

Reto Parcial 2 Análisis Numérico

Gabriel Andrés Niño Carvajal - Juliana García Mogollón

19 de abril de 2020

1 Reto de Interpolación

El objetivo propuesto es conseguir dibujar el mortero valenciano (Figure 1) usando superficies de Bezier y otro método (BSplines). Para ello se puede utilizar R (PathInterpolatR, gridBezier,vwline) o Python (griddata, matplotlib).



Figure 1: Mortero Valenciano

Se sugiere dividir la figura en cuatro cuadrantes, de manera que una vez construido uno, el resto puede representarse realizando rotaciones por ejemplo. Tenga en cuenta que la figura no puede representarse mediante una única superficie hay que dividirla de la manera eficiente. Tenga en cuenta que las zonas afiladas. Para el caso de superficies la derivada es obviamente direccional, pero la idea es la misma.

2 Solución Mortero Sencillo

De acuerdo con las instrucciones de la profesora, se eligió un nuevo mortero que se muestra en Figure 2.



Figure 2: Mortero Sencillo

Se busca graficar un mortero sencillo, ya que el mortero valenciano es más difícil de graficar debido a los picos característicos que maneja su figura. A partir de la imagen y utilizando el programa Paint 3D se dibujo un "esqueleto" de uno de los cuadrantes del mortero, y se utilizo el programa Paint para dibujar un sistema de coordenadas como se muestra en Figure 3:

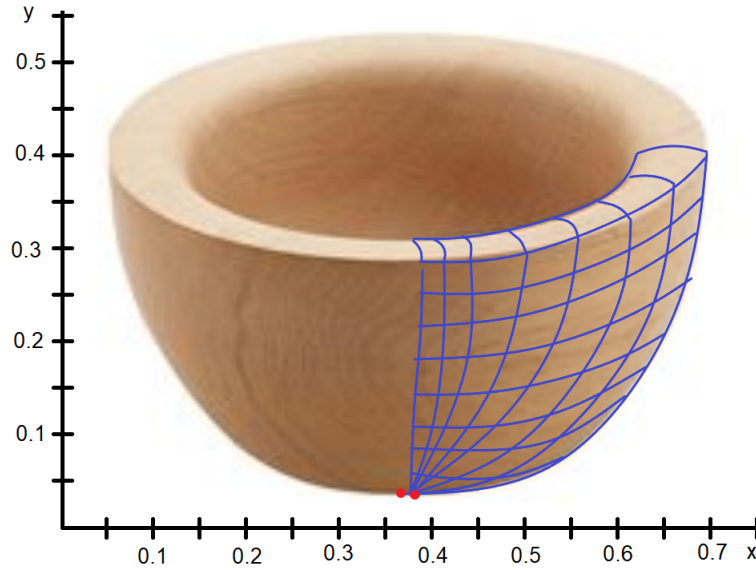


Figure 3: Coordenadas Mortero

El rango de x es $[0, 0.75]$ y el rango de y es $[0, 0.55]$, se manejaron puntos que estuviesen por debajo de 1 debido a que la función de Bezier que se utilizó no gráfica más allá de 1. A partir de Figure 3 se realizaron las mediciones correspondientes para graficar el mortero.

Para que la figura fuera lo más exacta posible a la imagen original, los puntos se midieron haciendo una regla de tres con las coordenadas de los pixeles de la imagen, esto puede determinarse gracias al programa Paint que, dependiendo de donde esté ubicado el cursor, muestra las coordenadas en pixeles. Se utilizaron las siguientes fórmulas de conversión:

$$x = (pixel_x - 142) * 0.75 / 865$$

$$y = (677 - pixel_y) * 0.55 / 638$$

Donde (x, y) representa un punto de la imagen.

2.1 Solución por Splines de Bezier

Primero se midieron los puntos más importantes para la figura, los puntos que tienen a y b en su nombre son los puntos intermedios que ayudan a formar el polígono de control de cada curva de Bezier.

$Pixel_x$	$Pixel_y$	Punto	x	y
580	636	Base	0.379768786	0.035344828
945	209	p1	0.696242775	0.403448276
794	656	p1a	0.565317919	0.018103448
941	512	p1b	0.692774566	0.142241379
588	346	p8	0.386705202	0.285344828
712	357	p8-1a	0.494219653	0.275862069
912	273	p8-1b	0.667630058	0.348275862
859	212	p9	0.621676301	0.400862069
580	317	p16	0.379768786	0.310344828
689	337	p16-9a	0.474277457	0.293103448
849	258	p16-9b	0.61300578	0.361206897

Table 1: Puntos Bezier Sencillos

En Figure 4 se muestran los polígonos de control:

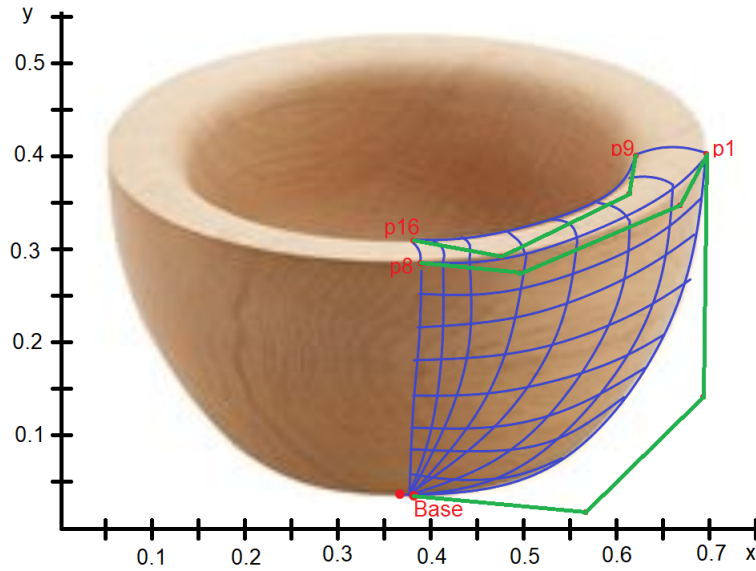


Figure 4: Mortero con Polígonos

Estos puntos representan sólo un cuadrante del mortero, de acuerdo con la sugerencia de la profesora, se utilizó el punto Base para hacer la rotación en x

y se utilizo el punto p1 para hacer la rotación en y .
 Teniendo en cuenta todo lo anterior, se escribió un programa en R para graficar el mortero, se utilizó la libreria gridBezier.

```
rm(list=ls())
#Splines de Bezier para graficar el Mortero
options(digits = 16)
library(grid)
library(gridBezier)
library(vwline)
grid.newpage()
#Punto Base-----
xbase<-0.379768786
ybase<-0.035344828
#Punto p1
xp1<-0.696242775
yp1<-0.403448276
#Lineas Verticales -----
#Base a p1-----
x<-c(xbase,0.565317919,0.692774566,xp1)
y<-c(ybase,0.018103448,0.142241379,yp1)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(xbase,2*xbase-x[2],2*xbase-x[3],2*xbase-x[4])
grid.Bezier(xrx,y)
#-----
#Borde superior-----
#p8 a p1-----
x<-c(0.386705202,0.494219653,0.667630058,xp1)
y<-c(0.285344828,0.275862069,0.348275862,yp1)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],2*xbase-x[4])
grid.Bezier(xrx,y)
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3],2*yp1-y[4])
grid.Bezier(x,yry)
#Reflejo xy
grid.Bezier(xrx,yry)
#p16 a p9-----
x<-c(0.379768786,0.474277457,0.61300578,0.621676301)
y<-c(0.310344828,0.293103448,0.361206897,0.400862069)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],2*xbase-x[4])
```

```

grid.Bezier(xrx,y)
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3],2*yp1-y[4])
grid.Bezier(x,yry)
#Reflejo xy
grid.Bezier(xrx,yry)
#_-----

```

La gráfica resultante fue la que se muestra en Figure 5:

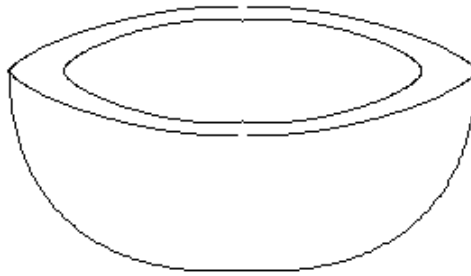


Figure 5: Mortero Sencillo

2.2 Mortero Splines Normales

Para este caso fue necesario utilizar más puntos:

$Pixel_x$	$Pixel_y$	Punto	x	y
580	636	Base	0.379768786	0.035344828
945	209	p1	0.696242775	0.403448276
903	254	p2	0.65982659	0.364655172
847	289	p3	0.611271676	0.334482759
784	316	p4	0.556647399	0.311206897
721	336	p5	0.502023121	0.293965517
652	346	p6	0.442196532	0.285344828
617	347	p7	0.411849711	0.284482759
588	346	p8	0.386705202	0.285344828
859	212	p9	0.621676301	0.400862069
847	240	p10	0.611271676	0.376724138
807	270	p11	0.576589595	0.350862069
755	291	p12	0.53150289	0.332758621
698	306	p13	0.482080925	0.319827586
643	315	p14	0.434393064	0.312068966
606	317	p15	0.402312139	0.310344828
580	317	p16	0.379768786	0.310344828
940	264	p17	0.691907514	0.356034483
934	309	p18	0.686705202	0.317241379
917	373	p19	0.671965318	0.262068966
891	437	p20	0.649421965	0.206896552
870	478	p21	0.631213873	0.171551724
844	515	p22	0.60867052	0.139655172
765	591	p23	0.54017341	0.074137931

Table 2: Puntos Mortero Spline

Los puntos utilizados se muestran Figure 6, se utilizó hasta el punto p23, los 7 puntos restantes no se utilizaron en esta solución:

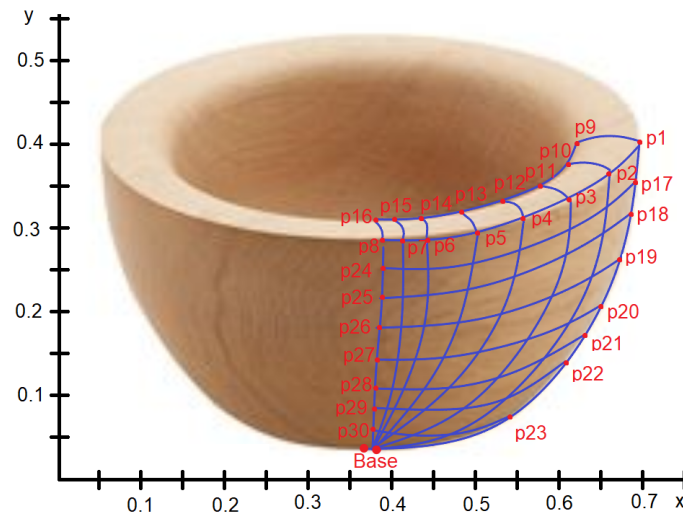


Figure 6: Mortero Sencillo

Nuevamente se hizo un programa en el lenguaje R para graficar el mortero pero esta vez utilizando el método Splines:

```
rm(list=ls())
options(digits=16)
#Interpolacion Mortero por Splines
#x-----
puntosx<-c(0.379768786,0.696242775,0.65982659
,0.611271676,0.556647399,0.502023121
,0.442196532,0.411849711,0.386705202
,0.621676301,0.611271676,0.576589595
,0.53150289,0.482080925,0.434393064
,0.402312139,0.379768786,0.691907514
,0.686705202,0.671965318,0.649421965
,0.631213873,0.60867052,0.54017341)

puntosy<-c(0.035344828,0.403448276,0.364655172
,0.334482759,0.311206897,0.293965517
,0.285344828,0.284482759,0.285344828
,0.400862069,0.376724138,0.350862069
,0.332758621,0.319827586,0.312068966
,0.310344828,0.310344828,0.356034483
,0.317241379,0.262068966,0.206896552
,0.171551724,0.139655172,0.074137931)
```



```

plot(puntosx,puntosy ,main=paste("Mortero"),xlim=c(0,0.75)
,ylim=c(0,0.55),col="red")
#Punto Base-----
xbase<-0.379768786
ybase<-0.035344828
#Punto p1
xp1<-0.696242775
yp1<-0.403448276
#Lineas Verticales -----
#Base a p1-----
x<-c(xbase,puntosx[18],puntosx[19],puntosx[20]
,puntosx[21],puntosx[22],puntosx[23],puntosx[24],xp1)
y<-c(ybase,puntosy[18],puntosy[19],puntosy[20]
,puntosy[21],puntosy[22],puntosy[23],puntosy[24],yp1)
lines(spline(x, y), col = "blue")
#Reflejo en x
xrx<-c(xbase,2*xbase-x[2],2*xbase-x[3],2*xbase-x[4]
,2*xbase-x[5],2*xbase-x[6],2*xbase-x[7],2*xbase-x[8]
,2*xbase-x[9])
lines(spline(xrx,y), col = "blue")
#-----
#Borde superior-----
#p8 a p1-----
x<-c(puntosx[2],puntosx[3],puntosx[4],puntosx[5]
,puntosx[6],puntosx[7],puntosx[8],puntosx[9])
y<-c(puntosy[2],puntosy[3],puntosy[4],puntosy[5]
,puntosy[6],puntosy[7],puntosy[8],puntosy[9])
lines(spline(x,y), col = "blue")
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3]
,2*xbase-x[4],2*xbase-x[5],2*xbase-x[6],2*xbase-x[7]
,2*xbase-x[8])
lines(spline(xrx,y), col = "blue")
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3],2*yp1-y[4]
,2*yp1-y[5],2*yp1-y[6],2*yp1-y[7],2*yp1-y[8])
lines(spline(x,yry), col = "blue")
#Reflejo xy
lines(spline(xrx,yry), col = "blue")
#p16 a p9-----
x<-c(puntosx[10],puntosx[11],puntosx[12],puntosx[13]
,puntosx[14],puntosx[15],puntosx[16],puntosx[17])
y<-c(puntosy[10],puntosy[11],puntosy[12],puntosy[13]
,puntosy[14],puntosy[15],puntosy[16],puntosy[17])
lines(spline(x,y), col = "blue")

```

```

#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3]
,2*xbase-x[4],2*xbase-x[5],2*xbase-x[6],2*xbase-x[7]
,2*xbase-x[8])
lines(spline(xrx,y), col = "blue")
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3],2*yp1-y[4]
,2*yp1-y[5],2*yp1-y[6],2*yp1-y[7],2*yp1-y[8])
lines(spline(x,yry), col = "blue")
#Reflejo xy
lines(spline(xrx,yry), col = "blue")

```

La gráfica resultante fue se muestra en Figure 7:

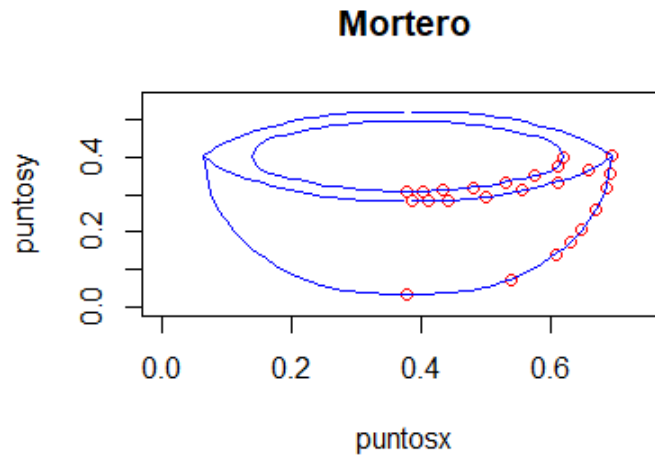


Figure 7: Mortero Splines

3 Solución Mortero con Volumen

Para graficar el mortero con volumen se necesitaron 31 puntos y 48 puntos intermedios que para formar los polígonos de control.

P_{ixel_x}	P_{ixel_y}	Punto	x	y
580	636	Base	0.379768786	0.035344828
945	209	p1	0.696242775	0.403448276
794	656	p1a	0.565317919	0.018103448
941	512	p1b	0.692774566	0.142241379
903	254	p2	0.65982659	0.364655172
746	620	p2a	0.523699422	0.049137931
905	425	p2b	0.661560694	0.217241379
847	289	p3	0.611271676	0.334482759
741	580	p3a	0.519364162	0.08362069
852	427	p3b	0.615606936	0.215517241
784	316	p4	0.556647399	0.311206897
720	529	p4a	0.501156069	0.127586207
780	437	p4b	0.553179191	0.206896552
721	336	p5	0.502023121	0.293965517
694	500	p5a	0.478612717	0.152586207
716	440	p5b	0.497687861	0.204310345
652	346	p6	0.442196532	0.285344828
646	518	p6a	0.43699422	0.137068966
662	444	p6b	0.450867052	0.200862069
617	347	p7	0.411849711	0.284482759
615	527	p7a	0.410115607	0.129310345
624	464	p7b	0.417919075	0.18362069
588	346	p8	0.386705202	0.285344828
712	357	p8-1a	0.494219653	0.275862069
912	273	p8-1b	0.667630058	0.348275862
593	336	p8-16a	0.391040462	0.293965517
590	324	p8-16b	0.388439306	0.304310345
859	212	p9	0.621676301	0.400862069
872	193	p9-1a	0.632947977	0.417241379
912	190	p9-1b	0.667630058	0.419827586
847	240	p10	0.611271676	0.376724138
863	228	p10-2a	0.625144509	0.387068966
899	233	p10-2b	0.656358382	0.382758621
807	270	p11	0.576589595	0.350862069
822	264	p11-3a	0.589595376	0.356034483
841	271	p11-3b	0.606069364	0.35
755	291	p12	0.53150289	0.332758621
776	288	p12-4a	0.549710983	0.335344828
788	299	p12-4b	0.560115607	0.325862069
698	306	p13	0.482080925	0.319827586
712	311	p13-5a	0.494219653	0.315517241
721	321	p13-5b	0.502023121	0.306896552
643	315	p14	0.434393064	0.312068966
650	317	p14-6a	0.440462428	0.310344828
654	328	p14-6b	0.443930636	0.300862069
606	317	p15	0.402312139	0.310344828
620	320	p15-7a	0.414450867	0.307758621
625	334	p15-7b	0.418786127	0.295689655

580	317	p16	0.379768786	0.310344828
689	337	p16-9a	0.474277457	0.293103448
849	258	p16-9b	0.61300578	0.361206897
940	264	p17	0.691907514	0.356034483
934	309	p18	0.686705202	0.317241379
917	373	p19	0.671965318	0.262068966
891	437	p20	0.649421965	0.206896552
870	478	p21	0.631213873	0.171551724
844	515	p22	0.60867052	0.139655172
765	591	p23	0.54017341	0.074137931
589	386	p24	0.387572254	0.250862069
678	400	p24-17a	0.464739884	0.238793103
931	277	p24-17b	0.684104046	0.344827586
588	424	p25	0.386705202	0.218103448
687	445	p25-18a	0.472543353	0.2
870	379	p25-18b	0.631213873	0.256896552
584	467	p26	0.383236994	0.181034483
646	486	p26-19a	0.43699422	0.164655172
878	420	p26-19b	0.638150289	0.221551724
582	511	p27	0.38150289	0.143103448
678	524	p27-20a	0.464739884	0.131896552
850	476	p27-20b	0.613872832	0.173275862
579	551	p28	0.378901734	0.10862069
654	566	p28-21a	0.443930636	0.095689655
779	539	p28-21b	0.552312139	0.118965517
578	580	p29	0.378034682	0.08362069
715	588	p29-22a	0.496820809	0.076724138
776	568	p29-22b	0.549710983	0.093965517
575	607	p30	0.375433526	0.060344828
614	623	p30-23a	0.409248555	0.046551724
702	623	p30-23b	0.485549133	0.046551724

Table 3: Puntos Mortero Bezier con Volumen

En este caso si se utilizarón todos los puntos que se muestran en Figure 8:

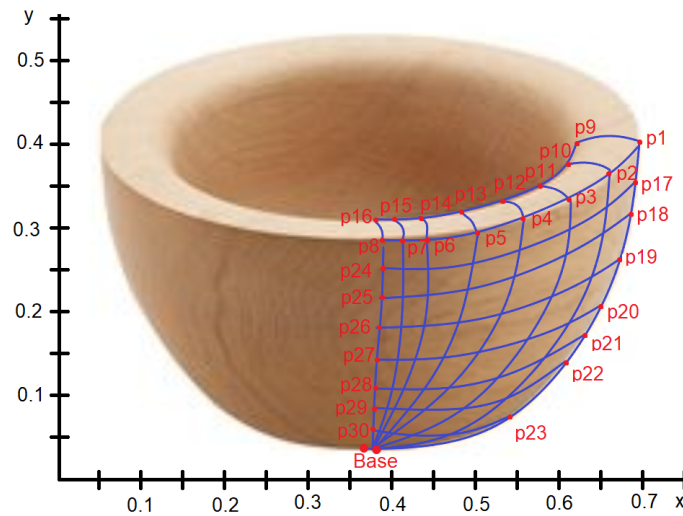


Figure 8: Puntos Mortero

Teniendo en cuenta lo anterior se hizo un programa en el lenguaje R para graficar el mortero utilizando los Splines de Bezier:

```
rm(list=ls())
#Splines de Bezier para graficar el Mortero
options(digits = 16)
library(grid)
library(gridBezier)
library(vwline)
grid.newpage()
#Punto Base-----
xbase<-0.379768786
ybase<-0.035344828
#Punto p1
xp1<-0.696242775
yp1<-0.403448276
#Lineas Verticales -----
#Base a p1-----
x<-c(xbase,0.565317919,0.692774566,xp1)
y<-c(ybase,0.018103448,0.142241379,yp1)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(xbase,2*xbase-x[2],2*xbase-x[3],2*xbase-x[4])
```

```

grid.Bezier(xrx,y)
#-----
#Base a p2-----
x<-c(xbase,0.523699422,0.661560694,0.65982659)
y<-c(ybase,0.049137931,0.217241379,0.364655172)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(xbase,2*xbase-x[2],2*xbase-x[3],2*xbase-x[4])
grid.Bezier(xrx,y)
#Base a p3-----
x<-c(xbase,0.519364162,0.615606936,0.611271676)
y<-c(ybase,0.08362069,0.215517241,0.334482759)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(xbase,2*xbase-x[2],2*xbase-x[3],2*xbase-x[4])
grid.Bezier(xrx,y)
#Base a p4-----
x<-c(xbase,0.501156069,0.553179191,0.556647399)
y<-c(ybase,0.127586207,0.206896552,0.311206897)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(xbase,2*xbase-x[2],2*xbase-x[3],2*xbase-x[4])
grid.Bezier(xrx,y)
#Base a p5-----
x<-c(xbase,0.478612717,0.497687861,0.502023121)
y<-c(ybase,0.152586207,0.204310345,0.293965517)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(xbase,2*xbase-x[2],2*xbase-x[3],2*xbase-x[4])
grid.Bezier(xrx,y)
#Base a p6-----
x<-c(xbase,0.43699422,0.450867052,0.442196532)
y<-c(ybase,0.137068966,0.200862069,0.285344828)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(xbase,2*xbase-x[2],2*xbase-x[3],2*xbase-x[4])
grid.Bezier(xrx,y)
#Base a p7-----
x<-c(xbase,0.410115607,0.417919075,0.411849711)
y<-c(ybase,0.129310345,0.18362069,0.284482759)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(xbase,2*xbase-x[2],2*xbase-x[3],2*xbase-x[4])
grid.Bezier(xrx,y)
#-----
#Lineas Horizontales-----

```

```

#p24 a p17-----
x<-c(0.387572254,0.464739884,0.684104046,0.691907514)
y<-c(0.250862069,0.238793103,0.344827586,0.356034483)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
2*xbase-x[4])
grid.Bezier(xrx,y)
#p25 a p18-----
x<-c(0.386705202,0.472543353,0.631213873,0.686705202)
y<-c(0.218103448,0.2,0.256896552,0.317241379)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
2*xbase-x[4])
grid.Bezier(xrx,y)
#p26 a p19-----
x<-c(0.383236994,0.43699422,0.638150289,0.671965318)
y<-c(0.181034483,0.164655172,0.221551724,0.262068966)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
2*xbase-x[4])
grid.Bezier(xrx,y)
#p27 a p20-----
x<-c(0.38150289,0.464739884,0.613872832,0.649421965)
y<-c(0.143103448,0.131896552,0.173275862,0.206896552)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
2*xbase-x[4])
grid.Bezier(xrx,y)
#p28 a p21-----
x<-c(0.378901734,0.443930636,0.552312139,0.631213873)
y<-c(0.10862069,0.095689655,0.118965517,0.171551724)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
2*xbase-x[4])
grid.Bezier(xrx,y)
#p29 a p22-----
x<-c(0.378034682,0.496820809,0.549710983,0.60867052)
y<-c(0.08362069,0.076724138,0.093965517,0.139655172)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
2*xbase-x[4])
grid.Bezier(xrx,y)

```

```

,2*xbase-x[4])
grid.Bezier(xrx,y)
#p30 a p23-----
x<-c(0.375433526,0.409248555,0.485549133,0.54017341)
y<-c(0.060344828,0.046551724,0.046551724,0.074137931)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
,2*xbase-x[4])
grid.Bezier(xrx,y)
#-----
#Borde superior-----
#p8 a p1-----
x<-c(0.386705202,0.494219653,0.667630058,xp1)
y<-c(0.285344828,0.275862069,0.348275862,yp1)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
,2*xbase-x[4])
grid.Bezier(xrx,y)
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3],2*yp1-y[4])
grid.Bezier(x,yry)
#Reflejo xy
grid.Bezier(xrx,yry)
#p16 a p9-----
x<-c(0.379768786,0.474277457,0.61300578,0.621676301)
y<-c(0.310344828,0.293103448,0.361206897,0.400862069)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
,2*xbase-x[4])
grid.Bezier(xrx,y)
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3],2*yp1-y[4])
grid.Bezier(x,yry)
#Reflejo xy
grid.Bezier(xrx,yry)
#p9 a p1-----
x<-c(0.621676301,0.632947977,0.667630058,xp1)
y<-c(0.400862069,0.417241379,0.419827586,yp1)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
,2*xbase-x[4])
grid.Bezier(xrx,y)

```



```

#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3],2*yp1-y[4])
grid.Bezier(x,yry)
#Reflejo xy
grid.Bezier(xrx,yry)
#p10 a p2-----
x<-c(0.611271676,0.625144509,0.656358382,0.65982659)
y<-c(0.376724138,0.387068966,0.382758621,0.364655172)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
,2*xbase-x[4])
grid.Bezier(xrx,y)
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3],2*yp1-y[4])
grid.Bezier(x,yry)
#Reflejo xy
grid.Bezier(xrx,yry)
#p11 a p3-----
x<-c(0.576589595,0.589595376,0.606069364,0.611271676)
y<-c(0.350862069,0.356034483,0.35,0.334482759)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
,2*xbase-x[4])
grid.Bezier(xrx,y)
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3],2*yp1-y[4])
grid.Bezier(x,yry)
#Reflejo xy
grid.Bezier(xrx,yry)
#p12 a p4-----
x<-c(0.53150289,0.549710983,0.560115607,0.556647399)
y<-c(0.332758621,0.335344828,0.325862069,0.311206897)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
,2*xbase-x[4])
grid.Bezier(xrx,y)
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3],2*yp1-y[4])
grid.Bezier(x,yry)
#Reflejo xy
grid.Bezier(xrx,yry)
#p13 a p5-----
x<-c(0.482080925,0.494219653,0.502023121,0.502023121)

```

```

y<-c(0.319827586,0.315517241,0.306896552,0.293965517)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
2*xbase-x[4])
grid.Bezier(xrx,y)
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3],2*yp1-y[4])
grid.Bezier(x,yry)
#Reflejo xy
grid.Bezier(xrx,yry)
#p14 a p6-----
x<-c(0.434393064,0.440462428,0.443930636,0.442196532)
y<-c(0.312068966,0.310344828,0.300862069,0.285344828)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
2*xbase-x[4])
grid.Bezier(xrx,y)
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3],2*yp1-y[4])
grid.Bezier(x,yry)
#Reflejo xy
grid.Bezier(xrx,yry)
#p15 a p7-----
x<-c(0.402312139,0.414450867,0.418786127,0.411849711)
y<-c(0.310344828,0.307758621,0.295689655,0.284482759)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
2*xbase-x[4])
grid.Bezier(xrx,y)
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3],2*yp1-y[4])
grid.Bezier(x,yry)
#Reflejo xy
grid.Bezier(xrx,yry)
#p8 a p16-----
x<-c(0.386705202,0.391040462,0.388439306,0.379768786)
y<-c(0.285344828,0.293965517,0.304310345,0.310344828)
grid.Bezier(x,y)
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3],
2*xbase-x[4])
grid.Bezier(xrx,y)
#Reflejo en y

```

```

yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3],2*yp1-y[4])
grid.Bezier(x,yry)
#Reflejo xy
grid.Bezier(xrx,yry)
#-----

```

El resultado se muestra en Figure 9:

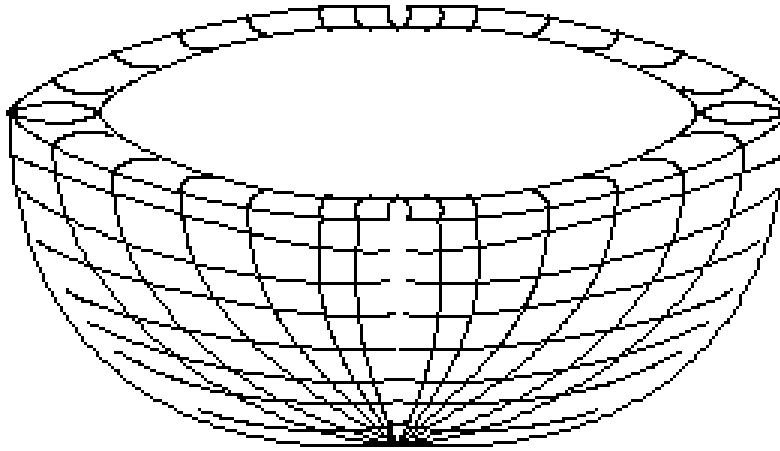


Figure 9: Mortero Splines Bezier

4 Conclusiones

Para comparar efectivamente los dos métodos anteriores hay que basarse en las graficas sencillas, ya que las soluciones de las graficas con volumen en realidad son relativas a la cantidad de lineas de volumen que se les quiera agregar, mientras que en las graficas sencillas solamente se manejan las lineas que son absolutamente necesarias para poder mostrar la figura del mortero.

Para el método de Splines de Bezier sólo se necesitaron 11 puntos mientras que en el método de Splines se necesitaron 24 puntos para generar la misma figura. Teniendo en cuenta lo anterior, se concluye que el método de Splines de Bezier es más eficiente porque requiere de menos puntos para obtener un resultado muy similar. También se concluye que el método de Splines de Bezier es más complejo de aplicar ya que la gráfica final depende del polígono de control y es difícil ubicarlo para obtener la línea deseada, mientras que en el método de Splines simplemente hay que colocar los puntos por donde se quiere que pase la línea.