Relación III Métodos Númericos I

Javier Gómez López

2020/2021

Ejercicio 1. Sean $x_0, x_1, \ldots, x_M$ $M+1$ números reales. Comprueba que

$$\det \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ x_0 & x_1 & \cdots & x_{M-1} & x_M \\ x_0^2 & x_1^2 & \cdots & x_{M-1}^2 & x_M \\ \vdots & \vdots & & \vdots & \vdots \\ x_0^M & x_1^M & \cdots & x_{M-1}^M & x_M^M \end{bmatrix} = \prod_{\substack{i,j=0 \\ i<j}}^{M} (x_j - x_i).$$

Deduce que si $x_0, y_0), (x_1, y_1), \ldots, (x_M, y_M) \in \mathbb{R}^2$ son tales que

$$i, j = 1, \ldots, M \Rightarrow x_i \neq x_j,$$

entonces existe una única función polinómica $p : \mathbb{R} \longrightarrow \mathbb{R}$ de grado menor o igual que M con

$$i = 0, 1, \ldots, M \Rightarrow p(x_i) = y_i.$$

Comencemos con la primera parte del ejercicio. Procederemos por inducción. Llamemos Z a la condición que nos pide demostrar el ejercicio y sea $A = \{n \in \mathbb{N} : n \text{ cumple } Z\}$ y probemos que es inductivo:

- Comprobamos el caso $n = 1$:

$$\det \begin{bmatrix} 1 & 1 \\ x_0 & x_1 \end{bmatrix} = x_1 - x_0$$

  Observamos que se cumple para este caso.

- Supongamos cierto para $n-1$ y veamos que es cierto para n:

$$\det \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_0 & x_1 & \cdots & x_n \\ x_0^2 & x_1^2 & \cdots & x_n^2 \\ \vdots & \vdots & & \vdots \\ x_0^n & x_1^n & \cdots & x_n^n \end{bmatrix} \overset{\substack{F_{n+1}=F_{n+1}-x_0F_n \\ \cdots \\ F_3=F_3-x_0F_2 \\ F_2=F_2-x_0F_1}}{=} \det \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & x_1 - x_0 & \cdots & x_n - x_0 \\ 0 & x_1^2 - x_0x_1 & \cdots & x_n^2 - x_0x_n \\ \vdots & \vdots & & \vdots \\ 0 & x_1^n - x_0x_1^{n+1} & \cdots & x_n^n - x_0x_n^{n-1} \end{bmatrix} =$$

$$= \det \begin{bmatrix} x_1 - x_0 & \cdots & x_n - x_0 \\ x_1^2 - x_0x_1 & \cdots & x_n^2 - x_0x_n^2 \\ \vdots & & \vdots \\ x_1^n - x_0x_1^{n-1} & \cdots & x_n^n - x_0x_n^{n-1} \end{bmatrix} \overset{\text{saco factor común por columnas}}{=}$$

$$(x_1-x_0)\cdots(x_n-x_0)\cdot\det\begin{bmatrix}1 & \cdots & 1\\ x_1 & \cdots & x_n\\ \vdots & & \vdots\\ x_1^{n-1} & \cdots & x_n^{n-1}\end{bmatrix} \overset{\text{hipótesis de inducción}}{=} (x_1-x_0)\cdots(x_n-x_0)\cdot\prod_{\substack{i,j=1\\ i<j}}^{M}(x_j-x_i) =$$

$$= \prod_{\substack{i,j=0\\ i<j}}^{M}(x_j - x_i)$$

y queda probado que el conjunto A es inductivo y por tanto Z se cumple $\forall n \in \mathbb{N}$.

Pasemos ahora a la segunda parte del ejercicio. Lo que se nos pregunta es si :

$$\exists! p \in \mathbb{P}_n : i = 0, 1, \ldots, n \Rightarrow p(x_i) = y_i$$

Esto ocurre si y solo si

$$\exists! a_0, a_1, \ldots, a_N : \begin{cases} a_0 + a_1x_0 + \cdots + a_nx_0^n = y_0\\ a_0 + a_1x_1 + \cdots + a_nx_1^n = y_1\\ \vdots\\ a_0 + a_1x_n + \cdots + a_nx_n^n = y_n\end{cases}$$

Estamos ante un sistema cuadrado cuya solución es única si es un sistema compatible determinado. Esto solo ocurre si

$$\det\begin{bmatrix}1 & x_0 & \cdots & x_0^N\\ 1 & x_1 & \cdots & x_1^N\\ \vdots & \vdots & & \vdots\\ 1 & x_n & \cdots & x_n^n\end{bmatrix} \neq 0$$

Recordemos la propiedad de los determinantes que dice que $\det(A) = \det(A^T)$. Por tanto estamos ante el caso probado anteriormente y podemos asegurar que

$$\det\begin{bmatrix}1 & x_0 & \cdots & x_0^N\\ 1 & x_1 & \cdots & x_1^N\\ \vdots & \vdots & & \vdots\\ 1 & x_n & \cdots & x_n^n\end{bmatrix} = \prod_{\substack{i,j=0\\ i<j}}^{M}(x_j - x_i) \neq 0$$

Puesto que hemos supuesto (en el enunciado lo especifíca) que $x_j \neq x_i$ para $j \neq i$, se verifica lo anterior y queda probado el ejercicio.

Ejercicio 2. Comprueba que el problema de interpolación anterior no está bien definido si el grado de la función polinómica p es distinto de M (número de datos menos 1).

Podemos afirmar que si $\text{gr}(p) \neq M \Rightarrow \text{gr}(p) < M$ o $\text{gr}(p) > M$. Distingamos casos:

- Supongamos que $\text{gr}(p) < M$. Entonces, tomando $N = M - K$ con $K \in \{1, \ldots, M\}$ el sistema que nos genera si imponemos que $p(x) = a_0 + a_1x + \cdots + a_nx^N$ sea tal que $p(x_i) = y_i$ con $i \in \{1, \ldots, N\}$ es el siguiente:

$$\begin{cases} a_0 + a_1x_0 + \cdots + a_nx_0^n = y_0\\ a_0 + a_1x_1 + \cdots + a_nx_1^n = y_1\\ \vdots\\ a_0 + a_1x_n + \cdots + a_nx_n^n = y_n\end{cases}$$

Observamos que se trata de un sistema de ecuaciones compatible determinado, por lo que el polinomio de la solución es único como se ha demostrado anteriormente. Pero dicho polinomio no cumpliria que

$$p(x_j) = y_j \quad j = n+1, \ldots, n$$

ya que esta condición no se impone en el sistema para obtener los coeficientes y por tanto no estaría bien planteado en dicho caso.

- Supongamos lo contrario, es decir, que $\text{gr}(p) > M$. Se entonces $N = M + K$ con $K \in \mathbb{N}$. Repetimos el proceso anterior tomando un polinomio genérico de grado N que verifique que $p(x_i) = y_i$ para $i = 1, \ldots, N$. Tendriamos el siguiente sistema:

$$\begin{cases} a_0 + a_1 x_0 + \cdots + a_n x_0^n = y_0 \\ a_0 + a_1 x_1 + \cdots + a_n x_1^n = y_1 \\ \vdots \\ a_0 + a_1 x_n + \cdots + a_n x_n^n = y_n \end{cases}$$

Sabemos que el determinante de la matriz de coeficientes es distinto de 0 (lo hemos probado en el ejercicio 1) y que el determinante de la matriz ampliada con los términos independientes también es máximo, por tanto sus rangos coinciden. Sea A la matriz de coeficientes y A' la matriz ampliada, entonces tenemos

$$\text{rg}(A) = \text{rg}(A') = M < N = \text{n}^{\text{o}} \text{ de incógnitas}$$

Por el teorema de Rouché-Frobenius tenemos un sistema de ecuaciones compatible indeterminado. Por tanto, se verifica que $p(x_i) = y_i$, $i = 1, \cdots, n$ pero no se cumple que dicho poliniomio sea único.

Ejercicio 3. Demuestra que si $a < b$, $f \in C^3([a,b])$ y $\mathbf{I}_2 f$ es el polinomio en $\mathbb{P}_2$ de forma

$$\mathbf{I}_2 f(x_0) = f(x_0),\ \mathbf{I}_2 f(x_1) = f(x_1),\ \mathbf{I}_2 f(x_2) = f(x_2),$$

con los nodos igualmente espaciados $x_0 = a$, $x_1 = (a+b)/2$ y $x_2 = b$, entonces el correspondiente error de la interpolación $\mathbf{E}_2 f$ verifica

$$||\mathbf{E}_2 f||_\infty \leq \frac{||f'''||_\infty}{9\sqrt{3}} h^3$$

siendo $h = (b-a)/2$.

Comencemos recordando la fórmula de el error de interpolación vista en teoría:

$$E_N f(x) = \frac{f^{N+1}(\varepsilon)}{(N+1)!} \omega_{N+1}(x), \quad \varepsilon \in]a,b[\tag{1}$$

De aquí obtenemos las expresiones de el control puntual y de el control uniforme, respectivamente:

$$E_2 f(x) = \frac{f'''(\varepsilon)}{3!} \omega_3(x) \qquad ||E_2 f(x)||_\infty \leq \frac{||f'''||_\infty}{6} ||\omega_3||_\infty$$

Por lo tanto, tratemos de acotar $\omega_3(x)$. Primero de todo, sea $x \in [a,b]$, entonces podemos afirmar que $\exists t \in [0,2] : x = a + th$. Ahora calculemos $\omega_3(x)$:

$$|\omega_3(x)| = |\omega_3(a+th)| = |(x-x_0)(x-x_1)(x-x_2)| = |(a+th-a)\left(a+th-\frac{a+b}{2}\right)(a+th-b)| =$$

$$= th \cdot \left| \frac{a-b}{2} + th \right| \cdot |a - b + th| = th \cdot | - h + th| \cdot | - 2h + th| = th^3 \cdot |t-1| \cdot |t-2| =$$

$$= t \cdot |t-1| \cdot (2-t)h^3$$

Por tanto, $\omega_3(x) = t \cdot |t-1| \cdot (2-t)h^3$. Ahora, sea $f : [0,2] \longleftrightarrow \mathbb{R}$, $f(t) = t \cdot |t-1| \cdot (2-t)$ y calculemos el máximo de esta función:

$$f(t) = \begin{cases} t \cdot (1-t) \cdot (2-t) & \text{si} \quad 0 \leq x \leq 1 \\ t \cdot (t-1) \cdot (2-t) & \text{si} \quad 1 < x \leq 2 \end{cases} \Rightarrow f(t) = \begin{cases} t^3 - 3t^2 + 2t & \text{si} \quad 0 \leq x \leq 1 \\ -t^3 + 3t^2 - 2t \cdot (2-t) & \text{si} \quad 1 < x \leq 2 \end{cases}$$

Calculemos ahora su derivada y observemos en que punto se anula. Podemos tomar cualquiera de las dos expresiones pues lo único que variaría sería el signo y esto no afecta en las raices cuadradas.

$$f'(t) = 3t^2 - 6t + 2 = 0 \Longleftrightarrow t = \frac{3 \pm \sqrt{3}}{3}$$

$$f''(t) = 6t - 6 \Rightarrow \begin{cases} f''\left(\frac{3+\sqrt{3}}{3}\right) = 3.46 > 0 \Rightarrow \text{ mínimo} \\ f''\left(\frac{3-\sqrt{3}}{3}\right) = -3.46 < 0 \Rightarrow \text{ máximo} \end{cases}$$

Por tanto,

$$f\left(\frac{3-\sqrt{3}}{3}\right) = \frac{2\sqrt{3}}{9}$$

es máximo de la función y podemos afirmar que $||\omega_3||_\infty \leq \frac{2\sqrt{3}}{9}h^3$. Teniendo en cuenta la expresión del control uniforme, deducimos que

$$||E_2 f(x)||_\infty \leq \frac{||f'''||_\infty}{6}||\omega_3||_\infty \leq \frac{||f'''||_\infty}{6} \cdot \frac{2\sqrt{3}}{9}h^3 = \frac{||f'''||_\infty}{9\sqrt{3}}h^3$$

y queda probado lo pedido.

Ejercicio 4. Calcula los 7 nodos de Chebyshev $x_0, x_1, x_2, x_3, x_4, x_5, x_6$ del intervalo [1.6,3] y úsalos para resolver el sistema de interpolación

$$\text{encontrar} p \in \mathbb{P}_6 : i = 0,1,2,3,4,5,6 \Rightarrow p(x_i) = \sqrt{|x_i - 2|}$$

mediante las fórmulas de Lagrange y Newton. Analiza el condicionamiento de este problema y obtén una estimación del error de interpolación.

Primero de todo, calculemos los nodos de Chebyshev en el intervalo [-1,1]:

(% i3) N:7;

$$7 \tag{N}$$

(% i4) v1: makelist(cos(%pi *(2*i+1)/(2*N)),i,0, N-1);

$$[\cos\left(\frac{\pi}{14}\right), \cos\left(\frac{3\pi}{14}\right), \cos\left(\frac{5\pi}{14}\right), 0, \cos\left(\frac{9\pi}{14}\right), \cos\left(\frac{11\pi}{14}\right), \cos\left(\frac{13\pi}{14}\right)] \tag{v1}$$

Ahora buscaremos un isomorfismo afín $g : [-1,1] \longrightarrow [1.6,3]$ tal que $g(-1) = 1.63$ y $f(1) = 3$):

$$g(x) = \alpha x + \beta \Rightarrow \begin{cases} \alpha \cdot (-1) + \beta = 1.63 \\ \alpha \cdot 1 + \beta = 3 \end{cases} \Rightarrow \begin{cases} \beta = 1.6 + \alpha \\ \alpha + 1.6 + \alpha = 3 \end{cases}$$

$$\alpha = 0.7 \quad \beta = 2.3 \Rightarrow g(x) = 0.7x + 2.3$$

Tras esto, usamos dicho isomorfismo para hallar los nodos de Chebyshev en el intervalo deseado:

(% i6) nodes:makelist(0.7 * v1[i] + 2.3, i, 1, N);

$$[0.7\cos\left(\frac{\pi}{14}\right)+2.3, 0.7\cos\left(\frac{3\pi}{14}\right)+2.3, 0.7\cos\left(\frac{5\pi}{14}\right)+2.3, 2.3, 0.7\cos\left(\frac{9\pi}{14}\right)+2.3, 0.7\cos\left(\frac{11\pi}{14}\right)+2.3, 0.7\cos\left(\frac{13\pi}{14}\right)+2.3] \quad \text{(nodes)}$$

(% i7) float(nodes);

$$[2.982449538527276, 2.847282037727621, 2.60371861738229, 2.3, 1.996281382617709, 1.752717962272379, 1.617550461472723] \quad (\%\ o7)$$

Ahora les aplicamos $f(x_i) = \sqrt{|x_i - 2|}$ para obtener los pares que nos hacen falta para hallar el polinomio de interpolación:

(% i8) f(x):=(sqrt(abs(x-2)));

$$\mathrm{f}(x) := \sqrt{|x-2|} \quad (\%\ o8)$$

(% i11) yi:makelist(f(nodes[i]),i,1,N);

$$[\sqrt{0.7\cos\left(\frac{\pi}{14}\right)+0.2999999999999998}, \sqrt{0.7\cos\left(\frac{3\pi}{14}\right)+0.2999999999999998}, \sqrt{0.7\cos\left(\frac{5\pi}{14}\right)+0.2999999999999998} \quad \text{(yi)}$$

$$, 0.547722557505166, \sqrt{-0.7\cos\left(\frac{9\pi}{14}\right)-0.2999999999999998}, \sqrt{-0.7\cos\left(\frac{11\pi}{14}\right)-0.2999999999999998}, \sqrt{-0.7\cos\left(\frac{13\pi}{14}\right)-0.2999999999999998}]$$

(% i12) float(yi);

$$[0.9911859253072939, 0.9204792435072182, 0.7769933187500975, 0.547722557505166, 0.06098046721935467, 0.4972746099768427, 0.6184250468143061] \quad (\%\ o12)$$

Ordenando por pares obtenemos que los puntos para los cuales queremos hallar el polinomio de interpolación son:

$$(2.982, 0.991) \quad (2.847, 0.920) \quad (2.603, 0.777) \quad (2.3, 0.548)$$

$$(1.996, 0.061) \quad (1.753, 0.497) \quad (1.618, 0.618)$$

Ahora ya tenemos calculados los pares, y podemos pasar a resolver el problema de interpolación:

- Lagrange

  Sabemos que $p(x) = \sum_{i=0}^{N} y_i l_i(x)$. Los y_i los hemos calculado previamente, pero aún tenemos que calcular los polinomios $l_i(x)$. Los calculamos usando la fórmula vista en teoría:

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

  Si usamos *Máxima* para realizar los cálculos de manera única y directa obtenemos que:

  (% i44) li(i,x):=product((x-nodes[j])/(nodes[i]-nodes[j]),j,1,i-1)*product((x-nodes[j])/(nodes[i]-nodes[j]),j,i+1,N);

  $$\mathrm{li}\,(i,x) := \prod_{j=1}^{i-1} \frac{x - nodes_j}{nodes_i - nodes_j} \prod_{j=i+1}^{N} \frac{x - nodes_j}{nodes_i - nodes_j} \quad (\%\ o44)$$

(% i45) lagrange(x):=sum(yi[i]*li(i,x),i,1,N);

$$\text{lagrange}(x) := \sum_{i=1}^{N} yi_i \, \text{li}\,(i,x) \qquad (\%\ \text{o45})$$

(% i46) float(expand(lagrange(x)));

$$-24.75292799807801x^6 + 349.9934972799464x^5 - 2041.754186135889 \qquad (\%\ \text{o46})$$

x^4+6285.746443881555x^3−10762.69525253002x^2+9711.293214095544x−3605.252707524353

Por tanto, podemos afirmar que el polinomio de Lagrange que interpola dichos puntos es $-24.752x^6 + 349.993x^5 - 2041.754x^4 + 6285.746x^3 - 10762.695x^2 + 9711.293x - 3065.2527$.

- Newton

  En este caso, el polinomio se calcula de la forma $p(x) = \sum_{i=1}^{N} \alpha_i \cdot \omega_i(x)$ de donde

$$\begin{cases} \alpha_i = f[x_0, x_1, \ldots, x_i] = \frac{f[x_1,\ldots,x_i] - f[x_0,\ldots,x_{i-1}]}{x_i - x_0} \Rightarrow \text{diferencias divididas} \\ \omega_i(x) = \begin{cases} \prod_{j=0}^{i-1}(x - x_j) & i > 0 \\ 1 & i = 0 \end{cases} \end{cases}$$

  Comenzamos calculando los α_i:

(% i22) A: genmatrix(lambda([i,j], 0), N, N+1);

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad (\text{A})$$

(% i23) for i:1 thru N do (A[i][1]: nodes[i], A[i][2]:yi[i]);

$$done \qquad (\%\ \text{o23})$$

(% i24) for j:3 thru N + 1 dofor i:(j-1) thru N doA[i][j]:(A[i,j-1]-A[i-1,j-1])/(nodes[i]-nodes[i-(j-2)]);

$$done \qquad (\%\ \text{o24})$$

(% i26) alphaN:float(A[N,N+1]);

$$-24.752927998078 \qquad (\text{alphaN})$$

Ahora calculamos los polinomios de la base:

(% i27) omega(i,x):=product((x-nodes[j]),j,1,i-1);

$$\text{omega}\,(i,x) := \prod_{j=1}^{i-1} x - nodes_j \qquad (\%\ \text{o27})$$

(% i28) newton(x):=sum(A[i,i+1]*omega(i,x),i,1,N);

$$\text{newton}(x) := \sum_{i=1}^{N} A_{i,i+1}\,\text{omega}\,(i,x) \qquad (\%\ \text{o28})$$

(% i29) expand(float(newton(x)));

$$-24.752927998078x^6 + 349.9934972799465x^5 - 2041.754186135889x^4 \qquad (\%\ \text{o29})$$

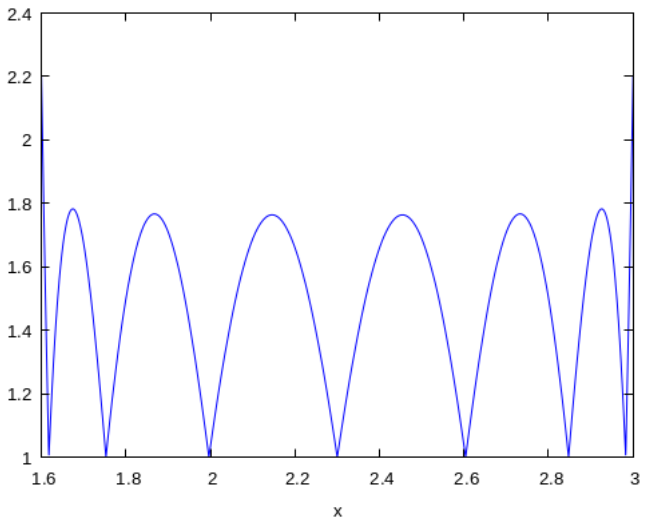
$$+6285.746443881544x^3 - 10762.69525253003x^2 + 9711.293214095553x - 3605.252707524355$$

Observamos que ambos polinomios coinciden.

Ahora pasaremos a estudiar su condicionamiento. Para medir dicho condicionamiento, calculemos la constante de Lebesgue

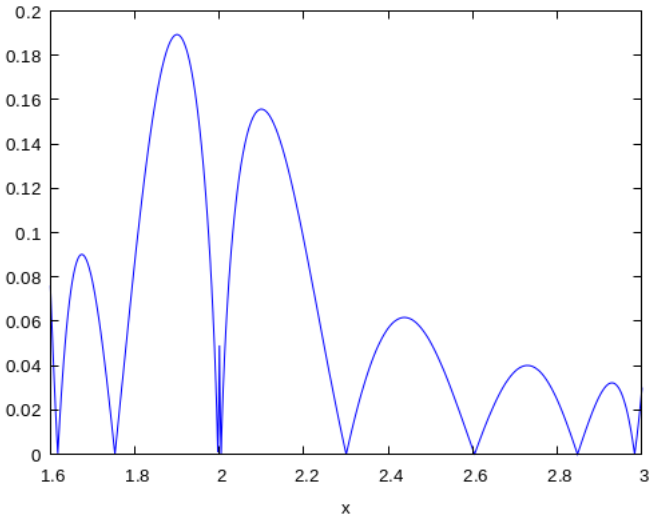
$$\wedge_N := \max_{x\in[a,b]} \sum_{i=0}^{N} |l_i(x)|$$

Representemos dicha función:



Observamos que $\wedge \approx 2.2$ lo cual es un condicionamiento bastante bueno, pues es un valor relativamente pequeño.

Por último, estudiemos el error de interpolación, el cual viene dado por la diferencia de la aproximación con la función real:



Observamos que el error de interpolación es pequeño pues no llega al 0.2.

Ejercicio 5. Considera en el intervalo [-1,1] 9 nodos x_i uniformemente distribuidos y los correspondientes 9 nodos de Chebyshev u_i. Estudia en cada caso el problema de interpolación

$$\text{encontrar } p \in \mathbb{P}_8 : i = 1,2,3,4,5,6,7,8 \Rightarrow p(x_i) = 2|x_i| + 1,$$

así como el análogo para los nodos u_i. Dibuja simultáneamente ambos interpolantes junto con la función $2|x| + 1$, $-1 \leq x \leq 1$.

Primero de todo, calculemos ambos nodos:

(% i1) a:-1;

$$-1 \qquad \text{(a)}$$

(% i3) b:1;

$$1 \qquad \text{(b)}$$

(% i4) N:8;

$$8 \qquad \text{(N)}$$

(% i7) f(x):=2*abs(x)+1;

$$\mathrm{f}(x) := 2\,|x| + 1 \qquad \text{(\% o7)}$$

(% i8) h:b-a;

$$2 \qquad \text{(h)}$$

(% i9) x1:makelist(a+i*h/N, i, 0,N);

$$[-1, -\frac{3}{4}, -\frac{1}{2}, -\frac{1}{4}, 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1] \tag{x1}$$

(% i10) y1:makelist(f(x1[i]),i,1,N+1);

$$[3, \frac{5}{2}, 2, \frac{3}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}, 3] \tag{y1}$$

(% i16) x2:makelist(cos(%pi*(2*i+1)/(2*(N+1))),i,0,N);

$$[\cos\left(\frac{\pi}{18}\right), \frac{\sqrt{3}}{2}, \cos\left(\frac{5\pi}{18}\right), \cos\left(\frac{7\pi}{18}\right), 0, \cos\left(\frac{11\pi}{18}\right), \cos\left(\frac{13\pi}{18}\right), -\frac{\sqrt{3}}{2}, \cos\left(\frac{17\pi}{18}\right)] \tag{x2}$$

(% i17) y2:makelist(f(x2[i]),i,1,N+1);

$$[2\cos\left(\frac{\pi}{18}\right)+1, \sqrt{3}+1, 2\cos\left(\frac{5\pi}{18}\right)+1, 2\cos\left(\frac{7\pi}{18}\right)+1, 1, 1-2\cos\left(\frac{11\pi}{18}\right), 1-2\cos\left(\frac{13\pi}{18}\right), \sqrt{3}+1, 1-2\cos\left(\frac{17\pi}{18}\right)] \tag{y2}$$

Ahora calculamos el polinomio de interpolación:
Primero para los nodos equidistantes

(% i18) li1(i,x):=product((x-x1[j])/(x1[i+1]-x1[j]),j,1,i)*
product((x-x1[j])/(x1[i+1]-x1[j]),j,i+2,N+1);

$$\mathrm{li1}\,(i,x) := \prod_{j=1}^{i} \frac{x - x1_j}{x1_{i+1} - x1_j} \prod_{j=i+2}^{N+1} \frac{x - x1_j}{x1_{i+1} - x1_j} \tag{% o18}$$

(% i19) lagrange1(x):=sum(y1[i+1]*li1(i,x),i,0,N);

$$\mathrm{lagrange1}(x) := \sum_{i=0}^{N} y1_{i+1}\,\mathrm{li1}\,(i,x) \tag{% o19}$$

(% i20) float(expand(lagrange1(x)));

$$-32.5079365079365x^8 + 62.57777777777777x^6 - 38.22222222222222x^4 + 10.15238095238095x^2 + 1.0 \tag{% o20}$$

(% i21) li2(i,x):=product((x-x2[j])/(x2[i+1]-x2[j]),j,1,i)*
product((x-x2[j])/(x2[i+1]-x2[j]),j,i+2,N+1);

$$\mathrm{li2}\,(i,x) := \prod_{j=1}^{i} \frac{x - x2_j}{x2_{i+1} - x2_j} \prod_{j=i+2}^{N+1} \frac{x - x2_j}{x2_{i+1} - x2_j} \tag{% o21}$$

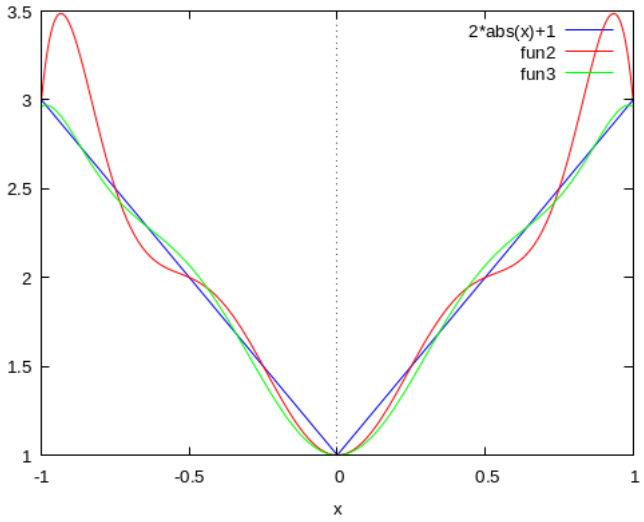
(% i22) lagrange2(x):=sum(y2[i+1]*li2(i,x),i,0,N);

$$\mathrm{lagrange2}(x) := \sum_{i=0}^{N} y2_{i+1}\,\mathrm{li2}\,(i,x) \tag{% o22}$$

(% i23) float(expand(lagrange2(x)));

$$-10.35293110801629x^8 - 2.36255459640233310^{-13}\,x^7 + 24.08456014962255x^6 + 2.99760216648792210^{-13}\,x^5 \tag{% o23}$$
$$-19.59667954343477x^4 - 1.42996725571720110^{-13}\,x^3 + 7.82698846136628x^2 + 9.4368957093138310^{-15}x + 1.0$$

Y por último representamos los polinomios como se nos indica en el enunciado:



Ejercicio 7. Dada la partición uniforme P del intervalo [-1,1] determinada por 6 puntos y la función de Runge f, determina el spline s que verifica

$$i = 0, 1, 2, 3, 4, 5 \Rightarrow s(-1 + 2i/5) = f(-1 + 2i/5),$$

siendo, o bien $s = \mathbf{S}_5^1 \in \mathbb{S}_0^1(P)$, o bien $s = \mathbb{S}_5^2 \in \mathbb{S}_3^2(P)$ con $s''(-1) = 0$ (natural). Ilustra con un ejemplo el principio de mínima energía para este último spline.

Comencemos con el spline lineal:

(% i1) N:5;

$$5 \qquad \text{(N)}$$

(% i2) a:-1;

$$-1 \qquad \text{(a)}$$

(% i3) b:1;

$$1 \qquad \text{(b)}$$

(% i5) h:(b-a)/N;

$$\frac{2}{5} \qquad \text{(h)}$$

(% i9) xi:makelist(-1+2*i/N, i, 0, N);

$$[-1, -\frac{3}{5}, -\frac{1}{5}, \frac{1}{5}, \frac{3}{5}, 1] \qquad \text{(xi)}$$

(% i7) runge(x):=1/(1+25*x^2);

$$\text{runge}(x) := \frac{1}{1 + 25x^2} \qquad \text{(\% o7)}$$

(% i23) B(i, x):= if i = 0 then (if x< xi[1] then 0 elseif x<= xi[2] then (xi[2] - x) /(xi[2] - xi[1]) else 0)
else if i<= N - 1 then (if x<= xi[i] then 0 elseif x<= xi[i + 1] then (x - xi[i]) /(xi[i + 1] - xi[i]) elseif x<= xi[i + 2] then (xi[i + 2] - x) /(xi[i + 2] - xi[i + 1]) else0)
else if i = N then(if x<= xi[N] then 0 else if x<= xi[N + 1] then (x - xi[N]) /(xi[N + 1] - xi[N]) else 0) ;

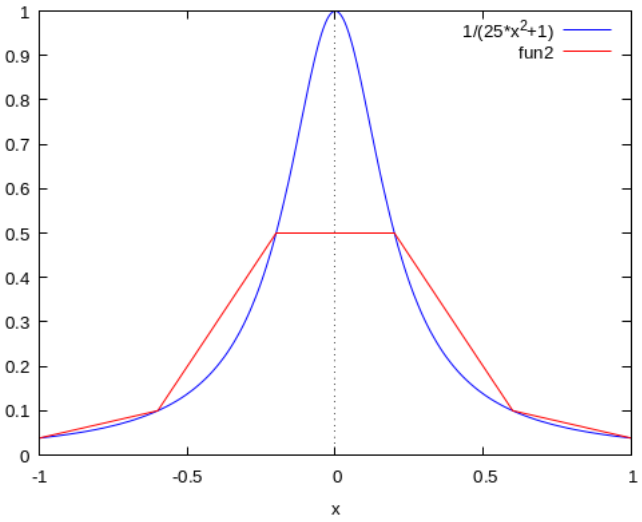
(% i24) s(x):=sum(runge(xi[i+1])*B(i,x),i,0,N);

$$\mathrm{s}(x) := \sum_{i=0}^{N} \mathrm{runge}\left(xi_{i+1}\right) \mathrm{B}\left(i, x\right) \qquad (\%\ \mathrm{o24})$$

(% i26) wxplot2d([runge(x), s(x)], [x, a, b]) ;

(% t26)

(% o26)



Pasemos ahora al spline cúbico:

(% i8) A:2*ident(N+1);

$$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix} \qquad (\mathrm{A})$$

(% i9) for i:2 thru N do (A[i,i-1]:1/2);

done (% o9)

(% i10) for i:3 thru N do (A[i-1,i]:1/2);

done (% o10)

(% i11) yi:makelist(0,i,1,N+1);

$$[0,0,0,0,0,0] \tag{yi}$$

(% i12) for i:2 thru N do (yi[i]: runge(xi[i+1])-2*runge(xi[i]) + runge(xi[i-1]));

$$done \tag{\% o12}$$

(% i13) yi;

$$[0,\frac{22}{65},-\frac{2}{5},-\frac{2}{5},\frac{22}{65},0] \tag{\% o13}$$

(% i14) cm:((3/h^2)*invert(A).yi);

$$\begin{pmatrix} 0 \\ \frac{1020}{247} \\ -\frac{945}{247} \\ -\frac{945}{247} \\ \frac{1020}{247} \\ 0 \end{pmatrix} \tag{cm}$$

(% i15) c:makelist(0,i,1,N+1);

$$[0,0,0,0,0,0] \tag{c}$$

(% i16) for i:1 thru N+1 do (c[i]: cm[i,1]);

$$done \tag{\% o16}$$

(% i17) c;

$$[0,\frac{1020}{247},-\frac{945}{247},-\frac{945}{247},\frac{1020}{247},0] \tag{\% o17}$$

(% i18) rungexi:makelist(runge(xi[i]),i,1,N+1);

$$[\frac{1}{26},\frac{1}{10},\frac{1}{2},\frac{1}{2},\frac{1}{10},\frac{1}{26}] \tag{rungexi}$$

(% i19) alpha:makelist(0,i,1,N);

$$[0,0,0,0,0] \tag{alpha}$$

(% i20) for i:1 thru N doalpha[i] :(rungexi[i + 1] - rungexi[i]) / h - (h / 6).(c[i + 1] - c[i]) ;

$$done \tag{\% o20}$$

(% i21) alpha;

$$[-\frac{30}{247},\frac{378}{247},0,-\frac{378}{247},\frac{30}{247}] \tag{\% o21}$$

(% i22) bet:makelist(0,i,1,N);

$$[0,0,0,0,0] \qquad \text{(bet)}$$

(% i23) for i:1 thru N dobet[i]: rungexi[i]-c[i]*(h^2)/6;

$$done \qquad (\%\ o23)$$

(% i24) bet;

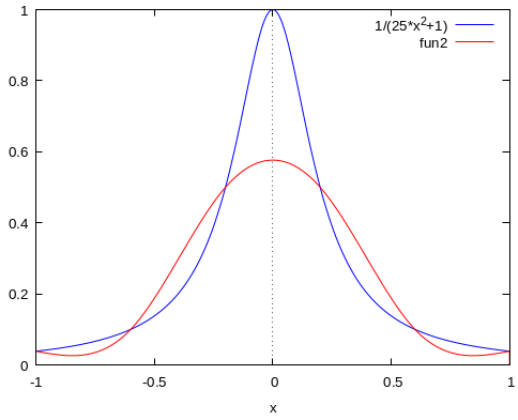
$$[\frac{1}{26}, -\frac{5}{494}, \frac{1487}{2470}, \frac{1487}{2470}, -\frac{5}{494}] \qquad (\%\ o24)$$

(% i26) for i:1 thru N dos(i,x):=(c[i]*(xi[i+1]-x)^3)/(6*h) + (c[i+1]*(x-xi[i])^3)/(6*h) + alpha[i]*(x-xi[i])+bet[i];

$$done \qquad (\%\ o26)$$

(% i27) p(i,x):= if x < xi[i] then 0 elseif x < xi[i+1] then s(i,x) else 0;
(% i28) splinecubic(x):= sum((p(i,x)),i,1,N);

$$\text{splinecubic}(x) := \sum_{i=1}^{N} \text{p}\,(i,x) \qquad (\%\ o28)$$



Por último, ilustraremos con un ejemplo el principio de mínima energía para este último spline, el cual dice así: Sí s es una función spline cúbica natural que satisface la misma condición de interpolación que otro polinomio $g \in C^2([a,b])$, entonces

$$\int_a^b s''(x)^2 dx \leq \int_a^b g''(x)^2 dx$$

dándose la igualdad si, y sólo si, $g = s$.

Primero debemos de calcular un poliniomio de interpolación por el método que prefiramos, en este caso Lagrange:

(% i42) li(i,x):= product((x-xi[j])/(xi[i+1]-xi[j]),j,1,i)*product((x-xi[j])/(xi[i+1]-xi[j]),j,i+2,N+1);

$$\text{li}\,(i,x) := \prod_{j=1}^{i} \frac{x - xi_j}{xi_{i+1} - xi_j} \prod_{j=i+2}^{N+1} \frac{x - xi_j}{xi_{i+1} - xi_j} \qquad (\%\ o42)$$

(% i43) lagrange(x):=sum(rungexi[i+1]*li(i,x),i,0,N);

$$\text{lagrange}(x) := \sum_{i=0}^{N} rungexi_{i+1}\,\text{li}\,(i,x) \qquad (\%\ o43)$$

(% i44) float(expand(lagrange(x)));

$$1.201923076923076x^4 - 1.73076923076923x^2 + 0.5673076923076923 \qquad (\%\ o44)$$

Ahora hacemos la comparación entre las integrales de ambas expresiones (spline y polinomio) de su derivada segunda usando *Máxima*:

Integramos el spline:

(% i37) float(integrate((diff(s(5,x),x,2))^2,x,xi[5],xi[6])+integrate((diff(s(4,x),x,2))^2,x,xi[4],xi[5])
+integrate((diff(s(3,x),x,2))^2,x,xi[3],xi[4])
+integrate((diff(s(2,x),x,2))^2,x,xi[2],xi[3])+integrate((diff(s(1,x),x,2))^2,x,xi[1],xi[2]));

$$14.64029897228277 \qquad (\%\ o37)$$

Por tanto tenemos que

$$\int_a^b s''(x)^2 dx = 14.64029897228277$$

Por otro lado, calculemos la integral del polinomio de Lagrange:

(% i39) float(integrate((diff(lagrange(x),x,2))^2,x,-1,1));

$$40.60650887573964 \qquad (\%\ o39)$$

Por tanto tenemos que

$$\int_a^b g''(x)^2 dx = 40.60650887573964$$

quedando así ilustrado el principio de mínima energía.