Constructing parametric quadratic curves

Caiming Zhang*,1, Fuhua (Frank) Cheng2
Department of Computer Science, University of Kentucky, Lexington, KY 40506, USA
Received 19 December 1997; received in revised form 25 August 1998

Abstract

Constructing a parametric spline curve to pass through a set of data points requires assigning a knot to each data point. In this paper we discuss the construction of parametric quadratic splines and present a method to assign knots to a set of planar data points. The assigned knots are invariant under affine transformations of the data points, and can be used to construct a parametric quadratic spline which reproduces parametric quadratic polynomials. Results of comparisons of the new method with several known methods are included. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Parametric quadratic curve; Interpolation; Knots; Spline

1. Introduction

Since their introduction by Schoenberg in 1946 [14], spline functions have been widely used in many areas of scientific computation, engineering design, and CAD due to their good properties such as local control capability, small strain energy, and numerical stability. The first property enables local modification of a spline function while the second one facilitates its geometric smoothness. Quadratic and cubic splines are the most frequently used splines in various applications.

In computer graphics, image processing, computer vision, pattern recognition, CAD, and many other fields, modeling the shape of an image or an object usually requires the technique of parametric curve interpolation [4], i.e., constructing a parametric curve that interpolates, or passes through, a given set of points, called interpolation points or data points. The parametric curve is usually constructed using a spline method. The resulting curve is then used as an approximation of the desired image or object. The data points should be selected in a way so that good approximation is guaranteed.

* Corresponding author. E-mail: czhang@csr.uky.edu.
1 Supported by Ford.
2 Supported by NSF (9722728), Ford and Honda.

0377-0427/99/$ see front matter © 1999 Elsevier Science B.V. All rights reserved.
PII: S0377-0427(98)00206-4
The precision of the above approximation process also depends on the selection of the parametric knots. Actually, appropriately spaced parametric knots also reduce the energy of the resulting curve, avoid the occurrence of ‘oscillations’ and ‘loops’, and facilitate the reproduction degree of the resulting curve. An interpolation scheme is said to have a parametric polynomial reproduction degree of \( n \) if it reproduces a parametric polynomial of degree \( n \) as long as the data points are taken from this parametric polynomial. Hence, a key issue in spline curve interpolation is the selection of parametric knots.

Presently there are four popular methods in use for choosing knot locations, namely, uniform parametrization (uniform method), accumulated chord length method (chord length method), centripetal model \([10]\), and affine invariant parametrization method (Foley/Nielson method) \([7, 11]\). The chord length method \([1, 3, 5, 6]\) seems to be the most widely used one. As far as reproduction degree is concerned, none of these methods provides a reproduction degree higher than one. A new approach recently proposed by Zhang et al. \([15]\) (referred as the ZCM method) has a reproduction degree of two. A deficiency of the chord length, centripetal, and ZCM methods is that the knots determined by these methods are not invariant under affine transformations of the data points. The uniform and the Foley/Nielson methods are invariant under affine transformations of the data points, but the uniform method is only suitable for data points which are evenly spaced.

This paper discusses the problem of constructing parametric quadratic splines, and proposes a method to choose knots for a set of planar data points. The selected knots are invariant under affine transformations of the data points. The quadratic spline constructed using the selected knots can reproduce a parametric polynomial of degree two.

The remaining part of this paper is arranged as follows. In Section 2, the equations that parametric quadratic splines are required to satisfy are discussed. A method to choose knots for a given set of planar points is given in Section 3. The comparison of the new method with the chord length method, the centripetal model, Foley/Nielson method and the ZCM method is performed in Section 4.

### 2. Constructing parametric quadratic spline

Let \( P_i = (x_i, y_i), \ i = 1, 2, \ldots, n, \) be an ordered sequence of distinct, planar data points, and \( t_i \) be the knots to be determined. For \( i \) from 2 to \( n \), let \( \bar{t}_i = (t_{i-1} + t_i)/2. \ \bar{t}_i, \ i = 1, \ldots, n - 1, \) are called nodes. Furthermore, we set \( t_0 = t_1 \) and \( t_n = t_{n-1} \). The goal is to construct a parametric quadratic spline \( P(t) \) with the following properties:

1. \( P(t) \) is a parametric quadratic polynomial on each subinterval \([\bar{t}_{i-1}, \bar{t}_i], \ i = 1, 2, \ldots, n; \)
2. \( P(t) \) is \( C^1 \) continuous on \([t_1, t_n]; \)
3. \( P(t_i) = P_i, \ i = 1, 2, \ldots, n. \)

Note that, for the specified knots, there are two additional degrees of freedom for constructing this quadratic spline.

Let \( \tilde{P}_i \) denote the value of \( P(t) \) at the node \( \bar{t}_i. \) The segment of the quadratic spline \( P(t) \) on the subinterval \([\bar{t}_{i-1}, \bar{t}_i] \) is denoted by \( \tilde{P}_i(t), \ i = 2, 3, \ldots, n - 1. \) \( P(t) \) is a quadratic polynomial passing through \( \tilde{P}_{i-1}, P_i \) and \( \tilde{P}_i. \) \( P(t) \) can be put in Lagrange’s form as follows:

\[
P(t) = \frac{2(t - t_i)}{t_{i+1} - t_{i-1}} \left[ \frac{2(t - \bar{t}_i)}{t_i - t_{i-1}} (\tilde{P}_{i-1} - P_i) + \frac{2(t - \bar{t}_{i-1})}{t_{i+1} - t_i} (\tilde{P}_i - P_i) \right] + P_i, \tag{1}
\]

where \( t \) is a parameter in \([\bar{t}_{i-1}, \bar{t}_i]. \)
From \( P'_i(t_i) = P'_{i+1}(t_i) \), it follows that the continuity equation for \( P(t) \) being \( C^1 \) continuous at node \( t_i \) is

\[
(1 - s_i) \frac{\tilde{P}_{i-1}}{h_{i-1}} + (3 - s_i + s_{i+1}) \frac{\tilde{P}_i}{h_i} + s_{i+1} \frac{\tilde{P}_{i+1}}{h_{i+1}} = \frac{1}{h_i} \left( \frac{P_i}{s_i} + \frac{P_{i+1}}{1 - s_{i+1}} \right), \quad i = 2, 3, \ldots, n - 2, \tag{2}
\]

where

\[
s_i = \frac{h_{i-1}}{h_{i-1} + h_i}, \quad h_i = t_{i+1} - t_i.
\]

This is a system of \((n - 3)\) equations in \((n - 1)\) unknowns \( \tilde{P}_i/h_i, \ i = 1, 2, \ldots, n - 1 \) (the reason for using \( \tilde{P}_i/h_i \) as the unknowns instead of \( \tilde{P}_i \) is to ensure that the resulting matrix is diagonally dominant). Two extra conditions are needed to solve this system of equations. One option in introducing two extra conditions is shown below. First, for the subinterval \([t_0, t_1]\), construct a parametric quadratic curve \( P_1(t) \) that interpolates \( P_1; P_1 \) and \( P_2 \) in Lagrange’s form, then from the requirement

\[
(1 + s_2) \frac{\tilde{P}_1}{h_1} + s_2 \frac{\tilde{P}_2}{h_2} = \frac{1}{2h_1} \left( P_1 + \frac{2h_1 + h_2}{h_2} P_2 \right). \tag{3}
\]

By constructing a parametric quadratic curve \( P_n(t) \) on the subinterval \([t_{n-1}, t_n]\) that interpolates \( P_{n-1}, P_{n-1} \) and \( P_n \) in Lagrange’s form, and then imposing the requirement that \( P'_n(t_{n-1}) = P'_n(t_{n-1}) \), one gets another condition

\[
(1 - s_{n-1}) \frac{\tilde{P}_{n-2}}{h_{n-2}} + (2 - s_{n-1}) \frac{\tilde{P}_{n-1}}{h_{n-1}} = \frac{1}{2h_{n-1}} \left( \frac{h_{n-2} + 2h_{n-1}}{h_{n-2}} P_{n-1} + P_n \right). \tag{4}
\]

If \( h_i, \ i = 1, 2, \ldots, n - 1, \) are appropriately selected, the quadratic spline constructed by (2)–(4) can reproduce parametric polynomials of degree two. Obviously, \( h_i, \ i = 1, 2, \ldots, n - 1, \) and \( zh_i, \ i = 1, 2, \ldots, n - 1, (z > 0), \) produce the same parametric quadratic spline \( P(t) \). In Section 3, we shall discuss how to determine knots \( t_i, \ i = 1, 2, \ldots, n, \) so that if \( P_i = (x_i, y_i) \), \( i = 1, 2, \ldots, n, \) are taken from a parametric quadratic curve, i.e.,

\[
x_i = a_1 u_i^2 + b_1 u_i + c_1, \quad y_i = a_2 u_i^2 + b_2 u_i + c_2, \tag{5}
\]

then

\[
t_{i+1} - t_i = z(u_{i+1} - u_i), \tag{6}
\]

for some \( z > 0 \).

**Theorem 1.** If the selected knots \( t_i, \ i = 1, 2, \ldots, n, \) satisfy (6), the parametric quadratic spline constructed by (2)–(4) reproduces the original parametric quadratic polynomial.

### 3. Determining knot \( t_i \)

According to (2), the knots \( t_i, \ i = 1, 2, \ldots, n, \) can be defined by

\[
t_1 = 0, \quad t_{i+1} = t_i + h_i, \quad i = 1, 2, \ldots, n - 1, \tag{7}
\]
where \( h_i \) satisfy the condition
\[
(1 - s_i)h_{i-1} = s_i h_i
\]
with \( s_i \) defined in (2). Hence, the issue is how to determine \( s_i, h_i, i = 1, 2, \ldots, n - 1 \). It will be shown later that the techniques for constructing \( s_i \) and \( h_i \) are invariant under affine transformation of the data points.

3.1. Determining \( s_i \)

To be symmetrical, five points \( P_j, i - 2 \leq j \leq i + 2 \), will be used to determine \( s_i \). Suppose that \( P_{i-1}, P_i \) and \( P_{i+1} \) are not collinear. Without loss of generality, we shall assume that the coordinates of the five points are \((x_{i-2}, y_{i-2}), (0, 1), (0, 0), (1, 0)\) and \((x_{i+2}, y_{i+2})\) (see Fig. 1). Otherwise, simply apply a transformation similar to the one in [15] to transform them into the above forms.

For the four points \( P_{i-1}, P_i, P_{i+1} \) and \( P_k \) with \( k \) being \( i - 2 \) or \( i + 2 \), let \( C_i(t) = (x(t), y(t)) \) be a parametric cubic polynomial that passes through these points at the corresponding knots. By applying the parametric transformation
\[
t = t_{i-1} + (t_{i+1} - t_{i-1})s
\]
to \( C_i(t) = (x(t), y(t)) \), one changes the corresponding knots \( t_{i-1}, t_i, t_{i+1} \) and \( t_k \) to \( 0, s_i, 1 \) and \( \tau_k = (t_k - t_{i-1})/(t_{i+1} - t_{i-1}) \), respectively, and changes \( C_i(t) \) to a function of \( s \) as follows:
\[
x = x(s) = s(s - s_i) + \frac{\lambda(s)}{\lambda(\tau_k)} \left( x_k - \frac{\tau_k(\tau_k - s_i)}{1 - s_i} \right),
\]
\[
y = y(s) = \frac{(s - s_i)(s - 1)}{s_i} + \frac{\lambda(s)}{\lambda(\tau_k)} \left( y_k - \frac{(\tau_k - s_i)(\tau_k - 1)}{s_i} \right),
\]
where
\[
\lambda(s) = s(s - s_i)(s - 1),
\]
and \( s_i \) satisfy (2).
The construction of a quadratic curve which interpolates four planar points has been studied in [9, 12]. In the following we discuss the corresponding relations between the points and their parameters. If $P_{i-1}$, $P_i$, $P_{i+1}$ and $P_k$ are points on a quadratic curve, then the cubic terms of (10) are zero, i.e.,

\begin{align}
&x_k(1 - s_i) - \tau_k(\tau_k - s_i) = 0, \\
y_k s_i - (\tau_k - s_i)(\tau_k - 1) = 0. 
\end{align}

(11)

Simple algebra shows that

\begin{align}
(1 - x_k - y_k)[(x_k + y_k)s_i^2 - 2x_k s_i] + (1 - x_k)x_k = 0, \\
\tau_k = x_k + (1 - x_k - y_k)s_i. 
\end{align}

(12)

Eq. (12) has two roots. But only one can be used to define $s_i$. If $k$ equals $i + 2$, from $\tau_{i+2} = (t_{i+2} - t_{i-1})/(t_{i+1} - t_{i-1}) > 1$ one gets the value of $s_i$, denoted by $s'_i$, as follows:

\begin{equation}
s_i = s'_i = \frac{1}{x_i + 2 + y_i + 2} \left( x_i + 2 - \sqrt{\frac{x_i + 2 y_i + 2}{x_i + 2 + y_i + 2} - 1} \right). 
\end{equation}

(13)

It follows from (11) that the condition for $0 < s_i < 1$ and $\tau_{i+2} > 1$ is

\begin{align}
x_i + 2 > 1 \quad \text{and} \quad y_i + 2 > 0,
\end{align}

(14)

i.e., $(x_i + 2, y_i + 2)$ is in the dotted region of Fig. 1 and Theorem 2 follows.

**Theorem 2.** Let $P_j = (x_j, y_j)$, $i - 1 \leq j \leq i + 2$, be four planar data points, and $P_0$ be the intersection of the line $P_{i-1}P_i$ and the line $P_{i+1}P_{i+2}$. If $P_0$ and $P_{i-1}$ are on different sides of $P_iP_{i+1}$, but $P_{i+1}$ and $P_{i+2}$ are on the same side of $P_iP_{i+1}$, then $P_i$, $i - 1 \leq j \leq i + 2$, determine a unique parametric quadratic polynomial that passes through these points successively.

In (12), if $k$ equals $i - 2$, then from the fact that $\tau_{i-2} < 0$ one gets the value of $s_i$, denoted by $s'_i$, as follows:

\begin{equation}
s_i = s'_i = \frac{1}{x_i - 2 + y_i - 2} \left( x_i - 2 + \sqrt{\frac{x_i - 2 y_i - 2}{x_i - 2 + y_i - 2} - 1} \right). 
\end{equation}

(15)

The condition for $0 < s_i < 1$ and $\tau_{i-2} < 0$ is

\begin{align}
x_i - 2 > 0 \quad \text{and} \quad y_i - 2 > 1.
\end{align}

(16)

By substituting $(x_i - 2, y_i - 2)$ and $(x_i + 2, y_i + 2)$ into (12), one gets two equations. The common root $s_i$ of these two equations, denoted by $s^m_i$, is

\begin{equation}
s_i = s^m_i = \frac{A(x_i + 2, y_i + 2)(x_i - 2, y_i - 2) - A(x_i - 2, y_i - 2)(x_i + 2, y_i + 2)}{2x_i + 2y_i - 2x_i^2 + 2y_i + 2},
\end{equation}

(17)

where

\[A(x, y) = (1 - x)x/(1 - x - y). \]

The geometric meaning of $s^m_i$ is shown in Fig. 2 (where $F = (1 - x_k - y_k)[(x_k + y_k)s_k^2 - 2x_k s_k] + (1 - x_k)x_k$, defined by (12)).
With these three possible values to define $s_i$, one option is to form $s_i$ as a linear combination of (13), (15) and (17). These values have different influences on the formation of $s_i$. The one close to 0.5 should have a bigger influence than the ones close to 0 or 1. $s_i$ is defined as follows:

$$s_i = \frac{w(s_i^t) s_i^t + w(s_i^m) s_i^m + w(s_i^r) s_i^r}{w(s_i^t) + w(s_i^m) + w(s_i^r)},$$

where $w(t) = t(1 - t)$ is a weight function whose maximum is at $t = 0.5$.

If the two quadratic curves shown in Fig. 2 have no intersection point in the interval $[0, 1]$, then $s_i^m$ in (17) does not satisfy $0 < s_i^m < 1$. In this case, $s_i$ is formed by the arithmetic mean of (13) and (15):

$$s_i = \frac{(s_i^l + s_i^r)}{2}.$$

If none of (14) and (16) hold, i.e., there does not exist $s_i$ and $\tau_k$ to make the coefficients of the cubic terms of (10) zero, then we determine $s_i$ by minimizing the sum of the squared coefficients of the cubic terms, i.e., for $k = i - 2$ or $i + 2$, if we set

$$H(s_i, \tau_k) = \frac{1}{2} \left[ \left( x_k - \frac{\tau_k (\tau_k - s_i)}{1 - s_i} \right)^2 + \left( y_k - \frac{(\tau_k - s_i)(\tau_k - 1)}{s_i} \right)^2 \right],$$

where $\lambda(s)$ is defined in (10), then the values of $s_i$ and $\tau_k$ are determined by minimizing $H(s_i, \tau_k)$. For any $0 < s_i < 1$, $H(s_i, \tau_k) \rightarrow 0$ when $\tau_k \rightarrow \infty$. This means that the solution one gets by minimizing $H(s_i, \tau_k)$ is not unique. To remove this shortcoming, the following remedy is used. For the four data points $P_{i-1}$, $P_i$, $P_{i+1}$ and $P_{i+2}$, a transformation is applied to transform their coordinates into $(0, 1), (0, 0), (1, 0)$ and $(v_{i+2}, w_{i+2})$. Let

$$v_j = \frac{t_j - t_{j-1}}{t_{i+2} - t_{i-1}}, \quad j = i - 1, i, i + 1, i + 2,$$

then a cubic polynomial $(x(v), y(v))$ passing through these four points is defined by

$$x(v) = \frac{v(v - v_i)}{v_{i+1}(v_{i+1} - v_i)} + \frac{z(v)}{\chi(1)} \frac{v_{i+2}}{v_{i+1}(v_{i+1} - v_i)},$$

$$y(v) = \frac{(v - v_i)(v - v_{i+1})}{v_i v_{i+1}} + \frac{z(v)}{\chi(1)} \frac{w_{i+2}}{v_{i+1}(v_{i+1} - v_i)},$$

(18)
where
\[ \chi(v) = v(v - v_i)(v - v_{i+1}). \]

The sum of the squared coefficients of the cubic terms of (18) is
\[ H_1(v_i, v_{i+1}) = \frac{1}{\chi(1)^2} \left[ \left( \frac{v_i}{v_i(v_{i+1} - v_i)} - 1 \right)^2 + \left( \frac{w_{i+2} - \chi(1)}{v_iv_{i+1}} \right)^2 \right]. \]

Similarly, one can transform the coordinates of the four points into \((v_i, 1), (0, v_{i+1}), (1, 0)\) and \((0, 0)\), then a cubic polynomial \((x(v), y(v))\) passing through these four points is defined by
\[ x(v) = \frac{(v - v_i)(v - v_{i+1})}{v_i(v_{i+1} - v_i)} + \frac{\zeta(v)}{\zeta(0)} \left( \frac{v_{i+1} - v_i}{v_i(v_{i+1} - v_i)} \right), \]
\[ y(v) = \frac{(v - v_{i+1})(v - 1)}{(v_i - v_{i+1})(v_i - 1)} + \frac{\zeta(v)}{\zeta(0)} \left( \frac{v_{i+1} - v_i}{(v_i - v_{i+1})(v_i - 1)} \right), \]

where
\[ \zeta(v) = (v - v_i)(v - v_{i+1})(v - 1). \]

The sum of the squared coefficients of the cubic terms of (19) is
\[ H_2(v_i, v_{i+1}) = \frac{1}{\zeta(0)^2} \left[ \left( \frac{v_{i+1}}{(v_i - v_{i+1})(v_i - 1)} - 1 \right)^2 + \left( \frac{v_i(v_{i+1} - v_i)}{(v_i - v_{i+1})(v_i - 1)} \right)^2 \right]. \]

The values of \(v_i\) and \(v_{i+1}\) are determined by minimizing the following function:
\[ H_1(v_i, v_{i+1}) + H_2(v_i, v_{i+1}). \]

We determine \(v_i\) and \(v_{i+1}\) using the following numerical method, where \(v_i\) and \(v_{i+1}\) are restricted to satisfy \(A \leq v_i \leq 1 - 2A\), \(v_{i+1} \leq v_{i+1} \leq 1 - A\), \(A\) is a small constant.

\[
\begin{align*}
&f_0 = 10^{10}; \quad \{Initialization\} \\
&\text{Initializing } A; \\
&\sigma = (1 - 3A)/50; \\
&\text{for } r_i = A \text{ to } 1 - 2A \text{ step } \sigma \text{ do} \\
&\quad \text{for } r_{i+1} = r_i + A \text{ to } 1 - A \text{ step } \sigma \text{ do} \\
&\quad \quad f_i = H_1(r_i, r_{i+1}) + H_2(r_i, r_{i+1}); \\
&\quad \quad \text{if } (f_0 > f_i) \text{ then} \\
&\quad \quad \quad \text{begin } f_0 = f_i; \quad v_i = r_i; \quad v_{i+1} = r_{i+1}; \quad \text{end}; \\
&\quad \quad \text{end; } \\
&\quad \text{end;} \\
&\tilde{s}_i = v_i/v_{i+1}; \\
\end{align*}
\]

Using \(P_{i-1}, P_i, P_{i+1}\) and \(P_{i+2}\) one can get an \(s_i\), denoted by \(\tilde{s}_i\). Using \(P_{i-2}, P_{i-1}, P_i\) and \(P_{i+1}\), one can get another \(s_i\), denoted by \(\tilde{\tilde{s}}_i\). The arithmetic mean of \(\tilde{s}_i\) and \(\tilde{\tilde{s}}_i\) is used to define \(s_i\) if none of (14) and (16) hold.
Fig. 3. \(P_i\), \(P_{i+1}\) and \(P_{i+2}\) are nearly collinear.

If \(P_{i-1}\), \(P_i\) and \(P_{i+1}\) are collinear, \(s_i\) is defined by

\[
s_i = \frac{|P_{i-1}P_i|}{|P_{i-1}P_i| + |P_iP_{i+1}|} \tag{22}
\]

The following illustration shows that this is a reasonable choice. Without loss of generality, we assume that \(P_j = (x_j,y_j), j = i-1, i, i+1\). A quadratic polynomial \((x(s), y(s))\) passing through these three points is defined by

\[
x(s) = \frac{(s - s_i)(s - 1)}{s_i}(x_{i-1} - x_i) + \frac{s(s - s_i)}{1 - s_i}(x_{i+1} - x_i) + x_i,
\]

\[
y(s) = y_i.
\]

For \(0 < s_i < 1\), \((x(s), y(s))\) is a straight line. The first derivative of \(x(s)\) being constant gives

\[
\frac{1}{s_i}(x_{i-1} - x_i) + \frac{1}{1 - s_i}(x_{i+1} - x_i) = 0.
\]

(22) now follows from the fact that \(x_i - x_{i-1} = |P_{i-1}P_i|\) and \(x_{i+1} - x_i = |P_{i+1}P_i|\).

The technique for determining \(s_i\), \(i = 3, 4, \ldots, n - 2\), is summarized as follows:

\[
s_i = \begin{cases} 
    \frac{|P_{i-1}P_i|}{|P_{i-1}P_i| + |P_iP_{i+1}|} & \text{if } P_{i-1}, P_i \text{ and } P_{i+1} \text{ are collinear,} \\
    \frac{w(s'_i)s'_i + w(s''_i)s''_i}{w(s'_i) + w(s''_i)} & \text{if } 0 < s''_i < 1, \text{ (14) and (16) hold,} \\
    (s'_i + s''_i)/2 & \text{if } s''_i(1 - s''_i) \leq 0, \text{ (14) and (16) hold,} \\
    s'_i & \text{if (16) holds, but (14) not true,} \\
    s''_i & \text{if (14) holds, but (16) not true,} \\
    (\bar{s}_i + \bar{s}_i)_i/2 & \text{else.}
\end{cases}
\tag{23}
\]

For the end points, (1) if the data points are not closed then \(s_2\) corresponding to \(C_2(s)\) is determined by (13) using four points \(P_j, j = 1, 2, 3, 4\), and similarly, \(s_{n-1}\) corresponding to \(C_{n-1}(s)\) is determined by (15); (2) if the data points are closed \((P_1 = P_n)\) then a cyclic solution is used.

Note that if the last three points of \(P_{i-1}, P_i, P_{i+1}\) and \(P_{i+2}\) are nearly collinear, \(s_i\) defined by (13) is close to 1. In this case, the constructed curve will not have the shape suggested by the data points. For example, if the coordinates of the four points are \((0,1),(0,0),(1,0)\) and \((1.5,0.01)\), respectively, then the resulting quadratic polynomial is of the shape shown in Fig. 3.
3.2. Determining $h_i$

To determine $h_i$, $i = 1, 2, \ldots, n - 1$, note that the system in (8)

$$(1 - s_i)h_{i-1} - s_i h_i = 0, \quad i = 2, 3, \ldots, n - 1,$$

has $(n - 2)$ equations but $(n - 1)$ unknowns $h_1, h_2, \ldots, h_{n-1}$. The system can be solved by adding an extra condition as follows:

$$h_1 = 1,$$

$$h_i = h_{i-1}(1 - s_i)/s_i, \quad i = 2, 3, \ldots, n - 1.$$

Although we have not seen unsatisfactory curves resulted from the above approach, it is obvious that $h_i$ and $h_j$ may have large difference when $j$ is much bigger than $i$.

Similar to [15], we recommend using two extra conditions $h_1$ and $h_{n-1}$ to solve the system. The remaining unknowns $h_2, h_3, \ldots$, and $h_{n-2}$ are determined by the least square method. Let $G(h_2, h_3, \ldots, h_{n-2})$ denote the sum of the squared deviations between $(1 - s_i)h_{i-1}$ and $s_i h_i$

$$G(h_2, h_3, \ldots, h_{n-2}) = \sum_{i=2}^{n-1} [(1 - s_i)h_{i-1} - s_i h_i]^2.$$

The $(n - 3)$ unknowns $h_2, h_3, \ldots, h_{n-2}$ are determined by minimizing the function $G(h_2, h_3, \ldots, h_{n-2})$. By setting the first partial derivative of $G(h_2, h_3, \ldots, h_{n-2})$ with respect to $h_i$ to zero:

$$\frac{\partial G(h_2, h_3, \ldots, h_{n-2})}{\partial h_i} = 0, \quad i = 2, 3, \ldots, n - 2,$$

and re-arranging the terms, we obtain

$$\begin{bmatrix} a_2 & -b_3 & 0 & \cdots & 0 \\ -b_3 & a_3 & -b_4 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & -b_{n-3} & a_{n-3} & -b_{n-2} \\ 0 & \cdots & 0 & -b_{n-2} & a_{n-2} \end{bmatrix} \begin{bmatrix} h_2 \\ h_3 \\ \vdots \\ h_{n-3} \\ h_{n-2} \end{bmatrix} = \begin{bmatrix} b_2 h_1 \\ 0 \\ \vdots \\ 0 \\ b_{n-1} h_{n-1} \end{bmatrix}, \quad (24)$$

where

$$b_i = s_i(1 - s_i), \quad i = 2, 3, \ldots, n - 1,$$

$$a_i = s_i^2 + (1 - s_{i+1})^2, \quad i = 2, 3, \ldots, n - 2.$$

If $h_1$ and $h_{n-1}$ are known, the solution of (24) is uniquely determined. One option in setting up $h_1$ and $h_{n-1}$ is shown below. If the given data points are closed, i.e., $(x_1, y_1) = (x_n, y_n)$, we simply take $h_1 = h_{n-1} = 1$. If the data points are not closed, $h_1$ and $h_{n-1}$ are determined by the following consideration: if $P_i$, $i = 1, 2, \ldots, n$, are defined by (5), then $h_1$ and $h_{n-1}$ shall satisfy the following conditions:

$$h_1 = \alpha(u_2 - u_1), \quad h_{n-1} = \alpha(u_n - u_{n-1}),$$
where $\alpha$ is a constant. These conditions make the solution of (24) to be of the following form
\[
h_i = \alpha (u_i - u_{i-1}), \quad i = 2, 3, \ldots, n - 2.
\]

Four points $P_1, P_2, P_{n-1}$ and $P_n$ are used to determine $h_1$ and $h_{n-1}$. After a parameter transformation similar to (9)
\[
t = t_1 + (t_{n-1} - t_1) s
\]
is applied to the four parameters corresponding to the four points, they become 0, $s_2, 1$ and $n$, respectively, where $s_2$ and $\tau_n$ are defined by (13) and (12) or by (21) ($h_1 = v_i, h_{n-1} = 1 - v_{i+1}$). If $s_2$ and $\tau_n$ are defined by (13) and (12), then $h_1$ and $h_{n-1}$ are taken as
\[
\begin{align*}
h_1 &= s_2 = (t_2 - t_1)/(t_{n-1} - t_1), \\
h_{n-1} &= \tau_n - 1 = (t_n - t_{n-1})/(t_{n-1} - t_1),
\end{align*}
\]
and Theorem 3 follows.

**Theorem 3.** If the knots $t_i, i = 1, 2, \ldots, n$, defined by (23)–(25) and (7) are used to construct a quadratic spline using (2)–(4), then the constructed spline reproduces parametric quadratic polynomials.

**Proof.** It is sufficient to prove that if the given data points $P_i, i = 1, 2, \ldots, n$, are defined by (5), then the determined $t_i, i = 1, 2, \ldots, n$, satisfy the following condition:
\[
t_i - t_{i-1} = \alpha(u_i - u_{i-1}), \quad i = 2, 3, \ldots, n - 2,
\]
for some $\alpha > 0$. (25) shows $\alpha = 1/(u_{n-1} - u_1)$. \(\square\)

Our experiments show that if the data points are not closed, simply taking $h_1 = h_{n-1} = 1$ also gives good results.

**Theorem 4.** The knots $t_i, i = 1, 2, \ldots, n$, defined by (7), (23) and (24) with the end conditions of setting $h_1$ and $h_{n-1}$ to (25), or simply to 1, are invariant under affine transformations of the data points.

**Proof.** It is sufficient to prove that $x_{i+2}$ and $y_{i+2}$ in (13) are invariant under affine transformations of the data points. Let $P_i$ be the projection of $P_{i+2}$ on the $w$ axis along the $v$ axis (see Fig. 1). Then $P_1 = (0, y_{i+2})$ and $P_0 = (0, y_{i+2}/(1 - x_{i+2}))$. Thus
\[
\begin{align*}
x_{i+2} &= |P_o P_{i+2}|/|P_0 P_i|, \\
y_{i+2} &= |P_0 P_{i+2}|/|P_0 P_{i+1}|.
\end{align*}
\]
Since an affine transformation does not change the ratios of these segments, $x_{i+2}$ and $y_{i+2}$ are invariant. \(\square\)

For three dimensional data points, the problem can be solved by extending the idea described in (18)–(21), with $v_i$ and $v_{i+1}$ determined by minimizing the corresponding function $H_i(v_i, v_{i+1}) +
Specifically, the coordinates of \( P_{i-1}, P_i, P_{i+1} \) and \( P_{i+2} \) are transformed into the form \((x, y, z) = (0, 1, 0), (0, 0, 0), (1, 0, 0), (v_{i+2}, w_{i+2}, 1), \) and \( H_1(v_i, v_{i+1}) \) is defined as

\[
H_1(v_i, v_{i+1}) = \frac{1}{z(1)^2} \left[ \left( \frac{v_{i+2} - v_i}{v_{i+1}(v_{i+1} - v_i)} \right)^2 + \left( \frac{w_{i+2} - v(1)}{v_{i+1}} \right)^2 + 1 \right].
\]

\( H_2(v_i, v_{i+1}) \) can be defined similarly.

4. Experiments

The new method has been implemented and compared with the chord length method, the centripetal model, the Foley/Nielson method, and the ZCM method, where the ZCM and the new methods are the versions that reproduce quadratic polynomials. In the new method, \( \lambda \) in (21) is set to 0.1 for determining \( s_i, i = 2, 3, \ldots, n - 2 \), and set to 0.01 for determining \( h_1 \) and \( h_{n-1} \). The comparison is performed on the quality of the parametric quadratic spline curves constructed using the knots determined by these methods. The chord length method is also used to determine knots in the construction of a cubic spline. The quadratic curve and cubic curve constructed by the chord length method are called chord2 and chord3, respectively.

Two types of data points are used to compare the quality of the methods. The first type of data points is taken from two primitive curves. These data points are used to compare the approximation precision of the constructed spline curves. Another type of data points is taken from \([2, 3, 8, 10]\). These data points are used to compare the shape of the constructed spline curves.

A knot computation method is considered to be better from the view point of approximation if the precision of the corresponding spline curve is better. The first data points used in comparing the approximation precision are taken from the following primitive cubic curve:

\[
F(s) = (x, y(s)),
\]

\[
x = x(s) = s(s - 1)(2s - 1)K + 3s^2(3 - 2s),
\]

\[
y = y(s) = s(1 - s)K,
\]

where \( K = 1, 2, \ldots, 12 \).

The cubic curve \( F(s) \) has the following properties: it is convex for \( K = 1, 2, 3, 4 \), it has two inflection points for \( K = 5, 6, 7, 8 \), it has one cusp for \( K = 9 \), and it has one loop for \( K = 10, 11, 12 \). Note that when \( K = 3 \), the curve becomes \( y = 3x(1 - x) \). For \( K = 3, 6, 9, 12 \), the figures of \( F(s) \) on the region \([0, 1]\) are shown in Fig. 4.

The five methods are compared on non-uniform data points generated by dividing the interval \([0, 1]\) into 20 subintervals using \( s_i \) defined as follows:

\[
s_i = [i + \lambda \sin (i * (20 - i))] / 20, \quad i = 0, 1, 2, \ldots, 20,
\]

where \( 0 \leq \lambda \leq 0.25 \).

The tangent vectors of the primitive cubic curve \( F(s) \) at the end points \( s = 0 \) and \( s = 1 \) are used as end conditions in the construction of the cubic spline. In order to offer more insight of the comparison, the knots defined by (27) are also used to construct a quadratic spline, called ‘exact spline’. The five methods and the exact spline are evaluated in terms of absolute error curve \( E(t) \).
defined by

\[ E(t) = \min \{|P(t) - F(s)|\} \]

\[ = \min \{|\tilde{P}_i(t) - F(s)|, s_{i-1} \leq s \leq s_i\}, \quad i = 0, 1, 2, \ldots, 19, \]

where \( P(t) \) is the spline curve constructed by one of the five methods or the exact spline, \( \tilde{P}_i(t) \) is the segment of \( P(t) \) on the subinterval \([t_{i-1}, t_i]\), and \( F(s) \) is the given primitive cubic curve.

When \( \lambda = 0.25 \) in (27), the maximum values of the error curve \( E(t) \) generated by the five methods and the exact spline are shown in Table 1. When \( K = 3 \), \( F(s) \) is a quadratic polynomial, the ZCM and the new methods and the exact spline can reproduce it exactly. The errors 3.30e-10 and 3.58e-10 in Table 1 are caused by computing error of the computer.

The five methods have also been compared on data points which divide \([0, 1]\) into 10, 40, \ldots etc. subintervals. The results are basically the same as those shown in Table 1.
The theoretical derivation in Sections 2 and 3 shows that when used to assign knots in the construction of polynomial interpolants to data points whose signs in convexity are the same, the new method will reproduce parametric quadratic polynomials. Therefore, we have also compared the five methods using data points taken from a convex elliptic curve:

$$x = 3 \cos(2\pi s), \quad y = 2 \sin(2\pi s).$$

The interval $[0, 1]$ is also divided into 20 sub-intervals to define data points. The maximum values of the error curve $E(t)$ generated by the five methods and the exact spline are shown in Table 2. Several methods for constructing quadratic splines to interpolate convex data points are available. The method presented in [13] can construct a visually $C^2$ ($G^2$) piecewise quadratic Bézier polynomial to interpolate convex data points.

Tables 1 and 2, as well as several other comparisons, show that (1) when the knots determined by the chord length method are used to construct splines, the resulting cubic curves have no obvious advantage in approximation precision over quadratic curves; (2) quadratic curves constructed by the new method generally have higher approximation precision than quadratic and cubic curves constructed by chord length method, as well as the ones constructed by the centripetal model and the Foley/Nielson method.

Finally, four sets of data points are used to compare the shape of the quadratic splines constructed by the five methods. The four sets of data points are

(Akima, 1970) $\{x, y\} = \{(0, 10), (2, 10), (3, 10), (5, 10), (6, 10), (8, 10), (9, 10.5), (11, 15), (12, 50), (14, 60), (15, 85);$

(Brodie, 1980) $\{x, y\} = \{(0, 1), (1, 1.1), (2, 1.1), (3, 1.2), (4, 1.3), (5, 7.2), (6, 3.1), (7, 2.6), (8, 1.9), (9, 1.7), (10, 1.6)\};$

(Fritsch and Carlson, 1980) $\{x, y\} = \{(7.99, 0), (8.09, 2.76429e-5), (8.19, 4.37498e-2), (8.7, 0.169183), (9.2, 0.469428), (10, 0.94374), (12, 0.998636), (15, 0.999919), (20, 0.999994)\}.$

(Lee, 1989) $\{x, y\} = \{(0, 0), (1.34, 5), (5, 8.66), (10, 10), (10.6, 10.4), (10.7, 12), (10.7, 28.6), (10.8, 30.2), (11.4, 30.6), (19.6, 30.6), (20.2, 30.2)\}.$

For the four sets of data points, the curves produced by the centripetal model, the Foley/Nielson method, the ZCM method and the new method are shown on the left hand side of Figs. 5–8. The curves on the right-hand side of Figs. 5–8 are interpolants to the data points $\{v_i, w_i\}$, $i = 1, 2, \ldots$, produced by applying the following transformation to the four sets of data points, $v_i = x_i$, $w_i = 0.25 \times y_i$, $i = 1, 2, \ldots$.
Fig. 5. Data points taken from (Akima, 1970), (a) Foley/Nielson method, (b) centripetal model, (c) ZCM method, (d) new method.

Fig. 6. Data points taken from (Brodlie, 1980), (a) Foley/Nielson method, (b) centripetal model, (c) ZCM method, (d) new method.

Figs. 5–8 show that the Foley/Nielson and the new methods are invariant under the affine transformation of the data points. For these four sets of data points, none of the spline curves produced by the chord length methods are visually pleasing.
Fig. 7. Data points taken from (Fritsch and Carlson, 1980), (a) Foley/Nielson method, (b) centripetal model, (c) ZCM method, (d) new method.

Fig. 8. Data points taken from (Lee, 1989), (a) Foley/Nielson method, (b) centripetal model, (c) ZCM method, (d) new method.

5. Conclusions

A new method for determining knots in parametric quadratic spline interpolation is presented. Advantages of the new method include: (1) knots determined by the new method are invariant under affine transformations of the data points; (2) quadratic spline curves constructed using knots
determined by the new method can reproduce parametric polynomials of degree two exactly. Experiments also indicate that when used to determine knots for parametric quadratic spline construction, the new method in general gives better approximation precision than the chord length method, the centripetal model, and Foley/Nielson method.

Our next work is to investigate if there is a method to determine knots so that the constructed parametric cubic splines reproduce parametric cubic polynomials.

References