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

На тему:

«Чисельні методи інтегрування»
з дисципліни «Чисельні методи»

Лектор:
доцент каф. ПЗ
Мельник Н. Б.

Виконав:
ст. гр. ПЗ-11
Морозов О. Р.

Прийняла:
доцент каф. ПЗ
Мельник Н. Б.

« __ » ________ 2022 р.

∑ = _____ _________ .

Львів – 2022

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

Теоретичні відомості

Метод прямокутників
Найпростішим методом наближеного обчислення інтеграла є метод прямокутників, суть якого зводиться до знаходження визначеного інтеграла як суми площ n прямокутників висотою $f(x)$ та основою $h = \Delta x_i = x_{i+1} - x_i$, отриманих шляхом розбиття відрізка інтегрування [a, b] на n рівних частин.
Розбиття на прямокутники виконують зліва направо або справа наліво.

Формула лівих прямокутників:
$$I_l = \int_a^b f(x)dx \approx h\left(f(x_0) + f(x_1) + \dots + f(x_{n-1})\right) = h\sum_{i=0}^{n-1} f(x_i);$$
Формула правих прямокутників:
$$I_r = \int_a^b f(x)dx \approx h\left(f(x_1) + f(x_2) + \dots + f(x_n)\right) = h\sum_{i=0}^{n} f(x_i);$$
Формула середніх прямокутників:
$$I_l = \int_a^b f(x)dx \approx h\left(f(x_0 + \tfrac{h}{2}) + f(x_1 + \tfrac{h}{2}) + \dots + f(x_{n-1} + \tfrac{h}{2})\right) = h\sum_{i=0}^{n-1} f(x_i + \tfrac{h}{2});$$

Метод трапецій
Метод трапецій полягає в тому, що відрізок інтегрування [a, b] розбивають на n рівних відрізків, а криву, описану під інтегральною функцією $f(x_i)$, замінюють на кожному з цих відрізків кусково-лінійною функцією $\varphi(x)$, отриманою стягуванням хорд, що проходять через точки $(x_{i-1}, f(x_{i-1}))$ та $(x_i, f(x_i))$.
Значення інтеграла знаходять як суму площ прямокутних трапецій з висотою $h = \frac{b-a}{n}$.
Площу кожної трапеції обчислюють за формулою: $S_i = h\dfrac{f(x_i)+f(x_{i+1})}{2}$

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

$$I_{mp} = \int_a^b f(x)dx \approx h\left(\frac{f(x_0)+f(x_1)}{2} + \frac{f(x_1)+f(x_2)}{2} + \dots + \frac{f(x_{n-1})+f(x_n)}{2}\right) =$$
$$= h\sum_{i=0}^{n-1}\frac{f(x_i)+f(x_{i+1})}{2}.$$

Також можемо переписати її як:

$$I_{mp} = \int_a^b f(x)dx \approx h\left(\frac{f(x_0)}{2} + f(x_1) + f(x_2) + \ldots + f(x_{n-1}) + \frac{f(x_n)}{2}\right) =$$

$$= h\left(\frac{f(x_0) + f(x_n)}{2} + \sum_{i=0}^{n-1} f(x_i)\right).$$

Метод Сімпсона

Даний метод полягає в тому, що криву, описану під інтегральною функцією f(x), на елементарних відрізках замінюють параболою.

Поділимо відрізок інтегрування [a, b] на парну кількість рівних частин з кроком: $h = \frac{b-a}{n}$. На кожному елементарному відрізку [x_0, x_2], [x_2, x_4], ... , [x_{i-1}, x_{i+1}], ... , [x_{n-2}, x_n] підінтегральну функцію f(x) замінимо інтерполяційним поліномом другого ступеня(квадратичною параболою). Тоді обчислення означеного інтеграла зводиться до обчислення суми площ криволінійних трапецій.

Площу S_i кожної трапеції визначають за формулою Сімпсона:

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

Загальна розрахункова формула в такому випадку виглядає так:

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

Індивідуальне завдання

Хід роботи

Код програми:

```
#include <iostream>
#include <iomanip>
#include <cmath>
#include <vector>

using namespace std;

double f(double x) {
	return (1 + (2 * x)) / pow(log(2 + pow(x, 2)), 2);
}

double LeftRectangles(double a, double b, int n) {
	double res = 0;
	double h = (b - a) / n;
	for(int i = 0; i < n; i++) {
```

```
            res += f(a + (i * h));
        }
        return res * h;
}

double RightRectangles(double a, double b, int n) {
        double res = 0;
        double h = (b - a) / n;
        for(int i = 1; i <= n; i++) {
            res += f(a + (i * h));
        }
        return res * h;
}

double MiddleRectangles(double a, double b, int n) {
        double res = 0;
        double h = (b - a) / n;
        for(int i = 0; i < n; i++) {
            res += f(a + (i * h) + (h/2));
        }
        return res * h;
}

double TrapezeMethod(double a, double b, int n) {
        double res = (f(a) + f(b)) / 2;
        double h = (b - a) / n;
        for(int i = 0; i < n; i++) {
            res += f(a + (i * h));
        }

        return res * h;
}

double GomerSimpsonMethod(double a, double b, int n) {
        double h = (b - a) / n;
        double res = 0, tmp, x = a, odd = 0, pair = 0;

        res += f(x);
        x += h;

        for(int i = 1; i <= n; ++i) {
            if(i == n) {
                res += f(x);
            } else {
                tmp = f(x);
                if((i % 2) == 0) {
                    pair += tmp;
```

```
            } else {
                  odd += tmp;
            }
      }
      x += h;
    }
    res += 4 * odd + 2 * pair;
    return res * h / 3;
}

int main() {

    cout << "[ Numerical Methods Of Integration ]" << endl << endl
           << "  / 4      1 + 2x" << endl
           << "  |   ----------------- dx;" << endl
           << "0 /    ln^2(2 + (x^2))" << endl;

    double a = 0, b = 4, e = 0;

    cout << endl << "Enter accuracy:" << endl;
    cin >> e;

    double res_prev, res_next;

    int h_prev = (b - a) / sqrt(e);
    int h_next;


//==========================================================================//

    do {
          res_prev = LeftRectangles(a, b, h_prev);
          h_next = h_prev * 2;
          res_next = LeftRectangles(a, b, h_next);
          h_prev = h_next;
    } while(fabs(res_prev - res_next) > e);

    cout << endl << "=====[ Left Rectangles Method ]=====" << endl
           << "n = " << h_next << endl
           << "Result = " << res_next << endl;


//==========================================================================//

    h_prev = (b - a) / sqrt(e);

    do {
```

```
        res_prev = RightRectangles(a, b, h_prev);
        h_next = h_prev * 2;
        res_next = RightRectangles(a, b, h_next);
        h_prev = h_next;
    } while(fabs(res_prev - res_next) > e);

    cout << endl << "=====[ Right Rectangles Method ]====" << endl
        << "n = " << h_next << endl
        << "Result = " << res_next << endl;


//========================================================================//

    h_prev = (b - a) / sqrt(e);

    do {
        res_prev = MiddleRectangles(a, b, h_prev);
        h_next = h_prev * 2;
        res_next = MiddleRectangles(a, b, h_next);
        h_prev = h_next;
    } while(fabs(res_prev - res_next) > e);

    cout << endl << "====[ Middel Rectangles Method ]====" << endl
        << "n = " << h_next << endl
        << "Result = " << res_next << endl;


//========================================================================//

    h_prev = (b - a) / sqrt(sqrt(e));

    do {
        res_prev = TrapezeMethod(a, b, h_prev);
        h_next = h_prev * 2;
        res_next = TrapezeMethod(a, b, h_next);
        h_prev = h_next;
    } while(fabs(res_prev - res_next) > e);

    cout << endl << "=========[ Trapeze Method ]=========" << endl
        << "n = " << h_next << endl
        << "Result = " << res_next << endl;


//========================================================================//

    h_prev = (b - a) / sqrt(sqrt(e));
```

```
	do {
		res_prev = GomerSimpsonMethod(a, b, h_prev);
		h_next = h_prev * 2;
		res_next = GomerSimpsonMethod(a, b, h_next);
		h_prev = h_next;
	} while(fabs(res_prev - res_next) > e);

	cout << endl << "=========[ Simpson Method ]=========" << endl
		<< "n = " << h_next << endl
		<< "Result = " << res_next << endl;


//======================================================================//

	return 0;
}
```

Результат:

```
[ Numerical Methods Of Integration ]

  / 4      1 + 2x
  |   ----------------- dx;
0 /    ln^2(2 + (x^2))

Enter accuracy:
0.00000001

=====[ Left Rectangles Method ]=====
n = 327680000
Result = 7.22872

=====[ Right Rectangles Method ]====
n = 327680000
Result = 7.22872

====[ Middel Rectangles Method ]====
n = 80000
Result = 7.22872

=========[ Trapeze Method ]=========
n = 838860800
Result = 7.22872

=========[ Simpson Method ]=========
n = 800
Result = 7.22872
```

Висновок

Виконавши дану лабораторну роботу, я ознайомився на практиці з методами чисельного інтегрування, та запрограмував метод трапецій, метод Сімпсона та методи лівих, правих та середніх прямокутників.