МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені ІГОРЯ СІКОРСЬКОГО»

Лабораторна робота №4

з дисципліни: «Чисельні методи»

16 Варіант

СПЕЦІАЛЬНОСТІ 121 – Інженерія програмного забезпечення

Виконав: Левак О. О.

Група: ІТ-91

Перевірила: Тимофєєва Ю.С.

Київ 2020

**Лабораторна робота №4. Чисельне інтегрування функцій**

**Мета:** навчитися чисельно інтегрувати функції за допомогою метода прямокутників, формули трапецій, формули Сімпсона та використовувати процедуру Рунге для оцінки похибки й уточнення формул чисельного інтегрування.

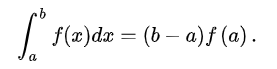
**ТЕОРЕТИЧНІ ВІДОМОСТІ**

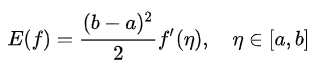
**Метод прямокутників**: найпростіший метод чисельного інтегрування, що полягає у заміні значень функції на проміжку значенням функції в деякій точці проміжку.

Види формули прямокутників:

Формула лівих прямокутників

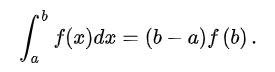
У цьому випадку береться значення функції на початку проміжку:



Похибка обчислення рівна: 

Формула правих прямокутників

У цьому випадку береться значення функції в кінці проміжку:

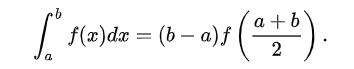


Як і в попередньому випадку похибка обчислень рівна:



Формула центральних прямокутників

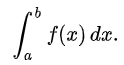
Ця формула має вид:



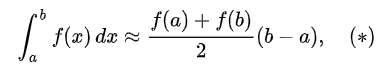
Похибка обчислень рівна:



**Метод трапецій**: метод наближеного обчислення значення визначеного інтегралу.

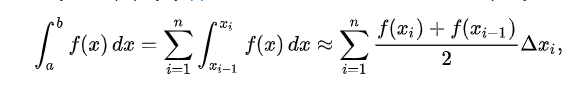


Ідея методу трапецій полягає в наближенні області під графіком функції f(x) трапецією та обчисленні її площі. Якщо застосувати цю ідею безпосередньо до інтервалу [a, b], то отримаємо



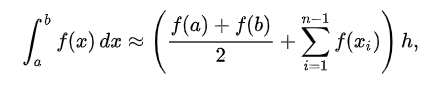
але це незадовільно через велику похибку.

Для точнішого обчислення значення інтегралу, слід попередньо розбити інтервал інтегрування [a, b] на n підінтервалів [a, x1], [x1, x2], … , [x(n-1), b] та застосувати формулу (\*) до кожного із них. Таким чином, отримуємо:



де 

У методі трапецій переважно застосується розбиття інтервалу інтегрування на n рівних відрізків довжиною h=Delta x=(b-a)/n. Тоді попередня формула перетворюється на таку:



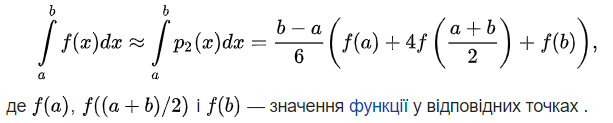
і похибка, так званий залишковий член E(f) не перевищує за



де  — це максимум другої похідної функції f(x) на всьому інтервалі. Відзначимо, що за збільшення числа n інтервалів розбиття, залишковий член зменшується як 

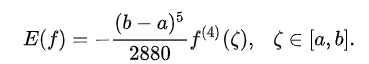
**Метод Сімпсона**: один із методів чисельного інтегрування. Названий на честь британського математика Томаса Сімпсона.

Формулою Сімпсона називається інтеграл від інтерполяційного многочлена другого степеня на відрізку [a,b]:

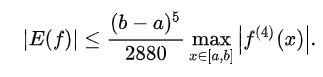


Похибка

При умові, що функція f(x) на відрізку [a,b] має похідну четвертого порядку, похибка E(f), дорівнює:



Зважаючи, що значення zeta переважно не є відомим, для оцінки похибки використовується нерівність:



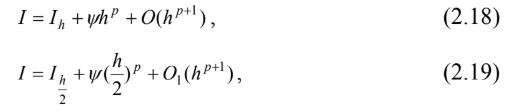
**Процедура Рунге** дозволяє оцінити похибку і підвищити на одиницю

порядок методу шляхом багаторазового (у найпростішому разі дворазового)

прорахунку з різними кроками.

Нехай використовується будь-який метод чисельного інтегрування із

кроками h і h/2. І нехай порядок обраного методу дорівнює p, тоді



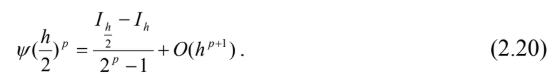
де I - точне значення інтеграла, I(h), I(h/2) - обчислені значення інтеграла із

кроком h і h/2 відповідно; другі доданки праворуч – головні члени похибки

методу чисельного інтегрування порядку p. Віднявши від виразу (2.19) вираз

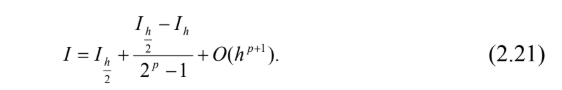
(2.18), отримаємо





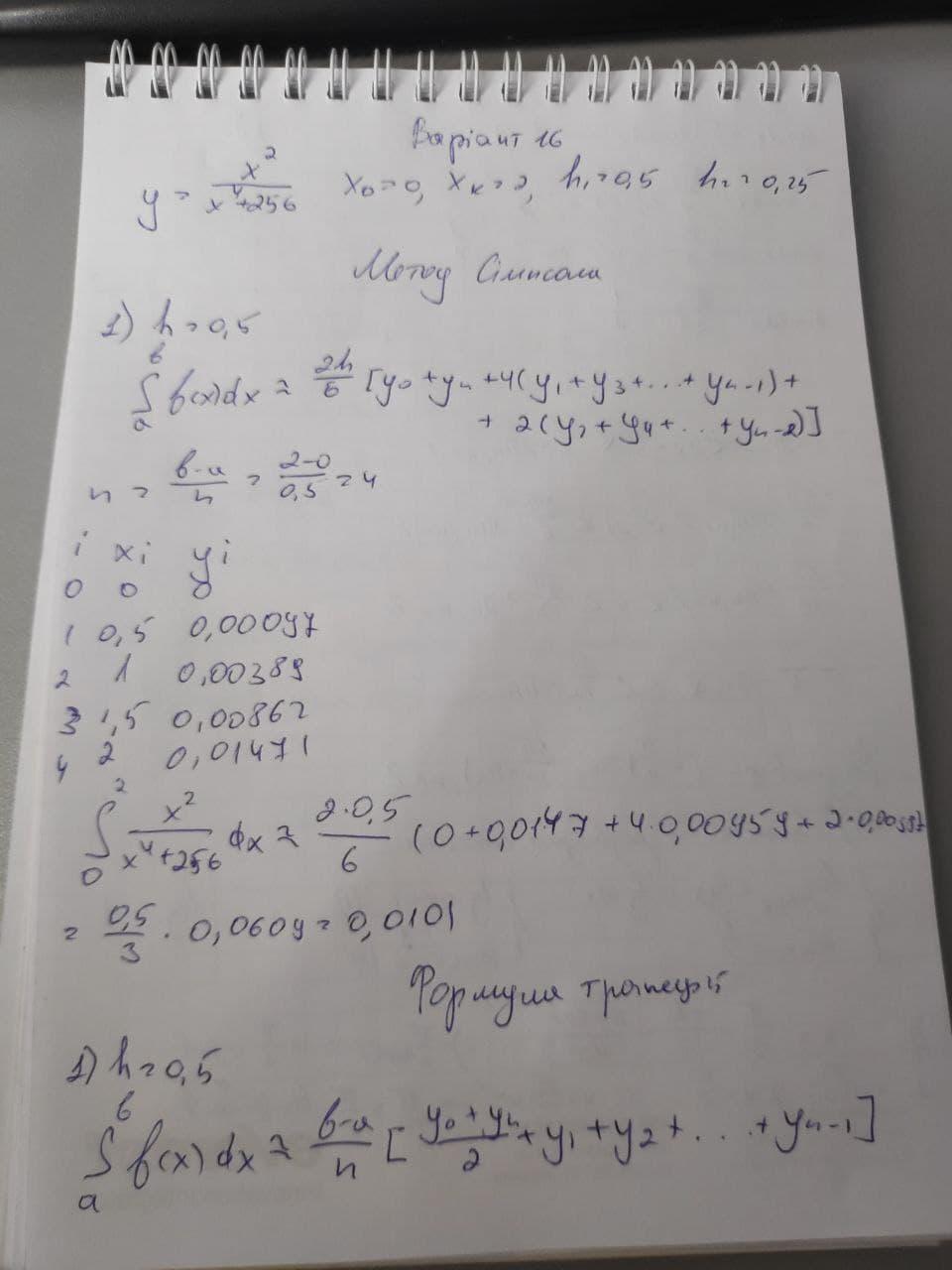
Вираз (2.20) дозволяє виконати апостеріорну оцінку похибки обчисленого значення визначеного інтеграла.

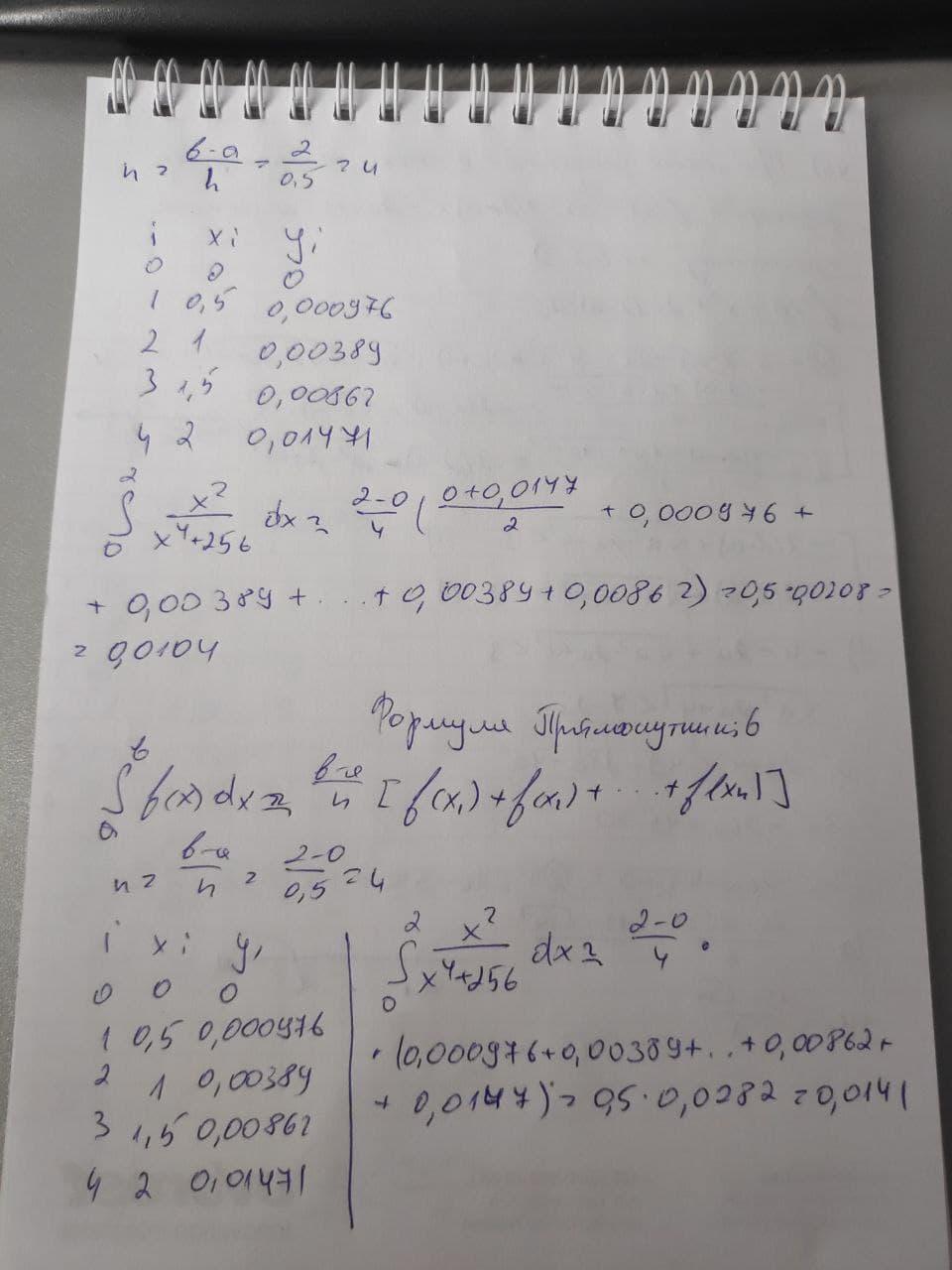
Підставимо (2.20) в (2.18), отримаємо формулу чисельного інтегрування

вже порядку p+1

Таким чином, формула (2.21) – найпростіша процедура Рунге уточнення на один порядок формули чисельного інтегрування.

**РОЗВ’ЯЗОК ЗАДАЧІ В АНАЛІТИЧНІЙ ФОРМІ**

****

****

**ЛІСТИНГ ПРОГРАМИ РОЗВ’ЯЗУВАННЯ ЗАДАЧІ**

**def rectangular(f, a, b, n):**

**h = float(b - a) / n**

**funcSum = f(a + 0.5 \* h)**

**for i in range(1, n):**

**funcSum += f(a + (0.5 + i) \* h)**

**result = h \* funcSum**

**return result**

**def simpson(f, a, b, n):**

**h = float(b - a) / n**

**firstItem = 0**

**secondItem = 0**

**n = round((b - a) / (2 \* h))**

**for i in range(1, n + 1):**

**firstItem += f(a + (2 \* i - 1) \* h)**

**for i in range(1, n):**

**secondItem += f(a + (2 \* i) \* h)**

**result = (h / 3) \* (f(a) + (4 \* firstItem) + (2 \* secondItem) + f(b))**

**return result**

**def trapezoidal(f, a, b, n):**

**h = float(b - a)/n**

**func = 0**

**for i in range(1, n):**

**func += f(a + i\*h)**

**result = h\*0.5\*(f(a) + f(b) + 2\*func)**

**return result**

**def rungeRomberg(f, mF, a , b, n):**

**h = mF(f, a, b, n)**

**hHalf = mF(f, a, b, round(n / 2))**

**result = hHalf - ((hHalf - h) / 3)**

**print('{0:10} {1}\n'.format('Runge-Romberg method result:', result))**

**def f(x):**

**return (x\*\*2)/(x\*\*4 + 256)**

**a = 0**

**b = 2**

**n1 = 4**

**n2 = 8**

**print("a:", a, "\tb:", b, "\tn:", n1)**

**print("------------------------------------")**

**print('{0:28} {1}'.format('Rectangular method result:', rectangular(f, a, b, n1)))**

**rungeRomberg(f, rectangular, a, b, n1)**

**print('{0:28} {1}'.format('Simpson method result:', simpson(f, a, b, n1)))**

**rungeRomberg(f, simpson, a, b, n1)**

**print('{0:28} {1}'.format('Trapezoidal method result:', trapezoidal(f, a, b, n1)))**

**rungeRomberg(f, trapezoidal, a, b, n1)**

**print("a:", a, "\tb:", b, "\tn:", n2)**

**print("------------------------------------")**

**print('{0:28} {1}'.format('Rectangular method result:', rectangular(f, a, b, n2)))**

**rungeRomberg(f, rectangular, a, b, n2)**

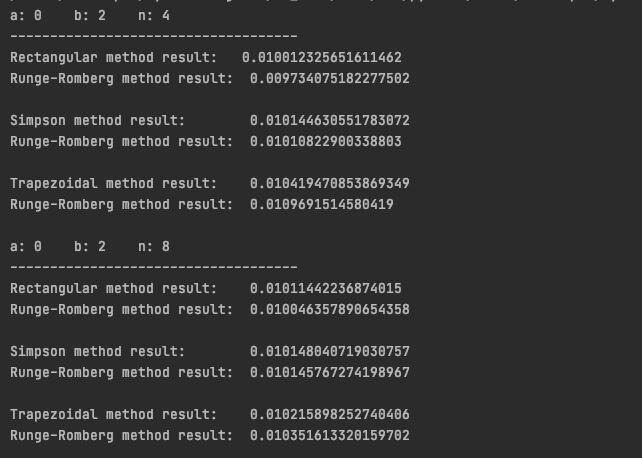
**print('{0:28} {1}'.format('Simpson method result:', simpson(f, a, b, n2)))**

**rungeRomberg(f, simpson, a, b, n2)**

**print('{0:28} {1}'.format('Trapezoidal method result:', trapezoidal(f, a, b, n2)))**

**rungeRomberg(f, trapezoidal, a, b, n2)**

**РЕЗУЛЬТАТИ ВИКОНАННЯ ПРОГРАМИ**

****

**Висновки :**

В ході лабораторної роботи ми навчилися чисельно інтегрувати функції за допомогою метода прямокутників, формули трапецій, формули Сімпсона та використовувати процедуру Рунге для оцінки похибки й уточнення формул чисельного інтегрування та реалізовувати згадані методи програмно. Для перевірки вірності методів ми вводили рівняння для прикладу та дані, запропоновані в задачі.