A New Local Basis for Designing with Tensioned Splines*

by

Elaine Cohen
Computer Science, University of Utah

UUCS Tech Report: 85-104
June 1985

* This work was supported in part by DARPA (DAAK1184K0017) and the National Science Foundation (MCS-8203692 and MCS-8121750). All opinions, findings, conclusions or recommendations expressed in this document are those of the author and do not necessarily reflect the views of the sponsoring agencies.

Table of Contents

A New Local Basis for Designing with Tensioned Splines

by
Elaine Cohen
Computer Science
University of Utah

Abstract

Recently there has been a great deal of interest in the use of "tension" parameters to augment control mesh vertices as design handles for piecewise polynomials. A particular local cubic basis called β-splines, which has been termed a "generalization of B-splines", has been proposed as an appropriate basis. These functions are defined only for floating knot sequences. This paper uses the known property of B-splines that with appropriate knot vectors they span what are called here spaces of tensioned splines, and that particular combinations of them, called LT-splines, form bases for the spaces of tensioned splines. In addition, this paper shows that these new proposed bases have the variation diminishing property, the convex hull property, and straightforward knot insertion algorithms, and that both curves and individual basis functions can be easily computed. Sometimes it is desirable to interpolate points and also use these tension parameters so interpolation methods using the LT-spline bases are presented. Finally, the above properties are established for uniform and nonuniform knot vectors, open and floating end conditions, and homogeneous and nonhomogeneous tension parameter pairs.

Key Words: CAGD, B-splines, ν-splines, β-splines, knot insertion, algorithms, convex hull property, variation diminishing property, geometric continuity, visual continuity
Computing Reviews Classification: I.3.5 Computational Geometry and Object Modelling

1. Introduction

Parametric functions are widely used and have some advantages as well as disadvantages over explicit and implicit functions [6]. Further, parametric piecewise polynomials (parametric splines) are used in design and to solve interpolation problems since they have more inherent flexibility than single polynomials and allow the user to avail himself of a richer family of curves.

Frequently second derivative continuity of the designed parametric spline curve is stated as a requirement. For explicit curves, second derivative continuity is directly related to curvature continuity; however, this is not always the case for parametric curves in general, and parametric splines in particular. Sometimes, a particular parametrization of a curve will not be second derivative continuous at certain points, but the curve will be curvature continuous. Recall that the curvature of a curve is simply the second derivative of a curve with respect to its arc length parametrization. Hence, if a curve γ is parametrized with respect to some arbitrary parametrization t, we know that the curve also has an arc length parametrization s such that $t = t(s)$. Thus, a curve $\gamma(t)$ is curvature continuous if $\frac{d^2\gamma(t)}{ds^2}$ exists and is continuous everywhere. The analogous result is true for the tangent vectors. For example, if the straight line, α, between two points P_1 and P_2 is arc length parametrized, its form is $\alpha(s) = P_1 + s\,(P_2 - P_1)/\| P_2 - P_1 \|$. We may parametrize it differently. Let f(t) and g(t) be two twice differentiable functions such that $0 \le f(t) \le 1/2$, for $t \in [0,1]$, and $1/2 \le g(t) \le 1$, for $t \in [1, 5]$, with $f(0) = 0$, $f(1) = 1/2$, $g(1) = 1/2$, and $g(5) = 1$.

$$\gamma(t) = \begin{cases} (1-f(t))P_1 + f(t)\, P_2,\ t \in [0,1) \\ (1-g(t))P_1 + g(t)\, P_2,\ t \in [1,5] \end{cases}$$

Now, $\gamma(t)$ is another parametrization of the line segment between P_1 and P_2. However, γ' is not necessarily continuous at t = 1. $\gamma'(1-) = f'(1)(P_2 - P_1)$, but $\gamma'(1+) = \gamma'(1)(P_2 - P_1)$. If $f(t) = (1/2)t$ and $g(t) = (t + 3)/8$, then the $\gamma'(1-) = (1/2)(P_2 - P_1)$, but $\gamma'(1+) = (1/8)(P_2 - P_1)$. If $f(t) = (1/2)t^3$, then $\gamma'(1-) = (3/2)(P_2-P_1)$. In this second case, $\gamma''(1-) = 3(P_2 - P_1)$, but $\gamma''(1+) = 0$. As this example shows, a curve need be neither parametric $C^{(2)}$ nor even parametric $C^{(1)}$ to be curvature continuous.

The B-spline representation for spline curves has become the predominant method of representing spline curves since it has so many interesting and useful computational and geometric properties. If S_τ is a space of splines and $\gamma(t) \in S_\tau$, $\gamma(t) = \sum Q_i B_{i,k,\tau}(t)$ is its B-spline representation. $\gamma(t)$ is in the convex hull of $\{Q_i\}$ (indeed, an even stronger local convex hull property applies). Also, $\gamma(t)$ is a variation diminishing approximation to the piecewise linear curve formed by connecting the coefficient points together in order. While these two concepts will be defined below, we note now that these two properties cause the spacial extent and behavior of the spline curve to have strong geometric relationships to the coefficient control polygon. Also, stable algorithms for evaluation [4, 9] and refinement and subdivision [7] exist. The parametric values, and the geometric locations on the curve, at which the different polynomial pieces meet are called the "knots". Since on either side of a knot, the parametric spline is a polynomial, it is certainly parametrically continuous for all possible derivatives as it approaches the knot. However, since the polynomial pieces meeting at the knot are generally different, the parametric derivatives will match only to a certain level, frequently made to be the second derivative. This choice of the second derivative is to gain curvature continuity. We have seen, however, that this is not necessary.

The work which uses this flexibility is frequently based on spline spaces of order 4 (cubic) splines. Unless another degree spline is specifically mentioned, all references in this paper are to cubics.

Manning [12] and Sabin [15] quite early proposed using this extra flexibility for design flexibility. Sabin, while recognizing that one of the tension parameters was basically a parametric rescaling, did not use it. However he noted that the other could be varied and still give curvature continuity. Sabin wanted to apply this concept to design by specifying points of interpolation. He presented not only the concept, but an iterative method for finding good values of the discrete tension parameters based on the data and a minimization of the resulting curvature. Manning also considered this problem of designing with interpolation. He used both of the scalar degrees of freedom afforded by curvature continuous, but only $C^{(0)}$ parametric, cubics. He also recognized the need and derived algorithms for automatically and iteratively determining two tension values at each knot which would have good design properties for both open and periodic curves. Both, however, used a uniform parametrization and the truncated power basis. Under these parametrization conditions the matrix formulations used by both are the same between each pair of knots. Since the piecewise power basis was used directly, the coefficient sequence involved did not have the strong geometric relevance that the B-spline coefficient sequence has. Nielson [13] proposed using parametrically $C^{(1)}$, but curvature continuous, splines in the plus function basis as a minimum solution of an interpolation problem. He presented this material as the solution to a discrete minimization problem analogous to the one solved by splines under tension. The extra flexibility occurs as a scalar at each knot, the "tension", and is used to tighten or loosen the curve around the interpolation value. The resulting curves, called ν-splines, are curvature continuous.

More recently, Barsky has proposed using parametrically $C^{(0)}$ piecewise cubic polynomials with curvature continuity for design. The idea is to use the additional flexibility with local basis functions represented in a piecewise power basis whose attributes are styled after the desirable attributes of B-splines. In particular, if a curvature continuous curve is represented in such a formulation, there should be geometrical properties relating the curve and the sequence of its vector coefficients. We shall call this general class of formulations the "design" formulation. While two tension parameters are theoretically allowed at each breakpoint, computational complexity for the formulation has meant that uniformly spaced breakpoints, floating curves, and most frequently just two tension values per curve are used, which he calls β_1 and β_2. Because of his formulation, the β-spline formulation, he has been restricted both theoretically and computationally to the use of coefficient polygons with "floating end conditions". No knot insertion algorithms exist in this formulation. It is difficult to understand, define, and evaluate the basis functions for the nonhomogeneous conditions using the β-spline representation. Farin [10] investigated developing the "interpolating" control polygon points for the piecewise cubic Bézier representation given an original control polygon, the "piecewise inner Bézier control points", and a particular way of specifying the two tension scalars.

In this paper we present a formulation for using the extra degrees of freedom allowed by requiring that the cubic parametric splines be only parametrically $C^{(0)}$ at the knots, but are still required to be curvature continuous at those points. The formulation is based on defining and using appropriate B-splines as the foundation for the new functions, called *local tensioned* (LT) splines which will be shown to be a local basis for the space of tensioned splines. It will be shown that the LT-spline basis supports knot insertion and also supports both floating and open end conditions, both uniform and nonuniform knot vectors, and both homogeneous and nonhomogenous tension values as particular instantiations. A curve and control polygon represention in the LT-basis has the "design" formulation characteristics. The power of B-splines may then be brought to bear on problems and functions of this type.

2. Background

In this section we define several of the desirable properties which B-spline representations of curves convey about the geometry of the curve. They are also, where appropriate, shared by the Bézier curve formulation and some have been shown for the β-spline curve formulation [3, 11]. Later in this paper, we shall show that the LT-splines also have all of these properties.

A parametric (vector) curve representation is used for many curve design schemes in practice and in theory. We shall adopt that format. For this investigation we suppose $\gamma(t) = \sum_i P_i f_i(t)$, where the coefficients, P_i, are vectors.

The vector coefficients, $\{P_i\}$, are often called the "control vertices" and can be connected sequentially by subscript to form a piecewise linear curve, L_γ. This is frequently called the "control polygon" or the "control net".

> **Definition 1:** If a straight line intersects $\gamma(t)$ no more often then it intersects L_γ, for all possible ordered collections of coefficients $\{P_i\}$, then $\gamma(t)$ is called a *variation diminishing* approximation to L_γ

The effect of this property is that the resulting vector-valued curve has no more undulations than the piecewise linear curve of vector-valued coefficients. This property also determines many features of the extent of the blended curve.

Another such property is the convex hull property.

Definition 2: A *convex set*, C, is a set of points such that if U, V $\in$ C, then cU + (1-c)V $\in$ C for all c $\in$ [0,1]. That means that the line segment joining U and V must be entirely in the set C.

Definition 3: The *convex hull* of a set of points is the smallest convex set containing those points.

Definition 4: The curve approximation of $\{P_i\}$ and $\gamma(t)$ is said to have the *convex hull property* if the curve $\gamma(t)$ is contained in the convex hull of the coefficient set $\{P_i\}$.

If a convex hull property exists, the extent of the curve is known. A more stringent form is a local convex hull property which can place the extent of the curve over subintervals within the convex hulls of known subsets of the coefficient sequence.

Features which make the basis functions $f_i(t)$ easier to work with include

- nonnegative values over the domain (necessary for the convex hull property, and
- local support of each blending function, which means it is nonzero only over a small interval of the whole function domain.

2.1. Polynomial Splines

In everyday use, the functions $f_i(t)$ are most often polynomials or piecewise polynomials. Every piecewise polynomial is a spline [5]; however the functions f_i can be any of many different bases for splines. One such basis for splines is the B-spline basis. B-splines are widely used for modelling and defined in many different places, so we shall assume the reader is familiar with their properties and here just summarize some of their properties.

A particular space of piecewise polynomials is completely characterized by its maximum degree, the parameter values where the polynomials may have a discontinuity in some derivative (also called the breakpoints), and a sequence specifying the highest derivative continuity required at each of the break points. Call the sequence of B-splines over that space $\{B_{i,k}(t)\}$, where k-1 is the maximal degree of the polynomial pieces. Then,

* $B_{i,k}(t) \geq 0$, for all t;

* $\sum_i B_{i,k}(t) \equiv 1$ for all t; and

*each $B_{i,k}(t)$ is local, and in fact the set of B-splines has the smallest support possible for any basis of the space.

*If $\gamma(t) = \sum_i P_i B_{i,k}(t)$ is the B-spline representation of a spline curve $\gamma(t)$ then the curve and control polygon have both the variation diminishing property and the convex hull property.

2.2. Curvature Continuity - Discrete, Single Value

Definition 5: A parametric curve C(t) = (x(t), y(t), z(t)) which is continuous in its domain is called *curvature continuous* if in its arc length parametrization C is $C^{(2)}$. Note that usually one does not have the arc length parametrization, so that one must find the constraints upon C(t) in the given parametrization. These amount to the following conditions. Given a point t_k in the domain of C, there exist scalars $\mu_k > 0$ and ν_k such that

$$C'(t_k^+) = \mu_k C'(t_k^-)$$
$$C''(t_k^+) = \mu_k^2 C''(t_k^-) + \nu_k C'(t_k^-) \qquad (1)$$

Note that for any fixed value of t, the scalars can vary arbitrarily and still preserve curvature continuity.

2.2.1. Curvature Continuous Piecewise Cubics

It is known that every piecewise cubic polynomial can be expressed as a linear combination of the appropriate B-spline basis functions. Hence, every piecewise cubic which is $C^{(2)}$ in the arc length parametrization satisfies equations 1 in its current parametrization and can be written as a combination of B-splines with triple knots at the desired locations. Unfortunately, the use of triple knots requires that the user or the system check that the conditions really are met, and change in the tangent or curvature parameters (μ and ν) requires solving a whole new system to find the appropriate coefficients. The reality is that the cubic spline space with that knot configuration contains the tensioned splines but is too big. Even worse, the coefficients have little geometric meaning. If one uses a truncated power basis, or defines a power basis over each interval, redefining the parametrization each time, one still faces the same problems, with the additional one of needing to also constrain and find the appropriate coefficients to keep the global curve continuous [2, 13, 1]. The tensioned splines, on the other hand, form a subspace.

We shall define a framework for comparing the various bases.

Definition 6: Given any strictly increasing sequence of real values t_0, t_1, ..., t_{n+1}, and sequences $\{\mu_i\}$ and $\{\nu_i\}$, i = 1,..., n, define the space $CCS_{t,\mu,\nu}[t_j,t_k]$ of all parametric functions f which are $C^{(0)}$ cubic polynomials on $[t_j,t_k]$, for $0 \le j < k \le n+1$, and which also satisfy equations 1 at each t_i, where $j < i < k$.

This space is the space of cubic tensioned splines. Clearly it is a subspace of the space of piecewise cubic, parametric, $C^{(0)}$ splines. We will find a local, minimal basis, $\{S_j\}$, for two types of these spaces. Within each type of space, the values in the sequences $\{\mu_j\}$ and $\{\nu_j\}$ are arbitrary, and the spacing between the breakpoints t_i is arbitrary.

1. The curvature continuous piecewise cubics with floating end conditions. The functions $\{S_j\}$ will form a basis over $CCS_{t,\mu,\nu}[t_3, t_{n-2}]$.
2. The curvature continuous piecewise cubics with open end conditions. The functions $\{S_j\}$ will form a basis over $CCS_{t,\mu,\nu}[t_0, t_{n+1}]$.

We shall show that the basis for 2 is the basis for 1 augmented by the boundary condition basis functions. There will be a distinct geometrical relevance here. The bases will have the property that if

$\gamma(t) = \Sigma P_j S_j(t)$ is a curve, then

1. $\gamma(t)$ will be a variation diminishing approximation to the control polygon,
2. $\gamma(t)$ will lie in convex hull of the P_j, and in fact a local convex hull property similar to that for B-spline curves will prevail,
3. for open end conditions, $\gamma(t_0) = P_0$ and $\gamma(t_{n+1}) = P_n$.

In the case that floating end conditions are desired, these functions will form an easily computed basis for the curvature continuous splines that allow the type of design promulgated by Barsky's β-splines. This basis allows arbitrary or uniform spacing at will and still remains easily computed, fitting within the same function definition. The ability to use tension with the designing polygon and have open end conditions is new. Both spaces have as local bases the appropriate collection of $\{S_j\}$. Hence, it is possible to write all curvature continuous cubic polynomials as combinations of local functions, which are themselves combinations of B-splines. The power of B-splines may then be brought to bear on problems and functions of this type.

3. Conventions

As yet we have not defined exactly what the functions S_j will look like. We will first define the formalism for identifying the vector spaces of linear combinations of them. As we have seen, these functions are dependent on the coefficients μ_k and ν_k, k = 1,...,n, which occur in equations 1 as well as on the values t_k, k = 0,...,n+1.

Definition 7: Define $T_p = T_{t,\mu,\nu,p} = \{S_j(t)\}$. The "openness" or "floatingness" is determined by the value of p, where p = f for floating and p = o, for open. Clearly these blending functions should be completely defined by the knot sequence, the specific values of μ_i and ν_i, i=1,...,n selected, and the open/floating decision. We also define $S_p = S(T_p) = \{ \Sigma P_j S_j(t): P_j \in \mathbf{R}^3$ and $S_j \in T_p\}$, the space of locally tensioned (LT-) splines.

The purpose of this framework is to define LT-splines, show that they are linearly independent, and then to show that they form a basis for *CCS*. We shall first define the underlying space of B-splines which will be used to define the functions S_j.

For a strictly increasing sequence of real values $t_0, t_1, \ldots t_{n+1}$, define the knot sequence $\tau = \{\tau_i\}$ where

$$\tau_i = \begin{cases} t_0, \; i = 0, 1, 2, 3, \\ t_j, \; i = 3j+1, 3j+2, 3j+3, \text{ for } j=1,\ldots,n, \\ t_{n+1}, \; i=3n+4, 3n+5, 3n+6, 3n+7. \end{cases} \tag{2}$$

Denote the cubic B-spline basis functions over this knot sequence as $\{N_k\}_{k=0}^{3n+3}$. Then $N_{3j}(t)$ is the $C^{(0)}$ piecewise cubic spline with maximum value 1 at t_j, and having support $[t_{j-1}, t_{j+1}]$, j=1,...,n.

We seek to define piecewise cubic functions $S_j(t) \in CCS$, j=0,...,n-3, each with support $[t_j, t_{j+4}]$, which are single polynomials between t_j and t_{j+1}, between t_{j+1} and t_{j+2}, between t_{j+2} and t_{j+3}, and between t_{j+3} and t_{j+4}. The support constraint in conjunction with the geometric continuity conditions at t_{j+k}, $k = 0,...,4$ impose the following continuity conditions, which we shall later verify:

$S_j(t)$ is $C^{(2)}$ at t_j and t_{j+4},
is curvature continuous at t_i, $i=j+1, j+2, j+3$, that is, satisfies equations 1, (3)

The open end conditions impose further constraints. We shall need to define six additional functions, three at each end, which will enforce the end conditions which we shall prescribe. The purpose of the end conditions will be to keep the same geometric effects that occur with open B-spline curves and to keep the support as minimal as possible subject to curvature continuity constraints. We shall require that:

$S_{-i}(t)$
has support on $[t_0, t_{4-i}]$
is in $C^{(2-i)}$ at t_0,
is curvature continuous at t_j, $j = 1, ..., 4-i$.

$i = 3, 2, 1.$

This last condition implies containment in $C^{(2)}$ at t_{4-i}. Functions S_{n-2}, S_{n-1}, and S_n are defined analogously.

3.1. S's as combinations of B-splines

First assume that $j \in \{0, 1, ..., n-3\}$ and that S_j is defined on $[t_j, t_{j+4}]$. That is, the function is not one of the boundary functions.

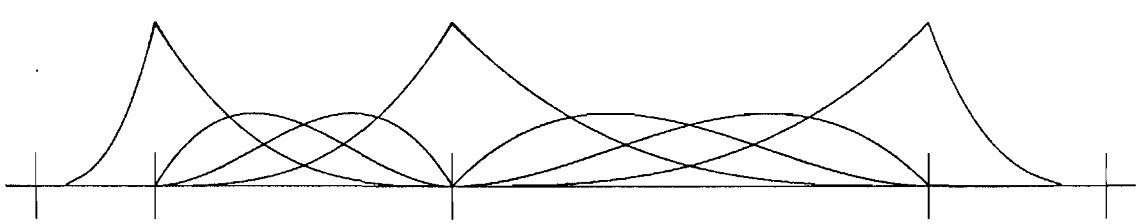

Figure 3-1: The B-splines composing the j-th S-function

Since we seek to write S_j as a combination of B-splines, and the support of S_j is $[t_j, t_{j+4}]$, we see that, for $i \le 3j$ and for $i \ge 3(j+4)$, $c_{j,i} = 0$, so that

$$S_j(t) = \sum_{i=3j+1}^{3(j+3)+2} c_{j,i} N_i(t).$$

This occurs by matching the supports of the various functions. We must further match the end continuities to check if all these coefficients will be nonzero.

Theorem 8: If S_j is constrained as defined above, then

$$S_j(t) = \sum_{i=3j+3}^{3j+9} c_{j,i}N_i(t),\ j = 0,\ldots,n-3.$$

Proof: S_j is curvature continuous t_j requires

$$\mu_j S_j'(t_j^-) = S_j'(t_j^+).$$

Rewriting in terms of basis functions,

$$0 = c_{j,3j+1}N_{3j+1}'(t_j^+) \quad \text{or} \quad 0 = c_{j,3j+1}$$

since the support of S_j is in $[t_j, t_{j+4}]$, and since $N_{3j+q}'(t_j^+) = 0$, if $q > 3j + 1$, and $N_{3j+1}'(t_j^+) \neq 0$.

Next, since

$$\mu_j^2 S_j''(t_j^-) = S_j''(t_j^+),$$

rewriting in terms of basis functions gives

$$0 = c_{j,3j+1}N_{3j+1}'(t_j^+) + c_{j,3j+2}N_{3j+2}''(t_j^+)$$
$$= c_{j,3j+2}N_{3j+2}''(t_j^+)$$

The conclusion,

$$c_{j,3j+2} = 0,$$

follows again from the facts that the support of S_j is in $[t_j, t_{j+4}]$, that $N_{3j+q}''(t_j^+) = 0$, if $q > 3j + 2$, and that $N_{3j+2}''(t_j^+) \neq 0$.

We have shown that $c_{j,3j+1} = c_{j,3j+2} = 0$. Analogously, we may show that $c_{j,3(j+3)+1} = c_{j,3(j+3)+2} = 0$ from matching continuity at t_{j+4}.

Now consider the boundary functions. The support of S_{-3} is $[t_0, t_1]$. Also, S_{-3} is $C^{(2)}$ at t_1 and discontinuous at t_0 require that $c_{-3,i} = 0, i > 0$.

Thus, $S_{-3}(t) = c_{-3,0}N_0(t)$.

Since the support of S_{-2} is $[t_0, t_{-2}]$, the function is $C^{(0)}$ at t_0, and S_{-2} is $C^{(2)}$ at t_2, $c_{-2,0} = 0$ and $c_{-2,i} = 0$, for $i > 3$ (for the same reasons that $c_{j,i} = 0$ for $i > 3j+9$, in the general case). So, $S_{-2}(t) = \sum_{i=1}^{3} c_{-2,i}N_i(t)$.

Since the support of S_{-1} is $[t_0, t_3]$, the function is $C^{(1)}$ at t_0, and the function is $C^{(2)}$ at t_3, $c_{-1,i} = 0$ for $i = 0, 1$, and $c_{-1,i} = 0$ for $i > 6$. Hence, $S_{-1}(t) = \sum_{i=2}^{6} c_{-1,i}N_i(t)$.

The conditions at the upper end of the domain are analogous, so, we have shown

Theorem 9: If tension splines with open end conditions are needed then the basis functions which show the effects of the boundary are written

$$S_{-3}(t) = c_{-3,0}N_0(t),$$

$$S_{-2}(t) = \sum_{i=1}^{3} c_{-2,i} N_i(t),$$

$$S_{-1}(t) = \sum_{i=2}^{6} c_{-1,i} N_i(t),$$

$$S_{n-2}(t) = \sum_{i=3(n-2)+3}^{3(n-2)+7} c_{n-2,i} N_i(t),$$

$$S_{n-1}(t) = \sum_{i=3(n-1)+3}^{3(n-1)+5} c_{n-1,i} N_i(t),$$

$$S_n(t) = c_{n,3n+3} N_{3n+3}(t).$$

where $N_i = N_{i,\tau}$ is the τ sequence defined above.

3.2. Normalization Conditions

The final unknowns develop from the normalization requirement that

$$\sum_j S_j(t) \equiv 1, \text{ for all t in the domain.} \tag{4}$$

This condition adds global constraints on finding the coefficients $c_{j,i}$. However, for a fixed j, the coefficients $c_{j,i}$ can be uniquely solved and will be shown to depend only on μ_k, ν_k, t_k, $k = j, ..., j+4$. That is, the function S_j is defined uniquely by values contained totally within its support.

Over the interval $[t_i, t_{i+1}]$, the normalization condition, Equation 4 will provide the final constraint for unique solution. For values of t in the interval $[t_i, t_{i+1}]$, the only S functions which can be nonzero are the four functions S_k, $k = i-3, ..., i$, and with the knot vector τ the only B-splines which can be nonzero are N_p, $p = 3i,...,3i+3$. Thus,

$$1 \equiv \sum_{k=i-3}^{i} S_k(t),$$

$$\equiv \sum_{k=i-3}^{i} \sum_{p=3k+3}^{3k+9} c_{k,p} N_p(t),$$

$$\equiv \sum_{p=3i}^{3i+3} \{ \sum_{k=i-3}^{i} c_{k,p} \} N_p(t),$$

But since over that interval

$$1 \equiv \sum_{p=3i}^{3i+3} 1 * N_p(t)$$

and the $\{N_p\}$ form a basis, that means that the two equations must have the same coefficients. Hence, we obtain

$$1 = c_{i-3,3i} + c_{i-2,3i} + c_{i-1,3i} \tag{5}$$
$$1 = c_{i-2,3i+1} + c_{i-1,3i+1} \tag{6}$$
$$1 = c_{i-2,3i+2} + c_{i-1,3i+2} \tag{7}$$
$$1 = c_{i-2,3(i+1)} + c_{i-1,3(i+1)} + c_{i,3(i+1)} \tag{8}$$

If we consider the normalization conditions over the interval $[t_{i+1}, t_{i+2}]$, we arrive at the following equations:

$$1 = c_{i-2,3(i+1)} + c_{i-1,3(i+1)} + c_{i,3(i+1)}$$
$$1 = c_{i-1,3(i+1)+1} + c_{i,3(i+1)+1} \tag{9}$$
$$1 = c_{i-1,3(i+1)+2} + c_{i,3(i+1)+2} \tag{10}$$
$$1 = c_{i-1,3(i+2)} + c_{i,3(i+2)} + c_{i+1,3(i+2)} \tag{11}$$

Over the interval $[t_{i+2}, t_{i+3}]$ the normalization conditions require that:

$$1 = c_{i-1,3(i+2)} + c_{i,3(i+2)} + c_{i+1,3(i+2)}$$
$$1 = c_{i,3(i+2)+1} + c_{i+1,3(i+2)+1} \tag{12}$$
$$1 = c_{i,3(i+2)+2} + c_{i+1,3(i+2)+2} \tag{13}$$
$$1 = c_{i,3(i+3)} + c_{i+1,3(i+3)} + c_{i+2,3(i+3)} \tag{14}$$

From these equations we see that equation 8 occurs as the first equation on the next interval, and equation 11 occurs as the first equation over the next interval. Equations 9 and 10 are unique to the (i+1)-st interval, and will be used to get the last degree of freedom resolved.

These equations are different in the case of the open end conditions only near the boundaries, and will be treated in the discussion of that case.

4. Solving The System Away From The Boundaries

To find the coefficients of N_i defining S_j, for a fixed j, we need only solve for the seven coefficients of the N_i. If either the μ and ν are not all the same values, or if the values of t_j are not uniformly spaced, S_j and S_p, $p \neq j$, will have different coefficients for their respective B-splines.

For a fixed j, We can determine six linearly independent conditions on the function S_j by considering the two constraints of equations 1 at the points t_k, $k = j+1, j+2, j+3$. That is,

$$S_j'(t_k^+) = \mu_k S_j'(t_k^-) \tag{15}$$
$$S_j''(t_k^+) = \mu_k^2 S_j''(t_k^-) + \nu_k S_j'(t_k^-) \tag{16}$$
$$k = j+1, j+2, j+3.$$

These equations become:

$$\sum_{i=3j+3}^{3j+9} c_{j,i}N_i'(t_k^+) = \mu_k \sum_{i=3j+3}^{3j+9} c_{j,i}N_i'(t_k^-)$$

$$\sum_{i=3j+3}^{3j+9} c_{j,i}N_i''(t_k^+) = \mu_k^2 \sum_{i=3j+3}^{3j+9} c_{j,i}N_i''(t_k^-) + \nu_k \sum_{i=3j+3}^{3j+9} c_{j,i}N_i'(t_k^-)$$

$$k = j+1, j+2, j+3.$$

In order to set up these six linear equations in the seven unknowns, we must symbolically evaluate the appropriate left and right sided first and second derivatives.

At t_{j+1}:

$$N'_{3j+3}(t_{j+1}^-) = \frac{3}{t_{j+1}-t_j} \qquad N'_{3j+3}(t_{j+1}^+) = -\frac{3}{t_{j+2}-t_{j+1}}$$

$$N'_{3j+4}(t_{j+1}^+) = \frac{3}{t_{j+2}-t_{j+1}}$$

$$N''_{3j+3}(t_{j+1}^-) = \frac{6}{(t_{j+1}-t_j)^2} \qquad N''_{3j+3}(t_{j+1}^+) = \frac{6}{(t_{j+2}-t_{j+1})^2}$$

$$N''_{3j+4}(t_{j+1}^+) = -\frac{12}{(t_{j+2}-t_{j+1})^2}$$

$$N''_{3j+5}(t_{j+1}^+) = \frac{6}{(t_{j+2}-t_{j+1})^2}$$

At t_{j+2}:

$$N'_{3j+5}(t_{j+2}^-) = -\frac{3}{t_{j+2}-t_{j+1}}$$

$$N'_{3j+6}(t_{j+2}^-) = \frac{3}{t_{j+2}-t_{j+1}} \qquad N'_{3j+6}(t_{j+2}^+) = -\frac{3}{t_{j+3}-t_{j+2}}$$

$$N'_{3j+7}(t_{j+2}^+) = \frac{3}{t_{j+3}-t_{j+2}}$$

$$N''_{3j+4}(t_{j+2}^-) = \frac{6}{(t_{j+2}-t_{j+1})^2}$$

$$N''_{3j+5}(t_{j+2}^-) = -\frac{12}{(t_{j+2}-t_{j+1})^2}$$

$$N''_{3j+6}(t_{j+2}^-) = \frac{6}{(t_{j+2}-t_{j+1})^2} \qquad N''_{3j+6}(t_{j+2}^+) = \frac{6}{(t_{j+3}-t_{j+2})^2}$$

$$N''_{3j+7}(t_{j+2}^+) = -\frac{12}{(t_{j+3}-t_{j+2})^2}$$

$$N''_{3j+8}(t_{j+2}^+) = \frac{6}{(t_{j+3}-t_{j+2})^2}$$

At t_{j+3}:

$$N'_{3j+8}(t_{j+3}^-) = -\frac{3}{t_{j+3}-t_{j+2}}$$

$$N'_{3j+9}(t_{j+3}^-) = \frac{3}{t_{j+3}-t_{j+2}} \qquad N'_{3j+9}(t_{j+3}^+) = -\frac{3}{t_{j+4}-t_{j+3}}$$

$$N''_{3j+7}(t_{j+3}^-) = \frac{6}{(t_{j+3}-t_{j+2})^2}$$

$$N''_{3j+8}(t_{j+3}^-) = -\frac{12}{(t_{j+3}-t_{j+2})^2}$$

$$N''_{3j+9}(t_{j+3}^-) = \frac{6}{(t_{j+3}-t_{j+2})^2} \qquad N''_{3j+9}(t_{j+3}^+) = \frac{6}{(t_{j+4}-t_{j+3})^2}$$

Evaluating 15 explicitly and regrouping:

$$c_{j,3j+4} - c_{j,3j+3} = \mu_{j+1}\left(\frac{t_{j+2}-t_{j+1}}{t_{j+1}-t_j}\right)c_{j,3j+3} \tag{17}$$

$$-c_{j,3j+6} + c_{j,3j+7} = \mu_{j+2}\left(\frac{t_{j+3}-t_{j+2}}{t_{j+2}-t_{j+1}}\right)(c_{j,3j+6} - c_{j,3j+5}) \tag{18}$$

$$-c_{j,3j+9} = \mu_{j+3}\left(\frac{t_{j+4}-t_{j+3}}{t_{j+3}-t_{j+2}}\right)(c_{j,3j+9} - c_{j,3j+8}) \tag{19}$$

Equation 16 after regrouping gives more complicated equations:

$$c_{j,3j+5} - 2c_{j,3j+4} + c_{j,3j+3} = [\mu_{j+1}\frac{t_{j+2}-t_{j+1}}{t_{j+1}-t_j}]^2 c_{j,3j+3} + \nu_{j+1}\frac{(t_{j+2}-t_{j+1})^2}{2(t_{j+1}-t_j)}c_{j,3j+3} \tag{20}$$

$$c_{j,3j+8} - 2c_{j,3j+7} + c_{j,3j+6} = [\mu_{j+2}\frac{t_{j+3}-t_{j+2}}{t_{j+2}-t_{j+1}}]^2(c_{j,3j+6} - 2c_{j,3j+5} + c_{j,3j+4})$$

$$+ \nu_{j+2}\frac{(t_{j+3}-t_{j+2})^2}{2(t_{j+2}-t_{j+1})}(c_{j,3j+6} - c_{j,3j+5}) \tag{21}$$

$$c_{j,3j+9} = [\mu_{j+3}\frac{t_{j+4}-t_{j+3}}{t_{j+3}-t_{j+2}}]^2(c_{j,3j+9} - 2c_{j,3j+8} + c_{j,3j+7})$$

$$+ \nu_{j+3} \frac{(t_{j+4}-t_{j+3})^2}{2(t_{j+3}-t_{j+2})} (c_{j,3j+9} - c_{j,3j+8}) \tag{22}$$

Clearly some simplification is necessary. We shall let:

$$\lambda_k = \frac{t_{k+1}-t_k}{t_k-t_{k-1}} \mu_k \tag{23}$$

$$\eta_k = \frac{(t_{k+1}-t_k)^2}{2(t_k-t_{k-1})} \nu_k \tag{24}$$

$$\Lambda_k = \lambda_k{}^2 + \lambda_k + \eta_k \tag{25}$$

$$\Delta_{j,k} = c_{j,k+1}-c_{j,k} \tag{26}$$

for all k.

Substituting 23, 24, and 26 into equations 17- 22 and rearranging within equations and order of equations once again gives:

$$\begin{array}{lll}
\lambda_{j+1} c_{j,3j+3} & = & \Delta_{j,3j+3} \\
(\lambda_{j+1}{}^2+\eta_{j+1}) c_{j,2j+3} & = & -\Delta_{j,3j+3} + \Delta_{j,3j+4} \\
0 & = & - \Delta_{j,3,+5}\lambda_{j+2} \quad + \Delta_{j,3,+6} \\
0 & = & +\Delta_{j,3j+4}\lambda_{j+2}{}^2 \; - \Delta_{j,3j+5}(\lambda_{j+2}{}^2+\eta_{j+2}) \; - \Delta_{j,3j+6} + \Delta_{j,3j+7} \\
c_{j,3j+9} & = & - \Delta_{j,3j+7}\lambda_{j+3}{}^2 + \Delta_{j,3j+8}(\lambda_{j+3}{}^2+\eta_{j+3}) \\
c_{j,3,+9} & = & - \Delta_{j,3j+8}\lambda_{j+3}
\end{array}$$

We can easily get each $\Delta_{j,3j+k}$, $k = 3,\ldots,8$ in terms of $c_{j,3j+3}$ and $c_{j,3j+9}$. Then using the fact that $c_{j,3j+9} - c_{j,3j+3} = \sum_{k=3j+3}^{3j+8} \Delta_{j,k}$, and letting

$$L_k = \Lambda_{k+1} + \Lambda_{k+1}\Lambda_k + \Lambda_{k+1}\lambda_k + \lambda_{k+1}{}^2\Lambda_k + \lambda_{k+1}{}^3\Lambda_k \tag{27}$$

we solve the system to get:

$$c_{j,3j+9} = \frac{L_{j+1}}{L_{j+2}} \lambda_{j+3}{}^3 c_{j,3j+3}, \tag{28}$$

and

$$\Delta_{j,3j+3} = \lambda_{j+1} c_{j,3j+3} \tag{29}$$

$$\Delta_{j,3j+4} = \Lambda_{j+1} c_{j,3j+3} \tag{30}$$

$$\Delta_{j,3j+5} = (\lambda_{j+2}^2 \frac{\Lambda_{j+1}}{\Lambda_{j+2}} - \frac{\Lambda_{j+3}}{\Lambda_{j+2}} \frac{L_{j+1}}{L_{j+2}}) c_{j,3j+3} \tag{31}$$

$$\Delta_{j,3j+6} = (\lambda_{j+2}^3 \frac{\Lambda_{j+1}}{\Lambda_{j+2}} - \lambda_{j+2} \frac{\Lambda_{j+3}}{\Lambda_{j+2}} \frac{L_{j+1}}{L_{j+2}}) c_{j,3j+3} \tag{32}$$

$$\Delta_{j,3j+7} = - \Lambda_{j+3} \frac{L_{j+1}}{L_{j+2}} c_{j,3j+3} \tag{33}$$

$$\Delta_{j,3j+8} = - \lambda_{j+3}^2 \frac{L_{j+1}}{L_{j+2}} c_{j,3j+3} \tag{34}$$

Since $c_{j,3j+p+1} = c_{j,3j+3} + \Sigma_{i=3}^{p} \Delta_{j,3j+i}$, for p = 3,...,8, we can easily obtain all the coefficients of the B-splines once $c_{j,3j+3}$ is known.

This value must be obtained from the normalization equations. Subtracting equation 9 from 10 gives

$$\Delta_{j-1,3(j-1)+7} = - \Delta_{j,3j+4} \tag{35}$$

If one replaces j by j-1 in equation 33, Equations 29, 33, and 35 can be used to obtain

$$c_{j-1,3(j-1)+3} = \frac{\Lambda_{j+1}}{\Lambda_{j+2}} \frac{L_{j+1}}{L_j} c_{j,3j+3} \tag{36}$$

In an analogous fashion, subtracting Equation 12 from Equation 13 gives:

$$\Delta_{j,3j+7} = - \Delta_{j+1,3(j+1)+4} \tag{37}$$

Using Equations 29, 33, and 37, but replacing j by j+1 in equation 33 gives:

$$c_{j+1,3(j+1)+3} = \frac{\Lambda_{j+3}}{\Lambda_{j+2}} \frac{L_{j+1}}{L_{j+2}} c_{j,3j+3} \tag{38}$$

It seems now that S_{j-1} and S_{j+1} can be written in terms of the coefficients of S_j, which indeed they can. Using Equation 11 now gives that final condition to uniquely solve the system for normalized functions. To do this, we must first obtain $c_{j-1,3(j-1)+9}$, $c_{j,3j+6}$, and $c_{j+1,3(j+1)+3}$ as functions of $c_{j,3j+3}$. However, we can replace j by j-1 in equation 28, and use Equation 36.

$$c_{j-1,3(j-1)+9} = \frac{L_j}{L_{j+1}} \lambda_{j+2}{}^3 c_{j-1,3(j-1)+3}$$

$$= \frac{L_j}{L_{j+1}} \lambda_{j+2}{}^3 \frac{\Lambda_{j+1}}{\Lambda_{j+2}} \frac{L_{j+1}}{L_j} c_{j,3j+3}$$

$$= \lambda_{j+2}{}^3 \frac{\Lambda_{j+1}}{\Lambda_{j+2}} c_{j,3j+3} \tag{39}$$

Since

$$c_{j,3j+6} = \Delta_{j,3j+5} + \Delta_{j,3j+4} + \Delta_{j,3j+3} + c_{j,3j+3}$$

$$= (\lambda_{j+2}{}^2 \frac{\Lambda_{j+1}}{\Lambda_{j+2}} - \frac{\Lambda_{j+3}}{\Lambda_{j+2}} \frac{L_{j+1}}{L_{j+2}} + \Lambda_{j+1} + \lambda_{j+1} + 1) c_{j,3j+3}. \tag{40}$$

Combining Equations 11, 38, 39, and 40 gives:

$$1 = c_{j-1,3(j-1)+9} + c_{j,3j+6} + c_{j+1,3(j+1)+3}$$

$$= \lambda_{j+2}{}^3 \frac{\Lambda_{j+1}}{\Lambda_{j+2}} c_{j,3j+3} + (\lambda_{j+2}{}^2 \frac{\Lambda_{j+1}}{\Lambda_{j+2}} - \frac{\Lambda_{j+3}}{\Lambda_{j+2}} \frac{L_{j+1}}{L_{j+2}} + \Lambda_{j+1} + \lambda_{j+1} + 1) c_{j,3j+3}$$

$$+ \frac{\Lambda_{j+3}}{\Lambda_{j+2}} \frac{L_{j+1}}{L_{j+2}} c_{j,3j+3}$$

$$= (\lambda_{j+2}{}^3 \frac{\Lambda_{j+1}}{\Lambda_{j+2}} + \lambda_{j+2}{}^2 \frac{\Lambda_{j+1}}{\Lambda_{j+2}} + \Lambda_{j+1} + \lambda_{j+1} + 1) c_{j,3j+3}.$$

Finally,

$$c_{j,3j+3} = \frac{\Lambda_{j+2}}{\lambda_{j+2}{}^3 \Lambda_{j+1} + \lambda_{j+2}{}^2 \Lambda_{j+1} + \Lambda_{j+1} \Lambda_{j+2} + \lambda_{j+1} \Lambda_{j+2} + \Lambda_{j+2}} \tag{41}$$

Using equations 29 through 34, 27, 41, and the fact that $c_{j,3j+p+1} = c_{j,3j+3} + \sum_{i=3}^{p} \Delta_{j,3j+i}$, for p = 3,...,8, we easily obtain:

$$c_{j,3j+3} = \frac{\Lambda_{j+2}}{L_{j+1}}$$

$$c_{j,3j+4} = (\lambda_{j+1} + 1) \frac{\Lambda_{j+2}}{L_{j+1}}$$

$$c_{j,3j+5} = (\Lambda_{j+1} + \lambda_{j+1} + 1)\frac{\Lambda_{j+2}}{L_{j+1}}$$

$$c_{j,3j+6} = (\Lambda_{j+3} + \lambda_{j+3}{}^3 + \lambda_{j+3}{}^2)\frac{\Lambda_{j+2}}{L_{j+2}} + \lambda_{j+2}\frac{\Lambda_{j+3}}{L_{j+2}} - \lambda_{j+2}{}^3\frac{\Lambda_{j+1}}{L_{j+1}}$$

$$= 1 - \frac{\Lambda_{j+3}}{L_{j+2}} - \lambda_{j+2}{}^3\frac{\Lambda_{j+1}}{L_{j+1}}$$

$$c_{j,3j+7} = (\Lambda_{j+3} + \lambda_{j+3}{}^3 + \lambda_{j+3}{}^2)\frac{\Lambda_{j+2}}{L_{j+2}}$$

$$c_{j,3j+8} = (\lambda_{j+3}{}^3 + \lambda_{j+3}{}^2)\frac{\Lambda_{j+2}}{L_{j+2}}$$

$$c_{j,3j+9} = \lambda_{j+3}{}^3\frac{\Lambda_{j+2}}{L_{j+2}}$$

where (42)

$$L_k = \Lambda_{k+1} + \Lambda_{k+1}\Lambda_k + \Lambda_{k+1}\lambda_k + \lambda_{k+1}{}^2\Lambda_k + \Lambda_k\lambda_k\lambda_{k+1}{}^3$$

and

$$\Lambda_k = \lambda_k{}^2 + \lambda_k + \eta_k$$

It is straightforward to show that this value for $c_{j,3j+3}$ will lead to values for all the other coefficients that satisfy the normalization requirements. Since the follow from straight substitution, but take some algebraic manipulation and space, they are omitted here.

4.1. The Floating Arbitrary Knot Curvature Continuous Spline

We wish to use the polygon $P_0,\ldots, P_{n-3}$ to define a curve using the basis functions S_j for which the domain is $[t_3,t_{n-2}]$ and with tension pairs (μ_i,η_i), for i = 1,...,n. There are a variety of questions which can be asked near the ends, but in general they follow the line of "Are $S_j(t)$,j = 0,1,2, completely defined?", and the corresponding functions for the functions whose support intersect the interval $[t_{n-2}, t_{n+1}]$. Conditions 3 which were used to determine the interior functions apply for the floating end conditions and hence Theorem 8 is still applicable. The remaining questions concern the normalization conditions on the functions S_j.

The S_j, j = 0,..., n-3 are all defined by conditions 28 through 34 and all must satisfy normalization conditons 5 through 14 over intervals $[t_3,t_4],\ldots, [t_{n-3},t_{n-2}]$. Looking at Equations 9 through 14 and letting i = 3 gives the normalization conditions for $c_{2,j}$, in terms of $c_{3,j}$. Since S_3 satisfies the interior function conditons for normalization, so must S_2. Analogously, S_i, i = 1 and i = 0 also satisfy the same conditions. At the opposite end, symmetric conditions hold, and those end functions satisfy the same conditions as the interior functions.

Hence, just evaluating the coefficients for the appropriate values of j in equations 42 gives the correct coefficients for the end functions in the floating end conditions. However, it is unnecessary to use all of these functions. Since S_0 is needed only on $[t_3,t_4]$, we need only evaluate $c_{0,9}$; since S_1 is needed on $[t_3,t_5]$, we need evaluate $c_{1,i}$, i = 6,...,9.

In the examples which follow all the polygons are the same. However, below is the table which refers to the different knot vectors used, and the different tension pairs used.

Figure	Knot Vector	μ Tension Vector	ν tension vector
4-1-a	{0, 2, 3, 5, 6, 8, 9, 11, 12, 15, 16}	{.5, .5, .5, .5, .5, .5, .5, .5}	{ 0, 0, 0, 0, 0, 0, 0, 0}
4-1-b	{0, 2, 3, 5, 6, 8, 9, 11, 12, 15, 16}	{ 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1}	{2, 2, 2, 20, 2, 20, 2, 2, 2}
4-2-a	{0.5, 1, 5, 6, 8, 9, 11, 14, 14.5, 15}	{.5, .5, .5, .5, .5, .5, .5, .5}	{ 0, 0, 0, 0, 0, 0, 0, 0}
4-2-b	{0.5, 1, 5, 6, 8, 9, 11, 14, 14.5, 15}	{ 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1}	{2, 2, 2, 20, 2, 20, 2, 2, 2}

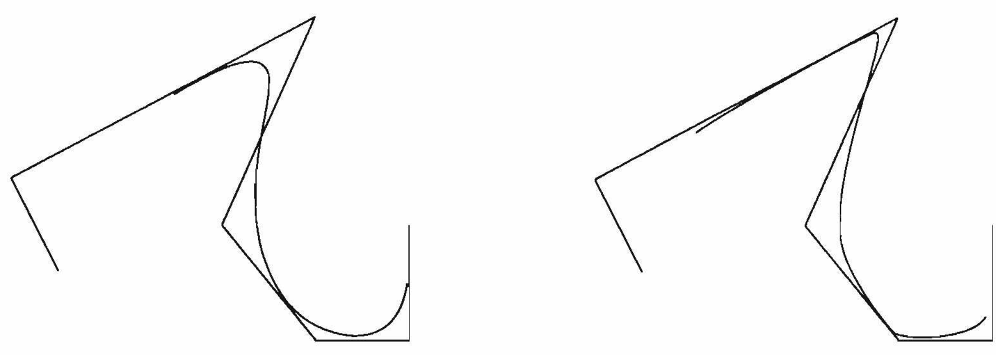

Figure 4-1: a. b.

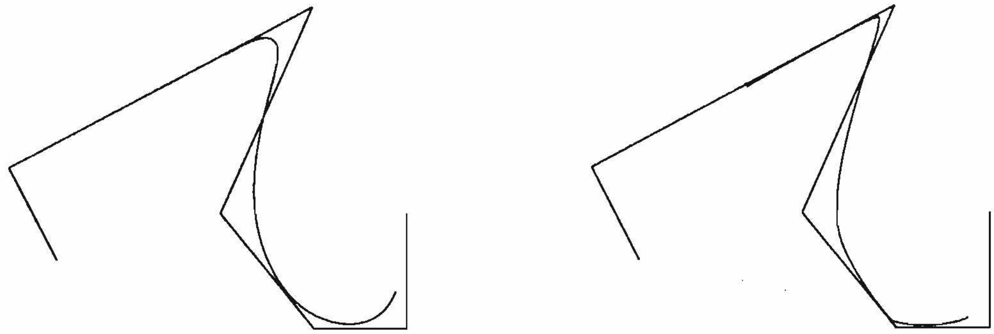

Figure 4-2: a. b.

4.2. Tradeoffs Between Uniform Knots And Single Tension Values

We have seen that the values for the coefficients, the $c_{j,k}$'s depend entirely on the various values of ν_k, λ_k, Λ_k, and L_k. These in turn depend on the values for the tension parameters at each value of t_j as well as the distance between each of the t_j's. Refreshing the reader's memory:

$$L_k = \Lambda_{k+1} + \Lambda_{k+1}\Lambda_k + \Lambda_{k+1}\lambda_k + \lambda_{k+1}^2\Lambda_k + \Lambda_k\lambda_k\lambda_{k+1}^3$$
$$\Lambda_k = \lambda_k^2 + \lambda_k + \eta_k$$

where

$$\lambda_k = \frac{t_{k+1}-t_k}{t_k-t_{k-1}}\mu_k$$

$$\eta_k = \frac{(t_{k+1}-t_k)^2}{t_k-t_{k-1}}\nu_k$$

Hence the values of the various λ, ν, and Λ can differ at two different knots in one of two ways. Either there is a difference in the tension values at the two knots, that is the pair of values (μ_k, ν_k) differs from (μ_j,ν_j), or else the pair of ratios $\left(\frac{t_{k+1}-t_k}{t_k-t_{k-1}}, \frac{(t_{k+1}-t_k)^2}{t_k-t_{k-1}}\right)$ and $\left(\frac{t_{j+1}-t_j}{t_j-t_{j-1}}, \frac{(t_{j+1}-t_j)^2}{t_j-t_{j-1}}\right)$ are different. The effects on the resulting spline are analogous. Note that even if the first ratio in the pairs are 1, the second may not be and hence the pairs themselves may not be equal.

4.3. Example: Uniform Floating Tensioned Spline

For this example it is assumed that $t_{j+1} - t_j = 1$, $j = 0,\dots,n$, and that there is only one distinct value for each of the sequences $\{\mu_j\}$ and $\{\nu_j\}$. Under this knot configuration, all the B-splines over the domain are just translates of each other, and the tension constraint means that there exists one scalar $\mu = \mu_j$, $j = 1,\dots,n$ and one scalar $\nu = \nu_j$, $j = 1,\dots,n$. These conditions are the most common hypothesis for actually using Barsky's β-splines. For our LT-functions it means that $S_j(t) = S_{j+1}(t+1)$.

For this special case, $\lambda_j = \lambda_1 = \mu$, for all j, so set $\lambda_j = \lambda = \mu$. Similarly, $\eta_j = \eta_1 = \nu$, for all j, so set $\eta_j =$

$\eta = \nu$. Then applying equations 42

$$\Lambda = \Lambda_j = \lambda^2 + \lambda + \eta, \text{ for all } j$$

and

$$L = L_j = \Lambda + \Lambda^2 + \Lambda\lambda + \lambda^2\Lambda + \lambda^3\Lambda.$$

This leads to coefficents:

$$c_{j,3j+3} = \frac{\Lambda}{L} = \frac{1}{1+\Lambda+\lambda+\lambda^2+\lambda^3} = C$$

$$c_{j,3j+4} = (1 + \lambda)C$$

$$c_{j,3j+5} = (1 + 2\lambda + \lambda^2 + \eta)C$$

$$c_{j,3j+6} = (2\lambda + 2\lambda^2 + \eta)C$$

$$c_{j,3j+7} = (\lambda + 2\lambda^2 + \lambda^3 + \eta)C$$

$$c_{j,3j+8} = (\lambda^2 + \lambda^3)C$$

$$c_{j,3j+9} = \lambda^3 C$$

5. Open End Conditons

In this case we shall have domain $[t_0, t_{n+1}]$, over which we will have n+4 S_j basis functions, $j = -3,\dots,n$. Given a polygon $P_{-3},\dots,P_0,\dots,P_n$, we wish to determine the class of tensioned spline curves, $\gamma(t) = \sum_{i=-3}^{n} P_i S_i(t)$ which will have similar properties to open B-spline curves. That means the curve should:

1. interpolate P_3 and P_n, that is $\gamma(t_0) = P_{-3}$ and $\gamma(t_{n+1}) = P_n$,
2. have a tangent direction tangent to the first and last "legs" of the polygon, that is, $\gamma'(t_0) = \alpha(P_{-2}-P_{-1})$ and $\gamma'(t_{n+1}) = \beta(P_n-P_{n-1})$, where α and β are some scalars determined by the functions S_i.

We must establish, if possible the values of the coefficients of the boundary S_i functions and ensure that they still satisfy and are consistent with the normalization conditions which tie all the functions together, and the geometry conditions.

Since

$$\begin{aligned}\gamma(t_0) &= P_{-3}\\ &= P_{-3}S_{-3}(t_0)\\ &= P_{-3}c_{-3,0}N_0(t_0)\\ &= P_{-3}c_{-3,0},\end{aligned}$$

this means that $c_{-3,0} = 1$. S_{-3} is then completely determined. Next, consider

$$\begin{aligned}\gamma'(t_0) &= \alpha(P_{-2} - P_{-3})\\ &= P_{-3}S_{-3}'(t_0) + P_{-2}S_{-2}'(t_0)\\ &= P_{-3}N_0'(t_0) + P_{-2}\textstyle\sum_{i=1}^{3} c_{-2,i}N_i'(t_0)\end{aligned}$$

$$= P_{-3}N_0'(t_0) + P_{-2}c_{-2,1}N_1'(t_0)$$

Since $N_0'(t_0) = -N_1'(t_0)$, this means

$$= (c_{-2,1}P_{-2} - P_{-3})N_1'(t_0)$$
$$= (c_{-2,1}P_{-2} - P_{-3})\frac{3}{t_1-t_0}$$

The only way that the first and last forms of this equation can be satisfied is if $c_{-2,1} = 1$.

Analogously, $c_{n,3n+3} = 1$ and $c_{n-1,3n+2} = -1$.

We consider the geometric constraints for the boundary functions individually, and then consider the applications of modified normalization constraints. Clearly, this must be done over all intervals that involve the boundary functions to insure consistency. Further, since the normalization conditions also lead to absolute determination of the coefficients, if the same values for $c_{-2,1}$, $c_{-3,0}$, $c_{n,3n+3}$, and $c_{n-1,3n+2}$ are arrived at, the consistency check is complete.

We initially constrain ourselves to the case for which $n \geq 2$, that is, the number of internal knots is at least two. We consider the special case of the fewer internal knots after this derivation. We first consider the boundary at t_0.

We shall develop the normalization conditions on all the intervals containing boundary functions, the intervals $[t_0, t_1]$, $[t_1, t_2]$, and $[t_2, t_3]$.

Over $[t_2, t_3]$ S_j, $j = -1,\ldots,2$, are nonzero based on their contributions from N_6, N_7, N_8, and N_9, so

$$1 \equiv \sum_{j=-1}^{2} S_j(t)$$
$$\equiv \sum_{i=2}^{6} c_{-1,i}N_i(t) + \sum_{i=3}^{9} c_{0,i}N_i(t) + \sum_{i=6}^{12} c_{1,i}N_i(t) + \sum_{i=9}^{15} c_{2,i}N_i(t)$$
$$\equiv [c_{-1,6} + c_{0,6} + c_{1,6}]N_6(t) + [c_{0,7} + c_{1,7}]N_7(t)$$
$$+ [c_{0,8} + c_{1,8}]N_8(t) + [c_{0,9} + c_{1,9} + c_{2,9}]N_9(t).$$

These equations lead to Equations 5 through 8 when $i = 2$ is substituted into those equations. Thus, the normalization conditions over $[t_2, t_3]$ are the same as over the interior intervals.

Now consider $[t_1,t_2]$. S_j, $j = -2,\ldots,1$, are nonzero based on their contributions from N_3, N_4, N_5, and N_6, so

$$1 \equiv \sum_{j=-2}^{1} S_j(t)$$
$$\equiv \sum_{i=1}^{3} c_{-2,i}N_i(t) + \sum_{i=2}^{6} c_{-1,i}N_i(t) + \sum_{i=3}^{9} c_{0,i}N_i(t) + \sum_{i=6}^{12} c_{1,i}N_i(t)$$
$$\equiv [c_{-2,3} + c_{-1,3} + c_{0,3}]N_3(t) + [c_{-1,4} + c_{0,4}]N_4(t)$$

$$+ [c_{-1,5} + c_{0,5}]N_5(t) + [c_{-1,6} + c_{0,6} + c_{1,6}]N_6(t).$$

Again, these equations lead to Equations 5 through 8 when i = 1 is substituted into those equations so the normalization conditions over $[t_1, t_2]$ are the same as over the interior intervals.

We are left with the interval $[t_0, t_1]$, which has functions S_j, j = -3,...,0 nonzero with contributions from N_0, N_1, N_2, and N_3.

$$1 \equiv \sum_{j=-3}^{0} S_j(t)$$
$$\equiv c_{-3,0}N_0(t) + \sum_{i=1}^{3} c_{-2,i}N_i(t) + \sum_{i=2}^{6} c_{-1,i}N_i(t) + \sum_{i=3}^{9} c_{0,i}N_i(t)$$
$$\equiv c_{-3,0}N_0(t) + c_{-2,1}N_1(t) + [c_{-2,2} + c_{-1,2}]N_2(t) + [c_{-2,3} + c_{-1,3} + c_{0,3}]N_3(t).$$

Several of the normalization conditions over this end interval are different. They are

$$1 = c_{-3,0} \qquad (43)$$
$$1 = c_{-2,1} \qquad (44)$$

The other two conditions are the same as Equations 7 and 8 with i = 0.

Equations 43 and 44 were already known from interpolation considerations, but the normalization constraints also ratify the validity of those values and interpolation constraints.

Next we must set up the linear system to solve for the coefficients by looking at the geometric constraints. We note that S_{-3} is already completely determined, and it has no additional geometric constraints. Applying Equations 1 to S_{-1} at t_{-1} and t_{-2} result in the following equations:

at t_2:

$$c_{-1,6}\left(-\frac{3}{t_3-t_2}\right) = \mu_2\left(\frac{3}{t_2-t_1}\right)(c_{-1,6}-c_{-1,5})$$

$$c_{-1,6}\left(\frac{6}{(t_3-t_2)^2}\right) = \mu_2^2\left(\frac{6}{(t_2-t_1)^2}\right)(c_{-1,6} - 2c_{-1,5} + c_{-1,4}) + \nu_2\left(\frac{3}{t_2-t_1}\right)(c_{-1,6}-c_{-1,5})$$

at t_1:

$$(c_{-1,4}-c_{-1,3})\left(\frac{3}{t_2-t_1}\right) = \mu_1\left(\frac{3}{t_1-t_0}\right)(c_{-1,3}-c_{-1,2})$$

$$\left(\frac{6}{(t_2-t_1)^2}\right)(c_{-1,5} - 2c_{-1,4} + c_{-1,3}) = \mu_1^2\left(\frac{6}{(t_1-t_0)^2}\right)(c_{-1,3} - 2c_{-1,2}) + \nu_1\left(\frac{3}{t_1-t_0}\right)(c_{-1,3}-c_{-1,2})$$

Thus the geometric tension constraints give

$$\Delta_{-1,3} = \lambda_1 \Delta_{-1,2}$$
$$\Delta_{-1,4} - \Delta_{-1,3} = \lambda_1^2(\Delta_{-1,2} - c_{-1,2}) + \eta_1 \Delta_{-1,2}$$
$$-c_{-1,6} = \lambda_2 \Delta_{-1,5}$$
$$c_{-1,6} = \lambda_2^2(\Delta_{-1,5} - \Delta_{-1,4}) + \eta_2 \Delta_{-1,5}$$

Solving this system gives:

$$\Delta_{-1,5} = -\frac{c_{-1,6}}{\lambda_2}$$

$$\Delta_{-1,4} = -\frac{\Lambda_2}{\lambda_2^3} c_{-1,6}$$

$$\Delta_{-1,3} = \frac{\lambda_1^3}{\Lambda_1} c_{-1,2} - \frac{\Lambda_2}{\Lambda_1} \frac{\lambda_1}{\lambda_2^3} c_{-1,6}$$

$$\Delta_{-1,2} = \frac{(\lambda_1)^2}{\Lambda_1} c_{-1,2} - \frac{\Lambda_2}{\Lambda_1} \frac{1}{(\lambda_2)^3} c_{-1,6}$$

Since

$$c_{-1,6} = c_{-1,2} + \Sigma_{j=2}^{5} \Delta_{-1,j},$$

$$c_{-1,6} = \frac{\lambda_2^3(\Lambda_1+\lambda_1^3+\lambda_1^2)}{L_1} c_{-1,2}$$

We may substitute that in the equations above to determine $\Delta_{-1,j}$, $j = 2,\ldots,5$ in terms of $c_{-1,2}$. Then we can solve for $c_{-1,j}$, $j = 3,\ldots,6$, in terms of $c_{-1,2}$ by using $c_{-1,j} = c_{-1,2} + \Sigma_{i=2}^{j-1} \Delta_{-1,i}$. One obviously first solves for $c_{-1,3}$ and then for $c_{-1,4}$, etc., to get:

$$c_{-1,3} = [1 + \frac{\lambda_1^2}{\Lambda_1} - \frac{\Lambda_2}{\Lambda_1} \frac{\lambda_1^3+\lambda_1^2+\Lambda_1}{L_1}]c_{-1,2}$$

$$c_{-1,4} = \frac{\lambda_1^3+\lambda_1^3+\Lambda_1}{\Lambda_1} [1 - \frac{\Lambda_2(1+\lambda_1)}{L_1}]c_{-1,2}$$

$$c_{-1,5} = \frac{(\lambda_1^3+\lambda_1^2+\Lambda_1)(\lambda_2^2+\lambda_2^3)}{L_1} c_{-1,2}$$

$$c_{-1,6} = \frac{\lambda_2^3(\lambda_1^3+\lambda_1^2+\Lambda_1)}{L_1} c_{-1,2}$$

We now use the normalization constraints on $[t_2, t_3]$, in particular Equation 5, and the values of $c_{0,6}$, $c_{1,6}$ to get

$$c_{-1,6} = 1 - c_{0,6} - c_{1,6} = 1 - \left(1 - \frac{\Lambda_3}{L_2} - \lambda_2^3 \frac{\Lambda_1}{L_1}\right) - \left(\frac{\Lambda_3}{L_2}\right)$$

$$= \lambda_2^3 \frac{\Lambda_1}{L_1} \tag{45}$$

Since all the rest of the coefficients are in terms of $c_{-1,2}$ we should solve for $c_{-1,2}$ next.

$$c_{-1,2} = \frac{L_1}{\lambda_2^3(\lambda_1^3+\lambda_1^2+\Lambda_1)} c_{-1,6} = \frac{L_1}{\lambda_2^3(\lambda_1^3+\lambda_1^2+\Lambda_1)} \lambda_2^3 \frac{\Lambda_1}{L_1}$$

$$= \frac{\Lambda_1}{(\lambda_1^3+\lambda_1^2+\Lambda_1)} \tag{46}$$

Substituting in the appropriate equations gives

$$c_{-1,3} = \frac{\Lambda_1+\lambda_1^2}{\lambda_1^3+\lambda_1^2+\Lambda_1} - \frac{\Lambda_2}{L_1} \tag{47}$$

$$c_{-1,4} = (\lambda_2^3 + \lambda_2^2 + \Lambda_2) \frac{\Lambda_1}{L_1} \tag{48}$$

$$c_{-1,5} = (\lambda_2^3 + \lambda_2^2) \frac{\Lambda_1}{L_1} \tag{49}$$

We continue towards the boundary by now considering $S_{-2}(t)$. Since it has exactly one additional geometric constraint, at t_1, we substitute into Equations 1 to get

$$-\frac{3}{t_2-t_1}c_{-2,3}=\mu_1\frac{3}{t_1-t_0}(c_{-2,3}-c_{-2,2})$$

$$\frac{6}{(t_2-t_1)^2}c_{-2,3}=\mu_1^2\frac{6}{(t_1-t_0)^2}(c_{-2,1}-2c_{-2,2}+c_{-2,3})+\nu_1\frac{3}{t_1-t_0}(c_{-2,3}-c_{-2,2})$$

From these equations we get that

$$\Delta_{-2,1}=-\frac{\Lambda_1}{\lambda_1^3}c_{-2,3} \qquad \Delta_{-2,2}=-\frac{c_{-2,3}}{\lambda_1}$$

Since $c_{-2,1}=1$, and $c_{-2,3}-c_{-2,1}=\Delta_{-2,1}+\Delta_{-2,2}$, we can solve to get

$$c_{-2,3}=\frac{\lambda_1^3}{\lambda_1^3+\lambda_1^2+\Lambda_1} \tag{50}$$

$$c_{-2,2}=\frac{\lambda_1^3+\lambda_1^2}{\lambda_1^3+\lambda_1^2+\Lambda_1} \tag{51}$$

To check for consistency, we can find these coefficients by using the normalization constraints over $[t_1,t_2]$, just as we solved for the coefficients for S_{-1}. In particular,

$$\begin{aligned} c_{-2,3} &= 1-c_{0,3}-c_{-1,3} \\ &= 1-\left(\frac{\Lambda_2}{L_1}\right)-\left(\frac{\Lambda_1+\lambda_1^2}{\lambda_1^3+\lambda_1^2+\Lambda_1}-\frac{\Lambda_2}{L_1}\right) \\ &= 1-\frac{\Lambda_1+\lambda_1^2}{\lambda_1^3+\lambda_1^2+\Lambda_1} \\ &= \frac{\lambda_1^3}{\lambda_1^3+\lambda_1^2+\Lambda_1} \end{aligned}$$

which is the same answer as obtained using the normalization conditions from the other direction. Thus, the functions are well defined and sum to one.

Symmetric geometric constraints and normalization conditions at the upper end of the knot value domain yield

$$c_{n,3n+3}=1 \tag{52}$$

$$c_{n-1,3(n-1)+3} = \frac{1}{1+\lambda_n+\Lambda_n} \tag{53}$$

$$c_{n-1,3(n-1)+4} = \frac{1+\lambda_n}{1+\lambda_n+\Lambda_n} \tag{54}$$

$$c_{n-1,3(n-1)+5} = 1 \tag{55}$$

$$c_{n-2,3(n-2)+3} = \frac{\Lambda_n}{L_{n-1}} \tag{56}$$

$$c_{n-2,3(n-2)+4} = \frac{\Lambda_n}{L_{n-1}}(1 + \lambda_{n-1}) \tag{57}$$

$$c_{n-2,3(n-2)+5} = \frac{\Lambda_n}{L_{n-1}}(1 + \lambda_{n-1} + \Lambda_{n-1}) \tag{58}$$

$$c_{n-2,3(n-2)+6} = 1 - \frac{\Lambda_{n-1}}{L_{n-1}}\lambda_n^3 - \frac{1}{1+\lambda_n+\Lambda_n} \tag{59}$$

$$c_{n-2,3(n-2)+7} = \frac{\Lambda_n}{1+\lambda_n+\Lambda_n} \tag{60}$$

Hence, in the general case the coefficients are known. The following figures illustrate the use of tension with open polygons. In the first of the two figures , the knot sequence is uniform open, in the last of the two figures the sequence is nonuniform open.

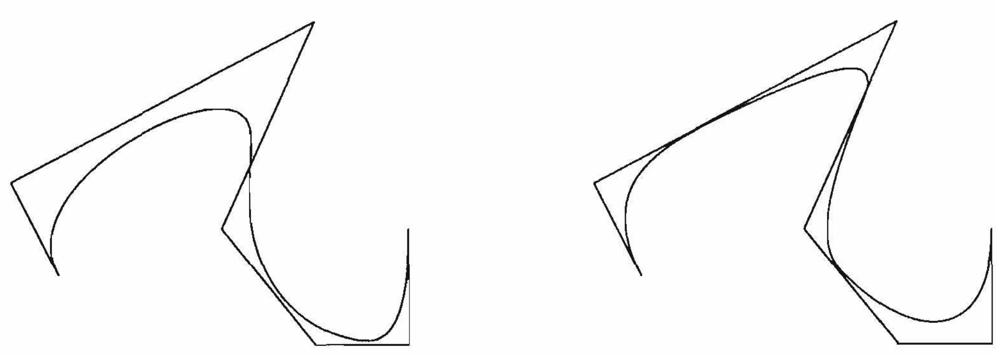

Figure 5-1: a. $\mu=\{.5, .5\}$, $\nu=\{0, 0\}$ b. $\mu=\{4., 4.\}$, $\nu=\{6., 6.\}$
knot vector={0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4}

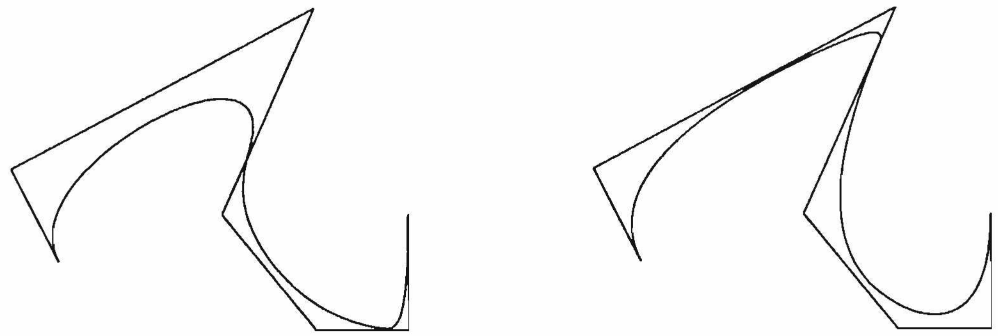

Figure 5-2: a. $\mu=\{.5, .5\}$, $\nu=\{0, 0\}$ b. $\mu=\{4., 4.\}$, $\nu=\{6., 6.\}$
knot vector=$\{0, 0, 0, 0, 3, 4, 7, 8, 8, 8, 8\}$

5.1. Open tensioned splines with small numbers of internal knots

The section has presumed that one need only assume two or more internal knots for these coefficients to hold. Here we determine the effects of having just one internal knot, and also justify the above stated assumption.

When n = 1, there are only three breakpoints in total, $\{t_0, t_1, t_2\}$, and just five blending functions $\{S_{-3}, S_{-2}, S_{-1}, S_0, S_1\}$. We consider the modified geometric constraints on these five equations. S_{-3}'s constraints are unchanged as are S_{-2}'s. Since 0 = n-1, and 1 = n, S_0 and S_1 satisfy the geometric constraints of S_{n-1} and S_n in the discussion above. Thus, these functions are combinations of the same B-splines as in the general case.

$$S_{-3}(t) = c_{-3,0}N_0(t)$$
$$S_{-2}(t) = \Sigma_{i=1}^{3} c_{-2,i}N_i(t)$$
$$S_0(t) = \Sigma_{i=3}^{5} c_{0,i}N_i(t)$$
$$S_1(t) = c_{1,6}N_6(t)$$

However, S_{-1} must serve dually as S_{-1}, a lower "boundary" function, and as S_{n-2}, an upper "boundary" function. The geometric conditions on it become that

S_{-1}
- has support on $[t_0, t_2]$,
- is C_1 at t_0,
- is C_1 at t_2,
- satisfies Equations 1 at t_1

Using those conditions

$$S_{-1}(t) = \Sigma_{i=2}^{4} c_{-1,i}N_i(t).$$

Once again, the normalization conditions on the intervals give,

over $[t_0, t_1]$:

$$1 \equiv S_{-3}(t) + S_{-2}(t) + S_{-1}(t) + S_0(t)$$
$$\equiv c_{-3,0}N_0(t) + c_{-2,1}N_1(t) + [c_{-2,2} + c_{-1,2}]N_2(t) + [c_{-2,3} + c_{-1,3} + c_{0,3}]N_3(t)$$

over $[t_1, t_2]$:

$$1 \equiv S_{-2}(t) + S_{-1}(t) + S_0(t) + S_1(t)$$
$$\equiv [c_{-2,3} + c_{-1,3} + c_{0,3}]N_3(t) + [c_{-1,4} + c_{0,4}]N_4(t) + c_{0,5}N_5(t) + c_{1,6}N_6(t).$$

and finally, that

$$1 = c_{-3,0}$$
$$1 = c_{-2,1}$$
$$1 = c_{-2,2} + c_{-1,2}$$
$$1 = c_{-2,3} + c_{-1,3} + c_{0,3}$$
$$1 = c_{-1,4} + c_{0,4}$$
$$1 = c_{0,5}$$
$$1 = c_{1,6}$$

We see that these conditions are identical to the first and last interval conditions in the general case, using $n = 1$. Since the geometric conditions on S_{-3} are unchanged and $c_{-3,0} = 1$ is the same as the general case, S_{-3} is the same. Similarly, the geometric conditions on S_{-2}, S_0, and S_1 are unchanged and from the fact that each has one coefficient unchanged, we can deduce that all the coefficients are unchanged and thus that the functions S_{-3}, S_{-2}, S_0, and S_1 are all identically the same. However, the function S_{-1} has a different definition, so we cannot use the same reasoning to obtain its coefficients. Since all the other functions are known, S_{-1} is completely determined by the normalization conditions.

Using the normalization equations and the known values for the coefficients of S_{-2} and S_0 gives

$$c_{-1,2} = 1 - c_{-2,2} = 1 - \frac{\lambda_1^3+\lambda_1^2}{\lambda_1^3+\lambda_1^2+\Lambda_1} \quad \text{by Equation 50}$$
$$= \frac{\Lambda_1}{\lambda_1^3+\lambda_1^2+\Lambda_1}$$

$$c_{-1,4} = 1 - c_{0,4} = 1 - \frac{1+\lambda_1}{1+\lambda_1+\Lambda_1}$$
$$= \frac{\Lambda_1}{1+\lambda_1+\Lambda_1}$$

$$c_{-1,3} = 1 - c_{-2,3} - c_{0,3} = 1 - \frac{\lambda_1^3}{\lambda_1^3+\lambda_1^2+\Lambda_1} - \frac{1}{1+\lambda_1+\Lambda_1}$$
$$= \frac{\Lambda_1(\lambda_1+\lambda_1^2+\Lambda_1)}{(1+\lambda_1+\Lambda_1)(\lambda_1^3+\lambda_1^2+\Lambda_1)}$$

We can check that these functions satisfy the geometric constraints at t_1. S_{-1} must satisfy

$$\Delta_{-1,3} = \lambda_1\Delta_{-1,2}$$

$$-c_{-1,4} - \Delta_{-1,3} = \lambda_1^2(\Delta_{-1,2} - c_{-1,2}) + \eta_1\Delta_{-1,2}$$

$$c_{-1,4} - c_{-1,2} = \Delta_{-1,3} + \Delta_{-1,2}.$$

Solving for the other coefficients in terms of $c_{-1,2}$

$$c_{-1,3} = \frac{\lambda_1+\lambda_1^2+\Lambda_1}{1+\lambda_1+\Lambda_1} c_{-1,2}$$

$$c_{-1,4} = \frac{\lambda_1^2+\lambda_1^3+\Lambda_1}{1+\lambda_1+\Lambda_1} c_{-1,2}$$

Using the normalization arrived at value for $c_{-1,2}$ in the above two equations gives the same answer as using just the normalization equations for all the coefficients, thus verifying consistency.

If there are two internal knots, then there are six blending functions which turn out to be the "boundary" blending functions of the general case. In that case, solving the geometric constraints gives the same Δ's as the general case, and the normalization constraints are the same. Hence, all the coefficients are identical to those arrived at in the general case for the boundary functions.

If there are more than two internal knots, then one starts to arrive at the "interior" functions. Since their geometric constraints are all the same as for the general case, and the normalization constraints are the same, and the boundary functions are all the same, these functions are the same, and the problem is solved.

Note that the only special case function for the open conditions occur when there is just a single internal knot. For all other knot configurations, the blending functions are standard.

6. Completeness of the Representation

Remember that $T_p = T_{t,\mu,v,p} = \{S_j(t)\}$. The "openness" or "floatingness" is determined by the value of p, where p = f for floating and p = o, for open. Also the span(T_p) = $S_p = S(T_p) = \{ \sum P_jS_j(t)$: $P_j \in \mathbf{R}^3$ and $S_j \in T_p\}$. It is clear that S_p is contained in the space $CCS_{t,\mu,v}$ defined with appropriate end conditions. It is now appropriate to ask if

- *CCS* is also contained in S_p, that is, is the span of T_p the whole space of curvature continuous piecewise cubic polynomials satisfying Equations 1?
- Is T_p a basis for this space? That is, are the functions $\{S_j(t)\}$ linearly independent?
- Finally, are these functions minimal support functions with the required geometric properties?

We shall investigate those questions in this section.

Suppose β is any curvature continuous piecewise cubic polynomial over the domain of the space. Since the functions $\{N_i\}$ form a basis for all parametrically $C^{(0)}$ piecewise cubic polynomials, there exists a unique sequence of vectors $\{Q_i\}$ such that

$$\beta(t) = \Sigma\, Q_i N_i(t).$$

Let us first show the independence of the elements of S_p. Over $[t_3, t_{n-2}]$

$$0 \equiv \Sigma\, P_j S_j(t)$$
$$\equiv \Sigma_{i=9}^{3n-6}\, Q_i N_i(t)$$

Since the N_i's form a basis, this means that $Q_i = 0$ for all i. Thus, for each k,

$$Q_{3k+1} = 0 = \sum_{i=k-2}^{k-1} c_{i,3k+1} P_i,$$
$$Q_{3k+2} = 0 = \sum_{i=k-2}^{k-1} c_{i,3k+2} P_i.$$

Solving for P_{k-2} and P_{k-1} then becomes a question of solving this homogeneous system of two equations in two unknowns. The four values of $c_{i,m}$, i = k-2, k-1, m = 3k+1, 3k+2, are not all zero, and since

$$\frac{c_{k-2,3k+1}}{c_{k-2,3k+2}} = \frac{\Lambda_{k+1}+\lambda_{k+1}{}^3+\lambda_{k+1}{}^2}{\lambda_{k+1}{}^3+\lambda_{k+1}{}^2}$$
$$\frac{c_{k-1,3k+1}}{c_{k-1,3k+2}} = \frac{\lambda_k+1}{\Lambda_k+\lambda_k+1}$$

the ratios of the coefficients of the respective P_i's are different. That means that one equation cannot be a constant scalar multiple of the other, and the equations are linearly independent. The only solution to a homogeneous system of m equations in m unknowns of rank m is the trivial solution, that is, $P_i = 0$, i = k-2, k-1. Since this follows true for all k, we are done. The open end conditions require slightly different ratios near the ends, but the proof is analogous. Hence, the functions $\{S_j\}$ are independent over this space.

Now, consider the case when p = f, that is, the space S_f with floating end conditions. Since the functions T_f are independent, the space S_f has dimension n-2, the number of elements in the set T_f. Since S_f is contained in $CCS[t_3, t_{n-2}]$, we need only show that the dimension of $CCS[t_3, t_{n-2}]$ is n-2 to have equality.

Over $[t_3, t_{n-2}]$, there are n-5 distinct intervals partitioned by the knot set t_j, j = 4, ..., n-3. Suppose it is desired to construct a curvature continuous piecewise cubic function β. Let us count the degrees of freedom. Suppose β is completely specified on $[t_i, t_{i+1}]$. At t_{i+1} it must be continuous and given fixed values of μ_{i+1} and ν_{i+1}, it must satisfy Equations 1, and hence if β is specified by $\beta_i = a_{i,0} + a_{i,1}(t-t_i) + a_{i,2}(t-t_i)^2 + a_{i,3}(t-t_i)^3$ over $[t_i, t_{i+1}]$, then the equations require that

$$a_{i+1,0} = \sum_{j=0}^{3} a_{i,j}(t_{i+1}-t_i)^j, \text{ for continuity}$$

$$a_{i+1,1} = \mu_{i+1} \sum_{j=0}^{2} (j+1)a_{i,j+1}(t_{i+1}-t_i)^j$$

$$2a_{i+1,2} = \mu_{i+1}^2 \sum_{j=0}^{1} (j+2)(j+1)a_{i,j+2}(t_{i+1}-t_i)^j + \nu_{i+1} \sum_{j=0}^{2} (j+1)a_{i,j+1}(t_{i+1}-t_i)^j$$

We see that three of the four coefficients for the next interval are completely specified, and the last remains completely unconstrained. Hence, counting degrees of freedom starting with i = 3,

interval	degrees of freedom
$[t_3,t_4]$	4
$[t_4,t_5]$	1
$[t_5,t_6]$	1
.	.
.	.
.	.
$[t_{n-3},t_{n-2}]$	1

Since there are n-5 intervals, one of which has 4 degrees of freedom, and n-6 of which have just 1 degree of freedom, $CCS[t_3, t_{n-2}]$ has dimension n-2, the dimension of S_f. Hence $S_f = CCS[t_3, t_{n-2}]$.

For the open case, since T_o is independent, S_o has dimension n+4, the number of elements of T_o. Once again, if we show that the dimension of $CCS[t_0, t_{n+1}]$ is n+4, we are done. However, exactly the same arguments as above, except here there are n+1 intervals, on which to apply it. It follows directly that the dimension of $CCS[t_0, t_{n+1}]$ is n+1, and $S_o = CCS[t_0, t_{n+1}]$.

Finally we consider the question of minimal local support. Suppose there is a function T ∈ *CCS* which has a smaller support than the elements of T_p. Suppose first that this function T does not meet the boundary. By the properties of polynomials and continuity, T must have support over $[t_k, t_{k+m}]$ where m < 4. Satisfaction of Equations 1 requires that T is $C^{(2)}$ at t_k and at t_{k+m}. Also, that means that $T(t) = \sum_{i=3k+3}^{3k+3(m-1)} d_i N_i(t)$. If m = 1 or m = 2 it is impossible to fulfill the geometric constraints unless $T(t) \equiv 0$, the trivial function. If m = 3, there would be just four B-splines that would enter into the definition for T. However, from the geometric constraints at t_{k+1} and t_{k+2}, one would get four homogeneous linear equations in four independent unknowns. The only solution again is the trivial solution. Thus, if T is not near the boundary, its support must include $[t_k, t_{k+4}]$. Now suppose that T has support over $[t_0, t_m]$ where m < 4, that is T is a boundary function. We have S functions with exactly those conditions. We must show that T can have no higher continuity class than S_{m-4} at each of its knots. $T(t) = \sum_{i=0}^{3(m-1)} d_i N_i(t)$, where the upper end is constrained by the geometric continuity conditions, leading to $C^{(2)}$ continuity at t_m. If m = 1, T(t) is just a scaled version of $S_{-3}(t)$ so that is consistent. For m > 1, if $d_0 \neq 0$, then T(t) can be decomposed as a sum of S_{-3} and another function in the class and is again not minimal. Thus, for m > 1, $T(t) = \sum_{i=1}^{3(m-1)} d_i N_i(t)$. The continuity constraints on T, however, at t_0 now match the constraints for S_{-2}, and the support of T contains the support of S_{-2}. Thus if $d_1 \neq 0$, then T(t) can be decomposed as a scaled version of a sum of S_{-2} and another function in the class, and is again not minimal. Hence, if m = 2 and T is a scaled version of S_{-2}, or else $d_1 = 0$. If m = 3, the final class then has form $T(t) = \sum_{i=2}^{6} d_i N_i(t)$, with continuity constraints matching the constraints on S_{-1}. Thus, only one degree of freedom remains, the normalization factor, and T then becomes a scaled version of S_{-1}. It is impossible then to have a function in *CCS* with support smaller than those defined in T_p. In that sense the basis is minimal.

7. Convex Hull and Variation Diminishing Properties

The discussion earlier in the paper has already indicated why these two properties are important and useful to any design scheme. Below we show that the curve forms which use the blending functions which we have developed, the $S_j(t)$, lie within the convex hull of the coefficient polygon and are a variation diminishing approximation to it.

7.0.1. Convex Hull Properties

We make the assumption that $\Lambda_j = \lambda_j^2 + \lambda_j + v_j > 0$ for all j. Note that λ_j must be greater than 0 to have geometric first derivative continuity. Hence, we require that $v_j > -(\lambda_j^2 + \lambda_j)$.

Theorem 10: The curve $\gamma(t) = \sum_{i=0}^{n-3} P_i S_i(t)$ lies in the convex hull of the points $\{P_i, i=0,\ldots,n-3\}$.

This is the result for the floating end conditions. The statement and for the open end conditions is obvious, and the proof is closely analogous to the following proof for the floating end conditions. In the floating case, the domain for the curve is $[t_3, t_{n-2}]$.

Proof: The curve lies in the convex hull of the points $\{P_i\}$ if

1. Each $S_i \geq 0$
2. $\sum_i S_i(t) \equiv 1$.

We have used condition 2. to arrive at the normalization conditions, and hence it is automatically satisfied. In order to have the convex hull property we must show only that $c_{i,j} \geq 0$, for all i and j.

It is easy to see from equations in 42 that for a given i, $c_{i,\,3i+k} > 0$, for k = 3, 4, 5, 7, 8, 9. Hence, we must show that $c_{i,3i+6} > 0$.

By equations 11 through 14,

$$\begin{aligned} c_{j-1,3(j-1)+8} + c_{j,\,3j+5} &= 1 \\ c_{j-1,3(j-1)+9} + c_{j,3j+6} + c_{j+1,3(j+1)+3} &= 1 \\ c_{j,3j+7} + c_{3(j+1)+4} &= 1 \end{aligned}$$

Now

$$\begin{aligned} \gamma(t) &= \sum_{i=0}^{n-3} P_i S_i(t) \\ &= \sum_{i=0}^{n-3} P_i \sum_{j=3i+3}^{3i+9} c_{i,j} N_j(t) \\ &= \sum_{j=9}^{3(n-2)} N_j(t) \sum_{ce(j/3)-3}^{fl(j/3)-1} c_{i,j} P_i \\ &= \sum_{j=9}^{3(n-2)} Q_j N_j(t), \end{aligned}$$

where

$$ce(a) = \min\{k: k \geq a \text{ and } k \text{ is an integer}\} \text{ and } fl(a) = \max\{k: k \leq a \text{ and } k \text{ is an integer}\}.$$

Thus,

$$Q_{3k} = \sum_{i=k-3}^{k-1} c_{i,3k} P_i$$

$$Q_{3k+1} = \sum_{i=k-2}^{k-1} c_{i,3k+1} P_i$$

$$Q_{3k+2} = \sum_{i=k-2}^{k-1} c_{i,3k+2} P_i$$

$$k = 3, \ldots, (n-3), \text{ and}$$

$$Q_{3(n-2)} = \sum_{i=n-5}^{k-3} c_{i,3(n-2)} P_i$$

Now,

$$c_{k-2,3(k-2)+7} = \frac{\Lambda_k}{L_k} (\lambda_{k+1}^3 + \lambda_{k+1}^2 + \Lambda_{k+1}) > 0,$$

$$c_{k-1,3(k-1)+4} = \frac{\Lambda_{k+1}}{L_k} (\lambda_k + 1) > 0$$

and

$$c_{k-2,3(k-2)+7} + c_{k-1,3(k-1)+4} = 1.$$

Hence, Q_{3k+1} is a convex combination of P_{k-2} and P_{k-1}, and lies on the line segment between them.

Similarly,

$$c_{k-3,3(k-3)+8} = \frac{\Lambda_{k-1}}{L_{k-1}} (\lambda_k^3 + \lambda_k^2) > 0,$$

$$c_{k-2,3(k-2)+5} = \frac{\Lambda_k}{L_{k-1}} (\Lambda_{k-1} + \lambda_{k-1} + 1) > 0$$

and

$$c_{k-3,3(k-3)+8} + c_{k-2,3(k-2)+5} = 1.$$

Thus, $Q_{3(k-1)+2} = Q_{3k-1}$ is a convex combination of P_{k-3} and P_{k-2} and lies on the line segment between them.

Finally, since $c_{k-3,3(k-3)+9} > 0$, $c_{k-1,3(k-1)+3} > 0$, and $c_{k-3,3(k-3)+9}\ c_{k-2,3(k-2)+6} + c_{k-1,3(k-1)+3} =$ 1, we need only have that $c_{k-2,3(k-2)+6} > 0$ to have Q_{3k} a convex combination of P_{k-3}, P_{k-2}, and P_{k-1}. We show positivity of $c_{k-2,3(k-2)+6}$.

$$c_{k-3,3(k-3)+9} = \lambda_{(k-3)+3}^3 \frac{\Lambda_{(k-3)+2}}{L_{(k-3)+2}} = \lambda_k^3 \frac{\Lambda_{k-1}}{L_{k-1}} = \frac{\lambda_k^3}{\lambda_k^3 + \lambda_k^2} \{\lambda_k^3 + \lambda_k^2\} \frac{\Lambda_{k-1}}{L_{k-1}}$$

$$= \frac{\lambda_k^3}{\lambda_k^3 + \lambda_k^2} c_{k-3,3(k-3)+8}$$

$$= \frac{\lambda_k}{\lambda_k+1} c_{k-3,3(k-3)+8}$$

and

$$c_{k-1,3(k-1)+3} = \frac{\Lambda_{(k-1)+2}}{L_{(k-1)+1}} = \frac{\Lambda_{k+1}}{L_k} = \frac{1}{\lambda_k+1}(\lambda_k+1)\frac{\Lambda_{k+1}}{L_k}$$
$$= \frac{1}{\lambda_k+1} c_{k-1,3(k-1)+4}.$$

Thus, using the normalizing equation,

$$c_{k-2,3(k-2)+6} = 1 - c_{k-3,3(k-3)+9} - c_{k-1,3(k-1)+3}$$
$$= \frac{\lambda_k+1}{\lambda_k+1} - c_{k-3,3(k-3)+9} - c_{k-1,3(k-1)+3}$$
$$= \frac{\lambda_k}{\lambda_k+1} - c_{k-3,3(k-3)+9} + \frac{1}{\lambda_k+1} - c_{k-1,3(k-1)+3}$$
$$= \frac{\lambda_k}{\lambda_k+1}(1 - c_{k-3,3(k-3)+8}) + \frac{1}{\lambda_k+1}(1 - c_{k-1,3(k-1)+4}).$$

Since $0 < c_{k-3,3(k-3)+8}, c_{k-1,3(k-1)+4} < 1$, $c_{k-2,3(k-2)+6}$ is a convex combination of numbers between zero and one and hence must also lie strictly between zero and one. Thus, each of the coefficients of the N_j's used to form S_k is nonnegative, and by choice of the normalizing equations, $\sum S_j \equiv 1$. Hence, the curve $\gamma(t)$ lies in the convex hull of the vertices $\{P_i\}$.

Corollary 11: For $t \in [t_j, t_{j+1}]$, $\gamma(t)$ is in the convex hull of $\{P_{j-3}, P_{j-2}, P_{j-1}, P_j\}$.
This follows from the convex hull property and the localness of the LT-spline representation.

In order to approach the variation diminishing issue, we look at one more consequence of Theorem 10.

Corollary 12: Q_{3k} is a convex combination of Q_{3k-1} and Q_{3k+1}.
This result must in fact be true for the curve to be geometrically derivative continuous. We show the result using the theorem above.

Proof:

$$Q_{3k} = P_{k-3}c_{k-3,3(k-3)+9} + P_{k-2}c_{k-2,3(k-2)+6} + P_{k-1}c_{k-1,3(k-1)+3}$$
$$= \frac{\lambda_k}{\lambda_k+1} c_{k-3,3(k-3)+8}P_{k-3}$$
$$+ \{\frac{\lambda_k}{\lambda_k+1}(1- c_{k-3,3(k-3)+8}) + \frac{1}{\lambda_k+1}(1- c_{k-1,3(k-1)+4})\}P_{k-2}$$
$$+ \frac{1}{\lambda_k+1} c_{k-1,3(k-1)+4}P_{k-1}$$
$$= \frac{\lambda_k}{\lambda_k+1}(c_{k-3,3(k-3)+8}P_{k-3} + c_{k-2,3(k-2)+5}P_{k-2})$$
$$+ \frac{1}{\lambda_k+1}(c_{k-2,3(k-2)+7}P_{k-2} + c_{k-1,3(k-1)+4}P_{k-1})$$
$$= \frac{\lambda_k}{\lambda_k+1} Q_{3k-1} + \frac{1}{\lambda_k+1} Q_{3k+1}$$

We see that Q_{3k} *must* fall on the interior of the line segment joining Q_{3k-1} and Q_{3k+1}.

7.0.2. Variation Diminishing Property

It is well known that:

Theorem 13: If f(t) is a continuous curve, and v(t) is a continuous, piecewise linear interpolant to f(t),then v(t) is a variation diminishing approximation to f.

Theorem 14: Suppose $\{B_{i,k}(t)\}$ are the B-splines of order k defined over a knot vector τ, and $\gamma(t) = \sum Q_i B_{i,k}(t)$. Consider the piecewise linear curve $\theta(t)$ which is defined by connecting the points Q_i in order. Then $\gamma(t)$ is a variation diminishing approximation to $\theta(t)$.

We state without proof the transitivity of variation diminishing relationships:

Theorem 15: If v_1 is a variation diminishing approximation to f(t) and $v_2(t)$ is a variation diminishing approximation to $v_1(t)$, then $v_2(t)$ is a variation diminishing approximation to f(t).

Let $\tau^*_i = \frac{\tau_{i+1}+\tau_{i+2}+\tau_{i+3}}{3}$ and consider a piecewise linear function Q(t) such that $Q(\tau^*_{3k+1}) = Q_{3k+1}$ and $Q(\tau^*_{3k+2}) = Q_{3k+2}$. If this function is parametrized linearly, $Q(\tau^*_{3k}) = Q_{3k}$, also. The Q function is a continuous, piecewise linear interpolant to the P-control polygon of the curve $\gamma(t)$. Hence, Q is a variation diminishing approximation to the P-control polygon. But, since $\gamma(t)$ is just the B-spline curve with control polygon $\{Q_j\}$, $\gamma(t)$ is a variation diminishing approximation to the control polygon, which is the function Q(t). Hence, by the transitivity result, $\gamma(t)$ is a variation diminishing approximation to the control polygon $\{P_i\}$.

This whole section has been based on the premise that $\Lambda_j > 0$ for all j. (This implies that $L_j > 0$ for all j also.) When that premise is not true, it is possible that the curve will not lie in the convex hull, as the following example illustrates.

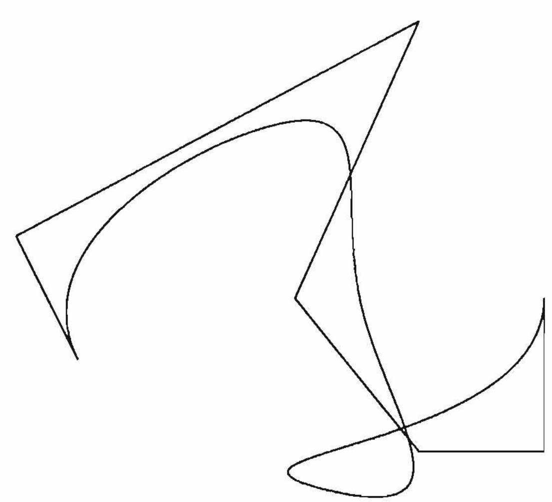

Figure 7-1: Uniforms tensions with negative Λ

8. Computing With Tensioned Splines

As we suggested in the introduction, one of the benefits of using B-splines to represent the local tensioned basis is that the computational benefits of B-splines may be invoked. In both this section and the next, we discuss techniques for using B-splines to help in computation and refinement of tensioned splines for rendering and hierarchical modelling.

Since all piecewise polynomials with knots at the t_j's can be represented as combinations of the N_i's, for all $\gamma(t)$,

$$\gamma(t) = \sum_{i=r}^{s} P_i S_i(t) = \sum_{j=3(r+3)}^{3s+3} Q_j N_j(t)$$

where r = 0, s = n-3 for the floating ends, $t \in [t_3, t_{n-2}]$, and r = -3, s = n for the open end conditions, $t \in [t_0, t_{n+1}]$,and where the Q's are computed by the following equations:

$$Q_0 = P_{-3}$$

$$Q_1 = P_{-2}$$

$$Q_2 = P_{-1}c_{-1,2} + P_{-2}c_{-2,2}$$
$$= P_{-2}[\frac{\lambda_1^3+\lambda_1^2}{\lambda_1^3+\lambda_1^2+\Lambda_1}] + P_{-1}[\frac{\Lambda_1}{\lambda_1^3+\lambda_1^2+\Lambda_1}]$$

$$Q_{3i} = P_{i-3}c_{i-3,3i} + P_{i-2}c_{i-2,3i} + P_{i-1}c_{i-1,3i}, \qquad i = 1,\ldots,n$$
$$= P_{i-2} + (P_{i-3} - P_{i-2})\lambda_i^3 \frac{\Lambda_{i-1}}{L_{i-1}} + (P_{i-1} - P_{i-2})\frac{\Lambda_{i+1}}{L_i}, \qquad i = 1,\ldots,n$$
$$= \frac{\lambda_i}{\lambda_i+1} Q_{3(i-1)+2} + \frac{1}{\lambda_i+1} Q_{3(i-1)+2}Q_{3i+1}, \qquad i=1,\ldots,n, \quad \text{by the proof of Corollary 12}$$

$$Q_{3i+1} = P_{i-2}c_{i-2,3i+1} + P_{i-1}c_{i-1,3i+1}, \qquad i = 1,\ldots,n$$
$$= P_{i-2}[(\Lambda_{i+1} + \lambda_{i+1}^3 + \lambda_{i+1}^2)\frac{\Lambda_i}{L_i}] + P_{i-1}[(\lambda_i + 1)\frac{\Lambda_{i+1}}{L_i}], \qquad i = 1,\ldots,n$$

$$Q_{3i+2} = P_{i-2}c_{i-2,3i+2} + P_{i-1}c_{i-1,3i+2}, i = 1,\ldots,n-1, \qquad i = 1,\ldots,n-1$$
$$= P_{i-2}[(\lambda_{i+1}^3 + \lambda_{i+1}^2)\frac{\Lambda_i}{L_i}] + P_{i-1}[(\Lambda_i + \lambda_i + 1)\frac{\Lambda_{i+1}}{L_i}], \qquad i=1,\ldots,n-1$$

$$Q_{3n+2} = P_{n-1}$$

$Q_{3n+3} = P_n$

Further, if we define end values for Λ_0, L_0, L_n, and Λ_{n+1}, then we can write all conditions, floating and open, in the same way. For

$$\Lambda_0 = 1, \tag{61}$$

$$L_0 = \lambda_1^3 + \lambda_1^2 + \Lambda_1, \tag{62}$$

$$L_n = 1 + \lambda_n + \Lambda_n, \tag{63}$$

$$\Lambda_{n+1} = 1, \tag{64}$$

$$\zeta_0 = 0; \quad \zeta_i = (1 + \lambda_i)\frac{\Lambda_{i+1}}{L_i}, \, i = 1,\ldots,n. \tag{65}$$

$$Q_0 = P_{-3} \qquad Q_{3n+3} = P_n \tag{66}$$

$$Q_{3i} = P_{i-2} + (P_{i-3} - P_{i-2})\lambda_i^3 \frac{\Lambda_{i-1}}{L_{i-1}} + (P_{i-1} - P_{i-2})\frac{\Lambda_{i+1}}{L_i}, \, i = 1, \ldots, n \tag{67}$$

$$= \frac{\lambda_i}{\lambda_i+1} Q_{3(i-1)+2} + \frac{1}{\lambda_i+1} Q_{3(i-1)+2} Q_{3i+1}, \qquad i=1,\ldots,n$$

$$Q_{3i+1} = (1-\zeta_i)\, P_{i-2} + \zeta_i P_{i-1}, \, i = 0,\ldots,n \tag{68}$$

$$Q_{3i+2} = (1 - \zeta_i - \frac{\Lambda_i \Lambda_{i+1}}{L_i})P_{i-2} + (\zeta_i + \frac{\Lambda_i \Lambda_{i+1}}{L_i})P_{i-1}, \, i = 0,\ldots,n \tag{69}$$

For floating end conditions just P_0 through P_{n-3} exist and just Q_9 through Q_{3n-6} are computed.

Thus we can see immediately two approaches for to evaluate points on a tensioned spline. First one can evaluate all the "c" coefficients in order to be able to evaluate each tensioned spline basis function and then evaluate a point on the curve by evaluating each of the tensioned basis functions at that domain value. Another approach is to selectively evaluate just those "c" coefficients necessary to evaluate the "Q" coefficients. The "Q" coefficients are the control points for the corresponding B-spline curve. One can evaluate points on the curve by evaluating this B-spline curve.

An even more efficient strategy is to directly evaluate the "Q" coefficients from the sequences of values of λ_i, Λ_i, and L_i. That is the recommended strategy.

Since there are many algorithms available for evaluating B-splines, this is a relatively simple procedure. An even more interesting procedure is to not evaluate points on the tensioned spline, but to simply render the corresponding B-spline curve using a refinement approach and the Oslo Algorithm for computation. This approach allows for adaptive rendering, putting more line segments where the curve has more complexity and curvature, and fewer segments where the curve has less curvature variation [7].

9. Interactive Modification of Tensioning Parameters

Consider the impact on all of these rendering techniques which occurs from modifying an element in the tensioned pair at a single particular knot. In particular, suppose that either μ_J or ν_J is modified. It is known that only S_{J-1}, S_{J-2}, and S_{J-3} are effected by such a change. If we again consider rendering by evaluating the "Q"'s, we need ask only which Q's change value. Since λ_J and Λ_J change value, so do L_{J-1} and L_J. Hence all "Q" coefficients which depend on those four scalars also change. In particular, just $Q_{3J-3}, Q_{3J-2},\ldots, Q_{3J+2}, Q_{3J+3}$ are modified. Just seven coefficients for the underlying B-splines. Hence one can implement computation of real time modifications by simply evaluating the seven appropriate Q's and then re-rendering the modified spans of the B-spline curve. Note that the curve is affected only over the interval $[t_{J-1}, t_{J+1}]$.

Note that the approach which uses the "Q" coefficients for rendering the tensioned splines carries a dual polygon to the "P" polygon. While this second polygon can be computed from the first, while having an interactive design session with the tensioned splines, interaction time is faster if only the modified "Q" coefficients must be recomputed for display. It is also faster to keep stored the values of λ_i, Λ_i, and L_i, $i = 0,\ldots,n$.

10. Knot Insertion with tensioned splines

Even with tensioned splines, there are times when it is desirable to add degrees of freedom. This issue remained unsolved in other formulations of the problem. Here we discuss the meaning of knot insertion for tensioned splines and provide a computational solution to finding the new design polygon

Given a sequences of real values and tension parameters, $t = \{t_0,\ldots,t_{n+1}\}$, $\mu = \{\mu_1,\ldots,\mu_n\}$, $\nu = \{\nu_1,\ldots,\nu_n\}$, consider another collection of vectors, $\underline{t}$, $\underline{\mu}$, $\underline{\nu}$, each with one additional element such that

$$\underline{t}_i = t_i,\ i = 0,\ldots,J$$
$$\underline{t}_i = t_{i-1},\ i = J+2,\ldots,n+2 \qquad\qquad t_J < \underline{t}_{J+1} < t_{J+1}$$

$$\underline{\mu}_i = \mu_i,\ i = 1,\ldots,J$$
$$\underline{\mu}_{J+1} = 1$$
$$\underline{\mu}_i = \mu_{i-1},\ i = J+2,\ldots,n+1$$

$$\underline{\nu}_i = \nu_i,\ i = 1,\ldots,J$$
$$\underline{\nu}_{J+1} = 0$$
$$\underline{\nu}_i = \nu_{i-1},\ i = J+2,\ldots,n+1$$

This means that

$$\underline{\lambda}_j = \begin{cases} \lambda_j, j \le J\text{-}1 \\ \lambda_{j-1}, j \ge J+3. \end{cases}$$

$$\underline{\Lambda}_j = \begin{cases} \Lambda_j, j \le J\text{-}1 \\ \Lambda_{j-1}, j \ge J+3 \end{cases}$$

$$\underline{L}_j = \begin{cases} L_j, j \le J\text{-}2 \\ L_{j-1}, j \ge J+3. \end{cases}$$

Initially, $\underline{\mu}_{J+1} = 1$ and $\underline{v}_{J+1} = 0$, and the space $CCS_{t,\mu,v}[t_0, t_{n+1}]$ and $CCS_{t,\mu,v}[t_3, t_{n-2}]$ are identical to $CCS_{\underline{t},\underline{\mu},\underline{v}}[t_0, t_{n+1}]$, and $CCS_{\underline{t},\underline{\mu},\underline{v}}[t_3, t_{n-2}]$, respectively. However, $\underline{\mu}_{J+1}$ and $\underline{v}_{J+1}$ can be modified, and this changes the tension space of the resulting curve So just as with design strategies for regular B-splines, one first represents the tensioned spline as one with an additional knot and tension parameters, but for which the tension parameters keep the required $C^{(2)}$ continuity at the new knot. After the new representation is found, these new tension values can also be adjusted, thus locally modifying the curve.

If one is given $\gamma(t) = \Sigma P_i S_i(t) \in CCS_{t,\mu,v}$, one wants to find the points $\underline{P}_j$ such that $\gamma(t) = \Sigma \underline{P}_j \underline{S}_j \in CCS_{\underline{t},\underline{\mu},\underline{v}}$. We shall use the computational ideas derived in Equations 68 and 69.

First, note that the knot sequence for generating the basis functions for the newly defined space is

$$\underline{\tau}_j = \begin{cases} \tau_j, & j \le 3J+3, \\ \underline{t}_{J+1}, & j = 3J+4, 3J+5, 3J+6, \\ \tau_{j-3}, & j > 3J+6 \end{cases}$$

and since $t_J < \underline{t}_{J+1} < t_{J+1}$ this means that

$$\underline{N}_j(t) = \begin{cases} N_j(t), j \le 3J\text{-}1, \\ N_{j-3}(t), j > 3J+6 \end{cases}$$

From discrete spline computations, however, it becomes evident that

$$\underline{Q}_j = \begin{cases} Q_j, j \le 3J, \\ Q_{j-3}, j \ge 3J+6 \end{cases}$$

Thus, one has to compute only five new values, $\underline{Q}_{3J+k}$, $k = 1,\ldots,5$. These values are easily determined using B-spline knot insertion algorithms.

However, to consider this function as a tensioned spline and to be able to interactively manipulate the values of $\underline{\mu}_{J+1}$ and $\underline{v}_{J+1}$, we must find the $\underline{P}_i$'s. Since the knot insertion only affects those tensioned basis functions overlapping the interval $[t_J, t_{J+1}]$, one new tensioned basis function is introduced, but it is clear that

$$\underline{S}_j(t) = \begin{cases} S_j(t), j \leq J\text{-}4 \\ S_{j-1}, j \geq J+2. \end{cases}$$

Hence, we need determine only $\underline{P}_{J-3}$, $\underline{P}_{J-2}$, $\underline{P}_{J-1}$, $\underline{P}_J$, and $\underline{P}_{J+1}$. We shall determine these new coefficient values by alternating inwards from the ends.

For appropriate scalars ζ,

$$\begin{aligned} Q_{3(J-2)+1} = \underline{Q}_{3(J-2)+1} &= (1 - \underline{\zeta}_{J-2})\underline{P}_{J-4} + \underline{\zeta}_{J-2}\underline{P}_{J-3} \\ &= (1 - \zeta_{J-2})P_{J-4} + \zeta_{J-2}\underline{P}_{J-3} \\ &= (1 - \zeta_{J-2})P_{J-4} + \zeta_{J-2}P_{J-3} \end{aligned}$$

By uniqueness of representation,

$$\underline{P}_{J-3} = P_{J-3}.$$

Analogously,

$$\begin{aligned} Q_{3(J+2)+1} = \underline{Q}_{3(J+3)+1} &= (1 - \underline{\zeta}_{J+3})\underline{P}_{J+1} + \underline{\zeta}_{J+3}\underline{P}_{J+2} \\ &= (1 - \zeta_{J+2})\underline{P}_{J+1} + \zeta_{J+2}P_{J+1} \\ &= (1 - \zeta_{J+2})P_J + \zeta_{J+2}P_{J+1} \end{aligned}$$

By uniqueness of representation,

$$\underline{P}_{J+1} = P_J.$$

We have discovered that there are only three coefficients in the new control polygon which need be computed, $\underline{P}_{J-2}$, $\underline{P}_{J-1}$, and $\underline{P}_J$.

$$\begin{aligned} Q_{3(J-1)+1} - Q_{3(J-1)+2} = \underline{Q}_{3(J-1)+1} - \underline{Q}_{3(J-1)+2} &= \frac{\underline{\Delta}_{J-1}\underline{\Delta}_J}{\underline{L}_{J-1}} [\underline{P}_{J-3} - \underline{P}_{J-2}] \\ &= \frac{\underline{\Delta}_{J-1}\underline{\Delta}_J}{\underline{L}_{J-1}} [P_{J-3} - \underline{P}_{J-2}] \end{aligned}$$

Reassociating yields,

$$\underline{P}_{J-2} = P_{J-3} + \frac{\underline{L}_{J-1}}{\underline{\Delta}_{J-1}\underline{\Delta}_J} [Q_{3(J-1)+2} - Q_{3(J-1)+1}]$$

In an analogous fashion,

$$\begin{aligned} Q_{3(J+1)+1} - Q_{3(J+1)+2} = \underline{Q}_{3(J+2)+1} - \underline{Q}_{3(J+2)+2} &= \frac{\underline{\Delta}_{J+2}\underline{\Delta}_{J+3}}{\underline{L}_{J+2}} [\underline{P}_J - \underline{P}_{J+1}] \\ &= \frac{\underline{\Delta}_{J+2}\underline{\Delta}_{J+3}}{\underline{L}_{J+2}} [\underline{P}_J - P_J] \end{aligned}$$

Reassociating yields,

$$\underline{P}_J = P_J + \frac{\underline{L}_{J+2}}{\underline{\Delta}_{J+2}\underline{\Delta}_{J+3}}[Q_{3(J+1)+1} - Q_{3(J+1)+2}].$$

The last unknown polygon point is $\underline{P}_{J-1}$, which can be found through two different applications of the same principles as above, using the already determined values for $\underline{P}_J$ or $\underline{P}_{J-2}$. Except for floating point error, they should give the same result.

$$\underline{P}_{J-1} = \begin{cases} \underline{P}_J + \dfrac{\underline{L}_{J+1}}{\underline{\Delta}_{J+1}\underline{\Delta}_{J+2}}[\underline{Q}_{3J+4} - \underline{Q}_{3J+5}] \\ \\ \underline{P}_{J-2} + \dfrac{\underline{L}_J}{\underline{\Delta}_J\underline{\Delta}_{J+1}}[\underline{Q}_{3J+2} - \underline{Q}_{3J+1}] \end{cases}$$

With the new P- polygon completely determined, the values of $\underline{\mu}_{J+1}$ and $\underline{\nu}_{J+1}$ can be interactively modified and the curve redrawn as suggested in the computational section. The example which follows shows knot insertion.

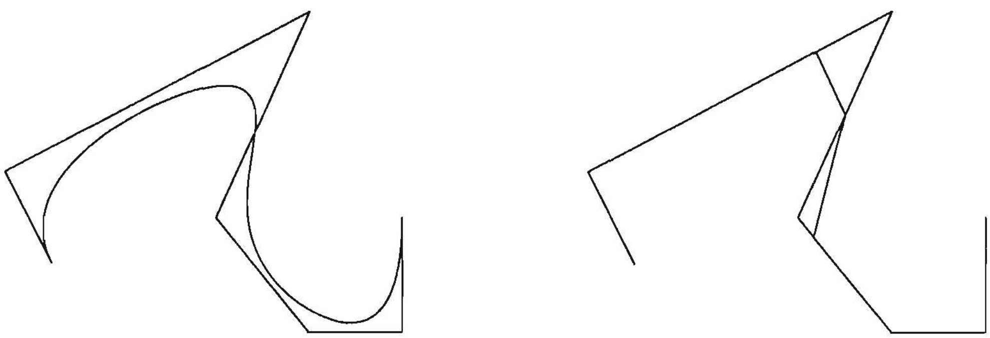

Figure 10-1: Original polygon and curve   Original and modified polygon

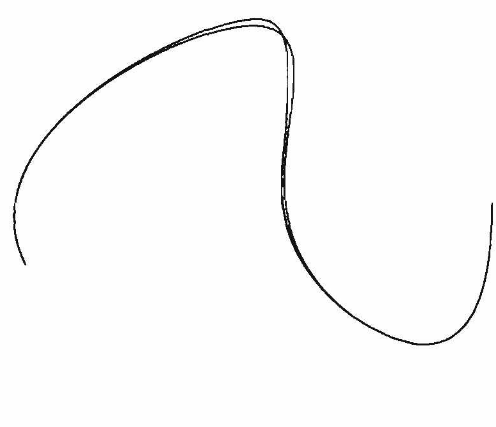

Figure 10-2: New $\underline{\mu}_{J+1}$, $\underline{\nu}_{J+1}$. Both old and new curves drawn.

11. Interpolation With Tensioned Splines

Since much of the original work in the use of tensioning with piecewise polynomials was done in the framework of interpolation methods, this work would be incomplete without treating that case.

We have discussed to this point the use of LT-splines as the blending functions in a "Bézier" type, or "B-spline" type curve formulation, noting that the geometrical aspects served by the LT control polygon are exactly analogous. When considering the use of the LT-splines to provide a basis for parametric interpolation, we must find the coefficients of the LT-spline which will interpolate specific points in space. Note that in using the "design" formulation one could change either the control polygon or the tension parameters at either one or many points to change the curve. In fact, a fixed control polygon led to different curves, one for each (μ, ν) sequence pair. The interpolation case is different. Here specific points which the curve must pass through are given, and the (μ, ν) pair is also given. Then one must solve for the LT-polygon, and use computational methods for rendering. If one wants to retain interpolation at the specified points and yet change some values in the tensioning sequences, one must resolve for yet another new LT-polygon. Below we discuss how to find the LT-polygon in this context and then present several examples.

The hypothesis is that given a sequence of increasing parameter values, $t_0 < ... < t_{n+1}$, tension sequences μ and ν, and n+4 vector values R_0', R_i, $i = 0,...,n+1$, R_{n+1}', we need to solve for a tensioned spline $\gamma(t)$ such that

$$\begin{aligned}
\gamma(t_i) &= R_i,\ i = 0, ..., n+1 \\
\gamma'(t_0) &= R_0' \\
\gamma'(t_{n+1}) &= R_{n+1}'
\end{aligned}$$

We also want $\gamma(t)$ to satisfy Equations 1 using the specified tension sequences at the given **t**-sequence values. This formulation of the problem leads to an open end condition LT-spline.

From B-spline properties, if $\gamma(t) = \sum Q_j N_j(t)$, where the knot sequence τ is defined as in Equation 2, and using Equations 61 through 69 we see that

$$Q_{3i} = R_i = \gamma(t_i),\ i = 0,...,n+1$$

which also means that

$$\begin{array}{llll}
P_{-3} & & & = R_0 \\
-P_{-3}\dfrac{3}{t_1-t_0} & +\,P_{-2}\dfrac{3}{t_1-t_0} & & = R_0' \\
P_{i-3}c_{i-3,3(i-3)+9} & +\,P_{i-2}c_{i-2,3(i-2)+6} & +\,P_{i-1}c_{i-1,3(i-1)+3} & = R_i,\ i = 1,...,n \\
 & -\,P_{n-1}\dfrac{3}{t_{n+1}-t_n} & +\,P_n\dfrac{3}{t_{n+1}-t_n} & = R_{n+1}' \\
 & & P_n & = R_{n+1}
\end{array}$$

The coefficients values $c_{i-3,3(i-3)+9}$, $c_{i-2,3(i-2)+6}$, and $c_{i-1,3(i-1)+3}$ are easily determined from their defining equations. This is a linear system of (n+4) equations in (n+4) unknowns. Further, the system is a tridiagonal system such that each element in each triple is nonnegative and each row sums to 1. This sparce system may quickly be solved for each different pair of tension sequence values, and then the LT-polygon and interpolation curve can be rendered using the earlier presented computational methods. In the following example, both the LT-polygon and the interpolating curve are shown; the data is shown with +-es. If just some tension pairs are modified, the data will look unchanged but the LT-polygons and resulting curves are different.

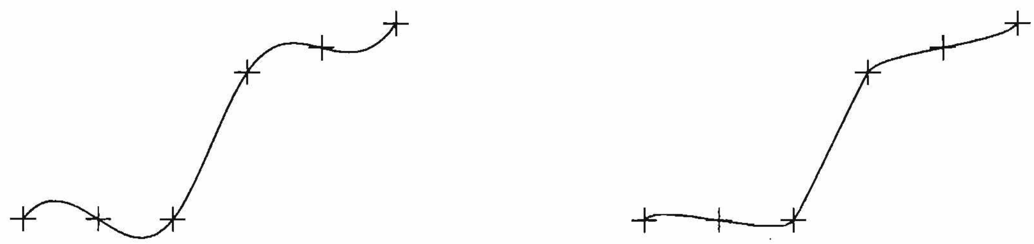

Figure 11-1: a. interpolated curve b. adjusted tensions

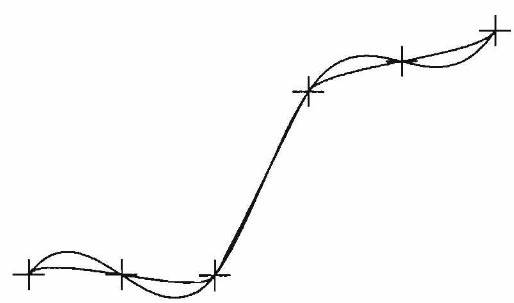

Figure 11-2: comparison of two interpolated curves

12. Tensioned Surfaces

Surfaces which may be straightforwardly computed from the LT-splines are of the form

$$S(u,v) = \sum_{j} \sum_{i} P_{i,j} S_{i,u}(u) S_{j,v}(v)$$

Where the functions $S_{i,u}$ are based on the u breakpoint sequence, and the functions $S_{j,v}$ are based on the v breakpoint sequence.

Computationally one can turn this equation into an appropriate one with the correct B-spline functions in both u and v. Fixing j, determine the B-spline coefficients for $N_{i,u}$, calling them $Q_i(j)$. This is done by using Equations 61 through 69 and the tension pair sequences for the "u" direction.

$$S(u,v) = \sum_j \sum_i Q_i(j) N_{i,u}(u) S_{j,v}(v)$$

Then using $Q_i(j)$ in place of P_j, the tension pair sequences for the "v" direction, and the same equations,

$$S(u,v) = \sum_i N_{i,u}(u) \sum_j Q_i(j) S_{j,v}(v)$$
$$S(u,v) = \sum_i \sum_j Q_{i,j} N_{i,u}(u) N_{j,v}(v)$$

Clearly, both open and floating tensioned surfaces are straightforwardly computed and can be modified by modifying just the appropriate Q values. Note that if a single tension pair in the u-direction is modified, then a strip of coefficients will be modified.

13. Other Possible Bases

As has been mentioned, others who have proposed using curvature continuous piecewise cubic functions have proposed a variety of other bases. The plus function formulation is very convenient for proving many interpolation and tangency properties. It is for computation that their global characteristics are undesirable. However, the plus function formulation does carry the continuity constraints implicitly. Unfortunately the coefficients of the plus functions do not seem to convey any geometric intuitions on the behavior of the curve.

Also proposed is the use of the power basis, with reparametrization over every interval. Barsky in 1981 [2] wanted to use the extra tensioning freedoms as design parameters, and hence wanted to have a control polygon of blending function coefficients which conveyed geometric intuitions, as occurs with B-splines (convex hull property and variation diminishing property). His initial approach was analogous to the early approach to B-splines. Uniform floating knot vectors were required. Also, all the elements in the μ sequence had the same value, and all the elements in the ν sequence had to have the same value, the homogeneous case. For these conditions, as for B-splines, as we have shown, all the local blending functions are just translates of each other. This being the case, he could solve for the one function in its power basis formulation over the interval [0,1] and then just translate it along. These functions are called β-splines. Further, solving for this function required the solution of a linear system which initially had sixteen unknowns. He reduced it to eleven unknowns. This approach does not seem to allow for open end conditions, nonuniform knot vectors, multivalued μ and ν sequences, nor refinement. Rendering is done by function evaluation and cannot be dynamically determined by the particular curve geometry, unless the curve is later converted to piecewise Bézier formulation. The conversion is not direct.

The general β-spline formulation proposed by Barsky uses a generalized divided difference formulation for the general floating case with multivalued μ and ν sequences (called the β_1 and β_2 sequences, respectively) and nonuniform knots. The divided difference used requires several steps and is not the standard definition. If the functions defined in that way are nonnegative, sum to one, and are variation diminishing with respect to the coefficient polygon, then they represent the same functions as the floating LT-splines, by uniqueness of bases. The variation diminishing property was proved for the ν-spline design formulation by Goodman [11] who also studied other properties of the nonuniform case.

While it is common practice to simulate the effects of open end conditions by placing multiple vertices at the ends, the two curves generated are not equivalent. If the lower end is made a triple vertex, then the first span of the curve will be in the convex hull spanned by the first four points, three of which are identically the same. That means that the first nondegenerate span is a straight line. The effect then of simulating open end conditions by multiple vertices is to imbed straight line segments into the curve near the ends. Use of the open end condition formulation allows selection of arbitrary curvature near the ends, and no straight lines are imbedded unless the designer wants to do so. Note that the multiple vertex approach leads to bilinear patches at the four corners of the tensor product surface, and leads to cubic-by-linear (and linear-by-cubic) patches along all four sides.

Recently others have looked at writing β-splines as piecewise Bézier curves, as noted earlier. Finally, one could just try to use the B-spline basis directly. Discovering the relationship between the B-spline basis and a control polygon with the characteristics of the LT-polygon might be difficult without the LT-formalism.

14. Conclusions

This work has provided a unifying conceptual and computational framework with which to attack the designing curves and surfaces with tensioned splines. Now one can use a control polygon for design with tensioned splined using standard open end conditions, and also there is an ability to add knots at arbitrary locations to either floating or open ended tension splines. This last property allows the capability of performing hierarchical design with tensioned splines in a straightforward manner. The use of B-splines as the underlying formulation, instead of the piecewise power basis expansion or a plus function expansion, permits one to use the computational and theoretical properties of B-splines. They have been used in behalf of deriving properties of tensioned splines and their coefficient control polygons and also used to directly access and employ subdivision techniques for dynamic rendering based on the curve characteristics.

This same approach can be used to derive curves with geometric continuity of higher degree.

15. Acknowledgements

The author would like to thank Jim Cobb for help in implementing the algorithms and generating the figures in the paper. The algorithms were implemented using the Alpha_1 geometric modelling system at the University of Utah. The author was pleased to use the facility which generates curves in Alpha_1 and translates them into postscript form for inclusion directly in the paper.

References

1. Barsky, B., Beatty, J., Bartells, *An Introduction to the Use of Splines in Computer Graphics,* Siggraph-84 course notes, 1984.

2. Barsky, B. A., "The Beta-spline: A Local Representation Based on Shape Parameters and Fundamental Geometric Measures," Ph.D. dissertation, , December 1981.

3. Bartels, Richard H. and Beatty, John C., "Beta-splines with a Difference," CS-83-40, Department of Computer Science, University of Waterloo, Waterloo, Ontario, 1983.

4. de Boor, C., "On Calculating with B-splines," *Journal of Approximation Theory,* Vol. 6, No. 1, July 1972, pp. 50-62.

5. de Boor, C., *A Practical Guide to Splines,* Springer-Verlag, Applied Mathematical Sciences, Vol. 27, 1978.

6. Cohen, E., Riesenfeld, R.F, *Introduction to Computer Aided Geometric Design,* to appear, 1988.

7. Cohen, E.; Lyche, T.; and Riesenfeld, R. F., "Discrete B-splines and Subdivision Techniques in Computer-Aided Geometric Design and Computer Graphics," *Computer Graphics and Image Processing,* Vol. 14, No. 2, October 1980, pp. 87-111, Also Tech. Report No. UUCS-79-117, , October 1979

8. Cohen, E.; Lyche, T.; and Morken, K., "Knot Line Refinement Algorithms for Tensor Product B-spline Surfaces," *CAGD,* to appear 1985.

9. Cox, M. G., "The Numerical Evaluation of B-splines," Report NPL-DNACS-4, Division of Numerical Analysis and Computing, National Physical Laboratory, Teddington, Middlesex, England, August 1971, Also in *J. Inst. Maths. Applics.*, Vol. 10, 1972, pp. 134-149

10. Farin, G., "Visually C^2 Cubic Splines," *Computer Aided Design,* Vol. 14, 1982, pp. 137-139.

11. Goodman, T., "Properties of β-splines," *Journal of Approximation Theory,* Vol. 44, 1985, pp. 132-153.

12. Manning, J. R., "Continuity Conditions for Spline Curves," , Vol. 17, No. 2, May 1974, pp. 181-186.

13. Nielson, G. M., "Some Piecewise Polynomial Alternatives to Splines under Tension" in *Computer Aided Geometric Design,* Barnhill, Robert E. and Riesenfeld, Richard F., ed., Academic Press, New York, 1974, pp. 209-235.

14. Nielson, G. M., "Computation of NU-splines," Tech. report 044-433-11, Department of Mathematics, Arizona State University, June 1974.

15. Sabin, M. A., "Parametric Splines in Tension," VTO/MS/160, British Aircraft Corporation, Weybridge, Surrey, England, 23 July 1970.