Aproximación de Funciones

Polinomio interpolante de Newton

A partir del sistema (22) se tiene para los 2 primeras ecuaciones

$$c_0 = f(x_0) \quad \text{y} \quad c_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

Introducimos lo que se conoce como notación de **diferencia dividida**.

La diferencia dividida cero para la función *f* con respecto a x_i es

$$f[x_i] = f(x_i)$$

La diferencia dividida uno para la función *f* con respecto a x_i y x_{i+1} es

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}. \tag{23}$$

La diferencia dividida dos para la función *f* con respecto a x_i , x_{i+1} y x_{i+2} es

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}. \tag{24}$$

Las diferencias divididas restantes se definen inductivamente.

Aproximación de Funciones

Polinomio interpolante de Newton

Cuando las *k-1* diferencias divididas

$$f[x_i,\ldots,x_{i+k-1}] \quad \text{y} \quad f[x_{i+1},\ldots,x_{i+k}]$$

han sido determinadas, la *k*-esima diferencia dividida de *f* respecto x_i, $x_{i+1},\ldots,x_{i+k}$, esta dada por

$$f[x_i,\ldots,x_{i+k}] = \frac{f[x_{i+1},\ldots,x_{i+k}] - f[x_i,\ldots,x_{i+k-1}]}{x_{i+k} - x_i}. \tag{25}$$

Usando esta notación, los otros coeficientes del polinomio interpolante (20), es decir c_2 hasta c_n, se pueden obtener consecutivamente de una manera similar a c_0 y c_1. Así,

$$c_0 = f[x_0], \quad c_1 = f[x_0,x_1], \quad \ldots, \quad c_k = f[x_0,\ldots,x_k]. \tag{26}$$

Ejercicio. Dado el polinomio interpolante (20), con c_0 y c_1 como en (26), usar $p(x_2)$ para demostrar que $c_2 = f[x_0,x_1,x_2]$.

Aproximación de Funciones

Polinomio interpolante de Newton

Con la notación (26), el polinomio interpolante dado por la ecuación (20) se puede escribir como

$$p(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots$$
$$+ f[x_0, \dots, x_n](x - x_0)(x - x_1)\cdots(x - x_{n-1})$$

$$p(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, \dots, x_k] \prod_{j=0}^{k-1} (x - x_j) \tag{27}$$

con x_0, x_1,…, x_n, puntos distintos.

La ecuación (27) se conoce como la fórmula de **diferencia dividida interpolante de Newton**. Esta nos da un procedimiento iterado para calcular los coeficientes del polinomio interpolante *p(x)*.

La determinación de las diferencias divididas para puntos de datos tabulados se bosqueja en la siguiente tabla.

Aproximación de Funciones

$$f[x_i,\ldots,x_{i+k}] = \frac{f[x_{i+1},\ldots,x_{i+k}] - f[x_i,\ldots,x_{i+k-1}]}{x_{i+k} - x_i}$$

Polinomio interpolante de Newton

	diferencia dividida					
	cero	uno	dos	tres	cuatro	cinco
x_0	$f[x_0]$					
		$f[x_0,x_1]$				
x_1	$f[x_1]$		$f[x_0,x_1,x_2]$			
		$f[x_1,x_2]$		$f[x_0,x_1,x_2,x_3]$		
x_2	$f[x_2]$		$f[x_1,x_2,x_3]$		$f[x_0,x_1,x_2,x_3,x_4]$	
		$f[x_2,x_3]$		$f[x_1,x_2,x_3,x_4]$		$f[x_0,x_1,x_2,x_3,x_4,x_5]$
x_3	$f[x_3]$		$f[x_2,x_3,x_4]$		$f[x_1,x_2,x_3,x_4,x_5]$	
		$f[x_3,x_4]$		$f[x_2,x_3,x_4,x_5]$		
x_4	$f[x_4]$		$f[x_3,x_4,x_5]$			
		$f[x_4,x_5]$				
x_5	$f[x_5]$					

Los elementos de la diagonal son los coeficientes del polinomio (27),

$$\boxed{p(x) = f[x_0] + \sum_{k=1}^{5} f[x_0,\ldots,x_k]\prod_{j=0}^{k-1}(x - x_j)\,.}$$

Este método permite agregar punto adicionales a bajo costo computacional.

Aproximación de Funciones

Polinomio interpolante de Newton

Algoritmo para calcular los coeficientes del polinomio

(El número de operaciones para este algoritmo es n^2 sumas y $n^2/2$ divisiones)

Leer x_i, $f_i = f(x_i)$, $i = 1, \dots, n$

Almacenar f_i en $c(i)$, $i = 1, \dots, n$

Para $j = 2$ hasta n

    Para $i = n$ *hasta* j

        $c(i) = (c(i) - c(i - 1)) / (x(i) - x(i - j + 1))$

    Fin para

Fin para

No es necesario almacenar todo el arreglo bidimensional, de la tabla anterior. Durante los cálculos el arreglo c se sobre-escribe, y al final contendrá los coeficientes del polinomio.

Ejercicio: Extender el algoritmo para almacenar todas la diferencias divididas.

Obs. La forma de diferencia dividida interpolante de Newton nos permite construir el polinomio de interpolación (20), y el algoritmo de evaluación anidada nos ayuda a evaluar este polinomio en puntos no tabulados.

Aproximación de Funciones

Polinomio interpolante de Newton

Ejemplo:

		diferencia dividida				
i	x_i	cero	uno	dos	tres	cuatro
1	1.0000	0.7651977				
2	1.3000	0.6200860	-0.4837			
3	1.6000	0.4554022	-0.5489	-0.1087		
4	1.9000	0.2818186	-0.5786	-0.0494	0.0659	
5	2.2000	0.1103623	-0.5715	0.0118	0.0681	0.0018

diferencia dividida $f[x_i,\ldots,x_{i+k}] = \dfrac{f[x_{i+1},\ldots,x_{i+k}] - f[x_i,\ldots,x_{i+k-1}]}{x_{i+k} - x_i}.$

$$p(x) = 0.7652 - 0.4837(x - x_1) - 0.1087(x - x_1)(x - x_2)$$
$$+ 0.0659(x - x_1)(x - x_2)(x - x_3) + 0.0018(x - x_1)(x - x_2)(x - x_3)(x - x_4)$$

Aproximación de Funciones

Polinomio interpolante de Newton

Teorema. Si $f \in C^n[a,b]$ y $x_0, x_1, \ldots, x_n \in [a,b]$ son distintos, entonces existe un número ξ en $[a,b]$ tal que

$$f[x_0,\ldots,x_n] = \frac{f^{(n)}(\xi)}{n!}.$$

Obs. Este resultado es una generalización de la aplicación del teorema del valor medio a

$$f[x_0,x_1] = \frac{f[x_1]-f[x_0]}{x_1 - x_0},$$

es decir, si f' existe, $f[x_0,x_1] = f'(\xi)$ para algún número ξ entre x_0 y x_1.

Teorema (término residual o cota de error). Sea p el polinomio interpolante de f en los puntos $x_0, x_1,\ldots, x_n$ (puntos diferentes) en $[a,b]$, dado por (27). Entonces para cada $x\in[a,b]$ existe un número $\xi(x)$ en (a,b) tal que

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)\cdots(x-x_n). \tag{28}$$

Aproximación de Funciones

Polinomio interpolante de Newton

Veamos la manera de cómo usar la acotación del error dado por (28) en un caso sencillo. Sea *l(x)* el polinomio lineal interpolante de *f*(*x*) en x_0 y x_1 y supongamos además que

$$|f''(x)| \leq M$$

en el intervalo de interés. Entonces, de (28)

$$|f(x) - l(x)| = \frac{|f''(\xi)|}{2}|(x - x_0)(x - x_1)| \leq \frac{M}{2}|(x - x_0)(x - x_1)|. \tag{29}$$

El problema del estudio de esta acotación depende en que *x* este fuera o dentro del intervalo $[x_0,x_1]$.

Si *x* esta fuera de $[x_0,x_1]$, se dice que estamos extrapolando para aproximar a *f*. Como $|(x-x_0)(x-x_1)|$ crece rápidamente a medida que *x* se aleja del intervalo $[x_0,x_1]$, extrapolación es un riesgo.

Si *x* esta dentro de $[x_0,x_1]$, se dice que estamos interpolando para aproximar *f*.

Aproximación de Funciones

Polinomio interpolante de Newton

En el caso de interpolación, podemos obtener cotas de error uniforme para (29). La función $|(x-x_0)(x-x_1)|$ alcanza su máximo en el punto $x = (x_0+x_1)/2$ siendo su máximo valor $(x_1-x_0)^2/4$. *De aquí*

$$x \in [x_0, x_1] \Rightarrow |f(x) - l(x)| \leq \frac{M}{2}|(x-x_0)(x-x_1)| \leq \frac{M}{8}(x_1-x_0)^2. \qquad (30)$$

Como una aplicación, supongamos que queremos calcular valores de la función sin(x) (rápido y fácil) almacenando valores de esta función en puntos equidistantes (distancia h) y usando interpolación lineal.

La pregunta que surge es: por ejemplo para un error de 10^{-4}

¿Cuán pequeño debe ser h para alcanzar una precisión dada?

Como la segunda derivada de sin(x) está acotada por 1, se tiene de (30)

$$|\sin(x) - l(x)| \leq \frac{h^2}{8}.$$

Tomando $\frac{h^2}{8} \leq 10^{-4}$ se tiene $h \leq 0.01\sqrt{8} = 0.0283\ldots$

Aproximación de Funciones

Polinomio interpolante de Newton

El método descrito para aproximar el sin(x) usa muchos puntos sobre intervalos pequeños (h es muy pequeño).

Otra posibilidad, es usar polinomios interpolantes de grado más alto para representar la función f.

Supongamos que para cada $n = 1, 2, 3, \ldots$, escogemos $n+1$ puntos equiespaciados y sea p_n el polinomio interpolante de f en esos puntos.

Si la sucesión de polinomios converge uniformemente a f, sabemos que existe un número n para el cual p_n está suficientemente de cerca de f para una precisión dada.

Veamos el siguiente ejemplo.

Aproximación de Funciones

Polinomio interpolante natural

Ejemplo: $(-2,-27),(0,-1),(1,0)$.

$$p_2(t) = x_1 + x_2 t + x_3 t^2$$

$$\begin{pmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ 1 & t_3 & t_3^2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

$$\begin{pmatrix} 1 & -2 & 4 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -27 \\ -1 \\ 0 \end{pmatrix}$$

Solución: $x = (-1, 5, -4)^t$. Entonces,

$$p_2(t) = -1 + 5t - 4t^2.$$

Aproximación de Funciones

Polinomio interpolante de Lagrange

Ejemplo: $(-2,-27),(0,-1),(1,0)$.

El polinomio $p_2(t)$ es

$$y_1\frac{(t-t_2)(t-t_3)}{(t_1-t_2)(t_1-t_3)}+y_2\frac{(t-t_1)(t-t_3)}{(t_2-t_1)(t_2-t_3)}+y_3\frac{(t-t_1)(t-t_2)}{(t_3-t_1)(t_3-t_2)}$$

$$p_2(t)=-27\frac{t(t-1)}{6}+\frac{(t+2)(t-1)}{2}.$$

Note: es el mismo polinomio hallado con la base de monomios, $p_2(t)=-1+5t-4t^2$.

Aproximación de Funciones

Polinomio interpolante de Newton

Ejemplo: $(-2,-27),(0,-1),(1,0)$.

t_i	$f[t_i]$	$f[t_i,t_{i+1}]$	$f[t_i,t_{i+1},t_{i+2}]$
-2	-27		
0	-1	$\frac{-1-(-27)}{0-(-2)}=13$	
1	0	$\frac{0-(-1)}{1-0}=1$	$\frac{1-13}{1-(-2)}=-4$

Solución:

$$p_2(t)=-27+13(t+2)-4(t+2)t$$
$$p_2(t)=-1+5t-4t^2.$$

Aproximación de Funciones

Polinomio interpolante de Newton

Consideremos la función $f(x) = \dfrac{1}{1+25x^2}$ para $x \in [-1,1]$.

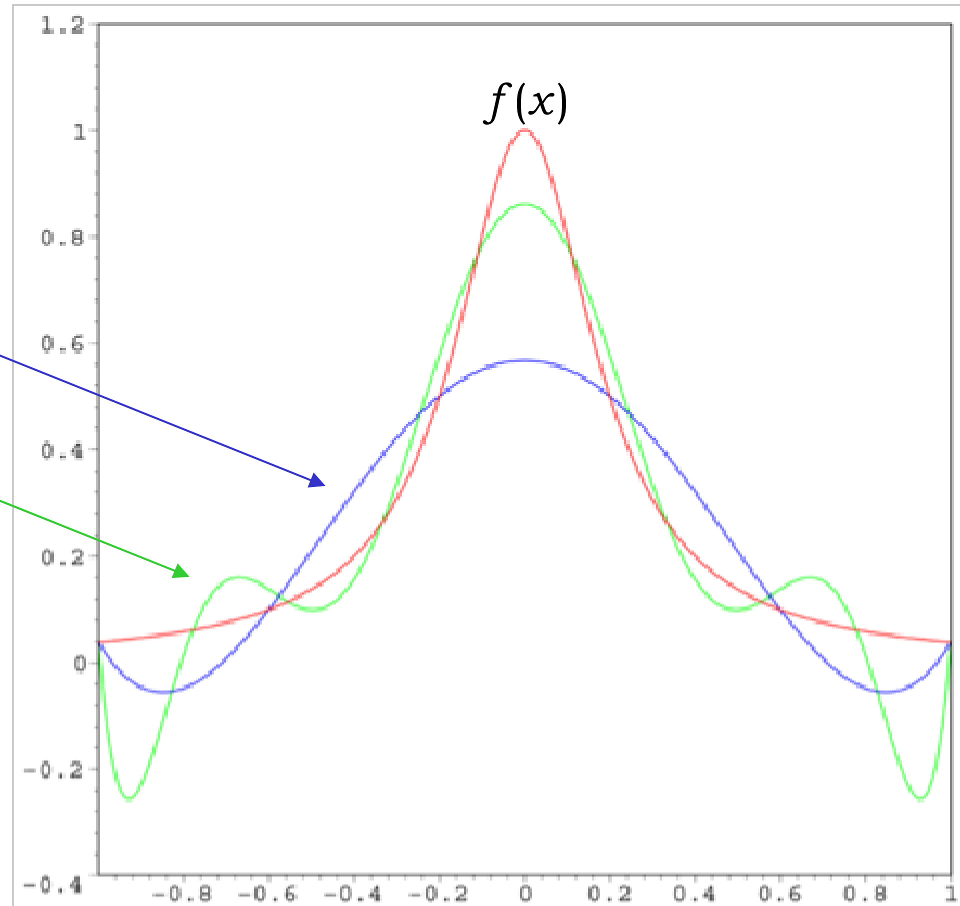
Seleccionamos los puntos

$$x_i = -1 + i\frac{2}{n}$$

para $i \in \{0,1,\ldots,n\}$

Polinomio interpolante de grado 5
(construido con 6 puntos)

Polinomio interpolante de grado 9
(construido con 10 puntos)



El error de interpolación tiende a infinito cuando el grado del polinomio crece:

$$\lim_{n\to\infty}\left(\max_{-1\le x\le 1}\left|f(x) - p_n(x)\right|\right) = \infty$$

Aproximación de Funciones

Polinomio interpolante de Newton

Para la función $f(x)=\dfrac{1}{1+25x^2}$ para $x\in[-1,1]$.

sus dos primeras derivadas son

$$f'(x)=-\frac{50x}{(1+25x^2)^2}\Rightarrow\left|f'(1)\right|=\frac{50}{26^2}=0.0740$$

$$f''(x)=-\frac{5000(1+25x^2)-50(1+25x^2)^2}{(1+25x^2)^4}\Rightarrow\left|f''(1)\right|=\frac{96200}{26^4}=0.2105$$

La magnitud de las derivadas de orden alto para esta función crecen más.

Vimos que $f(x)-p(x)=\dfrac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)\cdots(x-x_n)$,

por lo tanto, la cota del error de interpolación cuando se usan polinomios de grado alto crece mucho.

Aproximación de Funciones

Puntos de interpolación de Chebyshev

Para la función $f(x) = \dfrac{1}{1+25x^2} \quad \text{con} \quad x \in [-1,1],$

vimos que el error de interpolación tiende a infinito cuando el grado del polinomio crece. Este hecho se conoce como el fenómeno de Runge.

Los puntos de Chebyshev surgen del esfuerzo para ajustar los puntos de interpolación y tratar controlar el error de interpolación.

Dado *m,* los *m* puntos de Chebyshev en el intervalo [-*1,1*], es decir x_0, x_1, x_{m-1} se definen como

$$x_{i-1} = \cos\left(\frac{2i-1}{2m}\pi\right), \quad 1 \le i \le m. \tag{31}$$

Para definir *m* puntos de Chebyshev en un intervalo dado [*a,b*] se procede así

$$x_{i-1} = \frac{1}{2}(a+b) + \frac{1}{2}(b-a)\cos\left(\frac{2i-1}{2m}\pi\right), \quad 1 \le i \le m. \tag{32}$$

Aproximación de Funciones

Puntos de interpolación de Chebyshev

Caso 5 puntos en [-1, 1]

puntos igualmente espaciados

		diferencia dividida				
i	x_i	cero	uno	dos	tres	cuatro
0	-1.0000	0.0385				
1	-0.5000	0.1379	0.1989			
2	+0.0000	1.0000	1.7241	1.5252		
3	+0.5000	0.1379	-1.7241	-3.4483	-3.3156	
4	+1.0000	0.0385	-0.1989	1.5252	3.3156	3.3156

polinomio interpolante de Newton

$$p(x) = 0.0385 + 0.1989(x - x_0) + 1.5252(x - x_0)(x - x_1)$$
$$- 3.3156(x - x_0)(x - x_1)(x - x_2) + 3.3156(x - x_0)(x - x_1)(x - x_2)(x - x_3)$$

$$f[x_i, \ldots, x_{i+k}] = \frac{f[x_{i+1}, \ldots, x_{i+k}] - f[x_i, \ldots, x_{i+k-1}]}{x_{i+k} - x_i}.$$

Aproximación de Funciones

Puntos de interpolación de Chebyshev

Caso 5 puntos en [-1, 1]

puntos de Chebyshev

		diferencia dividida				
i	x_i	cero	uno	dos	tres	cuatro
0	-0.9511	0.0424				
1	-0.5878	0.1038	0.1691			
2	0.0000	1.0000	1.5248	1.4255		
3	+0.5878	0.1038	-1.5248	-2.5941	-2.6121	
4	+0.9511	0.0424	-0.1691	1.4255	2.6121	2.7465

polinomio interpolante de Newton

$$p(x) = 0.0424 + 0.1691(x - x_0) + 1.4255(x - x_0)(x - x_1)$$
$$-2.6121(x - x_0)(x - x_1)(x - x_2) + 2.7465(x - x_0)(x - x_1)(x - x_2)(x - x_3)$$

$$f[x_i, \ldots, x_{i+k}] = \frac{f[x_{i+1}, \ldots, x_{i+k}] - f[x_i, \ldots, x_{i+k-1}]}{x_{i+k} - x_i}.$$

Aproximación de Funciones

Caso 5 puntos en [-1, 1]

puntos igualmente espaciados				
-1.0000	-0.5000	0.0000	0.5000	1.0000
0.0385	0.1379	1.0000	0.1379	0.0385

puntos de Chebyshev				
-0.9511	-0.5878	0.0000	0.5878	0.9511
0.0424	0.1038	1.0000	0.1038	0.0424

Polinomio interpolante de Newton

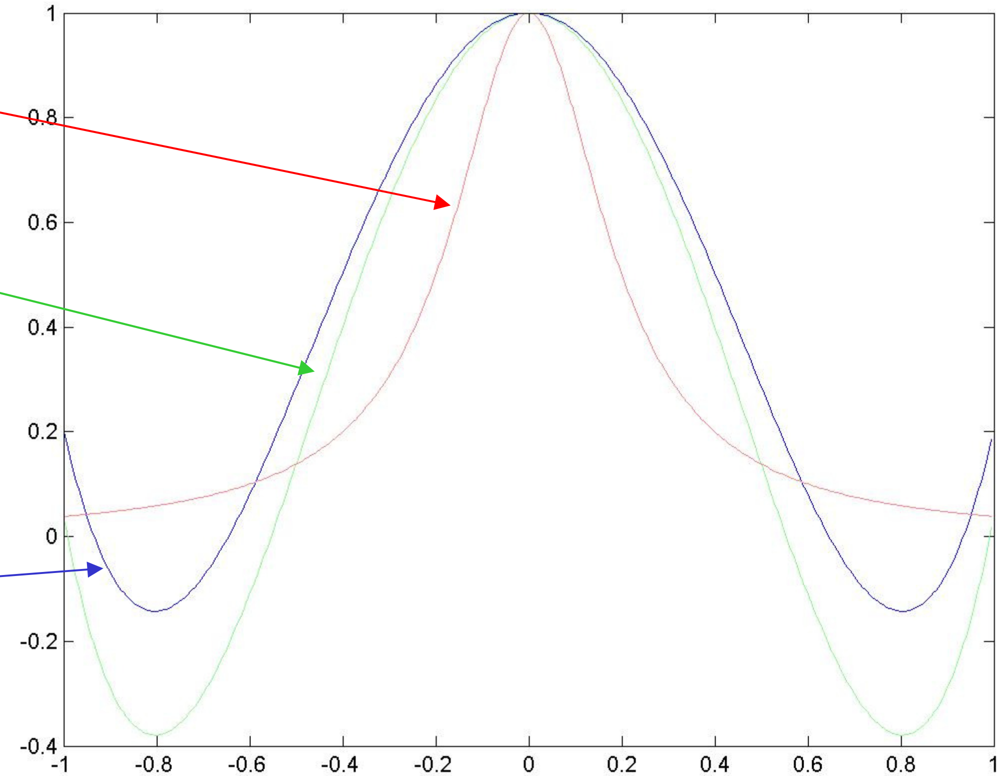
$$f(x) = \frac{1}{1+25x^2} \quad \text{para} \quad x \in [-1,1].$$

Coeficientes polinomio interpolante de Newton de grado 4 para puntos igualmente espaciados

0.0385 0.1989 1.5252 -3.3156 3.3156

Coeficientes polinomio interpolante de Newton de grado 4 para puntos de Chebyshev

0.0424 0.1691 1.4255 -2.6121 2.7465



El error de interpolación es menor

divdif_ap.m

Aproximación de Funciones

Puntos de interpolación de Chebyshev

El error de interpolación para el polinomio interpolante de Newton en *n+1* puntos es

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)\cdots(x - x_n).$$

Si suponemos que $\left|f^{(n+1)}(x)\right| \leq M \quad \text{si} \quad -1 \leq x \leq 1,$ entonces

$$\left|f(x) - p(x)\right| \leq \frac{M}{(n+1)!} \max_{x\in[-1,1]} \left|\prod_{i=0}^{n}(x - x_i)\right|.$$

Se puede demostrar que variando los puntos $x_i \quad 0 \leq i \leq n$

$$\min\left(\max_{x\in[-1,1]}\left|\prod_{i=0}^{n}(x - x_i)\right|\right) = 2^{-n}$$

y este mínimo se alcanza en los puntos de Chebyshev definidos por (31).

Aproximación de Funciones

Interpolación polinómica a trozos

Hasta el momento se ha realizado aproximación de funciones arbitrarias en intervalos cerrados usando polinomios. Este método es apropiado en muchas circunstancias, pero

- la naturaleza oscilatoria de los polinomios de grado alto y
- la propiedad de que una fluctuación sobre una porción pequeña del intervalo puede introducir fluctuaciones muy grandes sobre el rango entero,

restringe su uso cuando se aproximan muchas de las funciones que surgen en situaciones físicas.

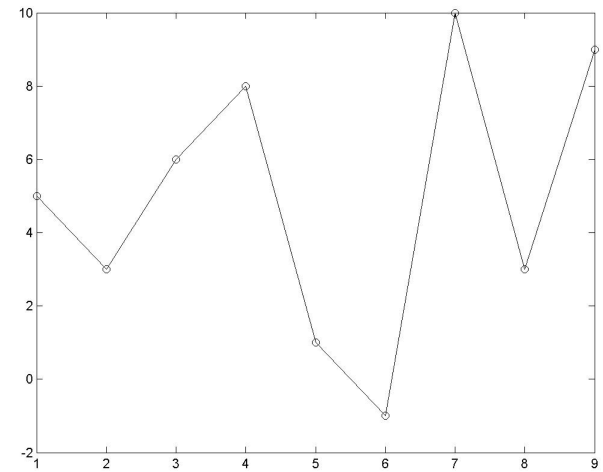
Un enfoque alternativo es dividir el intervalo en una colección de subintervalos y construir un polinomio interpolación diferente en cada subintervalo. La aproximación con funciones de este tipo se denomina interpolación polinómica a trozos (“splines”).

Aproximación de Funciones

Interpolación polinómica a trozos

El tipo más simple de interpolación polinómica a trozos es la lineal a trozos que consiste en unir un conjunto de *n+1* puntos

$$\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$$



con una serie de líneas rectas. Este es un método usado, por ejemplo, para funciones trigonométricas, cuando se quieren los valores intermedios de una colección de puntos tabulados.

La desventaja de enfocar un problema de interpolación usando funciones lineales es que en cada uno de los extremos de los subintervalos, no hay ninguna seguridad de diferenciabilidad, lo cual, geométricamente significa que la función interpolante no es "suave" en esos puntos.

Aproximación de Funciones

Interpolación polinómica a trozos

Definición. Dada una función f definida en $[a, b]$ y un conjunto de números, los cuales denominaremos los nodos, $a = x_0 < x_1 < \ldots < x_n = b$, un **spline cúbico** S para f es una función que satisface las condiciones

a) S es un polinomio de grado ≤ 3, denotado por S_j en el intervalo $[x_j, x_{j+1}]$ para cada $j = 0, 1, \ldots, n-1$

b) $S(x_j) = f(x_j)$ para cada $j = 0, 1, \ldots, n$

c) $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ para cada j = 0, 1, …, n-2

d) $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ para cada j = 0, 1, …, n-2

e) se satisface una del siguiente conjunto de condiciones de frontera

   i. $S''(x_0) = S''(x_n) = 0$ (frontera libre)

   ii. $S'(x_0) = f'(x_0)$ y $S'(x_n) = f'(x_n)$ (frontera amarrada)

En el caso (i) S se denomina spline cúbico natural.

Aproximación de Funciones

Interpolación polinómica a trozos

Para construir el spline cúbico para una función f, aplicamos las condiciones de la definición al polinomio cúbico

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3 \text{ para cada } j = 0, \ldots, n-1.$$

Claramente $\boxed{S_j(x_j) = a_j = f(x_j),}$

y aplicando la condición (b), se tiene que para *j* = *0*, …, *n-2*

$$a_{j+1} = S_{j+1}(x_{j+1}) = S_j(x_{j+1}) = a_j + b_j(x_{j+1} - x_j) + c_j(x_{j+1} - x_j)^2 + d_j(x_{j+1} - x_j)^3.$$

Introducimos la notación $h_j = (x_{j+1} - x_j)$ para cada *j* = *0*, …, *n-1*,

y si definimos $a_n = f(x_n)$ se tiene que

$$\boxed{a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \text{ para cada } j = 0, \ldots, n-1.} \qquad (33)$$

Aproximación de Funciones

Interpolación polinómica a trozos

Si definimos $b_n = S'(x_n)$ y observamos que

$$S'_j(x) = b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2$$

se tiene que $S'_j(x_j) = b_j$ para cada para cada *j* = *0*, …, *n-1.*

Aplicando la condición (c)

$$\boxed{b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2 \ \text{para cada} \ j = 0, \ldots, n-1.} \tag{34}$$

Otra relación entre los coeficientes de S_j se obtiene definiendo

$$c_n = S''_{n-1}(x_n)/2$$

y aplicando la condición (d), de donde

$$c_{j+1} = c_j + 3d_j h_j \ \text{para cada} \ j = 0, \ldots, n-1. \tag{35}$$

Aproximación de Funciones

Interpolación polinómica a trozos

Despejando d_j de (35)

$$\boxed{d_j = \frac{c_{j+1} - c_j}{3h_j}} \tag{36}$$

y sustituyendo en (33) y (34) se tiene que para cada $j = 0, \ldots, n-1$

$$a_{j+1} = a_j + b_j h_j + \frac{h_j^2}{3}(2c_j + c_{j+1}) \tag{37}$$

$$b_{j+1} = b_j + h_j(c_j + c_{j+1}) \tag{38}$$

Despejando b_j en la ecuación (37)

$$\boxed{b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1})} \tag{39}$$

y luego b_{j-1} en la misma ecuación (con una reducción del índice)

$$b_{j-1} = \frac{1}{h_{j-1}}(a_j - a_{j-1}) - \frac{h_{j-1}}{3}(2c_{j-1} + c_j). \tag{40}$$

Aproximación de Funciones

Interpolación polinómica a trozos

Reduciendo el índice en 1 para (38) se tiene $b_j = b_{j-1} + h_{j-1}(c_{j-1} + c_j)$.

Sustituyendo b_j y b_{j-1} dados por (39) y (40), en la ecuación anterior

$$\frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1}) =$$
$$\frac{1}{h_{j-1}}(a_j - a_{j-1}) - \frac{h_{j-1}}{3}(2c_{j-1} + c_j) + h_{j-1}(c_{j-1} + c_j)$$

Realizando algunos simplificaciones se obtiene el sistema lineal

$$h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1})$$

para cada $j = 1, \ldots, n-1,$

que escrito en forma matricial queda como

Aproximación de Funciones

Interpolación polinómica a trozos

$$
\begin{pmatrix}
h_0 & 2(h_0+h_1) & h_1 & & & \\
 & h_1 & 2(h_1+h_2) & h_2 & & \\
 & & & \ddots & & \\
 & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1}
\end{pmatrix}
\begin{pmatrix} c_0 \\ \vdots \\ \vdots \\ c_n \end{pmatrix}
=
\begin{pmatrix}
\frac{3}{h_1}(a_2-a_1)-\frac{3}{h_0}(a_1-a_0) \\
\frac{3}{h_2}(a_3-a_2)-\frac{3}{h_1}(a_2-a_1) \\
\vdots \\
\frac{3}{h_{n-1}}(a_n-a_{n-1})-\frac{3}{h_{n-2}}(a_{n-1}-a_{n-2})
\end{pmatrix}. \quad (41)
$$

El sistema (41) de *n-1* ecuaciones tiene como incógnitas los c_j, $j = 0, \ldots, n$, ya que los valores de h_j y a_j están dados por el espaciamiento entre los nodos y el valor de *f* en los nodos.

Aproximación de Funciones

Interpolación polinómica a trozos

Una vez determinados los c_j, los b_j se calculan usando (39) y los d_j de (36)

$$b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1}) \qquad d_j = \frac{c_{j+1} - c_j}{3h_j}$$

para cada $j = 0, \ldots, n-1$.

Finalmente se tiene los coeficientes del spline cúbico S_j en cada subintervalo $[x_j, x_{j+1}]$

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

para cada $j = 0, \ldots, n-1$.

La pregunta que surge es ¿cuándo el sistema lineal (41) tiene solución?

La respuesta está asociada con las condiciones de frontera dada en la parte (e) de la definición de spline cúbico.

Esto se resume en los 2 teoremas siguientes.

Aproximación de Funciones

Interpolación polinómica a trozos

Teorema: Sea *f* una función definida en [*a*, *b*], entonces *f* tiene un único spline cúbico *S* con condición de frontera libre, o sea un único spline cúbico que satisface las condiciones $S''(a) = S''(b) = 0$.

Siguiendo la notación usual $a = x_0 < x_1 < \ldots < x_n = b$, las condiciones de frontera en este caso implican que

$$c_n = S''_{n-1}(x_n)/2 = 0$$

$$0 = S''_0(x_0) = 2c_0 + 6d_j(x_0 - x_0) \Rightarrow c_0 = 0$$

Estas 2 condiciones, junto con el sistema lineal (41) producen el sistema $Ax = b$, donde A es una matriz $(n+1)\times(n+1)$ diagonal dominante estricta, y x y b como en (41).

$$A = \begin{pmatrix} 1 & 0 & & & & \\ h_0 & 2(h_0+h_1) & h_1 & & & \\ & h_1 & 2(h_1+h_2) & h_2 & & \\ & & & \ddots & & \\ & & h_{n-2} & & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ & & & & 0 & 1 \end{pmatrix}$$

A es no singular

Aproximación de Funciones

Interpolación polinómica a trozos

Teorema: Sea *f* una función definida en [*a*, *b*], entonces *f* tiene un único spline cúbico *S* con condición de frontera amarrada, o sea un único spline cúbico que satisface las condiciones *S'*(*a*) = *f'*(*a*) y *S'*(*b*) = *f'*(*b*).

Usando $f'(a) = S_0'(a) = S_0'(x_0) = b_0,$ la ecuación (39) con *j=0* se transforma

$$f'(a) = b_0 = \frac{1}{h_0}(a_1 - a_0) - \frac{h_0}{3}(2c_0 + c_1)$$

$$\Rightarrow 2h_0c_0 + h_0c_1 = \frac{3}{h_0}(a_1 - a_0) - 3f'(a) \tag{42}$$

Usando $f'(b) = b_n = S'(x_n)$ y la ecuación (38), se tiene

$$f'(b) = b_n = b_{n-1} + h_{n-1}(c_{n-1} + c_n),$$

La ecuación (39) con *j=n-1* se transforma

$$b_{n-1} = \frac{1}{h_{n-1}}(a_n - a_{n-1}) - \frac{h_{n-1}}{3}(2c_{n-1} + c_n)$$

Aproximación de Funciones

Interpolación polinómica a trozos

Combinando estas 2 últimas ecuaciones, se obtiene

$$f'(b) = \frac{1}{h_{n-1}}(a_n - a_{n-1}) - \frac{h_{n-1}}{3}(2c_{n-1} + c_n) + h_{n-1}(c_{n-1} + c_n)$$

$$= \frac{1}{h_{n-1}}(a_n - a_{n-1}) - \frac{h_{n-1}}{3}(c_{n-1} + 2c_n)$$

$$\Rightarrow h_{n-1}c_{n-1} + 2h_{n-1}c_n = 3f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1}) \qquad (43)$$

El sistema lineal (41) en conjunto con (42) y (43) conduce a un sistema lineal *Ax* = *b* donde A es una matriz $(n+1)\times(n+1)$ diagonal dominante estricta, y *x* como en (41) y *b* similar.

A es no singular

Aproximación de Funciones

Interpolación polinómica a trozos

$$2h_0c_0 + h_0c_1 = \frac{3}{h_0}(a_1 - a_0) - 3f'(a) \qquad h_{n-1}c_{n-1} + 2h_{n-1}c_n = 3f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1})$$

La matriz y término de la derecha del sistema son

$$A = \begin{pmatrix} 2h_0 & h_0 & & & & \\ h_0 & 2(h_0+h_1) & h_1 & & & \\ & h_1 & 2(h_1+h_2) & h_2 & & \\ & & & \ddots & & \\ & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ & & & h_{n-1} & 2h_{n-1} \end{pmatrix}$$

$$b = \begin{pmatrix} \frac{3}{h_0}(a_1 - a_0) - 3f'(a) \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \frac{3}{h_2}(a_3 - a_2) - \frac{3}{h_1}(a_2 - a_1) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 3f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1}) \end{pmatrix}.$$