CÁLCULO NUMÉRICO
2023
ENTREGABLE 3

Alumno: Bargas, Santiago

Tema: *Interpolación y aproximación de Funciones. Diferenciación e integración numérica*

Enunciado:

Se desea conocer la trayectoria de una partıcula que se mueve en el plano, dada por la curva (x(t), y(t)). Para ello se cuenta con dos sensores, que determinan la posición de la partícula: uno mide la posición en el eje x cada 2 segundos, y otro en el eje y cada 1 segundo. En el inicio de las mediciones (t = 0), se sabe que la part´ıcula se encuentra a 2 cm del origen en la dirección x y se mueve a una velocidad π 2 cm/s en la direcci´on y, y despu´es de 6 segundos llega al origen a la misma velocidad inicial, pero en dirección negativa de y. Las mediciones de posición de los sensores se muestran en la siguiente tabla:

t[s]	**Sensor x[cm]**	**Sensor y[cm]**
0	2.0	0.0
1	-	1.0
2	1.5	0.0
3	-	-1.0
4	0.5	0.0
5	-	1.0
6	0.0	0.0

(a) Realice interpolaciones por spline cubicos sujetos para determinar expresiones de x(t) y de y(t) utilizando los datos de la tabla, y las velocidades inicial y final que describe el problema

(b) Grafique la trayectoria de la partícula y determine la posición y el vector velocidad a los 3 segundos.

(c) Recuerde que la longitud de la trayectoria de la partícula durante los T primeros segundos está dada por $\int_0^T \sqrt{x_1'(t)^2 + x_2'(t)^2}dt$.

Estimar la distancia recorrida por la partıcula durante el proceso. Dar el resultado con 6 cifras exactas.

INTRODUCCIÓN:

Es común utilizar técnicas de interpolación para obtener expresiones que describen el movimiento de una partícula entre puntos de medición mientras se estudia su trayectoria en el plano.

Los splines cúbicos sujetos son un tipo de interpolación que utiliza polinomios cúbicos para ajustar una curva suave a un conjunto de puntos de datos. La principal característica de los splines cúbicos sujetos es que se definen mediante condiciones de frontera adicionales en los extremos de la curva.

En el caso de los splines cúbicos sujetos, se establecen condiciones de frontera conocidas como condiciones sujetas o condiciones naturales. Estas condiciones imponen restricciones en la curva en los extremos, generalmente relacionadas con la derivada de la función en esos puntos

En este caso, los datos de la posición de la partícula son proporcionados por los sensores en intervalos de tiempo diferentes.

El objetivo es determinar la trayectoria de la partícula, representada por las funciones x(t) y y(t) mediante interpolaciones de splines cúbicos sujetos. Además, la información sobre las velocidades iniciales y finales de la partícula es proporcionada.

Para realizar las interpolaciones en este informe, se utilizó los datos de posición proporcionados y las velocidades inicial y final mencionadas. Luego, se graficó la trayectoria de la partícula y determinamos su posición y vector de velocidad en un momento determinado. Finalmente, se calculó la distancia que recorre la partícula a lo largo del período de tiempo que se consideró.

Podemos obtener una representación suave y continua de la trayectoria de la partícula utilizando métodos numéricos e interpolación. Esto nos permitirá comprender mejor su movimiento y realizar cálculos adicionales, como estimar la distancia recorrida.

A continuación, se mostrará como se realizaron las interpolaciones por splines cúbicos sujetos y todos los puntos planteados en el problema.

DESARROLLO:

Para comenzar, lo primero que debemos hacer a la hora de realizar un problema que se nos propone es entender bien el enunciado, reconocer los datos que nos brindan y tener claro el problema que nos piden resolver.

Para el inciso **a)** debemos determinar expresiones de x(t) e y(t) analizando la tabla propuesta y sus velocidades iniciales y finales. El enunciado nos dice que en el tiempo 0 la partícula tiene $\frac{\pi}{2} cm/s$ en el eje y al tiempo 0, y $-\frac{\pi}{2} cm/s$ en el eje y al tiempo 6. Por otra parte sabemos que la velocidad en x al tiempo inicial 0 segundos y al tiempo final 6 segundos es 0, por tanto podemos definir estos datos como:

$$df1x = \ 0\ (velocidad\ de\ la\ partícula\ en\ x\ al\ tiempo\ t\ =\ 0s)$$
$$df2x = \ 0\ (velocidad\ de\ la\ partícula\ en\ x\ al\ tiempo\ t\ =\ 6s)$$

$$df1y = \ \frac{\pi}{2} cm/s\ (velocidad\ de\ la\ partícula\ en\ x\ al\ tiempo\ t\ =\ 0s)$$
$$df2y = \ -\frac{\pi}{2} cm/s\ (velocidad\ de\ la\ partícula\ en\ x\ al\ tiempo\ t\ =\ 6s)$$

Estos datos conocidos nos ayudaran para poder utilizar la función ***funcion_spline.m,*** que devolverá **S** y **dS** que representan la interpolación y su derivada respectivamente.

Definimos al tiempo x, $tx = [0\ ,\ 2\ ,\ 4\ ,\ 6]$ ya que el sensor x[cm] mide cada 2 segundos.

Definimos con los valores proporcionados en la tabla al sensor x como $x = [2, 1.5, 0.5, 0]$

Definimos al tiempo y, $ty = [0, 1,\ 2\ , 3,\ 4\ , 5,\ 6\ ]$ ya que el sensor y[cm] mide cada 1 segundo.

Definimos con los valores proporcionados en la tabla al sensor y como $y = [0, 1.0, 0.0, -\ 1.0, 0, 1.0, 0.0]$.

Una vez definidos los datos se utilizó la función spline, para hallar la interpolación y su derivada, proporcionando los datos correspondientes los cuales son:

```
[Sx,dSx,ax]=funcion_spline(tx,x,df1x,df2x);

[Sy,dSy,ay]=funcion_spline(ty,y,df1y,df2y);
```

Se obtuvo Sx y Sy que son las funciones que representan la curva suave interpolada

La fórmula de los polinomios cúbicos es:

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

para cada j=0,1, … , n-1

Los valores de a, b, c, y d correspondientes para Sx son:

Intervalo	a	b	c	d
[0,2]	2	0	-0.15	0.0125
[2,4]	1.5	-0.45	-0.075	0.025
[4,6]	0.5	-0.45	0.075	0.0125

Los valores de a, b, c, y d correspondientes para Sy son:

Intervalo	a	b	c	d
[0,1]	0	1.57	-0.12	-0.45
[1,2]	1	-0.02	-1.47	0.49
[2,3]	0	-1.50	-0.01	0.50
[3,4]	-1	0.00	1.50	-0.50
[4,5]	0	1.50	-0.01	-0.49
[5,6]	1	0.02	-1.47	0.45

Para el inciso **b)** que nos pide graficar la trayectoria de la partícula y determinar su posición y vector velocidad a los 3 segundos, se comenzó graficando x(t) en función de t, el cual vemos a continuación:

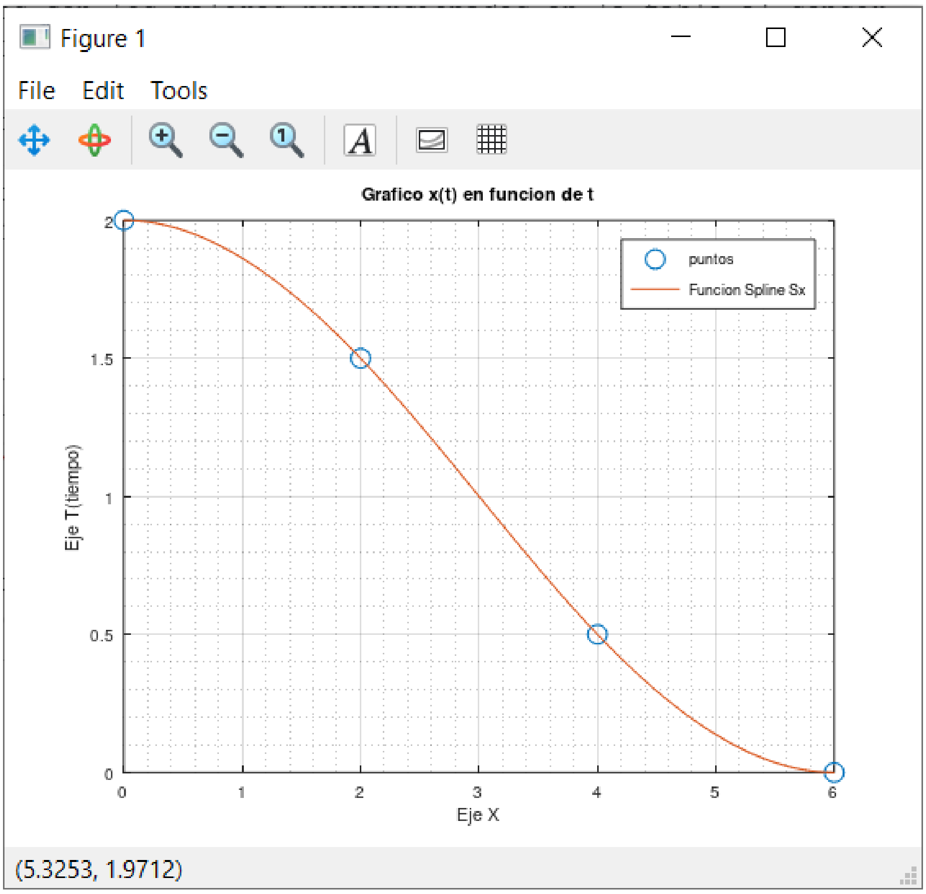

Figura 1. Gráfico x(t) en función de t

Para la Figura 1 se utilizó los siguientes valores:

$$plot(tdx, ax, 'o', xx, Sx(xx))$$

donde:

- $tdx = [0\ 2\ 4\ 6]$ valores de tiempo en el eje x que se muestran como puntos marcados en el gráfico.
- $ax = [ax, 0]$ valores de la variable x que corresponden a los puntos de datos, también se muestran como puntos marcados en el gráfico.
- $xx = linspace(0, 6, 201);$ valores de tiempo en el eje x que se utilizan para evaluar la función Sx
- $Sx(xx)$ Representa los valores de la función interpolada Sx evaluada en los puntos xx.

Luego se graficó y(t) en función de t, el cual vemos a continuación:

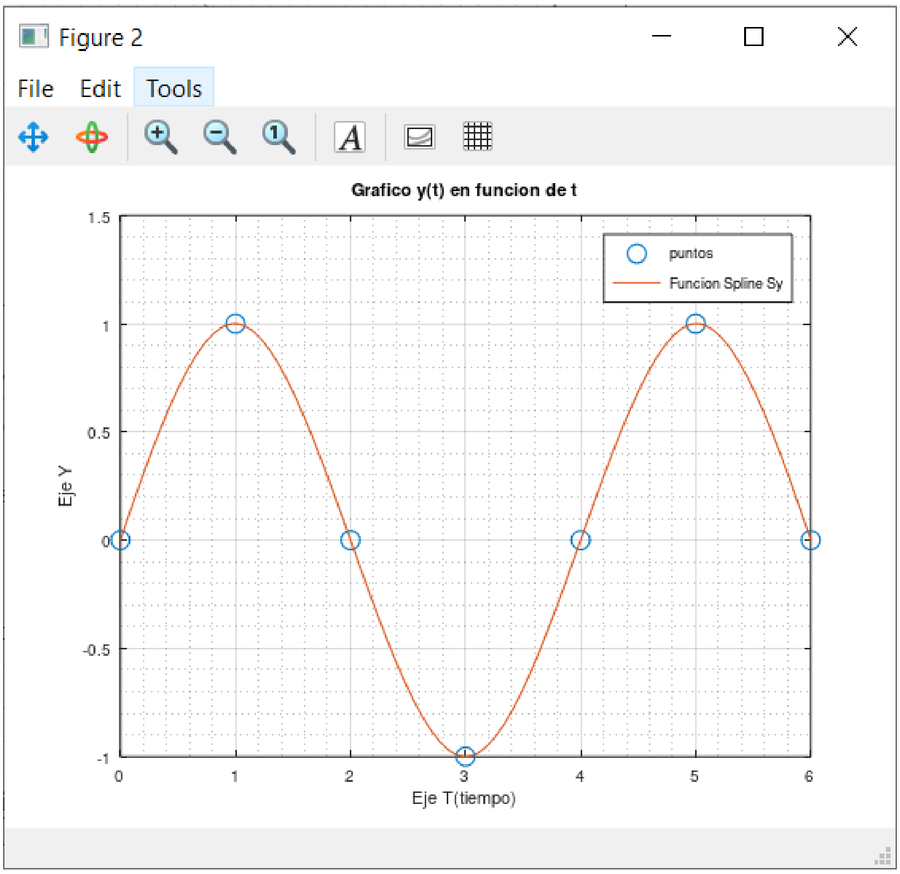

Figura 2. Gráfico y(t) en función de t

Para la Figura 2 se utilizó los siguientes valores:

$$plot(tdy, ay, 'o', yy, Sy(yy))$$

donde:

- $tdy = [0\ 1\ 2\ 3\ 4\ 5\ 6]$ valores de tiempo en el eje y que se muestran como puntos marcados en el gráfico.
- $ay = [ay, 0]$ valores de la variable y que corresponden a los puntos de datos, también se muestran como puntos marcados en el gráfico.
- $yy = linspace(0, 6, 201);$ valores de tiempo en el eje y que se utilizan para evaluar la función Sy
- $Sy(yy)$ Representa los valores de la función interpolada Sy evaluada en los puntos yy.

Luego se realizó el gráfico de X(t) en función de y(t), para la cual se hallaron los valores de x que faltaban en la tabla de los sensores. Para hallar los valores restantes se utilizó el polinomio spline Sx.

La tabla propuesta era

t[s]	Sensor x[cm]	Sensor y[cm]
0	2.0	0.0
1	-	1.0
2	1.5	0.0
3	-	-1.0
4	0.5	0.0
5	-	1.0
6	0.0	0.0

Entonces hallamos el vector X evaluando el polinomio Sx en los valores desconocidos del tiempo:

$X = [x(1)\ Sx(1)\ x(2)\ Sx(3)\ x(3)\ Sx(5)\ x(4)]$

el cual nos da como resultado $X = [2.0\ 1.86520\ 1.5\ 1\ 0.5\ 0.13750\ 0\ ]$ completando la tabla:

t[s]	Sensor x[cm]	Sensor y[cm]
0	2.0	0.0
1	1.86250	1.0
2	1.5	0.0
3	1.0	-1.0
4	0.5	0.0
5	0.13750	1.0
6	0.0	0.0

Para graficar en octave: $plot(Sx(xx), Sy(yy), 'b - ', X, y, 'r *- ');$

A continuación vemos la Figura 3 que representa X(t) en función de y(t):

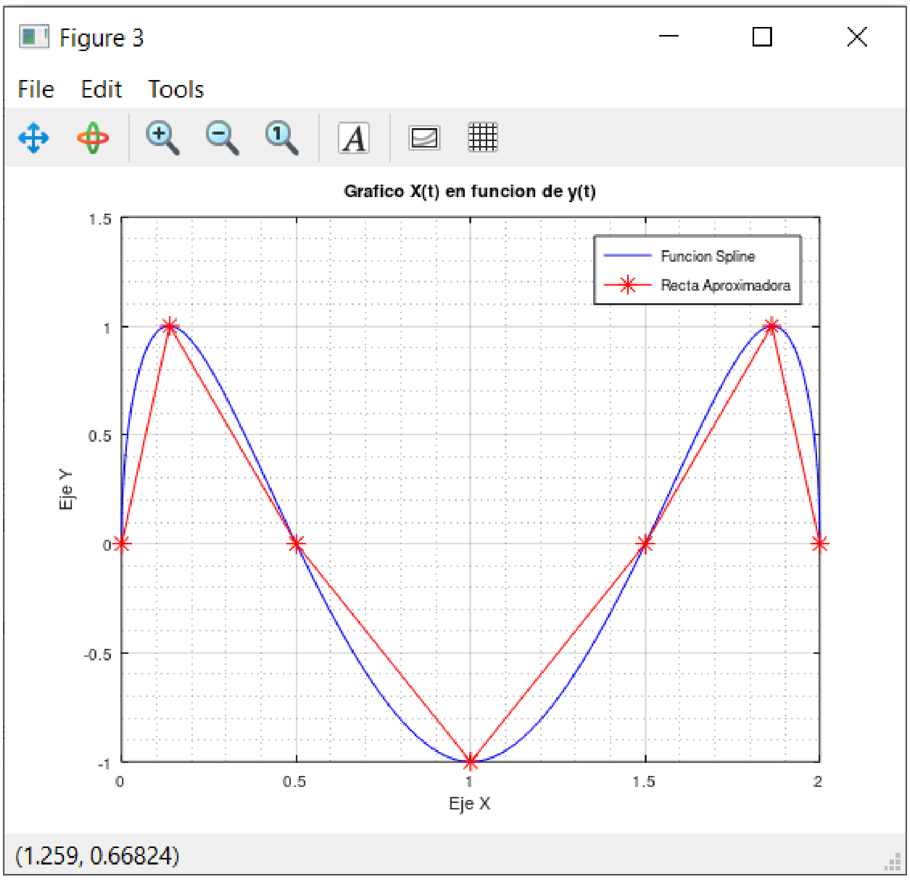

Figura 3. Gráfico X(t) en función de y(t)

Para determinar la posición a los 3 segundos se evaluó los polinomios splines Sx y Sy obtenidos en t=3s, y para obtener la velocidad resultante en 3 segundos se evaluó la derivada de los polinomios en t=3s.

La función funcion_spline.m arma el polinomio spline cúbico sujeto y además calcula su derivada, por ello es de tanta utilidad.

Como resultado:

La partícula se encuentra en x=1cm para t=3 segundos, y su componente de velocidad en x es: $-\ 0.5250\ cm/s$

La partícula se encuentra en y=-1cm para t=3 segundos, y su componente de velocidad en y es:$2.2205e^{-16}\ cm/s$

Y podemos calcular la velocidad resultante como:

$\sqrt{velocidad_x^{\ 2} + velocidad_y^{\ 2}}$ = $\sqrt{(-\ 0.5250)^2 + (2.2205)^2}$ = 0.52500 cm/s

$vector\ posicion\ al\ tiempo\ t\ =\ 3s:$ $\vec{i} + \vec{j}$

$vector\ velocidad\ al\ tiempo\ t\ =\ 3s:$ $-0.5250\vec{i} +\ 2.2205e^{16}\vec{j}$

Siguiendo, para el cálculo del punto **c)** se utilizó integración numérica, por lo cual se calculó el integrando, que es: $\sqrt{dSx^2 + dSy^2}$

Luego se integró entre 0 y T, donde 0 y T es el periodo a medir.

Para ello se utilizó el Método de La Cuadratura de Gauss. La cuadratura de Gauss es un método numérico que utiliza puntos y pesos específicos en un intervalo de integración para calcular una aproximación precisa de la integral de una función.

Los datos que utilizamos son:

- **a**,**b**: Límites inferior y superior del intervalo de integración.
- **L**: Número de subintervalos en los que se divide el intervalo [a, b].
- **n**: Número de puntos de Gauss a utilizar en cada subintervalo.

Se fijó un n=6 y se probó con distintas cantidades de subintervalos en los que se divide el intervalo, los resultados obtenidos fueron:

a=0 ; b=6; L=20; n=6; Distancia recorrida=6.514991382118884
a=0 ; b=6; L=40; n=6; Distancia recorrida=6.514997099454934
a=0 ; b=6; L=80; n=6; Distancia recorrida=6.514997683507586
a=0 ; b=6; L=160;n=6; Distancia recorrida=6.514997679106465

Observando los resultados obtenidos al variar la cantidad de subintervalos (L) en los que se divide el intervalo de integración, se puede concluir que a medida que se aumenta el número de subintervalos, la distancia recorrida por la partícula se acerca a un valor constante. Entonces, al aumentar la cantidad de subintervalos, se mejora la precisión de la aproximación de la integral numérica. Sin embargo, después de cierto punto, el aumento en la cantidad de subintervalos tiene un efecto insignificante en el resultado final, ya que la distancia recorrida se estabiliza alrededor de un valor específico.

Por lo tanto la distancia recorrida de la partícula durante el proceso es con L=160 es $6.514998\ cm$

Para obtener tambien se podrian haber utilizado otros métodos como:

Simpson compuesta: Es un método numérico que divide el intervalo de integración en subintervalos más pequeños y utiliza la regla de Simpson en cada uno de ellos para aproximar la integral de una función. La regla de Simpson es un método que utiliza una interpolación polinómica de segundo grado para aproximar la integral de una función, teniendo en cuenta la forma curva de la función y proporcionando una estimación más precisa de la integral.

Trapezoidal compuesta: Método numérico que divide el intervalo de integración en subintervalos y utiliza la regla del trapecio en cada uno de ellos para aproximar la integral de una función. La regla del trapecio es un método que aproxima la integral de una función utilizando una aproximación lineal mediante trapecios. Calcula el área de esos trapecios para estimar la integral en un intervalo dado.

Función quad de Octave: esta función se utiliza para aproximar numéricamente la integral definida de una función en un intervalo dado utilizando métodos de integración numérica.

Para finalizar podemos ver los incisos resueltos en octave y un resumen mostrado en la ventana de comandos de octave:

```
VALORES PARA LA TABLA
ans =

   2.000000000000000
   1.862500000000000
   1.500000000000000
   1.000000000000000
   0.500000000000000
   0.137500000000000
                   0

ans =

   0
   1
   0
  -1
   0
   1
   0

----------------------------------------------------------------------
posicion en x para t=3 segundos
posx3 = 1
componente velocidad en x
vx3 = -0.525000000000000
----------------------------------------------------------------------
posicion en y para t=3 segundos
posy3 = -1.000000000000000
componente velocidad en y
vy3 = 2.220446049250313e-16
----------------------------------------------------------------------
velocidad resultante en t=3 segundos
Velocidad_3 = 0.525000000000000
----------------------------------------------------------------------
Valor de la trayectoria a los 6 segundos: Gauss L=20 n=6:
Q2 = 6.514991382118884
----------------------------------------------------------------------
Valor de la trayectoria a los 6 segundos: Gauss L=40 n=6:
Q3 = 6.514997099454934
----------------------------------------------------------------------
Valor de la trayectoria a los 6 segundos: Gauss L=80 n=6:
Q4 = 6.514997683507586
----------------------------------------------------------------------
Valor de la trayectoria a los 6 segundos: Gauss L=160 n=6:
```

INCISO A:

```
%-------------------------------------------------
%----INCISO A---------
%x(t) en funcion de t;
tx=[0,2,4,6];
x=[2,1.5,0.5,0];
df1=df2=0; %pq en x no tiene velocidad
[Sx,dSx,ax]=funcion_spline(tx,x,df1,df2);
ax=[ax,0];

%y(t) en funcion de t;
ty=[0,1,2,3,4,5,6];
y=[0,1.0,0.0,-1.0,0,1.0,0.0];
df1=pi/2;
df2=-pi/2;
[Sy,dSy,ay]=funcion_spline(ty,y,df1,df2);
ay=[ay,0];
%-------------------------------------------------
```

INCISO B:

```
%------INCISO B----------

%grafico x(t) en funcion de t
tdx = [0 2 4 6];
xx = linspace(0,6,201);
figure(1)
plot(tdx,ax,'o',xx,Sx(xx));
title('Grafico x(t) en funcion de t')
xlabel('Eje X');
ylabel('Eje T(tiempo)');
legend('puntos','Funcion Spline Sx');
grid on;
grid minor;

%Grafico y(t) en funcion de t
tdy=[0 1 2 3 4 5 6 ];
yy=linspace(0,6,201);
figure(2)
plot(tdy,ay,'o',yy,Sy(yy))
title('Grafico y(t) en funcion de t');
xlabel('Eje T(tiempo)');
ylabel('Eje Y');
legend('puntos','Funcion Spline Sy');
grid on;
grid minor;

%graficamos X(t) en funcion y(t)
X=[x(1) Sx(1) x(2) Sx(3) x(3) Sx(5) x(4)];
figure(3)
%plot(Sx(xx),Sy(yy),'b-',X,y,'r*-');
plot(Sx(xx),Sy(yy),'b-',X,y,'r*-');
title('Grafico X(t) en funcion de y(t)')
xlabel('Eje X');
ylabel('Eje Y');
legend('Funcion Spline','Recta Aproximadora');
grid on;
grid minor;
disp("VALORES PARA LA TABLA")
X'
y'
```

```
%calculo para el tiempo de 3 segundos

disp("-----------------------------------------------
disp('posicion en x para t=3 segundos');
posx3=Sx(3)
disp('componente velocidad en x');
vx3=dSx(3)

disp("-----------------------------------------------
disp('posicion en y para t=3 segundos');
posy3=Sy(3)
disp('componente velocidad en x');
vy3=dSy(3)
disp('velocidad resultante en t=3 segundos')
Velocidad_3=sqrt(vx3^2+vy3^2)
```

INCISO C:

```
%-----INCISO C--------
f=@(t) sqrt(dSx(t).^2+dSy(t).^2);

%Aplicamos la integral para calcular la trayectoria
%Usamos Gauss-Legendre

disp("-------------------------------------------------------------------
a=0 ; b=6; L=20; n=6;
Q2=cuad_gauss_c(f,a,b,L,n);
disp('Valor de la trayectoria a los 6 segundos: Gauss L=20 n=6:')
Q2

disp("-------------------------------------------------------------------
a=0 ; b=6; L=40; n=6;
Q3=cuad_gauss_c(f,a,b,L,n);
disp('Valor de la trayectoria a los 6 segundos: Gauss L=40 n=6:')
Q3

disp("-------------------------------------------------------------------
a=0 ; b=6; L=80; n=6;
Q4=cuad_gauss_c(f,a,b,L,n);
disp('Valor de la trayectoria a los 6 segundos: Gauss L=80 n=6:')
Q4

disp("-------------------------------------------------------------------
a=0 ; b=6; L=160; n=6;
Q5=cuad_gauss_c(f,a,b,L,n);
disp('Valor de la trayectoria a los 6 segundos: Gauss L=160 n=6:')
Q5
```

Codigos usados:

```
function [S,dS,a]=funcion_spline(x1,y1,df1,df2)

[a,b,c,d] = Spline_Cubico_Sujeto(x1,y1,df1,df2);

S=@(x) a(1)*(x==x1(1));

M=[d' c' b' a'];
dM=[];
dS= @(x) 0;
for i=1:length(x1)-1
  dM=[dM;polyder(M(i,:))];
  S=@(x) S(x) +...
  polyval(M(i,:),x-x1(i)).*(x>x1(i)).*(x<=x1(i+1));
  dS=@(x) dS(x) +...
  polyval(dM(i,:),x-x1(i)).*(x>x1(i)).*(x<=x1(i+1));
endfor
```

```
function[a, b, c, d] = Spline_Cubico_Sujeto(x, y, dy0, dyn)
n = length(x);
h(1:n-1) = x(2:n)-x(1:n-1);
v = zeros(n,1);
v(1) = 3 * ( (y(2)-y(1) ) / h(1) - dy0 ); #paso 2
v(2:n-1) = 3 * ( (y(3:n)-y(2:n-1) ) ./ h(2:n-1) -( y(2:n-1) - y(1:n-2) ) ./ h(1:n-2));#paso3
v(n) = 3*( dyn - ( y(n) - y(n-1) ) / h(n-1) ); # paso 2
l(1) = 2*h(1);
u(1) = 0.5;
z(1) = v(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) = ( v(i) - h(i-1) * z(i-1) ) / l(i);
endfor
l(n) = h(n-1) * (2-u(n-1));
z(n) = ( v(n) - h(n-1)*z(n-1) ) / l(n);
c(n) = z(n);
for i = n-1:-1:1
  c(i) = z(i) - u(i) * c(i+1);
  b(i) = ( y(i+1)-y(i) ) / h(i) - h(i) * ( c(i+1) + 2 * c(i) ) / 3;
  d(i)=(c(i+1)-c(i))/(3*h(i));
endfor
  c = c(1:n-1);
  a = y(1:n-1);
endfunction
```

```
function Q=cuad_gauss_c(f,a,b,L,n)
[xg,w]=gauss_xw(n);
x=linspace(a,b,L+1);
h=(b-a)/L;
Q=0;
for i=1:L
t=h/2*(xg+1)+x(i);
Q+=h/2*(w'*f(t));
endfor
```

CONCLUSIÓN:

En resumen, el desarrollo realizado implica la resolución de un problema relacionado con una partícula en movimiento mediante el uso de técnicas de interpolación y técnicas de integración numérica.

Se utilizaron los datos proporcionados en una tabla para encontrar las expresiones de las funciones x(t) e y(t) en el inciso a).

Se utilizaron los polinomios spline obtenidos para crear gráficos de las funciones x(t) e y(t) en función del tiempo en el inciso b). Además, se representó la trayectoria de la partícula en un plano cartesiano mediante la representación de X(t) en función de y(t).

La distancia recorrida por la partícula durante un periodo de tiempo determinado se calculó utilizando el método numérico de integración conocido como cuadratura de Gauss en el inciso c).

Se realizaron verificaciones adicionales utilizando técnicas de integración numérica alternativas, como la composición Simpson y la composición trapezoidal, y se obtuvieron resultados consistentes.

En conclusión, el desarrollo del problema permitió la determinación de las expresiones de las funciones de posición de la partícula, la creación de gráficos de su trayectoria, el cálculo de la distancia recorrida mediante métodos de integración numérica y la validación de los resultados obtenidos mediante una variedad de técnicas. Esto mejora la comprensión del movimiento de la partícula y cómo se comporta en el tiempo.