Geometric Continuity, Shape Parameters, and Geometric Constructions for Catmull–Rom Splines

TONY D. DEROSE
University of Washington
and
BRIAN A. BARSKY
University of California

Catmull–Rom splines have local control, can be either approximating or interpolating, and are efficiently computable. Experience with Beta-splines has shown that it is useful to endow a spline with *shape parameters*, used to modify the shape of the curve or surface independently of the defining control vertices. Thus it is desirable to construct a subclass of the Catmull–Rom splines that has shape parameters.

We present such a class, some members of which are interpolating and others approximating. As was done for the Beta-spline, shape parameters are introduced by requiring *geometric* rather than *parametric* continuity. Splines in this class are defined by a set of control vertices and a set of shape parameter values. The shape parameters may be applied globally, affecting the entire curve, or they may be modified locally, affecting only a portion of the curve near the corresponding joint. We show that this class results from combining geometrically continuous (Beta-spline) blending functions with a new set of geometrically continuous interpolating functions related to the classical *Lagrange curves*.

We demonstrate the practicality of several members of the class by developing efficient computational algorithms. These algorithms are based on geometric constructions that take as input a control polygon and a set of shape parameter values and produce as output a sequence of Bézier control polygons that exactly describes the original curve. A specific example of shape design using a low-degree member of the class is given.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*curve, surface, solid, and object representations*; J.6 [**Computer Applications**]: Computer-Aided Engineering, Computer-Aided Design

General Terms: Algorithms, Design

Additional Key Words and Phrases: Approximation, Beta-splines, Bézier curves, Catmull–Rom splines, computer-aided geometric design, curves and surfaces, geometric continuity, interpolation, shape parameters

This work was supported in part by the Defense Advanced Research Projects Agency under contracts N00039-82-C-235 and N00039-84-C-0089, the National Science Foundation under grants ECS-8204381, CCR-8451997, and DMC-8602141, Control Data Corporation, and AT&T Bell Laboratories.
Authors' addresses: T. D. DeRose, Department of Computer Science, FR-35, University of Washington, Seattle, WA 98195; B. A. Barsky, Berkeley Computer Graphics Laboratory, Computer Science Division, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720.

1. INTRODUCTION

Many applications of computer-aided geometric design require the description of objects using mathematical functions called *splines*. A spline curve is a piecewise univariate function that satisfies a set of *continuity constraints* where the curve segments meet. A popular type of spline is the *parametric spline*, typically defined by a set of vector-valued *control vertices* and a set of polynomial *blending functions* used to weight the vertices [7].

A spline can be categoried as being *interpolating*, meaning that it is required to pass through its control vertices, or *approximating*, meaning that it is only required to pass "near" its control vertices. A spline can be further classified as being either a *global* or a *local* representation. In a global representation, the movement of a control vertex causes the entire spline to change. In a local representation it is possible to localize the change resulting from the perturbation of a control vertex, a property known as *local control*. The recent development of the *Beta-spline* [1, 3, 4, 6, 9] has shown that it is possible to extend the curve formulation by introducing *shape parameters*, which can be used to modify the shape of the curve independently of the control vertices. Experience has shown that shape parameters can provide a designer with intuitive control of shape.

From the standpoint of computer-aided geometric design, it is desirable to construct local splines with shape parameters. Since the choice of interpolation versus approximation is application dependent, both should be possible. The objective of this work is to develop a class of splines possessing local control that are either interpolating or approximating and have locally variable shape parameters. Catmull and Rom [11] introduced a class of local splines that could be made to either interpolate or approximate a set of control vertices.[1] To construct a class of splines with the enumerated properties, we need only to introduce shape parameters into the Catmull–Rom splines. As with Beta-splines, this is done by replacing *parametric* continuity with the less restrictive measure of *geometric* continuity [3–5, 13, 14].

We show how the relaxation to geometric continuity can yield a class of Catmull–Rom splines, either interpolating or approximating, whose shape can be modified via shape parameters. The interpolating splines that we present are particularly interesting owing to their shape parameters—they are local, polynomial, interpolating splines with locally variable shape parameters. Consequently, local modification of a shape parameter affects only a portion of the curve near the corresponding *joint* (a point where two curve segments abut).

The two interpolating members of lowest degree are studied further by developing efficient evaluation algorithms and by empirically investigating the behavior of the curves when shape parameters are varied. The evaluation algorithm is based on the construction of equivalent Bézier control polygons, one for each segment of the Catmull–Rom curve. Once the Bézier control polygons have been constructed, each segment can be evaluated using standard algorithms for Bézier curves, such as *recursive subdivision* [21, 22] or de Casteljau's algorithm [10].

[1] Unfortunately, the title of their paper did not reflect the fact that both approximating and interpolating splines are members of the class.

The presentation proceeds as follows: The class of Catmull–Rom splines is briefly reviewed in Section 2. In Section 3, the notion of geometric continuity is presented, and in Section 3.1 it is applied to the problem of constructing smooth piecewise Bézier curves. In Section 4, the G^1 and G^2 Beta-splines are derived using the results of Section 3.1. In Section 5, the class of geometrically continuous Catmull–Rom splines is introduced and properties of members of the class are identified. In Section 6, a practical general algorithm for the evaluation and rendering of geometrically continuous Catmull–Rom splines is developed. Finally, in Section 7, two of the interpolating members of low degree are studied and their evaluation algorithms are presented.

Of particular interest is the quintic interpolating spline discussed in Section 7.2. We believe that this spline may find application in problem domains where second-order smooth interpolation is required; so, to aid the implementor, we have included a detailed pseudocoded version of the evaluation algorithm.

1.1 Notation

Scalar quantities are written in italics as in x and $Y(u)$, and vectors and vector-valued functions are denoted by boldface type as in $\mathbf{V}$ and $\mathbf{Q}(u)$. Since it is necessary to distinguish between a piecewise function and the segments that compose it, we adhere to the convention that a piecewise function is denoted by an uppercase character, as in $\mathbf{H}_q(u)$, while its segments are indexed and written in the corresponding lowercase, as in $\mathbf{h}_{q,j}(u)$. Finally, the pth derivative of a function $W(u)$ from the left is written as $W^{(p)}(u^-)$, while the pth derivative from the right is written as $W^{(p)}(u^+)$; when no confusion can result, we simply write $W^{(p)}(u)$.

2. THE CLASS OF CATMULL–ROM SPLINES

Splines used in computer-aided geometric design are typically defined by a set of *control vertices* $\mathbf{V}_0, \ldots, \mathbf{V}_m$ and a set of blending functions $W_0(u), \ldots, W_m(u)$; that is,

$$\mathbf{Q}(u) = \sum_{i=0}^{m} \mathbf{V}_i W_i(u). \tag{2.1}$$

Catmull and Rom extended this form by replacing the vertices $\mathbf{V}_i$ with vector-valued *interpolating functions* $\mathbf{P}_i(u)$. Each $\mathbf{P}_i(u)$ is constructed to interpolate the $k + 1$ vertices $\mathbf{V}_i, \mathbf{V}_{i+1}, \ldots, \mathbf{V}_{i+k}$, for some nonnegative integer k. Intuitively, k sets the width of the *interpolating window* of the function $\mathbf{P}_i(u)$. Thus a Catmull–Rom spline takes the form

$$\mathbf{F}(u) = \sum_{i} \mathbf{P}_i(u) W_i(u). \tag{2.2}$$

Since each of the interpolating functions $\mathbf{P}_i(u)$ itself defines a curve, eq. (2.2) states that a Catmull–Rom spline $\mathbf{F}(u)$ is created by blending together curves rather than control vertices. For instance, if $k = 1$, the interpolating function $\mathbf{P}_i(u)$ is required to pass through the vertices $\mathbf{V}_i$ and $\mathbf{V}_{i+1}$. The simplest such function is a parameterized line connecting $\mathbf{V}_i$ and $\mathbf{V}_{i+1}$:

$$\mathbf{P}_i(u) = (1 - u)\mathbf{V}_i + u\mathbf{V}_{i+1}. \tag{2.3}$$

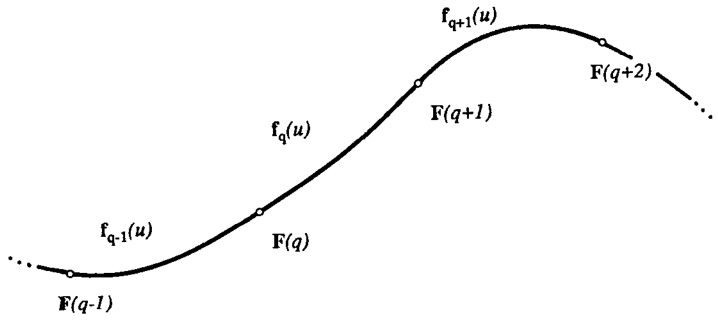

Fig. 1. Indexing of the piecewise function $\mathbf{F}(u)$. Note that the joints correspond to integral values of the domain parameter.

By blending these lines together with a set of blending functions $W_i(u)$, we obtain a Catmull–Rom spline with $k = 1$. An even simpler situation occurs when $k = 0$ since the function $\mathbf{P}_i(u)$ is only required to interpolate the single vertex $\mathbf{V}_i$. It is therefore sufficient for $\mathbf{P}_i(u)$ to be a constant function independent of u; that is, $\mathbf{P}_i(u) = \mathbf{V}_i$. In this case, eq. (2.2) is identical in form to eq. (2.1), showing that the Catmull–Rom splines generalize standard approximating techniques such as Bézier curves [7, 8, 10], B-splines [7, 10], and Beta-splines [1, 3, 7]. More generally, Catmull and Rom show that if the blending functions are nonzero over D parametric intervals, then a spline of the form given in eq. (2.2) will be approximating if $k < D - 2$, and interpolating otherwise (this result follows directly from eqs. (2.4)–(2.6)).

Throughout the remainder of our discussion, we make the following assumptions:

—The qth segment of $\mathbf{F}(u)$, denoted $\mathbf{f}_q(u)$, is traced out when u is on the half-open interval $[q, q + 1)$ (see Figure 1); thus $\mathbf{F}(u)$ has uniformly spaced *parametric breakpoints.*

—The blending functions have *local support*; that is, they are nonzero only over D parametric intervals. The ith such function $W_i(u)$ is nonzero only over the open interval $(i - 1, i - 1 + D)$.

—The blending functions form a *partition of unity*; that is, they satisfy

$$\sum_{i=0}^{m} W_i(u) = 1 \qquad \text{for all} \quad u \in [0, m). \tag{2.4}$$

This property is necessary if the blending functions are to describe a curve whose shape is independent of the coordinate system in which the control vertices are represented (cf. Bartels et al. [7]).

—The interpolating functions $\mathbf{P}_i(u)$ are constructed so that points of interpolation correspond to integral values of the domain parameter:

$$\mathbf{P}_i(q) = \mathbf{V}_q \qquad \text{for} \quad q = i, i + 1, \ldots, i + k. \tag{2.5}$$

With the above assumptions, the qth segment of $\mathbf{F}(u)$ can be written as

$$\mathbf{f}_q(u) = \sum_{i=2-D}^{1} \mathbf{P}_{q+i}(u) W_{q+i}(u), \qquad u \in [q, q + 1). \tag{2.6}$$

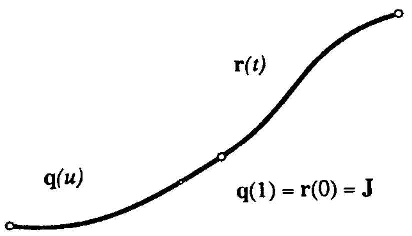

Fig. 2. Two curve segments $\mathbf{q}(u)$ and $\mathbf{r}(t)$ meeting at a joint $\mathbf{J}$.

3. GEOMETRIC CONTINUITY

Since splines are defined as piecewise functions, care must be taken to smoothly "stitch" the segments together where they abut.

The issue of exactly what is meant by "smooth" is a subtle one, ultimately leading to the distinction between parametric and geometric continuity. We present here an abbreviated development of geometric continuity; more complete treatments can be found in [5], [13], and [14].

Consider the situation shown in Figure 2 where two C^∞ parameterizations $\mathbf{q}(u)$, $u \in [0, 1]$, and $\mathbf{r}(t)$, $t \in [0, 1]$ (a parameterization is said to be C^∞ if it is infinitely differentiable) meet at a common point $\mathbf{J}$ such that

$$\mathbf{r}(0) = \mathbf{q}(1) = \mathbf{J}.$$

These parameterizations are said to meet with nth-*order parametric continuity*, denoted C^n, if the first n parametric derivatives match at $\mathbf{J}$, that is, if

$$\mathbf{r}^{(i)}(0) = \mathbf{q}^{(i)}(1), \qquad i = 1, \ldots, n.$$

Unfortunately, parametric continuity does not capture our intuitive notion of smoothness, as demonstrated by the next example.

Example 3.1. Consider the two parameterizations plotted in Figure 3:

$$\begin{aligned} \mathbf{q}(u) &= (2u, u), & u &\in [0, 1], \\ \mathbf{r}(t) &= (4t + 2, 2t + 1), & t &\in [0, 1]. \end{aligned} \tag{3.1}$$

These parameterizations meet with positional continuity at the joint $\mathbf{J} = (2, 1)$. Note, however, that their first derivative vectors do not match at the joint:

$$\left.\begin{aligned} \mathbf{q}^{(1)}(1) &= (2, 1) \\ \mathbf{r}^{(1)}(0) &= (4, 2) \end{aligned}\right\} \quad \text{implying that} \quad \mathbf{q}^{(1)}(1) \neq \mathbf{r}^{(1)}(0).$$

Thus these parameterizations do not meet with first-order parametric continuity, even though the plotted curve segments appear to meet very smoothly. □

To understand how to avoid situations like the one in Example 3.1, it is necessary to introduce the concept of *equivalent parameterizations*. Let $\mathbf{q}(u)$, $u \in [a, b]$, and $\tilde{\mathbf{q}}(\tilde{u})$, $\tilde{u} \in [\tilde{a}, \tilde{b}]$, be two *regular* C^∞ parameterizations (a parameterization is regular if its first derivative vector never vanishes). These parameterizations are said to be *equivalent*, that is, they describe the same

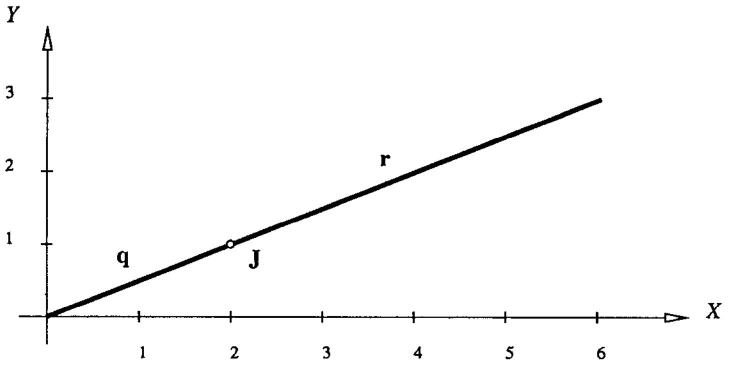

Fig. 3. Line segments meeting with G^1 but not C^1 continuity.

oriented curve, if there exists a C^∞ function f: $[\tilde{a}, \tilde{b}] \mapsto [a, b]$ such that

(i) $\tilde{\mathbf{q}}(\tilde{u}) = \mathbf{q}(f(\tilde{u}))$,
(ii) $f(\tilde{a}) = a$,
(iii) $f(\tilde{b}) = b$,
(iv) $f^{(1)} > 0$.

Intuitively, $\mathbf{q}$ and $\tilde{\mathbf{q}}$ trace out the same set of points in the same order. We also say that $\mathbf{q}$ has been reparameterized to obtain $\tilde{\mathbf{q}}$, and we call f an *orientation-preserving change of variables* (see Figure 4).

Example 3.2. As a concrete example of equivalent parameterizations, let $\mathbf{q}$ be as in Figure 3, and let $\tilde{\mathbf{q}}$ be defined by

$$\tilde{\mathbf{q}}(\tilde{u}) = (4\tilde{u}, 2\tilde{u}), \qquad \tilde{u} \in [0, \tfrac{1}{2}].$$

To show that $\mathbf{q}(u) = (2u, u)$ and $\tilde{\mathbf{q}}(\tilde{u}) = (4\tilde{u}, 2\tilde{u})$ are equivalent parameterizations, we observe that

$$\tilde{\mathbf{q}}(\tilde{u}) = \mathbf{q}(2\tilde{u}) \qquad \text{for all} \quad \tilde{u} \in [0, \tfrac{1}{2}].$$

Thus we have found a mapping f: $[0, \frac{1}{2}] \mapsto [0, 1]$ defined by $f(\tilde{u}) = 2\tilde{u}$ that satisfies property (i) above. It is easily verified that f satisfies the other three properties as well. We therefore conclude that $\mathbf{q}$ and $\tilde{\mathbf{q}}$ describe the same oriented curve, which in this case is the oriented line segment from (0, 0) to (2, 1). □

The key to geometric continuity is the following observation: Since reparameterization does not affect the shape of the curve being described, we should be free to reparameterize before determining continuity between two parameterizations such as $\mathbf{q}$ and $\mathbf{r}$ in Example 3.1. We are therefore led to the following definition.

Definition 3.1. Let $\mathbf{q}$ and $\mathbf{r}$ be two C^∞ parameterizations meeting at a point $\mathbf{J}$. They meet with nth-order geometric continuity, denoted G^n, if there exists a parameterization $\tilde{\mathbf{q}}$ equivalent to $\mathbf{q}$ such that $\tilde{\mathbf{q}}$ and $\mathbf{r}$ meet with C^n continuity at $\mathbf{J}$.

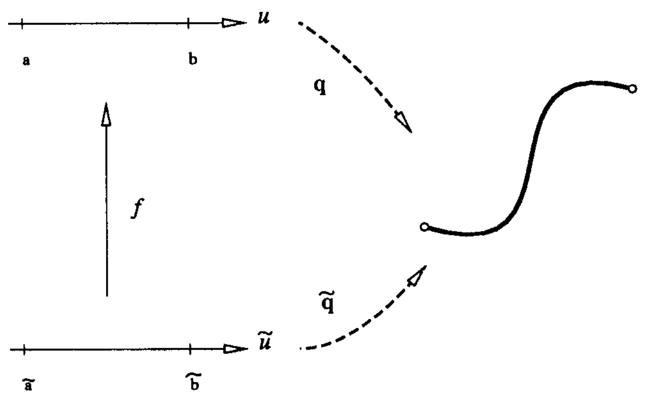

Fig. 4. $\mathbf{q}(u)$ is reparameterized by f to obtain $\tilde{\mathbf{q}}(\tilde{u})$.

Let us apply this definition of geometric continuity to the parameterizations of Example 3.1. In particular, if we choose $\tilde{\mathbf{q}}$ to be the equivalent parameterization constructed in Example 3.2, then we see that

$$\begin{aligned}\tilde{\mathbf{q}}^{(1)}(\tfrac{1}{2}) &= (4, 2),\\ \mathbf{r}^{(1)}(0) &= (4, 2),\end{aligned} \qquad \text{implying that} \quad \tilde{\mathbf{q}}^{(1)}(\tfrac{1}{2}) = r^{(1)}(0).$$

Thus $\tilde{\mathbf{q}}$ and $\mathbf{r}$ meet with C^1 continuity at $\mathbf{J} = (2, 1)$; hence $\mathbf{q}$ and $\mathbf{r}$ meet with G^1 continuity.

The characterization of geometric continuity based on the existence of equivalent parameterizations is a useful theoretical tool, which is used in Section 5.1. However, there are other characterizations that are more appropriate for applications. We now present one such characterization (for a more complete treatment see [5], [13], and [14]).

Let $\mathbf{q}(u)$, $u \in [0, 1]$ and $\mathbf{r}(t)$, $t \in [0, 1]$ be two regular C^∞ parameterizations meeting with G^n continuity at $\mathbf{q}(1) = \mathbf{r}(0)$, as shown in Figure 2. According to Definition 3.1, there must exist an orientation-preserving change of variables f: $[\tilde{a}, 1] \mapsto [0, 1]$ such that

$$\mathbf{r}^{(i)}(0) = \tilde{\mathbf{q}}^{(i)}(1), \qquad i = 1, \ldots, n, \tag{3.2}$$

where

$$\tilde{\mathbf{q}}(\tilde{u}) = \mathbf{q}(f(\tilde{u})), \qquad \tilde{u} \in [\tilde{a}, 1].$$

For simplicity (and without loss of generality) we have chosen $\tilde{b} = 1$. Using the chain rule from calculus, derivatives of $\tilde{\mathbf{q}}$ can be expanded in terms of derivatives of $\mathbf{q}$ and f. If the chain rule is applied i times, $\tilde{\mathbf{q}}^{(i)}$ can be expressed as a function, call it CR_i, of the first i derivatives of $\mathbf{q}$ and the first i derivatives of f:

$$\tilde{\mathbf{q}}(\tilde{u}) = \mathrm{CR}_i(\mathbf{q}^{(1)}(f(\tilde{u})), \ldots, \mathbf{q}^{(i)}(f(\tilde{u})), f^{(1)}(\tilde{u}), \ldots, f^{(i)}(\tilde{u})). \tag{3.3}$$

Evaluating this expression at $\tilde{u} = 1$ and using the fact that $f(1) = 1$, we find that

$$\begin{aligned}\tilde{\mathbf{q}}^{(i)}(1) &= \mathrm{CR}_i(\mathbf{q}^{(1)}(1), \ldots, \mathbf{q}^{(i)}(1), f^{(1)}(1), \ldots, f^{(i)}(1))\\ &= \mathrm{CR}_i(\mathbf{q}^{(1)}(1), \ldots, \mathbf{q}^{(i)}(1), \beta 1, \ldots, \beta i),\end{aligned} \tag{3.4}$$

where the substitutions

$$\beta j = f^{(j)}(1), \qquad j = 1, \ldots, i,$$

have been made. The quantities $\beta 1, \ldots, \beta i$ are real numbers, and since $f^{(1)}(1) > 0$ (property (iv) of an orientation-preserving change of variables), we can conclude that $\beta 1 > 0$. Substituting eq. (3.4) into eq. (3.2) yields the so-called *Beta-constraints*:

$$\mathbf{r}^{(i)}(0) = \mathrm{CR}_i(\mathbf{q}^{(1)}(1), \ldots, \mathbf{q}^{(i)}(1), \beta 1, \ldots, \beta i), \qquad i = 1, \ldots, n. \tag{3.5}$$

This argument shows that if $\mathbf{q}(u)$ and $\mathbf{r}(t)$ meet with G^n continuity, then there exist real parameters $\beta 1, \ldots, \beta n$, with $\beta 1 > 0$, commonly called *shape parameters*, satisfying the Beta-constraints. More important for applications, the converse is also true. That is,

THEOREM 3.1. $\mathbf{q}(u)$, $u \in [0, 1]$, *and* $\mathbf{r}(t)$, $t \in [0, 1]$, *meet with* G^n *continuity at* $\mathbf{r}(0) = \mathbf{q}(1)$ *if and only if there exist real numbers* $\beta 1, \ldots, \beta n$ *with* $\beta 1 > 0$ *such that eqs.* (3.5) *are satisfied.*

PROOF. For a rigorous proof see [13]. □

As an example of the form of the Beta-constraints, the constraints for G^3 continuity are

$$\begin{aligned}
\mathbf{r}^{(1)}(0) &= \beta 1\, \mathbf{q}^{(1)}(1),\\
\mathbf{r}^{(2)}(0) &= \beta 1^2 \mathbf{q}^{(2)}(1) + \beta 2 \mathbf{q}^{(1)}(1),\\
\mathbf{r}^{(3)}(0) &= \beta 1^3 \mathbf{q}^{(3)}(1) + 3\beta 1 \beta 2 \mathbf{q}^{(2)}(1) + \beta 3 \mathbf{q}^{(1)}(1),
\end{aligned} \tag{3.6}$$

where $\beta 2$ and $\beta 3$ are arbitrary, but $\beta 1$ is constrained to be positive.

For many practical applications only G^2 continuity is required and is equivalent to the satisfaction of the first two equations of (3.6). A more geometric statement of G^2 continuity is: $\mathbf{q}(u)$ and $\mathbf{r}(t)$ meet with G^2 continuity if and only if they have common unit tangent and curvature vectors [1, 3, 4, 13].

3.1 Piecewise Bézier Curves

In preparation for later sections, and to gain a feeling for the use of the Beta-constraints, we consider the problem of stitching Bézier curves together with G^1 and G^2 continuity. We first recall several important facts concerning Bézier curves [7, 10].

A Bézier curve $\mathbf{q}(u)$, $u \in [0, 1]$, of degree d is defined by a control polygon $\mathbf{V}_0, \ldots, \mathbf{V}_d$, also called a *Bézier polygon*, and takes the form

$$\mathbf{q}(u) = \sum_{i=0}^{d} \mathbf{V}_i B_i^d(u), \qquad u \in [0, 1],$$

where

$$B_i^d(u) = \binom{d}{i} u^i (1-u)^{d-i}$$

is the ith Bernstein polynomial of degree d. Describing the curve $\mathbf{q}(u)$ in Bézier form has many advantages, the foremost of which for our purposes is the simplicity with which derivatives at $u = 0$ and $u = 1$ can be expressed. Specifically, we use the following properties.

1. *Position*: $\mathbf{q}(u)$ interpolates $\mathbf{V}_0$ at $u = 0$, and $\mathbf{V}_d$ at $u = 1$:

$$\begin{aligned} \mathbf{q}(0) &= \mathbf{V}_0, \\ \mathbf{q}(1) &= \mathbf{V}_d. \end{aligned} \tag{3.7}$$

2. *First Derivatives*: The initial tangent vector is in the direction of the vector from $\mathbf{V}_0$ to $\mathbf{V}_1$, and the final tangent vector is in the direction of the vector from $\mathbf{V}_{d-1}$ to $\mathbf{V}_d$. More precisely, the initial and final first derivative vectors are

$$\begin{aligned} \mathbf{q}^{(1)}(0) &= d(\mathbf{V}_1 - \mathbf{V}_0), \\ \mathbf{q}^{(1)}(1) &= d(\mathbf{V}_d - \mathbf{V}_{d-1}). \end{aligned} \tag{3.8}$$

3. *Second Derivatives*: The initial second derivative vector depends only on $\mathbf{V}_0$, $\mathbf{V}_1$, and $\mathbf{V}_2$, and the final second derivative vector depends only on $\mathbf{V}_{d-2}$, $\mathbf{V}_{d-1}$, and $\mathbf{V}_d$; specifically,

$$\begin{aligned} \mathbf{q}^{(2)}(0) &= d(d-1)(\mathbf{V}_0 - 2\mathbf{V}_1 + \mathbf{V}_2), \\ \mathbf{q}^{(2)}(1) &= d(d-1)(\mathbf{V}_{d-2} - 2\mathbf{V}_{d-1} + \mathbf{V}_d). \end{aligned} \tag{3.9}$$

The specific problem we wish to address here is the following:

Given: The shape parameters $\beta 1$ and $\beta 2$, and the Bézier polygon $\mathbf{V}_0, \ldots, \mathbf{V}_d$ defining the parameterization

$$\mathbf{q}(u) = \sum_{i=0}^{d} \mathbf{V}_i B_i^d(u), \qquad u \in [0, 1].$$

Find: Constraints on the Bézier polygon $\mathbf{W}_0, \ldots, \mathbf{W}_d$ defining the parameterization

$$\mathbf{r}(t) = \sum_{j=0}^{d} \mathbf{W}_j B_j^d(t), \qquad t \in [0, 1].$$

Such that: $\mathbf{q}$ and $\mathbf{r}$ meet with G^1 (or G^2) continuity at $\mathbf{q}(1)$ with respect to $\beta 1$ (and $\beta 2$) (see Figure 5).

Since a Bézier curve interpolates its first and last control vertices, we can guarantee C^0 (and G^0) continuity by setting $\mathbf{W}_0 = \mathbf{V}_d$, as shown in Figure 5. To achieve G^1 continuity for a given $\beta 1 > 0$, we can find $\mathbf{W}_1$ by recalling the first equation of (3.6),

$$\mathbf{r}^{(1)}(0) = \beta 1 \mathbf{q}^{(1)}(1), \qquad \beta 1 > 0,$$

and using eq. (3.8) to yield

$$d(\mathbf{W}_1 - \mathbf{W}_0) = d\beta 1(\mathbf{V}_d - \mathbf{V}_{d-1}), \qquad \beta 1 > 0.$$

Simplification and rearrangement yield

$$\mathbf{W}_1 = \mathbf{W}_0 + \beta 1(\mathbf{V}_d - \mathbf{V}_{d-1}), \qquad \beta 1 > 0,$$

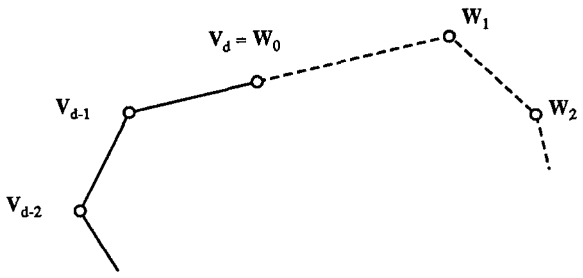

Fig. 5. The situation for stitching Bézier curves together.

and since $\mathbf{W}_0 = \mathbf{V}_d$,

$$\mathbf{W}_1 = \mathbf{V}_d + \beta 1(\mathbf{V}_d - \mathbf{V}_{d-1}), \qquad \beta 1 > 0. \tag{3.10}$$

Geometrically, eq. (3.10) states that $\mathbf{W}_1$ must lie on the half-infinite line starting at $\mathbf{V}_d$ ($= \mathbf{W}_0$), extending in the direction of the vector from $\mathbf{V}_{d-1}$ to $\mathbf{V}_d$. The length of the segment $\mathbf{W}_0\mathbf{W}_1$ relative to the length of $\mathbf{V}_{d-1}\mathbf{V}_d$ is given by the parameter $\beta 1$. Given $\mathbf{V}_{d-1}$, $\mathbf{V}_d$, and $\beta 1 > 0$, then the control vertices $\mathbf{W}_0$ and $\mathbf{W}_1$ can be determined geometrically, as shown in Figure 6, or algorithmically using the following construction:

Construction 1: *Joining Bézier curves with G^1 continuity.*

```
1. $\mathbf{W}_0 \leftarrow \mathbf{V}_d$,
2. $\mathbf{W}_1 \leftarrow \mathbf{W}_0 + \beta 1(\mathbf{V}_d - \mathbf{V}_{d-1})$.
```

Once $\mathbf{W}_0$ and $\mathbf{W}_1$ have been constrained subject to G^1 continuity, the control vertex $\mathbf{W}_2$ can be constrained to guarantee G^2 continuity for a given $\beta 2$ by recalling the second equation of (3.6),

$$\mathbf{r}^{(2)}(0) = \beta 1^2 \mathbf{q}^{(2)}(1) + \beta 2 \mathbf{q}^{(1)}(1),$$

and using eqs. (3.8) and (3.9):

$$\begin{aligned} & d(d-1)(\mathbf{W}_0 - 2\mathbf{W}_1 + \mathbf{W}_2) \\ & \quad = \beta 1^2 d(d-1)(\mathbf{V}_{d-2} - 2\mathbf{V}_{d-1} + \mathbf{V}_d) + \beta 2\, d(\mathbf{V}_d - \mathbf{V}_{d-1}). \end{aligned}$$

Simplification and rearrangement yield

$$\begin{aligned} \mathbf{W}_2 &= 2\mathbf{W}_1 - \mathbf{W}_0 + \beta 1^2(\mathbf{V}_{d-2} - 2\mathbf{V}_{d-1} + \mathbf{V}_d) + \frac{\beta 2(\mathbf{V}_d - \mathbf{V}_{d-1})}{d-1} \\ &= \beta 1^2 \mathbf{V}_{d-2} - \left(2\beta 1^2 + 2\beta 1 + \frac{\beta 2}{d-1}\right)\mathbf{V}_{d-1} \\ &\quad + \left(\beta 1^2 + 2\beta 1 + \frac{\beta 2}{d-1} + 1\right)\mathbf{V}_d. \end{aligned} \tag{3.11}$$

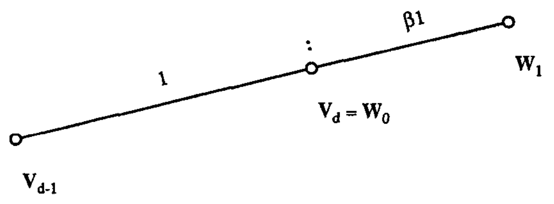

Fig. 6. Geometric interpretation of Construction 1.

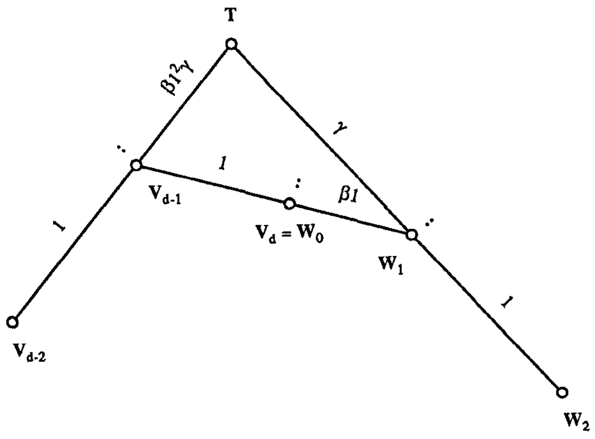

Fig. 7. The Farin–Boehm construction.

Rather than the algebraic approach given above for the determination of $\mathbf{W}_2$, Farin [16]—with a later improvement by Boehm [9]—developed a more geometric approach. For our purposes it is most convenient to think of the approach of Farin and Boehm as a convenient factorization of eq. (3.11), each term of which has a well-defined geometric interpretation. The Farin–Boehm construction takes as input the control polygon $\mathbf{V}_0, \ldots, \mathbf{V}_d$ and the shape parameters $\beta 1 > 0$ and $\beta 2$, and then produces as output the control vertices $\mathbf{W}_0$, $\mathbf{W}_1$, and $\mathbf{W}_2$ such that the curves meet with G^2 continuity with respect to $\beta 1$ and $\beta 2$. The construction may be stated as:

Construction 2: *The Farin–Boehm construction.*

```
1. $\gamma \leftarrow \dfrac{(d-1)(1+\beta 1)}{\beta 2 + \beta 1(d-1)(1+\beta 1)}$,
2. $\mathbf{W}_0 \leftarrow \mathbf{V}_d$,
3. $\mathbf{W}_1 \leftarrow \mathbf{W}_0 + \beta 1(\mathbf{V}_d - \mathbf{V}_{d-1})$,
4. $\mathbf{T} \leftarrow \mathbf{V}_{d-1} + \beta 1^2\gamma(\mathbf{V}_{d-1} - \mathbf{V}_{d-2})$,
5. $\mathbf{W}_2 \leftarrow \mathbf{W}_1 + \dfrac{1}{\gamma}(\mathbf{W}_1 - \mathbf{T})$.
```

The geometric interpretation of this construction is shown in Figure 7.

---

4. BETA-SPLINES

Given a control polygon $\mathbf{V}_0, \ldots, \mathbf{V}_m$ and a set of shape parameter values $\overline{\beta 1} = (\beta 1_0, \ldots, \beta 1_{m-1})$, the ith segment $\mathbf{q}_i(u)$ of a G^1 quadratic Beta-spline takes the form

$$\mathbf{q}_i(u) = \sum_{j=-1}^{1} \mathbf{V}_{i+j} b_{i+j,j}(\overline{\beta 1}; u), \qquad u \in [0, 1], \quad i = 1, \ldots, m-1, \tag{4.1}$$

where the functions $b_{i+j,j}(\overline{\beta 1}; u)$, called the G^1 Beta-spline blending functions, are quadratic polynomials constructed so that

$$\begin{aligned} \mathbf{q}_{i+1}(0) &= \mathbf{q}_i(1), \\ \mathbf{q}_{i+1}^{(1)}(0) &= \beta 1_i \mathbf{q}_i^{(1)}(1). \end{aligned} \tag{4.2}$$

The Beta-spline blending functions can be determined by solving symbolically a system of linear equations, as was done by the authors in [14]. A more elegant method, due to Farin [16] and Boehm [9], proceeds by describing each segment $\mathbf{q}_i(u)$ in Bézier form. In their approach the ith segment is written as

$$\mathbf{q}_i(u) = \sum_{j=0}^{2} \mathbf{W}_{i,j} B_j^2(u), \tag{4.3}$$

where the Bézier polygon $\mathbf{W}_{i,0}, \mathbf{W}_{i,1}, \mathbf{W}_{i,2}$ is constructed from the control vertices $\mathbf{V}_{i-1}, \mathbf{V}_i, \mathbf{V}_{i+1}$, and the shape parameters $\beta 1_{i-1}$ and $\beta 1_i$, as shown in Construction 3. Before presenting the construction, it may be helpful to note that if a point $\mathbf{C}$ divides a line segment $\mathbf{AB}$ into relative distances $a{:}b$ (see Figure 8), then $\mathbf{C}$ can be expressed as the *affine combination*

$$\mathbf{C} = \frac{b\mathbf{A} + a\mathbf{B}}{a + b}. \tag{4.4}$$

Construction 3: *The Bézier vertices for G^1 Beta-splines.*

1. The interior Bézier vertex $\mathbf{W}_{i,1}$ is defined simply by

$$\mathbf{W}_{i,1} \leftarrow \mathbf{V}_i. \tag{4.5}$$

2. The junction vertex $\mathbf{W}_{i,0}$ divides the line segment $\mathbf{V}_{i-1}\mathbf{V}_i$ into relative distances $1{:}\beta 1_{i-1}$, and the junction vertex $\mathbf{W}_{i,2}$ is set equal to $\mathbf{W}_{i+1,0}$. Algorithmically,

$$\begin{aligned} \mathbf{W}_{i,0} &\leftarrow \frac{\beta 1_{i-1}\mathbf{V}_{i-1} + \mathbf{V}_i}{1 + \beta 1_{i-1}}, \\ \mathbf{W}_{i,2} &\leftarrow \mathbf{W}_{i+1,0} = \frac{\beta 1_i \mathbf{V}_i + \mathbf{V}_{i+1}}{1 + \beta 1_i}. \end{aligned} \tag{4.6}$$

Comparison of Figures 9 and 6 shows that the segments thus constructed do indeed satisfy eqs. (4.2).

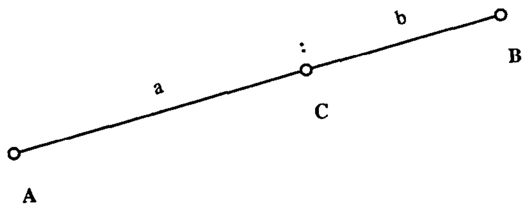

Fig. 8. C divides the line segment AB into relative distances $a:b$.

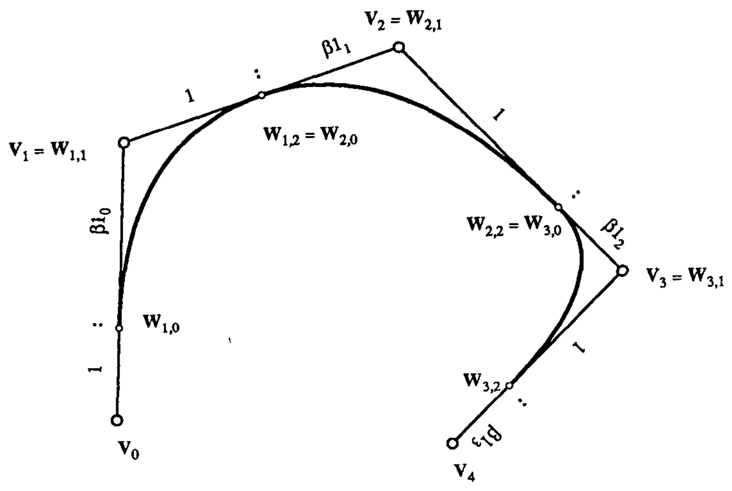

Fig. 9. Construction of the Bézier polygons for a G^1 Beta-spline.

Once the Bézier polygons have been constructed, each segment can be drawn using Bézier curve algorithms [7, 10, 21, 22].

Explicit expressions for the G^1 Beta-spline blending functions $b_{i+j,j}(\overline{\beta 1}; u)$ can be found by substituting eqs. (4.5) and (4.6) into eq. (4.3):

$$\begin{aligned}
\mathbf{q}_i(u) &= \sum_{j=0}^{2} \mathbf{W}_{i,j} B_j^2(u) \\
&= \left[\frac{\beta 1_{i-1}\mathbf{V}_{i-1} + \mathbf{V}_i}{1 + \beta 1_{i-1}}\right] B_0^2(u) + \mathbf{V}_i B_1^2(u) + \left[\frac{\beta 1_i \mathbf{V}_i + \mathbf{V}_{i+1}}{1 + \beta 1_i}\right] B_2^2(u) \\
&= \left[\frac{\beta 1_{i-1} B_0^2(u)}{1 + \beta 1_{i-1}}\right] \mathbf{V}_{i-1} \\
&\quad + \left[\frac{B_0^2(u)}{1 + \beta 1_{i-1}} + B_1^2(u) + \frac{\beta 1_i B_2^2(u)}{1 + \beta 1_i}\right] \mathbf{V}_i + \left[\frac{B_2^2(u)}{1 + \beta 1_i}\right] \mathbf{V}_{i+1}.
\end{aligned}$$

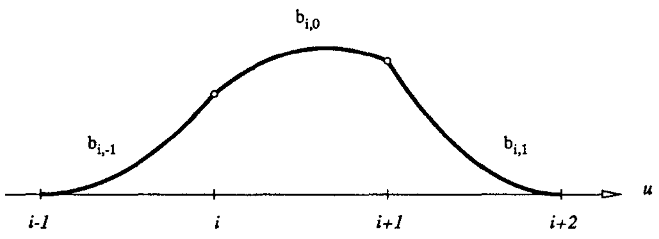

Fig. 10. The blending function $\mathscr{B}_i(\overline{\beta 1}; u)$.

Comparison with eq. (4.1) shows that the G^1 Beta-spline blending functions can be written as

$$\begin{aligned} b_{i-1,-1}(\overline{\beta 1}; u) &= \frac{\beta 1_{i-1} B_0^2(u)}{1 + \beta 1_{i-1}}, \\ b_{i,0}(\overline{\beta 1}; u) &= \frac{B_0^2(u)}{1 + \beta 1_{i-1}} + B_1^2(u) + \frac{\beta 1_i B_2^2(u)}{1 + \beta 1_i}, \\ b_{i+1,1}(\overline{\beta 1}; u) &= \frac{B_2^2(u)}{1 + \beta 1_i}. \end{aligned} \tag{4.7}$$

The strictly polynomial functions $b_{i,-1}(\overline{\beta 1}; u)$, $b_{i,0}(\overline{\beta 1}; u)$, and $b_{i,1}(\overline{\beta 1}; u)$ can be strung together as shown in Figure 10 to form a piecewise blending function $\mathscr{B}_i(\overline{\beta 1}; \bar{u})$, supported on $(i-1, i+2)$, that satisfies the first-order Beta-constraints

$$\begin{aligned} \mathscr{B}_i(\overline{\beta 1}; q^+) &= \mathscr{B}_i(\overline{\beta 1}; q^-), \\ \mathscr{B}_i^{(1)}(\overline{\beta 1}; q^+) &= \beta 1_q \mathscr{B}_i^{(1)}(\overline{\beta 1}; q^-), \end{aligned} \qquad q = i-1, i, i+1, i+2. \tag{4.8}$$

The previous constructions and definitions for G^1 Beta-splines can be extended to define G^2 Beta-splines. A G^2 Beta-spline is defined by a control polygon $\underline{\mathbf{V}_0}, \ldots, \mathbf{V}_m$, and a set of shape parameter values $\overline{\beta 1} = (\beta 1_0, \ldots, \beta 1_m)$ and $\overline{\beta 2} = (\beta 2_0, \ldots, \beta 2_m)$.[2] The ith segment of the curve, $i = 1, \ldots, m-2$, is given by

$$\mathbf{q}_i(u) = \sum_{j=-1}^{2} \mathbf{V}_{i+j} b_{i+j,j}(\overline{\beta 1}, \overline{\beta 2}; u), \qquad u \in [0, 1],$$

where the functions $b_{i+j,j}(\overline{\beta 1}, \overline{\beta 2}; u)$ are cubic polynomial functions constructed so that

$$\begin{aligned} \mathbf{q}_{i+1}(0) &= \mathbf{q}_i(1), \\ \mathbf{q}_{i+1}^{(1)}(0) &= \beta 1_i \mathbf{q}_i^{(1)}(1), \\ \mathbf{q}_{i+1}^{(2)}(0) &= \beta 1_i^2 \mathbf{q}_i^{(2)}(1) + \beta 2_i \mathbf{q}_i^{(1)}(1). \end{aligned} \tag{4.9}$$

Rather than construct the basis functions directly, we follow the approach of Farin and Boehm to construct the Bézier polygons of each of the segments. Let $\mathbf{W}_{i,0}, \ldots, \mathbf{W}_{i,3}$ denote the Bézier polygon of the ith segment, $i = 1, \ldots, m-2$. The first step of the construction proceeds by positioning, for each

[2] Notice that the number of shape parameters is more than twice that for G^1 Beta-splines. The reason for this difference is presented later in this section.

$i = 0, \ldots, m-1$, the two interior Bézier vertices $\mathbf{W}_{i,1}$ and $\mathbf{W}_{i,2}$ on the segment $\mathbf{V}_i\mathbf{V}_{i+1}$ so that the three segments $\mathbf{V}_i\mathbf{W}_{i,1}$, $\mathbf{W}_{i,1}\mathbf{W}_{i,2}$, and $\mathbf{W}_{i,2}\mathbf{V}_{i+1}$ are of relative lengths $\gamma_i : 1 : \beta 1_{i+1}^2 \gamma_{i+1}$ (see Figure 11a), where γ_i is defined as in Construction 2 with $d = 3$:

$$\gamma_i = \frac{2(1 + \beta 1_i)}{\beta 2_i + 2\beta 1_i(1 + \beta 1_i)}.$$

The second step of the construction positions the exterior Bézier vertex $\mathbf{W}_{i-1,3}$ (which is equal to $\mathbf{W}_{i,0}$) to divide the segment $\mathbf{W}_{i-1,2}\mathbf{W}_{i,1}$ into relative distances 1: $\beta 1_i$, as shown in Figure 11b. This construction guarantees that the Bézier vertices for adjacent segments are positioned as required by Figure 7. Hence adjacent segments meet with G^2 (unit tangent and curvature vector) continuity. More specifically, adjacent segments are guaranteed to satisfy eqs. (4.9).

This construction for G^2 Beta-splines may be stated algorithmically as follows:

Construction 4: *The Farin–Boehm construction for G^2 Beta-splines.*

1. For $i = 0, \ldots, m$, compute γ_i from $\beta 1_i$ and $\beta 2_i$:

$$\gamma_i \leftarrow \frac{2(1 + \beta 1_i)}{\beta 2_i + 2\beta 1_i(1 + \beta 1_i)}.$$

2. For $i = 0, \ldots, m-1$, compute the interior Bézier vertices:

$$\mathbf{W}_{i,1} \leftarrow \frac{(1 + \beta 1_{i+1}^2 \gamma_{i+1})\mathbf{V}_i + \gamma_i \mathbf{V}_{i+1}}{1 + \gamma_i + \beta 1_{i+1}^2 \gamma_{i+1}},$$

$$\mathbf{W}_{i,2} \leftarrow \frac{\beta 1_{i+1}^2 \gamma_{i+1} \mathbf{V}_i + (1 + \gamma_i)\mathbf{V}_{i+1}}{1 + \gamma_i + \beta 1_{i+1}^2 \gamma_{i+1}}.$$

3. For $i = 1, \ldots, m-2$, compute the exterior (junction) Bézier vertices:

$$\mathbf{W}_{i,0} \leftarrow \frac{\beta 1_i \mathbf{W}_{i-1,2} + \mathbf{W}_{i,1}}{1 + \beta 1_i},$$

$$\mathbf{W}_{i,3} \leftarrow \mathbf{W}_{i+1,0} = \frac{\beta 1_{i+1} \mathbf{W}_{i,2} + \mathbf{W}_{i+1,1}}{1 + \beta 1_{i+1}}.$$

To determine the number of shape parameters necessary to define a G^2 Beta-spline, we refer to Construction 4. It shows that the ith segment depends on the shape parameters $\beta 1_{i-1}$, $\beta 1_i$, $\beta 1_{i+1}$, $\beta 1_{i+2}$ and $\beta 2_{i-1}$, $\beta 2_i$, $\beta 2_{i+1}$, $\beta 2_{i+2}$. The totality of segments, numbered 1 through $m - 2$, therefore depend on shape parameters indexed from 0 to m, implying that the number of shape parameters is twice the number of control vertices.

As pointed out by Fournier and Barsky [18] and Boehm [9], when designing with Beta-splines, it is often more convenient for the designer to specify the interior Bézier points directly than have the system compute the shape parameters and place the junction points. In particular, the method of Farin and Boehm proceeds by having the designer specify a control polygon, $\mathbf{V}_0, \ldots, \mathbf{V}_m$, and a pair of points, $\mathbf{W}_{i,1}$ and $\mathbf{W}_{i,2}$, on each leg of the polygon. From this input, the

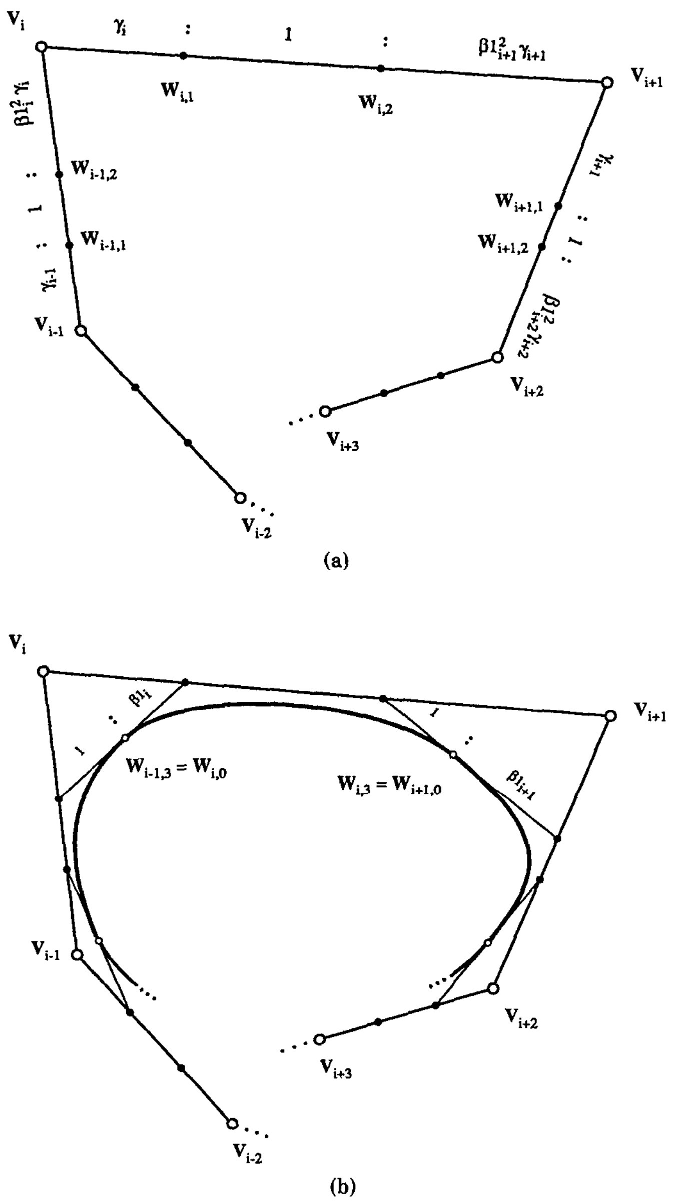

Fig. 11. The Farin–Boehm construction for G^2 Beta-splines. (a) The construction for the interior Bézier vertices. (b) The construction of the junction vertices.

shape parameters are uniquely determined (as long as $\mathbf{W}_{i,1} \neq \mathbf{W}_{i,2}$) and can be computed automatically by the system. The shape parameters are then used to compute the junction points, thereby completing the determination of each segment in Bézier form. The curve can then be drawn using standard techniques for Bézier curves [7, 10, 21, 22].

As was done for the G^1 Beta-spline, the G^2 Beta-spline basis functions $b_{i+j,j}(\overline{\beta 1}, \overline{\beta 2}; u)$ can be determined from the construction of the Bézier polygons (Construction 4). The basis functions thus determined can be strung together in a manner analogous to Figure 10 to define piecewise cubic blending functions $\mathscr{B}_i(\overline{\beta 1}, \overline{\beta 2}; \bar{u})$ that satisfy

$$\begin{aligned}
\mathscr{B}_i(\overline{\beta 1}, \overline{\beta 2}; q^+) &= \mathscr{B}_i(\overline{\beta 1}, \overline{\beta 2}; q^-),\\
\mathscr{B}_i^{(1)}(\overline{\beta 1}, \overline{\beta 2}; q^+) &= \beta 1_q \mathscr{B}_i^{(1)}(\overline{\beta 1}, \overline{\beta 2}; q^-),\\
\mathscr{B}_i^{(2)}(\overline{\beta 1}, \overline{\beta 2}; q^+) &= \beta 1_q^2 \mathscr{B}_i^{(2)}(\overline{\beta 1}, \overline{\beta 2}; q^-)\\
&\quad + \beta 2_q \mathscr{B}_i^{(1)}(\overline{\beta 1}, \overline{\beta 2}; q^-),
\end{aligned} \qquad q = i-1, \ldots, i+3. \tag{4.10}$$

Although we have demonstrated the construction of G^1 and G^2 Beta-splines, we have not established that Beta-splines of all orders exist. It is, in fact, possible to construct G^n Beta-splines for arbitrary $n \geq 1$, as recently shown by Goodman [20] and Dyn and Micchelli [15]. Thus, given a set of shape parameter values $\overline{\beta 1}, \ldots, \overline{\beta n}$, it is possible to find piecewise polynomial blending functions $\mathscr{B}_i(\overline{\beta 1}, \ldots, \overline{\beta n}; \bar{u})$ that satisfy the nth-order Beta-constraints with respect to the given shape parameters. Goodman and Dyn and Micchelli also show that these functions have local support and that their segments have degree $n + 1$. However, an algorithm for geometrically constructing the Bézier polygons of a G^n Beta-spline for arbitrary n and for arbitrary shape parameters is currently unknown.

5. GEOMETRICALLY CONTINUOUS CATMULL–ROM SPLINES

We now apply the notion of geometric continuity to the class of Catmull–Rom splines. The resulting class can conveniently be described by Table I. The rows of Table I correspond to k, the width of the interpolating window used in the construction of the function $\mathbf{P}_i(u)$ in eq. (2.6), and the columns correspond to n, the order of geometric continuity. The splines in the first column possess one shape parameter per joint, the splines in the second column possess two shape parameters per joint, and so on.

In Section 5.1 we show that the problem of constructing geometrically continuous Catmull–Rom splines can be decoupled into two simpler problems:

(1) constructing geometrically continuous blending functions, and
(2) constructing geometrically continuous interpolating functions.

For G^1 and G^2 continuity, the blending functions are known—they are the Beta-spline blending functions of Section 4. In Section 5.2.2 we demonstrate the construction of G^1 and G^2 geometrically continuous interpolating functions so as to complete the development of G^1 and G^2 Catmull–Rom splines.

In Section 5.3 we shall derive a form for the G^n Catmull–Rom blending functions, primarily to point out several properties of splines in the class. In practice, however, we do not recommend drawing these curves by explicitly computing the blending functions. More efficient evaluation algorithms are developed in Section 6. To aid the practitioner, pseudocode versions of these algorithms for the (G^1, $k = 1$) and the (G^2, $k = 2$) Catmull–Rom splines of Table I are also provided.

Table I

k	G^1	G^2	$\cdots$
0	Approximating	Approximating	$\cdots$
1	Interpolating[a]	Approximating	$\cdots$
2	Interpolating	Interpolating[a]	
$\vdots$	$\vdots$	$\vdots$	$\ddots$

[a] Constructed in Section 6.

5.1 Decoupling

When we say that the piecewise parameterization $\mathbf{F}(u)$ is G^n, we mean that successive segments of $\mathbf{F}(u)$ meet with G^n continuity. More precisely, for each q, there must exist a change of variables u_q: $[q-1, q] \mapsto [\tilde{a}, \tilde{b}]$ such that

$$\mathbf{f}_q^{(j)}(q) = \tilde{\mathbf{f}}_{q-1}^{(j)}(\tilde{b}), \qquad j = 1, \ldots, n, \tag{5.1}$$

where

$$\tilde{\mathbf{f}}_{q-1}(\tilde{u}) = \mathbf{f}_{q-1}(u_q(\tilde{u})).$$

We are free to choose $\tilde{b}$ (by applying an appropriate linear shift of the parameter line), and for our purposes the particular value of $\tilde{a}$ is irrelevant. By choosing $\tilde{b} = q$, the condition from eq. (5.1) can be stated in terms of $\mathbf{F}(u)$ as

$$\mathbf{F}^{(j)}(q^+) = \tilde{\mathbf{F}}^{(j)}(q^-), \qquad j = 1, \ldots, n, \tag{5.2}$$

where

$$\begin{aligned} \tilde{\mathbf{F}}(\tilde{u}) &= \mathbf{F}(u_q(\tilde{u})) \\ &= \sum_i \mathbf{P}_i(u_q(\tilde{u}))W_i(u_q(\tilde{u})) \\ &= \sum_i \tilde{\mathbf{P}}_i(\tilde{u})\tilde{W}_i(\tilde{u}). \end{aligned}$$

By rewriting eq. (5.2) as

$$\frac{d^j}{du^j}\left[\sum_i \mathbf{P}_i(q^+)W_i(q^+)\right] = \frac{d^j}{d\tilde{u}^j}\left[\sum_i \tilde{\mathbf{P}}_i(q^-)\tilde{W}_i(q^-)\right],$$

we see that it is sufficient to require that

$$\begin{aligned} &1.\ \mathbf{P}_i^{(j)}(q^+) = \tilde{\mathbf{P}}_i^{(j)}(q^-), \\ &2.\ W_i^{(j)}(q^+) = \tilde{W}_i^{(j)}(q^-), \end{aligned} \qquad j = 1, \ldots, n, \qquad q = i, i+1, \ldots, i+D-2. \tag{5.3}$$

Let us focus for the moment on the first set of conditions in eq. (5.3). Expanding the right side using the chain rule, we find that

$$\begin{aligned} \mathbf{P}_i^{(j)}(q^+) &= \mathrm{CR}_j(\mathbf{P}_i^{(1)}(q^-), \ldots, \mathbf{P}_i^{(j)}(q^-), u_q^{(1)}(q), \ldots, u_q^{(j)}(q)) \\ &= \mathrm{CR}_j(\mathbf{P}_i^{(1)}(q^-), \ldots, \mathbf{P}_i^{(j)}(q^-), \beta 1_q, \ldots, \beta j_q). \end{aligned} \tag{5.4}$$

A similar set of conditions hold for the blending functions $W_i(u)$:

$$\begin{aligned} W_i^{(j)}(q^+) &= \mathrm{CR}_j(W_i^{(1)}(q^-), \ldots, W_i^{(j)}(q^-), u_q^{(1)}(q), \ldots, u_q^{(j)}(q)) \\ &= \mathrm{CR}_j(W_i^{(1)}(q^-), \ldots, W_i^{(j)}(q^-), \beta 1_q, \ldots, \beta j_q). \end{aligned} \tag{5.5}$$

Note that the shape parameters that enter in eqs. (5.4) and (5.5) are the same. Thus, to construct a G^n Catmull–Rom spline $\mathbf{F}(u)$, it is sufficient to use interpolating functions $\mathbf{P}_i(u)$ and blending functions $W_i(u)$ that separately satisfy the nth-order Beta-constraints with respect to the *same* set of shape parameters.

5.2 Geometrically Continuous Interpolating Functions

Before embarking on the derivation of the geometrically continuous interpolating functions $\mathbf{P}_i(u)$, we examine the functions originally used by Catmull and Rom to show that the functions they chose do not satisfy the Beta-constraints for arbitrary shape parameters (eqs. (5.4)) and hence cannot be used to construct a geometrically continuous Catmull–Rom spline. However, a generalization of the functions used by Catmull and Rom can be used.

5.2.1 *Lagrange Interpolation.* Recall that the function $\mathbf{P}_i(u)$ must be constructed to interpolate the vertices $\mathbf{V}_i$, $\mathbf{V}_{i+1}$, ..., $\mathbf{V}_{i+k}$, for some nonnegative integer k. Catmull and Rom chose functions of the form

$$\mathbf{P}_i(u) = \sum_{j=0}^{k} \mathbf{V}_{i+j} L_j(k; u - i), \tag{5.6}$$

where the $L_j(k; u)$ are the classical Lagrange polynomials, defined by

$$L_j(k; u) = \prod_{\substack{p=0 \\ p \neq j}}^{k} \left(\frac{u - p}{j - p} \right). \tag{5.7}$$

It is easy to show that the Lagrange polynomials satisfy the *Kronecker delta relation*

$$L_j(k; r) = \delta_{j,r} = \begin{cases} 1 & \text{if } j = r, \\ 0 & \text{otherwise,} \end{cases} \qquad r = 0, \ldots, k. \tag{5.8}$$

In fact, any set of functions satisfying eq. (5.8) can be used to construct an interpolating function of the form given in (5.6).

We now examine the continuity of $\mathbf{P}_i(u)$. A function $\mathbf{P}_i(u)$ as in eq. (5.6) is a single polynomial of degree k and as such is everywhere C^∞ continuous. It is impossible for such a function to have the derivative discontinuities required by the Beta-constraints (eqs. (5.4)). Intuitively, the Lagrange polynomials are *too smooth* to be geometrically continuous with respect to arbitrary shape parameters.

5.2.2 *G^n Lagrange Interpolation.* Since a single polynomial is everywhere C^∞, we must resort to a piecewise polynomial representation to obtain interpolating functions that satisfy the Beta-constraints of eq. (5.4). We choose functions of the same form as eq. (5.6), but we replace the Lagrange polynomials with a set of piecewise polynomial functions $\Lambda_{i,j}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; u)$ to get

$$\mathbf{P}_i(u) = \sum_{j=0}^{k} \mathbf{V}_{i+j}\Lambda_{i,j}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; u). \tag{5.9}$$

The properties of $\Lambda_{i,j}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; u)$ naturally determine the behavior of $\mathbf{P}_i(u)$. For $\mathbf{P}_i(u)$ to interpolate $\mathbf{V}_i, \mathbf{V}_{i+1}, \ldots, \mathbf{V}_{i+k}$, the functions $\Lambda_{i,j}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; u)$ must satisfy the Kronecker delta relation

$$\Lambda_{i,j}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; r) = \delta_{j,r-i}, \qquad r = i, i+1, \ldots, i+k. \tag{5.10}$$

Moreover, since the continuity of $\Lambda_{i,j}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; u)$ is inherited by $\mathbf{P}_i(u)$, the functions $\Lambda_{i,j}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; u)$ must be constructed to satisfy the Beta-constraints at the parametric values $u = i, i+1, \ldots, i+D-2$. Note that the number of polynomial segments of $\Lambda_{i,j}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; u)$ is related to the width of the blending functions that will be used to weight the interpolating functions $\mathbf{P}_i(u)$. In particular, $\Lambda_{i,j}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; u)$ requires D segments.

Although the Lagrange polynomials have a concise definition (eq. (5.7)), we do not know of a closed form for the functions $\Lambda_{i,j}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; u)$ for arbitrary order of continuity (n), width of the interpolating window (k), and the support of the blending functions (D); currently, they must be constructed on a case-by-case basis for a given n, k, and D. However, in all the cases we consider, $D = n + 2$, so the functions $\Lambda_{i,j}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; u)$ are determined by the two integers n and k.

We demonstrate their construction by the following two examples, the results of which are used in Section 6 to construct the (G^1, $k = 1$) and (G^2, $k = 2$) splines of Table I. Rather than derive the Λ's directly, we take an approach similar to the one used in Section 3.1. Indeed, as the algorithms of Section 6 show, the functions $\Lambda_{i,j}$ are not (explicitly) needed to compute G^n Catmull–Rom splines.

Example 5.1. The interpolating functions $\mathbf{P}_i(u)$ for $k = 1$ and $n = 1$; that is, G^1 continuity with $D = 3$ must satisfy the four conditions:

$$\begin{aligned} \mathbf{P}_i(i) &= \mathbf{V}_i, \\ \mathbf{P}_i(i+1) &= \mathbf{V}_{i+1}, \\ \mathbf{P}_i^{(1)}(i^+) &= \beta 1_i \mathbf{P}_i^{(1)}(i^-), \\ \mathbf{P}_i^{(1)}(i+1^+) &= \beta 1_{i+1} \mathbf{P}_i^{(1)}(i+1^-). \end{aligned} \tag{5.11}$$

If it were not for the fact that $\mathbf{P}_i(u)$ must possess derivative discontinuities as prescribed by the last two constraints of eq. (5.11), we could define $\mathbf{P}_i(u)$ by a single linear polynomial as was done in eq. (2.3). To introduce the appropriate discontinuities, we break $\mathbf{P}_i(u)$ into three linear segments $\mathbf{p}_{i,0}$, $\mathbf{p}_{i,1}$, and $\mathbf{p}_{i,2}$ such that

$$\mathbf{P}_i(u) = \begin{cases} \mathbf{p}_{i,0}(u - (i-1)), & u < i, \\ \mathbf{p}_{i,1}(u - i), & i \le u < i+1, \\ \mathbf{p}_{i,2}(u - (i+1)), & i+1 \le u. \end{cases}$$

Given this form, conditions (5.11) are easily satisfied by expressing the segments of $\mathbf{P}_i(u)$ in Bézier form, as demonstrated by the next construction.

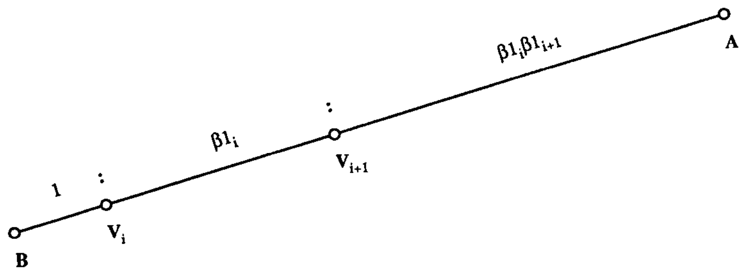

Fig. 12. Geometric interpretation of Construction 5.

Construction 5: *The Bézier vertices for G^1 Lagrange curves.* Given control vertices $\mathbf{V}_i$, and $\mathbf{V}_{i+1}$, shape parameters $\beta 1_i$, and $\beta 1_{i+1}$, construct

$$1.\ \mathbf{A} \leftarrow \mathbf{V}_{i+1} + \beta 1_{i+1}(\mathbf{V}_{i+1} - \mathbf{V}_i),$$
$$2.\ \mathbf{B} \leftarrow \mathbf{V}_i + \frac{1}{\beta 1_i}(\mathbf{V}_i - \mathbf{V}_{i+1}),$$

as shown in Figure 12. Satisfaction of eqs. (5.11) is then guaranteed by setting the Bézier polygon for $\mathbf{p}_{i,0}$ to $(\mathbf{B}, \mathbf{V}_i)$, setting the Bézier polygon for $\mathbf{p}_{i,1}$ to $(\mathbf{V}_i, \mathbf{V}_{i+1})$, and setting the Bézier polygon for $\mathbf{p}_{i,2}$ to $(\mathbf{V}_{i+1}, \mathbf{A})$.

The functions $\mathrm{A}_{i,j}(k;\ \overline{\beta 1}, \ldots, \overline{\beta n};\ u)$ of eq. (5.9) can be determined from Construction 5 in much the same way that the functions $\mathscr{B}_i(\overline{\beta 1};\ u)$ followed from Construction 3. □

Example 5.2. The interpolating functions $\mathbf{P}_i(u)$ for G^2 continuity and $k = 2$ must satisfy

$$\begin{aligned} \mathbf{P}_i(q) &= \mathbf{V}_q, \\ \mathbf{P}_i^{(1)}(q^+) &= \beta 1_q \mathbf{P}_i^{(1)}(q^-), \qquad\qquad q = i, i+1, i+2. \\ \mathbf{P}_i^{(2)}(q^+) &= \beta 1_q^2 \mathbf{P}_i^{(2)}(q^-) + \beta 2_q \mathbf{P}_i^{(1)}(q^-), \end{aligned} \tag{5.12}$$

Once again, we construct the Bézier polygons for the $(D = 4)$ quadratic segments of $\mathbf{P}_i(u)$, denoted by $\mathbf{p}_{i,0}, \ldots, \mathbf{p}_{i,3}$, as shown in Figure 13.

The interpolation conditions, that is, the first set of conditions in eq. (5.12) trivially determine the last Bézier vertex of $\mathbf{p}_{i,0}$, the first and last Bézier vertices of $\mathbf{p}_{i,1}$ and $\mathbf{p}_{i,2}$, and the first Bézier vertex of $\mathbf{p}_{i,3}$, leaving only the six unknown vertices labeled **A** through **F** in Figure 13.

Let us concentrate for the moment on the joint between $\mathbf{p}_{i,1}$ and $\mathbf{p}_{i,2}$. The interior Bézier vertices **A** and **B** must be positioned so that $\mathbf{p}_{i,1}$ and $\mathbf{p}_{i,2}$ meet with curvature continuity at $\mathbf{V}_{i+1}$ with respect to the shape parameters $\beta 1_{i+1}$ and $\beta 2_{i+1}$. The key to the solution of this problem is the Farin–Boehm construction (see Figure 7). We must construct **A** and **B** so that Figure 14 holds; stated more precisely, **A** and **B** must simultaneously satisfy the two equations

$$\begin{aligned} \mathbf{B} &= \mathbf{V}_{i+1} + \beta 1_{i+1}(\mathbf{V}_{i+1} - \mathbf{A}), \\ \mathbf{T} &= \mathbf{A} + \beta 1_{i+1}^2 \gamma_{i+1}(\mathbf{A} - \mathbf{V}_i) = \mathbf{B} + \gamma_{i+1}(\mathbf{B} - \mathbf{V}_{i+2}), \end{aligned} \tag{5.13}$$

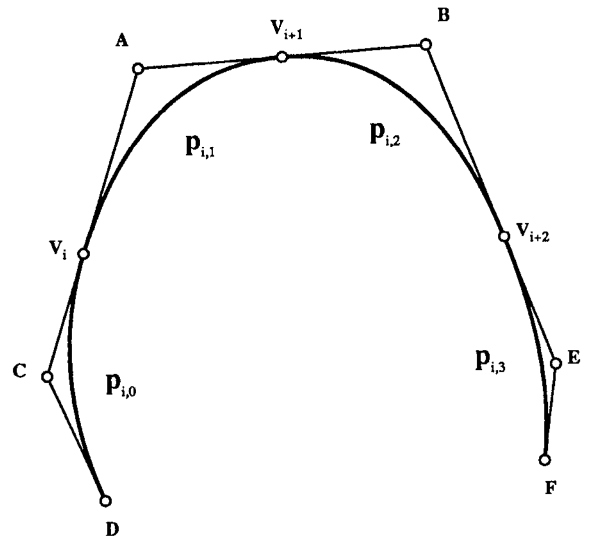

Fig. 13. The arrangement of the segments of $\mathbf{P}_i(u)$ for $k = 2$ and $D = 4$.

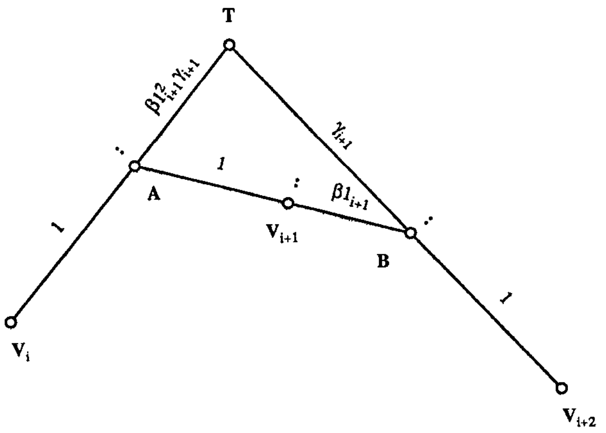

Fig. 14. The Farin–Boehm construction with interior points **A** and **B**.

where γ_{i+1} is computed from $\beta 1_{i+1}$ and $\beta 2_{i+1}$ as in the Farin–Boehm construction (Construction 2) with $d = 2$. Using a symbolic algebra system [17], the solution of this pair of equations for **A** was found to be

$$\mathbf{A} = \frac{\beta 1^2_{i+1}\gamma_{i+1}\mathbf{V}_i + (1 + \gamma_{i+1})(1 + \beta 1_{i+1})\mathbf{V}_{i+1} - \gamma_{i+1}\mathbf{V}_{i+2}}{(1 + \beta 1_{i+1})(1 + \beta 1_{i+1}\gamma_{i+1})}.$$

The Bézier vertex **B** can then be constructed using the first equation of (5.13). The four remaining unknown Bézier vertices **C**, **D**, **E**, and **F** can also be found using the Farin–Boehm construction, as shown in Figure 15. The complete

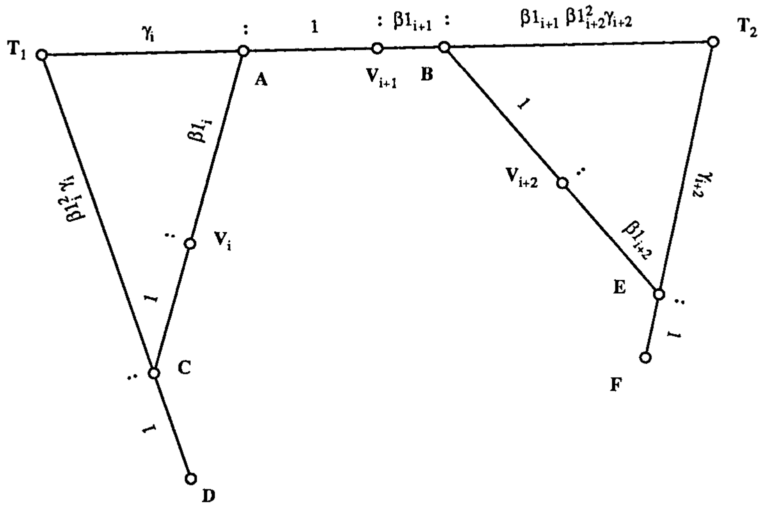

Fig. 15. The complete construction for a G^2 Lagrange curve.

construction of the Bézier polygons for the segments of a G^2 Lagrange curve through the vertices $\mathbf{V}_i$, $\mathbf{V}_{i+1}$, and $\mathbf{V}_{i+2}$ can be stated as:

Construction 6: *The Bézier vertices for a G^2 Lagrange curve interpolating* $\mathbf{V}_i$, $\mathbf{V}_{i+1}$, $\mathbf{V}_{i+2}$.

1. $\gamma_i \leftarrow \dfrac{1 + \beta 1_i}{\beta 2_i + \beta 1_i(1 + \beta 1_i)}$,
2. $\gamma_{i+1} \leftarrow \dfrac{1 + \beta 1_{i+1}}{\beta 2_{i+1} + \beta 1_{i+1}(1 + \beta 1_{i+1})}$,
3. $\gamma_{i+2} \leftarrow \dfrac{1 + \beta 1_{i+2}}{\beta 2_{i+2} + \beta 1_{i+2}(1 + \beta 1_{i+2})}$,
4. $\mathbf{A} \leftarrow \dfrac{\beta 1^2_{i+1}\gamma_{i+1}\mathbf{V}_i + (1 + \gamma_{i+1})(1 + \beta 1_{i+1})\mathbf{V}_{i+1} - \gamma_{i+1}\mathbf{V}_{i+2}}{(1 + \beta 1_{i+1})(1 + \beta 1_{i+1}\gamma_{i+1})}$,
5. $\mathbf{B} \leftarrow \mathbf{V}_{i+1} + \beta 1_{i+1}(\mathbf{V}_{i+1} - \mathbf{A})$,
6. $\mathbf{C} \leftarrow \mathbf{V}_i + \dfrac{1}{\beta 1_i}(\mathbf{V}_i - \mathbf{A})$,
7. $\mathbf{T}_1 \leftarrow \mathbf{A} + \gamma_i(\mathbf{A} - \mathbf{V}_{i+1})$,
8. $\mathbf{D} \leftarrow \mathbf{C} + \dfrac{1}{\beta 1^2_i \gamma_i}(\mathbf{C} - \mathbf{T}_1)$,
9. $\mathbf{E} \leftarrow \mathbf{V}_{i+2} + \beta 1_{i+2}(\mathbf{V}_{i+2} - \mathbf{B})$,
10. $\mathbf{T}_2 \leftarrow \mathbf{B} + \beta 1^2_{i+2}\gamma_{i+2}(\mathbf{B} - \mathbf{V}_{i+1})$,
11. $\mathbf{F} \leftarrow \mathbf{E} + \dfrac{1}{\gamma_{i+2}}(\mathbf{E} - \mathbf{T}_2)$.

The Bézier polygons for $\mathbf{p}_{i,0}(u), \ldots, \mathbf{p}_{i,3}(u)$ are then assigned as shown in Table II.

Table II

Segment	Bézier polygon
$\mathbf{p}_{i,0}(u)$	$(\mathbf{D}, \mathbf{C}, \mathbf{V}_i)$
$\mathbf{p}_{i,1}(u)$	$(\mathbf{V}_i, \mathbf{A}, \mathbf{V}_{i+1})$
$\mathbf{p}_{i,2}(u)$	$(\mathbf{V}_{i+1}, \mathbf{B}, \mathbf{V}_{i+2})$
$\mathbf{p}_{i,3}(u)$	$(\mathbf{V}_{i+2}, \mathbf{E}, \mathbf{F})$

The piecewise functions $\Lambda_{i,j}(2; \overline{\beta 1}, \overline{\beta 2}; u)$ can be determined from Construction 6 in much the same way the piecewise functions $\mathscr{B}_i(\overline{\beta 1}, \overline{\beta 2}; u)$ followed from Construction 4. □

Although we have no proof, we believe that it is always possible to find piecewise polynomial functions $\Lambda_{i,j}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; u)$ of degree $\max(n, k)$, subject to the nth-order Beta-constraints and the Kronecker delta relation of eq. (5.10).

5.3 The General Form

In this section we merge a set of geometrically continuous blending functions $W_i(u)$ with a set of geometrically continuous interpolating functions $\mathbf{P}_i(u)$ to produce a geometrically continuous Catmull–Rom spline. Using the decoupling result of Section 5.1, this is done by using G^n Beta-splines for the blending functions and G^n Lagrange curves for the interpolating functions. A convenient form for the segments of $\mathbf{F}(u)$ can be obtained by starting with eq. (2.6) for the qth segment $\mathbf{f}_q(u)$:

$$\mathbf{f}_q(u) = \sum_{i=2-D}^{1} \mathbf{P}_{q+i}(u) W_{q+i}(u), \qquad u \in [q, q+1).$$

By substituting the form for $\mathbf{P}_{q+i}(u)$ from eq. (5.9) and using the relation $D = n + 2$ for Beta-splines, we obtain

$$\mathbf{f}_q(u) = \sum_{i=-n}^{1} \left\{ \sum_{j=0}^{k} \mathbf{V}_{q+i+j} \Lambda_{q+i,j}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; u) \right\} \mathscr{B}_{q+i}(\overline{\beta 1}, \ldots, \overline{\beta n}; u),$$
$$u \in [q, q+1).$$

Changing summation indices yields

$$\mathbf{f}_q(u) = \sum_{l=-n}^{k+1} \mathbf{V}_{q+l} \left\{ \sum_{i+j=l} \Lambda_{q+i,j}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; u) \mathscr{B}_{q+i}(\overline{\beta 1}, \ldots, \overline{\beta n}; u) \right\},$$

where the inner summation is to be taken over all $i = -n, \ldots, +1$ and $j = 0, \ldots, k$ such that $i + j = l$. Finally, by setting

$$\Phi_{q,l}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; u) = \sum_{i+j=l} \Lambda_{q+i,j}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; u) \mathscr{B}_{q+i}(\overline{\beta 1}, \ldots, \overline{\beta n}; u), \qquad (5.14)$$

we obtain

$$\mathbf{f}_q(u) = \sum_{l=-n}^{k+1} \mathbf{V}_{q+l} \Phi_{q,l}(k; \overline{\beta 1}, \ldots, \overline{\beta n}; u), \qquad u \in [q, q+1). \qquad (5.15)$$

Equations (5.14) and (5.15) together define the class of geometrically continuous Catmull–Rom splines. A particular member of the class is determined by n, the order of geometric continuity, and k, the width of the interpolating window. The functions $\Phi_{q,l}(k; \beta 1, \ldots, \beta n; u)$ are called the *geometrically continuous Catmull–Rom blending functions*. Several important properties of the class can now be identified:

(1) Every member of the class has local control. From eq. (5.15), $\mathbf{f}_q(u)$ depends only on the $k + n + 2$ vertices $\mathbf{V}_{q-n}, \mathbf{V}_{q-n+1}, \ldots, \mathbf{V}_{q+k+1}$. Modification of vertices outside this range has no effect on the segment. Thus, perturbation of a given vertex will only affect $k + n + 2$ segments near it.

(2) Every member has shape parameters. The G^1 splines have one shape parameter per joint, the G^2 splines have two shape parameters per joint, and in general the G^n splines have n shape parameters per joint. Owing to the local control property (1), modification of a particular shape parameter affects at most $k + n + 2$ segments of the curve.

(3) Members of this class can be either interpolating or approximating. Since this class is a proper subclass of the Catmull–Rom splines, if $k < n$, the spline will approximate the vertices; otherwise, it will interpolate the vertices.

(4) In general, the (G^n, k) spline is (we believe) of polynomial degree

$$(n + 1) \cdot \max(n, k).$$

This follows from eq. (5.14), the fact that a G^n Beta-spline blending function $\mathscr{B}_{q+i}(\beta 1, \ldots, \beta n; u)$ is of degree $n + 1$, and the belief that a G^n interpolating blending function $\Lambda_{q+i,j}(k; \beta 1, \ldots, \beta n; u)$ is of degree $\max(n, k)$.

If a G^1 Catmull–Rom spline is desired (a member of the first column of Table I), the functions $\mathscr{B}_i(\beta 1; u)$ are the G^1 Beta-spline blending functions from eq. (4.8), and the functions $\Lambda_{i,j}(1; \beta 1; u)$ follow from Construction 5. To define a G^2 Catmull–Rom spline (a member of the second column of Table I), the $\mathscr{B}_i(\beta 1, \beta 2; u)$ are the G^2 Beta-spline blending functions from eq. (4.10), and the functions $\Lambda_{i,j}(2; \beta 1, \beta 2; u)$ follow from Construction 6.

6. EVALUATION AND RENDERING OF CATMULL–ROM SPLINES

Owing to the algebraic complexity of each of the terms in eq. (5.14), we do not recommend computing or rendering a G^n Catmull–Rom spline by repeated evaluation of eqs. (5.14) and (5.15). A better approach, which we now develop, is to compute the Bèzier polygon for each segment of the G^n Catmull–Rom curve. Once the Bèzier polygons have been constructed, the segments can be displayed using Bèzier curve algorithms, such as recursive subdivision [7, 21, 22] or de Casteljau's algorithm [10]. Explicit examples of this approach are given for the $(G^1, k = 1)$ and $(G^2, k = 2)$ splines of Table I.

The development of the general algorithm for arbitrary n and k begins by writing the qth segment of $\mathbf{F}(u)$ as in eq. (2.6), where the interpolating functions $\mathbf{P}_i(u)$ are the G^n Lagrange curves of Section 5.2.2, and the blending functions $W_i(u)$ are the G^n Beta-spline blending functions of Section 4. Let K denote the degree of the segments of $\mathbf{P}_i(u)$, and let $\mathbf{R}_{i+n,0}, \ldots, \mathbf{R}_{i+n,K}$ denote the Bèzier polygon for the segment of $\mathbf{P}_{q+i}(u)$ corresponding to the interval $u \in [q, q+1)$;[3] that is, let $\mathbf{R}_{i+n,0}, \ldots, \mathbf{R}_{i+n,K}$ be such that

$$\mathbf{P}_{q+i}(u) = \sum_{j=0}^{K} \mathbf{R}_{i+n,j} B_j^K(u-q), \qquad u \in [q, q+1). \tag{6.1}$$

Constructions 5 and 6 demonstrate that it is often easy to determine these Bézier polygons. Substituting eq. (6.1) and the relation $D = n + 2$ for Beta-splines into eq. (2.6) allows $\mathbf{f}_q(u)$ to be written as

$$\begin{aligned}\mathbf{f}_q(u) &= \sum_{i=-n}^{1} \left\{ \sum_{j=0}^{K} \mathbf{R}_{i+n,j} B_j^K(u-q) \right\} \mathscr{B}_{q+i}(\overline{\beta 1}, \ldots, \overline{\beta n}; u) \\ &= \sum_{j=0}^{K} \left\{ \sum_{i=-n}^{1} \mathbf{R}_{i+n,j} \mathscr{B}_{q+i}(\overline{\beta 1}, \ldots, \overline{\beta n}; u) \right\} B_j^K(u-q), \end{aligned} \qquad u \in [q, q+1).$$

The term in braces can be thought of as a Beta-spline curve with control polygon $\mathbf{R}_{i',j}$, $i' = 0, \ldots, n+1$, which for G^n continuity will be of degree $n+1$. Denote the Bézier polygon for this curve by $\mathbf{S}_{i',j}$, $i' = 0, \ldots, n+1$, the computation of which is quite simple for G^1 and G^2 continuity, as was shown in Section 4. With this notation,

$$\mathbf{f}_q(u) = \sum_{j=0}^{K} \left\{ \sum_{i'=0}^{n+1} \mathbf{S}_{i',j} B_{i'}^{n+1}(u-q) \right\} B_j^K(u-q), \qquad u \in [q, q+1].$$

Dropping the prime on i and rewriting $\mathbf{f}_q$ more conveniently as a [0, 1] parameterization yield

$$\mathbf{f}_q(u) = \sum_{j=0}^{K} \left\{ \sum_{i=0}^{n+1} \mathbf{S}_{i,j} B_i^{n+1}(u) \right\} B_j^K(u), \qquad u \in [0, 1]. \tag{6.2}$$

The right side of eq. (6.2) is a polynomial of degree $K + n + 1$, so it must be possible to find a Bézier polygon $\mathbf{b}_0, \ldots, \mathbf{b}_{K+n+1}$ such that

$$\mathbf{f}_q(u) = \sum_{l=0}^{K+n+1} \mathbf{b}_l B_l^{K+n+1}(u), \qquad u \in [0, 1].$$

This Bézier polygon is remarkably easy to compute from the Bézier polygons $\mathbf{S}_{i,j}$. The essential fact is contained in the following lemma.

LEMMA 6.1

$$B_i^{n+1}(u) B_j^K(u) = \begin{pmatrix} n+1, K \\ i, j \end{pmatrix} B_{i+j}^{n+K+1}(u),$$

where

$$\begin{pmatrix} n, K \\ i, j \end{pmatrix} = \frac{\binom{n}{i}\binom{K}{j}}{\binom{n+K}{i+j}}.$$

[3] Strictly speaking, this Bézier polygon should be ornamented with a superscript q; we have chosen to omit the superscript to simplify the notation. We have also chosen the first subscript on the **R**'s to be $i + n$ (rather than simply i) to simplify the indexing in the remainder of the derivation.

PROOF

$$\begin{aligned}
B_i^{n+1}(u)B_j^K(u) &= \binom{n+1}{i}u^i(1-u)^{n+1-i}\binom{K}{j}u^j(1-u)^{K-j}\\
&= \binom{n+1}{i}\binom{K}{j}u^{i+j}(1-u)^{K+n+1-i-j}\\
&= \binom{n+1}{i}\binom{K}{j}\frac{\binom{K+n+1}{i+j}}{\binom{K+n+1}{i+j}}u^{i+j}(1-u)^{K+n+1-(i+j)}\\
&= \frac{\binom{n+1}{i}\binom{K}{j}}{\binom{K+n+1}{i+j}}B_{i+j}^{K+n+1}(u)\\
&= \binom{n+1,\ K}{i,\ j}B_{i+j}^{n+K+1}(u).
\end{aligned}$$

□

Using Lemma 6.1, eq. (6.2) can be written as

$$\begin{aligned}
\mathbf{f}_q(u) &= \sum_{j=0}^{K}\sum_{i=0}^{n+1}\binom{n+1,\ K}{i,\ j}\mathbf{S}_{i,j}B_{i+j}^{K+n+1}(u)\\
&= \sum_{l=0}^{K+n+1}\left\{\sum_{i+j=l}\binom{n+1,\ K}{i,\ j}\mathbf{S}_{i,j}\right\}B_l^{K+n+1}(u)\\
&= \sum_{l=0}^{K+n+1}\mathbf{b}_l B_l^{K+n+1}(u),
\end{aligned}$$

where we have made the identification

$$\mathbf{b}_l = \sum_{i+j=l}\binom{n+1,\ K}{i,\ j}\mathbf{S}_{i,j}. \tag{6.3}$$

The summation in eq. (6.3) is taken over all values of $i = 0, \ldots, n+1$ and $j = 0, \ldots, K$ such that $i + j = l$.

Given an algorithm for constructing the Bézier polygons of the interpolating functions and an algorithm for constructing the Bézier polygons of a G^n Beta-spline, the steps leading from eq. (2.6) to eq. (6.3) define an algorithm for constructing the Bézier polygons of a G^n Catmull–Rom spline. The structure of this algorithm for arbitrary n and k can be explained schematically by arranging the points $\mathbf{R}_{i,j}$ of eq. (6.1), and the points $\mathbf{S}_{i,j}$ of eq. (6.2), in arrays using standard row-major ordering, as shown in Figure 16:

(1) The rows of the "**R**-array" correspond to the Bézier polygons of the interpolating functions $\mathbf{P}_i(u)$, and are computed from the Catmull–Rom polygon $\mathbf{V}_0, \ldots, \mathbf{V}_m$. Constructions 5 and 6 demonstrate the computation of the **R**-array for G^1 and G^2 continuity.

(2) The columns of the "**S**-array" are computed from the columns of the **R**-array using Beta-spline constructions similar to Constructions 3 and 4. More precisely, as indicated by Figure 16, the ith column of the **S**-array is the Bézier polygon of the Beta-spline curve whose control polygon is the ith column of the **R**-array.

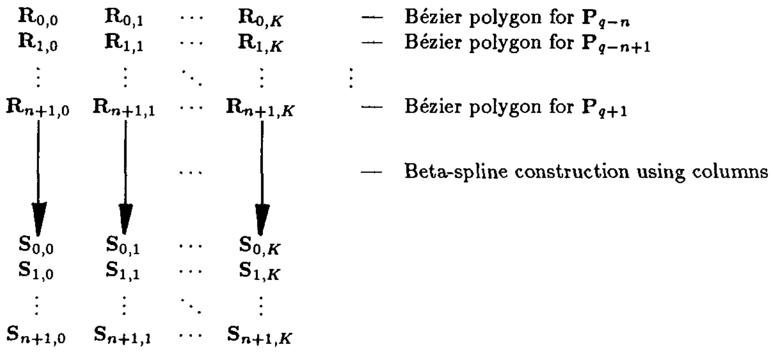
$\mathbf{R}_{0,0}$ $\mathbf{R}_{0,1}$ $\cdots$ $\mathbf{R}_{0,K}$ — Bézier polygon for $\mathbf{P}_{q-n}$
$\mathbf{R}_{1,0}$ $\mathbf{R}_{1,1}$ $\cdots$ $\mathbf{R}_{1,K}$ — Bézier polygon for $\mathbf{P}_{q-n+1}$
$\mathbf{R}_{n+1,0}$ $\mathbf{R}_{n+1,1}$ $\cdots$ $\mathbf{R}_{n+1,K}$ — Bézier polygon for $\mathbf{P}_{q+1}$
— Beta-spline construction using columns
$\mathbf{S}_{0,0}$ $\mathbf{S}_{0,1}$ $\cdots$ $\mathbf{S}_{0,K}$
$\mathbf{S}_{1,0}$ $\mathbf{S}_{1,1}$ $\cdots$ $\mathbf{S}_{1,K}$
$\mathbf{S}_{n+1,0}$ $\mathbf{S}_{n+1,1}$ $\cdots$ $\mathbf{S}_{n+1,K}$

Fig. 16. Schematic representation of the Bézier polygon construction algorithm.

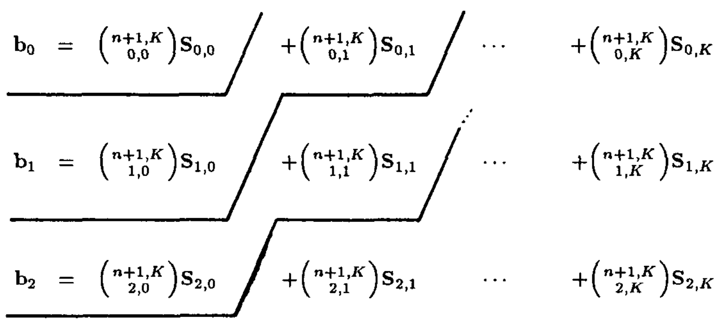
$\mathbf{b}_0 = \binom{n+1,K}{0,0}\mathbf{S}_{0,0} + \binom{n+1,K}{0,1}\mathbf{S}_{0,1} \cdots + \binom{n+1,K}{0,K}\mathbf{S}_{0,K}$
$\mathbf{b}_1 = \binom{n+1,K}{1,0}\mathbf{S}_{1,0} + \binom{n+1,K}{1,1}\mathbf{S}_{1,1} \cdots + \binom{n+1,K}{1,K}\mathbf{S}_{1,K}$
$\mathbf{b}_2 = \binom{n+1,K}{2,0}\mathbf{S}_{2,0} + \binom{n+1,K}{2,1}\mathbf{S}_{2,1} \cdots + \binom{n+1,K}{2,K}\mathbf{S}_{2,K}$

Fig. 17. Summation over skew diagonals.

(3) Finally, the Bézier polygon $\mathbf{b}_0, \ldots, \mathbf{b}_{K+n+1}$ for the Catmull–Rom segment is obtained from the $\mathbf{S}$-array by weighting $\mathbf{S}_{i,j}$ by the scalar coefficient $\binom{n+1,K}{i,j}$, then summing over the skew diagonals of the weighted $\mathbf{S}$-array, as shown in Figure 17. Specifically, $\mathbf{b}_l$ is obtained by summing over the lth skew diagonal (see Figure 17).

In the next section we demonstrate this general algorithm for $n = 1$ and $k = 1$, and for $n = 2$ and $k = 2$, that is, for the $(G^1, k = 1)$ and $(G^2, k = 2)$ splines of Table I.

7. EXAMPLES

To demonstrate the diversity of splines contained in the class of geometrically continuous Catmull–Rom curves we refer again to Table I. The splines in the first row (the $k = 0$ row) are the Beta-spline curves. Also of practical interest are the splines along the $k = n$ subdiagonal in that they are the lowest degree interpolating members of the class. Of these, the $(G^1, k = 1)$ and $(G^2, k = 2)$ splines are likely to be the most practical; they are of polynomial degree 3 and 5, respectively. In this section we study these splines more thoroughly by refining the Bézier polygon algorithm of Section 6 and by empirically investigating the effect of varying shape parameters.

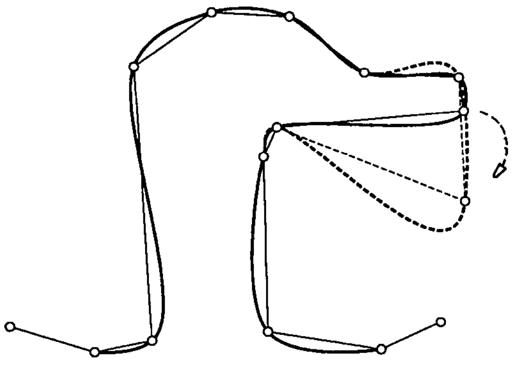

Fig. 18. Movement of a control vertex in a (G^1, $k = 1$) spline.

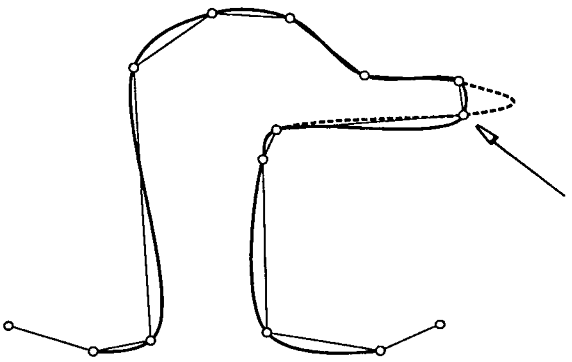

Fig. 19. Modification of a shape parameter in a (G^1, $k = 1$) spline.

7.1 The (G^1, $k = 1$) Spline

The (G^1, $k = 1$) spline of Table I is a cubic interpolating spline possessing one shape parameter per *interior control vertex*. That is, a (G^1, $k = 1$) spline with a control polygon $\mathbf{V}_0, \ldots, \mathbf{V}_m$ will have $m - 1$ shape parameters labeled $\beta 1_1, \ldots, \beta 1_{m-1}$. This spline has local control with respect to both vertex movement and shape parameter modification. However, the region of the curve that is affected differs depending on what kind of change is made. In particular, perturbation of a control vertex $\mathbf{V}_q$ affects four segments of the curve, $\mathbf{f}_{q-2}$, $\mathbf{f}_{q-1}$, $\mathbf{f}_q$, $\mathbf{f}_{q+1}$ (see Figure 18), whereas perturbation of a shape parameter $\beta 1_q$ affects only two segments, $\mathbf{f}_{q-1}$ and $\mathbf{f}_q$ (see Figure 19).

It should be mentioned that the (G^1, $k = 1$) spline is an alternate representation of an earlier spline in CAGD. It is, in a sense that will be elaborated on momentarily, equivalent to the standard C^1 cubic Catmull–Rom spline with nonuniformly spaced parametric breakpoints. That is, a C^1 cubic Catmull–Rom spline with nonuniformly spaced parametric breakpoints can be viewed as a G^1 cubic Catmull–Rom spline with uniformly spaced parametric breakpoints. If one were to draw a C^1 cubic Catmull–Rom spline using the breakpoints $u_0, \ldots, u_m$, the same curve could be reproduced by appropriately choosing the shape parameters $\beta 1_i$. In particular, it can be verified by direct calculation that the necessary choice for each $\beta 1_i$ is

$$\beta 1_i = \frac{u_{i+1} - u_i}{u_i - u_{i-1}}, \qquad i = 1, \ldots, m-1.$$

That is, $\beta 1_i$ measures the relative lengths of the parametric intervals $[u_{i-1}, u_i]$ and $[u_i, u_{i+1}]$.

The converse is also true: given a G^1 cubic Catmull–Rom spline with some assignment of the $\beta 1$'s, it is possible to find parametric breakpoints such that the G^1 curve is reproduced by a C^1 cubic Catmull–Rom spline.

Even in light of this equivalence, the (G^1, $k = 1$) spline is of interest for several reasons. First, from a user's standpoint, the use of shape parameters is frequently more natural than the specification of breakpoints. Second, the (G^1, $k = 1$) spline provides a good starting point for the study of the (G^2, $k = 2$) spline, which is presented in Section 7.2. It is important to note that, unlike the (G^1, $k = 1$) spline, the (G^2, $k = 2$) spline is *not* equivalent to the corresponding C^2 Catmull–Rom spline curve with nonuniformly spaced parametric breakpoints.

The construction of the Bézier polygons for the (G^1, $k = 1$) spline takes as input the vertices $\mathbf{V}_0, \ldots, \mathbf{V}_m$ and the shape parameters $\overline{\beta 1} = (\beta 1_1, \ldots, \beta 1_{m-1})$ and produces as output the Bézier polygons for the segments $\mathbf{f}_i(u)$, $i = 1, \ldots, m-2$. The Bézier polygon for the qth segment $\mathbf{f}_q(u)$ is constructed as follows:

Construction 7: The Bézier vertices for the qth *segment of a* (G^1, $k = 1$) *spline.*

1. Construct the Bézier polygons $(\mathbf{R}_{0,0}, \mathbf{R}_{0,1})$ for $\mathbf{P}_{q-1}(u)$, $(\mathbf{R}_{1,0}, \mathbf{R}_{1,1})$ for $\mathbf{P}_q(u)$, and $(\mathbf{R}_{2,0}, \mathbf{R}_{2,1})$ for $\mathbf{P}_{q+1}(u)$ (the only interpolating functions that contribute to $\mathbf{f}_q(u)$) using Construction 5 (see Figure 20a):

   a. $\mathbf{R}_{0,0} \leftarrow \mathbf{R}_{1,0} = \mathbf{V}_q$,

   b. $\mathbf{R}_{0,1} \leftarrow \mathbf{V}_q + \beta 1_q(\mathbf{V}_q - \mathbf{V}_{q-1})$,

   c. $\mathbf{R}_{1,1} \leftarrow \mathbf{R}_{2,1} = \mathbf{V}_{q+1}$,

   d. $\mathbf{R}_{2,0} \leftarrow \mathbf{V}_{q+1} + \dfrac{1}{\beta 1_{q+1}}(\mathbf{V}_{q+1} - \mathbf{V}_{q+2})$.

2. Construct the first column of the $\mathbf{S}$-array by blending the first column of the $\mathbf{R}$-array $(\mathbf{R}_{0,0}, \mathbf{R}_{1,0}, \mathbf{R}_{2,0})$ as a G^1 Beta-spline control polygon with shape parameters $\beta 1_q$ and $\beta 1_{q+1}$ as in Construction 3 (see Figure 20b):

   a. $\mathbf{S}_{0,0} \leftarrow \dfrac{\beta 1_q \mathbf{R}_{0,0} + \mathbf{R}_{1,0}}{1 + \beta 1_q} = \mathbf{V}_q \dfrac{1 + \beta 1_q}{1 + \beta 1_q} = \mathbf{V}_q$,

   b. $\mathbf{S}_{1,0} \leftarrow \mathbf{R}_{1,0} = \mathbf{V}_q$,

   c. $\mathbf{S}_{2,0} \leftarrow \dfrac{\beta 1_{q+1}\mathbf{R}_{1,0} + \mathbf{R}_{2,0}}{1 + \beta 1_{q+1}}$.

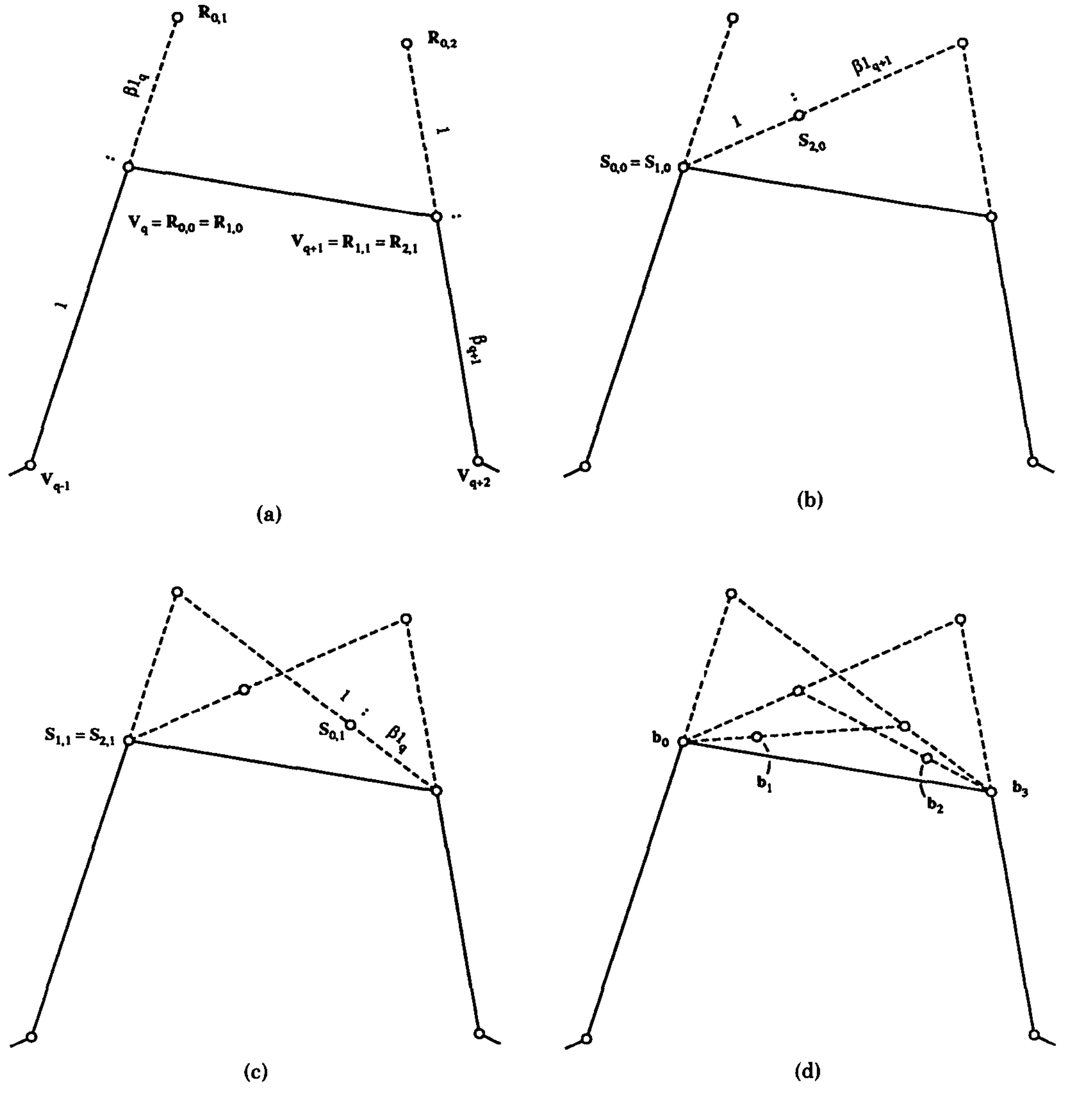

Fig. 20. The (G^1, $k = 1$) construction.

3. Construct the second column of the **S**-array using ($\mathbf{R}_{0,1}$, $\mathbf{R}_{1,1}$, $\mathbf{R}_{2,1}$) and $\beta 1_q$, $\beta 1_{q+1}$ in Construction 3 (see Figure 20c):

   a. $\mathbf{S}_{0,1} \leftarrow \dfrac{\beta 1_q \mathbf{R}_{0,1} + \mathbf{R}_{1,1}}{1 + \beta 1_q}$,

   b. $\mathbf{S}_{1,1} \leftarrow \mathbf{R}_{1,1} = \mathbf{V}_{q+1}$,

   c. $\mathbf{S}_{2,1} \leftarrow \dfrac{\beta 1_{q+1} \mathbf{R}_{1,1} + \mathbf{R}_{2,1}}{1 + \beta 1_{q+1}} = \mathbf{V}_{q+1} \dfrac{1 + \beta 1_{q+1}}{1 + \beta 1_{q+1}} = \mathbf{V}_{q+1}$.

4. Construct $\mathbf{b}_0, \ldots, \mathbf{b}_3$ from the S-array by summing over skew diagonals (see Figure 20d):

$$\begin{aligned} &\text{a.}\quad \mathbf{b}_0 \leftarrow \mathbf{S}_{0,0} = \mathbf{V}_q, \\ &\text{b.}\quad \mathbf{b}_1 \leftarrow \tfrac{2}{3}\mathbf{S}_{1,0} + \tfrac{1}{3}\mathbf{S}_{0,1}, \\ &\text{c.}\quad \mathbf{b}_2 \leftarrow \tfrac{1}{3}\mathbf{S}_{2,0} + \tfrac{2}{3}\mathbf{S}_{1,1}, \\ &\text{d.}\quad \mathbf{b}_3 \leftarrow \mathbf{S}_{2,1} = \mathbf{V}_{q+1}. \end{aligned}$$

By eliminating unnecessary computations this construction can be compressed into the following pseudocode:

```
procedure ComputeG1k1Bézier($\mathbf{V}_0, \ldots, \mathbf{V}_m, \overline{\beta 1}, q$)
  {Return the Bézier polygon for the qth segment}
  {of a ($G^1$, $k = 1$) Catmull-Rom spline}
  $\mathbf{R}_{0,1} \leftarrow \mathbf{V}_q + \beta 1_q(\mathbf{V}_q - \mathbf{V}_{q-1})$
  $\mathbf{R}_{2,0} \leftarrow \mathbf{V}_{q+1} + \frac{1}{\beta 1_{q+1}}(\mathbf{V}_{q+1} - \mathbf{V}_{q+2})$
  $\mathbf{S}_{0,1} \leftarrow \frac{\mathbf{V}_{q+1} + \beta 1_q \mathbf{R}_{0,1}}{1 + \beta 1_q}$
  $\mathbf{S}_{2,0} \leftarrow \frac{\beta 1_{q+1}\mathbf{V}_q + \mathbf{R}_{2,0}}{1 + \beta 1_{q+1}}$
  $\mathbf{b}_0 \leftarrow \mathbf{V}_q$
  $\mathbf{b}_1 \leftarrow \frac{2\mathbf{V}_q + \mathbf{S}_{0,1}}{3}$
  $\mathbf{b}_2 \leftarrow \frac{\mathbf{S}_{2,0} + 2\mathbf{V}_{q+1}}{3}$
  $\mathbf{b}_3 \leftarrow \mathbf{V}_{q+1}$
  return ($\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$)
end ComputeG1k1Bézier
```

A (G^1, $k = 1$) spline curve can be drawn by first computing the Bézier polygon for each segment q using the routine *ComputeG1k1Bézier*, then using standard techniques to draw the segments in Bézier form (cf. [10, 21, 22]). All (G^1, $k = 1$) figures in this section have been computed using *ComputeG1k1Bézier*, together with de Casteljau's algorithm [10] for computing points on a Bézier curve. Figure 21 has all shape parameters set to unity and is therefore equivalent to a cubic Catmull–Rom spline defined by uniformly spaced parametric breakpoints.

To demonstrate that the curve as a whole depends on shape parameters indexed from 1 to $m - 1$ (and not, e.g., from 0 to $m - 1$ as for G^1 Beta-splines), we can appeal to Figure 20a–d. These figures show that the qth segment of the Catmull–Rom curve depend only on the shape parameters $\beta 1_q$ and $\beta 1_{q+1}$. The entire curve, consisting of segments 1 through $m - 2$, therefore depends on the shape parameters indexed from 1 through $m - 1$.

Figures 18 and 19 demonstrate the property of local control with respect to control vertex movement and shape parameter modification, respectively. The curves of Figure 18 differ only in the position of the indicated vertex; the curves of Figure 19 differ only in the value of the shape parameter at the indicated vertex. Notice that four segments are affected by control vertex movement, whereas only two segments are affected by shape parameter modification.

Some of the irregularities in the curve of Figure 21 can be reduced by appropriately choosing the shape parameters. We currently have no rigorous analytic results to which we can appeal to provide automatic shape parameter settings for arbitrary control polygons. We do, however, have some heuristic rules based on empirical experience that we hope will lead to more analytic tools. Since $\beta 1$ measures the relative length of adjacent parametric intervals, we experimented with setting $\beta 1_i$ on the basis of the relative lengths of adjacent segments of the control polygon. In particular, we tried

$$\beta 1_i = \frac{\| \mathbf{V}_i \mathbf{V}_{i+1} \|}{\| \mathbf{V}_{i-1} \mathbf{V}_i \|}. \tag{7.1}$$

Compared with the curve corresponding to all $\beta 1$'s being set to unity, the curve produced by eq. (7.1) tended to overshoot what one might call the most desirable curve (see Figure 22). We therefore modified the heuristic to compute a shape parameter halfway between unity and the value produced by eq. (7.1). The improved heuristic is therefore

$$\beta 1_i = \frac{1}{2}\left(1 + \frac{\| \mathbf{V}_i \mathbf{V}_{i+1} \|}{\| \mathbf{V}_{i+1} \mathbf{V}_{i+2} \|}\right). \tag{7.2}$$

The effect of this heuristic on the curve of Figure 21 is shown in Figure 23.

7.2 The (G^2, $k = 2$) Spline

The (G^2, $k = 2$) spline of Table I is a quintic (degree 5) interpolating spline possessing two shape parameters per interior control vertex. Thus, given a (G^2, $k = 2$) spline with control polygon $\mathbf{V}_0, \ldots, \mathbf{V}_m$, there are $2(m - 1)$ shape parameters; the justification for this number of shape parameters is presented later in this section. As with the (G^1, $k = 1$) spline, the (G^2, $k = 2$) spline possesses local control with respect to control vertex change and shape parameter modification. More specifically, perturbation of a control vertex $\mathbf{V}_q$ affects six segments of the curve $\mathbf{f}_{q-3}$, $\mathbf{f}_{q-2}$, $\mathbf{f}_{q-1}$, $\mathbf{f}_q$, $\mathbf{f}_{q+1}$ (see Figure 24), whereas modification of a shape parameter $\beta 1_q$ or $\beta 2_q$ affects only four segments of the curve $\mathbf{f}_{q-2}$, $\mathbf{f}_{q-1}$, $\mathbf{f}_q$, $\mathbf{f}_{q+1}$ (see Figure 25).

The construction of the Bézier polygons for the (G^2, $k = 2$) spline takes as input the vertices $\mathbf{V}_0, \ldots, \mathbf{V}_m$ and the shape parameters $\overline{\beta 1} = (\beta 1_1, \ldots, \beta 1_{m-1})$ and $\overline{\beta 2} = (\beta 2_1, \ldots, \beta 2_{m-1})$. The following Bézier polygon algorithm follows from Constructions 4 and 6 in much the same way that the routine *ComputeG1k1Bézier* followed from Constructions 3 and 5.

One subtlety arises in the construction of the Bézier polygons for the (G^2, $k = 2$) spline. This did not appear for the (G^1, $k = 1$) spline in the previous section since the subtlety is introduced in the handling of $\beta 2$. It was shown in Section 5.1 that the interpolating functions $\mathbf{P}_i(u)$ and the blending functions $W_i(u)$ must satisfy the Beta-constraints with respect to the same shape parameters. That is, if $\mathbf{P}_i(u)$ satisfies the Beta-constraints for some choice

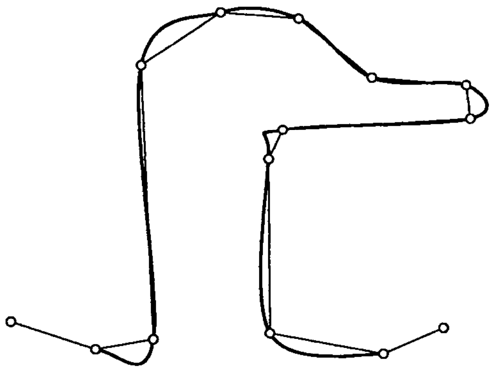
Fig. 21. A (G^1, k = 1) spline with $\beta 1_i$ set uniformly to unity.

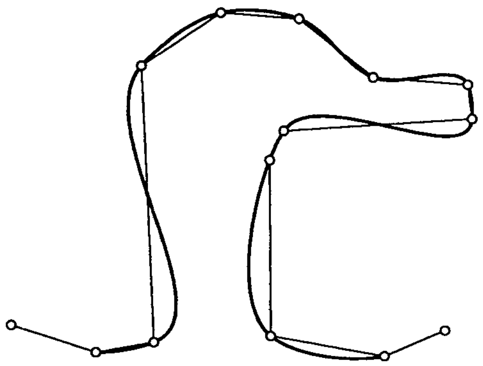
Fig. 22. Shape parameters set using the heuristic of eq. (7.1).

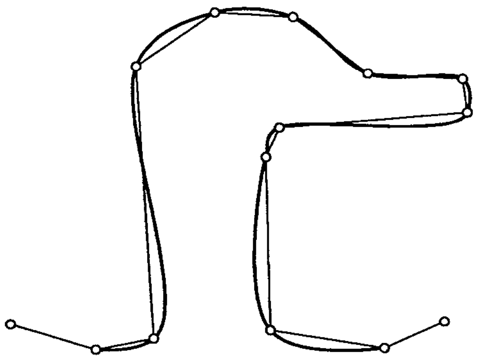
Fig. 23. Shape parameters set using the heuristic of eq. (7.2).

of $\beta 1, \ldots, \beta n$, at a joint, then $W_i(u)$ must satisfy the Beta-constraints for the same choice of β's at that joint.

In the G^2 constructions, that is, Constructions 4 and 6, it is γ, not $\beta 2$, that enters directly into the computation of the Bézier polygons. Notice that the quantity γ as defined in Construction 2 depends not only on $\beta 1$ and $\beta 2$, but also on the degree d of the equivalent Bézier curve. Since the degree of the G^2 interpolating functions from Construction 6 differs from the degree of

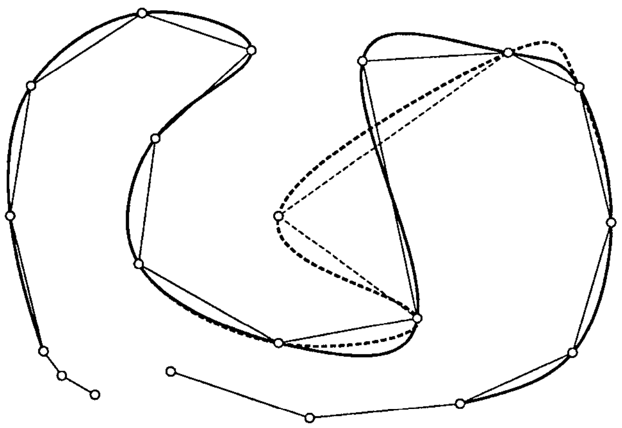
Fig. 24. Movement of a vertex in a (G^2, $k = 2$) spline.

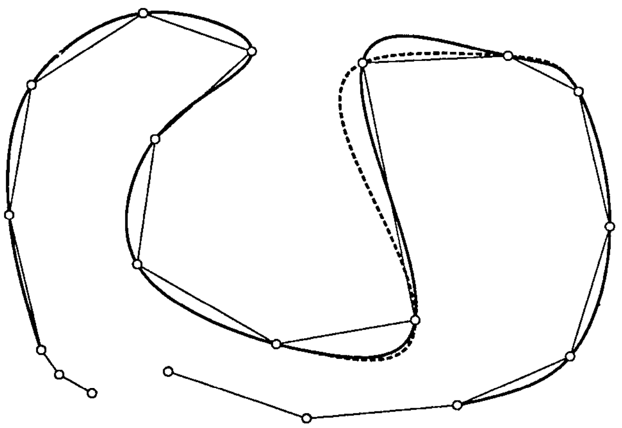
Fig. 25. Modification of a shape parameter in a (G^2, $k = 2$) spline.

the G^2 Beta-splines, two different γ's are needed: one for $d = 2$ and one for $d = 3$. In the algorithm to follow, the variables $\gamma 2_i$ and $\gamma 3_i$ represent the ith γ for $d = 2$ and $d = 3$, respectively.

One other preliminary comment is in order before presenting the (G^2, $k = 2$) construction algorithm. On the basis of the assumption that the Bézier polygons for all segments are to be computed, the following algorithm eliminates some redundant computations by precomputing $\gamma 2_i$, $\gamma 3_i$, and the Bézier polygons for each of the interpolating functions $\mathbf{P}_i(u)$ using Construction 6. In the following pseudocode routine, the points $\mathbf{Pb}_{i,j,0}$, $\mathbf{Pb}_{i,j,1}$, and $\mathbf{Pb}_{i,j,2}$ form the Bézier polygon for $\mathbf{p}_{i,j}(u)$, that is, the jth segment of $\mathbf{P}_i(u)$ (see Figure 13). Note that some unnecessary work is done in that the segments $\mathbf{p}_{0,0}$, $\mathbf{p}_{0,1}$, $\mathbf{p}_{0,2}$, $\mathbf{p}_{1,0}$, $\mathbf{p}_{1,1}$, $\mathbf{p}_{2,0}$ are computed but never needed, as are the segments $\mathbf{p}_{m-4,3}$, $\mathbf{p}_{m-3,2}$,

$\mathbf{p}_{m-3,3}$, $\mathbf{p}_{m-2,1}$, $\mathbf{p}_{m-2,2}$, $\mathbf{p}_{m-2,3}$. We have chosen to do this unnecessary calculation to simplify the pseudocode presentation of the algorithm. Removal of the unnecessary work is straightforward and should definitely be done in production implementations.

```
procedure Precompute($\mathbf{V}_0, \ldots, \mathbf{V}_m, \overline{\beta 1}, \overline{\beta 2}$)
{Precompute $\gamma 2$'s, $\gamma 3$'s and the Bézier polygons of each}
{segment $\mathbf{p}_{i,j}(u)$ of each interpolating function $\mathbf{P}_i(u)$}
begin
  $\beta 1_0 \leftarrow \beta 1_m \leftarrow 1$
  $\beta 2_0 \leftarrow \beta 2_m \leftarrow 0$
  for $i \leftarrow 1$ to $m - 1$ do
    $\gamma 2_i \leftarrow \dfrac{1 + \beta 1_i}{\beta 2_i + \beta 1_i(1 + \beta 1_i)}$
    $\gamma 3_i \leftarrow \dfrac{2(1 + \beta 1_i)}{\beta 2_i + 2\beta 1_i(1 + \beta 1_i)}$
  end
  for $i \leftarrow 0$ to $m - 2$ do
    $\mathbf{Pb}_{i,0,2} \leftarrow \mathbf{Pb}_{i,1,0} \leftarrow \mathbf{V}_i$
    $\mathbf{Pb}_{i,1,2} \leftarrow \mathbf{Pb}_{i,2,0} \leftarrow \mathbf{V}_{i+1}$
    $\mathbf{Pb}_{i,2,2} \leftarrow \mathbf{Pb}_{i,3,0} \leftarrow \mathbf{V}_{i+2}$
    $\mathbf{Pb}_{i,1,1} \leftarrow \dfrac{\beta 1_{i+1}^2 \gamma 2_{i+1} \mathbf{V}_i + (1 + \gamma 2_{i+1})(1 + \beta 1_{i+1})\mathbf{V}_{i+1} - \gamma 2_{i+1} \mathbf{V}_{i+2}}{(1 + \beta 1_{i+1})(1 + \beta 1_{i+1} \gamma 2_{i+1})}$
    $\mathbf{Pb}_{i,2,1} \leftarrow \mathbf{V}_{i+1} + \beta 1_{i+1}(\mathbf{V}_{i+1} - \mathbf{Pb}_{i,1,1})$
    $\mathbf{Pb}_{i,0,1} \leftarrow \mathbf{V}_i + \dfrac{1}{\beta 1_i}(\mathbf{V}_i - \mathbf{Pb}_{i,1,1})$
    $\mathbf{T}_1 \leftarrow \mathbf{Pb}_{i,1,1} + \gamma 2_i(\mathbf{Pb}_{i,1,1} - \mathbf{V}_{i+1})$
    $\mathbf{Pb}_{i,0,0} \leftarrow \mathbf{Pb}_{i,0,1} + \dfrac{1}{\beta 1_i^2 \gamma 2_i}(\mathbf{Pb}_{i,0,1} - \mathbf{T}_1)$
    $\mathbf{Pb}_{i,3,1} \leftarrow \mathbf{V}_{i+2} + \beta 1_{i+2}(\mathbf{V}_{i+2} - \mathbf{Pb}_{i,2,1})$
    $\mathbf{T}_2 \leftarrow \mathbf{Pb}_{i,2,1} + \beta 1_{i+2}^2 \gamma 2_{i+2}(\mathbf{Pb}_{i,2,1} - \mathbf{V}_{i+1})$
    $\mathbf{Pb}_{i,3,2} \leftarrow \mathbf{Pb}_{i,3,1} + \dfrac{1}{\gamma 2_{i+2}}(\mathbf{Pb}_{i,3,1} - \mathbf{T}_2)$
  end
end Precompute
```

The Bézier polygon of $\mathbf{f}_q(u)$, $q = 2, \ldots, m - 3$, can now be computed as follows:

```
procedure ComputeG2k2Bézier($\mathbf{V}_0, \ldots, \mathbf{V}_m, \overline{\beta 1}, \overline{\beta 2}, q$)
{Return the Bézier polygon for the $q$th segment of a ($G^2$, $k = 2$) Catmull–Rom spline}
  {Construct the R-array from the precomputed Pb's}
  for $r \leftarrow 0$ to 3 do
    {Construct row $r$ of the R-array}
    $\mathbf{R}_{r,0} \leftarrow \mathbf{Pb}_{q+r-2,3-r,0}$
    $\mathbf{R}_{r,1} \leftarrow \mathbf{Pb}_{q+r-2,3-r,1}$
    $\mathbf{R}_{r,2} \leftarrow \mathbf{Pb}_{q+r-2,3-r,2}$
end
```

Table III

Segment	Shape parameter dependence
$\mathbf{p}_{q-2,3}(u)$	$\beta 1_{q-1}, \beta 1_q, \beta 2_{q-1}, \beta 2_q$
$\mathbf{p}_{q-1,2}(u)$	$\beta 1_q, \beta 2_q$
$\mathbf{p}_{q,1}(u)$	$\beta 1_{q+1}, \beta 2_{q+1}$
$\mathbf{p}_{q+1,0}(u)$	$\beta 1_{q+1}, \beta 1_{q+2}, \beta 2_{q+1}, \beta 2_{q+2}$

```
    {Construct the S-array using Construction 4}
    for $c \leftarrow 0$ to 2 do
      {construct column c of the S-array}
      $\mathbf{T}_1 \leftarrow \dfrac{\beta 1_q^2 \gamma 3_q \mathbf{R}_{0,c} + (1 + \gamma 3_{q-1})\mathbf{R}_{1,c}}{1 + \gamma 3_{q-1} + \beta 1_q^2 \gamma 3_q}$
      $\mathbf{S}_{1,c} \leftarrow \dfrac{(1 + \beta 1_{q+1}^2 \gamma 3_{q+1})\mathbf{R}_{1,c} + \gamma 3_q \mathbf{R}_{2,c}}{1 + \gamma 3_q + \beta 1_{q+1}^2 \gamma 3_{q+1}}$
      $\mathbf{S}_{2,c} \leftarrow \dfrac{\beta 1_{q+1}^2 \gamma 3_{q+1} \mathbf{R}_{1,c} + (1 + \gamma 3_q)\mathbf{R}_{2,c}}{1 + \gamma 3_q + \beta 1_{q+1}^2 \gamma 3_{q+1}}$
      $\mathbf{T}_2 \leftarrow \dfrac{(1 + \beta 1_{q+2}^2 \gamma 3_{q+2})\mathbf{R}_{2,c} + \gamma 3_{q+1} \mathbf{R}_{3,c}}{1 + \gamma 3_{q+1} + \beta 1_{q+2}^2 \gamma 3_{q+2}}$
      $\mathbf{S}_{0,c} \leftarrow \dfrac{\beta 1_q \mathbf{T}_1 + \mathbf{S}_{1,c}}{1 + \beta 1_q}$
      $\mathbf{S}_{3,c} \leftarrow \dfrac{\beta 1_{q+1} \mathbf{S}_{2,c} + \mathbf{T}_2}{1 + \beta 1_{q+1}}$
    end

    {Compute the Bézier polygon from the S-array}
    $\mathbf{b}_0 \leftarrow \mathbf{S}_{0,0}$
    $\mathbf{b}_1 \leftarrow \dfrac{3\mathbf{S}_{1,0} + 2\mathbf{S}_{0,1}}{5}$
    $\mathbf{b}_2 \leftarrow \dfrac{3\mathbf{S}_{2,0} + 6\mathbf{S}_{1,1} + \mathbf{S}_{0,2}}{10}$
    $\mathbf{b}_3 \leftarrow \dfrac{\mathbf{S}_{3,0} + 6\mathbf{S}_{2,1} + 3\mathbf{S}_{1,2}}{10}$
    $\mathbf{b}_4 \leftarrow \dfrac{2\mathbf{S}_{3,1} + 3\mathbf{S}_{2,2}}{5}$
    $\mathbf{b}_5 \leftarrow \mathbf{S}_{3,2}$
    return($\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4, \mathbf{b}_5$)
end ComputeG2k2Bézier
```

The only shape parameters that affect the curve are those indexed from 1 to $m - 1$. To see this, we examine further the construction algorithm for the Bézier polygons of the segments of the (G^2, $k = 2$) spline. The qth segment depends on $\mathbf{p}_{q-2,3}(u)$, $\mathbf{p}_{q-1,2}(u)$, $\mathbf{p}_{q,1}(u)$, and $\mathbf{p}_{q+1,0}(u)$. The dependence of each of these segments on the shape parameters is shown in Table III. These segments are blended together using Beta-spline blending that depends on the shape parameters $\beta 1_{q-1}, \ldots, \beta 1_{q+2}$ and $\beta 2_{q-1}, \ldots, \beta 2_{q+2}$. Combining these dependencies shows that the qth segment depends on the shape parameters indexed from $q - 1$ to $q + 2$. The entire curve, consisting of segments 2 through $m - 3$, therefore depends on the shape parameters indexed from 1 through $m - 1$.

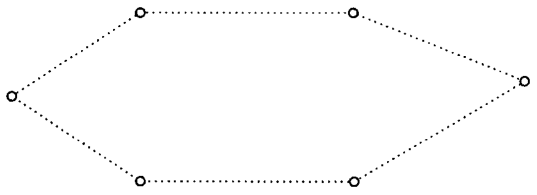
Fig. 26. Control polygon for a rough football.

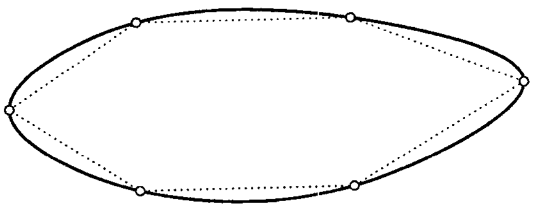
Fig. 27. Control polygon and the corresponding (G^2, $k = 2$) spline.

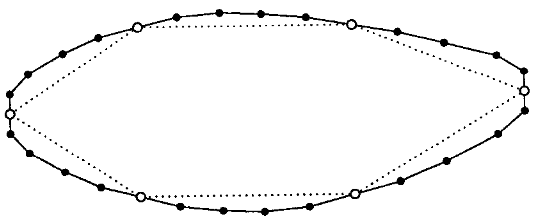
Fig. 28. The Bézier polygons for the curve of Figure 27.

To demonstrate the flexibility of the (G^2, $k = 2$) spline, consider the design of the outline of an icon in a typography system. A rough outline of the icon (a football) is first specified by the designer, as shown in Figure 26. The system provides periodic boundary conditions and default shape parameter settings, in this case $\beta 1_i = 1$ and $\beta 2_i = 0$ for all i, then generates and renders the (G^2, $k = 2$) spline segments defined by this input using the routines *Precompute* and *ComputeG2k2Bézier*. The resulting curve, together with its control polygon, is shown in Figure 27; for illustrative purposes the Bézier polygons are shown in Figure 28.

Since the shape in Figure 27 is not quite symmetrical, the control vertex at the far right can be moved into a more appropriate position. The system then responds with the new curve, as shown in Figure 29. Notice that only the portion of the curve near the modified vertex is perturbed. To further refine the shape, the ends of the football should be made "more pointed." This could be done by adding new vertices at the far left and far right; however, it is easier to change the value of $\beta 2$. Figure 30 shows the curve of Figure 29 superimposed with the curve (shown dashed) generated when the value of $\beta 2$ at the far right vertex is increased to 3. Notice that $\beta 2$ behaves in a "tensionlike" manner, producing much the same effect as $\beta 2$ in a G^2 cubic Beta-spline. For this reason we again call $\beta 2$ a *tension* parameter. Increasing the value of tension to 36 produces the heavily dotted curve shown in Figure 30. The design is completed by increasing the tension of the curve to 36 at the far left vertex, as shown in Figure 31.

The sample design session also seems to suggest that $\beta 2$ in a (G^2, $k = 2$) spline behaves just as it would for a G^2 Beta-spline. Although this is partially true,

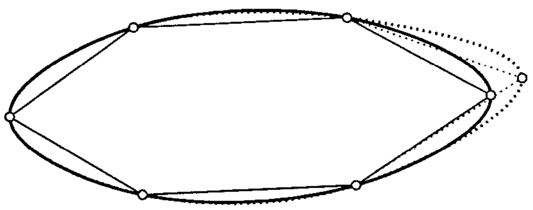
Fig. 29. Movement of a control vertex.

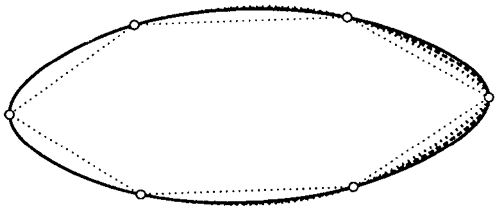
Fig. 30. Modification of $\beta 2$ at the far right vertex.

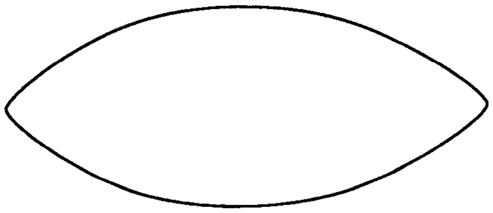
Fig. 31. The completed football.

there is one difference that should be pointed out. In a G^2 Beta-spline, if $\beta 2_i$ is taken to the limiting value of infinity for all i, the curve is guaranteed to converge its control polygon. This is not true of a (G^2, $k = 2$) spline. In the limit of infinite $\beta 2$, the curve is guaranteed to converge, but not necessarily to the defining control polygon. We currently do not have a good explanation of this behavior.

8. CONCLUSION

We have introduced a subclass of the Catmull–Rom splines that possesses geometric, rather than parametric, continuity. The replacement of parametric continuity with the less restrictive geometric analog allows the introduction of shape parameters that can be used to modify the shape of the spline without moving the control vertices. For nth-order geometric continuity there are n shape parameters per joint, which can be varied independently to control the shape of the curve. In addition to shape parameters, members of the class have local control. Some of the splines in the class interpolate the control vertices, whereas others approximate them (see Table I).

The class results from the combination of Beta-spline blending functions and a set of geometrically continuous functions related to the classical Lagrange curves, and is a proper generalization of the class of parametrically continuous Catmull–Rom splines. Moreover, the class includes the G^1 and G^2 Beta-splines, which are local, approximating, polynomial splines with shape parameters.

The evaluation and rendering of geometrically continuous Catmull–Rom splines is made practical by general algorithms that construct the Bézier control polygons for each of the segments of the spline. To further aid implementors of these techniques, the general algorithm has been further refined for the lowest degree interpolating members of the class, that is, the cubic (G^1, $k = 1$) spline and the quintic (G^2, $k = 2$) spline.

The sample design session in Section 7.2 brings to light the important properties of the quintic (G^2, $k = 2$) spline: interpolation of the control vertices, shape parameters, local control with respect to vertex movement, local control with respect to shape parameter modification, G^2 continuity, and relatively efficient computation. Previous interpolating splines either had shape parameters, but were global representations [2, 12, 23–26], or were local with no shape parameters [11]. Moreover, the shape parameter $\beta 2$ exhibits tensionlike behavior and is locally variable. These properties render the (G^2, $k = 2$) spline potentially useful for rapid free-form design.

During the course of this research a number of additional questions have been raised, some of which have been posed in previous sections. Here we summarize a more complete list of open questions:

—Do G^n Lagrange curves exist for all n? If so, what is their degree and do they have a simple closed form?

—What is the degree of a geometrically continuous Catmull–Rom spline for arbitrary n and k?

—What is the general construction algorithm for arbitrary n and k?

—Is there a fast way to modify (instead of completely recomputing) the Bézier control vertices when a control vertex or shape parameter is perturbed?

—Is there a general explanation for the behavior exhibited in the limit of infinite shape parameters?

—Do urn model descriptions [19] exist for geometrically continuous Catmull–Rom splines? We know that for (G^1, $k = 1$) this question is answered in the affirmative.

—Finally, can the techniques used in this work for curves be extended to tensor product surfaces and/or triangular patch surfaces? What is the behavior of such surfaces when shape parameters are changed? What is the nature of construction algorithms for determining the Bézier control meshes?

REFERENCES

1. BARSKY, B. A. The Beta-spline: A local representation based on shape parameters and fundamental geometric measures. Ph.D. dissertation, Dept. of Computer Science, Univ. of Utah, Salt Lake City, Utah, Dec. 1981.
2. BARSKY, B. A. Exponential and polynomial methods for applying tension to an interpolating spline curve. *Comput. Vision Graph. and Image Process. 27*, 1 (July 1984), 1–18.
3. BARSKY, B. A. *Computer Graphics and Geometric Modeling Using Beta-Splines.* Springer-Verlag, Heidelberg. To appear.
4. BARSKY, B. A., AND BEATTY, J. C. Local control of bias and tension in Beta-splines. *ACM Trans. Graph. 2*, 2 (Apr. 1983), 109–134. Also published in *SIGGRAPH '83 Conf. Proc. 17*, 3 (July 1983), 193–218.
5. BARSKY, B. A., AND DEROSE, T. D. Geometric continuity for parametric curves. Tech. Rep. UCB/CSD 84/205, Computer Science Div., Electrical Engineering and Computer Sciences Dept., Univ. of California, Berkeley, California, Oct. 1984.

6. Barsky, B. A., and DeRose, T. D. The Beta2-spline: A special case of the Beta-spline curve and surface technique. *IEEE Comput. Graph. Appl. 5*, 9 (Sept. 1985), 46–58. Errata in *IEEE Comput. Graph. Appl. 7*, 3 (Mar. 1987), 15. Earlier version published as Tech. Rep. UCB/CSD 83/152, Computer Science Div., Electrical Engineering and Computer Sciences Dept., Univ. of California, Berkeley, California, Nov. 1983.
7. Bartels, R. H., Beatty, J. C., and Barsky, B. A. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling.* Morgan-Kaufmann, Los Altos, Calif., 1987.
8. Bézier, P. E. Mathematical and practical possibilities of Unisurf. In *Computer Aided Geometric Design,* R. E. Barnhill and R. F. Riesenfeld, Eds. Academic Press, Orlando, 1974, pp. 127–152.
9. Boehm, W. Curvature continuous curves and surfaces. *Comput. Aided Geom. Des. 2*, 4 (Dec. 1985), 313–323.
10. Boehm, W., Farin, G., and Kahmann, J. A survey of curve and surface methods in CAGD. *Comput. Aided Geom. Des. 1*, 1 (July 1984), 1–60.
11. Catmull, E. E., and Rom, R. J. A class of local interpolating splines. In *Computer Aided Geometric Design,* R. E. Barnhill and R. F. Riesenfeld, Eds. Academic Press, Orlando, 1974, pp. 317–326.
12. Cline, A. K. Scalar- and planar-valued curve fitting using splines under tension. *Commun. ACM 17*, 4 (Apr. 1974), 218–220.
13. DeRose, T. D. Geometric continuity: A parameterization independent measure of continuity for computer aided geometric design. Ph.D. dissertation, Univ. of California, Berkeley, California, Aug. 1985. Available as Tech. Rep. UCB/CSD 86/255, Computer Science Div., Dept. of Electrical Engineering and Computer Sciences, Univ. of California, Berkeley.
14. DeRose, T. D., and Barsky, B. A. An intuitive approach to geometric continuity for parametric curves and surfaces. In *Proceedings of Graphics Interface '85* (Montreal, Quebec, May 27–31, 1985), pp. 343–351. Revised version published in *Computer-Generated Images—The State of the Art,* N. Magnenat-Thalmann and D. Thalmann, Eds. Springer-Verlag, 1985, Berlin, pp. 159–175. Extended abstract in *Proceedings of the International Conference on Computational Geometry and Computer-Aided Design* (New Orleans, La., June 5–8, 1985), pp. 71–75.
15. Dyn, N., and Micchelli, C. A. Piecewise polynomial spaces and geometric continuity of curves. IBM Res. Rep., Mathematical Sciences Dept., IBM T.J. Watson Research Center, Yorktown Heights, N.Y., 1985.
16. Farin, G. Visually C^2 cubic splines. *Comput.-Aided Des. 14*, 3 (May 1982), 137–139.
17. Fateman, R. J. Addendum to the MACSYMA Reference Manual for the VAX. Computer Science Div., Univ. of California, Berkeley, 1982.
18. Fournier, A., and Barsky, B. A. Geometric continuity with interpolating Bézier curves (extended summary), pp. 337–341. In *Proceedings of Graphics Interface '85* (Montreal, Quebec, May 27–31, 1985). Revised version published in *Computer-Generated Images—The State of the Art,* N. Magnenat-Thalmann and D. Thalmann, Eds. Springer-Verlag, Berlin 1985, pp. 153–158.
19. Goldman, R. N. An urnful of blending functions. *IEEE Comput. Graph. Appl. 3*, 7 (July 1983), 49–54.
20. Goodman, T. N. T. Properties of β-splines. *J. Approx. Theory 44*, 2 (June 1985), 132–153.
21. Lane, J. M., and Riesenfeld, R. F. A theoretical development for the computer generation and display of piecewise polynomial surfaces. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-2*, 1 (Jan. 1980), 35–46.
22. Lane, J. M., Carpenter, L. C., Whitted, J. T., and Blinn, J. F. Scan line methods for displaying parametrically defined surfaces. *Commun. ACM 23*, 1 (Jan. 1980), 23–34.
23. Nielson, G. M. Some piecewise polynomial alternatives to splines under tension. In *Computer Aided Geometric Design,* R. E. Barnhill and R. F. Riesenfeld, Eds. Academic Press, New York, 1974, pp. 209–235.
24. Nielson, G. M. Rectangular ν-splines. *IEEE Comput. Graph. Appl. 6*, 2 (Feb. 1986), 35–40.
25. Pilcher, D. T. Smooth approximation of parametric curves and surfaces. Ph.D. dissertation, Dept. of Computer Science, Univ. of Utah, Salt Lake City, Utah, Aug. 1973.
26. Schweikert, D. G. An interpolation curve using a spline in tension. *J. Math. Phys. 45* (1966), 312–317.

Received September 1984; revised May 1987; final revision accepted November 1987