

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



ЗВІТ

про виконання лабораторної роботи №4
з теми ЧИСЛОВЕ ОБЧИСЛЕННЯ ВИЗНАЧЕНИХ ІНТЕГРАЛІВ
з навчальної дисципліни: “Чисельні методи”

Виконав:

студент групи ІР-24

Шийка Андрій

Прийняв:

Сиротюк С. В.

Львів - 2025

Мета роботи: вивчити основні методи обчислення визначених інтегралів

Завдання:

1. Обчислити визначений інтеграл:
 - a. Інтеграл від 1 до 2 функції $f(x) = \sqrt{x^2 + 3} / x^2$
2. Точне значення обчислити за формулою Ньютона-Лейбніца, використовуючи первісну:
 - a. $F(x) = -(\sqrt{x^2 + 3}) / x + \ln(x + \sqrt{x^2 + 3})$
3. Чисельне інтегрування виконати методами:
 - a. Метод середніх прямокутників (кількість розбиттів $n = 30$)
 - b. Метод трапецій (кількість розбиттів $n = 30$)
 - c. Метод Сімпсона (кількість розбиттів $n = 2m = 20$, де $m = 10$)

Короткі теоретичні відомості:

У роботі використовуються такі методи наближеного обчислення визначеного інтеграла $I = \int[a, b] f(x) dx$:

1. Метод середніх прямокутників.

Проміжок $[a, b]$ розбивається на n рівних частин з кроком $h = (b-a)/n$. Висота прямокутника береться в середній точці кожного підінтервалу $x_i + h/2$.

Розрахункова формула:

$$I \approx h * \sum f(x_i + h/2), \text{ де } i \text{ від } 0 \text{ до } n-1.$$

Цей метод зазвичай точніший за методи лівих або правих прямокутників.

2. Метод трапецій.

Графік підінтегральної функції замінюється ламаною лінією (хордами). Площа фігури наближається сумою площ трапецій.

Розрахункова формула:

$$I \approx h * ((f(a) + f(b))/2 + \sum f(x_i)), \text{ де } i \text{ від } 1 \text{ до } n-1$$

3. Метод Сімпсона (парабол).

Підінтегральна функція на кожному подвійному кроці $2h$ замінюється параболою другого порядку. Метод вимагає парної кількості розбиттів $n = 2m$.

Розрахункова формула:

$$I \approx (h/3) * (f(a) + f(b) + 4 * \sum f(x_{\{2i-1\}}) + 2 * \sum f(x_{\{2i\}}))$$

Це один з найбільш точних методів для гладких функцій.

Список ідентифікаторів констант, змінних, функцій:

- a, b (float): нижня та верхня межі інтегрування (a=1.0, b=2.0).
- n (int): кількість розбиттів проміжку інтегрування.
- m (int): параметр для методу Сімпсона ($n = 2 * m$).
- h (float): крок інтегрування, $h = (b - a) / n$.
- integral (float): змінна для накопичення суми (квадратурної суми) в циклах.
- x (float): поточне значення аргументу функції.
- f(x): функція, що реалізує підінтегральний вираз.
- F(x): функція для обчислення первісної.
- exact_val (float): точне значення інтегралу (еталон).
- res_rect, res_trap, res_simp (float): результати обчислення інтегралу відповідними методами.
- err_rect, err_trap, err_simp (float): абсолютні похибки обчислень

Код програми:

```
import math

# Лабораторна робота №4. Числове інтегрування.
# Варіант 12.
# Функція: f(x) = sqrt(x^2 + 3) / x^2
# Проміжок: [1, 2]

def f(x):
    """Підінтегральна функція"""
    return math.sqrt(x**2 + 3) / (x**2)
```

```

def F(x):
    """Первісна функції (для точного розрахунку)"""
    return - (math.sqrt(x**2 + 3) / x) + math.log(x + math.sqrt(x**2 +
3))

def exact_integral(a, b):
    """Формула Ньютона-Лейбніца (1.2)"""
    return F(b) - F(a)

# -----
# 1. МЕТОД СЕРЕДНІХ ПРЯМОКУТНИКІВ
# -----

def method_middle_rectangles(a, b, n):
    integral = 0
    h = (b - a) / n

    # Цикл сумування значень у середніх точках
    for i in range(n):
        # x_mid - середина i-го відрізка: x_i + h/2
        x_mid = a + i * h + h / 2
        integral = integral + f(x_mid)

    # Остаточне множення на крок h
    result = integral * h
    return result

# -----
# 2. МЕТОД ТРАПЕЦІЙ
# -----

def method_trapezoidal(a, b, n):
    integral = 0
    h = (b - a) / n
    x = a + h
    fa = f(a)

```

```

fb = f(b)

for i in range(1, n):
    integral = integral + f(x)
    x = x + h

result = h * ((fa + fb) / 2 + integral)
return result

# -----
# 3. МЕТОД СІМПСОНА
# -----

def method_simpson(a, b, m):
    integral = 0
    n = 2 * m # Кількість розбиттів має бути парною
    h = (b - a) / n
    fa = f(a)
    fb = f(b)

    # Перший цикл (для непарних індексів: 1, 3, ..., 2m-1)
    # Множник 4
    for i in range(1, m + 1):
        x = a + (2 * i - 1) * h
        integral = integral + 4 * f(x)

    # Другий цикл (для парних індексів: 2, 4, ..., 2m-2)
    # Множник 2
    for i in range(1, m):
        x = a + 2 * i * h
        integral = integral + 2 * f(x)

    result = (h / 3) * (fa + fb + integral)
    return result

# -----

```

```

# Головна частина програми
# -----

def main():
    # Межі інтегрування
    a = 1.0
    b = 2.0

    # Точне значення
    exact_val = exact_integral(a, b)

    print(f"Точне значення (формула Ньютона-Лейбніца):
{exact_val:.10f}")

    print("-" * 60)

    print(f"{'Метод':<30} | {'Результат':<12} | {'Похибка':<12}")
    print("-" * 60)

    # 1. Метод середніх прямокутників (n=30)
    n_rect = 30
    res_rect = method_middle_rectangles(a, b, n_rect)
    err_rect = abs(res_rect - exact_val)

    print(f"{'Середніх прямокутників (n=30)':<30} | {res_rect:.10f} |
{err_rect:.2e}")

    # 2. Метод трапецій (n=30)
    n_trap = 30
    res_trap = method_trapezoidal(a, b, n_trap)
    err_trap = abs(res_trap - exact_val)

    print(f"{'Трапецій (n=30)':<30} | {res_trap:.10f} | {err_trap:.2e}")

    # 3. Метод Сімпсона (m=10 -> n=20)
    m_simpson = 10
    res_simp = method_simpson(a, b, m_simpson)
    err_simp = abs(res_simp - exact_val)

    print(f"{'Сімпсона (m=10, n=20)':<30} | {res_simp:.10f} |
{err_simp:.2e}")

    print("-" * 60)

```

```
if __name__ == "__main__":  
    main()
```

Результат:

```
na_p1_labs_iot_nulp on  lab/4 [?] via  v3.12.2 (.venv)  
> python lab4/lab4.py
```

Точне значення (формула Ньютона-Лейбніца): 1.1144651612

Метод	Результат	Похибка
Середніх прямокутників (n=30)	1.1143250574	1.40e-04
Трапецій (n=30)	1.1147454204	2.80e-04
Сімпсона (m=10, n=20)	1.1144665458	1.38e-06

Висновок:

В ході виконання лабораторної роботи було розроблено програму для числового обчислення визначеного інтеграла трьома методами: середніх прямокутників, трапецій та Сімпсона.

Аналіз результатів:

1. Точне значення інтегралу, обчислене аналітично, становить приблизно 1.1144651612.
2. Метод Сімпсона показав найвищу точність (похибка 1.38e-06) при найменшій кількості розбиттів (n=20), що підтверджує його ефективність (4-й порядок точності).
3. Метод середніх прямокутників (похибка 1.40e-04) виявився точнішим за метод трапецій (похибка 2.80e-04) при однаковій кількості розбиттів (n=30), що узгоджується з теоретичними очікуваннями.

Рекомендація: для досягнення високої точності при мінімальних обчислювальних витратах доцільно використовувати метод Сімпсона.