Computing Suppl. 10, 199–210 (1995)

Computing

© Springer-Verlag 1995
Printed in Austria

A Quartic Spline Based on a Variational Approach

B. Hamann, Mississippi State, and **T. A. Foley,** Tempe

Abstract. The C^2 continuous cubic spline can be viewed as the solution of a variational problem. The spline derived in this paper is obtained by solving a slightly different variational problem that depends on the input data. The goal is to obtain a spline that may have high second derivatives at the interpolated points and low second derivatives between two consecutive interpolated points. The solution is a C^2 continuous quartic spline.

Key words: Approximation, calculus of variation, interpolation, spline.

1. Introduction

Commonly, the C^2 continuous cubic spline interpolating the univariate data in $\{(x_i, F_i) | i = 0, \ldots, n\}$, where $x_{i+1} > x_i$, $i = 0, \ldots, (n-1)$, is derived using calculus of variation. The objective is to minimize an energy function, which has a piecewise cubic spline as solution. When dealing with parametric curves, one must interpolate the ordered points in $\{(x_i, y_i, z_i) | i = 0, \ldots, n\}$ considering an associated knot sequence $\{u_0 < \cdots < u_n\}$. Often, the solution of the univariate case is directly applied to the three coordinate functions of a parametric spline curve.

In the univariate case, the functional

$$\int (f''(x))^2 \, dx \tag{1.1}$$

is minimized on $[x_0, x_n]$. In this paper, this functional is modified slightly, which leads to a C^2 continuous quartic spline. The modification of the functional (1.1) is motivated by the fact that the cubic spline tends to "overshoot" in regions where the given data changes rapidly. In the context of shape preservation, one wants to obtain a spline without (or at least very small) "overshoots". Thus, a different variational problem must be solved. One objective is to create a spline, which differs very little from the polygon obtained by connecting consecutive original data points. Intuitively, a spline that satisfies this criterion has high second derivatives at the knots x_i and low second derivatives between two knots x_i and x_{i+1}. Another objective is to construct a spline curve that has a second derivative that closely approximates a given function that may be dependent on the input data. It is shown in the following section how these objectives can be achieved by adding a single term to the functional (1.1).

It is not possible to list all references dealing with tension splines, monotone splines, or other shape preserving splines. The derivation of the C^2 continuous quartic uses a similar paradigm. Some of the related shape preserving curve and surface modeling concepts are discussed in [1,5–14, 16]. General references covering additional material include [3, 4, 15].

2. The Modified Variational Approach

In order to incorporate the "shape" of the polygon implied by the original data points, it is proposed to subtract a polynomial $\omega(x)$ from $f''(x)$ appearing in the functional (1.1). It is assumed that the polynomial $\omega(x)$ reflects the "desired" behavior of the second derivative of an interpolant to the given data. Therefore, the function $\omega(x)$ must be computed from a preprocessing step (Section 4) that depends on the input data.

The quartic spline is derived for the univariate case. The minimization problem to be solved considers the space $L^2[x_0, x_n]$ of functions $f''(x) - \omega(x)$ that are square-integrable on the interval $[x_0, x_n]$. The modified functional is

$$\int (f''(x) - \omega(x))^2 \, dx, \tag{2.1}$$

where $\omega(x)$ is a polynomial. Several methods for defining the polynomial $\omega(x)$ (for each interval $[x_i, x_{i+1}]$) are discussed in Section 4. This polynomial reflects the "desired" second derivative of a shape preserving interpolant.

A necessary condition for minimizing the expression (2.1) is given by Euler's equation (see [2]):

$$\frac{\partial G}{\partial f} - \frac{d}{dx}\left(\frac{\partial G}{\partial f'}\right) + \frac{d^2}{dx^2}\left(\frac{\partial G}{\partial f''}\right) - \frac{d^3}{dx^3}\left(\frac{\partial G}{\partial f'''}\right) + \cdots - \cdots = 0, \tag{2.2}$$

where, $G = G(f, f', f'', \ldots)$ is the integrand of the integral in (2.1). Substituting the integrand of (2.1) into (2.2) yields the differential equation to be solved. The integrand in (2.1) is

$$G(f, f', f'') = (f'')^2 - 2\omega f'' + \omega^2, \tag{2.3}$$

and the differential equation (2.2) becomes

$$\frac{\partial G}{\partial f} - \frac{d}{dx}\left(\frac{\partial G}{\partial f'}\right) + \frac{d^2}{dx^2}\left(\frac{\partial G}{\partial f''}\right) = \frac{d^2}{dx^2}(2f'' - 2\omega) = 2(f'''' - \omega'') = 0, \tag{2.4}$$

which is equivalent to

$$f''''(x) = \omega''(x). \tag{2.5}$$

Equation (2.5) implies that the polynomial $\omega(x)$ must be at least of degree two for $\omega''(x)$ not to vanish. Therefore, a quadratic polynomial $\omega(x)$ is used in the following. This is the lowest-degree polynomial that yields a spline different from the cubic spline. If a parametric spline curve $\boldsymbol{c}(u) = ((x(u), y(u), z(u))$ is computed, the univariate approach is applied to the three coordinate functions. Examples will be given for both the non-parametric and the parametric case.

3. Computing the Quartic Spline Segments

In this section, it is assumed that a C^0 piecewise quadratic polynomial, denoted by $\omega_i(x)$, $x \in [x_i, x_{i+1}]$, is given. Integrating Eq. (2.5) implies that each spline segment is a quartic polynomial, which is written as

$$f_i(x) = \sum_{j=0}^{4} c_{i,j}(x - x_i)^j, \quad x \in [x_i, x_{i+1}], \quad i = 0, \ldots, (n-1). \tag{3.1}$$

Each quartic spline segment is determined by continuity conditions and the quadratic functions

$$\omega_i(x) = \sum_{j=0}^{2} \bar{c}_{i,j}(x - x_i)^j, \quad x \in [x_i, x_{i+1}], \quad i = 0, \ldots, (n-1). \tag{3.2}$$

Differentiating the quadratic functions (3.2), one obtains $\omega_i''(x) = 2\bar{c}_{i,2}$. Using Eq. (2.5) yields the coefficients for the quartic terms of each spline segment:

$$c_{i,4} = \frac{1}{12}\bar{c}_{i,2}, \quad i = 0, \ldots, (n-1). \tag{3.3}$$

Continuity conditions are imposed on the quartic spline segments; these are C^0, C^1, and C^2 continuity conditions. The quadratic polynomials $\omega_i(x)$ are required to be C^0 continuous at the interior knots, *i.e.*, $\omega_{i-1}(x_i) = \omega_i(x_i)$, $i = 1, \ldots, (n-1)$. The continuity conditions for the spline segments are stated next.

The interpolation conditions imply

$$f_i(x_i) = c_{i,0} = F_i, \quad i = 0, \ldots, (n-1), \tag{3.4}$$

the conditions for C^0 continuity are

$$f_i(x_{i+1}) = f_{i+1}(x_{i+1}), \quad i = 0, \ldots, (n-2), \tag{3.5}$$

the conditions for C^1 continuity are

$$f_i'(x_{i+1}) = f_{i+1}'(x_{i+1}), \quad i = 0, \ldots, (n-2), \tag{3.6}$$

and, finally, the C^2 conditions are

$$f_i''(x_{i+1}) = f_{i+1}''(x_{i+1}), \quad i = 0, \ldots, (n-2), \tag{3.7}$$

Defining $h_i = (x_{i+1} - x_i)$, Eq. (3.7) can be written as

$$c_{i,3} = \frac{1}{3h_i}(c_{i+1,2} - c_{i,2} - 6c_{i,4}h_i^2), \quad i = 0, \ldots, (n-1). \tag{3.8}$$

The coefficients $c_{i,1}$ are derived from (3.5) and (3.8) as

$$c_{i,1} = \frac{1}{h_i}(c_{i+1,0} - c_{i,0}) - \frac{h_i}{3}(2c_{i,2} + c_{i+1,2}) + h_i^3 c_{i,4}, \quad i = 0, \ldots, (n-1). \tag{3.9}$$

A linear system of $(n-1)$ equations and $(n+1)$ unknowns is obtained for the coefficients $c_{i,2}$ using (3.6). This linear system is given by

$$h_{i-1}c_{i-1,2}+2(h_{i-1}+h_i)c_{i,2}+h_ic_{i+1,2}$$
$$=3\left(\frac{1}{h_{i-1}}c_{i-1,0}-\left(\frac{1}{h_{i-1}}+\frac{1}{h_i}\right)c_{i,0}+\frac{1}{h_i}c_{i+1,0}+h_{i-1}^3c_{i-1,4}+h_i^3c_{i,4}\right),$$
$$i=1,\ldots,(n-1). \quad (3.10)$$

Assuming that values for $c_{0,2}$ and $c_{n,2}$ are known (using some end conditions for the spline), this linear system can be written in matrix form as

$$\begin{pmatrix} 2(h_0+h_1) & h_1 & 0 & \vdots & \vdots & \vdots \\ h_1 & 2(h_1+h_2) & h_2 & 0 & \vdots & \vdots \\ 0 & h_2 & 2(h_2+h_3) & h_3 & 0 & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & 0 & h_{n-3} & 2(h_{n-3}+h_{n-2}) & h_{n-2} \\ \vdots & \vdots & \vdots & 0 & h_{n-2} & 2(h_{n-2}+h_{n-1}) \end{pmatrix} \begin{pmatrix} c_{1,2} \\ c_{2,2} \\ c_{3,2} \\ \vdots \\ c_{n-2,2} \\ c_{n-1,2} \end{pmatrix}$$

$$=3\begin{pmatrix} \frac{1}{h_0}c_{0,0}-\left(\frac{1}{h_0}+\frac{1}{h_1}\right)c_{1,0}+\frac{1}{h_1}c_{2,0}+h_0^3c_{0,4}+h_1^3c_{1,4}-h_0\underline{c_{0,2}} \\ \frac{1}{h_1}c_{1,0}-\left(\frac{1}{h_1}+\frac{1}{h_2}\right)c_{2,0}+\frac{1}{h_2}c_{3,0}+h_1^3c_{1,4}+h_2^3c_{2,4} \\ \vdots \\ \frac{1}{h_{n-3}}c_{n-3,0}-\left(\frac{1}{h_{n-3}}+\frac{1}{h_{n-2}}\right)c_{n-2,0}+\frac{1}{h_{n-2}}c_{n-1,0}+h_{n-3}^3c_{n-3,4}+h_{n-2}^3c_{n-2,4} \\ \frac{1}{h_{n-2}}c_{n-2,0}-\left(\frac{1}{h_{n-2}}+\frac{1}{h_{n-1}}\right)c_{n-1,0}+\frac{1}{h_{n-1}}c_{n,0}+h_{n-2}^3c_{n-2,4}+h_{n-1}^3c_{n-1,4}-h_{n-1}\underline{c_{n,2}} \end{pmatrix}$$
$$(3.11)$$

The two underlined terms in the first and last row of the right hand side of (3.11) indicate that the end conditions for the spline determine those values. The coefficients $c_{0,2}$ and $c_{n,2}$ can be derived from a local polynomial approximation of the given data at the ends. Obviously, the linear system of equations with its symmetric, tridiagonal, and diagonally dominant matrix can be solved in linear time.

If periodic (cyclic) interpolation of the given data is desired, end conditions need not to be specified, since the equations $f_0(x_0)=f_{n-1}(x_n)$, $f'_0(x_0)=f'_{n-1}(x_n)$, $f''_0(x_0)=f''_{n-1}(x_n)$, and $h_0=h_n$ hold. These periodicity constraints imply a linear system of equations with a cyclic matrix given by

$$\begin{pmatrix} 2(h_{n-1}+h_0) & h_0 & 0 & \vdots & \vdots & h_{n-1} \\ h_0 & 2(h_0+h_1) & h_1 & 0 & \vdots & 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & 0 & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \vdots & 0 & h_{n-3} & 2(h_{n-3}+h_{n-2}) & h_{n-2} \\ h_{n-1} & \vdots & \vdots & 0 & h_{n-2} & 2(h_{n-2}+h_{n-1}) \end{pmatrix} \begin{pmatrix} c_{0,2} \\ c_{1,2} \\ c_{2,2} \\ \vdots \\ c_{n-3,2} \\ c_{n-2,2} \\ c_{n-1,2} \end{pmatrix}$$

$$=3\begin{pmatrix} \frac{1}{h_{n-1}}c_{n-1,0}-\left(\frac{1}{h_{n-1}}+\frac{1}{h_0}\right)c_{0,0}+\frac{1}{h_0}c_{1,0}+h_{n-1}^3c_{n-1,4}+h_0^3c_{0,4} \\ \vdots \\ \frac{1}{h_{n-2}}c_{n-2,0}-\left(\frac{1}{h_{n-2}}+\frac{1}{h_{n-1}}\right)c_{n-1,0}+\frac{1}{h_{n-1}}c_{0,0}+h_{n-2}^3c_{n-2,4}+h_{n-1}^3c_{n-1,4} \end{pmatrix}. \quad (3.12)$$

Solving the linear system provides the coefficients $c_{i,2}$. The coefficients $c_{i,3}$ and $c_{i,1}$ are then computed using (3.8) and (3.9).

4. Choosing the Quadratic Functions $\omega_i(x)$

The quadratic functions $\omega_i(x)$ are defined for each interval $[x_i, x_{i+1}]$, $i=0,\ldots,(n-1)$, interpolating second derivative estimates at x_i. They are C^0 continuous at the knots. These weight functions should have exactly one zero in the interior of $[x_i, x_{i+1}]$, if it is the desire to force the final interpolant to have second derivative extrema at the knots and small second derivatives between knots. In order to define the quadratic polynomials $\omega_i(x)$, it is necessary to derive estimates for the second derivatives at all knots. Several possibilities can be used to obtain these estimates.

In order to compute the quadratic weight function $\omega_i(x)$, a quadratic polynomial $q_i(x)$ is computed that interpolates the three data in

$$\{(x_{i-1},F_{i-1}),(x_i,F_i),(x_{i+1},F_{i+1})\}, \quad i=1,\ldots,(n-1). \quad (4.1)$$

Alternatively, a quartic polynomial $r_i(x)$ is computed that interpolates the five data in

$$\{(x_{i-2},F_{i-2}),(x_{i-1},F_{i-1}),(x_i,F_i),(x_{i+1},F_{i+1}),(x_{i+2},F_{i+2})\}, \quad i=2,\ldots,(n-2). \quad (4.2)$$

By differentiating these local approximants at $x=x_i$ one obtains values $q''_i(x_i)$ (or $r''_i(x_i)$). These values are used as second derivative estimates and are set equal to $\omega_{i-1}(x_i)=\omega_i(x_i)$, leaving one degree of freedom for defining the quadratic weight function. Obviously, special care is required at the ends. The values of $q''_1(x)(r''_2(x))$ and $q''_{n-1}(x)(r''_{n-2}(x))$ are used to obtain estimates at the ends. This, of course, is no problem in the periodic case, since the data are repeated cyclically.

The use of quartic polynomials $r_i(x)$ for second derivative estimation can be usd to generate a curve scheme with quartic precision. One approach is to define

$\omega_i(x)$ as the quadratic that interpolates the three values $r_i''(x_i)$, $r_{i+1}''(x_{i+1})$, and $\frac{1}{2}(r_i(\frac{1}{2}(x_i+x_{i+1}))+r_{i+1}(\frac{1}{2}(x_i+x_{i+1})))$, $i=2,\dots,(n-3)$. Special care is required for the end intervals when computing $\omega_0(x)$, $\omega_1(x)$, $\omega_{n-2}(x)$, and $\omega_{n-1}(x)$ (non-periodic case). A simple solution is the requirement $\omega_0(x)=\omega_1(x)=r_2''(x)$ and $\omega_{n-2}(x)=\omega_{n-1}(x)=r_{n-2}''(x)$.

Forcing quartic precision is effective if the data varies smoothly. Unfortunately, using quartic polynomials for second derivative estimation might not be "local enough" in certain cases and might lead to unwanted oscillations in the final spline if the data varies rapidly. For more local results, quadratic polynomials should be used for the estimation.

Having computed second derivative estimates at all knots, the quadratic polynomials $\omega_i(x)$ can be defined by specifying one more value for each interval. To generate a spline curve with small second derivative between the interpolated points, the weight function is computed using the following case distinctions:

- If the estimates at x_i and x_{i+1} are both zero set $\omega_i(x)=0$.
- If the estimate at one knot is zero, and the estimate at the other knot is positive (negative) let $\omega_i(x)$ be the quadratic polynomial, which has a double zero at the knot where it must interpolate the zero estimate.
- If the estimates at both knots are greater (smaller) than zero let $\omega_i(x)$ be the quadratic polynomial that has a double zero in the open interval (x_i, x_{i+1}).
- If the estimate at one knot is negative, and the estimate at the other knot is positive force $\omega_i(x)$ to have a zero at $(x_i+x_{i+1})/2$.

One can introduce real-valued parameters α_i to scale the second derivative estimates at each knot x_i. Obviously, if $\alpha_i=0$, $i=0,\dots,n$, the common cubic spline is obtained; if $\alpha_i=1$, $i=0,\dots,n$, the "standard" quartic spline is obtained. Scaling the estimates of the second derivatives at the knots using increasing positive scaling factors α_i generally produces splines which approach the polygon implied by the original data. This justifies the view of these scaling factors as tension parameters. They can be used as an interactive shaping tool for the quartic spline.

Unfortunately, increasing these scaling factors beyond a certain threshold can lead to splines with unwanted inflection points or even "loops". It is not clear at this point how these factors must be chosen in order to get the most pleasing spline with maximum shape preservation. Once determined, the functions $\omega_i(x)$ define the coefficients $c_{i,4}$ according to (3.3). Examples of possible quadratic polynomials $\omega_i(x)$ are shown in Fig. 1.

In the parametric case, the method presented is applied to the three coordinate functions $x(u)$, $y(u)$, and $z(u)$ of the parametric spline curve $\mathbf{c}(u)$. Three different sets of quadratic polynomials $\omega_i(u)$ are generated, one for each coordinate.

5. Conversion to Bernstein-Bézier Representation

The C^2 continuous quartic spline has been derived in monomial form. The transformation of a single polynomial $f_i(x)$ to its corresponding Bernstein-Bézier representation is given by

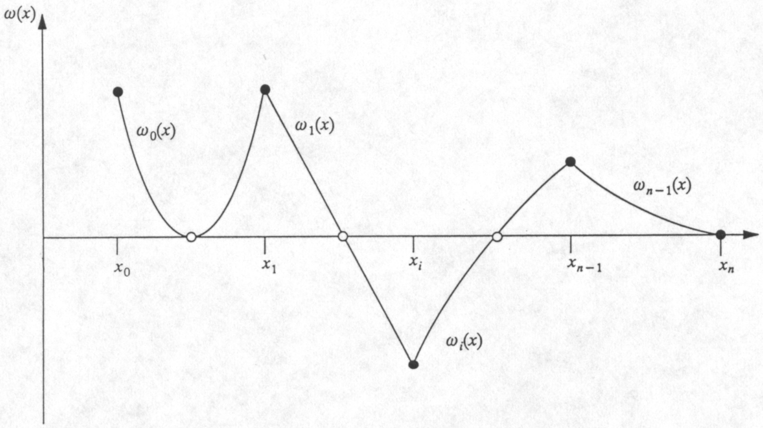

Figure 1. Possible quadratic polynomials $\omega_i(x)$

$$f_i(x)=\sum_{j=0}^{4} c_{i,j}(x-x_i)^j=\sum_{j=0}^{4} b_{i,j}B_j^4(x), \quad i=0,\dots,(n-1). \tag{5.1}$$

Here, x varies in $[x_i, x_{i+1}]$, $b_{i,j}$ are the Bézier ordinates for a single spline segment, and $B_j^4(x)$, $j=0,\dots,4$, are the quartic Bernstein-Bézier polynomials defined as

$$B_j^4(x)=\frac{1}{h_i^4}\binom{4}{j}(x_{i+1}-x)^{4-j}(x-x_i)^j, \quad i=0,\dots,(n-1). \tag{5.2}$$

In matrix notation, the conversion between monomial and Bernstein-Bézier representation is written as

$$\begin{pmatrix} c_{i,0} & -c_{i,1}x_i & +c_{i,2}x_i^2 & -c_{i,3}x_i^3 & +c_{i,4}x_i^4 \\ 0 & c_{i,1} & -2c_{i,2}x_i & +3c_{i,3}x_i^2 & -4c_{i,4}x_i^3 \\ 0 & 0 & c_{i,2} & -3c_{i,3}x_i & +6c_{i,4}x_i^2 \\ 0 & 0 & 0 & c_{i,3} & -4c_{i,4}x_i \\ 0 & 0 & 0 & 0 & c_{i,4} \end{pmatrix} = \frac{1}{h_i^4}$$

$$\cdot\begin{pmatrix} x_{i+1}^4 & -4x_ix_{i+1}^3 & 6x_i^2x_{i+1}^2 & -4x_i^3x_{i+1} & x_i^4 \\ -4x_{i+1}^3 & 4(3x_ix_{i+1}^2+x_{i+1}^3) & -12(x_i^2x_{i+1}+x_ix_{i+1}^2) & 4(x_i^3+3x_i^2x_{i+1}) & -4x_i^3 \\ 6x_{i+1}^2 & -12(x_ix_{i+1}+x_{i+1}^2) & 6(x_i^2+4x_ix_{i+1}+x_{i+1}^2) & -12(x_i^2+x_ix_{i+1}) & 6x_i^2 \\ -4x_{i+1} & 4(x_i+3x_{i+1}) & -12(x_i+x_{i+1}) & 4(3x_i+x_{i+1}) & -4x_i \\ 1 & -4 & 6 & -4 & 1 \end{pmatrix}$$

$$\cdot(b_{i,0} \quad b_{i,1} \quad b_{i,2} \quad b_{i,3} \quad b_{i,4})^T. \tag{5.3}$$

The Bézier ordinates $b_{i,j}$ can be computed directly from (5.3). Alternatively, it is possible to directly derive the quartic spline in Bernstein-Bézier representation.

6. Computing Biquartic Tensor Product Splines

In order to construct a tensor product surface method based on quartic splines, a set of cardinal basis functions is defined, where the quadratic weight functions depend on "cardinal data." More precisely, define the basis function $g_i(x)$ as the quartic spline interpolant that is zero at every knot except at x_i, where it is one. Thus, a C^2 continuous quartic spline can be written in cardinal form as

$$f(x) = \sum_{i=0}^{n} F_i g_i(x), \tag{6.1}$$

where the functions $g_i(x)$, $i = 0, \ldots, n$, are C^2 continuous quartic cardinal splines satisfying $g_i(x_k) = \delta_{i,k}$ (Kronecker delta). This linear combination interpolates the data, but it can be different from the method described earlier, where the quadratic weight functions depended on the given data. Using the cardinal form (6.1), the weight functions depend on the data $(x_k, \delta_{i,k})$.

Using the concept of cardinal spline bases, a C^2 continuous biquartic tensor product spline is written as

$$f(x, y) = \sum_{j=0}^{n} \sum_{i=0}^{m} F_{i,j} g_i(x) h_j(y) \tag{6.2}$$

interpolating the data in $\{(x_i, y_j, F_{i,j}) | x_{i+1} > x_i, y_{j+1} > y_j, i = 0, \ldots, m, j = 0, \ldots, n\}$. Again, the basis functions $g_i(x)$ and $h_j(y)$ satisfy the conditions $g_i(x_k) = \delta_{i,k}$ and $h_j(y_l) = \delta_{j,l}$. Both cardinal spline bases are computed using the univariate scheme.

In the parametric case, an interpolating parametric tensor product surface in three-dimensional space is written in cardinal form as

$$\mathbf{s}(u, v) = \sum_{j=0}^{n} \sum_{i=0}^{m} \mathbf{x}_{i,j} g_i(u) h_j(v), \tag{6.3}$$

where $\mathbf{x}_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$ is an interpolated point, and $g_i(u)$ and $h_j(v)$ are cardinal basis functons. Two increasing knot sequences are required in the parametric case, $\{u_0 < \cdots < u_m\}$ and $\{v_0 < \cdots < v_n\}$. A C^2 continuous quartic cardinal spline basis is shown in Fig. 2 using periodic end conditions, quadratic functions $\omega_i(x)$ for second derivative estimation, and tension parameters $\alpha_i = 1$.

7. Examples

The following examples demonstrate the new method for non-parametric and parametric cases. Figures 3 and 4 show C^2 continuous quartic splines/parametric spline curves obtained by increasing tension parameters α_i. The tension parameters used in Fig. 3 are 0 (= cubic spline, upper-left), 5 (upper-right), 10 (lower-left), and 15 (lower-right). For a particular spline/parametric spline curve, the tension parameters are the same at all knots. The second derivative estimates at the interior knots are based on local quadratic approximants q_i. In the non-periodic case, second derivative estimates at the end knots are zero, and natural end conditions are used

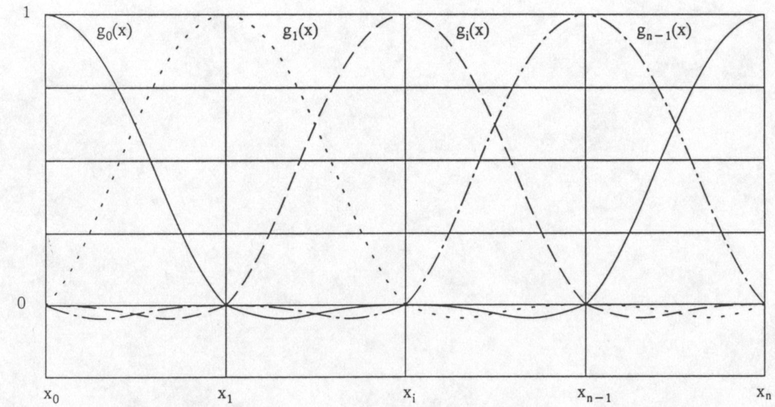

Figure 2. Cardinal spline basis functions using periodic end conditions

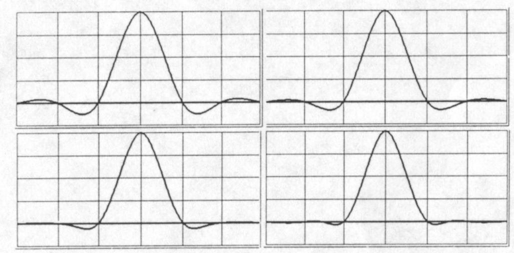

Figure 3. Quartic splines interpolating same data using increasing tension

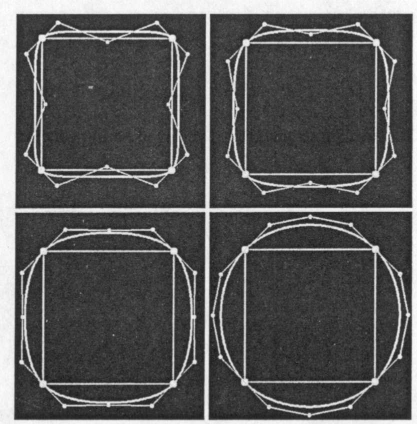

Figure 4. Quartic parametric spline curves interpolating same data using increasing tension

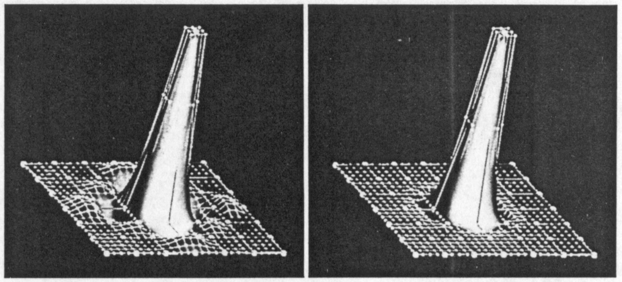

Figure 5. Product of two cubic and two quartic cardinal splines

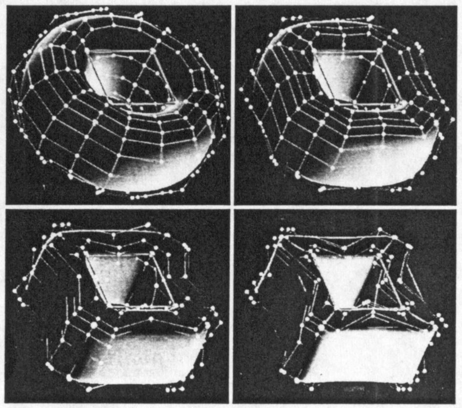

Figure 6. Modeling torus using increasing tension

for computing the quartic spline. Fig. 4 shows a periodic parametric quartic spline curve and its Bézier control polygon; the parametrization is based on chord length. The tension parameters used in Fig. 4 are 0 (= cubic spline, upper-left), 2.2 (upper-right), 4.4 (lower-left), and 6.6 (lower-right).

The next three examples show biquartic splines/parametric spline surfaces. Figure 5 shows the piecewise bicubic function (left)/piecewise biquartic function (right) defined as the product $f(x, y) = 6g_3(x)h_3(y)$ of the two cubic (left)/two quartic (right) cardinal splines $g_3(x)$ and $h_3(y)$. The tension parameters applied in x- and y-direction are 12 in the biquartic case. The knots are $x_i = i, i = 0, \ldots, 6$, and $y_j = j\ j = 0, \ldots, 6$. Figures 6 and 7 show the use of the method for interactively modeling parametric surfaces. The tension parameters used in Fig. 6 are 0 (= bicubic spline surface, upper-left), 6 (upper-right), 12 (lower-left), and 18 (lower-right). The tension parameters used in Fig. 7 are 0 (= bicubic spline surface, upper-left), 10 (upper-right), 15 (lower-left), and 20 (lower-right). All parametrizations are based on average chord

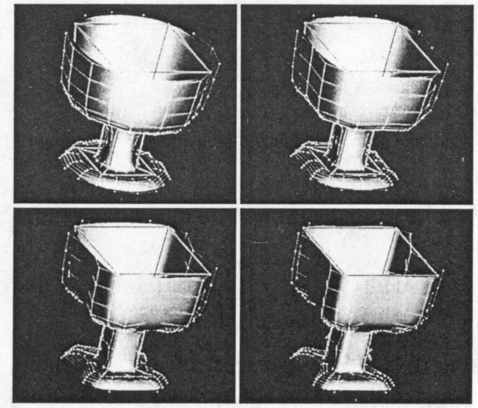

Figure 7. Modeling goblet using increasing tension

length, the same tension is applied in both parameter directions, and the associated Bézier control nets are shown.

8. Conclusions

A C^2 continuous quartic spline based on a minimization problem has been introduced. The spline is an alternative to the cubic spline. The quartic spline can be used as an interactive design tool by scaling second derivative estimates at the knots. This scaling parameter can be used to achieve interpolating curves of different shape with high second derivatives at the interpolated points. Since the quadratic polynomials ω_i depend on the knots, it is investigated how the quartic spline reacts to changes in the parametrization.

Acknowledgements

This research was supported by Department of Energy grant DE-FG02-87ER25041 and National Science Foundation grant DMS-9116930, both to Arizona State University, and by National Science Foundation grant ASC-9210439, to Mississippi State University. Algorithms were implemented on a Silicon Graphics workstation 4D/220 GTX. The authors wish to thank all members of the Computer Aided Geometric Design research group in the Computer Science Department at Arizona State University and all members of the National Grid Project team at the NSF Engineering Research Center for Computational Field Simulation at Mississippi State University for their advice and support.

References

[1] Akima, H.: A new method of interpolation and smooth curve fitting based on local procedures. J. Assoc. Comp. Mach. *17*, 589–602 (1970).
[2] Clegg, J. C.: Calculus of variations, Edinburgh: Oliver & Boyd 1968.
[3] de Boor, C.: A practical guide to splines. New York: Springer 1978.
[4] Farin, G.: Curves and surfaces for computer aided geometric design, 3rd edn. San Diego: Academic Press 1993.

[5] Foley, T. A.: Local control of interval tension using weighted splines. Comput. Aided Geom. Des. *3*, 281–294 (1986).

[6] Foley, T. A.: Interpolation with interval and point tension controls using cubic weighted ν-splines. ACM Trans. Math. Software *13*, 68–96 (1987).

[7] Foley, T. A.: A shape preserving interpolant with tension controls. Comput. Aided Geom. Des. 5, 105–118 (1987).

[8] Fritsch, F. N., Carlson, R. E.: Monotone piecewise cubic interpolation. SIAM J. Numer. Anal. *17*, 238–246 (1980).

[9] Goodman, T. N. T.: Shape preserving representations. In: Mathematical methods in computer aided geometric design (Lyche, T., Schumaker, L. L., eds.), pp. 333–351. San Diego: Academic Press 1989.

[10] Goodman, T. N. T. Unsworth, K.: Shape preserving interpolation by curvature continuous parametric curves. Comput. Aided Geom. Des. *5*, 323–340 (1988).

[11] Hagen, H., Schulze, G.: Automatic smoothing with geometric surface patches. Comput. Aided Geom. Des. *4*, 231–235 (1987).

[12] Hamann B., Farin, G., Nielson, G. M.: A parametric triangular patch based on generalized conics. In: NURBS for curve and surface design (Farin, G., ed.), pp 75–85. Philadelphia: SIAM 1991.

[13] Nielson, G. M.: Some piecewise alternatives to splines under tension. In: Computer aided geometric design (Barnhill, R. E., Riesenfeld, R. F., eds.), pp. 209–235. San Diego: Acadamic Press 1974.

[14] Salkauskas, K.: C^1 splines for interpolation of rapidly varying data. Rocky Mountain J. Math. *14*, 239–250 (1984).

[15] Schumaker, L. L.: Spline functions: basic theory. New York J. Wiley: 1961.

[16] Schweikert, D. G.: An interpolation curve using a spline in tension. J. Math. Phys. *45*, 312–317 (1996).

Dr. B. Hamann
NSF Engineering Research Center
for Computational Field Simulation
Mississippi State University
P.O. Box 6176,
Mississippi State
MS 39762, U.S.A.
or
Department of Computer Science
Mississippi State University
P.O. Drawer CS
Mississippi State
MS 39762, U.S.A.

Dr. T. A. Foley
Computer Science Department
Arizona State University
Tempe
AZ 85287-5406, U.S.A.