МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Воронежский государственный технический университет»

(ФГБОУ ВО «ВГТУ», ВГТУ)

Факультет Информационных Технологий и Компьютерной Безопасности

Кафедра систем управления и информационных технологий в строительстве

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

по дисциплине Информатика

### Тема «решение системы линейных алгебраических уравнений (СЛАУ) методом Якоб, методом Зейделя, матричным методом, методом Гаусса»

Выполнил студент: Рыженков Евгений

Группа: бИСТ-221

Руководитель: доцент, к.т.н. Ефимова О.Е.

Работа защищена « » 2023г.

С оценкой \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

подпись

Воронеж 2023

**Лабораторная работа №5.**

**Решение систем линейных алгебраических уравнений.**

Цель работы: найти решение системы линейных алгебраических уравнений разными способами: методом Гаусса, матричным методом, методом Якоби и методом Зейделя.

**Задания на лабораторную работу**

1. Вычислить определенный интеграл с заданной точностью на отрезках разбиения n = 10, n = 50, n = 100:
   1. Методом левых прямоугольников;
   2. Методом правых прямоугольников;
   3. Методом средних прямоугольников
   4. Методом трапеций;
   5. Методом Симпсона.
2. Вычислить точное значение заданного интеграла с помощью формулы Ньютона-Лейбница.
3. Каждый метод должен быть оформлен в виде процедуры.
4. Для каждого из трех случаев найти абсолютную погрешность. Вычисления округлить до 4 знаков после запятой.
5. Провести сравнительный анализ результатов.

**Результат Выполнения заданий**

|  |
| --- |
|  |
| Рисунок 1 – результаты для 10 разбиений. |

|  |
| --- |
|  |
| Рисунок 2 – результаты для 50 разбиений. |

|  |
| --- |
|  |
| Рисунок 3 – результаты для 100 разбиений. |

**Листинг программы на языке С++**

#define \_USE\_MATH\_DEFINES

#include <iostream>

#include <math.h>

#include <iomanip>

using namespace std;

double f(double x) {

return ((x \* x) / (sqrt(x \* x + 2)));

}

double center(double a, double b, int n) {

double h;

int i;

h = (b - a) / n;

double Integral = 0.0;

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

Integral += h \* f(a + h \* (i - 0.5));

}

return Integral;

}

double left(double a, double b, int n) {

double h;

int i;

h = (b - a) / n;

double Integral = 0.0;

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

Integral += h \* f(a + h \* i);

}

return Integral;

}

double right(double a, double b, int n) {

double h;

int i;

h = (b - a) / n;

double Integral = 0.0;

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

Integral += h \* f(a + h \* i);

}

return Integral;

}

double F(double x) {

return ((x \* sqrt(x \* x + 2)) / 2) - log(abs(sqrt(x \* x + 2) + x));

}

double newton\_leibniz(double a, double b) {

return F(b) - F(a);

}

double trapecia(double a, double b, int n) {

double h;

int i;

h = (b - a) / n;

double Integral = 0.0;

Integral = h \* (f(a) + f(b)) / 2.0;

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

Integral += h \* f(a + h \* i);

}

return Integral;

}

double parables(double a, double b, int n) {

double h;

int i;

h = (b - a) / n;

double Integral = 0.0; //интеграл

Integral = h \* (f(a) + f(b)) / 6.0;

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

Integral += 4.0 / 6.0 \* h \* f(a + h \* (i - 0.5));

}

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

Integral += 2.0 / 6.0 \* h \* f(a + h \* i);

}

return Integral;

}

int main() {

setlocale(LC\_ALL, "RUS");

int j = 1;

int i; //счётчик

double a = 0.6, b = 2; //отрезок интегрирования

int n = 10; //число разбиений n

int n1 = 50;

int n2 = 100;

cout << " Для 10 разбиений: " << endl;

cout << "Методом средних прямоугольников: " << center(a, b, n) << endl;

cout << "Абсолютная погрешность: " << abs(center(a, b, n) - (round((center(a, b, n)) \* 10000) / 10000)) << endl;

cout << "Методом правых прямоугольников: " << right(a, b, n) << endl;

cout << "Абсолютная погрешность: " << abs(right(a, b, n) - (round((right(a, b, n)) \* 10000) / 10000)) << endl;

cout << "Методом левых прямоугольников: " << left(a, b, n) << endl;

cout << "Абсолютная погрешность: " << abs(left(a, b, n) - (round((left(a, b, n)) \* 10000) / 10000)) << endl;

cout << "Формулой Ньютона-Лейбница: " << newton\_leibniz(a, b) << endl;

cout << "Абсолютная погрешность: " << abs(newton\_leibniz(a, b) - (round((newton\_leibniz(a, b)) \* 10000) / 10000)) << endl;

cout << "Методом трапеций: " << trapecia(a, b, n) << endl;

cout << "Абсолютная погрешность: " << abs(trapecia(a, b, n) - (round((trapecia(a, b, n)) \* 10000) / 10000)) << endl;

cout << "Методом трапеций: " << parables(a, b, n) << endl;

cout << "Абсолютная погрешность: " << abs(parables(a, b, n) - (round((parables(a, b, n)) \* 10000) / 10000)) << endl;

cout << endl << " Для 50 разбиений: " << endl;

cout << "Методом средних прямоугольников: " << center(a, b, n1) << endl;

cout << "Абсолютная погрешность: " << abs(center(a, b, n1) - (round((center(a, b, n1)) \* 10000) / 10000)) << endl;

cout << "Методом правых прямоугольников: " << right(a, b, n1) << endl;

cout << "Абсолютная погрешность: " << abs(right(a, b, n1) - (round((right(a, b, n1)) \* 10000) / 10000)) << endl;

cout << "Методом левых прямоугольников: " << left(a, b, n1) << endl;

cout << "Абсолютная погрешность: " << abs(left(a, b, n1) - (round((left(a, b, n1)) \* 10000) / 10000)) << endl;

cout << "Формулой Ньютона-Лейбница: " << newton\_leibniz(a, b) << endl;

cout << "Абсолютная погрешность: " << abs(newton\_leibniz(a, b) - (round((newton\_leibniz(a, b)) \* 10000) / 10000)) << endl;

cout << "Методом трапеций: " << trapecia(a, b, n1) << endl;

cout << "Абсолютная погрешность: " << abs(trapecia(a, b, n1) - (round((trapecia(a, b, n1)) \* 10000) / 10000)) << endl;

cout << "Методом трапеций: " << parables(a, b, n1) << endl;

cout << "Абсолютная погрешность: " << abs(parables(a, b, n1) - (round((parables(a, b, n1)) \* 10000) / 10000)) << endl;

cout << endl << " Для 100 разбиений: " << endl;

cout << "Методом средних прямоугольников: " << center(a, b, n2) << endl;

cout << "Абсолютная погрешность: " << abs(center(a, b, n2) - (round((center(a, b, n2)) \* 10000) / 10000)) << endl;

cout << "Методом правых прямоугольников: " << right(a, b, n) << endl;

cout << "Абсолютная погрешность: " << abs(right(a, b, n2) - (round((right(a, b, n2)) \* 10000) / 10000)) << endl;

cout << "Методом левых прямоугольников: " << left(a, b, n2) << endl;

cout << "Абсолютная погрешность: " << abs(left(a, b, n2) - (round((left(a, b, n2)) \* 10000) / 10000)) << endl;

cout << "Формулой Ньютона-Лейбница: " << newton\_leibniz(a, b) << endl;

cout << "Абсолютная погрешность: " << abs(newton\_leibniz(a, b) - (round((newton\_leibniz(a, b)) \* 10000) / 10000)) << endl;

cout << "Методом трапеций: " << trapecia(a, b, n2) << endl;

cout << "Абсолютная погрешность: " << abs(trapecia(a, b, n2) - (round((trapecia(a, b, n2)) \* 10000) / 10000)) << endl;

cout << "Методом трапеций: " << parables(a, b, n2) << endl;

cout << "Абсолютная погрешность: " << abs(parables(a, b, n2) - (round((parables(a, b, n2)) \* 10000) / 10000)) << endl;

}