Министерство образования и науки Российской Федерации

Санкт-Петербургский Государственный Политехнический Университет

—

Институт компьютерных наук и технологий

Высшая школа программной инженерии

Курсовая Работа

Вариант №24

по дисциплине «Вычислительная математика»

Выполнил:

студент гр. 3530904/10001 Боков Я. А.

Руководитель:

профессор С.М. Устинов

Санкт-Петербург

2023

**Задание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Ход работы**

Сначала рассчитываются парамерты E, A, B, F, G с помощью подпрограмм QUANC8 и DECOMP + SOLVE соответственно. Первая помогает рассчитать значение интеграла на промежутке, вторые решить систему уравнений. Мы формируем вектор Z = (dT/dx, T) = (z1, z2). Получаем систему уравнений:

Изображение выглядит как текст, письмо

Автоматически созданное описание

Ведём пристрелку по a с помощью подпрограммы ZEROIN, внутри которой вызывается RKF45 для решения системы дифференциальных уравнений. После нахождения подходящего значения z1(0) решаем систему уравнений снова через RKF45.

**Листинг программы**

#include "Forsythe.h"

#include <iostream>

#include <cmath>

////#include "decomp.cpp"

using namespace std;

static double E = 0.0;

static double epsrel = 1.0e-6;

static double epsabs = 1.0e-18;

const double L = 1;

double A, B, F, G;

double funE(double x)

{

return sqrt(1 + pow(x, 5));

}

void funcRKF(double x, double Z[], double dzdx[])

{

dzdx[0] = (x / E) \* (Z[1] - Z[0]);

dzdx[1] = Z[0] + 0 \* Z[1];

}

double F\_forZeroin(double param)

{

rkf base;

base.f = funcRKF;

const int neqn = 2;

//написано в документации

unsigned char work[6 \* (neqn \* sizeof(Float)) + sizeof(struct rkf\_inside)];

base.neqn = neqn;

base.work = work;

base.ae = epsabs;

base.re = epsrel;

base.flag = 1;

double Z[] = { param , A };

base.Y = Z;

base.t = 0;

base.tout = L;

rkf45(&base);

//cout << base.flag << "\n";

return Z[1] - B;

}

int main()

{

double end1 = 0;

double end2 = 0.5;

double errest;

int nofun;

double flag;

double integral = Quanc8(funE, end1, end2, epsabs, epsrel, &errest, &nofun, &flag);

E = 1.994827 \* integral;

double Brr[] = {23, 32, 36, 36};

double Arr[] = { 5, 7, 6, 5,

7, 10, 8, 7,

6, 8, 10, 9,

5, 7, 9, 10 };

int\* pivot = new int[4];

double cond;

Decomp(4, Arr, &cond, pivot);

Solve(4, Arr, Brr, pivot);

A = Brr[0];

B = Brr[1];

F = Brr[2];

G = Brr[3];

auto T0 = Zeroin(F\_forZeroin, F, G, epsabs);

cout << "Zeroin result: " << T0 << "\n\n";

rkf base;

base.f = funcRKF;

const int neqn = 2;

//написано в документации

unsigned char work[6 \* (neqn \* sizeof(Float)) + sizeof(struct rkf\_inside)];

base.neqn = neqn;

base.work = work;

base.ae = epsabs;

base.re = epsrel;

base.flag = 1;

double Z[] = { T0, 0 };

base.Y = Z;

base.t = 0;

double h\_print = 0.05;

base.tout = 0;

rkf45(&base);

cout << "A: " << A << "\nB: " << B << "\nF: " << F << "\nG: " << G << "\n\n";

cout << "x" << " \t" << "T" << " \t" << "flag\n";

cout << base.t << " \t" << Z[1]

<< " \t" << base.flag << endl;

for (double h = h\_print; h <= L; h += h\_print)

{

base.tout = h;

rkf45(&base);

cout << base.t << " \t" << Z[1]

<< " \t" << base.flag << endl;

}

base.tout = 1;

rkf45(&base);

cout << base.t << " \t" << Z[1]

<< " \t" << base.flag << endl;

}

**Результат работы программы:**

Zeroin result: 1.07999

A: -1.37033e-14

B: 1

F: 1

G: 2

x T flag

0 0 2

0.05 0.0539774 2

0.1 0.107828 2

0.15 0.161438 2

0.2 0.214709 2

0.25 0.267558 2

0.3 0.319917 2

0.35 0.371737 2

0.4 0.422981 2

0.45 0.473631 2

0.5 0.523681 2

0.55 0.573144 2

0.6 0.622044 2

0.65 0.670418 2

0.7 0.718319 2

0.75 0.765809 2

0.8 0.812962 2

0.85 0.859861 2

0.9 0.906599 2

0.95 0.953276 2

1 1 2

**График T(x)**

**Изображение выглядит как диаграмма

Автоматически созданное описание**

**Вывод**

Решая систему из задания 24В аналитически, получаем A=0. Тем не менее, при подстановке вместо A 0 во всех местах программы выходные даннве нисколько не меняются. Очевидно, что при указанных погрешностях, отклонения выходных данных не сильно влияют на конечный результат.