

# Métodos Numéricos y Optimización

## Trabajo Práctico N°1

### Interpolación y Búsqueda de Raíces

Manuela Gómez Pazos y Francesca Ragonesi

Septiembre 2023

#### Abstract

En el siguiente trabajo se analizaron métodos de interpolación y de búsqueda de raíces en múltiples dimensiones. Dividimos el mismo en dos secciones. En el primer problema se estudiaron dos funciones previamente provistas, mientras que en el segundo se estudió la trayectoria de un vehículo que atraviesa un campo.

## 1 Introducción

Uno de los problemas más comunes en el espacio de la aproximación numérica es el de encontrar raíces. En el primer problema se estudió el desempeño de dos funciones en distintos intervalos. Estas predicciones pueden ser obtenidas mediante funciones que se ajusten a dichos datos, este proceso es conocido como interpolación. Dentro de los distintos métodos de interpolación se utilizaron la interpolación lineal, la de Lagrange y la interpolación mediante Splines cúbicos y quinticos.

Por otro lado, en el caso dos, se describe la trayectoria de un tractor en un campo, dentro de ciertos límites. Uno de los puntos analizados fue el cálculo de la intersección entre dicha trayectoria y los límites. Para resolverlo se contó con el método de Newton-Raphson.

A su vez, al ser proporcionados un serie de datos, una pregunta común es si existe la posibilidad de realizar distintas estimaciones acerca del comportamiento de los mismos. Para el problema mencionado previamente, se emplearon dos tablas de datos, una con mediciones y otra con la información sobre el "ground truth" de las funciones no conocidas.

En este trabajo, se utilizarán distintos métodos tanto de búsqueda de raíces como de interpolación para resolver diversos enunciados provistos por la cátedra.

## 2 Métodos

Los siguientes métodos fueron utilizados a lo largo del trabajo.

### 2.1 Interpolación

En una de las secciones de la investigación se analizaron y estudiaron diversos procedimientos de interpolación aplicados a dos funciones distintas.

#### 2.1.1 Interpolación lineal

Este método permite ajustar un polinomio de grado uno a una serie de datos tabulados. Se utiliza la forma de una ecuación lineal

$$f(x) = mx + b$$

Para estimar un valor debemos emplear la unión de rectas y dos puntos conocidos. Si se tienen varios puntos, los mismos se proceden a unir de a pares con rectas.

### 2.1.2 Interpolación de Lagrange

Ajusta un único polinomio a los datos, de forma tal que este pase por cada uno de ellos. Dicho polinomio debe tener el mínimo grado posible y es obtenido mediante los polinomios interpoladores de Lagrange y la función original, evaluándola en los puntos brindados como datos.

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

Con  $L_n$  el polinomio de Lagrange asociado a cada punto de datos.

### 2.1.3 Interpolación con splines

Los métodos enunciados hasta el momento buscan ajustar un polinomio de menor grado que pasa por todos los puntos. Ahora bien, si los  $n$  puntos presentan una variación tal que puede generar un polinomio de grado  $n - 1$ , creando oscilaciones entre los puntos. Entonces los puntos que se interpolan muestran un comportamiento distinto al resto. Los splines generan un polinomio de un grado menor entre los puntos, buscando así evitar las oscilaciones y poder aproximar mejor la función a trozos. Dentro de los splines hay varias categorías, en este trabajo se utilizaron las siguientes.

#### Spline Cúbico:

Ajusta un polinomio de tercer grado entre cada par de puntos. El objetivo es obtener una curva suave y continua, por lo que se realiza la interpolación con la primera y segunda derivada del polinomio.

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$$

Con  $a_i, b_i, c_i, d_i$  constantes y  $x \in [x_0, x_n]$

#### Spline Quintico

Une cada pareja de puntos consecutivos con un polinomio de grado 5, asegurando que la trayectoria que une los puntos sea continua y suave.

$$S_i(x) = a_i(x - x_i)^5 + b_i(x - x_i)^4 + c_i(x - x_i)^3 + d_i(x - x_i)^2 + e_i(x - x_i) + f_i$$

Con  $a_i, b_i, \dots, f_i$  constantes y  $x \in [x_0, x_n]$

## 2.2 Búsqueda de raíces

En el inciso uno se determinó en qué coordenadas el tractor atravesó cada uno de los límites definidos. Para la determinación de dichas coordenadas, se empleó el método de Newton Raphson.

### 2.2.1 Newton Raphson

El método de Newton-Raphson se basa en la serie de Taylor, que aproxima  $f(x)$ .

$$p_n(x) = \sum_{k=0}^n \frac{f^k(x_0)}{k!} (x - x_0)^k$$

Para encontrar el valor de  $x$  donde  $f(x) = 0$ , se trunca la serie de Taylor en el segundo término y se iguala a 0. Esto nos da una aproximación de la raíz de  $f(x)$  y para conocer el valor de  $x$ , simplemente hay que despejarlo de la ecuación.

En el método iterativo  $x_{i+1}$  se utiliza para aproximar la raíz. Es importante comprobar si se cumple el criterio de convergencia. Si no se cumple, se debe iterar con un nuevo valor de  $x$ . Es necesario proporcionar un valor inicial  $x_0$ .

Aunque el método de Newton-Raphson puede ser un ciclo infinito en algunos casos, se puede incluir otra condición de parada que limite el número máximo de iteraciones.

$$X_{i+1} = X_i - \frac{f(X_i)}{f'(X_i)}$$

Donde  $x_{i+1}$  es una aproximación a la raíz.

El método de Newton-Raphson es conocido por converger rápidamente gracias a que utiliza la derivada de la función, lo cual permite que  $x$  se aproxime a la raíz de  $f(x)$ . No obstante, en algunos casos, la derivada puede alejar el valor de  $x$  de la raíz, llegando incluso a divergir. Es por ello que el algoritmo establece un límite máximo de iteraciones.

## 2.3 Puntos Chebyshev

A pesar de que los puntos Chebyshev no son un método de interpolación, ni de búsqueda de raíces, es necesario mencionarlos dentro del marco teórico debido a su aplicación durante el proceso experimental. Los puntos de Chebyshev, son un conjunto de puntos específicos utilizados en matemáticas y métodos numéricos, especialmente en la interpolación y aproximación de funciones. Estos puntos se obtienen aplicando la función de Chebyshev, la cual está definida matemáticamente como el conjunto de soluciones a la siguiente ecuación:

$$\text{Puntos Chebyshev} = \cos\left(\pi \cdot \frac{2 \cdot \text{arreglo}(n) + 1}{2 \cdot n}\right)$$

Estos puntos son útiles en una variedad de aplicaciones numéricas, ya que pueden mejorar significativamente la precisión y la estabilidad de los cálculos en comparación con puntos equiespaciados cuando se trabaja con polinomios de interpolación y aproximación de funciones.

## 3 Aplicación de los métodos

Los métodos desarrollados previamente fueron utilizados a lo largo de todo el proyecto. En el estudio de interpolación se trabajó con dos funciones dadas, las cuales fueron analizadas por separado.

En primera instancia, se estudió la siguiente función, la cual fue definida utilizando las librerías `math` y `np` de `numpy` en Python:

$$f_a(x) = 0.05^{|x|} \cdot \sin(5x) + \tanh(2x) + 2$$

Con  $x \in [-3, 3]$

Se definió un arreglo aplicando la herramienta de `linspace` en el intervalo definido previamente, para crear diez nodos equiespaciados.

Seguidamente, se interpoló aplicando los métodos de splines cúbicos, spline quínticos y polinomio de Lagrange, utilizando la librería `scipy.interpolate`.

Se generó un cálculo del error absoluto en relación a la función original para cada método empleado.

Luego, se analizó el cambio del error absoluto con respecto a la cantidad de nodos utilizados. Esta observación se hizo para cada uno de los métodos de interpolación empleados.

Mediante la fórmula de Chebyshev, implementada a partir de la librería `numpy.polynomial.chebyshev`, importando `chebpts2` se logró generar diez nodos no equiespaciados. Nuevamente se emplearon los mismos métodos de interpolación usados previamente en el caso de nodos equiespaciados y se calcularon los respectivos errores absolutos y la variación del error con respecto a la cantidad de nodos.

Para la segunda función proporcionada:

$$f_b(x) = 0.7 \exp \left( -\frac{(9x_1 - 2)^2}{4} - \frac{(9x_2 - 2)^2}{4} \right) + 0.45 \exp \left( -\frac{(9x_1 + 1)^2}{9} - \frac{(9x_2 + 1)^2}{5} \right) \\ + 0.55 \exp \left( -\frac{(9x_1 - 6)^2}{4} - \frac{(9x_2 - 3)^2}{4} \right) - 0.01 \exp \left( -\frac{(9x_1 - 7)^2}{4} - \frac{(9x_2 - 3)^2}{4} \right) \quad (1)$$

Con  $x \in [-1, 1]$

Se implementó el uso de las librerías numpy, math, scipy.interpolate, numpy.polynomial.chebyshev y mpl toolkit.mplot3d.

Nuevamente, se generó un arreglo con linspace en el intervalo  $[-1, 1]$ , produciendo diez nodos equiespaciados. A la hora de interpolar la función proporcionada, se recurrió al método de splines cúbicos. Debido a que la función está definida en dos variables, se tuvo que utilizar la función griddata, de la librería scipy.

Para la sección de la investigación que cubre el análisis de búsqueda de raíces, se utilizaron los datos de "ground truth" y mediciones provistos por la cátedra. Se generaron diez valores de tiempo que fueron interpolados, junto con las mediciones mencionadas, con el método de Lagrange.

Se interpoló en base a los datos aportados en el archivo de mediciones utilizando splines cúbicos.

Para realizar la búsqueda de las raíces, se definieron las funciones que representan las ecuaciones de las rectas con respecto a los límites. Posteriormente, se emplearon las funciones de interpolación con los límites de cada función restados. Teniendo en cuenta dichos límites, se llevó a cabo el método de Newton-Raphson con el objetivo de encontrar los puntos de intersección entre el tractor y el campo que lo delimita. Los puntos de intersección se identificaron mediante una estimación visual, a través del gráfico, eligiendo puntos cercanos al lugar de intersección.

## 4 Análisis de los resultados

### 4.1 Funciones interpoladas

En primer lugar, se analizó la primer función mencionada anteriormente. Se estudió en diez puntos equiespaciados distintos, generando trescientos puntos donde se evaluaron los resultados para obtener una representación de las funciones con mayor suavidad. Se estudiaron los tres métodos de interpolación diferentes mencionados previamente. Se graficaron las interpolaciones y la función original en un mismo gráfico para poder percibir claramente las diferencias entre cada método. Además se calcularon los errores absolutos relacionados a cada método de interpolación, lo cual permite una mejor visualización de los resultados. Luego, se realizó el mismo análisis pero esta vez observando diez nodos no equiespaciados. Esta interpolación fue generada con la función de Chebyshev. Los siguientes fueron los productos obtenidos al realizar el plot de las interpolaciones y los errores absolutos, generados mediante la librería matplotlib.pyplot:

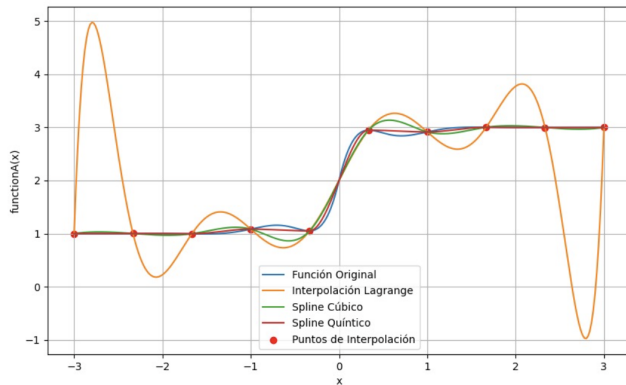


Figure 1: Interpolación con puntos equiespaciados

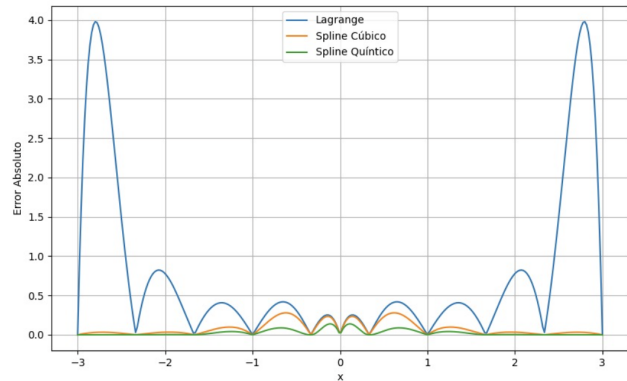


Figure 2: Error absoluto con puntos equiespaciados

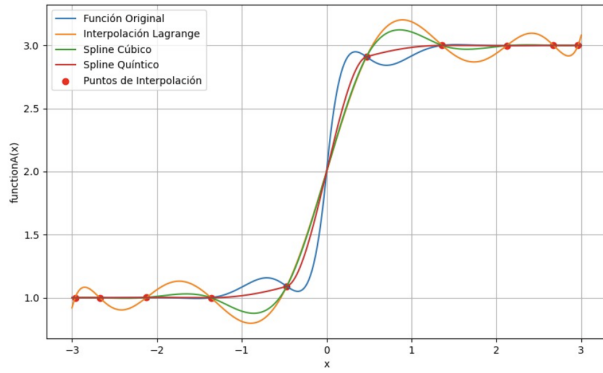


Figure 3: Interpolación con puntos no equiespaciados

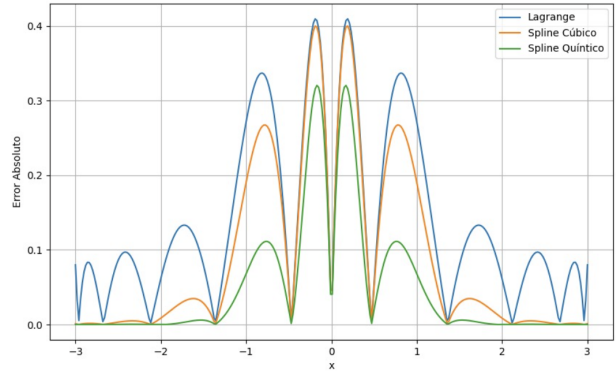


Figure 4: Error absoluto con puntos no equiespaciados

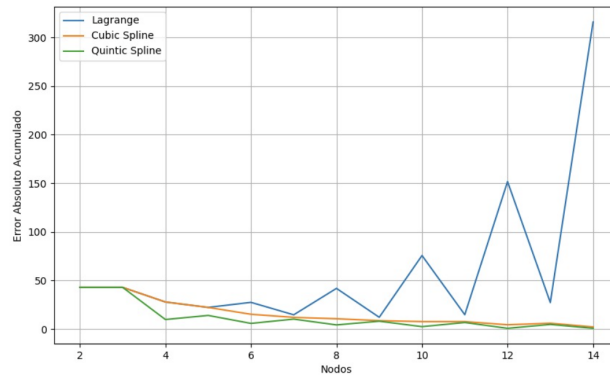


Figure 5: Error absoluto acumulado vs. cantidad de nodos

En las cuatro figuras se puede notar a simple vista que el método de interpolación de Lagrange es el que mayor error poseé, empeorando en los extremos del intervalo en el que se trabajó. En contraposición, los splines cúbicos y quinticos son los que mejor aproximan a la función original y lo hacen de manera similar. Esto sucede porque a diferencia de Lagrange, el método de interpolación con splines aproxima la función de a trozos, lo que hace que sean interpolaciones más locales, capaces de capturar mejor el comportamiento de los datos. A su vez, como se puede apreciar en la figura 5, a mayor cantidad de nodos, mayor error acumulado en el caso de la interpolación de Lagrange. Mientras tanto, en el caso de la interpolación con splines, se observa como a mayor cantidad de nodos, menor error absoluto acumulado, demostrando la eficacia de los splines sobre Lagrange para aproximar la función.

En segundo lugar, se analizó la función en  $\mathbb{R}^2$ . Para esta sección se empleó el uso de Axes3D, lo cual permitió plotear los gráficos necesarios en 3D. Se generaron dos arreglos de cien puntos equiespaciados utilizando linspace para poder pasárselos como parámetros a la función, calcular el "ground truth" y que se vea más suave. Seguidamente, se produjeron cien puntos equiespaciados para interpolar con mayor suavidad y se implementó el método de splines cúbicos mediante el uso de la función griddata, como puede verse en la figura 6.

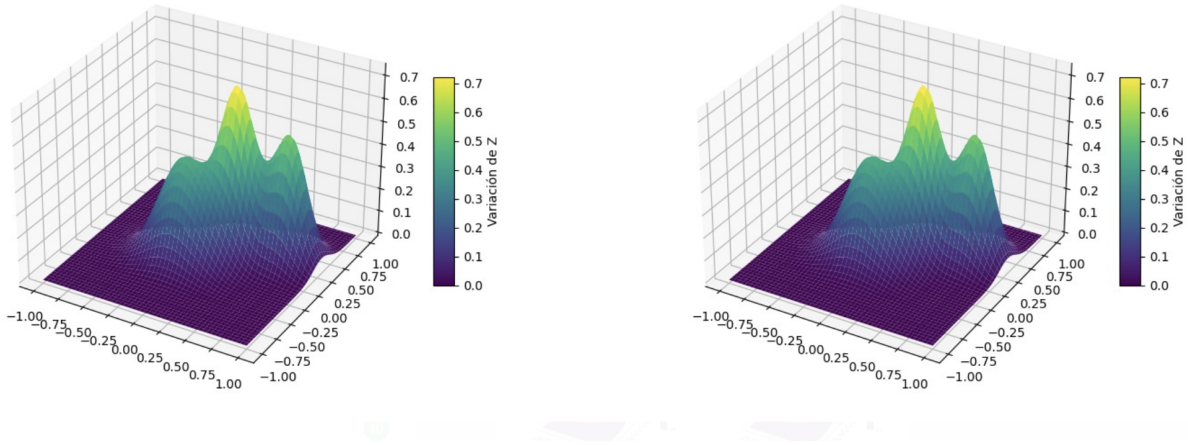


Figure 6: Función original e interpolación con puntos equiespaciados

Posteriormente, se utilizó Chebyshev nuevamente para generar diez nodos no equiespaciados en la función original y cien nodos para la interpolación con splines cúbicos. Ésta última fue implementada con griddata una vez más.

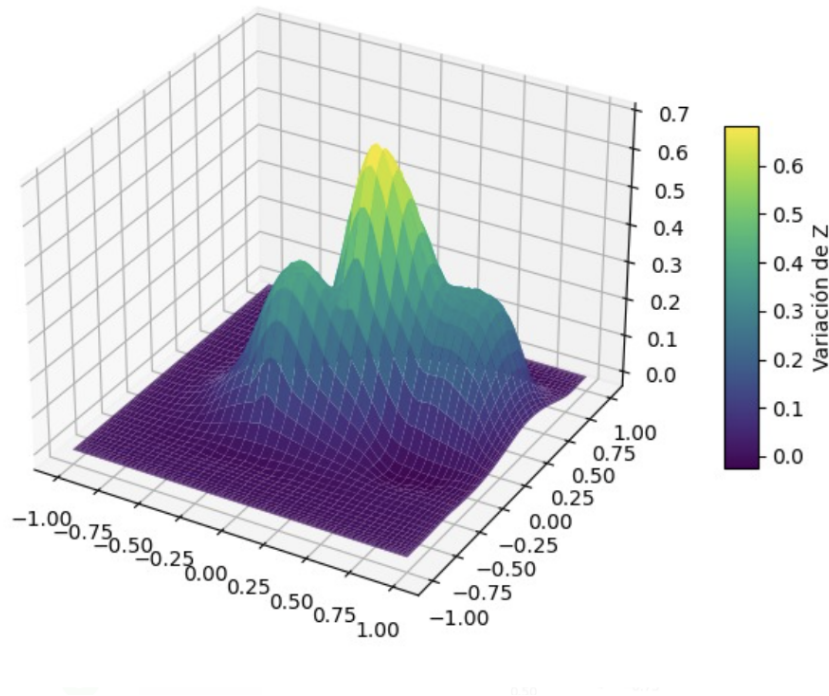


Figure 7: Interpolación con puntos Chebyshev

Luego de calcular las interpolaciones, se pasó a calcular el error absoluto en ambos casos para luego poder realizar en el análisis que de la figura 8 donde se representó el error absoluto acumulado con respecto a la cantidad de puntos de interpolación. Se pudo observar como, si se compara la interpolación hecha con puntos equiespaciados y puntos Chebyshev, la interpolación con puntos equiespaciados cuenta con un menor error absoluto frente a la interpolación con Chebyshev. Esto denota como el spline cúbico puede acoplarse mejor a la función original si interpola puntos equidistantes en 2D a trozos.

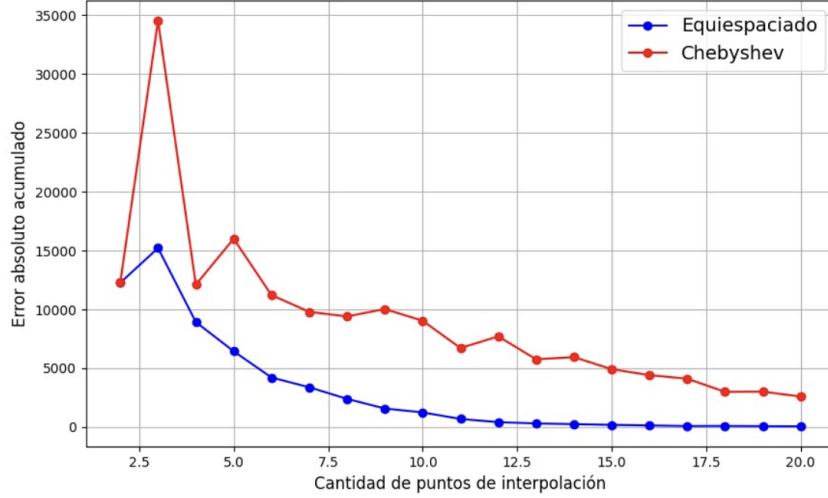


Figure 8: Cambio del error absoluto con respecto a la cantidad de puntos

## 4.2 Búsqueda de raíces

Por su parte, en el punto uno, se buscó interpolar la trayectoria de un tractor de riego que circulaba en un campo. Se proporcionaron dos archivos .csv, uno de ellos estaba compuesto por diez mediciones que representaban los puntos del terreno por los que pasó el tractor. El otro contaba con cien datos del "ground truth" del recorrido del mismo. Con las mediciones aportadas se interpoló mediante el método de splines cúbicos, el cuál vimos en el punto dos que se acopla de manera mas precisa que la interpolación lineal al "ground truth". A su vez, se informó que el campo posee distintas zonas delimitadas para separar cultivos, cuyos límites son los siguientes:

$$x1 = 10$$

$$0.35x1 + x2 = 3.6$$

La interpolación con splines y las rectas pueden verse en la figura 9, donde se aprecia como con splines cúbicos se puede obtener una interpolación que se asemeja bastante a la función ground truth. Esto se debe, como se menciono previamente durante el análisis del punto 2, a la aproximación a trozos que se lleva a cabo en este método de interpolación.

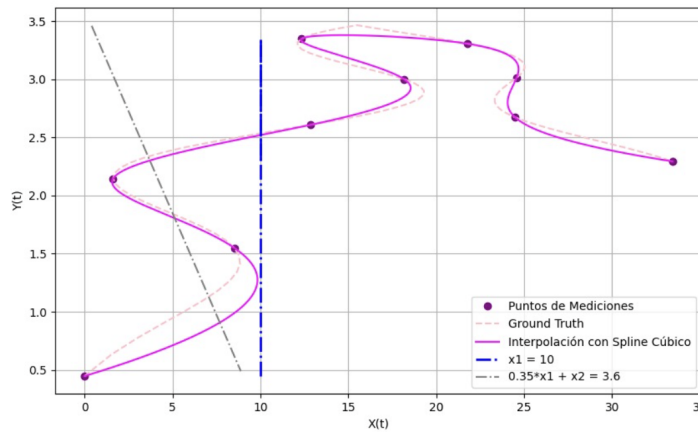


Figure 9: Comparación entre "ground truth" e interpolaciones

Luego de realizar la interpolación de las mediciones, se pasó a graficar la intersección entre la trayectoria interpolada y las rectas que delimitaban los cultivos. Estas intersecciones fueron obtenidas mediante el método de Newton-Raphson, el cual fue utilizado debido a su rápida convergencia. Al haber elegido el  $X_0$  inicial mediante una estimación visual, con pocas iteraciones y por ende en poco tiempo el método logró converger y encontrar las cuatro intersecciones entre la trayectoria del tractor y las dos rectas.

Dichas intersecciones fueron marcadas en la figura 10:

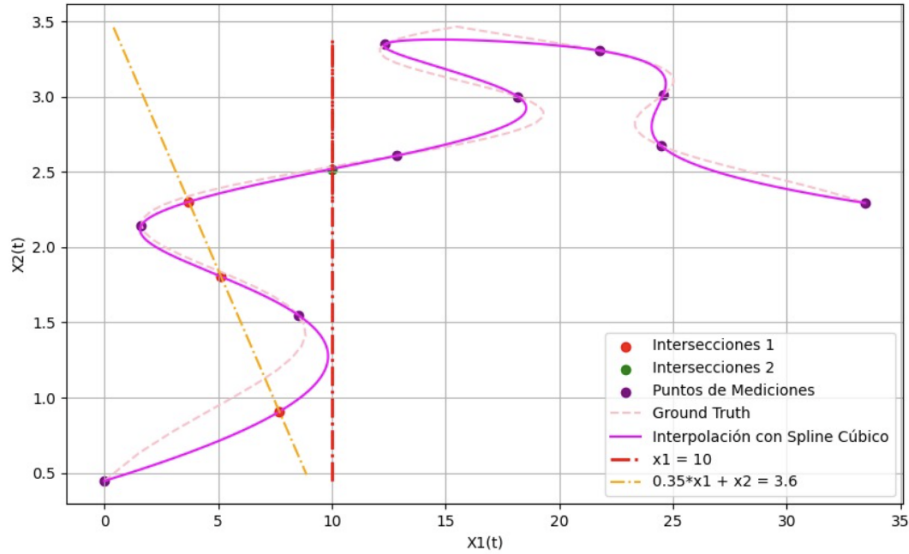


Figure 10: Intersecciones entre la función interpolada y los límites del terreno

## 5 Conclusiones

Los resultados de esta experimentación permitieron concluir en el hecho de que cada método de interpolación cuenta con un comportamiento diferente influenciado por factores como el tipo de función que se interpola, la cantidad de nodos y la forma de espaciar estos últimos. A través del punto uno y dos se pudo ver como las interpolaciones a trozos mediante el método de splines dan como resultado una aproximación mucho mas cercana al “ground truth” que el método de Lagrange, ya que pueden representar mejor el comportamiento de los nodos. Por lo que, se puede obtener una interpolación muy precisa, incluso con un spline cúbico y una sumamente similar a la función original con splines quinticos.

A su vez, la forma de distribución de los nodos, es otro factor que puede afectar la precisión de la interpolación. Como se pudo observar en los gráficos de error absoluto, los nodos no equiespaciados terminan dando aproximaciones menos certeras que lo equiespaciados, especialmente en el centro de las funciones. Por lo tanto, es preferible emplear nodos equiespaciados a la hora de interpolar.

Con respecto al ejercicio uno, específicamente la secciones de búsqueda de intersecciones, luego de emplear el método de Newton-Raphson fue posible llegar a una serie de conclusiones. En primer lugar, se debe tomar en cuenta que los puntos de intersección fueron encontrados de manera rápida y eficiente debido a que se determino el  $X$  inicial para las iteraciones mediante estimación visual del gráfico que representaba la interpolación con splines cúbicos y las dos rectas que delimitaban los cultivos. Facilitando de esta manera la convergencia y éxito del método a la hora de encontrar las intersecciones. Se reconoce que sin la ayuda de un gráfico o dato sobre puntos cercanos a la interseccion buscada puede resultar mucho más problemático dar con la raíz buscada. A pesar de esto, tomando el cuenta el numero de iteraciones y la baja complejidad para poder implementar dicho método, sigue siendo una opción sumamente viable a la hora de querer encontrar raíces de forma iterativa.



## Bibliografía

- Burden, Richard L., J. Douglas Faires, and Annette M. Burden. Numerical analysis. Cengage learning, 2015. (BF)
- Jiménez Bedolla, Juan Carlos. Métodos numéricos usando Python con aplicaciones a la Ingeniería Química. Universidad Nacional Autónoma de México, (2022).