Міністерство освіти і науки України
Національний університет “Львівська політехніка”
Інститут комп’ютерних наук та інформаційних технологій
Кафедра програмного забезпечення

Звіт

Про виконання лабораторної роботи №8

На тему:

«Наближення дискретних (таблично заданих) функцій»
з дисципліни «Чисельні методи»

Лекторка:
доцент каф. ПЗ
Мельник Н. Б.

Виконав:
ст. гр. ПЗ-11
Ясногородський Н.В.

Прийняла:
доцент каф. ПЗ
Мельник Н. Б.

« __ » ________ 2022 р.

∑ = _____ _________ .

Львів – 2022

Тема: Наближення дискретних (таблично заданих) функцій.
Мета: Ознайомитися з методом інтерполяції таблично заданих функцій.

Теоретичні відомості

Інтерполяційний поліном Лагранжа – поліном, в основу якого покладено те, що в одному довільному вузлі інтерполяції поліном приймає значення одиниці, а у всіх інших – нуль. Наближена функція матиме вигляд $F(x) = \sum_{i=0}^{n} P_i(x) f(x_i)$, де

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

Звідки отримаємо інтерполяційний многочлен Лагранжа

$$F(x) = \sum\nolimits_{i=0}^{n} \frac{\left(x - x_0\right)\left(x - x_1\right)\ldots\left(x - x_{i-1}\right)\left(x - x_{i+1}\right)\ldots\left(x - x_n\right)}{\left(x_i - x_0\right)\left(x_i - x_1\right)\ldots\left(x_i - x_{i-1}\right)\left(x_i - x_{i+1}\right)\ldots\left(x_i - x_n\right)} f(x_i)$$

Інтерполяційний поліном Ньютона – використовує дещо інший принцип побудови:

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

Де $f\left(x_0, x_1, \ldots, x_n\right) = \frac{f\left(x_1, \ldots, x_n\right) - f\left(x_0, \ldots, x_{n-1}\right)}{x_n - x_0}$ – розділена різниця n-го порядку для нерівновіддалених вузлів.

Індивідуальне завдання

Використовуючи інтерполяційні поліноми Лагранжа та Ньютона, обчислити значення табличної заданої функції у точці x0:

15-й варіант

x	0,115	0,120	0,125	0,130	0,135	0,140	0,145	0,150	0,165	0,170
y	4,48	4,95	5,47	5,99	6,05	6,68	6,909	7,38	8,166	9,025

$x_0 = 0{,}142$

Код функцій

```
import numpy as np

points = (
    np.array(
        [
            0.115,
            0.12,
            0.125,
            0.13,
            0.135,
            0.14,
            0.145,
            0.15,
            0.165,
            0.17,
        ]
    ),
    np.array([4.48, 4.95, 5.47, 5.99, 6.05, 6.68, 6.909, 7.38, 8.166, 9.025]),
)
x0 = 0.142


def lagrange_at_x(points, x0):
    print(f"\nLagrange method for x0={x0}:")
    x_points, y_points = points

    n = len(x_points)
```

```
    yp = 0  # interpolated value

    for i in range(n):
        p = 1
        for j in range(n):
            if i == j:
                continue
            p *= (x0 - x_points[j]) / (x_points[i] - x_points[j])
        print(f"step: {i+1}, y={yp}")
        yp += p * y_points[i]

    print(f"y0={yp}")
    return yp


def newton_at_x(points, x0):
    print(f"\nNewton method for x0={x0}:")
    x_points, y_points = points
    m = len(x_points)
    a_coff = np.copy(y_points)

    for k in range(1, m):
        a_coff[k:m] = (a_coff[k:m] - a_coff[k - 1]) / (x_points[k:m] - x_points[k -
1])

    n = len(x_points) - 1  # Degree of polynomial
    yp = a_coff[n]

    for k in range(1, n + 1):
        yp = a_coff[n - k] + (x0 - x_points[n - k]) * yp
        print(f"step: {k}, y={yp}")

    print(f"y0={yp}")
    return yp


if __name__ == "__main__":
    lagrange_at_x(points, x0)
    newton_at_x(points, x0)
```

Протокол роботи

```
Initial points:

X:
0.115   0.12    0.125   0.13    0.135   0.14    0.145   0.15    0.165   0.17

Y:
4.48    4.95    5.47    5.99    6.05    6.68    6.909   7.38    8.166   9.025

Lagrange method for x0=0.142:
Result: y0=6.877079707955199

Newton method for x0=0.142:

Newton differences table (first row):
4.48    94.00000000000013       999.9999999999707       -66666.66666666362      -27333333.3333335       5066666666.666663       -479199999999.99963     31738412698412.668      -10
68774603174601.4        3.1283116883116772e+16

Result: y0=6.877079707955198
```

Рис.1. Робота програми

Висновки

Виконуючи лабораторну роботу №8, я ознайомився з методом інтерполяції таблично заданих функцій, та склав програму для інтерполяції методом Лагранжа та методом Ньютона.