Министерство образования и науки Российской Федерации

Санкт-Петербургский политехнический университет Петра Великого

—

Институт компьютерных наук и технологий

Высшая школа программной инженерии

**Курсовая работа**

**по дисциплине «Вычислительная математика»**

Выполнил

Студент группы 5130904/20004 Шелковников Д.С.

Преподаватель Устинов С.М.

Оглавление

[Задание 2](#_Toc160626282)

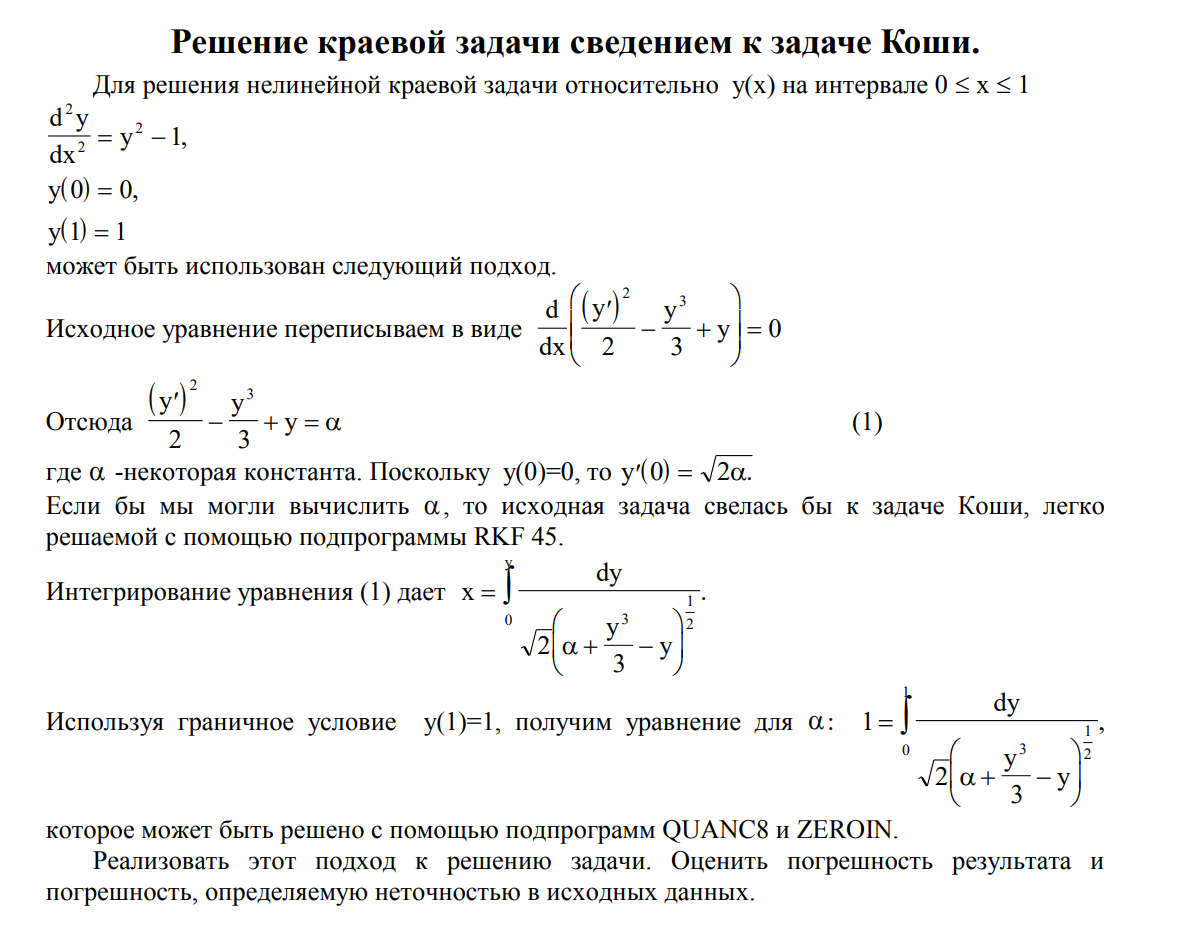
[Результаты 2](#_Toc160626283)

[Вывод 3](#_Toc160626284)

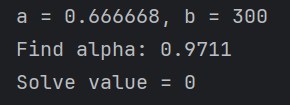
[Код программы 3](#_Toc160626285)

[<DIR>/computational\_mathematics/coursework/main.cpp 3](#_Toc160626286)

# Задание



# Результаты



# Вывод

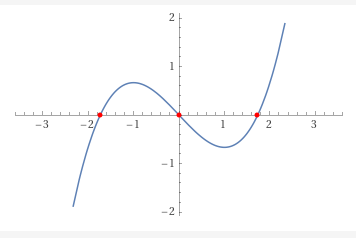
Определим сначала интервалы для поиска решения. 

График знаменателя (подкорневого выражения) имеет такой вид. По условию интегрирования нас интересует часть от 0 до 1. Решим неравенство, чтобы часть от 0 до 1 была неотрицательной. Это выполнится при добавлении . В таком случае знаменатель будет обращаться в 0, что нам не подходит. Тогда в коде программы укажем минимальную границу для поиска решения, как 2.0/3+0.000001, а верхнюю без ограничений. Дальше с помощью программы ZEROIN найдем решение.

Для решения дифференциального уравнения введем замену dy = x и получим систему из двух дифференциальных уравнений, где начальные параметры: 0 и , где α мы нашли ранее.

# Код программы

## <DIR>/computational\_mathematics/coursework/main.cpp

#include <iostream>

#include <cmath>

#include <functional>

#include "../common/Quanc8.h"

#include "zeroin.h"

double integrand(double y, double alpha) {

return 1.0 / sqrt(2 \* (alpha + pow(y, 3) / 3 - y));

}

double to\_solve(double alpha) {

double a = 0, b = 1;

double abserr = 1e-6, relerr = 1e-6;

auto func = std::bind(integrand, std::placeholders::\_1, alpha);

dimkashelk::Quanc8 quanc8(func, a, b, abserr, relerr);

return 1 - quanc8.getResult();

}

double find\_alpha(double a, double b) {

double tol = 1e-12;

int flag = 0;

double res = zeroin(a, b, to\_solve, tol, &flag);

return res;

}

int main() {

double a = 2.0/3+0.000001, b = 300;

double alpha = find\_alpha(a, b);

std::cout << "a = " << a << ", b = " << b << "\nFind alpha: " << alpha << "\nSolve value = " << to\_solve(alpha) << "\n\n";

return 0;

}

В программе также использовались файлы Quanc8.cpp и Quanc8.h из первой лабораторной работы и файл zeroin.h из стандартной библиотеки.