**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

Інститут **ІКНІ**

Кафедра **ПЗ**



### ЗВІТ

До лабораторної роботи №10

**На тему:** *“Чисельні методи інтегрування”*

**З дисципліни:** *“Чисельні методи”*

**Лектор:**

доцент каф. ПЗ

Мельник Н.Б.

**Виконав:**

ст. гр. ПЗ-16

Шеремета А.І.

**Прийняла:**

асистент кафедри ПЗ

Бутрак І. О.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2022 р.

∑= \_\_\_\_

Львів – 2022

**Тема роботи:** Чисельні методи інтегрування.

**Мета роботи:** ознайомлення на практиці з методами чисельного інтегрування.

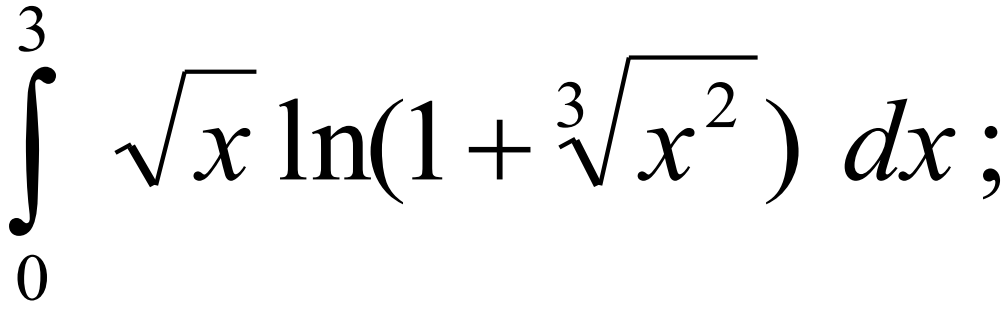
**Індивідуальне завдання**

Скласти програму чисельного інтегрування у відповідності до варіанту:

1) методом лівих, правих та середніх прямокутників;

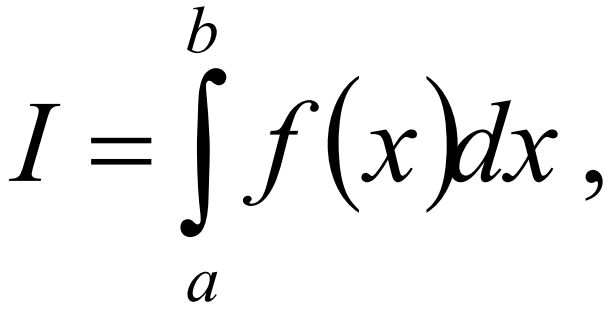
2) методом трапецій;

3) методом Сімпсона.

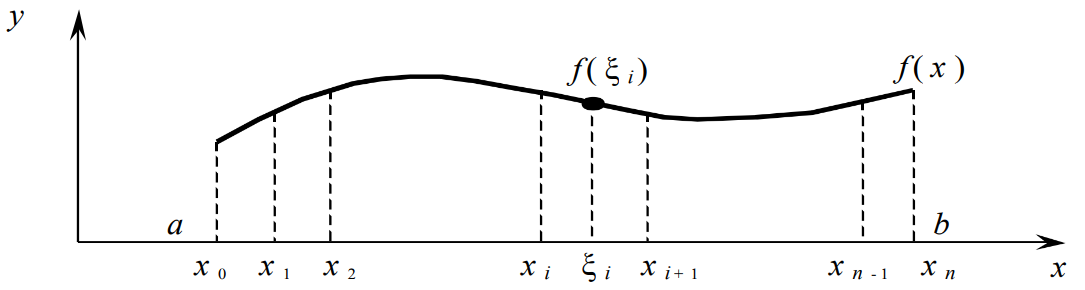


**Теоретичні відомості**

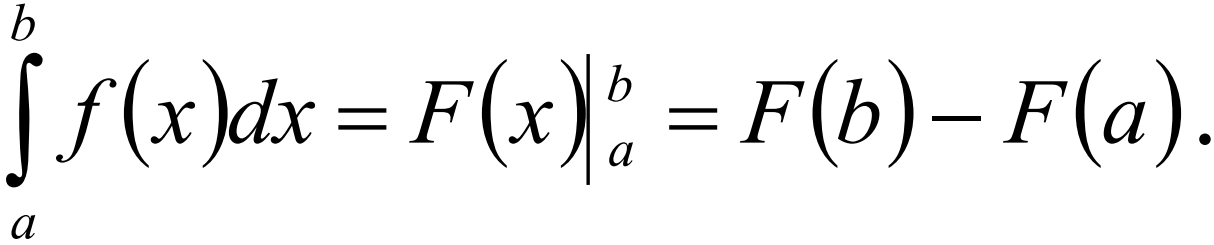
Багато наукових, технічних і практичних задач зводяться до інтегрування функцій. Зокрема, обчислення площ поверхонь, об’ємів тіл, моментів інерції і т.п. Нагадаємо, що геометричний зміст найпростішого означеного інтеграла



від додатньо визначеної неперервної функції *f* (*x*) ≥ 0 полягає у тому, що числове значення величини *I* – це площа, обмежена кривою *y* = *f* (*x*) , віссю абсцис та прямими *x* = *a* , *x* = *b* .



У випадках, коли підінтегральну функцію задано аналітично, причому вона є інтегровною, означений інтеграл обчислюють безпосередньо за допомогою формули Ньютона-Лейбніца. Ця формула полягає в тому, що означений інтеграл дорівнює приросту первісної *F*(*x*) на відрізку інтегрування



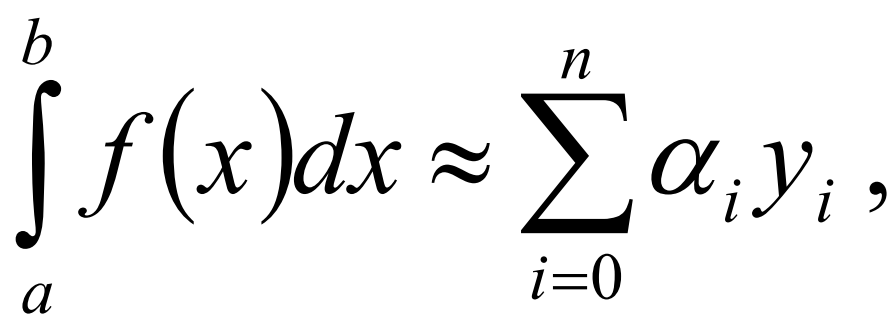
Однак на практиці цією формулою не завжди можна скористатися через дві основні причини:

1) функція *f* (*x*) не є інтегровною, тобто її первісну F(x) не можна зобразити елементарними функціями;

2) значення функції *f* (*x*) є відомим тільки на множині скінченної кількості точок

( *i* = 0, *n* ), тобто функцію задано у вигляді таблиці.

У цьому випадку застосовують методи чисельного інтегрування, які ґрунтуються на інтерполюванні підінтегральної функції за допомогою інтерполяційних поліномів. Така інтерполяція дає змогу наближено замінити означений інтеграл скінченною сумою

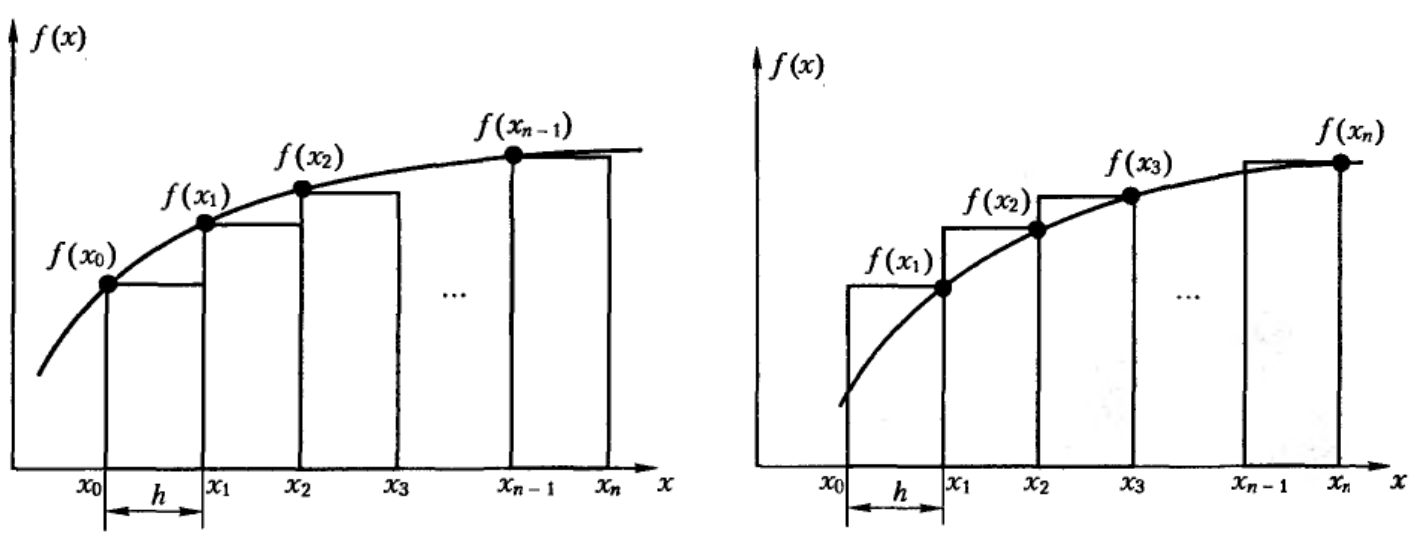


де – значення підінтегральної функції у вузлах інтерполяції; – числові коефіцієнти.

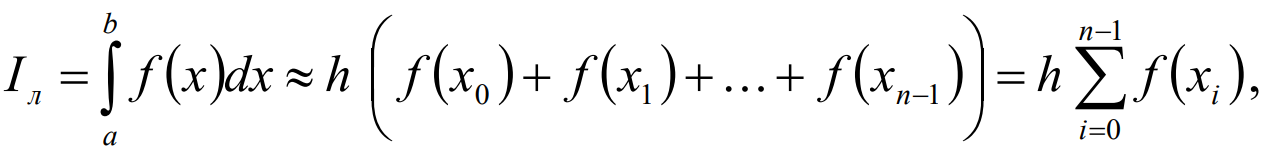
Співвідношення називають квадратурною формулою, а його праву частину – квадратурною сумою. У залежності від способу її обчислення існують різні методи чисельного інтегрування – метод прямокутників, трапецій, парабол (Сімпсона).

**Метод прямокутників**

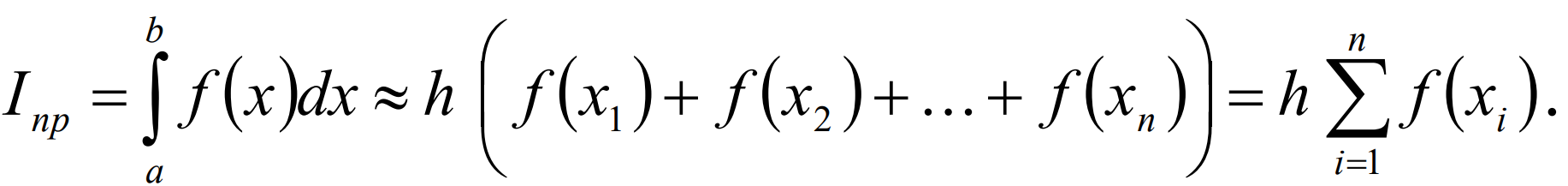
Найпростішим методом наближеного обчислення інтеграла є метод прямокутників, суть якого зводиться до знаходження означеного інтеграла як суми площ n прямокутників висотою та основою , отриманих шляхом розбиття відрізка інтегрування [*a,b*] на n рівних частин. Розбиття на прямокутники виконують зліва направо або справа наліво. При цьому висотою кожного елементарного прямокутника буде значення функції *y* = *f (x)* у крайній лівій або крайній правій точці відповідно.



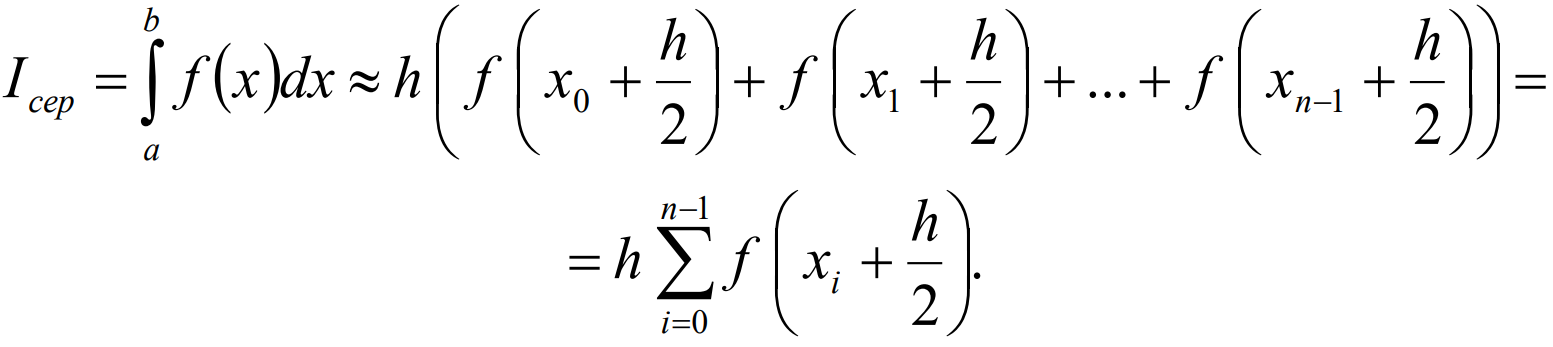
Для першого випадку отримуємо формулу лівих прямокутників



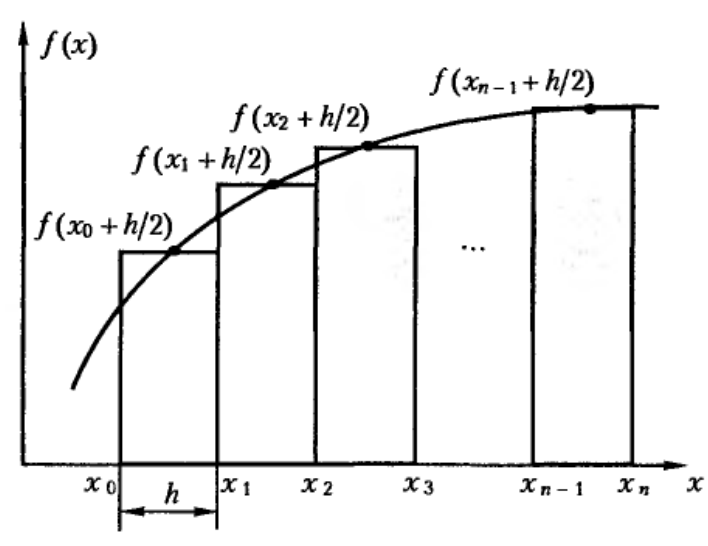
а для другого - формулу правих прямокутників



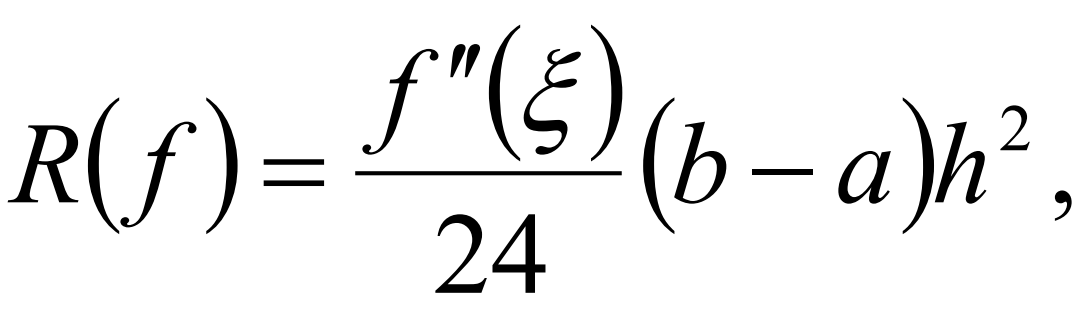
Тут крок інтегрування . Якщо функція *f* (*x*) монотонно зростає на відрізку [*a,b*] , то із використанням формул лівих і правих прямокутників отримують наближене значення інтеграла з недостачею та з надлишком відповідно. На практиці застосовують точнішу розрахункову формулу середніх (центральних) прямокутників, у результаті чого отримують точніше значення інтеграла



У цій формулі враховано значення функції в середніх точках, (*i* =1,*n*) елементарних відрізків.



Похибку обчислення інтеграла за методом прямокутників визначають за формулою

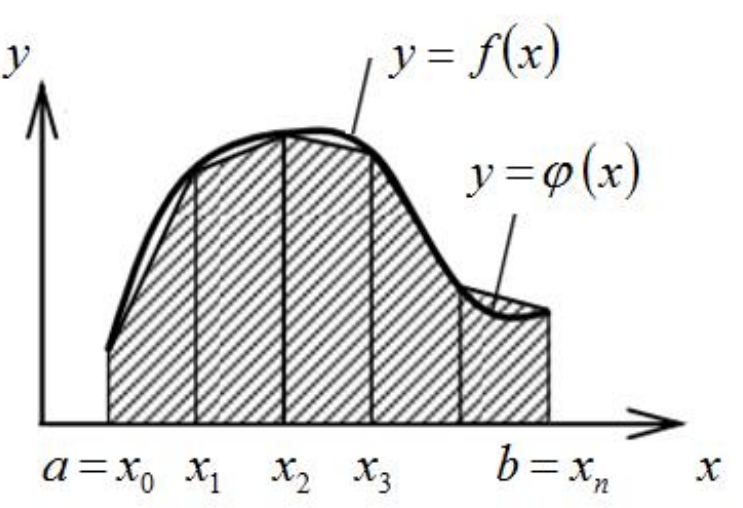


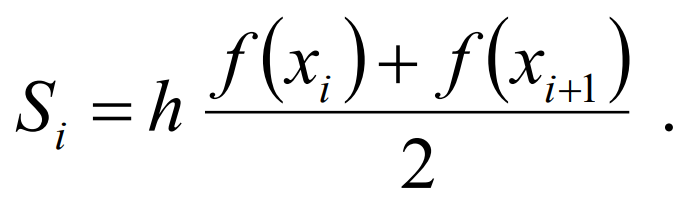
Метод трапецій

Метод трапецій полягає в тому, що відрізок інтегрування [*а,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 та (*i* =1,*n*)

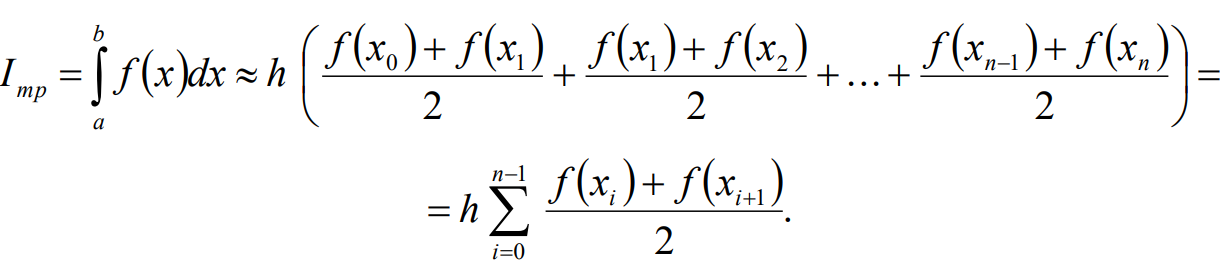
Значення інтеграла знаходять як суму площ (*i* = 0,*n*) прямокутних трапецій з висотою

.

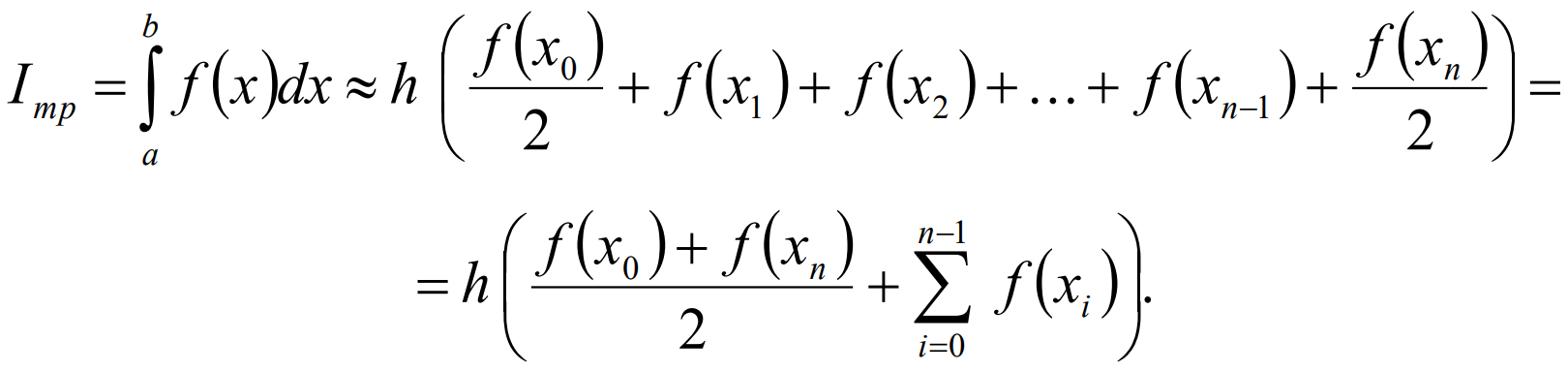


Площу кожної *i* -ої елементарної трапеції визначають за формулою

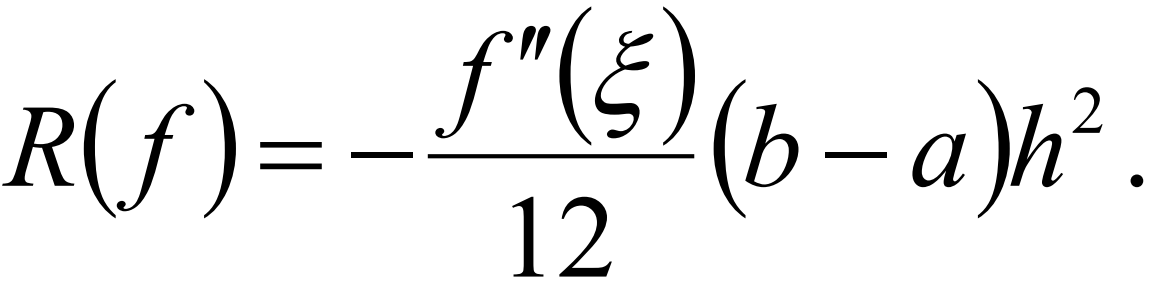
Відповідно на всьому відрізку інтегрування [*a,b*] площу складеної фігури визначають сумою площ усіх елементарних трапецій. У результаті отримують таку формулу



Оскільки в наведеній формулі під знаком суми величини *f* () , () зустрічаються двічі, то перепишемо її у вигляді



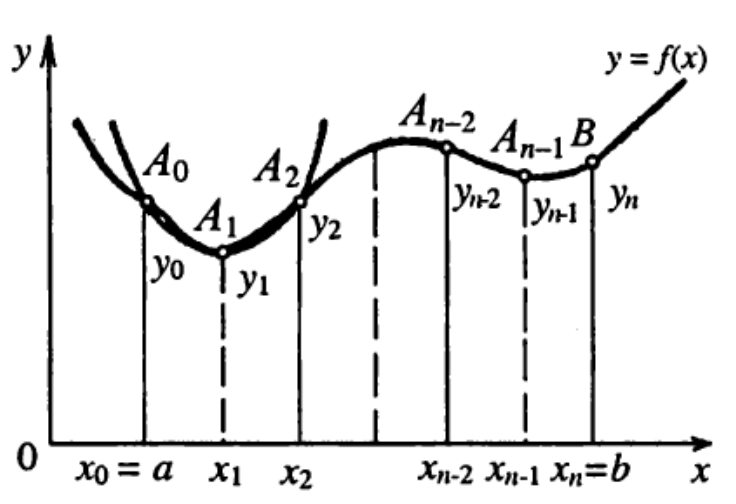
Похибку обчислення інтеграла з використанням формули трапецій визначають за формулою



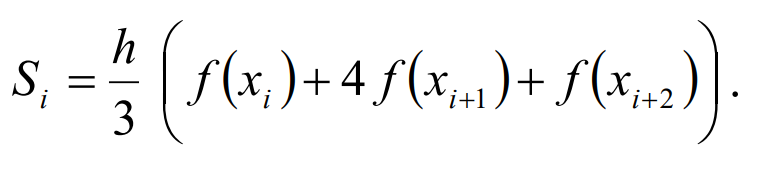
Метод Сімпсона

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

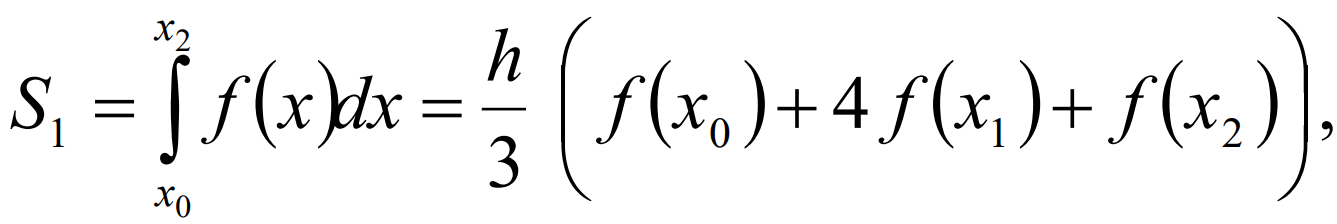
Поділимо відрізок інтегрування [*a,b*] на парну кількість n рівних частин з кроком . На кожному елементарному відрізку підінтегральну функцію *f* (*x*) замінимо інтерполяційним поліномом другого степеня (квадратичною параболою). Тоді обчислення означеного інтеграла зводиться до обчислення суми площ , (*i* =1,*n*) криволінійних трапецій

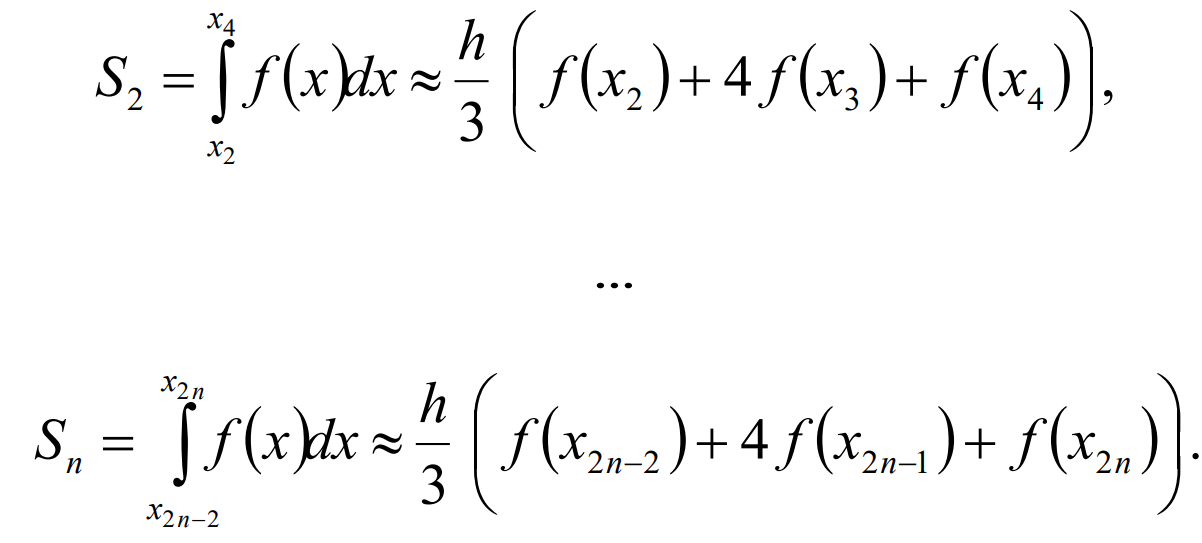


Площу кожної елементарної криволінійної трапеції визначають за формулою Сімпсона

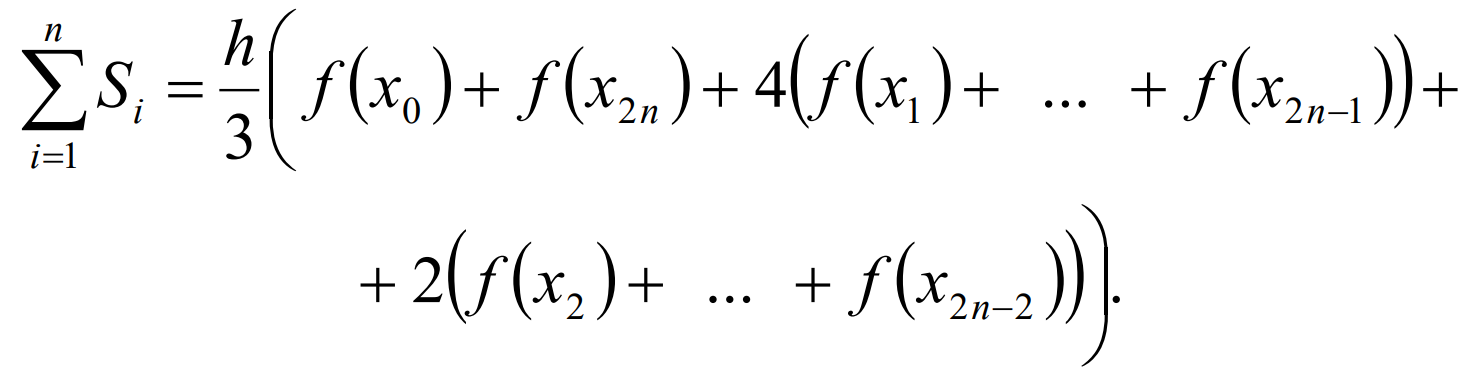


Послідовно обчислюємо за формулою площі *n* криволінійних трапецій , (*i* =1,*n*)

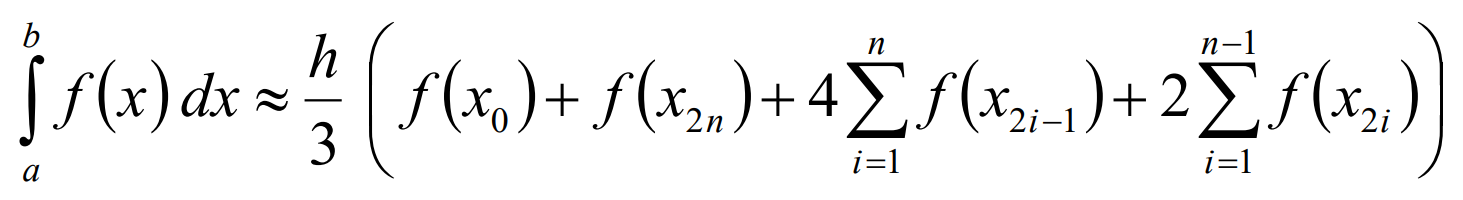




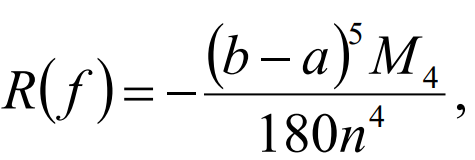
Знайдемо суму площ всіх криволінійних трапецій.

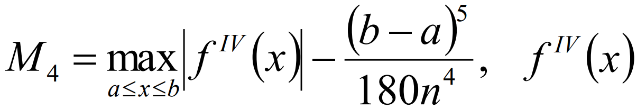


Тоді розрахункова формула методу Сімпсона набуде такого вигляду



Для визначення похибки обчислення інтеграла за формулою (11.14) використовують нерівність:





де - похідна четвертого порядку функції

*y* = *f* (*x*).

**Код програми**

#include <iostream>

#include <cmath>

double f(double x){

return sqrt(x)\* log(1+pow(x, (2/3)));

}

double f1(double x){

return (-9\*log(pow(x , (2/3))+1) - 8\*(2/(pow(x, 2/3)+1) + 1/(pow(x, (2/3)))/pow(x, (1/6)\*(pow(x, (2/3))+1))+(24/(pow(x, (2/3))+1)))/36);

}

double f4(double x){

return (-1/(2\*pow(x,(11/6))+2\*pow(x, (5/2))))

+(1/(3\*sqrt(x)\*(pow((pow(x, (2/3))+1), 3))))

+(3\*log(pow(x, (2/3))+1)/(8 \* pow(x, (5/2))))

-(8/(27\*pow(x, (7/6))\*pow((pow(x, (2/3)) + 1),3)))

-(1/(27\*pow(x, (11/6))\*pow((pow(x, (2/3)) + 1),3)));

}

void leftRectangleMethod(double a, double b, int steps){

double area = 0;

double h = (a+b)/steps;

for ( int i = 0; i < steps; ++i){

area += h \* (f(a + i \* h));

}

double MaxF = f1(a + h);

for( double x = a + 2 \* h; x < b; x+=h)

if(f1(x)> MaxF)

MaxF = f1(x);

double error = (MaxF / 24)\* (b - a) \* h \* h;

std::cout << "Left Rectangle method:\nResult: " << area << "\nError: " << error << '\n';

}

void rightRectangleMethod(double a, double b, double steps){

double area = 0;

double h = (a+b)/steps;

for( int i = 0; i < steps; i++){

area+= h \* f (a + i\*h);

}

double MaxF = f1(a+h);

for(double x = a + 2 \* h; x < b; x+=h)

if(f1(x) > MaxF)

MaxF = f1(x);

double error = -1 \* ((MaxF / 24) \* (b - a) \* h \* h);

std::cout << "Right Rectangle method:\nResult: " << area << "\nError: " << error << '\n';

}

void centerRectangleMethod(double a, double b, int steps){

double area = 0;

double h = (a+b)/steps;

for(int i = 0; i < steps; ++i)

area+= h \* (f((a + i \* h) + h / 2));

double MaxF = f4(a+h);

for(double x = a + 2 \* h; x < b;x += h)

if(f4(x) > MaxF)

MaxF = f4(x);

double error = (MaxF / 24) \* (b-a) \* h \*h;

std::cout << "Center Rectangle method:\nResult: " << area << "\nError: " << error << '\n';

}

void trapezoidalMethod(double a, double b, int steps){

double area = 0;

double h = (a + b) / steps;

area = h \* (f(a) + f(b)) / 2;

for(int i = 1; i < steps; i++)

area += h \* f(a + i \* h);

double MaxF = f1(a + h);

for(double x = a + 2 \* h; x < b; x+=h)

if(f1(x) > MaxF)

MaxF = f1(x);

double error = -1 \* (MaxF / 12) \* (b-a) \* h \* h;

std::cout << "Trapezpidal method:\nResult: " << area << "\nError :" << error << '\n';

}

void simpsonMethod(double a, double b, int steps){

double area = 0;

double h = (a + b) / steps;

area = f(a) + f(b);

for(int i = 1; i <= steps/2; i++)

area += 4 \* f(a + (2 \* static\_cast<double>(i)-1) \* h);

for(int i = 1; i <= steps / 2; i++)

area += 2 \* f(a + (2 \* static\_cast<double>(i)) \* h);

area \*= h/3;

double MaxF = f4(a+h);

for(double x = a + 2 \* h; x < b; x += h)

if(f4(x) > MaxF)

MaxF = f4(x);

double error = -1 \* (MaxF / 180) \* (b - a) \* h \* h \*h \* h;

std::cout << "Simpson method\nResult: " << area << "\nError: " << error << '\n';

}

int main(){

leftRectangleMethod(0, 3, 10);

rightRectangleMethod(0, 3, 10);

centerRectangleMethod(0, 3, 10);

trapezoidalMethod(0, 3, 10);

simpsonMethod(0, 3, 10);

return 0;

}

**Протокол роботи**

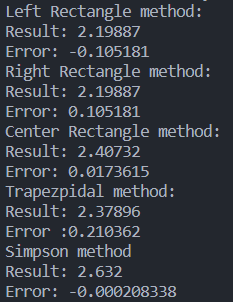


Рис.1. Результат виконання програми

**Висновки**

На даній лабораторній роботі я ознайомився на приктиці з методами чисельного інтегрування та склав програму чисельного інтегрування різними методами.