Chapter 5

B-SPLINE CURVES

Most shapes are simply 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" can also be used as a verb, as in "Spline together some cubic Bézier curves."

The word "spline" comes from the ship building industry, where it originally meant 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 5.1. We

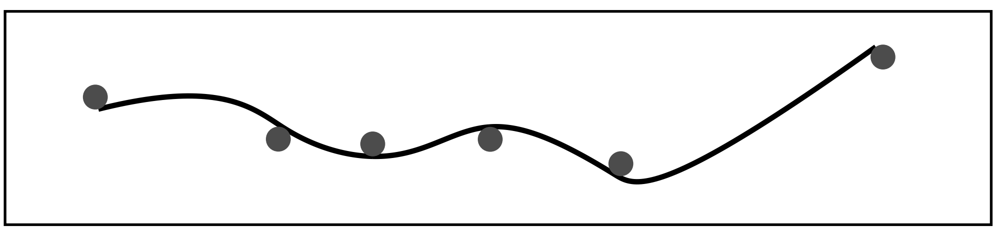

Figure 5.1: Spline and ducks.

know from basic structures theory 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.

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 the train would experience severe jolts. Car bodies are G^2 smooth, or else the reflection of straight lines would bend sharply.

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. In practical terms, 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 basic operation you need to understand about B-splines is how to extract the contituent Bézier curves. That understanding will provide you with a good working knowledge of B-spline curves.

5.1 Polar Form

Dr. Lyle Ramshaw of DEC Systems Research Center has developed a way of understanding B-splines based on what he calls *polar forms* [38, 39, 40]. This contrasts with the approach taken by conventional textbooks which begin by studying the B-spline basis functions. Experience has shown that Ramshaw's method allows students to attain a working knowledge of B-spline curves much faster, and to retain that "closed-book" knowledge far longer, than with traditional methods.

Ramshaw refers to this labeling scheme as *polar form.* In polar form, control points 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 degree n Bézier curves over the parameter interval $[a, b]$, the control points are relabeled $\mathbf{P}_i = \mathbf{P}(u_1, u_2, \ldots u_n)$ where $u_j = a$ if $j \leq n - i$ and otherwise $u_j = b$. For a degree two curve over the interval $[a, b]$,

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

For a degree three Bézier curve,

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

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

and so forth. Figure 5.2 shows two cubic Bézier curves labeled using polar values. The first curve is

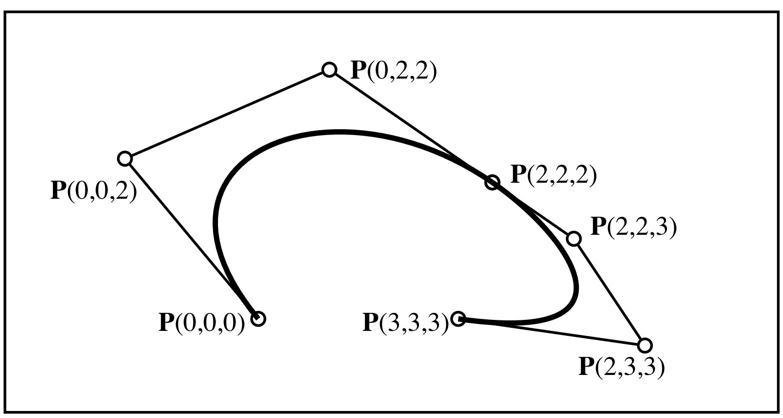

Figure 5.2: Bézier curves labeled using polar form.

defined over the parameter interval $[0, 2]$ and the second curve is defined over the parameter interval $[2, 3]$. Note that $\mathbf{P}(t, t, \ldots, t)$ is the point on a Bézier curve corresponding to parameter value t.

2. For a degree n B-spline with a *knot vector* (explained later) of

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

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 5.3.

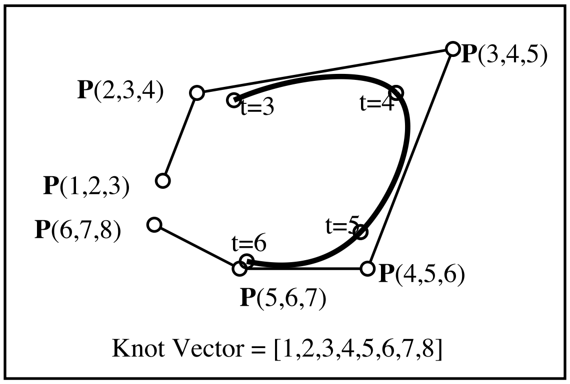

Figure 5.3: B-spline curve labeled using polar form.

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 5.4.

5.1.1 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 defined over the parameter interval $[0,1]$, we wish to split it into Bézier curves over the intervals $[0,t]$ and $[t,1]$. The control points of the original curve are labeled

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

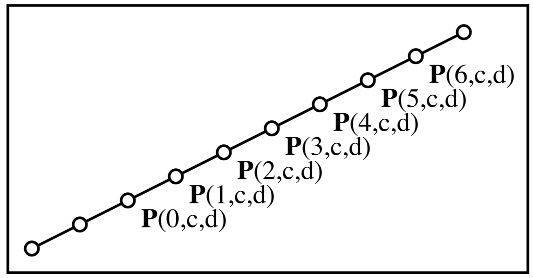

Figure 5.4: Affine map property of polar values.

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 5.5, 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.

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

5.2 Symmetric polynomials

The polar form of a Bézier curve is based on the notion of symmetric polynomials. The idea is to represent 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

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

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

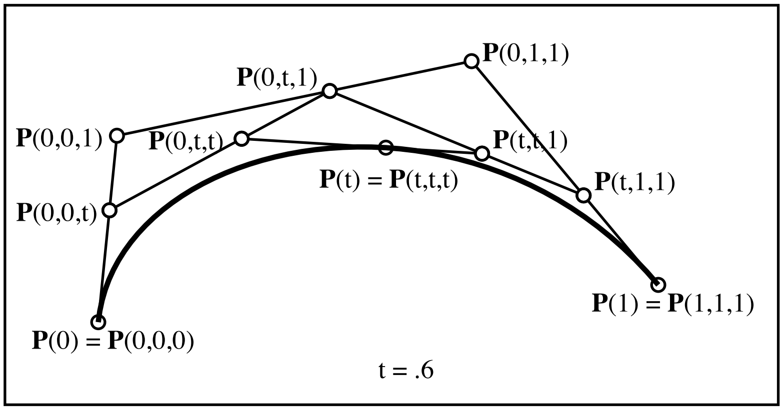

Figure 5.5: Subdividing a cubic Bézier curve.

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: $p[t_1, t_2, t_3] = 1 + 3\frac{t_1+t_2+t_3}{3} + 6\frac{t_1 t_2 + t_1 t_3 + t_2 t_3}{3} + t_1 t_2 t_3$.

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,

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

5.3 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 Bézier curves 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 5.7.

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 5.3 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 5.5. 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.

5.4 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 [8].

Consider the B-spline in Figure 5.3 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 5.6.

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 5.7. Finally, $\mathbf{P}(3, 3, 3)$ is seen to be half way between $\mathbf{P}(3, 2, 3)$ and $\mathbf{P}(3, 3, 4)$,

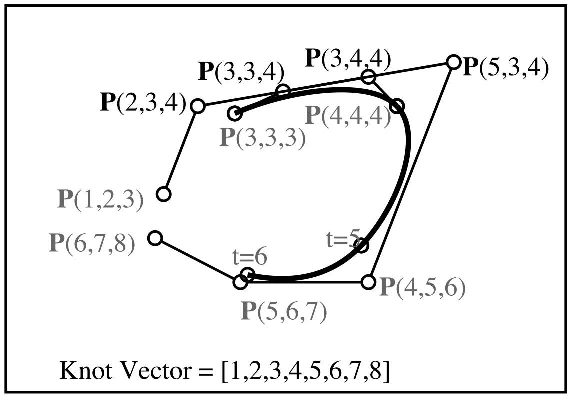

Figure 5.6: First step in Böhm algorithm.

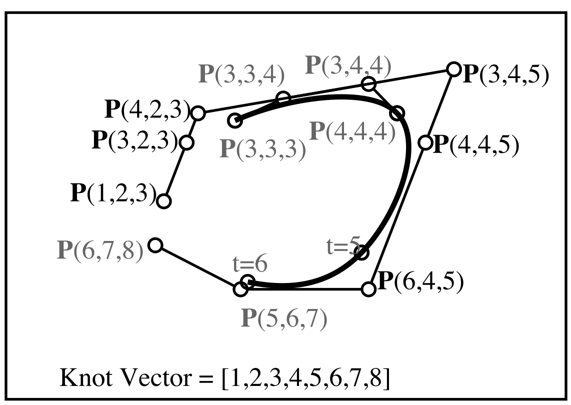

Figure 5.7: Second step in Böhm algorithm.

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.

5.5 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 5.8 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

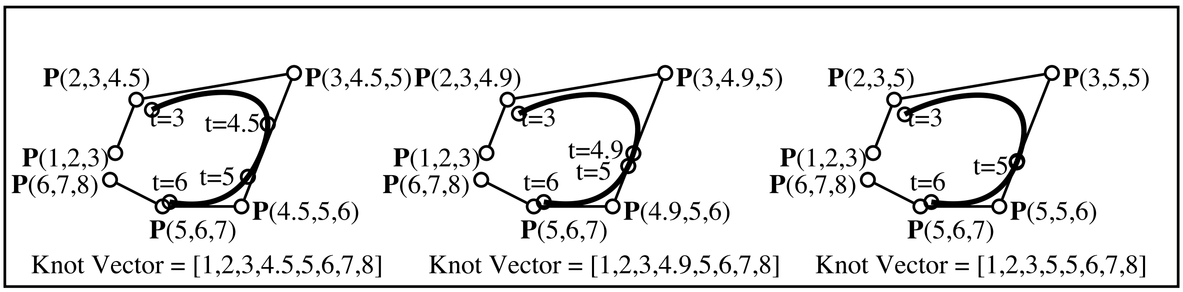

Figure 5.8: Double knot.

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

5.6 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 5.9.

5.7 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 5.10.

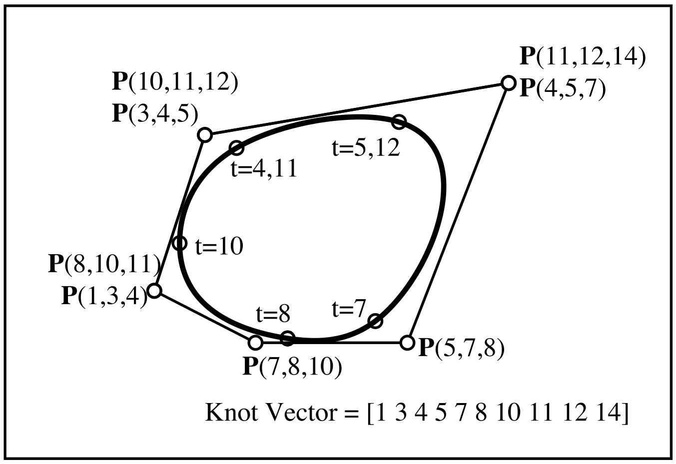

Figure 5.9: Periodic B-spline.

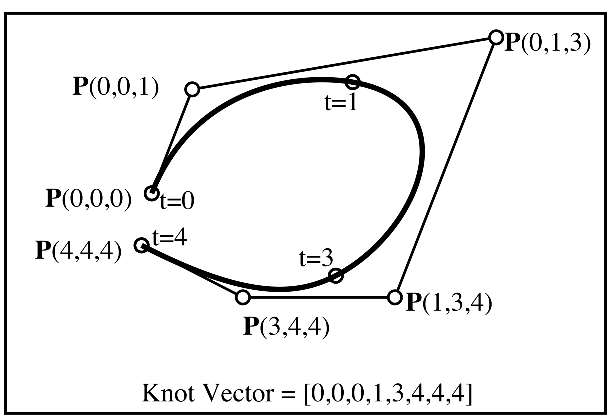

Figure 5.10: Bézier end condition.

	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)

Figure 5.11: Before and after.

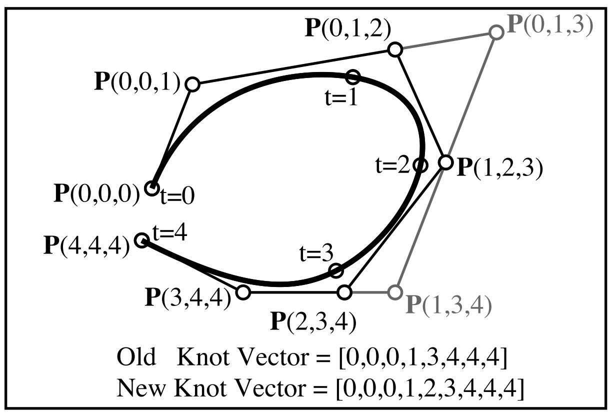

Figure 5.12: Knot insertion.

5.8 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 5.10. As shown in Figure 5.11, 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 5.12 shows the new set of control points, which are easily obtained using the affine and symmetry properties of polar values.

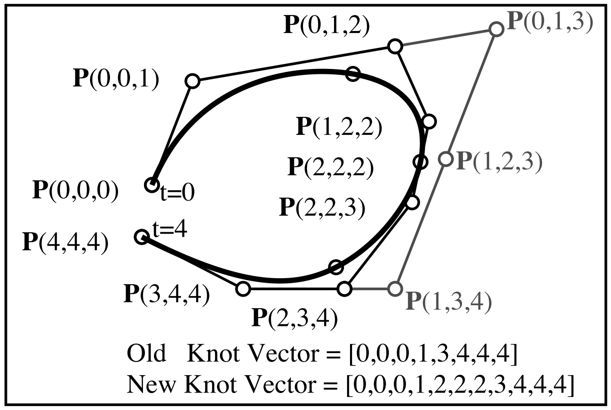

Figure 5.13: De Boor algorithm.

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

5.9 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 5.13.

5.10 Explicit B-splines

Section 2.12 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

$$x_i = \frac{1}{n}(t_i + t_{i+1} + \ldots + t_{i+n-1}); \quad i = 0 \ldots m - n. \tag{5.1}$$

5.11 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

$$\mathbf{H}_i = n\frac{(\mathbf{P}_{i+1} - \mathbf{P}_i)}{t_{i+n} - t_i} \tag{5.2}$$

where n is the degree.

5.12 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 [44] 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.

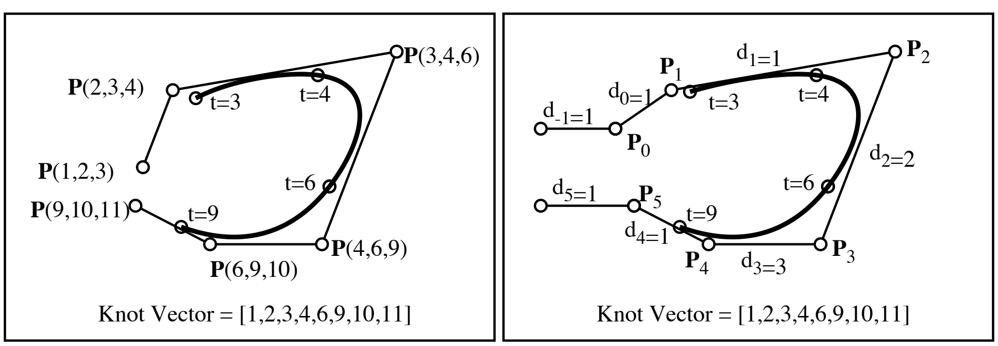

Figure 5.14: Sample cubic B-spline

Figure 5.14 shows a cubic B-spline curve. The control points in Figure 5.14.a are labeled with polar values, and Figure 5.14.b shows the control polygon edges labeled with knot intervals. End-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 5.15 shows two cubic periodic B-splines labelled with knot intervals. In this example, note

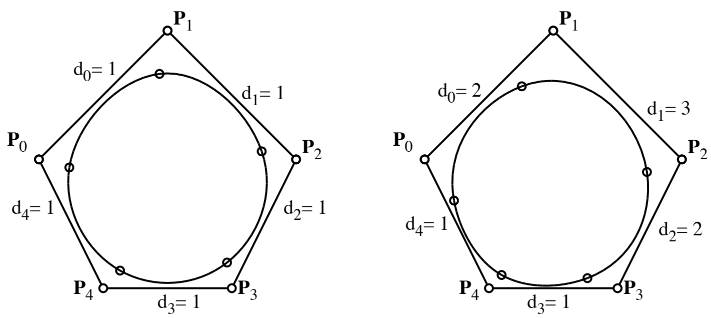

Figure 5.15: Periodic B-splines labelled with knot intervals

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

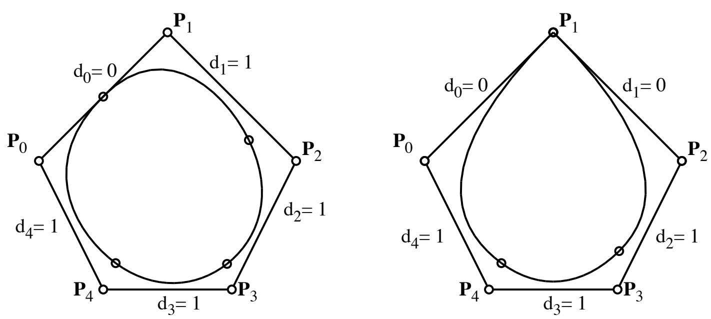

Figure 5.16: Periodic B-splines with double and triple knots.

Figure 5.16 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 [37] 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 5.17, we choose the knot origin to coincide with control points $\mathbf{P}_0$. Then the polar values are as shown in Figure 5.17.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 [23].

5.12.1 Knot Insertion

Knot intervals provide an easy-to-remember method for performing knot insertion. For a cubic B-spline, begin by splitting each edge $\mathbf{P}_i$–$\mathbf{P}_{i+1}$ of the control polygon into three segments whose lengths are proportional to d_{i-1}, d_i, and d_{i+1} as shown in Figure 5.18a. (for periodic B-splines, the subscript values are all modulo the number of edges in the control polygon). For a B-spline of even degree $2n$, each edge is split into $2n$ segments whose lengths are proportional to $d_{i-n}, \ldots, d_{i+n-1}$

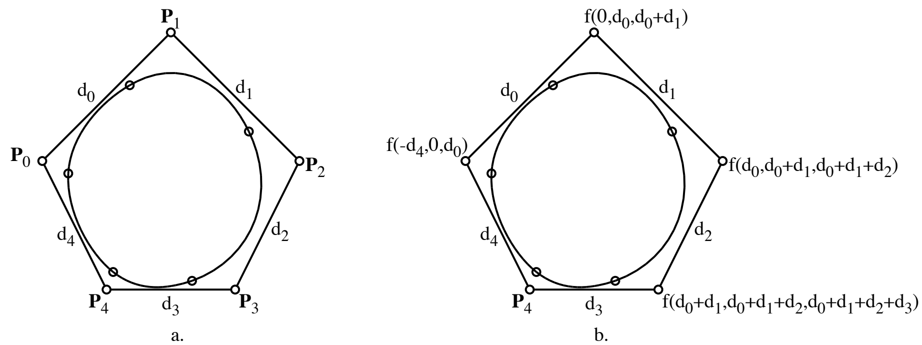

Figure 5.17: Inferring polar labels from knot intervals.

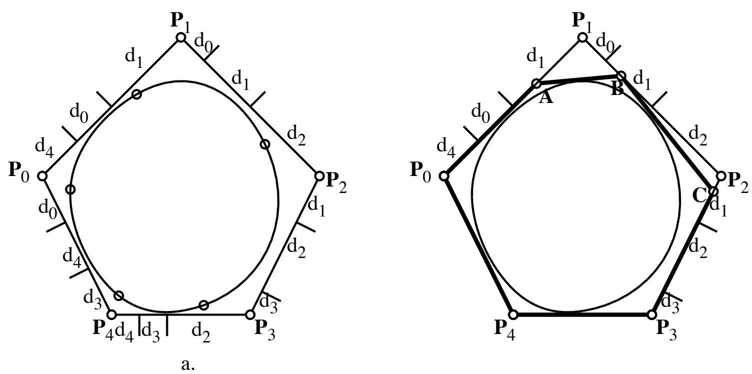

Figure 5.18: Knot insertion on a cubic B-spline.

and for a B-spline of odd degree $2n+1$, each edge is split into $2n+1$ segments whose lengths are proportional to $d_{i-n},\dots,d_{i+n}$.

Knot insertion in terms of knot intervals can be thought of as splitting a knot interval at some fraction $t \in [0,1]$. For example, suppose we wish to split knot interval d_1 in Figure 5.18a at $t=\frac{1}{3}$. We simply find each occurrence of d_1 on the control polygon edges, insert a control point $\frac{1}{3}$ of the way along each segment labelled d_1, and replace the control points $\mathbf{P}_1$ and $\mathbf{P}_2$ with $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ as shown in Figure 5.18.b.

Knot removal is the inverse of knot insertion. Thus, given the control polygon in Figure 5.18.b, knot removal would produce the control polygon in Figure 5.18.a. Knot removal is possible only when two adjacent curve segments are C^r with $r > n-m$ where n is the degree and m is the multiplicity of the knot; thus it is not generally possible to perform knot removal. We will say that a control polygon which cannot undergo knot removal is in *minimal form*, and the minimal form of a B-spline control polygon results when all knots have been removed that can be.

5.12.2 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:

$$\begin{aligned} \mathbf{Q}_{2i} &= \frac{(d_i + 2d_{i+1})\mathbf{P}_i + d_i\mathbf{P}_{i+1}}{2(d_i + d_{i+1})} \\ \mathbf{Q}_{2i+1} &= \frac{d_{i+1}\mathbf{P}_i + (2d_i + d_{i+1})\mathbf{P}_{i+1}}{2(d_i + d_{i+1})} \end{aligned} \tag{5.3}$$

as illustrated in Figure 5.19.

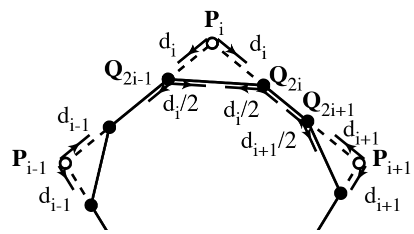

Figure 5.19: Interval halving for a non-uniform quadratic B-spline curve.

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:

$$\mathbf{Q}_{2i+1} = \frac{(d_i + 2d_{i+1})\mathbf{P}_i + (d_i + 2d_{i-1})\mathbf{P}_{i+1}}{2(d_{i-1} + d_i + d_{i+1})} \tag{5.4}$$

$$\mathbf{Q}_{2i} = \frac{d_i\mathbf{Q}_{2i-1} + (d_{i-1} + d_i)\mathbf{P}_i + d_{i-1}\mathbf{Q}_{2i+1}}{2(d_{i-1} + d_i)} \tag{5.5}$$

as shown in Figure 5.20.

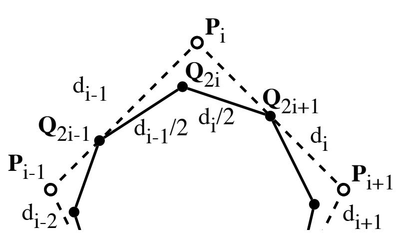

Figure 5.20: Interval halving for a non-uniform cubic B-spline curve.

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

5.12.3 Hodographs

The derivative $\mathbf{P}'(t)$ of a B-spline is called its hodograph. 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

$$\mathbf{Q}_i = c_i(\mathbf{P}_{i+1} - \mathbf{P}_i). \tag{5.6}$$

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 $n = 2m + 1$

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

5.12.4 Degree elevation

Ramshaw [37] presented an elegant insight into degree elevation using polar form. The symmetry property of polar labels demands that

$$f(a,b) = \frac{f(a) + f(b)}{2}; \qquad f(a,b,c) = \frac{f(a,b) + f(a,c) + f(b,c)}{3}; \tag{5.7}$$

$$f(a,b,c,d) = \frac{f(a,b,c) + f(a,b,d) + f(a,c,d) + f(b,c,d)}{4}; \quad \text{etc.} \tag{5.8}$$

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 5.21.a and b.

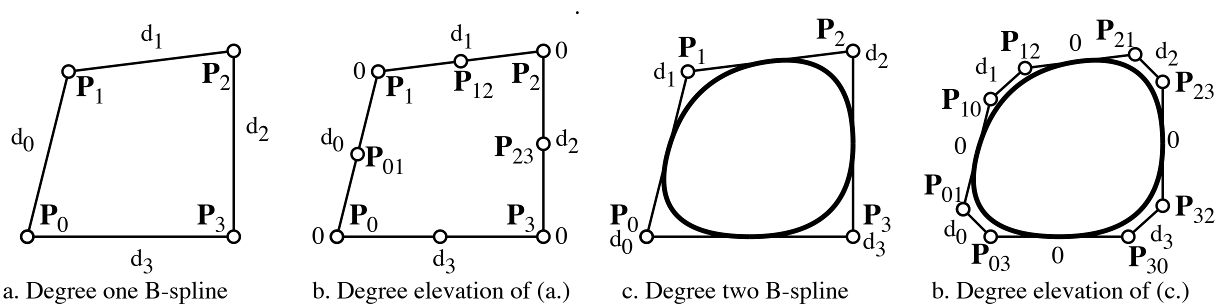

Figure 5.21: Degree elevating a degree one and degree two B-spline.

Degree elevation for a degree two B-spline is illustrated in Figures 5.21.c and d. The new control points are:

$$\mathbf{P}_{i,j} = \frac{(2d_i + 3d_j)\mathbf{P}_i + d_i\mathbf{P}_j}{3d_i + 3d_j}. \tag{5.9}$$

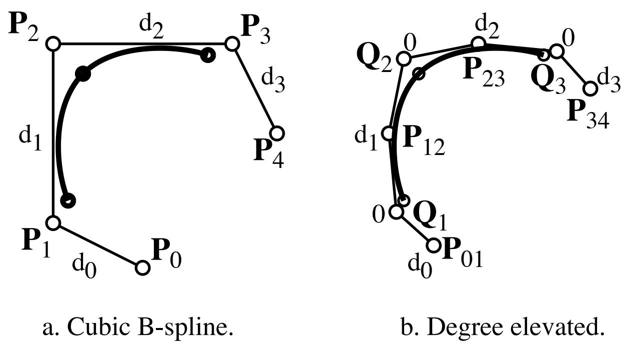

Figure 5.22: Degree elevating a degree three B-spline.

Figure 5.22 illustrates degree elevation from degree three to four. The equations for the new control points are:

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

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

5.13 Split-Interval Notation

A useful variation of knot interval notation is split-knot-interval notation in which a knot interval is split into a sequence of two or more non-negative knot intervals which sum to the original interval. Figure 5.23.a shows a periodic cubic B-spline whose top edge has a knot interval of 3. In Figure 5.23.b, that knot interval is divided and the edge is labeled with two knot intervals 1,2. Figure 5.23.c shows an equivalent representation in which the knot interval has actually been split by performing knot insertion, as discussed in Section 5.12.1. We will refer to this as the *expanded form* of the split-interval notation

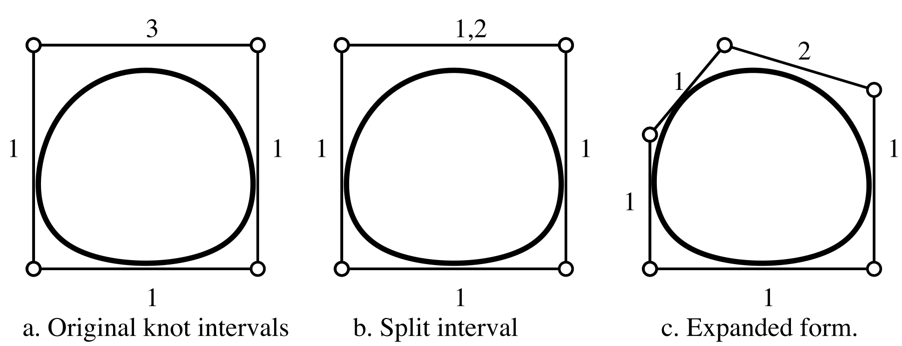

Figure 5.23: Cubic B-spline with two knot intervals on one edge.

A degree n B-spline is comprised of curve segments that meet with continuity C^{n-r} where r is the multiplicity of the knot. Curve segments mapped to in split-interval notation are C^{∞}.

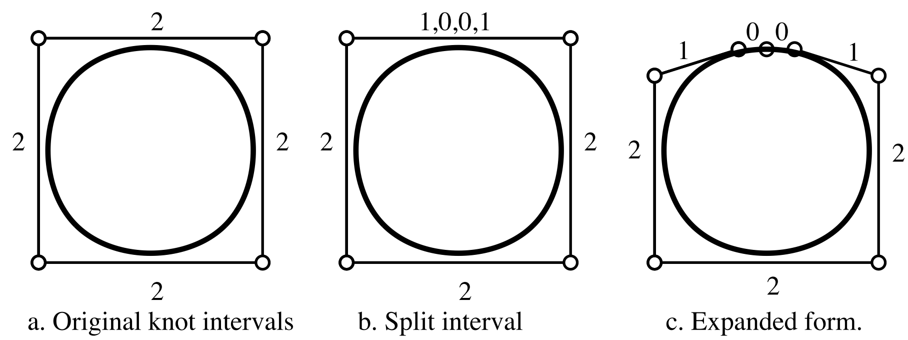

Figure 5.24: Four knot intervals on one edge of a cubic B-spline, expressing the de Boor algorithm.

A second example is presented in Figure 5.24. Here, the knot interval on the top edge is split into four intervals in Figure 5.24.b. The expanded form contains a triple knot. This is actually a way of denoting the de Boor algorithm using split interval notation. Using the terminology in Section 5.12.1, the control polygon in Figure 5.24.a is the minimal form of the control polygon in Figure 5.24.c.

5.14 Cubic B-Splines

The most common B-splines are probably those of degree three. This section presents closed-form expressions for the useful operations of conversion between Bézier and B-spline curves and knot insertion.

5.14.1 Splitting a B-spline into Bézier Curves

Cubic B-splines are defined by specifying $n > 3$ *control points* $\mathbf{P}_1, \ldots, \mathbf{P}_n$, and a *knot vector* $[k_{-2}, \ldots, k_{n+1}]$, which is a sequence of non-decreasing real numbers.

Every cubic B-spline with n control points can be decomposed into $n-3$ cubic Bézier curves. Conventionally, Bézier curves use the parameter range $0 \le t \le 1$. For the i^{th} Bézier curve in a B-spline, the parameter range is $k_i \le t \le k_{i+1}$. As far as the *appearance* of a Bézier curve is concerned, for four given control points, a Bézier curve defined over the range $k_i \le t \le k_{i+1}$ looks identical to one defined over the parameter interval $0 \le t \le 1$. So if one knows how to plot a cubic Bézier curve, all one needs to know to plot a B-spline is how to extract the control points of each of the $n-3$ Bézier curves comprising it.

Figure 5.25 shows the i^{th} Bézier curve (whose control points are labelled $\mathbf{Q}_0$, $\mathbf{Q}_1$, $\mathbf{Q}_2$, $\mathbf{Q}_3$) of a sample B-spline (whose control points are labelled $\mathbf{P}_{i-1} \ldots \mathbf{P}_{i+4}$). The Bézier control points are obtained using the formulas:

$$\mathbf{Q}_1 = \frac{(k_{i+2} - k_i)\mathbf{P}_{i+1} + (k_i - k_{i-1})\mathbf{P}_{i+2}}{k_{i+2} - k_{i-1}} \tag{5.10}$$

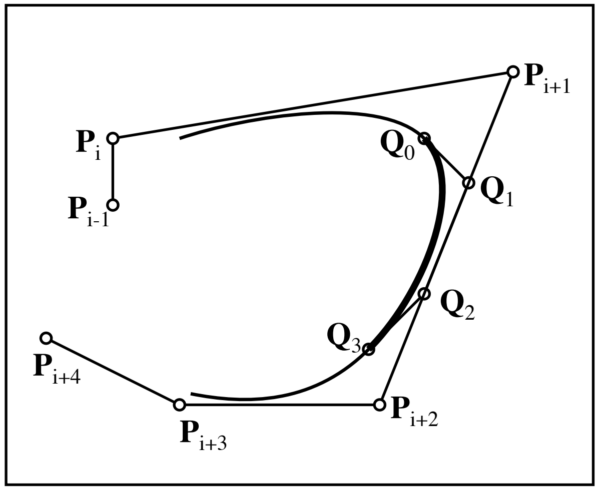

Figure 5.25: Extracting a Bézier curve from a B-spline.

$$\mathbf{Q}_0 = \frac{k_{i+1} - k_i}{k_{i+1} - k_{i-1}} \frac{(k_{i+1} - k_i)\mathbf{P}_i + (k_i - k_{i-2})\mathbf{P}_{i+1}}{k_{i+1} - k_{i-2}} + \frac{k_i - k_{i-1}}{k_{i+1} - k_{i-1}}\mathbf{Q}_1 \tag{5.11}$$

$$\mathbf{Q}_2 = \frac{(k_{i+2} - k_{i+1})\mathbf{P}_{i+1} + (k_{i+1} - k_{i-1})\mathbf{P}_{i+2}}{k_{i+2} - k_{i-1}} \tag{5.12}$$

$$\mathbf{Q}_3 = \frac{k_{i+1} - k_i}{k_{i+2} - k_i} \frac{(k_{i+3} - k_{i+1})\mathbf{P}_{i+2} + (k_{i+1} - k_i)\mathbf{P}_{i+3}}{k_{i+3} - k_i} + \frac{k_{i+2} - k_{i+1}}{k_{i+2} - k_i}\mathbf{Q}_2 \tag{5.13}$$

Notice from equations 5.10–5.13 that if $k_i = k_{i+1}$, Bézier curve i collapses to a single point:

$$\mathbf{Q}_0 = \mathbf{Q}_1 = \mathbf{Q}_2 = \mathbf{Q}_3 = \frac{(k_{i+2} - k_i)\mathbf{P}_{i+1} + (k_i - k_{i-1})\mathbf{P}_{i+2}}{k_{i+2} - k_{i-1}} \tag{5.14}$$

Thus, a B-spline with n control points can always be thought of as being made up of $n - 3$ Bézier curves, but some of those curves might be degenerate (zero length).

5.14.2 Knot Insertion

To insert a new knot k_j, first find where it fits in the knot vector by locating index i such that

$$k_i \le k_j \le k_{i+1}.$$

Then, replace control points $\mathbf{P}_{i+1}$ and $\mathbf{P}_{i+2}$ with the three control points

$$\mathbf{P}_A = \frac{(k_{i+1} - k_j)\mathbf{P}_i + (k_j - k_{i-2})\mathbf{P}_{i+1}}{k_{i+1} - k_{i-2}} \tag{5.15}$$

$$\mathbf{P}_B = \frac{(k_{i+2} - k_j)\mathbf{P}_{i+1} + (k_j - k_{i-1})\mathbf{P}_{i+2}}{k_{i+2} - k_{i-1}} \tag{5.16}$$

$$\mathbf{P}_C = \frac{(k_{i+3} - k_j)\mathbf{P}_{i+2} + (k_j - k_i)\mathbf{P}_{i+3}}{k_{i+3} - k_i}. \tag{5.17}$$

Of course, this requires renumbering of the control points $\mathbf{P}_j$, $j > i + 2$. The B-spline before knot insertion is identical to the one after knot insertion, except that the later has one additional knot and control point.

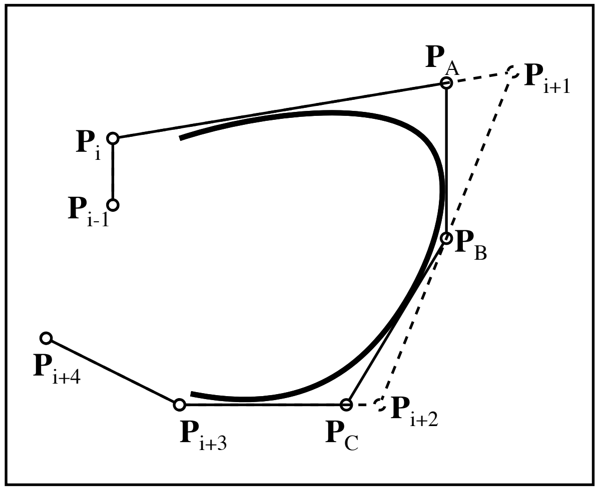

Figure 5.26: Knot insertion.

5.14.3 Combining Bézier curves into a B-spline

Here we suggest how to convert a string of cubic Bézier curves into a single B-spline. The process initializes by assigning the first four B-spline control points to be the control points of the first Bézier curve, and the knot vector is initially $[0,0,0,0,1,1,1,1]$.

Thereafter, each subsequent Bézier curve is analyzed to determine what order of continuity exists between it and the current B-spline, and it is appended to the B-spline as follows. Assume that at some step in this process, the B-spline has a knot vector $[k_{i-3}, k_{i-2}, k_{i-1}, k_i, k_{i+1}, k_{i+1}, k_{i+1}, k_{i+1}]$ with $k_{i-2} \le k_{i-1} \le k_i < k_{i+1}$, and the B-spline control points are labelled

$$\mathbf{P}_1, \ldots, \mathbf{P}_{n-3}, \quad \mathbf{P}_{n-2}, \quad \mathbf{P}_{n-1}, \quad \mathbf{P}_n.$$

The control points of the Bézier curve to be appended are

$$\mathbf{Q}_0 = \mathbf{P}_n, \quad \mathbf{Q}_1, \quad \mathbf{Q}_2, \quad \mathbf{Q}_3.$$

Then, depending on the continuity order between the B-spline and the Bézier curve, the B-spline after appending the Bézier becomes

Continuity	Knot Vector	Control Points
C^0	$[\ldots, k_{i-2}, k_{i-1}, k_i, k_{i+1}, k_{i+1}, k_{i+1}, e, e, e, e]$	$\ldots, \mathbf{P}_{n-3}, \mathbf{P}_{n-2}, \mathbf{P}_{n-1}, \mathbf{P}_n, \mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3$
C^1	$[\ldots, k_{i-2}, k_{i-1}, k_i, k_{i+1}, k_{i+1}, e, e, e, e]$	$\ldots, \mathbf{P}_{n-3}, \mathbf{P}_{n-2}, \mathbf{P}_{n-1}, \mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3$
C^2	$[\ldots, k_{i-2}, k_{i-1}, k_i, k_{i+1}, e, e, e, e]$	$\ldots, \mathbf{P}_{n-3}, \mathbf{P}_{n-2}, \mathbf{P}_{\alpha}, \mathbf{Q}_2, \mathbf{Q}_3$
C^3	$[\ldots, k_{i-2}, k_{i-1}, k_i, e, e, e, e]$	$\ldots, \mathbf{P}_{n-3}, \mathbf{P}_{\beta}, \mathbf{P}_{\gamma}, \mathbf{Q}_3$

C^0 continuity occurs if control points $\mathbf{P}_{n-1}$, $\mathbf{P}_n$, and $\mathbf{Q}_1$ are not collinear. If they are collinear, then the value of knot e is chosen so as to satisfy

$$|[\mathbf{P}_n - \mathbf{P}_{n-1}](k_{i+1} - k_i) - [\mathbf{Q}_1 - \mathbf{P}_n](e - k_{i+1})| < TOL$$

This provides for C^1 (not merely G^1) continuity. TOL is a small number which is needed to account for floating point error. An appropriate value for TOL is the width of the reverse map of a pixel into world space.

C^2 continuity occurs if, in addition to C^1 continuity, the relationship

$$|(\mathbf{P}_{n-2} - \mathbf{Q}_2)(k_{i+1} - k_{i-1})(k_{i+1} - e) + (\mathbf{P}_{n-1} - \mathbf{P}_{n-2})(e - k_{i-1})(k_{i+1} - e) + (\mathbf{Q}_2 - \mathbf{Q}_1)(k_i - e)(k_{i+1} - k_{i-1})| < TOL.$$

is satisfied. We can then compute

$$\mathbf{P}_\alpha = \frac{(k_{i+1} - e)\mathbf{P}_{n-2} + (e - k_{i-1})\mathbf{P}_{n-1}}{k_{i+1} - k_{i-1}} = \frac{(k_{i+1} - k_i)\mathbf{Q}_2 + (e - k_{i+1})\mathbf{Q}_1}{e - k_i}.$$

C^3 continuity occurs if, further, the relationship

$$\left|\mathbf{P}_\alpha - \frac{(e - k_{i+1})\mathbf{P}_\beta + (k_{i+1} - k_{i-1})\mathbf{P}_\gamma}{e - k_{i-1}}\right| < TOL$$

is satisfied, where

$$\mathbf{P}_\beta = \frac{(e - k_{i-2})\mathbf{P}_{n-2} + (k_{i-1} - e)\mathbf{P}_{n-3}}{k_{i-1} - k_{i-2}}$$

and

$$\mathbf{P}_\gamma = \frac{(k_{i+1} - k_i)\mathbf{Q}_3 + (k_i - e)\mathbf{Q}_2}{k_{i+1} - e}$$

5.15 B-spline blending functions

For completeness, this section discusses B-spline blending functions. Normally, those blending functions themselves are referred to as B-splines.

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

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

$$B_i^k(t) = \omega_i^k(t)B_i^{k-1}(t) + (1 - \omega_{i+1}^k(t))B_{i+1}^{k-1}(t) \tag{5.18}$$

where

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

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 5.18. The equation for the curve is:

$$\mathbf{P}(t) = \sum_{j=i}^{n+i} B_{j+1-n}^n \mathbf{P}(t_{j+1-n}, \ldots, t_j) \tag{5.19}$$