Informe III:
Trabajo Práctico 5

Edgardo Cipolatti
edgardocipolatti@hotmail.com

I. Ejercicio 7

(a) Modifique la función desarrollada en el ejercicio 6.(b) de manera que permita computar los coeficientes del trazador cúbico sujeto. Se debe prever el ingreso de los valores de la derivada de la función en los extremos del intervalo de interpolación.

(b) Encuentre el trazador cubico sujeto S(x) definido en el intervalo $[1, 3]$ tal que

$$S(x) = \begin{cases} S_0(x) = a_0 + b_0(x-1) + c_0(x-1)^2 + d_0(x-1)^3, \text{ si } 1 \leq x \leq 2 \\ S_1(x) = a_1 + b_1(x-2) + c_1(x-2)^2 + d_1(x-2)^3, \text{ si } 2 \leq x \leq 3 \end{cases}$$

siendo $f(1) = 0$, $f(2) = 4$, $f(3) = \frac{22}{3}$, y asumiendo que $f^{'}(1) = f^{'}(3) = 3$. Utilice la función `[a,b,c,d] = cubic_spline_clamped(x,f,df)` desarrollada en el inciso anterior.

(c) Se quiere determinar la trayectoria plana seguida por un brazo robot industrial (idealizado por un punto material) durante un ciclo de trabajo. El brazo robot debe satisfacer las siguientes restricciones: se debe encontrar en reposo en el punto $(0, 0)$ en el instante inicial. Luego de 1s se debe encontrar en el punto $(1, 2)$, 1s después debe alcanzar el punto $(4, 4)$ y detenerse allí, para recomenzar inmediatamente su movimiento y alcanzar, luego de otro segundo mas el punto $(3, 1)$ para finalmente retornar al origen luego de otro segundo más, donde quedará detenido para repetir el ciclo de trabajo. Encuentre el trazador cúbico sujeto correspondiente utilizando el código desarrollado en el primer inciso y luego grafique en el plano la trayectoria encontrada.

A. *Ejercicio 7a*

Interpolante de Trazador Cúbico
Dada una función f definida en $[a, b]$ y un conjunto de nodos $a = x_0, x_1, \ldots, x_n = b$, un **interpolante de trazador cúbico** S para f es una función que cumple con las siguientes condiciones:

a. $S(x)$ es un polinomio cúbico denotado $S_j(x)$ en el subintervalo $[x_j, x_{j+1}]$ para cada $j = 0, 1, \ldots, n-1$;

b. $S(x_j) = f(x_j)$ para cada $j = 0, 1, \ldots, n$;

c. $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ para cada $j = 0, 1, \ldots, n-2$;

d. $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ para cada $j = 0, 1, \ldots, n-2$;

e. $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ para cada $j = 0, 1, \ldots, n-2$;

f. Una de las siguientes condiciones de frontera se satisface:

   I. $S''(x_0) = S''(x_n) = 0$ (**frontera libre o natural**)

   II. $S''(x_0) = f(x_0)$ y $S''(x_n) = f(x_n)$ (**frontera sujeta**)

Deducción del Teorema 3.12
Si queremos construir el interpolante del trazador cúbico de una determinada función f aplicamos las condiciones de la definición a los polinomios cúbicos:

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

Cuando $x = x_j$ se tiene que $S_j(x_j) = a_j = f(x_j)$. Si además se define $a_n = f(x_n)$ y se aplica la condición **a.** a la expresión anterior se obtiene :

$$a_{j+1} = S_{j+1}(x_{j+1}) = S_j(x_{j+1}) = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \quad j = 0, 1, \ldots, n-1 \tag{1}$$

con $h_j = x_{j+1} - x_j$.

Entonces si también $a_n = f(x_n)$, la primera derivada del interpolante del trazador cúbico, S'(x) es:

$$S_j^{'} = \frac{dS_j(x)}{dx} = b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2$$

Al hacer $x = x_j \Rightarrow S_j^{'}(x_j) = b_j$ y habiendo definido $b_n = S^{'}(X_n)$ para cada $j = 0, 1, \ldots, n-1$. Al aplicar la condición **d.** Se tiene:

$$b_{j+1} = S_{j+1}^{'}(x+1) = S_j^{'}(x+1) = b_j + 2c_j(x_{j+1} - x_j) + 3d_j(x_{j+1} - x_j)^2$$

Entonces,

$$b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2 \tag{2}$$

para $j = 0, 1, \ldots, n-1$. La segunda derivada $S^{''}(x)$ es:

$$\frac{dS_j^{'}(x)}{dx} = 2c_j + 6d_j(x - x_j) = S_j^{''}(x)$$

Cuando $x = x_j$:

$$S_j^{''}(x_j) = 2c_j \Rightarrow c_j = \frac{S_j^{''}(x)}{2}$$

Luego al aplicar la condición **c.**

$$c_{j+1} = S_{j+1}^{''}(x+1) = S_j^{''}(x+1) = c_j + 3d_j(x_{j+1} - x_j)$$

Si $c_n = \frac{S^{''}(x_n)}{2}$, entonces:

$$c_{j+1} = c_j + 3d_j h_j \tag{3}$$

para $j = 0, 1, \ldots, n-1$. Al despejar d_j en (3) y sustituir en (1) se obtiene:

$$c_{j+1} = c_j + 3d_j h_j \Rightarrow d_j = \frac{c_{j+1} - c_j}{3h_j} \tag{4}$$

$$a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3$$

Usando (4):

$$a_j + b_j h_j + c_j h_j^2 + \frac{c_{j+1} - c_j}{3h_j} h_j^3 \Rightarrow$$

$$a_{j+1} = a_j + b_j h_j + c_j h_j^2 + \frac{c_{j+1} h_j^2}{3} - \frac{c_j h_j^2}{3} \tag{5}$$

$$\Rightarrow a_{j+1} = a_j + b_j h_j + \frac{h_j^2}{3}(2c_j + c_{j+1})$$

Reemplazando (4) en (2):

$$b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2 = b_j + 2c_j h_j + \frac{3(c_{j+1} - j)h_j^2}{3h_j}$$

$$\Rightarrow b_{j+1} = b_j + 2c_j h_j + c_{j+1} h_j - c_j h_j = b_j + h_j(c_j + c_{j+1}) \tag{6}$$

Reordenando (5):

$$b_j h_j = a_{j+1} - a_j - \frac{h_j^2}{3}(2c_j + c_{j+1})$$

$$b_j = \frac{a_{j+1} - a_j}{h_j} - \frac{h_j * (2c_j + c_{j+1})}{3} \tag{7}$$

Reduciendo en 1 el índice en (7):

$$b_{j-1} = \frac{a_j - a_{j-1}}{h_{j-1}} - \frac{-h_{j-1} * (2c_{j-1} + c_j)}{3} \tag{8}$$

Reduciendo en 1 el índice en (6):

$$b_j = b_{j-1} + h_{j-1} * (c_{j-1} + c_j)$$

Reemplazando en la anterior con (8) y operando algebraicamente:

$$b_j = \frac{a_j - a_{j-1}}{h_{j-1}} - \frac{2h_{j-1}c_{j-1}}{3} - \frac{h_{j-1}c_j}{3} + h_{j-1}c_{j-1} + h_{j-1}c_j$$
$$b_j = \frac{a_j - a_{j-1}}{h_{j-1}} + \frac{h_{j-1}c_{j-1}}{3} + \frac{2h_{j-1}c_j}{3}$$

Igualando la anterior a (7) y operando algebraicamente:

$$b_j = \frac{a_j - a_{j-1}}{h_{j-1}} + \frac{h_{j-1}c_{j-1}}{3} + \frac{2h_{j-1}c_j}{3} = \frac{a_{j+1} - a_j}{h_j} - \frac{h_j * (2c_j + c_{j+1})}{3}$$

$$\Rightarrow \frac{3(a_j - a_{j-1})}{h_{j-1}} + h_{j-1}c_{j-1} + 2h_{j-1}c_j = \frac{3(a_{j+1} - a_j)}{h_j} - 2h_jc_j - h_jc_{j+1}$$
$$\Rightarrow h_{j-1}c_{j-1} + 2h_{j-1}c_{j-1} + 2h_jc_j + h_jc_{j+1} = \frac{3 * (a_{j+1} - a_j)}{h_j} - \frac{3 * (a_j - a_{j-1})}{h_{j-1}}$$

$$\Rightarrow h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_jc_{j+1} = \frac{3 * (a_{j+1} - a_j)}{h_j} - \frac{3 * (a_j - a_{j-1})}{h_{j-1}} \tag{9}$$

La expresión (9), para $j = 1, 2, ..., n-1$ contiene solo $\{c_j\}_{J=0}^{n}$ como incógnitas, ya que los valores de $\{h_j\}_{J=0}^{n-1}$ y de $\{a_j\}_{J=0}^{n}$ están dados por el espaciado en los nodos $\{x_j\}_{j=0}^{n}$ y los valores de f en éstos. Sabiendo los c_j se pueden hallar los b_j y los d_j partiendo de las expresiones (7) y (4).

Resumiendo:

$$\begin{cases} a_j = S_j(x_j) = f(x_j) \\ b_j = \frac{a_{j+1} - a_j}{h_j} - \frac{h_j(2c_j + c_{j+1})}{3} \\ c_j \rightarrow \text{Sistema de ecuaciones originado en (9)} \\ d_j = \frac{c_{j+1} - c_j}{3h_j} \end{cases}$$

Se plantea el interrogante de si se pueden determinar los c_j por medio del sistema de ecuaciones dado en (9). El siguiente teorema indica que esto es posible cuando se establece la condición $f(i)$ de la definición de trazador cúbico.

Teorema 3.12: Si f está definida en $a = x_0 < x_1 < \ldots < x_n = b$ y es diferenciable en a y b, entonces f tendrá un interpolante único de trazador sujeto en los nodos $x_0, x_1, ..., x_n$; es decir, un interpolante de trazador que cumple con las condiciones de frontera $S^{'}(a) = f^{'}(a)$ y $S^{'}(b) = f^{'}(b)$.

Demostración: Se tiene que $S_j^{'}(x) = b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2$ luego en $x = x_j$ tenemos $S_j^{'}(x) = b_j$. En x_0 se tiene que $f^{'}(a) = S^{'}(a) = S^{'}(x_0) = b_0$.
De la expresión (7) se tiene:

$$b_0 = \frac{a_1 - a_0}{h_0} - \frac{h_0(2c_0 + c_1)}{3}$$
$$\Rightarrow f^{'}(a) = b_0 = \frac{a_1 - a_0}{h_0} - \frac{h_0(2c_0 + c_1)}{3}$$
$$\Rightarrow 3f^{'}(a) = \frac{3(a_1 - a_0)}{h_0} - 2h_0c_1$$

$$\Rightarrow 2h_0c_0 + h_0c_1 = \frac{3(a_1 - a_0)}{h_0} - 3f^{'}(a) \tag{10}$$

Por otro lado, a partir de la expresión (6) en $x = b$ se tiene:

$$f^{'}(b) = b_n = b_{n-1} + h_{n-1}(c_{n-1} + c_n)$$

Luego, a partir de la expresión (9) con $j = n$ se tiene:

$$b_{n-1} = \frac{a_n - a_{n-1}}{h_{n-1}} - \frac{h_{n-1}(2c_{n-1} + c_n)}{3}$$

$$\Rightarrow f'(b) = \frac{a_n - a_{n-1}}{h_{n-1}} - \frac{2h_{n-1}c_{n-1}}{3} - \frac{h_{n-1}c_n}{3} + h_{n-1}c_{n-1} + h_{n-1}c_n$$
$$\Rightarrow f'(b) = \frac{a_n - a_{n-1}}{h_{n-1}} + \frac{h_{n-1}c_{n-1}}{3} + \frac{2h_{n-1}c_n}{3}$$
$$\Rightarrow f'(b) = \frac{a_n - a_{n-1}}{h_{n-1}} - \frac{h_{n-1}c_{n-1}}{3} + \frac{2h_{n-1}c_n}{3}$$
$$\Rightarrow 3f'(b) = \frac{3(a_n - a_{n-1})}{h_{n-1}} + h_{n-1}c_{n-1} + 2h_{n-1}c_n$$

$$\Rightarrow h_{n-1}c_{n-1} + 2h_{n-1}c_n = 3f'(b) - \frac{a_n - a_{n-1}}{h_{n-1}} \tag{11}$$

Combinando las expresiones (9), (10) y (11) se obtiene el sistema lineal $Ax = b$, donde

$$A = \begin{bmatrix} 2h_0 & h_0 & 0 & \cdots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & & \\ 0 & h_1 & 2(h_1+h_2) & h_2 & \\ \vdots & & \ddots & \ddots & 0 \\ & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & \cdots & 0 & h_{n-1} & 2h_{n-1} \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} \frac{3}{h_0}(a_1 - a_0) - 3f'(a) \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 3f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1}) \end{bmatrix}, \quad \text{y} \quad \mathbf{x} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}$$

Fig. 1

La matriz A es estrictamente diagonal dominante, y por lo tanto cumple con la condición del teorema 6.19. En consecuencia el sistema tiene única solución $c_0, c_1, ..., c_n$.

B. Ejercicio 7b

Utilizamos el algoritmo 1 para obtener los coeficientes a, b, c y d de las ecuaciones de trazador cúbico sujeto. Los mismos se obtienen de introducir los valores de las condiciones iniciales. En el vector de x ponemos los puntos donde tenemos evaluada la función (dentro del intervalo). En f ponemos la función evaluada en los x anteriores y en df ponemos las derivadas en los extremos del vector.Lo hacemos así ya que son las condiciones de borde a considerar y completamos el vector con cualquier valor ya que el algoritmo no los utiliza.

Al utilizar el algoritmo 1 obtenemos los siguientes resultados:

$a_0 = 0$	$b_0 = 3$	$c_0 = 2$	$d_0 = -1$
$a_1 = 4$	$b_1 = 4$	$c_1 = -1$	$d_1 = 1/3$

Por ende en las funciones resultantes son:

$$S(x) = \begin{cases} S_0(x) = 0 + 3(x-1) + 2(x-1)^2 - (x-1)^3, \text{ si } 1 \le x \le 2 \\ S_1(x) = 4 + 4(x-2) - (x-2)^2 + \frac{(x-2)^3}{3}, \text{ si } 2 \le x \le 3 \end{cases}$$

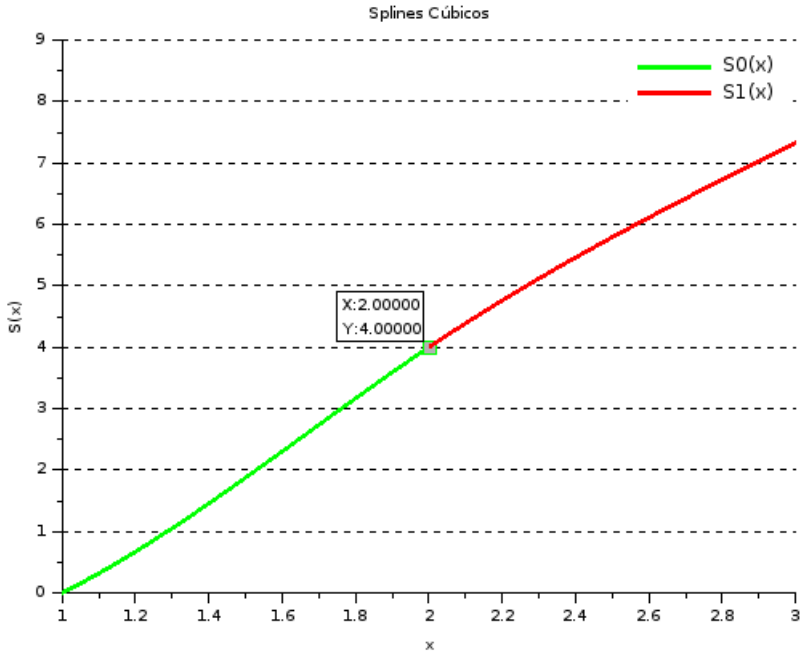

Fig. 2
SPLINE CÚBICO SUJETO

C. *Ejercicio 7c*

Primero Ordenamos los datos en una tabla de $Tiempo[seg]$, x e y:

Tiempo [seg]	0	1	2	3	4
x	0	1	4	3	0
y	0	2	4	1	0

Se nos da como dato que la velocidad es igual a cero en los puntos $(0,0)$ y $(3,1)$, es decir, que $\frac{dx}{dt} = 0$ y $\frac{dy}{dt} = 0$ en ambos puntos. Se procede a realizar una interpolación para y, y otra para x. Además conocer las derivadas nos permite desarmar el problema en dos curvas de trazadores cúbicos sujetos. Siendo finalmente los mismos S_0, S_1, S_2, S_3.
Los polinomios para los trazadores son:

$$
\begin{aligned}
x_0(t) &= a_0 + b_0 * (t-0) + c_0 * (t-0)^2 + d_0 * (t-0)^3 \\
x_1(t) &= a_1 + b_1 * (t-1) + c_1 * (t-1)^2 + d_1 * (t-1)^3 \\
y_0(t) &= r_0 + s_0 * (t-0) + t_0 * (t-0)^2 + u_0 * (t-0)^3 \\
y_1(t) &= r_1 + s_1 * (t-1) + t_1 * (t-1)^2 + u_1 * (t-1)^3 \\
x_2(t) &= a_2 + b_2 * (t-2) + c_2 * (t-2)^2 + d_2 * (t-2)^3 \\
x_3(t) &= a_3 + b_3 * (t-3) + c_3 * (t-3)^2 + d_3 * (t-3)^3 \\
y_2(t) &= r_2 + s_2 * (t-2) + t_2 * (t-2)^2 + u_2 * (t-2)^3 \\
y_3(t) &= r_3 + s_3 * (t-3) + t_3 * (t-3)^2 + u_3 * (t-3)^3
\end{aligned}
$$

Aplicando el algoritmo 1, en intervalos de tiempo entre $[0,2]$ y entre $[2,4]$ para los valores de x e y presentados en la tabla anterior. Se obtienen los coeficientes a, b, c, d del polinomio interpolante. Se los presenta a continuación:

$$
\begin{aligned}
x_0(t) &= 0 + 0 * (t-0) + 0 * (t-0)^2 + 1 * (t-0)^3 \\
x_1(t) &= 1 + 3 * (t-1) + 3 * (t-1)^2 - 3 * (t-1)^3 \\
y_0(t) &= 0 + 0 * (t-0) + 3 * (t-0)^2 - 1 * (t-0)^3 \\
y_1(t) &= 2 + 3 * (t-1) + 0 * (t-1)^2 - 1 * (t-1)^3 \\
x_2(t) &= 4 + 0 * (t-2) + 0 * (t-2)^2 - 1 * (t-2)^3 \\
x_3(t) &= 3 - 3 * (t-3) - 3 * (t-3)^2 + 3 * (t-3)^3 \\
y_2(t) &= 4 + 0 * (t-2) - 6 * (t-2)^2 + 3 * (t-2)^3 \\
y_3(t) &= 1 - 3 * (t-3) + 3 * (t-3)^2 - 1 * (t-3)^3
\end{aligned}
$$

Luego se construyeron los algoritmos 2, 3, 4, 5, 6, 7, 8 y 9 que representan a las polinomios antes listados. Utilizando estos algoritmos y evaluándolos en los intervalos $[0,1],[1,2],[2,3]$ y $[3,4]$ para cada caso, se procedió a graficar x vs y en forma logarítmica (10); se muestra dicha gráfica a continuación:

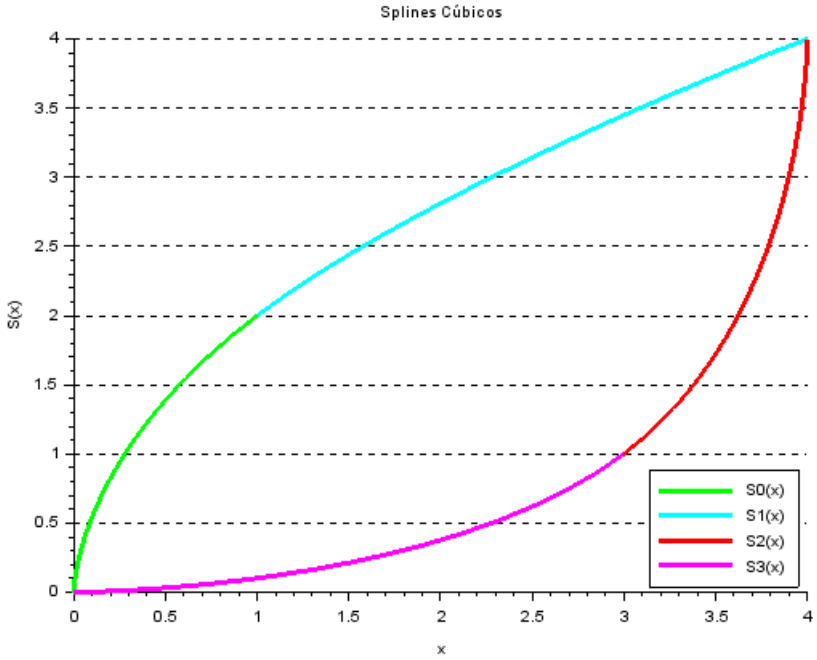

Fig. 3
GRÁFICA LOGARÍTMICA DE $y(t)$ VS. $x(t)$

II. EJERCICIO 9

En el siguiente conjunto de datos, presentados al Senate Antitrust Subcommittee de Estados Unidos, muestra las características comparativas de choque-supervivencia de automóviles de varios tipos. Obtenga la recta de mínimos cuadrados que aproxima estos datos. (tabla contiene el porcentaje de vehículos accidentados en los cuales la lesión más seria fue fatal o grave.

Tipo	Peso Promedio	ocurrencia
Regular de lujo, de fabricación nacional	4800lb	3.1
Regular intermedio, de fabricación nacional	3700lb	4.0
Económico regular, de fabricación nacional	3400lb	5.2
Compacto de fabricación nacional	2800lb	6.4
Compacto de fabricación extranjera	1900lb	9.6

El método de **mínimos cuadrados** para la resolución de este problema requiere determinar la mejor línea aproximante, cuando el error es la suma de los cuadrados de las diferencias entre los valores de y en la línea aproximante y los valores de y dados. Dada una colección de datos $\{(x_i,y_i)\}_{i=1}^m$ el método implica minimizar el error total, según:

$$E = E_2(a_0,a_1) = \sum_{i=1}^{m}[y_i - (a_1x_i + a_0)]^2$$

Por tanto, hay que encontrar los valores de las constantes a_0 y a_1 que minimicen el error. Para que haya un mínimo debemos tener:

$$0 = \frac{\partial}{\partial a_0}\sum_{i=1}^{m}[y_i - (a_1x_i + a_0)]^2 = 2\sum_{i=1}^{m}(y_i - a_1x_i - a_0)(-1)$$
$$0 = \frac{\partial}{\partial a_1}\sum_{i=1}^{m}[y_i - (a_1x_i + a_0)]^2 = 2\sum_{i=1}^{m}(y_i - a_1x_i - a_0)(-x_i)$$

Estas ecuaciones se simplifican en las ecuaciones normales:

$$a_0 \cdot m + a_1\sum_{i=1}^{m}x_i = \sum_{i=1}^{m}y_i \text{ y } a_0\sum_{i=1}^{m}x_i + a_1\sum_{i=1}^{m}x_i^2 = \sum_{i=1}^{m}x_iy_i$$

Operando algebraicamente estas ecuaciones se puede derivar en las expresiones finales para las constantes:

$$a_0 = \frac{\sum_{i=1}^{m} x_i^2 \sum_{i=1}^{m} y_i - \sum_{i=1}^{m} x_i y_i \sum_{i=1}^{m} x_i}{m\left(\sum_{i=1}^{m} x_i^2\right) - \left(\sum_{i=1}^{m} x_i\right)^2} \qquad a_1 = \frac{m\sum_{i=1}^{m} x_i y_i - \sum_{i=1}^{m} x_i \sum_{i=1}^{m} y_i}{m\left(\sum_{i=1}^{m} x_i^2\right) - \left(\sum_{i=1}^{m} x_i\right)^2}$$

A partir de estas expresiones se elaboró el algoritmo *least_squares* 11, el cual recibe como parámetros el conjunto de datos $\{(x_i, y_i)\}_{i=1}^{5}$ (ver tabla anterior). Con los datos de x_i e y_i y utilizando el algoritmo construimos la siguiente tabla:

x_i	y_i	$P(x_i) = -0{,}002255x_i + 13{,}1465$
4800	3,1	2.3225
3700	4	4.7579
3400	5,2	5.4795
2800	6,4	6.8325
1900	9,6	8.862

Finalmente utilizando el algoritmo *plot_ls* 12 y los datos obtenidos, se realiza la siguiente gráfica:

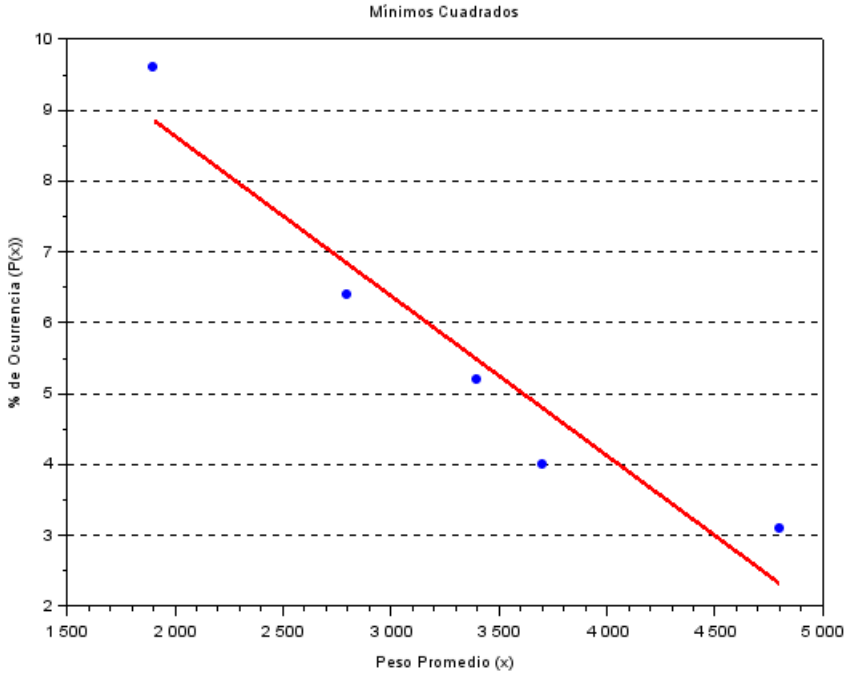

Fig. 4
CÁLCULO DE ECUACIÓN DE APROXIMACIÓN CON MÍNIMOS CUADRADOS

REFERENCIAS

[1] R. L. Burden, *Análisis Numérico*, 7th ed. Thomson Learning, 2002.

[2] V. Sonzogni, "Cálculo numérico - apuntes de cátedra," 2014.

ANEXO

A continuación se muestran los códigos utilizados para el desarrollo de este trabajo práctico:

```
function[a,b,c,d]=cubic_spline_clamped(x,f,df)
   n= length(x);
   h= x(2:n)-x(1:n-1); //Separacion entre dos nodos
   A = zeros(n,n);
   rhs = zeros(n,n); //Es el vector solución, vendria a ser el b de Ax=b
   A(1,1) = 2*h(1);
   A(1,2) = h(1);
   A(n,n-1)=h(n-1);
   A(n,n) = 2*h(n-1);
   rhs(1)=3*(f(2) - f(1)) / h(1) - 3* df(1);
   rhs(n)=3* df(n) - 3*(f(n) - f(n-1)) / h(n-1);

   for (i=2:n-1);
      A(i,i-1) = h(i-1);
      A(i,i) = 2*(h(i) + h(i-1));
      A(i,i+1) = h(i);
      rhs(i) = 3*(f(i+1) - f(i)) / h(i) -3*(f(i)-f(i-1))/h(i-1);
   end
   c = gauss(A,rhs);
   b = (f(2:n)-f(1:n-1))./h(1:n-1)-(2*c(1:n-1)+c(2:n)).*h(1:n-1)/3;
   d = (c(2:n)-c(1:n-1))./(3*h(1:n-1));
   a = f(1:n-1);
   c = c(1:n-1);
endfunction
```

Algoritmo 1: Trazador cúbico sujeto

```
function[r]=x0(t)
   r=1*(t^3);
endfunction
```

Algoritmo 2: Evalúa x_0 en t

```
function[r]=x1(t)
   r=1+3*(t-1)+3*((t-1)^2)-3*((t-1)^3);
endfunction
```

Algoritmo 3: Evalúa x_1 en t

```
function[r]=y0(t)
   r=3*(t^2)-1*(t^3);
endfunction
```

Algoritmo 4: Evalúa y_0 en t

```
function[r]=y1(t)
   r=2+3*(t-1)+0*((t-1)^2)-1*((t-1)^3);
endfunction
```

Algoritmo 5: Evalúa y_1 en t

```
function[r]=x2(t)
   r=4+0*(t-2)+0*((t-2)^2)-1*((t-2)^3);
endfunction
```

Algoritmo 6: Evalúa x_2 en t

```
function[r]=x3(t)
   r=3-3*(t-3)-3*((t-3)^2)+3*((t-3)^3);
endfunction
```

Algoritmo 7: Evalúa x_3 en t

```
function[r]=y2(t)
   r=4+0*(t-2)-6*((t-2)^2)+3*((t-2)^3);
endfunction
```

Algoritmo 8: Evalúa y_2 en t

```
function[r]=y3(t)
   r=1-3*(t-3)+3*((t-3)^2)-1*((t-3)^3);
endfunction
```

Algoritmo 9: Evalúa y_3 en t

```
function [] = plot_spline(x1,y1,x2,y2,x3,y3,x4,y4)
   scf(5);
   clf(5);

   plot2d("nn",x1,y1,style=3);
   p1 = get("hdl");
   p1.children.thickness = 3;
   plot2d("nn",x2,y2,style=4);
   p2 = get("hdl");
   p2.children.thickness = 3;
   plot2d("nn",x3,y3,style=5);
   p3 = get("hdl");
   p3.children.thickness = 3;
   plot2d("nn",x4,y4,style=6);
   p4 = get("hdl");
   p4.children.thickness = 3;
   legend(["S0(x)";"S1(x)";"S2(x)";"S3(x)"], with_box=%f, opt="?","in_lower_right");

   xtitle("Splines Cúbicos", "x", "S(x)"); // titulo, eje x, eje y
   set(gca(), "grid", [-1,1]);
endfunction
```

Algoritmo 10: Graficador de Splines

```
function [a0,a1,E] = least_sqares(x,y)
   [fx,cx] = size(x);
   [fy,cy] = size(y);
   nx = 0; // longitud de x
   ny = 0; // longitud de y
   m = 0;
   a0 = 0;
   a1 = 0;

   // Determina la longitud del vector x. También lo convierte a vector fila
   if fx == 1 then
      nx = cx;
      x = x';
   else
      nx = fx;
   end

   // Determina la longitud del vector y. También lo convierte a vector fila
   if fy == 1 then
      ny = cy;
      y = y';
   else
      ny = fy;
   end

   if nx ~= ny then
      disp("Dimensiones de los datos de entrada inconsistentes.");
      return;
   else
      m = nx;
   end

   a0 = (sum(x.^2)*sum(y) - sum(x.*y)*sum(x)) / (m * (sum(x.^2)) - (sum(x))^2);
   a1 = (m*sum(x.*y) - sum(x)*sum(y)) / (m*sum(x.^2) - (sum(x)^2));

   P = zeros(m,1);
   P(1:m) = a0 + a1 * x(1:m);
   E = sum((y(1:m) - P(1:m))^2);

endfunction
```

Algoritmo 11: Mínimos cuadrados

```
function [] = plot_ls(x,y,a0,a1)
   scf(5);
   clf(5);

   P = a0 + a1 * x;

   plot(x,y,".");

   plot2d("nn",x,P,style=5);
   p1 = get("hdl");
   p1.children.thickness = 3;


   xtitle("Mínimos Cuadrados", "Peso Promedio (x)", "% de Ocurrencia (P(x))"); // titulo,
       eje x, eje y
   set(gca(), "grid", [-1,1]);
endfunction
```

Algoritmo 12: Gráficas de Mínimos cuadrados