The Math Behind Bezier Cubic Splines

Don Lancaster
Synergetics, Box 809, Thatcher, AZ 85552

http://www.tinaja.com
don@tinaja.com
(928) 428-4073

Bezier **Cubic Splines** are an excellent and preferred method to draw the smooth continuous curves often found in typography, **CAD/CAM**, and graphics in general.

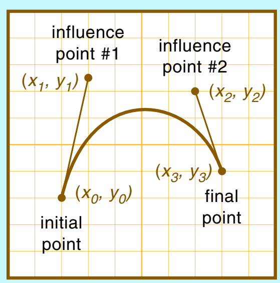
Among their many advantages is a **very sparse data set** allowing a mere **eight** values (or four **x,y** points) to **completely** define a full and carefully controlled and device independent curve. Many tutorials and examples are now present in our **Cubic Spline** Library. A brief and useful intro **appears here**.

While the underlying math behind Bezier cubic splines is amazingly simple, its derivation seems to be quite hard to find. So, what I thought we'd do here is find out exactly **where the funny numbers relating cubic spline power coefficients and control points come from.**

Let us once again begin by excerpting some key **Bezier Cubic Spline properties** from our **HACK62.PDF** tutorial...

Here is a cubic spline shown in its **graph space**...

The **first** influence point sets the direction and the enthusiasm that the spline **leaves** the **initial** point on the curve.



The **second** influence point sets the direction and the enthuasiasm that the spline **enters** the **final** point on the curve.

Here is how a cubic spline appears in its **equation space**...

$$x = At^3 + Bt^2 + Ct + D$$
$$y = Et^3 + Ft^2 + Gt + H$$

t (for time) always goes from **zero** at the **initial** point to a **one** at the **final** point.

This is a faster "cube free" form of the **equation space** math…

$$x = (((At) + B)t + C)t + D$$
$$y = (((Et) + F)t + G)t + H$$

How to get from **graph space** to **equation space**…

$A = x_3 - 3x_2 + 3x_1 - x_0$ $\quad$ $E = y_3 - 3y_2 + 3y_1 - y_0$
$B = 3x_2 - 6x_1 + 3x_0$ $\quad$ $F = 3y_2 - 6y_1 + 3y_0$
$C = 3x_1 - 3x_0$ $\quad$ $G = 3y_1 - 3y_0$
$D = x_0$ $\quad$ $H = y_0$

How to get from **equation space** to **graph space**…

$x_0 = D$ $\quad$ $y_0 = H$
$x_1 = D + C/3$ $\quad$ $y_1 = H + G/3$
$x_2 = D + 2C/3 + B/3$ $\quad$ $y_2 = H + 2G/3 + F/3$
$x_3 = D + C + B + A$ $\quad$ $y_3 = H + G + F + E$

You can easily take these numbers on faith and prove that they work. But to truly understand cubic splines, we need to take a closer look to see exactly **where** all those strange "threes" and "sixes" really come from.

In general, we can relate **any** four **x** control point variables to **any** four **A-D** cubic power coefficients. And similarly for **y**. The trick is to find a **useful** relationship between the two. And that is what Bezier Cubic Splines are fundamentally all about. And where some really sneaky math comes in.

It turns out there is a special class of math operators that are known as **Basis Functions**. Some of these are also called **Bernstein Polynomials**…

A Bernstein Polynomial of order three is exactly what you need to relate Bezier control points to cubic spline coefficients.

So, how do we create a Basis Function of order three? Start off with…

$$\mathbf{1 = 1}$$

… and then make this rather bizarre transformation to it…

$$\mathbf{(1 - t) + t = 1}$$

You often might restrict **t** (or "time") to values from **0** to **1**. To create most **any** group of Bernstein Polynomial Basis Functions, **you simply raise both sides of this equation to a desired power**. And then separate terms of interest.

Our crucial cubic Bernstein polynomial is...

Thus...

$$((1 - t) + t)^3 = 1$$

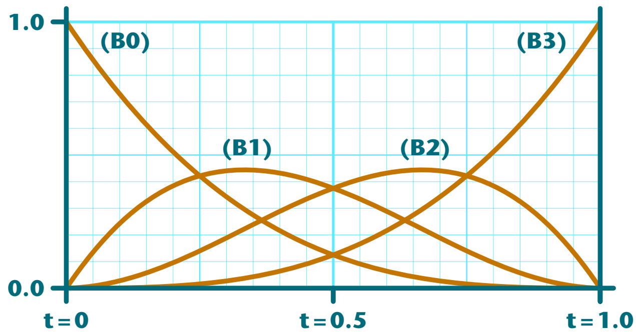
Expands to...

$$(1 - t)^3 + 3t(1-t)^2 + 3t^2(1 - t) + t^3 = 1$$

Whose terms can be redefined as...

$$B0(t) + B1(t) + B2(t) + B3(t) = 1$$

It is **very** interesting to plot these cubic basis functions...



For any given **t**, the four basis functions will **exactly** sum to give the **x** value for that **t**. Which leads to this crucially important Bezier Cubic Spline equation...

$$x(t) = x_0B0(t) + x_1B1(t) + x_2B2(t) + x_3B3(t)$$

... and the corresponding **y(t)** equivalent ...

$$y(t) = y_0B0(t) + y_1B1(t) + y_2B2(t) + y_3B3(t)$$

We can immediately see that initial point x_0 sets the **start** of the curve and that final point x_3 sets the **end** point for us. And that x_1 will have its strongest (but not total) influence **exactly** at $t = 1/3$. Similarly, x_2 will be strongest at $t = 2/3$.

And the magic derivation is...

The **x** equation deriving the underlying Cubic Spline math is simply...

$$x_0B0(t) + x_1B1(t) + x_2B2(t) + x_3B3(t) \quad = \quad At^3 + Bt^2 + Ct + D$$

Along with its **y** equivalent...

$$y_0B0(t) + y_1B1(t) + y_2B2(t) + y_3B3(t) \quad = \quad Et^3 + Ft^2 + Gt + H$$

Since the powers of **t** have to vary over a 0 to 1 region, **these two expressions can be equal only if the constant coefficients for each power of t match....**

$$x_3B3(t) = x_3t^3 = x_3t^3$$

$$x_2B2(t) = x_2(3t^2(1\text{-}t)) = -3x_2t^3 + 3x_2t^2$$

$$x_1B1(t) = x_1(3t(1\text{-}t)^2) = 3x_1t^3 - 6x_1t^2 + 3x_1t$$

$$x_0B0(t) = x_0(1\text{-}t)^3 = -x_0t^3 + 3x_0t^2 - 3x_0t + x_0$$

Now, "think vertically" and regroup to get...

$$A = x_3 - 3x_2 + 3x_1 - x_0$$

$$B = 3x_2 - 6x_1 + 3x_0$$

$$C = 3x_1 - 3x_0$$

$$D = x_0$$

And similarly for **y**...

$$E = y_3 - 3y_2 + 3y_1 - y_0$$

$$F = 3y_2 - 6y_1 + 3y_0$$

$$G = 3y_1 - 3y_0$$

$$H = y_0$$

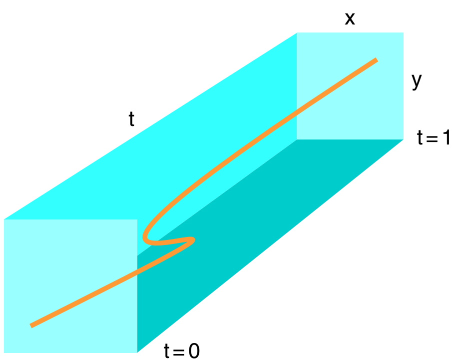
We've already seen that **x_0,y_0** sets the start of the curve and **x_3,y_3** the end. We've also seen that influence point **x_1,y_1** sets the **enthusiasm** or **tension** that peaks at **t = 1/3**. And that **x_2,y_2** sets the **enthusiasm** or **tension** that peaks at **t = 2/3**.

There are several ways of getting at the initial and final slopes of the final curve. By definition **C** will be the initial slope of the **x** versus **t** curve and will also equal **$3x_1 - 3x_0$**. On our **x** versus **y** curve, the slope will be **$(y_1 - y_0)/(x_1 - x_0)$**.

Thus, the **x_1,y_1** influence point will set the **initial** slope of the **x-y** cubic spline curve. And the **x_2,y_2** influence point will similarly set the **final** slope.

The "Snake in the Box"

It is important to remember that Bezier cubic splines **independently** relate **x** and **y** to a new **parametric** variable **t**. Which varies from **0** to **1** from the beginning to the end of the curve. Thus, your final two dimensional **x-y** plot is really one view of a three dimensional **x-y-t** plot. One good way to visualize this is to think of your parametric curve as a **snake in a box**...



You look into the **end** of the box to see how **x** varies with **y**. And into the **top** or **side** of the box to see the variation versus **t**.

In general, the relationship between **x** and **t** will be non-obvious, with **t** changing **faster** along the "more bent" portions of the curve. In addition, finding **t(x)** given **x(t)** is non trivial. And finding **y** given **x** involves first finding **t**. And really gets ugly fast because there can be multiple **y** values for a given **x** in a looped spline.

For More Help

Additional info on cubic splines can be found on our **Cubic Spline** library page. As are many dozens of examples of Bezier cubic spline techniques.

Additional consulting services are available per our **Infopack** services and on a contract or an hourly basis. Additional **GuruGrams** are found **here**.

Further **GuruGrams** await your ongoing support as a **Synergetics Partner**.