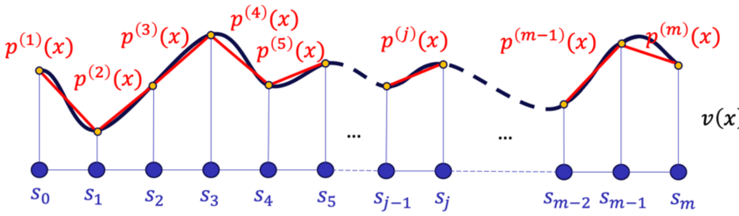
INTERPOLACIÓN POLINÓMICA POR TRAMOS

Grado 1 :



$$v(x) = \begin{cases} p^{(1)}(x) = a_1 + b_1\,x \quad \text{si } x \in [s_0, s_1] \\ p^{(2)}(x) = a_2 + b_2\,x \quad \text{si } x \in \,]s_1, s_2] \\ \quad \cdots \\ p^{(j)}(x) = a_j + b_j\,x \quad \text{si } x \in \,]s_{j-1}, s_j] \\ \quad \cdots \\ p^{(m)}(x) = a_m + b_m\,x \quad \text{si } x \in \,]s_{m-1}, s_m] \end{cases}$$

buscamos crear una función, v(x) creada por tramos p(x), funciones que utilizan 2 puntos de soporte para interpolar utilizando lineas rectas en cada intervalo Ij ([sj, s(j+1)]).
Esta función luego nos permitirá interpolar cualquier valor de x dentro de nuestro intervalo.

Resolviendo m sistemas

La interpolación por tramos de m sistemas de ecuaciones consiste en definir una función v(x) continua que aproxime una función f(x), mediante polinomios de grado 1 en cada intervalo [sj−1,sj]. Esto asegura continuidad y simplicidad en el cálculo.

A la hora de programar en R dividiremos las muestras en intervalos que contengan dos muestras y para cada tramo calcularemos un polinomio.

Ya que son polinomios de primer grado solo tendremos que calcular dos coeficientes , los cuales podemos hallar con las ecuaciones:

$$\begin{cases} p^{(j)}(s_{j-1}) = f(s_{j-1}) = a_j + b_j s_{j-1} \\ p^{(j)}(s_j) = f(s_j) = a_j + b_j s_j \end{cases} \Rightarrow \begin{cases} a_j = \dfrac{s_j \cdot f(s_{j-1}) - s_{j-1} \cdot f(s_j)}{s_j - s_{j-1}} \\ b_j = \dfrac{f(s_j) - f(s_{j-1})}{s_j - s_{j-1}} \end{cases}$$

Finalmente podiendo rellenar nuestra funcion v(x) con los pj(x) calulado para cada inter valo Ij.

Bases de newton

$$p^{(j)}(x) = f[s_{j-1}] + f[s_{j-1}, s_j](x - s_{j-1}) = f(s_{j-1}) + \frac{f(s_j) - f(s_{j-1})}{s_j - s_{j-1}}(x - s_{j-1})$$

$$v(x) = \begin{cases} p^{(1)}(x) = f[s_0] + f[s_0, s_1](x - s_0) \quad \text{si } x \in [s_0, s_1] \\ p^{(2)}(x) = f[s_1] + f[s_1, s_2](x - s_1) \quad \text{si } x \in \,]s_1, s_2] \\ \quad \cdots \\ p^{(j)}(x) = f[s_{j-1}] + f[s_{j-1}, s_j](x - s_{j-1})\text{si } x \in \,]s_{j-1}, s_j] \\ \quad \cdots \\ p^{(m)}(x) = f[s_{m-1}] + f[s_{m-1}, s_n](x - s_{n-1})\text{si } x \in \,]s_{m-1}, s_m] \end{cases}$$

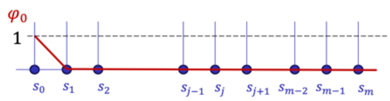
El metodo de newton, tiene el mismo fundamento que el de resolviendo m sistemas, siendo necesario apenas calcular pj(x) con la ecuación adecuada en cada intervalo Ij de 2 soportes, y rellenar la funcion v(x).

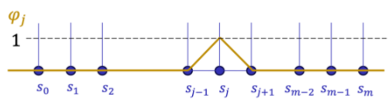
Bases de Lagrange $\mathcal{L}1$ (Δ)

$$v(x) = \sum_{j=0}^{m} f(s_j)\,\varphi_j(x)$$

La idea fundamental es crear nuestra función v(x) de esta manera.
Con lo cual es necesario calcular los coeficientes $\varphi j(x)$.

Mediante una inteprolación de grado 2 existen tres formas de caluclar φ:
$\varphi 0$ para el soporte inicial, φj para los soportes intermedios, y φn para el soporte final.





$$\varphi_0 = \begin{cases} \dfrac{x - s_1}{s_0 - s_1} & \text{si } x \in [s_0, s_1] \\ 0 & \text{en el resto} \end{cases} \qquad \varphi_j = \begin{cases} \dfrac{x - s_{j-1}}{s_j - s_{j-1}} & \text{si } x \in [s_{j-1}, s_j] \\ \dfrac{x - s_{j+1}}{s_j - s_{j+1}} & \text{si } x \in [s_j, s_{j+1}] \\ 0 & \text{en el resto} \end{cases} \qquad \varphi_m = \begin{cases} \dfrac{x - s_{m-1}}{s_m - s_{m-1}} & \text{si } x \in [s_{m-1}, s_m] \\ 0 & \text{en el resto} \end{cases}$$

Con nuestros coeficientes φ, a la hora de crear la funcion v(x) es importante recordar que al ser un sumatorio es necesario sumar f(sj) φj(x) si pertenecen al mismo intervalo de x. Aqui un ejemplo para que quede claro:

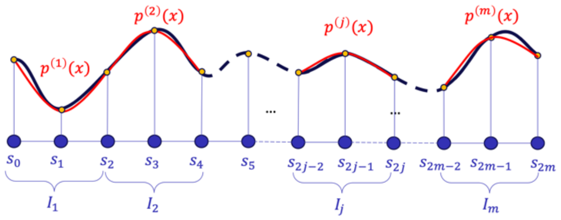
sj = 0, 2, 5
f(sj) = 3, 6, 10.5
intervalo: [0,2] [2,5]

$$v(x) = \sum_{j=0}^{1} f(s_j)\,\varphi_j(x) = 3\begin{cases} 1 - \dfrac{1}{2}x & \text{si } x \in [0,2] \\ 0 & \text{en el resto} \end{cases} + 6\begin{cases} \dfrac{1}{2}x & \text{si } x \in [0,2] \\ \dfrac{5}{3} - \dfrac{1}{3}x & \text{si } x \in [2,5] \\ 0 & \text{en el resto} \end{cases}$$

Es necesario sumar los trozos con el mismo intervalo:

$$v(x) = \begin{cases} 3(1 - 0.5x) + 6(0.5x) & [0,2] \\ 6\left(\dfrac{5}{3} - \dfrac{1}{3}x\right) & (2,5] \end{cases}$$

Grado 2:



$$v(x) = \begin{cases} p^{(1)}(x) = a_1 + b_1\,x + c_1x^2 \quad \text{si } x \in [s_0, s_2] & \text{.....Tiene que pasar por } f(s_0), f(s_1) \text{ y } f(s_2) \\ p^{(2)}(x) = a_2 + b_2\,x + c_2x^2 \quad \text{si } x \in\,]s_2, s_4] & \text{.....Tiene que pasar por } f(s_2), f(s_3) \text{ y } f(s_4) \\ \quad\cdots & \\ p^{(j)}(x) = a_j + b_j\,x + c_jx^2 \quad \text{si } x \in\,]s_{2j-2}, s_{2j}] & \text{.....Tiene que pasar por } f(s_{2j-2}), f(s_{2j-1}) \text{ y } f(s_{2j}) \\ \quad\cdots & \\ p^{(m)}(x) = a_m + b_m\,x + c_mx^2 \quad \text{si } x \in\,]s_{2m-2}, s_{2m}] & \text{.....Tiene que pasar por } f(s_{2m-2}), f(s_{2m-1}) \text{ y } f(s_{2m}) \end{cases}$$

En Interpolación de grado 2 se utilizan intervalos I_j de 3 soportes con el fin de crear una funcion v(x) a trozos, constituida por funciones pj(x) de grade 2.

Resolviendo m sistemas

La interpolación por tramos con polinomios de grado 2 utiliza un polinomio cuadrático para aproximar los valores de una función en intervalos definidos por puntos de soporte dados. Esta técnica es útil cuando se desea obtener una aproximación más precisa que la interpolación lineal.

$$\begin{cases} f(s_{2j-2}) = p^{(j)}(s_{2j-2}) = a_j + b_j\, s_{2j-2} + c_j(s_{2j-2})^2 \\ f(s_{2j-1}) = p^{(j)}(s_{2j-1}) = a_j + b_j\, s_{2j-1} + c_j(s_{2j-1})^2 \\ \quad f(s_{2j}) = p^{(j)}(s_{2j}) = a_j + b_j\, s_{2j} + c_j(s_{2j})^2 \end{cases}$$

Para cada intervalo, se calcula pj(x) resolviendo para **a**, **b** y **c** en este sistema de ecuaciones simultaneas

Newton

$$p^{(j)}(x) = f[s_{2j-2}] + f[s_{2j-2}, s_{2j-1}](x - s_{2j-2}) + f[s_{2j-2}, s_{2j-1}, s_{2j}](x - s_{2j-2})(x - s_{2j-1})$$

Para cada intervalo I_j calculas su pj(x) con el uso de tres soportes y rellenas la función v(x).

Bases de Lagrange $\mathcal{L}_2$ (Δ)

$$v(x) = \sum_{j=0}^{2m} f(s_j)\,\varphi_j(x)$$

Nuestra meta es poder expresar la función v(x) de esta manera, creando una `base´ de coeficientes $\boldsymbol{\varphi}$**j**.

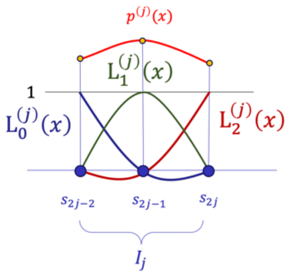
$$\varphi_0(x) = \begin{cases} \mathrm{L}_0^{(1)}(x) \text{ si } x \in [s_0, s_2] \\ 0 \quad \text{en el resto} \end{cases} \qquad \varphi_{2m}(x) = \begin{cases} \mathrm{L}_2^{(m)}(x) \text{si } x \in]s_{2m-2}, s_{2m}] \\ 0 \quad \text{en el resto} \end{cases}$$

Índice PAR

$$\varphi_{2j}(x) = \begin{cases} \mathrm{L}_2^{(j)}(x) & \text{si } x \in]s_{2j-2}, s_{2j}] \\ \mathrm{L}_0^{(j+1)}(x) & \text{si } x \in]s_{2j}, s_{2j+2}] \\ 0 & \text{en el resto} \end{cases}$$

Índice IMPAR

$$\varphi_{2j-1}(x) = \begin{cases} \mathrm{L}_1^{(j)}(x) & \text{si } x \in]s_{2j-2}, s_{2j}] \\ 0 & \text{en el resto} \end{cases}$$



Al estar utilizando Intervalos I_j de 3 muestras ($\mathcal{L}_2\Delta$), esto significa que los Indices par como S0 y S2 se calculen con dos particiones, mientras que muestras impar como S1 se calculen con una particion.
Para este metodo es fundamental entender la indexación de los polinomios de Lagrange Ljn. En un particion s0,s1,s2 le corresponde a cada su L0,L1,L2, ahora, para el proximo intervalo s3,s4,s5, tambien expresado como $s^1 0$,$s^1 1$,$s^1 2$ siendo los tres primeros soportes del intervalo I^1, correspondiendoles $L^1 0$, $L^1 1$, $L^1 2$.

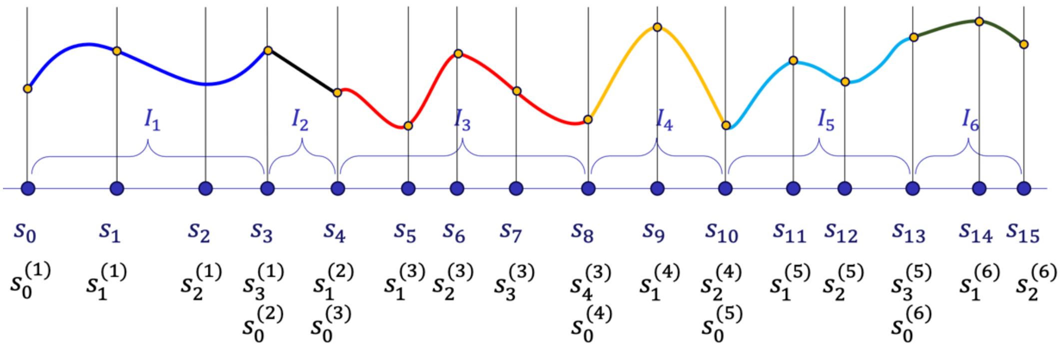
Una forma facil de calcular los polinomios de Lagrange:

$$\mathrm{L}_0^{(j)}(x) = \frac{(x - s_{2j-1})(x - s_{2j})}{(s_{2j-2} - s_{2j-1})(s_{2j-2} - s_{2j})}$$

$$\mathrm{L}_1^{(j)}(x) = \frac{(x - s_{2j-2})(x - s_{2j})}{(s_{2j-1} - s_{2j-2})(s_{2j-1} - s_{2j})}$$

$$\mathrm{L}_2^{(j)}(x) = \frac{(x - s_{2j-2})(x - s_{2j-1})}{(s_{2j} - s_{2j-2})(s_{2j} - s_{2j-1})}$$

Caso general:



Con el ultimo metodo, y el más exacto, se utilizan diferentes particiones para diferentes partes de la ecuación para dar un mejor aproximación, mientras usando 4 o más soportes.

<u>Solución por sistemas de ecuaciones</u>:

$$v^{(j)}(x) = a_0^{(j)} + a_1^{(j)}x + a_2^{(j)}x^2 + \cdots + a_{n_j}^{(j)}x^{n_j}$$

$$v^{(j)}\left(s_k^{(j)}\right) = f\left(s_k^{(j)}\right)$$

$n_j + 1$ incógnitas

$n_j + 1$ ecuaciones

$$\begin{bmatrix} 1 & s_0^{(j)} & \left(s_0^{(j)}\right)^2 & \dots & \left(s_0^{(j)}\right)^{n_j} \\ 1 & s_1^{(j)} & \left(s_1^{(j)}\right)^2 & \dots & \left(s_1^{(j)}\right)^{n_j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & s_{n_j}^{(j)} & \left(s_{n_j}^{(j)}\right)^2 & \dots & \left(s_{n_j}^{(j)}\right)^{n_j} \end{bmatrix} \begin{bmatrix} a_0^{(j)} \\ a_1^{(j)} \\ \vdots \\ a_{n_j}^{(j)} \end{bmatrix} = \begin{bmatrix} f\left(s_0^{(j)}\right) \\ f\left(s_1^{(j)}\right) \\ \vdots \\ f\left(s_{n_j}^{(j)}\right) \end{bmatrix}$$

Rellenando cada intervalo vj(x) resolviendo por los nj + 1 coeficientes a.

<u>Newton:</u>

j indexa el intervalo

$$v^{(j)}(x) = f\left[s_0^{(j)}\right] + \sum_{k=1}^{n_j} f\left[s_0^{(j)}, s_1^{(j)}, \dots, s_k^{(j)}\right] \prod_{i=0}^{k-1}\left(x - s_i^{(j)}\right)$$

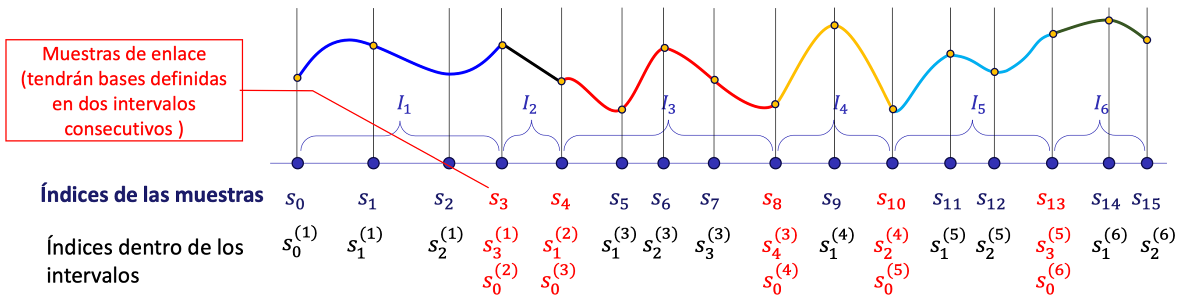
k indexa las muestras del intervalo

	$k=0$	$k=1$	$k=2$	$k=3$	
$s_0^{(j)}$	$f\left[s_0^{(j)}\right]$	$f\left[s_0^{(j)}, s_1^{(j)}\right]$	$f\left[s_0^{(j)}, s_1^{(j)}, s_2^{(j)}\right]$	$f\left[s_0^{(j)}, s_1^{(j)}, s_2^{(j)}, s_3^{(j)}\right]$	$i=0$
$s_1^{(j)}$	$f\left[s_1^{(j)}\right]$	$f\left[s_1^{(j)}, s_2^{(j)}\right]$	$f\left[s_1^{(j)}, s_2^{(j)}, s_3^{(j)}\right]$	0	$i=1$
$s_2^{(j)}$	$f\left[s_2^{(j)}\right]$	$f\left[s_2^{(j)}, s_3^{(j)}\right]$	0	0	$i=2$
$s_3^{(j)}$	$f\left[s_3^{(j)}\right]$	0	0	0	$i=3$

Simplemente calculando la tabla de diferencias divididas, y usando sus valores para rellenar la función vj(x).

Lagrange, $\mathcal{L}(\Delta)$

$$v(x) = \sum_{l=0}^{n} f(s_l)\,\varphi_l(x)$$



Teniendo claro que entre los intervalos existen muestras de enlace, podemos calcular los coeficientes φ:

muestra inicial: $s_0^{(1)}$

$$\varphi_0 = \varphi_0^{(1)} = \begin{cases} \mathrm{L}_0^{(1)}(x) & \text{si } s_0^{(1)} \le x \le s_{n_1}^{(1)} \\ 0 & \text{en el resto} \end{cases}$$

muestra de enlace: $s_{n_j}^{(j)} = s_0^{(j+1)}$

$$\varphi_l = \varphi_k^{(j)} = \varphi_0^{(j+1)} = \begin{cases} \mathrm{L}_{n_j}^{(j)}(x) & \text{si } s_0^{(j)} \le x \le s_{n_j}^{(j)} \\ \mathrm{L}_0^{(j+1)}(x) & \text{si } s_0^{(j+1)} \le x \le s_{n_j}^{(j+1)} \\ 0 & \text{en el resto} \end{cases}$$

muestra dentro del intervalo j

(no es de enlace: $s_k^{(j)}$ con $k \neq n_j$ y $k \neq 0$)

$$\varphi_l = \varphi_k^{(j)} = \begin{cases} \mathrm{L}_k^{(j)}(x) & \text{si } s_0^{(j)} \le x \le s_{n_j}^{(j)} \\ 0 & \text{en el resto} \end{cases}$$

muestra final: $s_n = s_{n_m}^{(m)}$

$$\varphi_n = \varphi_{n_m}^{(m)} = \begin{cases} \mathrm{L}_{n_m}^{(m)}(x) & \text{si } s_0^{(m)} \le x \le s_{n_m}^{(m)} \\ 0 & \text{en el resto} \end{cases}$$

$$\mathrm{L}_k^{(j)}(x) = \prod_{\substack{0 \le i \le n_j \\ i \neq k}} \frac{x - s_i^{(j)}}{s_k^{(j)} - s_i^{(j)}}$$