

# **Tecnológico de Monterrey Campus Guadalajara**

## **Métodos numéricos en ingeniería**

### **Segundo Proyecto Parcial**

Enrique Antonio Enciso De León A00367589

Antonio Reyes Espinoza A01224955

#### **Introducción**

En el siguiente documento se especifica el desarrollo de un programa capaz de resolver interpolaciones y extrapolaciones utilizando los métodos numéricos de Newton y Lagrange. El programa simplifica el proceso de tener que iterar manualmente las funciones para poder obtener el resultado requerido.

#### **Descripción**

Los métodos numéricos de interpolación y extrapolación de Newton y Lagrange son métodos muy eficientes y utilizados en varias ramas ingenieriles por su metodología simple.

El proyecto consiste en realizar una implementación de los métodos numéricos antes mencionados para ofrecer una simplicidad mayor en su uso. La programación de dicho proyecto debe incluir una forma de interactividad con el usuario para que pueda especificar el orden y hasta que valor desea realizar la interpolación o extrapolación según sea el caso, las opciones a desplegar incluyen elegir entre obtener como resultado un valor exacto o aproximado dependiendo de lo que solicite el usuario.

#### **Diseño de la solución**

El diseño del programa se basó principalmente en 4 clases, una clase main, la cual corre la interfaz principal con el usuario y llama todos los métodos necesarios según sea el caso del método elegido y sus distintos parámetros de ejecución, en esta clase también se llevan a cabo varias validaciones simples del sistema para reducir la probabilidad de que haya un error en la ejecución del mismo.

La clase newton, realiza todo lo necesario para poder resolver la interpolación o extrapolación por el método de Newton. Toma los valores necesarios y contiene métodos para extrapolar en caso de que los datos que se obtienen por la lectura de archivo que ofrece el usuario no sean suficientes para realizar dicha acción. Cuenta con métodos para extrapolar, calcular el intervalo en x de la función tabular, etc. El método fórmula es en si la resolución final del método de Newton.

La clase Lagrange es una clase desarrollada para poder resolver extrapolación e interpolación de una función tabular debido a que este método es más simple de desarrollar se decidió implementarlo en básicamente una función y no fue necesario hacer como lo es en el caso de newton varias funciones que agregaran al desarrollo otros parámetros como lo es la validación de si la función tabular cuenta con intervalos constantes ya que el método de LaGrange no revisa ese tipo de detalles.

La clase Lector es la encargada de abrir y procesar los datos obtenidos mediante un archivo de texto, estos datos son procesados como float por conveniencia en su manejo que se podría dar el caso de tener una función tabular con valores decimales.

Se seleccionó el lenguaje de programación Python para implementar la estructura de clases anterior debido a su simplicidad y elegancia, además por la pericia de los programadores de salir del desarrollo con lenguajes en los que estaban implementando varios proyectos durante su momento.

### **Documentación del programa Newton**

```
def formula(self, xUsuario, orden):
    k = self.fraccionIntervalo(xUsuario)
    self.extrapolacion(self.xAnterior) #aqui ta extrapolacion
    indice = self.lista[0].index(self.xAnterior)
    res = 1
    y0 = self.lista[1][indice]
    for i in range (self.k,0,-1):
        for j in range(0, i):
            res = res * (k - j)
        res = (res * self.lista[j+2][indice]) / math.factorial(j+1)
    y0 = y0 + res
    res = 1
```

## Sumas Finitas

```
def DiferenciasFinitas(matriz[][]):
    lista = []
    lista[lista.length] = (matriz[0])
    lista[lista.length] = matriz[1]
    igual = self.iguales(y)
    if igual != False:
        return False
    else:
        k = 0
        while igual != True:
            k = k + 1
            i = 0
            j = 1
            resultado = []
            while j < self.matriz[0]:
                resta = float(y[j]) - float(y[i])
                i = i + 1
                j = j + 1
                resultado.append(resta)
            igual = self.iguales(resultado)
            y = resultado
            lista.append(resultado)
        self.k = k
        self.lista = lista
        return True
```

## Lagrange

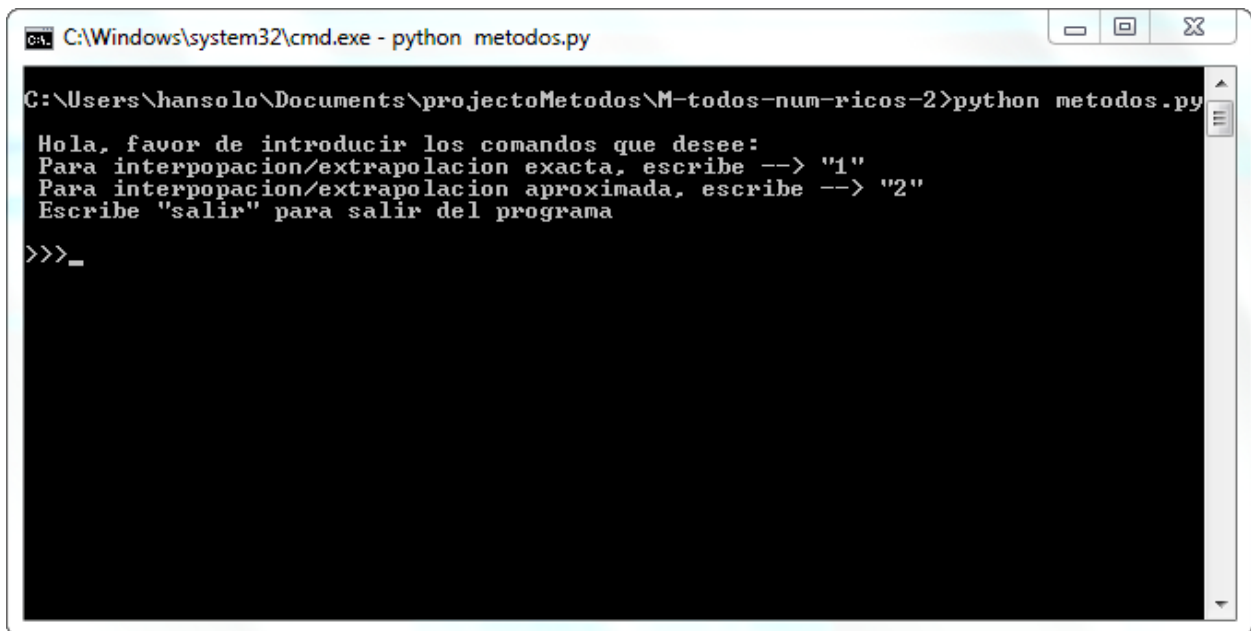
def formula(self, xuser):

```
    result = 0
    for i in range(0, len(self.matriz[0])):
        num = 1
        den = 1
        for j in range(0, len(self.matriz[0])):
            if(j != i):
                num = num * (xuser - self.matriz[0][j])
                den = den * (self.matriz[0][i] - self.matriz[0][j])
        num = num * self.matriz[1][i]
        #print num
        #print den
        print num/den
        result = result + (num/den)
    return result
```

## Casos de uso

### 1. inicio

El usuario inicia el programa y se muestran las siguientes opciones



```
C:\Windows\system32\cmd.exe - python metodos.py

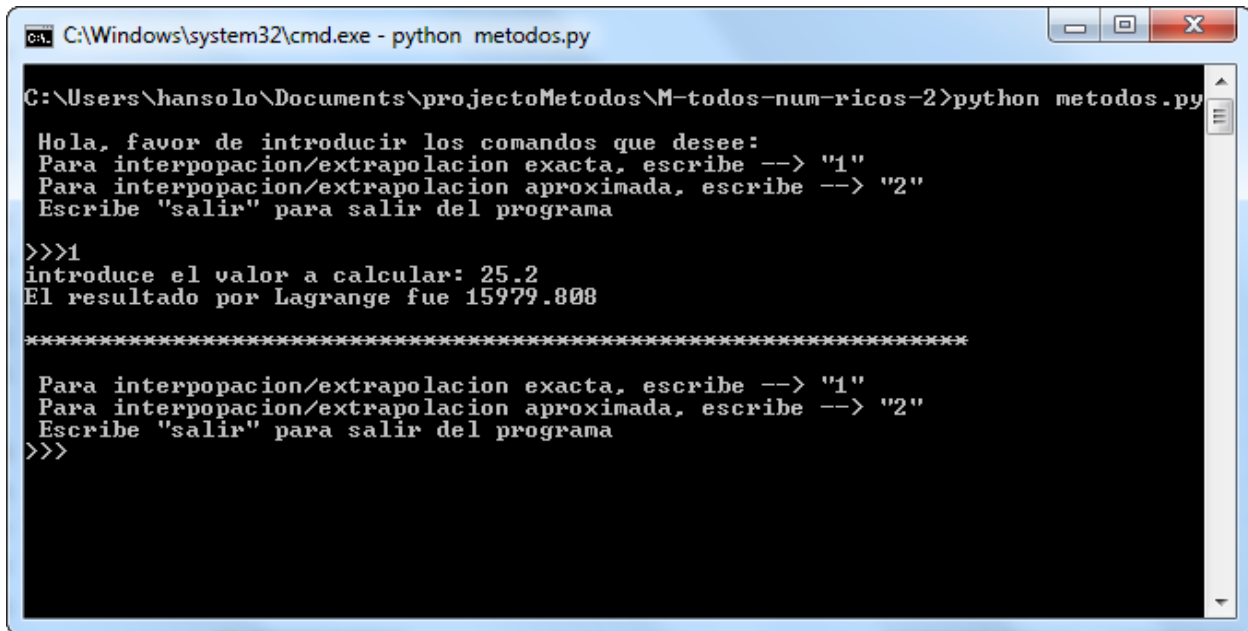
C:\Users\hansolo\Documents\proyectoMetodos\M-todos-num-ricos-2>python metodos.py

Hola, favor de introducir los comandos que desee:
Para interpolacion/extrapolacion exacta, escribe --> "1"
Para interpolacion/extrapolacion aproximada, escribe --> "2"
Escribe "salir" para salir del programa

>>>_
```

## 2. Interpolación/Extrapolación exacta

El usuario introduce el valor "1" y el valor a aproximar



```
C:\Windows\system32\cmd.exe - python metodos.py

C:\Users\hansolo\Documents\proyectoMetodos\M-todos-num-ricos-2>python metodos.py

Hola, favor de introducir los comandos que desee:
Para interpolacion/extrapolacion exacta, escribe --> "1"
Para interpolacion/extrapolacion aproximada, escribe --> "2"
Escribe "salir" para salir del programa

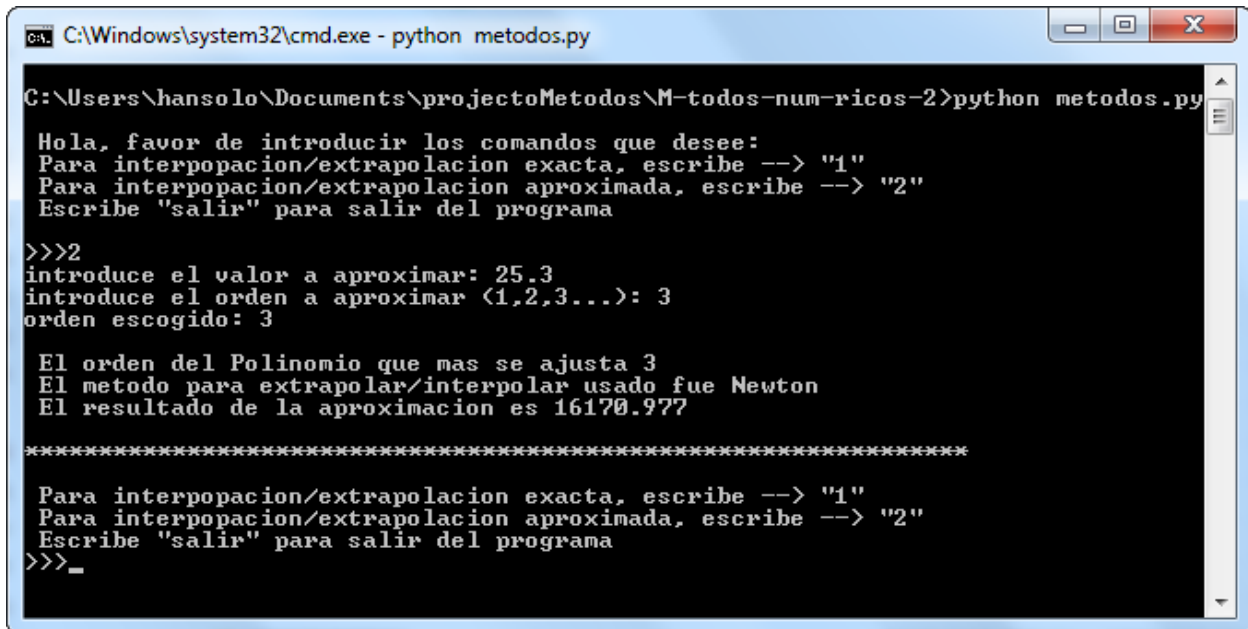
>>>1
introduce el valor a calcular: 25.2
El resultado por Lagrange fue 15979.808

*****

Para interpolacion/extrapolacion exacta, escribe --> "1"
Para interpolacion/extrapolacion aproximada, escribe --> "2"
Escribe "salir" para salir del programa
>>>
```

## 3. Extrapolación/Interpolación aproximada

El usuario introduce "2" en la consola, el valor a aproximar y el orden que quiere calcular, el programa le devuelve los siguiente



```
C:\Windows\system32\cmd.exe - python metodos.py

C:\Users\hansolo\Documents\proyectoMetodos\M-todos-num-ricos-2>python metodos.py

Hola, favor de introducir los comandos que desee:
Para interpolacion/extrapolacion exacta, escribe --> "1"
Para interpolacion/extrapolacion aproximada, escribe --> "2"
Escribe "salir" para salir del programa

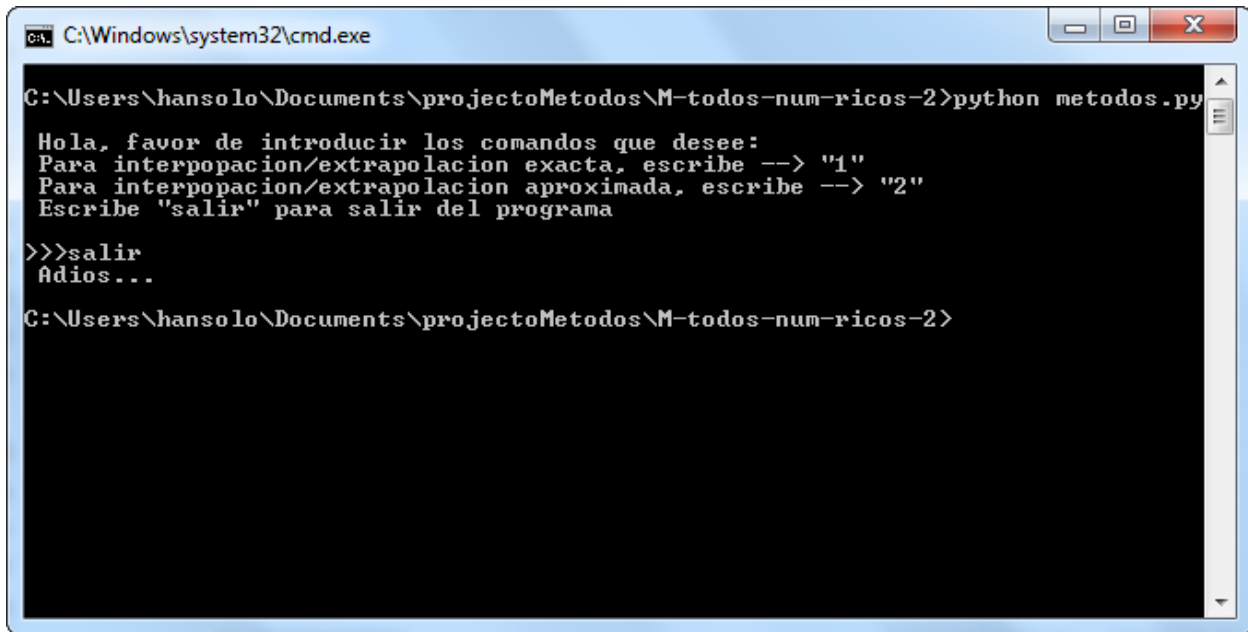
>>>2
introduce el valor a aproximar: 25.3
introduce el orden a aproximar (1,2,3...): 3
orden escogido: 3

El orden del Polinomio que mas se ajusta 3
El metodo para extrapolar/interpoliar usado fue Newton
El resultado de la aproximacion es 16170.977

*****

Para interpolacion/extrapolacion exacta, escribe --> "1"
Para interpolacion/extrapolacion aproximada, escribe --> "2"
Escribe "salir" para salir del programa
>>>_
```

4. El usuario para salir, introduce "salir"



```
C:\Windows\system32\cmd.exe

C:\Users\hansolo\Documents\proyectoMetodos\M-todos-num-ricos-2>python metodos.py

Hola, favor de introducir los comandos que desee:
Para interpolacion/extrapolacion exacta, escribe --> "1"
Para interpolacion/extrapolacion aproximada, escribe --> "2"
Escribe "salir" para salir del programa

>>>salir
Adios...

C:\Users\hansolo\Documents\proyectoMetodos\M-todos-num-ricos-2>
```

### Consideraciones de uso

Las siguientes son las consideraciones necesarias que deben ser aplicadas para que el programa funcione de la manera más óptima.

1. El programa funciona como un archivo ejecutable "exe" en el sistema operativo Windows xp o superior.
2. El archivo de entrada para el programa debe ser un archivo de texto, extensión .txt, el cual no se puede encontrar vacío, con caracteres que no sean números, o con cualquier otra circunstancia fuera de números separados por coma.
  - a. El primer renglón del archivo de texto representa en la función tabular la columna de x y el segundo renglón la columna de y.
3. Aunque algunas validaciones están realizadas para la parte de interacción con el usuario se debe tomar en cuenta solo seleccionar los valores que se le indiquen y hacer las entradas lo más simple posible.
4. Las expresiones mostradas por la consola carecen de acentos y se espera que los valores de ingreso de los usuarios de igual manera no incluyen acentos ni caracteres especiales.

## **Conclusiones**

Una vez terminados el diseño y la implementación del programa para desarrollar los métodos numéricos

Definitivamente, con el poder de la computación podemos utilizar los métodos numéricos para obtener aproximaciones matemáticas de un problema, los métodos numéricos tienen una importancia muy grande y con el desarrollo tecnológico podemos obtener de manera rápida los resultados de esos métodos numéricos.

En este caso, extrapolamos o intrapolamos una función tabular usando dos métodos, Lagrange y Newton. Con el poder computacional de las computadoras actuales, podemos obtener esos resultados de manera rápida solamente haciendo y diseñando el algoritmo e implementando el algoritmo.