

Pontificia Universidad
JAVERIANA
Bogotá

Pontificia Universidad Javeriana
Departamento de Matemáticas
Análisis Numérico

Interpolación de un mortero

Reto Segundo Corte
por

Monica Alejandra Alvarez Carrillo
monica_alvarez@javeriana.edu.co

Santiago Palacios Loaiza
palacios-santiago@javeriana.edu.co

Paula Catalina Piñeros Pardo
pineros.paula@javeriana.edu.co

Profesora: Eddy Herrera Daza

Bogotá D.C., 19 de Abril del 2020

1. Problema Planteado

En algunas regiones de España es muy común que en las cocinas de los hogares exista un mortero como el de la figura 1. Un mortero que se caracteriza por ser de color amarillo y esta decorado con manchas verdes, es la herramienta perfecta para la elaboración de salsas tradicionales que requieran de la tritución y mezcla de especias con otros elementos



Figura 1: Mortero Valenciano con su respectiva mano de mortero

Esta tradicional figura ha ido mas allá de la cocina para llegar al mundo de la interpolación, esto debido a sus características como superficie que comparten similitudes con una esfera pero también tiene suficientes modificaciones para presentar un problema de interpolación con una complejidad retadora e interesante. Precisamente ésta es la problemática que el presente documento pretende resolver inicialmente, sin embargo hay un añadido. Las curvas de Bezier son un instrumento matemático para la modelización de curvas y superficies. Este planteamiento se desarrolló alrededor de 1960 para el trazado de dibujos técnicos y es ampliamente usado en el diseño aeronáutico y automovilístico. Por su forma de crear una descripción matemática de curvas, Bezier es también una técnica común de interpolación y es por esto que es el añadido del problema.

Para resumir, el objetivo planteado es lograr recrear un mortero valenciano usando curvas de Bezier y con Splines.

2. Solución planteada

La idea principal para la solución del problema es relativamente sencilla, hay que aplicar funciones de Interpolación, tanto de Splines como con Curvas de Bezier, a una serie de puntos que representen la superficie objetivo, en ese caso un mortero. Por lo tanto el primer paso es encontrar un conjunto de puntos que representen vértices claves de la superficie del mortero, estos puntos deben ser organizados bajo algún criterio que facilite los siguientes pasos, los criterios usados en esta solución se explican mas adelante. Teniendo un conjunto de puntos ya definidos y organizados se procede a aplicar funciones de Interpolación sobre los puntos, esto sin tener en cuenta la dimensión Z . Los puntos de los resultados de las interpolaciones son los puntos que se van a graficar de manera tridimensional para finalmente lograr dibujar el mortero. El componente Z se agrega de manera manual, teniendo en cuenta el modelo del 3D y su crecimiento en el mismo componente anterior. La secuencia de pasos realizada para obtener la solución al problema se muestra a continuación.

2.1. Modelado del mortero

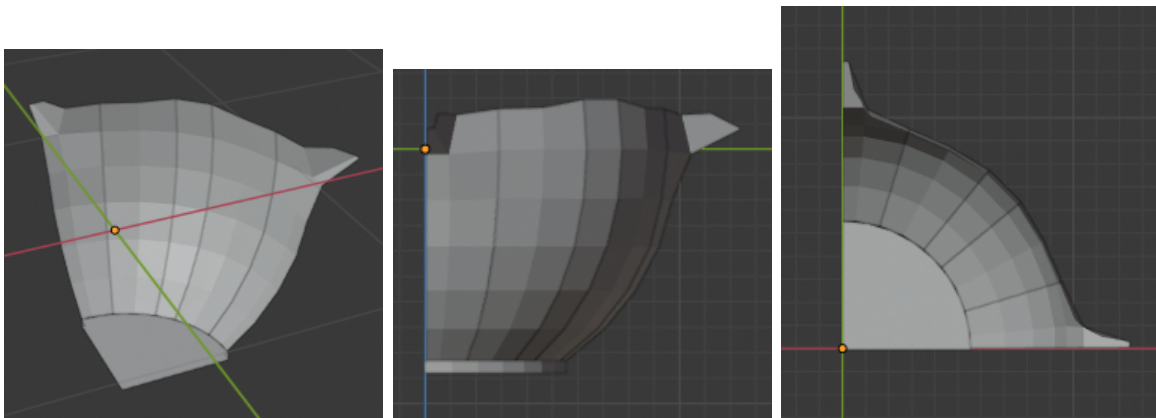
El primer inconveniente a la hora de pensar en la solución del problema planteado, es hallar una aproximación a los puntos que representan un mortero valenciano en el espacio tridimensional. Para solucionar este problema de la mejor manera, el equipo decidió realizar un modelo 3D en la herramienta Blender, una de las más famosas y más usadas en éste ámbito.

2.2. Obtención de los puntos

Para hallar los puntos se hizo uso del software Blender, el cual nos permite modelar objetos en 3D. Con las herramientas brindadas por el software y basándose en la (Figura 1, página 1) y en otras imágenes tomadas de internet se procedió a crear el objeto. Como únicos objetos se agregó una esfera y un cilindro, los cuales fueron cortados y modificados hasta acomodarse lo más posible al mortero. Esto se pudo lograr con acciones de Blender tales como: corte, transformar y deformación elástica. A continuación se muestran diferentes tomas del objeto modelado.



En este caso, una de las ventajas es la simetría del objeto. Es por ello, que éste se dividió en 4 partes y de esta forma facilitar el proceso de interpolación. A continuación se muestran las diferentes tomas del objeto que se uso para las interpolaciones.



El programa usado también nos permite exportar el modelo a un archivo del tipo objeto (.obj), el cual puede ser visualizado con cualquier editor de texto. Estos archivos se caracterizan por brindar un conjunto de datos bastante amplio del modelo realizado, tales como:

- Las coordenadas x, y, z de los vértices del objeto.
- Las coordenadas UV de la textura.
- Los vectores normales en sus tres componentes.
- Los índices de vértices/textura/normal requeridos para formar los triángulos que componen el modelo 3D.

Lo importante para llevar a cabo la interpolación del objeto en cuestión es la información del ítem 1, es decir, las coordenadas de los vértices que serán posteriormente organizados a nuestra conveniencia. Este orden se determinó según la distancia euclidiana entre los puntos; primero se parte de un punto inicial cualquiera, el siguiente punto será aquel punto cuya distancia euclidiana entre ellos sea menor y el siguiente punto igual, así hasta que todos los puntos queden ordenados. Lo siguiente es la lista de coordenadas de los vértices del mortero generado en Blender:

```

1
2 0.757972 0.195090 -0.622428
3 0.622771 0.195090 -0.757691
4
5 0.862122 0.162575 -0.461142
6 0.461532 0.162575 -0.861914
7
8 0.938587 0.158191 -0.287489
9 0.285035 0.158191 -0.939459
10
11 1.007241 0.134410 -0.123978
12 0.120758 0.134410 -1.008098
13
14 1.123987 0.140648 -0.050785
15 0.046877 0.140648 -1.124318
16
17 1.242845 0.080604 -0.029849
18 1.242894 0.080604 -0.004606
19 0.025441 0.080604 -1.242854
20 0.000199 0.080604 -1.242903
21
22 0.772822 0.000000 -0.634622
23 0.873309 0.000000 -0.467126
24 0.467521 0.000000 -0.873099
25 0.634972 0.000000 -0.772535
26 0.288205 -0.005570 -0.951784
27 0.951098 -0.005570 -0.290465
28 1.044185 -0.019903 -0.091954
29 1.044512 -0.019903 -0.003439
30 0.089114 -0.019903 -1.044502
31 0.000599 -0.019903 -1.044518
32

```

33	0.000723	-0.199724	-0.970937
34	0.095934	-0.199724	-0.970866
35	0.282667	-0.194153	-0.92932
36	0.458538	-0.194153	-0.856322
37	0.622771	-0.195090	-0.757691
38	0.757972	-0.195090	-0.622428
39	0.856529	-0.194153	-0.458151
40	0.928652	-0.194153	-0.284868
41	0.970636	-0.199724	-0.098240
42	0.970933	-0.199724	-0.003030
43			
44	0.000688	-0.382683	-0.919430
45	0.091244	-0.382683	-0.919362
46	0.268849	-0.382683	-0.883896
47	0.436123	-0.382683	-0.814463
48	0.586637	-0.382683	-0.713730
49	0.713995	-0.382683	-0.586314
50	0.814660	-0.382683	-0.435755
51	0.883257	-0.382683	-0.270943
52	0.919143	-0.382683	-0.093422
53	0.919426	-0.382683	-0.002867
54			
55	0.000619	-0.555570	-0.827465
56	0.082118	-0.555570	-0.827404
57	0.241958	-0.555570	-0.795485
58	0.392501	-0.555570	-0.732997
59	0.527960	-0.555570	-0.642340
60	0.642578	-0.555570	-0.527669
61	0.733174	-0.555570	-0.392169
62	0.794910	-0.555570	-0.243842
63	0.827207	-0.555570	-0.084077
64	0.827461	-0.555570	-0.002580
65			
66	0.000527	-0.707107	-0.703701
67	0.069835	-0.707107	-0.703649
68	0.205769	-0.707107	-0.676505
69	0.333794	-0.707107	-0.623363
70	0.448993	-0.707107	-0.546265
71	0.546468	-0.707107	-0.448746
72	0.623513	-0.707107	-0.333512
73	0.676015	-0.707107	-0.207371
74	0.703482	-0.707107	-0.071502
75	0.703698	-0.707107	-0.002194
76			
77	0.531142	-0.831470	-0.162930
78	0.552892	-0.831470	-0.001724
79	0.054869	-0.831470	-0.552853

```

80  0.262261 -0.831470 -0.489773
81  0.161671 -0.831470 -0.531527
82  0.000414 -0.831470 -0.552894
83  0.352771 -0.831470 -0.429198
84  0.489891 -0.831470 -0.262039
85  0.429357 -0.831470 -0.352577
86  -0.00089 -0.831470  0.00362
87  0.552722 -0.831470 -0.056179
88
89  0.000534 -0.879887 -0.551603
90  0.108912 -0.879887 -0.540928
91  0.213125 -0.879887 -0.509313
92  0.309168 -0.879887 -0.457973
93  0.393350 -0.879887 -0.388881
94  0.462437 -0.879887 -0.304692
95  0.513773 -0.879887 -0.208642
96  0.545386 -0.879887 -0.104422
97  0.556060 -0.879887  0.003964
98  0.555178 -0.879887 -0.004986
99  0.000534 -0.879887  0.003757

```

3. Curvas de Bezier

3.1. Teoría

Una curva de Bezier es una curva suave paramétrica generada a partir de dos puntos finales y uno o más puntos de control, dichos puntos no necesariamente caen en pasan por la curva, pero son necesarios para calcular la trayectoria de la dicha curva. Existen diversas curvas de Bezier, entre éstas se encuentran, las curvas lineales, las curvas cuadráticas y las curvas cúbicas.

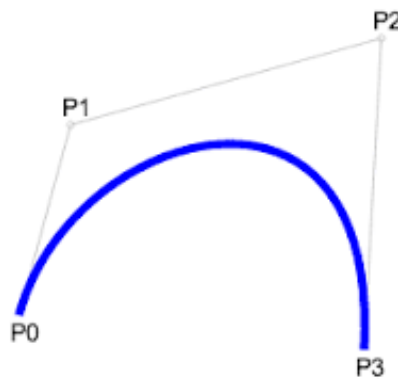


Figura 2: Curva de Bezier

Las curvas más comunes son las cúbicas que están dadas por cuatro puntos que las definen, donde existe un punto inicial, un punto final, y dos puntos de control que brindan información

direccional de la curva, es por ello que estos puntos jamás pasan por la ésta. La forma paramétrica de la curva cúbica de Bezier está definida como:

$$B(t) = P_0(1 - t)^3 + 3P_1t(1 - t)^2 + 3P_2t^3, t \in [0, 1]$$

3.2. Solución

Para la solución del problema por el método de curvas de Bezier el primer paso fue hacer una pequeña modificación a los puntos creados, mediante la creación de nuevos puntos. El proceso de creación de nuevos puntos esta definido de la siguiente manera:

Sea un punto (x_i, y_i) y un punto siguiente $(x_j, y_j); j = i + 1$ entonces se puede hallar un nuevo punto (x_p, y_p) de la siguiente manera:

$$(x_p, y_p) = \frac{(x_i, y_i) + (x_j, y_j)}{2}$$

La creación de estos nuevos puntos es necesaria para ayudar a formar curvas de Bezier que representen correctamente la superficie objetivo. Esto es importante para este método y no para el de Splines debido a que Bezier usa los puntos a manera de guía, pero sin tocarlos, esto se explica a continuación:.

Los puntos recibidos cuando se trabajan con Curvas de Bezier se podrían clasificar en dos tipos. La primera clasificación son los puntos de inicio y fin, los únicos que Bezier intercepta en la creación de su curva. La segunda clasificación son los puntos de control, Bezier usa los puntos de control como referencias para determinar la pendiente tangente de la línea. Por ejemplo:

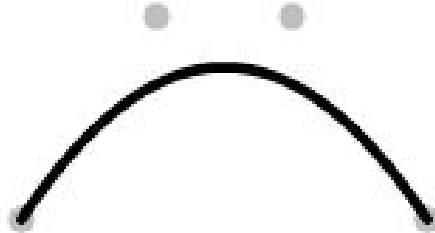


Figura 3: Ejemplo de una Curva de Bezier

Numerando los puntos de izquierda a derecha en la imagen anterior el primer y el cuarto punto cumplen la función de inicial y final, luego el segundo y tercer punto cumplen una función de guía para la pendiente de la tangente de la curva. Es importante notar que la curva se acerca pero nunca toca ningún punto de control a menos que sea el inicial o el final

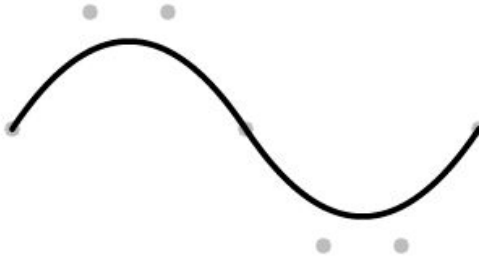


Figura 4: Ejemplo de dos Curvas de Bezier

Para ver otro ejemplo, en esta ocasión la figura esta siendo creada por 2 curvas diferentes. Donde el cuarto punto cumple la función de ser el punto final de la primera curva y al mismo tiempo el inicial de la primera.

Lo siguiente es el conjunto de puntos completo para ser utilizado mediante curvas de Bezier

```

1  y1 = c(-0.551603, -0.546265, -0.540928, -0.457973,
2      -0.4233915, -0.388881, -0.246651, -0.104422, 0.003964)
3  x1 = c(0.0053400, 0.057126, 0.108912, 0.309168, 0.351259,
4      0.393350, 0.469368, 0.545386, 0.556060)
5  y2 = c(-0.552894, -0.542210, -0.531527, -0.489773, -0.421175,
6      -0.352577, -0.257753, -0.162930, -0.001724)
7  x2 = c(0.000414, 0.081042, 0.161671, 0.262261, 0.345809,
8      0.429357, 0.480249, 0.531141, 0.552892)
9  y3 = c(-0.703701, -0.676505, -0.623363, -0.546265,
10     -0.439888, -0.333512, -0.207371, -0.1047825, -0.002194)
11 x3 = c(0.000527, 0.205769, 0.333794, 0.448993, 0.536253,
12     0.623513, 0.676015, 0.6898565, 0.703698)
13 y4 = c(-0.827465, -0.827404, -0.795485, -0.732997, -0.642340,
14     -0.527669, -0.392169, -0.243842, -0.002580)
15 x4 = c(0.000619, 0.082118, 0.241958, 0.392501, 0.527960,
16     0.642578, 0.733174, 0.794910, 0.827461)
17 y5 = c(-0.919430, -0.883896, -0.814463, -0.713730, -0.650022,
18     -0.586314, -0.435755, -0.270943, -0.002867)
19 x5 = c(0.000688, 0.268849, 0.436123, 0.586637, 0.650316,
20     0.713995, 0.814660, 0.883257, 0.919426)
21 y6 = c(-0.970937, -0.929325, -0.856322, -0.757691,
22     -0.690059, -0.622428, -0.458151, -0.284868, -0.003030)
23 x6 = c(0.000723, 0.282667, 0.458538, 0.622771, 0.690371, 0.757972,
24     0.856529, 0.928652, 0.970933)
25 y7 = c(-1.044518, -1.044502, -0.9888185, -0.880676, -0.772535,
26     -0.634622, -0.467126, -0.290465, -0.003439)
27 x7 = c(0.000599, 0.089114, 0.3620430, 0.498507, 0.634972,
28     0.772822, 0.873309, 0.951098, 1.044512)
29 y8 = c(-1.242903, -1.242854, -1.124318, -1.008098, -0.939459,
30     -0.861914, -0.757691, -0.622428, -0.461142, -0.287489, -0.029849,

```

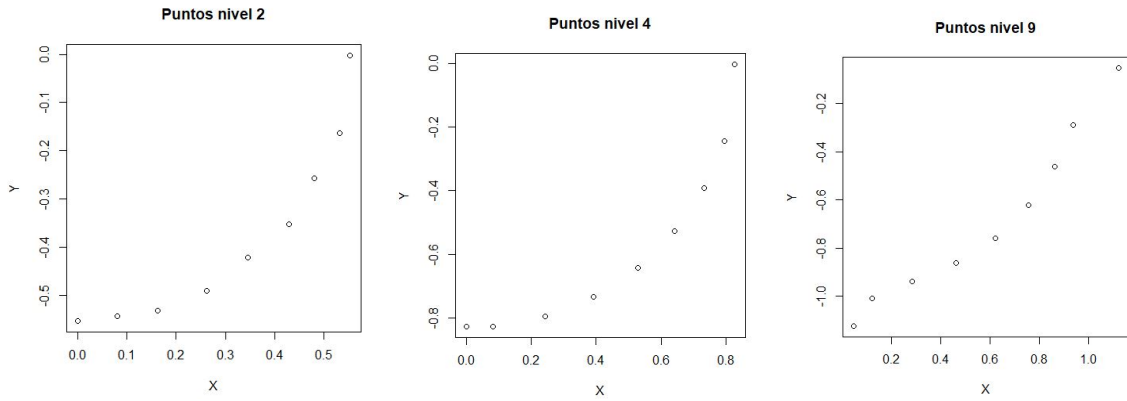


```

16      -0.004606)
      x8 = c( 0.000199,  0.025441,  0.046877,  0.120758,  0.285035,
              0.461532,  0.622771,  0.757972,  0.862122,  0.938587,  1.242845,
              1.242894)
17      y9 = c(-1.124318, -1.008098, -0.939459, -0.861914, -0.757691,
              -0.622428, -0.461142, -0.287489, -0.050785)
18      x9 = c( 0.046877,  0.120758,  0.285035,  0.461532,  0.622771,
              0.757972,  0.862122,  0.938587,  1.123987)

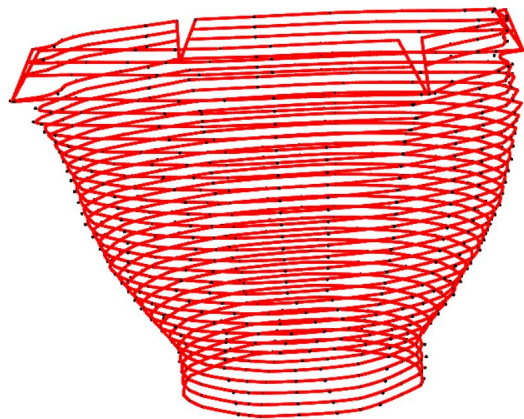
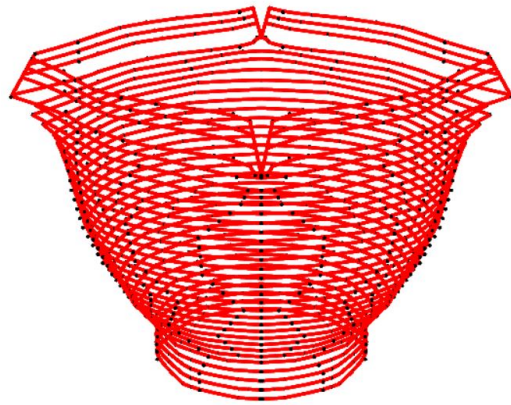
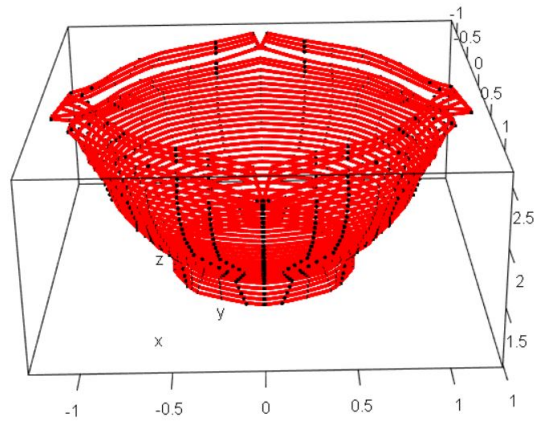
```

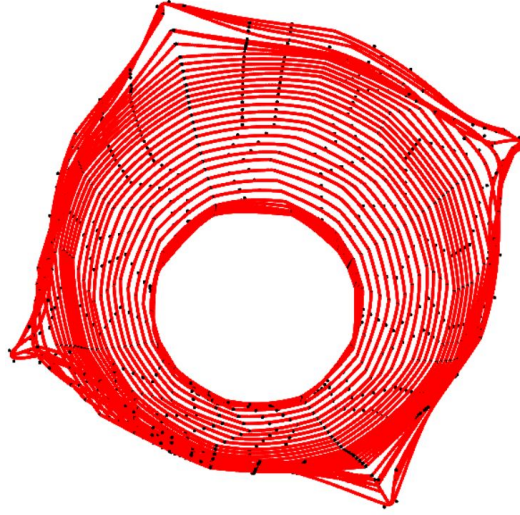
Tras tener una lista con suficientes puntos organizados de puntos se procede a realizar finalmente la interpolación mediante el método de Curvas de Bezier. El algoritmo comienza creando una matriz para cada nivel de Z , donde la primera columna representa las coordenadas de los puntos en X , la segunda las coordenadas en Y y la tercera las coordenadas en Z . Esta ultima tiene un mismo valor durante todas las filas de la tercera columna que es el valor correspondiente al nivel de Z de la matriz. A continuación se muestran graficas de la representación de las primeras columnas de las matrices de nivel 2,4 y 9.



Figuras 5,6,7: Gráficas (X,Y) de niveles de Z

Estas gráficas son también útiles para aclarar que la información hasta ahora consignada pertenece a solo un cuadrante del plano del problema, todos los pasos que incluyen graficación mas adelante alteran los ejes de los puntos anteriores, que se asemejan a semicírculos, para lograr la forma cilíndrica deseada en el mortero. Después de tener creada esta matriz con las coordenadas de todos los puntos de un nivel se invoca la función $bezier(t,p)$ que recibe como parámetros; t una secuencia de números entre 0 y 1, la cantidad de números es posteriormente la cantidad de puntos que retorna la función y p que es la matriz descrita anteriormente. El resultado de estas funciones es una cantidad de puntos n que representan la curva de bezier, son estos puntos los que se grafican para crear la superficie. El procedimiento anterior se realiza una vez por cada nivel de Z que tenemos del modelo. A continuación se muestran los resultados, es de resaltar que por fines estéticos se crearon mas niveles de una manera artificial, esto se hizo promediando un nivel con el nivel consecutivo con la intención de obtener una superficie mas completa.





Figuras 8,9,10,11: Perspectivas del Mortero, Bezier

La ultima imagen, la correspondiente a la figura 11 es especialmente interesante, puesto que, por su naturaleza de perspectiva cenital permite observar claramente en la boquilla del mortero como la curva de Bezier(rojo) se aproxima al punto guía(negro) y lo usa para guiarse sin tocarlo. Estas imágenes son entonces suficiente evidencia para concluir que se logra el objetivo de dibujar un mortero valenciano mediante interpolación de curvas de Bezier.

3.3. Errores obtenidos

La siguiente tabla muestra los errores obtenidos al interpolar las coordenadas x con las curvas de Bezier.

Teórico	Experimental	Absoluto	Relativo %
0.046877	0.0468770	0.000000000	0.0000000
0.120758	0.1538632	0.033105246	27.4145369
0.285035	0.2971901	0.012155133	4.2644353
0.461532	0.4509225	-0.010609548	2.2987676
0.622771	0.5990057	-0.023765305	3.8160583
0.757972	0.7334065	-0.024565527	3.2409545
0.862122	0.8545043	-0.007617704	0.8835993
0.938587	0.9743960	0.035809043	3.8152077
1.123987	1.1239870	0.000000000	0.0000000
*****	*****	*****	*****
0.000688	0.0006880	0.000000000	0.0000000
0.268849	0.2297922	-0.039056836	14.527425
0.436123	0.4031284	-0.032994553	7.565424
0.586637	0.5348709	-0.051766071	8.824208
0.650316	0.6374238	-0.012892159	1.982445
0.713995	0.7233600	0.009364968	1.311629
0.814660	0.8009115	-0.013748518	1.687639
0.883257	0.8693363	-0.013920680	1.576062
0.919426	0.9194260	0.000000000	0.0000000

Tabla de resultados con curvas de Bezier para x_9 y x_5

A continuación, los errores obtenidos al interpolar los puntos y :

Teórico	Experimental	Absoluto	Relativo %
-1.124318	-1.1243180	0.000000000	0.0000000
-1.008098	-1.0237611	-0.015663110	1.5537289
-0.939459	-0.9346555	0.004803518	0.5113069
-0.861914	-0.8395031	0.022410916	2.6001337
-0.757691	-0.7284415	0.029249498	3.8603465
-0.622428	-0.5968119	0.025616127	4.1155165
-0.461142	-0.4426325	0.018509544	4.0138491
-0.287489	-0.2633059	0.024183076	8.4118265
-0.050785	-0.0507850	0.000000000	0.0000000
*****	*****	*****	*****
-0.919430	-0.9194300	0.000000000	0.0000000
-0.883896	-0.8695207	0.014375328	1.626360
-0.814463	-0.8009341	0.013528924	1.661085
-0.713730	-0.7232264	-0.009496368	1.330527
-0.650022	-0.6372650	0.012756994	1.962548
-0.586314	-0.5349138	0.051400175	8.766663
-0.435755	-0.4037159	0.032039099	7.352549
-0.270943	-0.2312662	0.039676809	14.643969
-0.002867	-0.0028670	0.000000000	0.0000000

Tabla de resultados con curvas de Bezier para y_9 y y_5

4. Splines

4.1. Teoría

La definición matemática de un Spline hace referencia a una curva diferenciable que está formada por pequeños segmentos de línea definidos por polinomios. La interpolación con Splines es comúnmente usada porque es muy precisa y se pueden obtener resultados satisfactorios con el uso de simples polinomios. Igual que en el caso de las curvas de Bezier, los Splines pueden ser lineales, cuadráticos y cúbicos. En la interpolación segmentaria cuadrática, los splines se construyen a través de polinomios de grado 2:

$$P(x) = ax^2 + bx + c$$

4.2. Solución

Para la solución del problema usando Splines, a diferencia del anterior método se dejaron intactos los puntos obtenidos del modelo de Blender. La diferencia parte en que tuvo que dividirse dos veces, la primera dependía de su posición en el eje Z, y la otra separa este primer grupo en pequeños grupos de puntos que representan únicamente las coordenadas en X y en Y. Estos pequeños grupos de puntos fueron a los que se le aplicó el método de spline. Los puntos usados se listan a continuación:

```

1  y1 = c(-0.551603, -0.540928, -0.457973, -0.388881, -0.104422,
2      0.003964)
3  x1 = c(0.00534, 0.108912, 0.309168, 0.393350, 0.545386,
4      0.556060)
5  y2 = c(-0.552894, -0.531527, -0.489773, -0.352577, -0.162930,
6      -0.001724)
7  x2 = c(0.000414, 0.161671, 0.262261, 0.429357, 0.531141,
8      0.552892)
9  y3 = c(-0.703701, -0.676505, -0.623363, -0.546265, -0.333512,
10     -0.207371, -0.002194)
11 x3 = c(0.000527, 0.205769, 0.333794, 0.448993, 0.623513,
12     0.676015, 0.703698)
13 y4 = c(-0.827465, -0.827404, -0.795485, -0.732997, -0.642340,
14     -0.527669, -0.392169, -0.243842, -0.002580)
15 x4 = c(0.000619, 0.082118, 0.241958, 0.392501, 0.527960,
16     0.642578, 0.733174, 0.794910, 0.827461)
17 y5 = c(-0.919430, -0.883896, -0.814463, -0.713730, -0.586314,
18     -0.435755, -0.270943, -0.002867)
19 x5 = c(0.000688, 0.268849, 0.436123, 0.586637, 0.713995,
20     0.814660, 0.883257, 0.919426)
21 y6 = c(-0.970937, -0.929325, -0.856322, -0.757691, -0.622428,
22     -0.458151, -0.284868, -0.003030)
23 x6 = c(0.000723, 0.282667, 0.458538, 0.622771, 0.757972,
24     0.856529, 0.928652, 0.970933)
25 y7 = c(-1.044518, -1.044502, -0.9888185, -0.772535, -0.634622,
26     -0.467126, -0.290465, -0.003439)
27 x7 = c(0.000599, 0.089114, 0.3620430, 0.634972, 0.772822,
28     0.873309, 0.951098, 1.044512)
29 y8 = c(-1.242903, -1.124318, -0.939459, -0.861914, -0.757691,
30     -0.622428, -0.461142, -0.287489, -0.004606)
31 x8 = c(0.000199, 0.046877, 0.285035, 0.461532, 0.622771,
32     0.757972, 0.862122, 0.938587, 1.242894)
33 y9 = c(-1.124318, -0.939459, -0.861914, -0.757691, -0.622428,
34     -0.461142, -0.287489, -0.050785)
35 x9 = c(0.046877, 0.285035, 0.461532, 0.622771, 0.757972,
36     0.862122, 0.938587, 1.123987)
37 y10 = c(-0.004606, -0.050785)
38 x10 = c(1.242894, 1.123987)
39 y11 = c(-1.242903, -1.124318)
40 x11 = c(0.000199, 0.046877)

```

Los valores de Z fueron los siguientes:

```

1  z1 = -0.879887
2  z2 = -0.831470
3  z3 = -0.707107
4  z4 = -0.555570

```

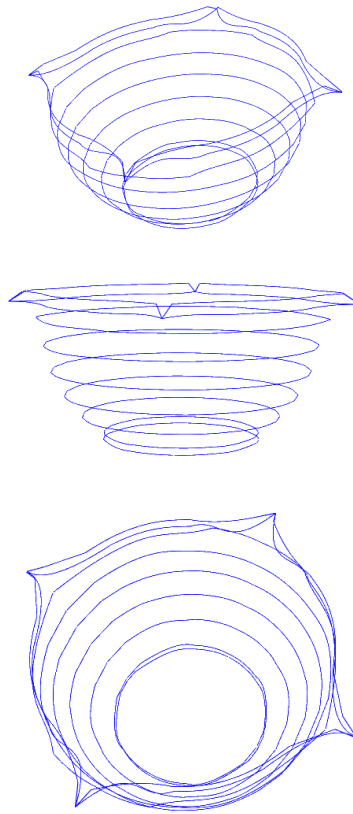
```

5      z5 = -0.382683
6      z6 = -0.199724
7      z7 = -0.019903
8      z8 =  0.08604
9      z9 =  0.162575

```

Como se explicó anteriormente cada x_n y y_n fue separado en pequeños intervalos a los cuales se les aplicó una función que ya tiene implementada R, la cual se llama *spline* y cuyos parámetros son un arreglo de X, un arreglo de Y y un valor de N (el número de puntos que se requiere que calcule la función) que en esta solución se uso siempre $n = 201$. dicha función retorna dos listas, una que representa las coordenadas de los 201 puntos en X, y la otra que representa los las coordenadas en y.

Al nosotros ya tener todos estos puntos guardados en otras variables se procedió a graficar dichos puntos obtenidos de los splines y los puntos de las diferentes alturas en Z . Para lograr esto se uso una función propia de R que es *plot3D*, cuyos parámetros son los valores de X, Y y Z. A parte se especifica el tamaño, el color de la gráfica y por ultimo se añade el argumento `.add = TRUE`. Al momento de graficar se tuvo que tener en cuenta que los puntos con los que contabamos únicamente representaban 1/4 parte del mortero. Por lo que gracias a la simetría del objeto simplemente se multiplicaron por -1 dichos puntos, obteniendo así el otro lado de del mortero, y para obtener los lados se multiplico solo x por -1, y para el último lado y. Así fue como se obtuvo la gráfica del mortero. A continuación se presentan algunas tomas del resultado.



Figuras 12,13,14: Perspectivas del Mortero, Splines

4.3. Errores obtenidos

A diferencia del otro método, en este caso los splines tocan el punto exacto que se le brindo, mientras que en Bezier, los puntos eran usados de guía. Es por eso que en método los errores son menores y también es por eso que los errores son tan pequeños que si se miran las mismas décimas de los puntos del spline que las decimas de los puntos dados parecieran el mismo punto. A continuación se presentan algunos los errores obtenidos usando el método de splines.

Teórico	Experimental	Absoluto	Relativo %
-0.001724	-0.001724	3.252607×10^{-18}	1.886663×10^{-13}
-0.002194	-0.002194	1.301043×10^{-18}	5.930003×10^{-14}
-0.00258	-0.00258	8.673617×10^{-19}	3.361867^{-14}
-0.002867	-0.002867	8.239937×10^{-18}	2.874062^{-13}
-0.00303	-0.00303	2.298509×10^{-17}	-7.585837^{-13}

Tabla de resultados con Splines

5. Conclusiones

Gracias al desarrollo de esta implementación se pudo comprender mejor los dos métodos puestos en cuestión, los tipos de curvas que existen en ambos casos y las diferentes aplicaciones. El cálculo de errores obtenidos en ambas implementaciones nos permite asegurar que realizar la interpolación por el método de Splines es mucho más precisa que realizar el método de curvas de Bezier, una de las razones se debe a la definición de los splines, ya que se trabaja con pequeños segmentos definidos por polinomios de bajo grado, esto garantiza mayor control en la formación de la curva. Mientras que el método de Bezier, al usar los puntos de control se puede ver viciada la dirección de la curva al trazarla.

