|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |
| --- | --- |

ФАКУЛЬТЕТ Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

**ЛАБОРАТОРНАЯ РАБОТА № 2**

**ПО КУРСУ:**

***«Численные методы»***

Студент *Виленский С.Д.*

Преподаватель *Домрачева А.Б.*

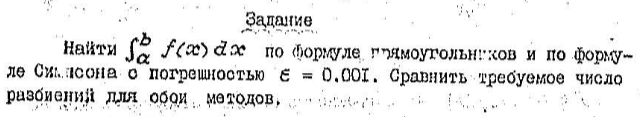
*Москва, 2024 г.*

**ОГЛАВЛЕНИЕ**

[1. Постановка задачи 3](#_heading=h.30j0zll)

[2. Практическая реализация 4](#_heading=h.1fob9te)

# 1. Постановка задачи



# 2. Практическая реализация

#include <cmath>

#include <numbers>

#include <iostream>

#include "../lab1/lib/Matrix.cpp"

// *Вариант 4. [0, π]. f(x) = 2x\*cos(x/2)*

#define EPSILON .001

auto f(double x) -> double {

return 2 \* x \* std::cos(x / 2);

}

auto f\_antiderivative(double x) -> double {

return 4 \* x \* std::sin(x / 2) + 8 \* std::cos(x / 2);

}

auto main() -> int {

double a = 0, b = std::numbers::pi;

double I = f\_antiderivative(b) - f\_antiderivative(a);

Matrix<double, 4, 3> dataTable{};

// *метод средних прямоугольников*

dataTable.at(0, 0) = 1;// *n*

dataTable.at(1, 0) = f((a + b) / 2) \* (b - a);// *I\_last*

do {

std::size\_t k = 2;

double n = dataTable.at(0, 0) + 1;

double h = (b - a) / n;

double I = 0;

for (std::size\_t i = 1; i != n + 1; ++i) {

I += f(a + h \* (i - .5));

}

I \*= h;

dataTable.at(0, 0) = n;

dataTable.at(2, 0) = (I - dataTable.at(1, 0)) / (std::pow(2, k) - 1);

dataTable.at(1, 0) = I;

} while (abs(dataTable.at(2, 0)) > EPSILON);

dataTable.at(3, 0) = dataTable.at(1, 0) + dataTable.at(2, 0);

// *метод трапеций*

dataTable.at(0, 1) = 1;// *n*

dataTable.at(1, 1) = ((f(a) + f(b)) / 2) \* (b - a);// *I\_last*

do {

std::size\_t k = 2;

double n = dataTable.at(0, 1) + 1;

double h = (b - a) / n;

double I = (f(a) + f(b)) / 2;

for (std::size\_t i = 1; i != n; ++i) {

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

}

I \*= h;

dataTable.at(0, 1) = n;

dataTable.at(2, 1) = (I - dataTable.at(1, 1)) / (std::pow(2, k) - 1);

dataTable.at(1, 1) = I;

} while (abs(dataTable.at(2, 1)) > EPSILON);

dataTable.at(3, 1) = dataTable.at(1, 1) + dataTable.at(2, 1);

// *метод Симпсона*

dataTable.at(0, 2) = 1;// *n*

dataTable.at(1, 2) = (f(a) + 4 \* f((a + b) / 2) + f(b))

\* (b - a) / 6;// *I\_last*

do {

std::size\_t k = 4;

double n = dataTable.at(0, 2) + 1;

double h = (b - a) / (2 \* n);

double I = f(a) - f(b);

for (std::size\_t i = 1; i != n + 1; ++i) {

I += 4 \* f(a + h \* (2 \* i - 1));

I += 2 \* f(a + h \* 2 \* i);

}

I = I \* h / 3;

dataTable.at(0, 2) = n;

dataTable.at(2, 2) = (I - dataTable.at(1, 2)) / (std::pow(2, k) - 1);

dataTable.at(1, 2) = I;

} while (abs(dataTable.at(2, 2)) > EPSILON);

dataTable.at(3, 2) = dataTable.at(1, 2) + dataTable.at(2, 2);

std::cout << "I = " << I << '\n';

std::cout << "\t\taverage\t\ttrapez\t\tSimpson\n";

std::cout << "n\t\t"

<< std::pow(2, dataTable.at(0, 0)) << "\t\t"

<< std::pow(2, dataTable.at(0, 1)) << "\t\t"

<< std::pow(2, dataTable.at(0, 2)) << '\n'

<< "I\*\_(h/2)\t"

<< dataTable.at(1, 0) << "\t\t"

<< dataTable.at(1, 1) << "\t\t"

<< dataTable.at(1, 2) << '\n'

<< "R\t\t"

<< dataTable.at(2, 0) << "\t"

<< dataTable.at(2, 1) << "\t"

<< dataTable.at(2, 2) << '\n'

<< "I\*\_(h/2)+R\t"

<< dataTable.at(3, 0) << "\t\t"

<< dataTable.at(3, 1) << "\t\t"

<< dataTable.at(3, 2) << '\n';

}

# 3. Результат работы программы

