An introduction to the mathematics and construction of splines

Stephen Weston
Addix Software Consultancy Limited
Version 1.6
September 2002

Contents

1 Introduction to splines

Everyone that has ever tried to apply simple linear interpolation to find a value between pairs of data points will be only too aware that such attempts are extremely unlikely to provide reliable results if the data being used is anything other than broadly linear. In an attempt to deal with inherent non-linearity, the next step usually involves some sort of polynomial interpolation. This generally leads to far more stable and robust interpolation and fitting, but is also potentially a difficult area as the end points, monotonicity, convexity and continuity of derivatives all make their influences felt in often contradictory ways.

One of the most popular ways of dealing with these issues is to use splines. In their most general form, splines can be considered as a mathematical model that associate a continuous representation of a curve or surface with a discrete set of points in a given space. Spline fitting is an extremely popular form of piecewise approximation using various forms of polynomials of degree n, or more general functions, on an interval in which they are fitted to the function at specified points, known as control points, nodes or knots. The polynomial used can change, but the derivatives of the polynomials are required to match up to degree $n-1$ at each side of the knots, or to meet related interpolatory conditions. Boundary conditions are also imposed on the end points of the intervals. The heart of spline construction revolves around how the selected control points are effectively "blended" together using the polynomial function of choice.

Given the various alternative forms of spline, the question of which type of spline is most applicable in any given situation naturally arises and is inevitably a difficult one to answer without clear criteria. Arguably the most important deciding question is whether the spline is required to approximate or interpolate the control points. In other words, does the user require the curve to pass through the control points with absolute precision, or is the overall shape of the curve more important. Beyond this critical requirement, Blanc and Schlick offer an extremely useful check-list of desirable properties that any spline model should possess. The list of properties can be regarded as a convenient set of features against which the usefulness of various spline types can be measured:

- *Affine invariance*: the affine transformation of a spline should be obtained by applying the transformation to its control points. This is usually expressed in terms of a normality constraint.

- *Convex hull*: the spline should be entirely contained in the convex hull of its control lattice. This is captured via a combination of the normality constraint with a positivity constraint.

- *Variance dimunition*: the number of intersections between the spline and a plane should be at most equal to the number of intersections between the plane and the control lattice, which means that the spline should have less oscillations than its control lattice. This property is ensured by combining the normality and positivity constraints with a regularity constraint.

- *Local control*: each control point should only exert influence on the shape of the spline in a restricted zone. This property is generated by a locality constraint. A given spline fitting method may offer varying degrees of local control depending on the influence of any given control point. Blanc and Schlick express this in terms of L^p locality, such that a spline can be said to have L^p locality when each control point influences at most p segments.

- *Smooth and sharp shapes*: the spline should permit the mixing of sharp and smooth sections within the same curve. Parametric continuity does not provide any information on the shape of the curve, so that geometric continuity can be imposed, with the requirement that sharp shapes are G^0 and smooth shapes are G^2 geometrically continuous. Parametric continuity is also required to ensure C^2 continuity.

- *Intuitive shape parameters*: in addition to the control parameters, the spline should also provide additional degrees of freedom (such as weights, tension, bias and curvature, generally referred to as shape parameters), which should allow the user to pull the spline locally toward one or more control points in an intuitive fashion.

- *Existence of refinement algorithms*: the spline model should lend itself to the use of refinement or subdivision techniques which serve to increase the degrees of freedom for a spline without modifying its shape.

- *Conic representation*: the spline model should permit the representation of conic sections and therefore support a wide range of curves and surfaces such as circles, ellipses, spheres and cylinders etc.

- *Approximation/interpolation*: spline models should provide both approximation and interpolation splines in a unified formulation.

The Addix Spline library provides the capability to interpolate using seven different spline blending techniques. Each must be used with care and some are more suitable than others for certain types of data. For example, cubic splines are probably the most well known and are particularly attractive as they provide second derivatives that vanish at the end points, thereby satisfying the boundary conditions. However, they tend not to work so well with non-uniform data and can have unpredictable results at the end points of a data set.

This document is designed to provide a short introduction to each of the types of spline covered in the Addix Spline library. There is also a bibliography at the end of this document that gives the principal references used for the construction of each type of spline, as well as further sources which provide a sound starting point for further investigation by the interested reader.

2 Auto-tensioning splines

Splines with uniform tension were first introduced by Schweikert (1966) to allow the use of tension factors as a versatile alternative to polynomial interpolation. Cline (1974) provides a methodology for interpolation and smoothing with uniform tension. However, the use of constant tension can result in more tension than is necessary over some intervals. Renka (1987) provides what is arguably the superior solution to the problem of finding a versatile method of interpolation and smoothing using tension splines. Renka's method of shape-preserving interpolation uses piece-wise exponential functions with a tension factor associated with each interval. Knots are set to coincide with data points and the interpolatory value is framed in terms of its values and first derivatives at these points. For a given set of derivatives, this permits the efficient computation of the minimum tension factor for which the interpolant satisfies locally defined properties such as monotonicity and convexity, as well as more general bounds on function values and derivatives, in each interval. Renka uses a local derivative estimation procedure to produce a C^1interpolant satisfying the constraints with minimum tension. He also uses a similar iterative approach to obtain a C^2 spline fit which satisfies the constraints. The following is a brief summary of the main features of Renka's method.

Given tension factors $\sigma_k \geqq 0$ associated with intervals $[x_k, x_{k+1}]$ of length $h_k = x_{k+1} - x_k$, define the tension spline f

$$f \in C^1 [x_1, x_n] \text{ and } f^{'''} - (\sigma_k/h_k)^2 f^{''} = 0 \text{ in } [x_k, x_{k+1}] \tag{1}$$

for $k = 1, ..., n-1$. Note that this definition differs from others used in the other methods presented in this paper in two respects, namely, that the continuity restriction on f has been relaxed and σ_k has been scaled by $1/h_k$. This normalisation makes the interpolant independent of a scaling of the abscissae (as well as the data values). When $\sigma_k = 0$ on $[x_k, x_{k+1}]$, it implies that f is cubic, whilst $\sigma_k > 0$ implies that $(\sigma_k/h_k)^2 f^{''} - f$ is linear and that f therefore approaches linear as σ_k increases.

In addition to letting y_1, y_2 and $y_1^{'}, y_2^{'}$ denote the data values and their associated derivatives respectively, make the following definitions

$$h = x_2 - x_1, \quad b = (x_2 - x_1)/h, \quad s = (y_2 - y_1)/h, \quad d_1 = s - y_1^{'}, \quad d_2 = y_2^{'} - s \tag{2}$$

Renka then uses the following two modified hyperbolic functions as the basis functions for the required interpolant

$$\sinh m(z) = \sinh(z) - z \quad \text{and} \quad \cosh m(z) = \cosh(z) - 1 \tag{3}$$

Further, define

$$E = \sigma * \sinh(\sigma) - 2\cos hm(\sigma) = \cosh m^2(\sigma) - \sinh m(\sigma)\sinh(\sigma) \tag{4}$$
$$\alpha_1 = \sigma * \cosh m(\sigma) d_2 - \sinh m(\sigma)(d_1 + d_2) \tag{5}$$
$$\alpha_2 = \sigma * \sinh(\sigma) d_2 - \cosh m(\sigma)(d_1 + d_2) \tag{6}$$

Then for $x \in [x_1, x_2]$ and $\sigma > 0$, the interpolant and its derivatives are given by the expressions

$$f(x) = y_2 - y_2^{'} hb + h/(\sigma E)\left[\alpha_1 \cosh m(\sigma b) - \alpha_2 \sinh m(\sigma b)\right] \tag{7}$$
$$f^{'}(x) = y_2^{'} - (1/E)[\alpha_1 \sinh(\sigma b) - \alpha_2 \cosh m(\sigma b)] \tag{8}$$
$$f^{''}(x) = \sigma/(hE)[\alpha_1 \cosh(\sigma b) - \alpha_2 \sinh(\sigma b)] \tag{9}$$

f interpolates the specified values and derivatives and satisfies the differential equation above.. Continuity of f and $f^{'}$across interval boundaries follows from the interpolatory conditions. When $\sigma = 0$ the standard cubic interpolatory expressions apply.

The next step it to calculate the minimum tension factors necessary to satisfy a constraint in an interval for which data values and derivatives are given at the endpoints. In the case of C^2 interpolation, tension factors can be selected by alternating between computation of nodal derivatives and the alternative procedures used by Renka suggested below. Renka treats three types of constraint, namely, bounds on function values, bounds on first derivative values and convexity or concavity. Clearly, for each type of constraint, it can either not be satisfied, satisfied by the cubic such that $\sigma = 0$, or $\sigma > 0$ is necessary and sufficient to satisfy the constraint.

Renka presents detailed consideration of each of these three situations, which the interested reader can consider, but which are beyond the scope of this introductory paper. Instead, this paper now moves on to considering numerically stable expressions for evaluating the interpolant, its derivatives and its integral over an arbitrary domain using expressions similar to those used to compute the tension factors. It is important to note that straightforward computation of the hyperbolic functions $\sinh m(z)$ and $\cosh m(z)$ may fail both in the case of small z, due to cancellation error and with large values of z, due to overflow. Renka avoids both problems by using polynomial approximations in the case of small tension factors and by scaling using $e^{-\sigma}$for large σ. Renka's method provides smooth transitions (defined as continuity with respect to σ) at $\sigma = 0$, $\sigma = 0.5$ and $\sigma = \sigma^c$ where $\exp(-\sigma^c)$ is equal to machine precision (the floating point values of $1 - e$ and $1 + e$ are unity for $\sigma > \sigma^c$.

Renka next considers local derivative estimation for a C^1 interpolant and a global procedure for which the nodal derivatives $y_i^{'}$ are selected to provide second derivative continuity with some ability to choose end point conditions. The C^2 interpolation method has an advantage in terms of accuracy and additional smoothness, which may be necessary in some applications, but is less efficient in both the preprocessing and evaluation phases of spline calculations. In the preprocessing phase, a linear system of equations must be solved for the derivatives and when constraints need to be satisfied it is necessary to iterate over both the calculation of the derivatives and tension factors. This process needs to be repeated if any further points are added , whereas local methods, in which $y_i^{'}$ depends on only a few nearby data points, allow for efficient updating to the data set. Imposing constraints, means that the C^2 interpolant generally has non-zero tension factors in most intervals, whereas using the C^1 method combined with an appropriate choice of derivatives, results in relatively few non-zero tension factors yielding greater efficiency in the evaluation phase.

Renka finally considers the problem of choosing the minimum tension factors required to satisfy constraints associated with the intervals when a C^2 interpolant is called for. In such a case, the equations for tension factors cannot be separated and a system of $2n - 1$ non-linear equations must be solved for the n nodal derivatives and $n - 1$ tension factors. Renka uses a simple iterative procedure to achieve this, which, beginning with zero tension factors, alternates between solving the linear system for nodal derivatives and choosing optimal tension factors for the given derivative estimates. Convergence is ensured (though not necessarily within an acceptable number of iterations) by opting to update a tension factor only if it increased from its value at a previous iteration and by placing a physical upper bound of 100 on tension factors. Constraints are therefore satisfied but with more tension than is absolutely necessary in some cases. This method has the features the it takes full advantage of the sparsity and linearity inherent in the problem, whilst also providing the flexibility to balance constraints and smoothness of the interpolant.

3 Basis splines

B or basis splines were developed in the 1970's and are derived from Bezier splines. A curve can be parameterised by

$$P_0 f_0(t) + P_1 f_1(t) + P_2 f_2(t) + P_3 f_3(t) \tag{10}$$

where f_0, f_1, f_2 and f_3 are the blending functions which indicate how the control points are blended together to form the spline curve. In the case of Bezier curves the blending functions are the Bernstein polynomials. In order to get a good fit, it is generally desirable that blending functions possess the following properties:

- Accurately and rapidly computable.
- Each function should be equal to zero most of the time.
- Smoothness.
- Permit interpolation of the control points if required.
- For each value of t, the sum of the blending functions should be unity.

In common with many types of spline, basis splines use piece-wise polynomials. The degree of the polynomials is one less than their order, with smoothness being two less than the order. So, for example, blending functions of order 4 will be piecewise polynomial of degree 3 whose smoothness or continuity is C^2. This of course results in low degree polynomials. More precisely, given $L+1$ knot points $\{P_0, P_1, ..., P_L\}$ define a basis or blending function for each knot point of order m

$$N_{0,m}(t), N_{1,m}(t), N_{2,m}(t), ..., N_{L,m}(t) \tag{11}$$

so that the spline curve will be parameterised

$$P_0 N_{0,m}(t) + P_1 N_{1,m}(t) + P_2 N_{2,m}(t) + ... + P_L N_{L,m}(t) \tag{12}$$

Basis functions can be parameterised recursively for order $m > 1$ (degree $m-1$ and smoothness or continuity C^{m-2})

$$N_{k,m}(t) = \frac{t-k}{m-1} N_{k,m-1}(t) + \frac{k+m-t}{m-1} N_{k+1,m-1}(t) \tag{13}$$

There is particular interest in the basis spline of order 3 since they are described by quadratics and are generally referred to a quadratic basis splines. Using the above definition, it is easy to define the first quadratic spline basis function

$$N_{0,3}(t) = \left\{ \begin{array}{c} \frac{1}{2}\left(t^2\right), \\ \frac{1}{2}\left(-2t^2+6t-3\right), \\ \frac{1}{2}\left(t^2-6t+9\right), \\ 0, \end{array} \right\} \begin{array}{c} \text{if } 0 < t \leqslant 1 \\ \text{if } 0 < t \leqslant 2 \\ \text{if } 2 < t \leqslant 3 \\ \text{otherwise} \end{array} \tag{14}$$

So, defining the points $\{P_0 = (x_0, y_0), P_1 = (x_1, y_1), P_2 = (x_2, y_2)\}$ and reducing the scaling to $[0, 1]$, gives

$$x = x_0 \frac{1}{2}\left(t^2 - 2t + 1\right) + x_1 \frac{1}{2}(-2t^2 + 2t + 1) + x_2 \frac{1}{2}(t^2) \tag{15}$$

There is of course an analogous expression for the y-values.

4 Catmull-Rom or Overhauster splines

The Catmull-Rom spline is a local interpolating spline developed for computer graphics purposes. Its initial use was in the design of curves and surfaces, but more recently its application has been extended. One of the most desirable features of the Catmull-Rom spline is that the specified curve will pass through all of the control or knot points, a feature which is not necessarily true of other spline methodologies.

The Catmull-Rom spline can be calculated in one of two ways. One uses four knot points and no tangents, while the second method uses two knot points and two tangents. The overall mathematics behind the two variations in the approach can be summarised simply as follows. In the case of the two knots and two tangents, the matrix form of the spline arises from a simple geometrical argument that attempts of fix the tangents at certain knot points to be the average of the slopes of the two line segments of the control polygon adjacent to each control point. The matrix equation is then just a simple result of matrix algebra.

Given the two knot points P_0 and P_1 and the slopes of the tangents $P_0^{'}$ and $P_1^{'}$ at each point, it is possible to define a parametric cubic curve that passes through P_0 and P_1, with the respective slopes $P_0^{'}$ and $P_1^{'}$ by equating the coefficients of the polynomial function

$$P(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{16}$$

with the above values:

$$P(0) = a_0 \tag{17}$$

$$P(1) = a_0 + a_1 + a_2 + a_3 \tag{18}$$

$$P^{'}(0) = a_1 \tag{19}$$

$$P^{'}(1) = a_1 + 2a_2 + 3a_3 \tag{20}$$

Solving the resulting simultaneous equations for a_0, a_1, a_2 and a_3 gives

$$
\begin{aligned}
a_0 &= P(0) && (21)\\
a_1 &= P^{'}(0) && (22)\\
a_2 &= 3[P(1) - P(0)] - 2P^{'}(0) - P^{'}(1) && (23)\\
a_3 &= 2[P(0) - P(1)] + P^{'}(0) + P^{'}(1) && (24)
\end{aligned}
$$

which, upon substitution into the original polynomial and subsequent simplification gives

$$
\begin{aligned}
P(t) &= (1 - 3t^2 + 2t^3)P(0) && (25)\\
&\quad +(3t^2 - 2t^3)P(1) && (26)\\
&\quad +(t - 2t^2 + t^3)P^{'}(0) && (27)\\
&\quad +(-t^2 + t^3)P^{'}(1) && (28)
\end{aligned}
$$

which is clearly in a cubic polynomial form. The method can be employed to obtain a curve through a more general set of knot points $\{P_0, P_1, ..., P_n\}$ by considering pairs of knot points and using the above method for two points. Of course, the computation requires the slopes of the tangents at each knot point.

To implement this variation of the Catmull-Rom approach given $n + 1$ knot points $\{P_0, P_1, ..., P_n\}$, the method involves finding a curve that passes through these knot points and is local in nature. Therefore, define the curve on each segment $\widehat{P_i P_{i+1}}$ by using the two knot points and specifying the tangent to the curve at each control point to be

$$
\frac{P_{i+1} - P_{i-1}}{2} \text{ and } \frac{P_{i+2} - P_i}{2} \qquad (29)
$$

respectively. Substituting these tangents into the above method, the following matrix equation arises

$$
P(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} P_i \\ P_{i+1} \\ \frac{P_{i+1} - P_{i-1}}{2} \\ \frac{P_{i+2} - P_i}{2} \end{bmatrix} \qquad (30)
$$

The second Catmull-Rom method involves using four knot points but no tangents and can be summarised as follows. To calculate a point on the curve, a total of four knot points are required - two on either side of the target value. Denoting these knot points as P_0, P_1, P_2 and P_3, and the value t as the required abscissae, the location of the point can be calculated as follows (assuming uniform spacing of the knot points)

$$
P(t) = 0.5 * \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \qquad (31)
$$

which in scalar notation is

$$P(t) = 0.5 * (2P_1 + (-P_0 + P_2) * t \quad (32)$$
$$+(2P_0 - 5P_1 + 4P_2 - P_3) * t^2 \quad (33)$$
$$+(-P_0 + 3P_1 - 3P_2 + P_3) * t^3) \quad (34)$$

Both methods provide a Catmull-Rom spline with the following characteristics:

- The spline passes through all of the knot points.
- The spline is C^1 continuous, so that there are no discontinuities in the tangent direction and magnitude.
- The spline is not C^2 continuous, as the second derivative is linearly interpolated within each segment, which causes the curve to vary linearly over the length of the segment.
- Points on a segment may lie outside of the domain $P_1 \rightarrow P_2$.

5 G-splines

G-splines are calculated used to interpolated Hermite-Birkoff (HB) data. Consider first the definition of an HB interpolation problem. This description draws closely on the ACM paper by Eidson and Schumaker (1974). Let $N \geq 2$ and $x_1 < x_2 < x_3 < ... < x_N$ be given and suppose that for each $j, 1 \leq j \leq N$, that z_j is a positive integer, $IM_{1,j} < IM_{2,j} < ... < IM_{zj,j}$ are positive integers and $y_{1,j}, y_{2,j}, , ..., y_{zj,j}$ are prescribed real double precision numbers. Then the HB interpolation problem is to determine s such that

$$s^{(IM_{ij}-1)}(x_j) = y_{i,j}, \qquad i = 1, 2, ..., z_j, \ j = 1, 2, ..., N \quad (35)$$

We can clearly see the z_j describes the number of derivatives prescribed at x_j while the vector $(IM_{1,j}, IM_{2,j}, ..., IM_{zj,j})$ describes which derivatives. If $z_j = 1$, then the interpolation problem is simple. However, in most cases the focus of attention is solving HB-interpolation problems with polynomial splines. Therefore, let M be an integer such that $M \geqslant IM_{2j,j}, \ j = 1, 2, ..., N$. Then there exists a function s satisfying 35 and

$$s^{(2,M)}(t) = 0, \quad x_j < t < x_{j+1}, \quad j = 1, 2, ..., N-1; \quad (36)$$
$$s^{(M)}(t) = 0, \quad t < x_1, \ t > x_N; \quad (37)$$
$$s \in C^{(M-1)}(-\infty, \infty); \quad (38)$$
$$s^{(2M-l)}(x_j+) = s^{(2M-l)}(x_j-), \text{ where} \quad (39)$$
$$l \in \{1, ..., M\} \backslash \{IM_{1,j}, ..., IM_{z_j,j}\} \quad (40)$$
$$j = 1, 2, ..., N \quad (41)$$

The function s is called a g-spline and is a polynomial spline of degree $2M-1$ (i.e. it is a piecewise polynomial of degree $2M-1$).

If the only polynomial that solves the homogeneous Hermite-Birkhoff interpolation problem is the identically zero polynomial, then the problem is said to be M-poised. In which case, there is a unique g-spline of degree $2M-1$ that solves the Hermite-Birkoff problem. Eidson and Schumaker (1974) consider only g-splines for M-poised Hermite-Birkoff problems.

Given an M-poised Hermite-Birkoff interpolation problem, the unique g-spline interpolant s, satisfying equations 36 to 41 above can be represented as:

$$s(t) = \left\{ \begin{array}{cc} p_1(t), & t \leq x_1 \\ p_j(t), & x_{j-1} < t \leq x_j, \quad j = 2, 3, ..., N \\ p_{N+1}(t), & t > x_N \end{array} \right\} \tag{42}$$

where for $j = 1, 2, ..., N$, $p_j(t)$ is a polynomial of the form

$$p_j(t) = \sum_{l=1}^{2M} C_{l,j}(t - x_j)^{l-1} \tag{43}$$

$$p_{N+1}(t) = \sum_{l=1}^{M} C_{l,N}(t - x_N)^{l-1} \tag{44}$$

6 Non-Uniform Rational Basis Splines

Non-Uniform Rational Basis Splines, or Nurbs for short, have their origins in the work of French automobile engineer Pierre Bézier in the late 1960's and early 1970's. Basis spline curves and surfaces quickly followed due to their greater power and flexibility. The subsequent development of rational and non-uniform rational basis spline curves and surfaces added even greater flexibility and precision in interpolation. One of the key attributes of Nurbs is their ability to provide a precise representation of both conic curves and surfaces. Nurbs therefore provide a single internal representation for a wide variety of both curves and surfaces, so that any number of different forms may then be embedded in single or multiple complex surfaces in a highly robust fashion, subject to geometric (or physical) or parametric (or mathematical) continuity requirements. Nurbs are therefore useful to groups as diverse as engineers, animators or even designers of computer games.

Addix.Spline provides the facility to calculate Nurbs, beginning with Bézier curves and covers the use of both open and periodic knot vectors, both with uniform and non-uniform spacing. Taking each variant in turn, a uniform knot vector has equally spaced knot values, with multiple knot values at either end (but nowhere else). Non-uniform knot vectors can have uneven spacing between knot points and/or multiple internal knot values. Open and periodic refer to whether or not the curve has fixed end points as would be the case in the event of defining the curve over a closed interval.

Nurbs curves are not only controlled by control points, but they can also be controlled by weights (at each knot point if required). A Nurbs curve is therefore a vector-valued piecewise rational polynomial function of the form:

$$C(u) = \frac{\sum_{i=0}^{n} w_i P_i N_{i,k(u)}}{\sum_{i=0}^{n} w_i N_{i,k(u)}} \tag{45}$$

where w_i are the weights, P_i are the knot points and $N_{i,k}$ are the normalised basis-spline basis function of degree k. The basis splines are defined recursively as:

$$N_{i,k(u)} = \frac{u - t_i}{t_{i+k} - t_i} N_{i,k-1(u)} + \frac{t_{i+k+1} - u}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1(u)} \tag{46}$$

and

$$N_{i,0(u)} = \{ \begin{array}{c} 1, \text{ if } t_i \leq u < t_{i+1} \\ 0 \text{ else} \end{array} \tag{47}$$

where t_i are the knots

$$U = \{t_0, t_1, t_2, ..., t_m\} \tag{48}$$

The knot vector uniquely determines the basis-splines as is clear from the above equations. The relation between the number of knots $(m+1)$, the degree (k) of $N_{i,k}$ and the number of control points $(n+1)$ is given by $m = n + k + 1$. The sequence of knots in the knot vector U, is assumed to be non-decreasing, such that. $t_i <= t_i + 1$, so each successive pair of knots represents an interval $[t_i$, $t_{i+1})$ for the parameter values to calculate a segment of a shape. The relative parametric intervals (or knot intervals) for Nurbs need not be the same for every shape segment, so that the knot spacing can be non-uniform, leading to a non-periodic knot vector of the form

$$U = \{a, ...a, t_{k+1}, ...t_{m-k-1}, b, ...b\} \tag{49}$$

where a and b can be repeated knot values with a multiplicity equal to $k + 1$. The multiplicity of a knot affects the parametric continuity at this knot. Non-periodic basis-splines, like Nurbs, are infinitely continuously differentiable in the interior of a knot span and $k - m - 1$ times continuously differentiable at a knot, where M is the multiplicity of the knot. This is in contrast to a periodic knot vector $U = \{0, 1, ..., n\}$, which is everywhere $k - 1$ times continuously differentiable. Considering the knot vector for Nurbs, the end knot points (t_k, t_{n+1}) with multiplicity $k + 1$ coincide with the end control points P_0, P_n.

Given that knot spacing can be non-uniform, the basis-splines are therefore no longer the same for each interval $[t_i,\ t_{i+1})$ and the degree of the basis-spline can also vary. Over the whole range of parameter values represented by the knot vector, the different basis-splines build up continuous (overlapping) blending functions $N_{i,k(u)}$, as previously defined, over this range (see Figure 1 below for details). These blending functions have the following properties:

- $N_{i,k(u)} >= 0$, for all i, k, u;
- $N_{i,k(u)} = 0$, if u not in $[t_i, t_{i+k+1})$, meaning local support of $k + 1$ knot spans, where $N_{i,k(u)}$ is nonzero;

if u in $[t_i, t_{i+1})$, the non-vanishing blending functions are $N_{i-k,k(u)}$, ..., $N_{i,k(u)}$
$\sum_{j=i-k}^{i} N_{j,k(u)} = \sum_{i=0}^{n} N_{i,k(u)} = 1$
in case of multiple knots, $\frac{0}{0}$ is deemed to be zero. Taken together, the result into the convex hull, the control points build up for a shape show that k+1 successive control points a shape segment is defined and a control point is involved in $k+1$ neighboring shape segments. Therefore, a change to a control point or weight influences only $k + 1$ shape segments, defined over the given interval.

The previous definition of a Nurbs-curve can be rewritten using rational basis functions:

$$R_{i,k(u)} = \frac{w_i N_{i,k(u)}}{\sum_{j=0}^{n} w_j N_{j,k(u)}} \tag{50}$$

into

$$C(u) = \sum_{i=0}^{n} P_i R_{i,k(u)} \tag{51}$$

Figure 1 shows how the blending functions and the Nurbs curve appear. Note that the rational basis functions have the same properties as the blending functions, most notably their invariance under affine transformations

As already mentioned above, when using weights to construct a curve, the weight w_i of a knot point P_i affects only the range $[t_i, t_{i+k+1})$, as can be seen from Figure 2. First, define the following points:

$$B = C(u; w_i = 0) \tag{52}$$
$$N = C(u; w_i = 1) \tag{53}$$
$$B_i = C(u; w_{inot} = \{0, 1\}) \tag{54}$$

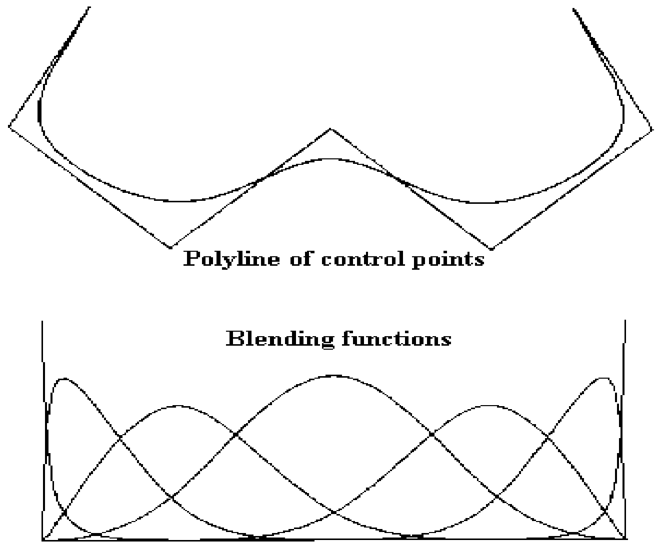
N and B_i can also be expressed as:

$$N = (1 - a)B + aP_i \tag{55}$$
$$B_i = (1 - b)B + bP_i \tag{56}$$

where

$$a = R_i, k(u; w_i = 1) \tag{57}$$
$$b = R_i, k(u) \tag{58}$$

Figure 1: Cubic Nurbs curve and blending functions



The following identity is obtained from the expression of a and b:

$$(1-a)/a : (1-b)/b = P_i N/BN : P_i B_i/BB_i = w_i \tag{59}$$

which is called the cross- or double ratio of the four points P_i, B, N, B_i. From these expressions, the effect of shape modification can be easily seen.as follows:

B_i sweeps out on a straight line segment, if $w_i = 0$, then P_i has no effect on shape; if w_i increases, so b and the curve is pulled toward P_i and pushed away from P_j, for $j \neq i$; if w_i decreases, so b and the curve are pushed away from P_i and pulled toward P_j, for $j \neq i$; if $wi \to \infty$ then $b \to 1$ and $B_i \to P_i$, if u in $[t_i, t_{i+k+1})$. The geometric impact and meaning of using weights can be clearly seen in Figure 2:

Nurbs have become extremely popular among many groups involved in constructing curves. However, it is in their ability to construct surfaces that Nurbs are probably of most use. Moving from curves to surfaces adds a number of complications, so in order to make exposition simpler and clearer, each will be considered in turn.

As already discussed, a rational basis-spline curve is the projection of non-rational polynomial basis-spline curve defined in four dimensional homogeneous coordinate space back into three dimensional physical splace. Projecting back into three dimensional space by dividing through by the homogeneous coordinate provides a rational basis-spline curve.

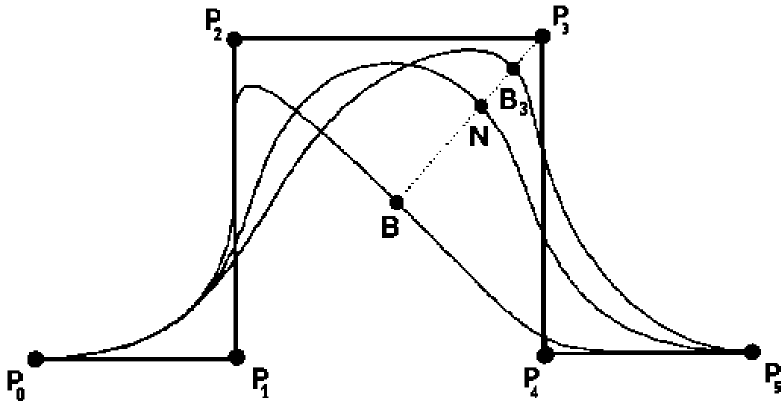


Figure 2: The Geometric effect of using weights

7 Quintic Splines

The use of a piece-wise interpolating quintic natural spline function, $S(x)$, for the set of monotonic data points (x_i, y_i), $i = 1, 2, ..., N$, is designed to produce a spline curve which is both smoother and less volatile in response to variations in knot points. The method presented in Addix.Spline principally follows the papers by Greville (1967) and Herriot and Reinsch (1976). The former shows that if $N > 2$, then there is an interpolating quintic natural spline function that has the following properties:

- $S(x)$ is a fifth degree polynomial in each interval, (x_i, x_{i+1})
- $S(x)$ and its first four derivatives, upto and including $S^{''''}(x)$ are continuous in $[x_1, x_N]$
- $S^{'''}(x_1) = S^{'''}(x_2) = S^{''''}(x_1) = S^{''''}(x_2) = 0$
- $S(x_i) = y_i$, $i = 1, 2, ..., N$

The spline function can be represented in the form:

$$S(x) = y_i + B_i t + C_i t^2 + D_i t^3 + E_i t^4 + F_i t^5 \tag{60}$$

with $t = x - x_i$ for $x_i \leq x < x_{i+1}$, $i = 1, 2, ..., N$. The code in Addix.Spline computes the coefficients, B_i, C_i, D_i, E_i and F_i for any arbitrary set of user supplied data points (x_i, y_i). The calculation method utilises minimum support B-splines of degree 2 to form a basis for the class of third derivatives of quintic natural splines.

The methodology is based on the assumption of strict monotonicity of the knots. Let $M_i(x)$ denote the B-spline of degree 2 that vanishes outside the interval (x_{i-1}, x_{i+2}) and let $h_i = x_{i+1} - x_i$, $t = x - x_{i-1}$, $u = x - x_i$ and $v = x - x_{i+1}$. Then

$$\begin{aligned} M_i(x) &= At^2 && x_{i-1} \le x < x_i \\ &= B + Cu - Du^2 && x_i \le x < x_{i+1} \\ &= E(v - h_{i+1})^2 && x_{i+1} \le x < x_{i+2} \end{aligned} \tag{61}$$

where

$$A = \frac{1}{[h_{i-1}(h_{i-1}+h_i)]}, \quad B = \frac{h_{i-1}}{(h_{i-1}+h_i)}, \quad C = \frac{2}{(h_{i-1}+h_i)} \tag{62}$$

and

$$D = \frac{(h_{i-1}+2h_i+h_{i+1})}{(h_{i-1}+h_i)h_i(h_i+h_{i+1})}, \quad E = \frac{1}{h_{i+1}(h_i+h_{i+1})} \tag{63}$$

Given that the third derivative, $S^{'''}(x)$ vanishes outside the interval (x_0, x_n), it will have a unique representation of the form

$$S^{'''}(x) = \sum_{j=1}^{n-2} 60\gamma_j M_j(x) \tag{64}$$

such that γ_j can be found by using the relation

$$\int_{-\infty}^{\infty} M_j(x) S^{'''}(x) dx = 2(S(x_i, x_{i+1}, x_{i+2}) - S(x_{i-1}, x_i, x_{i+1})) \tag{65}$$

Using integration by parts the following well-conditioned positive-definite pentadiagonal system of linear equations can be derived which can then be used to solve for γ_j

$$\begin{array}{lllllllllll} d_1\gamma_1 & + & e_1\gamma_2 & + & f_1\gamma_3 & & & & & = & c_1 \\ e_1\gamma_1 & + & d_2\gamma_2 & + & e_2\gamma_3 & + & f_2\gamma_4 & & & = & c_2 \\ f_{i-2}\gamma_{i-2} & + & e_{i-1}\gamma_{i-1} & + & d_i\gamma_i & + & e_i\gamma_{i+1} & + & f_i\gamma_{i+2} & = & c_i \\ f_{n-5}\gamma_{n-5} & + & e_{n-4}\gamma_{n-4} & + & d_{n-3}\gamma_{n-3} & + & e_{n-3}\gamma_{n-2} & & & = & c_{n-3} \\ & & f_{n-4}\gamma_{n-4} & + & e_{n-3}\gamma_{n-3} & + & d_{n-2}\gamma_{n-2} & & & = & c_{n-2} \end{array} \tag{66}$$

where

$$\begin{aligned} d_i &= T_1 + T_2 + T_3, && i = 1, 2, ..., n-2 \\ e_i &= T_4 + T_5 && i = 1, 2, ..., n-3 \\ f_i &= T_6 && i = 1, 2, ..., n-4 \\ c_i &= y_{i,i+1,i+2} - y_{i-1,i,i+1} && i = 1, 2, ..., n-2 \end{aligned} \tag{67}$$

and $y_{i,i+1,i+2}$ is the second divided difference[1] of the given $\{y_i\}$. The expressions for $T_{1\to 6}$ can be found after some algebra, such that 66 can be solved by Gaussian

[1]See de Boor (2001) pages 3-6 for a full explanation of the properties and consequences of divided differences. de Boor shows how divided differences can be used to build up the interpolating polynomial by adding knots in a piece-wise fashion.

elimination to find γ_j, thereby solving $S^{'''}(x)$. Recalling that $M_j(x)$ vanishes outside the interval (x_{j-1}, x_{j+2}) it is easy to derive the equations for D, E and F

$$D_j = \frac{10(\gamma_{i-1}h_i + \gamma_i h_{i-1})}{(h_{i-1} + h_i)} \tag{68}$$

$$E_i = \frac{5(\gamma_i - \gamma_{i-1})}{(h_{i-1} + h_i)} \tag{69}$$

$$F_i = (1/h_i)\left[\frac{(\gamma_{i+1} - \gamma_i)}{(h_i + h_{i+1})} - \frac{(\gamma_i - \gamma_{i-1})}{(h_{i-1} + h_i)}\right] \tag{70}$$

for $i = 2, 3, ..., n-3$. It is also possible to use the above formulae in the cases $i = 0, 1, n-2, n-1$, by adding the convention that $\gamma_{-1} = \gamma_0 = \gamma_{n-1} = \gamma_n = 0$. The last step is to make use of the continuity of $S(x)$ and its first four derivatives at x, to produce the results for B_i and C_i

$$B_i = \frac{h_{i-1}}{h_{i-1} + h_i}\frac{y_{i+1} - y_i}{h_i} + \frac{h_i}{h_{i-1} + h_i}\frac{y_i - y_{i-1}}{h_{i-1}} - D_i h_{i-1}h_i + \tag{71}$$

$$E_i h_{i-1}h_i(h_{i-1} - h_i) - \frac{h_{i-1}h_i}{h_{i-1} + h_i}(F_{i-1}h_{i-1}^3 F_i h_i^3) \tag{72}$$

$$C_i = \frac{1}{h_{i-1} + h_i}\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right) + D_i(h_{i-1} - h_i) - \tag{73}$$

$$E_i\left(\frac{h_{i-1}^3 + h_i^3}{h_{i-1} + h_i}\right) + \frac{1}{h_{i-1} + h_i}\left(F_i h_{i-1}^4 - F_i h_i^4\right) \tag{74}$$

where the above formulae are valid for $i = 1, 2, ..., n-1$. The end points are then given by the expressions

$$B_0 = \frac{(y_1 - y_0)}{h_0} - C_0 h_0 - F_0 h_0^4 \tag{75}$$

$$B_n = \frac{(y_n - y_{n-1})}{h_{n-1}} - C_n h_{n-1} - F_{n-1}h_{n-1}^4 \tag{76}$$

$$C_0 = C_1 - 10F_0 h_0^3 \tag{77}$$

$$C_n = C_{n-1} + 10F_{n-1}h_{n-1}^3 \tag{78}$$

Finally, it is also worth bearing in mind that the methodology will also deal with double and triple consecutive knot points in such a way taht if $x_j = x_{j+1}$, then $S(x_j) = y_j$ and $S^{'}(x_j) = y_{j+1}$; and if $x_j = x_{j+1}$, then in addition, $S^{''}(x_j) = y_{j+2}$.

8 Uniform tension splines

The method of minimum curvature is a popular approach to the construction of smooth surfaces from irregularly spaced data (see for example Briggs (1974) for a description of the general approach). The surface of minimum curvature corresponds to the minimum of a Laplacian power, or in an alternative formulation, satisfies the biharmonic differential equation. Physically, it models the behaviour of an elastic plate. In the one-dimensional case, the minimum curvature method leads to the natural cubic spline interpolation of de Boor (1978); whilst in the two dimensional case, a surface can be interpolated with biharmonic splines (see Sandwell (1987)), or gridded with an iterative finite difference method such as Swain (1976).

In most practical cases, the minimum curvature method produces a visually pleasing smooth surface. However, in cases of large changes in the surface gradient, the method can create strong artificial oscillations in the unconstrained regions. Switching to lower order methods, such as minimising the power of the gradient, solves the problem of extraneous inflections, but also removes the smoothness constraint and leads to gradient discontinuities (see Fomel and Claerbout (1995) for details). A remedy suggested by Schweikert (1966) is known as splines in tension. Splines in tension are constructed by minimising a modified quadratic form that includes a tension term. Physically, the additional term corresponds to tension in elastic plates as described by Timoshenko and Woinowsky-Kreiger (1968).

At a practical level, splines in tension use the traditional minimum variance criterion, which implies seeking a two-dimensional surface $f\,(x,y)$ in some region D, which corresponds to the minimum of the Laplacian power:

$$\iint\limits_D |\nabla^2 f\,(x,y)\,|^2 dxdy \tag{79}$$

where ∇^2 denotes the Laplacian operator

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \tag{80}$$

Alternatively, we can seek $f\,(x,y)$ as the solution of the biharmonic differential equation

$$(\nabla^2)^2 f\,(x,y) = 0 \tag{81}$$

Equation 79 corresponds to the normal system of equations in the standard least-square optimisation problem, the Laplacian operator being D and the surface $f\,(x,y)$ corresponding to the unknown model. Fung (1965) and Briggs (1974) derive 79 directly using variational calculus and Gauss's theorem.

Timoshenko and Woinowsky-Krieger (1968) make the interesting point that 79 approximates the strain energy of a thin elastic plate. Taking tension into account modifies both the energy formula and the corresponding equation 79. Smith and Wessel (1990) suggest the following form of the modified equation

$$\iint\limits_D \left[(1-\lambda)\,|\nabla^2 f\,(x,y)\,|^2 + \lambda|\nabla f\,(x,y)\,|^2\right] dxdy \tag{82}$$

Zero tension leads to the biharmonic equation 79 and corresponds to the minimum curvature construction. The case of $\lambda = 1$ corresponds to infinite tension. Although infinite tension is physically impossible, the resulting Laplace equation does have the physical interpretation of a steady-state temperature distribution. An important property of harmonic functions (solution of the Laplace equation) is that they cannot have local minima and maxima in the free regions. With respect to interpolation, this means that, in the case of $\lambda = 1$, the interpolation surface will be constrained to have its local extrema only at the input data locations.

It is also interesting to note that if the tension term $\lambda \nabla^2$ is written in the form $\nabla \cdot (\lambda \nabla)$, then the heat flow and electrostatics analogy can be followed and the tension parameter λ, can be generalised to a local function depending on x and y. In a more general form, λ could be a tensor allowing for an anisotropic smoothing in some pre-defined direction similarly to Clapp's steering-filter method in Clapp (1997).

To interpolate an irregular set of data values, f_k, at points (x_k, y_k), it is necessary to solve equation 82 under the constraint

$$f\left(x_k, y_k\right) = f_k \tag{83}$$

Practical implementation involves solving this constrained optimisation problem and using the results for successive interpolation of user supplied abscissae that have been presorted to ensure monotonicity and therefore maximum efficiency.

Whilst offering a distinct improvement over the basic cubic splines, single tension splines clearly suffer from the obvious deficiency that only one tension factor is applied over all sections of the data. This may of course not be appropriate for data sets that exhibit high volatility or strong trending behaviour. For such data it may be more appropriate to use an approach which endogenises the calculation of the tension factors for each segment to permit a stable and robust fit.

9 X-splines

The key problem, as with most spline methods is finding a family of functions $F_k(t)$ that sum to one whatever the value of t. X-splines are an entirely new approach to this problem in the calculation of splines. Blanc and Schlick construct their blending functions independently of any normality constraint, opting instead for a final normalisation step that replaces $F_k(t)$ by $\overline{F}_k(t)$:

$$\overline{F}_k(t) = \frac{F_k(t)}{\sum_{k=0}^{n} F_k(t)} \qquad \forall t \in [0,1] \tag{84}$$

These adapted blending functions, $\overline{F}_k(t)$, are normalised rational polynomials, which have the additional beneficial side-effect of projective invariance. The essential feature of the x-spline model is the non-null part of the blending function should be composed of only two segments. The first, referred to as $F_k^-(t)$, is defined between T_k^- and T_k, whilst the second is referred to as $F_k^+(t)$ is defined between T_k and T_k^+.

In order to illustrate the concept, consider a spline in which each control or knot point P_k influences four segments of the curve and therefore has L^4 locality - this would be the case for the classic cubic spline. By definition, for an L^4 spline, each blending function is non-null over four consecutive intervals of the knot vector, so that $F_k(t)$ becomes non-null at knot t_{k-2}, is maximal at knot t_k and becomes null again at knot t_{k+2} (as shown in figure 1). As $F_k(t)$ is composed of only two segments, it depends on t_{k-2} , t_k and t_{k+2}. There is therefore an alternation in the way the knots are taken into account such that even points use even knots whilst odd points use odd knots. In fact, the blending functions F_{k-2} and F_{k+2} cross each other at knot $t_k$and all the derivation of the model is based on this crossing. For that reason, Blanc and Schlick refer to their model as cross splines or x-splines for short.

Beginning with the case of a uniform knot vector, define Δ as follows

$$\forall_t \in [0,1] \qquad t_k - t_{k-1} = \Delta \tag{85}$$

so that applying the following re-parameterisation to the curve

$$u(t) = \frac{t - t_{k-2}}{t_k - t_{k-2}} = \frac{t - t_{k-2}}{2\Delta} \tag{86}$$

it is certain that $u = 0$ at knot t_{k-2} where $F_k(t)$ starts to increase and $u = 1$ at knot t_k where $F_k(t)$ reaches its maximum. It is therefore necessary to find a polynomial $f(u)$ defined on the range $[0,1]$ which can be linked to the left part of $F_k(t)$ by

$$F_k^-(t) = f\left(\frac{t - t_{k-2}}{2\Delta}\right) \tag{87}$$

Given the requirement for a C^2 continuous curve, the following constraints on the function $f(u)$ can be stated

$$f(0) = 0, \quad f^{'}(0) = 0 \;\text{ and }\; f^{''}(0) = 0 \tag{88}$$

it is also known that the maximum of the blending function is reached at $u = 1$ so that

$$f(1) = 1, \quad f^{'}(1) = 0 \;\text{ and }\; f^{''}(1) = -2p \tag{89}$$

such that the solution of this system of six constraints is the following polynomial

$$f_p(u) = u^3(10 - p + (2p - 15)u + (6 - p)u^2) \tag{90}$$

where $0 \leqslant p \leqslant 10$ due to regularity and the consequent need for a positive derivative. As the two functions $F_k^+(t)$ and $F_k^-(t)$ join at t_k, C^2 continuity is assured. The final formulation for an arbitrary segment of the curve $C(t)$ in the parameter range $[t_{k+1}, t_{k+2}]$ defined by the control points $P_k, P_{k+1}, P_{k+2}, P_{k+3}$ is

$$C(t) = \frac{A_0(t)P_k + A_1(t)P_{k+1} + A_2(t)P_{k+2} + A_3(t)P_{k+3}}{A_0(t) + A_1(t) + A_2(t) + A_3(t)} \tag{91}$$

where

$$\begin{array}{rr} A_0(t) = f_p\left(\frac{t_{k+2}-t}{2\Delta}\right) & A_1(t) = f_p(\frac{t_{k+3}-t}{2\Delta}) \\ A_2(t) = f_p(\frac{t-t_k}{2\Delta}) & A_3(t) = f_p(\frac{t-t_{k+1}}{2\Delta}) \end{array} \tag{92}$$

The final result is therefore a quintic rational approximation spline which includes the properties of normality, positivity, regularity, locality and C^2 continuity; along with p (in the range $p \in [0, 10]$) which provides a slight modification to the spline shape. It should be noted that when $p = 8$ the blending functions produce results very similar to the cubic spline method.

Blanc and Schlick take the x-spline model further by adding tension and angular shapes (G^0 continuity) via an extra parameter s which is a modified form of tension that determines the distance of points from the control lattice. Using s works as follows. Take the blending functions F_2, F_3 and F_4 as shown in figure 2. It is easily seen that F_3 reaches its maximum at t_3, but given that F_2 and F_4 are not null at t_3, the normalisation process sets the actual maximum to $F_3/(F_2+F_3+F_4)$. One way to increase the maximum in order to bring the curve closer to the knot point P_3 is to reduce $F_2(t_3)$ and $F_4(t_3)$. Within the region of interest, it is known that F_2 decreases whilst F_4 increases monotonically in the range $[t_2, t_4]$, so to obtain smaller values at t_3 it is necessary to increase the rate of change. To achieve these two operations simultaneously and symmetrically the crossing point of F_2 and F_4 is pushed down towards the horizontal axis. This is achieved by introducing a new degree of freedom $s_3 \in [0, 1]$ at point P_3. This parameter will be used first to compute the value of T_2^+ (where F_2 becomes zero) by interpolating between t_4 and t_3

$$T_2^+ = t_3 + s_3(t_4 - t_3) = t_3 + s_3\Delta \tag{93}$$

then to compute the value of T_4^- (where F_4 becomes non-zero) by interpolating between t_3 and t_2

$$T_4^- = t_3 + s_3(t_2 - t_3) = t_3 - s_3\Delta \tag{94}$$

The same operation being repeated for each k so that the resulting values T_k^- and T_k^+ have to be replaced in the reparameterisation equations as follows

$$F_k^-(t) = f_p\left(\frac{t - T_k^-}{t_k - T_k^-}\right) \qquad F_k^+(t) = f_p\left(\frac{t - T_k^+}{t_k - T_k^+}\right) \tag{95}$$

The two parts of F_k still join at t_k and their first derivatives are still null but their second derivatives are different

$$F_k^{''}(t_k^-) = \frac{-2p}{\left(t_k - T_k^-\right)^2} \quad \text{and} \quad F_k^{''}(t_k^+) = \frac{-2p}{\left(t_k - T_k^+\right)^2} \tag{96}$$

In order to equate the left and right expressions, the only the only solution is to use a specific value for p (denoted p_{k-1}) in F_k^- and a second denoted p_{k+1} in F_k^-

$$p_{k-1} = \frac{2\left(t_k - T_k^-\right)^2}{\Delta^2} \quad \text{and} \quad p_{k+1} = \frac{2\left(t_k - T_k^+\right)^2}{\Delta^2} \tag{97}$$

which gives

$$F_k^{''}(t_k^-) = F_k^{''}(t_k^+) = -\frac{4}{\Delta^2} \tag{98}$$

which provides C^2 continuity but also ensures that the parameters p_k are in the range $[0, 8]$ which is required to generate cubic B-splines at the limit.

It is therefore possible to produce a new formulation for the segment of the curve $C(t)$ on the range $[t_{k+1}, t_{k+2}]$ as defined by the four control points defined above

$$C(t) = \frac{A_0(t) P_k + A_1(t) P_{k+1} + A_2(t) P_{k+2} + A_3(t) P_{k+3}}{A_0(t) + A_1(t) + A_2(t) + A_3(t)} \tag{99}$$

$$\begin{array}{lll}
A_0(t) = t > T_k^+ & ?\ 0: & f_{pk-1}\left(\frac{t-T_k^+}{t_k - T_k^+}\right) \\
A_1(t) = t > T_{k+1}^+ & ?\ 0: & f_{pk}\left(\frac{t-T_{k+1}^+}{t_{k+1} - T_{k+1}^+}\right) \\
A_2(t) = t > T_{k+2}^- & ?\ 0: & f_{pk+1}\left(\frac{t-T_{k+2}^-}{t_{k+2} - T_{k+2}^-}\right) \\
A_3(t) = t > T_{k+3}^- & ?\ 0: & f_{pk+2}\left(\frac{t-T_{k+3}^-}{t_{k+3} - T_{k+3}^-}\right) \\
p_{k-1} = \frac{2}{\Delta^2}\left(t_k - T_k^+\right)^2 & & p_k = \frac{2}{\Delta^2}\left(t_{k+1} - T_{k+1}^+\right)^2 \\
p_{k+1} = \frac{2}{\Delta^2}\left(t_{k+2} - T_{k+2}^-\right)^2 & & p_{k+2} = \frac{2}{\Delta^2}\left(t_{k+3} - T_{k+3}^-\right)^2
\end{array} \tag{100}$$

The extended x-spline is therefore defined by a set of quadruples (x_k, y_k, z_k, s_k) with $k = 0, ..., n$ Clearly, the parameters $(x_k, y_k, z_k) \in \Re^3$ are the coordinates of the control points p_k. The parameter $s_k \in [0, 1]$ represents the distance between the curve and the control lattice, such that when $s_k = 1$ the curve passes relatively far away from the point P_k; and as s_k decreases the curve gets closer to P_k, finally passing through it when $s_k = 0$. By construction, the curve is always C^2 continuous, even when it interpolates the control point P_k, this enables the creation of sharp points or angular edges

The final stage is to include interpolation into the x-spline model. Achieving this means giving up positivity of the blending functions and the convex hull property. For some users this may be unacceptable, so Blanc and Schlick provide a second formulation of the x-spline model that will allow interpolation, thereby allowing users to choose which formulation most suits their purposes. The first step is to apply the following re-parameterisation of the curve

$$u(t) = \frac{t - t_k}{t_{k+1} - t_k} = \frac{t - t_k}{\Delta} \tag{101}$$

which ensures that $u = -1$ at knot t_{k-1} where F^-_{k+1} becomes negative, $u = 0$ at knot t_k where F^-_{k+1} becomes positive and $u = 1$ at knot t_{k+1} where F^-_{k+1} reaches its maximum. It is therefore necessary to find two polynomials: $g(u)$ defined on $[0, 1]$ which represents the positive part of F^-_{k+1} and $h(u)$ which represents the negative part. C^2 continuity is required at $u = 0$ where the two functions join. This gives rise to a series of 12 constraints

$$\begin{array}{lll} g(0) = 0 & g^{'}(0) = q & g^{''}(0) = 4q \\ g(1) = 1 & g^{'}(1) = 0 & g^{''}(1) = -2p \\ h(0) = 0 & h^{'}(0) = q & h^{''}(0) = 4q \\ h(-1) = 0 & h^{'}(-1) = 0 & h^{''}(-1) = 0 \end{array} \tag{102}$$

where q is a degree of freedom that controls the value of the first derivative at $u = 0$ All of the above constraints can be satisfied by the following two quintic polynomials

$$g(u) = qu + 2qu^2 + (10 - 12q - p)u^3 + \tag{103}$$

$$(2p + 14q - 15)u^4 + (6 - 5q - p)u^5 \tag{104}$$

$$h(u) = qu + 2u^2 - 2qu^4 + qu^5 \tag{105}$$

Starting with these polynomials and following the same construction process already described, a rational quintic spline interpolation model which includes the properties of normality, locality and C^2 continuity can be constructed. The parameter q allows modification of the shape of the curve. It is important to note that as with every spline model, the curve may exhibit unwanted oscillations. The way to limit this behaviour is to limit q to the range $[0, \frac{1}{2}]$. At $q = \frac{1}{2}$ the blending functions are very close to the Catmull-Rom spline model, but the functions are C^2 not C^1 continuous.

In order to simplify the user interface, it makes sense to merge the parameter s of the approximation model with the parameter q of the interpolation model. Blanc and Schlick use only one shape parameter using the following convention:

- When the user sets all s_k to be in the range $[0, 1]$, it is taken to imply that the user wishes to use approximation splines. In the particular case of $s_k = 1$ cubic B-splines will be approximated.
- When the users sets all s_k in the range $[-1, 0]$ it is taken to imply that the user wishes to use interpolation splines. In such a case, q_k is obtained from s_k by $q_k = -s_k/2$. Therefore, $s_k = -1$ gives $q = \frac{1}{2}$ which approximates the Catmull-Rom splines.

10 Some practical issues

Each type of spline model has its own particular advantages and disadvantages, the relevance of which depend upon the requirements of the task in hand. For example, the main problem with cubic b-splines is that they almost always force the second derivative to be zero at the beginning and end of the data set. Clearly, this is not a realistic general assumption and the result is that the interpolation can be fatally flawed around both the beginning and end of the data. X-splines on the other hand have the key advantage that they can support both approximation and interpolation methods using the same basic framework but different forms of the underlying quintic polynomial. The feature list provided courtesy of Blanc and Schlick at the beginning of this document is an invaluable aid to deciding which form of spline model best suits the task in hand. The main aim of this paper is not to recommend one particular spline model over another, rather it is to acquaint the user with the basic mathematical approach underlying the types of spline currently supported in this release of the Addix Spline library.

The bibliography at the end of this paper is a short list of the main papers connected with each of the types of spline implemented in this version of the Addix Spline Library. The interested reader can of course go much further...

References

[1] Blanc, C. and Schlick, C., "X-Splines: A Spline Model Designed for the End-User", Laboratoire Bordelais de Recherche en Informatique.

[2] de Boor, C., (1978), "A practical guide to splines", Springer-Verlag.

[3] Cline, A.K., (1974), "Scalar and planar-valued curve fitting using splines under tension", Communications of the Association of Computational Mathematics, 17, 4, April, 218-220.

[4] Catmull, E. and Rom, R., (1974), "A Class of Local Interpolating Splines", in Barnhill R.E. and Riesenfled (eds), *Computer Aided Geometric Design, Academic Press*, New York, 1974.

[5] Eidson, H.D, and Schumaker, L.L., (1974), "Computation of g-Splines via a Factorization Method", Communications of ACM, September 1974, Volume 17, No. 9.

[6] Farin, G., (1990), "Curves and Surfaces for Computer Aided Design - A Practical Guide", Academic Press.

[7] Farin, G., (1995), "NURB Curves and Surfaces: From Projective Geometry to Practical Use", Peters Press.

[8] Fomel, S. and Claerbout, J., (1995), "Searching the Sea of Galilee: the splendours and miseries of iteratively reweighted least squares", Preprint.

[9] Fung, Y.C., (1965), "Foundations of solid mechanics", Prentice-Hall.

[10] Grenville, T.N.E., (1967), “Introduction to spline functions”, In Theory and Applications of Spline Functions, Academic Press, New York.

[11] Herriot, J.G. and Reinsch, C.H., (1976), “Procedures for Quintic Natural Spline Interpolation”, Association for Computing Machinery, Transactions on Mathematical Software, Volume2, Number 3, September.

[12] Renka, R.J., (1987), “Interpolatory tension splines with automatic selection of tension factors”, SIAM Journal of Scientific and Statistical Computing, Volume 8, p. 393-415.

[13] Rogers, D.F., (2001), “An Introduction to NURBS with historical perspective”, Academic Press.

[14] Sandwell, D.T., (1987), “Biharmonic spline interpolation of GEOS-3 and SEASAT altimeter data”, Geophysical Research Letters, 14, no. 2, 139-142.

[15] Schweikert, D.G., (1966), “An interpolation curve using a spline in tension”, Journal of Mathematics and Physics, 45, 312-313.

[16] Swain, C.J., (1976), “Interpolating irregularly spaced data using the difference equations for minimum curvature”, Computers and Geosciences, 1, 231-240.

[17] Timoshenko, S. and Woinowsky-Kreiger, S., (1968), “Theory of plates and shells”, McGraw-Hill.