

---

# CÁLCULO NUMÉRICO

---

## **Entregable N° 3**

Métodos interpolantes, diferenciación e integración

---

Giuliana Cagnola

2023

---

---

## Contenido

INTRODUCCIÓN .....	3
PALABRAS CLAVE .....	3
MARCO TEÓRICO .....	3
CONSIGNAS.....	4
DESARROLLO.....	5
Análisis para X.....	6
Análisis para Y.....	8
Análisis conjunto para x e y .....	10
Longitud de arco .....	12
CONCLUSIÓN .....	12
ANEXO: Script .....	13

## INTRODUCCIÓN

Dado un conjunto discreto y finito de datos, frecuentemente es necesario conocer valores dentro de esos puntos, de los cuales no se tiene información. A través de los distintos métodos interpolantes es posible calcular ese valor con tanta precisión como sea necesario (siempre teniendo en cuenta los errores de redondeo y de truncamiento).

La interpolación de funciones permite construir una función continua que se ajuste a los valores conocidos, de manera que pueda utilizarse para calcular valores intermedios o extrapolados con mayor precisión. Esto es especialmente útil cuando se tienen datos discretos y se busca obtener una aproximación suave y continua de la función subyacente.

Por otra parte, a partir de esas funciones obtenidas podemos pensar en aplicar distintos métodos de diferenciación e integración que permitan construir numéricamente la derivada y la integral de la misma cuando el abordaje hacia la solución analítica resulta complejo o no se puede resolver por métodos matemáticos conocidos.

**PALABRAS CLAVE:** Interpolación, Spline cúbico sujeto, trayectoria, mediciones.

## MARCO TEÓRICO

Dada una serie de  $n+1$  datos  $(x_0, x_1, \dots, x_n)$  con sus correspondientes  $n+1$  valores  $(y_0, y_1, \dots, y_n)$  podemos pensar construir una función que los represente. A partir de esto podemos construir un polinomio  $P(x)$  que los aproxime:

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + a_2 x^2 + a_1 x + a_0$$

Si la función  $f(x)$  es la función de interés en relación a los puntos  $(x_i, y_i)$ , podemos encontrar un polinomio tal que su error al representar a la función sea tan pequeño como se desee, es decir  $|f(x) - P(x)| \leq \varepsilon \forall x \in [a, b]$ .

Como los polinomios presentan muchas variaciones a medida que aumenta el grado, las aproximaciones en puntos alejados al valor evaluado pueden diferir mucho del valor real. Para evitar esto podemos descomponer el intervalo en subintervalos y aproximar la función en cada segmento.

Para cada subintervalo  $[x_{i-1}, x_i]$   $S(x)$  es un polinomio de grado  $\leq k$  que tiene  $k-1$  derivadas continuas en  $[x_0, x_n]$ . Esto da lugar a los *splines* (o trazadores)

Un spline cúbico tiene la forma

$$\begin{cases} S_0(x), & x \in [x_0, x_1) \\ S_1(x), & x \in [x_1, x_2) \\ \vdots & \\ S_n(x), & x \in [x_{n-1}, x_n) \end{cases}$$

Cada  $S_i(x)$  es un polinomio cúbico

Podemos considerar un spline cúbico con condiciones de frontera libre, es decir  $S''(x_0)=0$  y  $S''(x_n)=0$  o un spline cúbico con condiciones de frontera sujeta:  $S'(x_0)=f'(x_0)$  y  $S'(x_n)=f'(x_n)$ .

A partir de todos estos datos y mediante procesos algebraicos podemos llegar a un sistema de ecuaciones de la forma  $Ax=b$ , el cual tiene única solución en el intervalo.

A partir de las funciones obtenidas podemos pensar en derivar e integrar numéricamente. El método de integración compuesta de Newton-Cotes tiene la forma  $I = \sum_{i=0}^n a_i f(x_i)$ , donde  $a_i = \int_a^b L_{n,i}(x) dx$ , siendo  $L_{n,i}(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x-x_j)}{(x_i-x_j)}$  el polinomio de

Lagrange de orden  $n$  asociado al punto  $i$ . Para usar este método de integración debemos ponderar los valores con pesos específicos, los cuales son elegidos de manera que la suma ponderada de los valores de la función en los nodos se aproxime a la integral exacta.

Todos estos métodos conllevan errores asociados en el resultado obtenido por aproximar numéricamente. El **error de redondeo** es causado por la limitación de la precisión numérica de la computadora, mientras que el **error de truncamiento** está relacionado con las aproximaciones y simplificaciones realizadas en los métodos numéricos. Para el caso de las fórmulas compuestas el error asociado al método no depende del tamaño del intervalo.

## CONSIGNAS

Se desea conocer la trayectoria de una partícula 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 partícula se encuentra a 2 cm del origen en la dirección  $x$  y se mueve a una velocidad  $\pi/2$  cm/s en la dirección  $y$ , y después 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

- Realice interpolaciones por spline cúbicos sujetos para determinar expresiones de  $\mathbf{x}(t)$  y de  $\mathbf{y}(t)$  utilizando los datos de la tabla, y las velocidades inicial y final que describe el problema.
- Grafique la trayectoria de la partícula y determine la posición y el vector velocidad a los 3 segundos.
- Recuerde que la longitud de la trayectoria de la partícula durante los  $T$  primeros segundos está dada por  $\int_0^T \sqrt{x_1'^2(t) + x_2'^2(t)}$ . Estimar la distancia recorrida por la partícula durante el proceso. Dar el resultado con 6 cifras exactas.

## DESARROLLO

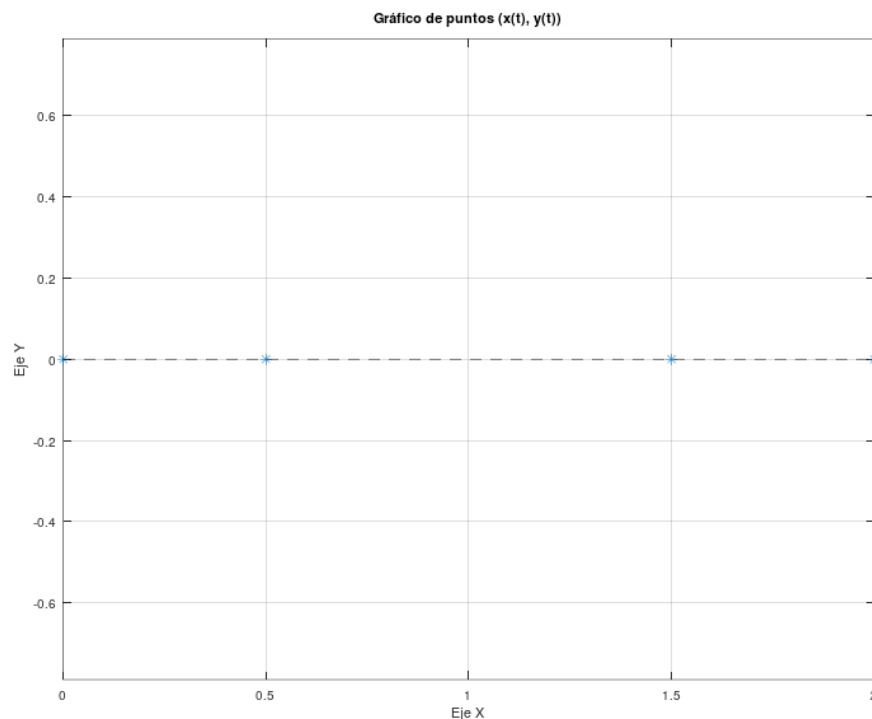
Este problema plantea una partícula que se desplaza a lo largo del tiempo en el plano  $XY$ . Esto genera una trayectoria que está dada por la curva  $(\mathbf{x}(t), \mathbf{y}(t))$ . Si bien podemos suponer que la curva es continua, los valores que se tienen para su movimiento son los correspondientes a los tiempos equiespaciados  $t_i$ ,  $i=0,1,\dots,6$  segundos, es decir, se tienen 7 momentos por los cuales se conocen una o ambas coordenadas de la partícula.

Para las mediciones en el eje  $x$  el sensor que mide la posición toma las mediciones cada 2 segundos, por lo tanto solo se tienen los valores de  $x$  para  $t_i$ ,  $i=0, 2, 4$  y  $6$ .

Para las mediciones de  $y$  el sensor mide cada un segundo, por lo tanto se tienen los 7 valores correspondientes a los 7 instantes medidos.

En el instante inicial ( $t=0$ ) la partícula se encuentra en el punto  $P_0 = (2,0)$  con una velocidad de  $\pi/2$  cm/s en la dirección  $y$ . Para el instante final ( $t=6$ ) la posición de la partícula es  $P_6 = (0,0)$  y la velocidad es  $-\pi/2$  cm/s, es decir, la misma magnitud que la velocidad en  $t=0$ .

A continuación se muestran los puntos correspondientes de los cuales se tiene la medición para ambos ejes. Como se puede ver, estos se encuentran sobre el eje  $x$ .

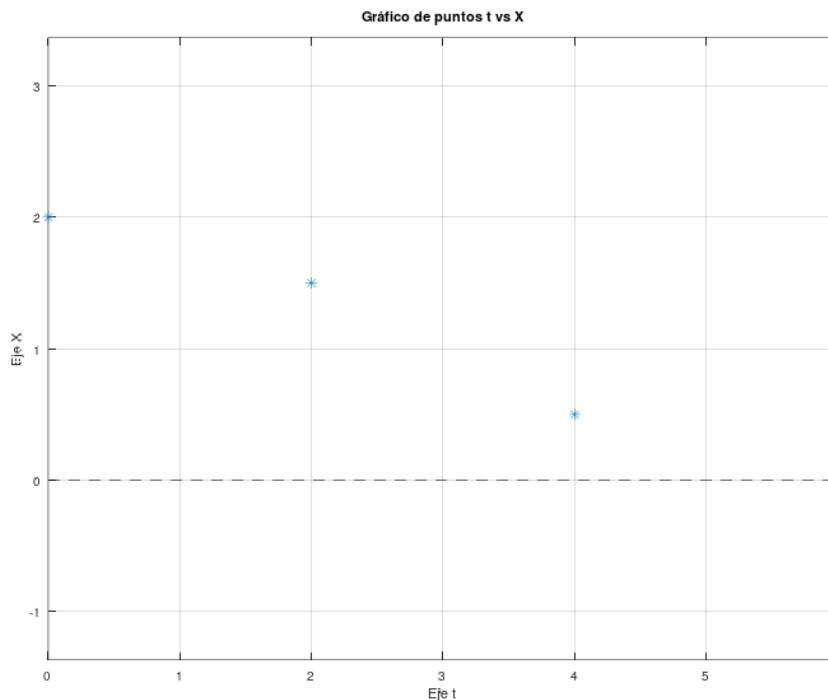


**Figura 1.** Gráfico de puntos  $(x(t), y(t))$

A partir de nuestras variables, podemos analizar la trayectoria que recorre la partícula para cada una de ellas respecto de nuestra variable independiente, en este caso el tiempo  $t$ .

### Análisis para X

Se considerarán los instantes de tiempo  $t = [0, 2, 4, 6]$  con sus respectivas posiciones  $x = [2, 1.5, 0.5, 0]$ . Podemos ver entonces que la partícula arranca a 2cm del origen y se va acercando hasta llegar al mismo.



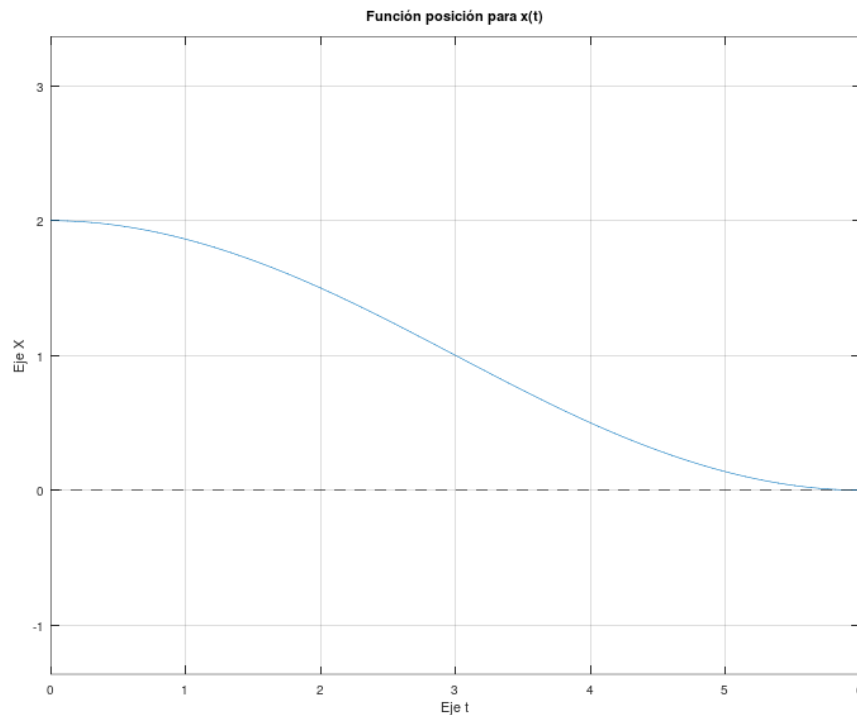
**Figura 2.** Gráfico de puntos de la posición en x respecto del tiempo

También podemos ver que las velocidades, es decir, la derivada de nuestra función posición tienen una componente completamente vertical, por lo que las derivadas en los puntos inicial y final son ambas 0. Con estos datos llamamos a nuestra función **[Sx, dSx]=función\_spline(t, x, 0, 0)** que a su vez invoca a la función **[a, b, c, d] = SplineCubicoSujeto(x, y, dy0, dyn)** y obtenemos las expresiones de las funciones posición y velocidad sobre el eje x respecto del tiempo.

Evalutando dichas funciones en  $t=3$  se obtiene:

- $x_3 = Sx(3) = 1.0000$
- $vx_3 = dSx(3) = -0.5250$

Es decir, a los 3 segundos la partícula se encuentra a 1cm del origen sobre el eje **x** y se desplaza con una velocidad instantánea de -0.5250 cm/s

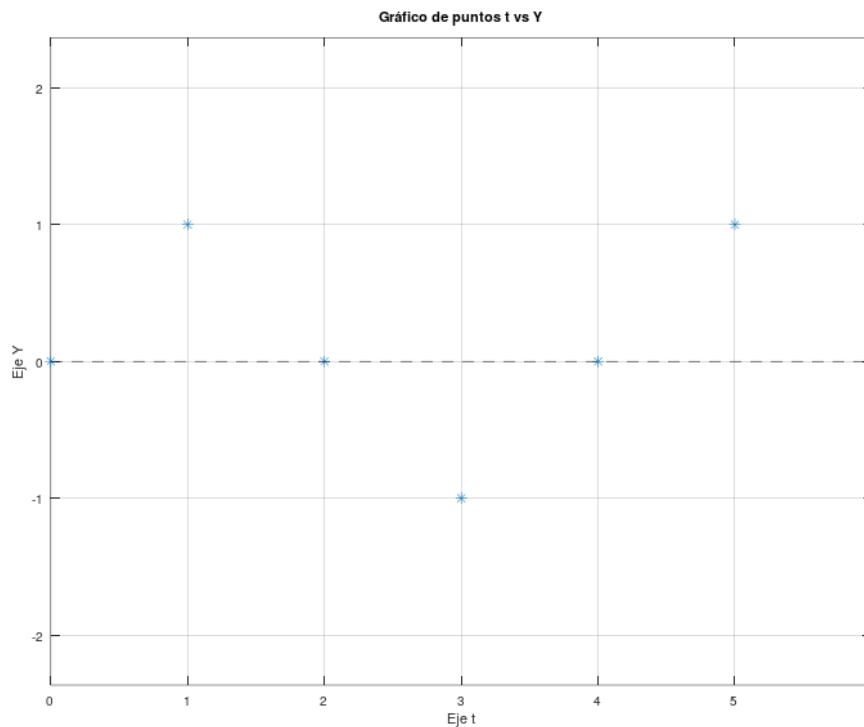


**Figura 3.** Trayectoria en el eje x obtenida por interpolación por Spline cúbico sujeto

### Análisis para Y

Para el análisis de la posición de la partícula sobre el eje de las abscisas se tienen los valores  $t = [0:6]$  y las posiciones  $y = [0 \ 1 \ 0 \ -1 \ 0 \ 1 \ 0]$ . A partir de los dichos valores para cada uno de los instantes de tiempo en los que el sensor midió la posición podemos inferir que la gráfica será una senoidal (o similar) ya que los valores son cíclicos para nuestro intervalo de tiempo.





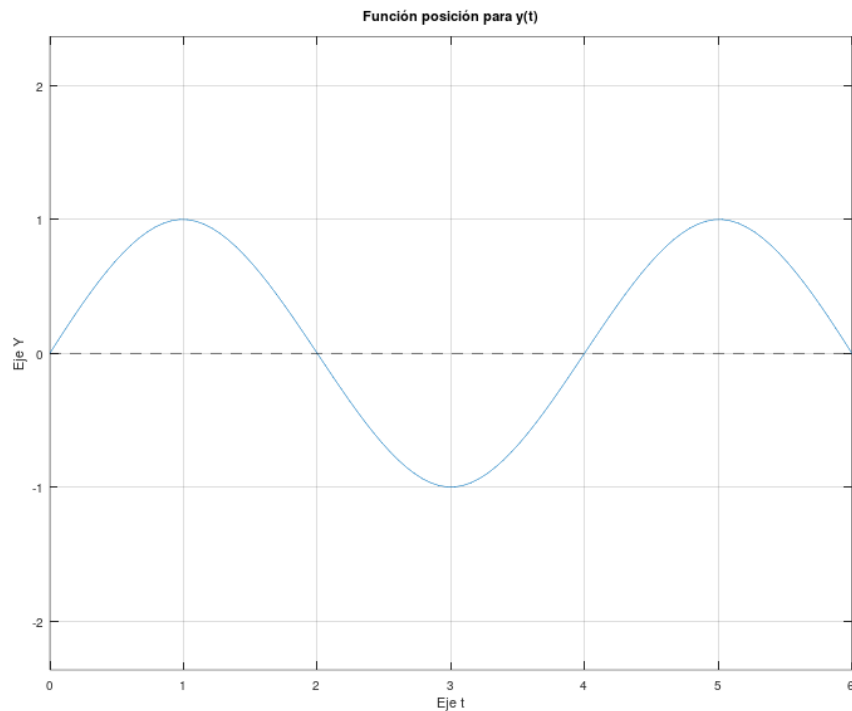
**Figura 4.** Gráfico de puntos de la posición en y respecto del tiempo

De acuerdo con el enunciado la velocidad inicial es  $\pi/2$  cm/s al principio y  $-\pi/2$  cm/s al final. Con estos datos nuevamente llamamos a la función **[Sy, dSy] = función\_spline(t, y, pi/2, -pi/2)** para obtener las funciones posición y velocidad en el eje **y** respecto del tiempo.

Las funciones evaluadas en  $t=3$  arrojan los siguientes resultados:

- $y_3 = Sy(3) = -1.0000$
- $vy_3 = dSy(3) = 2.2204e-16$

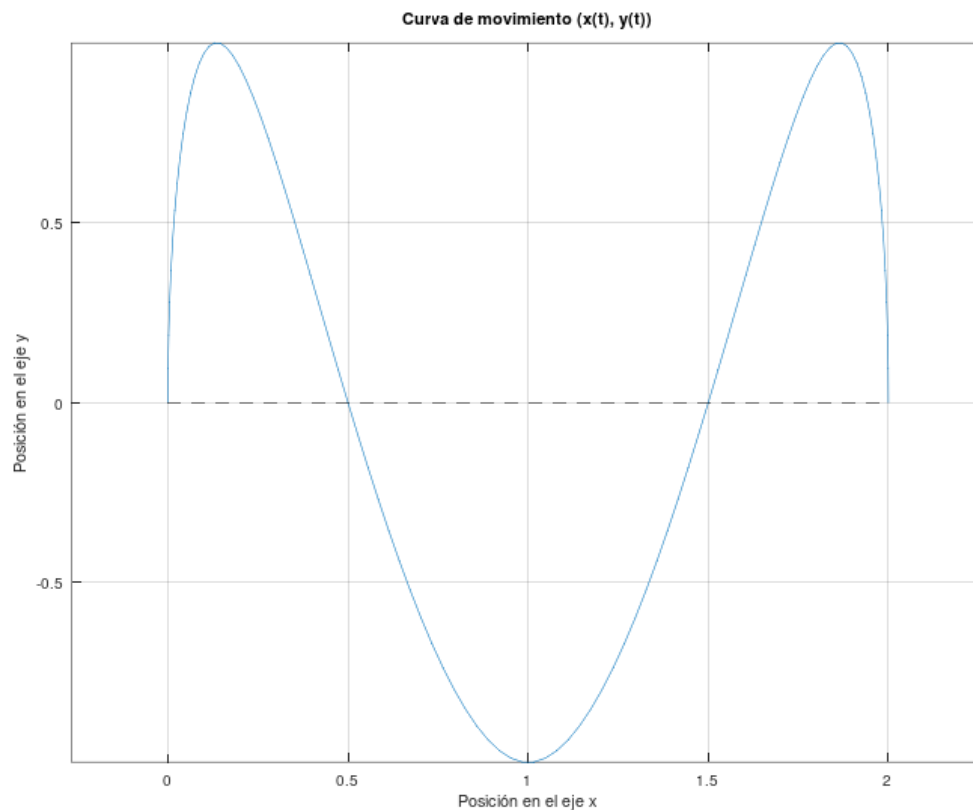
Es decir, a los 3 segundos de comenzadas las mediciones la posición en **y** es -1 cm, lo cual coincide con el valor reportado por el sensor en dicho instante, y la velocidad con la cual se desplaza es un valor muy cercano a 0, por lo que podemos considerarlo como el *épsilon de máquina* y asumir que en dicho valor la función asume un mínimo, ya que además la derivada evaluada en valores cercanos a la derecha y a la izquierda de  $t=3$  cambia de signo de negativo a positivo. Toda esta deducción se condice con lo expresado en el gráfico, por lo que podemos considerar que nuestra función posición es una buena representación de la trayectoria de la partícula para **y(t)**



**Figura 5.** Trayectoria en el eje  $y$  obtenida por interpolación por Spline cúbico sujeto

### Análisis conjunto para $x$ e $y$

A partir de los análisis individuales realizados para  $x$  e  $y$  podemos pensar en el movimiento que realiza la partícula a partir de la curva paramétrica que describe la trayectoria de la misma. Si consideramos las variables  $x = Sx(t)$  y  $y = Sy(t)$ , podemos graficar la curva para corroborar que los puntos medidos por los sensores se condicen con la gráfica de nuestra curva.



**Figura 6.** Trayectoria de la partícula a partir de las funciones  $x(t)$  e  $y(t)$

Cabe recordar que, como se ha explicado, la partícula arranca el movimiento en  $(2, 0)$  y se va acercando hacia el origen, por lo que la lectura de la misma a lo largo del tiempo es de derecha a izquierda.

Podemos ver que en  $t=0$  el movimiento arranca en  $(2,0)$ , y el vector velocidad es completamente vertical, lo cual se condice con las derivadas  $\mathbf{vx0} = \mathbf{dSx(0)} = 0$  y  $\mathbf{vy0} = \mathbf{dSy(0)} = 1.5708 \approx \pi/2$  con 4 decimales de precisión.

En  $t=1$  el sensor midió  $y=1$ , al igual que el valor que se muestra en la figura 6. Para  $x$  no se tenían mediciones, pero podemos calcularlas a partir de nuestro polinomio interpolante. El resultado es  $x1 = Sx(1) = 1.8625$ .

Otro punto a considerar será  $t=4$ , para el cual se tienen las mediciones de ambos sensores. A partir de esto queremos corroborar que las funciones sean coincidentes en dicho valor. Las variables  $x4 = Sx(4) = 0.5000$  y  $y4 = Sy(4) = 0$  se corresponden perfectamente con la medición obtenida.

Finalmente, para  $t=6$  obtenemos  $x_6 = Sx(6) = 5.5511e-17 \approx 0$ ,  $vx_6 = dSx(6) = 0$ ,  $y_6 = Sy(6) = 2.2204e-16 \approx 0$  y  $vy_6 = dSy(6) = -1.5708 \approx \pi/2$ . Todos estos valores son coincidentes con los datos del problema, por lo que la relación funcional de nuestro polinomio interpolante con dichos valores es muy buena.

### Longitud de arco

A partir de las funciones obtenidas  $dSx(t)$  y  $dSy(t)$  podemos pensar en calcular cuánto mide la trayectoria que realizó la partícula durante su movimiento. Dicha curva se representa por la función  $L=@(t) \text{sqrt}(dSx(t).^2 + dSy(t).^2)$

Se desea estimar la trayectoria con una precisión de 6 cifras, por lo que podemos pensar en una tolerancia de  $1e-5$  entre iteraciones sucesivas. Una vez que logremos que la diferencia entre la iteración actual  $i$  y la anterior  $i-1$  sea menor a dicha tolerancia, lograremos las 6 cifras de precisión que pide el enunciado

A partir de nuestra función  $L$  y el intervalo de integración de 0 a 6 llamamos a la función  $Q = \text{intNCcompuesta}(L, 0, 6, 12, 5)$  que realiza integración de Newton-Cotes compuesta particionando el intervalo en  $N$  subintervalos y tomando  $n=5$  puntos dentro de cada uno. Se ha tomado  $N=12$  para considerar al intervalo de tiempo cada 0.5 segundos y así obtener 5 decimales exactos.

El resultado de la trayectoria de la partícula es  $Q = 6.51493$  cm.

### CONCLUSIÓN

La interpolación e integración numérica son técnicas fundamentales en el campo de análisis numérico que nos permiten aproximar y calcular valores de funciones en intervalos específicos. En este informe, se exploraron diferentes conceptos y métodos relacionados con la interpolación e integración numérica en base a las consignas y a teoría que fundamenta a cada uno de los métodos.

En cuanto a interpolación se utilizó un spline de grado 3, sujeto a las derivadas en los puntos final e inicial para cada variable. Estos métodos resultan útiles para estimar valores de una función en puntos intermedios a partir de datos discretos.

En relación a la integración numérica, se utilizó el método de Newton-Cotes compuesta, ya que permite aproximar la integral con un bajo tiempo de ejecución y con un error apropiado al contexto.

Se implementaron varios métodos en Octave para ilustrar los conceptos y técnicas de interpolación e integración numérica. Estos métodos incluyeron la interpolación polinómica por splines cúbicos sujetos a las derivadas de cada una de las variables, así como el método de integración numérica de Newton-Cotes compuesta. Cada método se diseñó y utilizó con el propósito de proporcionar una comprensión práctica de los fundamentos teóricos y las aplicaciones prácticas de la interpolación y la integración numérica.

## ANEXO: Script

```
#Giuliana Cagnola
#CÓDIGO TP3
#-----
#Gráfico de puntos X vs Y
figure(1)
x = [2 1.5 0.5 0];
y = [0 0 0 0];
plot(x, y, '*')
hold on
plot([min(x), max(x)], [0, 0], 'k--')
xlabel('Eje X')
ylabel('Eje Y')
title('Gráfico de puntos (x(t), y(t))')
grid on
axis equal
hold off

#Gráfico de puntos t vs X
figure(2)
t = [0 2 4 6];
plot(t,x,'*')
hold on
plot([min(t), max(t)], [0, 0], 'k--')
xlabel('Eje t')
ylabel('Eje X')
title('Gráfico de puntos t vs X')
grid on
axis equal

#Función posición para X(t)
[Sx,dSx] = funcion_spline(t, x, 0, 0)

#Posición a los 3 segundos
x3 = Sx(3)

#Velocidad a los 3 segundos
vx3 = dSx(3)
```

```
#Gráfico de la función X(t)
figure(3)
t=linspace(0,6,200);
plot(t, Sx(t))
hold on
plot([min(t), max(t)], [0, 0], 'k--')
xlabel('Eje t')
ylabel('Eje X')
title('Función posición para x(t)')
grid on
axis equal

#-----

#Gráfico de puntos t vs Y
figure(4)
t = [0:6];
y = [0 1 0 -1 0 1 0];
plot(t,y,'*')
hold on
plot([min(t), max(t)], [0, 0], 'k--')
xlabel('Eje t')
ylabel('Eje Y')
title('Gráfico de puntos t vs Y')
grid on
axis equal

#Función posición para Y(t)
[Sy,dSy] = funcion_spline(t, y, pi/2, -pi/2)

#Posición a los 3 segundos
y3 = Sy(3)

#Velocidad a los 3 segundos
vy3 = dSy(3)

#Gráfico de la función Y(t)
figure(5)
t=linspace(0,6,200);
plot(t, Sy(t))
hold on
plot([min(t), max(t)], [0, 0], 'k--')
xlabel('Eje t')
ylabel('Eje Y')
title('Función posición para y(t)')
grid on
axis equal

#Gráfico de la curva (X(t), Y(t))
figure(6)
```

```
x = Sx(t);
y = Sy(t);
plot(x, y)
hold on
plot([min(x), max(x)], [0, 0], 'k--')
xlabel('Posición en el eje x')
ylabel('Posición en el eje y')
title('Curva de movimiento (x(t), y(t))')
grid on
axis equal
```

```
#-----
#Análisis en conjunto para x e y
x0 = Sx(0)
vx0 = dSx(0)
y0 = Sy(0)
vy0 = dSy(0)
```

```
x1 = Sx(1)
vx1 = dSx(1)
y1 = Sy(1)
vy1 = dSy(1)
```

```
x4 = Sx(4)
vx4 = dSx(4)
y4 = Sy(4)
vy4 = dSy(4)
```

```
x6 = Sx(6)
vx6 = dSx(6)
y6 = Sy(6)
vy6 = dSy(6)
```

```
#-----
#Longitud de arco
L=@(t) sqrt(dSx(t).^2 + dSy(t).^2)
Q = intNCcompuesta(L, 0, 6, 12, 5)
```