Министерство образования и науки Российской Федерации

Санкт-Петербургский Государственный Политехнический Университет

—

ЛАБОРАТОРНАЯ РАБОТА № 3

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**

Ход работы соответствует описанному в задании.

**Листинг программы**

#include <iostream>

#include <fstream>

#include "cmath.h"

#include "Forsythe.h"

#include <cmath>

#include <iomanip>

using namespace std;

void func(double x, double y[], double dydx[])

{

dydx[0] = (-14) \* y[0] + 13 \* y[1] + cos(1 + x);

dydx[1] = 20 \* y[0] - 30 \* y[1] + atan(1 + pow(x, 2));

}

void runge\_kutta3(void(\*f)(double x, double y[], double dydx[]),

double t, double out, double Zn[], double h)

{

double k1[2], k2[2], k3[2];

double func[2], temp[2];

double tn;

for (double t = 0.0; t <= out; t += h)

{

tn = t;

temp[0] = Zn[0];

temp[1] = Zn[1];

f(tn, temp, func);

k1[0] = h \* func[0];

k1[1] = h \* func[1];

tn = t + h / 2;

temp[0] = Zn[0] + k1[0] / 2;

temp[1] = Zn[1] + k1[1] / 2;

f(tn, temp, func);

k2[0] = h \* func[0];

k2[1] = h \* func[1];

tn = t + ((3 \* h) / 4);

temp[0] = Zn[0] + ((3 \* k2[0]) / 4);

temp[1] = Zn[1] + ((3 \* k2[1]) / 4);

f(tn, temp, func);

k3[0] = h \* func[0];

k3[1] = h \* func[1];

Zn[0] = Zn[0] + (2 \* k1[0] + 3 \* k2[0] + 4 \* k3[0]) / 9;

Zn[1] = Zn[1] + (2 \* k1[1] + 3 \* k2[1] + 4 \* k3[1]) / 9;

}

}

int main()

{

ofstream fout;

fout.open("output.txt");

double X[2] = {2, 0.5};

double EPS = 0.0001;

const double h\_print = 0.075;

const double t\_start = 0;

const double t\_end = 1.5;

const int neqn = 2;

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

rkf argument;

argument.f = func;

argument.neqn = neqn;

argument.re = EPS;

argument.ae = EPS;

argument.work = work;

argument.flag = 1;

argument.Y = X;

argument.t = t\_start;

fout << "t \tx[0] \tx[1]\n\n";

cout << "-----RKF45-----\n";

cout << "t \tx[0] \tx[1] \tflag\n\n";

for (double h = h\_print; h <= t\_end; h += 0.075)

{

argument.tout = h;

rkf45(&argument);

fout << argument.t << " \t" << X[0]

<< " \t" << X[1] << "\n";

cout << argument.t << " \t" << X[0]

<< " \t" << X[1]

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

}

cout << "\n-----RUNGE-KUTTA-3-----\n";

double h\_1 = h\_print;

fout << "\nt \tx[0] \tx[1]\n\n";

cout << "\n1) h= " << h\_1 << "\n\n";

cout << "t \tx[0] \tx[1]\n\n";

for (double tMid = h\_1; tMid <= t\_end; tMid += h\_1)

{

X[0] = 2;

X[1] = 0.5;

runge\_kutta3(func, t\_start, tMid, X, h\_1);

fout << tMid << " \t" << X[0] << " \t" << X[1] << "\n";

cout << tMid << " \t" << X[0] << " \t" << X[1] << endl;

}

double h\_2 = h\_print / 10.0;

int k = 0;

fout << "\nt \tx[0] \tx[1]\n\n";

cout << "\n2) h= " << h\_2 << "\n\n";

cout << "t \tx[0] \tx[1]\n\n";

for (double tMid = h\_2; tMid < t\_end + h\_2; tMid += h\_2)

{

k++;

X[0] = 2;

X[1] = 0.5;

runge\_kutta3(func, t\_start, tMid, X, h\_2);

if (k == 10)

{

fout << tMid << " \t" << X[0] << " \t" << X[1] << "\n";

cout << tMid << " \t" << X[0] << " \t" << X[1] << endl;

k=0;

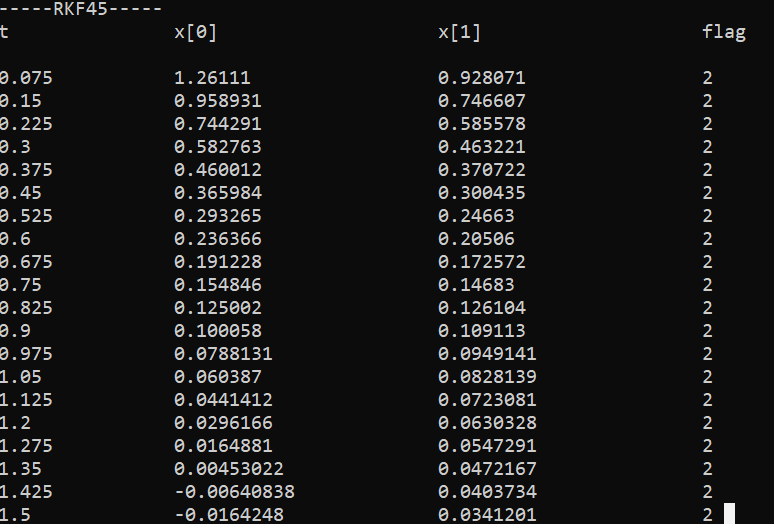
}

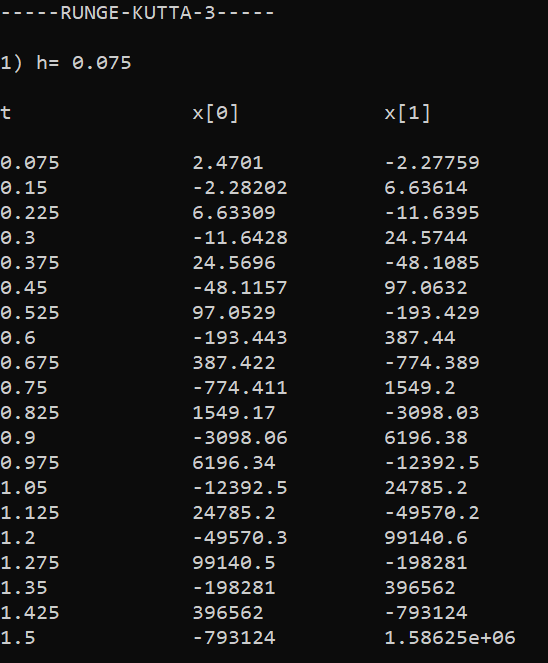
}

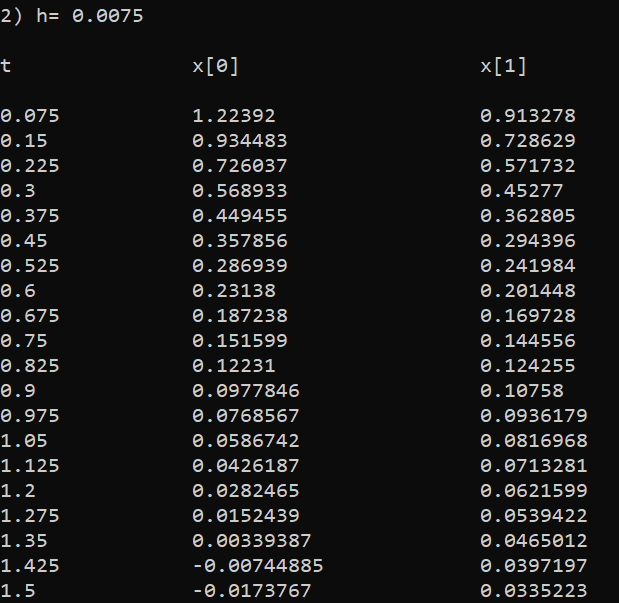
fout.close();

}

**Результат работы программы:**



****



**Вывод**

Из результатов работы программы видно, что один и тот же шаг интегрирования для разных методов даёт колоссальные различия. Это связано с тем, что шаг h = 0.075 для метода Рунге-Кутты 3 степени точности не удовлетворяет условию устойчивости. Чтобы получить качественно верное решение с помощью метода 3 степени точности, потребуется более маленький шаг, а следовательно, и большее количество вычислений.

Пренебрежение к условию устойчивости приводит к качественно неверному решению задачи.