МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Інститут **КНІТ**
Кафедра **ПЗ**

ЗВІТ

До лабораторної роботи № 9
На тему: “*Наближення функцій методом найменших квадратів*”
З дисципліни: “Чисельні методи”

Лектор:
доцент кафедри ПЗ
Мельник Н.Б.

Виконав:
студент групи ПЗ-16
Коваленко Д.М.

Прийняв:
асистент кафедри ПЗ
Гарматій Г.Ю.

«_____» _______ 2022 р.
$\sum =$ _____................

Львів — 2022

Тема. Наближення функцій методом найменших квадратів.

Мета. Ознайомлення на практиці з методом найменших квадратів апроксимації (наближення) функцій.

Теоретичні відомості

Інтерполяційний поліном Лагранжа

Наближену функцію $y = f(x)$ представимо у вигляді

$$\varphi(x) = L_n(x) = \sum_{i=0}^{n} P_i(x) f(x_i),$$

де $P_i(x)$ такий многочлен, що

$$P_i(x_j) = \begin{cases} 0, i \neq j \\ 1, i = j \end{cases}, \qquad i, j = 0, 1...n.$$

Оскільки точки $x_0, x_1, x_{i-1}, x_{i+1}, ..., x_n$ є коренями многочлена $P_i(x)$, то його можна записати наступним чином

$$P_i(x) = \frac{(x-x_0)(x-x_1)...(x-x_{i-1})(x-x_{i+1})...(x-x_n)}{(x_i-x_0)(x_i-x_1)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_n)}$$

а наближена функція, яку називають інтерполяційним многочленом Лагранжа, матиме вигляд

$$\varphi(x) = L_n(x) = \sum_{i=0}^{n} \frac{(x-x_0)(x-x_1)...(x-x_{i-1})(x-x_{i+1})...(x-x_n)}{(x_i-x_0)(x_i-x_1)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_n)} f(x_i)$$

Поліном Лагранжа для випадку рівновіддалених вузлів має вигляд

$$L_n(x) = \frac{1}{n!} \Pi_{n+1}(t) \sum_{i=0}^{n} (-1)^{n-i} \frac{C_n^i}{t-i} y_i$$

де

$$C_n^i = \frac{n!}{i!(n-i)!}$$

Інтерполяційний поліном Ньютона

Для загального випадку нерівновіддалених вузлів поліном $P_i(x)$ записують у вигляді

$$P_n(x) = f(x_0) + f(x_0, x_1)(x-x_0) + f(x_0, x_1, x_2)(x-x_0)(x-x_1) + \\ + ... + f(x_0, x_1, ..., x_n)(x-x_0)(x-x_1)...(x-x_{n-1}),$$

де скінченна різниця n-го порядку

$$\Delta^n f(x_i) = \sum_{j=0}^{n} (-1)^j C_n^j f(i+j)$$

Лабораторне завдання

Використовуючи інтерполяційні поліноми Лагранжа та Ньютона, обчислити значення табличної заданої функції у точці $x_0 = 0.455$.

x	0.45	0.46	0.47	0.48	0.49	0.50	0.51	0.52	0.53	0.54
y	20.19	19.61	18.94	18.17	17.30	16.31	15.19	13.94	12.55	10.99

Хід роботи

У даної функції усі вузли x рівновіддалені з кроком 0.01.

x_i	$f(x_i)$	$\Delta f(x_i)$	$\Delta^2 f(x_i)$	$\Delta^3 f(x_i)$	$\Delta^4 f(x_i)$
x_0	20.19	-0.58	-0.09	0.19	-0.19
x_1	19.61	-0.67	0.1	0	-
x_2	18.94	-0.77	0.1	-	-
x_3	18.17	-0.87	-	-	-
x_4	17.30	-	-	-	-

Інтерполяційний поліном Лагранжа

Використаю формулу

$$L_n(x) = \frac{1}{n!}\Pi_{n+1}(t)\sum_{i=0}^{n}(-1)^{n-i}\frac{C_n^i}{t-i}y_i$$

для знаходження полінома та підставлю $x_0 = 0.455$ в знайдений поліном щоб знайти значення фукції в цій точці.

Код програми *(файл lab_91.py):*

```
import numpy as np

def data_to_matrix(path):
    return (
        np.loadtxt(open(path, "rb"), delimiter=",")[0],
        np.loadtxt(open(path, "rb"), delimiter=",")[1],
    )

def lagrange(x, y, x0):
    n = np.shape(x)[0]
    yp = 0
    for i in range(n):
        p = 1
        for j in range(n):
            if i != j:
                p *= (x0 - x[j])/(x[i] - x[j])
        yp += p * y[i]
    return yp

path = input("Введіть шлях до файлу з даними: ") or "data.csv"
x, y = data_to_matrix(path)
print(f"Результат полінома Лагранжа у точці x0=0.455: {round(lagrange(x,
    y, 0.455), 4)}.")
```

Інтерполяційний поліном Ньютона

Для знаходження полінома та підставлю $x_0 = 0.455$ в знайдений поліном щоб знайти значення фукції в цій точці.

Код програми *(файл lab_92.py):*

```
import numpy as np

def data_to_matrix(path):
    return (
        np.loadtxt(open(path, "rb"), delimiter=",")[0],
        np.loadtxt(open(path, "rb"), delimiter=",")[1],
    )

def newton(x, y, r):
    n = len(x)
    a = [y[i] for i in range(n)]
    for j in range(1, n):
        for i in range(n-1, j-1, -1):
            a[i] = (a[i] - a[i-1]) / (x[i] - x[i-j])

    n = len(a) - 1
    temp = a[n] + (r - x[n])
    for i in range(n - 1, -1, -1):
        temp = temp * (r - x[i]) + a[i]
    return temp

path = input("Введіть шлях до файлу з даними: ") or "data.csv"
x, y = data_to_matrix(path)
print(f"Результат полінома Ньютона у точці x0=0.455: {round(newton(x, y,
    0.455), 4)}.")
```

Файл даних *(data.csv):*

```
0.45, 0.46, 0.47, 0.48, 0.49, 0.50, 0.51, 0.52, 0.53, 0.54
20.19, 19.61, 18.94, 18.17, 17.30, 16.31, 15.19, 13.94, 12.55, 10.99
```

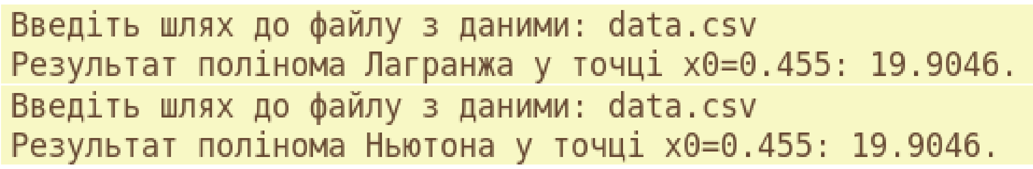

Рис. 1: Робота програми

Висновок

На лабораторній роботі я засвоїв практичні навички використання полінома Лагранжа та Ньютона для обчислення таблично заданої функції

x	0.45	0.46	0.47	0.48	0.49	0.50	0.51	0.52	0.53	0.54
y	20.19	19.61	18.94	18.17	17.30	16.31	15.19	13.94	12.55	10.99

у заданій точці $x_0 = 0.455$ та отримав результат 19.9046