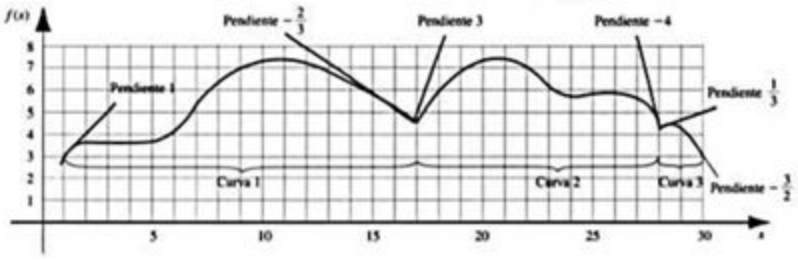
Análisis Numérico
Mateo Galvis
Sergio Posada
29/03/2019

INTERPOLACIÓN CON LAGRANGE BARICÉNTRICA

Utilizando el método de Lagrange Baricéntrica se logró graficar la silueta de un perro que está acostado; la imagen es la siguiente:



Se programó una solución interpolante en RStudio cuyo código está adjunto en la carpeta en la que se encuentra este documento con sus respectivos pasos y procedimientos requeridos.

1.Métodos utilizados en la solución del problema:

Se utilizó el método de Lagrange Baricéntrica y el Ajuste polinómico para la interpolación de los diferentes puntos para elaborar la respectiva figura del perro. Del primer método se obtienen los valores experimentales a partir de los valores teóricos siendo una implementación racional. Después, con estos nodos calculados obtenemos polinomio característicos que representa la función en dicho intervalo de puntos aplicamos el segundo método. Para finalizar, determinamos los siguientes aspectos: gráfica de la figura, error relativo por cada punto, error total relativo acumulado , nivel de eficiencia y la cota de Jaccard.

2. Implementación de las funciones

```
#Función de interpolación

DibujarLinea<-function(inicio, final,yCalculado)
{
  xi0 = x[inicio:final]                          #Determinar que puntuos inciales x voy a usar.
  yi0 = y[inicio:final]                          #Determinar que puntuos inciales y voy a usar.
  x0 <- seq(x[inicio], x[final],len=(length(xi0)+1))  #Secuencia entre los puntos, con n+1 nodos distintos
  y0 <- barylag(xi0, yi0, x0)                    #Método de Lagrange Baricentrica, para la interpolación
  lines(x0, y0, col="blue")                      #Linealización de la gráfica dado los puntos encontrados
  Ajuste_Polinomio = poly.calc(x0,y0)            #Cálculo del polinomio caracteristico, con los puntos interpolados
  for(i in inicio:final)
  {
    yCalculado[i]<-Ajuste_Polinomio(y[i])        #Determinar el punto teórico en el polinomio caracteristico
  }
  return(yCalculado)                             #Devoler arreglo con los "y"  experimentales
}
```

Nota: DibujarLinea: Es la función encargada de realizar la respectiva interpolación de los datos y calcular el polinomio característico para el intervalo de puntos.

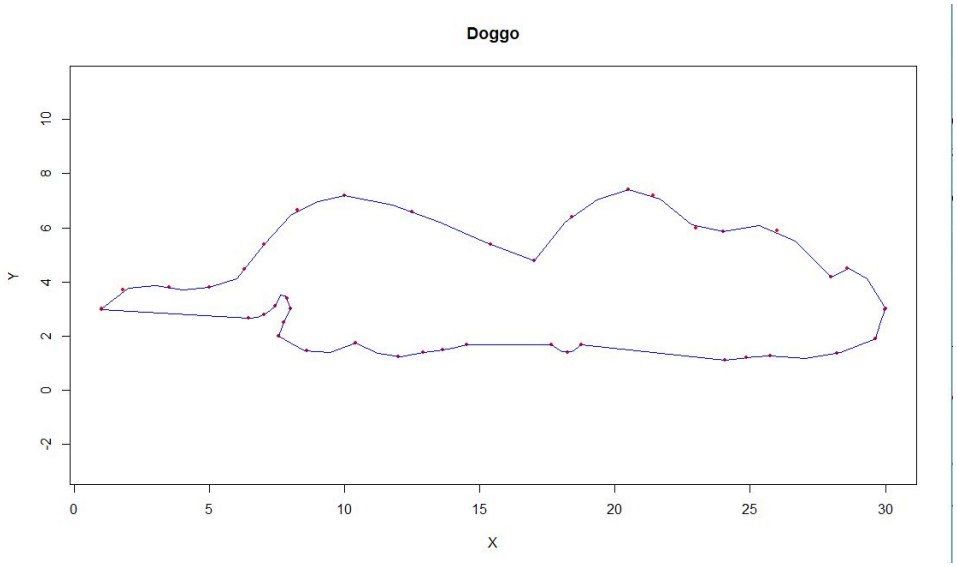
```
CalcularError<-function(inicio,final,yCalculado,Error)
{

    for(i in inicio:final)
    {
     Error[i]<-(abs(y[i]-yCalculado[i])/y[i])      #Cálcular el error que se genera este valor
                                                   #experimental, respecto al teórico
    }

  return(Error)                                       #Devoler arreglo con los errores
}
```

Nota: CalcularError: Calcula el error entre los puntos teóricos versus los puntos experimentales.

3.Resultados

Gráfica:

La gráfica final que se obtuvo de la silueta fue la siguiente:



Nodos y Errores:

A continuación se muestra la siguiente tabla: los nodos teóricos en el eje (X ,Y), el Y experimental medido en ese punto y el error relativo entre los valores originales y calculados.

	Datos 1	Datos 2	Datos 3	Datos 4	Datos 5	Datos 6	Datos 7
X	1.000000e+00	1.800000e+00	3.500000e+00	5.000000e+00	6.300000e+00	8.250000e+00	7.000000e+00
Y	3.000000e+00	3.700000e+00	3.800000e+00	3.800000e+00	4.480000e+00	6.650000e+00	5.400000e+00
Y1	3.874044e+00	3.766731e+00	3.751162e+00	7.307440e+00	4.585360e+00	4.936035e+00	3.718039e+00
Error	2.913480e-01	1.803527e-02	1.285217e-02	9.230105e-01	2.351783e-02	2.577391e-01	3.114743e-01
	Datos 8	Datos 9	Datos 10	Datos 11	Datos 12	Datos 13	Datos 14
X	1.000000e+01	1.250000e+01	1.540000e+01	1.700000e+01	1.840000e+01	2.050000e+01	2.140000e+01
Y	7.190000e+00	6.600000e+00	5.400000e+00	4.790000e+00	6.400000e+00	7.400000e+00	7.200000e+00
Y1	6.659137e+00	6.319255e+00	5.326696e+00	-2.104477e+03	-1.190361e+03	-8.060587e+02	-8.735918e+02
Error	7.383347e-02	4.253716e-02	1.357486e-02	4.403479e+02	1.869939e+02	1.099269e+02	1.223322e+02
	Datos 15	Datos 16	Datos 17	Datos 18	Datos 19	Datos 20	Datos 21
X	2.300000e+01	2.400000e+01	2.600000e+01	2.800000e+01	2.860000e+01	3.000000e+01	1.010000e+00
Y	6.000000e+00	5.850000e+00	5.900000e+00	4.180000e+00	4.500000e+00	3.000000e+00	2.990000e+00
Y1	-1.379921e+03	-7.551985e+01	-7.509546e+01	-4.752565e+02	-4.627818e+02	-5.226771e+02	2.866250e+00
Error	2.309868e+02	1.390938e+01	1.372804e+01	1.146977e+02	1.038404e+02	1.752257e+02	4.138796e-02
	Datos 22	Datos 23	Datos 24	Datos 25	Datos 26	Datos 27	Datos 28
X	6.450000e+00	7.000000e+00	7.440000e+00	7.860000e+00	8.000000e+00	7.740000e+00	7.560000e+00
Y	2.650000e+00	2.800000e+00	3.100000e+00	3.400000e+00	3.000000e+00	2.500000e+00	2.000000e+00
Y1	8.444215e+00	7.989256e+00	-1.317500e+02	-1.146990e+02	-6.615385e+00	-1.205556e+01	1.399898e+01
Error	2.186496e+00	1.853306e+00	4.350000e+01	3.473499e+01	3.205128e+00	5.822222e+00	5.999489e+00
	Datos 29	Datos 30	Datos 31	Datos 32	Datos 33	Datos 34	Datos 35
X	8.600000e+00	1.040000e+01	1.197000e+01	1.290000e+01	1.363000e+01	1.452000e+01	1.765000e+01
Y	1.450000e+00	1.760000e+00	1.240000e+00	1.400000e+00	1.500000e+00	1.700000e+00	1.700000e+00
Y1	1.601548e+01	2.237933e+01	2.457914e+01	7.441435e+00	7.327163e+00	1.700000e+00	2.693462e+02
Error	1.004516e+01	1.171553e+01	1.882189e+01	4.315311e+00	3.884775e+00	1.697988e-15	1.574390e+02
	Datos 36	Datos 37	Datos 38	Datos 39	Datos 40	Datos 41	Datos 42
X	1.826000e+01	1.876000e+01	2.406000e+01	2.486000e+01	2.575000e+01	2.820000e+01	2.962000e+01
Y	1.400000e+00	1.700000e+00	1.100000e+00	1.200000e+00	1.270000e+00	1.360000e+00	1.900000e+00
Y1	2.791754e+02	3.631321e+00	-1.673117e+01	-1.659081e+01	5.889325e+01	5.848653e+01	-7.976162e+01
Error	1.984110e+02	1.136071e+00	1.621015e+01	1.482568e+01	4.537264e+01	4.200480e+01	4.297980e+01
	Datos 43						
X	2.999000e+01						
Y	2.990000e+00						
Y1	-7.655054e+01						
Error	2.660219e+01						

Nota: En total se utilizaron 43 puntos para obtener la interpolación adecuada.

Error relativo Total y eficiencia

El error relativo total del programa fue del: 26.60219%
El número de operaciones realizadas en la interpolación es: 1092 operaciones lógicas.

Índice de Jaccard

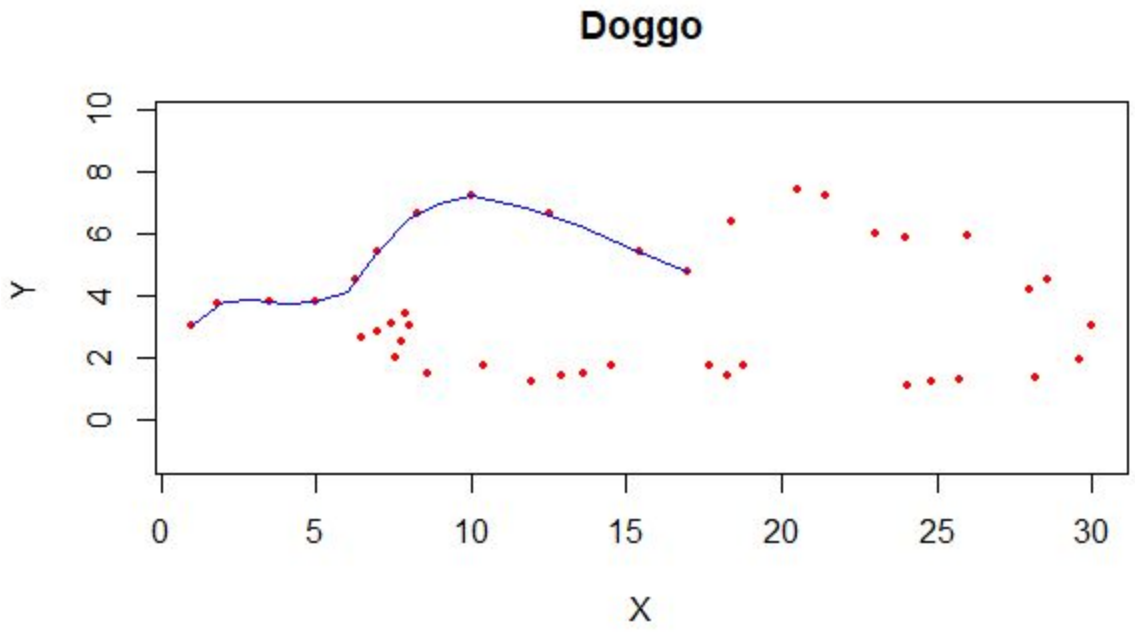
Para la implementación del índice de Jaccard se hallaron los valores teóricos que se aproximaban por lo menos con 1 décima de diferencia a los valores interpolados y también los que no cumplían con esta condición para lograr encontrar el índice de Jaccard que está dado por la siguiente función:
$J = |A \cap B| / |A \cup B|$
Donde se divide el número de aciertos encontrados según la condición que aplicamos (1 décima) sobre el total de puntos que se tienen.
La implementación de la función fue la que se muestra a continuación:

```
jaccard<-function(y, yCalculado)
{
  aciertos<-0
  desaciertos<-0
  for(i in 1: length(y))
  {
    if(y[i]==yCalculado[i] || (y[i]-yCalculado[i]<=1&&y[i]-yCalculado[i]>=0)  ||
       (yCalculado[i]-y[i]<=1&&yCalculado[i]-y[i]>=0))
    {
      aciertos<-aciertos+1
    }
    else
    {
      desaciertos<-desaciertos+1
    }
  }
  cat("Número de aciertos: ", aciertos)
  cat("Número de desaciertos: ", desaciertos)
  cat("Índice de Jaccard: ", aciertos/length(y))
}
```

Preguntas propuestas:

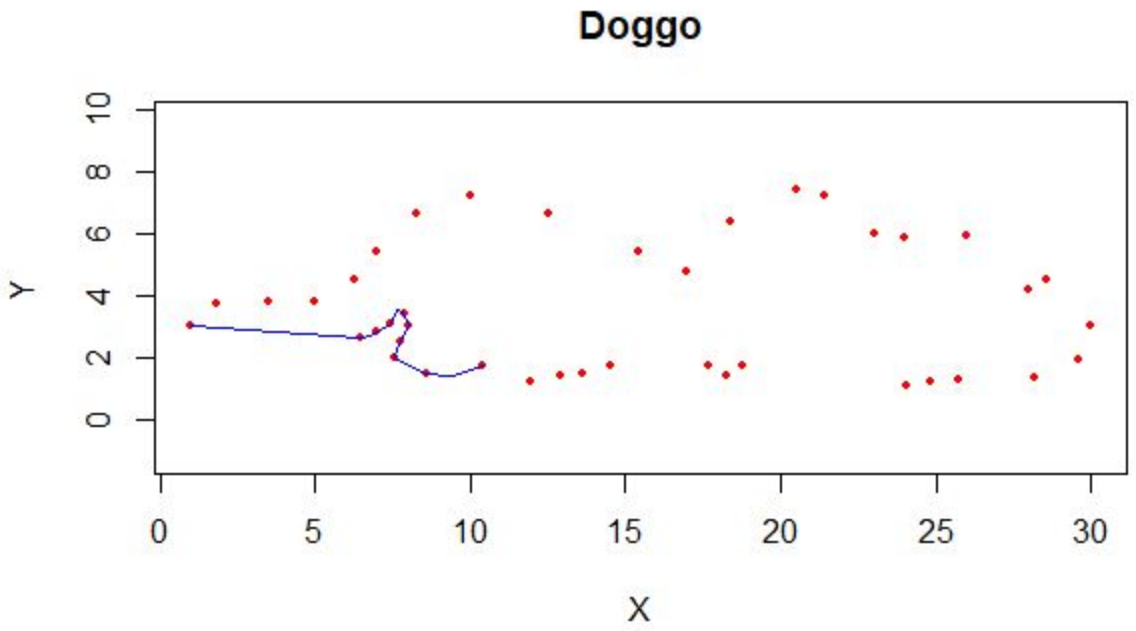
1. **El origen se puede modificar?**
   El origen se puede modificar dado que se creó una función que toma un punto inicial y un punto final para graficar con el método lagrange baricéntrica, el grado del polinomio que se genere en esta va a cambiar dependiendo de la cantidad de puntos que se le ingresen a esta. Para confirmar esto se mostrarán algunos de los puntos tomados inicialmente para graficar y cómo empieza la gráfica del perro para cada caso.
   - Origen desde la cola del perro (Parte superior):

```
#Parte superior del perro

yCalculado=DibujarLinea(1,4,yCalculado)
Error=CalcularError(1,4,yCalculado,Error)

yCalculado=DibujarLinea(4,8,Error)
Error=CalcularError(4,8,yCalculado,Error)

yCalculado=DibujarLinea(8,11,yCalculado)
Error=CalcularError(8,11,yCalculado,Error)

yCalculado=DibujarLinea(11,16,yCalculado)
Error=CalcularError(11,16,yCalculado,Error)

yCalculado=DibujarLinea(16,18,yCalculado)
Error=CalcularError(16,18,yCalculado,Error)

yCalculado=DibujarLinea(18,20,yCalculado)
Error=CalcularError(18,20,yCalculado,Error)
```



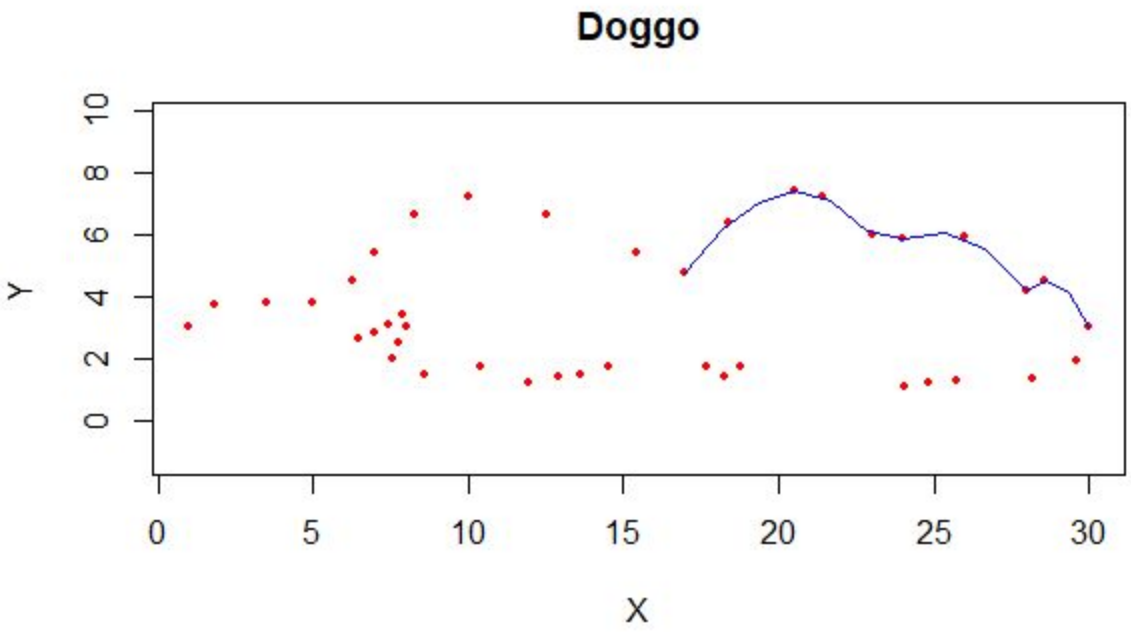
- Origen desde la cola del perro (Parte inferior)

```
yCalculado=DibujarLinea(21,22,yCalculado)
Error=CalcularError(21,22,yCalculado,Error)

yCalculado=DibujarLinea(22,24,yCalculado)
Error=CalcularError(22,24,yCalculado,Error)

yCalculado=DibujarLinea(24,26,yCalculado)
Error=CalcularError(24,26,yCalculado,Error)

yCalculado=DibujarLinea(26,27,yCalculado)
Error=CalcularError(26,27,yCalculado,Error)

yCalculado=DibujarLinea(27,28,yCalculado)
Error=CalcularError(27,28,yCalculado,Error)

yCalculado=DibujarLinea(28,30,yCalculado)
Error=CalcularError(28,30,yCalculado,Error)
```



- Origen desde la cabeza de perro

```
yCalculado=DibujarLinea(20,18,yCalculado)
Error=CalcularError(20,18,yCalculado,Error)

yCalculado=DibujarLinea(18,16,yCalculado)
Error=CalcularError(18,16,yCalculado,Error)

yCalculado=DibujarLinea(16,11,yCalculado)
Error=CalcularError(16,11,yCalculado,Error)

yCalculado=DibujarLinea(11,8,yCalculado)
Error=CalcularError(11,8,yCalculado,Error)

yCalculado=DibujarLinea(8,4,Error)
Error=CalcularError(8,4,yCalculado,Error)

yCalculado=DibujarLinea(4,1,yCalculado)
Error=CalcularError(4,1,yCalculado,Error)
```



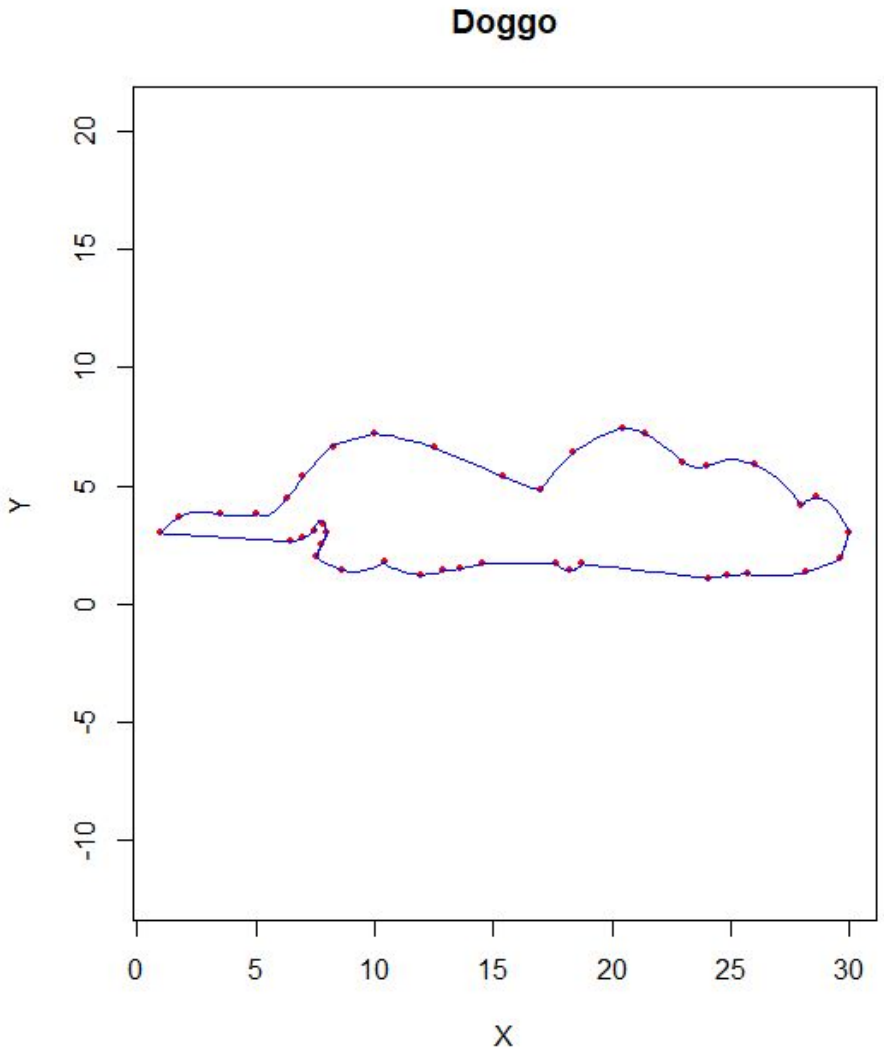
2. **Si tenemos nueva información osea nodos como podemos implementar esa información en el algoritmo de interpolación?**

   Si se agrega una nueva información se debe ingresar en los respectivos arreglos del eje x y del eje y en cualquier ordenen. Puesto que, la función no necesita de un origen específico, sino que puede partir de cualquier punto como se vio en la parte de arriba para realizar la respectiva gráfica. Lo único que se necesita es un nodo inicio y final que determine el trayecto para aplicar las dos métodos obteniendo los nuevos valores experimentales, polinomios característicos y errores.

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

   Nuestro método es de carácter no robusto debido a que con un mayor número de puntos en la gráfica se logra obtener una mejor interpolación. Con un más puntos las curvas tiende a ser más suaves y homogéneas comparado con el dibujo inicial. No obstante, al realizar varias ejecuciones determinados la cantidad mínima de puntos para la correcta funcionalidad del método para encontrar el polinomio característico del mismo con n+1 nodos distintos; logrando una precisión aceptable.

   Se corrió el algoritmo generando una secuencia de 920 nodos para la interpolación y el resultado es el siguiente:



Se obtuvo la misma gráfica pero las curvas del perro son mucho más suaves en comparación con de utilización de solo 43 puntos. La gráfica interpolada mejora considerablemente.
Para realizar el método con estos 920 puntos se modificó la función que dibuja un línea por el método lagrange de la siguiente manera:

```
DibujarLinea<-function(inicio, final,yCalculado)
{
  xi0 = x[inicio:final]
  yi0 = y[inicio:final]
  x0 <- seq(x[inicio], x[final],len=20)  #Secuencia
  y0 <- barylag(xi0, yi0, x0)
  lines(x0, y0, col="blue")
  Ajuste_Polinomio = poly.calc(x0,y0)
  for(i in inicio:final)
  {
    yCalculado[i]<-Ajuste_Polinomio(y[i])
  }
  return(yCalculado)
}
```

Se puede evidenciar que para cada llamado se tomarían 20 puntos por cada nodo.

4. **Suponga que tiene más puntos con más cifras significativas como se comporta su algoritmo ? la exactitud decae?**

   Con un mayor número de cifras significativas el algoritmo se comporta de una mejor manera, no decae en la exactitud. El método implementado es la Lagrange Baricéntrica combinado con un ajuste polinomio. Al tener un mayor números de cifras el error de redondeo disminuye considerablemente y permite trabajar con un error de menor grado.

```r
jaccard<-function(yCalculado, yAjustado)
{
  aciertos<-0
  desaciertos<-0
  for(i in 1:length(yCalculado))
  {
    x<-0
    for(j in 1:length(yAjustado))
    {
      if(yCalculado[i]==yAjustado[j])
      {
        x<-1
      }
    }
    if(x==1)
    {
      aciertos<-aciertos+1
    }
    else
    {
      desaciertos<-desaciertos+1
    }
  }

  cat("Numero de Aciertos: ",aciertos)
  cat("Numero de Desaciertos: ",desaciertos)
}
```