

## 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

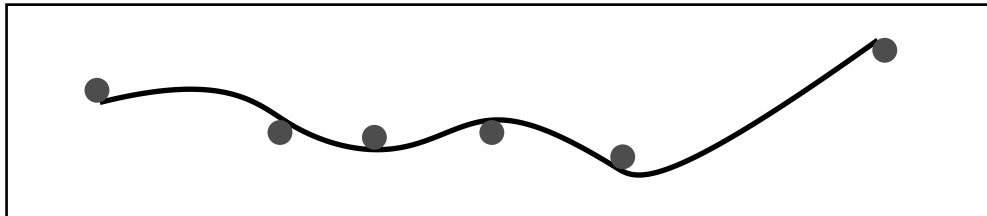


Figure 5.1: Spline and ducks

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 constituent 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* [35, 36, 37]. 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  $\mathbf{P}_i = \mathbf{P}(u_1, u_2, \dots, u_n)$  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

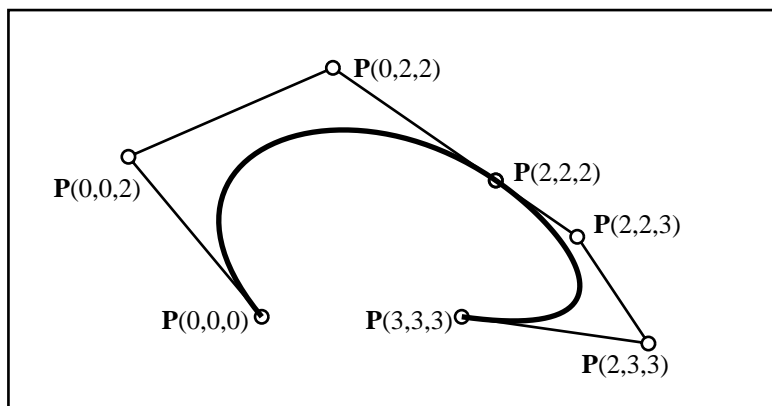


Figure 5.2: Bézier curves labeled using polar form.

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

2. For a degree  $n$  B-spline with a *knot vector* (explained later) 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  $i^{\text{th}}$  polar value being  $\mathbf{P}(t_i, \dots, t_{i+n-1})$ , as in Figure 5.3.

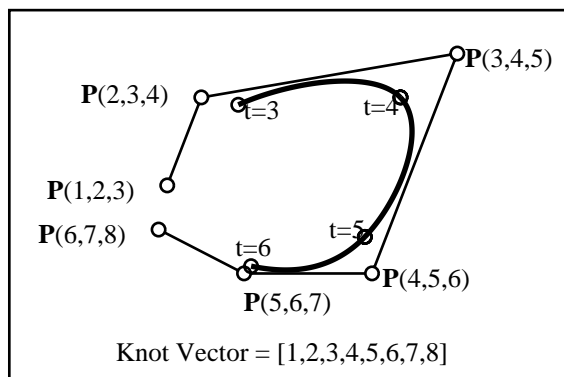


Figure 5.3: B-spline curve labeled using polar form.

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,

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

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

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

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

$$\begin{aligned} \mathbf{P}(0, t, 1) &= (1 - t) \times \mathbf{P}(0, 0, 1) + t \times \mathbf{P}(0, 1, 1), \\ \mathbf{P}(0, t) &= \frac{(4 - t) \times \mathbf{P}(0, 2) + (t - 2) \times \mathbf{P}(0, 4)}{2}, \\ \mathbf{P}(1, 2, 3, t) &= \frac{(t_2 - t) \times \mathbf{P}(2, 1, 3, t_1) + (t - t_1) \times \mathbf{P}(3, 2, 1, t_2)}{(t_2 - t_1)}. \end{aligned}$$

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

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

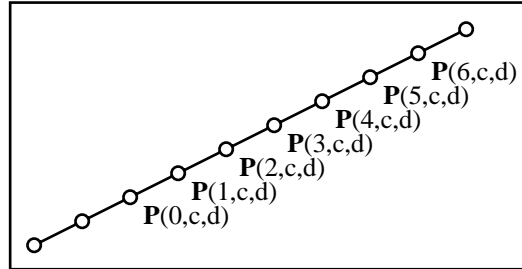


Figure 5.4: Affine map property of polar values.

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. Referring 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 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 5.6.

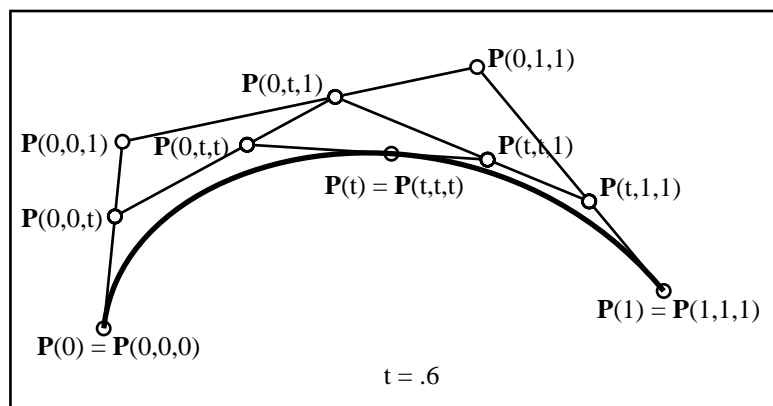


Figure 5.5: Subdividing a cubic Bézier curve.

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 5.4. 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.

### 5.3 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 [6].

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

$$\begin{aligned} & \mathbf{P}(3, 3, 3), \quad \mathbf{P}(3, 3, 4), \quad \mathbf{P}(3, 4, 4), \quad \mathbf{P}(4, 4, 4) \\ & \mathbf{P}(4, 4, 4), \quad \mathbf{P}(4, 4, 5), \quad \mathbf{P}(4, 5, 5), \quad \mathbf{P}(5, 5, 5) \\ & \mathbf{P}(5, 5, 5), \quad \mathbf{P}(5, 5, 6), \quad \mathbf{P}(5, 6, 6), \quad \mathbf{P}(6, 6, 6) \end{aligned}$$

respectively. 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 auxiliary 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  $\mathbf{P}(3, 3, 4)$ , and  $\mathbf{P}(4, 4, 4)$  is seen to be half way between  $\mathbf{P}(3, 4, 4)$  and  $\mathbf{P}(4, 4, 5)$ .

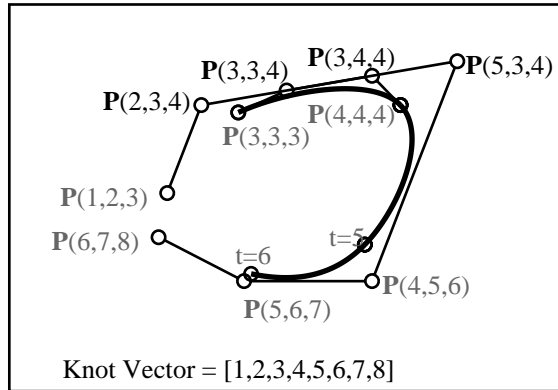


Figure 5.6: First step in Böhm algorithm.

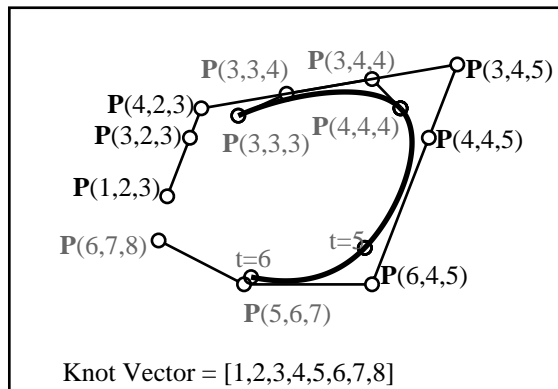


Figure 5.7: Second step in Böhm algorithm.

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.

## 5.4 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 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

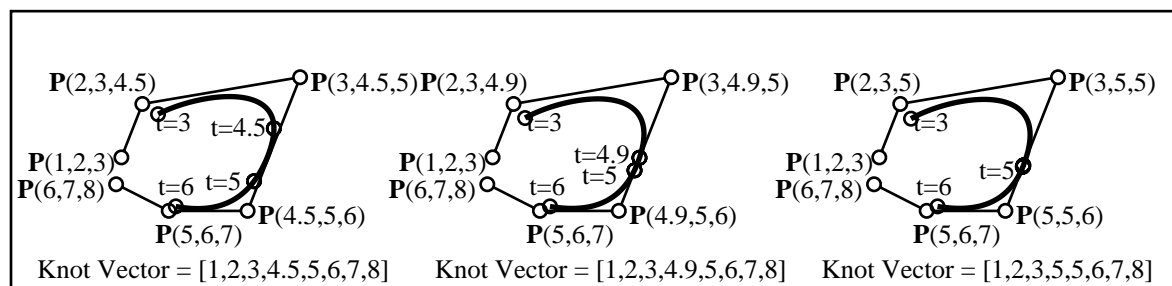


Figure 5.8: Double knot.

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

## 5.5 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 5.9.

## 5.6 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 (ie., 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.

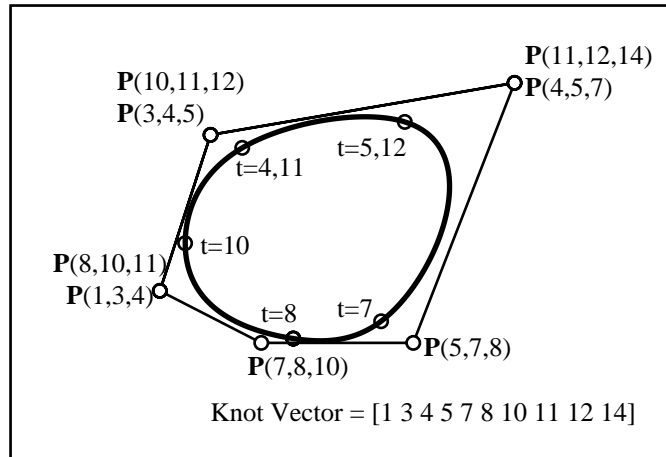


Figure 5.9: Periodic B-spline

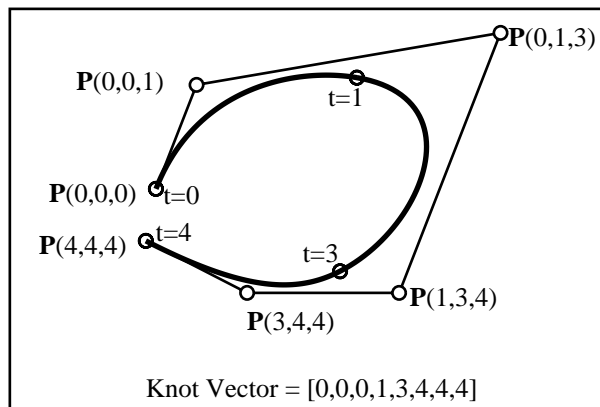


Figure 5.10: Bézier end condition.

	<b>Initial</b>	<b>After Knot Insertion</b>
Knot Vector:	[(0,0,0,1,3,4,4,4)]	[(0,0,0,1,2,3,4,4,4)]
Control Points:	<b>P</b> (0,0,0)	<b>P</b> (0,0,0)
	<b>P</b> (0,0,1)	<b>P</b> (0,0,1)
	<b>P</b> (0,1,3)	<b>P</b> (0,1,2)
	<b>P</b> (1,3,4)	<b>P</b> (1,2,3)
		<b>P</b> (2,3,4)
	<b>P</b> (3,4,4)	<b>P</b> (3,4,4)
	<b>P</b> (4,4,4)	<b>P</b> (4,4,4)

Figure 5.11: Before and after.

## 5.7 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 5.10. As shown in Figure 5.11, this involves replacing **P**(0, 1, 3) and **P**(1, 3, 4) with **P**(0, 1, 2), **P**(1, 2, 3), and **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.

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

## 5.8 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 **P**( $t$ ) has a polar value **P**( $t, t, \dots, t$ ), 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.

## 5.9 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

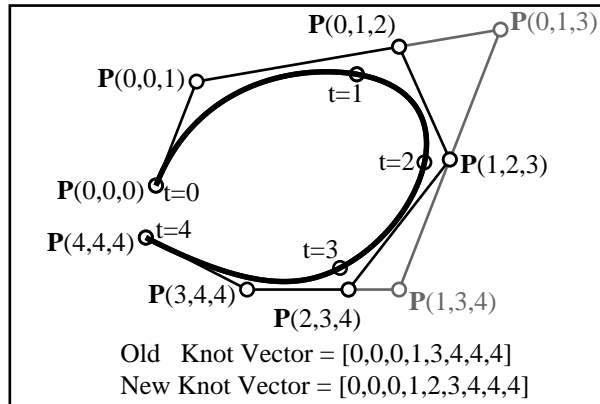


Figure 5.12: Knot insertion.

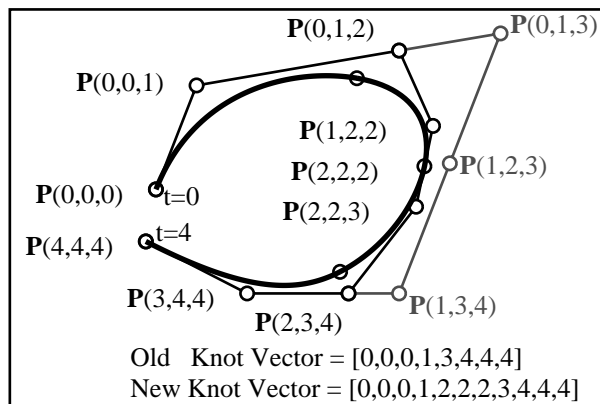


Figure 5.13: De Boor algorithm

known as *Greville abscissae*. For a degree  $n$  B-spline with  $m$  knots in the knot vector, the Greville abscissae are given by

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

## 5.10 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

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

where  $n$  is the degree.

## 5.11 Knot Intervals

B-splines originated in the field of approximation theory and were initially used for the purpose of approximating functions. In that context, parameter values are important, and hence, actual knot values are significant.

For use in curve design, we are almost never concerned about actual parameter values. It is obvious that B-spline curves are invariant under linear transformation of the knot vector—that is, if you add any constant to each knot (or multiply each knot by any constant) the curve does not change.

Another “problem” with knot vectors is that they are sort of isolated from the control polygon.

For these reasons, some people have begun to use “knot intervals” instead of knot vectors. Recall that for a cubic B-spline, each leg of the control polygon corresponds exactly to a single Bézier curve, and that each Bézier curve begins and ends at parameter values that belong to the knot vector. Let’s call the difference between two adjacent knots in a knot vector a knot interval. Then, a very nice way to specify the knot information for a B-spline is to simply assign one knot interval to each edge of the control polygon. (This works for even-degree B-splines; for odd degree, the knot interval should be assigned to the control points). Since the knot intervals convey all pertinent information of the knot vector, we don’t need polar values; we can label the control points as  $\mathbf{P}_i$ . For example, for the B-spline in Figure 5.14, the knot intervals are assigned as shown in Figure 5.15.

Notice that due to the end-condition knots, there are two more knot intervals than edges of the control polygon. Therefore, we must hang one knot interval off each end of the control polygon. For periodic B-splines, things are even more simple, since we don’t need to worry about end conditions. Figure 5.16 shows two periodic B-splines labelled with knot intervals. The small circles drawn on the curves show where the Bézier curves meet. Notice that when a knot interval increases, the length of the corresponding Bézier curve tends to increase also.

Figure 5.17 shows two periodic B-splines with a double knot (by setting  $d_0 = 0$ ) and a triple knot (by setting  $d_0 = d_1 = 0$ ).

We have seen how polar values provide an easy-to-remember method for doing knot insertion, and for splitting a B-spline into Bézier curves. Knot intervals can also give some nice insight for doing those tasks. The idea is to split 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 5.18. (For a B-spline of even

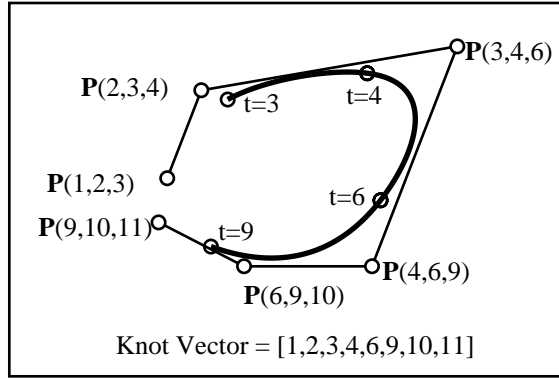


Figure 5.14: Sample B-spline

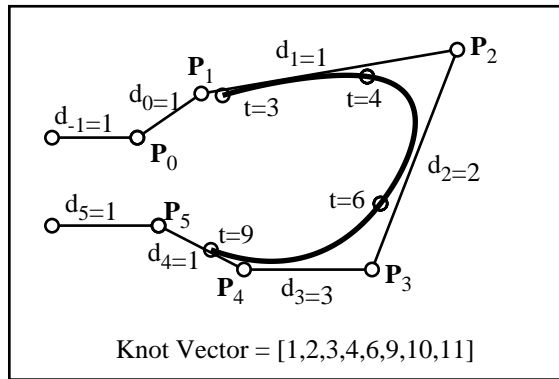


Figure 5.15: Sample B-spline with knot intervals

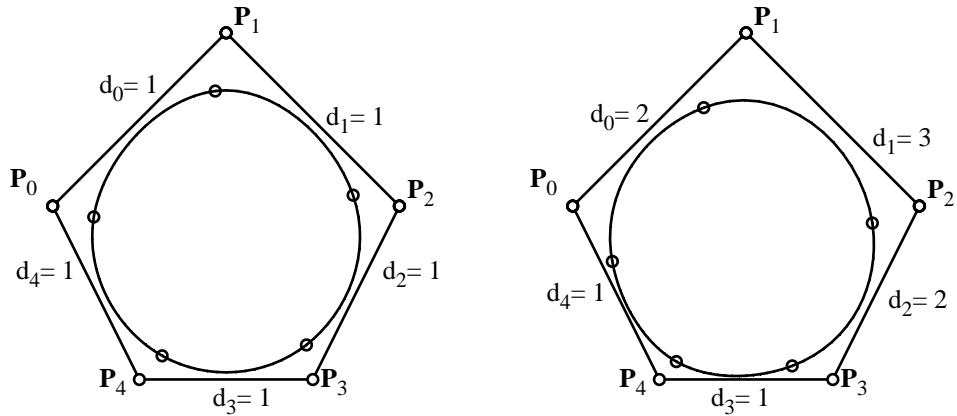


Figure 5.16: Periodic B-splines labelled with knot intervals

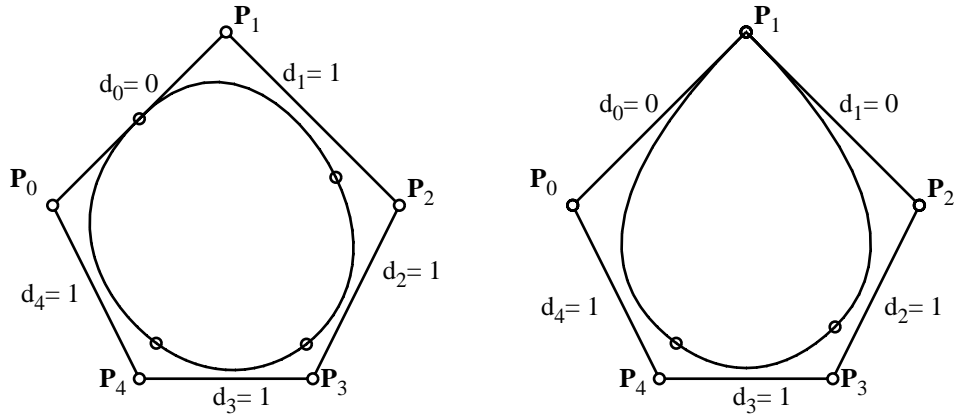


Figure 5.17: Periodic B-splines with double and triple knots.

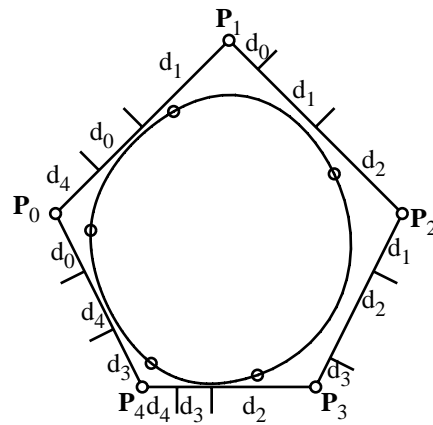


Figure 5.18: Cubic B-spline with control polygon marked.

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

Now, suppose we want to insert a knot. In terms of knot intervals, this really means splitting a knot interval at some value  $t \in [0, 1]$ . For example, suppose we wish to split knot interval  $d_1$  at

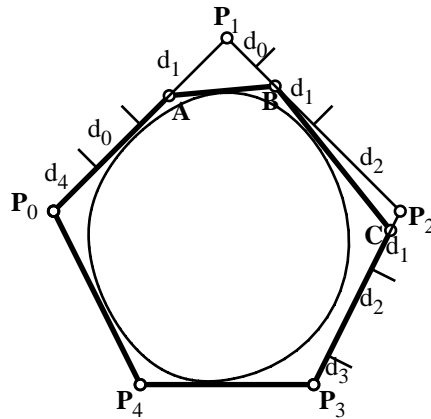


Figure 5.19: Knot insertion using knot intervals.

$t = 1/over3$ . 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 5.19.

## 5.12 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.

### 5.12.1 Splitting a B-spline into Bézier Curves

Cubic B-splines are defined by specifying  $n > 3$  control points  $\mathbf{P}_1, \dots, \mathbf{P}_n$ , and a knot vector  $[k_{-2}, \dots, k_{n+1}]$ , 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  $0 \leq t \leq 1$ . For the  $i^{\text{th}}$  Bézier curve in a B-spline, the parameter range is  $k_i \leq t \leq k_{i+1}$ . As far as the appearance of a Bézier curve is concerned, for four given control points, a Bézier curve defined over the range  $k_i \leq t \leq k_{i+1}$  looks identical to one defined over the parameter interval  $0 \leq t \leq 1$ . 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 5.20 shows the  $i^{\text{th}}$  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  $\mathbf{P}_{i-1} \dots \mathbf{P}_{i+4}$ ). The Bézier control points are

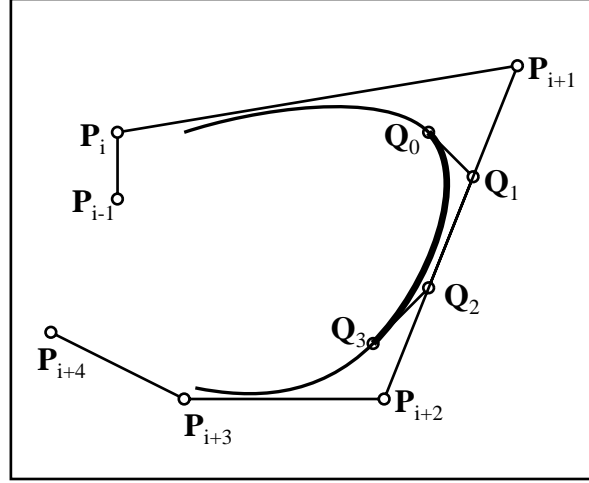


Figure 5.20: Extracting a Bézier curve from a B-spline

obtained using the formulas:

$$(5.3) \quad \mathbf{Q}_1 = \frac{(k_{i+2} - k_i)\mathbf{P}_{i+1} + (k_i - k_{i-1})\mathbf{P}_{i+2}}{k_{i+2} - k_{i-1}}$$

$$(5.4) \quad \mathbf{Q}_0 = \frac{k_{i+1} - k_i}{k_{i+1} - k_{i-1}} \frac{(k_{i+1} - k_i)\mathbf{P}_i + (k_i - k_{i-2})\mathbf{P}_{i+1}}{k_{i+1} - k_{i-2}} + \frac{k_i - k_{i-1}}{k_{i+1} - k_{i-1}} \mathbf{Q}_1$$

$$(5.5) \quad \mathbf{Q}_2 = \frac{(k_{i+2} - k_{i+1})\mathbf{P}_{i+1} + (k_{i+1} - k_{i-1})\mathbf{P}_{i+2}}{k_{i+2} - k_{i-1}}$$

$$(5.6) \quad \mathbf{Q}_3 = \frac{k_{i+1} - k_i}{k_{i+2} - k_i} \frac{(k_{i+3} - k_{i+1})\mathbf{P}_{i+2} + (k_{i+1} - k_i)\mathbf{P}_{i+3}}{k_{i+3} - k_i} + \frac{k_{i+2} - k_{i+1}}{k_{i+2} - k_i} \mathbf{Q}_2$$

Notice from equations 5.3–5.6 that if  $k_i = k_{i+1}$ , Bézier curve  $i$  collapses to a single point:

$$(5.7) \quad \mathbf{Q}_0 = \mathbf{Q}_1 = \mathbf{Q}_2 = \mathbf{Q}_3 = \frac{(k_{i+2} - k_i)\mathbf{P}_{i+1} + (k_i - k_{i-1})\mathbf{P}_{i+2}}{k_{i+2} - k_{i-1}}$$

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).

### 5.12.2 Knot Insertion

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

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

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

$$(5.8) \quad \mathbf{P}_A = \frac{(k_{i+1} - k_j)\mathbf{P}_i + (k_j - k_{i-2})\mathbf{P}_{i+1}}{k_{i+1} - k_{i-2}}$$

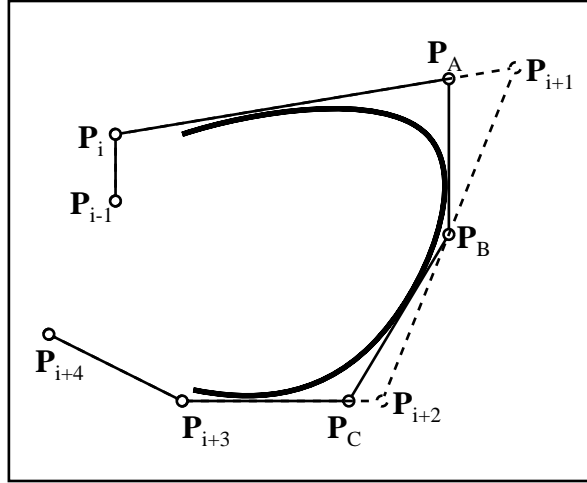


Figure 5.21: Knot insertion

$$(5.9) \quad \mathbf{P}_B = \frac{(k_{i+2} - k_j)\mathbf{P}_{i+1} + (k_j - k_{i-1})\mathbf{P}_{i+2}}{k_{i+2} - k_{i-1}}$$

$$(5.10) \quad \mathbf{P}_C = \frac{(k_{i+3} - k_j)\mathbf{P}_{i+2} + (k_j - k_i)\mathbf{P}_{i+3}}{k_{i+3} - k_i}.$$

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.

### 5.12.3 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  $k_{i-2} \leq k_{i-1} \leq k_i < k_{i+1}$ , and the B-spline control points are labelled

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

The control points of the Bézier curve to be appended are

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

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

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

$C^0$  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

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

## 5.13 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

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

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:

$$(5.11) \quad B_i^k(t) = \omega_i^k(t)B_i^{k-1}(t) + (1 - \omega_{i+1}^k(t))B_{i+1}^{k-1}(t)$$

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  $\{\dots, t_{i-1}, t_i, t_{i+1}, t_{i+2}, \dots\}$  has  $n + 1$  control points written as polar values

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

and blending functions  $B_i^n(t)$  which are obtained from equation 5.11. The equation for the curve is:

$$(5.12) \quad \mathbf{P}(t) = \sum_{j=i}^{n+i} B_{j+1-n}^n \mathbf{P}(t_{j+1-n}, \dots, t_j)$$