EJERCICIO COMPUTACIONAL Nº 3. INTERPOLACIÓN POLINÓMICA

Ángel Durán
Departamento de Matemática Aplicada
Universidad de Valladolid

28 de marzo de 2011

Contenidos

Representación del polinomio

$$\begin{array}{c|cccc} x & x_0 & x_1 & \cdots & x_N \\ \hline y & y_0 & y_1 & \cdots & y_N \end{array} \quad x_i \neq x_j \quad i \neq j, \quad N \geq 1$$

Representación de Lagrange

$$P_N(x) = y_0 L_0(x) + y_1 L_1(x) + \cdots y_N L_N(x).$$

$$L_j(x) = \frac{(x - x_0)\cdots(x - x_{j-1})(x - x_{j+1})\cdots(x - x_N)}{(x_j - x_0)\cdots(x_j - x_{j-1})(x_j - x_{j+1})\cdots(x_j - x_N)}.$$

Algoritmo de diferencias divididas

$$
\begin{array}{cccccc}
x_0 & y_0 & & & & \\
x_1 & y_1 & f[x_0, x_1] & & & \\
x_2 & y_2 & f[x_1, x_2] & f[x_0, x_1, x_2] & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & \\
x_N & y_N & f[x_{N-1}, x_N] & f[x_{N-2}, x_{N-1}, x_N] & \cdots & f[x_0, \ldots, x_N]
\end{array}
$$

$$f[x_0] = y_0$$

$$f[x_0, \ldots, x_j] = \frac{f[x_1, \ldots, x_j] - f[x_0, \ldots, x_{j-1}]}{x_j - x_0}, \quad j = 1, \ldots, N.$$

Representación de Newton

$$P_N(x) \quad = \quad f[x_0] + \sum_{j=1}^{N} f[x_0, \ldots, x_j](x - x_0) \cdots (x - x_{j-1})$$

Ejemplo 1

x	1	2	3	4	5
y	3,60	1,80	1,20	0,90	0,72

$$
\begin{array}{llllll}
1 & 3{,}60 & & & & \\
2 & 1{,}80 & -1{,}80 & & & \\
3 & 1{,}20 & -0{,}60 & \frac{-0{,}60+1{,}80}{3-1}=0{,}60 & & \\
4 & 0{,}90 & -0{,}30 & \frac{-0{,}30+0{,}60}{4-2}=0{,}15 & \frac{0{,}15-0{,}60}{4-1}=-0{,}15 & \\
5 & 0{,}72 & -0{,}18 & \frac{-0{,}18+0{,}30}{5-3}=0{,}06 & \frac{0{,}06+0{,}15}{5-2}=-0{,}03 & \frac{-0{,}03+0{,}15}{5-1}=0{,}03
\end{array}
$$

$$
\begin{aligned}
P_4(x) &= 3{,}6L_0(x)+1{,}80L_1(x)++1{,}20L_2(x)+0{,}90L_3(x)+0{,}72L_4(x)\\
&= 3{,}60-1{,}80(x-1)+0{,}60(x-1)(x-2)\\
&\quad -0{,}15(x-1)(x-2)(x-3)+0{,}03(x-1)(x-2)(x-3)(x-4).
\end{aligned}
$$

Ejemplo 2

x	0	$\pi/4$	$\pi/2$	$\pi/6$
y	0	$\sqrt{2}/2$	1	$1/2$

$$\begin{array}{ccccc}
0 & 0 & & & \\
\frac{\pi}{4} & \frac{\sqrt{2}}{2} & \frac{2\sqrt{2}}{\pi} & & \\
\frac{\pi}{2} & 1 & \frac{4-2\sqrt{2}}{\pi} & \frac{8-8\sqrt{2}}{\pi^2} & \\
\frac{\pi}{6} & \frac{1}{2} & \frac{3}{2\pi} & \frac{30-24\sqrt{2}}{\pi^2} & \frac{132-96\sqrt{2}}{\pi^3}
\end{array}$$

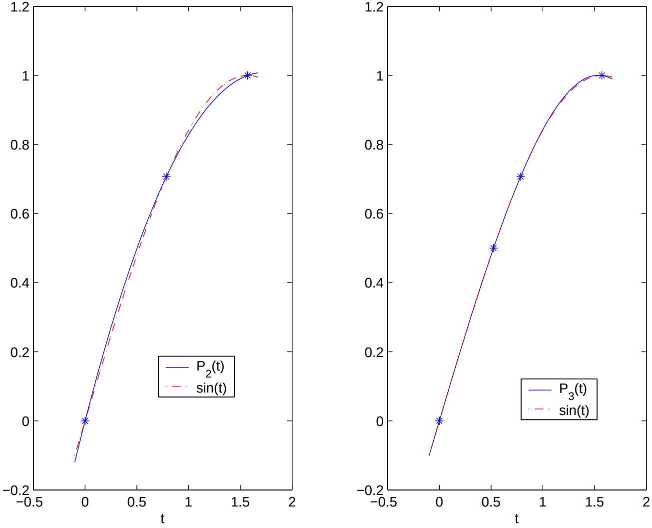
$$\begin{aligned}
P_2(x) &= \frac{2\sqrt{2}}{\pi}x + \frac{8-8\sqrt{2}}{\pi^2}x(x-\frac{\pi}{4}) \\
P_3(x) &= \frac{2\sqrt{2}}{\pi}x + \frac{8-8\sqrt{2}}{\pi^2}x(x-\frac{\pi}{4}) + \frac{132-96\sqrt{2}}{\pi^3}x(x-\frac{\pi}{4})(x-\frac{\pi}{2})
\end{aligned}$$

Algoritmo de diferencias divididas y evaluación

```
function [a]=difd(a,x)
// Algoritmo de diferencias divididas
// x: nodos de interpolación
// a: valores a interpolar
n = length(x);
for j = 2 : n
   for k = n : -1 : j
      a(k) = (a(k) - a(k-1))/(x(k) - x(k-j+1));
   end
end
endfunction
```

```
function [a,p]=difd2(a,x,t)
// Algoritmo de diferencias divididas y evaluación
// t: nodos de evaluación del polinomio
n = length(x);
for j = 2 : n
  for k = n : -1 : j
    a(k) = (a(k) - a(k - 1))/(x(k) - x(k - j + 1));
  end
end
b = zeros(size(a)); m = length(t);
for j = 1 : m
  b(n) = a(n);
  for k = n - 1 : -1 : 1
    b(k) = a(k) + (t(j) - x(k)) * b(k + 1);
  end
  p(j) = b(1);
end
endfunction
```

Implementación y representación



Fórmulas

$$\begin{array}{c|cccc} x & x_0 & x_1 & \cdots & x_N \\ \hline y & y_0 & y_1 & \cdots & y_N \end{array} \quad y_j = f(x_j), \quad j = 0, \ldots, N,$$

$$\begin{aligned} E_N(x) &= f(x) - P_N(x) = \frac{(x - x_0)\cdots(x - x_N)}{(N+1)!} f^{N+1)}(\xi), \\ & min\{x_0, \ldots, x_N, x\} < \xi < max\{x_0, \ldots, x_N, x\} \end{aligned}$$

$$|E_N(x)| = |f(x) - P_N(x)| \leq \frac{|x - x_0|\cdots|x - x_N|}{(N+1)!} \max_{\xi \in [a,b]} f^{N+1)}(\xi)$$

Ejemplo

x	0	$\pi/4$	$\pi/2$	$\pi/6$
y	0	$\sqrt{2}/2$	1	$1/2$

$[a,b]=[0,\pi/2], \quad f(x)=\sin x$

$$E_2(x)=\frac{f'''(\xi)}{3!}p(x), \quad |E_2(x)|\leq\frac{1}{6}|p(x)|, \quad p(x)=x(x-\frac{\pi}{4})(x-\frac{\pi}{2})$$

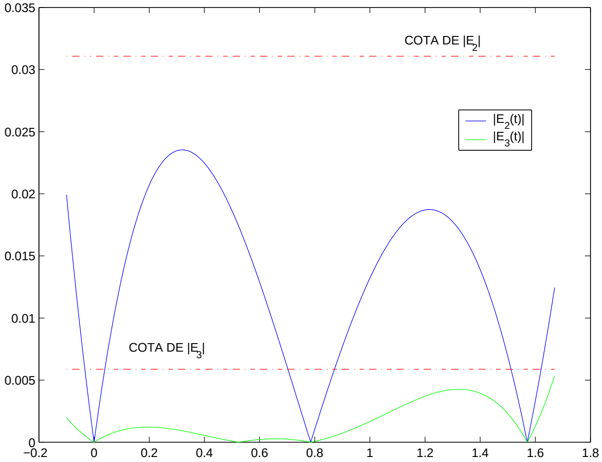
Máximo de $p(x)$ en $[0,\pi/2]$: $x^*=\frac{\pi}{12}(3-\sqrt{3})\Rightarrow p(x^*)=\frac{\pi^3\sqrt{3}}{288}$

$$|E_2(x)|\leq\frac{1}{6}\frac{\pi^3\sqrt{3}}{288}\approx 0{,}03108, \quad x\in[0,\pi/2]$$

$$E_3(x)=\frac{f^{iv)}(\xi)}{3!}x(x-\frac{\pi}{4})(x-\frac{\pi}{2})(x-\frac{\pi}{6})$$

$$|E_3(x)|\leq 5{,}8746\times 10^{-3}, \quad x\in[0,\pi/2]$$

Ejemplo


COTA DE $|E_2|$
COTA DE $|E_3|$
$|E_2(t)|$
$|E_3(t)|$
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
−0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8

Interpolación trigonométrica. Representación

x	x_0	x_1	$\cdots$	x_{N-1}
y	y_0	y_1	$\cdots$	y_{N-1}

$$F_N = \left(\omega^{(k-1)(l-1)}\right)_{k,l=1}^{N} = \left(\exp\left(\frac{-2\pi i(k-1)(l-1)}{N}\right)\right)_{k,l=1}^{N},$$

$$\omega = \omega_N = \exp\left(\frac{-2\pi i}{N}\right)$$

Representación

Sólo frecuencias positivas

$$P_{N-1}(x) = \sum_{k=0}^{N-1} c_k \exp\left(\frac{2\pi i k}{L}(x-a)\right), \quad c = \frac{1}{N} F_N y = \frac{1}{N} fft(y)$$

Centrado (positivas/negativas) $N = 2M$

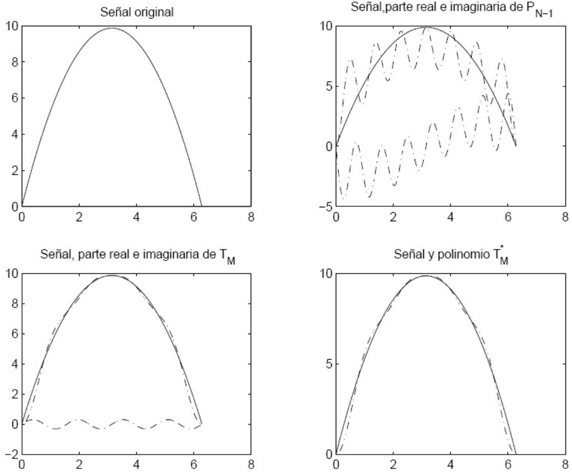
$$T_M(x) = \sum_{k=-M}^{M-1} c_k \exp\left(\frac{2\pi i k}{L}(x-a)\right), \quad c = \frac{1}{N} fftshift(fft(y))$$

Simétrico (real para datos reales)

$$T_M^*(x) = Re(T_M(x))$$

Ejemplo

$$x_j = 2\pi j/8, y_j = f(x_j), j = 0, \ldots, N-1 = 7, f(x) = 2\pi x - x^2$$



Evaluación: adaptación de algoritmo de Horner

$$z_0 = e^{\left(\frac{2\pi i}{L}(x^*-a)\right)}, \quad T_M(x^*) = \sum_{k=-M}^{M-1} c_k z_0^k = z_0^{-M} \sum_{k=-M}^{M-1} c_k z_0^{k+M}$$

```
function [p,c]=evalpoltrig(t,y)
// Algoritmo de evaluación del polinomio trigonométrico
// y: valores a interpolar. t: nodos de evaluación
n = length(y); c = (1/N) * fftshift(fft(y));
b = zeros(size(y)); m = length(t);
for j = 1 : m
   z = exp(2 * pi * i * t(k)); b(n) = c(n);
   for k = n - 1 : -1 : 1
      b(k) = c(k) + z * b(k + 1);
   end
   p(j) = b(1) * (z * *(-N/2));
end
endfunction
```

Convolución periódica

- Considerar la señal finita $\{y_k\}_{k=0}^{N-1}$ como una señal infinita y periódica

$$\ldots, y_{N-3}, y_{N-2}, y_{N-1}, y_0, y_1, \ldots, y_{N-2}, y_{N-1}, y_0, y_1, \ldots$$

- Dadas dos señales

$$\vec{x} = (x_0, \ldots, x_{N-1})^T, \vec{y} = (y_0, \ldots, y_{N-1})^T,$$

se define la convolución $\vec{x} * \vec{y}$ como la señal de N componentes

$$(\vec{x} * \vec{y})_j = \sum_{k=0}^{N-1} x_k y_{j-k} = \sum_{k=0}^{N-1} x_{j-k} y_k, j = 0, \ldots, N-1.$$

En la suma se entiende que se usa la interpretación como señal periódica.

Propiedades

(1) $\vec{x} * \vec{y}$ tiene período N.

(2) $\vec{x} * \vec{y} = \vec{y} * \vec{x}$.

(3) Si X es la matriz

$$X = \begin{pmatrix} x_0 & x_1 & x_2 & \cdots & x_{N-2} & x_{N-1} \\ x_{N-1} & x_0 & x_1 & \cdots & x_{N-3} & x_{N-2} \\ x_{N-2} & x_{N-1} & x_0 & \cdots & x_{N-4} & x_{N-3} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_1 & x_2 & x_3 & \cdots & x_{N-1} & x_0 \end{pmatrix}$$

entonces

$$\vec{x} * \vec{y} = X^T \vec{y}$$

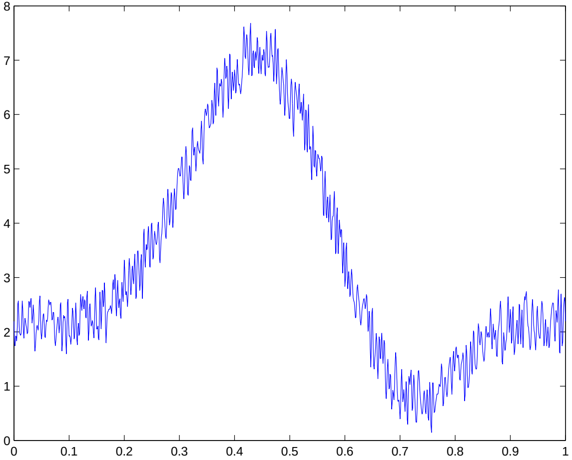
(4) Si $\hat{x} = F_N\vec{x}, \hat{y} = F_N\vec{y}$, entonces

$$\left(F_N(\vec{x} * \vec{y})\right)_j = \hat{c}_j\hat{d}_j.$$

La componente j de la DFT de $\vec{x} * \vec{y}$ es el producto de las componentes j de las DFT de $\vec{x}$ e $\vec{y}$.

Aplicación al filtrado de una señal. Un ejemplo

Señal distorsionada por ruido



Problema: recibida la señal distorsionada, cómo recuperar la información.
Una forma:

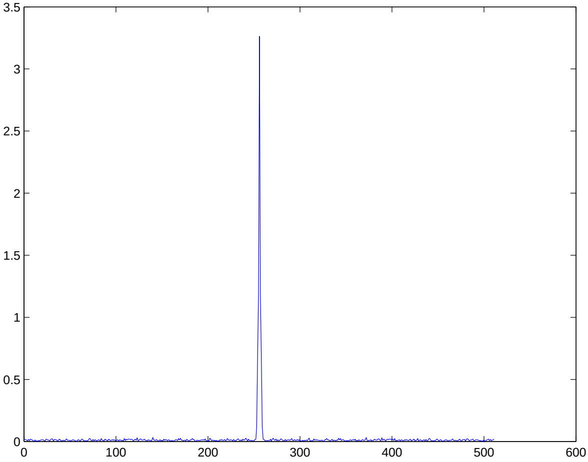
- Analizar frecuencias de señal recibida.
- Filtrar en frecuencia donde convenga.
- Recuperar la señal filtrada.

Paso 1: Representación y análisis de frecuencias

```
function filtrado
// Programa para filtrado de una señal
// Muestra distorsionada en senhal.dat
// almacenada en el vector y
// t: nodos de evaluación del polinomio
load senhal
t = 0 : 0,001 : 1; M = length(t); N = length(y);
[p, c] = evalpoltrig(t, y)
// Gráfico del polinomio y frecuencias
figure(1)
plot(t, real(p))
f = abs(c);
figure(2)
plot(0 : N − 1, f)
```

Paso 1: Representación y análisis de frecuencias

Espectro en frecuencia



Paso 2: filtro en frecuencia

$[w, jj] = max(f)$; $ese = 5$;
for $j = 1 : N$
if $(j < jj - ese)|(j > jj + ese)$
$c(j) = 0$;
end
end

Paso 3. Recuperación de señal filtrada

// Evaluación (se puede usar
// el programa evalpoltrig.m también)
$b = zeros(size(y));$
for $j = 1 : M$
  $z = exp(2 * pi * i * t(k));$
  $b(N) = c(N);$
  for $k = N - 1 : -1 : 1$
    $b(k) = c(k) + z * b(k + 1);$
  end
  $p(j) = b(1) * (z * *(-N/2));$
end
figure(1)
plot(t,real(p))
endfunction

Paso 3. Recuperación de señal filtrada

Señal filtrada