Міністерство освіти і науки України

Національний університет «Львівська Політехніка»

Інститут прикладної математики та фундаментальних наук

Кафедра прикладної математики

**Звіт**

про виконання розрахункової роботи №2

з дисципліни: «Чисельні методи ч.1»

на тему: «Дослідження похибки інтерполяції функції многочленом Ньютона»

**Виконав:**

студент групи ПМ-31

Богуш Н. А.

**Перевірив:**

канд. фіз-мат. наук, доц.

Ярополк ПІЗЮР

Львів 2025

**Варіант 3**

**Постановка задачі:**

Для функції f (x) = заданої на рівномірній сітці в точках a = 0, , , , , , π = b на інтервалі [0, π] :

1. Знайти аналітичний вираз залишкового члена інтерполяційного многочлена;
2. Обчислити його максимальне значення і значення в точках (не співпадають з вузлами інтерполяції) і в т. – вузлі інтерполяції;
3. Побудувати інтерполяційний многочлен Ньютона через розділені різниці (використати програму з лабораторної роботи №5);
4. Обчислити фактичні похибки інтерполяції в точках і порівняти їх із значеннями залишкового члена.

**Програма в середовищі Maple з дослідженням залишкового члена інтерполяційного многочлена та обчисленням його у точках та**   **- вузлі інтерполяції.**

* Ініціалізовуємо вхідні дані f(x), n, a, b

Изображение выглядит как текст, снимок экрана, Шрифт, линия

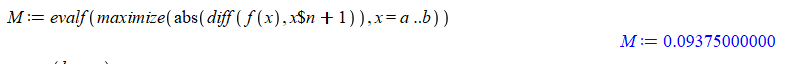
Содержимое, созданное искусственным интеллектом, может быть неверным.

* Обчислюємо похідну n+1-го порядку та будуємо її графік

Изображение выглядит как текст, линия, График, диаграмма

Содержимое, созданное искусственным интеллектом, может быть неверным.

* Максимум модуля похідної на відрізку [a; b]



Отримали значення M = 0.09375

* Обчислюємо крок h

Изображение выглядит как линия, диаграмма, белый, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

* Знаходимо аналітичний вираз многочлена w(x) та будуємо його графік

Изображение выглядит как текст, Шрифт, рукописный текст, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

Изображение выглядит как диаграмма, линия, текст, График

Содержимое, созданное искусственным интеллектом, может быть неверным.

* Максимум модуля w(x) на відрізку [a; b]



Значення Wmax = 1.034060113

* Максимум залишкового члену



Значення R = 0.0000192347

* Обчислюємо значення залишкового члена в т. x1

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

* Обчислюємо значення залишкового члена в т. x2

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

* Обчислюємо значення залишкового члена в т. x3

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

* Обчислюємо значення залишкового члена у вузлі інтерполяції x4

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

**Лістинг програми в середовищі Visual Studio з побудовою інтерполяційного многочлена Ньютона через розділені різниці на мові програмування Python:**

import math

def build\_divided\_differences\_table(x, y):

    n = len(x)

    table = [[0.0 for \_ in range(n)] for \_ in range(n)]

    for i in range(n):

        table[i][0] = y[i]

    for j in range(1, n):

        for i in range(n - j):

            table[i][j] = (table[i + 1][j - 1] - table[i][j - 1]) / (x[i + j] - x[i])

    return table

def print\_divided\_differences\_table(x, table):

    n = len(x)

    spacing = 1

    col\_width = 12

    height = n + (n - 1) \* spacing + (n - 1)

    visual\_table = [["".ljust(col\_width) for \_ in range(n + 1)] for \_ in range(height)]

    header = ["x".center(col\_width)] + [f"Δ^{i}f".center(col\_width) for i in range(n)]

    print(" | ".join(header))

    print("-" \* ((col\_width + 3) \* (n + 1)))

    for i in range(n):

        row\_idx = i \* (spacing + 1)

        visual\_table[row\_idx][0] = f"{x[i]:.2f}".rjust(col\_width)

        visual\_table[row\_idx][1] = f"{table[i][0]:.6f}".rjust(col\_width)

    for j in range(1, n):

        for i in range(n - j):

            row\_idx = i \* (spacing + 1) + j

            visual\_table[row\_idx][j + 1] = f"{table[i][j]:.6f}".rjust(col\_width)

    for row in visual\_table:

        if any(cell.strip() for cell in row):

            print(" | ".join(row))

def print\_newton\_polynomial(x\_data, table):

    n = len(x\_data)

    terms = [f"{table[0][0]:.6f}"]

    for i in range(1, n):

        coef = table[0][i]

        if abs(coef) < 1e-12:

            continue

        term = f"{coef:+.6f}"

        for j in range(i):

            term += f"\*(x - {x\_data[j]:.2f})"

        terms.append(term)

    print("P(x) =")

    for t in terms:

        print("     " + t)

def newton\_interpolation(x\_data, table, x):

    n = len(x\_data)

    result = table[0][0]

    product = 1.0

    for i in range(1, n):

        product \*= (x - x\_data[i - 1])

        result += table[0][i] \* product

    return result

*# --- Основна частина ---*

*# Налаштування задачі*

a = 0.0

b = math.pi

n = 56

x\_nodes = [0.0, math.pi/6, math.pi/3, math.pi/2, 2\*math.pi/3, 5\*math.pi/6, math.pi]

y\_nodes = [12 \* math.cos(1/2 \* x) for x in x\_nodes]

*# Точки для обчислення*

points = [0.13281, 0.432987, 1.23892, math.pi/6]

*# Побудова таблиці розділених різниць*

table = build\_divided\_differences\_table(x\_nodes, y\_nodes)

print("\n=== Таблиця розділених різниць ===")

print\_divided\_differences\_table(x\_nodes, table)

print("\n=== Інтерполяційний многочлен Ньютона ===")

print\_newton\_polynomial(x\_nodes, table)

print("\n=== Обчислення у точках ===")

for x in points:

    true\_value = 12 \* math.cos(1/2 \* x)

    interpolated\_value = newton\_interpolation(x\_nodes, table, x)

    error = abs(true\_value - interpolated\_value)

    print(f"\nx = {x}")

    print(f"f(x)  = {true\_value:.6f}")

    print(f"P(x)  = {interpolated\_value:.6f}")

    print(f"Фактична похибка = {error:.6f}")

**Результат виконання програми:**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Изображение выглядит как текст, Шрифт, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным. Изображение выглядит как текст, Шрифт, снимок экрана, число

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Таблиця 1 - результуючі значення:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | **Значення фактичної помилки** | **Значення залишкового члена** |
| 0,13281 | 11.973552 | 11.973540 | 0.000012 | 0.00001861894883 |
| 0,432987 | 11.719880 | 11.719877 | 0.000003 | 0.00000501489 |
| 1.23892 | 9.770305 | 9.770304 | 0.000002 | 0.000002354470 |
| 0.523598775 | 11.591110 | 11.591110 | 0 | 0 |

**Максимум залишкового члена R(n): 0.00001923474913**

**Висновок:** У цій розрахунковій роботі було досліджено точність інтерполяції функції f(x) = 12 cos(1/2 \* x) за допомогою інтерполяційного многочлена Ньютона. Побудовано таблицю розділених різниць, виведено аналітичний вираз залишкового члена та виконано обчислення у точках, що не збігаються з вузлами інтерполяції. Значення фактичних похибок не перевищували значення залишкового члена, що дозволило оцінити точність методу. Аналіз також показав, що похибка у вузлі інтерполяції дорівнює нулю, що підтверджує правильність побудови інтерполяційного многочлена Ньютона. Отримані результати свідчать про коректність обчислень і ефективність застосованого методу для заданої функції.