# Curvas de Bezier y sus algunas aplicaciones

## Bezier Curves and some applications

-Patricio Vejar

**Resumen:**

En este documento se abarcará una breve explicación de que son las curvas de Bezier. Como se puede evaluar hasta N puntos y algunos ejemplos para comprender sus aplicaciones. Utilizando como herramienta visual y matemática, Wolfram Alpha y un programa en Python trabajado por el autor.

Palabras clave: Bezier, Geometría diferencial, Parametrización, Graficado, Ecuaciones

**Summary:**

In this document, we will approach an brief explanation as to what are the Beizer Curves. How could one calculate up to N given points and some examples to comprehend it’s applications. Working with Wolfram Alpha and a Python program made by the author to aid as a visual/graphic and mathematical tool.

Keyword’s: Bezier, Differential Geometry, Parameterization, Equations, Graph

**Introducción:**

Entendiendo la naturaleza de las funciones parametrizadas y su practicidad. Podemos preguntarnos al igual que con una función cualquiera. Como podemos crear una función paramétrica a partir con solo Puntos dados.   
Del mismo modo que en una función cartesiana con puntos X,Y podemos armar una función de 2 dimensiones con una complejidad dependiente de la cantidad de puntos.

Uno puede crear una función paramétrica a partir de varios puntos dados en un sistema de coordenadas. Y al igual que cualquier función su complejidad depende de la cantidad de puntos dados.

Tomemos un espacio tridimensional. Si tomamos dos puntos podemos trazar una línea entre estos. En este escenario es simple la proposición. Uno genera una ecuación lineal donde se pueda evaluar los puntos dados.

B(t) = \*El punto inicial más la diferencia entre en punto final y el inicial por un tiempo \*

B(t) =

Esta ecuación lineal, se le llama curva lineal de Bezier. De esta ecuación podemos obtener una cantidad de ecuaciones igual a la cantidad de componentes de los puntos. Pues se calcula cada recorrido de la función. En el caso del espacio obtendríamos 3 ecuaciones, todas perteneciendo a B(t), la función paramétrica. Como sus componentes:

B(t) = (x(t) , y(t) , z(t)) =

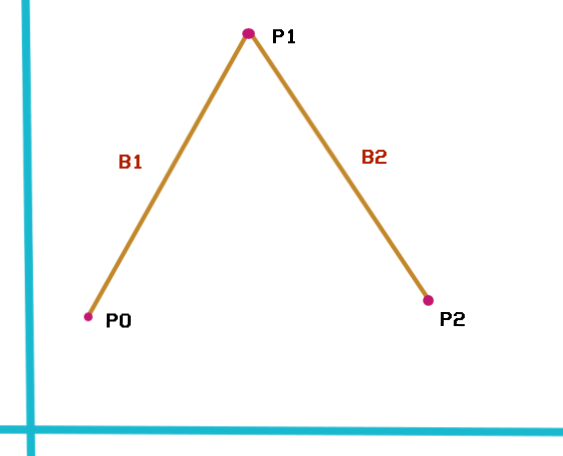
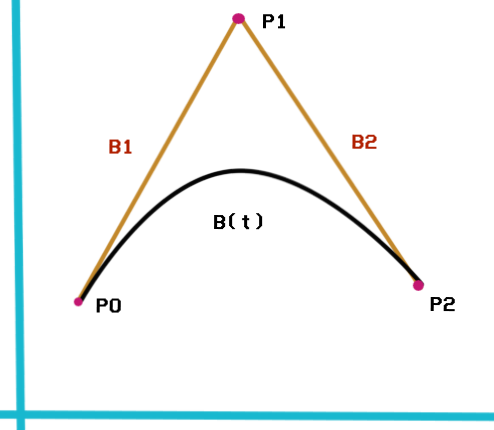
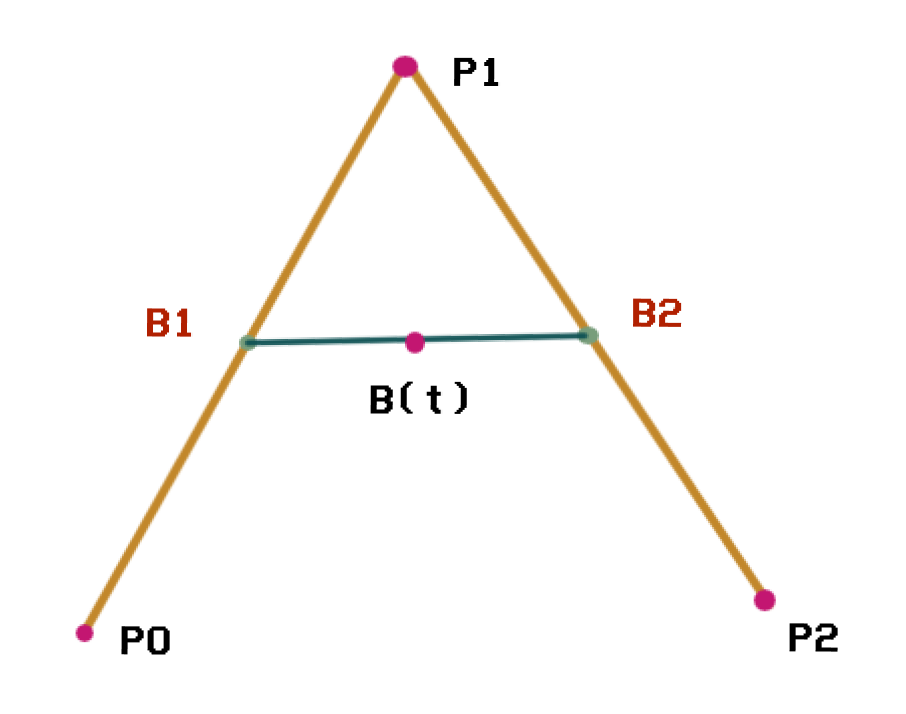
B(t) = (

Esta forma de cálculo de componentes se mantiene en toda forma al calcular las ecuaciones de Bezier. *La ecuación de los puntos determina la ecuación de todos sus componentes.*

Que ocurre si añadimos otro punto a la formula, que se tendría que hacer exactamente.

Bueno, entendiendo que en la primera formula B(t) era la suma de el punto inicial y la diferencia del punto final a la inicial en el tiempo. Lo que seria la traza del inicio al punto final.

Esto se replica en cada punto, siendo el punto anterior al final. Para entenderlo mejor visualizando dichas trazas:



Cada traza lineal será llamada líneas de control, son lo que lo que determina la forma de la parametrización, desde P\_0 se la función se dirigirá hasta P\_1 y conforme se acerca a P\_1 empezará su trayecto hacia P\_2.

B1 (t) =

B1 (t)=

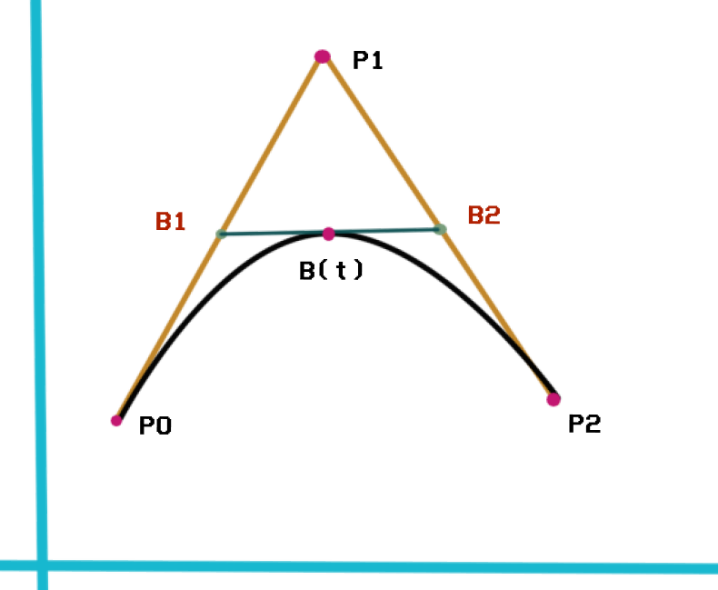
B2 (t)=

B2 (t)=

Ahora para poder obtener los puntos de B(t) y subsecuentemente su función, calculamos del mismo método que con los puntos:

B(t) = =

Ahora podemos con la función clara, podemos notar que es una función cuadrática:





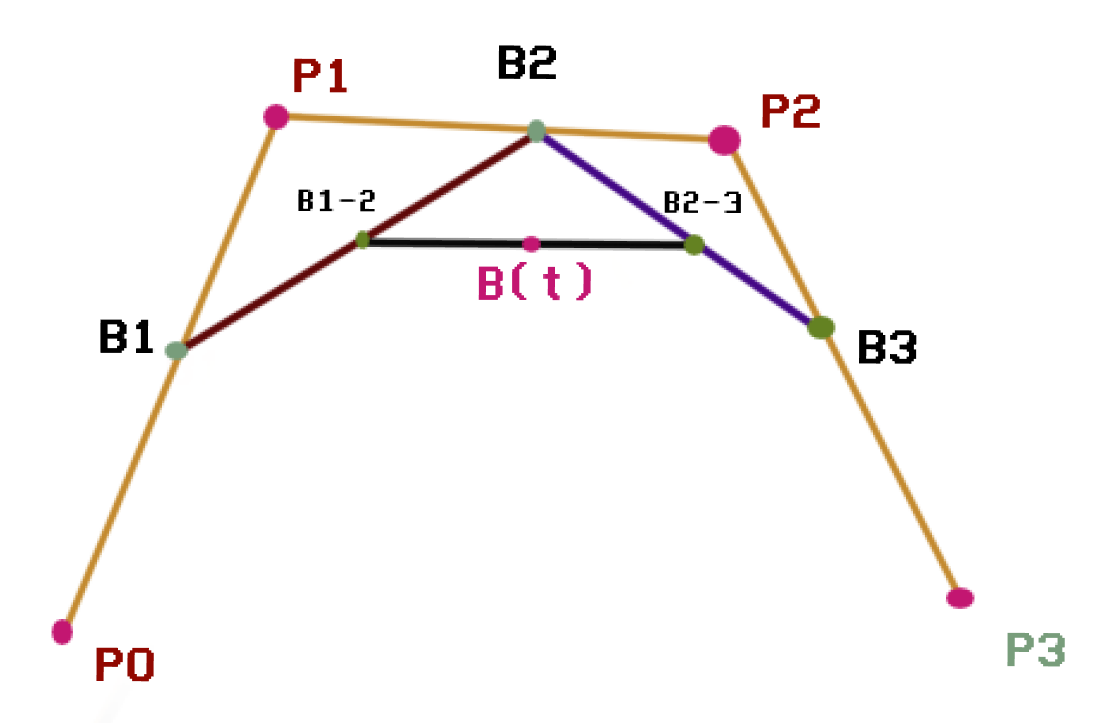
Un apunte que seguramente se notara conforme lee el documento es que las funciones de Bezier siempre van del punto inicial al punto final. Nunca tocando los puntos intermedios ; Ejemplos en dos dimensiones para explicar mejor la naturaleza de Bezier

¿Pero que le pasaría a la función si añadimos otro punto?

¿Como interpretaría el recorrido de este?

Si sabemos que la función cuadrática de Bezier es la función lineal entre las funciones lineales de todos los puntos, este razonamiento se mantiene para toda función de Breizer. Haciéndose mas y mas compleja pues la cantidad de líneas que hay entre los puntos para armar B(t) aumentaría:

En el caso de 4 puntos la gráfica de líneas de control seria:



Asique, la función B(t) sería la función lineal de las funciones lineal entre los puntos:

B1 (t) =

B1 (t)=

B2 (t)=

B2 (t)=

B3 (t)=

B3 (t)=

B 1-2 (t) = B 2-3 (t) =

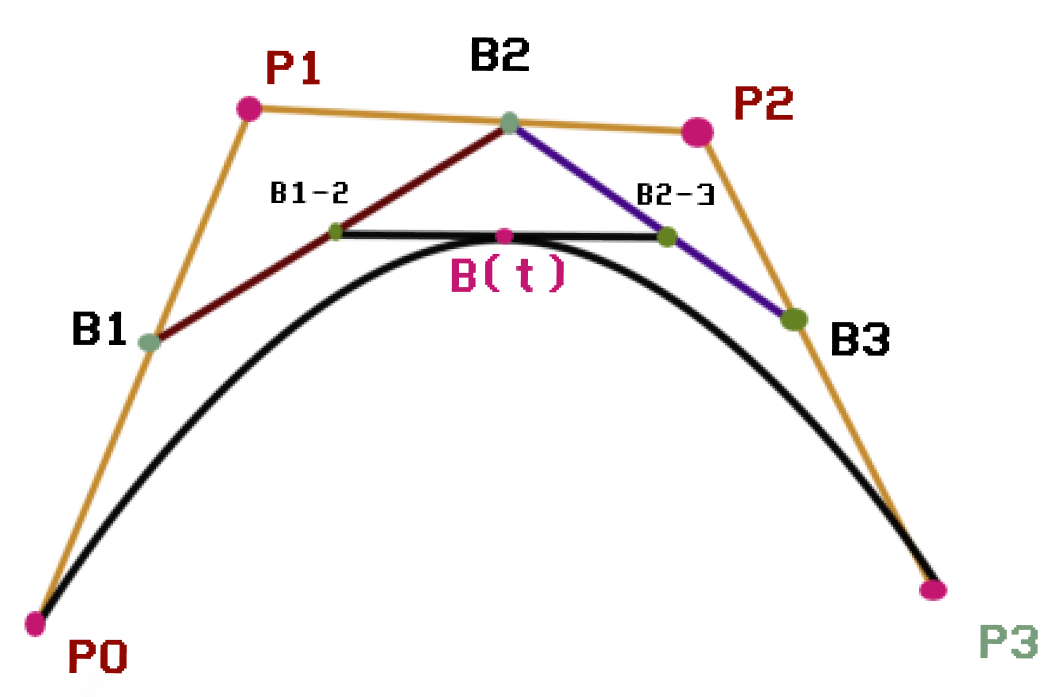
B(t) =

B(t) =

*Reemplazando y* simplificando*:*

B(t) =

De este modo obtenemos la función Cubica de Bezier.

 *Ahora para la parte divertida*

Si uno se fija notara cierto patrón conforme la curva de Bezier aumenta en complejidad. Asique con este patrón de cambio uno podría llevar estas ecuaciones de puntos limitados a una ecuación de N puntos,

Tomamos las ecuaciones dadas anteriormente, lineal, cuadrática y cubica.

Notar que para la lineal se dan 2 puntos, la cuadrática 3 y la cubica 4. Asique asumiendo que N siempre es positivo, al ser la cantidad de puntos a evaluar. Se diría que

Las curvas de Bezier con una cantidad N de puntos, Tienen complejidad N-1. Y notando cierto patrón en las curvas ya establecidas podemos darle una forma a esta curva N-1.

Primero excluimos la primera parte de las ecuaciones:

Este fragmento es una constante en todas las demás, con la diferencia que la diferencia de t está siendo elevada por N-1. Asique para la curva N-1 esa parte se escribiría:

Para el resto de la ecuación se acompleja un poco mas, pero con que entendamos que todos los fragmentos estas siendo sumados y la diferencia de t y t están siendo elevados en incrementos contrarios, podemos decir que:

\*

Juntamos los dos fragmentos de la ecuación final quedando. La función de Bezier de N-Puntos está dada por:

* B(t) =

Con esto obtendríamos una fórmula para la curva N-1 complejidad de Bezier.

Ex:

N=4:

N=5: =

*Un apunte Interesante es que esta formula fue encontrada durante un almuerzo familiar y esta escrita en una servilleta.*

Conclusión:

Al terminar este documento usted debería entender la naturaleza del actuar de cualquier curva de Bezier y el cómo armar funciones paramétricas gracias a esta.

Por parte del autor se diría que a través de Bezier y sin la limitación de la cantidad de puntos a evaluar. Uno podría aplicar estas fórmulas a una gran cantidad de escenarios relacionados con “pathfinding” en algoritmos de inteligencia artificial.

Referencias:

Wikipedia – Curvas de Bezier: <https://es.wikipedia.org/wiki/Curva_de_Bézier>

MathLike - ¿Cómo funcionan las curvas de Bezier?: https://youtu.be/oRd3l7ILTNY?si=Ce2dXAnVNfr7mhvo