Изображение выглядит как текст, Шрифт, логотип, Графика

Контент, сгенерированный ИИ, может содержать ошибки.

**Отчет по лабораторной работе №3**

**по дисциплине “Вычислительная математика” Вариант №4**

Выполнил студент гр. 5130904/30007 Голиков В.С.

Руководитель Скуднева Е.В.

**Оглавление**

[Формулировка задания 3](#_Toc192189829)

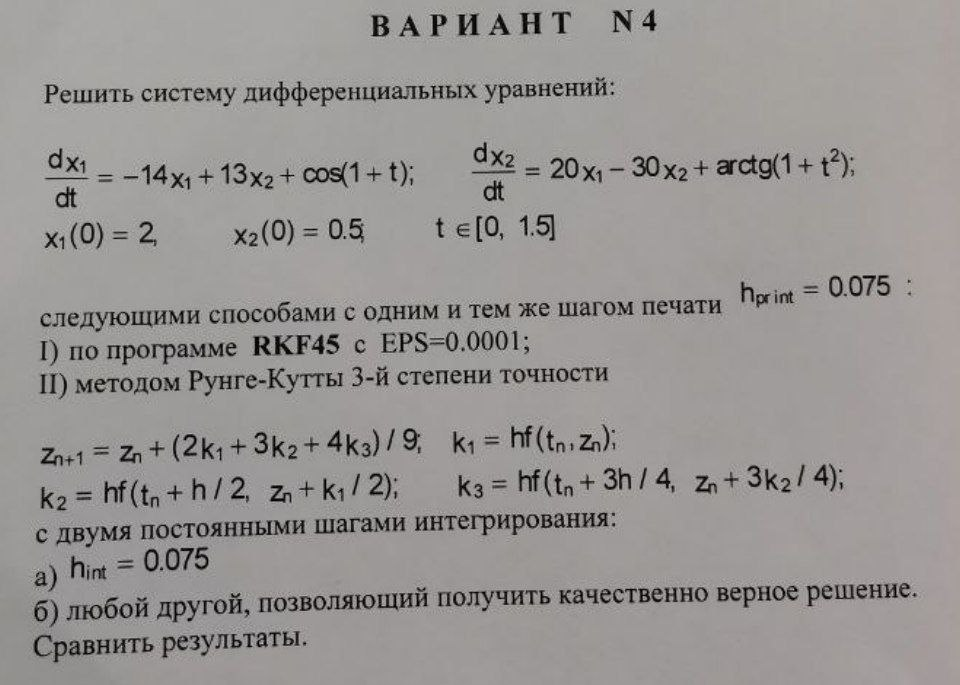
[Описание действий при выполнении 4](#_Toc192189830)

[Вывод 5](#_Toc192189831)

[Приложение. Скриншоты выполнения программы 6](#_Toc192189832)

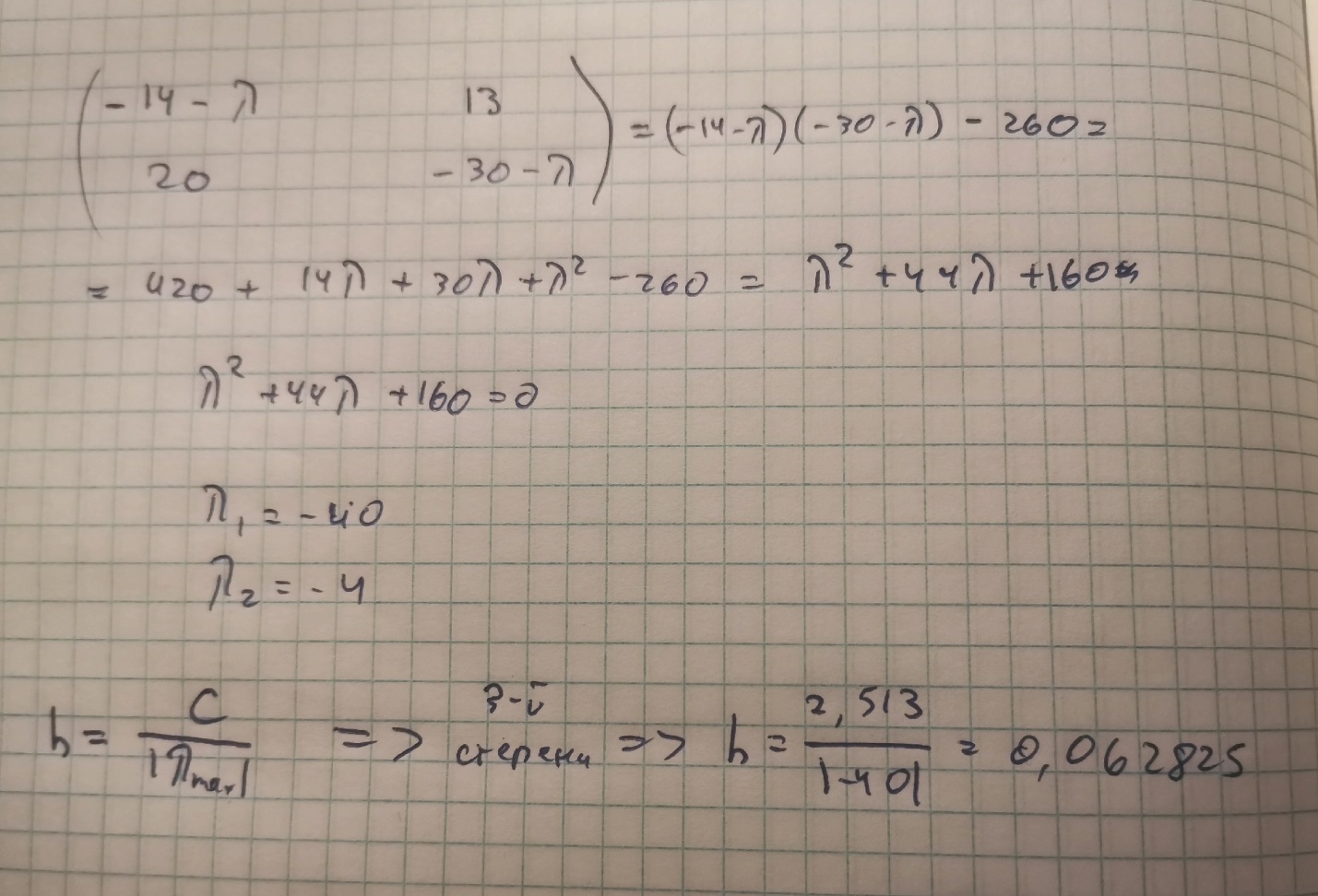
[Приложение. Код программы 7](#_Toc192189833)

# Формулировка задания



# Описание действий при выполнении

1. Использовав стандартную библиотеку с функцией **RKF45** предоставленную преподавателем (язык C++), я реализовал программу, которая решает систему дифференциальных уравнений с помощью подпрограммы **RKF45**.
2. Так же мною была реализована функция **runge3,** которая так же решает систему дифференциальных уравнений. Она представляет из себя метод Рунге-Кутты 3-й степени точности. Внутри нее реализовано определение каждого из k(1-3) и вычисляется решение СДУ
3. Первый способ решения вычисляется с шагом 0.075, второй с шагом 0.000001 для лучшей точности. Сравнение результатов приведу в выводе
4. Расчет критического шага:



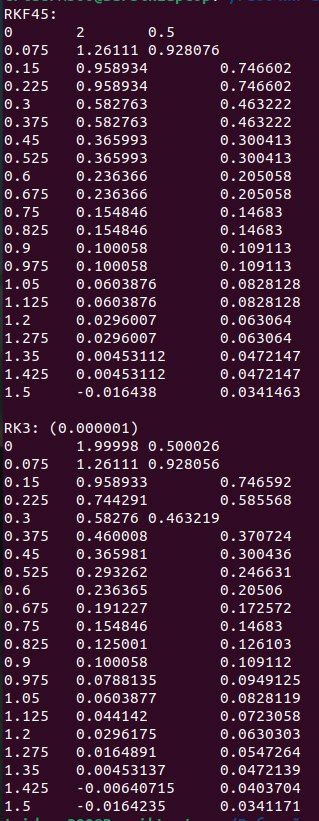
# Вывод

Я изучил подпрограмму **RKF45** практическим путем, написав для нее программу тестирования. Тестирование было выполнено на основе задания из моего варианта, а именно: решение системы дифференциальных уравнений.

В задании требовалось сравнить результаты двух вычислений: встроенная программа **RKF45** и собственноручно написанная функция для метода Рунге-Кутты 3-й степени. С результатами вычислений можно ознакомиться в скриншотах выполнения программы.

По полученным данным, можно сделать вывод, что при нарушении критического шага метод начинается расходиться с точными значениями **RKF45**. Таким образом, нельзя нарушать критический шаг, иначе точность значений будет нарушена. Порядок действий при выборе шага: рассчитать критический шаг -> выбрать свой шаг на основе критического

# Приложение. Скриншоты выполнения программы



# Приложение. Код программы

#include <iostream>

#include <iomanip>

#include "RKF45.CPP"

#include <stdio.h>

#include <cmath>

void runge3(void (\*f)(REAL T, REAL\* X, REAL\* dX), REAL\* X, REAL T, REAL H)

{

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

REAL temp[2];

f(T, X, k1);

temp[0] = (X[0] + (1 / 2) \* k1[0]);

temp[1] = (X[1] + (1 / 2) \* k1[1]);

f(T + H / 2, temp, k2);

temp[0] = (X[0] + (3 \* 1 / 4) \* k2[0]);

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

f(T + 3 \* H / 4, temp, k3);

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

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

}

void fun(REAL T, REAL\* X, REAL\* dX)

{

dX[0] = -14.0 \* X[0] + 13.0 \* X[1] + cos(1 + T);

dX[1] = 20 \* X[0] - 30 \* X[1] + atan(1 + T \* T);

}

bool isEqualToRange(REAL number) {

for (REAL i = 0; i <= 1.5; i += 0.075) {

if (std::abs(number - i) < 1e-10) { // Используем небольшую точность для сравнения

return true;

}

}

return false;

}

int main()

{

REAL X[2] = { 2.0, 0.5 };

REAL t = 0.0;

REAL tOut = 0.0;

REAL relerr = 0.0;

REAL abserr = 0.0001;

REAL H = 0.075;

REAL WORK[2 + 6 \* 2];

int flag = 1;

std::cout << "RKF45:" << std::endl;

while (tOut < 1.5)

{

RKF45(fun, 2, X, t, tOut, relerr, abserr, WORK, flag);

std::cout << tOut << "\t" << X[0] << "\t" << X[1] << std::endl;

tOut += H;

}

tOut = 0.0;

X[0] = 2.0;

X[1] = 0.5;

H = 0.000001;

std::cout << std::endl;

std::cout << "RK3: (0.000001)" << std::endl;

for (int i = 0; i <= 1500000; i++)

{

runge3(fun, X, tOut, H);

if ((i % 10000) == 0 and isEqualToRange(tOut) or i == 75000 or i == 225000 or i == 375000 or i == 525000 or i == 675000 or i == 825000 or i == 975000 or i == 1125000 or i == 1275000 or i == 1425000)

{

std::cout << tOut << "\t" << X[0] << "\t" << X[1] << std::endl;

}

tOut += H;

}

}