**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Основы информатики»

Отчет по лабораторной работе №6

« Численное интегрирование функции.»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-13Б |  | преподаватель каф. ИУ5 |
| Лачина Екатерина |  | Аксенова М.В. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2023 г.

1. Задача

Численное интегрирование функции с заданной точностью методом прямоугольников.

Вычислить определённый интеграл в пределах от ***a*** до ***b*** для четырех функций f1 = x, f2 = sin( 22 \* x ), f3 = x4 и f4 = arctg(x).

Вычисление интеграла оформить в виде функции IntRect.

Вычисления выполнить для пяти значений точности: 0.01, 0.001, 0.0001, 0.00001 и 0.000001.

Исследовать быстродействие алгоритма в зависимости от подынтегральной функции и требуемой точности (быстродействие алгоритма можно оценить числом элементарных прямоугольников ***n***).

Результаты представить в виде 5 таблиц, по одной таблице для каждого значения точности. В каждой таблице выводить данные для всех четырех функций.

Для печати таблицы результатов использовать функцию

void PrintTabl(I\_print i\_prn[],int k), приведенную в приложении 2.

Здесь i\_prn[] – массив структур типа I\_print размерностью k.

Вид таблицы приведен в Приложении 1.

2. Выполнить п.1, используя для интегрирования метод трапеций. Вычисление интеграла оформить в виде функции IntTrap.

Для печати таблиц результатов использовать ту же функцию, что и в методе прямоугольников.

1. Разработка алгоритма

Для решения задачи я создаю многофайловый проект, включающий следующие файлы: **file.h –** заголовочный файл, где я подключаю прототипы функций. **special.cpp –** файл, где прописаны все функции. **laba6kate.cpp** – файл, который вызывает функции и выводит на консоль результат.

double f1(double x), double f2(double x), double f3(double x), double f4(double x) – мат функции, для которых нужно будет искать интеграл. x – аргумент мат функции.

double f1\_t(double x), double f2\_t(double x), double f3\_t(double x), double f4\_t(double x) – преобразованные мат функции (от f1, f2, f3, f4 соответственно) для вычисления точного значения интеграла.

Структура I\_print – для хранения данных для печати результатов интегрирования

struct I\_print{

const char\* name; //название функции

double i\_sum; //значение интегральной суммы методом прямоугольника

double i\_toch; //точное значение интеграла

int n; //число разбиений области интегрирования при котором достигнута требуемая точность методом прямоугольника

double s\_trap; // значение интегральной суммы методом трапеции

int n\_trap; // число разбиений области интегрирования при котором достигнута требуемая точность методом трапеции

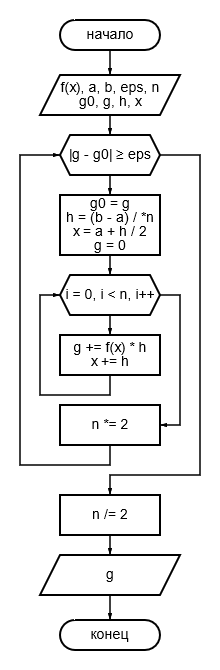
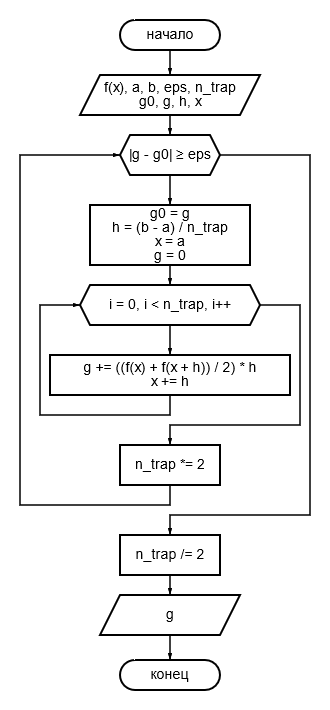
};

void PrintTabl(I\_print[], int) – функция вывода таблицы, передаю структура с данными для заполнения таблицы и количество строк в таблице.

double Trap(double (\*f)(double), double a, double b, double eps, int \*n\_trap) – функция вычисления интеграла методом прямоугольника, передаю функцию для интегрирования, границы для вычисления определённого интеграла, точность, количество разбиений.

double IntRect(double (\*f)(double), double a, double b, double eps, int\* n) – функция вычисления интеграла методом трапеции, передаю функцию для интегрирования, границы для вычисления определённого интеграла, точность, количество разбиений.

Метод прямоугольника Метод трапеции

1. Листинг

**head.h**

#pragma once

#include <cmath>

#include <iostream>

#include <iomanip>

double f1(double x);

double f2(double x);

double f3(double x);

double f4(double x);

double f1\_t(double x);

double f2\_t(double x);

double f3\_t(double x);

double f4\_t(double x);

struct I\_print{

const char\* name;

double i\_sum;

double i\_toch;

int n;

double s\_trap;

int n\_trap;

};

void PrintTabl(I\_print[], int);

double Trap(double (\*f)(double), double a, double b, double eps, int \*n\_trap);

double IntRect(double (\*f)(double), double a, double b, double eps, int\* n);

**special.cpp**

#include "head.h"

double f1(double x) {

return x;}

double f2(double x) {

return sin(22 \* x);}

double f3(double x) {

return pow(x, 4);}

double f4(double x) {

return atan(x);}

double f1\_t(double x) {

return pow(x, 2) / 2;}

double f2\_t(double x) {

return -cos(22 \* x) / 22;}

double f3\_t(double x) {

return pow(x, 5) / 5;}

double f4\_t(double x) {

return atan(x) \* x - 0.5 \* log(1 + pow(x, 2));}

double IntRect(double (\*f)(double), double a, double b, double eps, int\* n) {

\*n = 1;

double g0, g = 0;

double h, x;

do {

g0 = g;

h = (b - a) / \*n;

x = a + h / 2;

g = 0;

for (int i = 0; i < \*n; i++) {

g += f(x) \* h;

x += h;

}

\*n \*= 2;

} while (abs(g - g0) >= eps);

\*n /= 2;

return g;}

double Trap(double (\*f)(double), double a, double b, double eps, int \*n\_trap) {

\*n\_trap = 1;

double g0, g = 0;

double h, x;

do {

g0 = g;

h = (b - a) / \*n\_trap;

x = a;

g = 0;

for (int i = 0; i < \*n\_trap; i++) {

g += ((f(x) + f(x + h)) / 2) \* h;

x += h;

}

\*n\_trap \*= 2;

} while (abs(g - g0) >= eps);

\*n\_trap /= 2;

return g;}

void PrintTabl(I\_print i\_prn[], int k)

{

const int m = 6;//число столбцов таблицы

int wn[m] = { 12,18,18,10, 18, 10 };//ширина столбцов таблицы

const char\* title[m] = { "Function","Integral","IntSum","N ", "Strap ", "Ntrap "};

int size[m];

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

size[i] = strlen(title[i]);

//шапка таблицы

std::cout << char(124) << std::setfill(char(45));

for (int j = 0;j < m - 1;j++)

std::cout << std::setw(wn[j]) << char(124);

std::cout << std::setw(wn[m - 1]) << char(124) << std::endl;

std::cout << char(124);

for (int j = 0;j < m;j++)

std::cout << std::setw((wn[j] - size[j]) / 2) << std::setfill(' ') << ' ' << title[j]

<< std::setw((wn[j] - size[j]) / 2) << char(124);

std::cout << std::endl;

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

{//заполнение таблицы

std::cout << char(45) << std::fixed;

for (int j = 0;j < m - 1;j++)

std::cout << std::setfill(char(45)) << std::setw(wn[j]) << char(45);

std::cout << std::setw(wn[m - 1]) << char(124) << std::setfill(' ') << std::endl;

std::cout << char(124) << std::setw((wn[0] - strlen(i\_prn[i].name)) / 2) << ' ' << i\_prn[i].name

<< std::setw((wn[0] - strlen(i\_prn[i].name)) / 2) << char(124);

std::cout << std::setw(wn[1] - 1) << i\_prn[i].i\_toch << char(124)

<< std::setw(wn[2] - 1) << i\_prn[i].i\_sum << char(124)

<< std::setw(wn[3] - 1) << i\_prn[i].n << char(124)

<< std::setw(wn[4] - 1) << i\_prn[i].s\_trap << char(124)

<< std::setw(wn[3] - 1) << i\_prn[i].n\_trap << char(124) << std::endl;

}

//низ таблицы

std::cout << char(45) << std::setfill(char(45));

for (int j = 0;j < m - 1;j++)

std::cout << std::setw(wn[j]) << char(45);

std::cout << std::setw(wn[m - 1]) << char(45) << std::setfill(' ') << std::endl;

}

**laba6kate.cpp**

#include <iostream>

#include <cmath>

#include "head.h"

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

system("color F0");

double (\*mas[4]) (double) = { f1,f2,f3,f4 };

double (\*mas\_t[4]) (double) = { f1\_t,f2\_t,f3\_t,f4\_t };

I\_print m[6];

m[0].name = "y=x ";

m[1].name = "y=sin(22x)";

m[2].name = "y=x^4 ";

m[3].name = "y=arctg(x)";

int ch;

int\* n = &ch;

\*n = 0;

double eps = 0.01;

double a, b, sum;

int g;

int \*n\_trap = &g;

\*n\_trap = 0;

cin >> a >> b;

int k = 4;

while (eps >= 0.000001) {

for (int i = 0;i < 4;i++) {

m[i].i\_sum = IntRect(mas[i], a, b, eps, n);

m[i].n = \*n;

m[i].i\_toch = mas\_t[i](b) - mas\_t[i](a);

m[i].s\_trap = Trap(mas[i], a, b, eps, n\_trap);

m[i].n\_trap = \*n\_trap;

}

std::cout << fixed;

std::cout << setprecision(k);

std::cout << "Метод прямоугольника и трапеции для погрешности " << eps \* 10 << endl;

PrintTabl(m, 4);

std::cout << endl;

eps /= 10;

k += 1;

}

system("pause");}

4) Тестирование