Bézier curves and surfaces

Michael S. Floater

August 20, 2012

These notes continue a study of Bézier curves and introduce tensor-product Bézier surfaces.

1 Bézier curves on general domains

It is often useful to define a Bézier curve over a parameter domain other than $[0, 1]$. We can define a Bézier curve $\mathbf{p} : [a, b] \to \mathbb{R}^d$ for an arbitrary interval $[a, b]$ by the formula

$$\mathbf{p}(t) = \sum_{i=0}^{n} \mathbf{c}_i B_i^n(u), \qquad \mathbf{c}_i \in \mathbb{R}^d, \tag{1}$$

where

$$u = \frac{t-a}{b-a}, \tag{2}$$

with control points $\mathbf{c}_i \in \mathbb{R}^d$ and B_i^n is, as before, the Bernstein polynomial

$$B_i^n(x) = \binom{n}{i} x^i (1-x)^{n-i}.$$

Many of the properties of this more general Bézier curve are similar to those of the canonical one in which $a = 0$, $b = 1$. For example, the endpoint property is now

$$\mathbf{p}(a) = \mathbf{c}_0, \qquad \mathbf{p}(b) = \mathbf{c}_n.$$

The de Casteljau algorithm is similar to before, but is now, with $\mathbf{c}_i^0 = \mathbf{c}_i$,

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

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

Differentation and integration can be dealt with by the chain rule. Since

$$\frac{d}{dt} = \frac{1}{L}\frac{d}{du},$$

where

$$L = b - a,$$

differentiating (1) results in the formula

$$\mathbf{p}'(t) = \frac{n}{L}\sum_{i=0}^{n-1} \Delta\mathbf{c}_i B_i^{n-1}(u).$$

Or, treating the de Casteljau point $\mathbf{c}_i^r$ as a function of t,

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

Similarly, the r-th derivative is

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

with the special cases

$$\mathbf{p}^{(r)}(a) = \frac{n!}{(n-r)!L^r}\Delta^r\mathbf{c}_0 \quad \text{and} \quad \mathbf{p}^{(r)}(b) = \frac{n!}{(n-r)!L^r}\Delta^r\mathbf{c}_{n-r}, \tag{3}$$

or in terms of the de Casteljau points,

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

The integral formula becomes

$$\int_a^b \mathbf{p}(t)\,dt = L\frac{\mathbf{c}_0 + \mathbf{c}_1 + \cdots + \mathbf{c}_n}{(n+1)}.$$

2 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.

Suppose we restrict the Bézier curve $\mathbf{p}$ in (1) to the parameter interval $[a, b]$ and we want to join to it a second Bézier curve, $\mathbf{q}$, defined on the adjacent interval $[b, c]$, i.e.,

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

where

$$v = \frac{t-b}{c-b}.$$

We would like the composite curve

$$\mathbf{s}(t) := \begin{cases} \mathbf{p}(t), & a \leq t < b; \\ \mathbf{q}(t), & b \leq t < c; \end{cases}$$

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

$$\mathbf{d}_0 = \mathbf{c}_n. \tag{4}$$

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 $t = b$, written C^k, if

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

This is equivalent to saying that $\mathbf{s}$ is C^k at $t = b$. From equation (3) this condition is equivalent to

$$\frac{\Delta^r \mathbf{d}_0}{(c-b)^r} = \frac{\Delta^r \mathbf{c}_{n-r}}{(b-a)^r}, \qquad r = 0, 1, \ldots, k. \tag{5}$$

For example, the condition for C^1 smoothness is (4) combined with

$$\frac{\mathbf{d}_1 - \mathbf{d}_0}{c - b} = \frac{\mathbf{c}_n - \mathbf{c}_{n-1}}{b - a}.$$

This equation and (4) uniquely determine the two points $\mathbf{d}_0$ and $\mathbf{d}_1$ from the points $\mathbf{c}_n$ and $\mathbf{c}_{n-1}$,

$$\mathbf{d}_0 = \mathbf{c}_n, \qquad \mathbf{d}_1 = (1-\mu)\mathbf{c}_{n-1} + \mu\mathbf{c}_n, \tag{6}$$

where

$$\mu = \frac{c - a}{b - a}.$$

The coefficient μ is easy to remember because it is the 'coordinate' of c with respect to the interval $[a, b]$, analogous to u being the coordinate of t in (2). Notice, however, that while both coefficients u and $1 - u$ are non-negative in (2) for $t \in [a, b]$, the coefficient $(1 - \mu)$ in (6) is negative because $c > b$.

After a lengthier calculation using $r = 2$ in (5) it can be shown that the condition for C^2 continuity can be expressed as the two equations in (6) plus the equation

$$\mathbf{d}_2 = (1-\mu)^2\mathbf{c}_{n-2} + 2(1-\mu)\mu\mathbf{c}_{n-1} + \mu^2\mathbf{c}_n.$$

We will derive this equation in a different way later.

3 Tensor-product spaces

If we take the tensor-product of the two vector spaces of polynomials π_m and π_n, we obtain a new vector space

$$\pi_{m,n} = \pi_m \otimes \pi_n.$$

This space comprises all polynomials of two variables whose degree in the first variable is at most m and in the second at most n. A basis for these bivariate polynomials can be constructed by taking products of bases for π_m and π_n. For example, the monomial bases for π_2 and π_1 are $\{1, x, x^2\}$ and $\{1, y\}$ respectively, so that a basis for $\pi_{2,1}$ is

$$\{1, x, x^2, y, xy, x^2y\}.$$

Thus a polynomial in $\pi_{m,n}$ can be expressed uniquely in the form

$$p(x,y) = \sum_{i=0}^{m} \sum_{j=0}^{n} a_{i,j} x^i y^j.$$

An alternative representation is to use products of Bernstein polynomials,

$$p(x,y) = \sum_{i=0}^{m} \sum_{j=0}^{n} c_{i,j} B_i^m(x) B_j^n(y).$$

4 Tensor-product Bézier surfaces

We define a tensor-product Bézier surface in $\mathbb{R}^d$ as a parametric polynomial surface $\mathbf{p} : [a_1, b_1] \times [a_2, b_2] \to \mathbb{R}^d$ by the formula

$$\mathbf{p}(s,t) = \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{c}_{i,j} B_i^m(u) B_j^n(v), \qquad \mathbf{c}_{i,j} \in \mathbb{R}^d, \tag{7}$$

where

$$u = \frac{s - a_1}{b_1 - a_1}, \qquad v = \frac{t - a_2}{b_2 - a_2}. \tag{8}$$

In practice, the Euclidean space $\mathbb{R}^d$ will usually be $\mathbb{R}^3$. The *control net* of $\mathbf{p}$ is the network of points and line segments consisting of the control points $\mathbf{c}_{i,j}$ and all line segments of the form $[\mathbf{c}_{i,j}, \mathbf{c}_{i+1,j}]$ and $[\mathbf{c}_{i,j}, \mathbf{c}_{i,j+1}]$. Figure 1 shows a biquadratic surface, where $m = n = 2$, with its control net.

Bézier surfaces have various properties analogous to Bézier curves. On each of the four boundaries of the parameter domain $[a_1, b_1] \times [a_2, b_2]$ the surface $\mathbf{p}$ is a Bézier curve whose control polygon is one of the four boundaries of the control net of $\mathbf{p}$. For example,

$$\mathbf{p}(s, a_2) = \sum_{i=0}^{m} \mathbf{c}_{i,0} B_i^m(u).$$

At the corners of the parameter domain, the surface equals one of the corner control points, for example

$$\mathbf{p}(a_1, a_2) = \mathbf{c}_{0,0}.$$

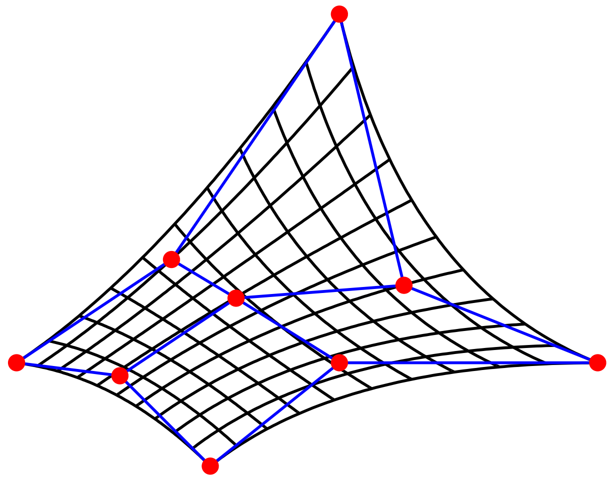

Figure 1: A biquadratic Bézier surface

Since the tensor-product Bernstein polynomials

$$B_{i,j} = B_i^m \otimes B_j^n,$$

sum to one:

$$\sum_{i=0}^{m}\sum_{j=0}^{n} B_{i,j}(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} B_i^m(u)B_j^n(v) = \sum_{i=0}^{m} B_i^m(u)\sum_{j=0}^{n} B_j^n(v) = 1,$$

every point $\mathbf{p}(s,t)$ is an affine combination of the control points $\mathbf{c}_{i,j}$ and so Bézier surfaces are affinely invariant. Since the $B_{i,j}$ are also non-negative in $[0,1]$, the surface $\mathbf{p}$ also has the convex hull and bounding box poperties.

5 The de Casteljau algorithm

Given a parameter pair $(s,t) \in [a_1,b_1] \times [a_2,b_2]$, one way of computing the point $\mathbf{p}(s,t)$ is to evaluate the Bernstein polynomials B_i^m and B_j^n at u and v respectively and then apply the formula (7). Alternatively one can apply de Casteljau's algorithm to the rows of points in the control net, in each of the two directions. We apply m steps of the algorithm with respect to u and n with respect to v. The last point generated will be the point $\mathbf{p}(s,t)$, no matter how we order these $m+n$ steps. Consider an example. Let $m=2$, $n=3$, and

$$\begin{bmatrix} \mathbf{c}_{00} & \mathbf{c}_{01} & \mathbf{c}_{02} & \mathbf{c}_{03} \\ \mathbf{c}_{10} & \mathbf{c}_{11} & \mathbf{c}_{12} & \mathbf{c}_{13} \\ \mathbf{c}_{20} & \mathbf{c}_{21} & \mathbf{c}_{22} & \mathbf{c}_{23} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 6 \\ 18 & 2 & 0 & 8 \\ 4 & 0 & 4 & 18 \end{bmatrix},$$

and suppose that $u = 1/2$ and $v = 2/3$ in (8). Applying de Castlejau's algorithm first with respect to u gives

$$\begin{bmatrix} 0 & 0 & 0 & 6 \\ 18 & 2 & 0 & 8 \\ 4 & 0 & 4 & 18 \end{bmatrix} \to \begin{bmatrix} 9 & 1 & 0 & 7 \\ 11 & 1 & 2 & 13 \end{bmatrix} \to \begin{bmatrix} 10 & 1 & 1 & 10 \end{bmatrix}.$$

Then, applying the algorithm with respect to v gives

$$\begin{bmatrix} 10 & 1 & 1 & 10 \end{bmatrix} \to \begin{bmatrix} 4 & 1 & 7 \end{bmatrix} \to \begin{bmatrix} 2 & 5 \end{bmatrix} \to \begin{bmatrix} 4 \end{bmatrix},$$

so that $\mathbf{p}(s,t) = 4$. Alternatively, we could apply the algorithm first with respect to v, giving

$$\begin{bmatrix} 0 & 0 & 0 & 6 \\ 18 & 2 & 0 & 8 \\ 4 & 0 & 4 & 18 \end{bmatrix} \to \begin{bmatrix} 0 & 0 & 4 \\ 22/3 & 1/3 & 16/3 \\ 1/3 & 8/3 & 40/3 \end{bmatrix} \to \begin{bmatrix} 0 & 8/3 \\ 26/9 & 34/9 \\ 20/9 & 88/9 \end{bmatrix} \to \begin{bmatrix} 16/9 \\ 94/27 \\ 196/27 \end{bmatrix},$$

and then with respect to u,

$$\begin{bmatrix} 16/9 \\ 94/27 \\ 196/27 \end{bmatrix} \to \begin{bmatrix} 71/27 \\ 145/27 \end{bmatrix} \to \begin{bmatrix} 4 \end{bmatrix},$$

yielding the same answer.