**Бігун Максим Сергійович, ФІТ 2-7, Варіант 2**

**Завдання:** Побудувати інтерполяційний багаточлен Лагранжа для функції , що задана таблицею:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | -1 | 0 | 2 | 3 |
|  | -16 | -7 | -1 | 20 |

Знайти наближене значення функції у заданих точках з точністю до 0.001.

Побудувати графік інтерполяційної функції за наявним набором точок.

*Розв’язання:*

У випадку розрахункова формула матиме вигляд:

Отже:

Знаходимо значення заданих точок:

# Задамо функцію

def f(x):

    return 2\*x\*\*3 - 4\*x\*\*2 + 3\*x - 7

# Задаємо точки для інтерполяції

x\_custom = np.array([-0.5, 0.5, 1.5, 2.5], dtype=float)

# Знаходимо значення функції в точках

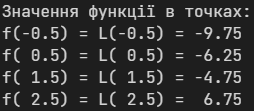
y\_custom = [f(i) for i in x\_custom]

# Виводимо результат

print("Значення функції в точках:")

for i in range(len(x\_custom)):

    print(f"f({x\_custom[i]: }) = L({x\_custom[i]: }) = {y\_custom[i]: }")



Побудуємо графік інтерполяційної функції за наявним набором точок використовуючи її програмну реалізацію:

# Імпортуємо бібліотеки

import numpy as np

import matplotlib.pyplot as plt

# Задаємо точки

xi = np.array([-1, 0, 2, 3], dtype=float)

yi = np.array([-16, -7, -1, 20], dtype=float)

# Функція Лагранжа

def lagranzha(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 \*= 1

                p2 \*= 1

            else:

                p1 \*= (t - x[i])

                p2 \*= (x[j] - x[i])

        z += y[j] \* (p1 / p2)

    return z

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

y\_new = [lagranzha(xi, yi, i) for i in x\_new]

# Виводимо графік

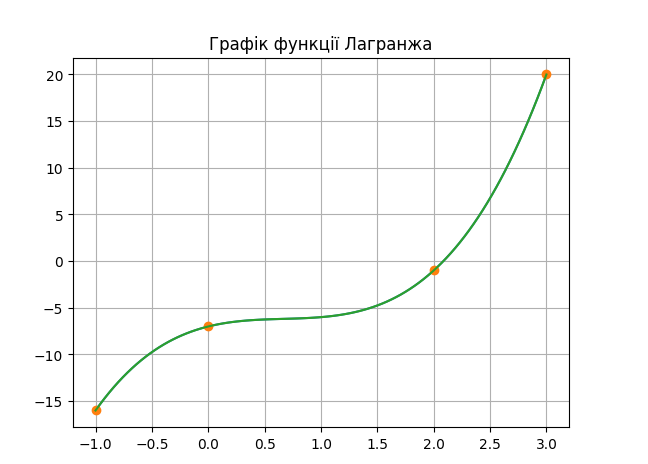
plt.plot(x\_new, y\_new)

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

plt.title('Графік функції Лагранжа')

plt.grid(True)

plt.show()



Перевіримо за допомогою бібліотеки SciPy:

# Імпортуємо lagrange

from scipy.interpolate import lagrange

#Перевірка

f = lagrange(xi, yi)

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

plt.title('Графік функції Лагранжа (SciPy)')

plt.grid()

plt.xlabel('x')

plt.ylabel('y')

plt.show()

