Министерство образования и науки РФ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа программной инженерии

Отчёт по лабораторной работе №1

по дисциплине «Вычислительная математика»

Выполнил студент гр. 3530904/00001 Корж В.Д.

Руководитель Устинов С.М.

**Оглавление**

[Постановка задачи 3](#__RefHeading___Toc1147_75006022)

[Код программы 4](#__RefHeading___Toc1149_75006022)

[Результат работы программы 6](#__RefHeading___Toc1151_75006022)

[Графики 7](#__RefHeading___Toc1153_75006022)

[Вывод 9](#__RefHeading___Toc1155_75006022)

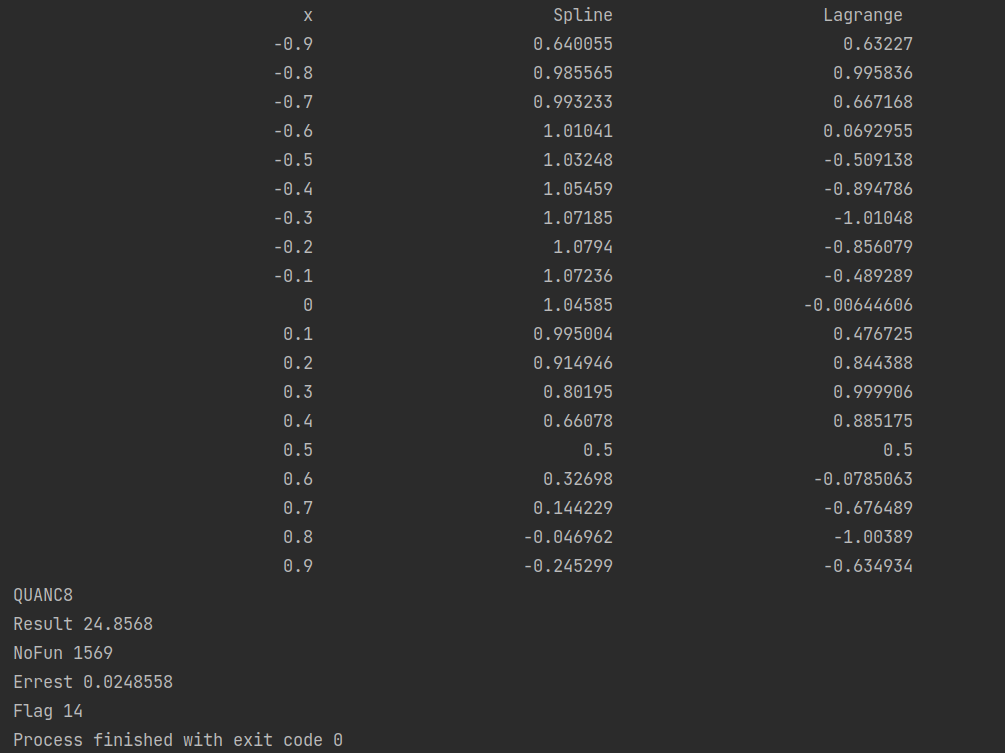
# Постановка задачи

# 

# Код программы

#include <iostream>  
#include <vector>  
#include <iomanip>  
#include <cmath>  
#include "FORSYTHE.H"  
  
struct Data {  
 std::vector <double> points = **{**-1.000, -0.960, -0.860, -0.790, 0.220, 0.500, 0.930**}**;  
 std::vector <double> values = **{**-1.000, -0.151, 0.894, 0.986, 0.895, 0.500, -0.306**}**;  
};  
  
double InterpolateLagrangePolynomial (double point, const std::vector<double>& xValues, const std::vector<double>& yValues, int n) {  
 double result = 0;  
  
 for (int i = 0; i < n; i++)  
 {  
 double temp = 1;  
 for (int j = 0; j < n; j++)  
 {  
 if (j != i)  
 {  
 temp \*= (point - xValues[j]) / (xValues[i] - xValues[j]);  
 }  
 }  
 result += temp \* yValues[i];  
 }  
  
 return result;  
}  
  
int main() {  
 Data object;  
 SPLINE spline = SPLINE(object.points.size() , object.points.data(), object.values.data());  
 const int POLYNOMIAL\_SIZE = object.points.size();  
  
 std::cout << std::setw(30) << "x" << std::setw(30) << "Spline" << std::setw(30) << "Lagrange\n";  
 for (int k = 1; k <= 19; ++k) {  
 double xk = -1 + 0.1 \* k;  
 double result = InterpolateLagrangePolynomial(xk, object.points, object.values, POLYNOMIAL\_SIZE);  
 std::cout << std::setw(30) << xk  
 << std::setw(30) << spline.Eval(xk) << std::setw(30) << result << '\n';  
 }  
  
 double result, errest, flag;  
 int nofun;  
 QUANC8([](double x) {  
 return tan(x) / x;  
 }, 1, 2, 0.0000001, 0.0000001, result, errest, nofun, flag);  
 std::cout << "QUANC8\n";  
 std::cout << "Result " << result << '\n'  
 << "NoFun " << nofun << '\n'  
 << "Errest " << errest << '\n'  
 << "Flag " << flag;  
}

# Результат работы программы

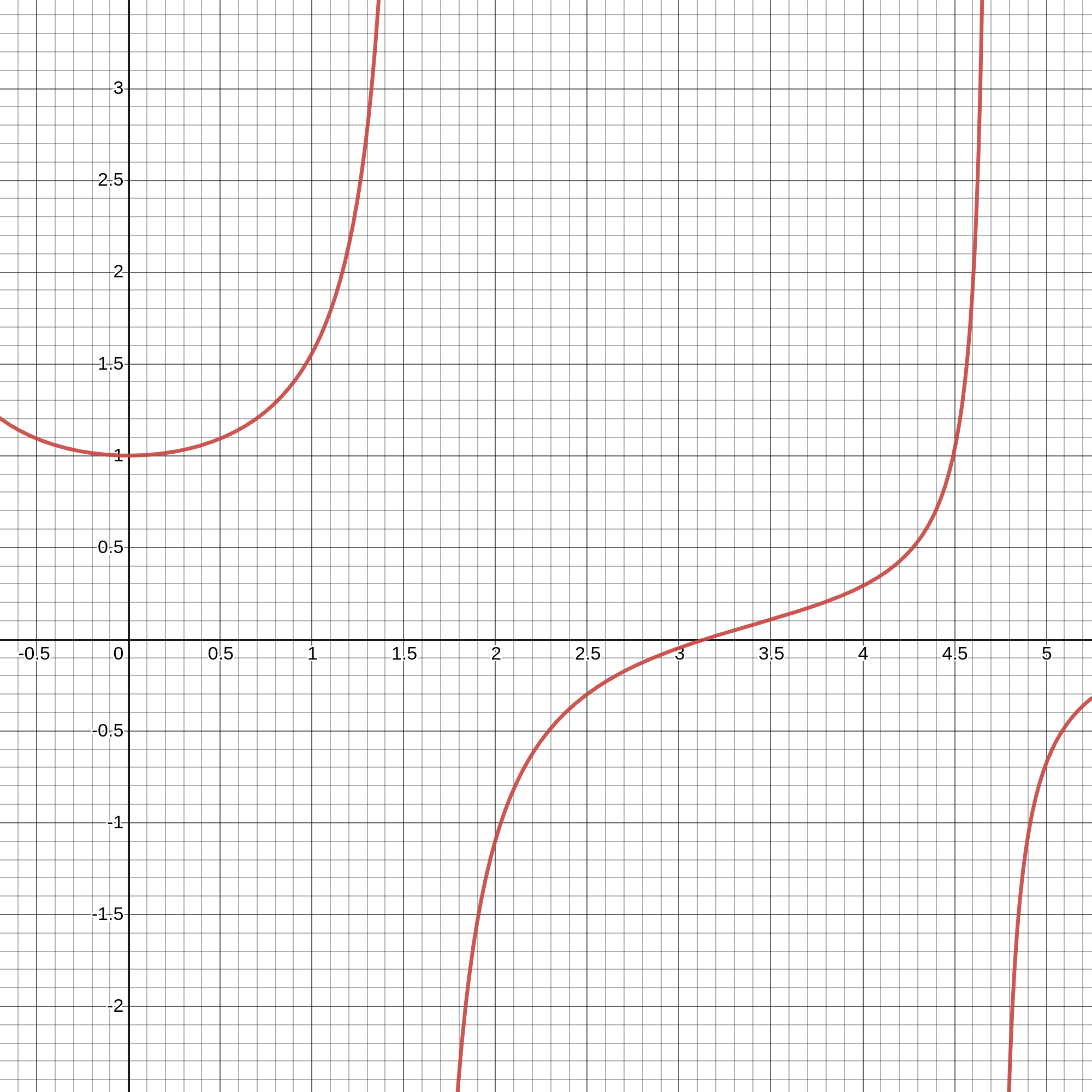
****

# Графики

*Интерполяционный полином Лагранжа*

*Spline*

*Подинтегральная функция (Quanc8)*

**

# Вывод

Таким образом, в ходе исполнения подпрограммы QUANC8, мы можем заметить, что подинтегральная функция является «неберущейся». Это видно в выходных параметрах, таких как: flag, errest. Flag сигнализирует нам о том, что на 14 интервалах не было сходимости, а errest говорит об оценке велечины действительной ошибки. Если построить график, можно заметить, что функция имеет разрывы в некоторых точках. Об этом и сведетельствует вывод работы подпрограммы.