Minimizar el error de interpolación considerando las raíces del polinomio de Chebyshev

Stefano Nasini

Dept. of Statistics and Operations Research
Universitat Politécnica de Catalunya

Un polinomio de interpolación es un polinomio que pasa exactamente a través de un conjunto dado de puntos. Supongamos que lo que se quiere es buscar un polinomio de grado finito que aproxime una función dada. Lo que resulta intuitivo es buscar que dicho polinomio tenga el mismo valor de la función en un conjunto de puntos dado.

Sabiendo que por *n* puntos pasa un único polinomio de grado *n-1*, podríamos argumentar que la única manera de buscar una aproximación mejor del polinomio a la función es la de escoger de formas distintas los puntos por los cuales el polinomio ha de pasar.

Dada una función *f* de la cual se conocen sus valores en un número finito de abscisas $x_0,x_1,...,x_m$, se llama interpolación polinómica al proceso de hallar un polinomio $p_m(x)$ de grado menor o igual a m, cumpliendo $p_m(x_k) = f(x_k)$ por cada k = 1, . . ., m.

Los coeficientes a_0 , a_1, a_2, . . ., a_n, de dicho polinomio se obtiene imponiendo al polinomio de pasar por los puntos fijados.

$$\begin{bmatrix} x_0^n & x_0^{n-1} & x_0^{n-2} & \dots & x_0 & 1 \\ x_1^n & x_1^{n-1} & x_1^{n-2} & \dots & x_1 & 1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ x_n^n & x_n^{n-1} & x_n^{n-2} & \dots & x_n & 1 \end{bmatrix} \begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \quad (1)$$

Este sistema es compatible determinado y a la matriz asociada se le suele denominar matriz de Vandermonde. La complejidad computacional para invertir la matriz es de $O(n^3)$. Por esta razón, han sido construido diferentes algoritmos que aprovechan de la particular estructura de de este sistema que reducen la complejidad a $O(n^2)$, como el método de las diferencias divididas de Newton o el método de Lagrange.

En este último caso, el polinomio, el polinomio interpolador de grado n de Lagrange es un polinomio de la forma

$$\sum_{j\in\{0\dots k\}} f_j l_j(x) \quad ; n \le m \quad [2]$$

donde $l_j(x)$ son los llamados polinomios de Lagrange, que se calculan de este modo:

$$l_j(x) = \prod_{j\neq i} \frac{x - x_i}{x_j - x_i} = \frac{(x-x_0)(x-x_1)\dots(x-x_{j-1})(x-x_{j+1})\dots(x-x_n)}{(x_j-x_0)(x_j-x_1)\dots(x_j-x_{j-1})(x_j-x_{j+1})\dots(x_j-x_n)} \quad [3]$$

Hagamos un ejemplo de polinomio interpolador de grado n de Lagrange. Se quiere hallar el valor de la función *f(x) = exp(x+1)* utilizando un polinomio interpolador de Lagrange de grado 2 que pase por los tres puntos *(0,f(0), (0.5,f(0.5)), (1,f(1)).*

Se usa primero el método directo para calcular el polinomio interpolador de Lagrange. Con las condiciones dadas, los polinomios de Lagrange son:

$$l_0(x) = \frac{(x-0.5)(x-1)}{0.5} = 2x^2 - 3x + 1$$

$$l_1(x) = \frac{x(x-1)}{-0.25} = -4x^2 + 4x \quad [4]$$

$$l_2(x) = \frac{x(x-0.5)}{0.5} = 2x^2 - x$$

Por consiguiente, el polinomio de interpolación de grado dos resultará es siguiente.

$$p_2(x) = \sum_{j\in\{0\ldots2\}} f_j l_j(x) = (2e - 4e^{3/2} + 2e^2)x^2 + (-3e - 4e^{3/2} + 2e^2)x + e \qquad [5]$$

Una pregunta que puede surgir al utilizar un polinomio de interpolación para aproximar una función es cuanto bueno es el ajuste del polinomio a la función originaria. Por esta razón consideramos el error de interpolación de un polinomio de grado *n* que pase por los puntos de una función f(x) en las abscisas $x_0,\ldots,x_n$.

Si f(x) es una función determinada en $x_0,\ldots,x_n$ y *n* veces diferenciable, entonces el error de interpolación puede calcularse como valor absoluto de la diferencia entre la función y el polinomio. Construimos una función *Φ(x)* por la cual se cumpla que

$$\phi(x) = f(x) - P_n(x) - a(x)(x-a_0)(x-a_0)\ldots(x-a_n)$$

$$\exists\bar{x}\in[-1,1] \quad a(\bar{x}) = [f(\bar{x}) - P_n(\bar{x})](\bar{x}-a_0)(\bar{x}-a_1)\ldots(\bar{x}-a_n) = 0 \qquad [6]$$

Esta función se anula en *n+2* puntos. Aplicando el teorema de Rolle[1] se tiene que una función que toma el mismo valor *n+2* veces tiene *n+1* puntos que anulan la derivada. A la vez, la derivada de esta función es tal que, teniendo *n+1* puntos con el mismo valor tendrá *n* puntos que anulan su derivada. Por lo tanto, derivando sucesivamente n+1 veces, tenemos que existirá un único punto ζ que anule la derivada (n+1)esima, es decir, $\phi^{(n+1}(\zeta) = 0$. Así, podemos asegurar que $\phi^{(n+1}$,tiene almenos una raíz, con lo cual resulta evidente que, siendo $p_n(x)$ un polinomio de grado manor que n-1, $\phi^{(n+1}(\zeta)$ resultará la siguiente.

$$\phi(x)^{(n+1)} = f(\bar{x})^{(n+1)} - a(\bar{x})(n+1)! \qquad [7]$$

Por consiguiente, habiendo puesto que en $\phi(\bar{x})^{(n+1)} = 0$, se tiene que

[1] Sea *f*:[a,b] → *R*. Si *f(x)* es continua en [a,b], derivable in *(a,b)* e *f(a) = f(b)* entonces existe un punto en [a, b] donde *f'(x) =0*.

$$a(\bar{x}) = \frac{f(\bar{x})^{(n+1)}}{(n+1)!} \qquad [8]$$

y en el caso de que f(x) sea n veces diferenciable en el dominio [-1, 1], el error de interpolación podrá definirse como

$$f(x) - P_n(x) \; = \; \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_i (x - x_i) \qquad [9]$$

Donde ξ es un punto que pertenece a [-1,1] por el cual $\phi^{(n+1}(\zeta) = 0$. Notemos que el resultado es trivialmente verdadero cuando *x = xi* ya que ambos lados de la expresión serán 0.

> **Corollarios** Condiciones de interpolación. Para cualquier valor de i, el error es 0 cuando *x*=*x*$_i$, ya que $\prod_i (x - x_i) = 0$.
>
> El error es 0 cuando los datos son medidas de un polinomio f(x) de exacto grado n poque entonces la (*n+1*)ésima derivada es igual a 0.

Adjudicando valores absolutos en la expresión del error de interpolación y maximizando ambos lados de la inecuación a lo largo del intervalo [-1,1] obtenemos la cota para dicho error:

$$\max_{-1 \le x \le 1} | f(x) - p_n(x) | \quad = \quad \left| \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_i (x - x_i) \right| \quad \le \quad \frac{\max_{-1 \le x \le 1} | f^{(n+1)}(x) |}{(n+1)!} \underbrace{\max_{-1 \le x \le 1} \prod_i (x - x_i)}_{\text{queremos minimizar este factor}} \qquad [10]$$

Dada la unicidad del polinomio de interpolación, las únicas dos cosas que podemos mover a la hora de reducir el error de interpolación es el grado del polinomio (por consiguiente, el número de puntos) y la localización de dichos puntos.

Se podría creer que al crecer del grado del polinomio el error de interpolación se reduzca. En realidad, pese al ser un resultado antiintuitivo, Carle David Tolmé Runge observó que el error de interpolación en un dato intervalo tiende a infinito cuando el grado del polinomio de interpolación tiende a infinito.

$$\lim_{n \longrightarrow \infty} \left(\max_{-1 \le x \le 1} | f(x) - p_n(x) | \right) = \infty \qquad [11]$$

El métodos que ilustraremos nos permiten proporcionar los puntos por los cuales hacer pasar el polinomio de interpolación de forma tal que la distancia máxima entre el polinomio interpolado y la función originaria sea mínima.

La oscilación observada por Runge se puede minimizar usando nodos de Chebyshev en lugar de equidistantes. En este caso se garantiza que el error máximo disminuye al crecer el orden polinómico. Esta es una propiedad que hace particularmente interesante el utilizar las raíces del polinomio de Chebyshev como puntos por donde interpolar el polinomio.

Para minimizar el último factor de la cota del error proporcionada en [10], Pafnuty Lvovich Chebyshev demostró que los puntos $x_0,...,x_n$ por los cuales hacer pasar el polinomio han de ser escogido de forma que

$$\max_{-1\le x\le 1}\prod_i (x-x_i) \quad = \quad \frac{1}{2^n}T_{n+1}(x) \qquad [12]$$

donde, $T_{n+1}(x)$ es el polinomio de Chebyshev de grado $n+1$. El polinomio de Chebyshev de primera especie es el

A partir de los polinomios de grado 0 y 1

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \end{aligned} \qquad [13]$$

el polinomio de Chebyshev de grado n se obtiene por medio de la siguiente definición recursiva.

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \qquad [14]$$

Con lo cual

$$
\begin{aligned}
T_0(x) &= 1 \\
T_1(x) &= x \\
T_2(x) &= 2x^2 - 1 \\
T_3(x) &= 4x^3 - 3x \\
T_4(x) &= 8x^4 - 8x^2 + 1 \\
T_5(x) &= 16x^5 - 20x^3 + 5x \\
T_6(x) &= 32x^6 - 48x^4 + 18x^2 - 1 \\
T_7(x) &= 64x^7 - 112x^5 + 56x^3 - 7x \\
T_8(x) &= 128x^8 - 256x^6 + 160x^4 - 32x^2 + 1 \\
T_9(x) &= 256x^9 - 576x^7 + 432x^5 - 120x^3 + 9x \\
&\vdots \\
T_n(x) &= 2xT_{n-1}(x) - T_{n-2}(x)
\end{aligned}
$$

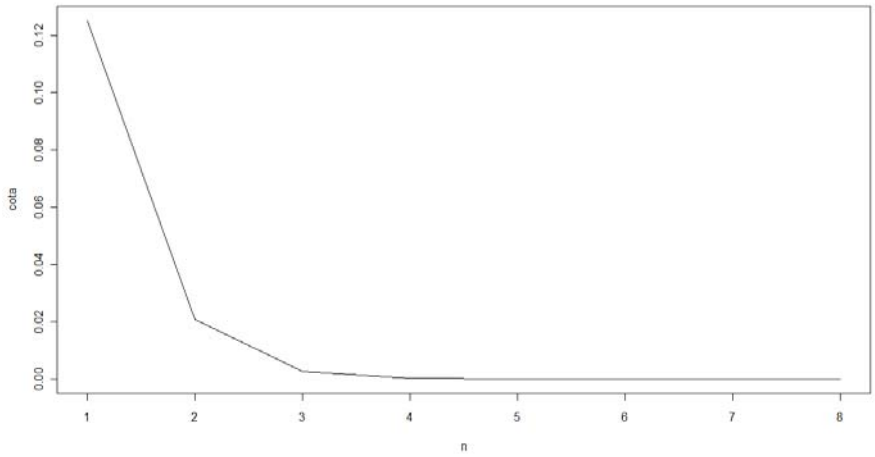
Entre todas las elecciones de los puntos $x_0,...,x_n$, elegirlos de forma que [7] se respete, garantiza que el polinomio así obtenido es el único que tenga la propiedad según la cual

$$
\begin{aligned}
\max_{-1\le x\le 1} T_n(x) &\le \max_{-1\le x\le 1} \prod_i (x - x_i) \\
\max_{-1\le x\le 1} T_n(x) &= \frac{1}{2^n} \\
\frac{1}{2^n} &\le \max_{-1\le x\le 1} \prod_i (x - x_i)
\end{aligned}
\qquad [15]
$$

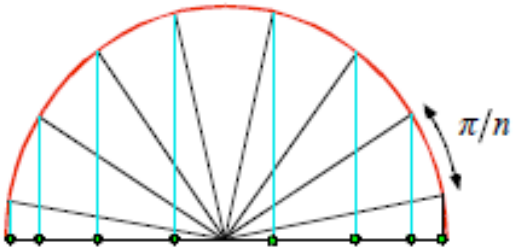
Por ende, se puede demostrar que el valor absoluto de la diferencia entre la función y el polinomio interpolado por las raíces del polinomio de Chebyshev resulta acotado de la siguiente forma:

$$
\left| f(x) - P_{n-1}(x) \right| \le \frac{1}{2^{n-1} n!} \max_{\xi \in [-1,1]} \left| f^{(n)}(\xi) \right| \qquad [16]
$$

Abajo mostramos la variación de dicha cota en el caso de f(x) = sin(x) aproximado en [-1, 1]. Como se nota, al aumentar del grado del polinomio la cota del error disminuye monótonamente.



Una interpretación geométrica de los nodos de Chebyshev es aquella según la cual estos se colocan en un segmento de longitud igual al diámetro de un círculo, cuya circunferencia repartimos en n partes iguales. Proyectando a lo largo del dicho segmento el punto medio de cada partición de la semicirconferencia obtenemos puntos que coinciden con las raíces del polinomio de Chebyshev.



La razón por la cual la aproximación de una función *f(x)* por un polinomio que interpole puntos escogidos de esta forma minimiza el efecto Runge es que la densidad de puntos resulta creciente desde el centro asta las extremidades. Para calcular dichas raices utiliza la identidad trigonometrica del polinomio de Chebyshev.

$$T_n(x) = \cos\left(n \arccos\ x\right) = \cosh\left(n \operatorname{arccosh}\ x\right)$$

Este coseno se anula cuando la expresión al interior es un múltiplo de 2Π y por lo tanto las raíces del polinomio de Chebyshev en [-1,1] son.

$$x_i = \cos\left(\frac{2i-1}{2n}\pi\right) \quad ; \qquad i = 1 \ldots n.$$

En el caso de que se quisiera definir el polinomio de Chebyshev en un intervalo cualquiera [a, b], las raices resultaran trasformadas de la siguiente forma.

$$x_i = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(\frac{(2i-1)}{2n}\pi\right) \quad ; \qquad i = 0,\ldots,n$$

La utilización de los nodos de Chebyshev nos permite también utilizar un método recursivo para la obtención de los coeficientes.

$$f(x) \approx \sum_{j=0}^{n} c_j T_j(x)$$

donde c_j son

$$c_0 = \frac{1}{n+1}\sum_{k=0}^{n} f(x_k)T_0(x_k) \;=\; \frac{1}{n+1}\sum_{k=0}^{n} f(x_k)$$

$$c_j = \frac{1}{n+1}\sum_{k=0}^{n} f(x_k)T_j(x_k)$$

Esta formula permita calcular los coeficientes del polinomio de interpolación con un coste computacional del orden de $O(n^2)$ operaciones.

El siguiente ejemplo muestra una aplicación de la aproximación de una función por polinomio interpolados en los nodos del polinomio de Chebyshev.

Consideramos la función: $f(x) = \frac{800x}{3+54x^4+x^2}$ y interpolamos dos polinomios de igual grado a los puntos de dicha función. El primer polinomio lo interpolamos en puntos equiespaciados y el segundo en las raíces del polinomio de Chebyshev. Utilizamos dos medidas de distancia para evaluar la bondad de la aproximación:

$$d(f,p) = \int_a^b \left(f(x) - p(x)\right)^2 dx$$

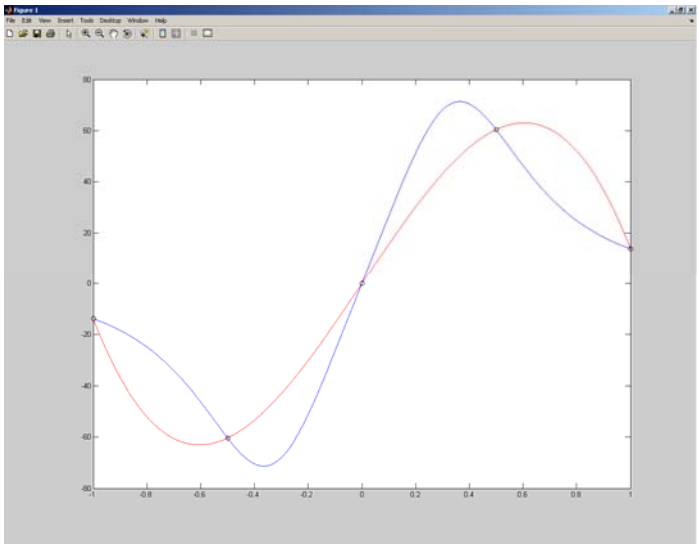
Donde calculamos el área de la diferencia entre la función y el polinomio de interpolación al cuadrado. Y:

$$d(f,p) = \max_x |\, f(x) - p(x)\,|$$

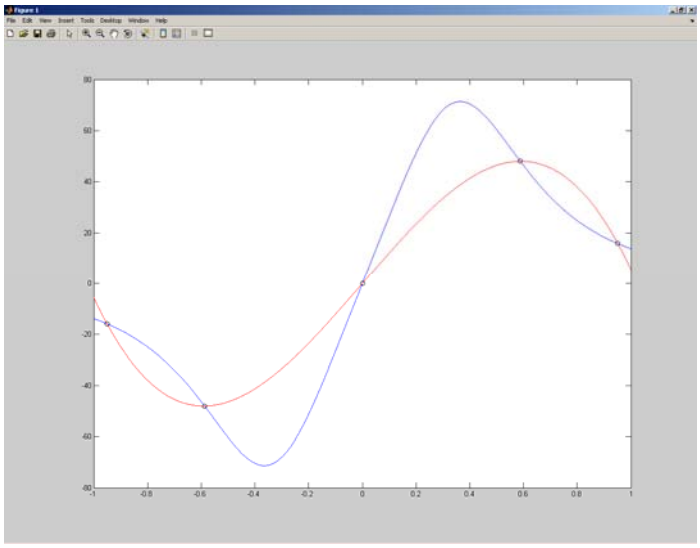
donde hallamos el valor de la máxima distancia.

Lo que resulta interesante en esta aproximación es que, mientras la distancia máxima entre la función y el polinomio resulta siempre menor cuando se utilizan las raíces del polinomio de Chebyshev, lo mismo no ocurre con la norma L2.

Como se nota, la norma L2 entre la función y el polinomio de grado 5 resulta 709 cuando se utilizan puntos uniformemente distribuidos y 715, cuando se utilizan las raíces del polinomio de Chebychev.



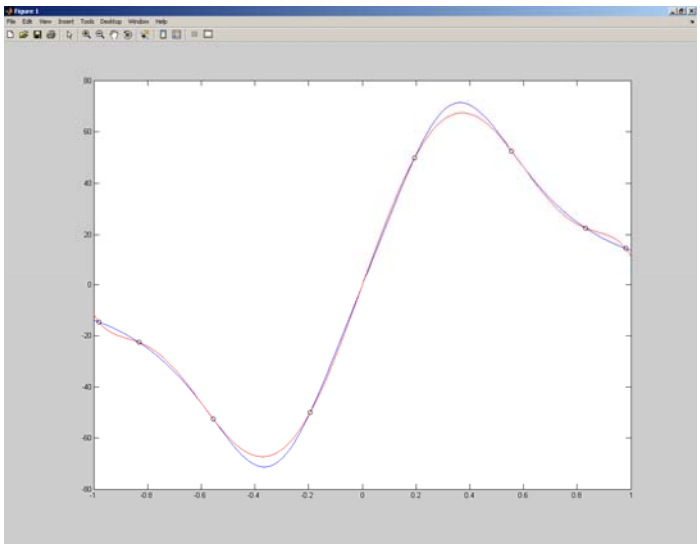
Puntos equidistantes



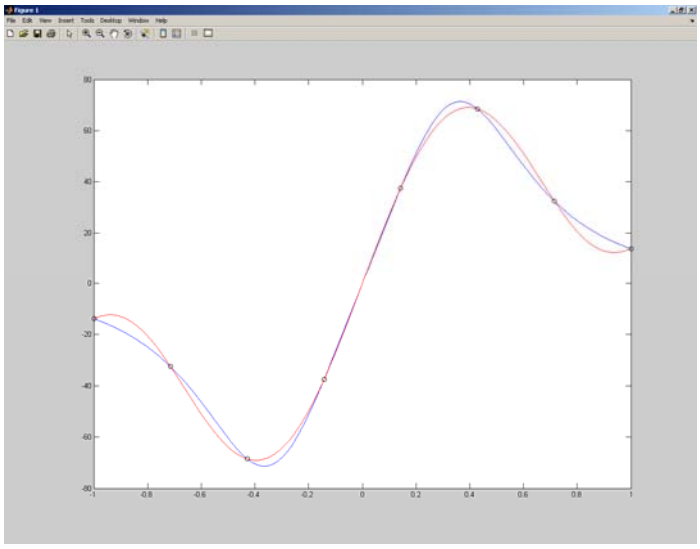
Nodos de Chebyshev

n = 5 L2 = 709 Min max = 34.6612	n = 5 L2 = 715 Min max = 28.1111

Para la interpolación con un polinomio de grado 5, a simple vista no se puede valorar cuales son los puntos con los que se interpola mejor, de hecho, existe una apreciación de máxima distancia en el polinomio hallado por el método de Chebyshev, pero si consultamos las medidas de error aclaramos que no es más que una ilusión óptica.



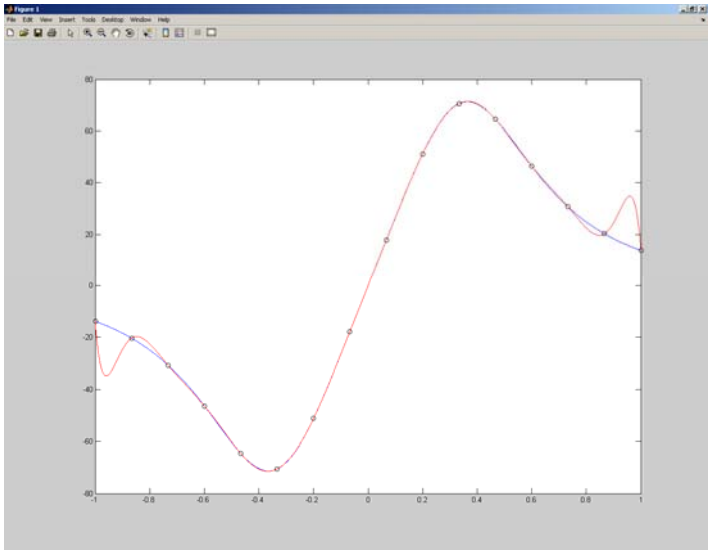
Nodos de Chebyshev



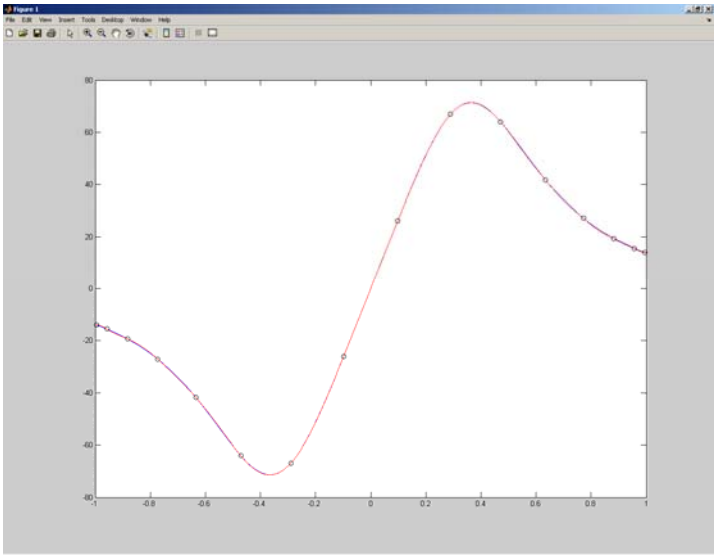
Puntos equidistantes

n = 8 L2 = 20 Min max = 5.7081	n = 8 L2 = 7 Min max = 4.1167

Cuando aumentamos el grado del polinomio, observamos que a pesar de que ambos métodos parecen ajustar bien, el hallado a partir de los nodos de Chebyshev es óptimo ya que podemos distinguir la cantidad de error sobre todo en los extremos, y lo corroboramos consultando las medidas de error.



Puntos equidistantes



Nodos de Chebyshev

n = 16 L2 = 44 Min max = 19.4396	n = 16 L2 = 0.001461 Min max = 0.3212

Por último, aumentamos el grado de polinomio a 16 y en este caso, apreciamos claramente en el primer gráfico una manifestación del fenómeno de Runge, del que ya hablamos anteriormente.

Lo que emerge claramente es que el método de Chebyshev resulta óptimo para la minimización de la mayor distancia existente entre la función que se desea aproximar y nuestro polinomio de Chebyshev (Minimax). Sin embargo, esto no garantiza de ninguna forma la minimización del otras distancias, como se observó en caso de la norma L2.

Un inconveniente considerable de dicho método es que si queremos añadir más nodos, se tendría que estimar de nuevo los coeficientes de los polinomios de Chebyshev.

Codigo MatLab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Sin Chebyshev
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n = 5 % grado del polinomio
x = (-1: 2/(n-1) :1)';

% seleccionamos los puntos equidistantes a partir de los cuales
% hallaremos el polinomio de interpolación

f = @(x) 800*x./(3+54*(x.^4)+(x.^2));
% definimos la función que queremos ajustar mediante el polinomio

y = feval(f,x);

% calculamos los valores de la función f en los puntos x definidos
% anteriormente

p = polyfit(x,y,n-1);

% hallamos los coeficientes del polinomio de interpolación

t = -1:0.001:1; % definimos el dominio
poly = polyval(p,t);

% calculamos los valores y del polinomio de interpolación en x

plot(t,f(t),'b',t,poly,'r',x,f(x),'ok')

%%%%%%%  Norm Min Max %%%%%%
% hallamos la máxima distancia


f = @(x) 800*x/(3+54*(x^4)+(x^2));

%función que queremos aproximar mediante g

g = @(x) p*[x^4 x^3 x^2 x 1]'; %polinomio de interpolación

dist = 0;

        for x = -1:0.0001:1
                dist = max (dist, abs(f(x) - g(x)));
end

dist


%%%%%%%  Norm L2 %%%%%%%%%%
% Calculamos el área de la diferencia entre la
% función y el polinomio de interpolación al cuadrado

syms x
```

```
L2 = int((f(x) - g(x))^2, -1,1)

round(L2)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Con Chebyshev
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n = 5; % grado del polinomio
x = cos((2*(1:n)'-1)*pi/(2*n)); % calculamos los nodos de Chebyshev
f = @(x) 800*x./(3+54*(x.^4)+(x.^2));

% definimos la función que queremos ajustar mediante el polinomio

y = feval(f,x);

% calculamos el valor de la función f en x

p = polyfit(x,y,n-1);

% hallamos los coeficientes del polinomio de interpolación a partir de % los
nodos de Chebyshev

t = -1:0.001:1; %definimos el dominio
poly = polyval(p,t);

% calculamos el valor y del polinomio de interpolación en x

plot(t,f(t),'b',t,poly,'r',x,f(x),'ok')

%%%%%%%  Norm Min Max %%%%%%

f = @(x) 800*x/(3+54*(x^4)+(x^2));

%función que queremos aproximar mediante g

g = @(x) p*[x^4 x^3 x^2 x 1]'; %polinomio de interpolación

dist = 0;

for x = -1:0.0001:1

        dist = [dist, abs(f(x) - g(x))];
end

dist

% hallamos la máxima distancia

%%%%%%%  Norm L2 %%%%%%%%%%

syms x

L2 = int((f(x) - g(x))^2, -1,1)

round(L2)
```

```
% Calculamos el área de la diferencia entre la función y el polinomio % de
interpolación al cuadrado


function [c,x] = chebpolfit1(fname,n)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%  Alternativamente, también podemos calcular el polinomio de interpolación
%  mediante los polinomios de Chebyshev.
%  Calcula los coeficientes c(i) para i=1, 2, …, n del polinomio de
%  interpolación: p(t)=c(1)*T_0(t)+...c(n)*T_{n-1}(t) donde T_j son los
%  polinomios de % Chebyshev de grado j
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x = cos((2*(1:n)'-1)*pi/(2*n)); % nodos de Chebyshev
y = feval(fname,x);
T = [zeros(n,1) ones(n,1)];
c = [sum(y)/n zeros(1,n-1)];
a = 1;
for k = 2:n
T = [T(:,2) a*x.*T(:,2)-T(:,1)];  % polinomios de Chebyshev
c(k) = sum( T(:,2) .* y)*2/n;

% coeficientes para los polinomios de Chebyshev


a = 2;
end

function u = chebpolval(c,t)

% esta función calcula los valores de la combinación lineal de los
% polinomios de
% Chebyshev en todo el dominio de t

n = length(c);
u = c(n)*ones(size(t));
if n > 1
ujp1 = u;
u = c(n-1) + 2*t*c(n);
for j = n-2:-1:1
ujp2 = ujp1;
ujp1 = u;
u = c(j) + 2*t.*ujp1 - ujp2;
end
u = u - t.*ujp1;
end


% y en la ventana de comando  de MatLab escribimos:


n=5 % grado del polinomio

f = @(x) 800*x/(3+54*(x^4)+(x^2)); % declaramos nuestra función

[c,x] = chebpolfit(f,n);
```

```
% calculamos los coeficientes y los polinomios de Chebyshev
t = -1:.01:1; % definimos el dominio en las abcisas
plot(t,f(t),'b',t,chebpolval(c,t),'r',x,f(x),'ok')
% graficamos la función en azul y el polinomio de interpolación en rojo
```