Министерство образования и науки Российской Федерации

Санкт-Петербургский политехнический университет Петра Великого

—

Институт компьютерных наук и технологий

Высшая школа программной инженерии

**Лабораторная работа №3**

**по дисциплине «Вычислительная математика»**

Выполнил

Студент группы 5130904/20004 Шелковников Д.С.

Преподаватель Устинов С.М.

Оглавление

[Задание 2](#_Toc160530783)

[Результаты 3](#_Toc160530784)

[Выводы 5](#_Toc160530785)

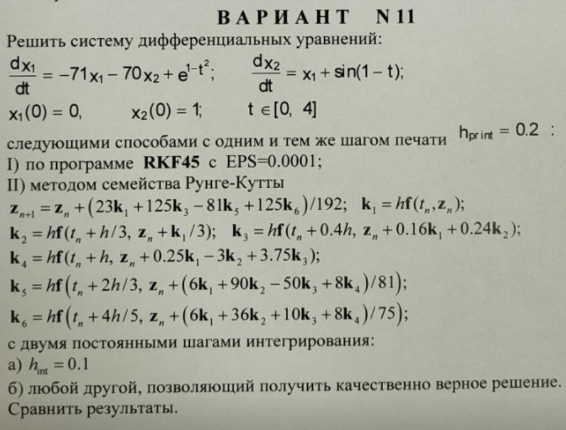
[Код программы 6](#_Toc160530786)

[<DIR>/computational\_mathematics/third\_lab/Rkf45.cpp 6](#_Toc160530787)

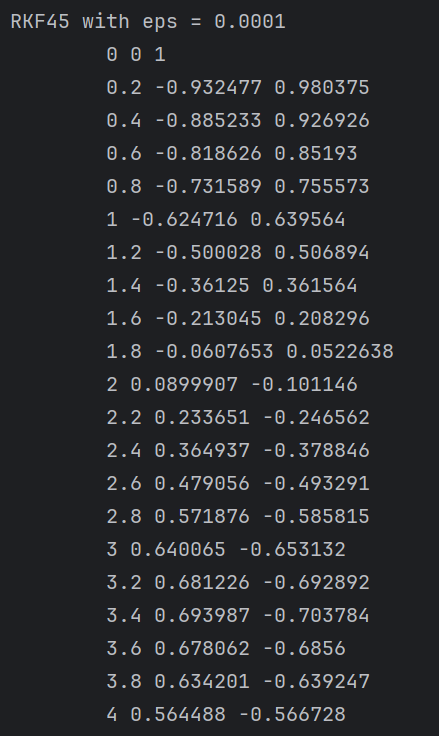
[<DIR>/computational\_mathematics/third\_lab/Rkf45.h 6](#_Toc160530788)

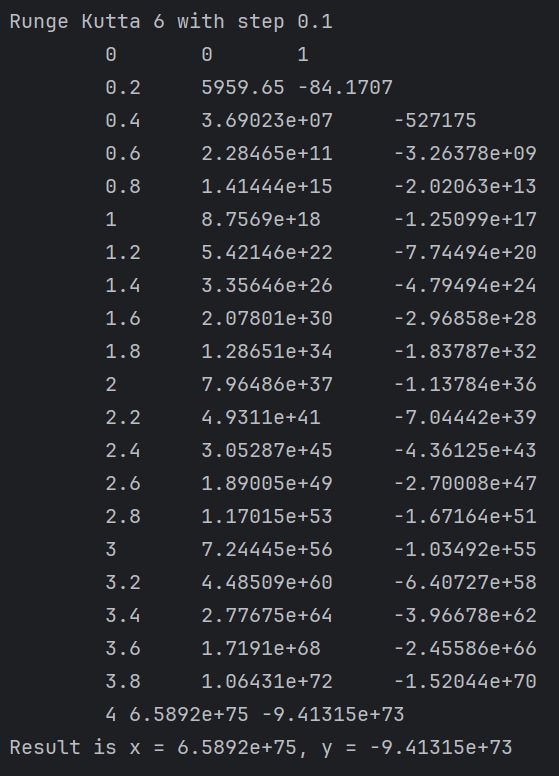
[<DIR>/computational\_mathematics/third\_lab/main.cpp 7](#_Toc160530789)

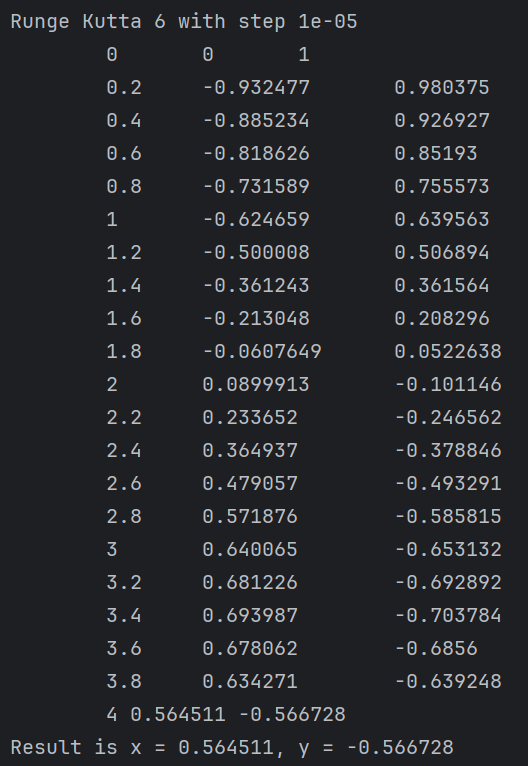
# Задание



# Результаты







# Выводы

Решения двух систем максимально схожи для rkf45 и Рунге-Кутты 6 порядка с шагом 1е-05. Можно говорить о трех знаках точности после запятой. Уменьшение шага в Рунге-Кутты ведет к увеличению времени вычисления, в то время, как 1е-05 вычисляется быстро и достаточно точно. Шаг 0.1 дает экспонентный рост результата из-за нестабильности системы, накопления численных ошибок при вычислении и коэффициентов при вычислении параметров метода. Есть два варианта решения проблема: другой шаг или адаптивная формула Рунге-Кутты 6 порядка.

# Код программы

## <DIR>/computational\_mathematics/third\_lab/Rkf45.cpp

#include "Rkf45.h"

#include "rkf.h"

#include <cmath>

#include <iostream>

void dimkashelk::Rkf45::calculate(int (\*F)(int n, double t, double y[], double yp[]),

double valueArray[],

double t, double tout) {

int flag = 0;

int NEQN = 2;

rkfinit(NEQN, &flag);

double dop[2]{0.0, 0.0};

flag = 1;

double ABS = 0.0001, REL = 0.0001;

double H[NEQN];

double STEP = 0.2;

int NFE = 0;

int MAXNFE = 100000;

while (t <= tout) {

std::cout << "\t" << t << " " << valueArray[0] << " " << valueArray[1] << "\n";

rkf45(F, NEQN, valueArray, dop, &t, t + STEP, &REL, ABS, H, &NFE, MAXNFE, &flag);

}

std::cout << "\t" << t << " " << valueArray[0] << " " << valueArray[1] << "\n";

rkfend();

}

## <DIR>/computational\_mathematics/third\_lab/Rkf45.h

#ifndef RKF45\_H

#define RKF45\_H

namespace dimkashelk {

class Rkf45 {

public:

static void calculate(int (\*F)(int n, double t, double y[], double yp[]),

double Y[],

double T,

double TOUT);

};

}

#endif

## <DIR>/computational\_mathematics/third\_lab/main.cpp

#include <iostream>

#include <iomanip>

#include <cmath>

#include <functional>

#include "Rkf45.h"

int func(int n, double t, double \*value, double \*res) {

res[0] = -71 \* value[0] - 70 \* value[1] + std::exp(1 - t \* t);

res[1] = value[0] + std::sin(1 - t);

return 0;

}

bool areEqualRel(float a, float b, float epsilon) {

return (fabs(a - b) <= epsilon \* std::max(fabs(a), fabs(b)));

}

using ODEFunction = std::function<double(double, double, double)>;

void rungeKutta6(ODEFunction f1, ODEFunction f2, double &y1, double &y2, double t0, double tEnd, double h) {

double k1 = 0.0, k2 = 0.0, k3 = 0.0, k4 = 0.0, k5 = 0.0, k6 = 0.0;

double l1 = 0.0, l2 = 0.0, l3 = 0.0, l4 = 0.0, l5 = 0.0, l6 = 0.0;

for (double t = t0, s = t0; t < tEnd; t += h) {

if (areEqualRel(s, t, 0.000001)) {

std::cout << "\t" << t << "\t" << y1 << "\t" << y2 << "\n";

s += 0.2;

}

k1 = h \* f1(t, y1, y2);

l1 = h \* f2(t, y1, y2);

k2 = h \* f1(t + h / 3, y1 + k1 / 3, y2 + l1 / 3);

l2 = h \* f2(t + h / 3, y1 + k1 / 3, y2 + l1 / 3);

k3 = h \* f1(t + 0.4 \* h, y1 + 0.16 \* k1 + 0.24 \* k2, y2 + 0.16 \* l1 + 0.24 \* l2);

l3 = h \* f2(t + 0.4 \* h, y1 + 0.16 \* k1 + 0.24 \* k2, y2 + 0.16 \* l1 + 0.24 \* l2);

k4 = h \* f1(t + h, y1 + 0.25 \* k1 - 3 \* k2 + 3.75 \* k3, y2 + 0.25 \* l1 - 3 \* l2 + 3.75 \* l3);

l4 = h \* f2(t + h, y1 + 0.25 \* k1 - 3 \* k2 + 3.75 \* k3, y2 + 0.25 \* l1 - 3 \* l2 + 3.75 \* l3);

k5 = h \* f1(t + 2 \* h / 3, y1 + (6 \* k1 + 90 \* k2 - 50 \* k3 + 8 \* k4) / 81,

y2 + (6 \* l1 + 90 \* l2 - 50 \* l3 + 8 \* l4) / 81);

l5 = h \* f2(t + 2 \* h / 3, y1 + (6 \* k1 + 90 \* k2 - 50 \* k3 + 8 \* k4) / 81,

y2 + (6 \* l1 + 90 \* l2 - 50 \* l3 + 8 \* l4) / 81);

k6 = h \* f1(t + 4 \* h / 5, y1 + (6 \* k1 + 36 \* k2 + 10 \* k3 + 8 \* k4) / 75,

y2 + (6 \* l1 + 36 \* l2 + 10 \* l3 + 8 \* l4) / 75);

l6 = h \* f2(t + 4 \* h / 5, y1 + (6 \* k1 + 36 \* k2 + 10 \* k3 + 8 \* k4) / 75,

y2 + (6 \* l1 + 36 \* l2 + 10 \* l3 + 8 \* l4) / 75);

y1 += (23 \* k1 + 125 \* k3 - 81 \* k5 + 125 \* k6) / 192;

y2 += (23 \* l1 + 125 \* l3 - 81 \* l5 + 125 \* l6) / 192;

}

std::cout << "\t" << tEnd << " " << y1 << " " << y2 << "\n";

}

int main() {

double data[2]{0.0, 1.0};

std::cout << "RKF45 with eps = 0.0001\n";

dimkashelk::Rkf45::calculate(func, data, 0, 4);

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

double y1 = 0.0, y2 = 1.0;

double t0 = 0.0, tEnd = 4.0, h = 0.1;

ODEFunction f1 = [](double t, double x, double y) {

return -71 \* x - 70 \* y + exp(1 - t \* t);

};

ODEFunction f2 = [](double t, double x, double y) {

return x + sin(1 - t);

};

std::cout << "Runge Kutta 6 with step " << h << "\n";

rungeKutta6(f1, f2, y1, y2, t0, tEnd, h);

std::cout << "Result is x = " << y1 << ", y = " << y2 << "\n\n\n";

y1 = 0.0, y2 = 1.0;

h = 0.00001;

std::cout << "Runge Kutta 6 with step " << h << "\n";

rungeKutta6(f1, f2, y1, y2, t0, tEnd, h);

std::cout << "Result is x = " << y1 << ", y = " << y2 << "\n\n\n";

return 0;

}