JOURNAL OF
COMPUTATIONAL AND
APPLIED MATHEMATICS

ELSEVIER Journal of Computational and Applied Mathematics 71 (1996) 383–393

A sinusoidal polynomial spline and its Bezier blended interpolant

K.F. Loe *

Department of Information Systems and Computer Science, National University of Singapore, Lower Kent Ridge 0511, Singapore

Received 28 June 1995; revised 19 October 1995

Abstract

Functional polynomials composed of sinusoidal functions are introduced as basis functions to construct an interpolatory spline. An interpolant constructed in this way does not require solving a system of linear equations as many approaches do. However there are vanishing tangent vectors at the interpolating points. By blending with a Bezier curve using the data points as the control points, the blended curve is a proper smooth interpolant. The blending factor has the effect similar to the "tension" control of tension splines. Piecewise interpolants can be constructed in an analogous way as a connection of Bezier curve segments to achieve C^1 continuity at the connecting points. Smooth interpolating surface patches can also be defined by blending sinusoidal polynomial tensor surfaces and Bezier tensor surfaces. The interpolant can very efficiently be evaluated by tabulating the sinusoidal function.

Keywords: Interpolant; Bezier curve; Sinusoidal spline; Tensor surface; Tension

AMS classification: 68 U05; 65 D07; 65 D10; 65 D17

1. Introduction

Curves and surfaces for geometrical design and modelling can basically be classified into two categories. For free-form design, a polygon defined by a set of control points is provided such that a curve defined in the convex hull of the polygon is obtained. For interpolating, a curve or a surface is plotted through the given data points. Free-form designs based on Bezier curves, B-splines or rational B-splines are widely used in CAD and graphics systems. There are many advantages of using Bezier curves and B-splines for free-form design. Once the control points are given, points on a curve or a surface can directly be evaluated using very efficient algorithms such as the De Casteljau algorithm. In the case of interpolating, theoretically, Lagrange polynomials can be used for directly defining a curve passing through a data set. In practice, a Lagrange interpolant has never been used. There

* e-mail: loekf@iscs.nus.sg.

SSDI 0377-0427(95)00241-3

are inherent problems in the approach. For interpolating through a large number of data points, a high degree polynomial is needed. In addition to the problem of inefficiency in evaluation, a high degree Lagrange polynomial exhibits wiggles through data points. To overcome these problems C^2 cubic splines with desirable geometrical properties were developed as discussed in [3, 10]. However cubic splines sometimes still exhibit undesirable oscillations and users do not have much control over the shape of the interpolants. Subsequently various methods have been developed to control the shape of an interpolating curve, such as those reported in [1, 2, 4, 6, 7, 9]. Barsky and Thomas [2] have developed the TRANSPLINE system which consist of different spline options from which users may select a spline that is more suitable for the type of shape control needed. For instance an exponential-based spline under tension [1] can be used to "flatten" or "tighten" curved segments between interpolating points, and piecewise cubic ν-splines [7–9] "tighten" corners of interpolating points. In [5] a weighted ν-spline with C^1 continuity as the generalization of C^2 cubic spline and ν-spline was developed with control to tighten both the intervals and interpolating points.

In this paper we introduce a new spline called sinusoidal spline which passes through control points at equal parametric intervals. Given a set of data to be interpolated, the spline can be constructed without having to solve a system of linear equations. It is unlike the cubic spline and many other spline interpolants which require solving for the control points from a system of linear equations. It seems that the sinusoidal spline is a good candidate for data interpolation because it can easily be constructed. However it has an undesirable feature; the spline has a vanishing tangent vector at the uniform parametric value intervals corresponding to the data points. Vanishing tangent vectors are usually considered as an undesirable feature in curve design as cusps may appear at those points. However we show that vanishing tangents at the data points are in fact an advantage when a sinusoidal spline is blended with a noninterpolatory spline such as a Bezier curve or a B-spline using the interpolatory data to define the convex hull. The blended curves can be made to pass through the data points as interpolants. The shape of a blended interpolant depends on the blending parameter which has an effect similar to the "tension" or ν control parameter in tension spline or ν-spline [5, 8, 11]. Such a blended interpolant is invariant under affine transformation.

In Section 2 we define and study the characteristics of sinusoidal splines. Section 3 introduces the blending of a sinusoidal spline and a Bezier curve for constructing a smooth blended interpolant. Section 4 discusses constructing composite interpolants by blending sinusoidal splines and Bezier curve segments. In Section 5 we generalize interpolating curves to tensor product interpolating surface. Section 6 is the conclusion.

2. Sinusoidal polynomial and spline

We define a family of sinusoidal functions as follows.

Definition 1. Sinusoidal functions are defined as follows:

$$A_{i,r}^{\hat{n}}(t) = \cos^2\left(\frac{\widehat{n}\pi}{\widehat{i}+1}\left(t - \frac{r}{\widehat{n}}\right)\right) \tag{1}$$

where n, $0 \leqslant i \leqslant n$ and $0 \leqslant r \leqslant \widehat{n}$ are integers, and $\widehat{k} = 2^k - 1$.

Proposition 2.

$$A^{\hat{n}}_{\hat{i},r}(t) = A^{\hat{n}}_{\hat{i},r+\hat{i}+1}(t) \quad \textit{for} \quad r = 0,1,\ldots,\hat{i}. \tag{2}$$

Proposition 3.

$$A^{\hat{n}}_{\hat{i},r}(t) + A^{\hat{n}}_{\hat{i},r+\widehat{i-1}+1}(t) = 1 \quad \textit{for} \quad r = 0,1,\ldots,\widehat{i-1}. \tag{3}$$

Proposition 4. *The first derivative of a sinusoidal function is*

$$(A^{\hat{n}}_{\hat{i},r}(t))' = -\frac{\hat{n}\pi}{\hat{i}+1}\left(2A^{\hat{n}}_{\hat{i},r}\left(t - \frac{\widehat{i-2}+1}{\hat{n}}\right) - 1\right).$$

Proposition 2 is a basic property of cosine function, i.e., $\cos^2(\theta+\pi) = \cos^2\theta$. Proposition 3 can be derived from the trigonometry identity $\cos^2(\theta-\pi/2) = \sin^2\theta$ and $\cos^2\theta+\sin^2\theta = 1$. Proposition 4 is obtained by differentiating cosine square and applying the trigonometry identity $2\cos\theta\sin\theta = \sin 2\theta$ and $\cos 2\theta = 2\cos^2\theta - 1$.

Lemma 5. *The first derivative of a sinusoidal function $A^{\hat{n}}_{\hat{i},r}(t)$ is zero at $t = r/\hat{n}$.*

Proof. From Proposition 4, when $t = r/\hat{n}$ we get

$$\begin{aligned}(A^{\hat{n}}_{\hat{i},r}(t))' &= -\frac{\hat{n}\pi}{\hat{i}+1}\left(2A^{\hat{n}}_{\hat{i},r}\left(\frac{\widehat{i-2}+1}{\hat{n}}\right) - 1\right)\\ &= -\frac{\hat{n}\pi}{\hat{i}+1}\left(2\cos^2\left(\frac{\pi}{4}\right) - 1\right)\\ &= 0. \qquad \square\end{aligned}$$

Definition 6. A degree $\hat{s}$ polynomial of sinusoidal functions is a functional polynomial of $A^{\hat{n}}_{\hat{i},r}(t)$ defined as follows:

$$S^{\hat{s}}_r(t) = \prod_{i=1}^{s} A^{\hat{s}}_{\hat{i},r}(t).$$

Lemma 7. *A summation of sinusoidal functional polynomials from $r = 0$ to $r = \hat{s}$ equals unity, i.e.,*

$$\sum_{r=0}^{\hat{s}} S^{\hat{s}}_r(t) = 1.$$

Proof. Define

$$R_k = \sum_{r=0}^{\hat{k}} \prod_{i=1}^{k} A^{\hat{s}}_{\hat{i},r}(t).$$

Obviously

$$R_s = \sum_{r=0}^{\hat{s}} S_r^{\hat{s}}(t).$$

Thus to prove the lemma is to show $R_s = 1$. We shall give an inductive proof as follows. For $k = 1$

$$\begin{aligned}
R_1 &= \sum_{r=0}^{\hat{1}} S_r^{\hat{s}}(t)\\
&= A_{\hat{1},0}^{\hat{s}}(t) + A_{\hat{1},1}^{\hat{s}}(t)\\
&= \cos^2 \frac{\widehat{s}\pi}{2} t \;+\; \cos^2 \frac{\widehat{s}\pi}{2}\left(t - \frac{1}{\widehat{s}}\right)\\
&= \cos^2 \frac{\widehat{s}\pi}{2} t \;+\; \sin^2 \frac{\widehat{s}\pi}{2} t = 1.
\end{aligned}$$

Assuming $R_{s-1} = 1$ is true for $k = s - 1$.

Now

$$\begin{aligned}
R_s &= \sum_{r=0}^{\hat{s}} \prod_{i=1}^{s} A_{\hat{i},r}^{\hat{s}}(t)\\
&= \sum_{r=0}^{\hat{s}} \left(\prod_{i=1}^{s-1} A_{\hat{i},r}^{\hat{s}}(t)\right) A_{\hat{s},r}^{\hat{s}}(t)\\
&= \sum_{r=0}^{\widehat{s-1}} \left(\prod_{i=1}^{s-1} A_{\hat{i},r}^{\hat{s}}(t)\right) A_{\hat{s},r}^{\hat{s}}(t) \;+\; \sum_{r=\widehat{s-1}+1}^{\hat{s}} \left(\prod_{i=1}^{s-1} A_{\hat{i},r}^{\hat{s}}(t)\right) A_{\hat{s},r}^{\hat{s}}(t)\\
&= \sum_{r=0}^{\widehat{s-1}} \left(\prod_{i=1}^{s-1} A_{\hat{i},r}^{\hat{s}}(t)\right) A_{\hat{s},r}^{\hat{s}}(t) + \sum_{r=0}^{\widehat{s-1}} \left(\prod_{i=1}^{s-1} A_{\hat{i},r+\hat{i}+1}^{\hat{s}}(t)\right) A_{\hat{s},r+\widehat{s-1}+1}^{\hat{s}}(t)\\
&= \sum_{r=0}^{\widehat{s-1}} \left(A_{\hat{s},r}^{\hat{s}}(t) + A_{\hat{s},r+\widehat{s-1}+1}^{\hat{s}}(t)\right) \prod_{i=1}^{s-1} A_{\hat{i},r}^{\hat{s}}(t) \quad \text{(by Proposition 2)}\\
&= \sum_{r=0}^{\widehat{s-1}} \prod_{i=1}^{s-1} A_{\hat{i},r}^{\hat{s}}(t) \quad \text{(by Proposition 3)}\\
&= R_{s-1} = 1.
\end{aligned}$$

This property is sometimes known as the Cauchy condition. Satisfying this condition ensures that a spline constructed with $S_r^{\hat{n}}(t)$ as basis functions is invariant under affine transformations.

Lemma 8. *The derivative of $S_r^{\hat{s}}(t)$ is zero at $t = r/\widehat{s}$.*

Proof. Based on Proposition 4, it can easily be shown that the first derivative of $S_r^{\hat{s}}(t)$ is given by

$$(S_r^{\hat{s}}(t))' = -\sum_{i=1}^{s} \frac{\hat{s}\pi}{\hat{i}+1} \prod_{j=1}^{s} \left[(1+\delta_{ij}) A_{\hat{j},r}^{\hat{s}} \left(t - \frac{\widehat{j-2}+1}{\hat{s}} \delta_{ij} \right) - \delta_{ij} \right],$$

where $\delta_{ij} = 1$ if $i = j$, else $\delta_{ij} = 0$. Based on Lemma 5 the proof immediately follows. □

Definition 9. A sinusoidal polynomial spline curve is defined with the sinusoidal polynomials $S_r^s(t)$ as the basis functions as follows:

$$S^{\hat{s}}(t) = \sum_{r=0}^{\hat{s}} Q_r S_r^{\hat{s}}(t),$$

where Q_r are some position vectors in the 3D space. According to this definition and Lemma 7 (Cauchy condition) a sinusoidal polynomial spline is invariant under affine transformations. In the following we shall establish a very useful theorem.

Theorem 10. *A sinusoidal polynomial spline given by Definition* 9 *satisfies the following discrete condition*:

$$S^{\hat{s}}\left(\frac{r}{\hat{s}}\right) = Q_r, \tag{4}$$

for $r = 0, 1, 2, \ldots, \hat{s}$.

Proof. Consider

$$A_{\hat{i},m}^{\hat{s}}\left(\frac{r}{\hat{s}}\right) = \cos^2 \frac{\pi(r-m)}{\hat{i}+1}.$$

For $m = r$ we get $A_{\hat{i},r}^{\hat{s}}(r/\hat{s}) = \cos^2 0 = 1$. Thus according to Definition 6, we have $S_r^{\hat{s}}(r/\hat{s}) = 1$.
Based on the unity condition of Lemma 7 we get $S_m^{\hat{s}}(r/\hat{s}) = 0$ when $m \neq r$.
Thus according to Definition 9 for $S^{\hat{s}}(t)$, the theorem immediately follows. □

This theorem implies that a sinusoidal spline passes through all the points Q_r. However there is an undesirable property in the sinusoidal spline given by the following theorem.

Theorem 11. *The tangent vector of a sinusoidal polynomial* $S^{\hat{s}}(t)$ *vanishes at* $t = r/\hat{s}$, *i.e.*, $(S^{\hat{s}}(t))' = 0$ *for* $r = 0, 1, \ldots, \hat{s}$.

Proof. Differentiating $S^{\hat{s}}(t)$ we get the tangent of the spline as follows:

$$(S^{\hat{s}}(t))' = \sum_{r=0}^{\hat{s}} Q_r (S_r^{\hat{s}}(t))'.$$

According to Lemma 8, $(S_r^{\hat{s}}(t))' = 0$ at $t = r/\hat{s}$, thus the theorem immediately follows. □

This theorem shows that a sinusoidal polynomial spline is not a regular curve with vanishing tangents at the interpolatory points (S_r for $r = 0, 1, \ldots, \hat{s}$). On the other hand, based on the geometrical properties of a sinusoidal spline given by the two theorems, a proper blended interpolant can be constructed by blending a sinusoidal spline with a well-behaved noninterpolatory curve such as a Bezier curve or a B-spline. In the following we shall only study interpolants formed by blending sinusoidal polynomial splines with Bezier curves. They can easily be extended to B-spline blended interpolants.

3. Interpolants formed by blending sinusoidal splines and Bezier curves

The blending of a degree $\widehat{s}$ sinusoidal spline and a same degree Bezier curve is defined as follows:

$$C^{\hat{s}}(t) = (1-\alpha)S^{\hat{s}}(t) + \alpha B^{\hat{s}}(t), \tag{5}$$

where $0 < \alpha < 1$ is called the blending factor. A Bezier curve of degree $\widehat{s}$ is given by

$$B^{\hat{s}}(t) = \sum_{r=0}^{\hat{s}} P_r B_r^{\hat{s}}(t),$$

where P_r $(r = 0, 1, \ldots, \widehat{s})$ denote the control points of the Bezier curve. We let the P_r be the set of given interpolatory points. Recall that

$$S^{\hat{s}}(t) = \sum_{r=0}^{\hat{s}} Q_r S_r^{\hat{s}}(t).$$

To have the blended curve pass through the data points P_r we need to solve for Q_r.

Based on Theorem 10 we get

$$P_r = (1-\alpha)Q_r + \alpha B^{\hat{s}}\left(\frac{r}{\widehat{s}}\right)$$

for $r = 0, 1, \ldots, \widehat{s}$. That is

$$Q_r = \frac{1}{1-\alpha}\left(P_r - \alpha B^{\hat{s}}\left(\frac{r}{\widehat{s}}\right)\right).$$

Thus Q_r are obtained by evaluating $B^{\hat{s}}(t)$ at $t = r/\widehat{s}$.

We shall consider a cubic polynomial sinusoidal spline given by $\widehat{s} = \widehat{2} = 3$, which can be explicitly written as

$$\begin{aligned} S^3(t) = {} & Q_0 \cos^2\frac{3\pi t}{2}\cos^2\frac{3\pi t}{4} + Q_1 \cos^2\frac{3\pi(t-1/3)}{2}\cos^2\frac{3\pi(t-1/3)}{4} \\ & + Q_2 \cos^2\frac{3\pi t}{2}\cos^2\frac{3\pi(t-2/3)}{4} + Q_3 \cos^2\frac{3\pi(t-1/3)}{2}\cos^2\frac{3\pi(t-1)}{4}. \end{aligned}$$

A cubic Bezier curve with four control points P_0, P_1, P_2 and P_3 is given by

$$B^3(t) = P_0(1-t)^3 + 3P_1(1-t)^2 t + 3P_2(1-t)t^2 + P_3 t^3.$$

The blending of the above two curves with blending factor α is

$$C^3(t) = (1-\alpha)S^3(t) + \alpha B^3(t).$$

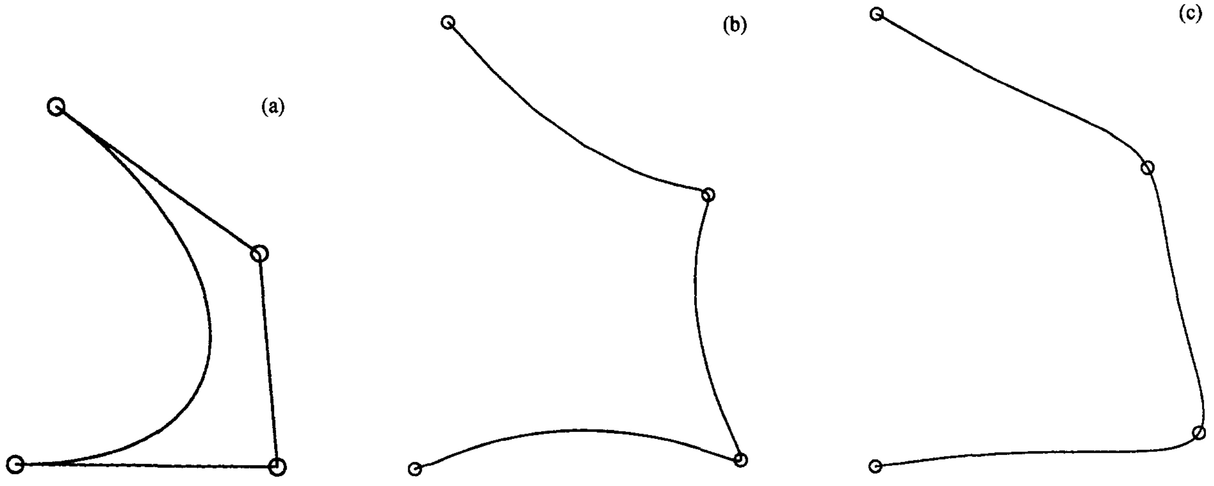

Fig. 1. (a) Bezier curve, (b) $\alpha = 0.3$ blended and (c) $\alpha = 0.9$ blended interpolants.

The $S^3(t)$ can be evaluated as efficiently as $B^3(t)$ if we tabulate the sinusoidal function $\cos^2((\pi/2)t)$ for $0 \leqslant t \leqslant 1$. Thus $C^3(t)$ can be evaluated efficiently.

Now let $C(t)$ equal the data points P_0 to P_3 at $t = 0, \frac{1}{3}, \frac{2}{3}, 1$, then Q_0 to Q_3 of $S^3(t)$ are given by

$$Q_i = \frac{1}{1-\alpha}(P_i - \alpha B^3(i/3)), \quad i = 0, \ldots, 3.$$

Fig. 1(a) is a Bezier curve defined in the convex hull of the control point polygon formed by P_0 to P_3 which are the data points to be interpolated. Fig. 1(b) is an interpolant passing through P_0 to P_3 constructed by blending a sinusoidal spline and a Bezier curve with a blending factor $\alpha = 0.3$, the interpolant is somewhat concave. Fig. 1(c) is the same interpolant with a blending factor $\alpha = 0.9$ that smoothly passes the data P_0 to P_3. It shows that increasing the blending factor α from 0.3 to 0.9 has the effect of tightening the interpolating intervals, similar to increasing tension value in a tension spline.

4. Composite interpolants

When drawing a Bezier curve over a set of control points, it is preferred to use a lower degree Bezier curve for efficiency reasons. A lowest degree smooth Bezier space curve is a cubic curve with four control points. When more control points are involved, segments of the cubic Bezier curve can be connected with C^1 continuity imposed at the connecting points. To satisfy the C^1 continuity at the connecting point of two cubic Bezier curves denoted as $P^1(t)$ and $P^2(t)$, the following linear condition must be satisfied:

$$\Delta_2(P_2^1 - P_c) = \Delta_1(P_c - P_1^2), \tag{6}$$

where $P_c = P_3^1 = P_0^2$ is the connecting point of the two curve segments. Δ_1 and Δ_2 are the parameter intervals between P_2^1 and P_c, and P_c and P_1^2 respectively.

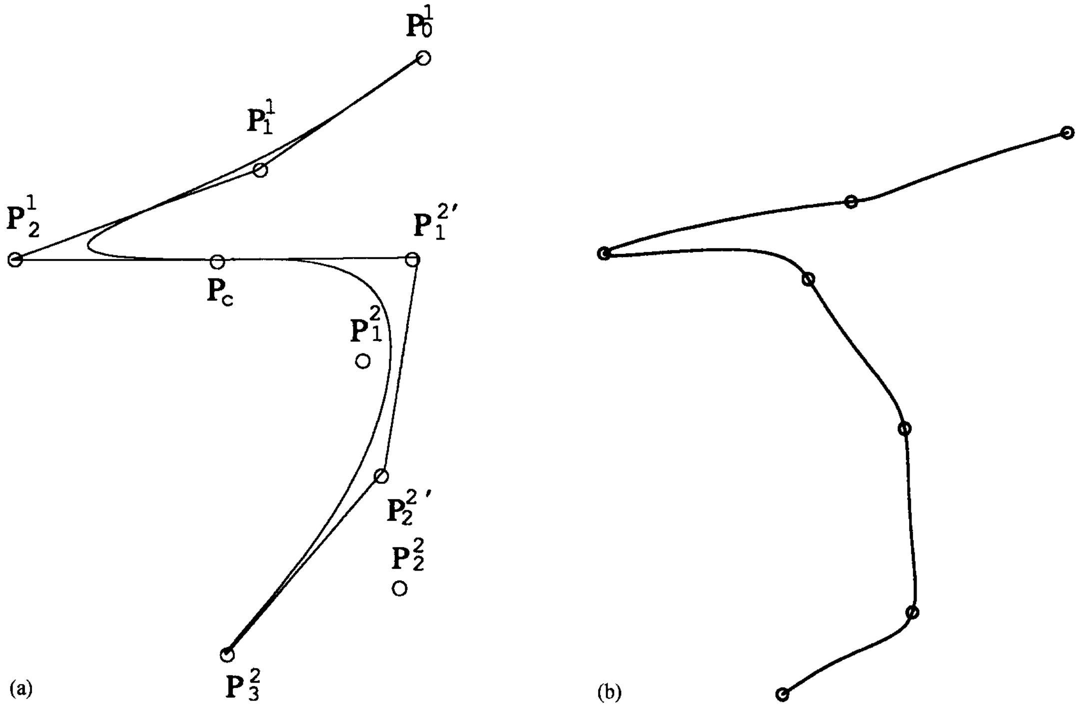

Fig. 2. Two composite segments of an interpolant.

It is also possible to define an interpolant by blending sinusoidal spline segments and Bezier curve segments. A C^1 continuity at a connecting point of two segments is completely defined by two Bezier curve segments because sinusoidal splines have zero derivatives at their end points as given by Theorem 11. If we require that every segment of an interpolant must pass through four data points, then the above condition for C^1 continuity would not usually be satisfied because three data points may not be collinear. Even when they are collinear, the C^1 continuity condition may still not be satisfied. This problem can be solved by introducing two complementary control points $(P_1^2)'$ and $(P_3^2)'$ to replace the data points as the control points for the next connecting Bezier curve segments as shown in Fig. 2(a).

Thus the second control point of the next Bezier curve segment is obtained from the above equation by replacing P_1^2 by $(P_1^2)'$ such that

$$(P_1^2)' = P_c - \frac{\Delta_2}{\Delta_1}(P_2^1 - P_c),$$

where Δ_1 is decided by the parameter interval of the control points of the previous segment and Δ_2 is given by

$$\Delta_2 = \frac{\Delta_1 |P_c - P_1^2|}{|P_2^1 - P_c|},$$

and P_1^2 is a data point to be interpolated, generally not equal to $(P_1^2)'$ as shown in Fig. 2(a). The next control point of the Bezier curve $(P_2^2)'$ is given by

$$(P_2^2)' = \frac{P_1^2 + P_2^2}{2}.$$

Fig. 2(b) shows the interpolant.

5. Tensor product interpolating patches

We can also construct tensor product interpolating patches in a similar way as we construct Bezier tensor surface patches. A bicubic Bezier patch is defined by a set of 4×4 control points forming a mesh denoted as P_{ij} with $i, j = 0, \ldots, 3$ given as follows,

$$B(u, v) = \sum_{i=0}^{3} \sum_{j=0}^{3} P_{ij} B_i(u) B_j(v),$$

where $B_i(u)$ and $B_j(v)$ are Bernstein polynomials.

We can define a sinusoidal patch in a similar way as

$$S(u, v) = \sum_{i=0}^{3} \sum_{j=0}^{3} Q_{ij} S_i(u) S_j(v),$$

where $S_i(u)$ and $S_j(v)$ are sinusoidal polynomials given by Definition 6. Without causing ambiguity we have omitted the superscripts which denote the degree of the polynomial.

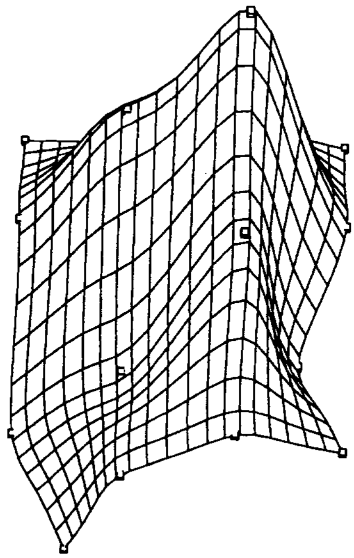

Fig. 3. Interpolating tensor surface.

Thus an interpolatory surface can be defined as follows:

$$C(u,v) = (1-\alpha)S(u,v) + \alpha B(u,v).$$

Now, the Q_{ij} can be obtained as follows:

$$Q_{ij} = \frac{1}{1-\alpha}(P_{ij} - \alpha B(i/3, j/3)), \quad i,j = 0,\ldots,3.$$

Fig. 3 shows a tensor interpolating surface patch defined by $C(u,v)$ passing through a set of data points.

6. Conclusion

A spline constructed by using sinusoidal polynomials as basis functions is proposed. The properties of the spline are studied. The most important properties are that it is an interpolating spline, and it possesses vanishing tangent values at the interpolating points. Such properties show that, standing alone, the spline is an improper interpolant. However, it is precisely because of these properties that the spline blends very well with Bezier curves to define interpolatory splines with blending factors which has a similar effect as the tension control parameters of tension splines. A blended interpolant is geometrically intuitive because its tangent at each data point is parallel to the underlying blending Bezier curve with the magnitute of the tangent equal to the Bezier tangent vector scaled by the blending factor at the corresponding point. Low degree such as cubic interpolant can be obtained by connecting blended curve segments satisfying C^1 continuity. Tensor surface can also be defined in an analogous way. To construct an interpolant using sinusoidal polynomial blending approach, no system of linear equations needs to be solved to obtain the control points as many interpolating techniques do. Also the blending factor provides an additional dimension for controlling the shape of the interpolant. The interpolants can very efficiently be evaluated if we tabulate the sinusoidal function.

Acknowledgements

I wish to thank Prof E. Goto for his hospitality when I was on leave in the RIKEN Goto Laboratory where part of the work was done. I also wish to thank C.L. Tai for comments and beneficial discussions.

References

[1] B.A. Barsky, Exponential and polynomial methods for applying tension to an interpolating spline curve, *Comput. Vision Graph. Image Process.* **27** (1984) 1–18.
[2] B.A. Barsky and S.W. Thomas, Transpline — a system for representing curves using transformations among four spline formulations, *Comput. J.* **24** (1981) 271–277.
[3] C. De Boor, *A Practical Guide to Splines* (Springer, New York, 1978).
[4] T.A. Foley, Local control of interval tension using weighted splines, *Comput. Aided Geometric Design* **3** (1986) 281–294.

[5] T.A. Foley, Interpolation with interval and point tension controls using cubic weighted ν-splines, *ACM Trans. Math Software* **13** (1987) 68–96.
[6] J.A. Gregory and M. Sarfraz, A rational cubic spline with tension, *Comput. Aided Geometric Design* **7** (1990) 1–13.
[7] G.M. Nielson, Some piecewise polynomial alternatives to splines in tension, in: R.E. Barnhill and R.F. Riesenfeld, Eds., *Computer Aided Geometric Design* (Academic Press, New York, 1975) 209–235.
[8] G.M. Nielson, A locally controllable spline with tension for interactive curve design, *Comput. Aided Geometric Design* **1** (1984) 199–205.
[9] G.M. Nielson, Rectangular ν-splines, *IEEE Comput. Graphic Appl.* **6** (2) (1986) 35–40.
[10] L.L. Schumaker, *Spline Functions: Basic Theory* (Wiley, New York, 1981).
[11] D.G. Schweikert, An interpolation curve using a spline in tension, *J. Math. Phys.* **45** (1966) 312–317.