BÉZIER CURVES

Bézier Curves are parametric polynomials used to make smooth lines and shapes. Being parametric means that instead of representing the curve as a function of spatial coordinates (x, y, z), we use a parameter that relates these values (t on [0,1]). The general form of a bezier polynomial is the expression:

$$B_n(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i P_i$$

Where $\binom{n}{i}$ is the binomial coefficient function $\dfrac{n!}{(n-i)!i!}$

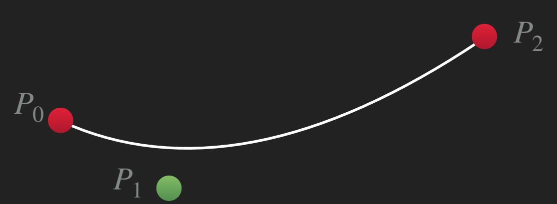
- n Is the degree of the polynomial and equal to (number of points - 1)
- i Is just the summation index
- P_i Is one of the points describing the curve. Generally this is in 2 or 3 dimensions (x, y) or (x, y, z).
- P_0 Is the starting point of the curve
- P_n Is the end point of the curve

All other points are known as control points and do not necessarily lie on the curve

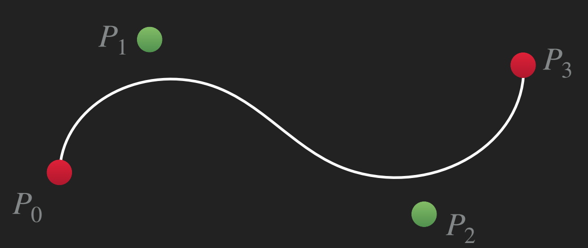
For most computer graphics purposes the we limit ourselves to 3rd degree polynomials because anything more than that is more computationally expensive than its worth. The 3 main bezier polynomials are as follows:

Linear $B_1(t) = (1-t)P_0 + tP_1$

Quadratic $B_2(t) = (1-t)^2P_0 + 2t(1-t)P_1 + t^2P_2$



Cubic $B_3(t) = (1-t)^3P_0 + 3t(1-t)^2P_1 + 3t^2(1-t)P_2 + t^3P_3$



Lots of methods are used for manipulating and creating bezier curves for particular applications. The forms presented are definitely the best for evaluating a point that lies on the curve. Try inputing [0, 1, and 1/2] for t into the functions and see what values you get. For splitting up curves into smaller segments, it's best to use the matrix form of the polynomial.

Matrix Form

Quadratic

$$\begin{aligned} B_2(t) &= (1-t)^2P_0 + 2t(1-t)P_1 + t^2P_2 \\ &= (1-2t+t^2)P_0 + (2t-2t^2)P_1 + t^2P_2 \\ &= \begin{bmatrix}1 & t & t^2\end{bmatrix}\begin{bmatrix}1\\-2\\1\end{bmatrix}P_0 + \begin{bmatrix}1 & t & t^2\end{bmatrix}\begin{bmatrix}0\\2\\-2\end{bmatrix}P_1 + \begin{bmatrix}1 & t & t^2\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix}P_2 \\ &= \begin{bmatrix}1 & t & t^2\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\-2 & 2 & 0\\1 & -2 & 1\end{bmatrix}\begin{bmatrix}P_0\\P_1\\P_2\end{bmatrix} \end{aligned}$$

Cubic

A similar process leads to the matrix form of the cubic curve

$$B_3(t) = \begin{bmatrix}1 & t & t^2 & t^3\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\-3 & 3 & 0 & 0\\3 & -6 & 3 & 0\\-1 & 3 & -3 & 1\end{bmatrix}\begin{bmatrix}P_0\\P_1\\P_2\\P_3\end{bmatrix}$$

Copy A Segment Of A Curve

Remember that t defined to range from 0 to 1 inclusively. This means if we want to find the segment of the curve defined from $t = s$ to $t = f$ where $f > s$, we just need to replace t with an expression that restricts our domain to the new smaller segment

$$t \to s + (f-s)t$$

Start value (basically the offset from 0)

The difference between the starting and end values

With this expression we can still keep t on the domain [0,1]. The curve segment is just restricted between s and f

$$B_2(t) = \begin{bmatrix}1 & (s+(f-s)t) & (s+(f-s)t)^2\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\-2 & 2 & 0\\1 & -2 & 1\end{bmatrix}\begin{bmatrix}P_0\\P_1\\P_2\end{bmatrix}$$

$$B_2(t) = \begin{bmatrix}1 & t & t^2\end{bmatrix}\begin{bmatrix}1 & s & s^2\\0 & f-s & 2s(f-s)\\0 & 0 & (f-s)^2\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\-2 & 2 & 0\\1 & -2 & 1\end{bmatrix}\begin{bmatrix}P_0\\P_1\\P_2\end{bmatrix}$$

Restricting Matrix **R** Coefficient Matrix **C**

By multiplying our expression by $\mathbf{CC}^{-1}$ we can pullout a bit of useful information

$$B_2(t) = \begin{bmatrix}1 & t & t^2\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\-2 & 2 & 0\\1 & -2 & 1\end{bmatrix}\underbrace{\begin{bmatrix}1 & 0 & 0\\1 & 1/2 & 0\\1 & 1 & 1\end{bmatrix}\begin{bmatrix}1 & s & s^2\\0 & f-s & 2s(f-s)\\0 & 0 & (f-s)^2\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\-2 & 2 & 0\\1 & -2 & 1\end{bmatrix}}\begin{bmatrix}P_0\\P_1\\P_2\end{bmatrix}$$

This portion coincidently represents a transformation that when applied to the points of the original curve returns the points defining the segment between s and f.

So the procedure here is

1. Create the coefficient matrix for the bezier curve and the restricting matrix for the segment.
2. Multiply the inverse of the coefficient matrix by the restricting matrix, then multiply that by the original coefficient matrix. $\mathbf{T} = \mathbf{C}^{-1}\mathbf{RC}$
3. Apply the transform to the original curves points

Its pretty weird because you basically can just throw away the first part of the expression and only use the transformation matrix and the points from the original curve

Subdividing Curves

Subdivision can be performed with respect to one of two metrics. Arc length scaled or parameter scaled. Arc length is the harder of the two being that it requires algorithms for computing the value of an elliptic integral. As such I will focus on subdividing the curve based on the parameter value t.

The procedure is to create n smooth segments from a single curve. Create n intervals from $\left[\dfrac{(i-1)}{n}, \dfrac{i}{n}\right]$ where i is an iteration value going from 1 to n.