

# Informe III:

## Trabajo Práctico 5

Edgardo Cipolatti  
edgardocipolatti@hotmail.com

### I. EJERCICIO 7

- Explique detalladamente cómo introduce las condiciones de borde de frontera sujeta en el sistema de ecuaciones lineales que permite encontrar los coeficientes del trazador cúbico sujeto. (**Observación: los pasos en la demostración del teorema 3.12 del libro de Burden no están completos, por lo que la transcripción directa de dicho teorema no bastarán para aprobar el práctico.**)
- 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.
- 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 instantes inicial. Luego de 1s se debe encontrar en el punto (1;5; 3), 1s después debe alcanzar el punto (4; 5) y detenerse allá, para recomenzar inmediatamente su movimiento y alcanzar, luego de otro segundo más 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  $a = x_0, x_1, \dots, x_n = b$ , un **interpolante de trazador cúbico**  $S$  para  $f$  es una función que cumple con las siguientes condiciones:

- $S(x)$  es un polinomio cúbico denotado  $S_j(x)$  en el subintervalo  $[x_j, x_{j+1}]$  para cada  $j = 0, 1, \dots, n-1$ ;
- $S(x_j) = f(x_j)$  para cada  $j = 0, 1, \dots, n$ ;
- $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$  para cada  $j = 0, 1, \dots, n-2$ ;
- $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$  para cada  $j = 0, 1, \dots, n-2$ ;
- $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$  para cada  $j = 0, 1, \dots, n-2$ ;
- Una de las siguientes condiciones de frontera se satisface:
  - $S'''(x_0) = S'''(x_n) = 0$  (**frontera libre o natural**)
  - $S'''(x_0) = f'(x_0)$  y  $S'''(x_n) = f'(x_n)$  (**frontera sujeta**)

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

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3 \quad (1)$$

para cada  $j = 0, 1, \dots, n-1$ . Entonces  $S_j(x_j) = a_j = f(x_j)$ . Si se aplica la condición **c.** a la expresión anterior se obtiene :

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

para  $j = 0, 1, \dots, n-2$  Por simplicidad reemplazamos  $(x_{j+1} - x_j)$  por  $h_j$ . Si definimos  $a_n = f(x_n)$  nos queda

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

de manera análoga se define  $b_n = S'(x_n)$  y al aplicar la condición **d.** tenemos

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

Ahora, al definir  $c_j = \frac{S''_j(x)}{2}$  y aplicar la condición **e.** se obtiene la relación entre los coeficientes de  $S_j$

$$c_{j+1} = c_j + 3d_j h_j \quad (4)$$

al despejar  $d_j$  y sustituir en 2 y 3 se obtiene

$$a_{j+1} = a_j + b_j h_j + \frac{h_j^2}{3}(2c_j + c_{j+1}) \quad (5)$$

$$b_{j+1} = b_j + h_j(c_j + c_{j+1}) \quad (6)$$

La relación que incluye los coeficientes se obtiene despejando  $b_j$  de 5, por lo que se consigue

$$b_{j-1} = \frac{a_j - a_{j-1}}{h_{j-1}} - \frac{-h_{j-1} * (2c_{j-1} + c_j)}{3} \quad (7)$$

si se cambia el subíndice  $j$  por  $j - 1$  y se sustituyen los valores en 6 obtenemos el sistema de ecuaciones lineales

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

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

**Demostración:** Puesto que  $f'(a) = S'(a) = S'(x_0) = b_0$  (esto sale de evaluar  $S'_0(x) = b_0 + 2c_0(x - x_0) + 3d_0(x - x_0)^2$  en  $x_0$ ) podemos ver que la ecuación 7 con  $j = 0$  implica que

$$f'(a) = \frac{1}{h_0}(a_1 - a_0) - \frac{h_0}{3}(2c_0 - c_1)$$

En consecuencia

$$2h_0c_0 + h_0c_1 = \frac{3}{h_0}(a_1 - a_0) - 3f'(a) \quad (9)$$

De manera semejante al evaluar

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

de modo que la ecuación 7 con  $j = n$  se tiene

$$\begin{aligned} b_{n-1} &= \frac{a_n - a_{n-1}}{h_{n-1}} - \frac{h_{n-1}(2c_{n-1} + c_n)}{3} \\ \Rightarrow f'(b) &= \frac{a_n - a_{n-1}}{h_{n-1}} - \frac{2h_{n-1}c_{n-1}}{3} - \frac{h_{n-1}c_n}{3} + h_{n-1}c_{n-1} + h_{n-1}c_n \\ \Rightarrow f'(b) &= \frac{a_n - a_{n-1}}{h_{n-1}} + \frac{h_{n-1}c_{n-1}}{3} + \frac{2h_{n-1}c_n}{3} \\ \Rightarrow f'(b) &= \frac{a_n - a_{n-1}}{h_{n-1}} - \frac{h_{n-1}c_{n-1}}{3} + \frac{2h_{n-1}c_n}{3} \\ \Rightarrow 3f'(b) &= \frac{3(a_n - a_{n-1})}{h_{n-1}} + h_{n-1}c_{n-1} + 2h_{n-1}c_n \\ \Rightarrow h_{n-1}c_{n-1} + 2h_{n-1}c_n &= 3f'(b) - \frac{a_n - a_{n-1}}{h_{n-1}} \end{aligned} \quad (10)$$

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

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

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

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

### B. Ejercicio 7b

Se presenta el algoritmo 1 el cual devuelve los coeficientes del spline y recibe como parámetro los valores de la variable independiente, los valores de la variable dependiente, y las derivadas en los extremos (un vector de derivadas pero solo considera el primer y ultimo elemento)

### 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	1.5	4	3	0
y	0	3	5	1	0

Se nos da como dato que la velocidad es igual a cero en los puntos (0,0) y (4,5), 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:

$$\begin{aligned}
 x_0(t) &= 0 + 0(t - 0) + 1,5(t - 0)^2 + 0(t - 0)^3 \\
 x_1(t) &= 1,5 + 3(t - 1) + 1,5(t - 1)^2 - 2(t - 1)^3 \\
 y_0(t) &= 0 + 0(t - 0) + 5,25(t - 0)^2 - 2,25(t - 0)^3 \\
 y_1(t) &= 3 + 3,75(t - 1) - 1,5(t - 1)^2 - 0,25(t - 1)^3 \\
 x_2(t) &= 4 + 0(t - 2) + 0(t - 2)^2 - 1(t - 2)^3 \\
 x_3(t) &= 3 - 3(t - 3) - 3(t - 3)^2 + 3(t - 3)^3 \\
 y_2(t) &= 5 + 0(t - 2) - 8,25(t - 2)^2 + 4,25(t - 2)^3 \\
 y_3(t) &= 1 - 3,75(t - 3) + 4,5(t - 3)^2 - 1,75(t - 3)^3
 \end{aligned}$$

Luego se construyó el algoritmo 2 que representa a los polinomios antes listados. Utilizando estos polinomios y evaluándolos en los intervalos [0,1],[1,2],[2,3] y [3,4] para cada caso, se procedió a graficar x vs y; se muestra dicha gráfica en la figura 1 a continuación:

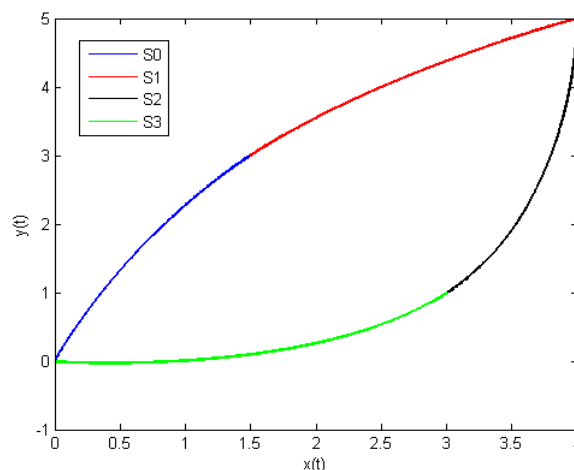


Fig. 1

GRÁFICA DE X(T) VS Y(T)

Tensión [MPa]	34,5	60,9	103,5	138,0
Deformación [mm] - Test 1	0,46	0,95	1,48	1,93
Deformación [mm] - Test 2	0,34	1,02	1,51	2,09
Deformación [mm] - Test 3	0,73	1,10	1,62	2,12

TABLA 1  
TABLA DE DATOS EJERCICIO 9 - ENSAYOS DE TRACCIÓN.

## II. EJERCICIO 9

Se realizaron tres ensayos de tracción sobre una barra de aluminio. En cada ensayo, se midieron cuatro valores de la deformación, para los mismos valores de la tensión aplicada entre ensayo y ensayo. Los resultados se muestran en la tabla 1. Utilizar regresión lineal para estimar el módulo de elasticidad de la barra.

El módulo de Young o módulo de elasticidad longitudinal es un parámetro que caracteriza el comportamiento de un material elástico, según la dirección en la que se aplica una fuerza.

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:

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

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:

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

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

Estas ecuaciones se simplifican en las ecuaciones normales:

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

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

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

A partir de estas expresiones se elaboró el algoritmo 4 "mínimos cuadrados" el cual recibe como parámetros el conjunto de datos  $(x_i, y_i)_{i=1}^{12}$  y nos devuelve la pendiente, la ordenada al origen y el error total de la recta de aproximación por mínimos cuadrados.

La recta que se obtuvo es la siguiente:  $P(x) = 0,0149 * x - 0,0017$  donde 0,0149 es la pendiente de la recta y  $-0,0017$  es la ordenada al origen. En este ejercicio en particular la pendiente representa el módulo de elasticidad del material (aluminio).

La datos utilizados se presenta en la tabla II y los resultados obtenidos se muestran en la figura 2 encontrados con el algoritmo 3

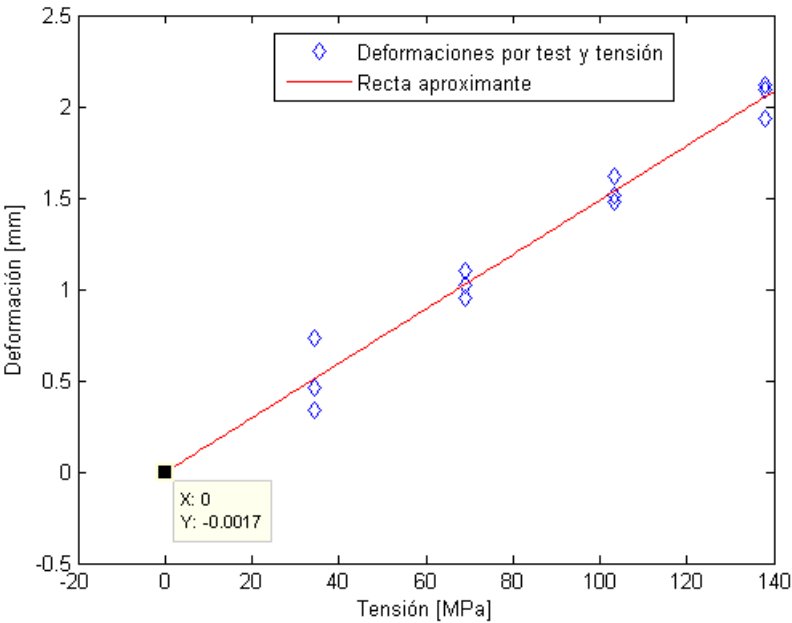


Fig. 2  
RECTA APROXIMANTE OBTENIDA POR MÍNIMOS CUADRADOS

$x_i$	$y_i$	$P(x_i) = 0,0149 * x_i - 0,0017$
34,5	0.46	0.5123
69,0	0.95	1.0264
103,5	1.48	1.5404
138,0	1.93	2.0545
34,5	0.34	0.5123
69,0	1.02	1.0264
103,5	1.51	1.5404
138,0	2.09	2.0545
34,5	0.73	0.5123
69,0	1.10	1.0264
103,5	1.62	1.5404
138,0	2.12	2.0545

TABLA 2  
DATOS Y RESULTADOS OBTENIDOS POR MÍNIMOS CUADRADOS

REFERENCIAS

[1] R. L. Burden, *Análisis Numérico*, 7th ed. Thomson Learning, 2002.  
[2] V. Sonzogni, “Cálculo numérico - apuntes de cátedra,” 2015.  
[3] “Módulo de young,” Wikipedia, 2015.

## ANEXO

## A. APÉNDICE

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

---

```
function [a,b,c,d]=cubic_spline_sujeto(x,f,df)
    n= length(x);
    h= x(2:n)-x(1:n-1);
    A = zeros(n,n);
    rhs = zeros(n,n);
    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(:,1));
    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);
end
```

---

**Algoritmo 1:** Cálculo de los coeficientes de los trazadores cúbicos

---

```
t0=0:0.001:1;
t1=1:0.001:2;
t2=2:0.001:3;
t3=3:0.001:4;

Sx0=0+0*(t0-0)+1.5*(t0-0).^2+0*(t0-0).^3;
Sx1=1.5+3*(t1-1)+1.5*(t1-1).^2-2*(t1-1).^3;
Sx2=4+0*(t2-2)+0*(t2-2).^2-1*(t2-2).^3;
Sx3=3-3*(t3-3)-3*(t3-3).^2+3*(t3-3).^3;

Sy0=0+0*(t0-0)+5.25*(t0-0).^2-2.25*(t0-0).^3;
Sy1=3+3.75*(t1-1)-1.5*(t1-1).^2-0.25*(t1-1).^3;
Sy2=5+0*(t2-2)-8.25*(t2-2).^2+4.25*(t2-2).^3;
Sy3=1-3.75*(t3-3)+4.5*(t3-3).^2-1.75*(t3-3).^3;

plot(Sx0,Sy0,'b','LineWidth',2)
hold on;
plot(Sx1,Sy1,'r','LineWidth',2)
hold on;
plot(Sx2,Sy2,'k','LineWidth',2)
hold on;
plot(Sx3,Sy3,'g','LineWidth',2)
legend('S0','S1','S2','S3')
xlabel('x(t)')
ylabel('y(t)')
```

---

**Algoritmo 2:** Cálculo y gráfico de la posición de robot en x e y para el tiempo t

---

```

plot(x,y,'d')
hold on
plot((-1:140),0.0149*(-1:140)-0.0017,'r')
xlabel('Tensión [MPa]');
ylabel('Deformación [mm]');
legend('Deformaciones por test y tensión','Recta aproximante')

```

---

**Algoritmo 3:** Grafica de los mínimos cuadrados

---

```

function [a0,a1,E] = minimos_cuadrados(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.
    if fx == 1
        nx = cx;
        x = x';
    else
        nx = fx;
    end

    % Determina la longitud del vector y.
    if fy == 1
        ny = cy;
        y = y';
    else
        ny = fy;
    end

    if nx ~= ny
        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);
end

```

---

**Algoritmo 4:** mínimos cuadrados