Alex Benny

Math 1C

Professor Mailhot

20 May 2022

Bezier Curves

Abstract:

The bézier curve is a cornerstone in computer graphics. It is a commonly used algorithm that applies parametric curves to graphics to create curves in things like animation tools or drawing tools. Say you are given two points with direction. The Bézier curve can use the two vectors to create a slope between the two points. This is very useful for modeling motion because it allows users to see the change in velocity between point A and B rather than just the change in distance.

My research includes the history of bézier curves, specifically why they were created and what prompted their use. The algorithm they follow will be included as well with a detailed analysis on what each part does, followed by real world applications of bézier curves with emphasis on how they have evolved over time and been adapted into newer technology.

Thesis:

In an electronic world with instantaneous results, bézier curves offer a simple solution to a complicated problem in graphing, and are vital for all computer graphics.

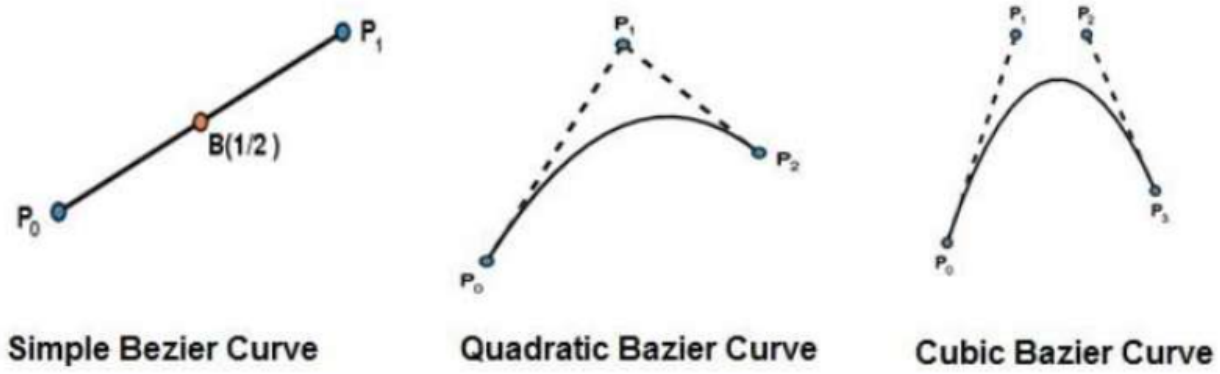
Introduction/History:

Bézier curves were a working concept starting in the mid 1960s, but the concept was actually invented in 1912 by Sergei Natanovich Bernstein. However, Bernstein's work was ahead of his time, as computers weren't even invented yet, and without this technology, his mathematical concept had a very limited industrial application. It would only be 50 years later

when Pierre Bézier made the theory his own. Bézier was looking for ways to increase fuel efficiency in vehicles, and referred to Bernstein's theory to help model curvy car bodies. It would take a few more years, however, for the bezier curve to see use in software application.

Concept:

Bezier curves generally come in three forms.



Simple Bezier Curve | Quadratic Bazier Curve | Cubic Bazier Curve

The difference between each of these is the number of control points used to generate the curve, but they all follow a similar procedure. Bezier curves are expressed with the parametric equations: $x(t) = (1-t)x_0 + tx_1, \quad y(t) = (1-t)y_0 + ty_1$

These equations are the fundamentals of bezier curves. They create the simple bezier curve described above, where x_0 and y_0 make up P_0, and x_1 and y_1 make up P_1. The line can also be described as $P(t) = (1-t)P_0 + tP_1$, since x_0 and y_0 represent P_0. Just remember that this equation is created by combining the two equations for simplification purposes. Also, notice how when t is 0, the equations equate to point P_0, and when t is 1, P_1 is created. So now we can set t to always be between 0 and 1. This is an important step when defining the parametric equations because we only want to focus on the line segment between the two control points. With this, we have successfully created a bezier curve. We can easily scale through different

points on the simple bezier curve by changing the value of t, as long as it remains within the set boundaries.

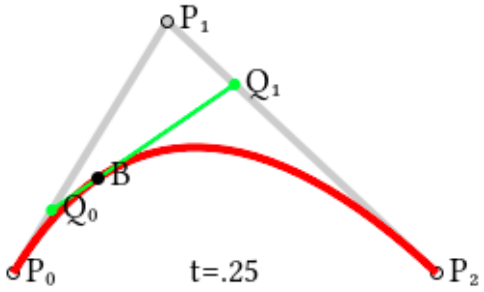
The quadratic bezier curve uses a similar concept, tracking a point Q_0 on line P_0 and P_1, and Q_1 on line P_1 and P_2. Using the equations listed earlier, we can create a parametric equation for each line and track points Q_0 and Q_1 by changing the value of t. They are defined as:

$$Q_0 = (1-t)P_0 + tP_1,$$
$$Q_1 = (1-t)P_1 + tP_2.$$

Now that we have defined the Q points, we can create another parametric equation that will give us the points used to model the quadratic bezier curve.

$$C(t) = (1-t)Q_0 + tQ_1$$

This equation defines a point between Q_0 and Q_1 that belongs on the bezier curve. For example, when t = 0.5, the Q points will be in the middle of the lines they reside in. Then, this equation takes the line between the two Q points and finds where t = 0.5, and plots it for the bezier curve to trace. This process is repeated until an acceptable curve is created, using various values of t between 0 and 1, and resulting in the quadratic bezier curve. Here is a more descriptive model of one.



In this example, C is displayed as B. Combining everything together gives us the parametric equation equation:

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

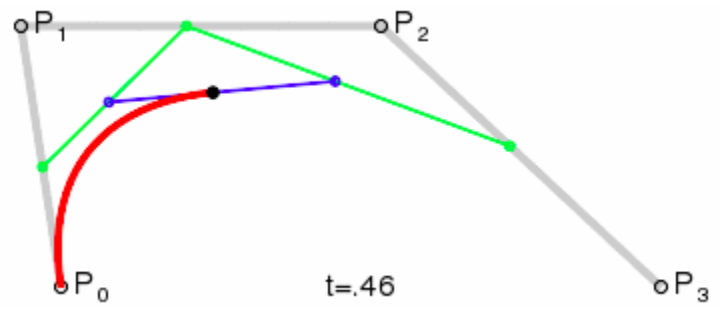
This is the final equation for creating a bezier curve with 3 points, the quadratic bezier curve formula.

Cubic bezier curves use four points, but the logic stays the same. There are three lines between the four points, meaning there are three Q points for a value of t. From there, we can resort to the methods we use when creating the quadratic bezier curve by establishing the Q

points as P points instead, and using the quadratic bezier curve formula we can find the point for the cubic bezier curve. The formula for quadratic bezier curves is

$$\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2 t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3$$

Here is an example of a quadratic bezier curve. In this sample, the green lines are the lines between the Q points for each line. From there, you can clearly see how the processes we used for quadratic bezier curves take over. Bezier curves don't end here, however. It is possible to use even more control points to model more complex structures. This idea leads into concepts like Bernstein polynomials, which I will not be covering. But it is possible, using the logic used to derive the formula for the quadratic and cubic bezier curves, to create a formula that uses more control points.



Software Application:

Bezier curves can be found anywhere a curve is found on a screen. From text to symbols, bezier curves are essential for creating any curves on a computer. When resizing text, what's really happening is the computer is resizing the control points used to create the text. This all stems from Adobe, the first group to use bezier curves in technology.

Adobe's first product actually relied heavily on bezier curves. The goal of Adobe initially was to break into the graphics computer language field. Their first product is called PostScript, a programming language meant to be used by printers to read text and print it out correctly. This language used bezier curves to accurately print out curves that couldn't be printed from older

programming languages. PostScript quickly became the global graphics language, used by any printer or desktop to efficiently render text.

Realizing the power of bezier curves in geometric construction, Adobe moved from the image processing industry to art. They wanted to have this technology to be applicable to anybody, since the use of PostScript was gated by programming knowledge. So, they developed a drawing interface that would allow users to interact with the PostScript code. On this drawing interface, users could set the points needed to create bezier curves, resulting in a useful tool for artists or designers to use to create smooth curves. This ended up being Adobe Illustrator, a tool that completely revolutionized the world of professional graphic designers.

Conclusion:

Bezier curves are a very useful algorithm that efficiently use parameterized curves to solve a simple problem. This research is to understand how they were created, and what they are useful for. Further research on this topic would go over how Bernstein polynomials are used in bezier curves, or how matrices can be used to model bezier curves. It would also include more detail of how bezier curves are used in modern day applications, like how they are used to create text, or how bezier curves with more than 4 control points work.

Work Cited

"Computer Graphics Curves." *Tutorials Point*, https://www.tutorialspoint.com/computer_graphics/computer_graphics_curves.htm.

"Finding T Vlaue in Bezier Curve." *MathOverflow*, 1 Feb. 1963, https://mathoverflow.net/questions/211804/finding-t-vlaue-in-bezier-curve.

Shiach, Jon. "Bezier Curves." *YouTube*, YouTube, 9 Mar. 2015, https://www.youtube.com/watch?v=2HvH9cmHbG4.

"What Is Postscript and Why Do Almost All High-End Printers Support It?" *Hackworth*, 26 Sept. 2013, https://www.hackworth.co/what-is-postscript-and-why-do-almost-all-high-end-printers-support-it.

Wickes, Takashi. "An Ode to the Bezier Curve." *Medium*, Prototypr, 4 June 2018, https://blog.prototypr.io/an-ode-to-the-bezier-curve-3eb9eca038ff.