Instituto Jorge Sabato, 25 años.
Comisión Nacional de Energía atómica.

Modelización de Materiales 2018

Interpolación

Mariano Forti - Ruben Weht

ruweht@cnea.gov.ar marianodforti@gmail.com

www.tandar.cnea.gov.ar/~weht/Modelizacion

https://mdforti.github.io/Modelizacion/

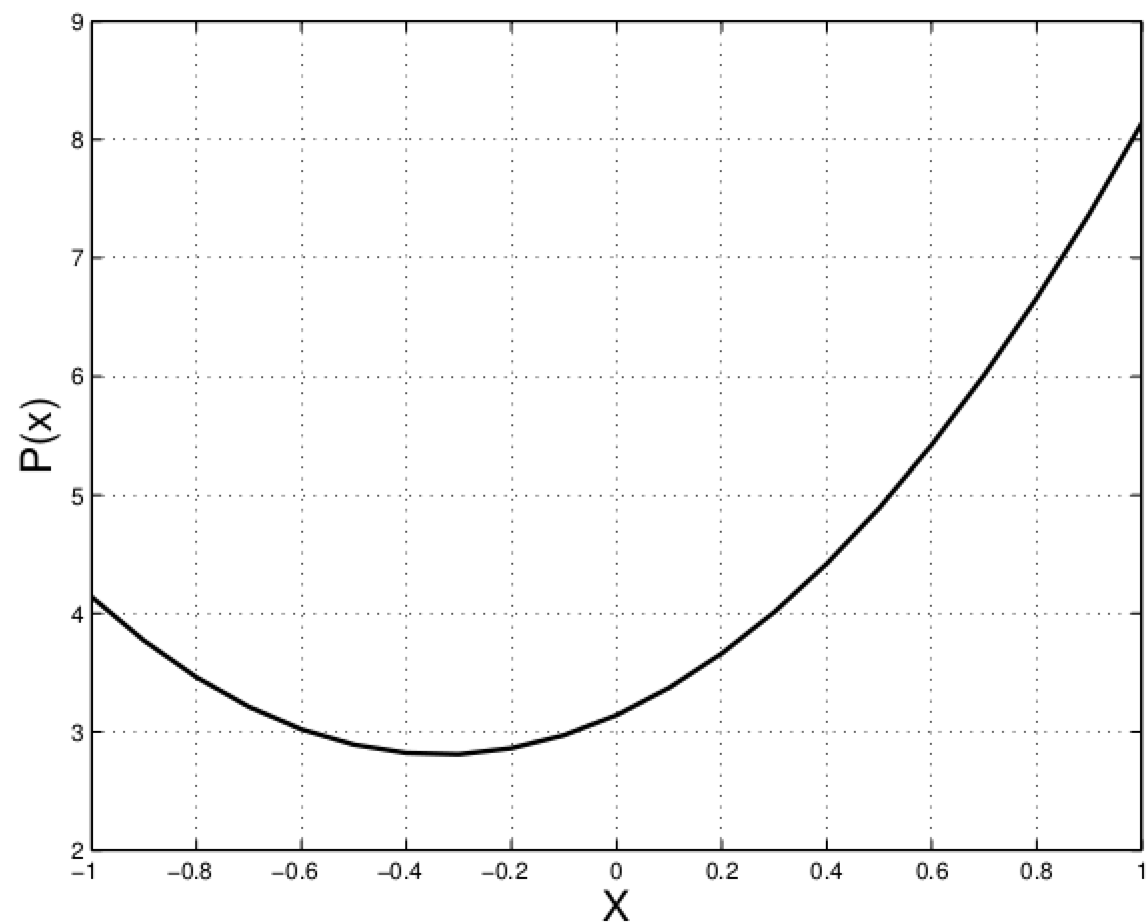
Polinomios en Matlab

Los Polinomios son Vectores

$$P(x) = 3x^2 + 2x + \pi$$

Evaluar polinomios:

```
Command Window
>> P = [ 3 2 pi ];
>> x = [-1:0.1:1];
>> y=polyval(P,x);
>>
```



$$y = P_1 \times x^2 + P_2 \times x + P_3$$

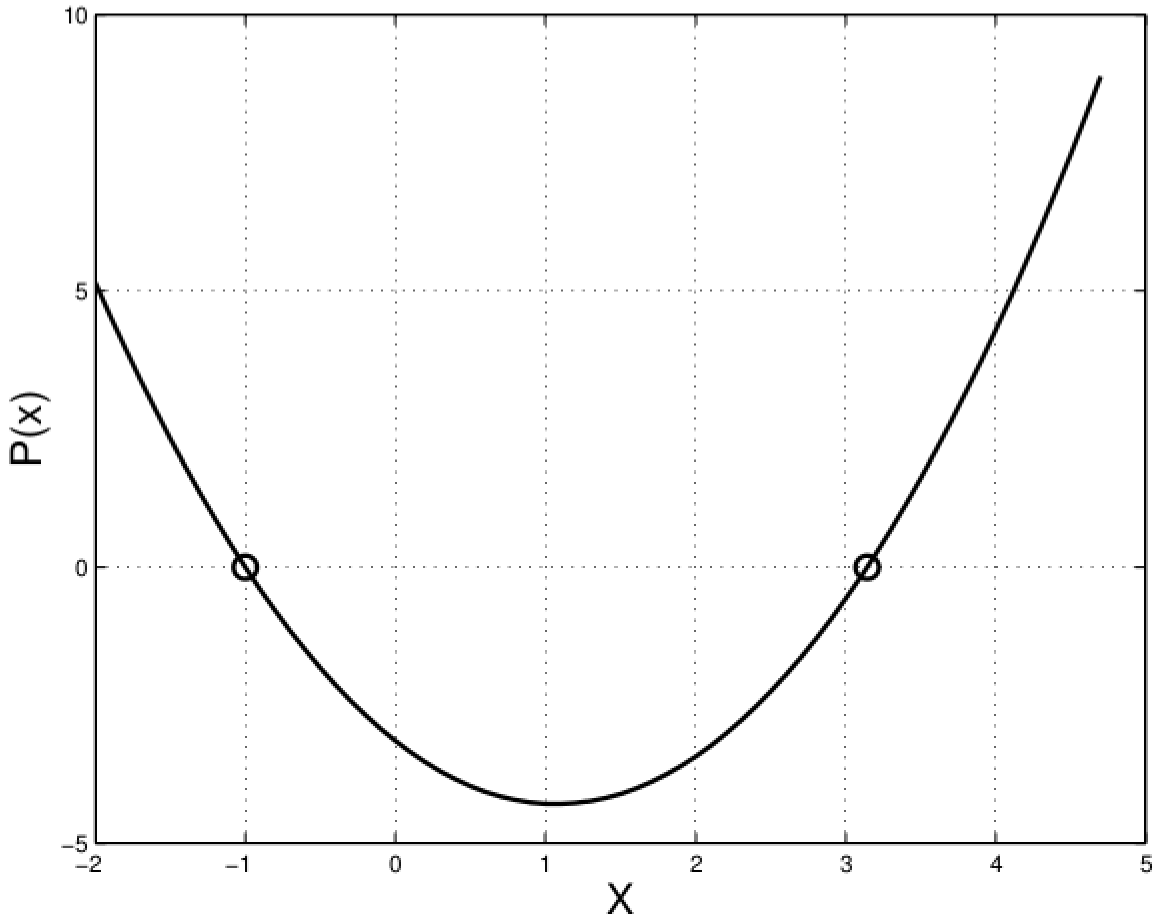
Polinomios en Matlab

Polinomio con raíces dadas

$$P(x) = (x - \pi)(x + 1)$$

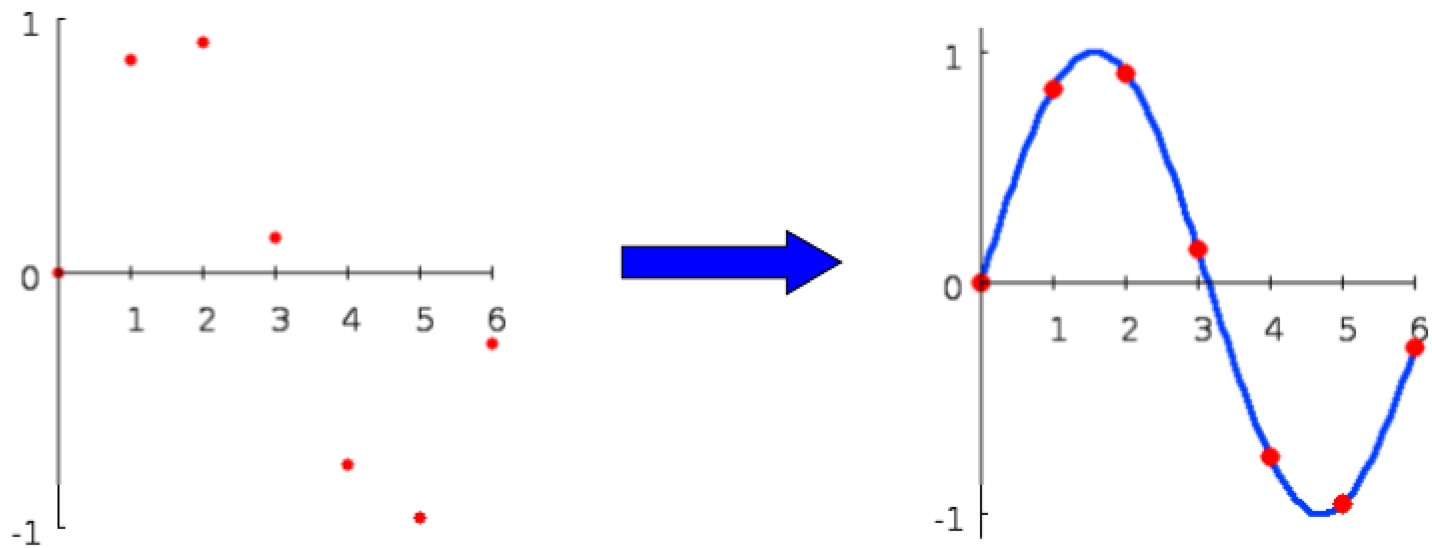
Command Window

```
>> x = [-2:0.1:1.5*pi];
>> xo = [ pi -1 ];
>> P = poly(xo);
>> y = polyval(P,x);
>>
```



Interpolación - Resumen

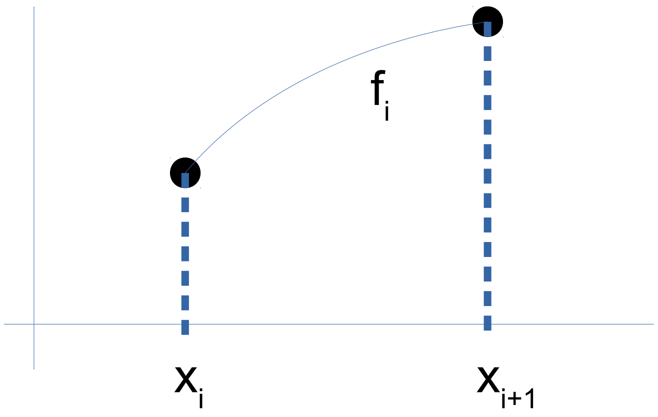
Se busca una curva que incluya a los puntos experimentales



N puntos   N-1 Intervalos

Interpolación - Resumen

Polinomios por tramos



Puntos experimentales:

$$\{(x_1, y_1), \ldots, (x_N, y_N)\}$$

Definición de Intervalos :

$$I_1=[x_1, x_2]; I_2=[x_2, x_3]; \ldots; I_i=[x_i, x_{i+1}]; \ldots; I_{N-1}=[x_{N-1}, x_N]$$

Tamaños de intervalo:

$$h_i = x_{i+1} - x_i$$

Interpolación:

$$f_i(x) = d_i + c_i(x-x_i) + b_i(x-x_i)^2 + a_i(x-x_i)^3 \quad ; \quad x \in [x_i, x_{i+1}]$$

Interpolación – Splines Cúbicos

Polinomio continuo y suave por tramos

En cada intervalo se aproxima por un polinomio de orden 3

$$f_i(x) = d_i + c_i(x - x_i) + b_i(x - x_i)^2 + a_i(x - x_i)^3 \quad ; \quad x \in [x_i, x_{i+1}]$$

4(N-1) incognitas

Que debe pasar por los puntos

$$f_i(x_i) = y_i \quad \Rightarrow d_i = y_i$$ N ecuaciones

Que deben formar una curva continua

$$f_i(x_{i+1}) = f_{i+1}(x_{i+1})$$ N-2 ecuaciones

Que deben formar una curva suave

$$f_i'(x_{i+1}) = f_{i+1}'(x_{i+1})$$ N-2 ecuaciones

La derivada segunda debe ser continua

$$f_i''(x_{i+1}) = f_{i+1}''(x_{i+1})$$ N-2 ecuaciones

4N-6 ecuaciones

Splines Cúbicos: Sistema Lineal

Resolviendo para los b :

$$3\left(\frac{y_{i+1}-y_i}{h_i}-\frac{y_i-y_{i-1}}{h_{i-1}}\right)=b_{i-1}h_{i-1}+2b_i\left(h_i+h_{i-1}\right)+b_{i+1}h_i$$

Es un sistema de N – 2 ecuaciones con N - 1 incognitas b_1 … b_{N-1}

Condiciónes extra: Derivadas segundas en los extremos

$$b_1=0$$

La condición de contorno fija el valor del b_1, reescribiendo la ecuación para i = 1

$$b_N=0$$

Es el coeficiente de un intervalo extra a la derecha del último punto. No nos interesan los otros coeficientes en ese intervalo.

Splines Cúbicos: Sistema Lineal

$$3\left(\frac{y_{i+1}-y_i}{h_i}-\frac{y_i-y_{i-1}}{h_{i-1}}\right)=b_{i-1}h_{i-1}+2b_i\left(h_i+h_{i-1}\right)+b_{i+1}h_i$$

Expandiendo cada ecuación en función de todas las b_i

i=1 : $$b_1=0 \;\Rightarrow\; 1\cdot b_1+0\cdot b_2+0\cdot b_3+\cdots=0$$

i=2 : $$b_1 h_1+2\left(h_2+h_1\right)b_2+b_3 h_2+0\cdot b_4+0\cdot b_5+\cdots=3\left(\frac{y_3-y_2}{h_2}-\frac{y_3-y_2}{h_2}\right)$$

i : $$0\cdot b_1+0\cdot b_2+\cdots+b_{i-1}h_{i-1}+2b_i\left(h_i+h_{i-1}\right)+b_{i+1}h_i+0\cdot b_{i+2}+\cdots+0\cdot b_N=3\left(\frac{y_{i+1}-y_i}{h_i}-\frac{y_i-y_{i-1}}{h_{i-1}}\right)$$

i=N : $$b_N=0 \;\Rightarrow\; \cdots+0\cdot b_{N-2}+0\cdot b_{N-1}+1\cdot b_N=0$$

Splines Cúbicos: Sistema Lineal

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ h_1 & 2(h_2+h_1) & h_2 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & h_2 & 2(h_3+h_2) & h_3 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & h_{N-3} & 2(h_{N-2}+h_{N-3}) & h_{N-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & h_{N-2} & 2(h_{N-1}+h_{N-2}) & h_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{N-2} \\ b_{N-1} \\ b_N \end{pmatrix} =$$

$$3\begin{pmatrix} 0 \\ \dfrac{y_3-y_2}{h_2}-\dfrac{y_2-y_1}{h_1} \\ \vdots \\ \dfrac{y_N-y_{N-1}}{h_{N-1}}-\dfrac{y_{N-1}-y_{N-2}}{h_{N-2}} \\ 0 \end{pmatrix}$$

Splines cúbicos: Resultado Final

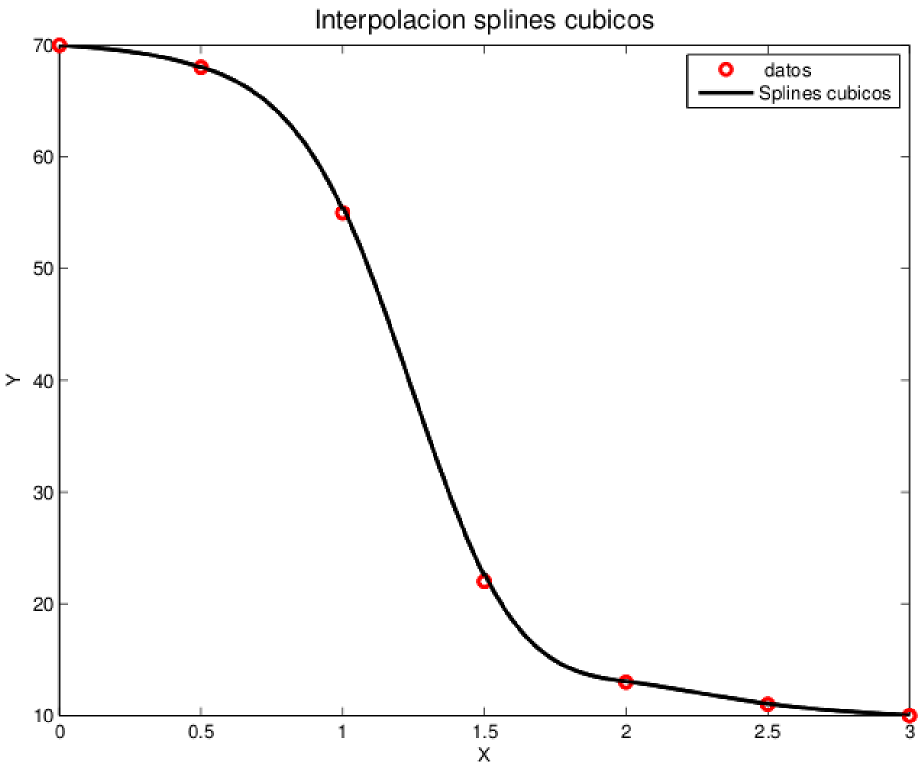
Resolución de todos los coeficientes.

$$f_i(x) = d_i + c_i(x - x_i) + b_i(x - x_i)^2 + a_i(x - x_i)^3 \quad ; \quad x \in [x_i, x_{i+1}]$$

$$d_i = y_i$$

$$c_i = \frac{(y_{i+1} - y_i)}{h_i} - b_i h_i - a_i h_i^2$$

$$a_i = \frac{1}{3} \frac{(b_{i+1} - b_i)}{h_i}$$



Splines Cúbicos: Polinomios por tramos

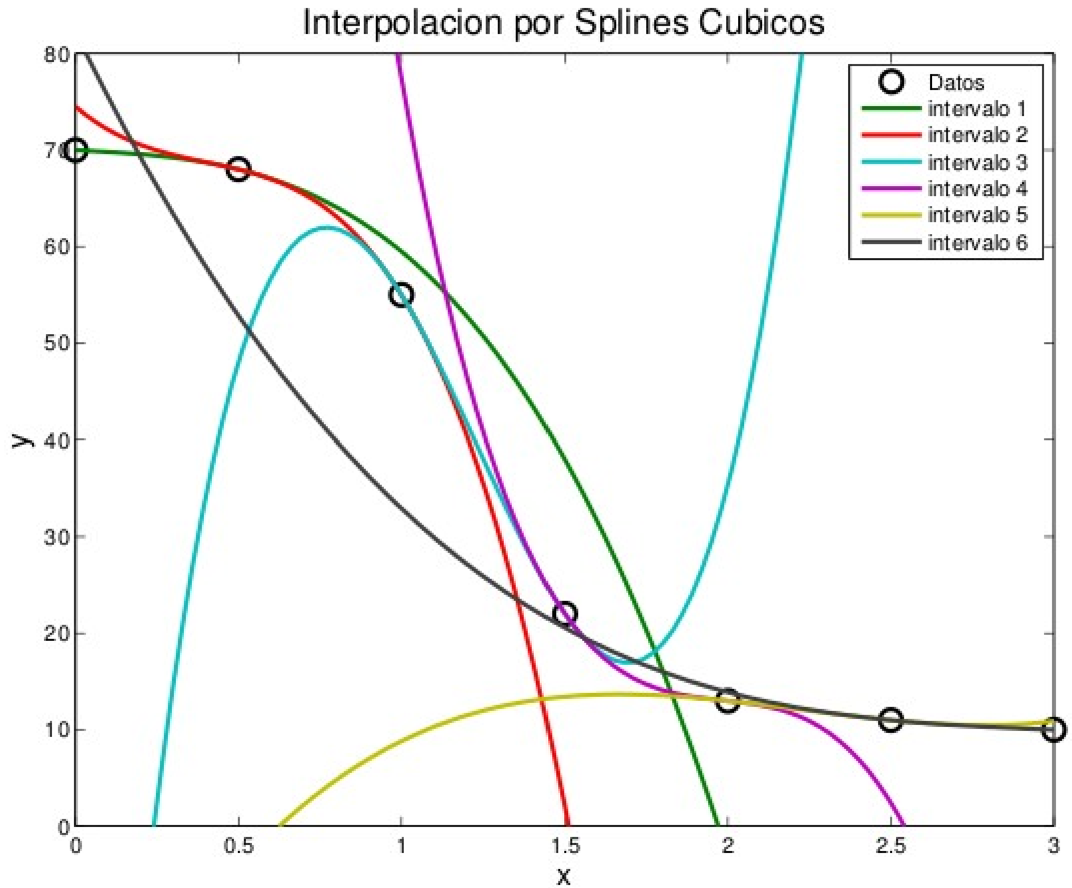
Los polinomios solo valen en los tramos en que se definen

$$f_i(x) = d_i + c_i(x - x_i) + b_i(x - x_i)^2 + a_i(x - x_i)^3 \quad ; \quad x \in [x_i, x_{i+1}]$$

$$d_i = y_i$$

$$c_i = \frac{(y_{i+1} - y_i)}{h_i} - b_i h_i - a_i h_i^2$$

$$a_i = \frac{1}{3} \frac{(b_{i+1} - b_i)}{h_i}$$



Ejercicio 1: Interpolación

Derivación directa de los splines

$$xo \in [x_i, x_{i+1}] \Rightarrow F(x_i) \cdot F(x_{i+1}) \leq 0$$

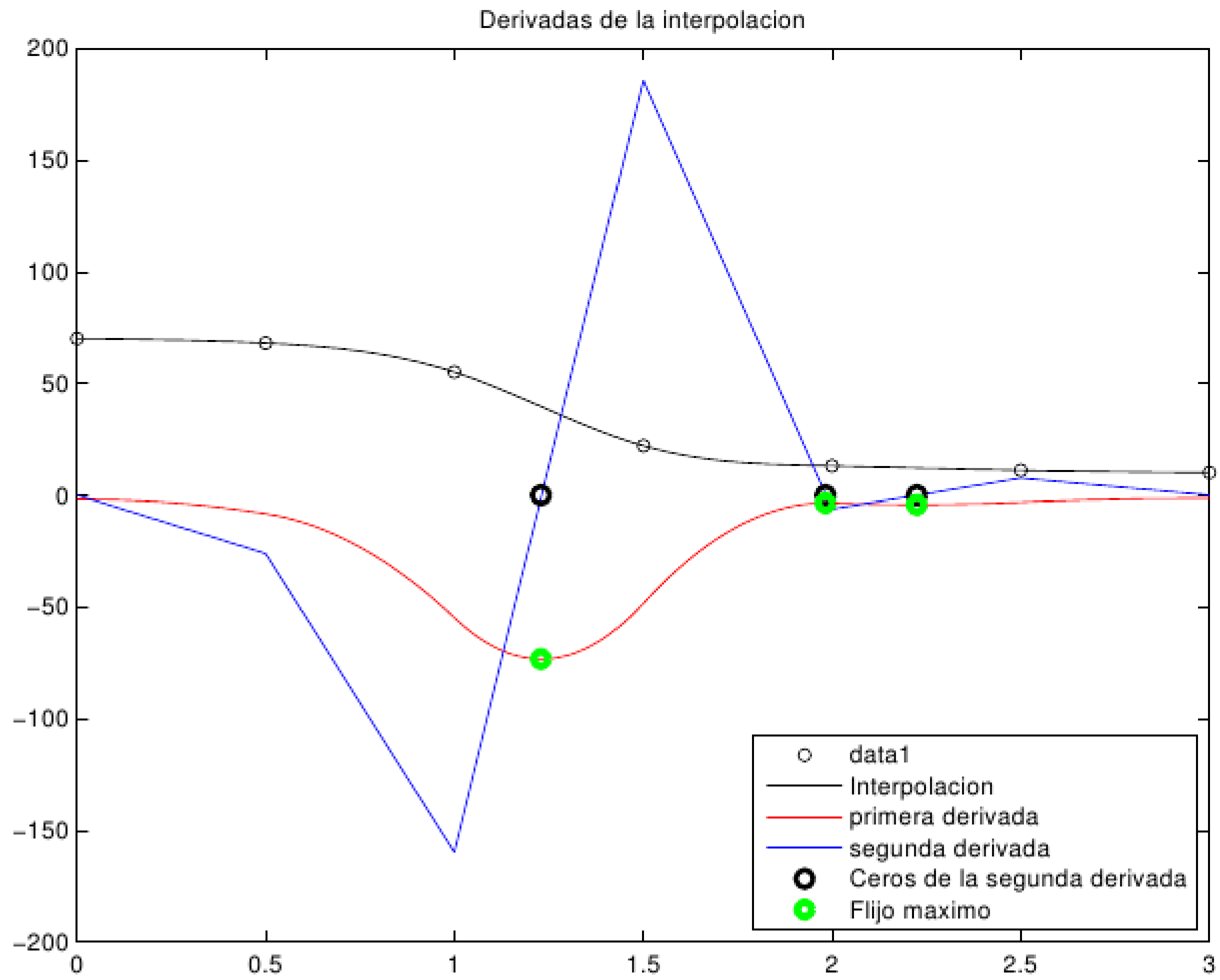
Este gráfico está bien para ilustrar el cumplimiento de las condiciones impuestas a los polinomios, pero la deri vada de los polinomios no necesariamente es una buena aproximación a las derivadas de la función. Por ejemplo, Observe que esa derivada segun da tiene varios ceros aunque la función tenga un único punto de inflección.

Ceros malos:

```
>> zomx

zomx =

    1.2298
    1.9823
    2.2247

>>
```

Flujo

```
>> DYom

DYom =

  -73.3144   -3.5336   -4.2959
```

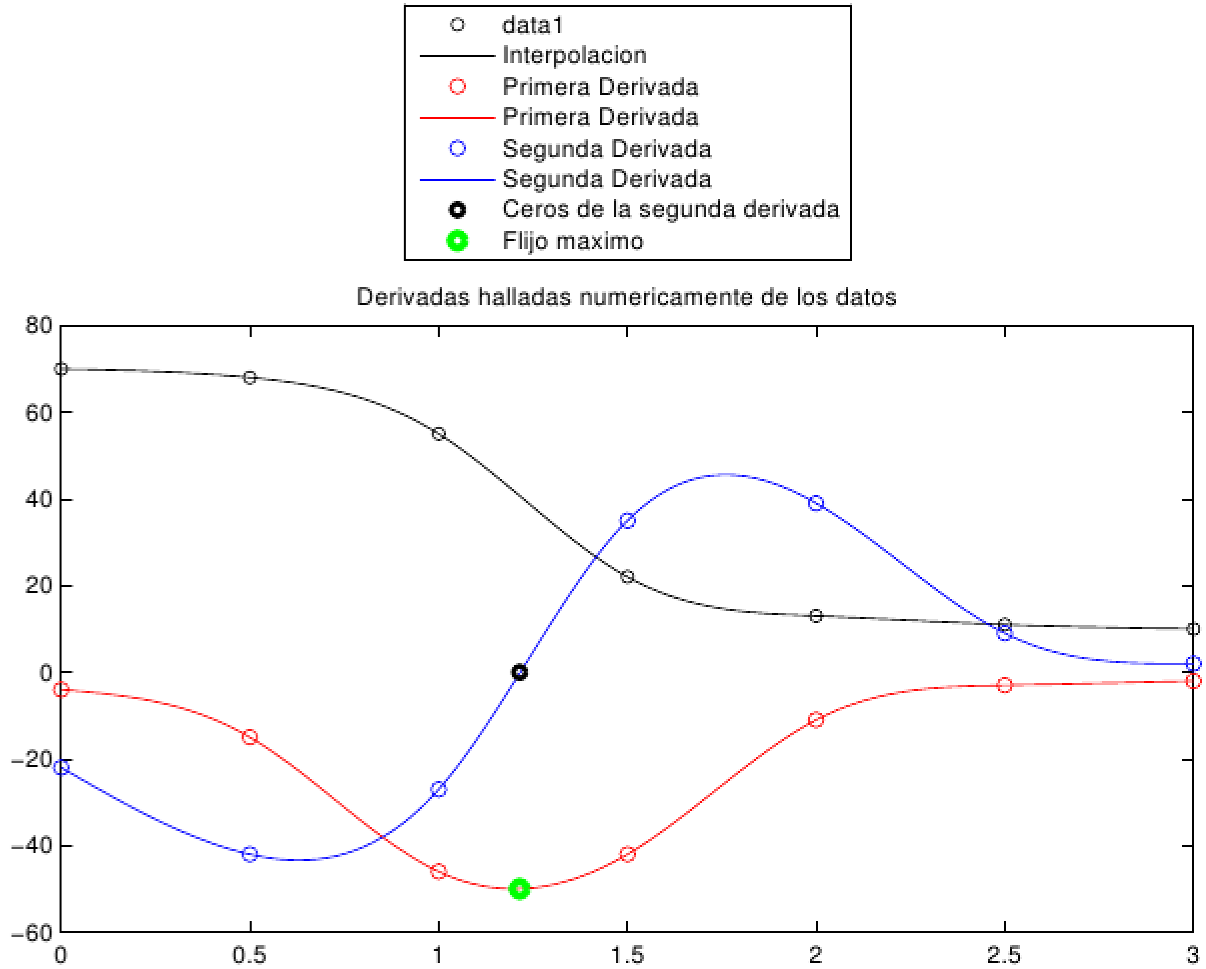


Ejercicio 1: Interpolación

Otra solución Posible

$$f'(x_i)=\frac{f(x_{i+1})-f(x_{i-1})}{x_{i+1}-x_{i-1}}+interpolación$$

```
zox =

    1.2146

DYo =

  -49.9587
```



Polinomios de Lagrange

$$L(x) = \sum_{k=1}^{N} Y_k L_k$$

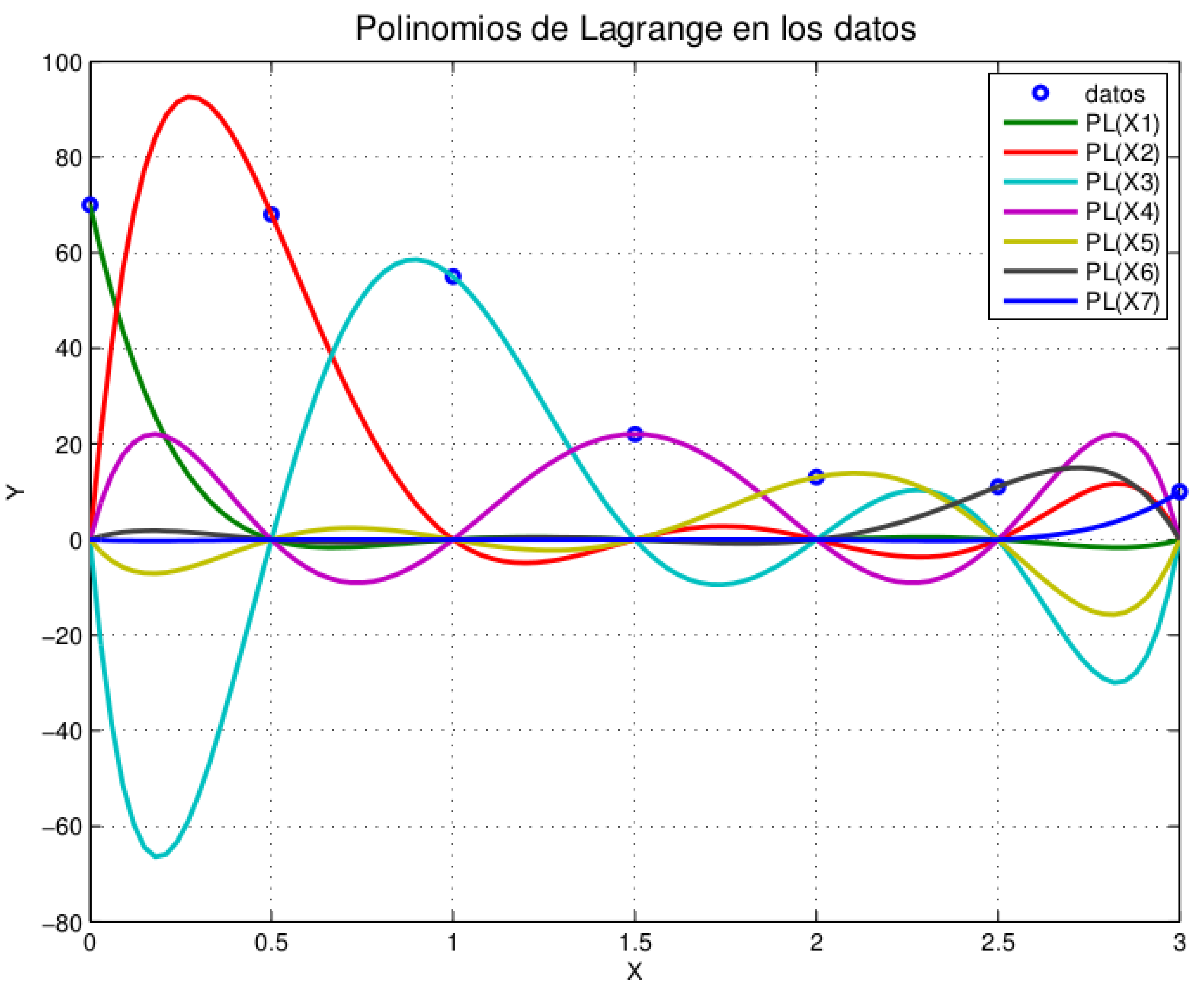
$$L_k = \prod_{i \neq k} \frac{(x - x_i)}{(x_k - x_i)}$$

Editor - /home/mariano/Documents/modelizacion/2012/P_02_Interpolacion/PL.m

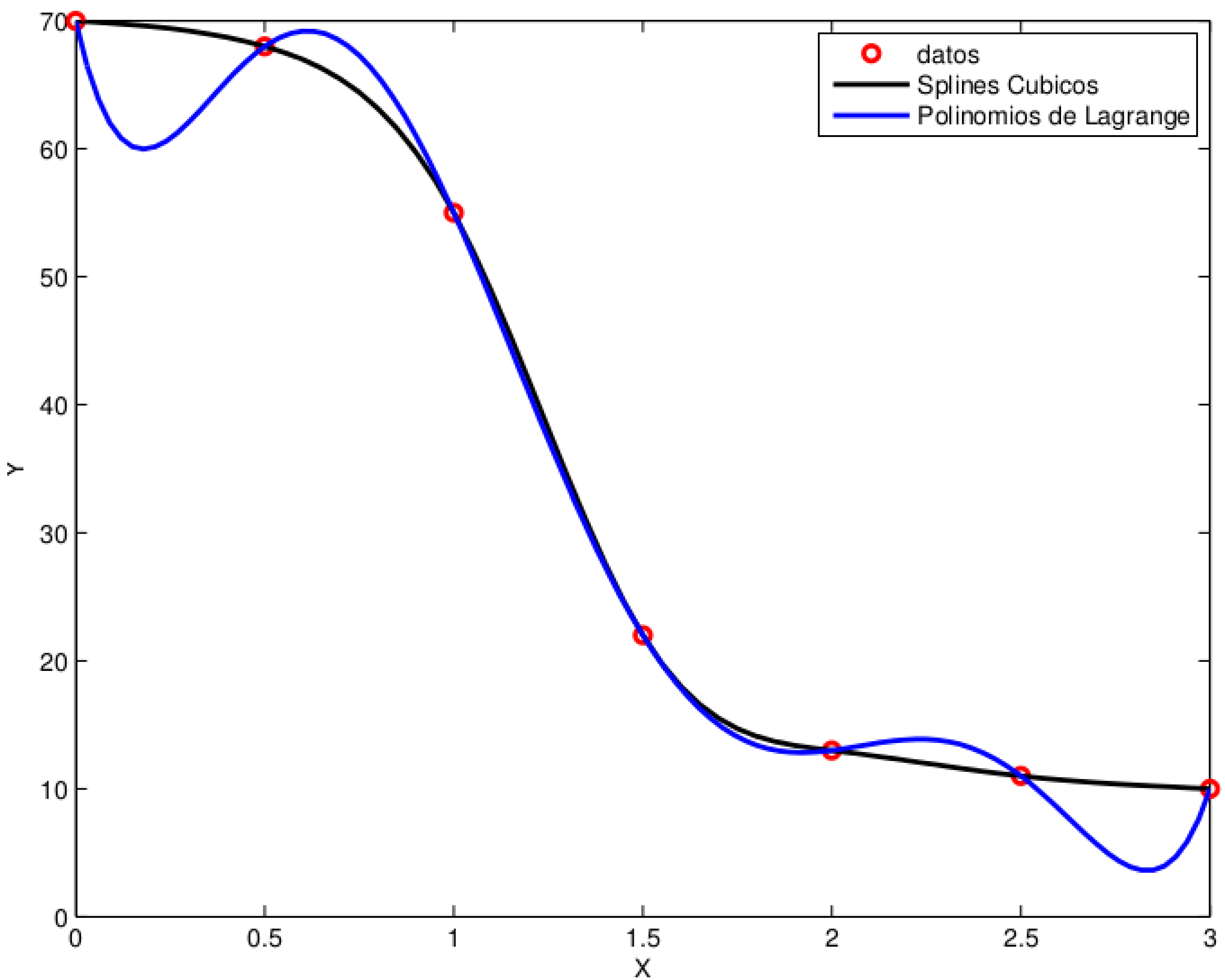
```
function [L,y]=PL(X,Y)

N = length(X);

for k=1:N
    l=1;
    for j=1:N
        if j~=k
            l=conv(l,poly(X(j))/(X(k)-X(j)));
        end
    end
    L(k,:)=Y(k)*l;
end

xmin=X(1); xmax = X(end);
x = [xmin:(xmax - xmin)/100:xmax];
y = zeros(1,length(x));

for i = 1:N
    y=y+polyval(L(i,:),x);
end

```

PLag.m | basic_plinomial.m | PL.m

Polinomios de Lagange



Polinomios de Lagrange



FIN – Consultas – Ejercicio