Interactive shape preserving interpolation by curvature continuous rational cubic splines

Chris Seymour and Keith Unsworth

Research Report No:99/01
January 1999

ISSN 1174-6696
Applied Computing, Mathematics and Statistics

The Applied Computing, Mathematics and Statistics Group (ACMS) comprises staff of the Applied Management and Computing Division at Lincoln University whose research and teaching interests are in computing and quantitative disciplines. Previously this group was the academic section of the Centre for Computing and Biometrics at Lincoln University.

The group teaches subjects leading to a Bachelor of Applied Computing degree and a computing major in the Bachelor of Commerce and Management. In addition, it contributes computing, statistics and mathematics subjects to a wide range of other Lincoln University degrees. In particular students can take a computing and mathematics major in the BSc.

The ACMS group is strongly involved in postgraduate teaching leading to honours, masters and PhD degrees. Research interests are in modelling and simulation, applied statistics, end user computing, computer assisted learning, aspects of computer networking, geometric modelling and visualisation.

Research Reports

Every paper appearing in this series has undergone editorial review within the ACMS group. The editorial panel is selected by an editor who is appointed by the Chair of the Applied Management and Computing Division Research Committee.

The views expressed in this paper are not necessarily the same as those held by members of the editorial panel. The accuracy of the information presented in this paper is the sole responsibility of the authors.

This series is a continuation of the series “Centre for Computing and Biometrics Research Report” ISSN 1173-8405.

Copyright

Copyright remains with the authors. Unless otherwise stated permission to copy for research or teaching purposes is granted on the condition that the authors and the series are given due acknowledgement. Reproduction in any form for purposes other than research or teaching is forbidden unless prior written permission has been obtained from the authors.

Correspondence

This paper represents work to date and may not necessarily form the basis for the authors’ final conclusions relating to this topic. It is likely, however, that the paper will appear in some form in a journal or in conference proceedings in the near future. The authors would be pleased to receive correspondence in connection with any of the issues raised in this paper. Please contact the authors either by email or by writing to the address below.

Any correspondence concerning the series should be sent to:

The Editor
Applied Computing, Mathematics and Statistics Group
Applied Management and Computing Division
PO Box 84
Lincoln University
Canterbury
NEW ZEALAND

Email: computing@lincoln.ac.nz
Interactive shape preserving interpolation by curvature continuous rational cubic splines

by

Chris Seymour and Keith Unsworth

Abstract

A scheme is described for interactively modifying the shape of convexity preserving planar interpolating curves. An initial curve is obtained by patching together rational cubic and straight line segments. This scheme has, in general, geometric continuity of order $2$ ($C^2$ continuity) and preserves the local convexity of the data. A method for interactively modifying such curves, while maintaining their desirable properties, is discussed in detail. In particular, attention is focused upon local changes to the curve, while retaining $C^2$ continuity and shape preserving properties. This is achieved by interactive adjustment of the Bézier control points, followed by automatic adjustment of the values of weights and curvatures in a prescribed manner. A number of examples are presented.

§1. Introduction

Suppose $I_i = (x_i, y_i), i = 0, \ldots, N$, are data points in the plane. In [4], Goodman suggests a scheme for generating a curve which interpolates these data, and which preserves the shape of the data by possessing the minimum number of inflection points compatible with the data. Such an interpolation scheme is said to be local convexity preserving, an idea which is discussed in more detail in [8]. In addition the scheme has the following properties which may also be desirable.

(1) The curve has, in general, geometric continuity of order 2 ($C^2$ continuity), i.e. the unit tangent vector and the curvature vary continuously along the curve. Note that the continuity of the curvature will not hold in certain situations involving collinear data, for example it will not be continuous at $I_i$ in the case that, say, $I_{i-2}, I_{i-1}, I_i$ are collinear but $I_{i-1}, I_i, I_{i+1}$ are not. This is discussed in more detail in §4.

(2) The method is local, i.e. a change in one of the data points or the addition of a new data point will only affect the shape of the curve in a small neighbourhood of this point.

(3) The curve is invariant under a rotation of the coordinate axes or a change in scale.
(4) The method is stable in the sense that a small change in the data, tangent directions or curvatures will produce a correspondingly small change in the shape of the curve.

(5) The method may be used to generate either closed curves, or open curves with appropriate end conditions.

The method for generating the interpolating curve has similarities with those of [3,8,9], in that a tangent vector $\mathbf{T}_i$ and a curvature value $\kappa_i$ need to be specified at each data point $I_i$ before the curve can be constructed. In [8] the interpolating curve is obtained by patching together polynomial segments of a certain degree $n(\geq 2)$ and straight line segments. In the cases in which $n \geq 3$, the curve has all of the properties listed above, with the curvature at each data point being zero. As a result, curves may be generated which contain “flat spots” and therefore may not be “visually pleasing”. The method described in [3,9] overcomes this problem for the case $n = 3$, allowing the curve to have a non-zero curvature value at each data point which is the end point of a non-linear segment. (Curvature discontinuity subsequently results in situations as described in (1) above, although [9] describes a technique for overcoming this, at the expense of stability). The construction of these curves, however, requires a positive solution of a system of two non-linear quadratic equations on each cubic segment. In order to guarantee that such a system has a unique positive solution, bounds are placed on the magnitudes of the curvatures at the data points at each end of the relevant curve segment, and the values of these curvatures must subsequently be chosen in order to satisfy these bounds. This requirement clearly restricts the choice of curvature values, and undesirably large curvature values may have to be used, resulting in curves which turn more sharply at a data point than is desired.

In the scheme outlined in [4], the curve is constructed by piecing together rational cubic and straight line segments. The use of rational cubic segments removes the need to solve a non-linear system and hence allows any non-zero values for the magnitudes of the curvatures at the data points. Discontinuity of the curvature still occurs in circumstances as described above. Furthermore, use of rational cubics allows the possibility of the interpolating curve being a conic: in particular the scheme can be used to construct an interpolating curve which reproduces circular arcs, provided that no arc between consecutive data points is larger than a semi-circle. The method has also been applied in [7] to the situation in which the interpolating curve is bounded by one or more straight line barriers.

Thus, the scheme in [4] produces curves with many desirable properties. (See [14] for further comments about the method.) Let us now consider its use in an interactive design situation; that is, if the curve produced by this method is treated as an initial attempt at an interpolating curve, can this curve be subsequently modified interactively by a user to give a more desirable shape? Such a scheme should have the following properties.

(P1) $C^2$ continuity should be preserved.

(P2) The curve must retain its local convexity preserving properties.
(P3) The effect on the shape of the curve must be intuitive to the user.
(P4) Changes to the shape of the curve must be as local as possible.
(P5) The scheme should be stable: i.e. as long as a small change in the data does not lead to curve segments changing their type (see §3), this should lead to a correspondingly small change in the way that the modification scheme can affect the shape of the curve.

We describe such a scheme in this paper. However, prior to considering the shape modification problem, we extend the work of [4] slightly to give a full description of the ways in which the Bézier points are specified for the initial curve.

A brief description of the paper is as follows. §§2, 3 and 4 give brief details of the initial values for the curvatures and tangent directions, and the subsequent initial assignment of Bézier points and weights. More details may be found in [4]; the brief description here is for completeness and to describe a small modification to the way in which the Bézier points may be calculated for a curve segment with an inflection (see §4). The method is extended to deal with open curves in §5. The scheme for the interactive modification of the curve is discussed in §6. Sample output is presented and discussed in §7. The paper is brought to a conclusion in §8.

There are many methods in the literature for generating interpolating curves. In particular, the method of Piegl[13] is closely related to the work of this paper, while that of Kaklis and Sapidis[12] offers insight into possible future work. Both are briefly discussed below.

In [13], Piegl considers the generation and interactive manipulation of parametric interpolating curves, in which the curves are made up of either rational quadratic or rational cubic Bézier segments. In the quadratic case, once an initial curve is generated, the shape of a curve segment may be subsequently modified by adjustment of the segment’s shoulder point. Two methods are offered in order to achieve the modification. One allows the shoulder point to be either pulled towards, or pushed away from, the inner control point. These push/pull operations allow the entire range of conic sections to be represented by the segment. The second method specifies the location of the shoulder point through which the curve must pass. For cubics, the second method is also offered, and is suggested as the preferred scheme to the corresponding push/pull scheme for cubics. In each case, once the new position for the shoulder point has been established, new weight(s) are calculated, and the curve segment is completely determined. The scheme has $G^1$ continuity. The methodology described in [13] is similar to that of this paper, in as much as an initial curve is generated and subsequently modified interactively. However, we deal with the cubic $G^2$ case, with shape modification achieved by moving the Bézier points and curvature continuity retained throughout.

Kaklis and Sapidis[12] consider the problem of convexity preserving interpolation using a piecewise polynomial approximation to the exponential spline in tension. The degrees of the polynomial curve segments need not be the same, and an iterative algorithm is presented that first initialises the
degrees, and then allows each to be incremented. The iterative process terminates as soon as the criteria for a visually pleasing interpolant, that also preserves the convexity of the data, are satisfied. It is shown that this interpolant is obtained after a finite number of iterations, and that the degrees are the smallest that can produce this interpolant. The effect of increasing the degree of a segment is analogous to applying "tension" to that segment, with the resulting tightening effect. This approach has since been applied successfully to space curves by Kaklis and Karavelas in [11]. We will return to this topic in the final section of this paper.

Schaback presents a survey of a number of parametric interpolation methods using piecewise rationals in [14] and includes the method of [4] in this study. Examples of the application of rational splines to non-parametric shape preserving interpolation may be found in [1,10]. The former describes a local \( C^1 \) scheme, and the latter a global \( C^2 \) scheme. [10] also contains a number of other references to the work of Gregory and Delbourgo.

§2. Interpolation between two data points

Consider a rational cubic arc in \( \mathbb{R}^2 \). It is shown in [4] that the arc may be represented, using the Bézier representation, in the form

\[
\mathbf{r}(t) = \frac{\mathbf{A}\alpha(1-t)^3 + \mathbf{B}t(1-t)^2 + \mathbf{C}t^2(1-t) + \mathbf{D}\beta t^3}{\alpha(1-t)^3 + t(1-t)^2 + t^2(1-t) + \beta t^3}, \quad 0 \leq t \leq 1, \tag{2.1}
\]

where \( \alpha, \beta \in \mathbb{R} \) and \( \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D} \in \mathbb{R}^2 \), these four points forming the corresponding Bézier polygon.

Assuming \( \alpha, \beta \neq 0 \), it follows from (2.1) that,

\[
\mathbf{r}(0) = \mathbf{A}, \quad \mathbf{r}(1) = \mathbf{D}, \tag{2.2}
\]
\[
\mathbf{r}'(0) = (\mathbf{B} - \mathbf{A})/\alpha, \quad \mathbf{r}'(1) = (\mathbf{D} - \mathbf{C})/\beta, \tag{2.3}
\]

so that the arc interpolates the end-points \( \mathbf{A} \) and \( \mathbf{D} \), while the directions of the tangents at \( \mathbf{A} \) and \( \mathbf{D} \) are parallel to the vectors \( \mathbf{B} - \mathbf{A} \) and \( \mathbf{D} - \mathbf{C} \) respectively.

Next consider the values of the curvatures \( K \) and \( L \) at \( \mathbf{A} \) and \( \mathbf{D} \) respectively. In order to do this some notation is introduced. Given two vectors \( \mathbf{A} = (A_1, A_2), \mathbf{B} = (B_1, B_2) \), define

\[
\mathbf{A} \times \mathbf{B} := A_1B_2 - A_2B_1 = |\mathbf{A}| |\mathbf{B}| \sin \theta
\]

where \( \theta \) represents the angle from \( \mathbf{A} \) to \( \mathbf{B} \). Then it can be shown [2, p. 175] that

\[
K = \frac{2\alpha[(\mathbf{B} - \mathbf{A}) \times (\mathbf{C} - \mathbf{B})]}{|\mathbf{B} - \mathbf{A}|^3}, \quad L = \frac{2\beta[(\mathbf{C} - \mathbf{B}) \times (\mathbf{D} - \mathbf{C})]}{|\mathbf{D} - \mathbf{C}|^3}, \tag{2.4}
\]

where the curvature is positive(negative) if the curve is turning anti-clockwise (clockwise).
Now consider two consecutive data points $I_i$ and $I_{i+1}$ and suppose tangent vectors $T_i$ and $T_{i+1}$ and curvature values $\kappa_i$ and $\kappa_{i+1}$ have been assigned at these points. For some $\gamma_i, \delta_i > 0$ the curve between $I_i$ and $I_{i+1}$ is defined by (2.1) with

$$A = I_i, \quad D = I_{i+1}, \quad B - A = \gamma_i T_i, \quad D - C = \delta_i T_{i+1}. \quad (2.5)$$

It follows from (2.2) and (2.3) that this curve interpolates $I_i$ and $I_{i+1}$ and has tangents at each of these points in the directions of $T_i$ and $T_{i+1}$ respectively.

Now suppose that the arc also interpolates the curvature values $\kappa_i, \kappa_{i+1}$ at each end-point respectively. A rearrangement of (2.4) gives

$$\alpha = \frac{\kappa_i |B - A|^3}{2[(B - A) \times (C - B)]}, \quad \beta = \frac{\kappa_{i+1} |D - C|^3}{2[(C - B) \times (D - C)]}, \quad (2.6)$$

and it follows therefore that if $A, B, C, D$ are known, values for $\alpha, \beta$ may be obtained from (2.6), and the right hand side of (2.1) is fully determined. Note that in order for (2.6) to make sense neither $A, B, C$ nor $B, C, D$ must be collinear.

Furthermore, assuming that neither $T_i$ nor $T_{i+1}$ is parallel to $I_{i+1} - I_i$, the convexity preserving properties require that at $I_i$ and $I_{i+1}$ the curve must turn towards the line joining $I_i$ and $I_{i+1}$, i.e.

$$\kappa_i [T_i \times (I_{i+1} - I_i)] > 0, \quad \kappa_{i+1} [(I_{i+1} - I_i) \times T_{i+1}] > 0. \quad (2.7)$$

(2.7) ensures that $\alpha, \beta > 0$, and that the curve satisfies the well known convex hull property. This is discussed further in §4 when choosing the curvature values.

It follows from the above discussion that in order to fully determine the curve, values for $\gamma_i$ and $\delta_i$ need to be chosen. This is considered in the next section.

**§3. The Initial Bézier Points**

The choices of values for $\gamma_i$ and $\delta_i$ for each curve segment depend upon a segment’s type, i.e. whether it is a convex segment or an inflection segment.

First, angles $a$ and $b$ are defined as follows:

$$\sin a = \frac{T_i \times (I_{i+1} - I_i)}{|T_i||I_{i+1} - I_i|}, \quad \sin b = \frac{(I_{i+1} - I_i) \times T_{i+1}}{|I_{i+1} - I_i||T_{i+1}|}.$$ 

It follows that

$$\sin(a + b) = \frac{T_i \times T_{i+1}}{|T_i||T_{i+1}|}.$$
Case 1. $\kappa_i \kappa_{i+1} > 0$: convex segment.

From (2.5) and (2.7) this implies that B and C lie on the same side of line joining A and D. Now in order for the final interpolating curve to be local convexity preserving it is required that for this particular case, the arc (2.1) should be convex. From the variation diminishing property of the Bézier representation this will be guaranteed if the polygonal arc ABCD is convex. Now consider the case in which $a + b < \pi$.

Referring to figure 1(a), by ensuring that the Bézier point B (resp. C) lies between A and E (resp. D and E), a convex segment is guaranteed. In addition, if $I_i, T_i, \kappa_i, I_{i+1}, T_{i+1}, \kappa_{i+1}$ arise from the arc of a circle which is no larger than a semi-circle, (2.1) should reproduce this arc. It is shown in [4] that both of these conditions can be satisfied with $|B - A|$ and $|D - C|$ chosen as follows:

$$|B - A| = \lambda_i |I_{i+1} - I_i|, \quad |D - C| = \mu_i |I_{i+1} - I_i|,$$

where

$$\begin{align*}
\lambda_i &= \frac{1}{m_i + (1 - m_i)|I_{i+1} - I_i|/2|\sin b| + |r_b|}, & 0 < m_i \leq 1, \\
\mu_i &= \frac{1}{n_i + (1 - n_i)|I_{i+1} - I_i|/2|\sin a| + |r_a|}, & 0 < n_i \leq 1, \\
r_a &= \sin(a + b)/\sin a, \\
r_b &= \sin(a + b)/\sin b.
\end{align*}$$

(3.2)

It is suggested in [4] that $m_i, n_i$ may be useful as input parameters in order to alter the shape of the curve. However, extensive testing with tangent directions and curvature values chosen as described in §4, failed to clarify what the effect of varying these “shape parameters” would be on an arbitrary set of data. (The one situation in which a certain forecast can be made is that in which the data come from a circle, since in this case variation of the parameters will have no effect whatsoever!) In practice therefore a fixed value of, say, $m_i = n_i = 0.5$ is used for each segment of the initial curve with subsequent shape modification carried out as described in §6.

For the cases in which $a + b = \pi$ (figure 1(b)) and $a + b > \pi$ (figure 1(c)), the need for convexity places no constraints upon the Bézier points. However, in case 1(b), an arc of a circle should be reproduced if possible, and case 1(c) should be continuous with that of 1(b). Hence (3.1), (3.2) are used in both cases.

Case 2. $\kappa_i \kappa_{i+1} < 0$: inflection segment.

In this case B and C lie on opposite sides of the line joining A and D. From the variation diminishing property it follows that the curve will have a single point of inflection. Hence arcs of circles cannot be reproduced and the curve will clearly not be convex. Two separate case are now considered depending upon the relative magnitudes of the angles $a$ and $b$. 
(a) \(|a| = |b|\)

See figure 1(d). Since the two tangent directions are parallel, there are no restrictions upon the locations of the Bézier points, and a simple rule for locating them would be:

\[ |\mathbf{B} - \mathbf{A}| = \lambda_i |\mathbf{I}_{i+1} - \mathbf{I}_i|, \quad |\mathbf{D} - \mathbf{C}| = \mu_i |\mathbf{I}_{i+1} - \mathbf{I}_i|, \quad 0 < \lambda_i, \mu_i < 0.5. \tag{3.3} \]

The upper bounds on \(\lambda_i\) and \(\mu_i\) which are suggested here are designed to ensure that the initial curve does not exhibit any sharp turns when \(|a|\) and \(|b|\) are small.

(b) \(|a| > |b|\)

Referring to figure 1(e), by ensuring that the Bézier point \(\mathbf{C}\) lies between \(\mathbf{D}\) and \(\mathbf{E}\), it is guaranteed that the curve segment will have an inflection point. In addition continuity with case 2(a) is also required. Both of these conditions can be satisfied with the following choices:

\[ |\mathbf{B} - \mathbf{A}| = \lambda_i |\mathbf{I}_{i+1} - \mathbf{I}_i|, \quad |\mathbf{D} - \mathbf{C}| = \mu_i |\mathbf{I}_{i+1} - \mathbf{I}_i|, \tag{3.4} \]

where

\[ \mu_i = \frac{\lambda_i}{1 + |r_a|} \]

and \(\lambda_i\) is as in (3.3).

The case in which \(|a| < |b|\) is completely analogous to this.

§4. Tangent directions and curvatures

The discussion in §3 assumes that values for \(T_i, \kappa_i, i = 0, \ldots, N\) have been assigned at the data points, with the resulting curve interpolating these values. We will now consider these assignments in order that the interpolating curve is local convexity preserving. The following discussion is appropriate for both closed and open curves with \(\mathbf{I}_{N+j} = \mathbf{I}_{j-1}\) and \(\mathbf{I}_{-j} = \mathbf{I}_{N-j+1}, j = 1, 2, 3\) for the closed curve case, (similarly for \(T_i\) and \(\kappa_i\)) and \(i \in [1, N - 1]\) for the open curve case. The cases \(i = 0, N\) for the open curve case are considered in §5.

As explained in [8], the curve is local convexity preserving provided

\[ T_i = a_i (\mathbf{I}_i - \mathbf{I}_{i-1}) + b_i (\mathbf{I}_{i+1} - \mathbf{I}_i), \quad a_i, b_i \geq 0, \tag{4.1} \]

where \(a_i = 0\), and hence \(T_i\) is in the direction of \(\mathbf{I}_{i+1} - \mathbf{I}_i\), if and only if \(\mathbf{I}_i, \mathbf{I}_{i+1}, \mathbf{I}_{i+2}\) are collinear and \(b_i = 0\), and hence \(T_i\) is in the direction of \(\mathbf{I}_i - \mathbf{I}_{i-1}\), if and only if \(\mathbf{I}_i, \mathbf{I}_{i-1}, \mathbf{I}_{i-2}\) are collinear. It is also required that if \(\mathbf{I}_{i-1}, \mathbf{I}_i, \mathbf{I}_{i+1}\) are not collinear then the curve should turn towards the polygonal arc \(\mathbf{I}_{i-1} \mathbf{I}_i \mathbf{I}_{i+1}\) at \(\mathbf{I}_i\), i.e.

\[ \kappa_i [(\mathbf{I}_i - \mathbf{I}_{i-1}) \times (\mathbf{I}_{i+1} - \mathbf{I}_i)] > 0. \]
First consider the choice of curvature values $\kappa_i$. In order to allow the possibility of the interpolating curve reproducing circular arcs, it is appropriate to choose each $\kappa_i$ to be the curvature of the circle passing through $I_{i-1}, I_i, I_{i+1}$, i.e. [2, p. 67],

$$\kappa_i = \frac{2(I_i - I_{i-1}) \times (I_{i+1} - I_i)}{|I_i - I_{i-1}| |I_{i+1} - I_i||I_i - I_{i+1}|}.$$  \hspace{1cm} (4.2)

Now suppose, assuming an appropriate value for $i$ for an open curve, $I_{i-2}, I_{i-1}, I_i$ are collinear. These three points would be connected by straight line segments, with corresponding zero curvature. If, however, $I_{i-1}, I_i, I_{i+1}$ are not collinear then $\kappa_i$, calculated from (4.2), will clearly be non-zero. Thus, this interpolation scheme has the same property as the method described in [3,9], namely that there is a discontinuity in the curvature at those data points which are at the join of a linear segment and a non-linear segment. Methods for overcoming this situation are discussed in [9], although as noted in that paper the price for retaining $C^2$ continuity may be loss of stability. In this paper, we have simply accepted the fact that we will only have $C^1$ continuity at such points.

As explained in [4], choosing $T_i$ based upon the same criterion as for $\kappa_i$, i.e. the circle passing through $I_{i-1}, I_i, I_{i+1}$, generates tangent vectors which do not satisfy the conditions stated immediately following (4.1) regarding collinear data. Instead we choose $T_i$ according to (4.1) with $a_i$ and $b_i$ as follows,

$$a_i = |\kappa_{i+1}| |I_{i+1} - I_i|^2, \quad b_i = |\kappa_{i-1}| |I_i - I_{i-1}|^2,$$  \hspace{1cm} (4.3)

with the non-zero curvature value selected at a point of curvature discontinuity.

This means that if $I_{i-2}, \ldots, I_{i+2}$ lie on a circular arc then (4.1), (4.3) give a tangent vector which has the same direction as that of this circular arc, again assuming an appropriate value for $i$ for the open curve case.

With the tangent directions and curvatures given by (4.1)-(4.3), the Bézier points and weights $\alpha, \beta$ may be derived from (2.5), (2.6), (3.1)-(3.4), with the user supplying values for $m_i, n_i$ in (3.2), and $\lambda_i, \mu_i$ in (3.3), (3.4). Hence the curve segment given by (2.1) is fully determined, providing the user with an initial closed curve.

Examples of two closed curves produced by this method are shown in figures 2 and 4. The corresponding curvature plots are shown in figures 3 and 5 respectively. Figure 2 is generated by interpolating three points which lie on a circle; an exact circle is reproduced, as shown by figure 3.

§5. Open Curves

To complete the definitions of the tangent directions and curvatures, the values of these quantities must be specified at $I_0$ and $I_N$ in the case of open curves. We will give the appropriate definitions for the case $I_0$, the values at $I_N$ are specified in an analogous manner.
First we consider $\kappa_0$. If $\mathbf{I}_0, \mathbf{I}_1, \mathbf{I}_2$ are collinear then $\kappa_0 = 0$, otherwise it has the value of the curvature of the circle passing through these three points. Thus $\kappa_0 = \kappa_1$, the latter being calculated from (4.2).

Now consider $\mathbf{T}_0$. If $\mathbf{I}_0, \mathbf{I}_1, \mathbf{I}_2$ are collinear then $\mathbf{T}_0$ has the same direction as $\mathbf{I}_1 - \mathbf{I}_0$, otherwise it has the same direction as the tangent to the circle which passes through these three points. We may write this as [2, p. 67],

$$\mathbf{T}_0 = (|\mathbf{I}_2 - \mathbf{I}_0|^2 - |\mathbf{I}_1 - \mathbf{I}_0|^2)(\mathbf{I}_1 - \mathbf{I}_0) - |\mathbf{I}_1 - \mathbf{I}_0|^2(\mathbf{I}_2 - \mathbf{I}_1).$$

An example of an open curve produced by this method is shown in figure 6, with the corresponding curvature plot shown in figure 7.

§6. Modifying the shape of the curve interactively

We now consider the problem of interactively modifying the shape of this initial curve, while satisfying the properties (P1)-(P5) given in §1. We will assume that the tangent directions do not change. Hence shape modification is achieved by adjusting one or more Bézier points interactively, with the weights and the curvature values then modified automatically in a manner which seeks to satisfy the specified properties.

6.1 Adjustment of the Bézier Points

For an arbitrary curve segment, we consider adjusting the location of $\mathbf{B}$. The treatment for $\mathbf{C}$ is completely analogous. From (3.1), (3.3), (3.4), we write, dropping the $i$ subscript,

$$\lambda = \frac{|\mathbf{B} - \mathbf{A}|}{|\mathbf{D} - \mathbf{A}|}, \quad \mu = \frac{|\mathbf{D} - \mathbf{C}|}{|\mathbf{D} - \mathbf{A}|} \tag{6.1}$$

Thus, the location of Bézier point $\mathbf{B}$ may be altered by a change in the value of $\lambda$. However, the precise amount by which this quantity can change will depend upon the type of curve segment under consideration. With reference to §3, there are two cases to consider. The first is that in which $\mathbf{B}$ is unconstrained, as in figures 1(b), 1(c), 1(c) and 1(e), while in the second case $\mathbf{B}$ is constrained as in figures 1(a) and the analogous case of 1(e) when $|a| < |b|$.

Unconstrained movement

Let the new value of $\lambda$ be $\tilde{\lambda}$. Since the only restriction on the movement of $\mathbf{B}$, along the given tangent direction, is that it does not cross the line $\mathbf{AD}$, we may write

$$\tilde{\lambda}(k) = k_\lambda \lambda, \quad k_\lambda > 0. \tag{6.2}$$

Thus, the user may specify any positive value for the “shape parameter” $k_\lambda$ in order to provide a new location for $\mathbf{B}$, using (6.1) and (6.2). Initially $k_\lambda = 1$, and $\mathbf{B}$ moves towards (resp. away from) $\mathbf{A}$ for $k_\lambda < 1$ (resp. $k_\lambda > 1$).
Constrained movement

In this case, \( B \) must be adjusted so that

\[ |B - A| < |E - A| \]

always remains true. As with the unconstrained case, we again allow the user to input any positive value for \( k_\lambda \). The new value of \( \lambda \) is now defined as

\[
\tilde{\lambda}(k_\lambda) = k_\lambda \lambda, \quad \begin{cases} \lambda \leq 1, \\ \lambda > 1, \end{cases} \quad 0 < k_\lambda \leq 1, \quad k_\lambda > 1, \quad (6.3)
\]

with \( r_b \) defined in (3.2). (6.3) defines a positive monotonically increasing \( C^1 \) function \( \tilde{\lambda} \) such that

\[ \tilde{\lambda} \to 1/r_b \text{ as } k_\lambda \to \infty. \]

That is,

\[ B \to E \text{ as } k_\lambda \to \infty. \]

For \( k_\lambda < 1 \), the effect is as in the unconstrained case. See figure 8 for a graph of \( \tilde{\lambda}(k_\lambda) \).

The treatment of \( C \) may be derived in a similar way, with \( \lambda, r_b, k_\lambda \) replaced by \( \mu, r_a, k_\mu \) in (6.3).

Hence (6.2), (6.3) offer a scheme for the interactive modification of the Bézier points, depending upon the type of curve segment which is being modified. Input of a positive value for \( k_\lambda \) (resp. \( k_\mu \)) will move \( B \) (resp. \( C \)) towards \( A \) (resp. \( D \)) if \( k_\lambda < 1 \) (resp. \( k_\mu < 1 \)), and in the opposite direction if \( k_\lambda > 1 \) (resp. \( k_\mu > 1 \)), in such a way that the local convexity of the curve is always maintained. Having adjusted the Bézier points, consideration must now be given to the curvature values \( K, L \), and the weights \( \alpha, \beta \), in order to satisfy properties (P1)-(P5).

6.2 Adjustment of the Curvatures and Weights

(a) Convex curve: \( a + b < \pi \)

Consider the convex curve segment of the type shown in figure 1(a), and depicted in more detail in figure 9. From (3.2) and (6.1) note that

\[ \lambda r_b = \frac{|B - A|}{|E - A|}, \quad \mu r_a = \frac{|D - C|}{|D - E|}. \]

From (2.4), in which \( \theta \) represents the angle from \( B - A \) to \( C - B \), it follows that

\[
\frac{K}{2\alpha} = \frac{|C - B| \sin \theta}{|B - A|^2} = \frac{|C - E| \sin(a + b)}{|B - A|^2} = \frac{(1 - \mu r_a)|D - E| \sin(a + b)}{|B - A|^2} = \frac{|D - C|}{|D - A|} \left( \frac{1 - \mu r_a}{\lambda^2} \right). \]
Similarly
\[ \frac{L}{2\beta} = \frac{\sin b}{|D - A|} \left[ \frac{1 - \lambda r_b}{\mu^2} \right]. \]

Hence, if the positions of B and/or C are adjusted as described in §6.1, it follows that
\[ K/\alpha \propto (1 - \mu r_a)/\lambda^2, \quad L/\beta \propto (1 - \lambda r_b)/\mu^2. \quad (6.4) \]

In seeking to maintain \( G^2 \) continuity, we must ensure that the values for \( K \) and \( L \) remain finite for all values of \( \lambda, \mu \). Also, in this case, we would expect \( K \) and \( L \) to be affected in a symmetric fashion by changes in \( \lambda, \mu \). Thus we write
\[
\begin{align*}
K &= K_{\lambda\mu}(1 - \mu r_a)^{m_1}(1 - \lambda r_b)^{m_2}, \\
L &= L_{\lambda\mu}(1 - \mu r_a)^{m_2}(1 - \lambda r_b)^{m_1},
\end{align*}
\]

where \( K_{\lambda\mu}, L_{\lambda\mu} \) are constants of proportionality that may be computed from the initial curve. We note immediately that \( K \leq K_{\lambda\mu}, L \leq L_{\lambda\mu} \) for all possible locations of \( B \) and \( C \). It follows from (6.4), (6.5) that
\[
\begin{align*}
\alpha &= C_\alpha \lambda^2(1 - \mu r_a)^{m_1-1}(1 - \lambda r_b)^{m_2}, \\
\beta &= C_\beta \mu^2(1 - \mu r_a)^{m_2}(1 - \lambda r_b)^{m_1-1},
\end{align*}
\]

where the constants of proportionality may be written as
\[
C_\alpha = \frac{K_{\lambda\mu}|D - A|}{2 \sin a}, \quad C_\beta = \frac{L_{\lambda\mu}|D - A|}{2 \sin b}. \]

Similar to the above observation with regard to the curvatures, we note that \( \alpha \leq C_\alpha/r_a^2, \beta \leq C_\beta/r_a^2 \) for all possible locations of \( B \) and \( C \).

Thus, the numerator of (2.1) may be written as
\[
A C_\alpha \lambda^2(1 - \mu r_a)^{m_1-1}(1 - \lambda r_b)^{m_2}(1 - t)^3 + ((1 - \lambda r_b)A + \lambda E)t(1 - t)^2
+ ((1 - \mu r_a)D + \mu E)t^2(1 - t) + DC_\beta \mu^2(1 - \mu r_a)^{m_2}(1 - \lambda r_b)^{m_1-1}t^3,
\]

which may be rewritten as
\[
A(1 - t)^2(1 - \lambda r_b)(C_\alpha \lambda^2(1 - \mu r_a)^{m_1-1}(1 - \lambda r_b)^{m_2-1}(1 - t) + t)
+ E t(1 - t) (\lambda (1 - t) + \mu t).
+ D t^2(1 - \mu r_a)(1 - t + C_\beta \mu^2(1 - \mu r_a)^{m_2-1}(1 - \lambda r_b)^{m_1-1}t).
\]

The denominator may be written as
\[
C_\alpha \lambda^2(1 - \mu r_a)^{m_1-1}(1 - \lambda r_b)^{m_2}(1 - t)^3 + t(1 - t)^2 + t^2(1 - t)
+ C_\beta \mu^2(1 - \mu r_a)^{m_2}(1 - \lambda r_b)^{m_1-1}t^3.
\]
This shows that as $\lambda \to 1/r_b$, the corresponding changes in $B$ and $\alpha$ have the effect of pulling the curve from $A$ to $E$. Similarly as $\mu \to 1/r_a$, the curve is pulled from $D$ to $E$. As $\lambda, \mu \to 0$, the curve is pulled towards the straight line $AD$. From (6.5), it follows that in this case the curvatures at $A$ and $D$ approach $K_{\lambda \mu}$ and $L_{\lambda \mu}$ respectively, and as noted above, remain bounded. We note that $B$ is not allowed to coincide with $A$, since the curve would reduce to a rational quadratic and would no longer be $G^1$, because the tangent at $A$ would then be in the direction of $C - A$. In addition, from (6.6), we have the further complication that $\alpha = 0$, which would mean that strictly the curve would be undefined at $A$.

In order to maintain $G^2$ continuity following adjustment of $K$ and/or $L$, Bézier points and/or weight values on each of the adjacent curve segments must also be changed. In order to keep the changes to the shape of the curve as local as possible, $G^2$ continuity is maintained in this case by modifying one weight value on each adjacent segment.

Suppose the initial values of the curvatures are $K_0, L_0$. Following the adjustment of the Bézier point(s), suppose $\tilde{K}, \tilde{L}$ are the new values for the curvatures.

Now consider the segment prior to the one that has been modified. We will use the $l$ superscript to denote the corresponding quantities on this segment. Let $\beta^l_0$ be the initial value for one of its weights. Since neither $B^l$ nor $C^l$ has been adjusted, it follows from (2.4) that $\beta^l$ must be modified to $\tilde{\beta}^l$ where

$$\tilde{\beta}^l = \tilde{K} \beta^l_0 / K_0.\quad (6.7)$$

Similarly, for the segment which follows, using the $r$ superscript,

$$\tilde{\alpha}^r = \tilde{L} \alpha^r_0 / L_0.\quad (6.8)$$

Thus from (6.5)

$$\tilde{\beta}^l \propto (1 - \mu r_a)^{m_1} (1 - \lambda r_b)^{m_2},$$
$$\tilde{\alpha}^r \propto (1 - \mu r_a)^{m_2} (1 - \lambda r_b)^{m_1}.\quad (6.9)$$

Hence, $\tilde{\beta}^l \to 0$ if either $\lambda \to 1/r_b$ or $\mu \to 1/r_a$, so that in each case the prior curve segment is pulled towards $C^l$ from $A$ ($= D^l$). Similarly, $\tilde{\alpha}^r \to 0$ and the following curve segment is pulled towards $B^r$ from $A$. If $\lambda \to 0$ and $\mu \to 0$ the weights will steadily approach a constant value, following the behaviour of $K$ and $L$.

(b) Convex curve: $a + b > \pi$

We note that in this case $r_a, r_b < 0$. Hence, if $m_1, m_2 \geq 1$, then from (6.5), (6.6), as $\lambda \to \infty$, so $K, L, \alpha \to \infty$. The overall effect would be the counter intuitive one of pushing the curve away from $B$. Similar effects are observed as $\mu \to \infty$.

These undesirable effects may be overcome by setting, for this type of convex curve segment,

$$m_1, m_2 \leq -2.$$
Now, as $\lambda, \mu \to \infty$, $K, L, \alpha^r, \beta^l \to 0$, and $\alpha, \beta$ will remain bounded. For example, if $m_1 = -2$, $\alpha$ will approach a constant as $\lambda \to \infty$, whereas $\alpha \to 0$ if $m_1 < 2$.

(c) Convex curve: $a + b = \pi$

We note that in this case $r_a, r_b = 0$. It can be seen from (6.5)-(6.8) that as $\sin(a + b) \to 0$, the effects described above become more and more local, so that in the limit only one curve segment is affected and the curvature values remain unchanged. However the changes to the curve are no longer intuitive. For example, as $\lambda \to \infty$, it follows from (6.6) that irrespective of the values of $m_1, m_2; \alpha \to \infty$ and the curve will pull away from $B$. Therefore, we need to amend (6.5), (6.6) in order to restore an intuitive behaviour, while maintaining the behaviours of cases (a) and (b) above and providing continuity between all three cases.

This can be achieved by dividing the expressions for $K, \alpha$ in (6.5), (6.6) by

$$(1 + \lambda)^{m_3}(1 - |\sin(a + b)|) + |\sin(a + b)|,$$

and those for $L, \beta$ by

$$(1 + \mu)^{n_3}(1 - |\sin(a + b)|) + |\sin(a + b)|,$$

where $m_3, n_3 \geq 3$.

Note that the behaviour of $\alpha^r, \beta^l$ will still depend directly upon $K, L$ as explained above.

Now, in this case, $K, \alpha \to 0$ as $\lambda \to \infty$ and the curve will be pulled towards $B$, providing a more intuitive modification to the shape of the curve as $B$ is moved. Similar behaviour is obtained for $L, \beta, \mu$ and $C$. Also as $\lambda, \mu \to 0$ the expected behaviour will still be observed (i.e. the curve segment will tighten) while $K, L$ remain finite. Note also that the influence of these additional terms decreases as $|\sin(a+b)|$ increases. In fact, when $|\sin(a+b)| = 1$, (6.5), (6.6) are reproduced exactly.

(d) Curve with Inflection

As noted in §6.1, the issues that need to be addressed for curves with inflections are the same as for convex curve segments. The only point to note is that, when $a + b \neq 0$, the Bézier points $B, C$ will need to be adjusted in different ways; one will be constrained and the other unconstrained. Hence (6.5), (6.6) need to be generalised slightly to allow for this non-symmetric behaviour.

Thus we arrive at our final specification for the curvatures and the weights which can be applied automatically depending upon the type of curve segment:

\[
K = \frac{K_\lambda (1 - \mu r_a)^{m_1} (1 - \lambda r_b)^{m_2}}{(1 + \lambda)^{m_3}(1 - |\sin(a + b)|) + |\sin(a + b)|},
\]

\[
L = \frac{L_\lambda (1 - \mu r_a)^{n_1} (1 - \lambda r_b)^{n_2}}{(1 + \mu)^{n_3}(1 - |\sin(a + b)|) + |\sin(a + b)|},
\]

(6.10)
\[
\alpha = \frac{C\lambda^2(1-\mu r_a)^{m_1-1}(1-\lambda r_b)^{m_2}}{(1+\lambda)^{m_3}(1-|\sin(a+b)|) + |\sin(a+b)|},
\beta = \frac{C\mu^2(1-\mu r_a)^{n_1}(1-\lambda r_b)^{n_2-1}}{(1+\mu)^{n_3}(1-|\sin(a+b)|) + |\sin(a+b)|},
\]

where for convex segments

\[
\begin{align*}
&n_1 = m_2, \quad n_2 = m_1, \\
&m_1, m_2 \geq 1 \quad \text{if } a + b < \pi, \\
&m_1, m_2 \leq -2 \quad \text{if } a + b > \pi,
\end{align*}
\]

for inflection segments

\[
\begin{align*}
&m_1, m_2 \geq 1 \quad n_1, n_2 \leq -2 \quad \text{if } r_a < 0, \\
&m_1, m_2 \leq -2 \quad n_1, n_2 \geq 1 \quad \text{if } r_a > 0,
\end{align*}
\]

and in all cases \( m_3, n_3 \geq 3 \), and (6.7), (6.8) are used to modify one weight on each adjacent segment.

§7. Results

We now present a selection of results that have been obtained through experimenting with the scheme derived above. In all cases, we have kept \( m_1 = m_2 = 1 \) for convex segments in which \( a + b < \pi \), and \( m_1 = m_2 = -2 \) for convex segments in which \( a + b \geq \pi \). For inflection segments, \( m_1, m_2, n_1, n_2 \) have been set to either 1 or -2 as appropriate. We have also set \( m_3 = n_3 = 3 \). Clearly more experimentation could be carried out by varying these values, however we feel that the results obtained and described below suffice in giving an overall impression of what a user would expect from the scheme.

Figure 10 refers to the data set taken from a circle, with the initial curve shown in figure 2. It shows the changes that result when modifying the curve segment that turns through an angle \( \pi \). It can be seen that as the Bézier points are moved in either direction from the initial curve, the curve follows the control polygon as expected. The effect on each adjacent segment is relatively minor, as would be expected from (6.9), (6.10) noting that \( r_a = r_b = 0 \). The corresponding curvature plots are shown in figure 11, demonstrating that \( G^2 \) continuity has been maintained in all cases.

Figures 12–20 consider the vase data, with the initial curve shown in figure 4. Let us suppose that a designer wishes to change this initial curve into a “wine glass” shape, with a straight stem and a wide base. In figure 12 part of this has been achieved with a tightening operation. Three curve segments, indicated by 1, 2 and 3, are tightened by moving the positions of \( \mathbf{B} \) and \( \mathbf{C} \) for each segment towards the corresponding data points. A wider base can then be obtained by increasing \( k_\lambda \) and \( k_\mu \) for the convex segment at the base. For this segment \( a + b > \pi \) and the movements of \( \mathbf{B} \) and \( \mathbf{C} \)
Rational Interpolation

are unlimited. The final shape, with $k_\lambda = k_\mu = 3$ for the base segment, is shown in figure 13 with the corresponding curvature plot in figure 14. For this segment, the curvatures $K, L$ both decrease, consequently, the weights $\alpha$ and $\beta$ also decrease. On each adjacent segment one weight, $\alpha^* + \beta^*$, also decrease, the result of this being that each of these segments is pulled towards the Bézier point $\mathbf{B}^*$, $\mathbf{C}^*$ respectively. For the three segments that have been tightened, the curvature at each data point increases, but remains finite, as the tightening effect increases. As a result, one weight on each of the corresponding adjacent segments increases, resulting in a slight flattening effect on each of these segments.

Figure 15 does not consider a design task as such but demonstrates the effect of varying $k_\mu$ and $k_\lambda$ for the base segment. Keeping $k_\lambda = 3$, it shows what happens when $k_\mu$ is reduced to 0.25. Qualitatively the effect on the shape of the curve is as expected with the curve being pulled back towards the data point. It can be seen that although the curve is still pulled towards $\mathbf{B}$, the reduction in $k_\mu$ has also reduced significantly the influence of the change in $\mathbf{B}$. This may be expected from (6.10) because of the influence of $\mu$ on $K$. The curvature plot is shown in figure 16.

Now suppose that a design is required that is of a similar shape to that of figure 13, but with less rounded sides above the stem. Two attempts at this design are shown in figures 17 and 19, in which $k_\lambda$ and $k_\mu$ are both increased to 15 for convex segments 1 and 3. For each segment $a + b < \pi$, so that $k_\lambda$ and $k_\mu$ are allowed to increase indefinitely, but as described in §6, neither $\mathbf{B}$ nor $\mathbf{C}$ will ever pass the intersection point $\mathbf{E}$ (see figure 9). Clearly, the curve of figure 19 is more successful in creating the desired effect even though the values for $k_\lambda$ and $k_\mu$ are the same in both cases. This serves to demonstrate that because the scheme depends to an extent upon what is considered to be the "initial curve", care must be taken sometimes when changing the positions of Bézier points on adjacent segments. In figure 17, segments 1 and 3 are modified before segment 2. Before segment 2 is modified, segments 1 and 3 are pulled very noticeably towards the new locations of $\mathbf{B}$ and $\mathbf{C}$. This is because of the decreases in $K, L$, $\alpha$ and $\beta$ for each segment. The appearances of segments 1 and 3 change however after segment 2 is tightened, because this operation increases $K$ and $L$ for segment 2, $\beta$ for segment 1 and $\alpha$ for segment 3. This leads to a shift away from $\mathbf{B}$ and $\mathbf{C}$ on segments 1 and 3. In figure 19, segments 1 and 3 are modified after segment 2, i.e. they are modified starting from figure 13. Now the strong pulling effects towards the pairs of points $\mathbf{B}$ and $\mathbf{C}$ are clearly visible. Visually, the effects of these changes on segment 2 are minimal because the locations of its Bézier points preclude any major changes. Note however that the curvature plots (figures 18 and 20) clearly show that for the curve in figure 19 there are expected decreases in $K$ and $L$ for segment 2 compared with the curve in figure 17. To sum up, the differences in the shapes of the same curve segments with the same Bézier points is caused by the different weight values, which in turn is caused by the order in which the adjacent segments are modified. Behaviour that is consistent in both cases is the flattening effect of the curve at the start (resp.
end) of segment 1 (resp. 3), and the pull towards \( C^* \) (resp. \( B^* \)). This is expected because of the decrease in \( K \) (resp. \( L \)) for segment 1 (resp. segment 3).

It should be noted that if the changes on adjacent segments are of the same type, i.e. both tightening or both loosening operations, the order in which the segments are modified is not so critical.

The final test case considers the open curve of figure 6. A new face is required that has a more rounded nose, but with a more prominent chin. This is achieved by a mixture of increasing and decreasing values for \( k_{\lambda} \) and \( k_{\mu} \) for three inflection segments. The resulting face is shown in figure 21 with the curvature plot in figure 22. The very large curvature value is achieved near the start of segment 2.

§8. Conclusions

We have described an interactive scheme for modifying the shape of local convexity preserving interpolating curves. These curves are constructed as piecewise rational cubics and straight line segments, and have a number of desirable properties including (in general) curvature continuity. As explained at the beginning of this paper, an interactive modification scheme should have a number of desirable properties; listed as (P1)-(P5) in §1. We will now re-visit these properties and assess the scheme against them.

(P1) \( G^2 \) continuity should be preserved: this is satisfied. Once the Bézier points have been moved, appropriate curvature values and weights are adjusted, using (6.7), (6.8) and (6.10) to ensure that the curvature is continuous at each data point. Together, these ensure that (6.4) is always satisfied. Discontinuities will only occur when a straight line segment meets a cubic segment, as explained in §4.

(P2) Local convexity must be preserved: this is satisfied. Figures 1(a) and 1(e) show points \( E \) beyond which the Bézier points \( B \) and \( C \) cannot move. Use of (6.3) guarantees that this will be the case. Since, in addition, the tangent directions remain unchanged from the initial curve then local convexity will be preserved: i.e. the curve will only have those inflection points that are imposed upon it by the data.

(P3) The scheme should be intuitive. This aspect has perhaps caused the greatest problem in deriving the scheme. It is well known from standard Bézier theory how the shape of a single Bézier curve segment is affected by moving a Bézier point, or changing the value of a weight, or changing the value of the curvature at an end-point. It is the intention that this scheme should reproduce these well known effects, so that a user will have a tool available with which he/she is comfortable and familiar. We believe that this has been achieved and that this has been demonstrated in §7.

(P4) Changes should be local. It is clear from (6.7)-(6.10) that no more than three curve segments will ever be affected by moving the Bézier point(s) of
Rational Interpolation

one segment. However, a study of these same expressions shows that the “local” behaviour will depend to an extent upon the data, via the terms $r_a, r_b$ introduced in (3.4): the smaller these terms the more local the behaviour.

(P5) The scheme should be stable. Note that (6.10), (6.11) change with the data continuously for the three different versions of convex segments (figures 1(a)-1(c)), and the two different versions of inflection segments (figures 1(d)-1(e)). This ensures stability when small changes occur in the data such that the type of a curve segment (ie. either convex or inflection) does not change. However, if a small change in the data does change the type of a curve segment (from convex to inflection, or vice versa), there will be an instantaneous and significant change in the way in which one of the Bézier points is allowed to move. It will change from being constrained to unconstrained, or vice versa. This action will be reflected in (6.10), (6.11), when either $m_1$ and $m_2$, or $n_1$ and $n_2$, change instantaneously from $\geq 1$ to $\leq -2$, or vice versa. Hence provided a small change in the data does not lead to curve segments changing their type, there will be a correspondingly small change in the way that the modification scheme can affect the shape of the curve. Hence the scheme is stable.

In summary, we believe that a scheme has been developed that achieves the goals set out in §1. Hence, curves generated by the method of [4] may be modified interactively while retaining their desirable properties. Note also that the modification scheme described in §§6, 7 may be applied to any rational cubic Bézier interpolating curve that requires $G^2$ continuity to be maintained and that is shape preserving in the sense described in this paper.

The results that have been shown in this paper have concentrated upon demonstrating the effects that users can expect from the scheme after Bézier points have been moved. These have, in general, been done with particular design tasks in mind. Another approach to the way that the scheme could be used is to concentrate upon “fairing” the curve to smooth out an initial curvature distribution. It is noted from (6.5) that the curvatures at the data points are always bounded, however a study has yet to be carried out on the effects that the modification scheme has upon the curvature away from the data points. This needs to be done before this “fairing” approach could be tried.

In [11], Kaklis and Karavelas succeed in extending the approach described in [12] to space curves. Splines of non-uniform degree are used to construct interpolants in $\mathbb{R}^3$ that are curvature and torsion continuous. They are also shape preserving in so far as the behaviour of the curve conforms to the discrete properties of the corresponding linear interpolant with regard to convexity, the sign of the torsion, coplanarity and collinearity. As with the two-dimensional case in [12], an increase in the degree of a curve segment is analogous to applying tension to the curve segment, so that it approaches the corresponding linear interpolant. Reference [11] is the first study to appear in the research literature that describes a method for generating shape preserving space curves.
Since then, Goodman and Ong[5, 6] have also considered this problem. In [5], the rational cubic scheme of [4] is extended to interpolate points in space. The end result is a scheme that preserves the convexity, inflections, signs of the torsion, coplanarity and collinearity of the data. The curves have continuous unit tangents, and although in general the curvature is continuous in magnitude, their osculating planes do not vary continuously across the data points. This drawback is addressed in [6], in which a $G^2$ piecewise rational cubic shape preserving scheme is described, with the osculating plane varying continuously along the curve.

It would be of interest to consider interactive modification of the curves generated by the method of [6], using techniques similar to those presented in this paper. Again, adjustment of the Bézier points could be considered while retaining the shape preserving properties of the initial curve. This remains a topic for future study.

§Acknowledgments

The authors wish to acknowledge the many helpful discussions with Tim Goodman while this work was in progress. They also wish to acknowledge the comments of the referees, which have helped to improve the presentation of the study.

References

Rational Interpolation


Chris I. Seymour and Keith Unsworth,
Applied Computing, Mathematics and Statistics Group,
Applied Management and Computing Division,
P.O. Box 84, Lincoln University, Canterbury, New Zealand.

email: unsworth@lincoln.ac.nz
Figure 1 The five types of Bézier polygons: (a) convex, $|a + b| < \pi$,  
(b) convex, $|a + b| = \pi$ (c) convex, $|a + b| > \pi$,  
(d) inflection, $|a| = |b|$ (e) inflection, $|a| > |b|$.

Figure 2 Circle data. Closed initial curve.  
O=data points, +=Bézier points.  
S = start point for curvature plot, with data points traversed in anti-clockwise order.

Figure 3 Curvature plot for data of figure 2.  
O=data points.

Figure 4 Vase data. Closed initial curve.  
O=data points, +=Bézier points.  
S = start point for curvature plot, with data points traversed in anti-clockwise order.

Figure 5 Curvature plot for data of figure 4.  
O=data points.

Figure 6 Face data. Open initial curve.  
O=data points, +=Bézier points.  
S = start point for curvature plot.

Figure 7 Curvature plot for data of figure 6.  
O=data points.

Figure 8 Graph of $\lambda(k_\alpha)$.

Figure 9 More detailed version of figure 1(a).

Figure 10 Circle data showing initial curve and three modified curves.  
Curve 1: $k_\alpha = k_\mu = 1$. Curve 2: $k_\alpha = k_\mu = 1.75$.  
Curve 3: $k_\alpha = k_\mu = 1.4$. Curve 4: $k_\alpha = k_\mu = 0.5$.  
O=data points, +=Bézier points.

Figure 11 Curvatures plots for data of figure 10.  
O=data points.

Figure 12 Vase data with three segments modified.  
Segments 1 and 3: $k_\alpha = k_\mu = 0.1$. Segment 2: $k_\alpha = k_\mu = 0.01$.  
O=data points, +=Bézier points.

Figure 13 Final "wine glass" shape.  
$k_\alpha = k_\mu = 3$ for base segment. Remaining values as for figure 12.  
O=data points, +=Bézier points.
Figure 14 Curvature plot for data of figure 13.
O=data points.

Figure 15 Vase data with one segment modified.
$k_{\lambda} = 3$, $k_{\mu} = 0.25$. O=data points, +=Bézier points.

Figure 16 Curvature plot for vase data of figure 15.
O=data points.

Figure 17 Vase data with "straight" edges.
Segments 1 and 3: $k_{\lambda} = k_{\mu} = 15$. Remaining values as for figure 13.
O=data points, +=Bézier points.

Figure 18 Curvature plot for vase data of figure 17.
O=data points.

Figure 19 Vase data with "straight" edges.
Segments 1 and 3: $k_{\lambda} = k_{\mu} = 15$. Remaining values as for figure 13.
O=data points, +=Bézier points.

Figure 20 Curvature plot for vase data of figure 19.
O=data points.

Figure 21 Face data with three segments modified.
Segment 1: $k_{\lambda} = 3$, $k_{\mu} = 1$. Segment 2: $k_{\lambda} = 0.1$ $k_{\mu} = 3$.
Segment 3: $k_{\lambda} = k_{\mu} = 2$. O=data points, +=Bézier points.

Figure 22 Curvature plot for face data of figure 19.
O=data points.
Figure 1
Figure 3
Figure 5
Figure 7
Figure 16
Figure 22