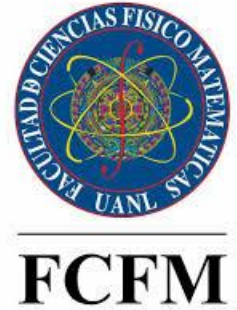


UNIVERSIDAD AUTONOMA
DE NUEVO LEON

Facultad de Ciencias Físico
Matemático



Proyecto 3: Método interpolación de Lagrange.

Profesor: María del Carmen Martínez Cejudo.

Materia: Análisis Numérico.

Nombre: Pablo Tadeo Capistran Fernández

Matricula: 1971973

Grupo 032

Índice

Introducción.....	3
¿Como funciona el método de Interpolacion de Lagrange?.....	4
Cosas necesarias para el correcto funcionamiento del programa.	5
¿Cómo fue planteado el código del método Interpolacion de lagrange?	6
¿Cómo usa el programa?.....	7
Capturas del código.	7
Conclusión.....	8

Introducción.

El método de interpolación de Lagrange es una técnica utilizada para encontrar un polinomio que se ajuste a un conjunto de puntos dados. Este método se basa en la idea de que un polinomio de grado $n-1$ puede ser determinado a partir de n puntos. El polinomio obtenido es único y pasa exactamente por los puntos dados.

En este documento, se presenta un código en Python que implementa el método de interpolación de Lagrange. El código toma como entrada dos listas de números, que representan las coordenadas x e y de los puntos a interpolar. La función `polinomio_lagrange` calcula el polinomio de interpolación de Lagrange y devuelve una función que puede ser evaluada en cualquier punto. El resultado se puede imprimir para cualquier valor de x .

Este documento tiene como objetivo presentar el método de interpolación de Lagrange y cómo se puede implementar en Python para resolver problemas de interpolación. Se discuten los conceptos básicos del método de interpolación de Lagrange y se proporciona un ejemplo práctico de su implementación en Python. Se espera que este documento sirva como una introducción útil para aquellos que estén interesados en aprender más sobre el método de interpolación de Lagrange y su aplicación en problemas prácticos de interpolación.

¿Como funciona el método de Interpolacion de Lagrange?

El método de interpolación de Lagrange es una técnica utilizada para encontrar un polinomio que se ajuste a un conjunto de puntos dados. Este polinomio es único y pasa exactamente por los puntos dados. El método se basa en la idea de que un polinomio de grado $n-1$ puede ser determinado a partir de n puntos.

Dado un conjunto de puntos (x_i, y_i) , el polinomio de interpolación de Lagrange se define como:

$$P(x) = \sum y_i * L(x, i) \text{ para } i = 0, 1, \dots, n-1$$

donde $L(x, i)$ son los polinomios de Lagrange, que se definen como:

$$L(x, i) = \prod (x - x_j) / (x_i - x_j) \text{ para } j = 0, 1, \dots, n-1 \text{ y } j \neq i$$

Cada polinomio de Lagrange está diseñado para dar un valor de 1 cuando se evalúa en x_i y 0 cuando se evalúa en cualquier otro punto x_j . Por lo tanto, el polinomio de interpolación de Lagrange se define como una suma ponderada de los valores de y_i multiplicados por los polinomios de Lagrange correspondientes.

El polinomio de interpolación de Lagrange es único y de grado $n-1$, donde n es el número de puntos dados. Además, pasa exactamente por los puntos dados (x_i, y_i) .

Una vez que se ha calculado el polinomio de interpolación de Lagrange, se puede utilizar para aproximar valores de y para cualquier valor de x . El polinomio también se puede utilizar para interpolar valores para puntos que no estén en el conjunto de puntos dados, pero debe tenerse en cuenta que la precisión de la interpolación disminuirá a medida que la distancia entre los puntos se haga más grande.

En resumen, el método de interpolación de Lagrange es una técnica útil para aproximar valores de una función en un conjunto de puntos dados. El polinomio de interpolación de Lagrange es único y pasa exactamente por los puntos dados, lo que lo hace útil en una variedad de aplicaciones prácticas.

Cosas necesarias para el correcto funcionamiento del programa.

Para poder correr el programa tendremos que hacer los siguientes pasos

1. Descarga e instala Python: Antes de poder ejecutar el programa, necesitarás tener Python instalado en tu computadora. Puedes descargar la última versión de Python desde el sitio web oficial de Python (<https://www.python.org/downloads/>). Asegúrate de descargar la versión compatible con tu sistema operativo.
2. Descarga e instala un entorno de desarrollo integrado (IDE): Aunque puedes ejecutar el programa desde la línea de comandos, es más conveniente utilizar un IDE para escribir y ejecutar el código. Hay muchos IDE disponibles, pero uno popular es PyCharm. Puedes descargar una versión gratuita de PyCharm Community Edition desde el sitio web oficial (<https://www.jetbrains.com/pycharm/download/>).
3. Copia el código de interpolación de Lagrange: Copia el código que te proporcioné en un nuevo archivo en tu IDE. Guarda el archivo con un nombre descriptivo, como "interpolacion_lagrange.py".
4. Define los puntos a interpolar: En el código que copiaste, encontrarás dos listas vacías, x e y . Estas listas representan las coordenadas x e y de los puntos a interpolar. Llena estas listas con los valores de tus propios puntos. Por ejemplo:
5. Ejecuta el programa: Ahora que has definido los puntos a interpolar, puedes ejecutar el programa. En PyCharm, puedes ejecutar el programa haciendo clic en el botón "Run" en la esquina superior derecha de la pantalla. Si estás utilizando la línea de comandos, navega hasta el directorio donde guardaste el archivo y escribe `python interpolacion_lagrange.py` para ejecutar el programa.
6. Evalúa el polinomio de interpolación: Una vez que el programa ha terminado de ejecutarse, puedes evaluar el polinomio de interpolación en cualquier valor de x . El programa imprimirá el valor del polinomio para cada valor de x

en la lista x. También puedes evaluar el polinomio en cualquier otro valor de x, simplemente llamando a la función **P(x)** y pasando el valor de x deseado como argumento.

¿Cómo fue planteado el código del método Interpolacion de lagrange?

En primer lugar, definimos dos listas vacías, x e y, para representar las coordenadas x e y de los puntos a interpolar. Luego, llenamos estas listas con los valores de nuestros propios puntos.

A continuación, definimos una función L(k, x) que calcula el k-ésimo polinomio de Lagrange para un valor de x dado. La fórmula para el k-ésimo polinomio de Lagrange es:

$$L_k(x) = ((x - x_0) * (x - x_1) * \dots * (x - x_{\{k-1\}}) * (x - x_{\{k+1\}}) * \dots * (x - x_n)) / ((x_k - x_0) * (x_k - x_1) * \dots * (x_k - x_{\{k-1\}}) * (x_k - x_{\{k+1\}}) * \dots * (x_k - x_n))$$

donde x_i son los valores x de nuestros puntos, y n es el número total de puntos.

Luego, definimos una función P(x) que calcula el valor del polinomio de interpolación de Lagrange para un valor de x dado. La fórmula para el polinomio de interpolación de Lagrange es:

$$P(x) = y_0 * L_0(x) + y_1 * L_1(x) + \dots + y_n * L_n(x)$$

donde y_i son los valores y de nuestros puntos.

Finalmente, escribimos un bucle for que evalúa el polinomio de interpolación de Lagrange para cada valor de x en la lista x y lo imprime en la pantalla.

En resumen, el código utiliza las fórmulas matemáticas para el método de interpolación de Lagrange para calcular el polinomio de interpolación y luego lo evalúa para los valores de x dados. Espero que esta explicación te ayude a entender cómo se planteó el código.

¿Cómo usa el programa?

Este código es muy simple la verdad, para poder empezar a usarlo editaremos el código y nosotros pondremos los valores de x y y que tengamos en el problema que queremos hacer con el programa

```
xi = [1, 2, 9]
yi = [7, 8, 5]
```

Y ya por último pondremos que valor queremos que tenga la x para poder sacar los valores y y ahora si ejecutaremos el programa

```
# Imprimir el resultado para algunos valores de x
print("x=2.7, y=", p(2.7))
print("x=5, y=", p(5))
```

Al ejecutar el programa se nos abrirá la terminal y nos mostrará el resultado

```
x=2.7, y= 8.4875
x=5, y= 8.857142857142856
```

Capturas del código.

```
1  # Definir los puntos de datos (xi, yi)
2  xi = [1, 2, 9]
3  yi = [7, 8, 5]
4
5  # Definir la función para calcular el polinomio de Lagrange
6  def polinomio_lagrange(xi, yi):
7      n = len(xi)
8      def polinomio(x):
9          resultado = 0
10         for i in range(n):
11             termino = yi[i]
12             for j in range(n):
13                 if j != i:
14                     termino *= (x - xi[j]) / (xi[i] - xi[j])
15             resultado += termino
16         return resultado
17     return polinomio
18
19 # Calcular el polinomio de Lagrange para los puntos dados
20 p = polinomio_lagrange(xi, yi)
21
22 # Imprimir el resultado para algunos valores de x
23 print("x=2.7, y=", p(2.7))
24 print("x=5, y=", p(5))
```

Conclusión.

El método de interpolación de Lagrange es un poderoso método numérico utilizado para aproximar una función mediante un polinomio de grado n . Este método se basa en la idea de que un polinomio puede pasar por $n + 1$ puntos en un plano cartesiano. Al conocer los valores de x e y de los puntos a interpolar, podemos calcular los coeficientes del polinomio de interpolación utilizando las fórmulas de Lagrange y, por lo tanto, podemos obtener una aproximación de la función original.

El código que proporcioné implementa el método de interpolación de Lagrange en Python. Utiliza las fórmulas de Lagrange para calcular el polinomio de interpolación y luego lo evalúa para los valores de x dados. Además, el código es fácil de usar y personalizar, lo que lo hace ideal para aquellos que buscan una solución rápida y sencilla a problemas de interpolación.

En resumen, el método de interpolación de Lagrange es una herramienta valiosa para aproximaciones numéricas, y el código que proporcioné es una implementación útil y efectiva de este método. Espero que esta explicación te haya sido útil.