Московский государственный технический

университет им. Н.Э. Баумана

Курс «Основы информатики»

Отчёт по лабораторной работе №6

Численное интегрирование функции.

Проверил:

Преподаватель каф. ИУ5

Аксёнова М.В.

Подпись и дата:

Факультет «Информатика и система управления»

Кафедра ИУ5 «Система обработки информации и управления»

Выполнил:

Студент группы ИУ5-13Б

Пермяков Дмитрий Кириллович

Подпись и дата:

**Постановка задачи.**

1. 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.

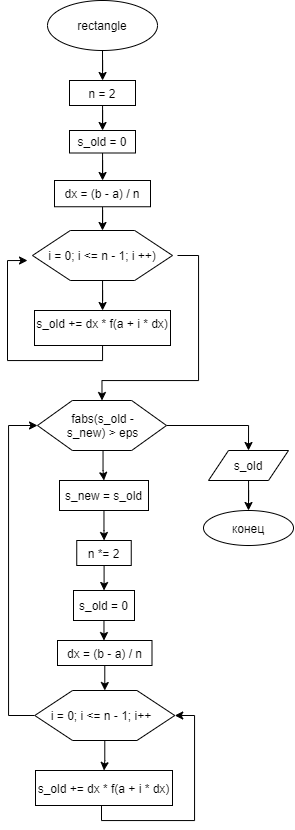
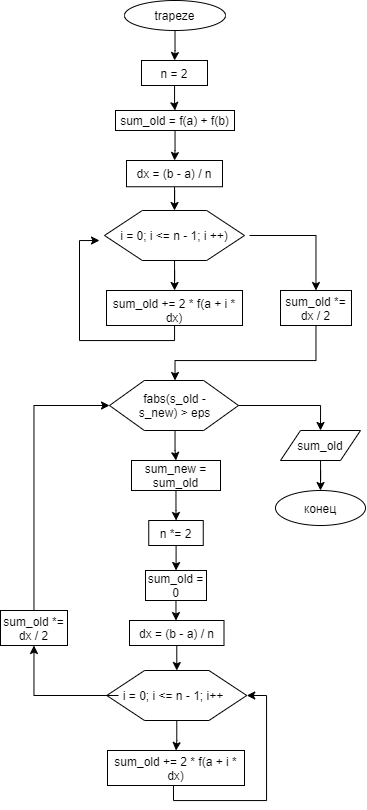
**Разработка алгоритма**

Переменные: double a, b – нижний и верхний предел интегрирования, int n – кол-во прямоугольников / трапеций, eps – возможная погрешность.

Функции: double f1(double x), double f2(double x), double f3(double x), double f4(double x) – принимают Х и возвращают значения функции в точках: х, sin(22x), x^4, atanx.

double rectangle (double (\*f) (double), double a, double b, double eps, int& n) – принимает функции f1-f4, a и b, eps, n и возвращает значение интеграла методом прямоугольников.

double trapeze (double (\*f) (double), double a, double b, double eps, int& n) – принимает функции f1-f4, a и b, eps, n и возвращает значение интеграла методом трапеций.

Функция метода прямоугольников Функция метода трапеций

**Код программы**

**Заголовки**

#pragma once

#include <iostream>

#include <cmath>

#include <math.h>

#include <iomanip>

#define NAME1 new char[7]{ "y = x " }

#define NAME2 new char[12]{ "y=sin(22x)" }

#define NAME3 new char[12]{ "y = x^4 " }

#define NAME4 new char[12]{ "y = atanx " }

#define NAME5 new char[12]{ "y = x^5 " }

using namespace std;

typedef double(\*TPF)(double);

typedef double(\*TPFEV)();

enum constant\_variables

{

A = -1,

B = 3,

SIZE = 5

};

struct I\_print

{

char\* name;

double i\_sum;

double i\_toch;

int n;

};

double Exactness\_F1();

double Exactness\_F2();

double Exactness\_F3();

double Exactness\_F4();

double Exactness\_F5();

double F1(double x);

double F2(double x);

double F3(double x);

double F4(double x);

double F5(double x);

double rectangle(TPF, double a, double b, double eps, int& n);

double trapeze(TPF, double a, double b, double eps, int& n);

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

Функции

#include "Header.h"

double F1(double x)

{

return x;

}

double F2(double x)

{

return sin(22 \* x);

}

double F3(double x)

{

return x \* x \* x \* x;

}

double F4(double x)

{

return atan(x);

}

double F5(double x) // доп задание

{

return x \* x \* x \* x \* x;

}

double Exactness\_F1()

{

return (B \* B - A \* A) / 2.0;

}

double Exactness\_F2()

{

return (cos(A \* 22.0) - cos(B \* 22.0)) / 22.0;

}

double Exactness\_F3()

{

return (B \* B \* B \* B \* B - A \* A \* A \* A \* A) / 5.0;

}

double Exactness\_F4()

{

return B \* atan((double)B) - A \* atan((double)A) - (log(B \* B + 1) - log(A \* A + 1)) / 2.0;

}

double Exactness\_F5()

{

return (B \* B \* B \* B \* B \* B - A \* A \* A \* A \* A \* A) / 6.0;

}

double rectangle(TPF F, double a, double b, double eps, int& n)

{

n = 2;

double s\_old = 0, s\_new = 0, dx;

dx = (b - a) / n;

for (size\_t i = 0; i <= n; i++)

s\_old += dx \* F(a + (i-0.5) \* dx);

do

{

s\_new = s\_old;

n \*= 2;

s\_old = 0;

dx = (b - a) / n;

for (size\_t i = 0; i <= n; i++)

s\_old += dx \* F(a + (i - 0.5) \* dx);

} while (fabs(s\_old - s\_new) > eps);

return s\_old;

}

double trapeze(TPF F, double a, double b, double eps, int& n)

{

n = 2;

double dx = (b - a) / n;

double sum = 0, sumNew = 0;

for (size\_t step = 0; step < n; step++)

{

double x1 = a + step \* dx;

double x2 = a + (step + 1) \* dx;

sum += 0.5 \* (x2 - x1) \* (F(x1) + F(x2));

}

do

{

sumNew = sum;

sum = 0;

n \*= 2;

dx = (b - a) / n;

for (size\_t step = 0; step < n; step++)

{

double x1 = a + step \* dx;

double x2 = a + (step + 1) \* dx;

sum += 0.5 \* (x2 - x1) \* (F(x1) + F(x2));

}

} while (fabs(sumNew - sum) > eps);

return sum;

}

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

{

const int m = 4;//число столбцов таблицы

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

char title[m][20] = { "Function","Integral","IntSum","N " };

int size[m];

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

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

//шапка таблицы

cout << char(218) << setfill(char(196));

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

cout << setw(wn[j]) << char(194);

cout << setw(wn[m - 1]) << char(191) << endl;

cout << char(179);

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

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

<< setw((wn[j] - size[j]) / 2) << char(179);

cout << endl;

//заполнение таблицы

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

{

cout << char(195) << fixed;

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

cout << setfill(char(196)) << setw(wn[j]) << char(197);

cout << setw(wn[m - 1]) << char(180) << setfill(' ') << endl;

cout << char(179) << setw((wn[0] - strlen(i\_prn[i].name)) / 2) << ' ' << i\_prn[i].name

<< setw((wn[0] - strlen(i\_prn[i].name)) / 2) << char(179);

cout << setw(wn[1] - 1) << setprecision(10) << i\_prn[i].i\_toch << char(179)

<< setw(wn[2] - 1) << i\_prn[i].i\_sum << setprecision(6) << char(179)

<< setw(wn[3] - 1) << i\_prn[i].n << char(179) << endl;

}

//низ таблицы

cout << char(192) << setfill(char(196));

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

cout << setw(wn[j]) << char(193);

cout << setw(wn[m - 1]) << char(217) << setfill(' ') << endl;

}

**Main**#include "Header.h"

int main()

{

double a = -1, b = 3; int n = 2;

I\_print integrals[SIZE]{};

TPF functions[] = { F1, F2, F3, F4, F5 };

TPFEV func\_toch[] = { Exactness\_F1 ,Exactness\_F2 ,Exactness\_F3 ,Exactness\_F4 ,Exactness\_F5 };

char\* func\_names[] = { NAME1, NAME2, NAME3, NAME4, NAME5 };

for (size\_t i = 0; i < SIZE; i++)

{

integrals[i].name = func\_names[i];

integrals[i].i\_toch = func\_toch[i]();

}

cout << "\x1b[31m Method of rectangle \x1b[0m\n";

double eps = 0.01;

while (eps > 0.000001)

{

cout << "\x1b[34m" << "eps = " << eps << "\x1b[0m\n";

for (size\_t i = 0; i < SIZE; i++)

{

integrals[i].i\_sum = rectangle(functions[i], a, b, eps, n);

integrals[i].n = n;

}

eps \*= 0.1;

PrintTabl(integrals, SIZE);

}

cout << "\x1b[31m Method of trapeze \x1b[0m \n";

eps = 0.01;

while (eps > 0.000001)

{

cout << "\x1b[34m" << "eps = " << eps << "\x1b[0m\n";

for (size\_t i = 0; i < SIZE; i++)

{

integrals[i].i\_sum = trapeze(functions[i], a, b, eps, n);

integrals[i].n = n;

}

eps \*= 0.1;

PrintTabl(integrals, SIZE);

}

return 27;

}

Тестирование

