Interpolación - Mortero

Camilo Ruiz[1], Alex Barreto[2], and Sebastian Roberts[3]

[1]ruizcamilo@javeriana.edu.co
[2]barreto.alex@javeriana.edu.co
[3]sroberts@javeriana.edu.co

Octubre del 2019

1. Introduction interpolación

En interpolación son varios los métodos que existen para modelar curvas y superficies. No obstante, surje la pregunta de como realizar aproximaciones de objetos reales buscando que el modelo sea lo más cercano al objeto real. Por su parte, el uso de los splines conlleva que, hay que ir reproduciendo la curva a tramos pequeños, con la diferencia de que además hay que saber de cálculo funcional que relacione la información de forma matemática.

Por su parte, el problema de los polinomios de Lagrange, que si pueden reproducir un tramo largo de curva, es que para obtener una buena aproximación hay que elevar el grado del polinomio, a veces demasiado, complicando así su representación gráfica, pero una pequeña modificación de los puntos de interpolación haría muy probable que ya no nos sirviese el mismo polinomio y sería necesario calcularlo de nuevo.

2. Curvas de Bezier

Las más importantes curvas que se usan actualmente en el diseño computacional son las curvas de Bézier y los B-Splines. Es gracias al trabajo de dos matemáticos franceses, Bézier y de Casteljau, que se desarrollan aplicaciones para el incipiente diseño computacional de los años 60, bajo el alero de la industria automotriz.

El resultado son herramientas que los diseñadores ocupan regularmente, bajo el nombre genérico de «trazados» o diseño vectorial, cuya matemática opera tras bambalinas con las ideas que estos matemáticos franceses, y otros más, desarrollaron.

La primera versión de las curvas de Bézier fue en realidad tridimensional, un esquema basado en curvas cuadráticas construidas dentro de un cubo, método que le permitirá describir cualquier curva de grado 2 a partir de sólo cuatro puntos.

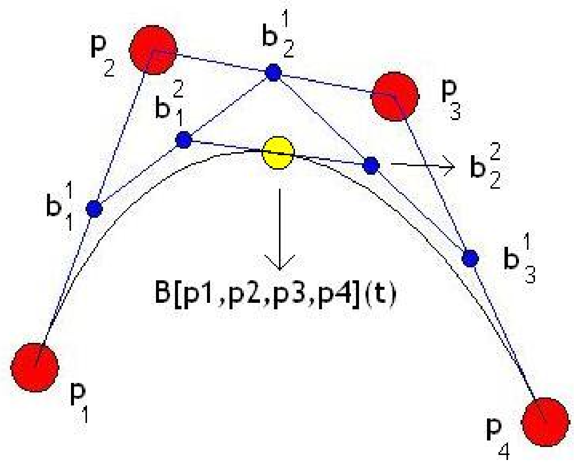

Figura 1: Curva de Bezier

3. Problema

El objetivo propuesto para este proyecto es conseguir dibujar el mortero valenciano usando superficies de Bezier y otro método (BSplines). Para ello se puede utilizar R(PathInterpolatR, gridBezier,vwline) o Python(griddata, matplotlib).
Se suguiere dividir la figura en cuatro cuadrantes, de manera que una vez construido uno, el resto puede representarse realizando rotaciones por ejemplo.

Figura 2: Mortero Valenciano

4. Método de solución

Para iniciar, es necesario indicar que hemos trabajado con curvas de bezier en R. Para ello hemos hecho uso de unas librerias o paquetes que nos proporciona R que son:

- bezier
- gridBezier
- plot3d
- rgl

Nuestro metodo de solución al problema de construcción estuvo compuesto de 4 partes:

- **Obtención de los puntos:** los puntos de la que sería la curva de bezier original serían hallados con un tercero que sería Geogebra. Dentro de geogebra graficamos con figuras vectoriales el mortero a partir de semiesferas como muestra la siguiente figura:

  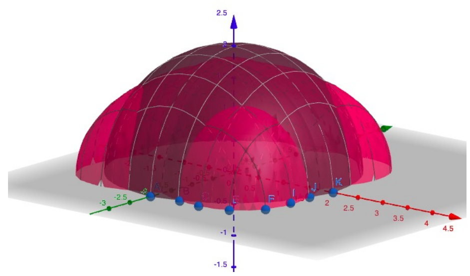

  Figura 3: Mortero en Geogebra

  De allí sacamos el contorno de puntos de la figura para ver cuales podrían ser los puntos para la curva de Bezier en R.
  Viendo estos puntos desde el eje z, estos fueron los resultados:

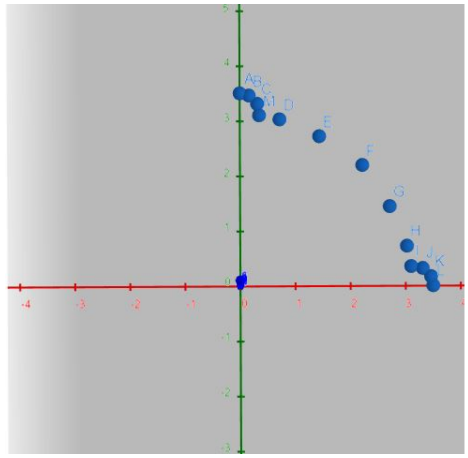

Figura 4: Puntos en Geogebra

Siendo estos los puntos:

x	y	z
0	3.5	0.5
0.17	3.46	0.5
0.32	3.31	0.5
0.35	3.1	0.5
0.72	3.03	0.5
1.44	2.72	0.5
2.22	2.19	0.5
2.72	1.44	0.5
3.03	0.72	0.5
3.1	0.35	0.5
3.31	0.32	0.5
3.46	0.17	0.5
3.5	0	0.5

Lo que se esta mostrando allí es el contorno del primer cuadrante de nuestro mortero.

- **Graficando el primer cuadrante:** el primer cuadrante se haría a partir de hacer distintas lineas de bezier que marcan los puntos que fueran del contorno previamente calculado. Estas líneas están compuestas por muchos puntos muy cercanos que hacen que pareciesen líneas.

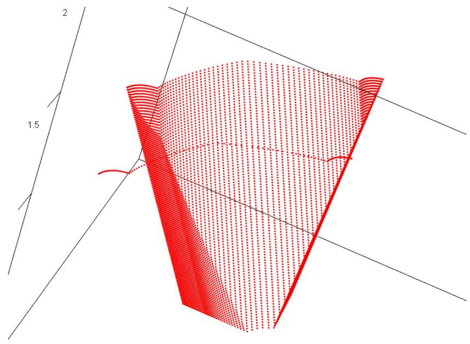

Figura 5: Primer cuadrante en R

A partir de eso se ira generando una forma que le da la perspectiva de ser un solido y, por consiguiente, nuestro primer cuadrante.

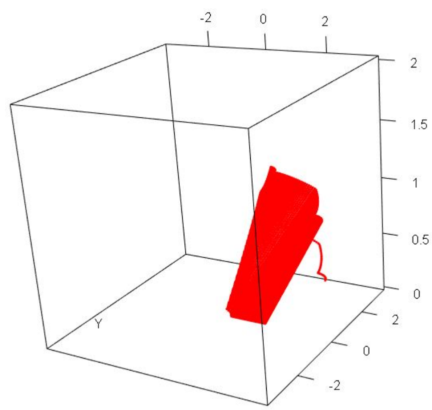

Figura 6: Primer cuadrante en R

Notese que hay una línea detrás, esa es la línea de bezier original.

- **Replicando los otros cuadrantes:** A partir de ese primer cuadrante que hemos hecho, se va replicando lo que se hizo. Es decir, vamos haciendo que se llenen los otros cuadrantes a partir del mismo metodo y del cuadrante inicial por medio de alterar los puntos dependiendo del cuadrante en el que se encuentre.

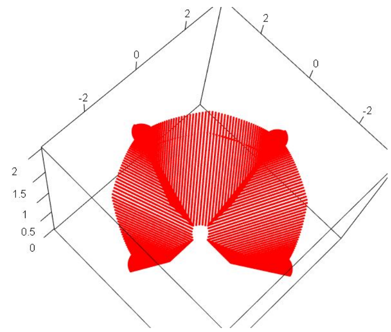

Figura 7: Primeros tres cuadrantes en R

Notese que en el ejemplo se han hecho ya 3 cuadrantes y solo falta uno.

- **Llenando el fondo del mortero:** Con la figura que tenemos después de haber completado los 4 cuadrantes queda un problema, hay un hoyo en el fondo de nuestro mortero.

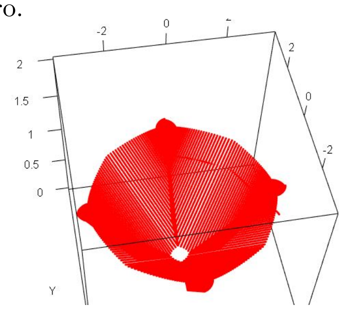

Figura 8: Figura incompleta cuatro cuadrantes en R

Para ello hacemos uso de una función que nos calcula todos los puntos por donde no llega ningun solido para x y para y, pero para zeta solo dejamos todos los valores en 0 para que esten alineados al fondo de nuestro mortero.

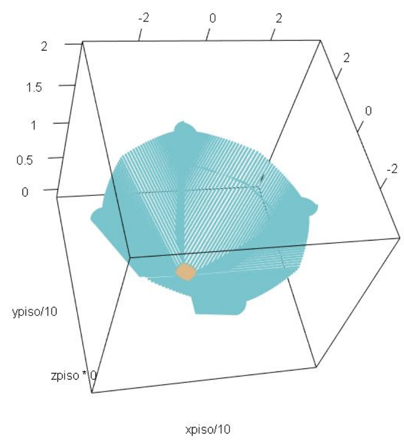

Figura 9: Mortero final en R

Notese el cambio de colores, este genera una forma más fácil de evidenciar los resultados.

5. Comparación

La comparaciòn la hicimos a partir del mismo solido resultante. Procedimos a coger el solido resultante y a hacer una versiòn .ºptimizada”de él, donde le redujimos la cantidad de puntos iniciales. Debido a que el suelo es el mismo

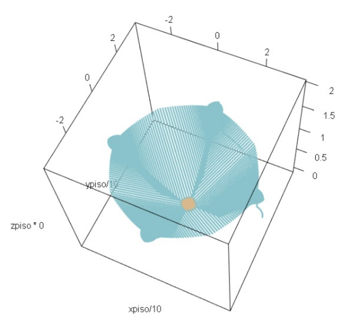

Figura 10: Primer solido formado.

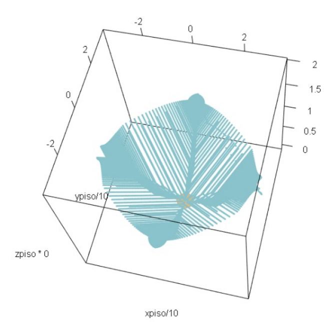

Figura 11: Solido con menos puntos.

para los dos casos ya que considerabamos que los puntos se cerraban casi en el mismo espacio las variables a comparar son matrices de puntos en x,y y z que conforman cada una de las figuras azules en la imagen.
Cada una de las figuras tiene sus propias matrices en x, en y, y en z. Entonces lo que hicimos fue comparar las posiciones en x de la primera figura con la de la segunda, las de y igual y las de z igual.

Se parte de la base de que para cada figura:

Los puntos en las matrices tienen un orden determinado

El criterio de comparación fue a traves de la ecuación de distancia entre puntos que se muestra a continuación:

Para $P_1 = (x_1, y_1)$ y $P_2 = (x_2, y_2)$ se tiene que

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Figura 12: Ecuación: distancia entre dos puntos.

Lo que hicimos fue tomar de a 100 puntos por cuadrante y comparar el punto 0 de la figura A con el punto 0 de la figura B. En un modelo ideal, la comparación entre ellos dos, es decir su distancia, debería de ser 0.
Estos fueron los resultados:

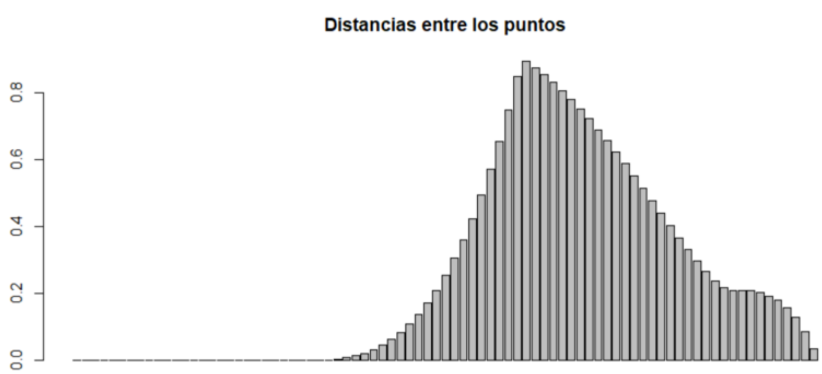

Figura 13: Grafica Errores: Punto vs Error

Esta se entiende como los resultados finales de las distancias entre los puntos tomando una muestra cada 100 puntos y comparandola con esa misma muestra de la otra figura. Lo que se muestra en la grafica anterior es el calculo de error de un solo cuadrante con 100 puntos representativos de los 10000.
Si volvemos a ver en la *Figura 4* podemos ver que los puntos son mucho más exactos hacia los extremos del mortero, lo que explicaría el bajo error en los extremos de la Figura 13. Pero en los puntos medios el error aumenta porque estan más dispersos los puntos y ,de paso si se le restaron puntos para una figura diferente, pues la cantidad de error va a aumentar en esa zona.