**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Основы информатики»

Отчет по лабораторной работе №6

# «Численное интегрирование функции»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-11 |  | преподаватель каф. ИУ5 |
| Анцифров Никита |  | Аксёнова М.В. |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2019 г.

**Постановка задачи**

1. Вычислить определённый интеграл  в пределах от **a** до **b** для четырех функций

f1 = x, f2 = sin(22\*x), f3 = x4 и f4 = arctg(x) методом прямоугольников.

Вычисление интеграла с использованием метода прямоугольников оформить в виде функции IntRect.

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

1. 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.

1. Выполнить п.1, используя для интегрирования метод трапеций.

Вычисление интеграла оформить в виде функции IntTrap.

**Описание входных, выходных и вспомогательных данных**

*Переменные:*

int **n** – количество прямоугольников

double **a** и double **b** – границы интервала

double **eps** – точность

I\_print **StructTable[4]** – переменная для записи найденных данных

int **i** и int **j** – счетчики циклов

const int **m** – вспомогательная переменная для обозначения размеров массивов

I\_print **i\_prn[]** – переданная в функцию структурная переменная

int **k** – переданная в функцию переменная, обозначающая количество строк

int **wn[m]** иint **size[m]** – вспомогательные массивы для создания «рамки» таблицы

char **\*title[m]** – наименование столбцов в таблице

TPF **f** – переданная в функцию одна из четырех функций

double **I1** и **I2** – значение интегралов

double **x** – значение аргумента

double **dx** – шаг подсчета интеграла

*Структуры:*

struct I\_print

{

char\* **name**; - название функции

double **i\_sum**; - полученное значение интеграла методом прямоугольника/трапеций

double **i\_toch**; - значение интеграла, рассчитанного по формуле

int **n**; - число прямоугольников, при котором достигнута требуемая точность

};

**Описание прототипов функций**

double **fi** (double)**; -** функции принимают значения аргумента и возвращает значение функции (1<=**i**<=4)

void **PrintTabl** (I\_print, int); - функция построения таблицы: принимает значение структурной переменной и числа строк

double **IntRect** (TPF, double, double, double, int\*) – функция принимает ссылку на одну из четырех заданных функций, значения границ интервала, значение точности и адрес на переменную, отвечающую за число прямоугольников, при котором достигнута требуемая точность; возвращает значение интеграла, посчитанного методом прямоугольников

double **IntTrap** (TPF, double, double, double, int\*) – функция принимает ссылку на одну из четырех заданных функций, значения границ интервала, значение точности и адрес на переменную, отвечающую за число прямоугольников, при котором достигнута требуемая точность; возвращает значение интеграла, посчитанного методом трапеций

**Алгоритм**

Функция **main**







Функция **f1**



Функция **f2**



Функция **f3**



Функция **f4**



Функция **PrintTabl**

 

Функция **IntRect**



Функция **IntTrap**



**Текст программы**

*L6.cpp*

#include <iostream>

#include <iomanip>

#include <cmath>

using namespace std;

typedef double (\*TPF)(double);

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);}

TPF functions[]={f1, f2, f3, f4};

struct I\_print

{

char\* name;

double i\_sum;

double i\_toch;

int n;

};

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

{

const int m=4;

int wn[m]={12, 18, 18, 10};

char \*title[m]={"Function", "Integral", "IntSum", "n "};

int size[m];

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

{

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

}

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++)

{

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;

}

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;

}

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

{

double I1=0, I2=0, x=0, dx=0;

dx=b-a;

x=a+dx/2;

I1=f(x)\*dx;

do

{

if ((\*n)!=1)

{

I1=I2;

}

(\*n)\*=2;

I2=0;

x=a+dx/(2\*(\*n));

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

{

I2+=f(x)\*(dx/(\*n));

x=x+dx/(\*n);

}

}

while ((abs(I2-I1)/3)>eps);

return I2;

}

double IntTrap (TPF f, double a, double b, double eps, int\* n)

{

double I1=0, I2=0, x=0, dx=0;

dx=b-a;

x=a;

I1=((f(a)+f(b))/2)\*dx;

do

{

if ((\*n)!=1)

{

I1=I2;

}

(\*n)\*=2;

I2=0;

x=a;

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

{

I2+=((f(x)+f(x+dx/(\*n)))/2)\*(dx/(\*n));

x=x+dx/(\*n);

}

}

while (abs(I2-I1)>eps);

return I2;

}

int main()

{

int n=1;

double a=-1, b=3, eps=0.01;

I\_print StructTable[4];

cout << "I.METHOD OF RECTANGLES" << endl;

for (int j=0; j<5; j++)

{

cout << "Accuracy: " << eps << endl;

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

StructTable[0].i\_sum=IntRect(functions[0],a,b,eps,&n);

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

StructTable[0].n=n;

n=1;

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

StructTable[1].i\_sum=IntRect(functions[1],a,b,eps,&n);

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

StructTable[1].n=n;

n=1;

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

StructTable[2].i\_sum=IntRect(functions[2],a,b,eps,&n);

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

StructTable[2].n=n;

n=1;

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

StructTable[3].i\_sum=IntRect(functions[3],a,b,eps,&n);

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

StructTable[3].n=n;

PrintTabl(StructTable, 4);

eps/=10;

cout << endl;

}

eps=0.01;

n=1;

cout << "II.METHOD OF TRAPEZES" << endl;

for (int j=0; j<5; j++)

{

cout << "Accuracy: " << eps << endl;

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

StructTable[0].i\_sum=IntTrap(functions[0],a,b,eps,&n);

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

StructTable[0].n=n;

n=1;

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

StructTable[1].i\_sum=IntTrap(functions[1],a,b,eps,&n);

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

StructTable[1].n=n;

n=1;

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

StructTable[2].i\_sum=IntTrap(functions[2],a,b,eps,&n);

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

StructTable[2].n=n;

n=1;

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

StructTable[3].i\_sum=IntTrap(functions[3],a,b,eps,&n);

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

StructTable[3].n=n;

PrintTabl(StructTable, 4);

eps/=10;

cout << endl;

}

system("pause");

return 0;

}

*func\_proto.h*

typedef double (\*TPF)(double);

struct I\_print;

double f1 (double);

double f2 (double);

double f3 (double);

double f4 (double);

void PrintTabl (I\_print, int);

double IntRect (TPF, double, double, double, int\*);

double IntTrap (TPF, double, double, double, int\*);

**Анализ результатов**





