the chord length method, Foley's method [8], the centripetal method [9], the rational chord length method [22], and the refined centripetal method [12], all of which offer only linear precision. Experiments show that approximation precision of our method is better than the method proposed in [14], and obviously better than the other five methods.

As the knot is determined based on only three consecutive points, our method is not invariant under affine transformation, which is the same as the chord length method, Foley's method [8], the centripetal method [9], the rational chord length method [22] and the refined centripetal method [12]. For future studies, we plan to study how to make the knot computation method invariant under affine transformation and with quadratic polynomial precision. In a future study we plan to undertake, we intend extend the new method for data parameterization for constructing surface to fit the scattered data points. For each local region, the parameters associated with the data points will be computed using a local method, and the surface constructed will have GC^1 continuity.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61572292, 61373078), NSFC Joint Fund with Zhejiang Integration of Informatization and Industrialization under Key Project (Grant No. U1609218).

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Ahlberg J H, Nilson E N, Walsh J L. The Theory of Splines and Their Applications. New York: Academic Press, 1967
2 Boor C. A Practical Guide to Splines. Berlin: Springer, 1978
3 Brodie K W. A review of methods for curve and function drawing. In: Mathematical Methods in Computer Graphics and Design. London: Academic Press, 1980. 1–37
4 Faux I D, Pratt M J. Computational Geometry for Design and Manufacture. New York: Halsted Press, 1979
5 Su B Q, Liu D Y. Computational Geometry. Shanghai: Shang Hai Academic Press, 1982
6 Zhao G, Li W, Zheng J, et al. Target curvature driven fairing algorithm for planar cubic b-spline curves. Comput Aid Geom Des, 2004, 21: 499–513
7 Lü W. Curves with chord length parameterization. Comput Aid Geom Des, 2009, 26: 342–350
8 Farin G. Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide. London: Academic Press, 1990
9 Lee E. Choosing knots in parametric curve interpolation. Comput Aid Des, 1989, 21: 363–370
10 Jeong S Y, Yun J C, Park P G. Parametric interpolation using sampled data. Comput Aid Des, 2006, 38: 39–47
11 Yuksel C, Schaefer S, Keyser J. Parameterization and applications of catmull-rom curves. Comput Aid Des, 2011, 43: 747–755
12 Fang J J, Hung C L. An improved parameterization method for b-spline curve and surface interpolation. Comput Aid Des, 2013, 45: 1005–1028
13 Zhang C, Cheng F, Miura K T. A method for determining knots in parametric curve interpolation. Comput Aid Geom Des, 1998, 15: 399–416
14 Zhang C, Wang W, Wang J, et al. Local computation of curve interpolation knots with quadratic precision. Comput Aid Des, 2013, 45: 853–859
15 Hartley P J, Judd C J. Parametrization and shape of b-spline curves for cad. Comput Aid Des, 1980, 12: 235–238
16 Marin S P. An approach to data parametrization in parametric cubic spline interpolation problems. J Approx Theory, 1984, 41: 64–86
17 Floater M S, Reimers M. Meshless parameterization and surface reconstruction. Comput Aid Geom Des, 2001, 18: 77–92
18 Gotsman C. Fundamentals of spherical parameterization for 3d meshes. Acm Trans Graph, 2003, 22: 28–29
19 Gu X, Yau S T. Global conformal surface parameterization. In: Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, Aachen, 2003. 127–137
20 Xie H, Qin H. A novel optimization approach to the effective computation of nurbs knots. Int J Shape Model, 2011, 7: 199–227
21 Bastl B, Juettler B, Lavicka M. Spherical quadratic bézier triangles with chord length parameterization and tripolar coordinates in space. Comput Aid Geom Des, 2011, 28: 127–134
22 Bastl B, Juettler B, Lavicka M. Curves and surfaces with rational chord length parameterization. Comput Aid Geom Des, 2012, 29: 231–241
23 Tsuchie S, Okamoto K. High-quality quadratic curve fitting for scanned data of styling design. Comput Aid Des, 2016, 71: 39–50
24 Han X. A class of general quartic spline curves with shape parameters. Comput Aid Geom Des, 2011, 28: 151–163
25 Antonelli M, Beccari C V, Casciola G. High quality local interpolation by composite parametric surfaces. Comput Aid Geom Des, 2016, 46: 103–124