Reto Parcial 2 Análisis Numérico

Gabriel Andrés Niño Carvajal - Juliana García Mogollón

19 de abril de 2020

1 Introduccion

El objetivo de esta sección es hacer una breve descripción del contenido del documento. Primero, se describe el problema. Luego, se describe la solución sencilla del problema, es decir, se realizaron graficas sencillas del mortero preocupandose sólo por su contorno. Después, se describe la solución compleja al problema, es decir, se realizaron graficas complejas del mortero preocupandose también por mostrar el volumen del objeto. Finalmente, se describe un conjunto de conclusiones donde se compara el funcionamiento e implementación de los dos métodos empleados para resolver el problema: Splines y Splines de Bezier.

2 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.

3 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 gráficar debido a los picos característicos que maneja su figura.
A partir de Figure 2 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:

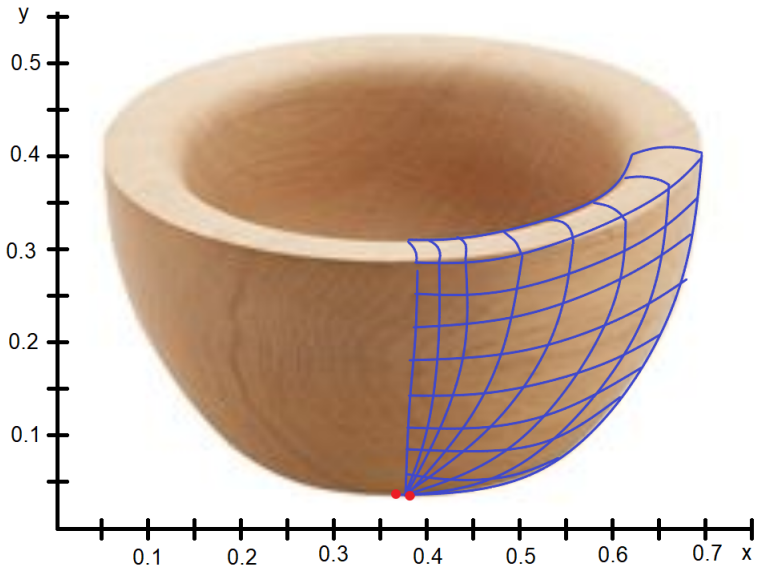

Figure 3: Coordenadas Mortero

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 alla de 1. A partir de Figure 3 se realizaron las mediciones correspondientes para gráficar 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 del mortero dentro del plano cartesiano establecido. Para los calculos se utilizó una tolerancia de 10^{-6}.

3.1 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.

Punto	$Pixel_x$	$Pixel_y$
Base	580	636
p1	945	209
p1a	794	656
p1b	941	512
p8	588	346
p8-1a	712	357
p8-1b	912	273
p9	859	212
p16	580	317
p16-9a	689	337
p16-9b	849	258

Table 1: Puntos Bezier Sencillos

En Figure 4 se muestran los polígonos de control:

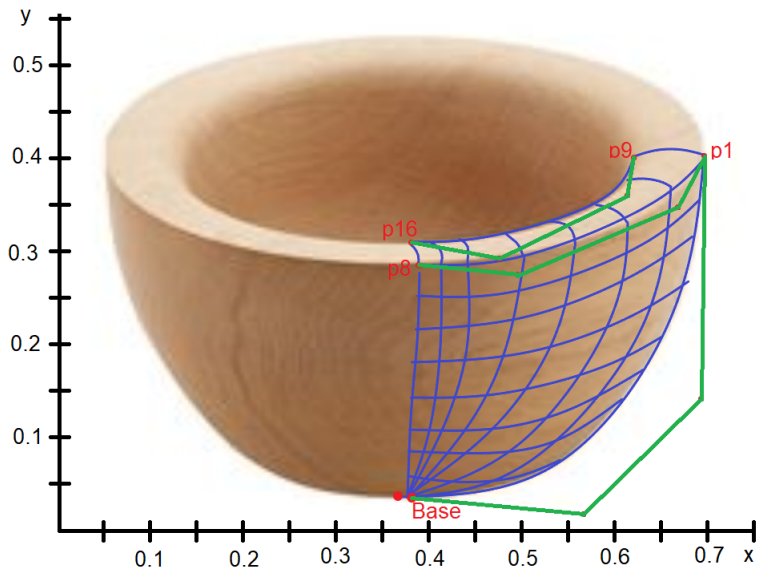

Figure 4: Mortero con Polígonos

Estos puntos representan sólo un cuadrante del mortero, de acuerdo con la sugerencia de la profesora, se utilizo 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, el nombre del programa es BezierSencillo.R.

```
#BezierSencillo implementacion con sencilla
rm(list=ls())
#Splines de Bezier para graficar el Mortero
options(digits = 16)
library(grid)
library(gridBezier)
library(vwline)
grid.newpage()
#Funciones--------------------------------
pixelx
pixely
px<-function(pixelx){
  (pixelx - 142)*0.75/865
}
py<-function(pixely){
  (677 - pixely)*0.55/638
}
#Punto Base-------------------------------
xbase<-px(580)
ybase<-py(636)
#Punto p1
#Punto p1
xp1<-px(945)
yp1<-py(209)
#Lineas Verticales------------------------------
#Base a p1--------------------------------
x<-c(xbase,px(794),px(941),xp1)
y<-c(ybase,py(656),py(512),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(px(588),px(712),px(912),xp1)
y<-c(py(346),py(357),py(273),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(px(580),px(689),px(849),px(859))
y<-c(py(317),py(337),py(258),py(212))
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:

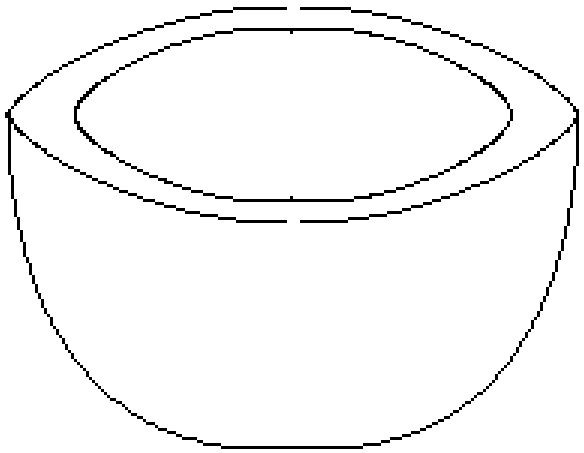

Figure 5: Mortero Sencillo

3.2 Mortero Splines Normales

Para este caso fue necesario utilizar más puntos:

Punto	$Pixel_x$	$Pixel_y$
Base	580	636
p1	945	209
p2	903	254
p3	847	289
p4	784	316
p5	721	336
p6	652	346
p7	617	347
p8	588	346
p9	859	212
p10	847	240
p11	807	270
p12	755	291
p13	698	306
p14	643	315
p15	606	317
p16	580	317
p17	940	264
p18	934	309
p19	917	373
p20	891	437
p21	870	478
p22	844	515
p23	765	591

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:

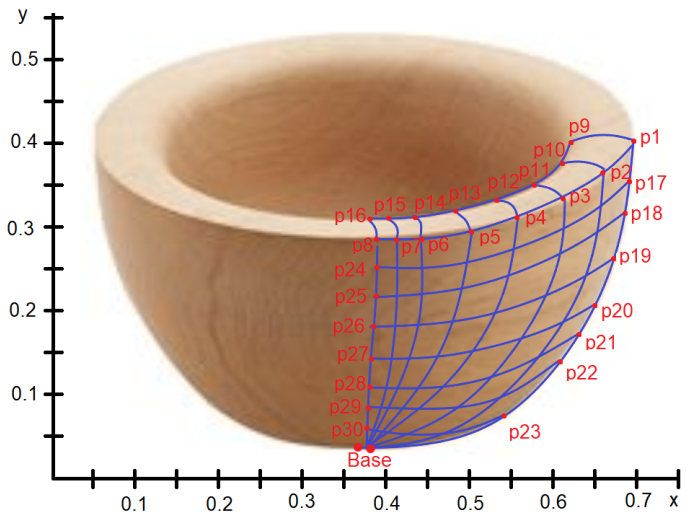

Figure 6: Mortero Sencillo

Nuevamente se hizo un programa en el lenguaje R para graficar el mortero pero esta vez utilizando el método Splines, el nombre del programa es Splines-Mortero.R:

```
#SplinesMortero implementacion sencilla
rm(list=ls())
options(digits=16)
#Interpolacion Mortero por Splines
#Funciones--------------------------------
pixelx
pixely
px<-function(pixelx){
  (pixelx - 142)*0.75/865
}
py<-function(pixely){
  (677 - pixely)*0.55/638
}

#Pixeles en x-----------------------------
pixelesx<-c(580,945,903,847,784,721,652,617
            ,588,859,847,807,755,698,643,606
            ,580,940,934,917,891,870,844,765)
#Pixeles en y-----------------------------
pixelesy<-c(636,209,254,289,316,336,346,347
```

```
            ,346,212,240,270,291,306,315,317
            ,317,264,309,373,437,478,515,591)
#x----------------------------------------
puntosx<-px(pixelesx)
#y----------------------------------------
puntosy<-py(pixelesy)
#Grafica base
plot(puntosx,puntosy, main=paste("Mortero")
  ,xlim=c(0,0.75),ylim=c(0,0.55),col="red")
#Punto Base--------------------------------
xbase<-px(580)
ybase<-py(636)
#Punto p1
xp1<-px(945)
yp1<-py(209)
#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 se muestra en Figure 7:

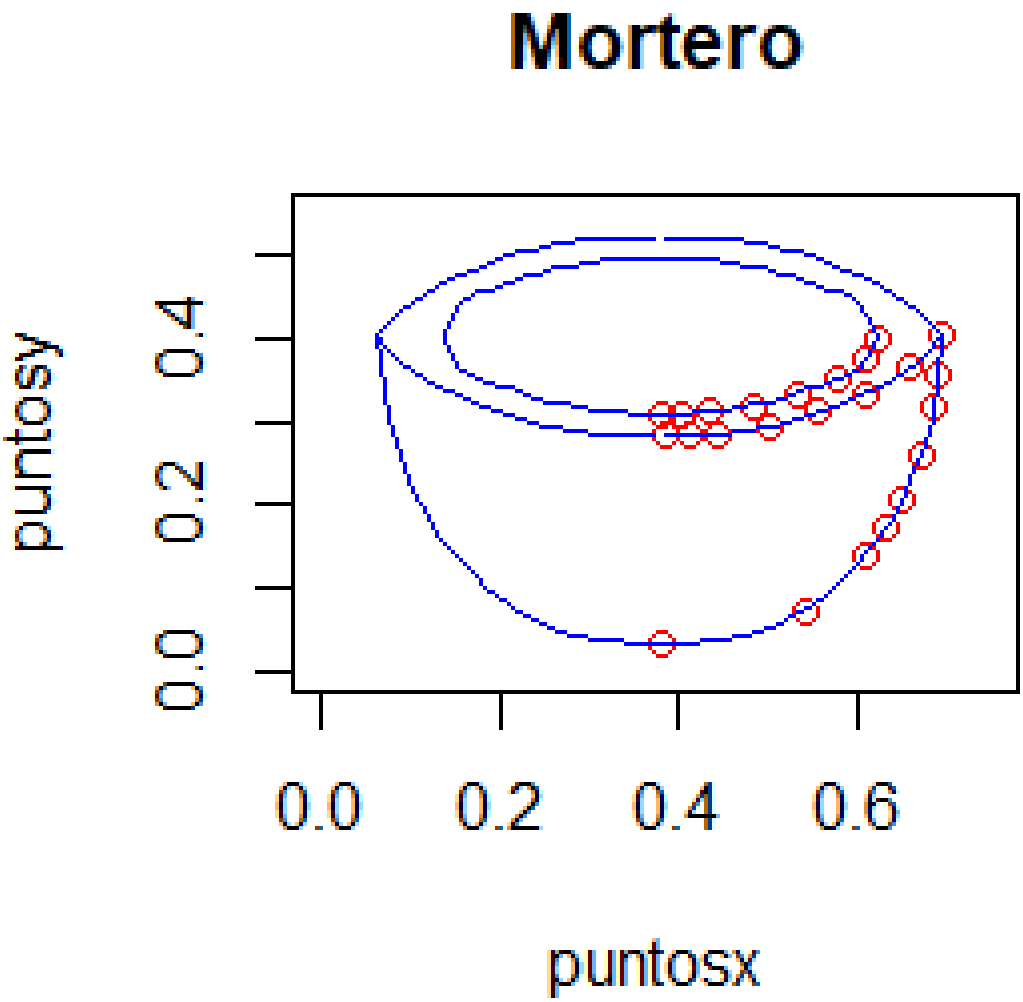

Figure 7: Mortero Splines

4 Solución Mortero con Volumen

4.1 Splines de Bezier con Volumen

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

Punto	$Pixel_x$	$Pixel_y$
Base	580	636
p1	945	209
p1a	794	656
p1b	941	512
p2	903	254
p2a	746	620
p2b	905	425
p3	847	289
p3a	741	580
p3b	852	427
p4	784	316
p4a	720	529
p4b	780	437
p5	721	336
p5a	694	500
p5b	716	440
p6	652	346
p6a	646	518
p6b	662	444
p7	617	347
p7a	615	527
p7b	624	464
p8	588	346
p8-1a	712	357
p8-1b	912	273
p8-16a	593	336
p8-16b	590	324
p9	859	212
p9-1a	872	193
p9-1b	912	190
p10	847	240
p10-2a	863	228
p10-2b	899	233
p11	807	270
p11-3a	822	264
p11-3b	841	271
p12	755	291
p12-4a	776	288
p12-4b	788	299
p13	698	306
p13-5a	712	311
p13-5b	721	321
p14	643	315
p14-6a	650	317
p14-6b	654	328
p15	606	317
p15-7a	620	320
p15-7b	625	334

p16	580	317
p16-9a	689	337
p16-9b	849	258
p17	940	264
p18	934	309
p19	917	373
p20	891	437
p21	870	478
p22	844	515
p23	765	591
p24	589	386
p24-17a	678	400
p24-17b	931	277
p25	588	424
p25-18a	687	445
p25-18b	870	379
p26	584	467
p26-19a	646	486
p26-19b	878	420
p27	582	511
p27-20a	678	524
p27-20b	850	476
p28	579	551
p28-21a	654	566
p28-21b	779	539
p29	578	580
p29-22a	715	588
p29-22b	776	568
p30	575	607
p30-23a	614	623
p30-23b	702	623

Table 3: Puntos Mortero Bezier con Volumen

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

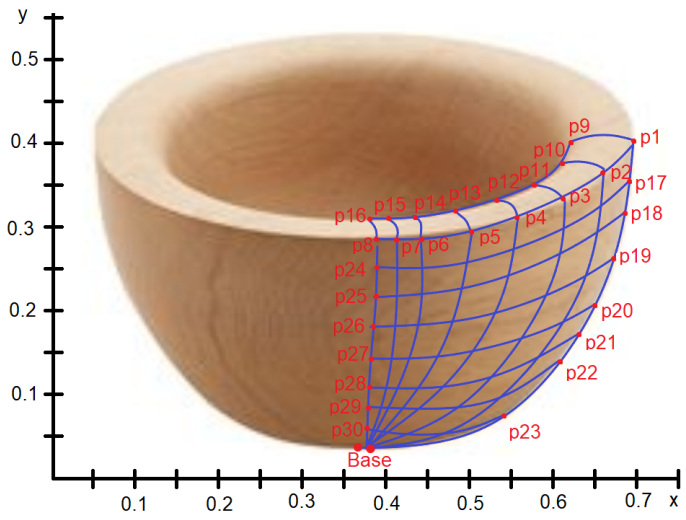

Figure 8: Puntos Mortero

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

```
#Bezier implementacion con Volumen
rm(list=ls())
#Splines de Bezier para graficar el Mortero
options(digits = 16)
library(grid)
library(gridBezier)
library(vwline)
grid.newpage()
#Funciones--------------------------------
pixelx
pixely
px<-function(pixelx){
  (pixelx - 142)*0.75/865
}
py<-function(pixely){
  (677 - pixely)*0.55/638
}
#Punto Base-------------------------------
xbase<-px(580)
```

```
ybase<-py(636)
#Punto p1
#Punto p1
xp1<-px(945)
yp1<-py(209)
#Lineas Verticales--------------------------------
#Base a p1---------------------------------
x<-c(xbase,px(794),px(941),xp1)
y<-c(ybase,py(656),py(512),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,px(746),px(905),px(903))
y<-c(ybase,py(620),py(425),py(254))
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,px(741),px(852),px(847))
y<-c(ybase,py(580),py(427),py(289))
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,px(720),px(780),px(784))
y<-c(ybase,py(529),py(437),py(316))
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,px(694),px(716),px(721))
y<-c(ybase,py(500),py(440),py(336))
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,px(646),px(662),px(652))
y<-c(ybase,py(518),py(444),py(346))
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,px(615),px(624),px(617))
y<-c(ybase,py(527),py(464),py(347))
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(px(589),px(678),px(931),px(940))
y<-c(py(386),py(400),py(277),py(264))
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(px(588),px(687),px(870),px(934))
y<-c(py(424),py(445),py(379),py(309))
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(px(584),px(646),px(878),px(917))
y<-c(py(467),py(486),py(420),py(373))
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(px(582),px(678),px(850),px(891))
y<-c(py(511),py(524),py(476),py(437))
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(px(579),px(654),px(779),px(870))
```

```
y<-c(py(551),py(566),py(539),py(478))
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(px(578),px(715),px(776),px(844))
y<-c(py(580),py(588),py(568),py(515))
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)
#p30 a p23--------------------------------
x<-c(px(575),px(614),px(702),px(765))
y<-c(py(607),py(623),py(623),py(591))
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(px(588),px(712),px(912),xp1)
y<-c(py(346),py(357),py(273),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(px(580),px(689),px(849),px(859))
y<-c(py(317),py(337),py(258),py(212))
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(px(859),px(872),px(912),xp1)
y<-c(py(212),py(193),py(190),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(px(847),px(863),px(899),px(903))
y<-c(py(240),py(228),py(233),py(254))
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(px(807),px(822),px(841),px(847))
y<-c(py(270),py(264),py(271),py(289))
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(px(755),px(776),px(788),px(784))
y<-c(py(291),py(288),py(299),py(316))
```

```
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(px(698),px(712),px(721),px(721))
y<-c(py(306),py(311),py(321),py(336))
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(px(643),px(650),px(654),px(652))
y<-c(py(315),py(317),py(328),py(346))
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(px(606),px(620),px(625),px(617))
y<-c(py(317),py(320),py(334),py(347))
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(px(588),px(593),px(590),px(580))
y<-c(py(346),py(336),py(324),py(317))
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:

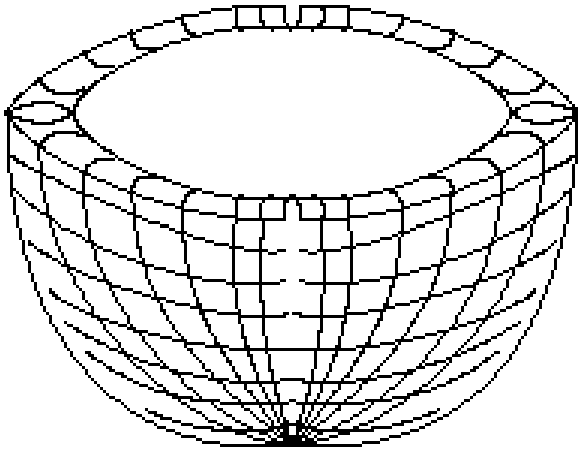

Figure 9: Mortero Splines Bezier

4.2 Splines con volumen

Para graficar el mortero con volumen se necesitaron 31 puntos principales y 50 puntos intermedios para formar las curvas.

Punto	$Pixel_x$	$Pixel_y$
Base	580	636
p1	945	209
p2	903	254
p3	847	289
p4	784	316
p5	721	336
p6	652	346
p7	617	347
p8	588	346
p9	859	212
p10	847	240
p11	807	270
p12	755	291
p13	698	306
p14	643	315
p15	606	317
p16	580	317
p17	940	264
p18	934	309
p19	917	373
p20	891	437
p21	870	478
p22	844	515
p23	765	591
p24	589	386
p25	588	424
p26	584	467
p27	582	511
p28	579	551
p29	578	580
p30	575	607
i1	900	299
i2	895	336
i3	879	398
i4	848	462
i5	814	511
i6	760	566
i7	682	614
i8	850	331
i9	844	367
i10	823	426
i11	789	485
i12	749	537
i13	703	579
i14	633	617
i15	779	361
i16	772	399
i17	753	449
i18	724	500

i19	680	550
i20	648	580
i21	608	613
i22	714	379
i23	708	416
i24	694	461
i25	672	510
i26	646	554
i27	626	580
i28	600	611
i29	652	386
i30	651	422
i31	646	467
i32	635	511
i33	619	553
i34	606	581
i35	591	611
i36	618	387
i37	618	425
i38	616	467
i39	608	512
i40	602	553
i41	593	579
i42	583	610
j1	905	201
j2	880	240
j3	832	274
j4	776	298
j5	711	317
j6	651	326
j7	618	329
j8	587	330

Table 4: Puntos Mortero Spline

En este caso si se utilizarón todos los puntos que se muestran en Figure 10:

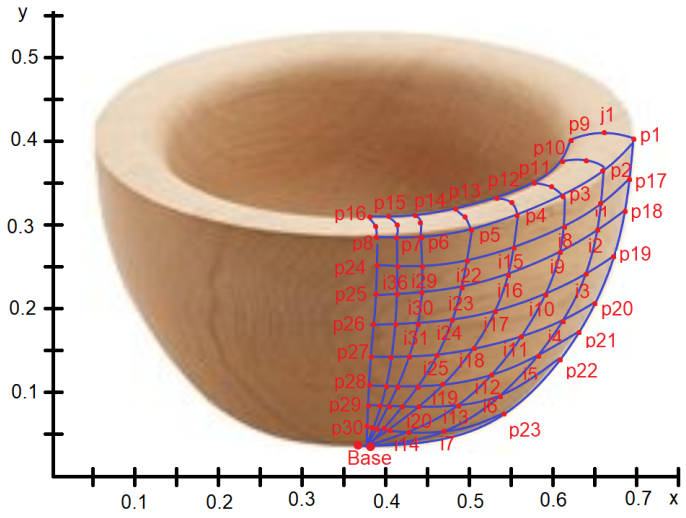

Figure 10: Mortero con muchos puntos

Teniendo en cuenta lo anterior se hizo un programa en el lenguaje R para graficar el mortero utilizando los Splines normales, el nombre del programa es SplinesVolumen.R:

```
#SplinesVolumen implementacion con Volumen
rm(list=ls())
options(digits=16)
#Interpolacion Mortero por Splines
#Funciones-------------------------------
pixelx
pixely
px<-function(pixelx){
        (pixelx - 142)*0.75/865
}
py<-function(pixely){
        (677 - pixely)*0.55/638
}

#Pixeles en x---------------------------------
pixelesx<-c(580,945,903,847,784,721,652,617
            ,588,859,847,807,755,698,643,606
            ,580,940,934,917,891,870,844,765
            ,589,588,584,582,579,578,575,900
```

```
            ,895,879,848,814,760,682,850,844
            ,823,789,749,703,633,779,772,753
            ,724,680,648,608,714,708
            ,694,672,646,626,600,652,651,646
            ,635,619,606,591,618,618,616,608
            ,602,593,583,905,880,832,776,711
            ,651,618,587)
#Pixeles en y-------------------------------
pixelesy<-c(636,209,254,289,316,336,346,347
            ,346,212,240,270,291,306,315,317
            ,317,264,309,373,437,478,515,591
            ,386,424,467,511,551,580,607,299
            ,336,398,462,511,566,614,331,367
            ,426,485,537,579,617,361,399,449
            ,500,550,580,613,379,416
            ,461,510,554,580,611,386,422,467
            ,511,553,581,611,387,425,467,512
            ,553,579,610,201,240,274,298,317
            ,326,329,330)
#x-----------------------------------------
puntosx<-px(pixelesx)
#y-----------------------------------------
puntosy<-py(pixelesy)
#Grafica base
plot(puntosx,puntosy, main=paste("Mortero")
     ,xlim=c(0,0.75),ylim=c(0,0.55),col="red")
#Punto Base--------------------------------
xbase<-px(580)
ybase<-py(636)
#Punto p1
xp1<-px(945)
yp1<-py(209)
#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")
```

```
#Base a p2-------------------------------
x<-c(xbase,puntosx[38],puntosx[37],puntosx[36]
,puntosx[35],puntosx[34],puntosx[33],puntosx[32]
,puntosx[3])
y<-c(ybase,puntosy[38],puntosy[37],puntosy[36]
,puntosy[35],puntosy[34],puntosy[33],puntosy[32]
,puntosy[3])
lines(spline(x[1:4],y[1:4]), col = "blue")
linea = spline(y[4:9],x[4:9])
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, 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],2*xbase-x[9])
lines(spline(xrx[1:4],y[1:4]), col = "blue")
linea = spline(y[4:9],xrx[4:9])
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#Base a p3-------------------------------
x<-c(xbase,puntosx[45],puntosx[44],puntosx[43]
,puntosx[42],puntosx[41],puntosx[40],puntosx[39]
,puntosx[4])
y<-c(ybase,puntosy[45],puntosy[44],puntosy[43]
,puntosy[42],puntosy[41],puntosy[40],puntosy[39]
,puntosy[4])
#lines(spline(x,y), col = "blue")
linea = spline(y,x)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, 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],2*xbase-x[9])
linea = spline(y,xrx)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#Base a p4-------------------------------
```

```
x<-c(xbase,puntosx[52],puntosx[51],puntosx[50]
,puntosx[49],puntosx[48],puntosx[47],puntosx[46]
,puntosx[5])
y<-c(ybase,puntosy[52],puntosy[51],puntosy[50]
,puntosy[49],puntosy[48],puntosy[47],puntosy[46]
,puntosy[5])
linea = spline(y,x)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, 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],2*xbase-x[9])
linea = spline(y,xrx)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#Base a p5--------------------------------
x<-c(xbase,puntosx[58],puntosx[57],puntosx[56]
,puntosx[55],puntosx[54],puntosx[53],puntosx[52]
,puntosx[6])
y<-c(ybase,puntosy[58],puntosy[57],puntosy[56]
,puntosy[55],puntosy[54],puntosy[53],puntosy[52]
,puntosy[6])
linea = spline(y,x)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, 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],2*xbase-x[9])
linea = spline(y,xrx)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#Base a p6--------------------------------
x<-c(xbase,puntosx[65],puntosx[64],puntosx[63]
,puntosx[62],puntosx[61],puntosx[60],puntosx[59]
,puntosx[7])
y<-c(ybase,puntosy[65],puntosy[64],puntosy[63]
```

```
,puntosy[62],puntosy[61],puntosy[60],puntosy[59]
,puntosy[7])
linea = spline(y,x)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, 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],2*xbase-x[9])
linea = spline(y,xrx)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#Base a p7--------------------------------
x<-c(xbase,puntosx[72],puntosx[71],puntosx[70]
,puntosx[69],puntosx[68],puntosx[67],puntosx[66]
,puntosx[8])
y<-c(ybase,puntosy[72],puntosy[71],puntosy[70]
,puntosy[69],puntosy[68],puntosy[67],puntosy[66]
,puntosy[8])
linea = spline(y,x)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, 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],2*xbase-x[9])
linea = spline(y,xrx)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#--------------------------------
#Lineas Horizontales--------------------------------
#p17 a p24--------------------------------
x<-c(puntosx[18],puntosx[32],puntosx[39],puntosx[46]
     ,puntosx[53],puntosx[60],puntosx[67],puntosx[25])
y<-c(puntosy[18],puntosy[32],puntosy[39],puntosy[46]
     ,puntosy[53],puntosy[60],puntosy[67],puntosy[25])
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")
#p18 a p25--------------------------------
x<-c(puntosx[19],puntosx[33],puntosx[40],puntosx[47]
     ,puntosx[54],puntosx[61],puntosx[68],puntosx[26])
y<-c(puntosy[19],puntosy[33],puntosy[40],puntosy[47]
     ,puntosy[54],puntosy[61],puntosy[68],puntosy[26])
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")
#p19 a p26--------------------------------
x<-c(puntosx[20],puntosx[34],puntosx[41],puntosx[48]
     ,puntosx[55],puntosx[62],puntosx[69],puntosx[27])
y<-c(puntosy[20],puntosy[34],puntosy[41],puntosy[48]
     ,puntosy[55],puntosy[62],puntosy[69],puntosy[27])
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")
#p20 a p27--------------------------------
x<-c(puntosx[21],puntosx[35],puntosx[42],puntosx[49]
     ,puntosx[56],puntosx[63],puntosx[70],puntosx[28])
y<-c(puntosy[21],puntosy[35],puntosy[42],puntosy[49]
     ,puntosy[56],puntosy[63],puntosy[70],puntosy[28])
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")
#p21 a p28--------------------------------
x<-c(puntosx[22],puntosx[36],puntosx[43]
     ,puntosx[50],puntosx[57]
     ,puntosx[64],puntosx[71],puntosx[29])
y<-c(puntosy[22],puntosy[36],puntosy[43]
     ,puntosy[50],puntosy[57]
     ,puntosy[64],puntosy[71],puntosy[29])
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")
#p22 a p29--------------------------------
x<-c(puntosx[23],puntosx[37],puntosx[44]
     ,puntosx[51],puntosx[58]
     ,puntosx[65],puntosx[72],puntosx[30])
y<-c(puntosy[23],puntosy[37],puntosy[44]
     ,puntosy[51],puntosy[58]
     ,puntosy[65],puntosy[72],puntosy[30])
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")
#p23 a p30--------------------------------
x<-c(puntosx[24],puntosx[38],puntosx[45]
     ,puntosx[52],puntosx[59]
     ,puntosx[66],puntosx[73],puntosx[31])
y<-c(puntosy[24],puntosy[38],puntosy[45]
     ,puntosy[52],puntosy[59]
     ,puntosy[66],puntosy[73],puntosy[31])
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")
#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")
#p1 a p9---------------------------------
x<-c(puntosx[2],puntosx[74],puntosx[10])
y<-c(puntosy[2],puntosy[74],puntosy[10])
lines(spline(x,y), col = "blue")
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3])
lines(spline(xrx,y), col = "blue")
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3])
lines(spline(x,yry), col = "blue")
#Reflejo xy
lines(spline(xrx,yry), col = "blue")
#p2 a p10--------------------------------
x<-c(puntosx[3],puntosx[75],puntosx[11])
y<-c(puntosy[3],puntosy[75],puntosy[11])
lines(spline(x,y), col = "blue")
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3])
lines(spline(xrx,y), col = "blue")
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3])
lines(spline(x,yry), col = "blue")
#Reflejo xy
lines(spline(xrx,yry), col = "blue")
#p3 a p11--------------------------------
```

```
x<-c(puntosx[4],puntosx[76],puntosx[12])
y<-c(puntosy[4],puntosy[76],puntosy[12])
lines(spline(x,y), col = "blue")
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3])
lines(spline(xrx,y), col = "blue")
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3])
lines(spline(x,yry), col = "blue")
#Reflejo xy
lines(spline(xrx,yry), col = "blue")
#p4 a p12--------------------------------
x<-c(puntosx[5],puntosx[77],puntosx[13])
y<-c(puntosy[5],puntosy[77],puntosy[13])
lines(spline(x,y), col = "blue")
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3])
lines(spline(xrx,y), col = "blue")
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3])
lines(spline(x,yry), col = "blue")
#Reflejo xy
lines(spline(xrx,yry), col = "blue")
#p5 a p13--------------------------------
x<-c(puntosx[6],puntosx[78],puntosx[14])
y<-c(puntosy[6],puntosy[78],puntosy[14])
lines(spline(x,y), col = "blue")
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3])
lines(spline(xrx,y), col = "blue")
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3])
lines(spline(x,yry), col = "blue")
#Reflejo xy
lines(spline(xrx,yry), col = "blue")
#p6 a p14--------------------------------
x<-c(puntosx[7],puntosx[79],puntosx[15])
y<-c(puntosy[7],puntosy[79],puntosy[15])
linea = spline(y,x)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3])
linea = spline(y,xrx)
```

```
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3])
linea = spline(yry,x)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#Reflejo xy
linea = spline(yry,xrx)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#p7 a p15---------------------------------
x<-c(puntosx[8],puntosx[80],puntosx[16])
y<-c(puntosy[8],puntosy[80],puntosy[16])
linea = spline(y,x)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3])
linea = spline(y,xrx)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3])
linea = spline(yry,x)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#Reflejo xy
linea = spline(yry,xrx)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#p8 a p16---------------------------------
```

```
x<-c(puntosx[9],puntosx[81],puntosx[17])
y<-c(puntosy[9],puntosy[81],puntosy[17])
linea = spline(y,x)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#Reflejo en x
xrx<-c(2*xbase-x[1],2*xbase-x[2],2*xbase-x[3])
linea = spline(y,xrx)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#Reflejo en y
yry<-c(2*yp1-y[1],2*yp1-y[2],2*yp1-y[3])
linea = spline(yry,x)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
#Reflejo xy
linea = spline(yry,xrx)
l = linea$x
linea$x = linea$y
linea$y = l
lines(linea, col = "blue")
```

El resultado se muestra en Figure 11:

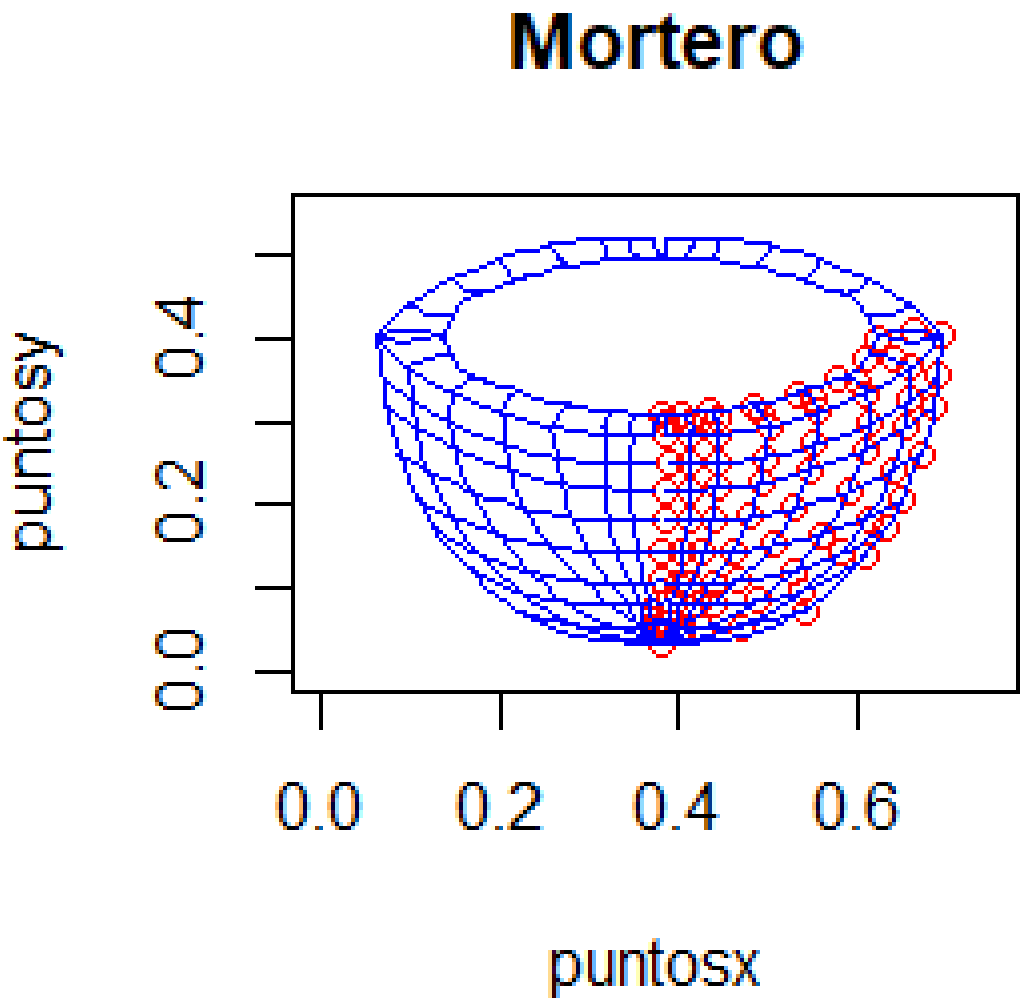

Figure 11: Mortero Splines Bezier

5 Conclusiones

Para comparar los Splines de Bezier y los Splines se tendran en cuenta 3 factores: Cantidad de puntos, facilidad de comprensión e implementación.

5.1 Cantidad de puntos

En las graficas sencillas 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.
En las graficas con volumen para el método de Splines de Bezier se necesitaron 79 puntos mientras que en el método de Splines se neceitaron 81 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.

5.2 Facilidad de comprensión

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.
Teniendo en cuenta lo anterior se concluye que el método de Splines es más facil de comprender que los Splines de Bezier.

5.3 Implementación

Los para utilizar la función de Bezier sólo se necesita manejar los cuatro puntos del polígono de control. Para los Splines normales sólo hay que poner los puntos por donde se quiere que pase la línea, sin embargo, hay que tener cuidado con el manejo de la función pues muchas veces falla y retorna líneas sin sentido que dañan la figura.
Teniendo en cuenta lo anterior, se concluye que los Splines de Bezier son más fáciles de implementar que los Splines normales

5.4 Conclusión final

Finalmente se concluye que el mejor método para graficar es el de Splines de Bezier ya que una vez que se entiende la teoría es fácil de implementar y requiere de menos puntos para obtener una gráfica.