Métodos Numéricos
Práctica 7: Interpolación polinómica y aproximación de funciones

3. Se desea tabular la función de Bessel de orden 0

$$J_0(x) = \frac{1}{\pi}\int_0^\pi \cos(x\sin t)dt$$

en el intervalo $[0,1]$ en abscisas equidistantes. ¿Qué paso de tabulación ha de usarse para que todos los valores obtenidos por interpolación lineal tengan un error (debido a la interpolación) menor que $\frac{1}{2}10^{-6}$? ¿Y si se realiza una interpolación cuadrática?

Solución:

Recordemos que el error de interpolar a J_0 por un polinomio p_n de grado n en $x_0, x_1, \ldots, x_n \in [0,1]$ está dado por

$$E_n(x) = |J_0(x) - p_n(x)| = \frac{|\Phi_n(x)|}{(n+1)!}|J_0^{(n+1)}(\xi(x))|,$$

donde $\Phi_n(x) = (x-x_0)(x-x_1)\ldots(x-x_n)$ y $\xi(x) \in (0,1)$.

Puesto que queremos analizar las interpolaciones lineal y cuadrática, basta con analizar las derivadas de orden 2 y 3 de J_0.

Definamos $f(x,t) = cos(x\sin t)$. Entonces f es continua, con derivadas continuas, por lo que podemos derivar J_0 derivando bajo el signo integral, resultando que

$$J_0'(x) = -\frac{1}{\pi}\int_0^\pi \sin(x\sin t)\sin t\,\mathrm{d}t.$$

Utilizando el mismo argumento, vemos que

$$J_0''(x) = -\frac{1}{\pi}\int_0^\pi \cos(x\sin t)\sin^2 t\,\mathrm{d}t,$$

$$J_0'''(x) = \frac{1}{\pi}\int_0^\pi \sin(x\sin t)\sin^3 t\,\mathrm{d}t.$$

Teniendo en cuenta que las funciones $\sin x$ y $\cos x$ están acotadas entre -1 y 1, obtenemos que

$$|J_0''(x)| = \left|\frac{1}{\pi}\int_0^\pi \cos(x\sin t)\sin^2 t\,\mathrm{d}t\right| \leq \frac{1}{\pi}\int_0^\pi |\cos(x\sin t)||\sin t|^2\,\mathrm{d}t \leq \frac{1}{\pi}\int_0^\pi 1\,\mathrm{d}t = \frac{1}{\pi}\pi = 1,$$

$$|J_0'''(x)| = \left|\frac{1}{\pi}\int_0^\pi \sin(x\sin t)\sin^3 t\,\mathrm{d}t\right| \leq \frac{1}{\pi}\int_0^\pi |\sin(x\sin t)||\sin t|^3\,\mathrm{d}t \leq \frac{1}{\pi}\int_0^\pi 1\,\mathrm{d}t = \frac{1}{\pi}\pi = 1.$$

Luego, los errores de interpolar a J_0 por un polinomio lineal o un polinomio cuadrático está acotado por

$$E_2(x) \leq \frac{|x - x_i||x - x_{i+1}|}{3!},$$
$$E_3(x) \leq \frac{|x - x_i||x - x_{i+1}||x - x_{i+2}|}{4!}.$$

Por otro lado, queremos que las abscisas $x_0, x_1, \ldots, x_n$ elegidas sean equidistantes, por lo que $x_i = \frac{i}{n}$. Luego, si para interpolar J_0 en un punto $x \in [x_i, x_{i+1}]$ utilizamos las abscisas x_i, x_{i+1} para la interpolación lineal, tenemos que

$$|x - x_i| \leq |x_{i+1} - x_i| = \left|\frac{i+1}{n} - \frac{i}{n}\right| = \frac{1}{n}$$

y análogamente $|x - x_{i+1}| \leq \frac{1}{n}$, por lo que

$$E_2(x) \leq \frac{|x - x_i||x - x_{i+1}|}{3!} \leq \frac{1}{6n^2}.$$

Por lo tanto, para que el error sea menor que $\frac{1}{2}10^{-6}$, debe ser

$$\frac{1}{6n^2} \leq \frac{1}{2}10^{-6} \Rightarrow \frac{10^6}{6} \leq n^2 \Rightarrow 409 \leq n.$$

De forma similar, tenemos que, si x pertenece a $[x_i, x_{i+2}]$ para algún $i = 0, 1, \ldots, n-2$, entonces

$$|x - x_i| \leq \frac{2}{n}, \quad |x - x_{i+1}| \leq \frac{2}{n}, \quad |x - x_{i+2}| \leq \frac{2}{n},$$

de donde

$$E_3(x) \leq \frac{|x - x_i||x - x_{i+1}||x - x_{i+2}|}{4!} \leq \frac{8}{24n^3} = \frac{1}{3n^3}.$$

Entonces, para que el error sea menor que $\frac{1}{2}10^{-6}$, debe ser

$$\frac{1}{3n^3} \leq \frac{1}{2}10^{-6} \Rightarrow \frac{2}{3}10^6 \leq n^3 \Rightarrow 88 \leq n.$$

4. Se dispone de la siguiente tabla

x	2.0	2.1	2.2	2.3	2.4	2.5
$J_0(x)$	0.2239	0.1666	0.1104	0.0555	0.0025	-0.0484

para la función de Bessel de orden 0 definida en el ejercicio (3), usar el método de las diferencias divididas de Newton para hallar los valores de $J_0(2.15)$ y $J_0(2.35)$ con errores menores que $\frac{1}{2}10^{-6}$.

Solución:

En Scilab, programamos una función que devuelva el polinomio de interpolación por diferencias divididas para un conjunto de datos dados. Comenzamos con la diferencia dividida de orden k.

//Diferencia dividida de orden k

//x, y vectores de valores donde (x(i),y(i)) son los datos dados para la interpolación, y D es la diferencia dividida de orden k correspondiente

function D=DIFDIVK(x,y)
  $k = length(x);$
  if k==2 **then**
    $D = (y(2) - y(1))/(x(2) - x(1));$
  else
    $D = (DIFDIVK(x(2:k), y(2:k)) - DIFDIVK(x(1:k-1), y(1:k-1)))/(x(k) - x(1));$
  end if
end function

Luego, damos el siguiente programa para el polinomio de interpolación por diferencias divididas:

//Polinomio de interpolación por diferencias divididas

//v, w vectores de valores donde (v(i),w(i)) son los datos dados para la interpolación. P es el polinomio de interpolación por diferencias divididas correspondiente

function P=POLYDIFDIV(v,w)
  $n = length(v);$
  $P = w(1);$
  for k = 2:n **do**
    $P = P + DIFDIVK(v(1:k), w(1:k)) * poly(v(1:k-1), "x", ["roots"]);$
  end for
end function

Definimos los vectores x e y correspondientes a los valores dados en la tabla:

```
--> x=[2, 2.1, 2.2, 2.3, 2.4, 2.5]
 x  =

   2.   2.1   2.2   2.3   2.4   2.5


--> y=[0.2239, 0.1666, 0.1104, 0.0555, 0.0025, -0.0484]
 y  =

   0.2239   0.1666   0.1104   0.0555   0.0025  -0.0484
```

Luego, calculamos el polinomio de interpolación por diferencias divididas de Newton usando los puntos dados:

```
--> P=polydifdiv(x,y)
 P  =

                             2             3      4             5
   38.8381 -84.5671x +75.23x  -33.633333x  +7.5x  -0.6666667x
```

Finalmente, evaluamos dicho polinomio en $x = 2.15$ y en $x = 2.35$:

```
--> aprox1=horner(P,2.15)
 aprox1  =

   0.1383688


--> aprox2=horner(P,2.35)
 aprox2  =

   0.0287313
```

Teniendo en cuenta el teorema 3, acotaremos el error de aproximación en ambos casos.

Por un lado, utilizando el mismo argumento que en el ejercicio 3, obtenemos que

$$J_0^{(6)}(x) = -\frac{1}{\pi}\int_0^\pi \cos(x\sin t)\sin^6 t\,\mathrm{d}t,$$

de donde

$$|J_0^{(6)}(x)| = \left|\frac{1}{\pi}\int_0^\pi \cos(x\sin t)\sin^6 t\,\mathrm{d}t\right| \leq \frac{1}{\pi}\int_0^\pi |\cos(x\sin t)||\sin t|^6\,\mathrm{d}t \leq \frac{1}{\pi}\int_0^\pi 1\,\mathrm{d}t = \frac{1}{\pi}\pi = 1.$$

Entonces,

$$\begin{aligned}|J_0^{(6)}(2.15) - P(2.15)| &\leq \frac{|2.15-2||2.15-2.1||2.15-2.2||2.15-2.3||2.15-2.4||2.15-2.5|}{6!}\\ &\leq \frac{1}{2\cdot 6!}10^{-6} < \frac{1}{2}10^{-6},\end{aligned}$$

$$
\begin{aligned}
|J_0^{(6)}(2.35) - P(2.35)| &\leq \frac{|2.35-2||2.35-2.1||2.35-2.2||2.35-2.3||2.35-2.4||2.35-2.5|}{6!} \\
&\leq \frac{1}{2\cdot 6!}10^{-6} < \frac{1}{2}10^{-6},
\end{aligned}
$$

por lo que las aproximaciones obtenidas tienen un error menor a $\frac{1}{2}10^{-6}$.

8. Dado el conjunto de datos:

i	1	2	3	4	5	6	7	8	9	10
x_i	4	4.2	4.5	4.7	5.1	5.5	5.9	6.3	6.8	7.1
y_i	102.56	113.18	130.11	142.05	167.53	195.14	224.87	256.73	299.5	326.72

a) Construya una aproximación por mínimos cuadrados de grado 1, 2 y 3. Calcule el error.
b) Utilizando Scilab grafique los datos de la tabla y las sucesivas funciones aproximantes obtenidas en el ítem anterior.

Solución:

a) En Scilab, programamos una función que devuelva una aproximación por mínimos cuadrados. Siguiendo lo visto en teoría, utilizamos la forma

$$f(x) = a_0\Phi_0(x) + a_1\Phi_1(x) + a_2\Phi_2(x) + \cdots + a_n\Phi_n(x),$$

para $\Phi_i(x) = x^i$. Luego, por Teorema 2, sabemos que existe una solución de mínimos cuadrados única si y sólo si $rango(A) = n+1$, en cuyo caso $x = (A^TA)^{-1}A^Tb$, donde

$$A = \begin{bmatrix} \Phi_0(x_1) & \dots & \Phi_n(x_1) \\ \vdots & & \vdots \\ \Phi_0(x_m) & \dots & \Phi_n(x_m) \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & & & & \vdots \\ 1 & x_{m-1} & x_{m-1}^2 & \dots & x_{m-1}^n \\ 1 & x_m & x_m^2 & \dots & x_m^n \end{bmatrix},$$

$$b = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}, \qquad x = \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix}$$

//Aproximacion de minimos cuadrados

//x,y vectores tales que $(x(i), y(i))$ son los puntos dados para realizar la aproximación, n es el grado del polinomio, P es el polinomio de aproximación de mínimos cuadrados, y ERR el error cometido

function [P,ERR] = MINIMOSCUADRADOSN(x,y,n)
$m = length(x)$;
$A = zeros(m, n+1)$; // A contiene los valores de las funciones $\Phi_0(x_i), \Phi_1(x_i), \dots, \Phi_n(x_i)$ en la fila i-ésima, por lo que tiene $n+1$ columnas

// Definimos la matriz A
for i = 1:m **do**
for j = 1:n+1 **do**
$A(i, j) = x(i)\hat{}(j-1)$;
end for

end for

// Calculamos $A^T A$ y $A^T b$
$Amin = A' * A;$
$bmin = A' * y';$

// Calculamos la solución $'a'$
$a = inv(Amin) * bmin;$

// Definimos el polinomio de aproximación
$P = poly(a, "s", ["coeff"]);$

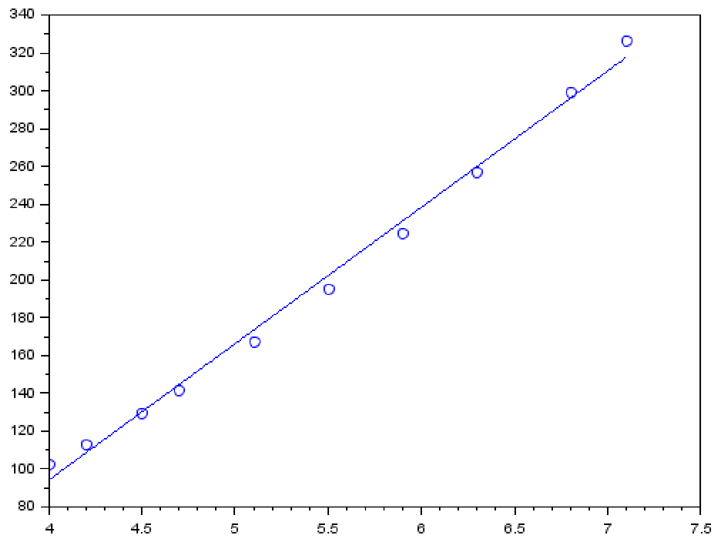
// Calculamos el error de aproximación
$ERR = norm(A * a - y');$

end function

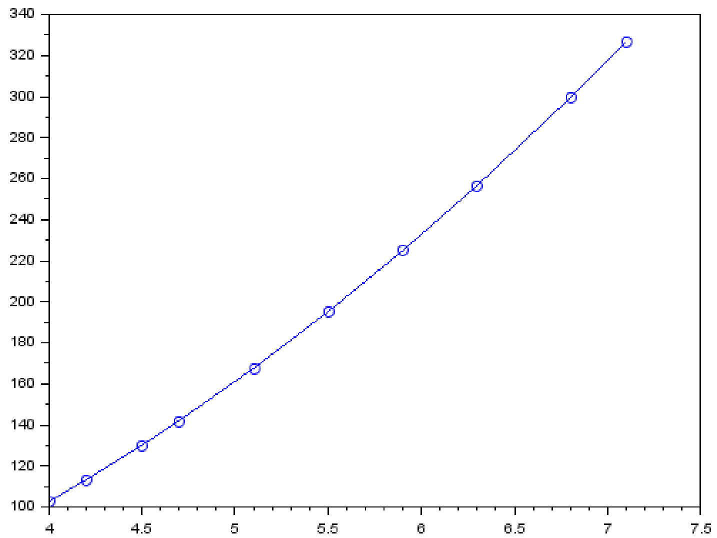
Calculamos los polinomios de aproximación por mínimos cuadrados de grado 1,2 y 3 para los datos dados, junto con sus errores de aproximación

```
--> [P,err]=minimoscuadradosn(x,y,1)
 err  =

   18.138721

 P  =

  -194.13824 +72.084518s


--> [P,err]=minimoscuadradosn(x,y,2)
 err  =

   0.0379857

 P  =
                                  2
   1.2355604 -1.1435234s +6.6182109s


--> [P,err]=minimoscuadradosn(x,y,3)
 err  =

   0.0229639

 P  =
                                  2            3
   3.4290944 -2.3792211s +6.8455778s  -0.0136746s
```

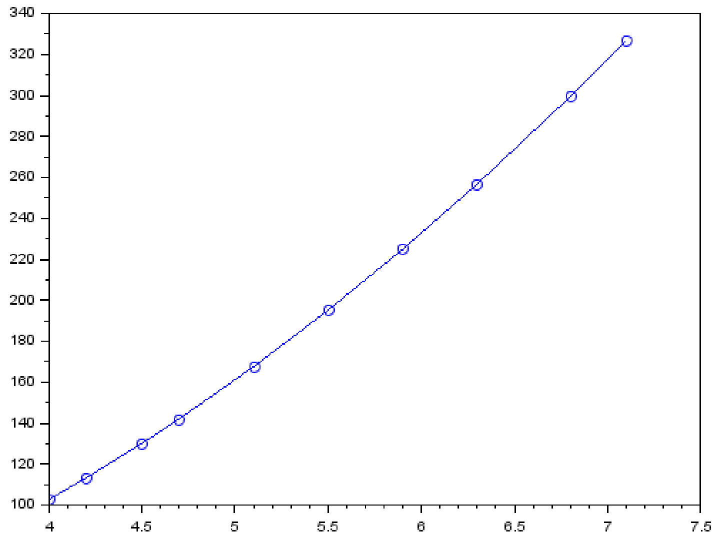
b) Mediante los comandos ‘scatter’ y ‘plot’ graficamos los datos de la tabla y las aproximaciones respectivamente para cada caso:



Gráfica de los datos y la aproximación por mínimos cuadrados de grado 1



Gráfica de los datos y la aproximación por mínimos cuadrados de grado 2



Gráfica de los datos y la aproximación por mínimos cuadrados de grado 3