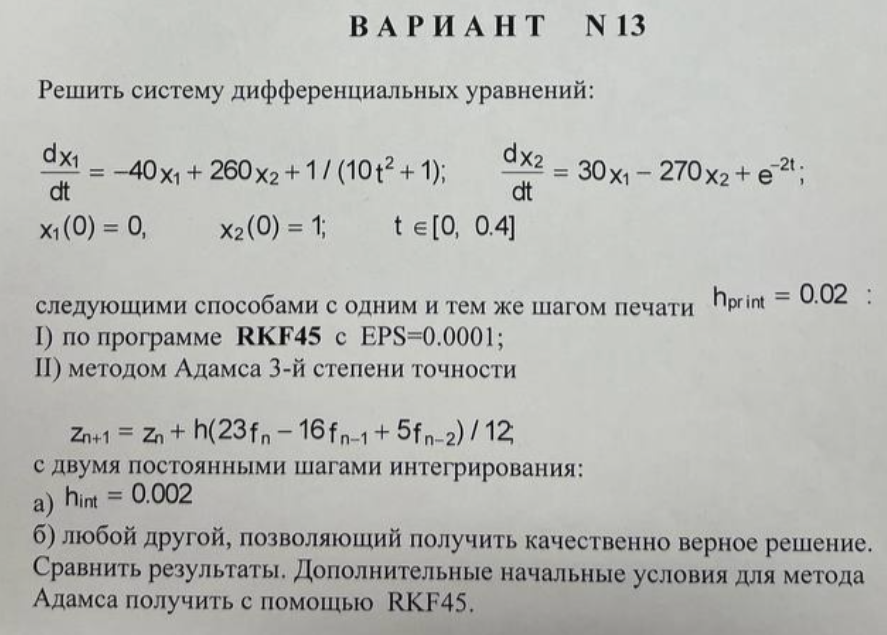
**Лабораторная работа №3**

по дисциплине «Вычислительная математика»

Выполнил

Студент группы 5130904/20001 Набережнов Д.А.

Преподаватель Устинов С.М.

1. **Задание**
2. **Текст программы**

#include <iostream>

#include "Forsythe.h"

#include <iomanip>

void function(Float t, Float\* X, Float\* dX) {

dX[0] = -40 \* X[0] + 260 \* X[1] + 1 / (10 \* t \* t + 1);

dX[1] = 30 \* X[0] - 270 \* X[1] + std::exp(-2 \* t);

}

void adams(Float\* X0, Float\* X1, Float\* X2, Float h, Float t) {

Float\* dX0 = new Float[2];

Float\* dX1 = new Float[2];

Float\* dX2 = new Float[2];

function(t, X0, dX0);

function(t, X1, dX1);

function(t, X2, dX2);

X0[0] = X1[0];

X0[1] = X1[1];

X1[0] = X2[0];

X1[1] = X2[1];

X2[0] = X2[0] + h \* (23 \* dX2[0] - 16 \* dX1[0] + 5 \* dX0[0]) / 12;

X2[1] = X2[1] + h \* (23 \* dX2[1] - 16 \* dX1[1] + 5 \* dX0[1]) / 12;

delete[] dX0;

delete[] dX1;

delete[] dX2;

}

int main() {

Float\* X0 = new Float[2];

X0[0] = 0.0;

X0[1] = 1.0;

Float\* X1 = new Float[2];

Float\* X2 = new Float[2];

Float\* X = new Float[2];

X[0] = 0.0;

X[1] = 1.0;

unsigned char\* work = new unsigned char[6 \* 2 \* sizeof(Float) + sizeof(struct rkf\_inside)];

rkf parameters;

parameters.f = function;

parameters.neqn = 2;

parameters.Y = X;

parameters.t = 0.0;

parameters.tout = 0.0;

parameters.re = 0.0001;

parameters.ae = 0.0001;

parameters.work = work;

parameters.flag = 1;

std::cout << "RKF45\n";

for (Float i = 0.0; i < 0.41; i += 0.02) {

parameters.tout = i;

rkf45(&parameters);

std::cout << std::fixed << std::setprecision(2) << "t = " << i << " | " << std::setprecision(6) << X[0] << " " << X[1] << "\n";

parameters.flag = 1;

}

std::cout << "Adams 3, h = 0.002\n";

X[0] = 0.0;

X[1] = 1.0;

std::cout << std::fixed << std::setprecision(2) << "t = " << 0.0 << " | " << std::setprecision(6) << X[0] << " " << X[1] << "\n";

parameters.t = 0.0;

parameters.tout = 0.002;

rkf45(&parameters);

X1[0] = X[0];

X1[1] = X[1];

parameters.flag = 1;

parameters.tout = 0.004;

rkf45(&parameters);

X2[0] = X[0];

X2[1] = X[1];

for (Float t = 0.006; t < 0.401; t += 0.002) {

adams(X0, X1, X2, 0.002, t);

if ((int)(t \* 1000) % 10 == 0 && (((int)(t \* 100)) % 2 == 0)) {

std::cout << std::fixed << std::setprecision(2) << "t = " << t << " | " << std::setprecision(6) << X2[0] << " " << X2[1] << "\n";

}

}

std::cout << "Adams 3, h = 0.001\n";

X0[0] = 0.0;

X0[1] = 1.0;

X[0] = 0.0;

X[1] = 1.0;

std::cout << std::fixed << std::setprecision(2) << "t = " << 0.0 << " | " << std::setprecision(6) << X[0] << " " << X[1] << "\n";

parameters.t = 0.0;

parameters.tout = 0.002;

rkf45(&parameters);

X1[0] = X[0];

X1[1] = X[1];

parameters.flag = 1;

parameters.tout = 0.004;

rkf45(&parameters);

X2[0] = X[0];

X2[1] = X[1];

for (Float t = 0.006; t < 0.401; t += 0.001) {

adams(X0, X1, X2, 0.001, t);

if ((int)(t \* 1000) % 10 == 0 && (((int)(t \* 100)) % 2 == 0)) {

std::cout << std::fixed << std::setprecision(2) << "t = " << t << " | " << std::setprecision(6) << X2[0] << " " << X2[1] << "\n";

}

}

delete[] X;

delete[] X0;

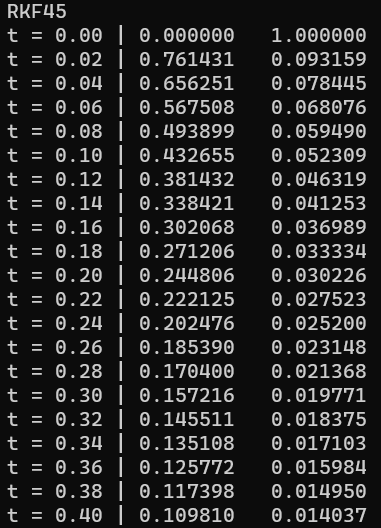
delete[] X1;

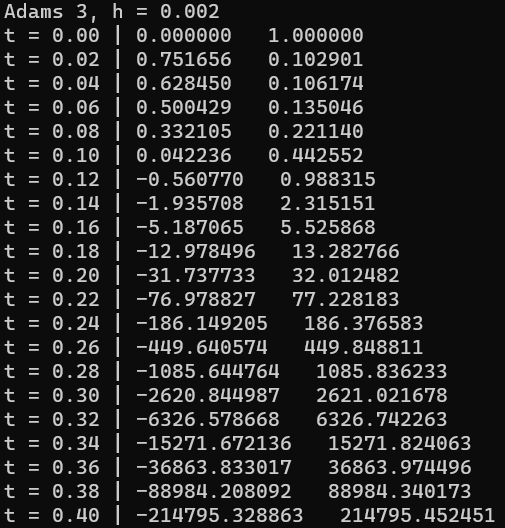
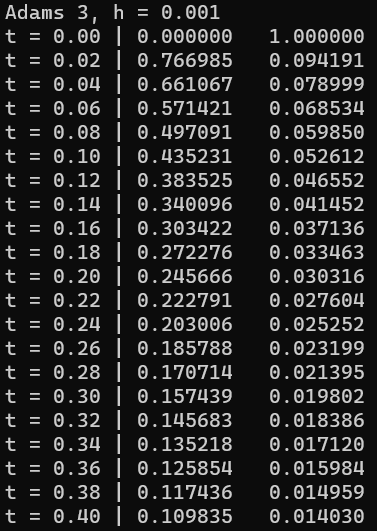
delete[] X2;

delete[] work;

return 0;

}

1. **Результат**

****Расчёт критического по устойчивости шага для явного метода Адамса 3 степени точности:

Формула: h|| < 6/11 (1)

A =

Вычислив собственные значения матрицы A получим: = -300; = -10

Подставив максимальное по модулю собственное значение матрицы A в формулу (1) получим: h < 0.0018

1. **Вывод**

Таким образом, нарушение ограничения на шаг всего на 0.0002 привело к неустойчивости решения и “взрыву погрешности”. При шаге 0.001, удовлетворяющем условию (1), получили вполне приемлемый результат.