

UNA INTRODUCCIÓN A LA INTERPOLACIÓN POLINOMIAL

F. VADILLO

RESUMEN. Uno de los problemas clásicos del Análisis Matemático es el de construir polinomios que aproximen a una función $f(x)$ en el entorno de un punto x_0 o bien coincida con los valores de la función f en unos nodos determinados x_0, \dots, x_N . En estas notas se explican las interpolación de Taylor y Lagrange con sus correspondiente análisis de errores. Después de comprobar las dificultades para aproximar fundamentalmente en los extremos de los intervalos, se explica la interpolación a trozos con especial interés en los splines cúbicos. Finalmente se comenta brevemente los problemas más relevantes no estudiados con referencias bibliográficas.

ÍNDICE

1. Interpolación de Taylor	2
2. Interpolación de Lagrange	3
2.1. La forma de Lagrange	4
2.2. La forma de Newton: diferencias divididas	4
3. El error de interpolación	5
4. Interpolación a trozos	7
5. Interpolación cúbica a trozos: splines cúbicos	8
6. Interpolación con MATLAB	10
7. Algunos problemas de interpolación no estudiados	11
7.1. Interpolación de Hermite	11
7.2. Interpolación trigonométrica y transformad discreta de Fourier	11
7.3. Interpolación de Chebyshev	11
Referencias	13

1. INTERPOLACIÓN DE TAYLOR

En el problema de interpolación de Taylor (1685-1731) se supone conocidas las N primera derivadas de la función $f(x)$ en un punto x_0 y se trata de encontrar un polinomio $p(x)$ de grado menor o igual que N tal que

$$(1.1) \quad p(x_0) = f(x_0), p'(x_0) = f'(x_0), \dots, p^{(N)}(x_0) = f^{(N)}(x_0).$$

Cualquier texto elemental de Análisis Matemático demuestra que el problema de interpolación de Taylor tiene una única solución que se denomina **N-ésimo polinomio de Taylor** de la función $f(x)$ en el punto x_0 que se escribe de la forma

$$(1.2) \quad \begin{aligned} p_N(x) = & f(x_0) + f'(x_0)(x - x_0) + f''(x_0) \frac{(x - x_0)^2}{2!} + \\ & \dots + f^{(N)}(x_0) \frac{(x - x_0)^N}{N!}, \end{aligned}$$

con la expresión de Lagrange (1736-1813) para el error

$$(1.3) \quad f(x) - p_N(x) = f^{(N+1)}(\xi) \frac{(x - x_0)^{N+1}}{(N+1)!},$$

donde ξ es un punto comprendido estrictamente entre x y x_0 . Vea por ejemplo los clásicos [11] ó [15].

La pregunta que ahora se puede plantear es si aumentando N podemos acercar el valor de $p_N(x)$ tanto como queramos al valor de la función $f(x)$, es decir, si es cierto o no que

$$(1.4) \quad f(x) = \lim_{N \rightarrow \infty} p_N(x) = \lim_{N \rightarrow \infty} \sum_{n=0}^N f^{(n)}(x_0) \frac{(x - x_0)^n}{n!}.$$

Si la función tiene derivadas de todos los ordenes en el punto x_0 , evidentemente se puede construir su serie de Taylor

$$(1.5) \quad \sum_{n=0}^{\infty} f^{(n)}(x_0) \frac{(x - x_0)^n}{n!},$$

Otra cuestión es si esta serie converge o diverge para un valor de x , y en caso de convergencia si la suma es el valor de la función en dicho punto. Por ejemplo, las derivadas de la función

$$(1.6) \quad f(x) = \begin{cases} \exp(-\frac{1}{x^2}) & x \neq 0, \\ 0 & x = 0, \end{cases}$$

en $x_0 = 0$ son todas cero por lo que la serie suma cero en todos los puntos x que evidentemente no coincide con su valor $\exp(-1/x^2)$.

En Matemática Aplicada es muy frecuente utilizar polinomios y series de Taylor de una función dada, si bien, en lugar de calcular las derivadas se utilizan desarrollos conocidos de otras funciones. La tabla siguiente da algunos ejemplos de polinomios de Taylor en $x_0 = 0$ de las funciones que se indican con los correspondientes intervalos de convergencia para su serie de potencias.

$f(x)$	$p_n(x)$	intervalo
$(1+x)^{-1}$	$1-x+\cdots+(-1)^n x^n$	$(-1, 1)$
$\ln(1+x)$	$x-\frac{x^2}{2!}+\cdots+(-1)^{n+1}\frac{x^n}{n!}$	$(-1, 1)$
e^x	$1+x+\cdots+\frac{x^n}{n!}$	$(-\infty, \infty)$
$\sin x$	$x-\frac{x^3}{3!}+\frac{x^5}{5!}-\dots$	$(-\infty, \infty)$
$\cos x$	$1-\frac{x^2}{2!}+\frac{x^4}{4!}-\dots$	$(-\infty, \infty)$

Por otra parte, el manipulador simbólico de MATLAB tiene un comando llamado **taylortool** que abre una ventana gráfica como la de la figura 1 donde además de calcular el polinomio de Taylor lo compara gráficamente con la función en el intervalo que decida el usuario.

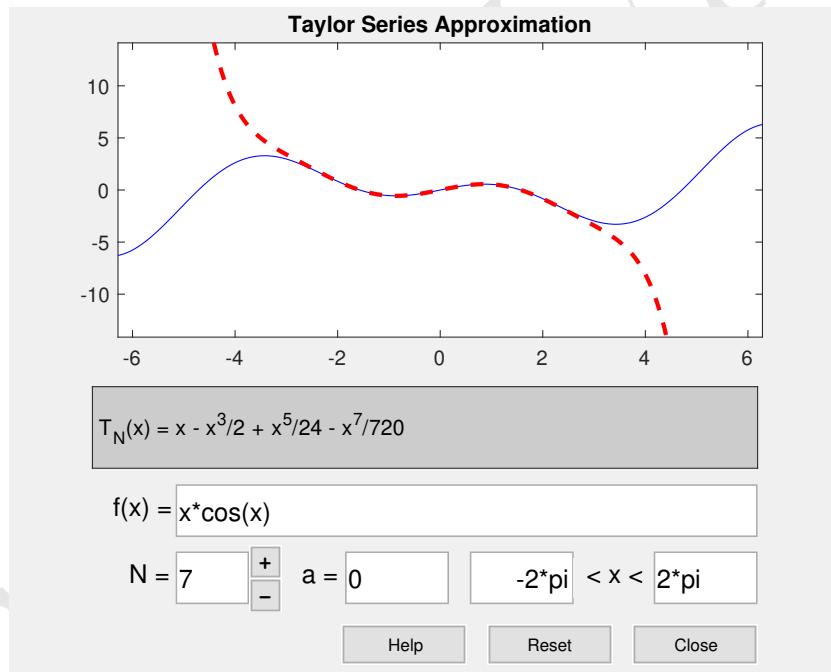


FIGURA 1. Desarrollo de Taylor de $x \cos x$.

2. INTERPOLACIÓN DE LAGRANGE

En el problema de interpolación de Lagrange se dan $N + 1$ puntos x_0, \dots, x_N distintos dos a dos y los correspondientes valores $f(x_0), \dots, f(x_N)$ de una función.

El problema de interpolación de Lagrange construye el polinomio $p(x)$ de grado $\leq N$ tal que

$$(2.1) \quad p(x_0) = f(x_0), \quad p(x_1) = f(x_1), \quad \dots, \quad p(x_N) = f(x_N),$$

donde los puntos x_i son denominados nodos de la interpolación.

Teorema 2.1. *El problema de interpolación de Lagrange tiene solución única.*

*Demuestra*ción. Escribiendo $p_N(x) = a_0 + a_1x + \dots + a_Nx^N$ entonces los coeficientes son la solución del sistema lineal

$$\begin{aligned} a_0 + a_1x_0 + \dots + a_Nx_0^N &= f(x_0), \\ a_0 + a_1x_1 + \dots + a_Nx_1^N &= f(x_1), \\ &\vdots \quad \vdots \quad \vdots \\ a_0 + a_1x_N + \dots + a_Nx_N^N &= f(x_N), \end{aligned}$$

cuya matriz es de tipo Vandermonde(1735-1796) con nodos diferentes y por tanto no singular. \square

2.1. La forma de Lagrange. Aunque el teorema anterior puede construir el polinomio de interpolación de Lagrange es laborioso y por tanto conveniente hacerlo de otra manera. La expresión más compacta es la forma de Lagrange.

Para cada $i = 0, \dots, N$ se define el **polinomio de Lagrange** de la forma

$$(2.2) \quad l_i(x) = \frac{\prod_{j=0, j \neq i}^N (x - x_j)}{\prod_{j=0, j \neq i}^N (x_i - x_j)},$$

tales que $l_i(x_j) = \delta_{ij}$ la delta de Kronecker (1823-1891), de donde se deduce que el polinomio de interpolación es

$$(2.3) \quad p_N(x) = \sum_{i=0}^N f(x_i)l_i(x),$$

llamada **forma de Lagrange del polinomio interpolador**.

2.2. La forma de Newton: diferencias divididas. Otra forma de construir el polinomio de interpolación es debida a Sir Isaac Newton (1643-1727) cuya principal idea es construir la solución en pasos sucesivos, primero se construye p_0 de grado ≤ 0 que coincide con el valor $f(x_0)$, luego $p_1(x)$ de grado ≤ 1 que interpola en x_0 y x_1 por lo que

$$p_1(x) = p_0 + q_1(x), \quad \text{tal que } q_1(x_0) = 0,$$

por lo que $q_1(x) = c_1(x - x_0)$ con c_1 calculado en la igualdad $f(x_0) + c_1(x_1 - x_0) = f(x_1)$.

Después

$$p_2(x) = p_1(x) + q_2(x), \quad \text{tal que } q_2(x_0) = q_2(x_1) = 0,$$

por lo que $q_2(x) = c_2(x - x_0)(x - x_1)\dots$

En definitiva,

$$\begin{aligned} p_N(x) &= c_0 \\ &+ c_1(x - x_0) \\ &+ c_2(x - x_0)(x - x_1) \\ &\vdots \\ &+ c_N(x - x_0)(x - x_1) \cdots (x - x_{N-1}), \end{aligned}$$

donde los coeficientes c_i son las diferencias divididas de la función con la notación

$$\begin{aligned} p_N(x) &= f[x_0] \\ &+ f[x_0, x_1] (x - x_0) \\ &+ f[x_0, x_1, x_2] (x - x_0)(x - x_1) \\ &\vdots \\ &+ f[x_0, x_1, \dots, x_N] (x - x_0)(x - x_1) \cdots (x - x_{N-1}), \end{aligned}$$

Es decir, que la diferencia dividida de orden n es el coeficiente de la potencia $n - sima$ del polinomio de interpolación.

Esta claro entonces que la diferencia dividida de orden cero es el valor de la función, es decir, $f[x_i] = f(x_i)$ y las diferencias de orden $N \geq 1$ pueden calcularse de las de orden $N - 1$ con el siguiente teorema

Teorema 2.2. *Si x_0, \dots, x_N son $N + 1$ puntos distintos dos a dos, entonces*

$$(2.4) \quad f[x_0, x_1, \dots, x_N] = \frac{f[x_1, \dots, x_N] - f[x_0, \dots, x_{N-1}]}{x_N - x_0}.$$

Demostración. Ver [16, pag. 49]. □

Utilizando este teorema se puede construir la siguiente tabla. Las dos primeras columnas son las abscisas y los valores de la función, después cada elemento se obtiene a partir de los que están a su oeste y noroeste.

x_0	$f[x_0]$					
x_1	$f[x_1]$	$f[x_0, x_1]$				
x_2	$f[x_2]$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$			
\vdots	\vdots	\vdots	\vdots	\ddots		
x_N	$f[x_N]$	$f[x_{N-1}, x_N]$	$f[x_{N-2}, x_{N-1}, x_N]$	\cdots	$f[x_0, \dots, x_N]$	

Finalmente, los elementos diagonales son los coeficiente del polinomio interpolador en la forma de Newton.

3. EL ERROR DE INTERPOLACIÓN

Construido el polinomio de interpolación de una u otra manera, ¿Qué diferencia hay entre el valor del polinomio de interpolación en un punto dado x y el verdadero valor de la función $f(x)$?

Teorema 3.1. Suponiendo de $f(x)$ tiene $N \geq 1$ derivadas continuas en un intervalo $[a, b]$ que contiene a los nodos de interpolación y además existe $f^{(N+1)}(x)$ en el intervalo abierto (a, b) . Entonces para cada $x \in [a, b]$ existe un punto ξ con

$$\min\{x_0, \dots, x_N, x\} < \xi < \max\{x_0, \dots, x_N, x\}$$

para el cual

$$(3.1) \quad f(x) - P_N(x) = \frac{f^{(N+1)}(\xi)}{(N+1)!} \prod_{j=0}^N (x - x_j).$$

Demostración. Ver [16, pag. 53]. \square

Según este resultado, si se supone además que $|f^{(N+1)}(x)| \leq K_{N+1}$ en (a, b) , el error de interpolación esta acotado de la forma

$$(3.2) \quad \max_{a \leq x \leq b} |f(x) - P_N(x)| \leq \frac{K_{N+1}}{(N+1)!} \max_{a \leq s \leq b} \prod_{j=0}^N |s - x_j|.$$

Con este resultado, se puede abordar el problema de la convergencia, es decir, estudiar si el error de interpolación $f(x) - p_N(x)$ puede hacerse arbitrariamente pequeño incrementando el número de nodos N . Más concretamente, si $f(x)$ es una función definida en un intervalo $[a, b]$

1. Se elige un punto $x_0^{(0)}$ y se interpola con un polinomio constante p_0 .
2. Se toman dos nodos $x_0^{(1)}, x_1^{(1)}$ y se construye el polinomio lineal $p_1(x)$.
3. Se toman tres nodos $x_0^{(2)}, x_1^{(2)}, x_2^{(2)}$ y se calcula el interpolador $p_2(x)$
4.

¿Será cierto que $\lim_{N \rightarrow \infty} p_N(x) = f(x)$? .

La respuesta en general es negativa, es decir: **aumentar el número de nodos y el grado del polinomio de interpolación no da siempre buenos resultados**. El program `rungeinterp.m` de [13] implementa un famoso problema de interpolación de Runge(1856-1927). La figura 2 compara la función

$$(3.3) \quad f(x) = \frac{1}{1 + 25x^2},$$

y su polinomio de interpolación de orden 7 donde se puede claramente apreciar los grandes errores en los extremos del intervalo. Es decir: **incrementar el grado de los polinomios de interpolación no es siempre aconsejable**. Naturalmente el resultado no es tan pesimista como el ejemplo de Runge, analizando la cota de error (3.2) todo dependerá de como se comporten las derivadas de la función cuando incrementan su orden.

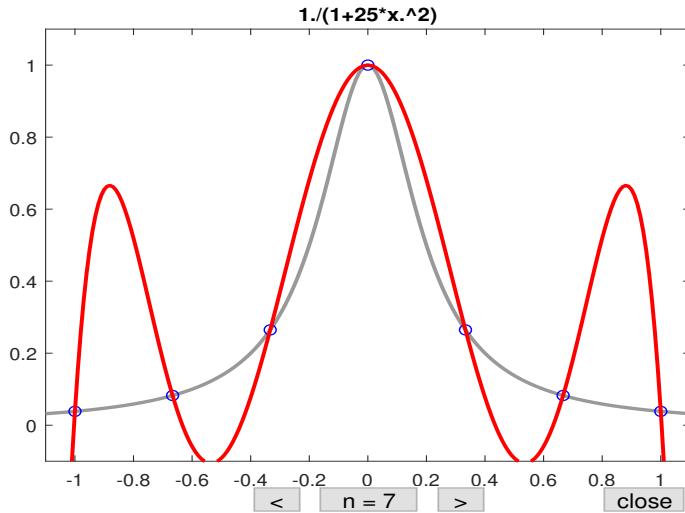


FIGURA 2. Fenómeno de Runge.

4. INTERPOLACIÓN A TROZOS

Como resultado de la divergencia del polinomio de interpolación, para conseguir mejores aproximaciones polinomiales lo habitual es interpolar a trozos. La idea básicamente consiste en tomar una partición del intervalo:

$$(4.1) \quad \Delta : a = x_0 < x_1 < x_2 < \dots < x_N = b,$$

y después interpolar en cada uno de los trozos $[x_k, x_{k+1}]$ para cada $k = 0, \dots, N-1$.

El caso más simple tomaría las líneas recta que unes los puntos de interpolación $(x_k, f(x_k))$ y $(x_{k+1}, f(x_{k+1}))$ de forma que para $x_k \leq x < x_{k+1}$

$$(4.2) \quad s(x) = f(x_k) + (x - x_k) \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k}.$$

Entonces $s(x)$ sería el interpolador lineal a trozos de la función $f(x)$ para la partición Δ , y por construcción la linea quebrada $s(x)$ es continua en todo el intervalo, esto se indica diciendo que $s \in M_0^1(\Delta)$.

Para estimar el error de interpolación se supone que $f(x)$ tiene derivada segunda en $[a, b]$ acotada en valor absoluto por una constante K_2 , y usando el teorema 3.1 se tiene que para $x_k \leq x < x_{k+1}$

$$(4.3) \quad |f(x) - s(x)| \leq \frac{1}{8}(x_{k+1} - x_k)^2 K_2.$$

Llamando ahora $h_k = x_{k+1} - x_k$ y $h = \max_k h_k$ el diámetro de la partición, se tiene

$$(4.4) \quad |f(x) - s(x)| \leq \frac{1}{8}h^2 K_2, \quad \forall x \in [a, b],$$

de donde se concluye que cuando $h \rightarrow 0$, $s(x) \rightarrow f(x)$ con convergencia cuadrática, es decir, cuando se reduce el diámetro de la partición a la mitad, la cota de error se divide por cuatro.

Además de la interpolación lineal a trozos se pueden considerar **funciones cuadráticas a trozos**, es decir, funciones que en cada sub-intervalo $[x_k, x_{k+1}]$ son polinomios de segundo grado. El resultado $q(x)$ es una función cuadrática a trozos y continua que se escribe $q \in M_0^2(\Delta)$.

Para determinar $q(x)$ en cada sub-intervalo $[x_k, x_{k+1}]$ además de imponer las dos condiciones $q(x_k) = f(x_k)$ y $q(x_{k+1}) = f(x_{k+1})$ se necesita q coincida con f en algún otro punto, por ejemplo el punto medio $\frac{x_{k+1}+x_k}{2}$, de esta forma se puede también obtener una cota de error:

$$(4.5) \quad |f(x) - q(x)| \leq \frac{\sqrt{3}}{216} h^3 K_3, \quad \forall x \in [a, b],$$

donde K_3 es una cota para el valor absoluto de la tercera derivada en $[a, b]$. Ahora la **convergencia es cúbica**.

5. INTERPOLACIÓN CÚBICA A TROZOS: SPLINES CÚBICOS

En los splines cúbicos se construyen polinomios de tercer grado en cada sub-intervalo $[x_k, x_{k+1}]$ con continuidad hasta la segunda derivada en el todo el intervalo, es decir

$$(5.1) \quad \begin{cases} s_k(x_k) = f(x_k), & k = 0, \dots, N-1, \\ s_k(x_{k+1}) = f(x_{k+1}), & k = 0, \dots, N-1, \\ s'_k(x_{k+1}) = s'_{k+1}(x_{k+1}), & k = 0, \dots, N-2, \\ s''_k(x_{k+1}) = s''_{k+1}(x_{k+1}), & k = 0, \dots, N-2, \end{cases}$$

por lo que queda claro que no puede resolver en cada sub-intervalo independientemente de los otros, es decir, que el spline es global y no local.

De esta forma, el spline cúbico estará definido de la forma

$$v(x) = s_k(x), \quad x_k \leq x \leq x_{k+1} \quad k = 0, \dots, N-1,$$

donde

$$\begin{aligned} s_k(x) &= a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3, \\ s'_k(x) &= b_k + 2c_k(x - x_k) + 3d_k(x - x_k)^2, \\ s''_k(x) &= 2c_k + 6d_k(x - x_k), \end{aligned}$$

es decir, se tienen $4N$ parámetros.

De la primera condición en (5.1) es evidente que

$$\begin{aligned} a_k &= f(x_k) & k = 0, \dots, N-1, \\ a_k + b_k h_k + c_k h_k^2 + d_k h_k^3 &= f(x_{k+1}), & k = 0, \dots, N-1. \end{aligned}$$

con $h_k = x_{k+1} - x_k$ y reemplazando la primera en la segunda y dividiendo por h_k la N ecuaciones

$$(5.2) \quad b_k + c_k h_k + d_k h_k^2 = f[x_k, x_{k+1}], \quad k = 0, \dots, N-1.$$

Por otra parte, las otras dos condiciones en (5.1) dan las $2(N-1)$ ecuaciones

$$(5.3) \quad \begin{aligned} b_k + 2c_k h_k + 3d_k h_k^2 &= b_{k+1} & k = 0, \dots, N-2, \\ c_k + 3d_k h_k &= c_{k+1}, & k = 0, \dots, N-2, \end{aligned}$$

que se pueden extender para $k = N-1$ definiendo

$$b_N = v'(x_N), \quad c_N = \frac{1}{2} v''(x_N).$$

Despejando d_k en la segunda ecuación se tiene

$$(5.4) \quad d_k = \frac{c_{k+1} - c_k}{3 h_k}, \quad k = 0, \dots, N-1,$$

y entonces se despejan en (5.2) los

$$(5.5) \quad b_k = f[x_k, x_{k+1}] - \frac{h_k}{3}(2 c_k + c_{k+1}), \quad k = 0, \dots, N-1.$$

En definitiva, si se calculan los c_k se tiene todo el spline porque las dos ecuaciones anteriores calculan los b_k y d_k .

Para calcular los coeficientes c_k en la primera ecuación de (5.3) corremos los índices de la forma

$$(5.6) \quad b_{k-1} + 2 c_{k-1} h_{k-1} + 3 d_{k-1} h_{k-1}^2 = b_k \quad k = 1, \dots, N-1,$$

que utilizando (5.4) y (5.5) queda

$$\begin{aligned} & f[x_{k-1}, x_k] - \frac{h_{k-1}}{3}(2 c_{k-1} + c_k) + 2 c_{k-1} h_{k-1} + h_{k-1}(c_k - c_{k-1}) \\ &= f[x_k, x_{k+1}] - \frac{h_k}{3}(2 c_k + c_{k+1}), \end{aligned}$$

que finalmente da el sistema de ecuaciones lineal

$$(5.7) \quad c_{k-1} h_{k-1} + 2(h_k + h_{k-1})c_k + h_k c_{k+1} = 3(f[x_k, x_{k+1}] - f[x_{k-1}, x_k]),$$

para $k = 1, \dots, N-1$. En definitiva, se tiene $N-1$ ecuaciones para las incognitas c_0, \dots, c_N por lo que se precisan completar con alguna condición extra. La más habitual coloca son los **splines naturales** en los que

$$(5.8) \quad s''_0(x_0) = s''_{N-1}(x_N) = 0 \Rightarrow c_0 = c_N = 0.$$

Existen más tipo de tipo de splines por ejemplo

- **Periódicos** $s'_0(x_0) = s'_{N-1}(x_N), s''_0(x_0) = s''_{N-1}(x_N)$.
- **Pendiente fija en los extremos** $s'_0(x_0) = y'_0, s'_{N-1}(x_N) = y'_N$.
- **Not-a-Knot** $s'''_0(x_1) = s'''_1(x_1), s'''_{N-2}(x_{N-1}) = s'''_{N-1}(x_{N-1})$.

En cuanto al error se demuestra el siguiente teorema

Teorema 5.1. Si $f \in C^4[a, b]$ entonces para todo $x \in [a, b]$ el spline cúbico satisface las siguientes cotas

$$(5.9) \quad |f(x) - s(x)| \leq \frac{5}{384} h^4 \|f^{(iv)}\|_\infty,$$

$$(5.10) \quad |f'(x) - s'(x)| \leq \frac{1}{24} h^3 \|f^{(iv)}\|_\infty,$$

$$(5.11) \quad |f''(x) - s''(x)| \leq \frac{1}{3} h^2 \|f^{(iv)}\|_\infty.$$

Demostración. Ver [8]. □

Las referencias más completas para Splines son el clásico texto de C. de Boor [4] y más recientemente [17].

6. INTERPOLACIÓN CON MATLAB

El paquete MATLAB representa el polinomio

$$p(x) = p_1x^n + p_2x^{n-1} + \cdots + p_nx + p_{n+1},$$

en un vector $[p(1), \dots, p(n+1)]$ con sus coeficiente.

Las tres principales tareas con polinomio son tres:

Evaluación: El comando `polyval` utiliza el algoritmo del Horner: `y=polyval(p,z)` evalúa en z el polinomio de coeficientes p .

Estimación de sus raíces: El comando `roots` estima las raíces de un polinomio.

Adaptación a datos o interpolación: El program `polyinterp.m` de [13] realiza la interpolación de Lagrange, mientras que el comando `interp1` realiza interpolaciones a trozos. Mas concretamente `Yi=interp1(X,Y,Xi, método)` devuelve el polinomio de interpolación por X e Y evaluado en Xi y calculado con el método

- `linear`
- `spline`
- `nearest`
- `cubic`

En la figura 3 se ha interpolado la función $1/(x + (1 - x)^2)$ en diez punto equidistantes, en verde la interpolación de Lagrange, en negro la lineal a trozos y en magenta el spline cúbico.

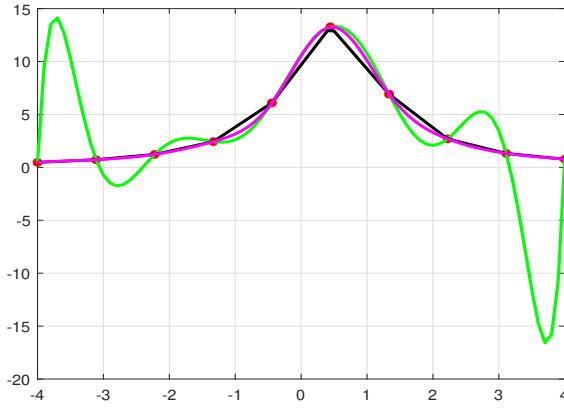


FIGURA 3. Interpolación de la función $1/(x + (1 - x)^2)$.

Otros polinomios:: El comando `polyfit(x,y,grado)` devuelve el polinomio mínimo cuadrático del grado que se indica.

7. ALGUNOS PROBLEMAS DE INTERPOLACIÓN NO ESTUDIADOS

7.1. Interpolación de Hermite. En muchos problemas no sólo se conoce valores de la función en los nodos, sino también valores de sus derivadas. En esta situación, el polinomio de interpolación debe coincidir con los valores de la función, sino también sus derivadas, son los que se conocen como problemas de interpolación de Hermite. Una sencilla introducción a dicho problema se puede leer en [1, pag. 319] y algo más completa en [2, pag. 137], [10, pag. 105], [14, pag. 314] y [3, pag. 52].

7.2. Interpolación trigonométrica y transformad discreta de Fourier. Como el nombre indica, se trata de representar funciones por polinomios trigonométrico que recorre todo el Análisis de Fourier [7, Cap. 6 y 7] y [1, Cap. 13]. En este tema, la bibliografía es amplísima. Desde el punto de la Matemática Aplicada y el Análisis Numérico una buena introducción es [14, ppag. 435] y [19, pag. 150] aunque la más original está en [13, Chap. 8].

Sin duda, las aplicaciones más relevantes de esta técnica estén en la resolución numérica de Ecuaciones en Derivadas Parciales en los métodos espectrales [18].

7.3. Interpolación de Chebyshev. Considerando la expresión (3.1) para el error de interpolación y su cota (3.2), en general como no se conoce el valor del punto ξ ni cotas para las derivadas de la función f la mejor opción para minimizar los errores será elegir los nodos que minimizan la función

$$\max_{a \leq s \leq b} \prod_{j=0}^N |s - x_j|,$$

lo que conduce a los **nodos de Chebysev** que en el intervalo $[-1, 1]$ son

$$(7.1) \quad x_i = \cos\left(\frac{2i+1}{2(N+1)}\pi\right), \quad i = 0, \dots, N.$$

Como se demuestra en [1, pag. 377], estos nodos de Chebyshev son las raíces de los polinomios del mismo nombre que resuelve el problema

$$(7.2) \quad \beta = \min_{x_0, \dots, x_N} \max_{-1 \leq x \leq 1} \prod_{j=0}^N |x - x_j|,$$

con $\beta = 2^{-N}$ por lo que ahora

$$\max_{-1 \leq x \leq 1} |f(x) - P_N(x)| \leq \frac{K_{N+1}}{2^N (N+1)!}.$$

Considerando entonces la fórmula de Stirling: $n! \approx \sqrt{2\pi n} (\frac{n}{e})^n$ se tiene que

$$\max_{-1 \leq x \leq 1} |f(x) - P_N(x)| \leq \frac{e}{N+1} \left[\frac{e}{2(N+1)} \right]^N \frac{1}{\sqrt{2\pi(N+1)}} K_{N+1},$$

que tiende a cero con velocidad exponencial, es decir, muy rápido.

En la figura 4 se puede apreciar la diferencia de errores al interpolar la función tipo Runge $\frac{1}{1+16x^2}$ en 17 nodos primero equidistantes y luego de Chebyshev.

En realidad, este resultados tan espectacular se obtiene de una teoría del potencial que se puede consultar en [18, Cap. 5] y [5, Cap. 3]. Los nodos de Chebyshev son las raíces de los **Polinomios de Chebyshev** que se aplican en diversos temas del Análisis Numérico. Una introducción sencilla se puede encontrar en [9, pag. 316], [1,

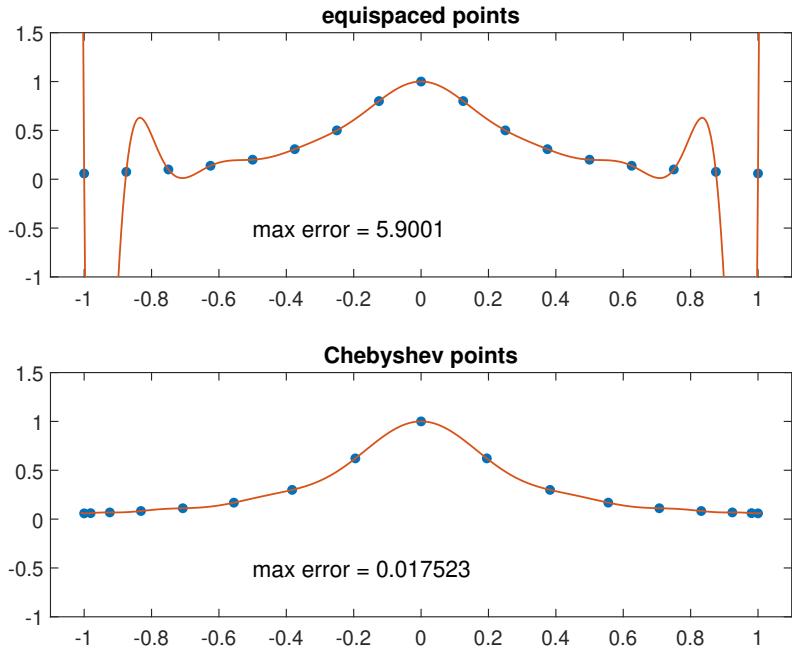


FIGURA 4. Interpolador de la función $\frac{1}{1+16x^2}$ en 17 nodos diferentes

Cap. 12], [19, pag. 124] y la más completas [6, Chap. 3] y el monográfico [12]. Como resultado de sus múltiples aplicaciones el paquete **Chebfun** que se puede consultar en <https://www.chebfun.org> y el más reciente **Chebops** para simular problemas diferenciales.

REFERENCIAS

1. U.M. Ascher and C. Greif, *A First Course in Numerical Methods*, SIAM, 2011.
2. K.E. Atkinson, *An Introduction to Numerical Analysis*, Wiley, 1978.
3. R. Bulirsch and J. Stoer, *Introduction to Numerical Analysis*, Springer, 1980.
4. C. de Boor, *A Practical Guide to Splines: Revised Edition*, Springer, 2001.
5. B. Fornberg, *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, 1998.
6. A. Gil, J. Segura, and N.M. Temme, *Numerical Methods for Special Functions*, SIAM, 2007.
7. B. Gustafson, *Fundamentals of Scientific Computing*, Springer, 2011.
8. C.A. Hall and W.W. Meyer, *Optimal Error Bounds for Cubic Spline Interpolation*, Journal Of Approximation Theory **16** (1976), no. 1, 105–122.
9. M.H. Holmes, *Introduction to Scientific Computing and Data Analysis*, Springer, 2016.
10. C.F. Van Loan, *Introduction to Scientific Computing*, Prentice-Hall, 2000.
11. J.E. Marsden and M.J. Hoffman, *Ánalisis clásico elemental. Segunda edición*, Addison-Wesley Iberoamericana, 1998.
12. J.C. Mason and D.C. Handscomb, *Chebyshev Polynomials*, Chapman and Hall/CRC, 2003.
13. C.B. Moler, *Numerical Computing with MATLAB*, SIAM, 2004.
14. R. Sacco, F. Saleri, and A. Quarteroni, *Numerical Mathematics*, Springer, 2000.
15. S.L. Salas, E. Hille, and G.J. Etgen, *Calculus. Una y varias variables. Cuarta edición*, Editorial Reverte, 2002.
16. J.M. Sanz-Serna, *Diez lecciones de Cálculo Numérico*, Universidad de Valladolid, 2010.
17. L.L. Schumaker, *Spline Functions. More Computational Methods*, SIAM, 2025.
18. L.N. Trefethen, *Spectral Methods in MATLAB*, SIAM, 2000.
19. W.Y. Yang, W. Cao, T.S. Chung, and J. Morris, *Applied Numerical Methods Using MATLAB*, Wiley Interscience, 2005.

DEPARTAMENTO DE MATEMÁTICAS DE LA UPV/EHU

Email address: fernando.vadillo@ehu.es