

RETO #2

Juan Alejandro Diaz Lote

Jalejandro_diaz@javeriana.edu.co

Cristian Camilo Benítez Peña

Benitez.cc@javeriana.edu.co

Andrés Ricardo Porras Escobar

Andres_porras@javeriana.edu.co

I. INTRODUCCION

Para la realización de este segundo reto, el equipo conformado por Alejandro Diaz, Cristian Benítez y Andrés Porras, decidieron realizar el problema *Interpolación o ajuste de curvas para datos faltantes en una serie de datos climáticos*. Para la solución de dicho problema, debíamos tener en cuenta los diferentes métodos de interpolación (Spline univariante interpolado, Interpolación de Lagrange, Interpolación de diferencias divididas – newton), utilizamos Python para la ejecución del mismo y Excel con fin de representar los datos y convertirlo en un modelo de lectura del lenguaje de programación ya mencionado. Para la solución final, se utilizó modelo del spline univariante interpolado ya que nos proporcionó un gráfico más acertado.

II. SOLUCION.

Para la solución del problema, debíamos tener en cuenta una serie de datos climáticos suministrados en un documento Excel, el cual estaba categorizado en ciudades. Para hacer más amena la lectura de dichos datos, se escogieron un número determinado de datos climáticos (Temperatura interna suministrada) y se ubicaron en un documento en Excel con el fin de evidenciar como fueron determinados los puntos dentro de un plano. Primero hicimos una aproximación métodos vistos en clase como interpolación de Lagrange o Newton, con esto nos dimos cuenta que no era la mejor aproximación, por ende, decidimos emplear el método de spline univariante interpolado ya que el que nos brindó una mejor solución al problema planteado.

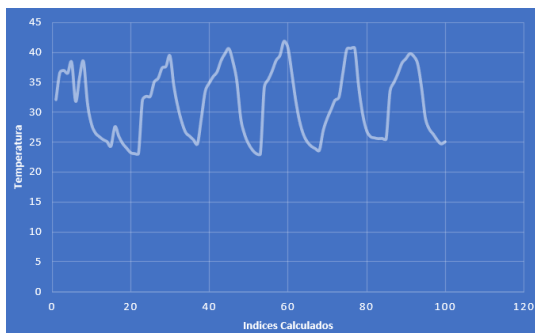


Fig. 1. Interpretación de 100 puntos de temperatura interna en Excel.

Teniendo en cuenta que para realización del problema se utilizó python, hicimos disposición de una serie de librerías como scipy, numpy, matplotlib, pandas. A continuación, una breve descripción de ellas:

- SciPy es una biblioteca libre y de código abierto para Python. Se compone de herramientas y algoritmos matemáticos. SciPy contiene módulos para optimización, álgebra lineal, integración, interpolación entre otras tareas para la ciencia e ingeniería.
- NumPy es una biblioteca que da soporte para crear vectores y matrices grandes multidimensionales, junto con una gran colección de funciones matemáticas de alto nivel para operar con ellas.
- Matplotlib es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays en Python.
- Pandas es una biblioteca de software escrita como extensión de Numpy para manipulación y análisis de datos. En particular ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales.

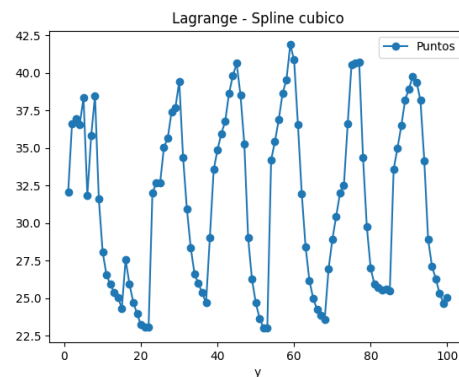


Fig. 2. Interpretación de 100 puntos de temperatura interna con métodos de Lagrange y spline cubico.

```

import sympy as sp
import matplotlib.pyplot as plt

def intLagrange(x,y,u=None):
    n=len(x)
    if u==None:
        t=sp.Symbol('t')

    else:
        t=u
        p=0
        for i in range(0,n):
            l=1
            for j in range(0,n):
                if j!=i:
                    l=l*(t-x[j])/(x[i]-x[j])
            p=p+y[i]*l
        p=sp.expand(p)
        return p

x=[]
y=[]
p=intLagrange(x,y)
print(p) #Polinomio de interpolacion
a=intLagrange(x,y,100) #Evaluar el polinomio en otro punto
print(a)

plt.plot(x,y,'o-', label = 'Puntos')
plt.legend()
plt.xlabel('x')
plt.ylabel('y')
plt.title('spline cubico')
plt.show()

```

Fig. 3. Código interpolación de Lagrange.

En el código de la figura 3, se recibe un conjunto de puntos y entrega el polinomio de interpolación en forma algebraica usando la fórmula de Lagrange. Opcionalmente, si la función recibe como parámetro adicional el valor a interpolar, el resultado entregado es un valor numérico, resultado de la interpolación. Para complementar un poco disposición del método, le implementamos la librería matplotlib con el fin de graficar dichos puntos dados.

```

from scipy import interpolate
import numpy as np
import matplotlib.pyplot as plt

def f(x, xp, xy):
    tck = interpolate.splrep(xp, xy)
    return interpolate.splev(x, tck)

x = []
y = []
print(f(20,x,y))

plt.plot(x,y,'o-', label = 'Puntos')
plt.legend()
plt.xlabel('x')
plt.ylabel('y')
plt.title('spline cubico')
plt.show()

```

Fig. 4. Código interpolación con splines cúbicos.

En la figura 4, se hace uso de una función la cual recibe como parámetros una variable independiente, y dos vectores en los cuales están definidos los datos a interpolar. Después de quemar dichos datos de las variables, la función permite determinar, gracias a el valor de la variable independiente, un resultado parametrizado dentro de los datos interpolados. Para nuestro caso, decidimos agregarle nuevamente la función matplotlib con el fin de que grafique dichos puntos determinados

```

#REALIZADO POR:
#ANDRES RICARDO PORRAS ESCOBAR
#CRISTIAN CAMILO BENITEZ
#JUAN ALEJANDRO DIAZ LOTE

#IMPORTACIONES
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import InterpolatedUnivariateSpline

#DECLARACIONES
xls=pd.ExcelFile('pruebasTemperatura.xlsx')
datos=xls.parse('Hoja1')

#INTERPOLACION
x= np.linspace(min(datos['DataN']), max(datos['DataN']), num=10001)
yld= np.interp(x,datos['DataN'],datos['TemperaturaSelect'])
ysp= InterpolatedUnivariateSpline(datos['DataN'],datos['TemperaturaSelect'])(x)

plt.plot(datos['DataN'],datos['TemperaturaSelect'],'o',mcw=2)
plt.plot(x,yld)
plt.plot(x,ysp)
leg = plt.legend(['Datos','Lineal',' Spline Cubico'])
leg.get_frame().set_facecolor('#fafafa')
plt.show()

```

Fig. 5. Código Spline univariante interpolado.

Gracias al código evidenciado en la figura 5, Se procedió a leer un archivo en Excel en donde se encontraban definidos los datos de la temperatura interna, ejecutando el algoritmo obtuvimos los siguientes resultados (véase figura 6, 7 y 8):

130, 217, 720

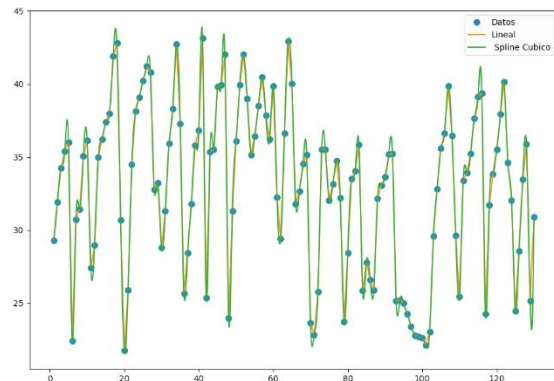


Fig. 6. Código interpolación de Lagrange.

En la figura 6, se obtuvo dicho grafico mediante la disposición de 130 puntos deferentes de temperatura interna.

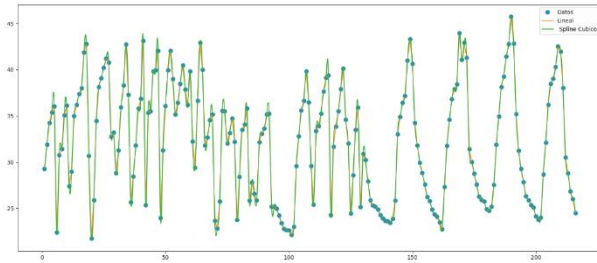


Fig. 7. Código interpolación de Lagrange.

En la figura 7, se obtuvo dicho grafico mediante la disposición de 217 puntos deferentes de temperatura interna.

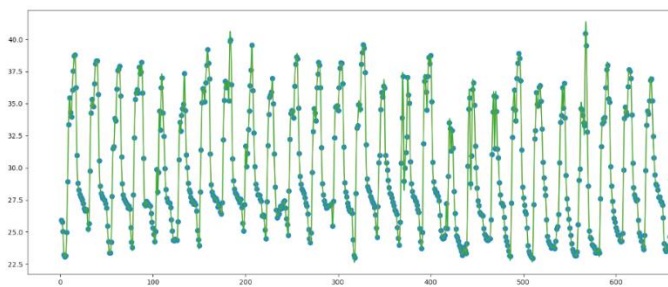


Fig. 8. Código interpolación de Lagrange.

En la figura 8, se obtuvo dicho grafico mediante la disposición de 720 puntos deferentes de temperatura interna.

Ahora para analizar lo errores de cada gráfica, el proceso que se realizo fue tomar varios puntos del excel dado por la profesora, estos puntos se usaron al momento de realizar la interpolación lo cual permitió; restar el punto que teníamos en el excel y con el resultado dado por la gráfica. Luego se tomó el valor absoluto, un ejemplo de esto es para la temperatura con valor 38,32 la gráfica nos muestra 38,15 nos da un error de 0,17; este proceso se repitió para varios puntos de la gráfica donde el error era muy similar.

III. CONCLUSIONES.

Para este reto podemos concluir que con los primeros métodos en este tipo de problemas se pueden presentar varias dificultades debido a que métodos como LaGrange nos dan curvas suaves, lo que significa que se puede obtener un error bastante grande, por lo cual para este tipo de datos es mejor hacer uso de métodos más precisos como lo sería en este caso el método de splines el cual presenta una mayor tolerancia a los picos que se presentan en la gráfica al momento de interpolar.

Referencias

- [1] «<https://www.analyticslane.com/2018/07/30/guardar-y-leer-archivos-excel-en-python/>,» [En línea].
- [2] «<https://es.stackoverflow.com/questions/103733/cómo-puedo-tomar-información-de-excel-en-python-y-guardarla-en-arreglos>,» [En línea].
- [3] «<https://living-sun.com/es/python/684428-interpolating-climate-data-with-irregular-measurement-intervals-in-python-with-pandas-and-traces-python-pandas.html>,» [En línea].
- [4] «<http://research.iac.es/sieinvens/python-course/source/scipy.html>,» [En línea].
- [5] «<https://www.analyticslane.com/2018/07/30/guardar-y-leer-archivos-excel-en-python/>,» [En línea].
- [6] «<https://living-sun.com/es/python/684428-interpolating-climate-data-with-irregular-measurement-intervals-in-python-with-pandas-and-traces-python-pandas.html>,» [En línea].