Computational Visual Media
DOI 10.1007/s41095-xxx-xxxx-x Vol. x, No. x, month year, xx–xx

Article Type (Research/Review)

Computing knots by quadratic and cubic polynomial curves

Fan Zhang[1,2] ✉ , Jinjiang Li[1,2], Peiqiang Liu[1,2],and Hui Fan[1,2]

Abstract A new method is presented to construct a model for computing parameter value (knot) for each data point. With four adjacent data points, a quadratic polynomial curve can be determined uniquely if the four points form a convex polygon. When the four data points do not form a convex polygon, a cubic polynomial curve with one degree of freedom (a variable) is used to interpolate the four points, to make the cubic polynomial curve have the better shape or approximate the polygon formed by the four data points. The degree of freedom is determined by minimizing the cubic coefficient of the cubic polynomial curve. The first advantage of the new method is that the knots computed by the new method has quadratic polynomial precision, in the sense that if the data points are sampled from an underlying quadratic polynomial curve, and the knots computed by new method are used to construct quadratic polynomial, the resulting interpolation curve reproduces the underlying quadratic curve. The second advantage of the new method is that it is affine invariant, which is very important while most parameterization methods do not have this property. And the third advantage of new method is that it computes knots as a local method. Experiments show that the curves constructed with the knots computed by the new method have better interpolation precision than the existing methods.

Keywords Knot, Polynomial curve, Minimizing, Affine invariant.

1 School of Computer Science and Technology, Shandong Technology and Business University, Yantai, 264005, China. E-mail: zhangfan51@sina.com, lijinjiang@gmail.com.

2 Co-Innovation Center of Shandong Colleges and Universities: Future Intelligent Computing, Yantai 264005, China. E-mail: liupq@126.com, fanlinw@263.net.

Manuscript received: 2014-12-31; accepted: 2015-01-30.

1 Introduction

In the fields of computer-aided design, engineering, scientific computing, and computer graphics, one of the fundamental problems that needs to be confronted is the construction of curves and surfaces with high precision and smoothness. The constructed curves and surfaces require high precision and unique attributes in different applications [1, 6, 7, 11, 24, 31]. To meet these requirements, better interpolation techniques and parameterization methods are needed. For scientific computation and engineering application, the method of constructing curves and surfaces with high polynomial accuracy is an ideal method. This paper focuses on how to determine the parameters (knots) of a given set of points with high precision.

Previous work: For a given set of data points, $P_i = (x_i, y_i)$, $i = 1, 2, ..., n$, the aim of parameterization is to assign a parameter value t_i, $t_0 < t_1 < t_n$, known as knots for each P_i. Interpolation curve can be seen as the movement of particle sequence through the position space (i.e., data points). Parameter t could be regarded as time, then the parameterization of the data points is equal to the time that a particle in turn arrives at the position space. For the same set of data, even with the same interpolation methods, constructing curve with different parameterization will result in different approximation result. That means the choice of a parameterization method will have a noticeable effect on the interpolated curve. Uniform parameterization is only suitable for some occasion that the intervals between consecutive data points are even. In application, three non-uniform parameterization strategies are widely used: the chord length method [22], Foley's method [9] and the centripetal method [18]. The chord length method was an ideal parameterized method, for it can reflect well the distribution according to chord length between consecutive data points. However, this interpolation only works well when the parametric curve is a straight line. The centripetal method assumes that, for a single arc, the centripetal force is proportional to the corner of curve tangent vector from start to end of the arc. Foley's method was an adapting chord parameterization method and good planar parameterization results can be derived from this method. But in terms of the approximation error, our experiments show that none of them can produce a satisfactory result.

The work by Lee [18], Jeong *et al.* [17] had a strengthened approach effect on those curves/surfaces whose curvature changes greatly and is irregular. However, our experiments show that the Jeong's method [17] generally results in more errors than the method [18]. In addition, among the chord length method, Foley's method [9] and the centripetal method [18], only the centripetal method can assure no local self-intersections on the constructed curve [28]. Yuksel *et al.* [28] gave an analysis of these three methods, showed that the curve produced by centripetal parameterization was more visually appropriate curves than those by the two others. But no mathematical explanation was given. Therefore, though the centripetal method is especially suitable for the unevenly data points distribution, the resulting interpolation curve of these methods do not always catch all the data characteristic. Fang *et al.* [8] refined the interpolation results of the centripetal method [18] to improve the wiggle evaluation, especially for abrupt chang data interpolation. A new universal parametrization [20] for B-spline interpolation is presented, which could improve the performance of the existing parametrizations such as the ones [9, 18, 22] by using the nature of B-spline basis function, the experiments showed that this method could not improve the precision of the interpolation curves.

Note that the interpolation precision of the knot locating methods previously mentioned is only linear, which implies that for the knot-set computed by above methods are used to construct interpolation curve, the interpolating curve will not be a quadratic polynomial curve if the data points are sampled from a quadratic polynomial curve. To solve the problem of that the data points are sampled from a non-linear curve, e.g., a quadratic or cubic polynomial curve, the underlying high-order curve [29, 30] with a higher interpolation precision is required to reconstruct. Zhang *et al.* [29] proposed a global method for choosing knots. Constructed interpolation by the chosen knots can exactly reproduce the quadratic polynomial curve where the data points are taken from. The approximation is better than linear precision methods in terms of error evaluation in the associated Taylor series. Based on the method of [29], a local method for determining knots with quadratic precision was introduced in Zhang *et al.* [30]. Even though this method employs a local computation, it has the ability of preserving quadratic precision. Hartley and Judd [16] discussed two ways of choosing knots: an iterative method and a simple formula. Due to the B-spline nodes were used as parameter values, method [16] can achieve good shape and good parametrization. Martin [23] proposed a method of choosing knots through optimization for parametric cubic spline interpolation. In the earlier article [19], the key technology was to generate a unique curve by minimizing its stress and stretching energies. An explicit function with high precision was constructed to compute the knots directly, which avoid solving non-linear optimization problems. Unfortunately, this method was still not invariant under affine transformation for the knot was determined based on only three consecutive points. The number of control points for constructing the curve plays an important role. To solve problems that control points are redundant or inadequate, for the space case, paper [21] extended the planar case [27], and proposed an adaptive removing and adding processes to refine the control points for the B-spline curve. Some articles also discuss the parameterization problems of spatial data points for other applications, article [12–14, 26]constructed a parametric surface by using the parameterized results.

Parameterization for curve and surface construction is still an unresolved problem and has attracted considerable attention. Motivated by the work [10], Łü [22] identified a family curves that can be parameterized by rational chord-length, and studied how the rational quartic and cubic curves were applied to G^1 Hermite interpolation. Similar to Łü [22], Bastl *et al.* [4, 5] also replaced arc length segments with chord lengths approximately, but extended the property of chord length parameterization of rational curves by a family of RCL surfaces to any dimension. Tsuchie and Okamato [25] introduced a curvature continuous G^2 quadratic B-spline curve for fitting planar curve. The G^2 curve is constructed with non-uniform knots to ensure the G^2 condition, thereby, to reduce the redundant segments in comparison to the use of uniform knots. To simplify the complicated optimization problem, the method [25] calculated the control points and adjusted the knot vector of the B-spline curve separately. Han [15] also discussed geometric continuous splines in curve design. A class of general quartic splines is presented for a non-uniform knot vector. The generated quartic spline curves had C^2 continuity with three local adjustable shape parameters, which had great influence on the shape of the spline curves. Bashir [3] presented the rational quadratic trigonometric Bézier curve with two shape parameters. Two segments of the objective curve can be joined with G^2 and C^2 continuity. Different from the classical tensor product setting, paper [2] assigned a different parameter interval to each mesh edge, which allows interpolation of each section polyline at parameter values that can prevent wiggling or other interpolation artifacts, and yields high-quality interpolating surfaces.

Proposed method: This paper provides a new method for computing knots. The new method is derived based on the assumption that the given set of data points are sampled from a parametric curve that can be approximated well by piecewise quadratic or cubic polynomial curves. In particular, the new method assumes that each curve segment between four adjacent points can be approximated by a quadratic polynomial or a cubic polynomial. If the four adjacent consecutive data points form a convex polygon, four data points are sufficient for determining a unique interpolation quadratic polynomial curve. Otherwise, when the four data points do not form a convex polygon, a cubic polynomial curve with one degree of freedom (a variable) is used to interpolate the four points. The degree of freedom is determined by minimizing the cubic coefficient of the cubic polynomial curve. This technique makes the method of constructing quadratic polynomial curve and cubic polynomial curve consistent in the sense that for quadratic polynomial curve, its cubic coefficient is zero, while for cubic polynomial curve, its cubic coefficient is as small as possible. Minimizing the cubic coefficient of the cubic polynomial curve could make the cubic polynomial curve approximate the polygon formed by the four data points well, and hence have the excellent shape. As the knots are determined by the quadratic curve and the cubic curve, they can reflect the distribution of the data points well. When the quadratic and cubic polynomial functions are determined, computing knot for each data point is an easy task.

The first advantage of the new method is that the knots computed by this method has quadratic polynomial precision in the sense that if the data points are sampled from an underlying quadratic polynomial curve, and the knots computed by the new method is used to construct quadratic polynomial, the resulting interpolation curve reproduces the underlying quadratic curve. Therefore, when used for curve construction, the resulting curve could have higher precision than the methods with linear precision. The second advantage of the new method is that it is affine invariant which is very important. Further more, 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 the curves constructed with the knots computed by the new method have better interpolation precision than the ones constructed using the knots by the existing methods. Experiments show that approximation precision with our method is better than the ones proposed in [5, 8, 9, 18, 19, 22, 30].

2 Basic idea of new method

Let $P_i = (x_i, y_i), 1 \le i \le n$, be a given set of distinct data points. The goal is that corresponding to each point P_i, a knot t_i, $t_i = 1, 2, ..., n$, is computed. And when the knots are used to construct a parametric curve $P(t)$ interpolating $P_i = (x_i, y_i)$, $1 \le i \le n$, using an existing interpolation method, $P(t)$ should have a quadratic polynomial precision in the case that if the given set of data points is sampled from an underlying quadratic polynomial curve, and the knots computed by the new method are used to construct quadratic polynomial, the constructed interpolation curve reproduces the underlying quadratic curve.

Now, the main idea of the new method is described briefly as follows. For each point P_i, we locally compute a knot t_i with the consecutive data points. For the two sets of consecutive data points corresponding to P_i, $\{P_{i-2}, P_{i-1}, P_i, P_{i+1}\}$ and $\{P_{i-1}, P_i, P_{i+1}, P_{i+2}\}$, two curves $P_i(t)$ and $P_{i+1}(t)$ passing the two sets, are constructed respectively. The two curves $P_i(t)$ and $P_{i+1}(t)$ are used to computed the knot t_i associated with P_i.

When the given set of data points, $P_i = (x_i, y_i), 1 \le i \le n$, is sampled from a quadratic polynomial curve, we compute the t_i which satisfy the following condition that, if the P_i are taken from a parametric quadratic polynomial $P(u) = (x(u), y(u))$ defined by

$$\begin{aligned} x(u) &= X_2 u^2 + X_1 u + X_0 \\ y(u) &= Y_2 u^2 + Y_1 u + Y_0. \end{aligned} \tag{1}$$

i.e., $P_i = P(u_i)$, then

$$t_i = \alpha u_i + \beta, \qquad 1 \le i \le n \tag{2}$$

for some constants α and β. This will ensure the quadratic precision, since a linear transform of the

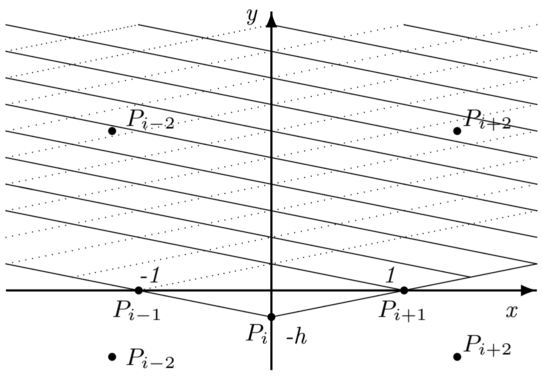

Fig. 1 Five data points after transform

knots does not change the shape of a curve.

If the data points P_i, $1 \leq i \leq n$, are taken from a quadratic polynomial defined by Eq. (1), any four consecutive data points $\{P_{i-2}, P_{i-1}, P_i, P_{i+1}\}$, $i = 3, 4, \cdots, n-1$ will uniquely determine a quadratic polynomial curve $P_i(t)$ which is the same as $P(u)$ in Eq.(1), but possibly with a different parameterization. Let $t_j^i = \alpha_i u_j + \beta_i$ be the knots computed with respect to $P_i(t)$ passing the four data points $\{P_{i-2}, P_{i-1}, P_i, P_{i+1}\}$. Let $t_j^{i+1} = \alpha_{i+1} u_j + \beta_{i+1}$ be the knots computed with respect to $P_{i+1}(t)$ passing the four data points $\{P_{i-1}, P_i, P_{i+1}, P_{i+2}\}$. Although $P_i(t)$ and $P_{i+1}(t)$ are the same with the quadratic curve $P(u)$ in Eq.(1), they could have two different parameterizations. Thus, we will have two sets of knot values t_j^i and t_j^{i+1} for the three data points P_j, $j = i-1, i, i+1$. Since the two sequences of knots t_j^i and t_j^{i+1}, $j = i-1, i, i+1$, are both linearly related to u_i, it is possible to use a linear mapping to match up the two sequences. For each point P_i, $1 \leq i \leq n$, a knot t_i is locally computed using two curves $P_i(t)$ and $P_{i+1}(t)$, all the t_i could have the different parameterizations, one needs to reparameterize t_i in a parameter space.

To develop a complete solution based on the idea above, we face two tasks:

1) Computing the local knot sequence t_j from the two groups of four consecutive data points separately;

2) Merging all these local knot sequences into a global knot sequence in a parameter space, which has quadratic precision.

These two steps will be explained in the following sections.

3 Computing Knot s_i from Consecutive Data Points

In this section, the main structure, functions and calculation methods of the knots of three consecutive points $\{P_{i-1}, P_i, P_{i+1}\}$ from their neighboring points are discussed in detail. For each set of four neighboring points $\{P_{i-2}, P_{i-1}, P_i, P_{i+1}\}$, a quadratic curve or cubic curve $Q_i(s)$ can be uniquely defined. But meanwhile, by the next set of four points $\{P_{i-1}, P_i, P_{i+1}, P_{i+2}\}$, a quadratic curve or cubic curve also can be determined . Here the three points $\{P_{i-1}, P_i, P_{i+1}\}$ are owned by these two sequences. Note that the group of each three neighboring points is a participant in at least two adjacent sequences. The key is the combination of the two sequences of s_i and s_{i+1}.

3.1 Computing s_i by a quadratic polynomial

For the given five consecutive sequent points $P_j = (x_j, y_j)$, $j = i-2, i-1, i, i+1, i+2$, if P_{i-1}, P_i and P_{i+1} are non-collinear, then, by the following transform

$$\begin{aligned} x &= a_{11}(x - x_i) + a_{12}(y - y_i) \\ y &= a_{21}(x - x_i) + a_{22}(y - y_i) - h, \end{aligned} \tag{3}$$

where

$$\begin{aligned} a_{11} &= \frac{y_{i-1} - 2y_i + y_{i+1}}{r} \\ a_{12} &= \frac{-x_{i-1} + 2x_i - x_{i+1}}{r} \\ a_{21} &= \frac{h(y_{i-1} - y_{i+1})}{r} \\ a_{22} &= \frac{h(x_{i+1} - x_{i-1})}{r} \\ r &= (x_{i+1} - x_i)(y_{i-1} - y_i) - (x_{i-1} - x_i)(y_{i+1} - y_i). \end{aligned} \tag{4}$$

The coordinates of P_{i-1}, P_i and P_{i+1} can be transformed as $(-1, 0)$, $(0, -h)$, $(1, 0)$, as shown in Figure 1. The quadratic polynomial $P_i(s)$ to interpolate points $(-1, 0)$, $(0, -h)$ and $(1, 0)$ is as follows:

$$\begin{aligned} x &= \frac{(s - s_i)(1 - s)}{s_i} + \frac{s(s - s_i)}{1 - s_i} \\ y &= \frac{s(s - 1)}{s_i(1 - s_i)} h, \end{aligned} \tag{5}$$

where $0 < s_i < 1$ is a parameter to be determined, which satisfies

$$s_i = \frac{t_i - t_{i-1}}{t_{i+1} - t_{i-1}}. \tag{6}$$

Theorem 1 In the formula(5), the relationship between parameter s and point (x, y) is defined by

$$s = \frac{1 + x + y(1 - 2s_i)/h}{2}. \tag{7}$$

Prof. The formula(5) are rewritten as

$$\begin{aligned} x &= \frac{(s-s_i)(1-s)(1-s_i)+s(s-s_i)s_i}{s_i(1-s_i)} \\ \frac{y}{h} &= \frac{s(s-1)}{s_i(1-s_i)}. \end{aligned} \tag{8}$$

We have

$$\frac{xh}{y} = \frac{(s-s_i)(1-s)(1-s_i)+s(s-s_i)s_i}{s(s-1)}. \tag{9}$$

Now

$$\frac{xh}{y} = \frac{(s-s_i)(1-s-s_i+2s_is)+s_i(1-s_i)}{s(s-1)} - \frac{h}{y}. \tag{10}$$

By simple algebra calculation, it follows from Eq.(10) that formula Eq.(7) holds.

Eq.(7) indicates that, once parameter s_i associated with (x_i, y_i) has been identified, parameter value s at point (x, y) can be achieved by Eq.(7). How to compute the parameter value s_i with point (x, y) is discussed below. Similarly, by simple algebra calculation, the relationship between parameter s_i and point (x, y) can be obtained from Eqs. (7) and (5), which is as follows:

$$a(x,y)s_i^2 + b(x,y)s_i + c(x,y) = 0 \tag{11}$$

where,

$$\begin{aligned} a(x,y) &= 4y(y+h) \\ b(x,y) &= -2y(\sigma(x,y)+\rho(x,y)+2h) \\ c(x,y) &= \rho(x,y)\sigma(x,y), \end{aligned} \tag{12}$$

with

$$\begin{aligned} \rho(x,y) &= h(1+x)+y \\ \sigma(x,y) &= h(x-1)+y. \end{aligned}$$

Then parameter s_i is the solution of Eq.(11). The five points P_j, $j = i-2, i-1, i, i+1, i+2$, can be mapped via an affine mapping Eq.(3), with the coordinates (x_{i-2}, y_{i-2}), $(0, 1)$, $(0, -h)$, $(1, 0)$ and (x_{i+2}, y_{i+2}), respectively, as shown in Figure 1, thus, the Eq.(11) is invariant under such an affine mapping, and hence, the parameter s_i is invariant.

If the coordinate values of point (x_{i+2}, y_{i+2}) satisfy $y_{i+2} + h(x_{i+2} - 1) > 0$ and $y_{i+2} - h(x_{i+2} - 1) > 0$, i.e., point P_{i+2} locates in the solid line area, then the four points $\{P_{i-1}, P_i, P_{i+1}, P_{i+2}\}$ form a convex polygon, as shown in Figure 1. When $(x, y) = (x_{i+2}, y_{i+2})$, the root of Eq.(11) is defined as follows:

$$s_i^r = \frac{-b(x_{i+2}, y_{i+2}) - \sqrt{G(x_{i+2}, y_{i+2})}}{2a(x_{i+2}, y_{i+2})}, \tag{13}$$

where,

$$G(x_k, y_k) = b(x_k, y_k)^2 - 4a(x_k, y_k)c(x_k, y_k),\ k = i+2.$$

Similarly, if the coordinate values of point (x_{i-2}, y_{i-2}) satisfy $y_{i-2} - h(x_{i-2} + 1) > 0$ and $y_{i-2} + h(x_{i-2} + 1) > 0$, i.e., point P_{i-2} locates in the dotted line area, as shown in Figure 1. The root of Eq.(11) with $(x, y) = (x_{i-2}, y_{i-2})$ is defined as follows:

$$s_i^l = \frac{-b(x_{i-2}, y_{i-2}) + \sqrt{G(x_{i-2}, y_{i-2})}}{2a(x_{i-2}, y_{i-2})}, \tag{14}$$

where $G(x_{i-2}, y_{i-2})$ is defined by Eq.(13).

On the two roots s_i^r (13) and s_i^l (14), we have the following theorem 2.

Theorem 2 When $(x, y) = (x_{i+2}, y_{i+2}), (x_{i-2}, y_{i-2})$, the roots of equation(11) are s_i^r(13) and s_i^l(14), respectively.

Prof. For simplicity, (x, y) in Eq.(11) is set as $(0, y)$, then $a(x, y)$, $b(x, y)$ and $c(x, y)$ in Eq.(12) becomes

$$\begin{aligned} a(x,y) &= 4y(y+h) \\ b(x,y) &= -4y(y+h) \\ c(x,y) &= (y+h)(y-h). \end{aligned}$$

Then, the two solutions of Eq.(11) are as follows:

$$s_i = \frac{y \pm \sqrt{hy}}{2y}. \tag{15}$$

Substituting Eq.(15) into Eq.(7) gets

$$s_i = \frac{1}{2}(1 - \frac{\pm\sqrt{hy}}{h}). \tag{16}$$

For $(x, y) = (x_{i+2}, y_{i+2})$, s_i^r(13) should satisfies $s_i^r > 1$, as $y > h$, it follows from Eq.(16) that s_i^r should be defined by Eq.(13). Similarly, for $(x, y) = (x_{i-2}, y_{i-2})$, $s_i^l < 0$ should be defined by Eq.(14). This completes the proof of Theorem 2.

3.2 s_i computed by a cubic polynomial

In addition to above two cases, however, we must mention another one, which is that the coordinate values (x_j, y_j), $j = i-2, i+2$, fail to meet $y_{i+2} - h(x_{i+2} - 1) > 0$ and $y_{i-2} + h(x_{i-2} + 1) > 0$, i.e., points P_{i-2} and P_{i+2} do not locate in the dotted line area and the solid line area, respectively, as shown in Figure 1. By this stage, points $(-1, 0)$, $(0, -h)$, $(1, 0)$ and (x_j, y_j), $j = i-2, i+2$, do not form a convex polygon, it is necessary to construct a cubic polynomial to interpolate the four sequent points. Denote the knot of points $P_j, j = i-2, i+2$, as s_j, then

$$s_j = \frac{1 + x_j + y_j(1-2s_i)/h}{2}. \tag{17}$$

The cubic polynomial interpolating the four sequent points is defined by

$$\begin{aligned} x &= -\frac{(s-s_i)(s-1)}{s_i} + \frac{s(s-s_i)}{1-s_i} + \frac{W(s)}{W(s_j)}X_j \\ y &= \frac{s(s-1)}{s_i(1-s_i)}h + \frac{W(s)}{W(s_j)}Y_j, \end{aligned} \tag{18}$$

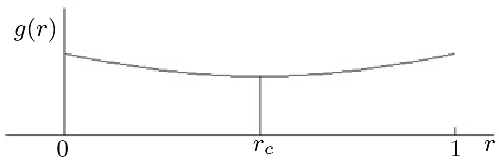

Fig. 2 The plot of the formula (23)

where,

$$\begin{aligned} W(s) &= s(s-s_i)(s-1) \\ X_j &= x_j + \frac{(s_j-s_i)(s_j-1)}{s_i} - \frac{s_j(s_j-s_i)}{1-s_i} \\ Y_j &= y_j - \frac{s_j(s_j-1)}{s_i(1-s_i)}h. \end{aligned} \tag{19}$$

Parameter s_i of Eq.(18) is determined by minimizing the cubic coefficient of Eq.(18), i.e., by minimizing the following objective function:

$$G(s_i) = \frac{X_j^2 + Y_j^2}{W(s_j)^2}. \tag{20}$$

The definition of objective function Eq.(20) is reasonable. When $(-1,0)$, $(0,-h)$, $(1,0)$ and (x_j,y_j) form a convex polygon, the cubic coefficient of the curve function is zero. While, when $(-1,0)$, $(0,-h)$, $(1,0)$ and (x_j,y_j) do not form a convex polygon, the cubic coefficient of the cubic curve should be as small as possible, thus enabling a slow and stable change of the curve shape in both cases.

3.3 Computing s_i

When the five points $P_j = (x_j, y_j)$, $i-2 \le j \le i+2$, are taken from the same quadratic curve, $s_i^l = s_i^r$. However, for data points given in general positions (but still assumed to form a convex chain), these five points may not lie on the same underlying quadratic curve, so $s_i^l \neq s_i^r$. In this case, we would need to reconcile the two values to determine a knot s_i for P_i. An obvious choice would be to set $s_i = (s_i^l + s_i^r)/2$. In the following, a more elaborate scheme to get s_i from s_i^l and s_i^r will be proposed, to further improve the estimate of s_i.

Reconsidering Eq.(11), let

$$h(x,y,s) = a(x,y)s^2 + b(x,y)s + c(x,y) = 0. \tag{21}$$

Then, s_i^l and s_i^r are roots of $h(x_{i-2}, y_{i-2}, s)$ and $h(x_{i+2}, y_{i+2}, s)$, respectively, as shown in Figure 3. Let

$$s_i = s_i^l + r(s_i^r - s_i^l) \tag{22}$$

and

$$g(r) = h(x_{i-2}, y_{i-2}, s_i)^2 + h(x_{i+2}, y_{i+2}, s_i)^2. \tag{23}$$

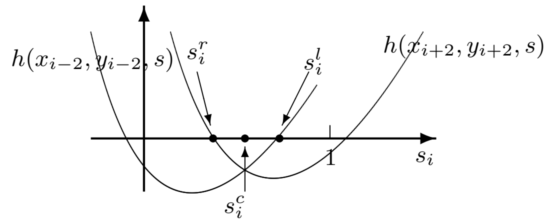

Fig. 3 Positions of s_i^l, s_i^c and s_i^r

It's obvious that better s_i in Eq.(22) makes the $g(r)$ have small value, the reason is that it follows from Eqs.(5)-(14) that small $g(r)$ means $P_i(s)$ (5) approximating the five data points (x_{i-2}, y_{i-2}), $(0,1)$, $(0,0)$, $(1,0)$ and (x_{i+2}, y_{i+2}) well. The shape of $g(r)$ is given in Figure 2. Our goal is to find a r^c so that

$$s_i^c = s_i^l + r^c(s_i^r - s_i^l),$$

makeing $g(r^c)$ being the minimum value of $g(r)$, as shown in Figure 2, which is defined by

$$g(r^c) = h(x_{i-2}, y_{i-2}, s_i^c)^2 + h(x_{i+2}, y_{i+2}, s_i^c)^2. \tag{24}$$

The value of r^c is determined by

$$\frac{dg(r)}{dr} = 0. \tag{25}$$

The Eqs.(21)-(23) show that Eq.(25) is a cubic equation, hence, it is easy to solve. Although there is no case that the Eq.(25) has no root in the interval $(0,1)$, for some cases in our experiments, we handle these special cases by the following way. If Eq.(25) has no root in the interval $(0,1)$ for some case, r_i^c is defined by

$$s_i^c = \begin{cases} s_i^l, \ if \ g(0) < g(1). \\ s_i^r, \ else \end{cases} \tag{26}$$

There are three estimates s_i^l, s_i^c and s_i^r defined by Eqs.(13), (14) and (22), respectively, for s_i, as indicated in Figure 3. Now we are going to compute s_i as a combination of s_i^l, s_i^c and s_i^r. We first discuss how to define the weight $\omega(s_i)$ associated with knot s_i. If one of s_i^l, s_i^c and s_i^r is closer to 0.5 than the other two, i.e., s_i^l is closer to 0.5, then s_i^l should have a bigger affect to the formation of s_i, thus the weight $\omega(s_i^l)$ is proportional to $s_i^l(1-s_i^l)$. Let

$$w(s) = \sqrt{h(x_{i-2}, y_{i-2}, s)^2 + h(x_{i+2}, y_{i+2}, s)^2}. \tag{27}$$

Obviously, for s_i^l, s_i^c and s_i^r, the best case is that they satisfying $w(s_i^l) = w(s_i^c) = w(s_i^r) = 0$. As in this case, each of s_i^l, s_i^c and s_i^r makes the curves in Eq.(5) or (18) interpolate the five points $P_j = (x_j, y_j)$, $j = i-2, i-1, i, i+1, i+2$. Thus, the weight $\omega(s_i)$ associated with knot s_i is inversely proportional to $w(s_i)$. Based on the discussion above, weight $\omega(s_i)$ is defined as

$$\omega(s_i) = \frac{s_i^2(1-s_i)^2}{w(s_i)}. \tag{28}$$

If one of $w(s_i^l)$, $w(s_i^c)$ and $w(s_i^r)$ is zero, for example, $w(s_i^c) = 0$, then s_i is defined by

$$s_i = s_i^c. \tag{29}$$

If none of $w(s_i^l)$, $w(s_i^c)$ and $w(s_i^r)$ is zero, then, s_i is defined by the weighted combination of s_i^l, s_i^c and s_i^r. The discussion above shows that s_i can be defined as follows

$$s_i = \frac{\omega(s_i^l)s_i^l + \omega(s_i^c)s_i^c + \omega(s_i^r)s_i^r}{\omega(s_i^l) + \omega(s_i^c) + \omega(s_i^r)}. \tag{30}$$

3.4 Discussion

So far we have excluded the case where some three consecutive points of them are collinear. Now we need to address this case. When P_{i-1}, P_i and P_{i+1} are on a straight line, we set

$$s_i = \frac{|P_{i-1}P_i|}{|P_{i-1}P_i| + |P_iP_{i+1}|}. \tag{31}$$

This choice makes the quadratic polynomial which passes P_{i-1}, P_i and P_{i+1} be a straight line with the magnitude of the first derivative being a constant. Such a straight line is the most naturally defined curve one can get in this case.

Finally, for the end data points, s_2 corresponding to $Q_2(s)$ is determined using the four points $P_j, j = 1,2,3,4$, and s_{n-1} corresponding to $Q_{n-1}(s)$ is determined using the points $P_j, j = n-3, n-2, n-1, n$.

4 Computing t_i with a local method

Based on the discussion above, with two sets of four data points $\{P_{j-1}, P_j, P_{j+1}, P_{j+2}\}$, $j = i-1, i$, one can construct two quadratic curves $P_i(s)$ and $P_{i+1}(s)$, and there are two knot intervals $1 - s_i$ and s_{i+1} for P_i and P_{i+1}, respectively. For $P_i(s)$, the knot interval for P_{i-1} and P_{i+1} is set as $[0, 1]$. While for $P_{i+1}(s)$, the knot interval for P_i and P_{i+2} is set as $[0, 1]$. Hence $P_i(s)$ and $P_{i+1}(s)$ are defined on different parametric spaces. The reason is as follows. For $P_i(s)$, the knot interval for P_{i-1} and P_{i+1} is set as $[0, 1]$, based on Eq.(6), the knot corresponding to P_{i+2} should be

$$s_{i+2} = (t_{i+2} - t_{i-1})/(t_{i+1} - t_{i-1}). \tag{32}$$

Thus, for $P_i(s)$, the knots corresponding to P_i, P_{i+1} and P_{i+2} are s_i,1 and s_{i+2}, respectively. Since P_{i+2} could have any possible position, in general, s_i,1 and s_{i+2} will not be 0, s_{i+1} and 1 through a translation transformation defined in Eq.(6), i.e., for the two sets $\{s_i, 1, s_{i+2}\}$ and $\{0, s_{i+1}, 1\}$, they generally do not satisfy

$$\frac{1 - s_i}{s_{i+2} - 1} = \frac{s_{i+1}}{1 - s_{i+1}}.$$

In the following, we will use a normal form of a quadratic curve introduced in the work [30] to translate all $P_i(s)$, $1 < i < n$, into the same parameter space, then to compute the knot interval for P_i and P_{i+1} by merging $1 - s_i$ and s_{i+1}. All the knot intervals corresponding to data pairs P_{i-1} and P_i, $i = 2, 3, ..., n-1$, are put together to form a consistent global knot sequence with respect to the same parameterization of a quadratic curve.

If the knots corresponding to P_{i-1}, P_i and P_{i+1} are 0, s_i and 1, respectively, then, the quadratic polynomial $P_i(s)$ passing these three data points can be written as

$$\begin{aligned} x_i(s) &= a_i s^2 + b_i s + x_{i-1} \\ y_i(s) &= d_i s^2 + e_i s + y_{i-1}, \end{aligned} \tag{33}$$

where

$$\begin{aligned} a_i &= \frac{(x_{i-1} - x_i)(1 - s_i) + (x_{i+1} - x_i)s_i}{s_i(1 - s_i)} \\ b_i &= -\frac{a_i s_i^2 + x_{i-1} - x_i}{s_i} \\ d_i &= \frac{(y_{i-1} - y_i)(1 - s_i) + (y_{i+1} - y_i)s_i}{s_i(1 - s_i)} \\ e_i &= -\frac{d_i s_i^2 + y_{i-1} - y_i}{s_i}. \end{aligned} \tag{34}$$

If $P_i(s)$ and $P_{i+1}(s)$ represent the same curve, they can be transformed into the normal form Eq.(37), and they will have the same knot interval between P_i and P_{i+1}.

Suppose that in Eq.(34), $a_i \neq 0$ or $d_i \neq 0$. By the following transformation Eq.(35)

$$\begin{aligned} \bar{x} &= x\cos\theta_i + y\sin\theta_i \\ \bar{y} &= -x\sin\theta_i + y\cos\theta_i, \end{aligned} \tag{35}$$

where,

$$\begin{aligned} \cos\theta_i &= \frac{a_i + d_i}{\sqrt{a_i^2 + d_i^2}} \\ \sin\theta_i &= \frac{d_i - a_i}{\sqrt{a_i^2 + d_i^2}} \end{aligned}$$

and a linear reparameterization Eq.(36)

$$t = \left(a_i^2 + d_i^2\right)^{\frac{1}{4}} s. \tag{36}$$

$P_i(s)$ (33) can be transformed into the following normal form

$$\begin{aligned} \bar{x}_i(t) &= t^2 + \bar{b}_i t + \bar{d}_i \\ \bar{y}_i(t) &= t^2 + \bar{e}_i t + \bar{f}_i, \end{aligned} \tag{37}$$

where,

$$\begin{aligned} \bar{d}_i &= \cos\theta_i x_{i-1} + \sin\theta_i y_{i-1} \\ \bar{f}_i &= -\sin\theta_i x_{i-1} + \cos\theta_i y_{i-1} \\ \bar{b}_i &= \frac{\cos\theta_i b_i + \sin\theta_i e_i}{\sqrt{\cos\theta_i a_i + \sin\theta_i d_i}} \\ \bar{e}_i &= \frac{-\sin\theta_i b_i + \cos\theta_i e_i}{\sqrt{\cos\theta_i a_i + \sin\theta_i d_i}}. \end{aligned} \tag{38}$$

Also, the property of above argument is invariant under such an normal form transformation of quadratic polynomial.

When the quadratic curve $P_i(s)$ (33) is transformed into the normal form in Eq.(37) by the reparameterization process Eqs.(35) and (36), the knot intervals s_i and $1-s_i$ in Eq.(33) become Δ_{i-1}^i and Δ_i^i, respectively, which are defined by

$$\begin{aligned}\Delta_{i-1}^i &= \left(a_i^2+d_i^2\right)^{\frac{1}{4}} s_i\\ \Delta_i^i &= \left(a_i^2+d_i^2\right)^{\frac{1}{4}}\left(1-s_i\right),\end{aligned} \tag{39}$$

where a_i and d_i are defined in Eq.(34).

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}, $2 \le i \le n-1$. 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, $\{\Delta_i\}, i=1,2,...,n-1$, using the following formula

$$\begin{aligned}&\Delta_1 = \Delta_1^2\\ &\Delta_i = \alpha_i\Delta_i^i + \beta_i\Delta_i^{i+1},\ i=2,3,...,n-2\\ &\Delta_{n-1} = \Delta_{n-1}^{n-1},\end{aligned} \tag{40}$$

where α_i and β_i are the weight function, satisfying $\alpha_i + \beta_i = 1$.

We now discuss the computation of α_i and β_i in Eq.(40). If all the data points are taken from the same quadratic curve, then

$$\begin{aligned}&\alpha_i = \beta_i = 0.5\\ &\alpha_i\Delta_i^i - \beta_i\Delta_i^{i+1} = 0.\end{aligned} \tag{41}$$

For the case that all the data points are not taken from the same quadratic curve, the values of α_i and β_i should have different values, and hence Δ_i^i and Δ_i^{i+1} have different effects on the formation of Δ_i. Corresponding to P_i and P_{i+1}, there are two knot intervals $1-s_i$ and s_{i+1}. If $s_i(1-s_i) > s_{i+1}(1-s_{i+1})$, in general, $|d_i - d_{i-1}| < |d_{i+1} - d_i|$, which means that s_i has higher precision than s_{i+1}, hence, Δ_i^i should have a bigger effect on the formation of Δ_i than Δ_i^{i+1}. On the other hand, if $s_i > 1-s_i$, Δ_i^i has higher precision than Δ_{i-1}^i as in the case $d_{i-1} > d_i$, and similarly, if $1-s_{i+1} > s_{i+1}$, Δ_i^{i+1} should have higher precision than Δ_{i+1}^{i+1}. This means that α_i and β_i should be proportional to the values $s_i^2(1-s_i)$ and $s_{i+1}(1-s_{i+1})^2$, respectively. For convenience, we first define two knot affect factors

$$\begin{aligned}\alpha_i^0 &= \frac{s_i^2(1-s_i)}{s_i^2(1-s_i)+s_{i+1}(1-s_{i+1})^2}\\ \beta_i^0 &= \frac{s_{i+1}(1-s_{i+1})^2}{s_i^2(1-s_i)+s_{i+1}(1-s_{i+1})^2}.\end{aligned}$$

To determine α_i and β_i, based on Eq.(41), we first define the following objective function

$G(\alpha_i^1,\beta_i^1) = (\alpha_i^1-\alpha_i^0)^2+(\beta_i^1-\beta_i^0)^2+(\alpha_i^1\Delta_i^i-\beta_i^1\Delta_i^{i+1})^2$

By minimizing $G(\alpha_i^1,\beta_i^1)$ yields

$$\begin{aligned}\alpha_i^1 &= \frac{\alpha_i^0(\Delta_i^{i+1}\Delta_i^{i+1}+1)+\beta_i^0\Delta_i^i\Delta_i^{i+1}}{2(\Delta_i^i)^2+2(\Delta_i^{i+1})^2+2}\\ \beta_i^1 &= \frac{\beta_i^0(\Delta_i^i\Delta_i^i+1)+\alpha_i^0\Delta_i^i\Delta_i^{i+1}}{2(\Delta_i^i)^2+2(\Delta_i^{i+1})^2+2}.\end{aligned} \tag{42}$$

In general, $\alpha_i^1+\beta_i^1 \neq 1$, they can not be used to define α_i and β_i directly. The factors α_i^0 and β_i^0 will be used to define the final α_i and β_i again. Now α_i and β_i in Eq.(40) are defined by

$$\begin{aligned}\alpha_i &= \frac{\alpha_i^0\alpha_i^1}{\alpha_i^0\alpha_i^1+\beta_i^0\beta_i^1}\\ \beta_i &= \frac{\beta_i^0\beta_i^1}{\alpha_i^0\alpha_i^1+\beta_i^0\beta_i^1}.\end{aligned} \tag{43}$$

Furthermore, for the end data points, there is only one knot interval, Δ_2^1, for the pair P_1 and P_2; and there is one knot interval, Δ_{n-1}^{n-1}, for the pair P_{n-1} and P_n. So Δ_1 and Δ_{n-1} are defined by

$$\begin{aligned}\Delta_1 &= \Delta_2^1\\ \Delta_{n-1} &= \Delta_{n-1}^{n-1}.\end{aligned} \tag{44}$$

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

$$\begin{aligned}&t_1 = 0\\ &t_{i+1} = t_i + \Delta_i, i=1,2,...,n-1.\end{aligned} \tag{45}$$

5 Experiments

The experiments are given in this section. The comparisons are between our method (New) with the explicit function method(M0) [19], the chord length method (M1), Foley's method (M2), the centripetal method (M3), the quadratic polynomial precision method (M4) [30], the rational chord length method (M5) [5] and the refined centripetal method (M6) [8]. The comparison is carried out by three type data points which are sampled from three sets of the primitive curves. To ensure the consistency, two of them are taken from existing studies [29, 30]. The 8 methods are compared by computing knots for constructing interpolation curves. The way of constructing curves and computing tangent vector at each point are the same as Li [19]. Afterwards, we compare the interpolation precision of piecewise cubic Hermite curves constructed by these 8 methods, consider the performance of the algorithms.

The first type data points are sampled from a family of elliptic curves, $F_1(k,t) = (x_1(k,t), y_1(k,t))$, defined

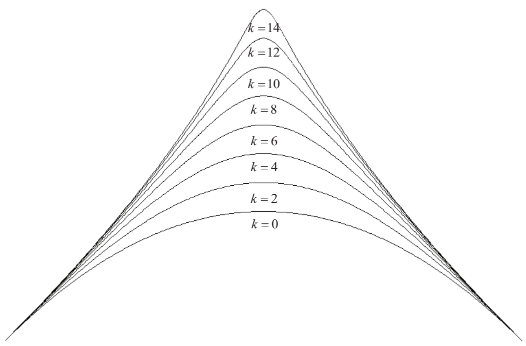

Fig. 4 The plots of $F_2(k,t)$

by

$$\begin{aligned} x_1(k,t) &= (2+0.5k)cos(2\pi t) \\ y_1(k,t) &= 2sin(2\pi t), \end{aligned} \tag{46}$$

where $k = 0, 1, ..., 13$. The second type data points are sampled from a family of cubic Hermite curves, $F_2(k,t) = (x_2(k,t), y_2(k,t))$, $k = 1, 2, ..., 14$, which is defined by

$$\begin{aligned} x_2(k,t) &= df_1(t) + 3g_0(t) + dg_1(t) \\ y_2(k,t) &= df_1(t) - dg_1(t), \end{aligned} \tag{47}$$

where $d = 3 + 0.5k$, and $f_0(t)$, $f_1(t)$, $g_0(t)$, $g_1(t)$ are cubic Hermite basic functions on region $[0,1]$.

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

The knots computed by the new method and the method M4 are exact when the data points are taken from a quadratic polynomial curve, and as $F_2(k,t)$ is a quadratic polynomial at $k = 0$, the case when $k = 0$ is discarded here. The plots of $F_2(k,t) = (x_2(k,t), y_2(k,t))$, $k = 0, 2, 4, ..., 14$, are given in Figure 4.

The third type data points are taken from four basic curves, $F_l(t) = (x_l(t), y_l(t))$, $l = 3, 4, 5, 6$, which are defined respectively as follows.

$$\begin{aligned} x &= t \\ y &= sin(\pi t). \end{aligned} \tag{48}$$

$$\begin{aligned} x &= t \\ y &= e^{\pi t}. \end{aligned} \tag{49}$$

$$\begin{aligned} x &= t \\ y &= \sqrt{1+(\pi t)^2}. \end{aligned} \tag{50}$$

$$\begin{aligned} x &= t \\ y &= \frac{1}{1+(t-0.5)^2}. \end{aligned} \tag{51}$$

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

$$t_i = [i + \lambda \sin((20-i)i)]/20, i = 0, 1, 2, \cdots, 20, \tag{52}$$

where $0 < \lambda \le 0.25$ to ensure the data points are non-uniformly distributed [29, 30], and meet $Max\{d_{i-1}, d_i\} \le 3Min\{d_{i-1}, d_i\}$.

For $F_2(k,t)$ and $F_l(t)$, $l = 3, 4, 5, 6$, are not closed curves, it's therefore easy to reach the maximum error value at the end points. Instead, 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 absolute error curves of $F_1(k,t)$, $F_2(k,t)$ and $F_l(t)$, $l = 3, 4, 5, 6$, are used to evaluate the 8 algorithms' performance, which defined as below [29, 30].

$$\begin{aligned} E_j(k,t) &= |P(s) - F_j(k,t)| \\ &= \min\{|P_i(s) - F_j(k,t)|\},\ j = 1, 2 \\ E_l(t) &= |P(s) - F_l(t)| \\ &= \min\{|P_i(s) - F_l(t)|\},\ l = 3, 4, 5 \\ & s_i \le s \le s_{i+1},\ i = 0, 1, 2, ..., 19 \end{aligned} \tag{53}$$

where $P(s)$ denotes one of the cubic Hermite curves constructed by the 8 methods. $F_j(k,t)$, $j = 1, 2$, and $F_l(t)$, $l = 3, 4, 5, 6$, are defined by Eqs.(46)-(51), respectively. $P_i(s)$ denotes the part of $P(s)$ on the interval $[s_i, s_{i+1}]$. The distance from $P(s)$ to $F_j(k,t)$ and $F_l(t)$ are defined as $|P(s) - F_j(k,t)|$, $j = 1, 2$, and $|P(s) - F_l(t)|$, $l = 3, 4, 5, 6$.

Comparison results of these 8 methods for the first and second types of data points are given first. Table 1 and table 2 describe the maximum values of the error curves $E_1(k,t)$ and $E_2(k,t)$ generated by the 8 methods, where, $E_1(k,t)$, $k = 0, 1, 2, \cdots, 13$, and $E_2(k,t)$, $k = 1, 2, 3, \cdots, 14$, when $\lambda = 0.15$ in Eq.(52). Table 1 and table 2 highlight the minimum values of maximum errors. The results displayed in table 1 and table 2 clearly demonstrate that new method has more minimum of maximum errors in most cases. Figure 5 is the error curves $E_1(k,t)$ and $E_2(k,t)$ at $k = 6$, $\lambda = 0.15$ produced by the seven methods, which gives the visual feeling to the precision of the curves constructed by these methods. The error curve by M5 is discarded because of its similarity to the one by M3. The results in table 1 and table 2 and Figure 5 fully display the higher precision of the curves constructed by the new method, followed by M0 method and M4 method.

Further experiment results on the third type data points, which are sampled from four basic curves, $F_l(t) = (x_l(t), y_l(t))$, $l = 3, 4, 5, 6$, defined by Eqs.(48)-(51), are provided in table 3. Table 3 is the Maximum errors for the set of data points sampled from $F_1(t)$,

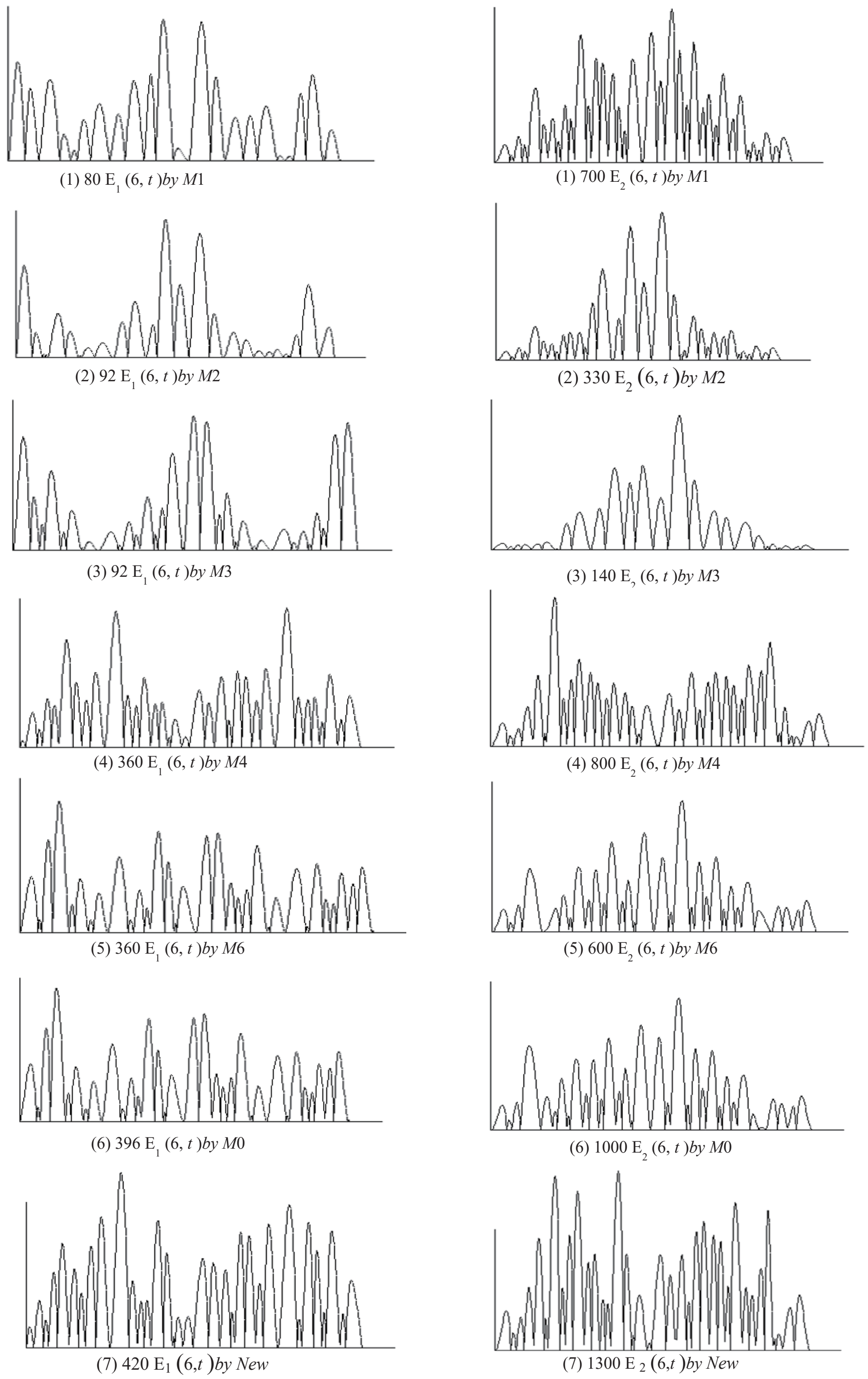

Fig. 5 Error curves by six methods

Tab. 1 Maximum errors of $E_1(k,t)$ for $\lambda = 0.15$

$E_1(k,t)$	New	M0	M1	M2	M3	M4	M5	M6
K=0	1.33e-3	1.64e-3	**1.02e-3**	5.95e-3	1.03e-2	1.11e-3	1.76e-2	8.96e-3
K=1	1.37e-3	1.78e-3	2.18e-3	7.67e-3	1.24e-2	**1.24e-3**	2.09e-2	1.09e-2
K=2	**1.71e-3**	2.04e-3	3.49e-3	9.14e-3	1.40e-2	1.75e-3	2.36e-2	1.23e-2
K=3	**2.09e-3**	2.39e-3	5.59e-3	1.03e-2	1.51e-2	2.24e-3	2.58e-2	1.32e-2
K=4	**2.46e-3**	2.70e-3	7.99e-3	1.14e-2	1.57e-2	2.71e-3	2.77e-2	1.36e-2
K=5	**2.79e-3**	2.96e-3	1.07e-2	1.27e-2	1.58e-2	3.19e-3	2.93e-2	1.37e-2
K=6	**3.10e-3**	3.19e-3	1.35e-2	1.42e-2	1.60e-2	3.63e-3	3.06e-2	1.50e-2
K=7	**3.38e-3**	3.39e-3	1.73e-2	1.55e-2	1.76e-2	4.03e-3	3.17e-2	1.66e-2
K=8	3.64e-3	**3.58e-3**	2.20e-2	1.66e-2	1.91e-2	4.41e-3	3.26e-2	1.82e-2
K=9	3.88e-3	**3.75e-3**	2.71e-2	1.76e-2	2.04e-2	4.75e-3	3.34e-2	1.97e-2
K=10	4.10e-3	**3.93e-3**	3.23e-2	1.84e-2	2.17e-2	5.08e-3	3.40e-2	2.10e-2
K=11	4.30e-3	**4.22e-3**	3.76e-2	1.91e-2	2.28e-2	5.38e-3	3.45e-2	2.21e-2
K=12	**4.58e-3**	4.71e-3	4.30e-2	1.97e-2	2.38e-2	5.66e-3	3.50e-2	2.30e-2
K=13	**4.97e-3**	5.51e-3	4.83e-2	2.02e-2	2.46e-2	5.93e-3	3.54e-2	2.39e-2

Tab. 2 Maximum errors of $E_2(k,t)$ for $\lambda = 0.15$

$E_2(k,t)$	New	M0	M1	M2	M3	M4	M5	M6
K=1	2.23e-5	4.18e-5	7.87e-5	2.24e-4	8.09e-4	**2.15e-5**	1.26e-4	7.60e-4
K=2	**4.58e-5**	5.49e-5	1.01e-4	2.76e-4	8.83e-4	4.64e-5	1.57e-4	8.32e-4
K=3	**7.15e-5**	7.69e-5	1.24e-4	3.31e-4	9.53e-4	7.37e-5	1.78e-4	9.01e-4
K=4	**9.99e-5**	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**	**1.33e-4**	2.06e-4	4.45e-4	1.07e-3	1.51e-4	2.30e-4	1.02e-3
K=6	**1.31e-4**	1.67e-4	2.47e-4	4.99e-4	1.12e-3	1.85e-4	2.66e-4	1.06e-3
K=7	**1.82e-4**	2.05e-4	2.83e-4	5.49e-4	1.14e-3	1.93e-4	2.97e-4	1.09e-3
K=8	**2.09e-4**	2.45e-4	3.76e-4	5.90e-4	1.15e-3	2.49e-4	3.24e-4	1.10e-3
K=9	3.34e-4	**2.86e-4**	4.92e-4	6.58e-4	1.12e-3	5.03e-4	3.46e-4	1.07e-3
K=10	**3.16e-4**	3.33e-4	6.39e-4	7.23e-4	1.04e-3	4.18e-4	3.97e-4	1.01e-3
K=11	**3.49e-4**	3.97e-4	9.32e-4	7.76e-4	9.93e-4	5.09e-4	4.70e-4	9.73e-4
K=12	**4.44e-4**	4.50e-4	1.35e-3	8.06e-4	1.03e-3	5.78e-4	5.54e-4	1.02e-3
K=13	**3.84e-4**	6.19e-4	1.88e-3	7.97e-4	1.04e-3	5.42e-4	6.48e-4	1.04e-3
K=14	**4.99e-4**	8.43e-4	2.50e-3	7.23e-4	9.85e-4	8.13e-4	7.48e-4	9.79e-4

Tab. 3 Maximum errors of $F_1(t)$

$F_1(t)$	New	M0	M1	M2	M3	M4	M5	M6
$\lambda = 0.05$	**4.28e-5**	1.56e-4	4.15e-4	2.87e-4	3.87e-4	5.29e-5	2.80e-4	5.46e-4
$\lambda = 0.10$	**4.51e-5**	1.64e-4	4.25e-4	4.52e-4	6.09e-4	5.68e-5	3.08e-4	1.03e-3
$\lambda = 0.15$	**4.82e-5**	1.73e-4	4.32e-4	6.19e-4	9.59e-4	6.05e-5	3.39e-4	1.58e-3
$\lambda = 0.20$	**5.75e-5**	1.82e-4	4.36e-4	8.46e-4	1.36e-3	6.38e-5	3.71e-4	2.17e-3
$\lambda = 0.25$	7.27e-5	1.90e-4	4.37e-4	1.13e-3	1.82e-3	**6.68e-5**	4.09e-4	2.83e-3

when $\lambda = 0.05i$, $i = 1,2,3,4,5$ in Eq.(52). Table 3 reveals that, for $F_1(t)$, the precision of the new method is much greater than those of the other 7 methods. Similar comparison experiments in other $F_l(t) = (x_l(t), y_l(t))$, $l = 4,5,6$, show the similar result. Table 3 also indicates that, when construct curves interpolate the third type data points, the new method has obviously advantage in curve precision than the other seven methods. Among the rest seven methods, M0 has better results than M1-6.

6 Conclusion

The discussion in this paper shows that computing knots for a given set of data points is equivalent to the problem of constructing the quadratic polynomial or cubic polynomial curve. We propose a new method that based on the fact that the curve segment between four adjacent points can be approximated by a quadratic polynomial or a cubic polynomial. If four adjacent consecutive data points form a convex polygon, they can determine a unique interpolation quadratic polynomial curve. When the four data points do not form a convex polygon, a cubic polynomial curve with one variable is used to interpolate the four points, the variable is determined by minimizing the cubic coefficient of the cubic curve. The method for constructing quadratic and cubic polynomials makes the method of computing knots consistent in the sense as follows. The cubic coefficient for the constructed quadratic polynomial curve is zero, while the cubic coefficient for the constructed cubic polynomial curve is as small as possible. Minimizing the cubic coefficient of the cubic polynomial curve could make the cubic polynomial curve approximate the polygon composed of the four data points well, and hence make the curve have the shape suggested by the four data points. As the knots are determined by the quadratic curve (four data points form a convex polygon) and the cubic curve(four data points do not form a convex polygon), they can reflect the distribution of the data points. When the quadratic and cubic polynomial functions are determined, computing knot for each data point is an easy task. One of the advantages of the new method is that the knots computed by the new method have quadratic polynomial precision, while the ones proposed in [5, 8, 9, 18, 19, 22] have only linear precision. This means that from the approximation point of view, the new method and the one [30] are better than the other six methods. Therefore, when used for curve construction, the resulting curve could have higher precision than the methods with linear precision. The second advantage of the new method is that it is affine invariant which is very important. Further more, 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 [5, 8, 9, 18, 19, 22, 30].

It is known that, when constructing a cubic spline interpolant, with the suitable end conditions and the knots, the constructed parametric cubic spline reproduces parametric cubic polynomials. Our next plan is to investigate whether there is a method of choosing knots with cubic precision. We also intend to extend the new method to 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 by using a local method, and the constructed surface will have GC^1 continuity.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61602277, 61772319, in part by the Natural Science Foundation of Shandong Province under Grant ZR2016FQ12, ZR2018BF009, in part by Key Research and Development Program of Yantai City under Grant 2017ZH065, in part by CERNET Innovation Project under Grant NGII20161204, and in part by the Science and Technology Innovation Program for Distributed Young Talents of Shandong Province Higher Education Institutions under Grant 2019KJN042.

References

[1] J. H. Ahlberg, E. N. Nilson, and J. L. Walsh. *The theory of splines and their applications.* Academic Press, 1967.

[2] M. Antonelli, C. V. Beccari, and G. Casciola. High quality local interpolation by composite parametric surfaces. *Computer Aided Geometric Design*, 46:103–124, 2016.

[3] U. Bashir, M. Abbas, and J. M. Ali. The g2 and c2 rational quadratic trigonometric bézier curve with two shape parameters with applications. *Applied Mathematics and Computation*, 219(20):10183–10197, 2013.

[4] B. Bastl, B. Juettler, M. Lavicka, J. Schicho, and Z. Sir. Spherical quadratic bézier triangles with chord length parameterization and tripolar coordinates in space. *Computer Aided Geometric Design*, 28(2):127–134, 2011.

[5] B. Bastl, B. Juettler, M. Lavicka, and Z. Sir. Curves and surfaces with rational chord length parameterization. *Computer Aided Geometric Design*, 29(5):231–241, 2012.

[6] K. W. Brodlie. A review of methods for curve and function drawing. *Mathematical methods in computer graphics and design*, pages 1–37, 1980.

[7] C. Deboor. *A practical guide to splines*, volume 27. springer-verlag New York, 1978.

[8] J. J. Fang and C. L. Hung. An improved parameterization method for b-spline curve and surface interpolation. *Computer-Aided Design*, 45(6):1005–1028, 2013.

[9] G. Farin. *Curves and surfaces for computer-aided geometric design: a practical guide.* Elsevier, 1989.

[10] G. Farin. Rational quadratic circles are parametrized by chord length. *Computer Aided Geometric Design*, 23(9):722–724, 2006.

[11] I. D. Faux and M. J. Pratt. *Computational geometry for design and manufacture.* Horwood Chichester, 1979.

[12] M. S. Floater and M. Reimers. Meshless parameterization and surface reconstruction. *Computer Aided Geometric Design*, 18(2):77–92, 2001.

[13] C. Gotsman, X. f. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3d meshes. volume 22, pages 358–363, 2003.

[14] X. F. Gu and S. T. Yau. Global conformal surface parameterization. pages 127–137, 2003.

[15] X. Han. A class of general quartic spline curves with shape parameters. *Computer Aided Geometric Design*, 28(3):151–163, 2011.

[16] P. J. Hartley and C. J. Judd. Parametrization and shape of b-spline curves for cad. *Computer-Aided Design*, 12(5):235–238, 1980.

[17] S. Y. Jeong, Y. J. Choi, and P. Park. Parametric interpolation using sampled data. *Computer-Aided Design*, 38(1):39–47, 2006.

[18] E. T. Lee. Choosing nodes in parametric curve interpolation. *Computer-Aided Design*, 21(6):363–370, 1989.

[19] X. M. Li, F. Zhang, G. N. Chen, and C. M. Zhang. Formula for computing knots with minimum stress and stretching energies. *Science China Information Sciences*, 61(5):052104, 2017.

[20] C. G. Lim. A universal parametrization in b-spline curve and surface interpolation. *Computer Aided Geometric Design*, 16(5):407–422, 1999.

[21] F. M. Lin, L. Y. Shen, C. Yuan, and Z. Mi. Certified space curve fitting and trajectory planning for cnc machining with cubic b-splines. *Computer-Aided Design*, 106:13–29, 2019.

[22] W. Lv. Curves with chord length parameterization. *Computer Aided Geometric Design*, 26(3):342–350, 2009.

[23] S. P. Marin. An approach to data parametrization in parametric cubic spline interpolation problems. *Journal of Approximation Theory*, 41(1):64–86, 1984.

[24] B. Q. Su and D. Y. Liu. *Computational Geometry*. Shang Hai Academic Press, 1982.

[25] S. Tsuchie and K. Okamoto. High-quality quadratic curve fitting for scanned data of styling design. *Computer-Aided Design*, 71:39–50, 2016.

[26] H. Xie and H. Qin. A novel optimization approach to the effective computation of nurbs knots. *International Journal of Shape Modeling*, 7(2):199–227, 2001.

[27] Z. Y. Yang, L. Y. Shen, C. M. Yuan, and X. S. Gao. Curve fitting and optimal interpolation for cnc machining under confined error using quadratic b-splines. *Computer-Aided Design*, 66:62–72, 2015.

[28] C. Yuksel, S. Schaefer, and J. Keyser. Parameterization and applications of catmull–rom curves. *Computer-Aided Design*, 43(7):747–755, 2011.

[29] C. M. Zhang, F. Cheng, and K. T. Miura. A method for determining knots in parametric curve interpolation. *Computer Aided Geometric Design*, 15(4):399–416, 1998.

[30] C. M. Zhang, W. P. Wang, J. Y. Wang, and X. M. Li. Local computation of curve interpolation knots with quadratic precision. *Computer-Aided Design*, 45(4):853–859, 2013.

[31] G. Zhao, W. Li, and J. Zheng. Target curvature driven fairing algorithm for planar cubic b-spline curves. *Computer Aided Geometric Design*, 21(5):499–513, 2004.

Fan Zhang Fan Zhang received his B.S. and Ph.D. degrees in computer science from Shandong University in 2009 and 2015, respectively. From 2012 to 2014, he was invited to visit the Department of Computer Science, University of Kentucky, USA, as a joint-training Ph.D. student. He is currently an associate professor with the School of Computer Science and Technology, Shandong Business and Technology University, Yantai, Shandong. His research interests include image processing, computer vision, computer graphics and CAGD.

Jinjiang Li Jinjiang Li received the B.S. and M.S. degrees in computer science from the Taiyuan University of Technology, Taiyuan, China, in 2001 and 2004, respectively, and the Ph.D. degree in computer science from Shandong University, Jinan, China, in 2010. From 2004 to 2006, he was an Assistant Research Fellow with the Institute of Computer Science and Technology, Peking University, Beijing, China. From 2012 to 2014, he was a Post-Doctoral Fellow with Tsinghua University, Beijing. He is currently a Professor with the School of Computer Science and Technology, Shandong Technology and Business University. His research interests include image processing, computer graphics, computer vision, and machine learning.

Peiqiang Liu PEIQIANG LIU received the Ph.D. degree in computer software and theory from Shandong University, Jinan, China, in 2013. He is currently a Professor at Shandong Technology and Business University. His research interests include algorithms and complexity theory, and computational biology.

Hui Fan Hui Fan received the B.S. degrees in computer science from Shandong University, Jinan, China, in 1984. He received the Ph.D. degree in computer science from Taiyuan University of Technology, Taiyuan, China, in 2007. From 1984 to 2001, he was a Professor at the computer department of Taiyuan University Technology. He is currently a Professor at Shandong Technology and Business University. His research interests include computer aided geometric design, computer graphics, information visualization, virtual reality, and image processing.