РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

ОТЧЕТ

по лабораторной работе № 5

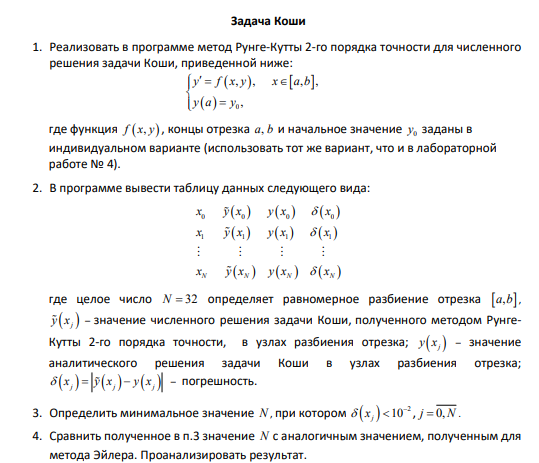
дисциплина: Вычислительные методы

Студент:

Группа:

**МОСКВА**

2021 г.

**Постановка задачи:**

**Индивидуальный вариант:**

**Выполнение работы:**

Точность метода Эйлера можно повысить, если воспользоваться для аппроксимации интеграла более точной формулой интегрирования – формулой трапеций.

****

Данная формула оказывается неявной относительно *yk+1* (это значение есть и в левой и в правой части выражения), то есть является уравнением относительно *yk+1*. Однако, можно поступить иначе и приблизительно вычислить значение функции в узле *k+1* с помощью обычной формулы Эйлера:

**,**

которое можно использовать при вычислении (4).

Таким образом, метод Рунге-Кутта 2-го порядка (метод Гюна) получается с пересчетом, т.е. как бы усовершенствованием метода Эйлера. Для каждого узла интегрирования производится следующая цепочка вычислений:

****

****

***Код программы***

#include <iostream>

#define \_USE\_MATH\_DEFINES

#include <cmath>

using namespace std;

double calculate\_y(double x) //analyticheskoe reshenie

{

return exp(x\*x);

}

double calculate\_f(double x, double y) //funkcia iz zadania

{

return 2\*x\*y;

}

int main()

{

double a=0; // levaya granica

double b=1.0; // pravaya granica

double y0=1.0; // uslovie zadachi Koshi iz zadania

int n=32; // melkost razbieneia

double h=(b-a)/n; // shag algoritma

double x[n+1]; // massiv znacheiy x

double y[n+1]; //massiv znacheiy y

x[0]=a;

y[0]=y0;// uslovie zadachi Koshi iz zadania

for (int i=1; i<=n; i++) // metod eilera

{

x[i]=x[i-1]+h;

y[i]=y[i-1]+(h/2.0)\*(calculate\_f(x[i-1], y[i-1]) + calculate\_f(x[i-1]+h, y[i-1]+h\*calculate\_f(x[i-1], y[i-1])) );

}

cout.precision(10); // fix dlya vivoda na ekran

cout.setf(ios::fixed);

cout<<"x\t\t"<<"y\t\t"<<"runge\t\t"<<"err"<< endl;

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

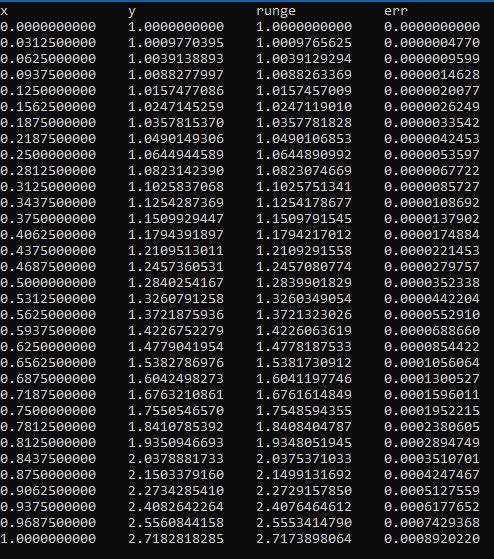
{

double error=abs(y[i]-calculate\_y(x[i])); // vichislenie otkloneniy

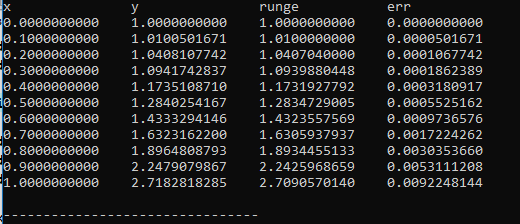
cout<<x[i]<<'\t'<<calculate\_y(x[i])<<'\t'<<y[i]<<'\t'<<error<< endl; // vivod na ekran: x, tochoe reshenie, reshenie eilera, otklonenie

}

}



Нужная точность достигается при N= 10 (в случае метода Эйлера при N = 350)



**Вывод** – Познакомились с задачей нахождения частного решения диф уравнения методом Рунге-Кутты 2го порядка