Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный университет

телекоммуникаций и информатики»

кафедра ПМ и К

КУРСОВАЯ РАБОТА

По дисциплине “Вычислительная математика”

Выполнил: студент гр. ИП-814

Якунин А. В.

Проверил: Доцент кафедры ПМ и К

Рубан А.А.

Новосибирск 2020г.

**Содержание**

1. Задание 3

2.Описание работы программы 4

3.Скриншоты работы программы 7

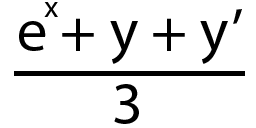
4.Листинг 9

**1.Задание**

Реализовать метод Рунге – Кутты не ниже 2 порядка с двойным пересчетом, для решения задачи Коши

**2.Описание работы программы**

Для реализации был выбран метод Рунге – Кутты 4 порядка



(Задача Коши)

Формулы метода Рунге – Кутты 4 порядка





В двойном пересчете для метода Рунге – Кутты, мы должны сравнивать парные элементы и так же в зависимости от порядка будут разные условия. Для 4 порядка условие таково





Реализация Рунге – Кутты 4 порядка

void du\_4(double x\_first, double x\_last, double h, double \*y\_mas, int n, double \*\*y\_mas\_for\_chek, int n\_shag)

{

int du\_step = 1;

double \*\*k\_mas = new double\* [4];

for (int I = 0; I < 4; i++) k\_mas[i] = new double [n];

double \*save\_y\_mas = nullptr;

save\_y\_mas = new double [n];

for (int I = 0; I < n; i++) save\_y\_mas[i] = y\_mas[i];

for(double x = x\_first; du\_step – 1 < n\_shag; x += h, du\_step++)

{

cout << endl << “x = “ << setprecision(15) << x + h;

for(int q = 0; q < n – 1; q++) k\_mas[0][q] = save\_y\_mas[q + 1];

k\_mas[0][n – 1] = formula(x, save\_y\_mas);

for(int I = 0; I < n; i++) y\_mas[i] = save\_y\_mas[i] + (h / 2.0) \* k\_mas[0][i];

for(int q = 0; q < n; q++) k\_mas[1][q] = y\_mas[q + 1];

k\_mas[1][n – 1] = formula(x + (h / 2.0), y\_mas);

for(int I = 0; I < n; i++) y\_mas[i] = save\_y\_mas[i] + (h / 2.0) \* k\_mas[1][i];

for(int q = 0; q < n; q++) k\_mas[2][q] = y\_mas[q + 1];

k\_mas[2][n – 1] = formula(x + (h / 2.0), y\_mas);

for(int I = 0; I < n; i++) y\_mas[i] = save\_y\_mas[i] + h \* k\_mas[2][i];

for(int q = 0; q < n; q++) k\_mas[3][q] = y\_mas[q + 1];

k\_mas[3][n – 1] = formula(x + h, y\_mas);

for(int I = 0; I < n; i++)

{

y\_mas[i] = save\_y\_mas[i] + (h / 6.0) \* (k\_mas[0][i] + (2.0 \* k\_mas[1][i]) + (2.0 \* k\_mas[2][i]) + k\_mas[3][i]);

save\_y\_mas[i] = y\_mas[i];

y\_mas\_for\_chek[i][du\_step – 1] = y\_mas[i];

}

for(int I = 0; I < n; i++) cout << “ | “ << setprecision(15) << y\_mas[i] << “ |”;

}

for (int I = 0; I < 4; i++) delete [] k\_mas[i];

delete [] k\_mas;

}

Реализация двойного пересчета

bool good = false, test\_all\_mas = false;

while(good == false)

{

double \*\*prev\_y\_mas = new double\* [n];

for (int I = 0; I < n; i++) prev\_y\_mas[i] = new double [n\_shag];

for(int I = 0; I < n; i++) y\_mas[i] = temp\_y\_mas[i];

du\_4(x\_first, x\_last, h, y\_mas, n, prev\_y\_mas, n\_shag);

step++;

prev\_shag = n\_shag;

h /= 2;

n\_shag = (x\_last / h) – (x\_first / h);

double \*\*next\_y\_mas = new double\* [n];

for (int I = 0; I < n; i++) next\_y\_mas[i] = new double [n\_shag];

cout << endl << “h = “ << h << endl;

for(int I = 0; I < n; i++) y\_mas[i] = temp\_y\_mas[i];

du\_4(x\_first, x\_last, h, y\_mas, n, next\_y\_mas, n\_shag);

for(int I = 0; I < n; i++)

{

for(int j = 0; j < prev\_shag; j++)

{

if(fabs((prev\_y\_mas[i][j] – next\_y\_mas[i][j \* 2 + 1])) < (15 \* eps)) test\_all\_mas = true;

else

{

test\_all\_mas = false;

break;

}

}

if(test\_all\_mas == false) break;

}

for (int I = 0; I < n; i++)

{

delete [] prev\_y\_mas[i];

delete [] next\_y\_mas[i];

}

delete [] prev\_y\_mas;

delete [] next\_y\_mas;

if(test\_all\_mas == true)

{

cout << endl << “----------------------------------“;

cout << endl << “Otvet: “ << endl;

for(int I = 0; I < n; i++) cout << endl << “ “ << setprecision(15) << y\_mas[i] << endl;

cout << endl << “Kolvo shagod: “ << step << endl;

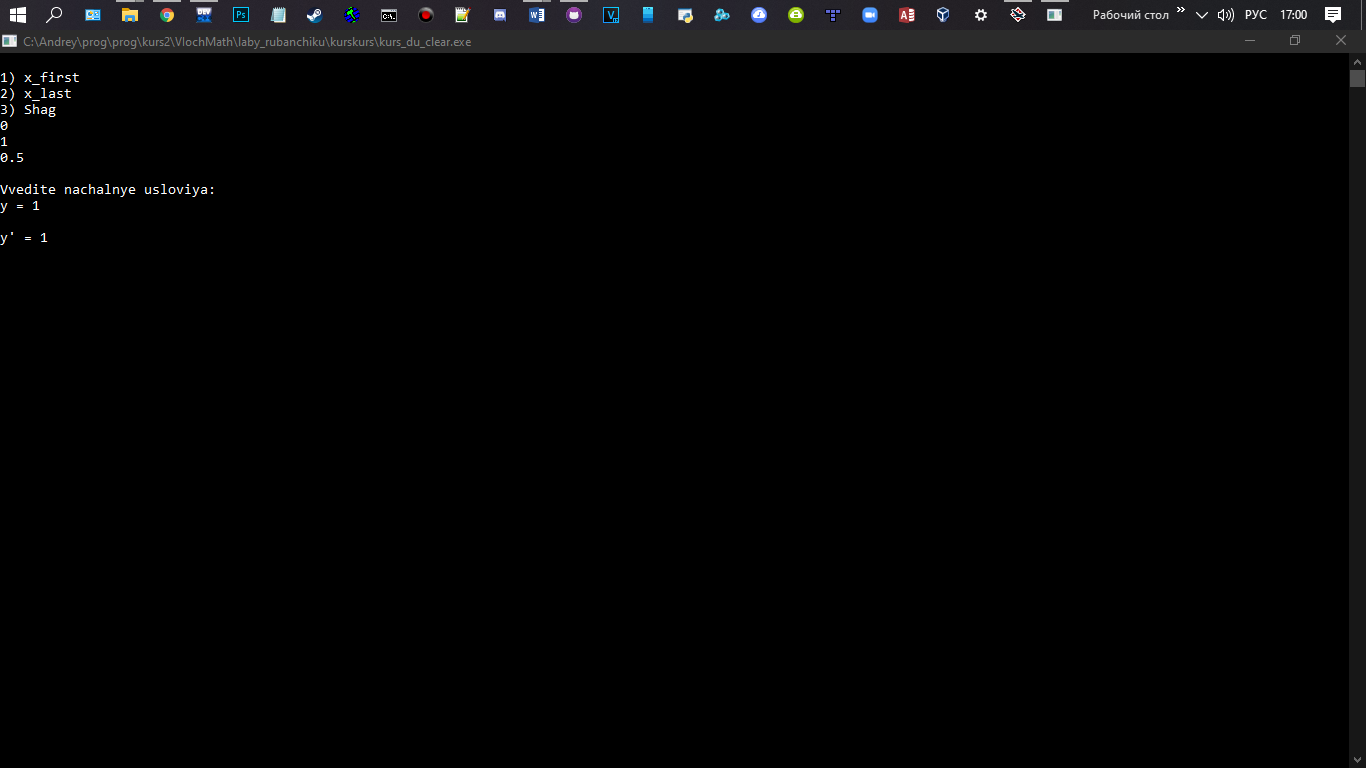
cout << endl << “h = : “ << h;

good = true;

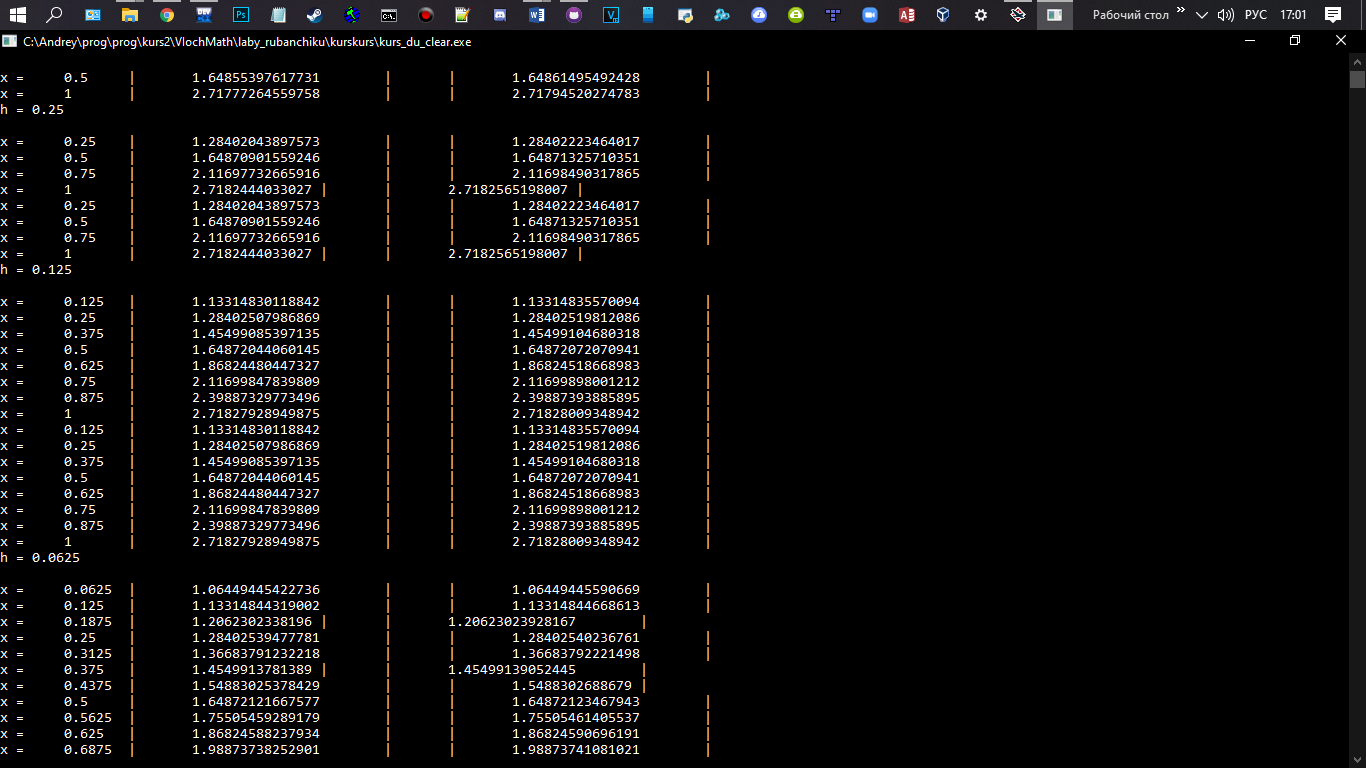
}

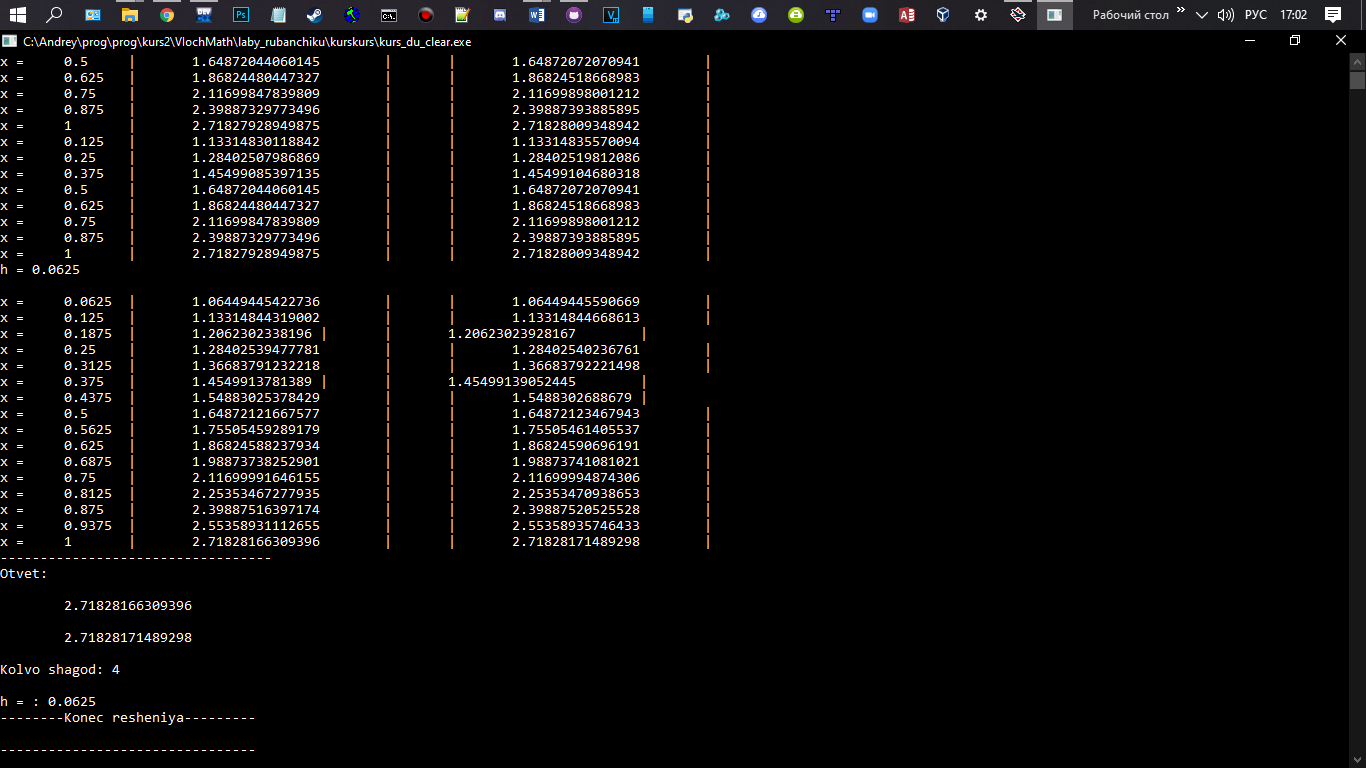
}

**3.** **Скриншоты работы программы**



**-**Вводим крайние точки, шаг и начальные условия (точность -6 степень)**-**





**-**Получаем ответ, при введенных данных, точность была достигнута при шаге h = 0.0625**-**

**4.Листинг**

**#include <stdio.h>**

**#include <iostream>**

**#include <cmath>**

**#include <string>**

**#include <iomanip>**

**using namespace std;**

**const double eps = 0.000001;**

**int opredelitel()**

**{**

**return 2;**

**}**

**double formula(double x, double \*y\_mas)**

**{**

**return (exp(x) + y\_mas[0] + y\_mas[1]) / 3.0;**

**}**

**void du\_4(double x\_first, double x\_last, double h, double \*y\_mas, int n, double \*\*y\_mas\_for\_chek, int n\_shag)**

**{**

**int du\_step = 1;**

**double \*\*k\_mas = new double\* [4];**

**for (int i = 0; i < 4; i++) k\_mas[i] = new double [n];**

**double \*save\_y\_mas = nullptr;**

**save\_y\_mas = new double [n];**

**for (int i = 0; i < n; i++) save\_y\_mas[i] = y\_mas[i];**

**for(double x = x\_first; du\_step - 1 < n\_shag; x += h, du\_step++)**

**{**

**cout << endl << "x = " << setprecision(15) << x + h;**

**for(int q = 0; q < n - 1; q++) k\_mas[0][q] = save\_y\_mas[q + 1];**

**k\_mas[0][n - 1] = formula(x, save\_y\_mas);**

**for(int i = 0; i < n; i++) y\_mas[i] = save\_y\_mas[i] + (h / 2.0) \* k\_mas[0][i];**

**for(int q = 0; q < n; q++) k\_mas[1][q] = y\_mas[q + 1];**

**k\_mas[1][n - 1] = formula(x + (h / 2.0), y\_mas);**

**for(int i = 0; i < n; i++) y\_mas[i] = save\_y\_mas[i] + (h / 2.0) \* k\_mas[1][i];**

**for(int q = 0; q < n; q++) k\_mas[2][q] = y\_mas[q + 1];**

**k\_mas[2][n - 1] = formula(x + (h / 2.0), y\_mas);**

**for(int i = 0; i < n; i++) y\_mas[i] = save\_y\_mas[i] + h \* k\_mas[2][i];**

**for(int q = 0; q < n; q++) k\_mas[3][q] = y\_mas[q + 1];**

**k\_mas[3][n - 1] = formula(x + h, y\_mas);**

**for(int i = 0; i < n; i++)**

**{**

**y\_mas[i] = save\_y\_mas[i] + (h / 6.0) \* (k\_mas[0][i] + (2.0 \* k\_mas[1][i]) + (2.0 \* k\_mas[2][i]) + k\_mas[3][i]);**

**save\_y\_mas[i] = y\_mas[i];**

**y\_mas\_for\_chek[i][du\_step - 1] = y\_mas[i];**

**}**

**for(int i = 0; i < n; i++) cout << " | " << setprecision(15) << y\_mas[i] << " |";**

**}**

**for (int i = 0; i < 4; i++) delete [] k\_mas[i];**

**delete [] k\_mas;**

**}**

**main()**

**{**

**double x\_first, x\_last, h;**

**cout << endl << "1) x\_first\n2) x\_last\n3) Shag" << endl;**

**cin >> x\_first >> x\_last >> h;**

**int n, step = 1, n\_shag = (x\_last / h) - (x\_first / h), prev\_shag;**

**string temp\_y = "y";**

**n = opredelitel();**

**double \*y\_mas = nullptr;**

**y\_mas = new double [n];**

**double \*temp\_y\_mas = nullptr;**

**temp\_y\_mas = new double [n];**

**cout << endl << "Vvedite nachalnye usloviya: ";**

**for(int i = 0; i < n; i++)**

**{**

**cout << endl << temp\_y << " = ";**

**cin >> y\_mas[i];**

**temp\_y\_mas[i] = y\_mas[i];**

**temp\_y += "'";**

**}**

**bool good = false, test\_all\_mas = false;**

**while(good == false)**

**{**

**double \*\*prev\_y\_mas = new double\* [n];**

**for (int i = 0; i < n; i++) prev\_y\_mas[i] = new double [n\_shag];**

**for(int i = 0; i < n; i++) y\_mas[i] = temp\_y\_mas[i];**

**du\_4(x\_first, x\_last, h, y\_mas, n, prev\_y\_mas, n\_shag);**

**step++;**

**prev\_shag = n\_shag;**

**h /= 2;**

**n\_shag = (x\_last / h) - (x\_first / h);**

**double \*\*next\_y\_mas = new double\* [n];**

**for (int i = 0; i < n; i++) next\_y\_mas[i] = new double [n\_shag];**

**cout << endl << "h = " << h << endl;**

**for(int i = 0; i < n; i++) y\_mas[i] = temp\_y\_mas[i];**

**du\_4(x\_first, x\_last, h, y\_mas, n, next\_y\_mas, n\_shag);**

**for(int i = 0; i < n; i++)**

**{**

**for(int j = 0; j < prev\_shag; j++)**

**{**

**if(fabs((prev\_y\_mas[i][j] - next\_y\_mas[i][j \* 2 + 1])) < (15 \* eps)) test\_all\_mas = true;**

**else**

**{**

**test\_all\_mas = false;**

**break;**

**}**

**}**

**if(test\_all\_mas == false) break;**

**}**

**for (int i = 0; i < n; i++)**

**{**

**delete [] prev\_y\_mas[i];**

**delete [] next\_y\_mas[i];**

**}**

**delete [] prev\_y\_mas;**

**delete [] next\_y\_mas;**

**if(test\_all\_mas == true)**

**{**

**cout << endl << "----------------------------------";**

**cout << endl << "Otvet: " << endl;**

**for(int i = 0; i < n; i++) cout << endl << " " << setprecision(15) << y\_mas[i] << endl;**

**cout << endl << "Kolvo shagod: " << step << endl;**

**cout << endl << "h = : " << h;**

**good = true;**

**}**

**}**

**cout << endl << "--------Konec resheniya---------" << endl;**

**return 0;**

**}**