**­**

**Лабораторна робота №2.2**

З дисципліни «Програмування»

Тема: НАБЛИЖЕНЕ ОБЧИСЛЕННЯ ІНТЕГРАЛА.

Виконав: Землянський Едуард

Група: КВ-22

**Постановка задачі**

Написати програму для обчислення з точністю  значення .

**Вимоги до програми**

Для обчислення значення визначеного інтегралу  з заданою точністю  використати квадратурну формулу виду , де , а  та  визначаються квадратурною формулою. Квадратурна формула, , ,  визначаються варіантом завдання (див. табл. 3).

Алгоритм квадратурної формули необхідно оформити у вигляді функції *з використанням функції як одного з параметрів.*

1. Для забезпечення необхідної точності при наближеному обчисленні інтеграла за квадратурною формулою необхідно обрати відповідне значення кроку  (чи, що те ж саме, число кроків , на яке ділиться відрізок інтегрування). Для цього можна скористатися формулою залишкового члену квадратурної формули, однак ця оцінка значення відповідної похідної підінтегральної функції, яка входить до формули залишкового члену, нерідко викликає великі труднощі. Крім цього, цю оцінку доводиться робити для кожної нової підінтегральної функції.

На практиці для отримання бажаної точності обчислення за квадратурною формулою інтеграла часто використовується метод послідовного подвоєння кроків, який полягає у наступному.

Інтеграл  обчислюється за квадратурною формулою двічі: спочатку при кількості кроків, рівному , а потім при кількості кроків, рівній . Похибка  наближеного значення інтеграла , обчисленого за квадратурною формулою при кількості кроків, рівній , обчислюється наближено за правилом Рунге: . Для формул прямокутників і трапецій , для формули Сімпсона .

Таким чином,  обчислюється для послідовних значень , ,  і т.д., де  - початкова кількість кроків. Процес обчислення закінчуються, коли для чергового значення  буде отримана похибка . Початкову кількість кроків слід обирати від 10 до 50.

2. Для отримання достатньо ефективної програми необхідно врахувати, що у формулах трапецій і Сімпсона при подвоєнні кількості кроків немає необхідності обчислювати значення підінтегральної функції заново у всіх вузлах сітки, так як всі вузли сітки, отриманні при кількості кроків, рівній , є вузлами сітки й при кількості кроків, рівній .

1. Для відлагодження функції інтегрування слід обирати такі підінтегральні функції та такі межі інтегрування, щоб відповідь була відома заздалегідь та обчислення інтеграла не потребувало великих затрат машинного часу. Необхідно потурбуватися про те, щоб для цієї функції насправді реалізовувався процес послідовного подвоєння числа кроків для отримання результату з заданою точністю.
2. Слід звернути увагу на те, що деякі помилки у програмі при реалізації метода подвоєння числа кроків (наприклад, врахування зайвого ) не впливають на точність результату, але ця точність досягається шляхом використання додаткового машинного часу.

**Варіант завдання**

****

****

формула Сімпсона:

**,**

де **, , , ** - парне;

залишковийчлен**: , .**

**Текст програми**

#include <stdio.h>

#define \_USE\_MATH\_DEFINES

#include <math.h>

double f(double x){

    if (x == 0) x += pow(0.01, 10);

    return log(1/(5 - 3\*cos(x)) + 1/(pow(x, 2)));

}

double sympson(double a, double b, int N){

    double h = (b - a) / N;

    double sum = f(a) + f(b);

    int k;

    for (int i = 1; i <= N - 1; i++){

        k = 2 + 2 \* (i % 2);

        sum += k \* f(a + i \* h);

    }

    sum \*= h / 3;

    return sum;

}

int main(){

    double a = 0;

    double b = 2 \* M\_PI;

    double eps = 0.001;

    int k = 10;

    int i = 0;

    double diff;

    do

    {

        i++;

        diff = fabs(sympson(a, b, k \* i) - sympson(a, b, k \* (i \* 2)));

    } while (diff > eps);

    printf("Integral value = %f", sympson(a, b, k \* (i \* 2)));

    return 1;

}

//      -3.648431486288209

**Тестування програми**

Результат з калькулятора інтегралів:

**-3.648431486288209**

Результати програми:

При **eps = 0.01**



При **eps = 0.001**

****