Министерство науки и высшего образования РФ Севастопольский государственный университет

Кафедра «Информатика и управление в технических системах»

**Лабораторная работа №2**

Численные методы интегрирования функций

Выполнил: ст. гр. УТС/б-21-1-о

Рябый А.И.

Проверил: ст.пр. Липко И.Ю.

Севастополь 2021

**Цель работы**

Реализация алгоритмов численного интегрирования путём их программной реализации и исследования ошибок интегрирования.

**Результаты**

**Вариант 16:** Вычисления определенного интеграла от некоторой функции

на промежутке [a, b]=[0, -1]

Код программы:

#include <iostream>

#include <cmath>

//var16

/\*

func (-3)/((15\*x-9)\*(15\*x-9)\*(15\*x-9))

D: [-1;0]

\*/

using namespace std;

float Sum1 = 0; //начало отсчёта каждой суммы 1,2,3...

float Sum2 = 0;

float Sum3 = 0;

float b = 0; //конец интегрирования

float a = -1; //начало интегрирования

float n =2400; //2400

float x = a;

float h = (b-a)/n;

float h1 = (b-a)/(n\*2);

float Z = - 0.00106096; //полный результат интегрирования

float error1 = 0;

float error2 = 0;

float **f**(float x){

return ((-3)/((15\*x-9)\*(15\*x-9)\*(15\*x-9))) ;

}

void **funcPrammoug**(){

for(int j = 0; j<n;j++){

Sum1-=f(x)\*h;

x+=h;

}

error1 = Sum1;

std::cout<<"Pryamoug^-^ = "<<Sum1<<std::endl;

}

void **funcSimpson**(){

float x1,x2;

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

x1 = a + i \* h;

x2 = a + (i+1)\*h;

Sum2 -= (x2-x1)/6.0\*(f(x1)+4.0\*f(0.5\*(x1+x2))+f(x2));

}

error2 = Sum2 ;

std::cout<<"Simpson^-^ = "<<Sum2;

std::cout<<endl;

}

void **trapeciya**(){

float Sum3 =0;

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

float x1 = a + i\*h;

float x2= a+(i+1)\*h;

Sum3-= 0.5\*(x2-x1)\*(f(x1)+f(x2));

}

std::cout<<"Trapeciya^-^ = "<<Sum3;

std::cout<<endl;

}

float **runge**(float s1, float s2)

{

s1 = (fabs(s1 - s2))/15;

return s1;

}

int **main**()

{

std::cout<<"n = "<<n<<std::endl;

funcPrammoug();

funcSimpson();

trapeciya();

std::cout<<endl;

std::cout <<"Runge \*\_\* = " << runge(Sum1, Sum2) << std::endl;

std::cout<<endl;

//формулы вычисления разности методов и результата калькулятора

error1 = fabs(Z - error1);

error2 = fabs(Z - error2);

//вывод ошибок

std::cout << "ERROR\_1: " << error1 << std::endl;

std::cout << "ERROR\_2: " << error2 << std::endl;

return 0;

}

Рис.1 график Метода прямоугольника.

Рис.2 график Метода Симпсона.

Рис.3 график Метода Трапеций.

**Вывод**

В ходе работы были проверены методы численного интегрирования функции такие, как метод средних прямоугольников, трапеций, Симпсона. Была написана программа, с помощью которой путём нахождения погрешности были проверены вышеперечисленные методы. После нахождения погрешностей каждого метода были построены графики, на основание которых можно сделать следующий вывод:

Наименьшую погрешность в сравнении со всеми методами имеет способ Симпсона, его погрешность практически гораздо быстрее спадает до незначительных значений. Самым нестабильным методом оказался метод средних прямоугольников, но несмотря на это самым точным оказался метод Симпсона.