|  |  |  |
| --- | --- | --- |
| **Лабораторная работа № 1****По вычислительной математике**«Вычисление определенного интеграла» | Студент | Муха А. В. |
| Группа | ИВТ-262 |
| Дата выполнения |  |
| Оценка |  |
| Преподаватель | Скворцов М.И. |

1. **Постановка задачи**

Вычисление определённого интеграла с заданной погрешностью

 $F\left(x\right)=\frac{x}{1+tgx}$

 На интервале [0, π/6]

Используемые методы

1. Метод средних прямоугольников.
2. Метод трапеций.
3. Метод Симпсона.

**2. Графики**

1. Метод средних прямоугольников



**2.Метод трапеций**



**3.Метод Симпсона**

****

**3.Блок-схема**

1: Ср.прямоуг

2: Трапеций

3: Симпсона

 choice

choice ≠ 4

 1

 2

 3

да

нет

choice

Ср.прямоуг.

1

2

 3

Трапеций

Симпсона

 4

**4.Таблица**

|  |  |
| --- | --- |
| ξ | Методы |
| Средних прямоугольников | Трапеций | Симпсона |
| Результатинтеграл | Числоотрезков**N** | Результатинтеграл | Числоотрезков**N** | Результатинтеграл | Числоотрезков**N**  |
| 10-4 |  0.101193 | 16 | 0.101106 | 16 | 0.101155 | 4 |
| 10-5 | 0.101171 | 32 | 0.101160 | 64 | 0.101155 | 4 |
| 10-6 | 0.101164 | 128 | 0.101163 | 128 | 0.101163 | 8 |

**Код программы**

#include "stdafx.h"

#include <conio.h>

#include <math.h>

#include <windows.h>

#include <iostream>

using namespace std;

void Rectangle(double lim1, double lim2, double epsil);

void Trapeze(double lim1, double lim2, double epsill);

void Simpson(double lim1, double lim2, double epsil);

void Output(double result, int n);

double Func(double x);

int \_tmain(int argc, \_TCHAR\* argv[])

{

 char choice; // переменная выбора

 char buf[54]; // строка

 double lim1, // нижний предел

 lim2, // верхний предел

 epsil; // погрешность

 do

 {

 system("cls");

 CharToOem("Программа вычисления определённого интеграла!", buf);

 cout << buf << endl << endl;

 CharToOem("\t x\nФункция: -------\n\t 1 + tg x", buf);

 cout << buf << endl << endl;

 CharToOem("1. Метод средних прямоугольников.", buf);

 cout << buf << endl;

 CharToOem("2. Метод трапеций.", buf);

 cout << buf << endl;

 CharToOem("3. Метод Симпсона.", buf);

 cout << buf << endl;

 CharToOem("4. Выход.", buf);

 cout << buf << endl << endl;

 do

 {

 CharToOem("Введите цифру, соответствующую методу: ", buf);

 cout << buf;

 fflush(stdin);

 cin >> choice;

 cout << endl << endl;

 }

 while ((int)choice < 49 || (int)choice > 52);

 if ((int)choice == 52)

 {

 CharToOem("Программа завершена", buf);

 cout << buf << endl << endl << endl;

 // Задержка

 system("pause");

 return 0;

 }

 ////////////////////////////////////////////////

 // Ввод пределов интегрирования

 do

 {

 fflush(stdin);

 CharToOem("Введите пределы интегрирования [0, pi/6]\na = ", buf);

 cout << buf;

 cin >> lim1;

 fflush(stdin);

 CharToOem("b = ", buf);

 cout << buf;

 cin >> lim2;

 if ((lim1 < 0) || (lim1 >= 0.5235987756) || (lim2 > 0.5235987756) || (lim2 <= 0))

 {

 CharToOem("\nОшибка в указании интервала!!\a", buf);

 cout << buf << endl << endl << endl;

 }

 }

 while ((lim1 < 0) || (lim1 >= 0.5235987756) || (lim2 > 0.5235987756) || (lim2 <= 0));

 // Ввод погрешности

 do

 {

 CharToOem("\nВведите погрешность вычислений\nepsil = ", buf);

 cout << buf;

 cin >> epsil;

 }

 while (1 != 1);

 if ((int)choice == 49)

 {

 Rectangle(lim1, lim2, epsil);

 }

 else if ((int)choice == 50)

 {

 Trapeze(lim1, lim2, epsil);

 }

 else if ((int)choice == 51)

 {

 Simpson(lim1, lim2, epsil);

 }

 else

 {

 CharToOem("Программа завершена", buf);

 cout << buf << endl << endl << endl;

 }

 // Задержка

 system("pause");

 }

 while (choice != 4);

 return 0;

}

void Rectangle(double lim1, double lim2, double epsil)

{

 double result;

 int i;

 int n;

 double h;

 double s1;

 double s2;

 n = 1;

 h = lim2 - lim1;

 s2 = h \* Func((lim1 + lim2) / 2);

 do

 {

 n \*= 2;

 s1 = s2;

 h /= 2;

 s2 = 0;

 i = 1;

 do

 {

 s2 = s2 + Func(lim1 + h/2 + h\*(i - 1));

 i++;

 }

 while (i <= n);

 s2 = s2 \* h;

 }

 while (fabs(s2 - s1) > 3\*epsil);

 result = s2;

 Output(result, n);

}

void Trapeze(double lim1, double lim2, double epsil)

{

 double result;

 int i;

 int n;

 double h;

 double s1;

 double s2;

 n = 1;

 h = lim2 - lim1;

 s2 = h \* (Func(lim1) + Func(lim2)) / 2;

 do

 {

 s1 = s2;

 s2 = 0;

 i = 1;

 do

 {

 s2 = s2 + Func(lim1 - h/2 + h\*i);

 i++;

 }

 while(i <= n);

 s2 = s1/2 + s2 \* h/2;

 n \*= 2;

 h /= 2;

 }

 while(fabs(s2 - s1) > 3\*epsil);

 result = s2;

 Output(result, n);

}

void Simpson(double lim1, double lim2, double epsil)

{

 double result;

 int n;

 double h;

 double s;

 double s1;

 double s2;

 double s3;

 double x;

 n = 1;

 s2 = 1;

 h = lim2 - lim1;

 s = Func(lim1) + Func(lim2);

 do

 {

 n \*= 2;

 s3 = s2;

 h /= 2;

 s1 = 0;

 x = lim1 + h;

 do

 {

 s1 = s1 + 2\*Func(x);

 x = x + 2\*h;

 }

 while(x < lim2);

 s = s + s1;

 s2 = (s + s1) \* h / 3;

 x = fabs(s3 - s2) / 15;

 }

 while(x > epsil);

 result = s2;

 Output(result, n);

}

void Output(double result, int n)

{

 char buf[13];

 // Вывод результатов

 CharToOem("\nИнтеграл = ", buf);

 cout << buf << result;

 CharToOem("\tN = ", buf);

 cout << buf << n << endl << endl << endl;

}

double Func(double x)

{

 return x / (1 + sin(x) / cos(x));

}