Chapter 6

B-Spline Curves

Most shapes are too complicated to define using a single Bézier curve. A spline curve is a sequence of curve segments that are connected together to form a single continuous curve. For example, a piecewise collection of Bézier curves, connected end to end, can be called a spline curve. Overhauser curves are another example of splines. The word "spline" is sometimes used as a verb, as in "Spline together some cubic Bézier curves." In approximation theory, spline is defined as a piecewise polynomial of degree n whose segments are C^{n-1}.

The word "spline" comes from the ship building industry, where it originally referred to a thin strip of wood which draftsmen would use like a flexible French curve. Metal weights (called ducks) were placed on the drawing surface and the spline was threaded between the ducks as in Figure 6.1. We know from structural mechanics that the bending moment M is an infinitely continuous function along the spline except at a duck, where M is generally only C^0 continuous. Since the curvature of the spline is proportional to M ($\kappa = M/EI$), the spline is everywhere curvature continuous.

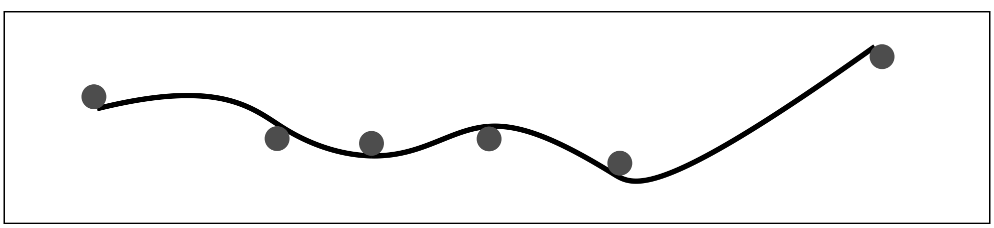


Figure 6.1: Spline and ducks.

Curvature continuity is an important requirement for the ship building industry, as well as for many other applications. For example, railroad tracks are always curvature continuous, or else a moving train would experience severe jolts. Car bodies are G^2 smooth, or else the reflection of straight lines would appear to be G^0.

While C^1 continuity is straightforward to attain using Bézier curves (for example, popular design software such as Adobe Illustrator use Bézier curves and automatically impose tangent continuity as you sketch), C^2 and higher continuity is cumbersome. This is where B-spline curves come in. B-spline curves can be thought of as a method for defining a sequence of degree n Bézier curves that join automatically with C^{n-1} continuity, regardless of where the control points are placed.

Whereas an open string of m Bézier curves of degree n involve $nm + 1$ distinct control points (shared control points counted only once), that same string of Bézier curves can be expressed using only $m+n$ B-spline control points (assuming all neighboring curves are C^{n-1}). The most important operation you need to understand to have a working knowledge of B-splines is how to extract the contituent Bézier curves. The traditional approach to teaching about B-Splines centered on basis functions and recurrence relations. Experience has shown that polar form (Section 6.1) and knot intervals (Section 6.12) provide students with such a working knowledge more quickly than recurrence relations.

6.1 Polar Form

Polar form, introduced by Dr. Lyle Ramshaw [Ram87, Ram89b, Ram89c], can be thought of as simply an alternative method of labeling the control points of a Bézier or B-Spline curve. These labels are referred to as *polar values*. These notes summarize the properties and applications of polar form, without delving into derivations. The interested student can study Ramshaw's papers.

All of the important algorithms for Bézier and B-spline curves can be derived from the following four rules for polar values.

1. For a degree n Bézier curve $\mathbf{P}_{[a,b]}(t)$, the control points are relabeled $\mathbf{P}_i = \mathbf{P}(u_1, u_2, \dots u_n)$ where $u_j = a$ if $j \le n - i$ and otherwise $u_j = b$. For a degree two Bézier curve $\mathbf{P}_{[a,b]}(t)$,

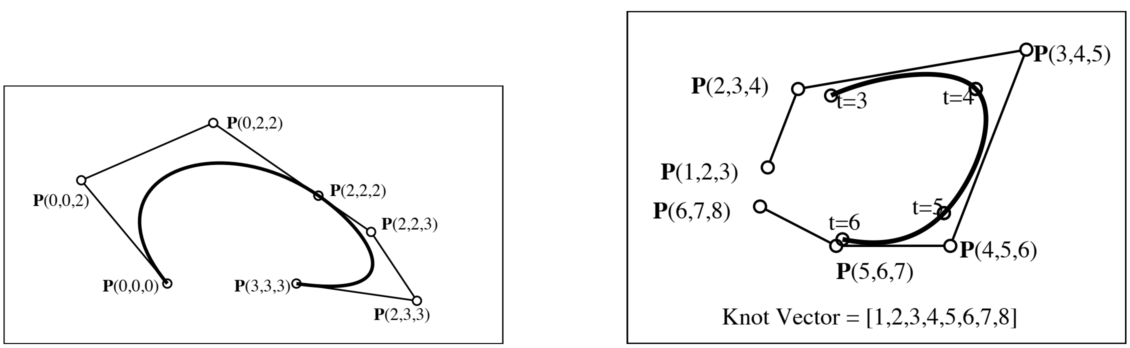
$$\mathbf{P}_0 = \mathbf{P}(a,a); \quad \mathbf{P}_1 = \mathbf{P}(a,b); \quad \mathbf{P}_2 = \mathbf{P}(b,b).$$

For a degree three Bézier curve $\mathbf{P}_{[a,b]}(t)$,

$$\mathbf{P}_0 = \mathbf{P}(a,a,a); \quad \mathbf{P}_1 = \mathbf{P}(a,a,b);$$

$$\mathbf{P}_2 = \mathbf{P}(a,b,b); \quad \mathbf{P}_3 = \mathbf{P}(b,b,b),$$

and so forth.



(a) Bézier curves with polar labels.

(b) B-Spline Curve with Polar Labels

Figure 6.2: Polar Labels.

Figure 6.2.a shows two cubic Bézier curves labeled using polar values. The first curve is defined over the parameter interval $[0, 2]$ and the second curve is defined over the parameter interval $[2, 3]$. Note that $\mathbf{P}(t, t, \dots, t)$ is the point on a Bézier curve corresponding to parameter value t.

2. For a degree n B-spline with a *knot vector* (see Section 6.2) of

$$[t_1, t_2, t_3, t_4, \dots],$$

the arguments of the polar values consist of groups of n adjacent knots from the knot vector, with the i^{th} polar value being $\mathbf{P}(t_i, \ldots, t_{i+n-1})$, as in Figure 6.2.b.
3. A polar value is symmetric in its arguments. This means that the order of the arguments can be changed without changing the polar value. For example,

$$\mathbf{P}(1,0,0,2) = \mathbf{P}(0,1,0,2) = \mathbf{P}(0,0,1,2) = \mathbf{P}(2,1,0,0),\ \textit{etc.}$$

4. Given $\mathbf{P}(u_1, u_2, \ldots, u_{n-1}, a)$ and $\mathbf{P}(u_1, u_2, \ldots, u_{n-1}, b)$ we can compute $\mathbf{P}(u_1, u_2, \ldots, u_{n-1}, c)$ where c is any value:

$$\mathbf{P}(u_1, u_2, \ldots, u_{n-1}, c) =$$
$$\frac{(b-c)\mathbf{P}(u_1, u_2, \ldots, u_{n-1}, a) + (c-a)\mathbf{P}(u_1, u_2, \ldots, u_{n-1}, b)}{b-a}$$

$\mathbf{P}(u_1, u_2, \ldots, u_{n-1}, c)$ is said to be an *affine combination* of $\mathbf{P}(u_1, u_2, \ldots, u_{n-1}, a)$ and $\mathbf{P}(u_1,\ u_2, \ldots, u_{n-1}, b)$. For example,

$$\mathbf{P}(0,t,1) = (1-t) \times \mathbf{P}(0,0,1) + t \times \mathbf{P}(0,1,1),$$

$$\mathbf{P}(0,t) = \frac{(4-t) \times \mathbf{P}(0,2) + (t-2) \times \mathbf{P}(0,4)}{2},$$

$$\mathbf{P}(1,2,3,t) = \frac{(t_2-t) \times \mathbf{P}(2,1,3,t_1) + (t-t_1) \times \mathbf{P}(3,2,1,t_2)}{(t_2-t_1)}.$$

What this means geometrically is that if you vary one parameter of a polar value while holding all others constant, the polar value will sweep out a line at a constant velocity, as in Figure 6.3.

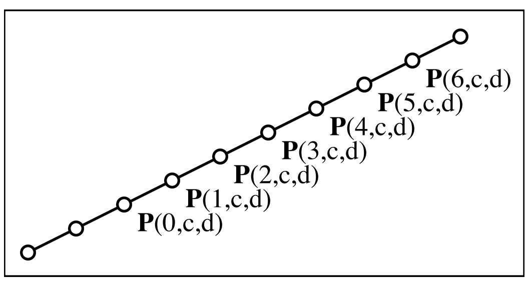


Figure 6.3: Affine map property of polar values.

6.1.1 Subdivision of Bézier Curves

To illustrate how polar values work, we now show how to derive the de Casteljau algorithm using only the first three rules for polar values.

Given a cubic Bézier curve $\mathbf{P}_{[0,1]}(t)$, we wish to split it into $\mathbf{P}_{[0,t]}$ and $\mathbf{P}_{[t,1]}$. The control points of the original curve are labeled

$$\mathbf{P}(0,0,0), \quad \mathbf{P}(0,0,1), \quad \mathbf{P}(0,1,1), \quad \mathbf{P}(1,1,1).$$

The subdivision problem amounts to finding polar values

$$\mathbf{P}(0,0,0), \quad \mathbf{P}(0,0,t), \quad \mathbf{P}(0,t,t), \quad \mathbf{P}(t,t,t),$$

and

$$\mathbf{P}(t,t,t), \quad \mathbf{P}(t,t,1), \quad \mathbf{P}(t,1,1), \quad \mathbf{P}(1,1,1).$$

These new control points can be derived by applying the symmetry and affine map rules for polar values. Refering to Figure 6.4, we can compute

STEP 1.

$$\mathbf{P}(0,0,t) = (1-t)\times\mathbf{P}(0,0,0) + (t-0)\times\mathbf{P}(0,0,1);$$
$$\mathbf{P}(0,1,t) = (1-t)\times\mathbf{P}(0,0,1) + (t-0)\times\mathbf{P}(0,1,1);$$
$$\mathbf{P}(t,1,1) = (1-t)\times\mathbf{P}(0,1,1) + (t-0)\times\mathbf{P}(1,1,1).$$

STEP 2.

$$\mathbf{P}(0,t,t) = (1-t)\times\mathbf{P}(0,0,t) + (t-0)\times\mathbf{P}(0,t,1);$$
$$\mathbf{P}(1,t,t) = (1-t)\times\mathbf{P}(0,t,1) + (t-0)\times\mathbf{P}(t,1,1);$$

STEP 3.

$$\mathbf{P}(t,t,t) = (1-t)\times\mathbf{P}(0,t,t) + (t-0)\times\mathbf{P}(t,t,1);$$

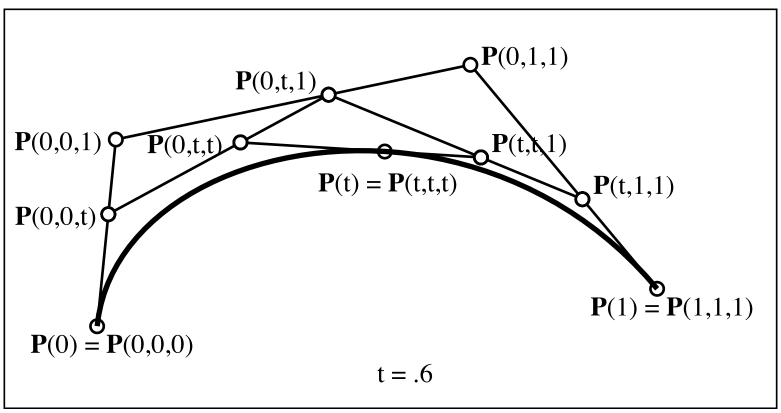


Figure 6.4: Subdividing a cubic Bézier curve.

6.2 Knot Vectors

A knot vector is a list of parameter values, or *knots*, that specify the parameter intervals for the individual Bézier curves that make up a B-spline. For example, if a cubic B-spline is comprised of four Béziercurves with parameter intervals $[1,2]$, $[2,4]$, $[4,5]$, and $[5,8]$, the knot vector would be

$$[t_0, t_1, 1, 2, 4, 5, 8, t_7, t_8].$$

Notice that there are two (one less than the degree) extra knots prepended and appended to the knot vector. These knots control the *end conditions* of the B-spline curve, as discussed in Section 6.6.

For historical reasons, knot vectors are traditionally described as requiring n end-condition knots, and in the real world you will always find a (meaningless) additional knot at the beginning and end of a knot vector. For example, the knot vector in Figure 6.2.b would be $[t_0, 1, 2, 3, 4, 5, 6, 7, 8, t_9]$, where the values of t_0 and t_9 have absolutely no effect on the curve. Therefore, we ignore these dummy knot values in our discussion, but be aware that they appear in B-spline literature and software.

Obviously, a knot vector must be non-decreasing sequence of real numbers. If any knot value is repeated, it is referred to as a *multiple knot*. More on that in Section 6.4. A B-spline curve whose knot vector is evenly spaced is known as a *uniform* B-spline. If the knot vector is not evenly spaced, the curve is called a *non-uniform* B-spline.

6.3 Extracting Bézier Curves from B-splines

We are now ready to discuss the central practical issue for B-splines, namely, how does one find the control points for the Bézier curves that make up a B-spline. This procedure is often called the Böhm algorithm after Professor Wolfgang Böhm [B8̈1].

Consider the B-spline in Figure 6.2.b consisting of Bézier curves over domains $[3,4]$, $[4,5]$, and $[5,6]$. The control points of those three Bézier curves have polar values

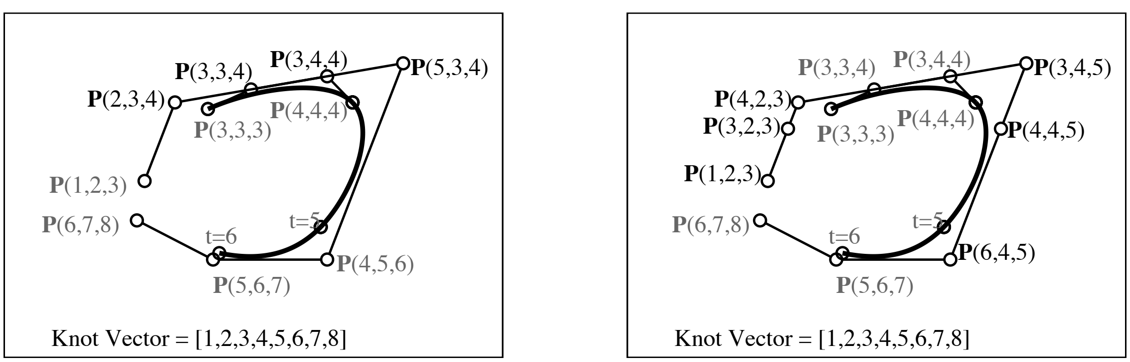
$$\mathbf{P}(3,3,3),\quad \mathbf{P}(3,3,4),\quad \mathbf{P}(3,4,4),\quad \mathbf{P}(4,4,4)$$

$$\mathbf{P}(4,4,4),\quad \mathbf{P}(4,4,5),\quad \mathbf{P}(4,5,5),\quad \mathbf{P}(5,5,5)$$

$$\mathbf{P}(5,5,5),\quad \mathbf{P}(5,5,6),\quad \mathbf{P}(5,6,6),\quad \mathbf{P}(6,6,6)$$

respectivly. Our puzzle is to apply the affine and symmetry properties to find those polar values given the B-spline polar values.

For the Bézier curve over $[3,4]$, we first find that $\mathbf{P}(3,3,4)$ is 1/3 of the way from $\mathbf{P}(2,3,4)$ to $\mathbf{P}(5,3,4) = \mathbf{P}(3,4,5)$. Likewise, $\mathbf{P}(3,4,4)$ is 2/3 of the way from $\mathbf{P}(3,4,2) = \mathbf{P}(2,3,4)$ to $\mathbf{P}(3,4,5)$. See Figure 6.5.a.



(a) First Step. (b) Second Step.

Figure 6.5: Böhm algorithm.

Before we can locate $\mathbf{P}(3,3,3)$ and $\mathbf{P}(4,4,4)$, we must find the auxilliary points $\mathbf{P}(3,2,3)$ (2/3 of the way from $\mathbf{P}(1,2,3)$ to $\mathbf{P}(4,2,3)$) and $\mathbf{P}(4,4,5)$ (2/3 of the way from $\mathbf{P}(3,4,5)$ to $\mathbf{P}(6,4,5)$)

as shown in Figure 6.5.b. Finally, $\mathbf{P}(3,3,3)$ is seen to be half way between $\mathbf{P}(3,2,3)$ and $\mathbf{P}(3,3,4)$, and $\mathbf{P}(4,4,4)$ is seen to be half way between $\mathbf{P}(3,4,4)$ and $\mathbf{P}(4,4,5)$.

Note that the four Bézier control points were derived from exactly four B-spline control points; $\mathbf{P}(5,6,7)$ and $\mathbf{P}(6,7,8)$ were not involved. This means that $\mathbf{P}(5,6,7)$ and $\mathbf{P}(6,7,8)$ can be moved without affecting the Bézier curve over $[3,4]$. In general, the Bézier curve over $[t_i, t_{i+1}]$ is only influenced by B-spline control points that have t_i or t_{i+1} as one of the polar value parameters. For this reason, B-splines are said to possess the property of *local control*, since any given control point can influence at most n curve segments.

6.4 Multiple knots

If a knot vector contains two identical non-end-condition knots $t_i = t_{i+1}$, the B-spline can be thought of as containing a zero-length Bézier curve over $[t_i, t_{i+1}]$. Figure 6.9 shows what happens when two knots are moved together. The Bézier curve over the degenerate interval $[5,5]$ has polar values $\mathbf{P}(5,5,5)$, $\mathbf{P}(5,5,5)$, $\mathbf{P}(5,5,5)$, $\mathbf{P}(5,5,5)$, which is merely the single point $\mathbf{P}(5,5,5)$. It can be

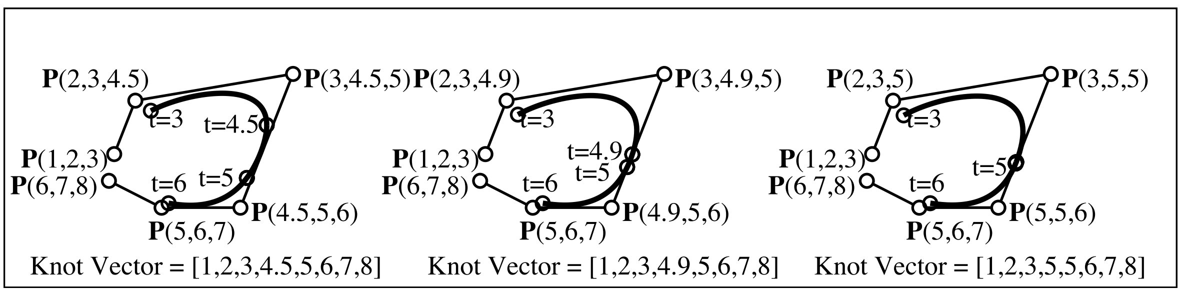


Figure 6.6: Double knot.

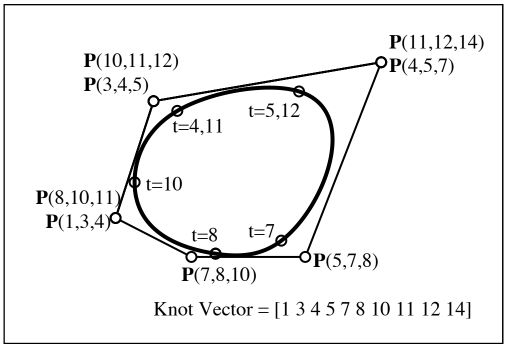
shown that a multiple knot diminishes the continuity between adjacent Bézier curves. The continuity across a knot of multiplicity k is generally $n - k$.

6.5 Periodic B-splines

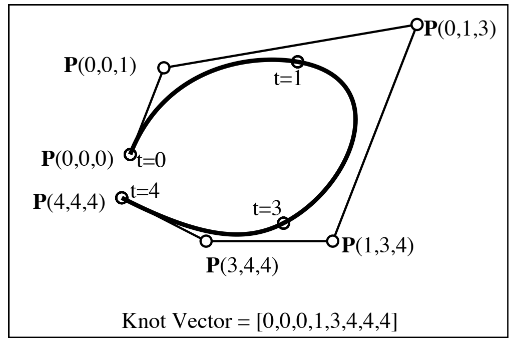
A periodic B-spline is a B-spline which closes on itself. This requires that the first n control points are identical to the last n, and the first n parameter intervals in the knot vector are identical to the last n intervals as in Figure 6.7.a.

6.6 Bézier end conditions

We earlier noted that a knot vector always has $n-1$ extra knots at the beginning and end which do not signify Bézier parameter limits (except in the periodic case), but which influence the shape of the curve at its ends. In the case of an open (i.e., non-periodic) B-spline, one usually chooses an n-fold knot at each end. This imposes a Bézier behavior on the end of the B-spline, in that the curve interpolates the end control points and is tangent to the control polygon at its endpoints. One can verify this by noting that to convert such a B-spline into Bézier curves, the two control points at each end are already in Bézier form. This is illustrated in Figure 6.7.b.



(a) Periodic B-Spline.



(b) Bézier End Conditions.

Figure 6.7: Special B-Spline Curves.

6.7 Knot insertion

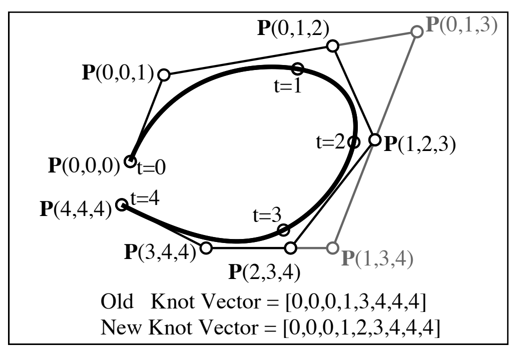
A standard design tool for B-splines is knot insertion. In the knot insertion process, a knot is added to the knot vector of a given B-spline. This results in an additional control point and a modification of a few existing control points. The end result is a curve defined by a larger number of control points, but which defines exactly the same curve as before knot insertion.

Knot insertion has several applications. One is the de Boor algorithm for evaluating a B-spline (discussed in the next section). Another application is to provide a designer with the ability to add local details to a B-spline. Knot insertion provides more local control by isolating a region to be modified from the rest of the curve, which thereby becomes immune from the local modification.

Consider adding a knot at $t = 2$ for the B-spline in Figure **??**. As shown in Figure 6.8.a, this involves replacing $\mathbf{P}(0,1,3)$ and $\mathbf{P}(1,3,4)$ with $\mathbf{P}(0,1,2)$, $\mathbf{P}(1,2,3)$, and $\mathbf{P}(2,3,4)$. Figure 6.8.b shows the new set of control points, which are easily obtained using the affine and symmetry properties of polar values.

	Initial	**After Knot Insertion**
Knot Vector:	[(0,0,0,1,3,4,4,4)]	[(0,0,0,1,2,3,4,4,4)]
Control Points:	**P**(0,0,0)	**P**(0,0,0)
	P(0,0,1)	**P**(0,0,1)
		P(0,1,2)
	P(0,1,3)	
		P(1,2,3)
	P(1,3,4)	
		P(2,3,4)
	P(3,4,4)	**P**(3,4,4)
	P(4,4,4)	**P**(4,4,4)

(a) Knots and Polar Labels.



(b) Control Points.

Figure 6.8: Knot Insertion.

Note that the continuity at $t = 2$ is C^∞.

Example

This degree four polynomial Bézier curve begins at $t = 1$ and ends at $t = 5$, and is therefore also a B-spline with knot vector [11115555]. Insert a knot at $t = 3$. State the control point coordinates and knot vector after this knot insertion is performed.

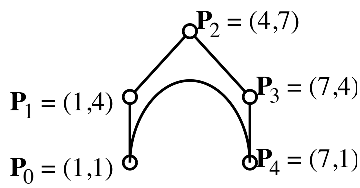


Figure 6.9: B-spline with knot vector [11115555].

Solution

The control points whose polar labels are $f(1,1,1,5) = (1,4)$, $f(1,1,5,5) = (4,7)$ and $f(1,5,5,5) = (7,4)$ will be replaced with polar labels $f(1,1,1,3) = (1,2.5)$, $f(1,1,3,5) = (2.5,5.5)$, $f(1,3,5,5) = (5.5,5.5)$, and $f(3,5,5,5) = (7,2.5)$.

6.8 The de Boor algorithm

The de Boor algorithm provides a method for evaluating a B-spline curve. That is, given a parameter value, find the point on the B-spline corresponding to that parameter value.

Any point on a B-spline $\mathbf{P}(t)$ has a polar value $\mathbf{P}(t, t, \ldots, t)$, and we can find it by inserting knot t n times. This is the de Boor algorithm. Using polar forms, the algorithm is easy to figure out.

The de Boor algorithm is illustrated in Figure 6.10.

6.9 Explicit B-splines

Section 2.11 discusses explicit Bézier curves, or curves for which $x(t) = t$. We can likewise locate B-spline control points in such a way that $x(t) = t$. The x coordinates for an explicit B-spline are known as *Greville abscissae*. For a degree n B-spline with m knots in the knot vector, the Greville abscissae are given by

$$x_i = \frac{1}{n}(t_i + t_{i+1} + \ldots + t_{i+n-1}); \quad i = 0 \ldots m - n. \tag{6.1}$$

6.10 B-spline hodographs

The first derivative (or hodograph) of a B-spline is obtained in a manner similar to that for Bézier curves. The hodograph has the same knot vector as the given B-spline except that the first and last knots are discarded. The control points are given by the equation

$$\mathbf{H}_i = n\frac{(\mathbf{P}_{i+1} - \mathbf{P}_i)}{t_{i+n} - t_i} \tag{6.2}$$

where n is the degree.

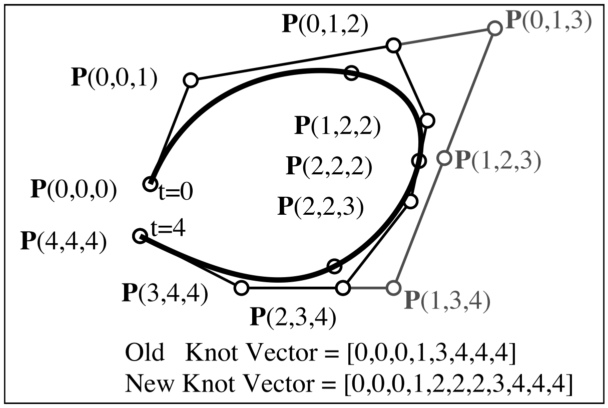


Figure 6.10: De Boor algorithm.

6.11 Symmetric polynomials

We introduced polar form as simply a labeling scheme, with rules defined for creating new control points with different labels. As we have seen, this level of understanding is sufficient to perform many of the basic operations on B-Spline curves. However, there is some beautiful mathematics behind these labels, based on symmetric polynomials.

A symmetric polynomial represents a degree m polynomial in one variable, $p(t)$, as a polynomial in $n \geq m$ variables, $p[t_1, \ldots, t_n]$, that is degree one in each of those variables and such that

$$p[t, \ldots, t] = p(t).$$

$p[t_1, \ldots, t_n]$ is said to be symmetric because the value of the polynomial will not change if the arguments are permuted. For example, if $n = 3$, $p[a, b, c] = p[b, c, a] = p[c, a, b]$ etc.

A symmetric polynomial has the form

$$p[t_1, \ldots, t_n] = \sum_{i=0}^{n} c_i p_i[t_1, \ldots, t_n]$$

where

$$p_0[t_1, \ldots, t_n] = 1; \quad p_i[t_1, \ldots, t_n] = \frac{\sum_{j=1}^{n} t_j p_{i-1}[t_1, \ldots, t_{j-1}, t_{j+1}, \ldots, t_n]}{n}, \quad i = 1, \ldots n.$$

For example,

$$p[t_1] = c_0 + c_1 t_1,$$

$$p[t_1, t_2] = c_0 + c_1 \frac{t_1 + t_2}{2} + c_2 t_1 t_2,$$

$$p[t_1, t_2, t_3] = c_0 + c_1 \frac{t_1 + t_2 + t_3}{3} + c_2 \frac{t_1 t_2 + t_1 t_3 + t_2 t_3}{3} + c_3 t_1 t_2 t_3,$$

and

$$p[t_1, t_2, t_3, t_4] = c_0 + c_1 \frac{t_1 + t_2 + t_3 + t_4}{4} + c_2 \frac{t_1 t_2 + t_1 t_3 + t_1 t_4 + t_2 t_3 + t_2 t_4 + t_3 t_4}{6}$$
$$+ c_3 \frac{t_1 t_2 t_3 + t_1 t_2 t_4 + t_1 t_3 t_4 + t_2 t_3 t_4}{4} + c_4 t_1 t_2 t_3 t_4.$$

The symmetric polynomial $b[t_1, \ldots t_n]$ for which $p[t, \ldots t] = p(t)$ is referred to as the *polar form* or *blossom* of $p(t)$.

Example Find the polar form of $p(t) = t^3 + 6t^2 + 3t + 1$.
Answer: $p[t_1, t_2, t_3] = 1 + 3\frac{t_1+t_2+t_3}{3} + 6\frac{t_1t_2+t_1t_3+t_2t_3}{3} + t_1t_2t_3$.

Theorem For every degree m polynomial $p(t)$ there exists a unique symmetric polynomial $p[t_1, \ldots, t_n]$ of degree $n \geq m$ such that $p[t, \ldots, t] = p(t)$. Furthermore, the coefficients b_i of the degree n Bernstein polynomial over the interval $[a, b]$ are

$$b_i = p[\underbrace{a, \ldots, a}_{n-i}, \underbrace{b, \ldots, b}_{i}]$$

Example Convert $p(t) = t^3 + 6t^2 + 3t + 1$ to a degree 3 Bernstein polynomial over the interval $[0, 1]$. We use the polar form of $p(t)$: $p[t_1, t_2, t_3] = 1 + 3\frac{t_1+t_2+t_3}{3} + 6\frac{t_1t_2+t_1t_3+t_2t_3}{3} + t_1t_2t_3$. Then,

$$b_0 = p[0, 0, 0] = 1, \quad b_1 = p[0, 0, 1] = 2, b_2 = p[0, 1, 1] = 5, b_3 = p[1, 1, 1] = 11.$$

6.12 Knot Intervals

B-spline curves are typically specified in terms of a set of control points, a knot vector, and a degree. Knot information can also be imposed on a B-spline curve using knot intervals, introduced in [SZSS98] as a way to assign knot information to subdivision surfaces. A knot interval is the difference between two adjacent knots in a knot vector, i.e., the parameter length of a B-spline curve segment. For even-degree B-spline curves, a knot interval is assigned to each control point, since each control point in an even-degree B-spline corresponds to a curve segment. For odd-degree B-spline curves, a knot interval is assigned to each control polygon edge, since in this case, each edge of the control polygon maps to a curve segment.

While knot intervals are basically just an alternative notation for representing knot vectors, knot intervals offer some nice advantages. For example, knot interval notation is more closely coupled to the control polygon than is knot vector notation. Thus, knot intervals have more geometric meaning than knot vectors, since the effect of altering a knot interval can be more easily predicted. Knot intervals are particularly well suited for periodic B-splines.

Knot intervals contain all of the information that a knot vector contains, with the exception of a knot origin. This is not a problem, since the appearance of a B-spline curve is invariant under linear transformation of the knot vector—that is, if you add any constant to each knot the curve's appearance does not change. B-splines originated in the field of approximation theory and were initially used to approximate functions. In that context, parameter values are important, and hence, knot values are significant. However, in curve and surface shape design, we are almost never concerned about absolute parameter values.

For odd-degree B-spline curves, the knot interval d_i is assigned to the control polygon edge $\mathbf{P}_i$—$\mathbf{P}_{i+1}$. For even-degree B-spline curves, knot interval d_i is assigned to control point $\mathbf{P}_i$. Each vertex (for even degree) or edge (for odd degree) has exactly one knot interval. If the B-spline is not periodic, $\frac{n-1}{2}$ "end-condition" knot intervals must be assigned past each of the two end control points. They can simply be written adjacent to "phantom" edges or vertices sketched adjacent to the end control points; the geometric positions of those phantom edges or vertices are immaterial.

Figure 6.11 shows a cubic B-spline curve. The control points in Figure 6.11.a are labeled with polar values, and Figure 6.11.b shows the control polygon edges labeled with knot intervals. End-

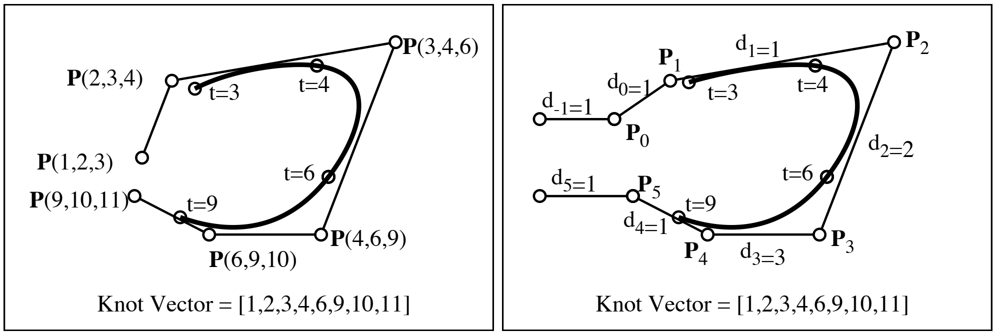


Figure 6.11: Sample cubic B-spline

condition knots require that we hang one knot interval off each end of the control polygon. Note the relationship between the knot vector and the knot intervals: Each knot interval is the difference between two consecutive knots in the knot vector.

For periodic B-splines, things are even simpler, since we don't need to deal with end conditions. Figure 6.12 shows two cubic periodic B-splines labelled with knot intervals. In this example, note

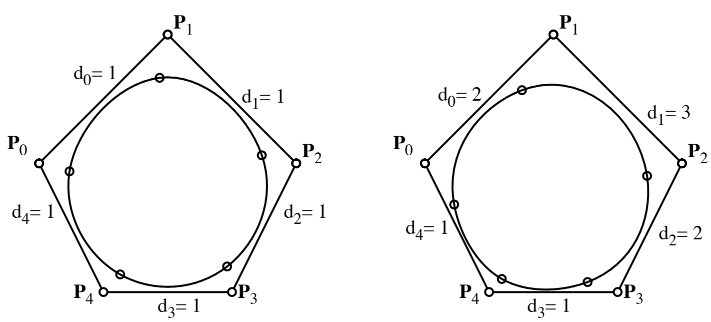


Figure 6.12: Periodic B-splines labelled with knot intervals

that as knot interval d_1 changes from 1 to 3, the length of the corresponding curve segment increases.

Figure 6.13 shows two periodic B-splines with a double knot (imposed by setting $d_0 = 0$) and a triple knot (set $d_0 = d_1 = 0$).

In order to determine formulae for operations such as knot insertion in terms of knot intervals, it is helpful to infer polar labels for the control points. Polar algebra [Ram89a] can then be used to create the desired formula. The arguments of the polar labels are sums of knot intervals. We are free to choose any knot origin. For the example in Figure 6.14, we choose the knot origin to coincide with control points $\mathbf{P}_0$. Then the polar values are as shown in Figure 6.14.b.

The following subsections show how to perform knot insertion and interval halving, and how to compute hodographs using knot intervals. These formulae can be verified using polar labels. The expressions for these operations written in terms of knot vectors can be found, for example, in [HL93].

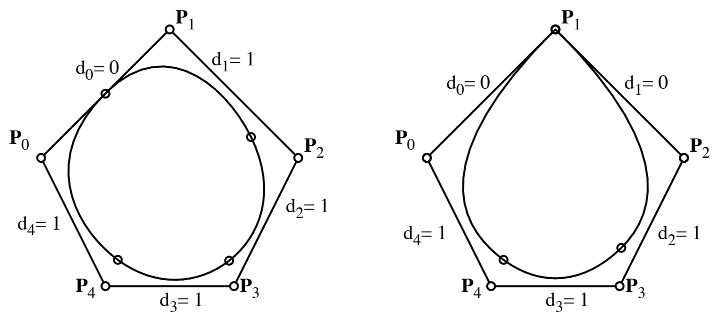


Figure 6.13: Periodic B-splines with double and triple knots.

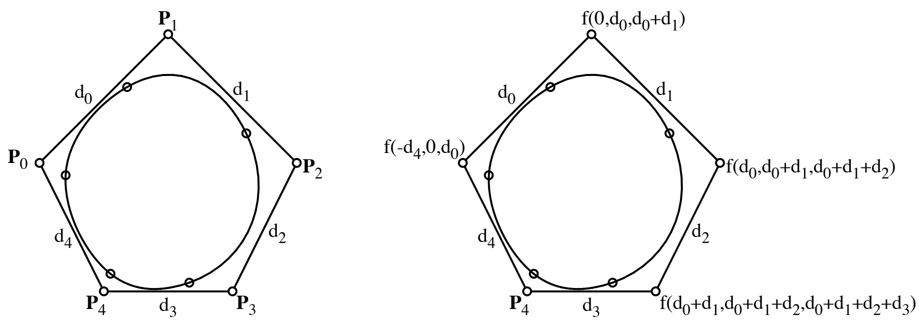


Figure 6.14: Inferring polar labels from knot intervals.

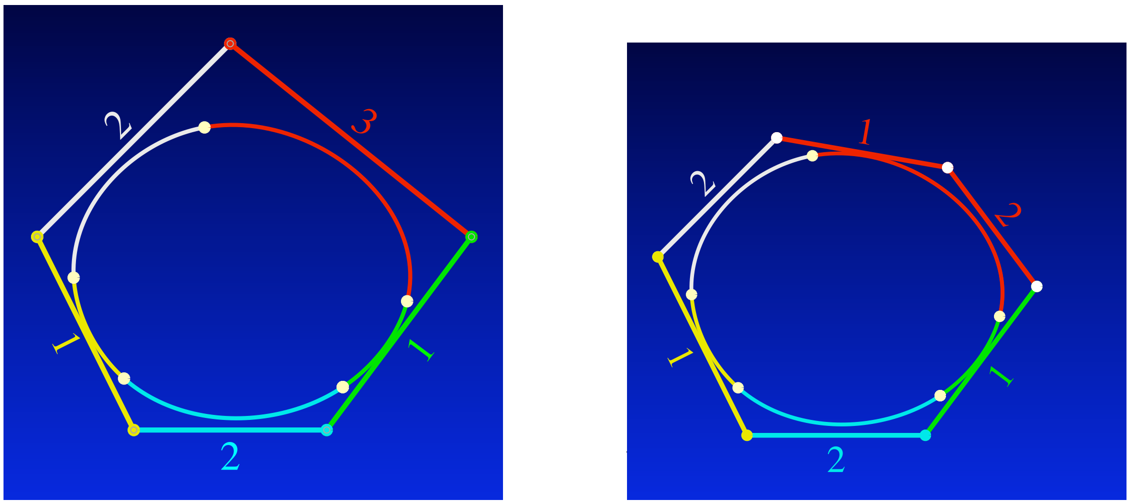
6.12.1 Knot Insertion

Recall from Section 6.7 that knot insertion is a fundamental operation whereby control points are added to a B-Spline control polygon without changing the curve. In Section 6.7, knot insertion is discussed in terms of knot vectors and polar form. We can perform knot insertion using knot intervals, as well. In fact, the construction algorithm is perhaps even easier to remember and to carry out than the polar form construction.

Using knot intervals, it is more natural to speak of interval splitting than of knot insertion. For example, Figure 6.15 shows a cubic periodic B-Spline in which the red knot interval is split into two knot intervals that sum to the original knot interval. The only question we need to address is, how can we compute the Cartesian coordinates of the white control points in Figure 6.15.b.

Figure 6.16 illustrates how to compute those control points. Split the white, red, and green edges of the control polygon into three segments each whose lengths are proportional to the neighboring knot intervals, as shown in Figure 6.16.a. Each of those three edges contains a segment labeled with a red "3." Split each of those segments into two pieces whose lengths are proportional to 1:2, as shown in Figure 6.16.b. This produces the three desired control points. Figure 6.17 shows the procedure for splitting the interval "3" into two equal intervals.

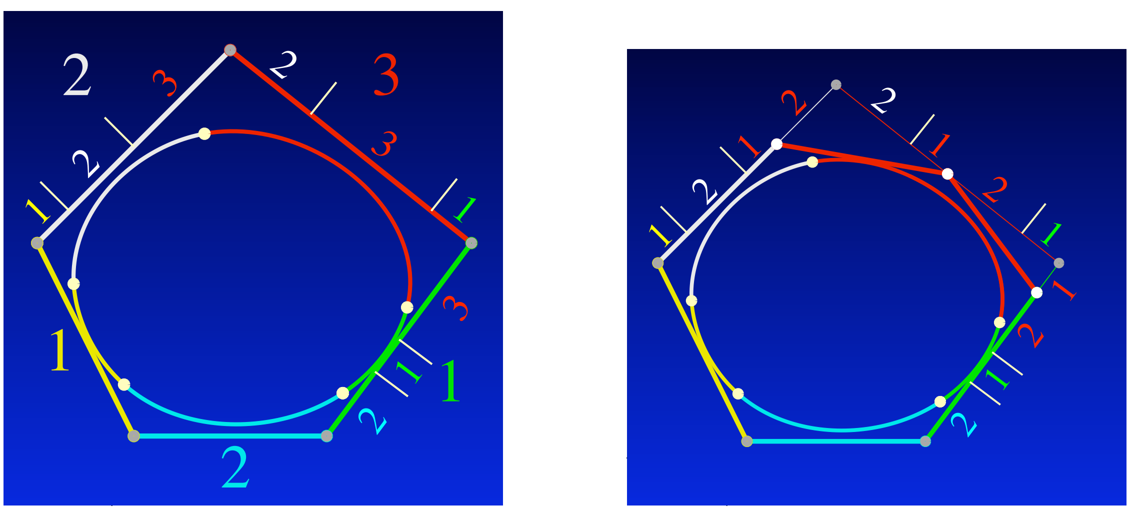
An important special case involves inserting a zero knot interval. In knot vector jargon, this is the same as inserting a double knot. In knot interval jargon, we can say that we are splitting a knot interval into two intervals, one of which is zero, and the other of which is the original knot interval. This special case is illustrated in Figure 6.18.a and Figure 6.18.b. If we repeat this operation three



(a) Periodic Cubic B-Spline Labeled with Knot Intervals.

(b) Splitting Interval 3 into Intervals 1 and 2.

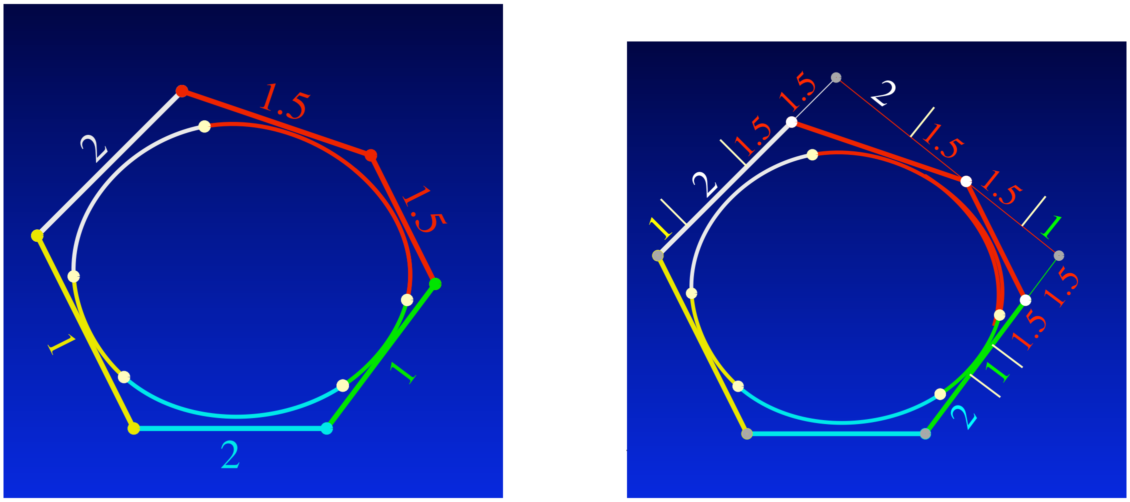
Figure 6.15: Knot Insertion using Knot Intervals.



(a) Dividing Edges of the Control Polygon into Three Segments Whose Lengths are Proportional to the Neighboring Knot Intervals.

(b) Splitting Intervals Labeled with a 3 into Intervals 1 and 2.

Figure 6.16: "Interval Splitting" using Knot Intervals.

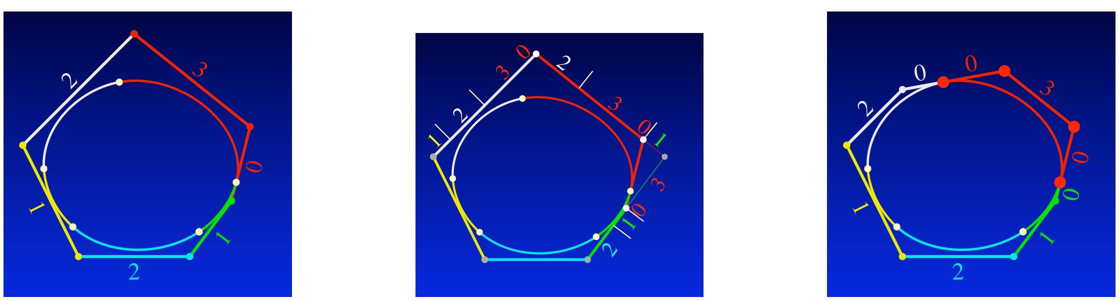


(a) Splitting the Interval Labeled with a 3 into Intervals 1.5 and 1.5.

(b) Construction.

Figure 6.17: “Interval Splitting” using Knot Intervals.

more times, arriving at the knot interval configuration shown in Figure 6.18.c, we uncover the Bézier control points for the red Bézier curve. The reason for this will be clear if you recall that a Bézier curve is a special case of B-Spline curve with knot vector $[a, a, a, b, b, b]$. Converting from knot vector to knot interval notation, we see that the two pairs of adjacent zero knot interval in Figure 6.18.c represent two sets of triple knots in the knot vector.



(a) Splitting the Interval Labeled with a 3 into Intervals 3 and 0.

(b) Construction.

(c) Bézier control points.

Figure 6.18: Introducing Zero Knot Intervals.

6.12.2 Interval Halving

Subdivision surfaces such as the Catmull-Clark scheme are based on the notion of inserting a knot half way between each existing pair of knots in a knot vector. These methods are typically restricted to uniform knot vectors. Knot intervals help to generalize this technique to non-uniform B-splines. Using knot intervals, we can think of this process as cutting in half each knot interval. For a quadratic non-uniform B-spline, the interval halving procedure is a generalization of Chaikin's algorithm, but the placement of the new control points becomes a function of the knot interval values. If each knot interval is cut in halve, the resulting control polygon has twice as many control points, and their coordinates $\mathbf{Q}_k$ are:

$$\begin{aligned}\mathbf{Q}_{2i} &= \frac{(d_i + 2d_{i+1})\mathbf{P}_i + d_i\mathbf{P}_{i+1}}{2(d_i + d_{i+1})} \\ \mathbf{Q}_{2i+1} &= \frac{d_{i+1}\mathbf{P}_i + (2d_i + d_{i+1})\mathbf{P}_{i+1}}{2(d_i + d_{i+1})}\end{aligned} \tag{6.3}$$

as illustrated in Figure 6.19.

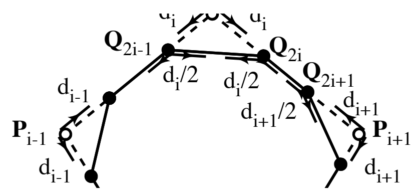


Figure 6.19: Interval halving for a non-uniform quadratic B-spline curve.

For non-uniform cubic periodic B-spline curves, interval halving produces a new control point corresponding to each edge, and a new control point corresponding to each original control point. The equations for the new control points $\mathbf{Q}_k$ generated by interval halving are:

$$\mathbf{Q}_{2i+1} = \frac{(d_i + 2d_{i+1})\mathbf{P}_i + (d_i + 2d_{i-1})\mathbf{P}_{i+1}}{2(d_{i-1} + d_i + d_{i+1})} \tag{6.4}$$

$$\mathbf{Q}_{2i} = \frac{d_i\mathbf{Q}_{2i-1} + (d_{i-1} + d_i)\mathbf{P}_i + d_{i-1}\mathbf{Q}_{2i+1}}{2(d_{i-1} + d_i)} \tag{6.5}$$

as shown in Figure 6.20.

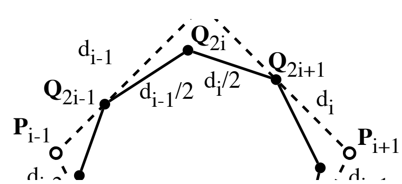


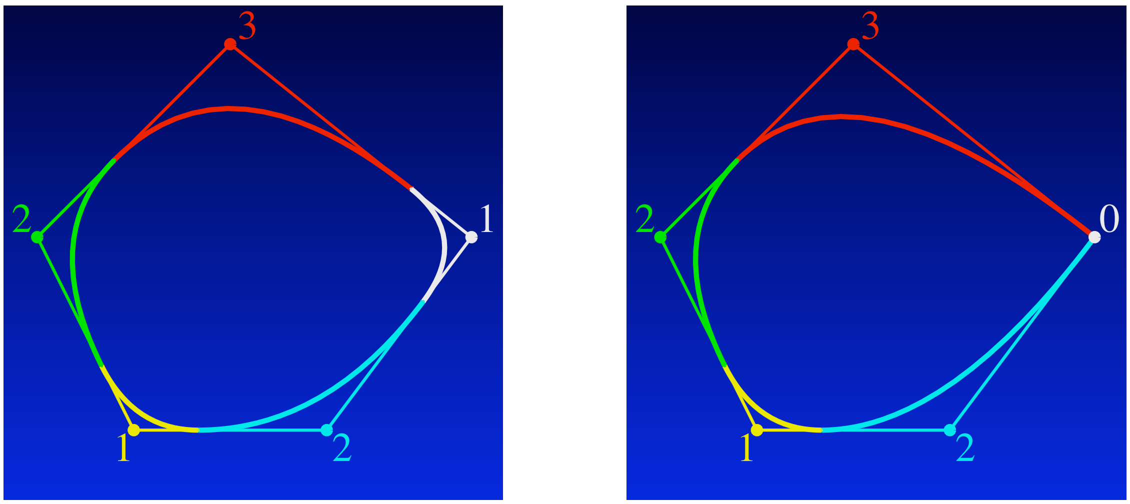
Figure 6.20: Interval halving for a non-uniform cubic B-spline curve.

Note that each new knot interval is half as large as its parent.

6.12.3 Degree-Two B-Splines using Knot Intervals

For any odd-degree B-Spline curve, the knot intervals are associated with the edges of the control polygon. For even-degree, the knot intervals are associated with the control points. Figure 6.21 shows two quadratic B-Splines with different knot intervals.

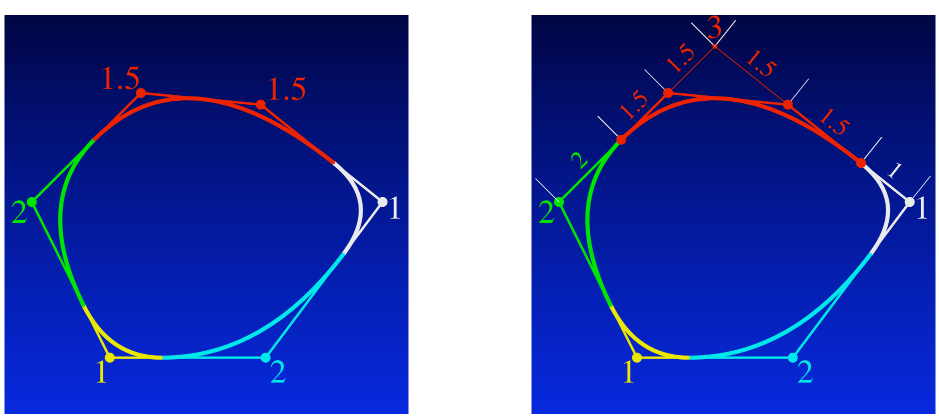
Figure 6.22 illustrates how to perform interval splitting for a quadratic B-Spline. Figure 6.23 illustrates how to split off a zero-knot interval on a quadratic B-Spline. To find the Bézier control points in a quadratic B-Spline, it suffices to insert a single zero knot interval on each side of a non-zero knot interval.



(a) Example 1.

(b) Example with Zero Knot Interval.

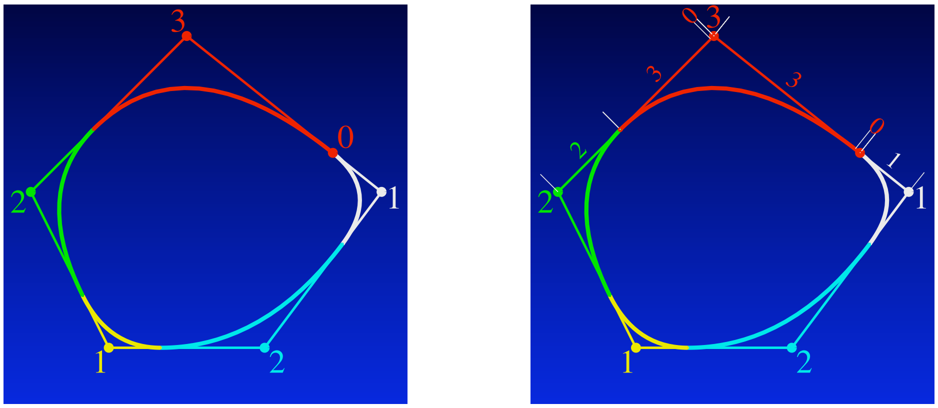
Figure 6.21: Quadratic B-Spline Curves.



(a) Splitting the Knot Interval 3 into two equal knot intervals.

(b) Construction for the Interval Splitting.

Figure 6.22: Interval Splitting of a Quadratic B-Spline Curve.



(a) Splitting the Knot Interval 3 into two equal knot intervals.

(b) Construction for the Interval Splitting.

Figure 6.23: Interval Splitting of a Quadratic B-Spline Curve.

6.12.4 Hodographs

Section 2.7 discusses the hodograph of a polynomial Bézier curve. The hodograph of a polynomial B-Spline curve can be constructed in a similar manner.

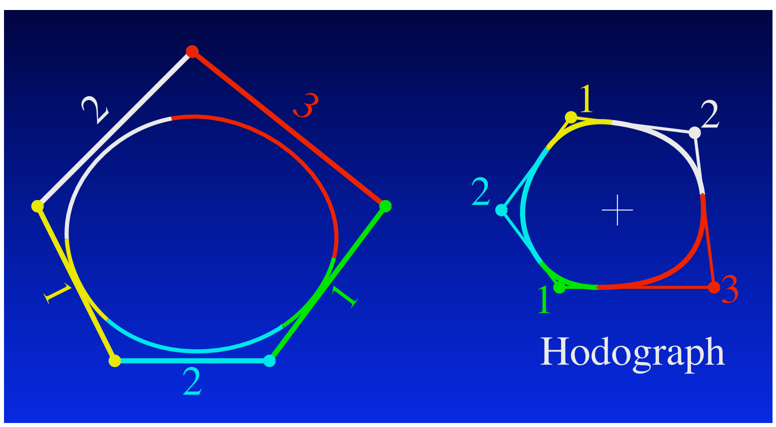
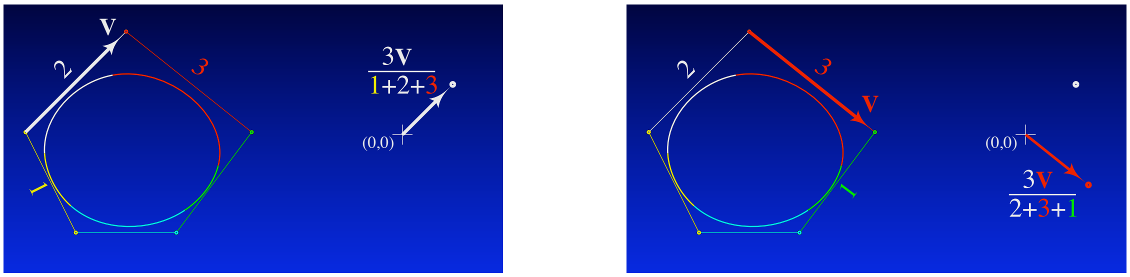


Figure 6.24: Hodograph of a Degree 3 Polynomial B-Spline Curve.

Figure 6.24 shows a cubic B-Spline curve and its hodograph, represented as a degree-two B-Spline curve. Note the relationship between the knot vectors in the original curve and in its hodograph. The control points for the hodograph are found in a manner similar to the control points for a Bézier hodograph.

The exact formula for the control points, as illustrated in Figure 6.25, is as follows. The hodograph of a degree n B-spline $\mathbf{P}(t)$ with knot intervals d_i and control points $\mathbf{P}_i$ is a B-spline of degree $n-1$ with the same knot intervals d_i and with control points $\mathbf{Q}_i$ where

$$\mathbf{Q}_i = c_i(\mathbf{P}_{i+1} - \mathbf{P}_i). \tag{6.6}$$



(a) (b)

Figure 6.25: Finding the Control Points of a B-Spline Hodograph.

The scale factor c_i is the inverse of the average value of n neighboring knot intervals. Specifically, if the curve is even-degree $n = 2m$, then

$$c_i = \frac{n}{d_{i-m+1} + \ldots + d_{i+m}}$$

and if the curve is odd degree $n = 2m + 1$

$$c_i = \frac{n}{d_{i-m} + \ldots + d_{i+m}}$$

6.12.5 Degree elevation

Ramshaw [Ram89a] presented an elegant insight into degree elevation using polar form. The symmetry property of polar labels demands that

$$f(a,b) = \frac{f(a) + f(b)}{2}; \qquad f(a,b,c) = \frac{f(a,b) + f(a,c) + f(b,c)}{3}; \tag{6.7}$$

$$f(a,b,c,d) = \frac{f(a,b,c) + f(a,b,d) + f(a,c,d) + f(b,c,d)}{4}; \quad \text{etc.} \tag{6.8}$$

The procedure of degree elevation on a periodic B-spline that is labeled using knot intervals results in two effects. First, an additional control point is introduced for each curve segment. Second, if the sequence of knot intervals is initially $d_1, d_2, d_3, \ldots$, the sequence of knot intervals on the degree elevated control polygon will be $d_1, 0, d_2, 0, d_3, 0, \ldots$. The zeroes must be added because degree elevation raises the degree of each curve segment without raising the continuity between curve segments,

Degree elevation of a degree one B-spline is simple: merely insert a new control point on the midpoint of each edge of the control polygon. The knot intervals are as shown in Figures 6.26.a and b.

Degree elevation for a degree two B-spline is illustrated in Figures 6.26.c and d. The new control points are:

$$\mathbf{P}_{i,j} = \frac{(2d_i + 3d_j)\mathbf{P}_i + d_i\mathbf{P}_j}{3d_i + 3d_j}. \tag{6.9}$$

Figure 6.27 illustrates degree elevation from degree three to four. The equations for the new control points are:

$$\mathbf{P}_{i,i+1} = \frac{(d_i + 2d_{i+1})\mathbf{P}_i + (2d_{i-1} + d_i)\mathbf{P}_{i+1}}{2(d_{i-1} + d_i + d_{i+1})}$$

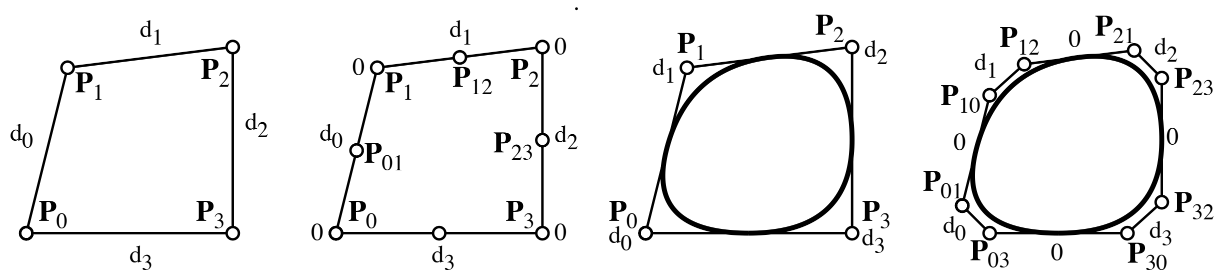


Figure 6.26: Degree elevating a degree one and degree two B-spline.

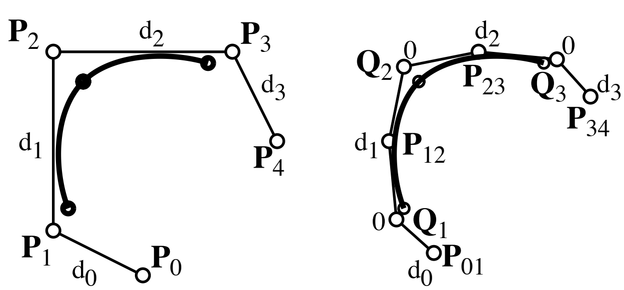


Figure 6.27: Degree elevating a degree three B-spline.

$$\mathbf{Q}_i = \frac{d_i}{4(d_{i-2}+d_{i-1}+d_i)}\mathbf{P}_{i-1}+\left(\frac{d_{i-2}+d_{i-1}}{4(d_{i-2}+d_{i-1}+d_i)}+\frac{d_i+d_{i+1}}{4(d_{i-1}+d_i+d_{i+1})}+\frac{1}{2}\right)\mathbf{P}_i+\frac{d_{i-1}}{4(d_{i-1}+d_i+d_{i+1})}\mathbf{P}_{i+1}$$

6.13 B-spline Basis Functions

A degree d B-spline curve with n control points can be expressed

$$\mathbf{P}(t) = \sum_{i=1}^{n} \mathbf{P_i} B_i^d(t).$$

The knot vector for this curve contains $n+d-1$ knots, denoted $[t_{1-d/2},\ldots,t_{n+(d-1)/2}]$.

Each B-spline basis function $B_i^d(t)$ can be expressed as an explicit B-spline curve whose x coordinates are the Greville abscissae (Section 6.9) and whose y coordinates are all zero except $y_i = 1$. Therefore, $B_i^d(t) = 0$ for $t \le t_{i-(d+1)/2}$ and $t \ge t_{i+(d+1)/2}$. Each basis function is a piecewise polynomial that has $d+1$ non-zero segments. For example, the cubic B-spline in Figure 6.28, $n = 7$ and the knot vector is $[0,0,0,2,3,4,5,5,5]$. The basis function $B_3^3(t)$ is shown in Figure **??**.

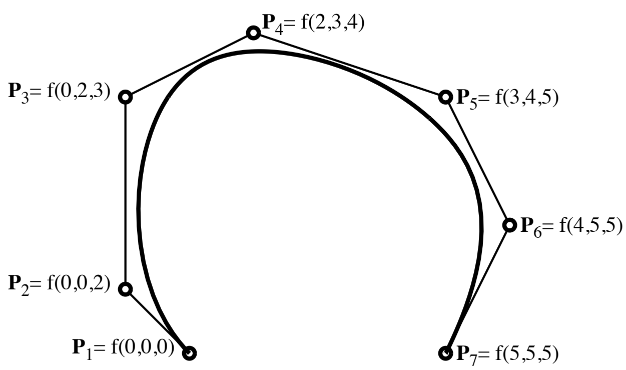


Figure 6.28: Cubic B-Spline Curve.

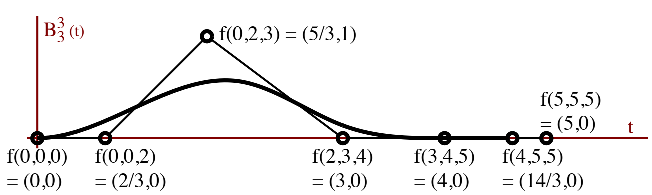


Figure 6.29: Basis function $B_3^3(t)$.

The basis function in Figure 6.29 is just like any other B-Spline curve. To evaluate $B_3^3(t)$, we compute polar value $f(t,t,t)$ using knot insertion. The y coordinate of $f(t,t,t)$ is $B_3^3(t)$, and the x-coordinate of $f(t,t,t)$ is t.

6.13.1 B-Spline Basis-Functions using Knot Intervals

Basis functions for B-Splines defined using knot intervals can likewise be represented as explicit B-Spline curves (Section 6.9). For the curve in Figure 6.30, the basis function for $\mathbf{P}_i$ is illustrated in Figure 6.31.

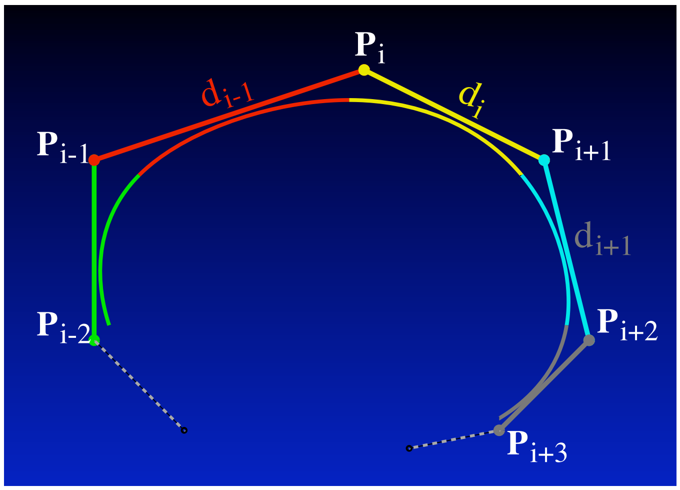


Figure 6.30: Sample Cubic B-Spline Curve.

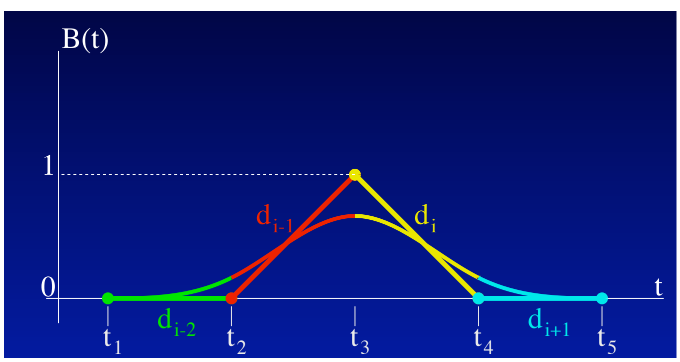


Figure 6.31: B-Spline Basis Function for Control Point $\mathbf{P}_i$ in Figure 6.30.

$B(t)$ is the y-coordinates the curve in Figure 6.31. Those y-coordinates are easily evaluated in the same manner as you would evaluate a point on any B-Spline curve. Note that the y-coordinates of the control points of this explicit B-Spline curve are all zero, except for the control point whose basis function we desire.

If we now evaluate a point on this explicit B-Spline curve using the knot-interval version of the de Boor algorithm, we arrive at the following equations for the B-Spline basis function in Figure 6.31. For $t \in [t_1, t_2]$:

$$B(t) = \frac{(t - t_1)^3}{(t_4 - t_1)(t_3 - t_1)_{(}t_2 - t_1)} \tag{6.10}$$

For $t \in [t_2, t_3]$:

$$B(t) = \frac{(t-t_2)^2(t_5-t)}{(t_5-t_2)(t_4-t_2)(t_3-t_2)} + \frac{(t-t_2)(t-t_1)(t_4-t)}{(t_4-t_1)(t_4-t_2)(t_3-t_2)} + \frac{(t_3-t)(t-t_1)^2}{(t_4-t_1)(t_3-t_1)(t_3-t_2)} \quad (6.11)$$

For $t \in [t_3, t_4]$:

$$B(t) = \frac{(t_4-t)^2(t-t_1)}{(t_4-t_1)(t_4-t_2)(t_4-t_3)} + \frac{(t_4-t)(t_5-t)(t-t_2)}{(t_5-t_2)(t_4-t_2)(t_4-t_3)} + \frac{(t-t_3)(t_5-t)^2}{(t_5-t_2)(t_5-t_3)(t_4-t_3)} \quad (6.12)$$

For $t \in [t_4, t_5]$:

$$B(t) = \frac{(t_5-t)^3}{(t_5-t_2)(t_5-t_3)(t_5-t_4)} \quad (6.13)$$

These four polynomials are C^2, with

$$B'(t_1) = B''(t_1) = 0, \quad B'(t_2) = \frac{3(t_2-t_1)}{(t_3-t_1)(t_4-t_1)}, \quad B''(t_2) = \frac{6}{(t_3-t_1)(t_4-t_1)},$$

$$B'(t_3) = \frac{3(t_1(t_2-t_3) - t_2t_3 + t_3t_4 + t_3t_5 - t_4t_5)}{(t_1-t_4)(t_2-t_4)(t_2-t_5)}, \quad B''(t_3) = \frac{6(t_1+t_2-t_4-t_5)}{(t_1-t_4)(t_4-t_2)(t_2-t_5)}$$

$$B'(t_4) = -\frac{3(t_5-t_4)}{(t_5-t_2)(t_5-t_3)}, \quad B''(t_4) = \frac{6}{(t_5-t_2)(t_5-t_3)}$$

6.13.2 Refinement of B-Spline Basis Functions

Denote by

$$N[s_0, s_1, \ldots, s_{n+1}](s)$$

a B-Spline basis function of degree n over a knot vector $\{s_0, s_1, \ldots, s_{n+1}\}$ and with support $[s_0, s_{n+1}]$. Upon inserting a knot k for which $s_i \le k \le s_{i+i}$, the basis function is split into two scaled basis functions:

$$N[s_0, s_1, \ldots, s_{n+1}](s) = c_i N[s_0, \ldots, s_i, k, s_{i+1}, \ldots, s_n](s) + d_i N[s_1, \ldots, s_i, k, s_{i+1}, \ldots, s_{n+1}](s)$$

where

$$c_i = \begin{cases} \frac{k-s_0}{s_{n+1}-s_0} & k < s_n \\ 1 & k \ge s_n \end{cases}$$

$$d_i = \begin{cases} \frac{s_{n+1}-k}{s_{n+1}-s_1} & k > s_1 \\ 1 & k \le s_1 \end{cases}$$

6.13.3 Recurrence Relation

B-spline basis functions can also be defined using the following recurrence relations.

A degree zero B-spline curve is defined over the interval $[t_i, t_{i+1}]$ using one control point, $\mathbf{P}_0$. Its basis function, which we will denote $B_i^0(t)$ is the step function

$$b_i^0(t) = \begin{cases} 1 & \text{if } t_i \le t \le t_{i+1} \\ 0 & \text{otherwise.} \end{cases}$$

The curve $B_i^0(t)\mathbf{P}_0$ consists of the discrete point $\mathbf{P}_0$.

Blending functions for higher degree B-splines are defined using the recurrence relation:

$$B_i^k(t) = \omega_i^k(t)B_i^{k-1}(t) + (1 - \omega_{i+1}^k(t))B_{i+1}^{k-1}(t) \tag{6.14}$$

where

$$\omega_i^k(t) = \begin{cases} \frac{t-t_i}{t_{i+k-1}-t_i} & \text{if } t_i \neq t_{i+k-1} \\ 0 & \text{otherwise.} \end{cases}$$

A degree one B-spline curve is defined over the interval $[t_i, t_{i+1}]$ using two control points, which we will denote as polar values $\mathbf{P}(t_i)$ and $\mathbf{P}(t_{i+1})$. The curve is simply the line segment joining the two control points:

$$\mathbf{P}(t) = \frac{t_{i+1} - t}{t_{i+1} - t_i}\mathbf{P}(t_i) + \frac{t - t_i}{t_{i+1} - t_i}\mathbf{P}(t_{i+1}).$$

A single degree n B-spline curve segment defined over the interval $[t_i, t_{i+1}]$ with knot vector $\{\ldots, t_{i-1}, t_i, t_{i+1}, t_{i+2}, \ldots\}$ has $n + 1$ control points written as polar values

$$\mathbf{P}(t_{i+1-n}, \ldots, t_i), \ldots, \mathbf{P}(t_{i+1}, \ldots, t_{i+n})$$

and blending functions $B_i^n(t)$ which are obtained from equation 6.14. The equation for the curve is:

$$\mathbf{P}(t) = \sum_{j=i}^{n+i} B_{j+1-n}^n(t)\mathbf{P}(t_{j+1-n}, \ldots, t_j) \tag{6.15}$$

In approximation theory, the basis functions themselves are referred to as B-splines.