Министерство образования и науки РФ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа программной инженерии

Курсовая работа

по дисциплине «Вычислительная математика»

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

Руководитель Устинов С.М.

**Оглавление**

[Постановка задачи 3](#__RefHeading___Toc1147_75006022)

[Алгоритм работы 4](#__RefHeading___Toc1149_750060221)

[Код программы 5](#__RefHeading___Toc1149_75006022)

[Результат работы программы 7](#__RefHeading___Toc2736_1577260916)

[Проверка на чувствительность к возмущениям 8](#__RefHeading___Toc1153_750060221)

[Вывод 9](#__RefHeading___Toc1153_75006022)

[Список литературы 10](#__RefHeading___Toc2027_3411755071)

# **Постановка задачи**

# 

# **Алгоритм работы**

Прежде чем приступить к написаннию программы, мы могли заметить, что поставленную задачу лучше всего начинать решать с конца. В поcледнем уравнении мы можем увидеть в знаменателе коэффицент alpha, который потребуется для решения всей задачи. Мы находим его с помощью программ ZEROIN и QUANC8, последовательным вложением одной в другую. Необходимость в QUANC8 возникает из за интеграла в представленном уравнении. После нахождения alpha, мы будем использовать RKF45 — это подпрограмма для решения нежестких и слабо жестких дифферинциальных уравнении. Подставив наше уравнение, интервал от 0 до 1, шаг в 0.1 и остальные праметры получаем ответ.

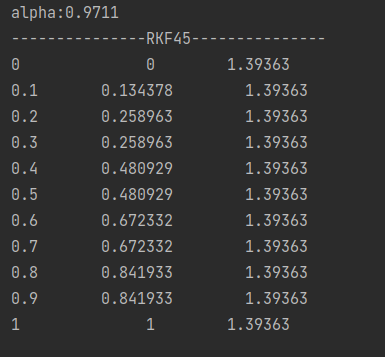
# **Код программы**

#include <iostream>  
#include <cmath>  
#include <iomanip>  
#include "FORSYTHE.H"  
  
double alpha = 0.0;  
  
void firstFunction(double t, double\* y, double\* dX);  
double secondFunction(double value);  
double auxiliaryFunction(double y);  
  
int main() {  
 double ax = 0.7;  
 double bx = 1.5;  
 alpha = ZEROIN(ax, bx, secondFunction, 0.0000001);  
 std::cout << "alpha:" << alpha << '\n';  
  
 double t = 0.0;  
 double tOut = 0.0;  
 double h = 0.1;  
 double X[2] = {0.0,sqrt(2 \* alpha)};  
 double work[3 + 6 \* 1];  
 double relerr = 0.0000001;  
 double abserr = 0.0000001;  
 int flag = 1;  
  
 std::cout << "---------------RKF45---------------\n";  
 for (int i = 0; i <= 1 / h; i++)  
 {  
 RKF45(firstFunction, 1, X, t, tOut, relerr, abserr, work, flag);  
 std::cout << tOut << std::setw(15) << X[0] << std::setw(15) << X[1] << '\n';  
 tOut += h;  
 }  
}  
  
void firstFunction(double t, double\* y, double\* dX) {  
 dX[0] = sqrt(2 \* ((alpha + pow(y[0], 3) / 3 - y[0])));  
}  
  
double secondFunction(double value) {  
 alpha = value;  
 double result, errest, flag;  
 int nofun;  
 QUANC8(auxiliaryFunction, 0.0, 1.0, 0.0000001, 0.0000001, result, errest, nofun, flag);  
 return result - 1;  
}  
  
double auxiliaryFunction(double y) {  
 return 1 / sqrt(2 \* (alpha + pow(y, 3) / 3 - y));  
}

# **Результат работы программы**

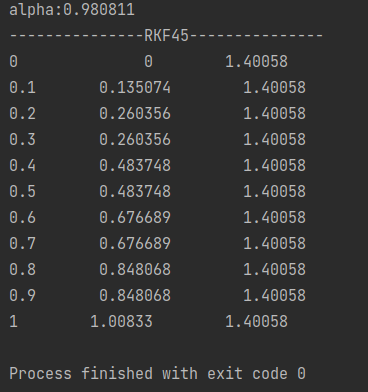
***Alpha: 0.9711***

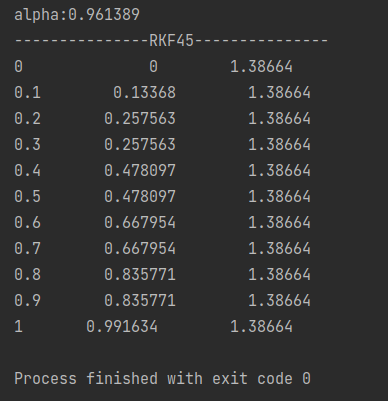
|  |  |
| --- | --- |
| 0 | 0 |
| 0.1 | 0.134378 |
| 0.2 | 0.258963 |
| 0.3 | 0.258963 |
| 0.4 | 0.480929 |
| 0.5 | 0.480929 |
| 0.6 | 0.672332 |
| 0.7 | 0.672332 |
| 0.8 | 0.841933 |
| 0.9 | 0.841933 |
| 1 | 1 |

****

# **Проверка на чувствительность к возмущениям**

Изменим alpha, тем самым показывая, что решение изменится на такой же процент как и коэффицент. Решение изменилось на такой же процент, следовательно оно устойчиво.

Рисунок 1: alpha \* 1.01

Рисунок 2: alpha \* 0.99

# **Вывод**

Проделав данную курсовую работу можно сделать вывод о том, что полученное решение полностью совпадает с желаемым результатом. Используя подпрограммы: ZEROIN, QUANC8, RKF45. Нам всё же удалось решить поставленную задачу и даже проверить его на устойчивость.

# **Список литературы**

1. С.М.Устинов, В.А.Зимницкий. Вычислительная математика. – СПб.: БХВ-Петербург, 2009. – 336с. – (Учебное пособие.)

2. Форсайт Дж., Малькольм М., Моулер К. Машинные методы математических вычислений. – М.: Мир, 1980. – 280с.