**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Основы программирования»

Отчет по лабораторной работе №6

«Численное интегрирование функции»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-12Б  Казицин Алексей |  | преподаватель каф. ИУ5  Козлов А.Д. |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2021 г.

Постановка задачи

**Цель работы**. На примере разработки программы для численного интегрирования функции с заданной точностью методом прямоугольников и методом трапеций освоить следующие приемы программирования:

- передача в функцию параметров «по значению» и «по адресу»;

- передача в функцию имени функции;

- передача одномерных массивов в функцию;

- объединение разнородных данных в структуру;

- использование массивов из элементов типа структура;

**Задание.**

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

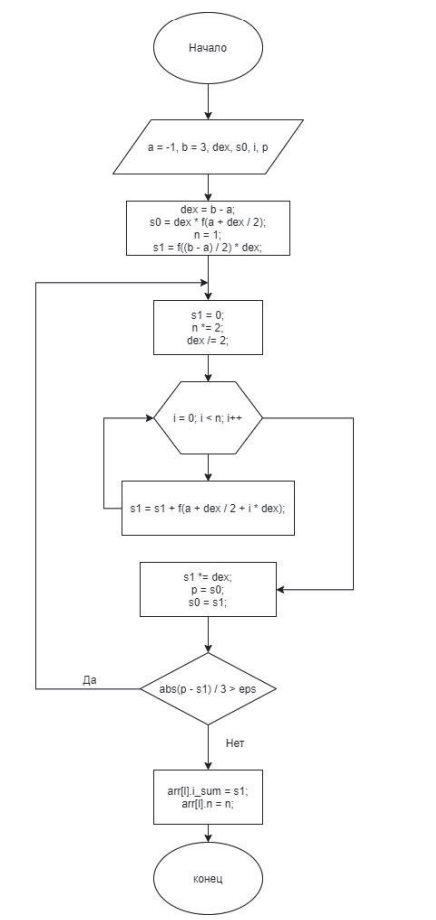
**Описание используемых переменных с указанием наименования, типа (int, float, и т.п.) и назначения в программе:**

double s1(значение функции); double n(счетчик);

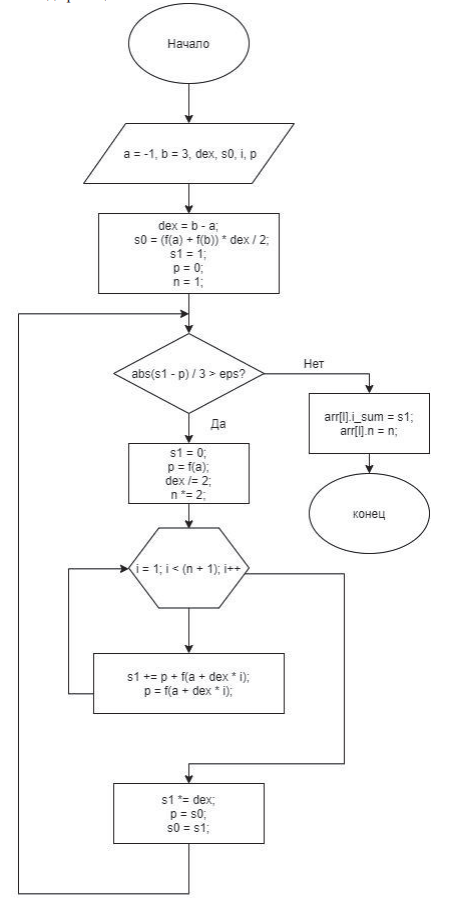
double a(левая граница интеграла); double b(правая граница интеграла, значение f1,f2,f3,f4); double dex(дельта х); double s0(вспомогательное значение функции);

arr[4](массив с названием функции, значением интегральной суммы, точным значением интеграла и числом разбитий области интегрирования)(структурная переменная I\_print); double i(счетчик, значение точности); int l(счетчик);

Метод прямоугольников



Метод трапеций



Текст программы

#include <iostream>

#include <string>

#include <iomanip>

#include <cmath>

#include <cstring>

#include <windows.h>

using namespace std;

struct I\_print { //данные для печати результатов интегрирования

char\* name; //название функции

double i\_sum; //значение интегральной суммы

double i\_toch; //точное значение интеграла

int n; //число разбиений области интегрирования

//при котором достигнута требуемая точность

};

typedef double (\*TPF)(double);

double f0(double p) {return p;}

double f1(double p) {return sin(22 \* p);}

double f2(double p) {return p \* p \* p \* p;}

double f3(double p) {return atan(p);}

void IntRect(double& s1, double& n, I\_print\* arr, double eps, TPF f, int l){

double a = -1, b = 3, dex, s0, i, p;

dex = b - a;

s0 = dex \* f(a + dex / 2);

n = 1;

s1 = f((b - a) / 2) \* dex;

do{

s1 = 0;

n \*= 2;

dex /= 2;

for (i = 0; i < n; i++)

s1 = s1 + f(a + dex / 2 + i \* dex);

s1 \*= dex;

p = s0;

s0 = s1;

} while (abs(p - s1) / 3 > eps);

arr[l].i\_sum = s1;

arr[l].n = n;

}

void IntTrap(double& s1, double& n, I\_print\* arr, double eps, TPF f, int l){

double a = -1, b = 3, dex, s0, i, p;

dex = b - a;

s0 = (f(a) + f(b)) \* dex / 2;

s1 = 1;

p = 0;

n = 1;

while (abs(s1 - p) > eps){

s1 = 0;

p = f(a);

dex /= 2;

n \*= 2;

for (i = 1; i < (n + 1); i++){

s1 += p + f(a + dex \* i);

p = f(a + dex \* i);

}

s1 \*= dex / 2;

p = s0;

s0 = s1;

}

arr[l].i\_sum = s1;

arr[l].n = n;

}

void zapoln(I\_print\* arr){

double n = 1, s1, a = -1, b = 3;

arr[0].name = new char[strlen("y=x ") + 1];

strcpy\_s(arr[0].name, strlen("y=x ") + 1, "y=x ");

arr[1].name = new char[strlen("y=sin22\*x ") + 1];

strcpy\_s(arr[1].name, strlen("y=sin22\*x ") + 1, "y=sin22\*x ");

arr[2].name = new char[strlen("y=x^4 ") + 1];

strcpy\_s(arr[2].name, strlen("y=x^4 ") + 1, "y=^4x ");

arr[3].name = new char[strlen("y=arctgx ") + 1];

strcpy\_s(arr[3].name, strlen("y=arctgx ") + 1, "y=arctgx ");

arr[0].i\_toch = (b \* b - a \* a) / 2.0;

arr[1].i\_toch = (cos(a \* 22.0) - cos(b \* 22.0)) / 22.0;

arr[2].i\_toch = (b \* b \* b \* b \* b - a \* a \* a \* a \* a) / 5.0;

arr[3].i\_toch = b \* atan(b) - a \* atan(a) - (log(b \* b + 1) - log(a \* a + 1)) / 2.0;

}

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

system("chcp 866>nul");

const int m = 4;//число столбцов таблицы

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

char title[m][9] = { "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;

for (int i = 0;i < k;i++){//заполнение таблицы

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 << 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) << endl;

//восстановление первоначальных значений

cout << setprecision(6) << setfill(' ');

SetConsoleCP(65001);

SetConsoleOutputCP(65001);

}

int main(){

SetConsoleCP(65001);

SetConsoleOutputCP(65001);

double n = 1, s1, i;

int l;

I\_print arr[4];

zapoln(arr);

TPF A[4] = { f0, f1, f2, f3 };

cout << "Область интегрирования функций: -1 <= x <= 3" << endl;

cout << "Метод прямоугольников" << endl;

for (i = 0.01; i >= 0.000001; i \*= 0.1){

cout << "Для eps=" << i << endl;

for (l = 0; l < 4;l++){

IntRect(s1, n, arr, i, A[l], l);

}

PrintTabl(arr, 4);

}

cout << "Метод трапеций" << endl;

for (i = 0.01; i >= 0.000001; i \*= 0.1){

cout << "Для eps=" << i << endl;

for (l = 0; l < 4;l++){

IntTrap(s1, n, arr, i, A[l], l);

}

PrintTabl(arr, 4);

}

return 0;

}

Анализ результатов