Investigating Bézier Curves

Pranav Joneja
HL Mathematics IA
Mr. White

Introduction:

At some point during his or her study of mathematics, every student has tried to make drawings, write messages, or doodle on their graphing calculator. By this point in their high school career, everyone already knows how to make straight lines using linear functions to make simple illustrations. However, many students are stumped when it comes to drawing more complex drawings that involve curved lines. One creative solution to this problem utilizes Bézier curves.

The elegant simplicity of Bézier curves comes from the fact that two or more simple linear segments can be used to construct any curved shape imaginable. In other words, if one is given any shape that can be drawn freehand using a pencil and paper, it is possible to mathematically model it using Bézier curves.

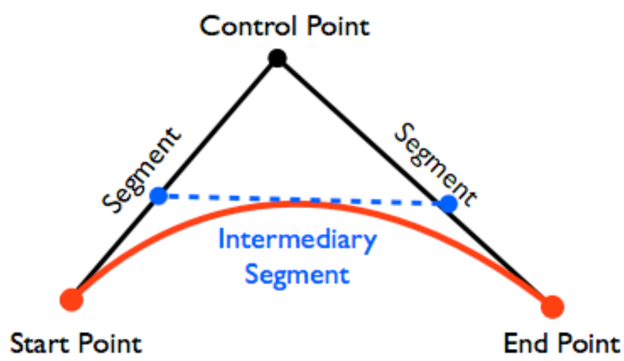
In this investigation, I will explore some of the properties of Bézier curves that make it so simple to replicate drawings. To do this, I will attempt to model several famous logos using as few Bézier curves as possible. I will then condense the curves from component equations down to simple parametric functions, which can be graphed (on a graphing calculator, for example) to recreate the original logo.

Characteristics and Definitions:

This section will define the parts of Bézier curves. These definitions will be referred to throughout this paper.

Points:
For the purposes of this paper, the points that form a Bézier curve are annotated in Figure 1. The Bézier curve itself is drawn in red. In addition, each point exists on a Cartesian plane, and therefore has a co-ordinate pair (x,y) associated with it.

Figure 1 - The names of points and segments that make up a Bézier curve



The black linear segments are defined by expressions in terms of t. Furthermore, t is always limited to $0 \leq t \leq 1$ for the rest of the investigation (Sederberg).

For example, the parametric equation of a segment could look like this:

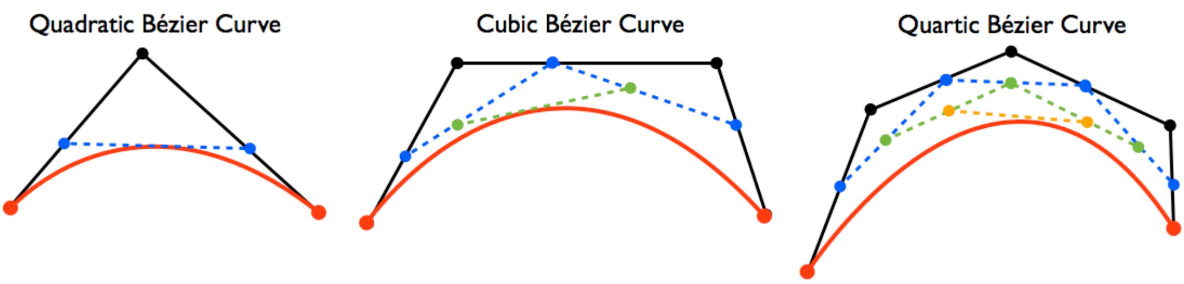
$x = 2 + t$
$y = 3 + 2t$ Indicating that the segment starts at the position (2,3) and then has a vector $\begin{pmatrix}1\\2\end{pmatrix}$, ending at the control point (3,5) when t is at its maximum, t = 1.

The blue intermediary segment is an imaginary segment that has endpoints on other segments. It is useful to draw when creating Bézier curves, but is not pertinent to defining the curve itself. The curve is defined solely by the start and end points and the control points in between them (Sederberg).

<u>Order</u>

Bézier curves can be constructed in higher orders, such as cubic, quartic, quantic etc. This refers to the number of linear segments used to construct a Bézier curve. It's important to note that this is not the same as the number of <u>intermediary</u> segments (Huynh).

Figure 2 - The order of a Bézier curve is determined by the number of segments used to construct it



...and so on.

<u>Investigation – Model 1:</u>

The first logo I attempted to model is the logo for Princeton University, which is one of the colleges I am applying to. For further simplicity, I opted to only draw the outline of the shield. I chose this logo because it is involves relatively simple curves that can be modeled by four quadratic Bézier curves. In addition, it is symmetrical, opening the door to exploring properties of symmetry that make constructing Bézier curves easier.

Figure 3 - Princeton University logo

Start and End Points
The first property of Bézier curves I discovered was quite obvious to begin with:

The curve begins at the start point of the first segment and ends at the end point of the last segment.

This means that when drawing the Princeton University logo, finding the start and end points of the Bézier curves is simply a matter of overlaying it on a Cartesian plane and finding the co-ordinates that correspond to the points on the edge of the shield.

In my model, these four points on the edge of shield were:
(3.15, 15.8) ; (10.08, 0.56) ; (17.01, 15.8) ; (10.08, 18.45)
These points are also shown in Figure 4 below. They will be the start and end points of the four Bézier curves that will be constructed.

Control Points
What remains is finding the control points for each of the four Bézier curves. While fiddling with the construction, I discovered the second property:

The first and last segments are tangent to the curve at the start and end point respectively.

In the context of the Princeton University logo, this meant that the line tangent to the curve at the start point was the slope or vector direction of the first segment. And, that the line tangent to the curve at the end point was the slope or vector direction of the last segment.

In my model, the tangent line were measured as shown in Figure 4:

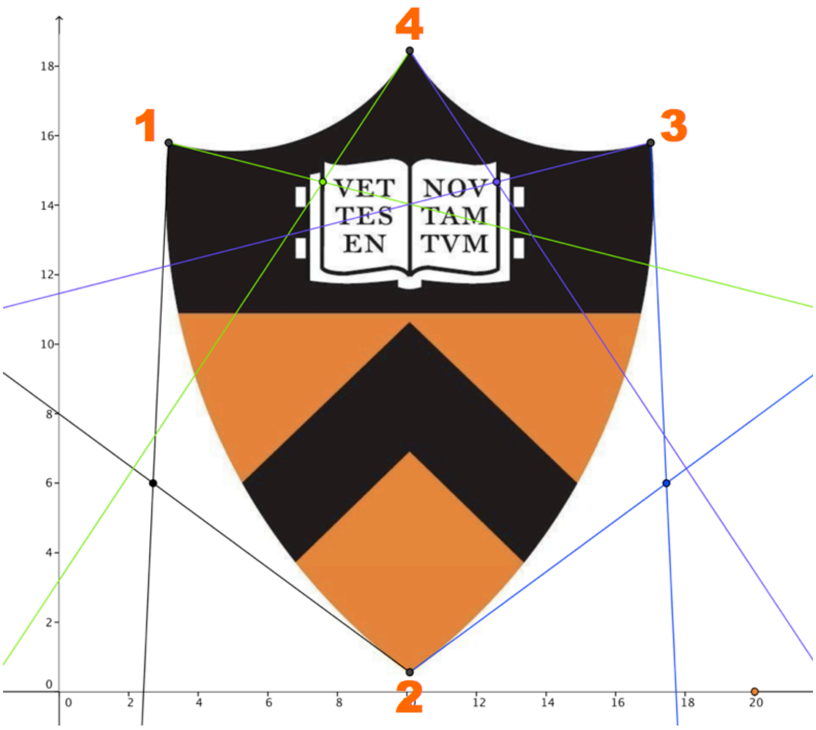


Figure 4 - Princeton University logo with 4 start/end points and corresponding pairs of tangent lines

The four pairs of colored tangent lines refer to the four Bézier curves that will be drawn.

The slopes of the pairs of tangent lines were measured and are recorded in Figure 5:

Figure 5 - Slopes of tangent lines at the start and end points of all four Bézier curves

	Black	Blue	Green	Purple
Slope of tangent line at start point	21.7	-21.7	-0.255	0.255
Slope of tangent line at end point	-0.737	0.737	1.51	-1.51

The slopes of the tangent lines were found to be the direction of the segments used to construct the Bézier curves. There are two segments for each curve because they are quadratic curves.

Armed with the starting/ending points and slopes, it was possible to use simple linear algebra to find where the lines intersect. It follows that this point of intersection will be the control point.

The following is a sample calculation that shows how the control point was found (for the **Black** pair in this example)

$$y = mx + b$$
$$15.8 = (21.7)(3.15) + b$$
$$b = -52.55$$

$$\therefore y = 21.7x - 52.55$$

$$y = mx + b$$
$$0.56 = (-0.737)(10.08) + b$$
$$b = 7.99$$

$$\therefore y = -0.737x + 7.99$$

Set equations equal to each other

$$21.7x - 52.55 = -0.737x + 7.99$$
$$x = 2.69$$

Plug into either equation to find y

$$y = -0.737x + 7.99$$
$$y = -0.737(2.69) + 7.99$$
$$y = 6$$

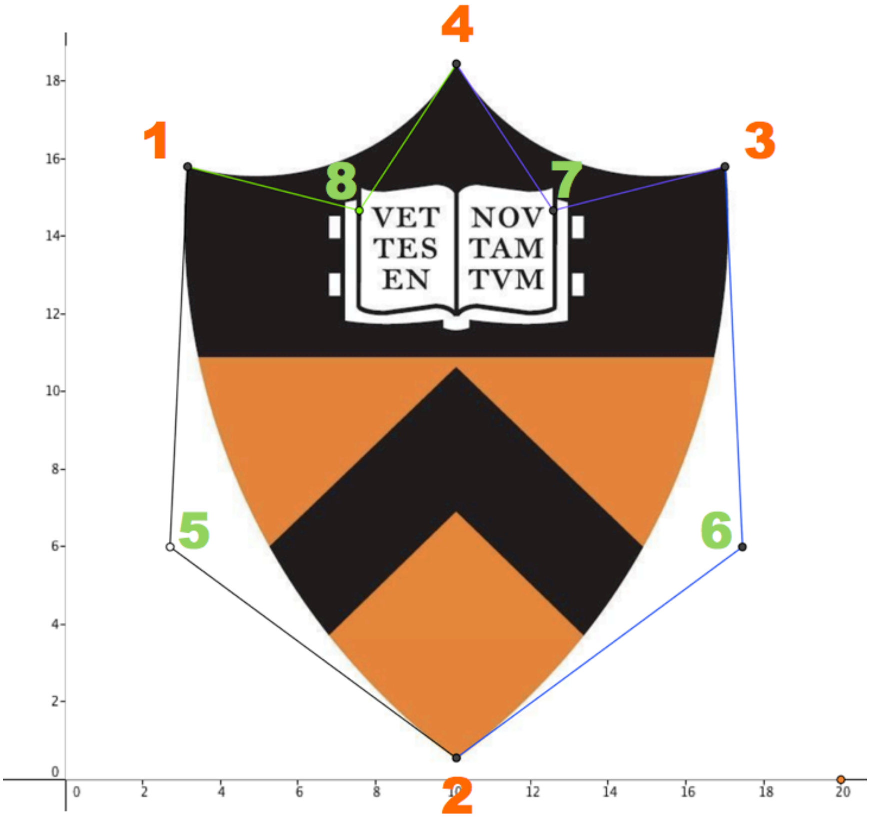
Therefore, it was found that the point of intersection is (2.69,6). This is the control point for the **Black** Bézier curve.

Thus, the control points for all four Bézier curves are as follows (and are also shown in Figure 6:
(2.69,6) ; (17.46,6) ; (7.58,14.67) ; (12.58,14.67)

Now that the start point, control point and end points have been found, it is possible to join all these points together to make the segments. As mentioned earlier, these segments will be the basis for the construction the curve.

At this point, the model of the Princeton University logo is shown in Figure 6:

Figure 5 - Princeton University logo with segments for all four Bézier curves. Start/end points are in orange, control points are in green

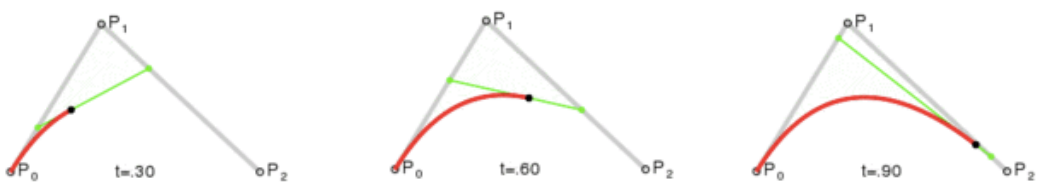


Intermediary segments

According to my research into Bézier curves, the easiest way to finish the modeling process is to create intermediary segments (Huynh). Although they do not directly lend themselves to the shape of the curve in the end (the start/end point and control point determine the shape of the curve), they are useful in making sense of the mathematics.

The endpoints of the intermediary segment lie on each of the segments. The endpoints are defined parametrically in terms of t. This means that as t varies from $0 \leq t \leq 1$, the position of the endpoints also varies. In reality, the endpoints can be seen sliding along the segments at a constant rate, influencing the intermediary segment. This 'sliding' can be seen in Figure 7 and animated in Animation 1 (attached).

Figure 6 - Sliding endpoints of the green intermediary segment (t = 0.3 ; 0.6 ; 0.9)



The 'sliding' begins at the start point of the Bézier curve (the start point of one segment) and continues towards the control point as *t* varies. This can be modeled like this:

(start point) + (**distance moved** towards control point in terms of *t*) × *t*

In general, the mathematic way to express this in parametric form of a point is as follows:

Let Point A (A_x, A_y) be the **start point**.
Let Point B (B_x, B_y) be the control point.
It follows that B – A is the **distance moved** towards the control point

Thus (Huynh),

$$\left(A_x + t(B_x - A_x), A_y + t(B_y - A_y)\right)$$

In the context of modeling the Princeton University logo, the intermediary segments were created using the component form above.

The following sample calculation for the **Black** Bézier curve shows how the intermediary segment was created by defining the endpoints in terms of *t*.

Start point: A_x (3.15, 15.8)
Control point: B_x (2.69,6)

$$\left(A_x + t(B_x - A_x), A_y + t(B_y - A_y)\right)$$

$$\left(3.15 + t(2.69 - 3.15), 15.8 + t(6 - 15.8)\right)$$

$$(3.15 - 0.46t, 15.8 - 9.8t)$$

The same process can be adapted for the other endpoint of the intermediary segment, this time using the control point and the end point of the Bézier curve

Control point: B_x (2.69,6)
End point: C_x (10.08, 0.56)

$$\left(B_x + t(C_x - B_x), B_y + t(C_y - B_y)\right)$$

$$\left(2.69 + t(10.08 - 2.69), 6 + t(0.56 - 6)\right)$$

$$(2.69 + 7.39t, 6 - 5.44t)$$

After doing the same process for the other three Bézier curves, Animation 2 was created, showing the intermediary segments sliding on the original segments.

After creating Animation 2, the third useful property of Bézier curves is clearly observable:

The final intermediary segment slides so as to always remain tangent to the curve being modeled.

Drawing the curve

The intermediary segment is tangent to the curve at all times. It makes sense, then, to find a point that is on both the intermediary segment and the curve (the point at which the intermediary segment is tangent to the curve). By defining this point for all values of *t*, the curve can be modeled with a tracing point.

Once again, the tracing point can be modeled in the following component form:

Let Point T (T_x, T_y) and Point U (U_x, U_y) be the endpoints of the intermediary segment.

$$\left(T_x + t(U_x - T_x),\ T_y + t(U_y - T_y)\right)$$

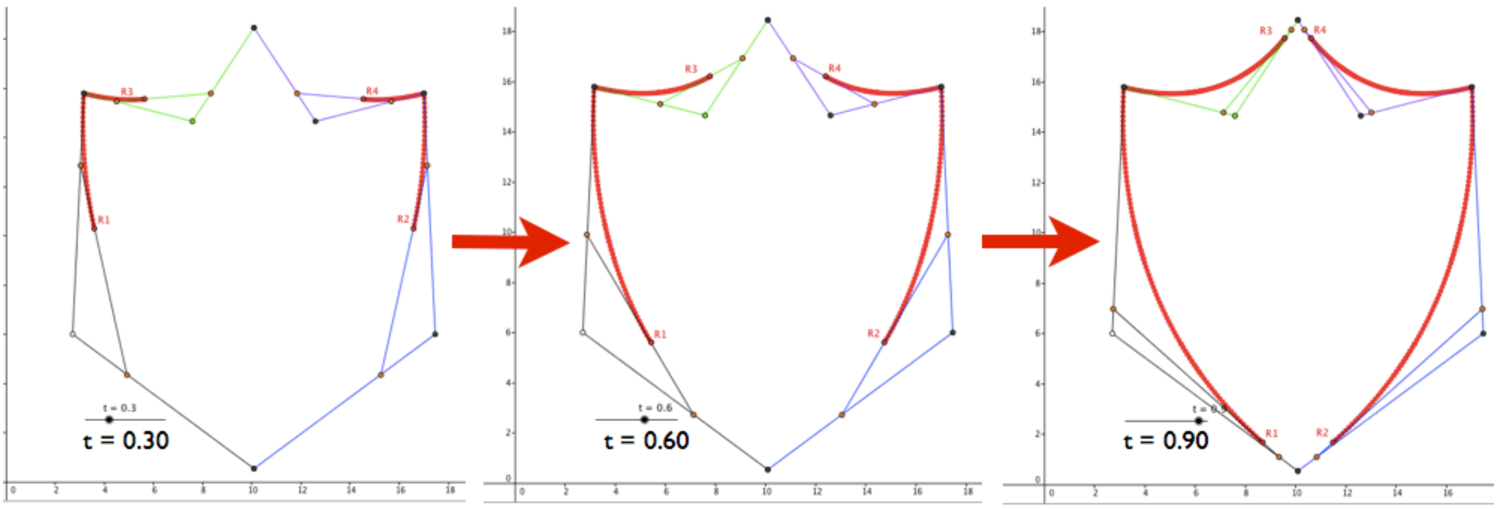
Substitute T and U with expressions in terms of the start point A (A_x, A_y), control point B (B_x, B_y) and end point C (C_x, C_y) of the Bézier curve.

$$\left(A_x+t(B_x-A_x) + t(B_x+t(C_x-B_x) - A_x-t(B_x-A_x)),\ A_y+t(B_y-A_y) + t(B_y+t(C_y-B_y) - A_y-t(B_y-A_y))\right)$$

This idea is known as **iteration** – substituting the point back into the general component form, over and over again for more iterations. In other words, the idea of *starting point + distance moved* × *t*, was iterated again to make each tracing point.

In total, to draw the shield shape, four tracing points were used, each drawing one Bézier curve. Figure 8 below shows this at *t=0.30; 0.60* and *0.90.*

Figure 8 - Bézier curves being drawn by tracing points to create Princeton University shield



Animation 2 (with the photo) and Animation 3 (without the background photo) show the Bézier curves in motion.

Investigation – Model 2:

The next logical step in my investigation was to take this concept and apply it to drawing more complex curves. Once again, I took inspiration from popular culture to find a drawing I could model with Bézier curves. I decided to model the cover of Bruno Mars' album *Doo-wops & Hooligans*, pictured to the right. The reason this was chosen was that it included a variety of tight curves, some of which that doubled back on themselves. Hence, if a Bézier curve model can be successfully created, then it shows that the concept as a whole is useful in graphic design applications.

Figure 9 - Bruno Mars' album cover *Doo-wops & Hooligans*

To begin this model, the start and end points were found on the curve itself. This follows the first property as discovered in the first model –the curve begins at the start point and ends at the end point.

Next, following the second property, the first and last segments were drawn, tangent to the curve at the start and end points respectively. The start/ends points and tangent lines are shown in Figure 10 below:

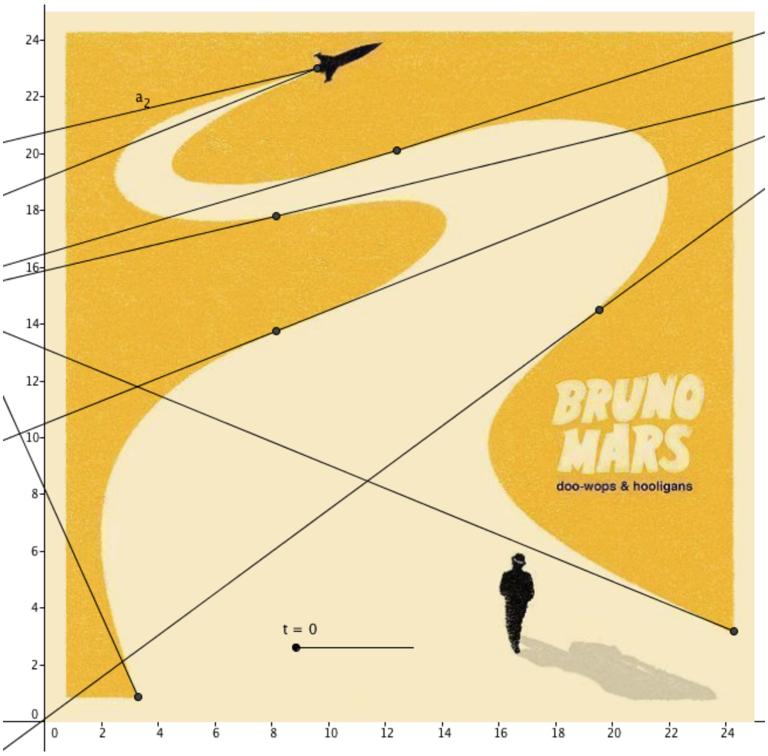


Figure 10 - Bruno Mars album cover with start/end points and tangent lines

Two Control Points:
In this model, many of the curves double back on themselves, which means that cubic (third order) Bézier curves are required. This part of the investigation shows how two control points are necessary to make a cubic Bézier curve.

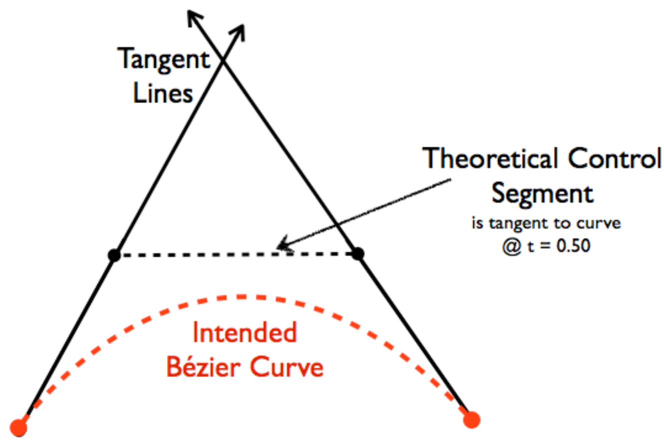


Figure 11 - Control segments and its relevance to the two control points in a cubic Bézier curve

Cubic Bézier curves require a segment between the two tangents, called the 'control segment'. The investigation showed that this control segment connects the two original tangent lines. In addition, the control segment is parallel to the 'peak' of the curve being drawn, at approximately $t = 0.5$, as shown in Figure 11.

Figure 11 also shows that the endpoints of the control segment are on the tangent lines. These are the two control points to be used in this cubic Bézier curve. By connecting the start point, two control points and the end point, the intermediary segments can be made, as shown in Figure 2 above.

Using this process, the start/end points, and control points were determined and are presented in the table below (Figure 12):

Figure 12 - Table showing start/end points and control points for Bruno Mars' album cover

Bézier curve number [polynomial order]	Start Point	Control Point(s)	End Point
R1 [quadratic]	(3.27, 0.87)	(-0.90, 10.15)	(8.15,13.75)
R2 [cubic]	(8.15, 13.75)	(17.2, 17.35); (15.2, 19.5)	(8.15, 17.8)
R3 [cubic]	(8.15, 17.8)	(2.68, 16.51); (-1.88, 20.29)	(9.6, 23.0)
R4 [cubic]	(24.28, 3.18)	(15.22, 6.89); (12.79, 9.54)	(19.53, 14.5)
R5 [cubic]	(19.53, 14.5)	(23.94, 17.73); (23.07, 23.52)	(12.4, 20.1)
R6 [cubic]	(12.4, 20.1)	(3.21, 17.4); (1.57, 19.76)	(9.6, 23.0)

Note that quadratic curves [R1] only have one control point, while cubic curves [R2 through R5] have two.

<u>Iterations:</u>
For cubic Bézier curves, the tracing point is defined by iterating the x- and y-component form of <u>three</u> times. This begins to look very complicated, although is not very conceptually difficult to understand. It is the same as in Model 1, only with one more iteration.

For brevity, the points are defined as:
Let Point C *(C_x, C_y)* be the start point
Let Points D *(D_x, D_y)* and E *(E_x, E_y)* be the two control points
Let Points F *(F_x, F_y)* be the end point

The first iteration (same as before):

$$\left(C_x + t(D_x-C_x),\ C_y + t(D_y-C_y)\right)$$

This defines the sliding endpoints of the first level of intermediary segments. The points move in a linear fashion.

The second iteration (same as before):

$$\left(C_x+t(D_x-C_x) + t(D_x+t(E_x-D_x) - C_x-t(D_x-C_x)),\ C_y+t(D_y-C_y) + t(D_y+t(E_y-D_y) - C_y-t(D_y-C_y))\right)$$

This defines the points sliding on the intermediary segments themselves. Two of them on two intermediary segments create the endpoints of the final moving segment that is always parallel to the curve (third property). These points resemble quadratic Bézier curves)

The third iteration:
x-coordinate:

$$C_x+t(D_x-C_x) + t(D_x+t(E_x-D_x) - C_x-t(D_x-C_x)) +$$
$$t(D_x+t(E_x-D_x) + t(E_x+t(F_x-E_x)-D_x-t(E_x-D_x)) - C_x - t(D_x-C_x) - t(D_x+t(E_x-D_x) - C_x-t(D_x-C_x)))$$

y-coordinate:

$$C_y+t(D_y-C_y) + t(D_y+t(E_y-D_y) - C_y-t(D_y-C_y)) +$$
$$t(D_y+t(E_y-D_y) + t(E_y+t(F_y-E_y)-D_y-t(E_y-D_y)) - C_y - t(D_y-C_y) - t(D_y+t(E_y-D_y) - C_y-t(D_y-C_y)))$$

This defines the tracing point of the cubic Bézier curve.

In summary, start/end points were found, tangents were drawn, and control segments were drawn to find the position of control points. Then, intermediary segments were created (in two iterations) and the curve itself was drawn with a tracing point (of the third iteration), This expands upon Model 1 of the Princeton University logo.

Figure 13 on the next page shows how the mathematical representation of Bruno Mars' album cover was finally drawn.

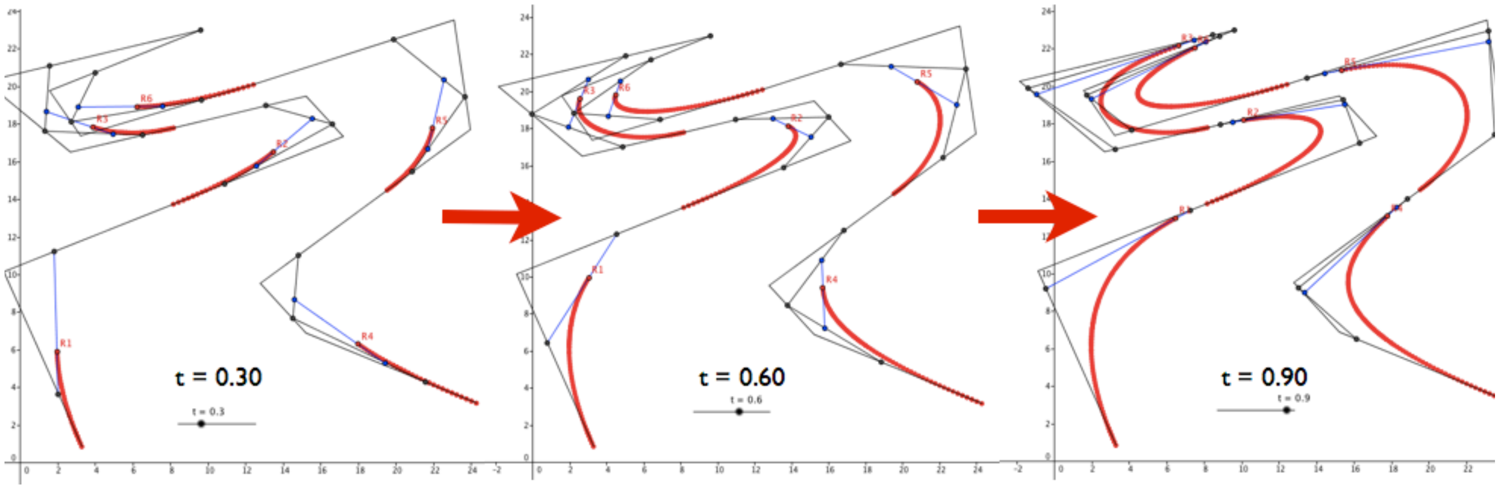


Figure 13 – Bézier curves being drawn by tracing points to create mathematical representation of Bruno Mars' album cover

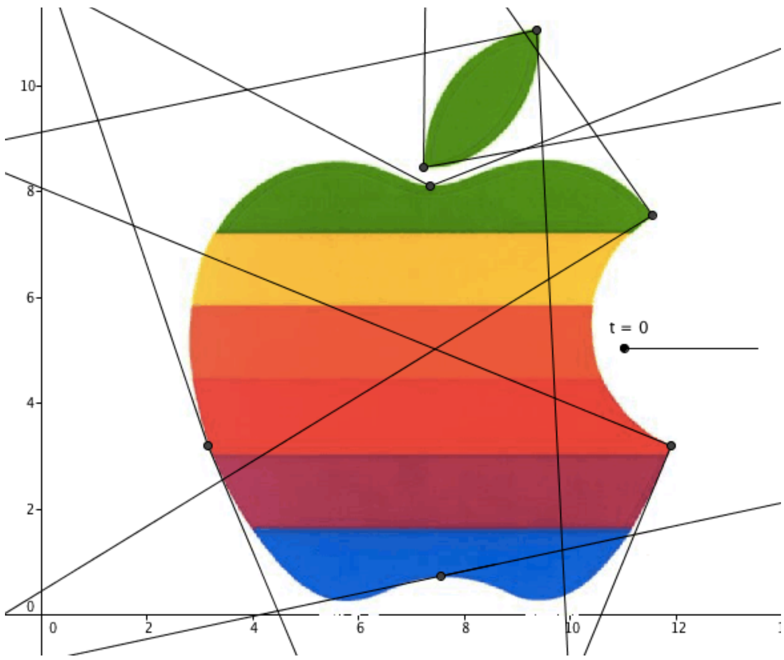
Animation 4 (with the photo) and Animation 5 (without the background photo) show the Bézier curves in motion.

Investigation - Model 3:

The final model in my investigation was to apply all the concepts I had discovered and apply it to the most ubiquitous and famous logo of all – the Apple inc. logo. Seeing as I work on my Apple laptop for many hours each day, this was the most obvious choice.

Once again, the model began with finding start/end points and drawing tangents, as shown in Figure 14 below.

Figure 14 - Apple logo with start/end points and tangent lines



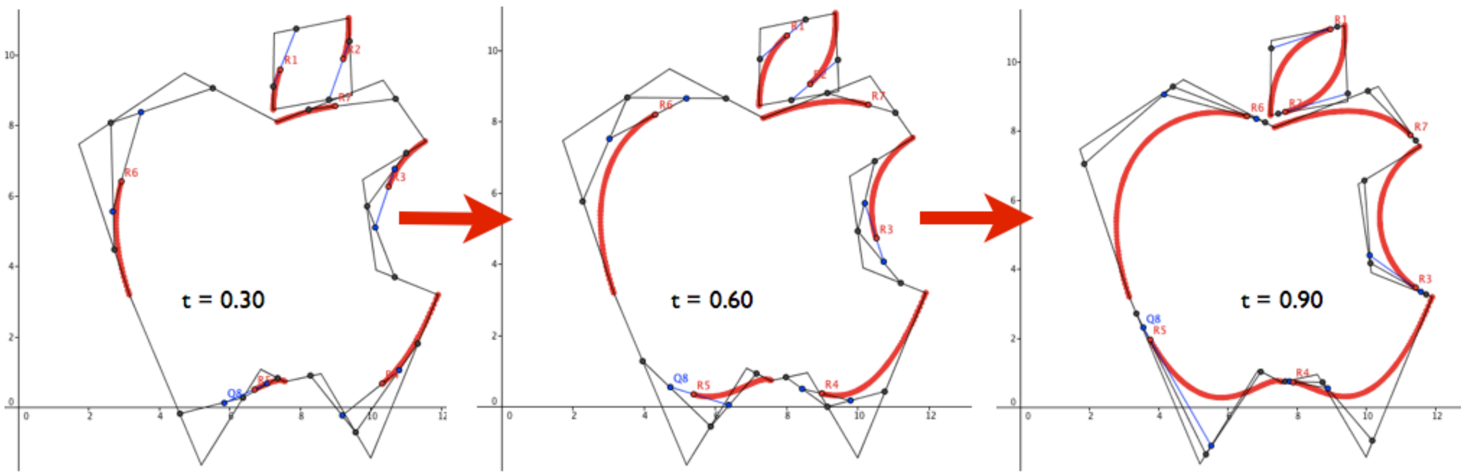
The table (Figure 15) shows the coordinates of the start points, numerous control points and end points.

Figure 15 - Table showing start/end points and control points for Apple logo

Bézier curve number [polynomial order]	Start Point	Control Point(s)	End Point
R1 [quadratic]	(7.23, 8.46)	(7.25, 10.62)	(9.36, 11.06)
R2 [quadratic]	(7.23, 8.46)	(9.47, 11.06)	(9.36, 11.06)
R3 [cubic]	(11.53, 7.56)	(9.77, 6.47); (10.15, 3.92)	(11.9, 3.21)
R4 [cubic]	(11.9, 3.21)	(9.98, -1.42); (8.59, 0.97)	(7.55, 0.75)
R5 [cubic]	(7.55, 0.75)	(6.88, 1.08); (5.18, -1.62)	(3.14, 3.21)
R6 [cubic]	(3.14, 3.21)	(1.7, 7.49); (4.72, 9.49)	(7.34, 8.11)
R7 [quadratic]	(7.34, 8.11)	(10.33, 9.28)	(11.53, 7.56)

The appropriate segments between these points were created using the same iterative process until finally arriving at the seven tracing points that drew the seven Bézier curves that make up the Apple logo. Figure 16 shows how the Apple logo was finally drawn, at *t=0.30; 0.60* and *0.90*.

Figure 16 - Bézier curves being drawn to mathematically represent the Apple logo



Animation 6 (with the photo) and Animation 7 (without the background photo) show the Bézier curves in motion.

The Final Step – Making 'Graphable' Parametric Functions

In it's current form, expressed as x- and y-components, the Bézier curves I have created are not particularly useful when it comes to actually making drawings on my graphing display calculator. Instead, the most appropriate way to transfer these drawings to my GDC would be to make parametric functions out of them. The following math achieves just that.

Quadratic Bézier Curves:
The investigation(s) above revealed the component form of a general, quadratic Bézier curve to be:

$$\left(A_x+t(B_x-A_x) + t(B_x+t(C_x-B_x) - A_x-t(B_x-A_x)),\ A_y+t(B_y-A_y) + t(B_y+t(C_y-B_y) - A_y-t(B_y-A_y))\right)$$

This can be combined into a single expression as:

$$A + t(B - A) + t(B + t(C - B) - A - t(B - A))$$

Expanding it gives:

$$A + Bt - At + Bt + Ct^2 - Bt^2 - At - Bt^2 - At^2$$

Combining like terms of *t* gives:

$$(C - 2B + A)t^2 + 2(B - A)t + A$$

As a general parametric function, it would be:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} (C_x - 2B_x + A_x)t^2 + 2(B_x - A_x)t + A_x \\ (C_y - 2B_y + A_y)t^2 + 2(B_y - A_y)t + A_y \end{pmatrix}$$

Cubic Bézier Curves:

As a single expression:

$$C+t(D-C)+t(D+t(E-D)-C-t(D-C))+t(D+t(E-D)+t(E+t(F-E)-D-t(E-D))-C-t(D-C)-t(D+t(E-D)-C-t(D-C)))$$

Expanding it (and already combining terms for simplicity):

t^3: $Ft^3 - Et^3 - Et^3 + Dt^3 - Et^3 + Dt^3 + Dt^3 - Ct^3$

t^2: $Et^2 - Dt^2 - Dt^2 + Ct^2 + Et^2 - Dt^2 + Et^2 - Dt^2 - Dt^2 + Ct^2 - Dt^2 + Ct^2$

t: $Dt - Ct + Dt - Ct + Dt - Ct$

1: C

Combining like terms of *t* gives:

$$(F - 3E + 3D - C)t^3 + (3E - 6D + 3C)t^2 + (3D - 3C)t + C$$

Simplifying:

$$(F - 3E + 3D - C)t^3 + 3(E - 2D + C)t^2 + 3(D - C)t + C$$

Expressed as a general parametric function:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} (F_x - 3E_x + 3D_x - C_x)t^3 + 3(E_x - 2D_x + C_x)t^2 + 3(D_x - C_x)t + C_x \\ (F_y - 3E_y + 3D_y - C_y)t^3 + 3(E_y - 2D_y + C_y)t^2 + 3(D_y - C_y)t + C_y \end{pmatrix}$$

The two general parametric functions, for quadratic and for cubic curves, are written in terms of their start/end points and control points. To represent the curves in each of the three models above as a parametric function, it was simply a matter of substituting the appropriate points to their corresponding place. The substitution was done using Microsoft Excel, enabling me to quickly process the numbers for all the curves. The result is a single parametric equation for each curve. These can be easily graphed on a graphing display calculator, allowing me to show doodles on my calculator. The following screenshots present the final simplified polynomial parametric functions graphed using GDC emulator software

Figure 17 - Princeton logo graphed on a GDC (emulator)

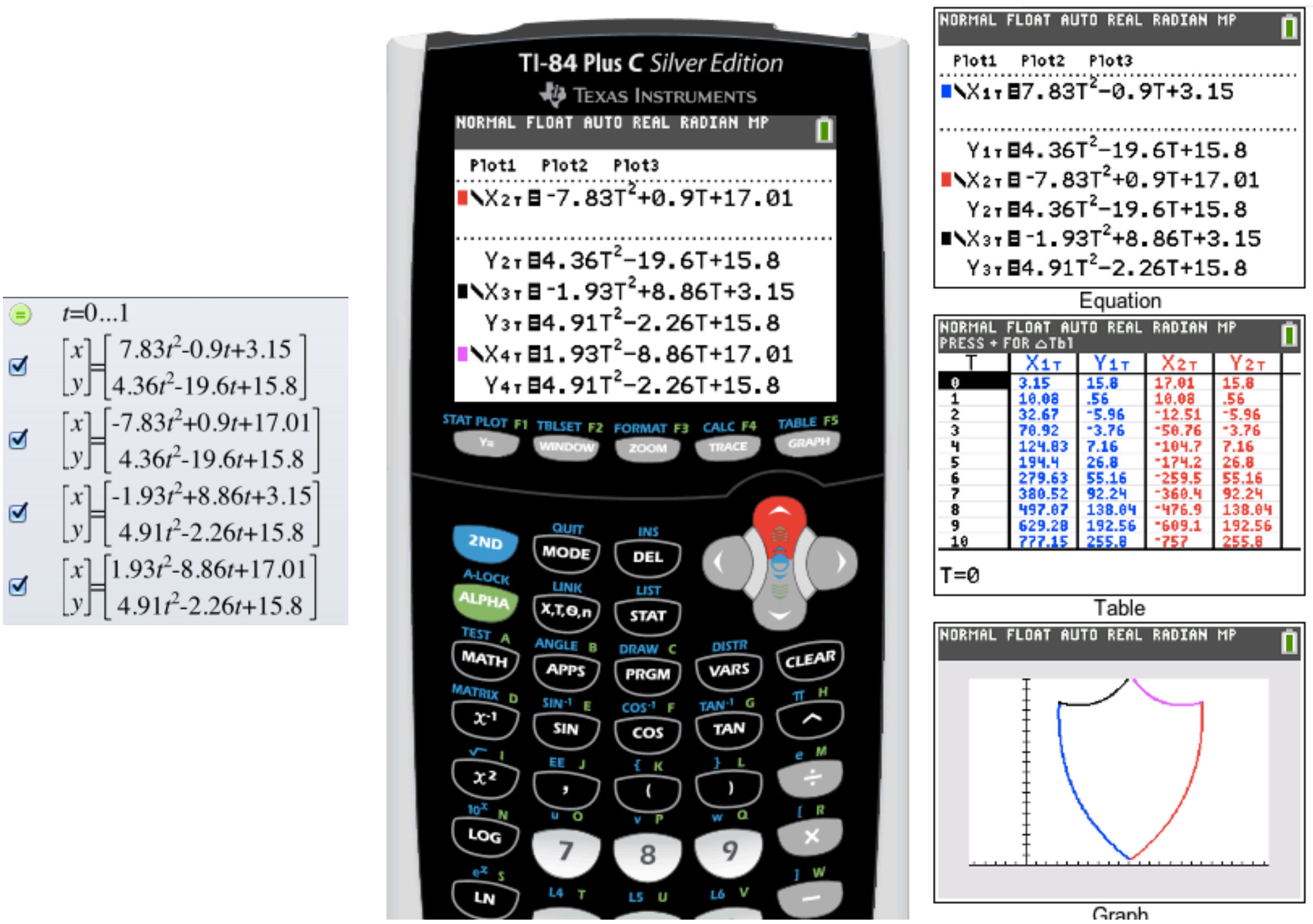


Figure 18 - Bruno Mars' album cover graphed on a GDC (emulator)

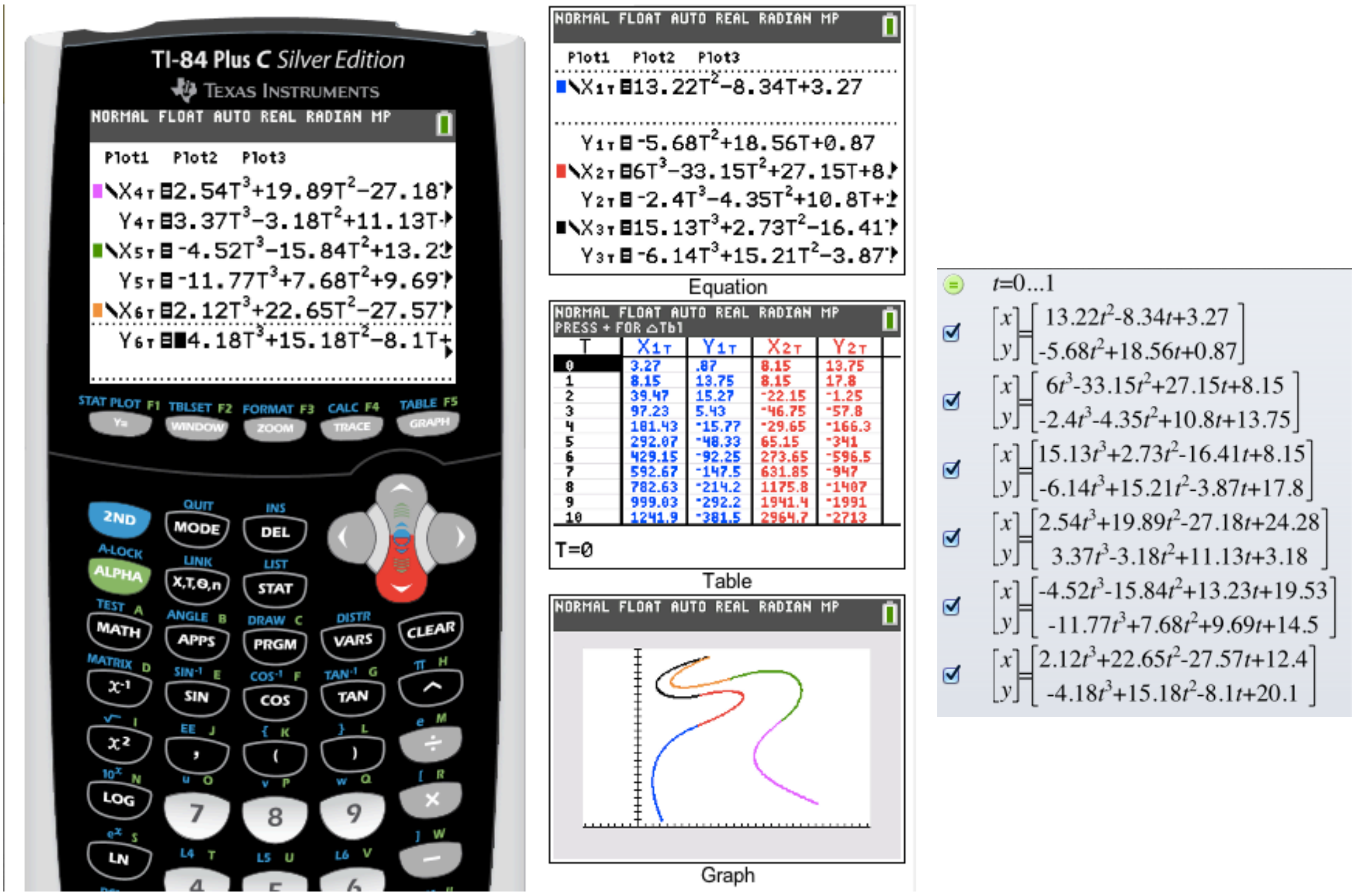
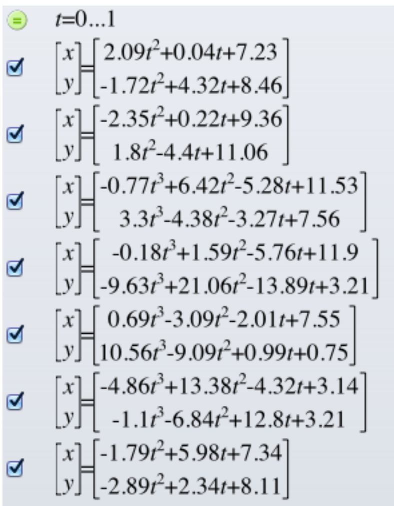
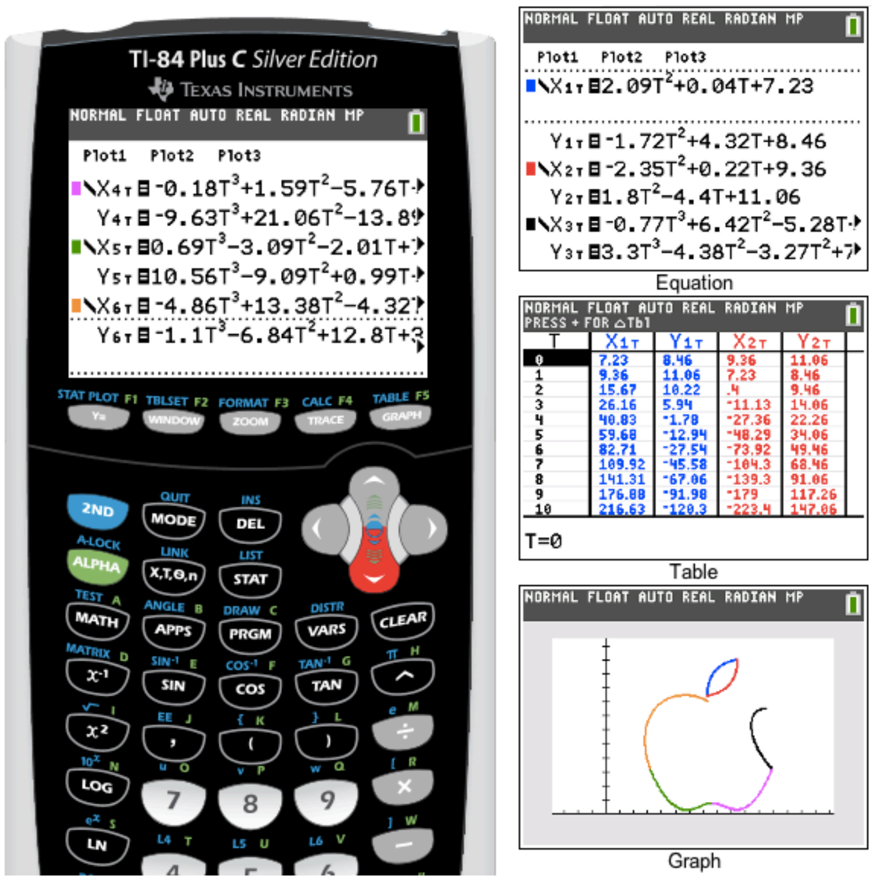


Figure 19 - Apple logo graphed on a GDC (emulator)





Note: the seventh parametric function could not be graphed due to limitations of the emulator software (max. 6 functions). See Appendix for full graph.

Conclusion and Evaluation:

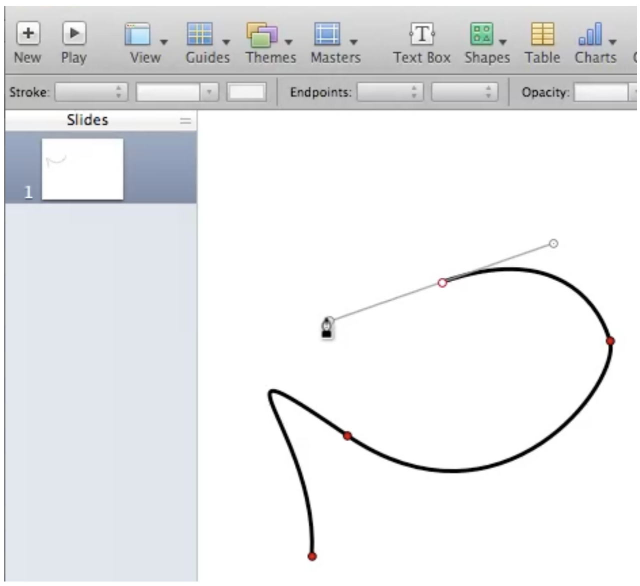
This investigation was inspired by some doodles on my graphing display calculator. I wanted to be able to draw more than just line segments and simple curves, so I did some research into how curves can be illustrated on a simple, Cartesian plane. My research pointed me towards Bézier curves as they are an actual mathematical tool used by graphic designers, and are implemented in graphic software.

My systematic investigation helped to reach several conclusions, namely:

1. *The curve begins at the start point of the first segment and ends at the end point of the last segment.*
2. *The first and last segments are tangent to the curve at the start and end point respectively.*
3. *The final intermediary segment slides so as to always remain tangent to the curve being modeled.*
4. *The control point in a quadratic Bézier curve is found by calculating the intersection of the two original tangent lines*
5. *The control point(s) in a cubic Bézier curve is found by drawing a line parallel to the peak of the intended curve, and setting that to $t = 0.50$. The two points where this new parallel line intersects the two original tangent lines are the two control points.*

These five properties of Bézier curves I discovered provide reasons for why this tool is so simple and useful. The implication is that complex curves can be drawn by simply 'eyeballing' the start and end points and constructing a few tangent lines in between. This is the reason that Bézier curves are so ubiquitous in the graphic design industry. In particular, basic computer applications as common as Pages and Keynote use a stylized version of Bézier curves to enable users to draw curves. Based on what I can see in Figure 20, I am quite certain that curves are in fact modeled by computer code based on Bézier curves. The link can be seen even more clearly in Animation 8.

Figure 20 - Screenshot of curve drawing tool based off Bézier curves in Keynote, a graphic presentation application



Other applications of Bézier curves include interpolation of data sets, fonts (alphabet glyphs) and geometric invariance ("Bezier Curve.").

The aim of this investigation was achieved when I managed to show that complex curved drawings could be modeled on my quotidian GDC, as shown in Figures 17-19 above.

This investigation was limited to only quadratic and cubic functions. The reason for this is that many, if not all, shapes can be broken up into curved segments that can be modeled by combinations of these two types of curves. This is evident in the three models I created. In addition, I found that lines were defined by 3 (3^1) terms, quadratic Bézier curves by 9 (3^2) terms, and cubic functions by 27 (3^3) terms. This pattern would continue for higher order polynomial curves, with the number of terms growing exponentially (3^n) like a nuclear fission reaction, and would quickly become unmanageable. I was already making mistakes with juggling 27 terms, so I really did not feel 81 terms would have been feasible.

This idea directly lends itself to where this investigation could be directed in the future. With some computer programming and coding skills, I could conceivably begin to work with quartic, quintic and even higher order polynomials. While this does move away from simple, classroom doodles, some curves would actually look more elegant if they were modeled by higher order polynomials. For example, I can already see that the bottom half of the Apple logo (Model 3) would look much better as a quintic function, rather than two, awkward-looking cubic Bézier curves.

Perhaps in the future, as I begin looking into a profession in the graphic design industry more seriously, I could explore how calculus could be used to determine the length of the arc drawn by a Bézier curve function.

In addition, future investigations could also look at how Bézier curves can be extended into the third dimension by adding a third parameter (in the z-direction) to the function.

Bibliography:

"Apple Logo." *Logo Pictures*. Blogspot, 25 Apr. 2013. Web. 19 Nov. 2013. <http://hms-somerset-co.blogspot.com/2012/12/apple-logo.html>.

"Bezier Curve." *Wolfram*. Wolfram Research, n.d. Web. 19 Nov. 2013. <http://reference.wolfram.com/mathematica/ref/BezierCurve.html>.

"Bruno Mars, Doo-Wops & Hooligans." *FYM Stories*. FYM Stories, 1 June 2011. Web. 19 Nov. 2013. <http://www.fymstories.com/2011/06/review-bruno-mars-doo-wops-hooligans.html>.

Huynh, Joe. "Bezier Curve Directed Investigation." *WebTemplate*. Catholic Education South Australia, n.d. Web. 19 Nov. 2013. <http://cesa.webtemplate.com.au/sites/2823/__files/f/7715/Joe%20Huynh.pdf>.

Sederberg, T. W. "Chapter 2 - Bézier Curves." *BYU Computer Science*. Brigham Young University, n.d. Web. 19 Nov. 2013. <http://cagd.cs.byu.edu/~557/text/ch2.pdf>.

"Top 10 Universities in the USA." *Entire Education*. Entire Education, 22 Nov. 2011. Web. 19 Nov. 2013. <http://www.entireeducation.com/top-10-universities-in-the-usa-2011/>.

Appendix 1:

This appendix includes all three mathematical representations on separate Cartesian planes. This is the formal presentation of the final result of this investigation.

Photo 1 - Princeton University logo

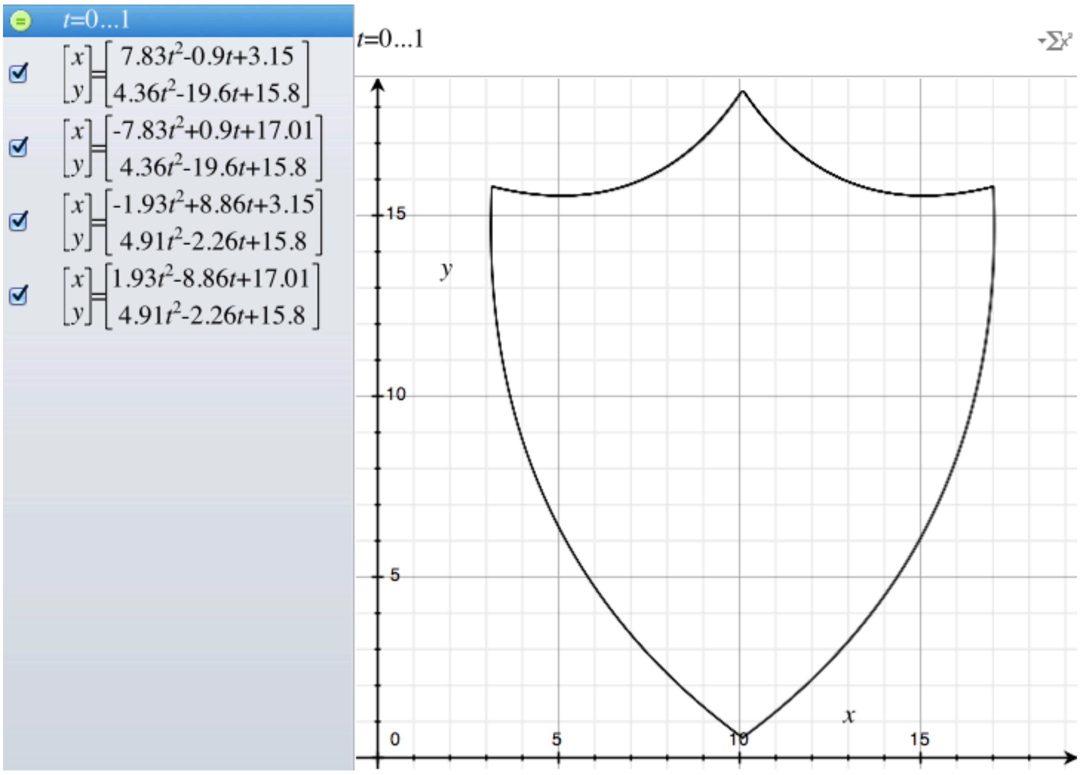


Photo 2 - Bruno Mars album cover

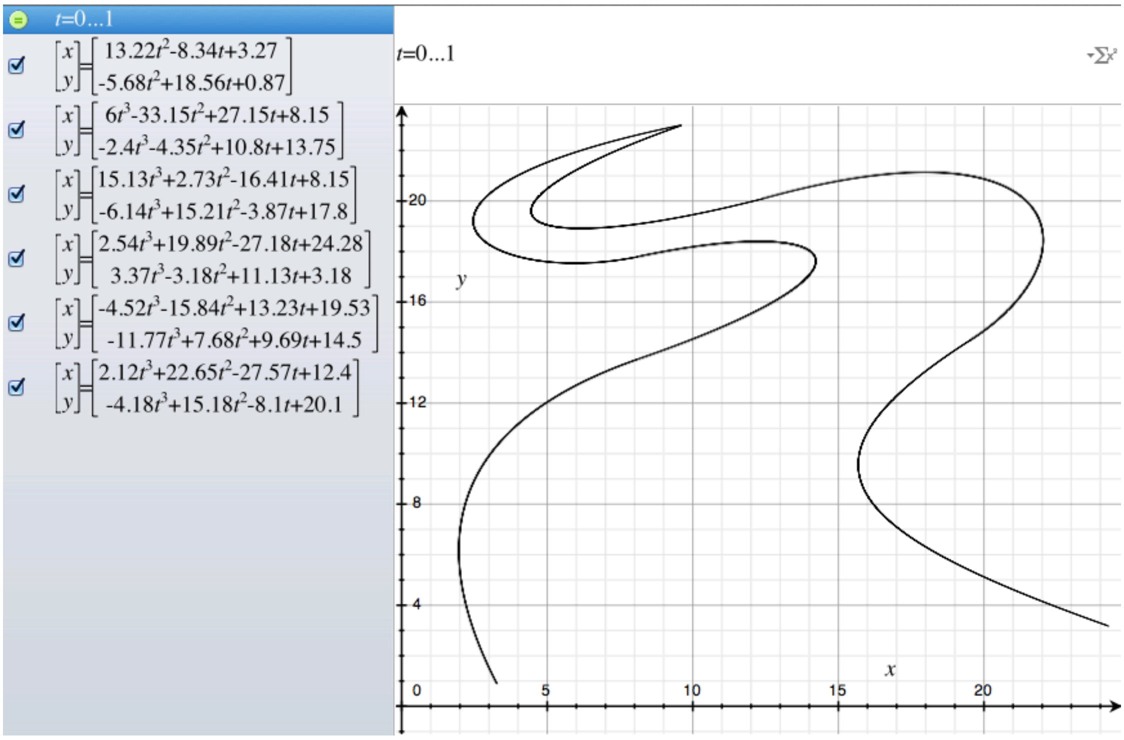


Photo 3 - Apple logo