Lecture 8: Bézier curves

Michael S. Floater

October 8, 2018

These notes provide an introduction to Bézier curves.

1 Introduction

Computer-aided geometric design (CAGD) is the design of geometrical shapes using computer technology, and is used extensively in many applications, such as the automotive, shipbuilding, and aerospace industries, architectural design, and computer animation. A popular way of modelling geometry in CAGD is to represent the outer surface, or curve, of the object as a patchwork of parametric polynomial pieces. Bézier curves and surfaces are a representation of such polynomial pieces that makes their interactive design easier and more intuitive than with other representations. They were developed in the 1960's and 1970's by Paul de Casteljau and Pierre Bézier, for use in the automotive industry.

2 Bernstein polynomials

Recall that a real polynomial of a real variable $t \in \mathbb{R}$, with degree $\leq n$, is a function of the form

$$p(t) = a_0 + a_1 t + \cdots + a_n t^n = \sum_{i=0}^{n} a_i t^i, \qquad a_i \in \mathbb{R}.$$

We will denote by π_n the linear (vector) space of all such polynomials. The actual degree of p is the largest i for which a_i is non-zero. The functions $1, t, \ldots, t^n$ form a *basis* for π_n, known as the *monomial basis*, and the *dimension* of the space π_n is therefore $n + 1$.

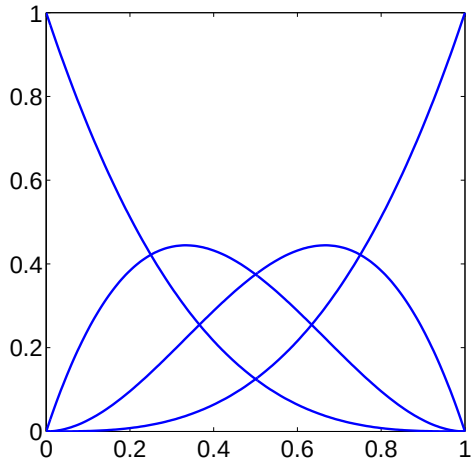


Figure 1: The cubic Bernstein polynomials

Bernstein polynomials are an alternative basis for π_n, and are used to construct Bézier curves. The i-th Bernstein polynomial of degree n is

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \tag{1}$$

where $0 \le i \le n$ and

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}.$$

The first few examples are

$$B_0^0(t) = 1,$$

$$B_0^1(t) = 1 - t, \quad B_1^1(t) = t,$$

$$B_0^2(t) = (1-t)^2, \quad B_1^2(t) = 2t(1-t), \quad B_2^2(t) = t^2,$$

$$B_0^3(t) = (1-t)^3, \quad B_1^3(t) = 3t(1-t)^2, \quad B_2^3(t) = 3t^2(1-t), \quad B_3^3(t) = t^3.$$

The cubic ones are shown in Figure 1. These polynomials are defined for all $t \in \mathbb{R}$, but are usually restricted to $t \in [0,1]$ in practice. They have various important properties. They are *linearly independent*, for if

$$\sum_{i=0}^{n} c_i t^i (1-t)^{n-i} = 0, \qquad t \in (0,1),$$

then, by dividing by $(1-t)^n$ and letting $y = t/(1-t)$, we see that

$$\sum_{i=0}^{n} c_i y^i = 0, \qquad y > 0,$$

which implies that $c_0 = c_1 = \cdots = c_n = 0$. Since there are $n+1$ Bernstein polynomials of degree n, they do indeed form a basis for π_n. They are positive for t in the open interval $(0,1)$ and at the endpoints,

$$B_i^n(0) = \begin{cases} 1 & i = 0; \\ 0 & i = 1, \ldots, n, \end{cases} \quad \text{and} \quad B_i^n(1) = \begin{cases} 0 & i = 0, \ldots, n-1; \\ 1 & i = n. \end{cases} \tag{2}$$

They form a *partition of unity*: by the binomial theorem,

$$\sum_{i=0}^{n} B_i^n(t) = (t + (1-t))^n = 1^n = 1.$$

They satisfy the *recursion formula*,

$$B_i^n(t) = t B_{i-1}^{n-1}(t) + (1-t) B_i^{n-1}(t), \tag{3}$$

which follows from the definition (1) and the binomial identity

$$\binom{n}{i} = \binom{n-1}{i-1} + \binom{n-1}{i}.$$

In (3) and elsewhere we define

$$B_{-1}^n = B_{n+1}^n = 0.$$

The computation of the Bernstein polynomials up to degree n can be arranged in the following triangular scheme, with each column being computed from the previous column, starting from the left:

$$\begin{array}{ccccccc} 1 & = & B_0^0 & B_0^1 & B_0^2 & \cdots & B_0^n \\ & & & B_1^1 & B_1^2 & \cdots & B_1^n \\ & & & & B_2^2 & \cdots & B_2^n \\ & & & & & \ddots & \vdots \\ & & & & & & B_n^n \end{array}$$

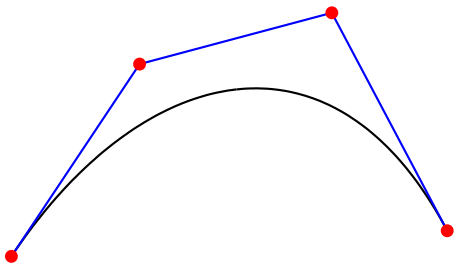


Figure 2: A cubic Bézier curve

3 Bézier curves

When modelling geometry in some Euclidean space $\mathbb{R}^d$ we often model a curve, or part of a curve, as a *parametric* polynomial,

$$\mathbf{p}(t) = \sum_{i=0}^{n} \mathbf{a}_i t^i, \qquad \mathbf{a}_i \in \mathbb{R}^d. \tag{4}$$

Since the Bernstein polynomials of degree n form a basis for π_n, $\mathbf{p}$ can also be represented in the Bernstein form,

$$\mathbf{p}(t) = \sum_{i=0}^{n} \mathbf{c}_i B_i^n(t), \qquad \mathbf{c}_i \in \mathbb{R}^d. \tag{5}$$

A polynomial curve expressed in this form is known as a *Bézier curve* and the points $\mathbf{c}_i$ are known as the *control points* of $\mathbf{p}$. The curve is usually restricted to the parameter domain (parameter interval) $[0, 1]$, but is well defined also for t outside $[0, 1]$. The polygon formed by connecting the sequence of control points $\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_n$ is known as the *control polygon* of $\mathbf{p}$. Figure 2 shows a cubic Bézier curve with its control polygon. To a large extent the shape of a Bézier curve relects the shape of its control polygon, which is why it is a popular choice for designing geometry in an interactive graphical environment. As the user moves the control points interactively, the shape of the Bézier curve tends to change in an intuitive and predictable way.

Various properties of Bézier curves follow from properties of the Bernstein polynomials. For example, from (2), we obtain the *endpoint property* of Bézier curves,

$$\mathbf{p}(0) = \mathbf{c}_0, \qquad \mathbf{p}(1) = \mathbf{c}_n.$$

Since the Bernstein polynomials sum to one, every point $\mathbf{p}(t)$ is an *affine combination* of the control points $\mathbf{c}_0, \ldots, \mathbf{c}_n$. From this it follows that Bézier curves are *affinely invariant*, i.e., if Φ is an affine map in $\mathbb{R}^d$ then the mapped curve $\Phi(\mathbf{p})$ has control points $\Phi(\mathbf{c}_i)$. So see this, recall that an affine map has the form

$$\Phi(\mathbf{x}) = A\mathbf{x} + \mathbf{b}, \qquad \mathbf{x} \in \mathbb{R}^d,$$

for some matrix A of dimension $d \times d$ and a vector $\mathbf{b}$ of length d. Since the Bernstein polynomials sum to one,

$$\Phi(\mathbf{p}(t)) = \sum_{i=0}^{n} A\mathbf{c}_i B_i^n(t) + \mathbf{b} = \sum_{i=0}^{n} (A\mathbf{c}_i + \mathbf{b}) B_i^n(t) = \sum_{i=0}^{n} \Phi(\mathbf{c}_i) B_i^n(t).$$

Since the Bernstein polynomials are non-negative in $[0, 1]$, the point $\mathbf{p}(t)$, with $t \in [0, 1]$, is a *convex combination* of the control points $\mathbf{c}_0, \ldots, \mathbf{c}_n$, and so $\mathbf{p}$, restricted to $[0, 1]$, lies in the *convex hull* of its control points:

$$C := \{\sum_{i=0}^{n} \lambda_i \mathbf{c}_i : \lambda_0, \ldots, \lambda_n \geq 0, \sum_i \lambda_i = 1\}.$$

By treating each of the d coordinates of $\mathbf{p}$ separately, a similar reasoning shows that $\mathbf{p}$ restricted to $[0, 1]$ also lies in the *bounding box*

$$[\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times \cdots \times [\alpha_d, \beta_d],$$

where, if the point $\mathbf{c}_i$ has coordinates $c_{i1}, \ldots, c_{id}$,

$$\alpha_k = \min_{0 \leq i \leq n} c_{ik} \quad \text{and} \quad \beta_k = \max_{0 \leq i \leq n} c_{ik}, \qquad k = 1, \ldots, d.$$

Bounding boxes are used in various algorithms, and are easier to compute than convex hulls.

4 The de Casteljau algorithm

One way of computing a point $\mathbf{p}(t)$ of the Bézier curve $\mathbf{p}$ is first to evaluate the Bernstein polynomials B_i^n at t and then use the formula (5). Another, more direct method is de Casteljau's algorithm. We first set $\mathbf{c}_i^0 = \mathbf{c}_i$, and then for each $r = 1, \ldots, n$, let

$$\mathbf{c}_i^r = (1 - t)\mathbf{c}_i^{r-1} + t\mathbf{c}_{i+1}^{r-1}, \qquad i = 0, 1, \ldots, n - r. \tag{6}$$

The last point computed in this algorithm is the point on the curve:

$$\mathbf{p}(t) = \mathbf{c}_0^n.$$

We can show this by showing more generally that for any $r = 0, 1, \ldots, n$,

$$\mathbf{p}(t) = \sum_{i=0}^{n-r} \mathbf{c}_i^r B_i^{n-r}(t). \tag{7}$$

This equation clearly holds for $r = 0$, and we can use induction on r for the general case. Suppose (7) holds for some $r \leq n-1$. Then applying the recursion formula (3) to the right hand side gives

$$\begin{aligned}\mathbf{p}(t) &= \sum_{i=0}^{n-r} \mathbf{c}_i^r \left(tB_{i-1}^{n-r-1}(t) + (1-t)B_i^{n-r-1}(t)\right) \\ &= \sum_{i=0}^{n-r-1} \left(t\mathbf{c}_{i+1}^r + (1-t)\mathbf{c}_i^r\right) B_i^{n-r-1}(t).\end{aligned}$$

This shows that (7) also holds with r replaced by $r+1$, and therefore, by induction, for all $r = 0, 1, \ldots, n$. Like the recursive algorithm for computing Bernstein polynomials, the de Casteljau algorithm can be viewed as a triangular scheme, here arranged row-wise, with each row being computed from the row above:

$$\begin{array}{ccccccccc}
\mathbf{c}_0^0 & & \mathbf{c}_1^0 & & \mathbf{c}_2^0 & & \cdots & & \mathbf{c}_n^0 \\
& \mathbf{c}_0^1 & & \mathbf{c}_1^1 & & \cdots & & \mathbf{c}_{n-1}^1 & \\
& & \ddots & & & & \iddots & & \\
& & & \mathbf{c}_0^{n-1} & & \mathbf{c}_1^{n-1} & & & \\
& & & & \mathbf{c}_0^n & & & &
\end{array}$$

Figure 3 illustrates the algorithm applied to the Bézier curve of Figure 2 with $t = 0.3$.

A numerical example with $n = 3$ is $t = 2/3$, $d = 1$ and

$$[\mathbf{c}_0 \quad \mathbf{c}_1 \quad \mathbf{c}_2 \quad \mathbf{c}_3] = [4 \quad 0 \quad 4 \quad 18].$$

The three steps of the algorithm are then:

$$[4 \quad 0 \quad 4 \quad 18] \to [4/3 \quad 8/3 \quad 40/3] \to [20/9 \quad 88/9] \to [196/27].$$

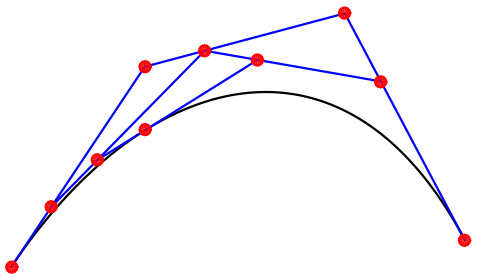


Figure 3: de Casteljau algorithm, $t = 0.3$

The point $\mathbf{c}_i^r$ of the de Casteljau algorithm is itself the result of applying r steps of the algorithm to the subsequence of control points $\mathbf{c}_i, \mathbf{c}_{i+1}, \ldots \mathbf{c}_{i+r}$, and so $\mathbf{c}_i^r$, viewed as a function of t, is itself a Bézier curve of degree r:

$$\mathbf{c}_i^r = \mathbf{c}_i^r(t) = \sum_{j=0}^{r} \mathbf{c}_{i+j} B_j^r(t). \tag{8}$$

5 Derivatives

To find the first derivative of $\mathbf{p}$ in (5), we use a formula for the derivative of a Bernstein polynomial. Differentiating (1) with respect to t and using the product rule gives

$$\frac{d}{dt} B_i^n(t) = n \left(B_{i-1}^{n-1}(t) - B_i^{n-1}(t) \right). \tag{9}$$

From these observations we find that the derivative of the Bézier curve in (5) with respect to its parameter t is

$$\mathbf{p}'(t) = n \left(\sum_{i=0}^{n-1} \mathbf{c}_{i+1} B_i^{n-1}(t) - \sum_{i=0}^{n-1} \mathbf{c}_i B_i^{n-1}(t) \right). \tag{10}$$

Thus one way of expressing the derivative is as

$$\mathbf{p}'(t) = n \sum_{i=0}^{n-1} \Delta \mathbf{c}_i B_i^{n-1}(t), \tag{11}$$

where Δ denotes the forward difference operator,

$$\Delta \mathbf{c}_i = \mathbf{c}_{i+1} - \mathbf{c}_i.$$

This means that the derivative of $\mathbf{p}$ is itself a Bézier curve with control points, which we now view as vectors, $n\Delta\mathbf{c}_i$. For each $t \in [0,1]$, the tangent vector $\mathbf{p}'(t)$ lies in the convex cone of the vectors $\Delta\mathbf{c}_i$, i.e., in

$$C := \{\sum_{i=0}^{n-1} \lambda_i \Delta\mathbf{c}_i : \lambda_0, \ldots, \lambda_{n-1} \geq 0\}.$$

and by the endpoint property of Bézier curves,

$$\mathbf{p}'(0) = n\Delta\mathbf{c}_0 \quad \text{and} \quad \mathbf{p}'(1) = n\Delta\mathbf{c}_{n-1}.$$

An alternative way of expressing the derivative is in terms of the intermediate points (8) of the de Casteljau algorithm. Setting $r = n-1$ in (8), it follows from (10) that

$$\mathbf{p}'(t) = n\left(\mathbf{c}_1^{n-1}(t) - \mathbf{c}_0^{n-1}(t)\right) = n\Delta\mathbf{c}_0^{n-1}(t). \tag{12}$$

6 Higher derivatives

Applying the derivative formula (11) repeatedly leads to

$$\mathbf{p}^{(r)}(t) = \frac{d^r}{dt^r}\mathbf{p}(t) = \frac{n!}{(n-r)!}\sum_{i=0}^{n-r} \Delta^r\mathbf{c}_i B_i^{n-r}(t),$$

for any $r = 1, \ldots, n$, where Δ^r is the r-th forward difference operator

$$\Delta^r\mathbf{c}_i = \Delta^{r-1}\mathbf{c}_{i+1} - \Delta^{r-1}\mathbf{c}_i.$$

At the endpoints of the curve, the r-th derivative depends only on the first or last $r+1$ control points:

$$\mathbf{p}^{(r)}(0) = \frac{n!}{(n-r)!}\Delta^r\mathbf{c}_0 \quad \text{and} \quad \mathbf{p}^{(r)}(1) = \frac{n!}{(n-r)!}\Delta^r\mathbf{c}_{n-r}.$$

Alternatively, in terms of the intermediate de Casteljau points, differentiating (12) repeatedly gives

$$\mathbf{p}^{(r)}(t) = \frac{n!}{(n-r)!}\Delta^r\mathbf{c}_0^{n-r}(t).$$

7 Joining curves together

If we tried to model a complex curve with a single Bézier curve we would need a polynomial of high degree. It is easier in practice to create complex geometry by joining together several Bézier curves, each with relatively low degree. Degrees such as two and three often (though not always) provide enough flexibility. Due to the Bernstein basis, the conditions for joining Bézier curves together smoothly are quite simple: they would not be so simple if we used the monomial basis for example.

Consider again the Bézier curve

$$\mathbf{p}(t) = \sum_{i=0}^{n} \mathbf{c}_i B_i^n(t), \qquad \mathbf{c}_i \in \mathbb{R}^d, \tag{13}$$

for $t \in [0, 1]$, and suppose we want to join a second Bézier curve, $\mathbf{q}$, to $\mathbf{p}$, where $\mathbf{q}$ is defined on the adjacent interval $[1, 2]$, i.e.,

$$\mathbf{q}(t) = \sum_{i=0}^{n} \mathbf{d}_i B_i^n(t-1), \qquad \mathbf{d}_i \in \mathbb{R}^d,$$

for $t \in [1, 2]$. We would like the composite curve

$$\mathbf{s}(t) := \begin{cases} \mathbf{p}(t), & 0 \le t < 1; \\ \mathbf{q}(t), & 1 \le t < 2; \end{cases}$$

to be at least continuous at the breakpoint $t = 1$. For $\mathbf{s}$ to be continuous, we require that $\mathbf{q}(1) = \mathbf{p}(1)$, which, by the endpoint property of Bézier curves is simply the condition

$$\mathbf{d}_0 = \mathbf{c}_n. \tag{14}$$

How do we make $\mathbf{p}$ and $\mathbf{q}$ join with higher orders of smoothness (continuity)? The curves $\mathbf{p}$ and $\mathbf{q}$ join with continuity of order k at $t = 1$, written C^k, if

$$\mathbf{q}^{(r)}(1) = \mathbf{p}^{(r)}(1) \qquad \text{for all } r = 0, 1, \ldots, k.$$

This is equivalent to saying that $\mathbf{s}$ is C^k at $t = 1$. From the formula for end point derivatives, this condition is equivalent to

$$\Delta^r \mathbf{d}_0 = \Delta^r \mathbf{c}_{n-r}, \qquad r = 0, 1, \ldots, k. \tag{15}$$

For example, the condition for C^1 smoothness is

$$\begin{aligned} \mathbf{d}_0 &= \mathbf{c}_n, \\ \mathbf{d}_1 - \mathbf{d}_0 &= \mathbf{c}_n - \mathbf{c}_{n-1}, \end{aligned}$$

and the condition for C^2 smoothness is

$$\begin{aligned} \mathbf{d}_0 &= \mathbf{c}_n, \\ \mathbf{d}_1 - \mathbf{d}_0 &= \mathbf{c}_n - \mathbf{c}_{n-1}, \\ \mathbf{d}_2 - 2\mathbf{d}_1 + \mathbf{d}_0 &= \mathbf{c}_n - 2\mathbf{c}_{n-1} + \mathbf{c}_{n-2}. \end{aligned}$$

8 Exercises

Exercise 1 *Let* $\mathbf{p}(t) = \sum_{i=0}^{n} \mathbf{c}_i B_i^n(t)$, $\mathbf{c}_i \in \mathbb{R}^d$, *be a Bezier curve in* $\mathbb{R}^d$. *Show that its length*

$$L = \int_0^1 \|\mathbf{p}'(t)\|\, dt,$$

is less than or equal to the length of its control polygon.

Exercise 2 *Consider the composite curve* $\mathbf{s} : [0, 2] \to \mathbb{R}^2$, *with pieces*

$$\mathbf{p}(t) = \sum_{i=0}^{2} \mathbf{c}_i B_i^2(t), \qquad 0 \le t < 1,$$

$$\mathbf{q}(t) = \sum_{i=0}^{2} \mathbf{d}_i B_i^2(t-1), \qquad 1 \le t < 2,$$

where $\mathbf{c}_0 = (-1, 1)$, $\mathbf{c}_1 = (-1, 0)$, $\mathbf{c}_2 = (0, 0)$, *and* $\mathbf{d}_0 = (0, 0)$, $\mathbf{d}_1 = (1, 0)$, $\mathbf{d}_2 = (2, 1)$. *(a) What is the order of continuity of* $\mathbf{s}$ *at* $(0, 0)$*? (b) The curvature of a parametric curve* $\mathbf{r}$ *in* $\mathbb{R}^2$ *is*

$$\kappa(\mathbf{r}(t)) = \frac{\mathbf{r}'(t) \times \mathbf{r}''(t)}{\|\mathbf{r}'(t)\|^3},$$

where $(a_1, a_2) \times (b_1, b_2) := a_1 b_2 - a_2 b_1$. *What are the curvatures of* $\mathbf{p}$ *and* $\mathbf{q}$ *at* $(0, 0)$*?*

Exercise 3 *Bezier curves can be generalized to rational Bezier curves, which have the form*

$$\mathbf{r}(t) = \frac{\sum_{i=0}^{n} w_i \mathbf{c}_i B_i^n(t)}{\sum_{i=0}^{n} w_i B_i^n(t)},$$

where $w_0, w_1, \ldots, w_n > 0$ *are chosen user-defined weights. If the weights are equal we recover the usual polynomial Bezier curve. If we increase the weight* w_i *we tend to drag the curve towards the control point* $\mathbf{c}_i$*.*

Consider the rational quadratic Bezier curve

$$\mathbf{r}(t) = \frac{(1-t)^2 w_0 \mathbf{c}_0 + 2t(1-t)w_1 \mathbf{c}_1 + t^2 \mathbf{c}_2}{(1-t)^2 w_0 + 2t(1-t)w_1 + t^2}$$

in the plane, with control points $\mathbf{c}_0 = (1,0)$*,* $\mathbf{c}_1 = (1,1)$*,* $\mathbf{c}_2 = (0,1)$*. Let* $w_0 = w_2 = 1$*. Find* w_1 *such that* $\mathbf{r}$ *is a circular arc.*