15 Parametric Curves And Surfaces

15.1 Parametric Curves

Designing Curves

- We don't want only polygons.
- Curves are used for design. Users require a simple set of controls to allow them to edit and design curves easily.
- Curves should have infinite resolution, so we can zoom in and still see a smooth curve.
- We want to have a compact representation.

Parametric functions are of the form $x(t) = f(t)$ and $y(t) = g(t)$ in two dimensions. This can be extended for arbitrary dimensions. They can be used to model curves that are *not* functions of any axis in the plane.

Curves can be defined as polynomials, for example $x(t) = 5t^{10} + 4t^9 + 3t^8 + ...$ However, coefficients are not intuitive editing parameters, and these curves are difficult to control. Hence, we will consider more intuitive parameterizations.

15.2 Bézier curves

We can define a set of curves called Bézier curves by a procedure called the de Casteljau algorithm. Given a sequence of control points $\bar{p}_k$, de Casteljau evaluation provides a construction of smooth parametric curves. Evaluation proceeds by repeatedly defining new, smaller point sequences until we have a single point at the value for t for which we are evaluating the curve.

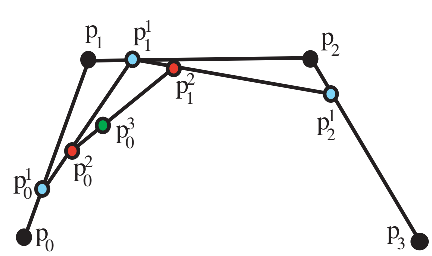

Figure 1: de Casteljau evaluation for $t = 0.25$.

$$\bar{p}_0^1(t) = (1-t)\bar{p}_0 + t\bar{p}_1 \tag{1}$$

$$\bar{p}_1^1(t) = (1-t)\bar{p}_1 + t\bar{p}_2 \quad (2)$$
$$\bar{p}_2^1(t) = (1-t)\bar{p}_2 + t\bar{p}_3 \quad (3)$$

$$\bar{p}_0^2(t) = (1-t)\bar{p}_0^1(t) + t\bar{p}_1^1(t) \quad (4)$$
$$= (1-t)^2\bar{p}_0 + 2t(1-t)\bar{p}_1 + t^2\bar{p}_2 \quad (5)$$
$$\bar{p}_1^2(t) = (1-t)\bar{p}_1^1(t) + t\bar{p}_2^1(t) \quad (6)$$
$$= (1-t)^2\bar{p}_1 + 2t(1-t)\bar{p}_2 + t^2\bar{p}_3 \quad (7)$$

$$\bar{p}_0^3(t) = (1-t)\bar{p}_0^2(t) + t\bar{p}_1^2(t) \quad (8)$$
$$= (1-t)^3\bar{p}_0 + 3(1-t)^2t\bar{p}_1 + 3(1-t)t^2\bar{p}_2 + t^3\bar{p}_3 \quad (9)$$

The resulting curve $\bar{p}_0^3$ is the cubic Bézier defined by the four control points. The curves $\bar{p}_0^2$ and $\bar{p}_1^2$ are quadratic Bézier curves, each defined by three control points. For all Bézier curves, we keep t in the range $[0...1]$.

15.3 Control Point Coefficients

Given a sequence of points $\bar{p}_0$, $\bar{p}_1$, ..., $\bar{p}_n$, we can directly evaluate the coefficient of each point. For a class of curves known as Bézier curves, the coefficients are defined by the Bernstein polynomials:

$$\bar{p}_0^n(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i \bar{p}_i = \sum_{i=0}^{n} B_i^n(t)\bar{p}_i \quad (10)$$

where

$$B_i^n(t) = \binom{n}{i}(1-t)^{n-i}t^i \quad (11)$$

are called the *Bernstein basis functions.*

For example, cubic Bézier curves have the following coefficients:

$$B_0^3(t) = (1-t)^3 \quad (12)$$
$$B_1^3(t) = 3(1-t)^2t \quad (13)$$
$$B_2^3(t) = 3(1-t)t^2 \quad (14)$$
$$B_3^3(t) = t^3 \quad (15)$$

Figure 2 is an illustration of the cubic Bernstein basis functions.

Similarly, we define basis functions for a linear curve, which is equivalent to the interpolation $\bar{p}(t) = \bar{p}_0(1-t) + \bar{p}_1 t$. These are shown in Figure 3.

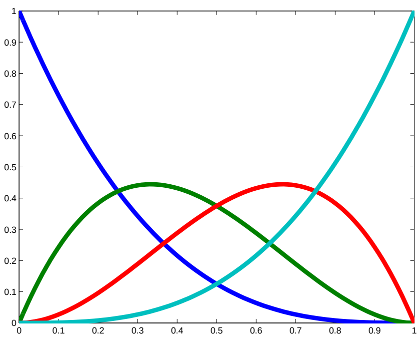

Figure 2: Degree three basis functions for Bézier curves. $B_0^3(t)$ (dark blue), $B_1^3(t)$ (green), $B_2^3(t)$ (red), and $B_3^3(t)$ (light blue).

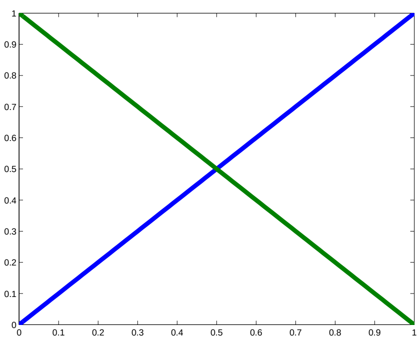

Figure 3: Degree one basis functions for Bézier curves. $B_0^1(t)$ (green) and $B_1^1(t)$ (blue).

15.4 Bézier Curve Properties

- **Convexity of the basis functions.** For all values of $t \in [0...1]$, the basis functions sum to 1:

$$\sum_{i=0}^{n} B_i^n(t) = 1 \tag{16}$$

In the cubic case, this can be shown as follows:

$$((1-t)+t)^3 = (1-t)^3 + 3(1-t)^2 t + 3(1-t)t^2 + t^3 = 1 \tag{17}$$

In the general case, we have:

$$((1-t)+t)^n = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i = 1 \tag{18}$$

Similarly, it is easy to show that the basis functions are always non-negative: $B_i^n(t) \geq 0$.

- **Affine Invariance**

  What happens if we apply an affine transformation to a Bézier curve?

  Let $\bar{c}(t) = \sum_{j=0}^{n} \bar{p}_j B_j^n(t)$, and let $F(\bar{p}) = \mathbf{A}\bar{p} + \vec{d}$ be an affine transformation. Then we have the following:

$$\begin{aligned} F(\bar{c}(t)) &= \mathbf{A}\bar{c}(t) + \vec{d} && (19)\\ &= \mathbf{A}\left(\sum \bar{p}_j B_j^n(t)\right) + \vec{d} && (20)\\ &= \sum \left(\mathbf{A}\bar{p}_j\right) B_j^n(t) + \vec{d} && (21)\\ &= \sum \left(\mathbf{A}\bar{p}_j + \vec{d}\right) B_j^n(t) && (22)\\ &= \sum B_j^n(t)\bar{q}_j && (23) \end{aligned}$$

  $\bar{q}_j = \mathbf{A}\bar{p}_j + \vec{d}$ denotes the transformed points. This illustrates that the transformed curve we get is the same as what we get by transforming the control points. (The third statement follows from the fact that $\sum_{j=0}^{n} B_j^n(t) = 1$.)

- **Convex Hull Property**

  Since $B_i^N(t) \geq 0$, $\bar{p}(t)$ is a convex combination of the control points. Thus, Bézier curves *always* lie within the convex hull of the control points.

- **Linear Precision**

  When the control points lie on a straight line, then the corresponding Bézier curve will also be a straight line. This follows from the convex hull property.

- **Variation Diminishing**

  No straight line can have more intersections with the Bézier curve than it has with the control polygon. (The control polygon is defined as the line segments $\overline{p_j p_{j+1}}$.)

- **Derivative Evaluation**

  Letting $\bar{c}(t) = \sum_{j=0}^{N} \bar{p}_j B_j^N(t)$, we want to find the following:

$$\bar{c}'(t) = \frac{d\bar{c}(t)}{dt} = \left(\frac{dx(t)}{dt}, \frac{dy(t)}{dt}\right) \tag{24}$$

  Letting $\vec{d}_j = \bar{p}_{j+1} - \bar{p}_j$, it can be shown that:

$$\tau(t) = \frac{d}{dt}\bar{c}(t) = \frac{d}{dt}\sum_{j=0}^{N} \bar{p}_j B_j^N(t) = N\sum_{j=0}^{N-1} \vec{d} B_j^{N-1}(t) \tag{25}$$

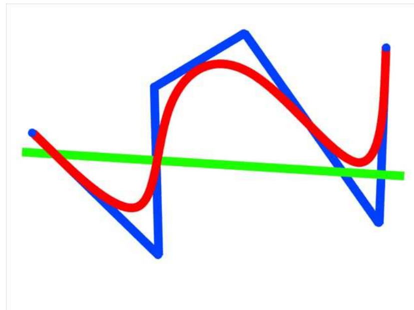

Figure 4: The line (green) will always intersect the curve less often than or as many times as the control polygon.

Thus, $\bar{c}(t)$ is a convex sum of the points $\bar{p}_j$ and is a point itself. $\tau(t)$ is a convex sum of vectors and is a vector.

Example: What is $\tau(0)$ when $N = 3$, given $(\bar{p}_0, \bar{p}_1, \bar{p}_2, \bar{p}_3)$?

Since $B_j^3(0) = 0$ for all $j \neq 0$ and $B_0^3(0) = 1$,

$$\tau(0) = N \sum \vec{d}_j B_j^{N-1}(t) = 3\vec{d}_j = 3(\bar{p}_1 - \bar{p}_0) \quad (26)$$

Therefore, the tangent vector at the endpoint is parallel to the vector from the endpoint to the adjacent point.

- **Global vs. Local Control**

  Bézier curves that approximate a long sequence of points produce high-degree polynomials. They have global basis functions; that is, modifying any point changes the entire curve. This results in curves that can be hard to control.

15.5 Rendering Parametric Curves

Given a parameter range $t \in [0, 1]$, sample t by some partition Δt, and draw a line connecting each pair of adjacent samples.

- This is an expensive algorithm.
- This does not adapt to regions of a curve that do not require as many samples.
- It's difficult to determine a sufficient number of samples to render the curve such that it appears smooth.

There are faster algorithms based on adaptive refinement and subdivision.

15.6 Bézier Surfaces

Cubic Bézier patches are the most common parametric surfaces used for modeling. They are of the following form:

$$\mathbf{s}(\alpha, \beta) = \sum_{k=0}^{3} \sum_{j=0}^{3} B_j^3(\alpha) B_k^3(\beta) \bar{p}_{j,k} = \sum_k B_k^3(\beta) \bar{p}_k(\alpha) \tag{27}$$

where each $\bar{p}_k(\alpha)$ is a Bézier curve:

$$\bar{p}_k(\alpha) = \sum_j B_j^3(\alpha) \bar{p}_{j,k} \tag{28}$$

Rather than considering only four points as in a cubic Bézier curve, consider 16 control points arranged as a 4 x 4 grid:

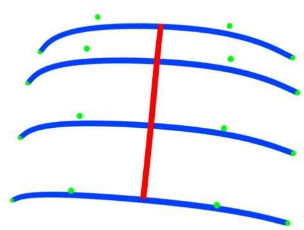

Figure 5: Evaluation of any point can be done by evaluating curves along one direction (blue), and evaluating a curve among points on these curves with corresponding parameter values.

For any given α, generate four points on curves and then approximate them with a Bézier curve along β.

$$\bar{p}_k(\alpha) = \sum_{j=0}^{3} B_j^3(\alpha) \bar{p}_{jk} \tag{29}$$

To connect multiple patches, we align adjacent control points. to ensure C^1 continuity, we also have to enforce colinearity of the neighboring points.
The surface can also be written in terms of 2D basis functions $B_{j,k}^3(\alpha, \beta) = B_j^3(\alpha) B_k^3(\beta)$:

$$\mathbf{s}(\alpha, \beta) = \sum_{k=0}^{3} \sum_{j=0}^{3} B_{j,k}^3(\alpha, \beta) \bar{p}_{j,k} \tag{30}$$