

RETO DEL PERRITO



**PROFESORA:
EDDY HERRERA**

**INTEGRANTES:
DAVID HERRERA CAICEDO
PABLO ALEJANDRO PULIDO**



**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
ANÁLISIS NUMÉRICO
2019**

RETO DEL PERRITO

INTRODUCCIÓN:

El reto del perrito consiste en construir un Interpolador (no necesariamente en forma polinómica) utilizando la menor cantidad de puntos k (parte superior y/o inferior o en total) y reproducir el dibujo del contorno completo del perrito sin bigotes (mejor exactitud) con la información dada.

Criterios entregados:

1. Metodología que explique cómo se seleccionaron k puntos con $k \leq n$ con n el total de puntos dados (Selección de más puntos o de los puntos de la parte de abajo).
2. Algoritmo que se aplicó (justificación) por ejemplo, interpolación polinómica y como soluciono el sistema.
3. Validación del resultado.

Producto:

1. Algoritmo, requerimientos, codificación.
2. Codificación, tabla donde está la interpolación en los $n-k$ puntos (no seleccionados), el polinomio o la función interpolante. En un plano los puntos originales, los utilizados, el contorno y el interpolado (utilice el grosor mínimo para la curva).
3. Calcular la cota de error de su método con los datos experimentales y compárela con la cota teórica.
4. Tabla donde estén los valores interpolados (tenga en cuenta los que no utilizo), los originales y el error relativo, calcule un error relativo total como la suma de los errores relativos.
5. Cree una función que cuente el número aciertos y el número de diferencias en una cifra entre su función de interpolación y los originales y impleméntelo como el índice de Jaccard.

6. Cree una función que muestre la eficiencia de su método.

Preguntas:

1. ¿El origen se puede modificar?
2. ¿Si tenemos nueva información ósea nodos como podemos implementar esa información en el algoritmo de interpolación?
3. ¿Su método es robusto, en el sentido que si se tienen más puntos la exactitud no disminuye?
4. ¿Suponga que tiene más puntos con más cifras significativas cómo se comporta su algoritmo? ¿La exactitud decae?

DESARROLLO

Criterios entregados:

1. Metodología que explique cómo se seleccionaron k puntos con $k < n$ con n el total de puntos dados (Selección de más puntos o de los puntos de la parte de abajo).

La metodología en la que se seleccionaron los puntos consistió principalmente de:

- Para la selección de los puntos de la parte superior se evidenciaban los máximos y mínimos relativos que formaban el contorno de la espalda, cabeza, hocico y pata del perro. Por definición estos puntos son necesarios para mantener la suavidad del spline cúbico. De igual forma se escogieron los dos puntos más cercanos a estos máximos y mínimos con la finalidad de poder quitar los puntos máximos ya mencionados anteriormente.
- Para la selección de los puntos de la parte inferior, considerando que el número de curvas es menor y son más amplias e incluso pequeñas, se decidió tomar intervalos más espaciados, con prioridad de puntos a las curvas pequeñas ya que estas serían segmentadas.
- Se determinaron los puntos que se encontraban al inicio y final del perro, siendo estos el inicio de su cola y el fin de la pata estirada del perro.

- Otra condición fue que la parte de inferior y superior del perro se pudieran conectar a partir de un punto en común, que no incomodara ninguno de los dos intervalos, para esto se terminó escogiendo el primer punto en cuya coordenada en x fuera la menor y el punto cuyo coordenada en x fuera la mayor.

2. Algoritmo que se aplicó (justificación) por ejemplo, interpolación polinómica y como soluciono el sistema.

Un trazador o spline se usa para dibujar curvas suaves a través de un conjunto de puntos. **Los trazadores cúbicos** (cubic splines) **naturales** se utilizan para crear una función que interpola un conjunto de puntos de datos. Esta función consiste en una unión de polinomios cúbicos, uno para cada intervalo, y está construido para ser una función con derivada primera y segunda continuas. Dado que en los puntos exactos no se tienen derivadas exactas, el método de spline cúbico permite fácilmente y con bajo margen de error acercarse a los datos verdaderos.

Supongamos que tenemos $n + 1$ puntos $(x_0, y_0), \dots, (x_n, y_n)$ con $x_0 < x_1 < \dots < x_n$. En vez de interpolar f con un solo polinomio que pase por todos estos puntos, interpolamos la función f en cada subintervalo $[x_k, x_{k+1}]$ con un polinomio cúbico $S_k(x)$ de tal manera que el polinomio cúbico (o trazador cúbico) $S_i(x)$ en $[x_i, x_{i+1}]$ y el trazador cúbico $S_{i+1}(x)$ en $[x_{i+1}, x_{i+2}]$, coinciden en x_{i+1} y que también sus derivadas primera y segunda coincidan en este punto. Cada trazador cúbico coincide con f en los extremos de cada intervalo.

El uso de splines cúbicos permite también reducir la segmentación manual, cosa que en otros métodos tendría que haber sido escogida y modificada cada vez que se eliminara o añadiera un punto.

3. Validación del resultado.

Producto:

1. Algoritmo, requerimientos, codificación.

```
SplineCubico_Aux<- function(x,y){
  a=rep(y)
  n=length(x)
  h<-(c(x,0)-c(0,x))[2:n]
  #Cuando (i,j) i = j = 2 (hi+hi+1)
  #Cuando (i,j-1) = hi
  #Cuando (i,j+1) = hi+1
  A <- c(1,rep(0,times=n-1))
  for (i in 1:(n-2)) {
    # rep(0,times = i-1) y rep (0,times = n-i-2)
    # son para poner 0 en elementos que no compongan
    # la diagonal
    A <- rbind(A,c( rep(0,times=i-1) ,
                    c(h[i],2*(h[i]+h[i+1]),h[i+1]) ,
                    rep(0,times=n-i-2) ) )
  }
  A <- rbind(A,c(rep(0,times=n-1),1))
  casoBase<-(3/c(1,h,1,1)*(c(a,1,1) - c(1,a,1)) -
             3/c(1,1,h,1)*(c(1,a,1)-c(1,1,a)))[3:n]
  b <- c(0,casoBase,0)
  #Despeje de la ecuacion de la forma Ax = b
  c <- solve(A, b)
  #Se calculan los coeficientes del polinomio de
  # Spline
  b <- ((c(a,0) - c(0,a))/c(1,h,1) -
        c(1,h,1)/3*(c(c,0) + 2*c(0,c)))[2:n]
  d <- ((c(c,0) - c(0,c))/(3*c(1,h,1)))[2:n]

  print("Primera")
  print(b)
  print("Segunda")
  print(d)
  return( rbind(a[1:n-1],b,c[1:n-1],d) )
}
```

```
SplineCubico<-function(x,y){
  t = 1:length(x)
```

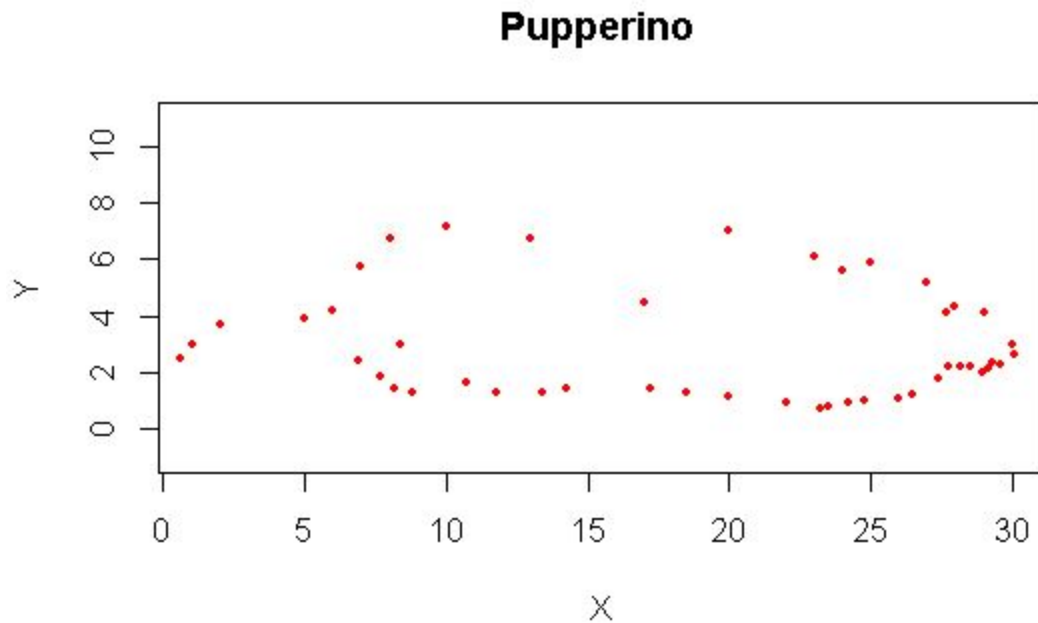
```

sx = SplineCubico_Aux(t,x)
sy = SplineCubico_Aux(t,y)
for (i in 1:(length(t)-1)) {
  dat<- data.frame(t=seq(t[i],t[i+1], by=0.1) )
  fx <- function(x) (sx[1,i] + sx[2,i]*(x-t[i]) +
sx[3,i]*(x-t[i])^2 + sx[4,i]*(x-t[i])^3)
  fy <- function(x) (sy[1,i] + sy[2,i]*(x-t[i]) +
sy[3,i]*(x-t[i])^2 + sy[4,i]*(x-t[i])^3)
  dat$y=fy(dat$t)
  dat$x=fx(dat$t)
  points(dat$x,dat$y,type='l', col='brown',lwd=1)
}
}

```

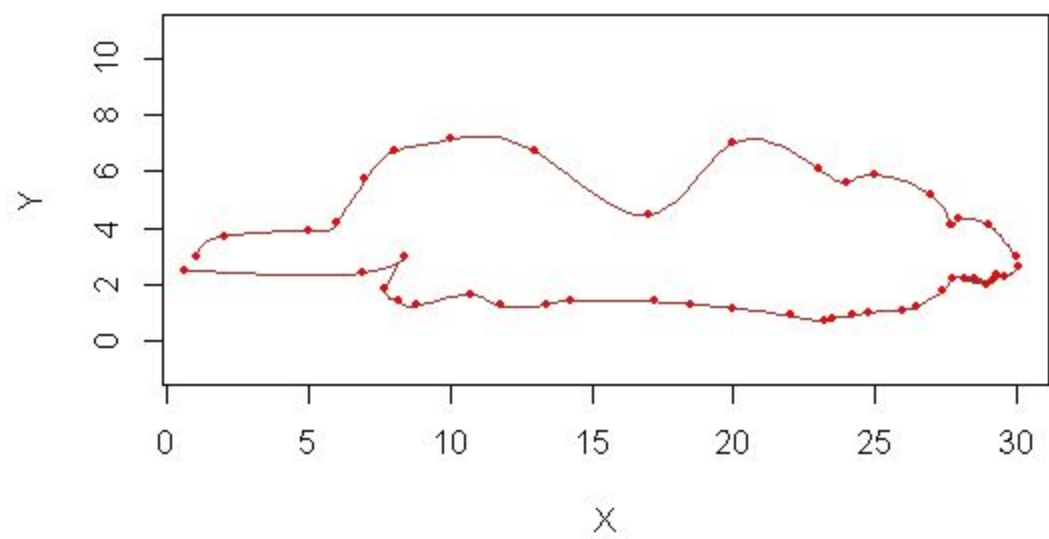
2. Codificación, tabla donde está la interpolación en los n-k puntos (no seleccionados), el polinomio o la función interpolante. En un plano los puntos originales, los utilizados, el contorno y el interpolado (utilice el grosor mínimo para la curva).

Puntos Originales:

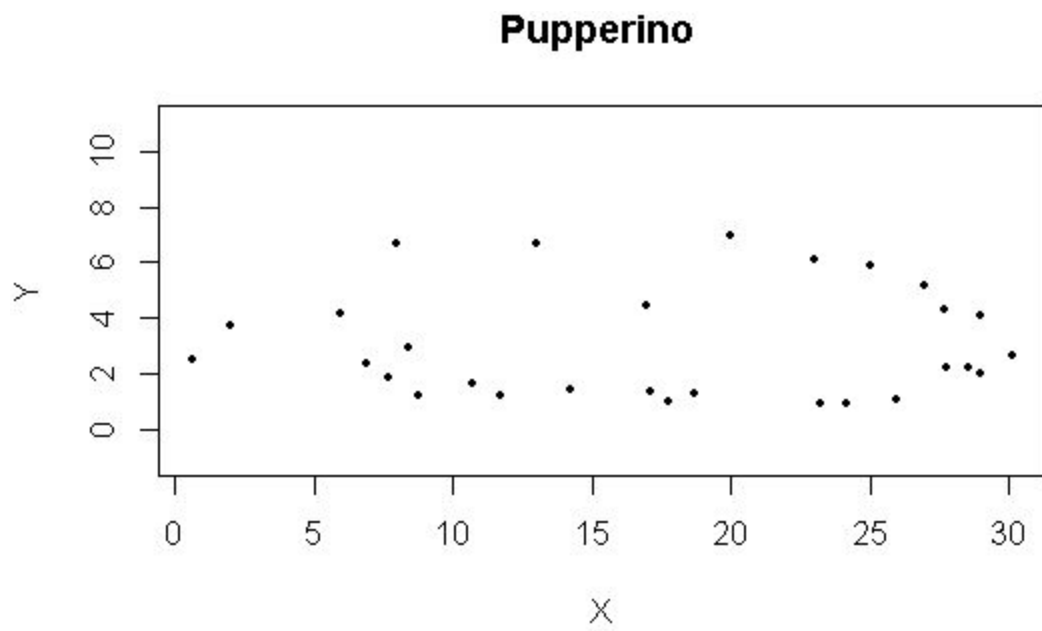


Contorno Original:

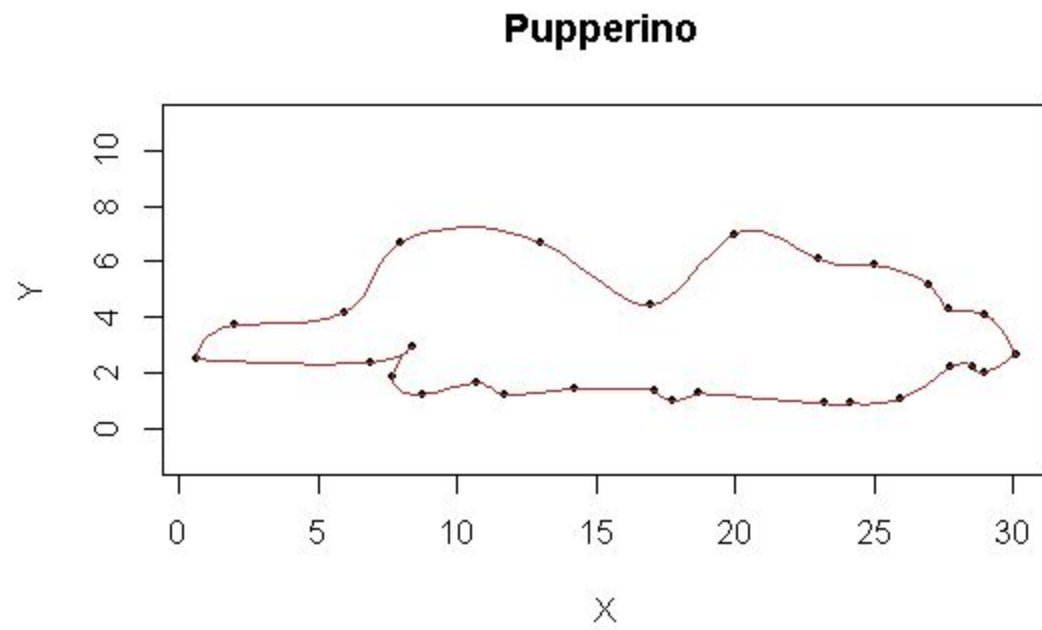
Pupperino



Puntos Interpolados:



Contorno Interpolado:



3. Calcular la cota de error de su método con los datos experimentales y compárela con la cota teórica.

La cota de error para S si es el interpolante único se calcula a partir de la siguiente función:

$$\max_{a \leq x \leq b} |f(x) - S(x)| \leq \frac{5M}{384}$$

Para esta cota es necesario tomar los valores originales de la figura y reemplazarlos en f, es decir hallar la cuarta derivada de la función en esas coordenadas junto con los valores interpolados.

Valores Originales:

x	y	$f^{(iii)}(x,y)$	$f^{(iv)}(x,y)$
1	3	0.78	-0.08
2	3.7	0.52	0.03
5	3.9	-0.07	0.23
6	4.2	1.00	-0.53
7	5.7	1.46	0.08
8	6.69	0.60	0.05
10	7.12	0.04	-0.03
13	6.7	-0.56	0.06
17	4.45	0.34	-0.09
20	7	0.50	0.04
23	6.1	-0.69	0.19
24	5.6	-0.10	-0.20
25	5.87	0.43	-0.17
27	5.15	-1.87	2.35
27.7	4.1	0.03	-5.71
28	4.3	0.78	0.32

29	4.1	-0.86	0.11
30	3	01.03	-2.65
0.62	2.5	-2.11	1.47
6.9	2.4	06.07	-0.73
8.38	2.95	-3.15	-6.01
7.7	1.84	-0.03	0.68
8.16	1.4	-0.15	-0.19
8.8	1.25	-0.33	0.27
10.7	1.65	0.18	-0.06
11.75	1.25	0.12	-0.02
13.39	1.28	-0.06	0.08
14.25	1.42	-0.08	0.05
17.2	1.39	-0.07	0.08
18.5	1.3	-0.29	-0.02
20	1.17	0.25	0.20
22	0.9	0.30	-1.15
23.2	0.7	0.12	0.29
23.5	0.8	0.13	-0.19
24.2	0.9	0.17	0.10
24.8	1	0.32	-0.06
26	1.10	01.07	0.23
26.5	1.23	0.86	-1.71
27.39	1.77	-0.67	0.24
27.78	2.2	0.12	2.83
28.56	2.22	1.21	2.93
28.96	02.02	1.51	2.89
28.19	2.18	0.85	-32.27
29.17	2.15	-0.59	16.28
29.31	2.36	1.22	2.15
29.57	2.25	1.12	1.14
30.12	2.64	0	0
Total		174624	193884

Valores Interpolados:

x	y	$f_{(iii)}(x,y)$	$f_{(iv)}(x,y)$
0.62	2.5	0	0
2	3.7	0.52	0.06
6	4.2	0.83	-0.12
8	6.69	1.15	0.01
13	6.7	-0.84	0.04
17	4.45	0.43	-0.09
20	7	0.39	0.06
23	6.1	-0.37	-0.02
25	5.87	0.03	-0.11
27	5.15	-1.20	0.90
27.7	4.3	-0.78	-0.58
29	4.1	-0.52	0.31
30.12	2.64	-1.35	3.39
0.62	2.5	2.66	-1.47
6.9	2.4	-5.15	5.97
8.38	2.95	11.18	6.77
7.7	1.84	-4.05	-1.05
8.8	1.25	0.67	0.71
10.7	1.65	-0.64	-0.08
11.75	1.25	0.25	0.01
14.25	1.42	-0.78	0.43
17.1	1.39	-0.24	-0.29
17.1	1.39	-1.25	2.005
17.74	1	-3	-0.02
18.7	1.28	-0.02	0.06
18.7	1.28	0.45	-0.12
23.2	0.9	0.39	-0.17
24.2	0.9	-0.44	1.52

26	1.10	-0.30	-0.31
27.78	2.2	-6.67	0.30
28.56	2.22	0.02	-0.34
28.96	02.02	-0.04	-0.73
30.12	2.64	0	0
Total		-3	2005

Reemplazando en la función anteriormente descrita tenemos como resultado lo siguiente:

$$\max_{a \leq x \leq b} |193.884 - 2.005| \leq \frac{5(193.884)}{384}$$

$$\max_{a \leq x \leq b} |191.879| \leq 2.524,53$$

4. Tabla donde estén los valores interpolados (tenga en cuenta los que no utilizo), los originales y el error relativo, calcule un error relativo total como la suma de los errores relativos.

Teniendo en cuenta de la longitud de la tabla, se decidió anexar al final la misma. Finalmente, se calculó que el error relativo total fue de **17.64694**.

El algoritmo usado genera esta tabla cada vez que se llama a la función spline cúbico, por lo que a la hora de ejecutar el archivo .R se generan varias tablas, ya que se usó esta función varias veces para los casos de prueba de la resolución de preguntas que se hará más adelante

5. Cree una función que cuente el número aciertos y el número de diferencias en una cifra entre su función de interpolación y los originales y implementarlo como el índice de Jaccard.

La función planteada calcula el número de aciertos y de diferencias a partir de los postulados del índice de Jaccard, estimando un valor relativo de igualdad entre los conjuntos de coordenadas que trazan el contorno del perro, permitiendo visualizar la efectividad del método utilizado para este caso Spline Cúbico.

$V_o \rightarrow \text{Valores Originales}$

$V_m \rightarrow \text{Valores interpolados}$

$V_o \cap V_m \rightarrow \text{Número de Aciertos}$

$V_o \cup V_m \rightarrow \text{Total puntos}$

$$J(V_o, V_m) = \frac{V_o \cap V_m}{V_o \cup V_m}$$

$$V_o = 47$$

$$V_m = 33$$

$$V_o \cap V_m = 24$$

$$V_o \cup V_m = 47$$

$$J(67, 23) = \frac{24}{47} = 0,51$$

Según el resultado anterior se puede deducir que se presentan varias coordenadas en común o muy cercanas entre ambos conjuntos específicamente de 24 puntos todos pertenecientes al conjunto del interpolador.

6. Cree una función que muestre la eficiencia de su método.

Para verificar la eficiencia del método se calculó la cantidad de operaciones que se realizaban respecto a los valores originales de la función y los valores interpolados a la hora de graficar el contorno del perro.

- Cantidad de Operaciones para los valores interpolados:

Contador de operaciones = 91

Contador de operaciones = 49

Contador de operaciones = 16

Contador de operaciones = 41

Total = 197

- Cantidad de Operaciones para los valores originales:

```
Contador de operaciones = 166
Contador de operaciones = 419
Total = 585
```

- **Función de complejidad:**

O(nk), donde k son los nodos correspondientes al contorno del perro.

Cómo es posible observar al tener solo 33 puntos se realizan solamente 197 operaciones para poder trazar el perro, caso contrario para los 47 puntos originales donde se deben ejecutar 585 operaciones. Lo cual indica que entre más nodos existan más operaciones se deben realizar. Sin embargo la complejidad del algoritmo continúa siendo de orden lineal, lo cual indica un alto grado de eficiencia sin importar la cantidad de nodos..

Preguntas:

1. ¿El origen se puede modificar?

Inicialmente, se planteó el dibujo del perrito con origen en la cola, de ahí en adelante se iba dibujando por partes, siendo la parte superior, la primera en ser dibujada. Luego, la parte inferior del dibujo se dividió en tres secciones: antes de la oreja, oreja, y después de la oreja. En este caso también se dibujó desde la cola hasta la pata delantera(junto al hocico)

Para probar los cambios que podría tener el programa al cambiar el origen, se decidió invertir los puntos, para que el dibujo iniciara desde el hocico y no desde la cola. Por consiguiente, los arreglos de puntos quedaron de la siguiente forma:

```
y2 <- c(2.64 , 4.1 , 4.3 , 5.15 , 5.87 , 6.1 , 7 , 4.45 , 6.7 , 6.69 , 4.2 , 3.7 , 2.5 )
```

```
x2 <- c(30.12, 29 , 27.7 , 27 , 25 , 23 , 20 , 17 , 13 , 8 , 6 , 2 , 0.62)
```

```
xb2 <- c(17.1, 14.25 , 11.75 , 10.7 , 8.8 , 7.7 , 8.38 , 6.9 , 0.62)
```

```
yb2 <- c(1.39, 1.42 , 1.25 , 1.65 , 1.25 , 1.84 , 2.95 , 2.4 , 2.5 )
```

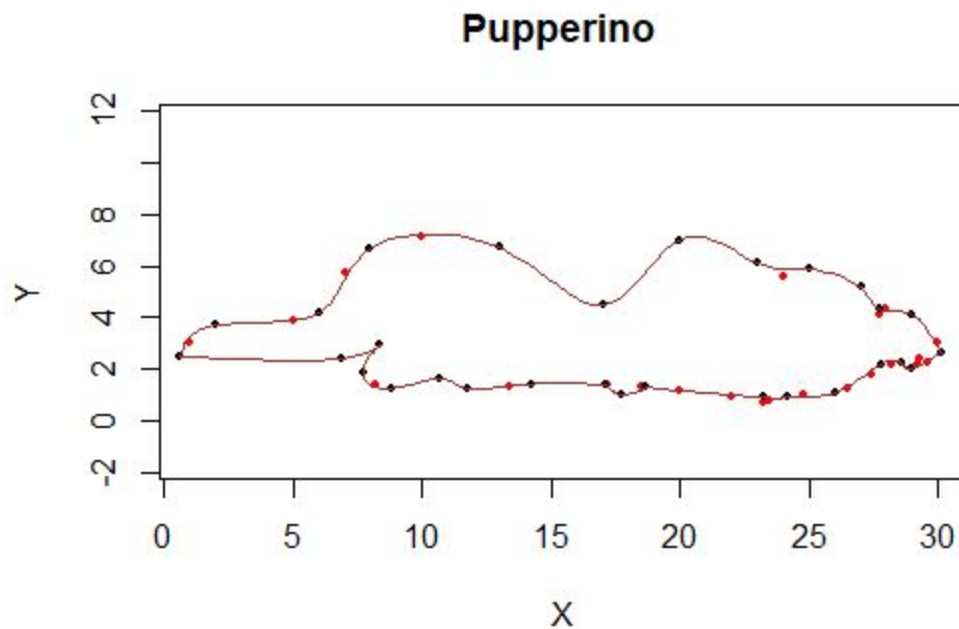
```
xb2o <- c(18.7, 17.74 , 17.1)
```

```
yb2o <- c(1.28 , 1 , 1.39)
```

```
xb3 <- c(30.12, 28.96 , 28.56 , 27.78 , 26 , 24.2 , 23.2 , 18.7)
```

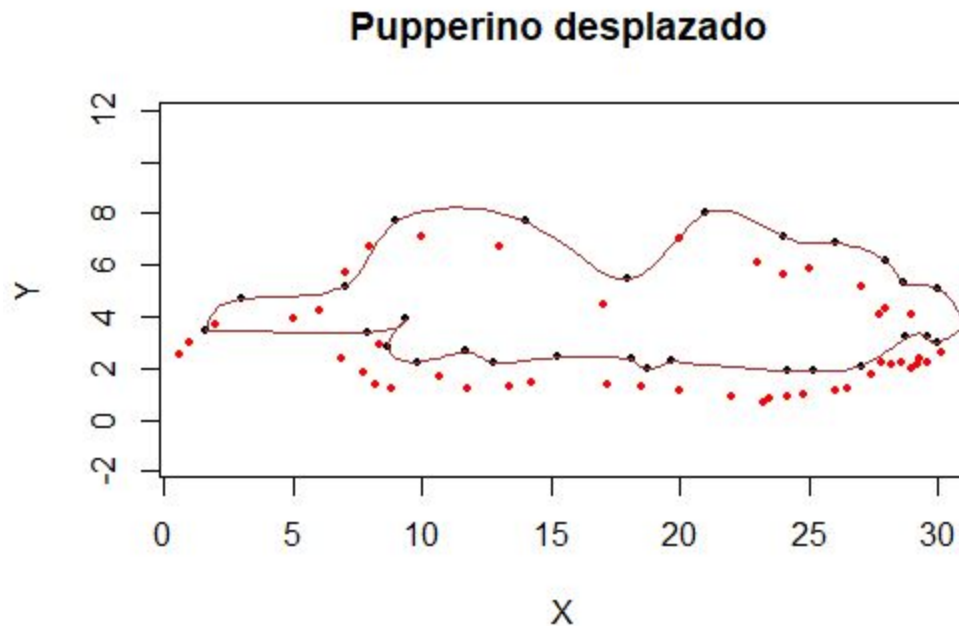
```
yb3 <- c(2.64, 2.02 , 2.22 , 2.2 , 1.10 , 0.9 , 0.9 , 1.28)
```

la graficación de estos puntos, muestra la siguiente imagen:



Como podemos observar, no existe un cambio notable con respecto al aspecto del perrito, demostrando que el código implementado no requiere de un orden estricto para poder graficar correctamente. Esto quiere decir que se podrá ingresar de varias maneras los puntos(derecha a izquierda, izquierda a derecha, etc), pero el dibujo no presentará un cambio radical que perjudique la calidad del mismo

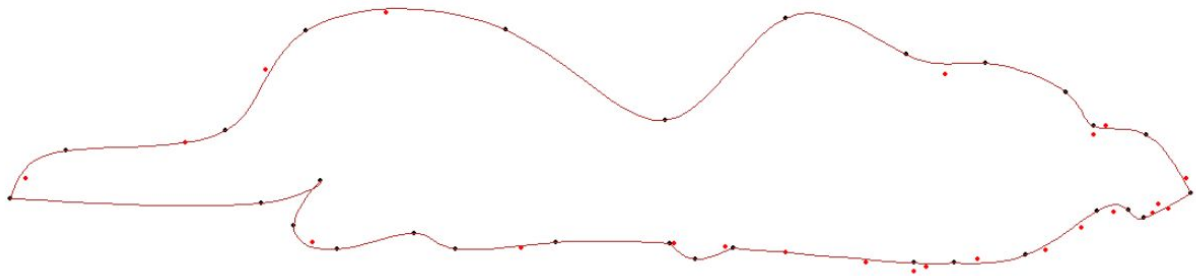
Adicionalmente, se decidió desplazar el origen de la figura para comprobar que la graficación de los puntos no dependiera de la ubicación de los mismos



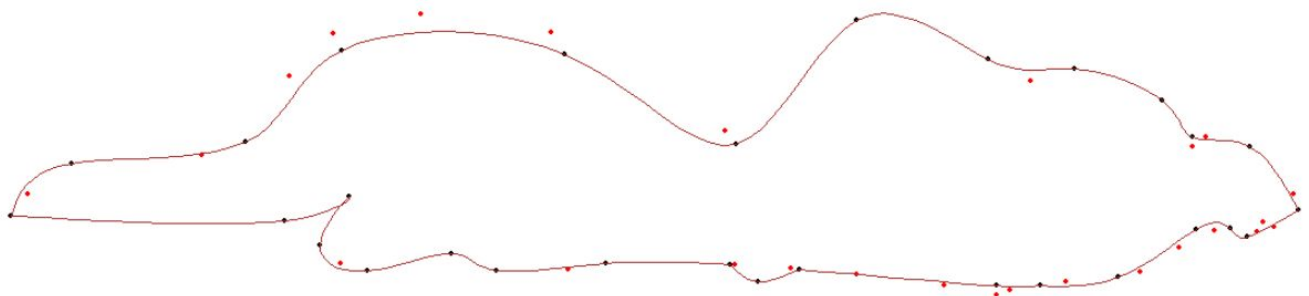
los puntos rojos son los los puntos originales no interpolados, sirven como guia, mostrando el desplazamiento que se hizo de la figura original, se aumentó 1 una unidad en x, así como también se hizo en y

2. ¿Si tenemos nueva información ósea nodos como podemos implementar esa información en el algoritmo de interpolación?

Para responder esta pregunta, se cambiarán puntos en la parte superior e inferior del perro, esto con la finalidad de observar cambios notorios en la forma de este, los puntos escogidos para cambiar serán los puntos críticos utilizados al momento de seleccionar los puntos ya que estos influyen de mayor forma en la interpolación.

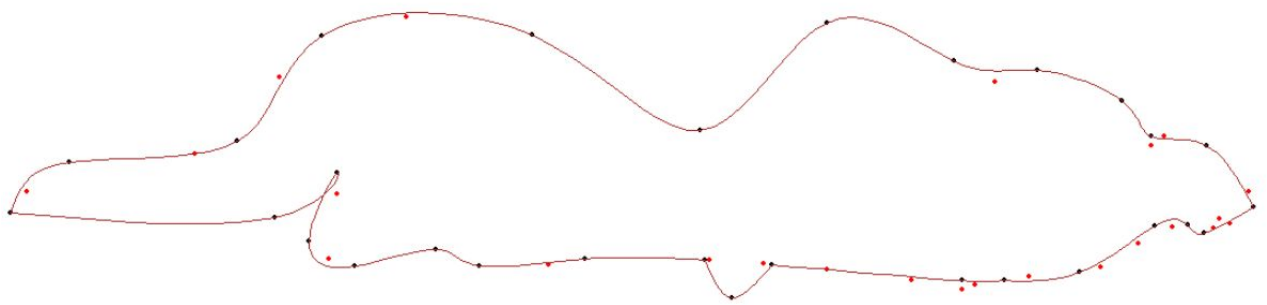


Se cambiaron los puntos de la parte superior (8,6.69), (13,6.7) y (17,4.45) por (8.2,6.3), (13.3,6.2) y (17.23,4.15), esto se realizó con la intención de probar que tanto la integridad de la espalda del perro se mantendría en forma, y si estos cambios llegarán a afectar la forma de la cabeza.



Como se puede ver, la forma de la espalda se mantiene igual de suave y aunque la forma de la cabeza se deforma un poco, la integridad y suavidad de sus curvas se mantiene.

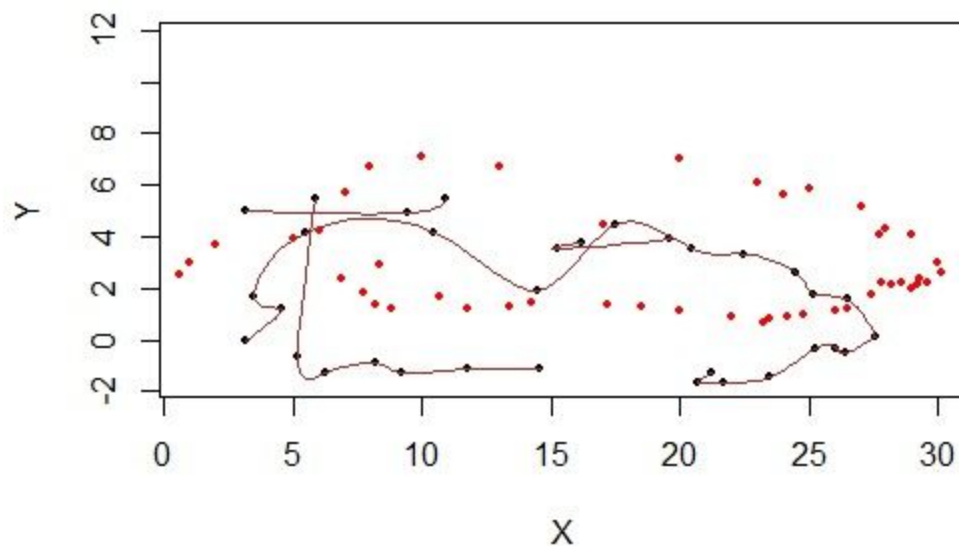
Para el lado inferior del perro, se alejaran los puntos que componen la pata y oreja del perro, esto para ver si produce algún tipo de deformación o anomalía en la forma de este. Los puntos que se alejaran serán (8.38, 2.95) y (17.74,1) por (8.38,3.46) y (17.74,0.5) respectivamente.



Como se puede ver, mientras que los puntos nuevos no se encuentren cerca de otros puntos, el método tratará de suavizar las curvas sin importar que tan alejados se encuentren dichos puntos.

Finalmente, se desea observar qué comportamiento tendrá el gráfico una vez que se modifiquen los puntos teniendo en cuenta la cota de error calculada anteriormente. para esto, se definió que cada punto aumente o disminuya su valor(aleatoriamente) en una magnitud de 2.524.

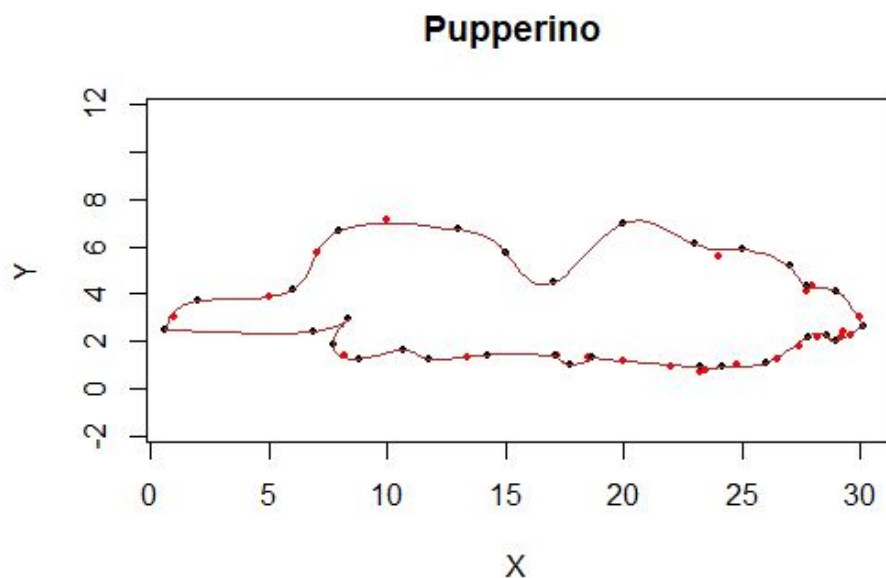
Pupperino erroneo



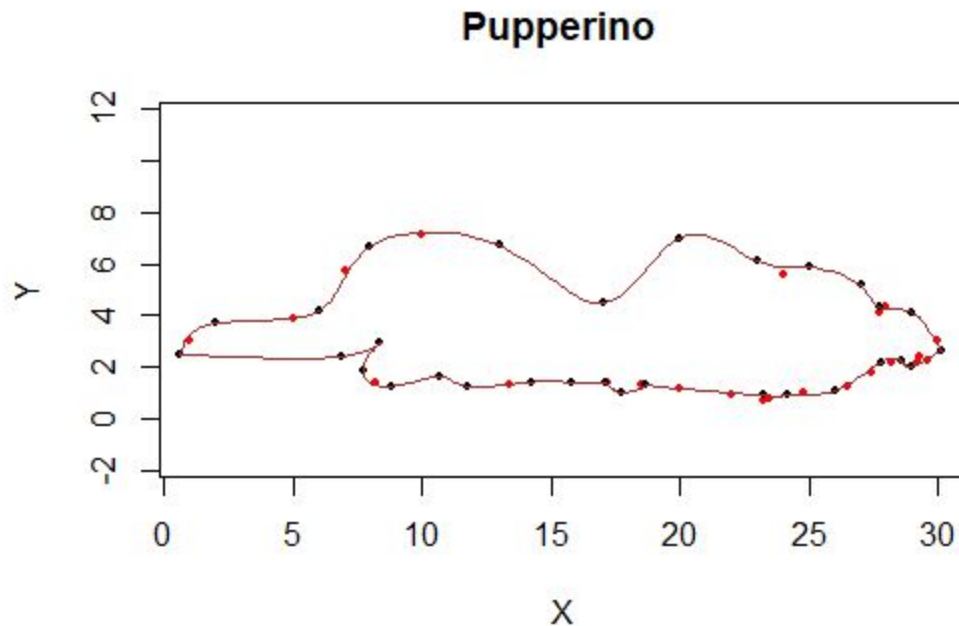
como se puede observar en la imagen, el hecho de que se modifiquen los valores de forma aleatoria puede perjudicar la integridad de la gráfica. Para que esto no se presente, es recomendable que el aumento o decremento de la magnitud de los puntos se haga en un mismo sentido, para que se mantenga la integridad de la figura

3. ¿Su método es robusto, en el sentido que si se tienen más puntos la exactitud no disminuye?

Recordando cómo funciona el método spline cúbico, se evidencia que si se llega a ingresar más puntos, se debe calcular nuevamente el nodo que contiene estos puntos. Esto se observa de mejor manera si el punto agregado es un punto crítico, ya que puede generar anomalías a la hora de graficar por la corta distancia que hay entre estos dos puntos. Pero, si se esta adición de puntos se hace en otros espacios, no afectará la gráfica.



Para este caso, se agregó un punto cerca al lomo del perrito (15,5.7). Se puede observar las anomalías descritas anteriormente. Esta área es definida como crítica por lo cual la adición de un punto y de manera tan cercana, hace que la figura presente una anomalía ligera a la hora de dibujar



Por otro lado, en esta figura se puede ver como la adición de un punto en la parte inferior del perro(15.8,1.4) no presentó anomalías a la hora de dibujar, ya que no hace parte de una zona crítica.

4. ¿Suponga que tiene más puntos con más cifras significativas cómo se comporta su algoritmo? ¿La exactitud decae?

Teniendo en cuenta la naturaleza del método spline cúbico, y los criterios de selección de puntos, se aclara que la adición de más puntos con un mayor número de cifras significativas no afectaría negativamente al funcionamiento del método empleado para graficar.

Sin embargo, se debe tener en cuenta la forma de cómo se ingresan estos puntos, ya que como se ha mencionado anteriormente, si estos puntos no son ingresados de forma correcta(o muy lejos o muy cerca) harán que la gráfica tenga un comportamiento anormal. Si las reglas de ordenamiento de los puntos y de los nodos son cumplidas, el hecho de agregar más puntos con más cifras significativas, hace que la precisión de gráfico aumente, contrario a lo que se podría creer por lo indicado en esta pregunta.

Anexos: tabla de errores, comparacion metodo spline cúbico contra metodo Mono H.FC

	X	Y Spline	MonoH.FC	Error
1	0.6200000	2.500000	2.500000	0.00000000
2	0.6682072	2.655493	2.542974	-0.11251869
3	0.7218564	2.808835	2.593092	-0.21574363
4	0.7863895	2.957875	2.656142	-0.30173358
5	0.8672487	3.100462	2.738607	-0.36185454
6	0.9698758	3.234444	2.847231	-0.38721289
7	1.0997128	3.357670	2.987774	-0.36989632
8	1.2622017	3.467990	3.162175	-0.30581523
9	1.4627846	3.563253	3.362946	-0.20030693
10	1.7069033	3.641306	3.563180	-0.07812624
11	2.0000000	3.700000	3.700000	0.00000000

	X	Y Spline	MonoH.FC	Error
1	2.000000	3.700000	3.700000	0.00000000
2	2.344695	3.738634	3.765675	0.02704175
3	2.732319	3.762312	3.818570	0.05625800
4	3.151384	3.777590	3.857453	0.07986251
5	3.590398	3.791024	3.885821	0.09479734
6	4.037873	3.809169	3.910577	0.10140839
7	4.482318	3.838580	3.940036	0.10145617
8	4.912243	3.885813	3.981654	0.09584077
9	5.316158	3.957424	4.039877	0.08245265
10	5.682574	4.059968	4.114517	0.05454892
11	6.000000	4.200000	4.200000	0.00000000

	X	Y Spline	MonoH.FC	Error
1	6.000000	4.200000	4.200000	0.000000e+00
2	6.260995	4.381662	4.375862	-5.800299e-03
3	6.474307	4.599437	4.636995	3.755848e-02
4	6.652736	4.845394	4.911654	6.626010e-02
5	6.809079	5.111603	5.178778	6.717524e-02
6	6.956133	5.390132	5.440868	5.073602e-02
7	7.106697	5.673050	5.709003	3.595333e-02
8	7.273568	5.952427	5.992518	4.009074e-02
9	7.469543	6.220332	6.287234	6.690257e-02
10	7.707421	6.468833	6.555650	8.681721e-02
11	8.000000	6.690000	6.690000	-8.881784e-16

	X	Y Spline	MonoH.FC	Error
1	8.000000	6.690000	6.690000	0.000000e+00
2	8.356327	6.876948	6.691885	-1.850634e-01
3	8.770451	7.026980	6.693903	-3.330768e-01
4	9.232671	7.138443	6.695917	-4.425261e-01
5	9.733286	7.209685	6.697787	-5.118982e-01
6	10.262594	7.239054	6.699380	-5.396742e-01
7	10.810895	7.224900	6.700578	-5.243212e-01
8	11.368487	7.165569	6.701285	-4.642837e-01
9	11.925669	7.059410	6.701434	-3.579754e-01
10	12.472741	6.904771	6.701000	-2.037708e-01
11	13.000000	6.700000	6.700000	-8.881784e-16

	X	Y Spline	MonoH.FC	Error
1	13.00000	6.700000	6.700000	0.0000000000

2	13.49970	6.446364	6.594638	0.1482742180
3	13.97189	6.156803	6.338991	0.1821882821
4	14.41858	5.847175	6.003629	0.1564541715
5	14.84178	5.533338	5.641128	0.1077902778
6	15.24349	5.231151	5.290076	0.0589252043
7	15.62572	4.956471	4.978384	0.0219126253
8	15.99048	4.725158	4.726023	0.0008647695
9	16.33978	4.553070	4.547271	-0.0057994170
10	16.67562	4.456064	4.452559	-0.0035057987
11	17.00000	4.450000	4.450000	0.0000000000

	X	Y Spline	MonoH.FC	Error
--	---	----------	----------	-------

1	17.00000	4.450000	4.450000	0.00000000
2	17.31488	4.544855	4.556508	0.01165284
3	17.62200	4.727088	4.761462	0.03437402
4	17.92301	4.977278	5.035153	0.05787501
5	18.21960	5.276004	5.352488	0.07648479
6	18.51345	5.603843	5.691521	0.08767823
7	18.80621	5.941375	6.032110	0.09073484
8	19.09958	6.269178	6.354653	0.08547554
9	19.39521	6.567831	6.638858	0.07102772
10	19.69480	6.817912	6.862481	0.04456899
11	20.00000	7.000000	7.000000	0.00000000

	X	Y Spline	MonoH.FC	Error
--	---	----------	----------	-------

1	20.00000	7.000000	7.000000	0.000000e+00
2	20.31177	7.099965	7.047727	-5.223833e-02
3	20.62813	7.124843	7.027709	-9.713484e-02

4	20.94637	7.086962	6.952179	-1.347827e-01
5	21.26381	6.998647	6.835747	-1.629006e-01
6	21.57773	6.872228	6.694199	-1.780290e-01
7	21.88543	6.720030	6.543292	-1.767372e-01
8	22.18421	6.554381	6.397609	-1.567718e-01
9	22.47136	6.387608	6.269531	-1.180765e-01
10	22.74420	6.232039	6.168415	-6.362333e-02
11	23.00000	6.100000	6.100000	8.881784e-16

	X	Y	Spline	MonoH.FC	Error
1	23.00000	6.100000	6.100000	0.00000000	
2	23.23705	6.000904	6.058736	0.05783242	
3	23.45750	5.932498	6.032102	0.09960476	
4	23.66450	5.889615	6.014235	0.12462064	
5	23.86117	5.867087	6.001003	0.13391587	
6	24.05065	5.859747	5.989371	0.12962444	
7	24.23609	5.862427	5.976914	0.11448738	
8	24.42060	5.869960	5.961412	0.09145201	
9	24.60733	5.877178	5.940493	0.06331511	
10	24.79942	5.878914	5.911282	0.03236781	
11	25.00000	5.870000	5.870000	0.00000000	

	X	Y	Spline	MonoH.FC	Error
1	25.00000	5.870000	5.870000	0.00000000	
2	25.21104	5.846360	5.823218	-0.023142027	
3	25.42987	5.808286	5.778686	-0.029600203	
4	25.65264	5.757160	5.732618	-0.024542308	
5	25.87552	5.694365	5.680762	-0.013603879	

6	26.09466	5.621285	5.619246	-0.002039024
7	26.30623	5.539302	5.545416	0.006114103
8	26.50639	5.449799	5.458550	0.008750699
9	26.69130	5.354159	5.360363	0.006204160
10	26.85712	5.253765	5.255222	0.001457254
11	27.00000	5.150000	5.150000	0.000000000

	X	Y	Spline	MonoH.FC	Error
--	---	---	--------	----------	-------

1	27.00000	5.150000	5.150000	0.000000000
2	27.11749	5.044316	5.027343	-0.016972546
3	27.21263	4.938439	4.895430	-0.043008721
4	27.28985	4.834165	4.777367	-0.056798173
5	27.35356	4.733291	4.678260	-0.055031483
6	27.40820	4.637613	4.595480	-0.042132863
7	27.45818	4.548925	4.523918	-0.025006904
8	27.50793	4.469024	4.458712	-0.010311674
9	27.56186	4.399705	4.397128	-0.002577650
10	27.62441	4.342765	4.340860	-0.001905946
11	27.70000	4.300000	4.300000	0.000000000

	X	Y	Spline	MonoH.FC	Error
--	---	---	--------	----------	-------

1	27.70000	4.300000	4.300000	0.000000000
2	27.79190	4.272258	4.274104	0.001846403
3	27.89880	4.256599	4.251004	-0.005595464
4	28.01826	4.249138	4.232373	-0.016765723
5	28.14783	4.245989	4.218393	-0.027596121
6	28.28504	4.243265	4.207769	-0.035496119
7	28.42744	4.237079	4.197967	-0.039112425

8	28.57260	4.223546	4.185644	-0.037902462
9	28.71804	4.198780	4.167202	-0.031578044
10	28.86133	4.158893	4.139407	-0.019486230
11	29.00000	4.100000	4.100000	0.000000000

	X	Y	Spline	MonoH.FC	Error
1	29.00000	4.100000	4.100000	0.000000e+00	
2	29.13213	4.019244	4.027164	7.920297e-03	
3	29.25791	3.917884	3.911568	-6.316094e-03	
4	29.37804	3.798211	3.767647	-3.056357e-02	
5	29.49321	3.662512	3.606441	-5.607130e-02	
6	29.60415	3.513078	3.436381	-7.669781e-02	
7	29.71156	3.352198	3.263941	-8.825715e-02	
8	29.81614	3.182161	3.094190	-8.797092e-02	
9	29.91861	3.005256	2.931254	-7.400253e-02	
10	30.01966	2.823773	2.778720	-4.505318e-02	
11	30.12000	2.640000	2.640000	4.440892e-16	

> cont = cont + SplineCubico(xb21,yb21)

	X	Y	Spline	MonoH.FC	Error
1	0.6200	2.500000	2.500000	0.0000000000	
2	1.3668	2.473912	2.472946	-0.0009662742	
3	2.1064	2.448800	2.424295	-0.0245050709	
4	2.8316	2.425638	2.367012	-0.0586250194	
5	3.5352	2.405400	2.313082	-0.0923180231	
6	4.2100	2.389063	2.272495	-0.1165671639	
7	4.8488	2.377600	2.252437	-0.1251627172	

8	5.4444	2.371987	2.256721	-0.1152660286
9	5.9896	2.373200	2.285527	-0.0876733288
10	6.4772	2.382213	2.335466	-0.0467469626
11	6.9000	2.400000	2.400000	0.0000000000

	X	Y	Spline	MonoH.FC	Error
1	6.9000	2.400000		2.400000	0.00000000
2	7.2532	2.427212		2.491585	0.06437233
3	7.5416	2.463200		2.598536	0.13533584
4	7.7724	2.506987		2.695711	0.18872339
5	7.9528	2.557600		2.774246	0.21664605
6	8.0900	2.614063		2.833376	0.21931379
7	8.1912	2.675400		2.875766	0.20036623
8	8.2636	2.740638		2.905109	0.16447133
9	8.3144	2.808800		2.925083	0.11628317
10	8.3508	2.878913		2.939039	0.06012672
11	8.3800	2.950000		2.950000	0.00000000

> cont = cont + SplineCubico(xb22,yb22)

	X	Y	Spline	MonoH.FC	Error
1	8.38000	2.950000		2.950000	0.000000e+00
2	8.39683	2.977473		2.977472	-2.220446e-15
3	8.41264	3.003280		3.003280	8.881784e-16
4	8.42641	3.025758		3.025757	-4.440892e-16
5	8.43712	3.043240		3.043240	-8.881784e-16
6	8.44375	3.054063		3.054063	8.881784e-16
7	8.44528	3.056560		3.056560	-8.881784e-16

8	8.44069	3.049068	3.049067	-1.776357e-15
9	8.42896	3.029920	3.029920	1.776357e-15
10	8.40907	2.997453	2.997453	8.881784e-16
11	8.38000	2.950000	2.950000	0.000000e+00

	X	Y Spline	MonoH.FC	Error
--	---	----------	----------	-------

1	8.38000	2.950000	2.950000	0.000000e+00
2	8.34107	2.886453	2.886452	-8.881784e-16
3	8.29296	2.807920	2.807920	4.440892e-16
4	8.23669	2.716068	2.716068	8.881784e-16
5	8.17328	2.612560	2.612560	1.776357e-15
6	8.10375	2.499063	2.499063	1.332268e-15
7	8.02912	2.377240	2.377240	4.440892e-16
8	7.95041	2.248757	2.248758	2.220446e-15
9	7.86864	2.115280	2.115280	-8.881784e-16
10	7.78483	1.978473	1.978472	-8.881784e-16
11	7.70000	1.840000	1.840000	0.000000e+00

> cont = cont + SplineCubico(xb23,yb23)

	X	Y Spline	MonoH.FC	Error
--	---	----------	----------	-------

1	7.700000	1.840000	1.840000	0.00000000
2	7.780158	1.747880	1.794984	0.04710382
3	7.862124	1.657767	1.745434	0.08766687
4	7.947708	1.571669	1.690998	0.11932903
5	8.038718	1.491593	1.631368	0.13977496
6	8.136962	1.419545	1.566558	0.14701208
7	8.244249	1.357535	1.497279	0.13974400

8	8.362388	1.307567	1.425460	0.11789300
9	8.493187	1.271651	1.354988	0.08333733
10	8.638455	1.251793	1.292737	0.04094448
11	8.800000	1.250000	1.250000	0.00000000

	X	Y	Spline	MonoH.FC	Error
--	---	---	--------	----------	-------

1	8.800000	1.250000	1.250000	0.00000000
2	8.978623	1.267263	1.237351	-0.02991162
3	9.171089	1.300502	1.255637	-0.04486477
4	9.373155	1.345621	1.301989	-0.04363214
5	9.580578	1.398524	1.369022	-0.02950173
6	9.789115	1.455114	1.446344	-0.00877007
7	9.994522	1.511295	1.522443	0.01114894
8	10.192558	1.562970	1.586691	0.02372074
9	10.378978	1.606044	1.631151	0.02510718
10	10.549540	1.636419	1.651956	0.01553652
11	10.700000	1.650000	1.650000	0.00000000

	X	Y	Spline	MonoH.FC	Error
--	---	---	--------	----------	-------

1	10.70000	1.650000	1.650000	0.0000000000
2	10.82770	1.643979	1.627428	-0.0165506493
3	10.93632	1.620705	1.592691	-0.0280142725
4	11.03112	1.583817	1.553712	-0.0301052426
5	11.11737	1.536953	1.513371	-0.0235814131
6	11.20033	1.483750	1.471933	-0.0118173883
7	11.28526	1.427847	1.428539	0.0006921688
8	11.37743	1.372883	1.382343	0.0094601362
9	11.48210	1.322495	1.333749	0.0112541312

10	11.60454	1.280321	1.286316	0.0059953448
----	----------	----------	----------	--------------

11	11.75000	1.250000	1.250000	0.0000000000
----	----------	----------	----------	--------------

	X	Y	Spline	MonoH.FC	Error
--	---	---	--------	----------	-------

1	11.75000	1.250000	1.250000	0.0000000000
---	----------	----------	----------	--------------

2	11.92238	1.234191	1.228614	-0.005577157
---	----------	----------	----------	--------------

3	12.12003	1.231636	1.216700	-0.014936294
---	----------	----------	----------	--------------

4	12.33996	1.240100	1.216990	-0.023109723
---	----------	----------	----------	--------------

5	12.57914	1.257345	1.230460	-0.026885555
---	----------	----------	----------	--------------

6	12.83457	1.281136	1.256169	-0.024967556
---	----------	----------	----------	--------------

7	13.10323	1.309236	1.291296	-0.017940666
---	----------	----------	----------	--------------

8	13.38212	1.339409	1.331354	-0.008054850
---	----------	----------	----------	--------------

9	13.66822	1.369418	1.370568	0.001149598
---	----------	----------	----------	-------------

10	13.95851	1.397027	1.402371	0.005343673
----	----------	----------	----------	-------------

11	14.25000	1.420000	1.420000	0.0000000000
----	----------	----------	----------	--------------

	X	Y	Spline	MonoH.FC	Error
--	---	---	--------	----------	-------

1	14.25000	1.420000	1.420000	0.0000000000
---	----------	----------	----------	--------------

2	14.54020	1.436587	1.426137	-0.01045018
---	----------	----------	----------	-------------

3	14.82875	1.446989	1.428340	-0.01864953
---	----------	----------	----------	-------------

4	15.11585	1.451893	1.427363	-0.02452957
---	----------	----------	----------	-------------

5	15.40167	1.451985	1.423934	-0.02805113
---	----------	----------	----------	-------------

6	15.68639	1.447955	1.418755	-0.02919970
---	----------	----------	----------	-------------

7	15.97021	1.440487	1.412506	-0.02798088
---	----------	----------	----------	-------------

8	16.25329	1.430271	1.405855	-0.02441589
---	----------	----------	----------	-------------

9	16.53583	1.417993	1.399455	-0.01853728
---	----------	----------	----------	-------------

10	16.81801	1.404340	1.393955	-0.01038455
----	----------	----------	----------	-------------

11	17.10000	1.390000	1.390000	0.0000000000
----	----------	----------	----------	--------------

```
> cont = cont + SplineCubico(xb2o,yb2o)
```

	X	Y	Spline	MonoH.FC	Error
1	17.10000	1.390000		1.390000	0.000000000
2	17.15608	1.334418		1.353806	0.019388876
3	17.21264	1.279840		1.314001	0.034160515
4	17.27016	1.227272		1.271346	0.044073209
5	17.32912	1.177720		1.226656	0.048935562
6	17.39000	1.132188		1.180906	0.048718065
7	17.45328	1.091680		1.135360	0.043680369
8	17.51944	1.057203		1.091724	0.034521549
9	17.58896	1.029760		1.052321	0.022561433
10	17.66232	1.010358		1.020320	0.009962118
11	17.74000	1.000000		1.000000	0.000000000

	X	Y	Spline	MonoH.FC	Error
1	17.74000	1.0000000		1.0000000	0.000000000
2	17.82232	0.9993575		0.9930108	-0.006346678
3	17.90896	1.0077600		0.9975963	-0.010163653
4	17.99944	1.0242025		1.0134256	-0.010776862
5	18.09328	1.0476800		1.0394677	-0.008212324
6	18.19000	1.0771875		1.0740330	-0.003154526
7	18.28912	1.1117200		1.1148421	0.003122110
8	18.39016	1.1502725		1.1591182	0.008845714
9	18.49264	1.1918400		1.2036999	0.011859884
10	18.59608	1.2354175		1.2451706	0.009753056
11	18.70000	1.2800000		1.2800000	0.000000000

```
> cont = cont + SplineCubico(xb4, yb4)
```

	X	Y	Spline	MonoH.FC	Error
1	18.70000	1.2800000		1.2800000	0.000000000
2	19.24814	1.2311341		1.2287616	-0.002372566
3	19.79032	1.1829268		1.1710249	-0.011901965
4	20.32062	1.1360366		1.1116118	-0.024424768
5	20.83307	1.0911220		1.0549644	-0.036157530
6	21.32173	1.0488415		1.0047728	-0.044068691
7	21.78065	1.0098537		0.9636842	-0.046169438
8	22.20388	0.9748171		0.9331181	-0.041698939
9	22.58549	0.9443902		0.9132049	-0.031185304
10	22.91951	0.9192317		0.9028607	-0.016370962
11	23.20000	0.9000000		0.9000000	0.000000000

	X	Y	Spline	MonoH.FC	Error
1	23.20000	0.9000000		0.9	0.000000000
2	23.42346	0.8870751		0.9	0.0129248780
3	23.59619	0.8797229		0.9	0.0202770732
4	23.72691	0.8769307		0.9	0.0230692683
5	23.82438	0.8776859		0.9	0.0223141463
6	23.89732	0.8809756		0.9	0.0190243902
7	23.95447	0.8857873		0.9	0.0142126829
8	24.00458	0.8911083		0.9	0.0088917073
9	24.05638	0.8959259		0.9	0.0040741463
10	24.11860	0.8992273		0.9	0.0007726829
11	24.20000	0.9000000		0.9	0.000000000

	X	Y Spline	MonoH.FC	Error
1	24.20000	0.9000000	0.9000000	0.000000e+00
2	24.30731	0.8977854	0.9000424	2.257012e-03
3	24.43932	0.8943415	0.9004701	6.128606e-03
4	24.59283	0.8919805	0.9020789	1.009836e-02
5	24.76463	0.8930146	0.9061730	1.315834e-02
6	24.95151	0.8997561	0.9145550	1.479889e-02
7	25.15027	0.9145171	0.9294273	1.491026e-02
8	25.35770	0.9396098	0.9532113	1.360157e-02
9	25.57061	0.9773463	0.9882981	1.095174e-02
10	25.78577	1.0300390	1.0367529	6.713905e-03
11	26.00000	1.1000000	1.1000000	-1.110223e-15

	X	Y Spline	MonoH.FC	Error
1	26.00000	1.100000	1.100000	0.000000000
2	26.21047	1.188483	1.189987	0.001503886
3	26.41596	1.292511	1.309587	0.017076204
4	26.61563	1.408047	1.446713	0.038665903
5	26.80864	1.531056	1.590549	0.059493748
6	26.99415	1.657500	1.731811	0.074310885
7	27.17134	1.783344	1.862937	0.079592432
8	27.33935	1.904553	1.978213	0.073660578
9	27.49736	2.017089	2.073822	0.056733302
10	27.64452	2.116917	2.147813	0.030896584
11	27.78000	2.200000	2.200000	0.000000000

	X	Y Spline	MonoH.FC	Error
1	27.78000	2.200000	2.200000	0.000000e+00

2	27.90322	2.263301	2.233341	-2.996024e-02
3	28.01459	2.307774	2.252952	-5.482195e-02
4	28.11479	2.335370	2.262445	-7.292574e-02
5	28.20449	2.348043	2.264674	-8.336872e-02
6	28.28437	2.347744	2.261875	-8.586870e-02
7	28.35510	2.336425	2.255775	-8.065075e-02
8	28.41736	2.316040	2.247687	-6.835281e-02
9	28.47181	2.288539	2.238591	-4.994739e-02
10	28.51913	2.255875	2.229198	-2.667686e-02
11	28.56000	2.220000	2.220000	-4.440892e-16

	X	Y	Spline	MonoH.FC	Error
--	---	---	--------	----------	-------

1	28.56000	2.220000	2.220000	0.00000000
2	28.59528	2.182773	2.208600	0.02582719
3	28.62665	2.145674	2.194055	0.04838071
4	28.65596	2.110092	2.177543	0.06745129
5	28.68507	2.077413	2.159091	0.08167852
6	28.71585	2.049024	2.138162	0.08913789
7	28.75017	2.026314	2.114139	0.08782451
8	28.78988	2.010669	2.086928	0.07625851
9	28.83684	2.003477	2.057961	0.05448441
10	28.89293	2.006125	2.031962	0.02583703
11	28.96000	2.020000	2.020000	0.00000000

	X	Y	Spline	MonoH.FC	Error
--	---	---	--------	----------	-------

1	28.96000	2.020000	2.020000	0.00000000
2	29.03939	2.046048	2.026797	-0.019251001
3	29.13034	2.083450	2.046914	-0.036536176

4	29.23157	2.130943	2.082753	-0.048189585
5	29.34179	2.187266	2.135181	-0.052084986
6	29.45972	2.251159	2.203344	-0.047814732
7	29.58407	2.321358	2.284650	-0.036707788
8	29.71355	2.396603	2.374909	-0.021694162
9	29.84689	2.475633	2.468602	-0.007031262
10	29.98280	2.557186	2.559271	0.002085500
11	30.12000	2.640000	2.640000	0.000000000