Trabajo práctico Nº5

Darién Julián Ramírez
Universidad Nacional del Litoral. ianshalaga@gmail.com

I. Ejercicio 7.

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

Función desarrollada en el ejercicio 6.b): Trazador cúbico natural

```
function [a,b,c,d] = cubic_spline_natural(x,f)
    n = length(x);
    b = zeros(n,1);
    c = zeros(n,1);
    d = zeros(n,1);
    for i=1:n-1
        h(i)=x(i+1)-x(i);
    end
    for i=2:n-1
        alfa(i) = ((3*(f(i+1)-f(i)))/h(i)) - ((3*(f(i)-f(i-1)))/h(i-1));
    end
    l(1) = 1;
    u(1) = 0;
    z(1) = 0;
    for i=2:n-1
        l(i) = (2*(x(i+1)-x(i-1))) - (h(i-1)*u(i-1));
        u(i) = h(i)/l(i);
        z(i) = (alfa(i)-(h(i-1)*z(i-1)))/l(i);
    end
    l(n) = 1;
    z(n) = 0;
    c(n) = 0;
    for j=n-1:-1:1
        c(j) = z(j)-(u(j)*c(j+1));
        b(j) = ((f(j+1)-f(j))/h(j)) - ((h(j)*(c(j+1)+(2*c(j))))/3);
        d(j) = (c(j+1)-c(j))/(3*h(j));
    end
    a = f';
endfunction
```

Trazador cúbico sujeto:

```
function [a,b,c,d] = cubic_spline_clamped(x,f,df)
    n = length(x);
    b = zeros(n,1);
    c = zeros(n,1);
    d = zeros(n,1);
    for i=1:n-1
        h(i)=x(i+1)-x(i);
    end
    alfa(1) = ((3*(f(2)-f(1)))/h(1)) - (3*df(1));
    alfa(n) = (3*df(2)) - ((3*(f(n)-f(n-1)))/h(n-1));
    for i=2:n-1
        alfa(i) = ((3*(f(i+1)-f(i)))/h(i)) - ((3*(f(i)-f(i-1)))/h(i-1));
    end
    l(1) = 2*h(1);
    u(1) = 0.5;
    z(1) = alfa(1)/l(1);
    for i=2:n-1
```

```
        l(i) = (2*(x(i+1)-x(i-1))) - (h(i-1)*u(i-1));
        u(i) = h(i)/l(i);
        z(i) = (alfa(i)-(h(i-1)*z(i-1)))/l(i);
    end
    l(n) = h(n-1)*(2-u(n-1));
    z(n) = (alfa(n)-(h(n-1)*z(n-1)))/l(n);
    c(n) = z(n);
    for j=n-1:-1:1
        c(j) = z(j)-(u(j)*c(j+1));
        b(j) = ((f(j+1)-f(j))/h(j)) - ((h(j)*(c(j+1)+(2*c(j))))/3);
        d(j) = (c(j+1)-c(j))/(3*h(j));
    end
    a = f';
endfunction
```

El *Teorema 3.12* [1] justifica el algoritmo anterior:
Si f definida en $a=x_0<x_1<\ldots<x_n=b$, y es diferenciable en a y b, entonces f tiene un único trazador sujeto que interpola los nodos $x_0,x_1,\ldots,x_n$, es decir, un interpolante de trazador que cumple las condiciones de frontera $S'(a)=f'(a)$ y $S'(b)=f'(b)$.
En el proceso de construcción del interpolante del trazador cúbico[1], se llega a que el sistema de ecuaciones lineales está dado por $h_{(j-1)}c_{(j-1)}+2(h_{(j-1)}+h_j)c_j+h_j c_{(j+1)}=\frac{3}{h_j}(a_{(j+1)}-a_j)-\frac{3}{h_{(j-1)}}(a_j-a_{(j-1)})$ para *j=1,..,n-1*.

Debido a que este sistema de ecuaciones requiere dos renglones más para poder llegar a una solución, se imponen las condiciones para el trazador sujeto:

Por la definición de trazador sujeto, se debe cumplir $f'(x_0)=S'(x_0)\, y\, f'(x_n)=S'(x_n)$ con $f'(x_0)$ y $f'(x_n)$ dados. Además $S_j(x)=a_j+b_j(x-x_j)+c_j(x-x_j)^2+d_j(x-x_j)^3 \Rightarrow S'_j(x)=b_j+2c_j(x-x_j)+3d_j(x-x_j)^2$ por lo tanto $S'(x_0)=b_0=f'(x_0)$.
De la expresión para la relación entre los coeficientes dada en la ecuación: $b_j=\frac{1}{h_j}(a_{(j+1)}-a_j)-\frac{h_j}{3}(2c_j+c_{(j+1)})$ [2]*(1)* evaluada en *j=0*, se llega a $f'(a)=\frac{1}{h_0}(a_1-a_0)-\frac{h_0}{3}(2c_0+c_1)$. Colocando del lado izquierdo los c_i y el resto del lado derecho, se llega a $2h_0c_0+h_0c_1=\frac{3}{h_0}(a_1-a_0)-3f'(a)$. Lo mismo ocurre con x_n, pero en este caso, usando la relación $b_{j+1}=b_j+h_j(c_j+c_{j+1})$ se llega a $f'(x_n)=b_n=b_{n-1}+h_{n-1}(c_{n-1}+c_n)$. Usando la ecuación *(1)*, en *j=n-1*, se reemplaza: $f'(b)=\frac{a_n-a_{n-1}}{h_{n-1}}-\frac{h_{n-1}}{3}(2c_{n-1}+c_n)+h_{n-1}(c_{n-1}+c_n)=\frac{a_n-a_{n-1}}{h_{n-1}}+\frac{h_{n-1}}{3}(c_{n-1}+2c_n)$. Reagrupando: $h_{n-1}c_{n-1}+2h_{n-1}c_n=3f'(b)-\frac{3}{h_{n-1}}(a_n-a_{n-1})$
En síntesis, se obtuvieron los primeros dos renglones del sistema, los cuales al depender del valor de la primera derivada están definidos por:

$$2h_0c_0+h_0c_1=\frac{3}{h_0}(a_1-a_0)-3f'(a)$$
$$h_{(n-1)}c_{(n-1)}+2h_{(n-1)}c_n=3f'(b)-\frac{3}{h_{(n-1)}}a_n-a_{(n-1)}$$

Esto junto al sistema mencionado en el primer parrafo, determina el siguiente sistema tridiagonal:

$$A\boldsymbol{x}=\boldsymbol{b},\ \ donde$$

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

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

[1]Páginas 143-145 del libro "Análisis numérico, séptima edición" Burden y Faires. Thomson Learning.
[2]Ecuación 3.20 página 144 del libro "Análisis numérico, séptima edición" Burden y Faires. Thomson Learning.

Para verificar si el sistema tiene solución, se puede probar que la matriz A es estrictamente diagonal dominante. Esto se verifica teniendo en cuenta:

- h_i es siempre positivo, ya que es la resta de dos posiciones x_{i+1} y x_i (h_i=x_{i+1}-x_i).
- En el primer renglón de A, se tiene que el elemento de la diagonal ($2h_0$) es mayor que la suma del resto de los elementos del renglón (h_i+0).
- En el último renglón ocurre lo mismo que en el primero, ya que el elemento de la diagonal ($2h_{n-1}$) es mayor que la suma del resto de los elementos del renglón ($h_{n-1}+0$).
- Para el resto de los renglones, se tiene en cuenta lo mismo que en los ítems anteriores. Ya que h_i y h_{i+1} son positivos, el doble de su suma es siempre mayor que su suma.

Como los elementos de la diagonal son siempre mayores a la suma del resto de los elementos de cada renglón, la matriz A es estrictamente diagonal dominante, y por lo tanto es no singular[3] (posee una solución única).

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 1 [seg.] se debe encontrar en el punto (2, 4), 1 [seg.] después debe alcanzar el punto (6, 6) y detenerse allí. En una segunda etapa retoma inmediatamente su movimiento y alcanza, luego de otro segundo más el punto (3, 2) 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 realice las siguientes gráficas: (i) x vs. t (ambos tramos en la misma gráfica), (ii) y vs. t (idem anterior), y finalmente (iii) en el plano xy la trayectoria encontrada.

Del enunciado se determinan los siguientes valores de t, x, y:

t = [0 1 2 3 4]
x = [0 2 6 3 0]
y = [0 4 6 2 0]

Se fijaran las derivadas de los extremos en 0 puesto que es donde se encuentra la velocidad cero del movimiento del brazo mecánico: df = [0 0]

Aplicando el algoritmo se obtienen los coeficientes de los trazadores:

t	x	a	b	c	d
0	0	0	0	1,5	0,5
1	2	2	4,5	3	-3,5
2	6	6	-	-	-

t	x	a	b	c	d
2	6	6	0	-4,5	1,5
3	3	3	-4,5	0	1,5
4	0	0	-	-	-

t	y	a	b	c	d
0	0	0	0	7,5	-3,5
1	4	4	4,5	-3	0,5
2	6	6	-	-	-

t	y	a	b	c	d
2	6	6	0	-7,5	3,5
3	2	2	-4,5	3	-0,5
4	0	0	-	-	-

Se confeccionan los siguientes trazadores:

```
function [xvst0] = xvst0(x)
  xvst0 = 0+0*(x-0)+1.5*(x-0).^2+0.5*(x-0).^3;
```

[3] Teorema 6.19 (página 398) del libro "Análisis numérico, séptima edición" Burden y Faires. Thomson Learning.

```
endfunction

function [xvst1] = xvst1(x)
  xvst1 = 2+4.5*(x-1)+3*(x-1).^2-3.5*(x-1).^3;
endfunction

function [xvst2] = xvst2(x)
  xvst2 = 6+0*(x-2)-4.5*(x-2).^2+1.5*(x-2).^3;
endfunction

function [xvst3] = xvst3(x)
  xvst3 = 3-4.5*(x-3)+0*(x-3).^2+1.5*(x-3).^3;
endfunction

function [yvst0] = yvst0(x)
  yvst0 = 0+0*(x-0)+7.5*(x-0).^2-3.5*(x-0).^3;
endfunction

function [yvst1] = yvst1(x)
  yvst1 = 4+4.5*(x-1)-3*(x-1).^2+0.5*(x-1).^3;
endfunction

function [yvst2] = yvst2(x)
  yvst2 = 6+0*(x-2)-7.5*(x-2).^2+3.5*(x-2).^3;
endfunction

function [yvst3] = yvst3(x)
  yvst3 = 2-4.5*(x-3)+3*(x-3).^2-0.5*(x-3).^3;
endfunction
```

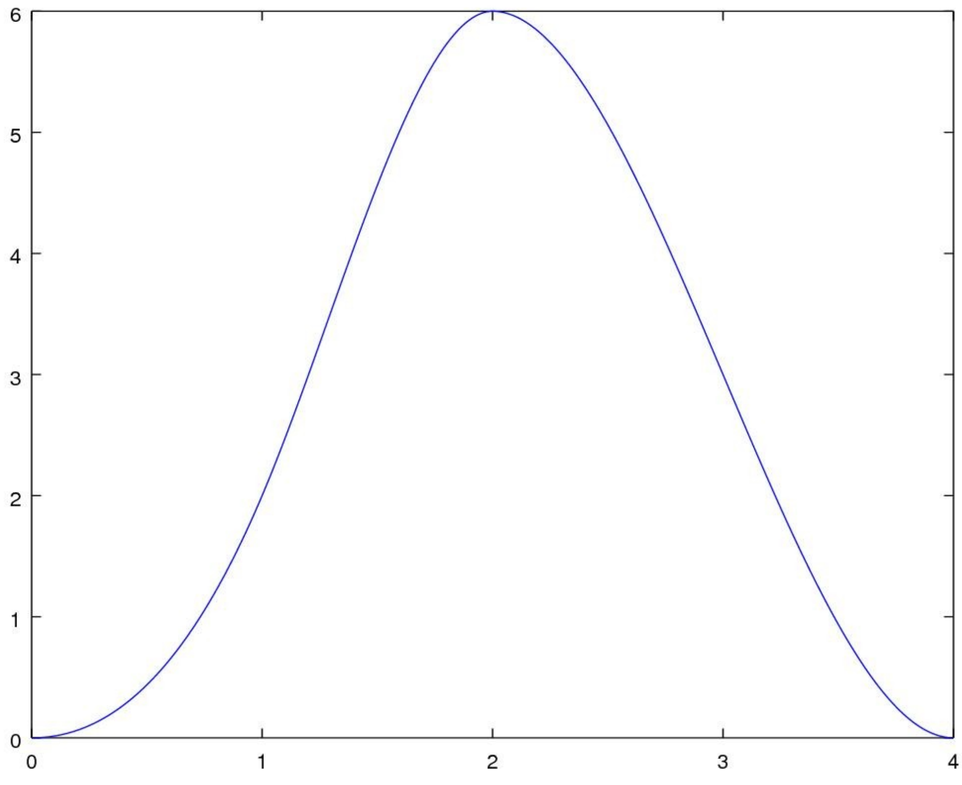

Figura 1: x vs t

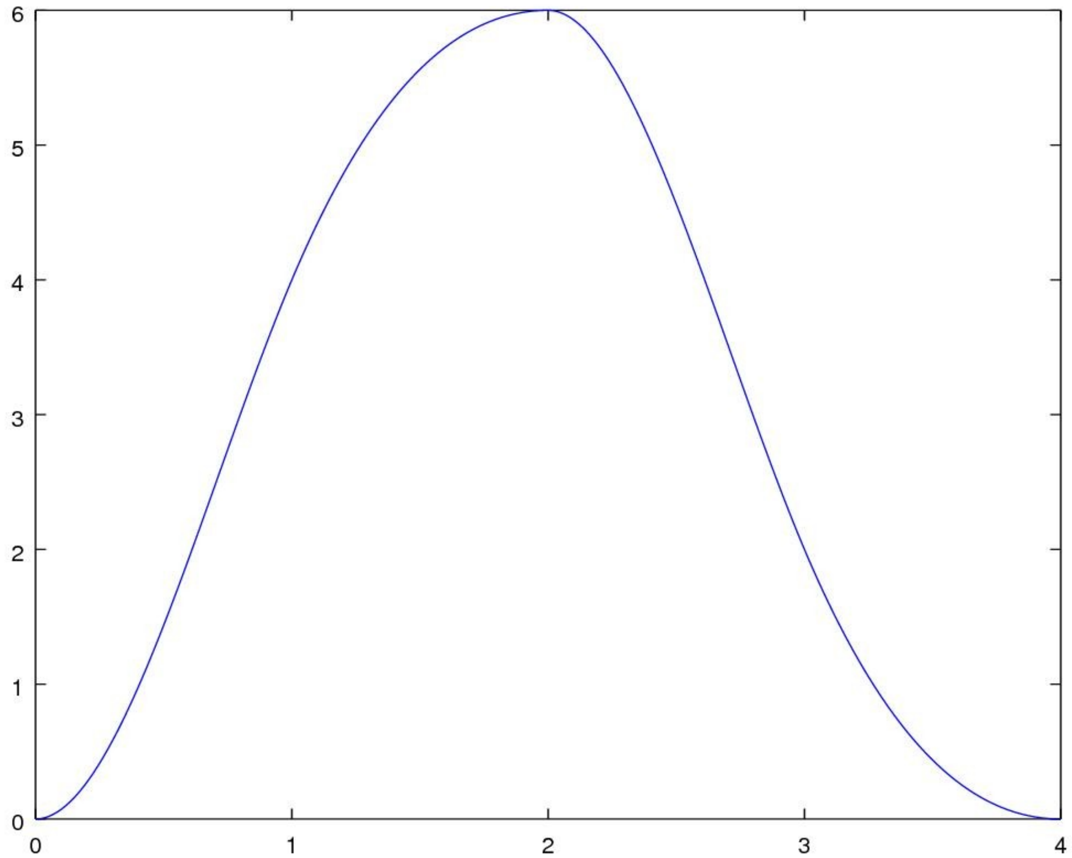

Figura 2: y vs t

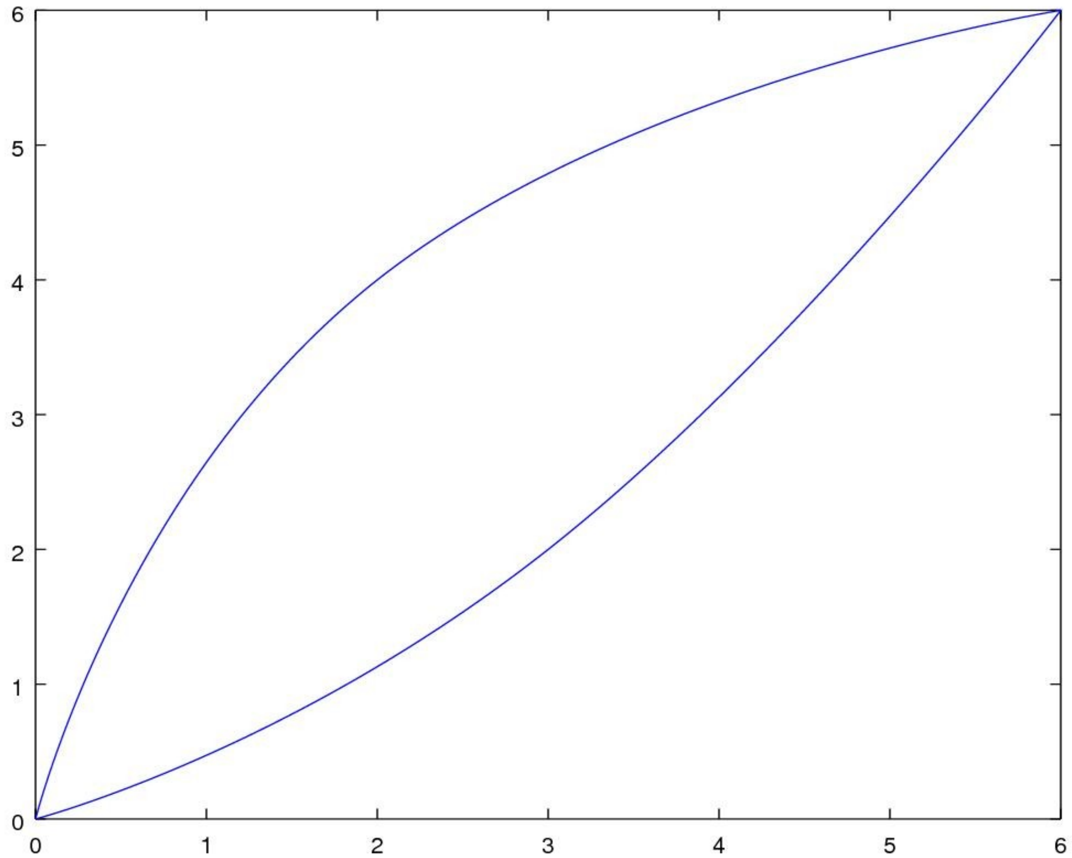

Figura 3: x vs y

REFERENCIAS

[1] Richard L. Burden & J.Douglas Faires, *Análisis Numérico*.