Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення

ЗВІТ

Про виконання лабораторної роботи № 10

«Чисельні методи інтегрування»

з дисципліни «Чисельні методи»

Лектор:

Мельник Н.Б.

Виконав:

студент групи ПЗ-11

Солтисюк Д.А.

Прийняла:

Мельник Н.Б.

«___» ______ 2022 р.

∑ = ____ ___________

Львів – 2022

Тема роботи: Чисельні методи інтегрування.
Мета роботи: Ознайомлення на практиці з методами чисельного інтегрування.

Теоретичні відомості

Метод прямокутників

Найпростішим методом наближеного обчислення інтеграла є метод прямокутників, суть якого зводиться до знаходження означеного інтеграла як суми площ n прямокутників висотою $f(x_i)$ та основою $h = x_i$, отриманих шляхом розбиття відрізка інтегрування [a,b] на n рівних частин.

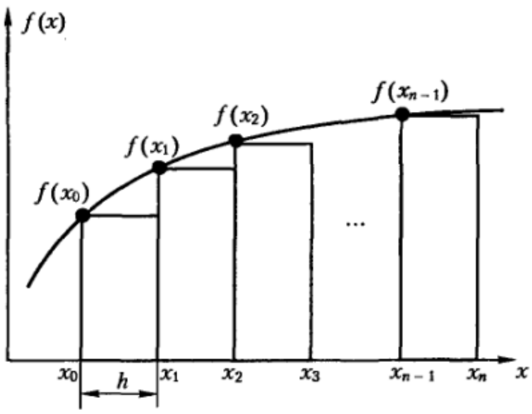

Рис.1 Метод лівих прямокутників

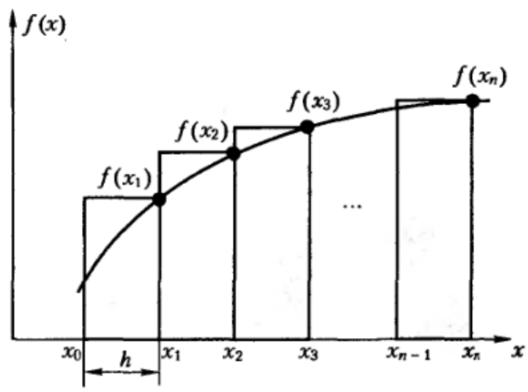

Рис.2 Метод правих прямокутників

Формула лівих прямокутників:

$$I_л = \int_a^b f(x)dx \approx h \sum_{i=0}^{n-1} f(x_i)$$

Формула правих прямокутників:

$$I_{пр} = \int_a^b f(x)dx \approx h \sum_{i=1}^{n} f(x_i)$$

Де $h = \dfrac{b - a}{n}$

Однак ці методи не є дуже точними, оскільки в першому є нестача, а в другому надлишок інтегралу. Тому існує метод середніх прямокутників.

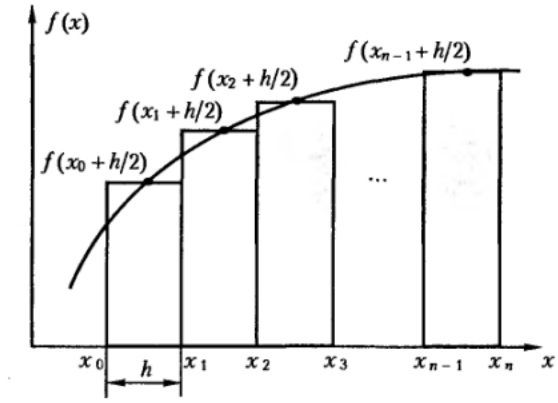

Рис.3 Метод середніх прямокутників.

Формула середніх прямокутників:

$$I_{сер} = \int_a^b f(x)dx \approx h\sum_{i=0}^{n-1} f(x_i + \frac{h}{2})$$

Метод трапецій

Метод трапецій полягає в тому, що відрізок інтегрування [a,b] розбивають на n рівних відрізків, а криву, описану підінтегральну функцією f (x), замінюють на кожному із цих відрізків кусково-лінійною функцією φ(x), отриманою стягуванням хорд, які проходять через точки $(x_{i-1}, f(x_{i-1}))$ та $(x_i, f(x_i))$ $(i = 1, n)$.

Значення інтеграла знаходять як суму площ S_i $(i = 0, n)$ прямокутних трапецій з висотою $h = \frac{b - a}{n}$.

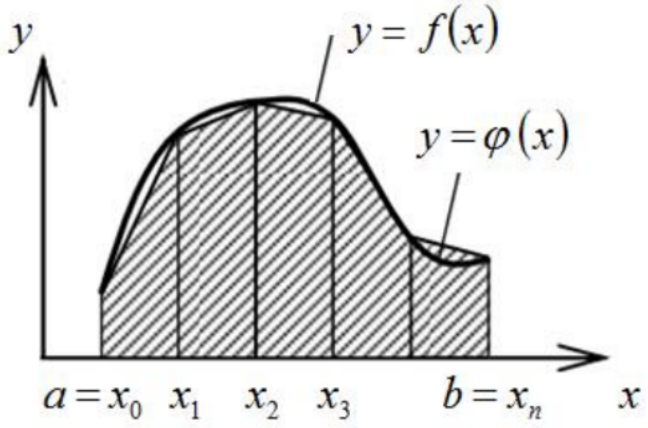

Рис.4 метод трапецій

Площу кожної і-ої елементарної трапеції визначають за формулою

$$S_i = h\frac{f(x_i) + f(x_{i+1})}{2}$$

Відповідно отримуємо таку формулу трапецій для обчислення інтегралу:

$$I_{тр} = \int_a^b f(x)dx \approx h(\frac{f(x_0) + f(x_n)}{2} + \sum_{i=1}^{n-1} f(x_i))$$

Метод Сімпсона

Даний метод полягає в тому, що криву, описану підінтегральною функцією f (x) , на елементарних відрізках заміняють параболою.

Поділимо відрізок інтегрування [a,b] на парну кількість n рівних частин з кроком

$h = \frac{b - a}{n}$. На кожному елементарному відрізку $[x_0,\ x_2],\ [x_2,\ x_4],\ \ldots\ ,\ [x_{i-1},\ x_{i+1}], \ldots, [x_{n-2},\ x_n]$ підінтегральну функцію f (x) замінимо інтерполяційним поліномом другого степеня (квадратичною параболою). Тоді обчислення означеного інтеграла зводиться до обчислення суми площ S_i , (i =1,n) криволінійних трапецій.

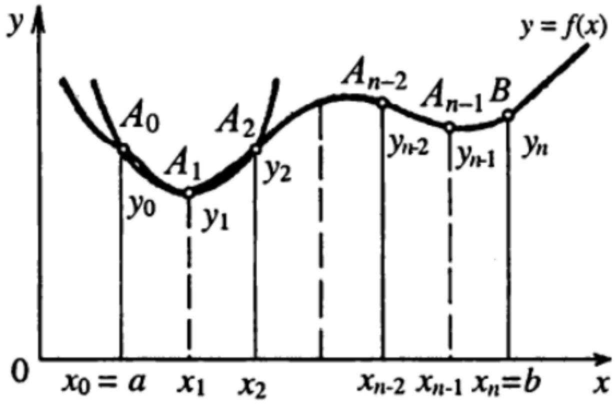

Рис.5 Метод Сімпсона

Площу S_i кожної елементарної криволінійної трапеції визначають за формулою Сімпсона

$$S_i = \frac{h}{3}(f(x_i) + 4f(x_{i+1}) + f(x_{i+2}))$$

Формула Сімпсона:

$$\int_a^b f(x)dx \approx \frac{h}{3}(f(x_0) + 4\sum_{i=1}^{n} f(x_{2i-1}) + 2\sum_{i=1}^{n-1} f(x_{2i}) + f(x_{2n}))$$

Індивідуальне завдання

Скласти програму чисельного інтегрування у відповідності до варіанту:

1) Методом лівих, правих та центральних прямокутників;
2) Методом трапецій;
3) Методом Сімпсона

Варіант 4

4. $\int\limits_0^2 \frac{sh\,x}{1-2x+3x^2}\,dx;$

Результат виконання роботи

rectangles.py

```
from typing import Callable, Dict, Literal, Tuple

import numpy as np
from common.main import IntegrationOrientedMethod

RectangleMethodType = Literal["right", "left", "centered", "trapezium"]


class RectangleMethod(IntegrationOrientedMethod):

    method_mappings: Dict[RectangleMethodType, Tuple[Callable, list[int]]]
    method_type: RectangleMethodType

    def __init__(self, method_type: RectangleMethodType, f: Callable, a: float,
                 b: float, eps: float) -> None:
        super().__init__(f"{method_type.capitalize()} rectangles", f, a, b,
                         eps)

        self.method_type = method_type
        self.method_mappings = {
            "left": (lambda i: self.f_xi(i), list(range(self.n - 1))),
            "right": (lambda i: self.f_xi(i), list(range(1, self.n))),
            "centered":
            (lambda i: self.f_xi(i + 0.5), list(range(self.n - 1))),
            "trapezium": (lambda i: (self.f_xi(i) + self.f_xi(i + 1)) / 2,
                          list(range(self.n - 1)))
        }

    def execute_method(self):
        return self.h * np.sum([
            self.method_mappings[self.method_type][0](i)
            for i in self.method_mappings[self.method_type][1]
        ])
```

simpson.py

```
from typing import Callable

import numpy as np
from common.main import IntegrationOrientedMethod


class SimpsonMethod(IntegrationOrientedMethod):

    def __init__(self, f: Callable, a: float, b: float, eps: float) -> None:
        super().__init__("Simpson", f, a, b, eps)

    def execute_method(self):
        return (self.h / 3) * (
            self.f_xi(0) + self.f_xi(self.n) +
            4 * np.sum([self.f_xi(i - 1) for i in range(1, self.n) if i % 2]) +
            2 * np.sum([self.f_xi(i) for i in range(1, self.n - 1) if i % 2]))
```

main.py

```python
from math import sinh
from typing import List

from common.rectangles import RectangleMethod, RectangleMethodType
from common.simpson import SimpsonMethod

if __name__ == "__main__":

    def f(x):
        return sinh(x) / (1 - 2 * x + pow(3 * x, 2))

    a = 0
    b = 2
    eps = 0.01

    rectangle_methods: List[RectangleMethodType] = [
        "left", "right", "centered", "trapezium"
    ]

    for rectangle_method in rectangle_methods:
        RectangleMethod(rectangle_method, f, a, b, eps).compile()

    SimpsonMethod(f, a, b, eps).compile()
```

Результат виконання програми

```
Initial values
Bounds [0, 2]


Results
0.2898697587574605


>>> Right rectangles method

Initial values
Bounds [0, 2]


Results
0.3008773805544572


>>> Centered rectangles method

Initial values
Bounds [0, 2]


Results
0.2966362949642391


>>> Trapezium rectangles method

Initial values
Bounds [0, 2]


Results
0.29537356965595885


>>> Simpson method

Initial values
Bounds [0, 2]


Results
0.29452238021488303
```

Висновки

Виконуючи лабораторну роботу №10, я ознайомився на практиці з методами чисельного інтегрування.