Министерство образования и науки Российской Федерации

Санкт-Петербургский политехнический университет Петра Великого

—

Институт компьютерных наук и кибербезопастности

Высшая школа программной инженерии

**Лабораторная работа №1**

**по дисциплине «Вычислительная математика»**

Выполнил

Студент группы 5130904/20004 Машкин А.А.

Преподаватель Устинов С.М.

Оглавление

[Задание 2](#_Toc162000585)

[Результаты 3](#_Toc162000586)

[Вывод 5](#_Toc162000587)

[Код программы 6](#_Toc162000588)

[<DIR>/computational\_mathematics/lab\_1/Langrage.cpp 6](#_Toc162000589)

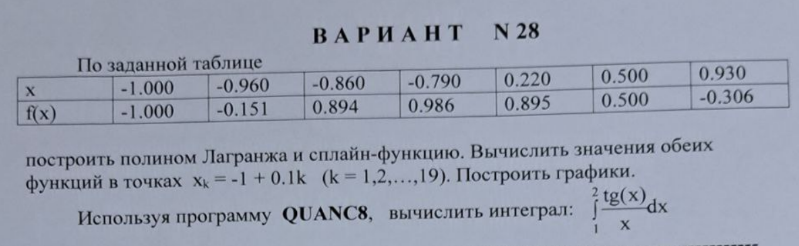
[<DIR>/computational\_mathematics/lab\_1/Langrage.h 7](#_Toc162000590)

[<DIR>/computational\_mathematics/lab\_1/main.cpp 7](#_Toc162000591)

[<DIR>/computational\_mathematics/lab\_1/function.cpp 8](#_Toc162000592)

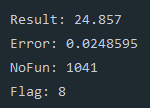
[<DIR>/computational\_mathematics/lab\_1/function.h 8](#_Toc162000593)

# Задание



# Результаты

При eps = 0.0001



# Вывод

В лабароторной работе на графике у меня видны разные результаты. Это связанно с двумя факторами. Первый - мне не известно какого вида фукнция, колебательная или какого-то иного вида. Второй - в нашей таблице есть большой промежуток, где нет известных значений (с -0,7 до 0, 2) и из-за этого вид обеих функций очень разный, если возможно получить значение на этом промежутке, то станет яснее, какой вид принимает истинная фукнция. Результат программы QUANC8 говорит нам о том, что в нашей подинтегральной функции деление промежутка пополам более 30 раз произошло 8 раз и погрешность в этом случае не установлена точно. Это может быть связано с разрывами подынтегральной функции или ее «зашумлением» вычислительной погрешностью.

# Код программы

## <DIR>/computational\_mathematics/lab\_1/Langrage.cpp

## #include "Lagrange.h"

## REAL Lagrange(REAL\* f, REAL\* x, REAL z, int size)

## {

## REAL var = 1.0;

## REAL result = 0.0;

## for (auto k = 0; k <= size; k++)

## {

## for (auto i = 0; i <= size; i++)

## {

## if (k == i)

## {

## continue;

## }

## var \*= (z - x[i]) / (x[k] - x[i]);

## }

## result += var \* f[k];

## var = 1.0;

## }

## return result;

## }

## <DIR>/computational\_mathematics/lab\_1/main.cpp

#include <iostream>

#include <functional>

#include <function.h>

#include <fstream>

#include <iomanip>

#include "Lagrange.h"

#include "FORSYTHE.H"

int main(int argc, char\*\* argv)

{

if (argc != 2)

{

std::cout << "Not enough argument\n";

return 1;

}

std::ofstream outFile;

outFile.exceptions(std::ofstream::badbit | std::ofstream::failbit);

try

{

outFile.open(argv[1]);

}

catch (const std::exception& ex)

{

std::cerr << ex.what() << "\n";

return 1;

}

double a, b, epsabs, epsrel, result, errest;

int nfe = 0;

double flag = 0.0;

REAL x[] = {-1.000, -0.960, -0.860, -0.790, 0.220, 0.500, 0.930};

REAL y[] = {-1.000, -0.151, 0.894, 0.986, 0.895, 0.500, -0.306};

auto size = sizeof(x) / sizeof(double);

SPLINE spline(7, x, y);

std::cout << " x | Lagrange | Spline\n";

for (auto i = -1.0; i < -1.0 + 0.1 \* 19; i += 0.1)

{

outFile << i << "," << Lagrange(y, x, i, size) << "," << spline.Eval(i) << "\n";

std::cout << std::setw(12) << i << " | ";

std::cout << std::setw(13) << Lagrange(y, x, i, size) << " | ";

std::cout << std::setw(12) << spline.Eval(i) << "\n";

}

outFile.close();

a = 1.0;

b = 2.0;

epsrel = 0.0001;

epsabs = 0.0;

QUANC8(f, a, b, epsabs, epsrel, result, errest, nfe, flag);

std::cout << "\n\n";

std::cout << "Result: " << result << "\n";

std::cout << "Error: " << errest << "\n";

std::cout << "NoFun: " << nfe << "\n";

std::cout << "Flag: " << flag << "\n";

}

## <DIR>/computational\_mathematics/lab\_1/function.cpp

#include "function.h"

#include <math.h>

double f(double x)

{

return std::tan(x) / x;

}

## <DIR>/computational\_mathematics/lab\_1/function.h

#ifndef LIBRARY\_FUNCTION\_H

#define LIBRARY\_FUNCTION\_H

#include <functional>

double f(double x);

#endif