MA 323 Geometric Modelling
Course Notes: Day 13
Bezier Curves & Bernstein Polynomials

David L. Finn

Over the past few days, we have looked at de Casteljau's algorithm for generating a polynomial curve, and we have examined the subdivision method for approximating the curve generated by de Casteljau's algorithm. de Casteljau's algorithm is useful as a computational and theoretical device, but it is also frequently useful to look at a basis function form for the curve given by de Casteljau's algorithm. Specifically, today, we examine curves described by affine combinations

$$c(t) = \alpha_0(t)\, p_0 + \alpha_1(t)\, p_1 + \cdots + \alpha_n(t)\, p_n,$$

and examine the properties of the coefficients $\alpha_i(t)$. The basis function form for de Casteljau's algorithm is also called the Bezier form or the Bernstein-Bezier form, after Pierre Bezier an engineer working for Renault who developed some of the methods for creating such curves and working with a basis function approach. It is also typical to called the curves produced by de Casteljau's algorithm Bezier curves, after Bezier.

You may wonder why both Bezier's name and de Casteljau's name are both attached to the same type of curve. The reason for attaching both names to the same type of curve is simple. In the early days of computer aided geometric design, the primary methods were developed as industrial secrets. de Casteljau's algorithm was developed by P. de Casteljau working for Citroen and Bezier curves were developed by P. Bezier working for Renault. Both of these methods were developed in the early 1960's when CAD/CAM was first beginning to become an industrial tool. The original methods were carefully guarded industrial secrets for well-over a decade. The original papers in which they were developed were never published except as technical reports to the interior research and development teams at Renault and Citroen. When the methods later surfaced, Bezier's name was attached to the curves as his methods were circulated first, and later de Casteljau's methods were learned.

Before examining the basis functions in particular, we first look at how one can define derivatives and perform the basic operations of calculus on curves defined by affine combinations. It is important to note that we can perform calculus on the curves, without working through coordinates as in multivariate calculus. We then will derive the explicit form for the basis functions of a Bezier curve and then examine some of the properties of the basis functions. The basis functions for an nth degree Bezier curve are nth degree polynomials known as Bernstein polynomials.

13.1 Calculus of Curves Defined by Affine Combinations

Through de Casteljau's algorithm, we have defined a polynomial curve as an affine combination (once we have expanded the coefficients). This means, we have a curve $c(t)$ defined by

$$c(t) = \alpha_0(t)\,p_0 + \alpha_1(t)\,p_1 + \cdots + \alpha_n(t)\,p_n.$$

for some functions $\alpha_i : [0,1] \to \mathbf{R}$ with the property

$$1 = \alpha_0(t) + \alpha_1(t) + \cdots + \alpha_n(t),$$

we examine in this subsection the derivatives of such a curve. First, recall the definition of the derivative of such a curve is defined as

$$c'(t) = \lim_{h\to 0} \frac{c(t+h) - c(t)}{h}.$$

Note that the derivative implies that the derivative is a vector, because the derivative is defined as the difference of two points then multiplied by a length. The calculation of the derivative can be accomplished by differentiating the coefficients, as the points do not depend on t. In fact, in this manner, one is never concerned with the location of the points to determine properties of the derivatives.

Several properties of the derivatives of curves defined by affine combinations arise by implicit differentiation of the equation

$$1 = \alpha_0(t) + \alpha_1(t) + \cdots + \alpha_n(t).$$

In particular, implicit differentiation implies that

$$0 = \alpha_0'(t) + \alpha_1'(t) + \cdots + \alpha_n'(t).$$

Note that as an affine combination this means the derivative expressed as

$$c'(t) = \alpha_0'(t)\,p_0 + \alpha_1'(t)\,p_1 + \cdots + \alpha_n'(t)\,p_n$$

is a vector, as when the coefficients sum to zero the quantity is a vector, opposed to when the coefficients sum to one which implies the quantity is a point. All other derivatives are also vectors.

13.2 Bernstein Polynomials and Bezier Curves

An important property of de Casteljau's algorithm (at least for computational purposes) is the explicit form of the barycentric coordinates for the control points. To determine the coefficients, it is useful to follow the algorithm through completely for say four control points.

Given four control points p_0, p_1, p_2, p_3. The zeroth iteration of de Casteljau's algorithm produces the points

$$p_0^0(t) = p_0, \quad p_1^0(t) = p_1, \quad p_2^0(t) = p_2, \quad p_3^0(t) = p_3.$$

The first iteration of de Casteljau's algorithm then produces the points

$$\begin{aligned} p_0^1 &= (1-t)\,p_0 + t\,p_1, \\ p_1^1(t) &= (1-t)\,p_1 + t\,p_2, \\ p_2^1(t) &= (1-t)\,p_2 + t\,p_3. \end{aligned}$$

The second iteration of de Casteljau's algorithm then produces the points

$$\begin{aligned} p_0^2 &= (1-t)^2\,p_0 + 2t(1-t)\,p_1 + t^2\,p_2, \\ p_1^2 &= (1-t)^2\,p_1 + 2t(1-t)\,p_2 + t^2\,p_3. \end{aligned}$$

Finally, we obtain the curve

$$c(t) = (1-t)^3\,p_0 + 3t(1-t)^2\,p_1 + 3t^2(1-t)\,p_2 + t^3\,p_3.$$

Notice that the intermediate points obtained after each iteration have similar coefficients. In fact, the curves obtained after the first iteration are lines, and the curves obtained after the second iteration are parabolas (possibly degenerate). And lastly, the curve itself is a cubic curve.

To derive the general form of the coefficients, we use an inductive argument. We first write an nth degree curve as

$$c(t) = \beta_0^n(t)\,p_0 + \beta_1^n(t)\,p_1 + \cdots + \beta_n^n(t)\,p_n$$

where $\beta_i^n(t)$ are the basis functions for the nth degree curve. By the above computation, we have the coefficients for a "cubic curve" given by

$$\beta_i^3(t) = \binom{3}{i}\,t^i\,(1-t)^{3-i}.$$

The conjecture is that the coefficients of an nth degree curve ($n+1$ control points) are given by

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

where $\binom{n}{i}$ is the binomial coefficient,

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

The crucial observation in deriving the above general form is that the control points p_0, p_1, $\cdots$, p_{n-1} generate the $(n-1)$th degree $p_0^{n-1}(t)$ and the control points p_1, p_2, $\cdots$, p_n generate the $(n-1)$th degree curve $p_1^{n-1}(t)$. We then use the definition of $p_0^n(t) = (1-t)\,p_0^{n-1}(t) + p_1^{n-1}(t)$ and some algebra to derive the above formula by induction.

We have already shown by direct computation that the general formula is true for $n = 0, 1, 2, 3$. Expanding the sum

$$(1-t)\,p_0^{n-1}(t) + t\,p_1^{n-1}(t),$$

using the formula for $n-1$ degree curves, we have $\beta_0^n(t) = (1-t)^n$, $\beta_n^n(t) = t^n$, and

$$\beta_i^n(t) = \left(\binom{n-1}{i} + \binom{n-1}{i+1}\right)\,t^i\,(1-t)^{n-i} = \binom{n}{i}\,t^i\,(1-t)^{n-i},$$

by using Pascal's triangle interpretation of binomial coefficients.

The coefficients $\beta_i^n(t) = \binom{n}{i}\,t^i(1-t)^{n-i}$ are called the Bernstein polynomials. Notice for $0 \le t \le 1$, we have $0 \le \beta_i^n(t) \le 1$, and

$$\beta_0^n(t) + \beta_1^n(t) + \cdots + b_n^n(t) = 1.$$

This last fact is a direct consequence of noting that the Bernstein polynomials arise from expanding $(1-t+t)^n = 1^n$.

We note that the Bernstein polynomials posses the symmetry $\beta_i^n(t) = \beta_{n-i}^n(1-t)$, and as a result Bezier curves have a nice symmetry; if you reverse the order of the control points one obtains the same curve. Further, we note that $\beta_i^n(0) = 0$ for $i = 1, 2, \cdots, n$ and $\beta_0^n(0) = 1$, and $\beta_i^n(1) = 0$ for $i = 0, 1, \cdots, n-1$ and $\beta_n^n(1) = 1$, which implies that the curve passes through the points p_0 and p_n. To see a graphical depiction of the basis functions, and the properties mentioned above, see the figure below.

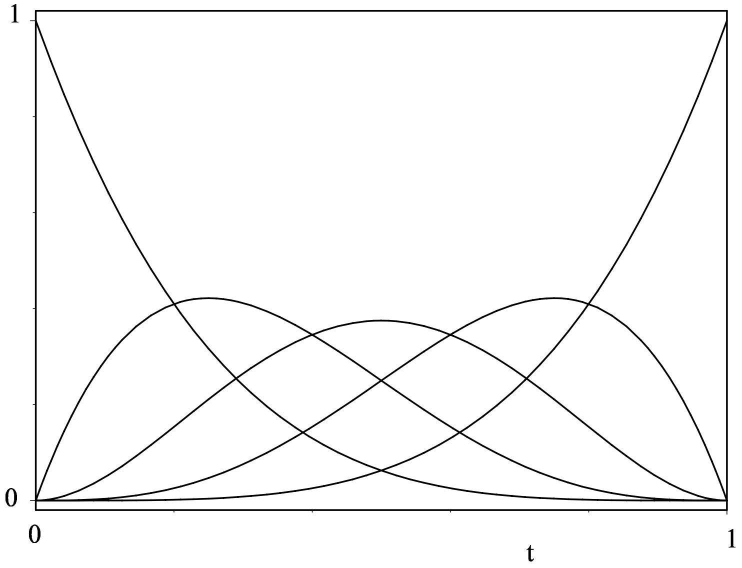

Figure 1: Basis functions $\beta_i^4(t)$

13.3 Derivatives of Bezier Curves

Given the control points p_0, p_1, $\cdots$, p_n, a Bezier curve with these control points is given by

$$B(t) = \beta_0^n(t)\, p_0 + \beta_1^n(t)\, p_1 + \cdots + \beta_n^n(t)\, p_n.$$

This definition of curves by barycentric coordinates is useful as long as the degree remains fixed. For the remainder of our discussion, we view the curve as defined by a set number of control and do not consider the addition or subtraction of a control point to the control polyline.

Formulas for differentiating a Bezier curve can be developed by examining the derivatives of the Bernstein polynomials

$$\beta_i^n(t) = \binom{n}{i}\, t^i\, (1-t)^{n-i}.$$

By direct computation, we have

$$\begin{aligned}\frac{d}{dt}\left(\beta_i^n(t)\right) &= \binom{n}{i}\left(i\,t^{i-1}\,(1-t)^{n-i} - (n-i)\,t^i\,(1-t)^{n-i-1}\right)\\ &= \frac{n!}{i!(n-i)!}i\,t^{i-1}\,(1-t)^{n-i} - \frac{n!}{i!(n-i)!}(n-i)\,t^i\,(1-t)^{n-i}\\ &= n\frac{(n-1)!}{(i-1)!(n-i)!},t^{i-1}\,(1-t)^{n-i} - n\frac{(n-1)!}{i!(n-i-1)!}\,t^i\,(1-t)^{n-i-1}\\ &= n\left(\beta_{i-1}^{n-1}(t) - \beta_i^{n-1}(t)\right).\end{aligned}$$

In the above, it is to be understood that $\beta_i^n(t) = 0$ if $i < 0$ and $\beta_i^n(t) = 0$ if $i > n$. Therefore, we have

$$\begin{aligned}B'(t) = &- n\beta_0^{n-1}(t)\,p_0 + n(\beta_0^{n-1}(t) - \beta_1^{n-1}(t))\,p_1 + \cdots\\ &+ n(\beta_{n-2}^{n-1}(t) - \beta_{n-1}^{n-1}(t))\,p_{n-1} + n\beta_{n-1}^{n-1}(t)\,p_n.\end{aligned}$$

Rearranging the terms, we have

$$B'(t) = n\beta_0^{n-1}\,(p_1 - p_0) + n\beta_1^{n-1}\,(p_2 - p_1) + \cdots + n\beta_{n-1}^{n-1}(t)\,(p_n - p_{n-1}).$$

Notice that the derivative is a Bezier curve with n control points equal to the vectors $n\,\Delta_i p = n(p_{i+1} - p_i)$. Likewise, we can define higher order derivatives. By applying the same type of rule.

The formulas for Bezier curves and derivatives of Bezier curves are greatly simplified by introducing sum notation. A Bezier curve is

$$B(t) = \sum_{i=0}^{n} \beta_i^n(t)\,p_i$$

and the derivative of a Bezier curve is

$$B'(t) = n\sum_{i=0}^{n-1} \beta_i^{n-1}(t)\,\Delta_i p$$

where $\Delta_i p = p_{i+1} - p_i$. The quantities $\Delta_i p$ is called the first difference of the control points. Higher order derivatives are obtained as generalizations of these formulas, see exercises.

13.4 Exercises

1. Complete the interactive exercises for Bezier curves and the Bernstein polynomials.

2. Given the control points $p_0 = [-1, 0]$, $p_1 = [1, 1]$, $p_2 = [2, 1]$, $p_3 = [1, 3]$.

   (a) Write the coordinate functions of the Bezier curve $B(t)$ with these control points.

   (b) Differentiate the coordinate functions and evaluate the derivative when $t = 0$, $t = 1/2$ and $t = 1$.

   (c) Apply the formula $B'(t) = \sum n\,\beta_i^{n-1}(t)\,\Delta_i p$ to compute the derivative show this yields the same answer as in (b).

3. Derive a formula for the second derivative of $B(t)$. Express this derivative as a Bezier curve on the second differences $\Delta_i^2 p = \Delta_{i+1}p - \Delta_i p = p_{i+2} - 2p_{i+1} + p_i$.

4. Given the control points $p_0 = [1, 1]$, $p_1 = [2, 3]$, $p_2 = [3, 0]$, $p_3 = [1, 2]$.
   (a) Write the coordinate functions of the Bezier curve through these control points.
   (b) Compute the second derivative (by differentiating the expression in (a) twice).
   (c) Compute the second differences, and write the second derivative in terms of the second differences.
   (d) Verify the two expressions of the second derivative are equal.

5. Derive a formula for the third, fourth, fifth, et cetera derivatives of a Bezier curve in terms of higher order differences.

6. Given the figure below, draw the first, second and third differences as vectors. Where should each difference be naturally based? [$\Delta_0 p = p_1 - p_0$ is naturally based at p_0. Why?]

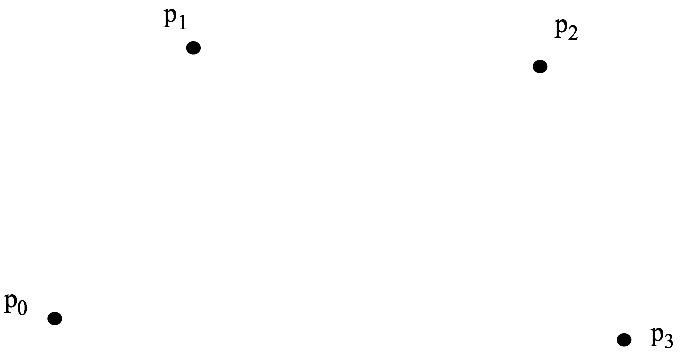

Figure 2: Control Points