Relación de ejercicios 3 Javier Gómez López

1 Programa la forma de Lagrange del polinomio p de grado menor o igual que N que resuelve el problema de interpolación polinómica: dados (x0, y0), (x1, y1),...,(xN, yN) (abscisas distintas dos a dos), se tiene que j=0,1,...,N = p(xj) = yj. Para ello, calcula previamente los poliniomios de Lagrange. Aplícalo al problema: encuentra el polinimio p de grado menor o igual que 8 de forma que p(j/8)=sen(j/8) - j/4, (j=0,1,...8). Dibuja simultáneamente las gráficas de p y de f(x)=senx-2x en el intervlao [0,1].

Primero declaramos el numero de puntos, su lista y sus imágenes

```
--> x : makelist( j / 8, j, 0, 8) ;
```

$$[0, \tfrac{1}{8}, \tfrac{1}{4}, \tfrac{3}{8}, \tfrac{1}{2}, \tfrac{5}{8}, \tfrac{3}{4}, \tfrac{7}{8}, 1]$$

```
--> y : makelist( sin( j / 8) − j / 4, j, 0, 8) ;
```

$$[0, \sin\left(\tfrac{1}{8}\right)-\tfrac{1}{4}, \sin\left(\tfrac{1}{4}\right)-\tfrac{1}{2}, \sin\left(\tfrac{3}{8}\right)-\tfrac{3}{4}, \sin\left(\tfrac{1}{2}\right)-1, \sin\left(\tfrac{5}{8}\right)-\tfrac{5}{4}, \sin\left(\tfrac{3}{4}\right)-\tfrac{3}{2}, \sin\left(\tfrac{7}{8}\right)-\tfrac{7}{4}, \sin(1)-2]$$

```
--> N : length( x) ;
```

$$9$$

Ahora calculamos I_i(x). Para ello se calcula como el producto desde j=0 hasta N con j!=i de (x-x_j)/(x_i-x_j)

```
--> I[ i, z] : = product((( z − x[ j]) /( x[ i] − x[ j])), j, 1, i − 1)· product((( z − x[ j]) /( x[ i] − x[ j])), j, i + 1, N) ;
```

$$I_{i,z} := \prod_{j=1}^{i-1} \frac{z-x_j}{x_i-x_j} \prod_{j=i+1}^{N} \frac{z-x_j}{x_i-x_j}$$

Ahora calculamos p(x)

```
--> p( z) : = sum( y[ i]· I[ i, z], i, 1, N) ;
```

$$p(z) := \sum_{i=1}^{N} y_i I_{i,z}$$

```
--> float( expand( p( z))) ;
```

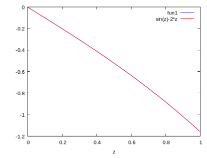
$$1.18287365467040210^{-5}z^8-2.20308210600705910^{-4}z^7+2.2417555328502210^{-5}z^6+0.008319591712734109z^5+5.11623886723100410^{-6}z^4-0.1666677845536526z^3+1.29241286117576210^{-7}z^2-1.000000005932017z$$

Ahora dibujamos sus gráficas para comprobar el resultado

```
--> f( x) : = sin( x) − 2· x ;
```

$$f(x) := \sin(x)-2x$$

```
--> wxplot2d([ p( z), f( z)], [ z, 0, 1]) $
```



2 Programa la forma de Newton del polinomio p que resuelve del problema de la interpolación polinómica anterior. Para ello, calcula previamente los polinomios nodales y las diferentes divididas. Aplicalo a los mismos datos del ejercicio anterior y comprueba el resultado que obtienes es el mismo.

De nuevo, debemos declarar una lista con sus puntos, otra con sus imágenes, y el número de puntos. Por otro lado, declaramos una matriz auxiliar para el cálculo de las diferencias divididas.

```
--> x ;
```

$$[0, \tfrac{1}{8}, \tfrac{1}{4}, \tfrac{3}{8}, \tfrac{1}{2}, \tfrac{5}{8}, \tfrac{3}{4}, \tfrac{7}{8}, 1]$$

```
--> y ;
```

$$[0, \sin\left(\tfrac{1}{8}\right)-\tfrac{1}{4}, \sin\left(\tfrac{1}{4}\right)-\tfrac{1}{2}, \sin\left(\tfrac{3}{8}\right)-\tfrac{3}{4}, \sin\left(\tfrac{1}{2}\right)-1, \sin\left(\tfrac{5}{8}\right)-\tfrac{5}{4}, \sin\left(\tfrac{3}{4}\right)-\tfrac{3}{2}, \sin\left(\tfrac{7}{8}\right)-\tfrac{7}{4}, \sin(1)-2]$$

```
--> N ;
```

$$9$$

```
--> difs : genmatrix( lambda([ i, j], 0), N, N) ;
```

$$\begin{pmatrix} 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0 \end{pmatrix}$$

La primera columna son las imágenes de los puntos

```
--> for i : 1 thru N do difs[ i, 1] : y[ i] ;
```

done

```
--> difs ;
```

$$\begin{pmatrix} 0&0&0&0&0&0&0&0&0\\ \sin\left(\frac{1}{8}\right)-\frac{1}{4}&0&0&0&0&0&0&0&0\\ \sin\left(\frac{1}{4}\right)-\frac{1}{2}&0&0&0&0&0&0&0&0\\ \sin\left(\frac{3}{8}\right)-\frac{3}{4}&0&0&0&0&0&0&0&0\\ \sin\left(\frac{1}{2}\right)-1&0&0&0&0&0&0&0&0\\ \sin\left(\frac{5}{8}\right)-\frac{5}{4}&0&0&0&0&0&0&0&0\\ \sin\left(\frac{3}{4}\right)-\frac{3}{2}&0&0&0&0&0&0&0&0\\ \sin\left(\frac{7}{8}\right)-\frac{7}{4}&0&0&0&0&0&0&0&0\\ \sin(1)-2&0&0&0&0&0&0&0&0 \end{pmatrix}$$

Ahora calculo las diferencias divididas.

```
--> for i : 2 thru N do(
      for j : 2 thru i do(
        difs[ i, j] : float(( difs[ i, j − 1] − difs[ i − 1, j − 1]) /( x[ i] − x[ i − j + 1]))
        )
      ) ;
```

done

```
--> difs ;
```

$$\begin{pmatrix} 0&0&0&0&0&0&0&0&0\\ \sin\left(\frac{1}{8}\right)-\frac{1}{4}&-1.002602132918178&0&0&0&0&0&0&0\\ \sin\left(\frac{1}{4}\right)-\frac{1}{2}&-1.018166193045638&-0.06225624050983924&0&0&0&0&0&0\\ \sin\left(\frac{3}{8}\right)-\frac{3}{4}&-1.049051441347803&-0.1235409932086595&-0.163426007196854&0&0&0&0&0\\ \sin\left(\frac{1}{2}\right)-1&-1.094775923854756&-0.1828979300278135&-0.1582851648510773&0.01028168469155355&0&0&0&0\\ \sin\left(\frac{5}{8}\right)-\frac{5}{4}&-1.154626125309926&-0.2394008058206793&-0.1506743354476422&0.01522165880687015&0.007903958584506566&0&0&0\\ \sin\left(\frac{3}{4}\right)-\frac{3}{2}&-1.227668103337024&-0.2921679121083933&-0.1407122834339039&0.01992410402747663&0.007523912352970364&-5.06728308714935810^{-4}&0&0\\ \sin\left(\frac{7}{8}\right)-\frac{7}{4}&-1.312762062298456&-0.3403758358457267&-0.1285544632995557&0.02431564026869637&0.007026457985951584&-6.63272489358372710^{-4}&-1.78907635021070810^{-4}&0\\ \sin(1)-2&-1.408580139425044&-0.3832723085063527&-0.1143905937616693&0.02832773907577268&0.006419358091322103&-8.09466526172641910^{-4}&-1.67078899216307610^{-4}&1.18287358047631710^{-5} \end{pmatrix}$$

Ahora calculamos los w_i de forma que w_0=1 y w_i=(x-x_i)*w_i-1

```
--> w( i, z) : = product( z − x[ j + 1], j, 0, i − 1) ;
```

$$w(i,z) := \prod_{j=0}^{i-1} z-x_{j+1}$$

Y ahora:

```
--> p( z) : = sum( difs[ i + 1, i + 1]· w( i, z), i, 0, N − 1) ;
```

$$p(z) := \sum_{i=0}^{N-1} \mathrm{difs}_{i+1,i+1}\, w(i,z)$$

```
--> expand( p( z)) ;
```

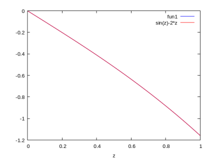
$$1.18287358047631710^{-5}z^8-2.20308210337741810^{-4}z^7+2.24175602330897510^{-5}z^6+0.00831959171958372z^5+5.11624782881192210^{-6}z^4-0.1666677845544147z^3+1.29241221623945110^{-7}z^2-1.000000005932023z$$

Ahora comprobamos

```
--> f( x) : = sin( x) − 2· x ;
```

$$f(x) := \sin(x)-2x$$

```
--> wxplot2d([ p( z), f( z)], [ z, 0, 1]) $
```



3 Sea f la función en C[-1,1] definida como f(x):7.21 cos(2x/π). Considera los nodos x=1-2j/21, j=0,1,...,21, los datos (xj,f(xj)) y los datos perturbados (xj, fj) siendo fj=f(xj)+10^(-3) (-1)^j.

3.1 Halla max(|f(xj)-fj|:j=0,...21).

```
--> f( x) : = 7 . 21· cos( 2· x / %pi) ;
```

$$f(x) := 7.21\cos\left(\frac{2x}{\pi}\right)$$

```
--> xj : makelist( 1 − 2· j / 21, j, 0, 21) ;
```

$$[1, \tfrac{19}{21}, \tfrac{17}{21}, \tfrac{5}{7}, \tfrac{13}{21}, \tfrac{11}{21}, \tfrac{3}{7}, \tfrac{1}{3}, \tfrac{5}{21}, \tfrac{1}{7}, \tfrac{1}{21}, -\tfrac{1}{21}, -\tfrac{1}{7}, -\tfrac{5}{21}, -\tfrac{1}{3}, -\tfrac{3}{7}, -\tfrac{11}{21}, -\tfrac{13}{21}, -\tfrac{5}{7}, -\tfrac{17}{21}, -\tfrac{19}{21}, -1]$$

```
--> fxj : f( xj) ;
```

$$[7.21\cos\left(\tfrac{2}{\pi}\right), 7.21\cos\left(\tfrac{38}{21\pi}\right), 7.21\cos\left(\tfrac{34}{21\pi}\right), 7.21\cos\left(\tfrac{10}{7\pi}\right), 7.21\cos\left(\tfrac{26}{21\pi}\right), 7.21\cos\left(\tfrac{22}{21\pi}\right), 7.21\cos\left(\tfrac{6}{7\pi}\right), 7.21\cos\left(\tfrac{2}{3\pi}\right), 7.21\cos\left(\tfrac{10}{21\pi}\right), 7.21\cos\left(\tfrac{2}{7\pi}\right), 7.21\cos\left(\tfrac{2}{21\pi}\right), 7.21\cos\left(\tfrac{2}{21\pi}\right), 7.21\cos\left(\tfrac{2}{7\pi}\right), 7.21\cos\left(\tfrac{10}{21\pi}\right), 7.21\cos\left(\tfrac{2}{3\pi}\right), 7.21\cos\left(\tfrac{6}{7\pi}\right), 7.21\cos\left(\tfrac{22}{21\pi}\right), 7.21\cos\left(\tfrac{26}{21\pi}\right), 7.21\cos\left(\tfrac{10}{7\pi}\right), 7.21\cos\left(\tfrac{34}{21\pi}\right), 7.21\cos\left(\tfrac{38}{21\pi}\right), 7.21\cos\left(\tfrac{2}{\pi}\right)]$$

```
--> float( fxj) ;
```

[5.797631861529588, 6.04669437645163, 6.273535736759698, 6.477322317232071, 6.657305217257767, 6.812823012990908, 6.943304188034422, 7.048269233720542, 7.127332411269688, 7.180203169351961, 7.206687211841779, 7.206687211841779, 7.180203169351961, 7.127332411269688, 7.048269233720542, 6.943304188034422, 6.812823012990908, 6.657305217257767, 6.477322317232071, 6.273535736759698, 6.04669437645163, 5.797631861529588]

Ahora calculamos fj (nodos perturbados)

```
--> fj : makelist( f( xj[ i]) + 10 ^( − 3)·( − 1) ^( i − 1), i, 1, 21 + 1) ;
```

$$[7.21\cos\left(\tfrac{2}{\pi}\right)+\tfrac{1}{1000}, 7.21\cos\left(\tfrac{38}{21\pi}\right)-\tfrac{1}{1000}, 7.21\cos\left(\tfrac{34}{21\pi}\right)+\tfrac{1}{1000}, 7.21\cos\left(\tfrac{10}{7\pi}\right)-\tfrac{1}{1000}, 7.21\cos\left(\tfrac{26}{21\pi}\right)+\tfrac{1}{1000}, 7.21\cos\left(\tfrac{22}{21\pi}\right)-\tfrac{1}{1000}, 7.21\cos\left(\tfrac{6}{7\pi}\right)+\tfrac{1}{1000}, 7.21\cos\left(\tfrac{2}{3\pi}\right)-\tfrac{1}{1000}, 7.21\cos\left(\tfrac{10}{21\pi}\right)+\tfrac{1}{1000}, 7.21\cos\left(\tfrac{2}{7\pi}\right)-\tfrac{1}{1000}, 7.21\cos\left(\tfrac{2}{21\pi}\right)+\tfrac{1}{1000}, 7.21\cos\left(\tfrac{2}{21\pi}\right)-\tfrac{1}{1000}, 7.21\cos\left(\tfrac{2}{7\pi}\right)+\tfrac{1}{1000}, 7.21\cos\left(\tfrac{10}{21\pi}\right)-\tfrac{1}{1000}, 7.21\cos\left(\tfrac{2}{3\pi}\right)+\tfrac{1}{1000}, 7.21\cos\left(\tfrac{6}{7\pi}\right)-\tfrac{1}{1000}, 7.21\cos\left(\tfrac{22}{21\pi}\right)+\tfrac{1}{1000}, 7.21\cos\left(\tfrac{26}{21\pi}\right)-\tfrac{1}{1000}, 7.21\cos\left(\tfrac{10}{7\pi}\right)+\tfrac{1}{1000}, 7.21\cos\left(\tfrac{34}{21\pi}\right)-\tfrac{1}{1000}, 7.21\cos\left(\tfrac{38}{21\pi}\right)+\tfrac{1}{1000}, 7.21\cos\left(\tfrac{2}{\pi}\right)-\tfrac{1}{1000}]$$

```
--> float( fj) ;
```

[5.798631861529588, 6.04569437645163, 6.274535736759699, 6.47632231723207, 6.658305217257768, 6.811823012990907, 6.944304188034422, 7.047269233720542, 7.128332411269689, 7.179203169351961, 7.20768721184178, 7.205687211841779, 7.181203169351962, 7.126332411269688, 7.049269233720542, 6.942304188034422, 6.813823012990908, 6.656305217257767, 6.478322317232071, 6.272535736759698, 6.047694376451631, 5.796631861529588]

```
--> maximo : apply( max, float( abs( f( xj) − fj))) ;
```

0.001

```
-->
```

3.2 Estima gráficamente la distancia (norma infinito) de los interpolantes obtenidos para las dos series de datos. ¿Qué se puede decir del condicionamiento del problema de interpolación anterior?

Primero calcularemos el polinomio de interpolación de los nodos originiales, luego para los nodos perturbados. Los representamos gráficamente después.

Comencemos calculando los polinomios de Lagrange

```
--> li( i, x) : = product(( x − xj[ j]) /( xj[ i + 1] − xj[ j]), j, 1, i)· product(( x − xj[ j]) /( xj[ i + 1] − xj[ j]), j, i + 2, 21 + 1) ;
```

$$\mathrm{li}(i,x) := \prod_{j=1}^{i} \frac{x-\mathrm{xj}_j}{\mathrm{xj}_{i+1}-\mathrm{xj}_j} \prod_{j=i+2}^{21+1} \frac{x-\mathrm{xj}_j}{\mathrm{xj}_{i+1}-\mathrm{xj}_j}$$

```
--> lagrangeoriginal( x) : = sum( fxj[ i + 1]· li( i, x), i, 0, 21) ;
```

$$\mathrm{lagrangeoriginal}(x) := \sum_{i=0}^{21} \mathrm{fxj}_{i+1}\, \mathrm{li}(i,x)$$

```
--> lagrangeperturbados( x) : = sum( fj[ i + 1]· li( i, x), i, 0, 21) ;
```

$$\mathrm{lagrangeperturbados}(x) := \sum_{i=0}^{21} \mathrm{fj}_{i+1}\, \mathrm{li}(i,x)$$

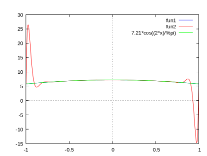
```
--> float( expand( lagrangeoriginal( x))) ;
```

$$5.36990709431685510^{-8}x^{21}+1.55591806105803710^{-9}x^{20}-2.14587113297740110^{-7}x^{19}+1.78906702785752710^{-8}x^{18}+4.23533619307828210^{-7}x^{17}-5.88170223636552710^{-9}x^{16}-4.52616784930258210^{-7}x^{15}+8.90304363565519510^{-9}x^{14}+1.53255762129691710^{-7}x^{13}-9.10858943825587710^{-9}x^{12}-5.29123906408585310^{-8}x^{11}-1.91037088370649110^{-8}x^{10}+1.42754045000593910^{-8}x^9+4.82325072326261810^{-6}x^8-1.75761982954820910^{-9}x^7-6.66628118826340110^{-4}x^6+1.13873937674432510^{-10}x^5+0.04934515470271$$

```
--> float( expand( lagrangeperturbados( x))) ;
```

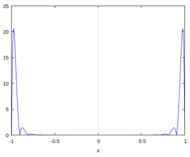
$$114356.6113686633x^{21}-8.85802364791743510^{-9}x^{20}-453796.0768591593x^{19}+1.23092991088924410^{-6}x^{18}+750535.492452174x^{17}-4.02933983423281510^{-6}x^{16}-674245.0213915462x^{15}-8.52683501761930610^{-6}x^{14}+359608.8977162528x^{13}-3.24438849474972810^{-6}x^{12}-116982.3038132274x^{11}-4.70130174790028810^{-7}x^{10}+22975.78652666949x^9+4.81027733911787410^{-6}x^8-2604.655642032605x^7-6.66629871274970810^{-4}x^6+155.28540594411x^5+0.04934515456368745x^4-4.044180081334606x^3-1.46105146812313$$

```
--> wxplot2d([ lagrangeoriginal( x), lagrangeperturbados( x), f( x)],[ x, − 1, 1]) ;
```



Ahora representamos la distancia entre los interpolantes

```
--> wxplot2d([ abs( lagrangeoriginal( x) − lagrangeperturbados( x))],[ x, − 1, 1]) ;
```



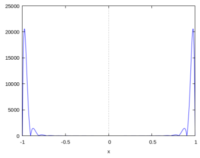
Podemos observar que según nos acercamos a +1 y a -1, la distancia entre los polinomios interpolantes es relativamente elevada. Esto nos indica que el problema no es estable, puesto que las perturbaciones de la imagen son muy grandes antes pequeñas perturbaciones de la x

3.3 Da una estimación de la constante de Lebesgue L y relaciona el valor de este número con el apartado anterior.

```
--> lebesgue( x) : = sum( abs( li( i, x)), i, 0, 21) ;
```

$$\mathrm{lebesgue}(x) := \sum_{i=0}^{21} |\mathrm{li}(i,x)|$$

```
--> wxplot2d([ lebesgue( x)],[ x, − 1, 1]) ;
```



La constante de Lebesgue es aproximadamente 2000. Esto nos indica un mal condicionamiento, y observamos que se alcanza justo cerca de los puntos donde el problema no era estable.

3.4 Determina los 22 nodos de Chebyshev en el intervalo [-1,1] y resuelve el problema de interpolación para esos nodos y la misma función f. Analia el condicionamiento de este nuevo problema.

```
--> N : 21 ;
```

$$21$$

```
--> chebysev : float( makelist( cos( %pi·( 2· i + 1) /( 2·( N + 1))), i, 0, N)) ;
```

$$[0.9974521146102535, 0.9771468659711595, 0.9369497249997617, 0.8776789895672557, 0.8005412409243604, 0.7071067811865475, 0.599277666511347, 0.479248986720057, 0.3494641795990984, 0.2125652895529766, 0.07133918319923236, -0.07133918319923224, -0.2125652895529767, -0.3494641795990983, -0.4792489867200565, -0.599277666511347, -0.7071067811865475, -0.8005412409243603, -0.8776789895672557, -0.9369497249997616, -0.9771468659711594, -0.9974521146102535]$$

```
--> liche( i, x) : = product(( x − chebysev[ j]) /( chebysev[ i + 1] − chebysev[ j]), j, 1, i)· product(( x − chebysev[ j]) /( chebysev[ i + 1] − chebysev[ j]), j, i + 2, N + 1) ;
```

$$\mathrm{liche}(i,x) := \prod_{j=1}^{i} \frac{x-\mathrm{chebysev}_j}{\mathrm{chebysev}_{i+1}-\mathrm{chebysev}_j} \prod_{j=i+2}^{N+1} \frac{x-\mathrm{chebysev}_j}{\mathrm{chebysev}_{i+1}-\mathrm{chebysev}_j}$$

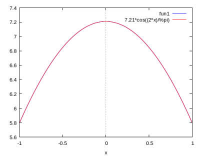
```
--> lagrangeche( x) : = sum( f( chebysev[ i + 1])· liche( i, x), i, 0, N) ;
```

$$\mathrm{lagrangeche}(x) := \sum_{i=0}^{N} f\left(\mathrm{chebysev}_{i+1}\right)\mathrm{liche}(i,x)$$

```
--> float( expand( lagrangeche( x))) ;
```

$$-8.1250802410272410^{-10}x^{21} + 1.45519152283668510^{-10}x^{20} + 7.62473823478617210^{-9}x^{19} + 1.33877620100975 10^{-8}x^{18} + 6.59471771940827310^{-9}x^{17} + 1.58324837684631310^{-8}x^{16} + 2.86960723130371110^{-8}x^{15} + 4.28408384323120110^{-8}x^{14} + 6.95137809251444 10^{-8}x^{13} + 1.02387275546789110^{-7}x^{12} + 9.32205398495681 10^{-8}x^{11} + 3.38186509907245610^{-8}x^{10} + 2.23301807002845810^{-8}x^{9} + 4.83157054986804710^{-6}x^{8} + 1.76497890980275910^{-9}x^{7} - 6.66627547616371910^{-4}x^{6} + 5.88899698944309810^{-11}x^{5} + 0.049345154$$

```
--> wxplot2d([ lagrangeche( x), f( x)],[ x, − 1, 1]) ;
```

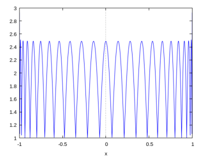


En este caso el ajuste es perfecto porque los nodos están bien seleccionados. Veamos ahora el condicionamiento del problema en este caso.

```
--> lebesguechebysev( x) : = sum( abs( liche( i, x)), i, 0, N) ;
```

$$\mathrm{lebesguechebysev}(x) := \sum_{i=0}^{N} |\mathrm{liche}(i,x)|$$

```
--> wxplot2d([ lebesguechebysev( x)],[ x, − 1, 1]) ;
```



En este caso, la constante de Lebesgue es mucho menor, 2.5 aproximadamente, lo que indica un buen condicionamiento del problema.

4 Considera un intervalo real cualquiera [a,b], con a<b, y una partición suya P={x0,x1,...,xN}.

4.1 Halla una base del espacio E de funciones splines continuas y afines a trozos.

Hacemos una base genérica, B con los siguientes parámetros:
-i: Elemento i-ésimo de la base
-z: La incógnita en la que se expresa B
-x: Vector de coordenadas x de los nodos
-y: Vector de coordenadas y de los nodos

```
--> B( i, z, x) : =
      if i = 1 then(
        if z< x[ i] then 0
        else if z< = x[ i + 1] then ( x[ i + 1] − z) /( x[ i + 1] − x[ i])
        else 0
      )
      else if i< length( x) then(
          if z< x[ i − 1] then 0
          else if z< x[ i] then ( z − x[ i − 1]) /( x[ i] − x[ i − 1])
          else if z< = x[ i + 1] then ( x[ i + 1] − z) /( x[ i + 1] − x[ i])
          else 0
      )
      else if i = length( x) then(
        if z< x[ i − 1] then 0
        else if z< = x[ i] then ( z − x[ i − 1]) /( x[ i] − x[ i − 1])
        else 0
      )
      else
        0 ; ;
```

$$B(i,z,x) := \mathrm{if}\, i=1\, \mathrm{then}\, \mathrm{if}\, z < x_i\, \mathrm{then}\, 0\, \mathrm{elseif}\, z <= x_{i+1}\, \mathrm{then} \frac{x_{i+1}-z}{x_{i+1}-x_i} \mathrm{else}\, 0\, \mathrm{elseif}\, i < \mathrm{length}(x)\, \mathrm{then}\, \mathrm{if}\, z < x_{i-1}\, \mathrm{then}\, 0\, \mathrm{elseif}\, z < x_i\, \mathrm{then} \frac{z-x_{i-1}}{x_i-x_{i-1}} \mathrm{elseif}\, z <= x_{i+1}\, \mathrm{then} \frac{x_{i+1}-z}{x_{i+1}-x_i} \mathrm{else}\, 0\, \mathrm{elseif}\, i=\mathrm{length}(x)\, \mathrm{then}\, \mathrm{if}\, z < x_{i-1}\, \mathrm{then}\, 0\, \mathrm{elseif}\, z <= x_i\, \mathrm{then} \frac{z-x_{i-1}}{x_i-x_{i-1}} \mathrm{else}\, 0\, \mathrm{else}\, 0$$

4.2 Utiliza la base anterior para encontrar el único elemento s de dicho espacio E de forma que s(xj)=aj, (j=0,1,...,N), siendo aj's escalares dados.

Para este apartado, tomamos la base anterior. Calculamos una partición en el intervalo dado, según las condiciones que nos den. En este caso usaremos la del apartado siguiente:

```
--> x :[ 0 . 4, 0 . 5, 2 . 34, 3 . 45, 4 . 567, 5 . 081, 5 . 26] ;
```

$$[0.4, 0.5, 2.34, 3.45, 4.567, 5.081, 5.26]$$

Ahora definimos la f del siguiente apartado

```
--> f( z) : = 1 − z ^ 2 / 20 . 78 ;
```

$$f(z) := 1 - \frac{z^2}{20.78}$$

```
--> n : length( x) ;
```

$$7$$

Ahora calculamos la imagen de los puntos de la partición dada

```
--> y : makelist( f( x[ i]), i, 1, n) ;
```

$$[0.9923002887391723, 0.9879692011549567, 0.7364966313763235, 0.4272136669874879, -0.003729018286814156, -0.2423754090471608, -0.331453320500481]$$

Por último comprobamos que podemos encontrar un elemento del espacio E que verifica la igualdad pedida, y que es único (pues forma base)

```
--> genmatrix( lambda([ i, j], B( i, x[ j], x)), n, n) ;
```

$$\begin{pmatrix} 1.0 & 0.0 & 0 & 0 & 0 & 0 & 0 \\ 0.0 & 1.0 & 0.0 & 0 & 0 & 0 & 0 \\ 0 & 0.0 & 1.0 & 0.0 & 0 & 0 & 0 \\ 0 & 0 & 0.0 & 1.0 & 0.0 & 0 & 0 \\ 0 & 0 & 0 & 0.0 & 1.0 & 0.0 & 0 \\ 0 & 0 & 0 & 0 & 0.0 & 1.0 & 0.0 \\ 0 & 0 & 0 & 0 & 0 & 0.0 & 1.0 \end{pmatrix}$$

Observamos que forma base y que es único

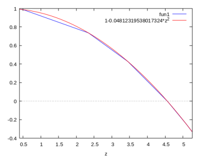
4.3 Aplica lo anterior a la partición P={x0=0.4,x1=0.5,x2=2.34,x3=3.45,x4=4.567,x5=5.081,x6=5.26} del intervalo [0.4,5.26] para encontrar el único elemento s de E de forma que s(xj)=1-xj^2/20.78, j=(0,1,...,6). Dibuja conjuntamente las gráficas de s y de f(x)=1-x^2/20.78.

Ahora construimos la función del spline

```
--> spline( z) : = sum( y[ i]· B( i, z, x), i, 1, n) ;
```

$$\mathrm{spline}(z) := \sum_{i=1}^{n} y_i B(i,z,x)$$

```
--> wxplot2d([ spline( z), f( z)],[ z, 0 . 4, 5 . 26]) ;
```



5 Partiendo de una partición uniforme P={x0,x1,...,xN} de un intervalo real cualquiera [a,b],

5.1 halla el único spline natural s de clase 2 y grado 3 de forma que s(xj)=aj, j=(0,1,...,N), siendo aj escalares dados,

El procedimiento lo aplicamos directamente para que sea más ilustrativo. Seguimos los pasos de las transparencias de clase.

5.2 aplica lo anterior a la partición P del intervalo [-2.09,4.56] en 8 subintervalos iguales y con s(xj)=log √(1 + |xj|), (j=0,1,...,8) y dibuja conjuntamente las gráficas de s y de f(x)=log √(1 + |xj|).

```
--> N : 8 ;
```

$$8$$

```
--> a : − 2 . 09 ;
```

$$-2.09$$

```
--> b : 4 . 56 ;
```

$$4.56$$

```
--> h :( b − a) / N ;
```

$$0.8312499999999999$$

Creamos la partición del intervalo

```
--> xi : makelist( a + h· i, i, 0, N) ;
```

$$[-2.09, -1.25875, -0.4275, 0.40375, 1.234999999999999, 2.06625, 2.8975, 3.72875, 4.56]$$

Definimos la función

```
--> f( x) : = log( sqrt( 1 + abs( x))) ;
```

$$f(x) := \log\left(\sqrt{1 + |x|}\right)$$

Aplicamos al procedimiento visto en clase

```
--> A : 2· ident( N + 1) ;
```

$$\begin{pmatrix}2&0&0&0&0&0&0&0&0\\0&2&0&0&0&0&0&0&0\\0&0&2&0&0&0&0&0&0\\0&0&0&2&0&0&0&0&0\\0&0&0&0&2&0&0&0&0\\0&0&0&0&0&2&0&0&0\\0&0&0&0&0&0&2&0&0\\0&0&0&0&0&0&0&2&0\\0&0&0&0&0&0&0&0&2\end{pmatrix}$$

```
--> for i : 2 thru N do A[ i, i − 1] : 1 / 2 ;
```

done

```
--> for i : 3 thru N + 1 do A[ i − 1, i] : 1 / 2 ;
```

done

```
--> A ;
```

$$\begin{pmatrix}2&0&0&0&0&0&0&0&0\\\frac{1}{2}&2&\frac{1}{2}&0&0&0&0&0&0\\0&\frac{1}{2}&2&\frac{1}{2}&0&0&0&0&0\\0&0&\frac{1}{2}&2&\frac{1}{2}&0&0&0&0\\0&0&0&\frac{1}{2}&2&\frac{1}{2}&0&0&0\\0&0&0&0&\frac{1}{2}&2&\frac{1}{2}&0&0\\0&0&0&0&0&\frac{1}{2}&2&\frac{1}{2}&0\\0&0&0&0&0&0&\frac{1}{2}&2&\frac{1}{2}\\0&0&0&0&0&0&0&0&2\end{pmatrix}$$

```
--> yi : makelist( 0, i, 1, N + 1) ;
```

[0, 0, 0, 0, 0, 0, 0, 0, 0]

```
--> for i : 2 thru N do ( yi[ i] : f( xi[ i + 1]) − 2· f( xi[ i]) + f( xi[ i − 1])) ;
```

done

```
--> yi ;
```

[0, −0.07276368619545992, 0.2210547324493801, 0.2409357182362643, −0.07443995623499308, −0.03816704158289874, −0.02327721517196257, −0.01569418185895899, 0]

Ahora creamos la matriz de las c

```
--> cm :( 3 / h ^ 2)· invert( A). yi ;
```

$$\begin{pmatrix}0.0\\-0.2639287184254621\\0.4238814065967182\\0.4879014754534119\\-0.2833551177217908\\-8.70132142422709810^{-4}\\-0.04458261291668435\\-0.02292391448997783\\0.0\end{pmatrix}$$

```
--> c : makelist( 0, i, 0, N) ;
```

[0, 0, 0, 0, 0, 0, 0, 0, 0]

```
--> for i : 1 thru N + 1 do c[ i] : cm[ i][ 1] ;
```

done

```
--> c ;
```

[0.0, −0.2639287184254621, 0.4238814065967182, 0.4879014754534119, −0.2833551177217908, $-8.70132142422709810^{-4}$, −0.04458261291668435, −0.02292391448997783, 0.0]

Ahora pasamos a calcular los alphas y los betas

```
--> fxi : makelist( f( xi[ i + 1]), i, 0, N) ;
```

[0.564085545454827, 0.4074057814621505, 0.1779623312740141, 0.1695736135352579, 0.402120614032766, 0.560227658295281, 0.6801676609748972, 0.7768304484825509, 0.8577990541312456]

```
--> alpha : makelist( 0, i, 1, N) ;
```

[0, 0, 0, 0, 0, 0, 0, 0]

```
--> for i : 1 thru N do
       alpha[ i] :(( fxi[ i + 1] − fxi[ i]) /( h) −( h / 6)·( c[ i + 1] − c[ i])) ;
```

done

```
--> alpha ;
```

[−0.1519218095946672, −0.3713125567858325, −0.01896113755231057, 0.386606963918338, 0.1510680223120994, 0.1503447249687106, 0.1132854279817167, 0.09422992406192267]

```
--> bet : makelist( 0, i, 1, N) ;
```

[0, 0, 0, 0, 0, 0, 0, 0]

```
--> for i : 1 thru N do
       bet[ i] :( fxi[ i] −( c[ i])·( h ^ 2 / 6)) ;
```

done

```
--> bet ;
```

[0.564085545454827, 0.4378005412292599, 0.1291469784010366, 0.1133855328106785, 0.4347525715677969, 0.5603278651147297, 0.6853019177449703, 0.7794704297547723]

Por último calculamos los splines

```
--> for i : 1 thru N do
       s( 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

Y por último definimos el polinomio

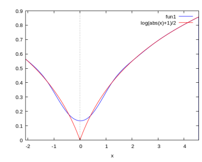
```
--> p( i, x) : = if x< xi[ i] then 0
       else if x< xi[ i + 1] then s( i, x)
       else 0 ;
```

$p(i,x) := \text{if} x < \xi_i\ \text{then}0\text{elseif}x < \xi_{i+1}\ \text{then}s(i,x)\text{else}0$

```
--> splinecubico( x) : = sum(( p( i, x)), i, 1, N) ;
```

$\text{splinecubico}(x) := \sum_{i=1}^{N} p(i,x)$

```
--> wxplot2d([ splinecubico( x), f( x)],[ x, a, b]) ;
```



Created with wxMaxima.