

# Análisis de performance de métodos de interpolación y aplicación en problemas de trazado de trayectorias

Maximiliano Lombardia

Universidad de San Andrés, Buenos Aires, AR

2do Semestre 2024

## Resumen

Se sabe que la interpolación, como técnica, consta de una utilidad superlativa a la hora de obtener funciones que permitan realizar una aproximación en base a información disponible[1]. En particular, es deseable entender qué metodos tienen mejor desempeño para permitir una mejor aproximación, y utilizarlo en problemas que requieran de una aplicación práctica. A continuación, se mostrará un análisis de varios métodos de interpolación, mostrando distintas métricas que permiten determinar cuál es el más performante, para luego proceder a mostrar en un caso práctico su rendimiento.

## 1. Introducción

La interpolación tiene una utilidad esencial a la hora de interpretar datos que no se encuentran nucleados en una función (más bien, que no fueron medidos en base a una). Mediante su técnica, permite representar una función mediante la cual se puede obtener una tendencia de aquellos intervalos que no están contemplados por mediciones.

Existen múltiples métodos de interpolación, siendo la interpolación Lagrange y la interpolación Spline dos variantes ampliamente conocidas y utilizadas.

En este trabajo se procederá a estudiar, en base a funciones arbitrarias, el rendimiento de éstos métodos, para determinar que el Spline cúbico es el que mejor performa, para luego aplicarlo en un problema donde se debe generar una

trayectoria en base a interpolación de puntos obtenidos mediante otro método (explicado en la sección 2.4)

## 2. Métodos y reglas a aplicar

### 2.1. Interpolación de Lagrange

El método de interpolación de Lagrange es un método que permite interpolar datos mediante polinomios de grado  $n-1$ , siendo  $n$  la cantidad de puntos a utilizar para la interpolación [2]. Para calcular la interpolación, se usa la forma del polinomio interpolador de Lagrange:

$$P(x) = \sum_{k=0}^n f(x_k) L_{n,k}(x) \quad (1)$$

donde

$$L_{n,k} = \prod_{i=0, i \neq k}^n \frac{(x - x_i)}{(x_k - x_i)} \quad (2)$$

## 2.2. Interpolación Spline

El método de interpolación por Splines tiene como premisa interpolar por trozos (*piece-wise interpolation*) para evitar oscilaciones importantes generadas por usar un único polinomio de grado  $n$  [2]. Las dos formas que se usarán en este desarrollo son la lineal, que consiste en definir múltiples funciones lineales en cada sub-intervalo deseado, y la cúbica, donde ya cada sub-intervalo tendrá un polinomio de grado 3 de la forma

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

y debe cumplir que cada  $S_j$  sea un polinomio cúbico en  $[x_j, x_{j+1}] \forall j = 0, \dots, n - 1$ , cada punto evaluado en  $S_j$  debe ser igual a evaluarlo en la  $f$  deseada ( $S_j(x_j) = f(x_j)$ ) y que evaluar un punto que conecta  $S_j$  con  $S_{j+1}$  (ocurriendo igual en la primera y segunda derivada) deben dar igual en ambas.

## 2.3. Nodos/raíces de Chebyshev

Los nodos de primer tipo de Chebyshev (también conocidos como *raíces de Chebyshev*) son conjuntos de números algebraicos acotados entre  $[-1, 1]$  y que cumplen ser raíces de los polinomios de Chebyshev. Estas raíces son de la forma[1]

$$x_k = \cos\left(\frac{2k - 1}{2n}\pi\right), \quad (4)$$

con  $k = 1, 2, \dots, n$ . Es importante destacar que también se puede extender a intervalos distintos a  $[-1, 1]$  con la forma general [3]

$$x_k = \frac{(a + b)}{2} + \frac{(b - a)}{2} \cos\left(\frac{2k + 1}{2n}\pi\right), \quad (5)$$

con  $k = 0, 1, \dots, n - 1$ .

## 2.4. Método de Newton-Raphson

Para el segundo experimento, en el contexto de resolver un sistema de ecuaciones para el método de trilateración de posiciones se debe utilizar el Método de Newton-Raphson pero para múltiples variables [2], cuyo paso iterativo queda de la siguiente forma:

$$\vec{x}_{n+1} = \vec{x}_n - J^{-1}(\vec{x}_n)f(\vec{x}) \quad (6)$$

Este paso iterativo se repite hasta alcanzar una diferencia mínima entre valores (es decir, hasta que  $J^{-1}(\vec{x}_n)f(\vec{x}) < tol$ , siendo  $tol$  la tolerancia definida para la variación entre iteraciones), o hasta iterarse un máximo de iteraciones definido.

## 2.5. Cálculo de errores

A lo largo de este trabajo se utilizarán dos tipos de errores: el error relativo promedio, y el error máximo. El primero consiste en obtener, para cada  $n$ , cada valor interpolado, generar la diferencia en módulo con el valor exacto y generar un promedio de todos los elementos:

$$x_n = mean(|\vec{i}_n - \vec{r}_n|) \quad (7)$$

donde  $mean$  es la función promedio, y  $|\vec{i}_n - \vec{r}_n|$  representa el módulo de la resta miembro a miembro entre los puntos evaluados en la interpolación y la función real para un determinado  $n$ . Por otro lado, el error máximo surge de realizar la misma operación, pero analizando el valor máximo en vez de realizar un promedio de todos:

$$x_n = max(|\vec{i}_n - \vec{r}_n|) \quad (8)$$

donde  $max$  es la función que busca el máximo.

### 3. Experimentos numéricos

El trabajo se divide en dos partes: el análisis de los métodos en funciones específicas, y luego la aplicación de la interpolación en un ambiente práctico.

Respecto a la implementación, se utilizó Python como lenguaje para implementar los experimentos, y se utilizaron tanto librerías propias del lenguaje como librerías de código abierto para utilidades varias.

Del lenguaje, se utilizó la librería *math*, la cual provee acceso a las funciones matemáticas definidas en el estándar C. En este caso en particular, fue de utilidad para calcular funciones periódicas y el uso de constantes irracionales como  $\pi$ .

Respecto a librerías abiertas, se utilizaron *numpy*, *matplotlib*, *pandas*, *sklearn* y *scipy*.

La librería *numpy* aporta una interfaz muy completa para todo lo referido a manejo de estructuras con varias dimensiones, además de para poder utilizar operaciones como cálculo de exponenciales o de promedios de un vector de elementos dados. Esto se complementa con el uso de *scipy*, que permite utilizar implementaciones con buena performance para las interpolaciones estudiadas (Lagrange, splines, y la extensión a 3D).

Finalmente, *matplotlib* ofrece un conjunto de herramientas muy amplio para realizar gráficos en 2 y 3 dimensiones (incluso con varios tipos de escalada, como la semi-logarítmica), con lo cual se utilizó ésta para los gráficos del experimento; *sklearn* fue utilizada brevemente para el error cuadrático medio, y *pandas* para poder leer archivos con información de posición y mediciones para la segunda actividad.

#### 3.1. Análisis de funciones

La primera función a analizar es de la forma

$$f_1(x) = -0.4 \tanh(50x) + 0.6, \quad (9)$$

con  $x \in [-1, 1]$ .

La segunda función a analizar es de la forma

$$\begin{aligned} f_2(\mathbf{x}) = & 0.75 \exp\left(-\frac{(9x_1 - 2)^2}{4} - \frac{(9x_2 - 2)^2}{4}\right) \\ & + 0.75 \exp\left(-\frac{(9x_1 + 1)^2}{49} - \frac{(9x_2 + 1)^2}{10}\right) \\ & + 0.5 \exp\left(-\frac{(9x_1 - 7)^2}{4} - \frac{(9x_2 - 3)^2}{4}\right) \\ & - 0.2 \exp\left(-\frac{(9x_1 - 7)^2}{4} - \frac{(9x_2 - 3)^2}{4}\right) \end{aligned} \quad (10)$$

con  $x_1, x_2 \in [-1, 1]$ .

Para ambas funciones, se han realizado experimentos con varios valores de puntos, tomados tanto de forma equiespaciada como también tomando raíces asociadas a los nodos de Chebyshev.

#### 3.2. Trilateración de posiciones

En esta segunda parte, ya se procede a estudiar un problema práctico con el método que mejor desempeño tenga, para resolver el sistema de ecuaciones que utiliza el método de trilateración de posiciones para recuperar la posición de una partícula:

$$\begin{aligned} \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2} &= d_1, \\ \sqrt{(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2} &= d_2, \\ \sqrt{(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2} &= d_3, \end{aligned} \quad (11)$$

donde  $(x, y, z)$  es la posición de la partícula en el momento dado, y  $(x_i, y_i, z_i)$  (con  $i = 1, 2, 3$ ) son las posiciones de los sensores y  $d_i$  son las mediciones realizadas. Se debe recuperar la trayectoria

de una partícula, obteniendo primero los puntos que recorrió la misma en base a las mediciones tomadas y la posición de los sensores, para luego interpolar éstos y obtener la trayectoria aproximada.

## 4. Resultados y análisis

### 4.1. Análisis de funciones

En el caso de Lagrange, en primer lugar se empleó la técnica de puntos equiespaciados con  $n = 5$  ascendiendo progresivamente para valores más altos. En la figura 1 se puede apreciar la evaluación en los puntos del polinomio generado con 10 puntos ( $n = 10$ ).

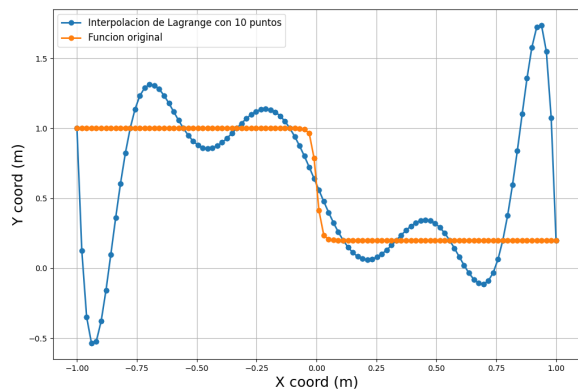


Figura 1: Interpolación de Lagrange tomando 10 puntos equiespaciados

Distinto fue el caso de aplicar las raíces de Chebyshev para conseguir puntos de interpolación: en las figuras 2 y 3 se puede ver, usando  $n = 10$  y  $n = 20$  cómo la trayectoria de la curva se ajusta mejor a la función original a medida que se aumenta  $n$ .

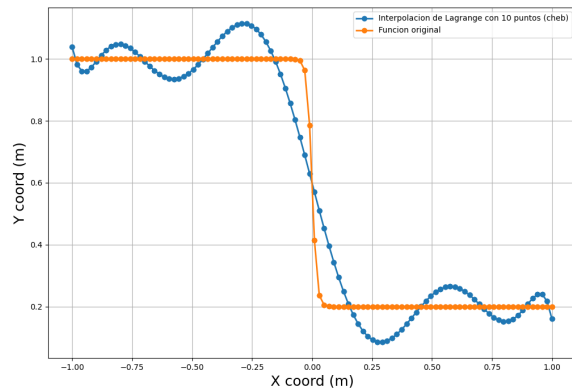


Figura 2: Interpolación de Lagrange tomando 10 puntos con nodos de Chebyshev

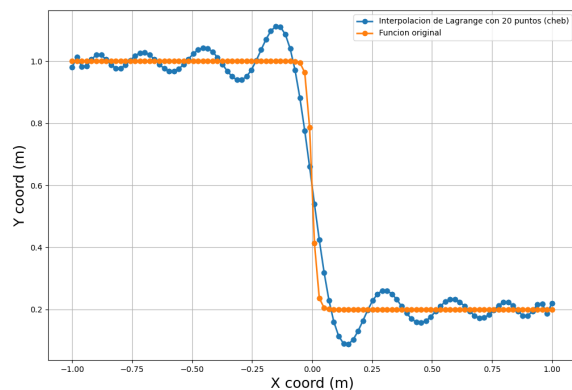


Figura 3: Interpolación de Lagrange tomando 20 puntos con nodos de Chebyshev

En el caso de Spline, se podría decir que ocurre lo contrario a lo visto en Lagrange: si bien la aproximación mejora a medida que se aumenta el  $n$ , en base a las figuras 4, 5, 6 y 7 el método aparenta tener mejor performance eligiendo puntos equiespaciados contra usar las raíces de Chebyshev.

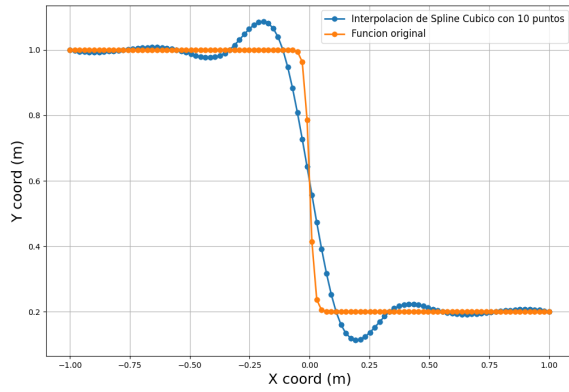


Figura 4: Interpolación Spline tomando 10 puntos equiespaciados

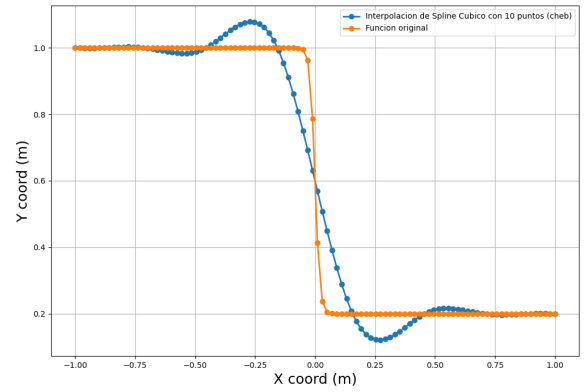


Figura 6: Interpolación Spline tomando 10 puntos con nodos de Chebyshev

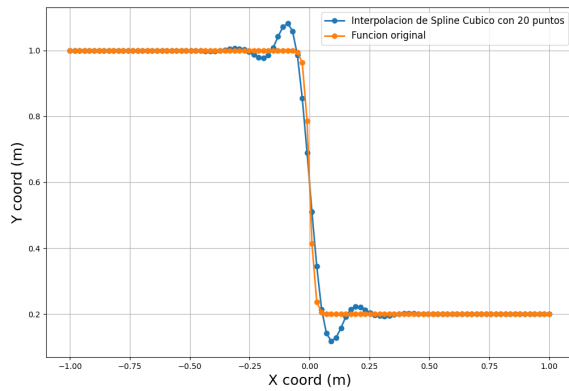


Figura 5: Interpolación Spline tomando 20 puntos equiespaciados

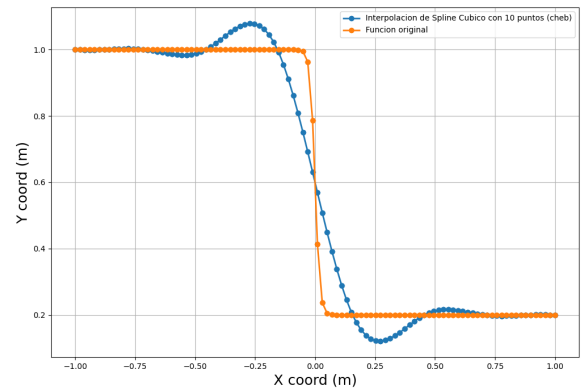


Figura 7: Interpolación Spline tomando 20 puntos con nodos de Chebyshev

Si bien los gráficos anteriores aportan información valiosa para entender el rendimiento de cada método en múltiples situaciones, son útiles para tener un entendimiento inicial de la situación. Obteniendo y graficando los errores se puede ver que se condicen con lo visto previamente. En el caso de la figura 8 se puede ver que el método de Lagrange con puntos equiespaciados tiene un error que tiende a crecer, y luego

de los  $n$  a partir del 10 llega a valores bastante elevados comparado con los otros métodos.

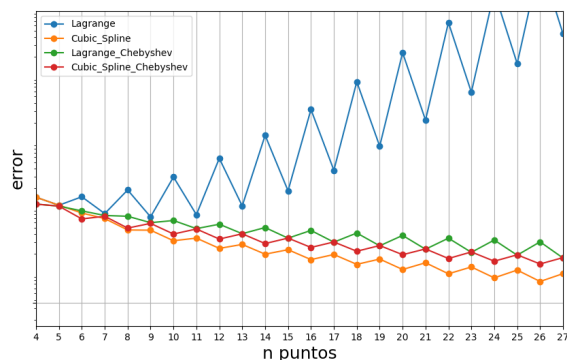


Figura 8: Comparación de promedio de errores relativos de todas las variantes

Comparando las otras tres variantes en función a métodos y reglas para elegir puntos, en la figura 9 se puede ver cómo la combinación que presenta menor error en todos los  $n$  relevantes para el estudio es el Spline con puntos equiespaciados, puesto en comparación con usar raíces de Chebyshev. En el caso de Lagrange, usar raíces de Chebyshev permite poder compararlo en rendimiento con Splines, debido a la mala performance con puntos equidistantes.

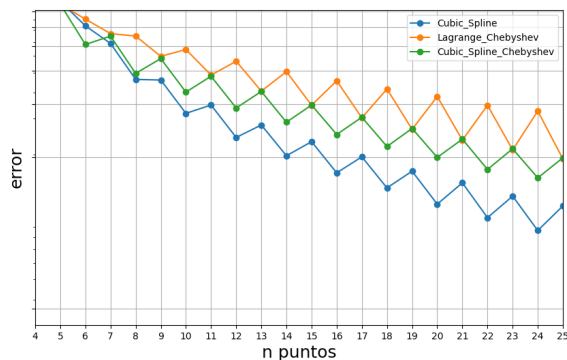


Figura 9: Comparación de promedio de errores relativos de todas las variantes, excepto interpolación de Lagrange con puntos equiespaciados

Para la segunda función, al ser en tres dimensiones, se consideró necesario, en primera instancia, revisar el gráfico de la función por separado y entender qué curvas e inclinación se deberían apreciar.

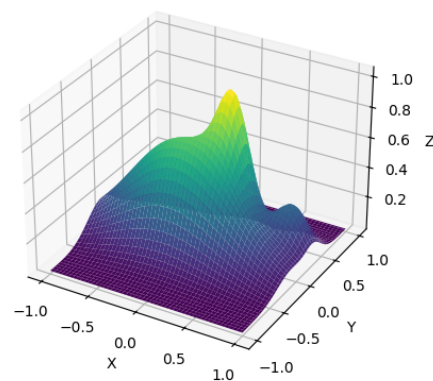


Figura 10: Gráfico de la función  $f_2$

Se estudió luego la evolución de los métodos con el incremento de  $n$ . En este caso, se consideró relevante para explicar la evolución los casos de  $n = 3$  y  $n = 20$  en las 4 variantes posibles. Las

ternas de figuras 11,12,13 y 14,15,16 representan lo que se vió en  $n$  intermedios: a medida que avanza  $n$ , se va construyendo la estructura básica de la curva tridimensional. Lo que se puede apreciar comparando los pares de figuras 11,14, 12,15 y 13,16 es que el método de Spline cúbica aporta mayor precisión en la curvatura de los picos, aportando una aproximación con más similitud con la función original que el método lineal.

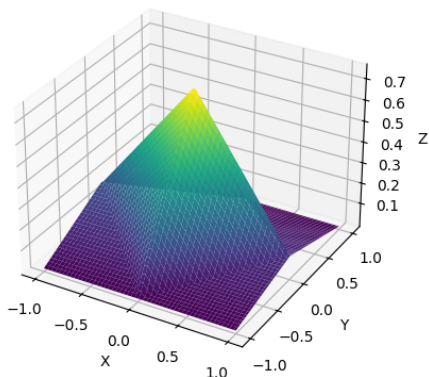


Figura 11: Interpolación lineal tomando 9 puntos equiespaciados ( $n = 3$ )

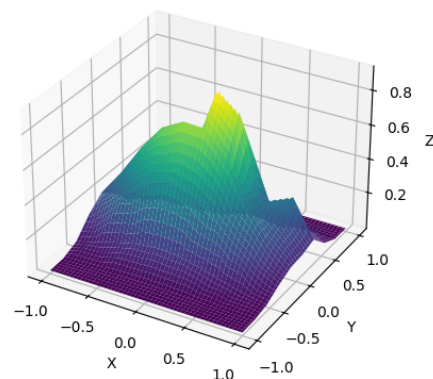


Figura 12: Interpolación lineal tomando 100 puntos equiespaciados ( $n = 10$ )

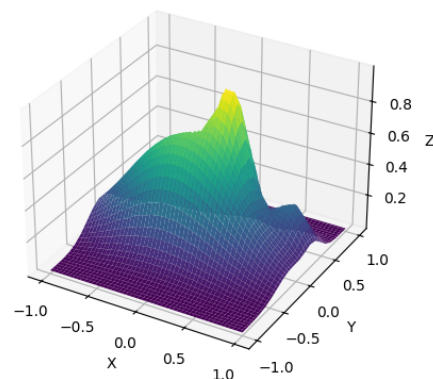


Figura 13: Interpolación lineal tomando 400 puntos equiespaciados ( $n = 20$ )

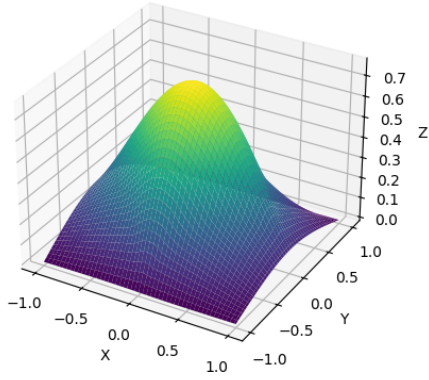


Figura 14: Interpolación Spline cúbica tomando 9 puntos equiespaciados ( $n = 3$ )

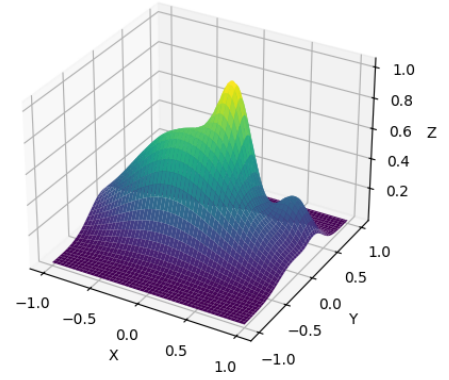


Figura 16: Interpolación Spline cúbica tomando 400 puntos equiespaciados ( $n = 20$ )

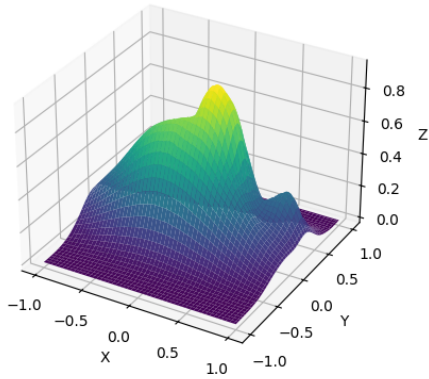


Figura 15: Interpolación Spline cúbica tomando 100 puntos equiespaciados ( $n = 10$ )

Para explicar la desestimación de las raíces de Chebyshev para elegir los puntos se puede recurrir a los gráficos de error relativo y máximo en cada caso. Comparando los pares de figuras 17,18 y 19,20 se puede apreciar que con puntos equidistantes el error de los métodos decrecen monótonicamente a medida que se aumenta el valor de  $n$ , alcanzando cierta estabilidad con el avance de  $n$ , mientras que con las raíces de Chebyshev esto no es así, y esta propiedad no se puede marcar, apreciando errores oscilantes con el aumento de la cantidad de puntos elegidos para interpolar.



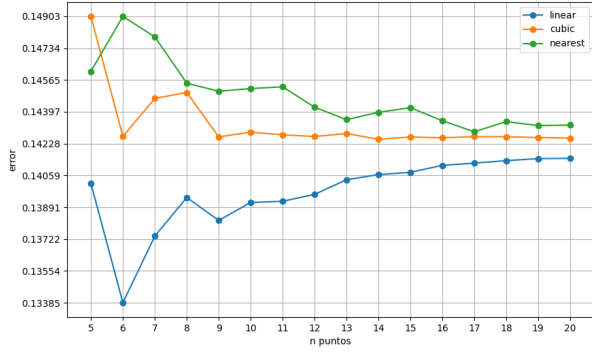


Figura 17: Comparación de promedio de errores relativos de los diferentes métodos con puntos equiespaciados

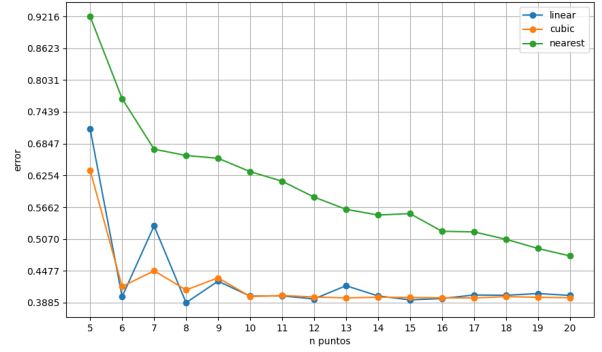


Figura 19: Comparación de errores máximos de los diferentes métodos con puntos equiespaciados

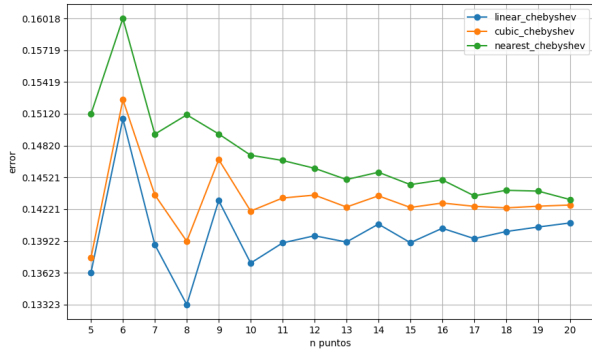


Figura 18: Comparación de promedio de errores relativos de los diferentes métodos con nodos de Chebyshev

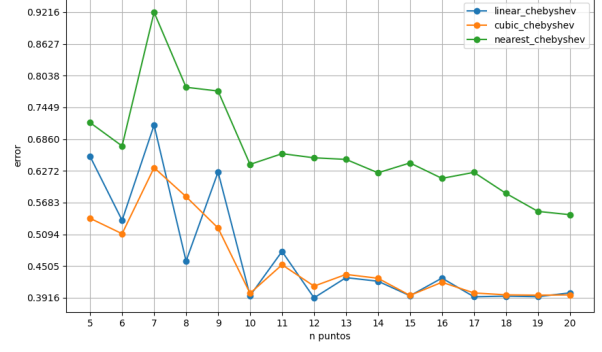


Figura 20: Comparación de errores máximos de los diferentes métodos con nodos de Chebyshev

## 4.2. Trilateración de posiciones

En base a los resultados vistos en el experimento anterior, y debido a las limitaciones presentadas por el método de Lagrange, a la hora de elegir un método de interpolación se escogió por probar con el Spline cúbico, obteniendo como resultado una trayectoria similar a la real, ambas comparadas en la figura 21.

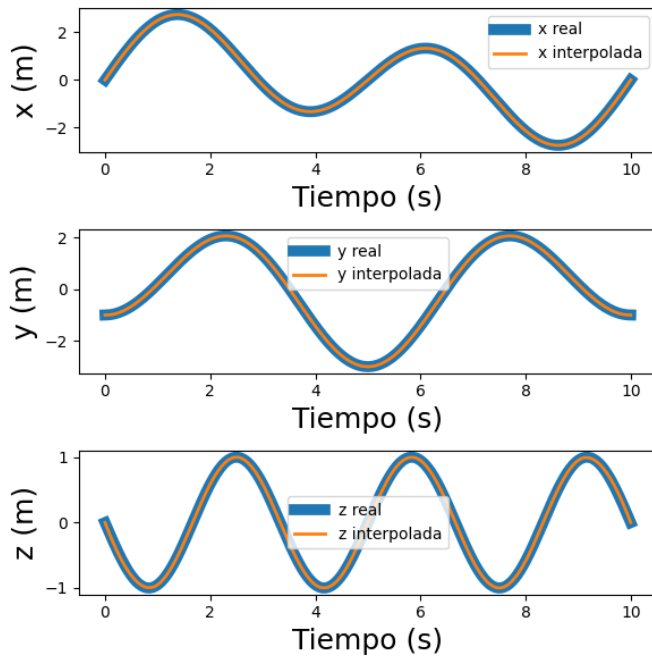


Figura 21: Comparación de trayectoria recuperada con interpolación con la trayectoria real (*ground truth*).

A la hora de analizar la máxima cantidad de iteraciones en base a la cual se estabiliza, se puede ver (analizando los errores cuadráticos medios entre el *ground truth* y el interpolado) que a partir de 4 iteraciones se estabiliza el error:

$n$	$x$	$y$	$z$
3	0.13368e-2	0.16141e-2	0.44788e-2
4	0.13345e-2	0.16114e-2	0.44772e-2
5	0.13345e-2	0.16114e-2	0.44772e-2
10	0.13345e-2	0.16114e-2	0.44772e-2

Tabla 1: Tabla con los errores cuadráticos medios en base a  $n$ .

## 5. Conclusión

Para sintetizar lo realizado en este experimento, se ha estudiado en funciones de múltiples

dimensiones el rendimiento de distintos métodos de interpolación, analizando cuáles aportaban mejores resultados. Una vez hecho ese análisis, se procedió a resolver el problema que planteaba mediante el sistema de ecuaciones que conlleva resolver la trilateración de posiciones.

Entre los descubrimientos realizados en el ejercicio de estos experimentos, se pueden destacar tres de mayor importancia.

Por un lado, es importante evaluar distintas opciones para conseguir puntos de interpolación porque en el caso del polinomio de Lagrange, si se utilizaban puntos equidistantes no era posible realizar el experimento correctamente, mientras que si se usaban las raíces de Chebyshev era posible comparar su performance con otros métodos.

Además, lo que se puede confirmar es que la interpolación como técnica, junto con otros métodos numéricos, puede ser de gran ayuda para problemas donde se cuenta con información sin una tendencia clara, y se necesita generar una función para aproximar valores intermedios. Entre los métodos estudiados, se pudo concluir que el spline cúbico es el que mejor se desempeña.

Finalmente, cabe destacar que el tipo de cálculo de error elegido para analizar la exactitud de un método es un punto clave para decir que método es mejor para cada situación.

Como futuros pasos, se puede contemplar un estudio de más métodos de interpolación, y compararlos partiendo de los resultados ya obtenidos en este informe. Además, se alienta a explorar la utilidad de la interpolación en otros ámbitos donde no parezca que se pueda utilizar métodos numéricos para llegar a una solución, o que cuenten con datos de la misma naturaleza que los expuestos aquí (es decir, sin ninguna función evidente, como análisis de mercado).

## Referencias

- [1] Annette Burden Richard Burden Douglas Faires. *Numerical Analysis*. Cengage, 2014.
- [2] Martin Palazzo Joaquin Cullen. *Apuntes de cátedra*. Universidad de San Andrés, 2024.
- [3] Wikipedia. *Chebyshev nodes*. URL: [https://en.wikipedia.org/wiki/Chebyshev\\_nodes](https://en.wikipedia.org/wiki/Chebyshev_nodes).