Лабораторна робота №3

*Наближення функцій. Інтерполяційний багаточлен Лагранжа*

Виконала Гальчинська Софія, студентка ФІТ 2-8. Варіант 4.

[*https://github.com/GalchynskaSofiia/Numerical\_Methods\_Of\_Programming*](https://github.com/GalchynskaSofiia/Numerical_Methods_Of_Programming)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Задана таблиця значень функції** | | | | |
| ***xi*** | -1 | 0 | 1 | 2 |
| ***f(xi)*** | -20 | -5 | 6 | 25 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Задані точки** | | | | |
| ***xi*** | -0,5 | 0,5 | 1,5 | 2,5 |
| ***f(xi)*** | ? | ? | ? | ? |

**Обчислення інтерполяційного багаточлена та його значень у заданих точках:**

import numpy as np

import matplotlib.pyplot as plt

from scipy.interpolate import lagrange

x=np.array([-1.,0.,1.,2.], dtype=float)

y=np.array([-20.,-5.,6.,25.], dtype=float)

x1\_test = -0.5

x2\_test = 0.5

x3\_test = 1.5

x4\_test = 2.5

def lagrange\_interpolation(x, y, x1\_test):

n = len(x)

p = np.zeros(n)

for i in range(n):

p\_i = 1

for j in range(n):

if i != j:

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

p[i] = p\_i

return np.dot(y, p)

f\_interp = lagrange\_interpolation(x, y, x1\_test)

print("Значення функції у точці f(-0.5) =", f\_interp.round(4))

f\_interp = lagrange\_interpolation(x, y, x2\_test)

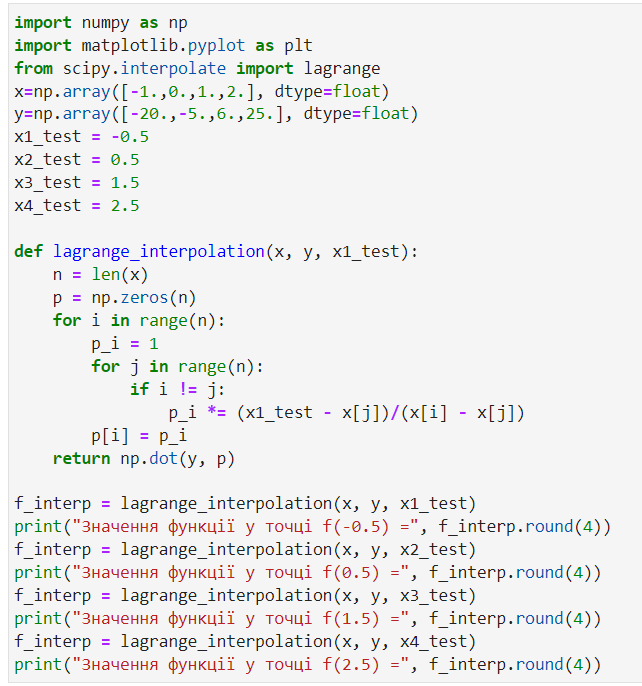
print("Значення функції у точці f(0.5) =", f\_interp.round(4))

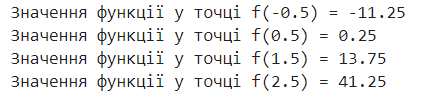
f\_interp = lagrange\_interpolation(x, y, x3\_test)

print("Значення функції у точці f(1.5) =", f\_interp.round(4))

f\_interp = lagrange\_interpolation(x, y, x4\_test)

print("Значення функції у точці f(2.5) =", f\_interp.round(4))





**Побудова графіка функції Лангранжа:**

xnew=np.linspace(np.min(x),np.max(x),100)

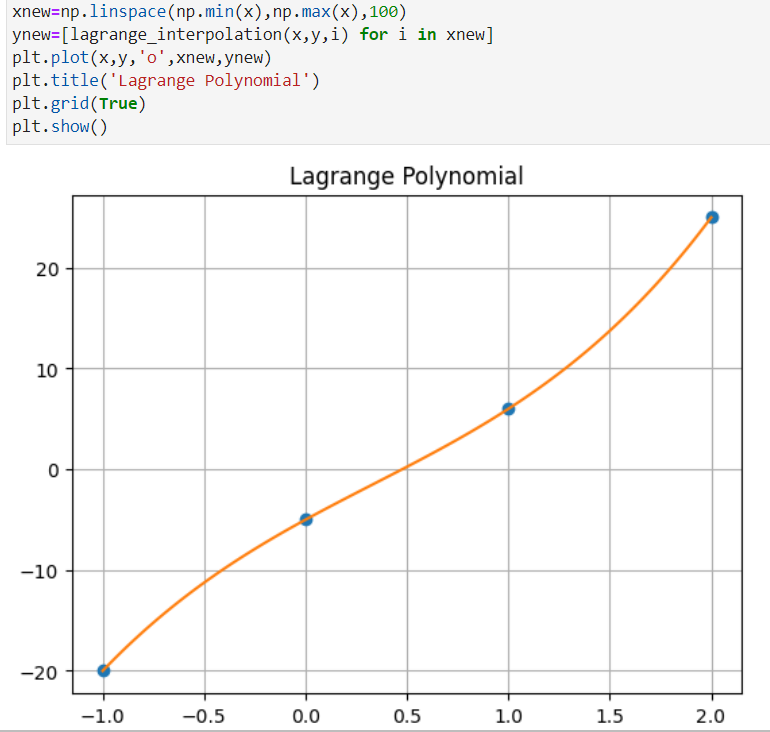
ynew=[lagrange\_interpolation(x,y,i) for i in xnew]

plt.plot(x,y,'o',xnew,ynew)

plt.title('Lagrange Polynomial')

plt.grid(True)

plt.show()



**Перевірка:**

f = lagrange(x, y)

fig = plt.figure(figsize = (7,5))

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

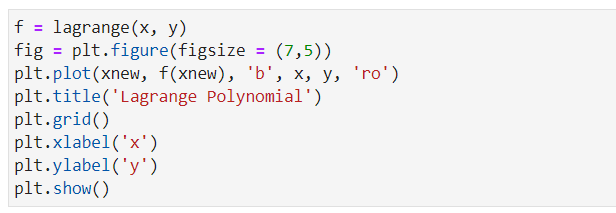
plt.title('Lagrange Polynomial')

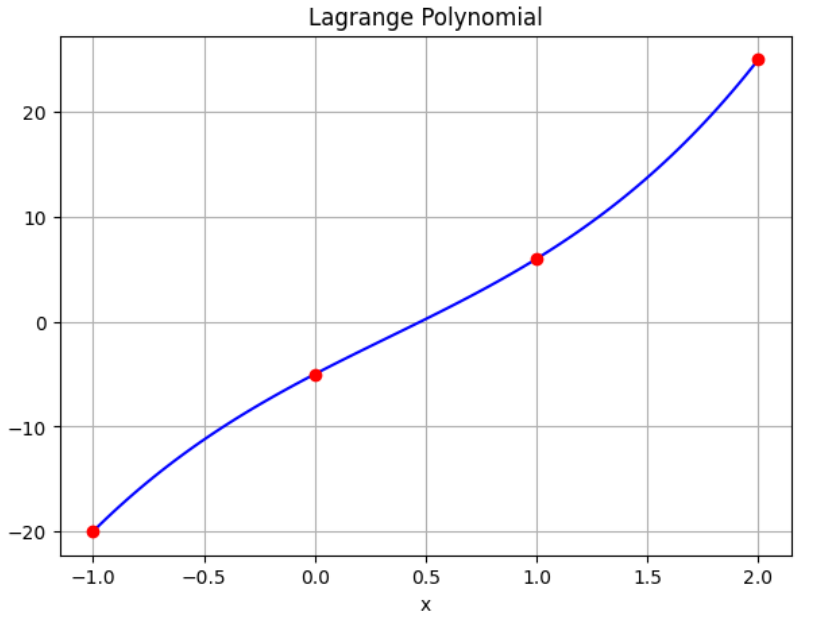
plt.grid()

plt.xlabel('x')

plt.ylabel('y')

plt.show()





Висновки: Отже, під час виконання практичної роботи ми навчились будувати інтерполяційний багаточлен Лагранжа для функції, що задана таблицею, а також обчислювати наближені значення функції в заданих точках. Побудова була проведена за допомогою програмування, зокрема використання Python для обчислення багаточлена Лагранжа та його значень у заданих точках. Також було побудовано графік полінома Лагранжа з відображенням на графіку точок (*x, f(x)*). У результаті перевірки всі значення співпали, що означає правильність виконання завдання. Тож після проведених обчислень маємо:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Задані точки** | | | | |
| ***xi*** | -0,5 | 0,5 | 1,5 | 2,5 |
| ***f(xi)*** | -11,25 | 0,25 | 13,75 | 41,25 |