import numpy as np

import matplotlib.pyplot as plt

from scipy.interpolate import lagrange

x=np.array([-4, -1, 1,2])

y=np.array([-6, 3, 11, -6])

def lagranz(x,y,t):

z = 0

for j in range(len(y)):

p1 = 1; p2 = 1

for i in range(len(x)):

if i==j:

p1=p1\*1; p2=p2\*1

else:

p1=p1\*(t-x[i])

p2=p2\*(x[j]-x[i])

z=z+y[j]\*p1/p2

return z

x\_new = np.linspace(np.min(x),np.max(x),100)

y\_new = [lagranz(x,y,i) for i in x\_new]

plt.plot(x,y,'o', x\_new, y\_new)

plt.show()

poly = lagrange(x, y)

print(poly)

x1 = -3

yp = 0

n=3

for i in range(n):

p = 1

for j in range(n):

if i != j:

p = p \* (x1 - x[j])/(x[i] - x[j])

yp = yp + p \* y[i]

print('Interpolated value at ', x1, ' = ', yp)

x2 = -2

for i in range(n):

p = 1

for j in range(n):

if i != j:

p = p \* (x2 - x[j])/(x[i] - x[j])

yp = yp + p \* y[i]

print('Interpolated value at ',x2 , ' = ', yp)

x3 = -0.5

for i in range(n):

p = 1

for j in range(n):

if i != j:

p = p \* (x3 - x[j])/(x[i] - x[j])

yp = yp + p \* y[i]

print('Interpolated value at ',x3 , ' = ', yp)

x4 = 2

for i in range(n):

p = 1

for j in range(n):

if i != j:

p = p \* (x4 - x[j])/(x[i] - x[j])

yp = yp + p \* y[i]

print('Interpolated value at ',x4 , ' = ', yp)

from scipy.interpolate import lagrange

f = lagrange(x, y)

fig = plt.figure(figsize = (10, 8))

plt.plot(x\_new, f(x\_new), 'b', x, y, 'ro')

plt.title('Langrage Polynomial')

plt.grid()

plt.xlabel('x')

plt.ylabel('y')

plt.show()