Министерство образования и науки РФ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа программной инженерии

**Отчёт по лабораторной работе №3**

1. по дисциплине «Вычислительная математика»
2. Выполнил
3. студент гр. 3530904/90002 Мэн Цзянин

Преподаватель С.П.Воскобойников

1. Санкт-Петербург
2. 2020

Оглавление

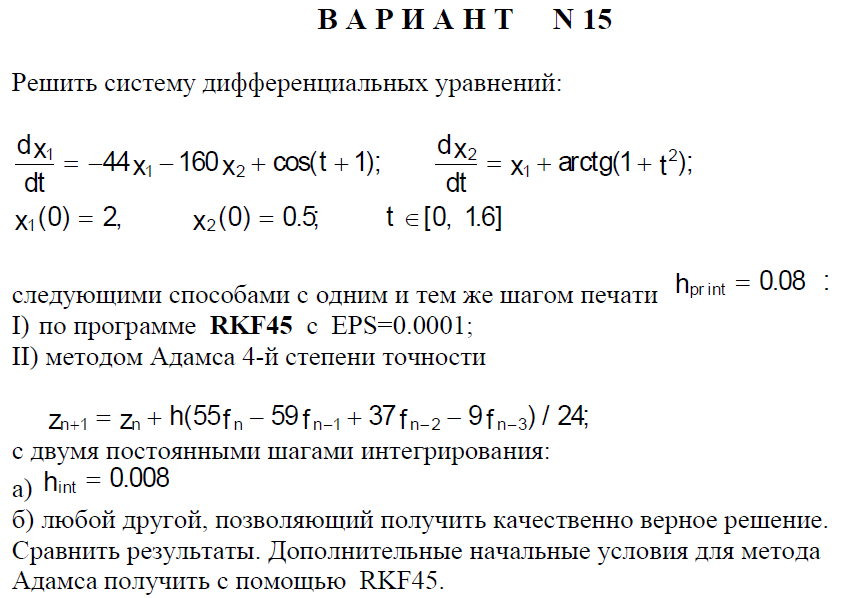
[1. Постановка задачи: 3](#_Toc64690199)

[2. Код программы： 3](#_Toc64690200)

[3. Результат работы программы: 5](#_Toc64690201)

[4. Вывод: 23](#_Toc64690202)

# 1. Постановка задачи:



# 2. Код программы：

#include <math.h>

#include <stdio.h>

#include "cmath.h"

#include <iostream>

using namespace std;

double x, y[], dydx[];

double a = 0, b = 1.6;

const int MAX\_NUM\_POINTS = 20000;

const int number\_stating\_points = 4;

int rkf(double input\_h);

void adams\_method(double input\_h); //Adams

int main()

{

rkf(0.08);

adams\_method(0.008);

adams\_method(0.001);

return 0;

}

double adamsf0(double x, double y0, double y1)

{

return(-44 \* y0 - 160 \* y1 + cos(1 + x));

}

double adamsf1(double x, double y0, double y1)

{

return(y0 + atan(1 + x \* x));

}

int f(int n, double x, double y[], double dydx[])

{

dydx[0] = (-44 \* y[0] - 160 \* y[1] + cos(1 + x));

dydx[1] = (y[0] + atan(1 + x \* x));

return (0);

}

int rkf(double input\_h) //RKF45

{

double h, relerr, abserr, x1, x2;

int n, flag, nfe, maxfe, fail, step;

double y[2], yp[2];

n = 2;

flag = 1;

maxfe = 5000;

relerr = 1.0e-4;

abserr = 1.0e-4;

rkfinit(n, &fail);

if (fail == 0){

y[0] = 2.0;

y[1] = 0.5;

double t1 = 0.0, t2 = 0.0;

double t\_end = 1.6;

size\_t count = static\_cast<size\_t>((t\_end - t1) / input\_h);

printf(" t x[0] x[1]\n");

printf("-------------- rkf ----------------\n");

while (t1 <= t\_end) {

t2 = t1 + input\_h;

rkf45(f, n, y, yp, &t1, t2, &relerr, abserr, &h, &nfe, maxfe, &flag);

printf("%10.6f %10.6f %10.6f \n", t1, y[0], y[1]);

}

rkfend();

}

return (0);

}

void adams\_method(double input\_h) //Adams

{

double h, relerr, abserr, x1, x2;

int n, flag, nfe, maxfe, fail, step;

double y[2], yp[2];

n = 2;

flag = 1;

maxfe = 5000;

relerr = 1.0e-4;

abserr = 1.0e-4;

rkfinit(n, &fail);

if (fail == 0)

{

y[0] = 2.0;

y[1] = 0.5;

double y0\_adams[MAX\_NUM\_POINTS];

double y1\_adams[MAX\_NUM\_POINTS];

const int number\_points = (b - a) / input\_h;

printf(" t x[0] x[1]\n");

cout << "------------- adams, h=" << input\_h << " ---------------\n";

x1 = a;

x2 = a + input\_h;

y0\_adams[0] = 2;

y1\_adams[0] = 0.5;

for (int i = 1; i <= 3; i++)

{

rkf45(f, n, y, yp, &x1, x2, &relerr, abserr, &h, &nfe, maxfe, &flag);

y0\_adams[i] = y[0];

y1\_adams[i] = y[1];

x1 = x2;

x2 += input\_h;

}

// 4-step Adams method

for (int i = number\_stating\_points; i <= number\_points; i++)

{

x2 = input\_h \* i;

y0\_adams[i] =

y0\_adams[i - 1] + input\_h / 24 \*

(55 \* adamsf0(x2, y0\_adams[i - 1], y1\_adams[i - 1]) -

59 \* adamsf0(x2 - input\_h, y0\_adams[i - 2], y1\_adams[i - 2]) +

37 \* adamsf0(x2 - 2 \* input\_h, y0\_adams[i - 3], y1\_adams[i - 3]) -

9 \* adamsf0(x2 - 3 \* input\_h, y0\_adams[i - 4], y1\_adams[i - 4]));

y1\_adams[i] =

y1\_adams[i - 1] + input\_h / 24 \*

(55 \* adamsf1(x2, y0\_adams[i - 1], y1\_adams[i - 1]) -

59 \* adamsf1(x2 - input\_h, y0\_adams[i - 2], y1\_adams[i - 2]) +

37 \* adamsf1(x2 - 2 \* input\_h, y0\_adams[i - 3], y1\_adams[i - 3]) -

9 \* adamsf1(x2 - 3 \* input\_h, y0\_adams[i - 4], y1\_adams[i - 4]));

}

// output Adams result

for (int i = 0; i <= number\_points; i++)

{

x1 = input\_h \* i;

printf("%10.6f %10.6f %10.6f \n", x1, y0\_adams[i], y1\_adams[i]);

}

rkfend();

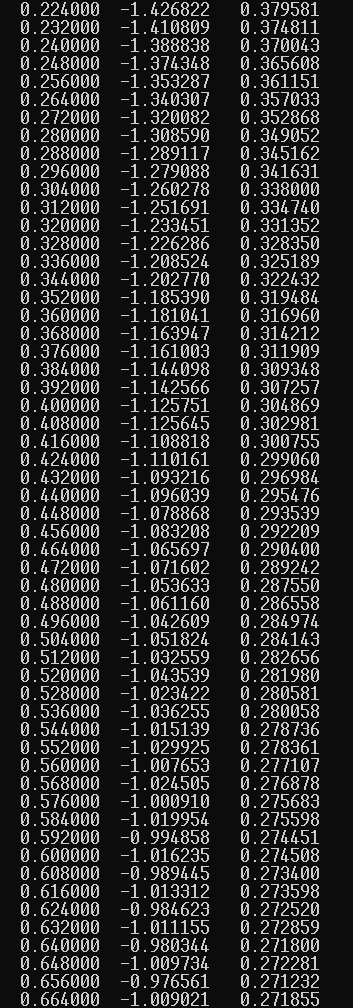
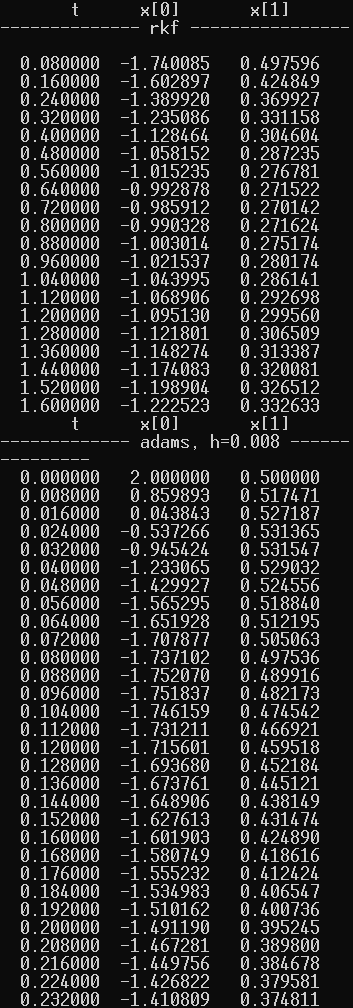
}

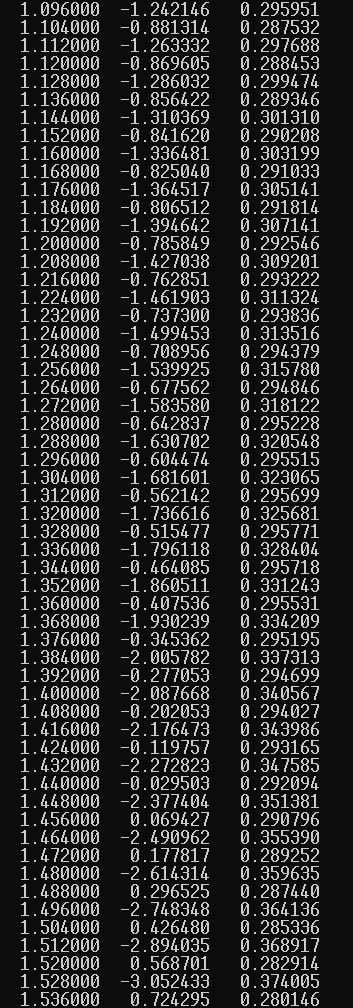
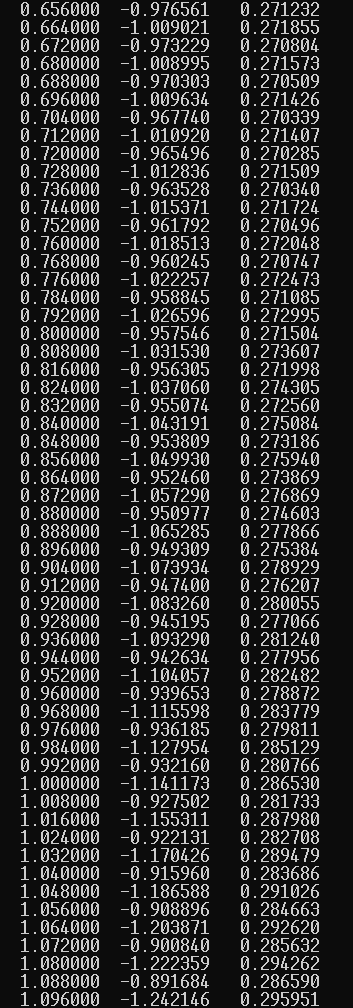
return ;

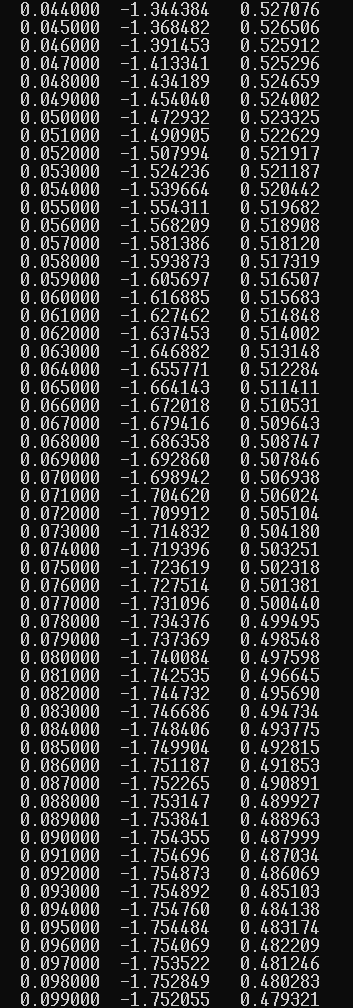
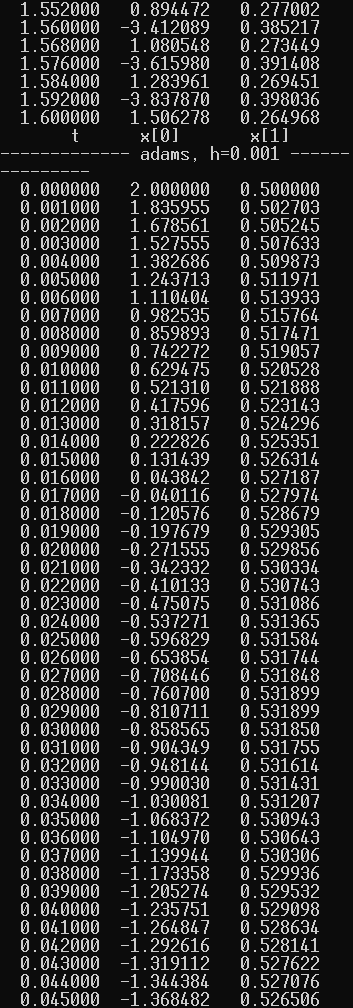
}

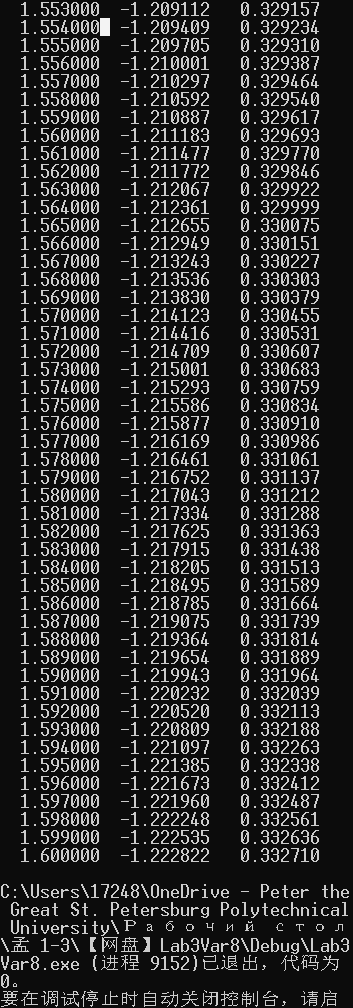
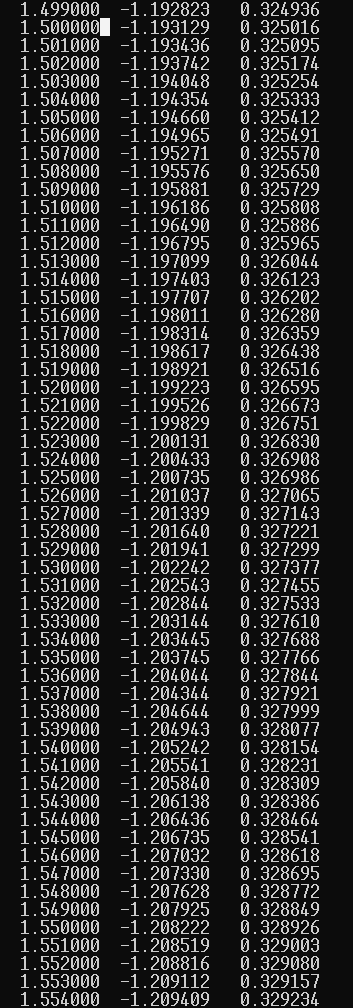
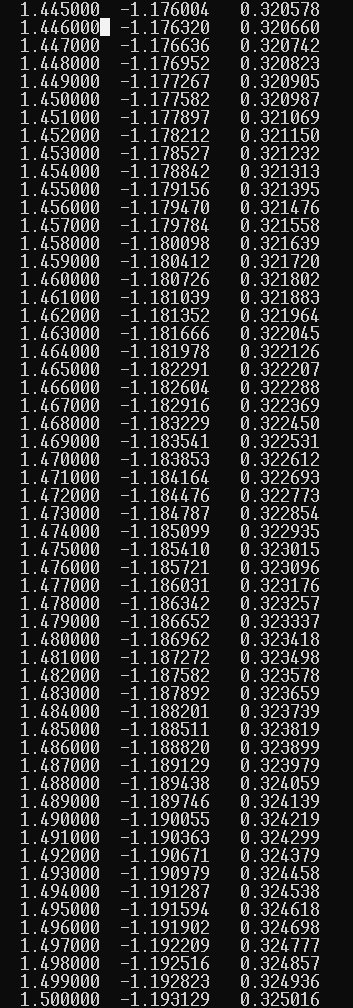
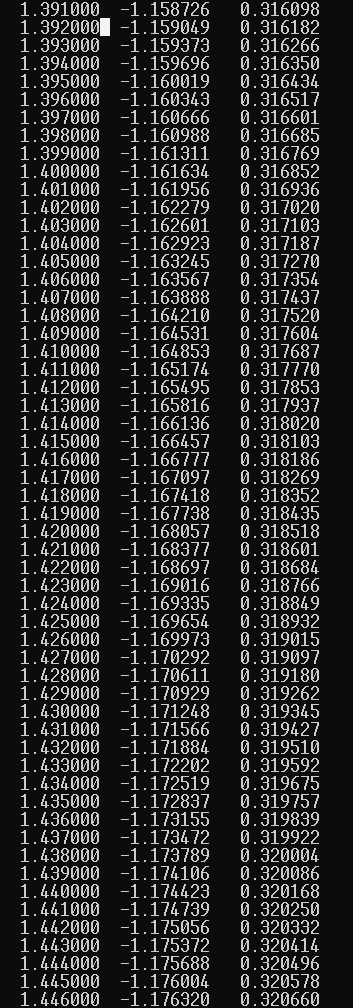
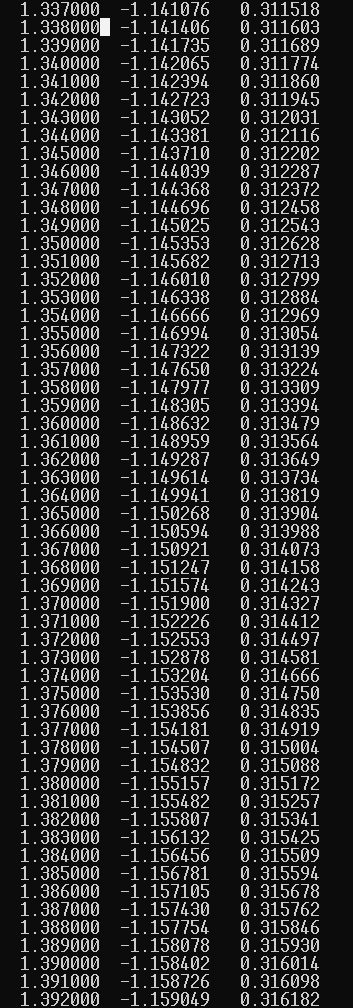
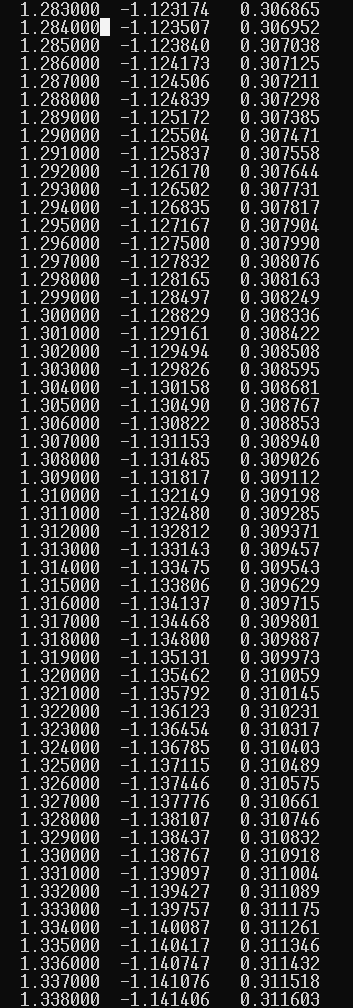
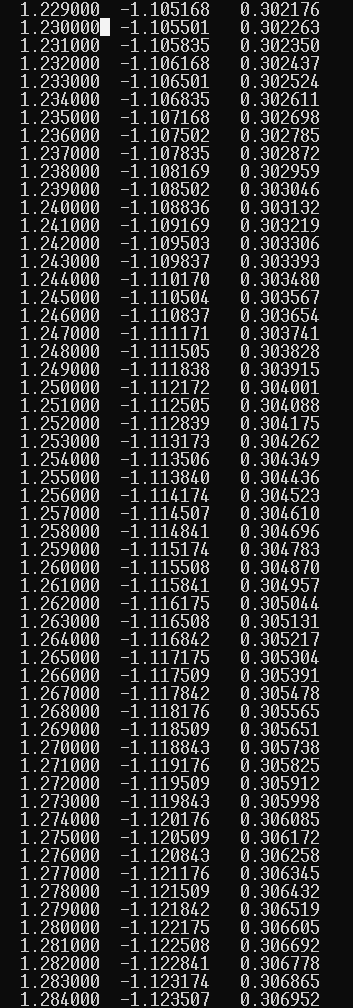
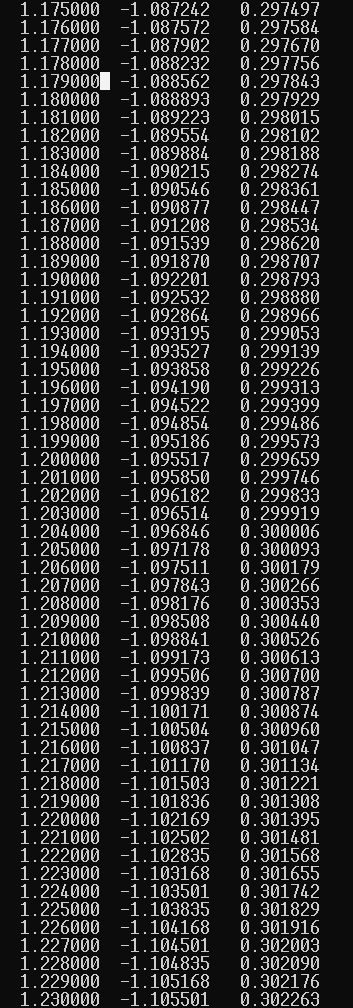
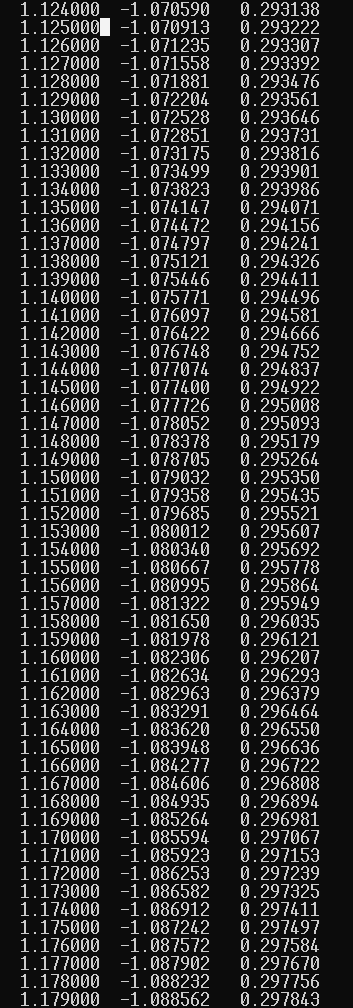
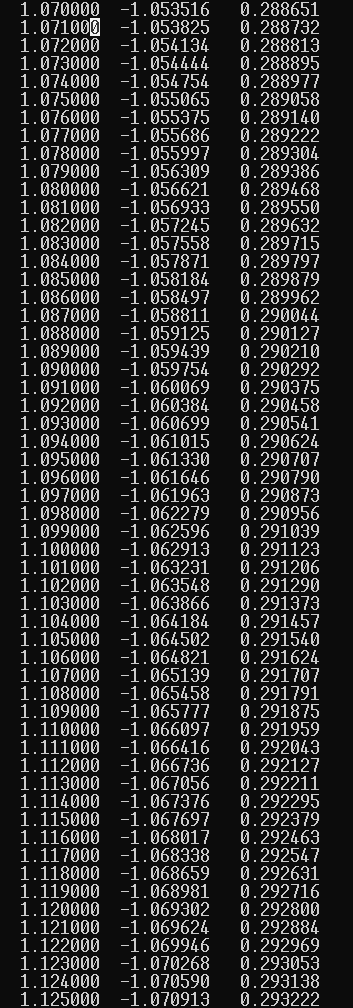
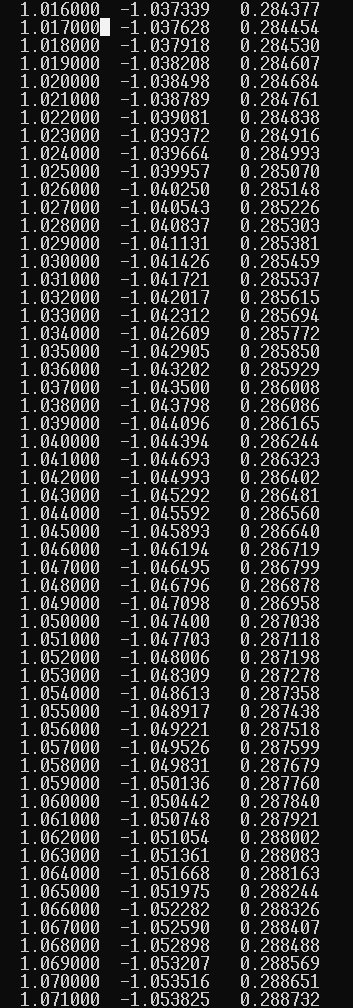
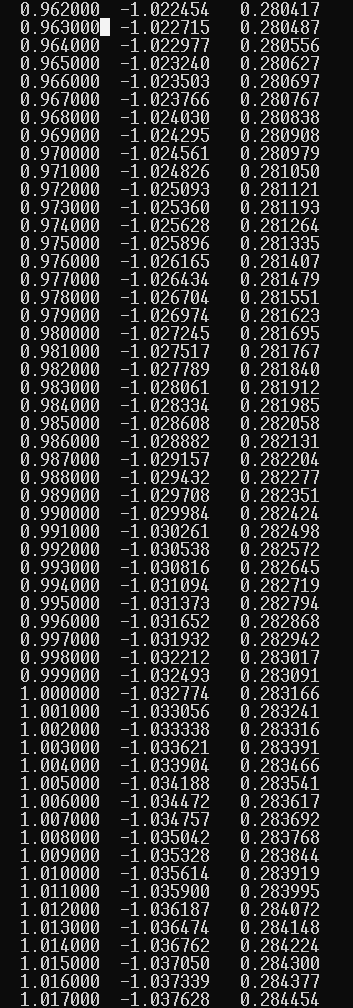
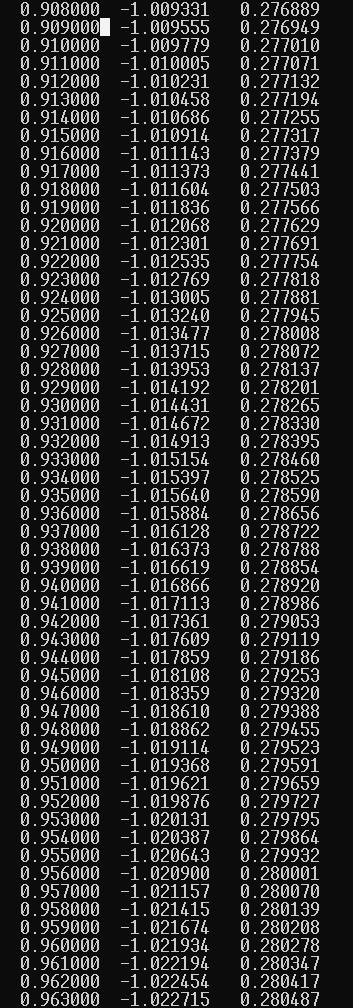
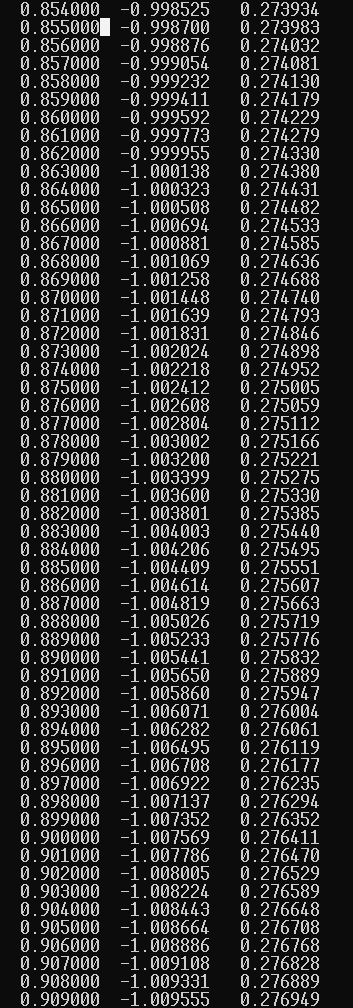
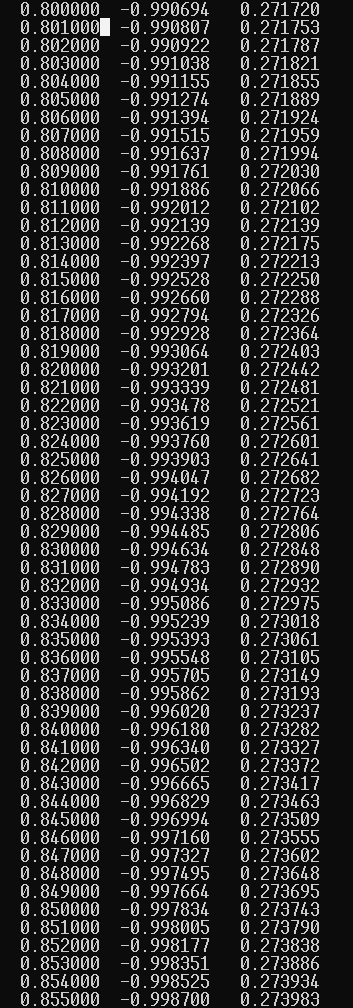
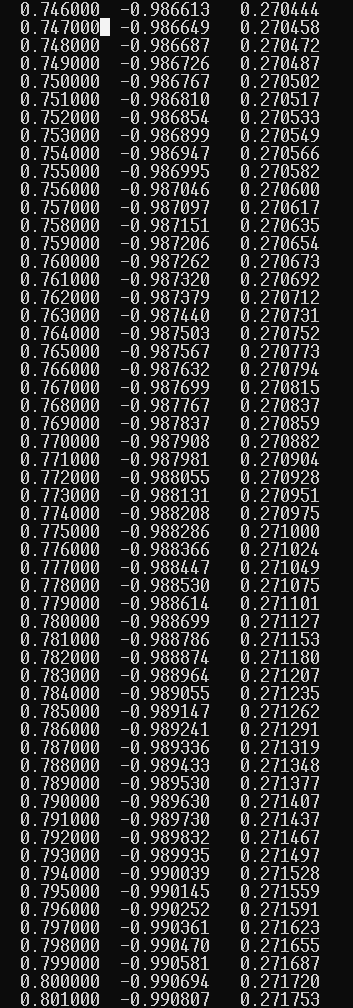
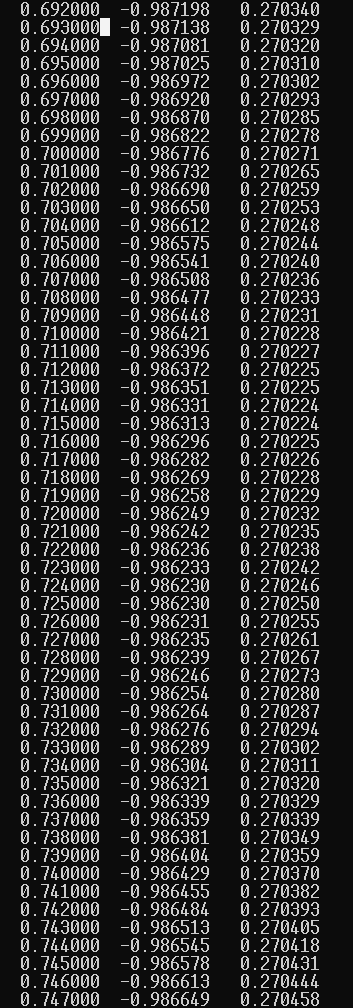
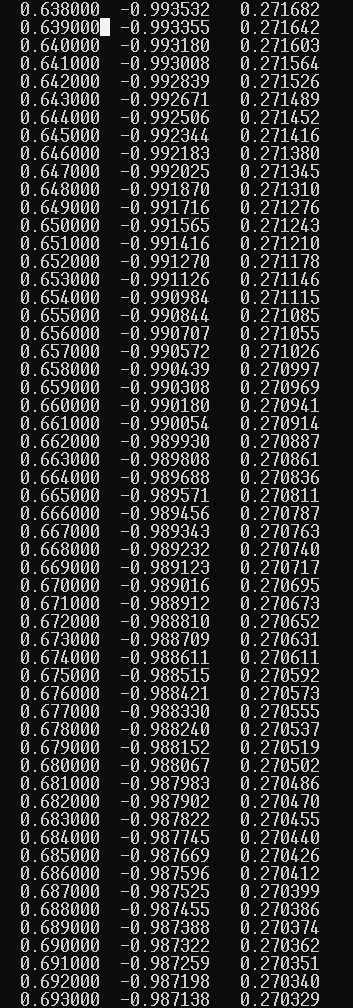
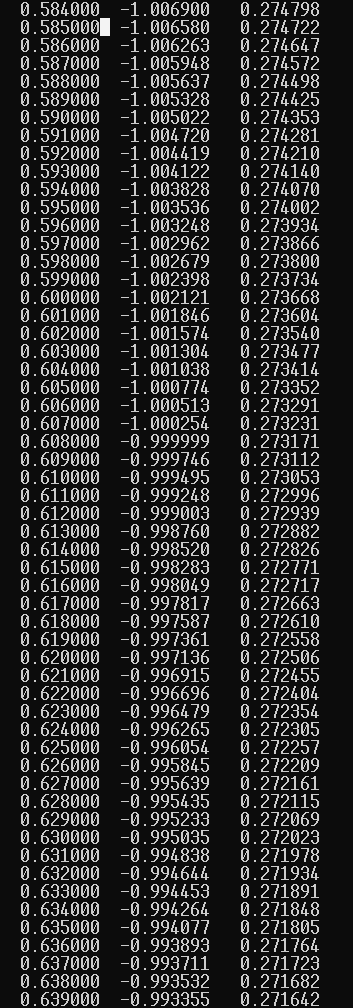
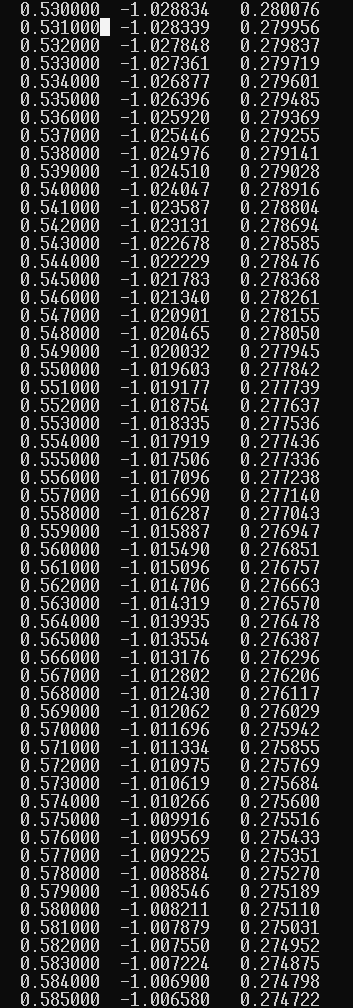
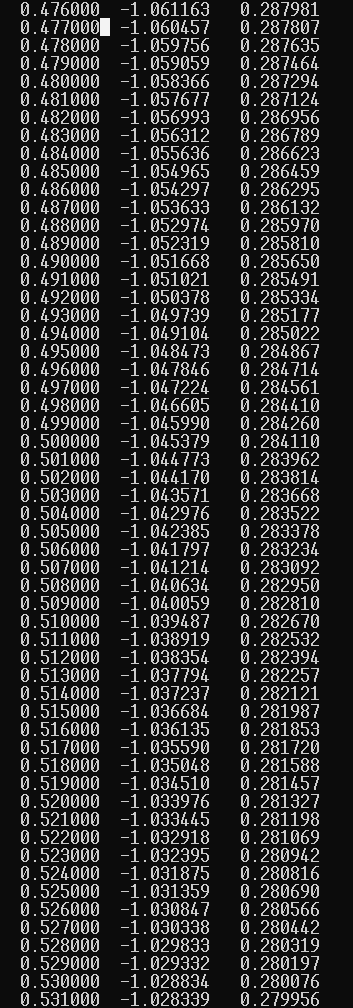
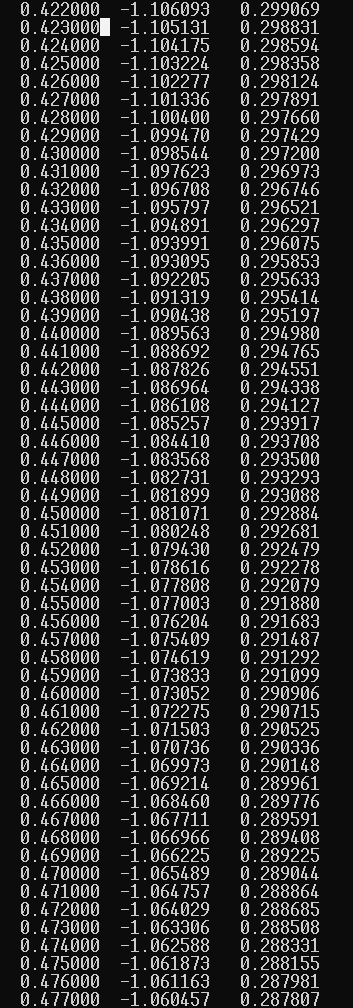
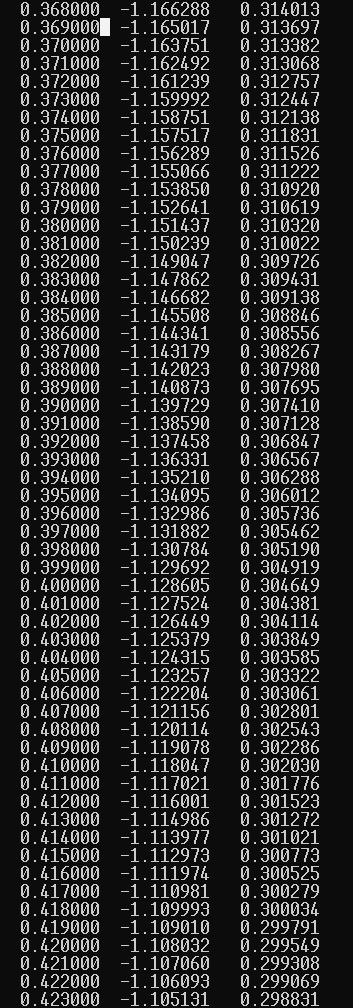
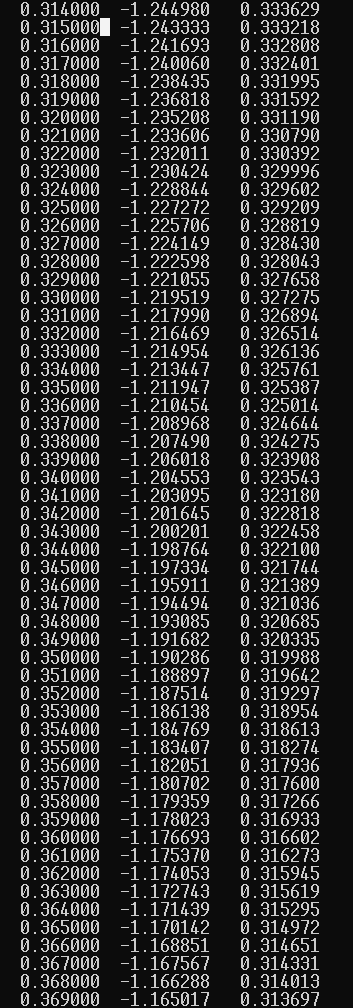
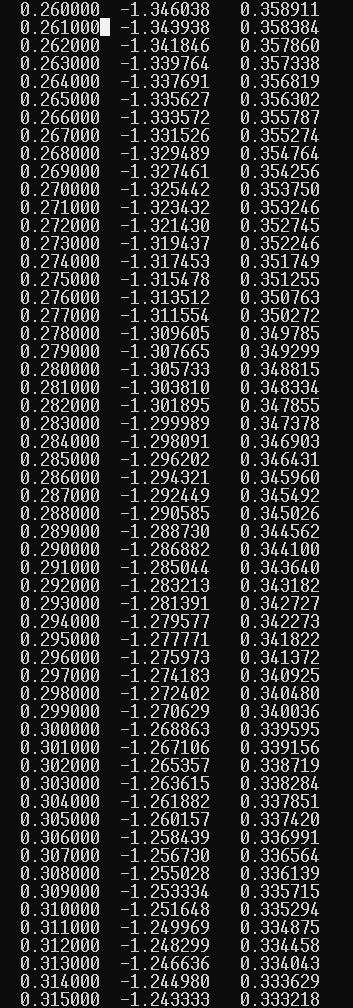
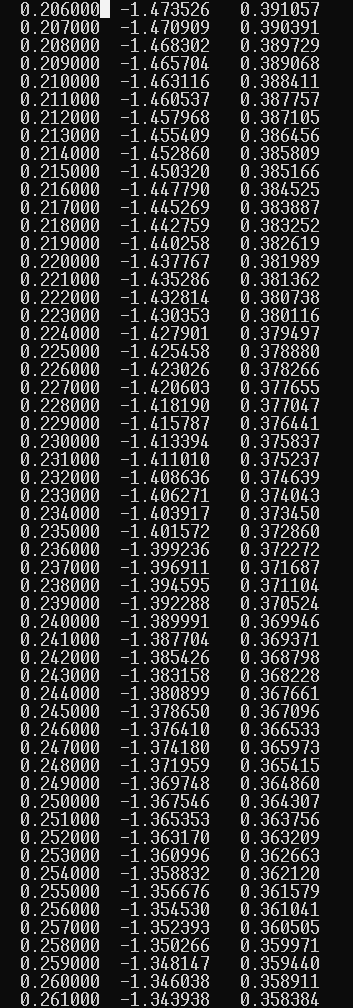
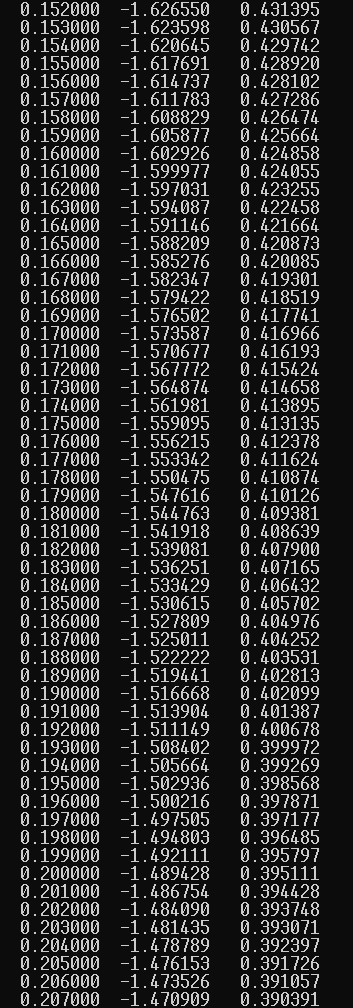
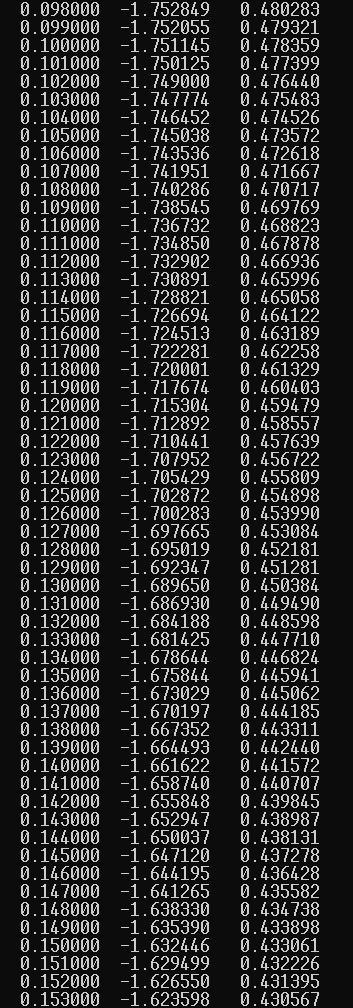
# 3. Результат работы программы:











С помощью программы RKF45 с шагом 0,08 было получено качественное решение. Усовершенствованным методом Адамса для шага 0,008 удалось дать даже близкого результата. Этим же методом, с шагом 0,001 удалось добиться точности.

# 4. Вывод:

В ходе данной лабораторной работы были получены навыки работы с программой RKF45 и была написана процедуру для решения системы дифференциальных уравнений на основе усовершенствованного метода Адамса.