

```

import numpy as np
import matplotlib.pyplot as plt
import sympy as sym

def interpola_dfinitas(xi, fi):
    # Tabla de diferencias finitas
    titulo = ['i', 'xi', 'fi']
    n = len(xi)
    # cambia a forma de columnas
    i = np.arange(1, n + 1, 1)
    i = np.transpose([i])
    xi = np.transpose([xi])
    fi = np.transpose([fi])
    # Añade matriz de diferencias
    dfinita = np.zeros(shape=(n, n), dtype=float)
    tabla = np.concatenate((i, xi, fi, dfinita), axis=1)
    # Sobre matriz de diferencias, por columnas
    [n, m] = np.shape(tabla)
    c = 3
    diagonal = n - 1
    while (c < m):
        # Aumenta el título para cada columna
        titulo.append('df' + str(c - 2))
        # calcula cada diferencia por fila
        f = 0
        while (f < diagonal):
            tabla[f, c] = tabla[f + 1, c - 1] - tabla[f,
c - 1]
            f = f + 1

        diagonal = diagonal - 1
        c = c + 1

```

```

# POLINOMIO con diferencias finitas
# caso: puntos en eje x equidistantes
dfinita = tabla[:, 3:]
n = len(dfinita)
x = sym.Symbol('x')
h = xi[1, 0] - xi[0, 0]
polinomio = fi[0, 0]
for c in range(1, n, 1):
    denominador = np.math.factorial(c) * (h ** c)
    factor = dfinita[0, c - 1] / denominador
    termino = 1
    for f in range(0, c, 1):
        termino = termino * (x - xi[f])
    polinomio = polinomio + termino * factor
# simplifica polinomio, multiplica los (x-xi)
polinomio = polinomio.expand()
#np.set_printoptions(precision=2)
return (polinomio)

```

```

#yi=[10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70]
#vi=[108, 123, 139, 188, 275, 378, 441, 469, 552, 670,
701, 716, 858]

```

```

yi1=[20, 25, 30, 35, 40 ] #se utiliza interpolación por
diferencia finitas avanzadas
vi1=[139, 188, 275, 378, 441]

```

```

polinomio11=interpola_dfinitas(yi1,vi1)

```

```

# SALIDA
print("Con 5 puntos usando diferencias finitas
avanzadas")

```

```

print(polinomio11)
#plt.title("Con 5 puntos usando diferencias finitas
avanzadas")
plt.title("Velocidad vs Voltaje")
plt.plot(yi1,vi1, "o", color="green")
plt.xlabel("Velocidad km/h")
plt.ylabel("Voltaje mV")
plt.plot(yi1,vi1,'o', label = 'Puntos')
plt.plot(yi1,vi1, label = 'Polinomio')
plt.legend()
plt.plot(yi1,vi1)
plt.show()

```

```

def interpola_lagrange(xi,yi):
    #Interpolación con método de Lagrange resultado:
    polinomio en forma simbólica
    # PROCEDIMIENTO
    n = len(xi)
    x = sym.Symbol('x')
    # Polinomio
    polinomio = 0
    for i in range(0,n,1):
        # Termino de Lagrange
        termino = 1
        for j in range(0,n,1):
            if (j!=i):
                termino =
termino*(x-xi[j])/(xi[i]-xi[j])
        polinomio = polinomio + termino*yi[i]
    # Expande el polinomio
    polinomio = polinomio.expand()
    return(polinomio)

```

```
yi2=[0, 0.05, 0.11, 0.16, 0.21, 0.26, 0.32, 0.38, 0.44]#  
se utiliza lagrange porque los desplazamientos no son  
iguales  
vi2=[0, 5, 10, 15, 20, 25, 30, 35, 40]  
  
polinomio12=interpola_lagrange(yi2,vi2)  
  
# SALIDA  
#print("Con 5 puntos usando lagrange")  
print("Interpolación inversa:")  
print(polinomio12)  
plt.title("Voltaje vs Velocidad")  
plt.plot(yi2,vi2, "o", color="green")  
plt.ylabel("Velocidad km/h")  
plt.xlabel("Voltaje mV")  
plt.plot(yi2,vi2,'o', label = 'Puntos')  
plt.plot(yi2,vi2, label = 'Polinomio')  
plt.legend()  
plt.plot(yi2,vi2)  
plt.show()
```