ГОУ ВПО «Донецкий национальный университет»

Физико-технический факультет

Кафедра компьютерных технологий

ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

ЛАБОРАТОРНАЯ РАБОТА №5.

**Численное решение задачи Коши**

Выполнил:

Давыденко Дмитрий

Студент III курса группы ИВТ-6

Проверил: асс. Пшеничный К.А.

Донецк 2022

Решить задачу Коши методом Эйлера и Рунге-Кутта 4-го порядка:

, .

Докажите для вашего варианта существование решения на заданном отрезке ().

Для тех, кто не программирует, во-первых, решите задачу с помощью **WolframAlpha** (<http://www.wolframalpha-ru.com/2013/07/wolframalpha_17.html>). Сделайте выводы. Во-вторых, сделайте соответствующее расчётное задание с преподавателем в лаборатории.

Для тех, кто программирует, решите задачу на трёх сетках при *N* =10, 50, 500. Постройте графики решений для обоих методов на всех сетках. Сделайте выводы.



Код программы

#include <iostream>

#include <fstream>

#include <cmath>

using namespace std;

double F(double t, double v) {

return 1 / t;

}

int main(int argc, char\*\* argv)

{

double a = 1; double b = 5;

double n = 50;

double h = (b - a) / n;

std::unique\_ptr<double[]> T(new double[(int)n]);

std::unique\_ptr<double[]> V(new double[(int)n]);

T[0] = a; V[0] = 0;

//метод Эйлера

for (int i = 1; i <= n - 1; i++) {

T[i] = a + i \* h;

V[i] = V[i - 1] + h \* F(T[i - 1], V[i - 1]);

}

std::ofstream output;

output.open("Ex1\_Euler.txt");

for (int i = 0; i <= n - 1; i++) {

output << T[i] << '\t' << V[i] << std::endl;

}

// std::unique\_ptr<double[]> T(new double[(int)n]);

//std::unique\_ptr<double[]> V(new double[(int)n]);

std::unique\_ptr<double[]> V1(new double[(int)n]);

std::unique\_ptr<double[]> V2(new double[(int)n]);

std::unique\_ptr<double[]> V3(new double[(int)n]);

std::unique\_ptr<double[]> V4(new double[(int)n]);

//метод Рунге-Кутта

for (int i = 1; i <= n - 1; i++) {

T[i] = a + i \* h;

V1[i] = h \* F(T[i - 1], V[i - 1]);

V2[i] = h \* F(T[i - 1] + h / 2.0, V[i - 1] + V1[i] / 2.0);

V3[i] = h \* F(T[i - 1] + h / 2, V[i - 1] + V2[i] / 2);

V4[i] = h \* F(T[i - 1] + h, V[i - 1] + V3[i]);

V[i] = V[i - 1] + (V1[i] + 2 \* V2[i] + 2 \* V3[i] + V4[i]) / 6;

}

std::ofstream otput\_2;

otput\_2.open("Ex1\_Runge–Kutta.txt");

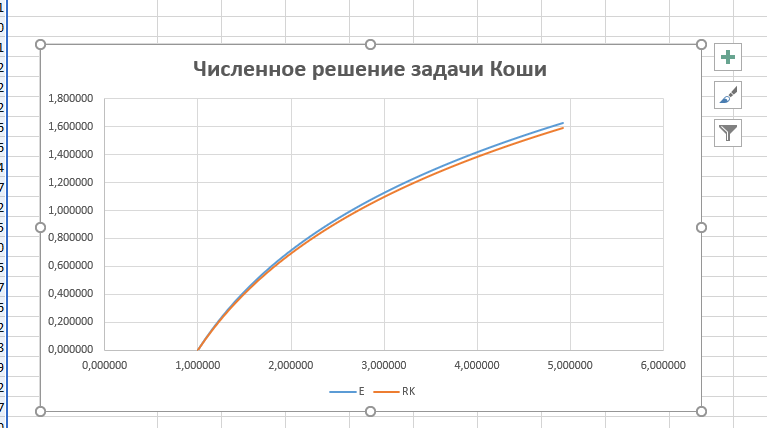
for (int i = 0; i <= n - 1; i++) {

otput\_2 << T[i] << '\t' << V[i] << std::endl;

}

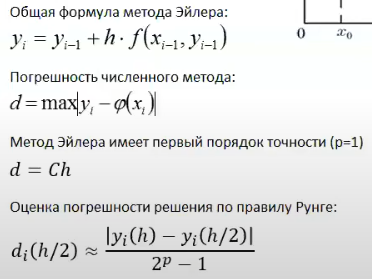
return 0;

}

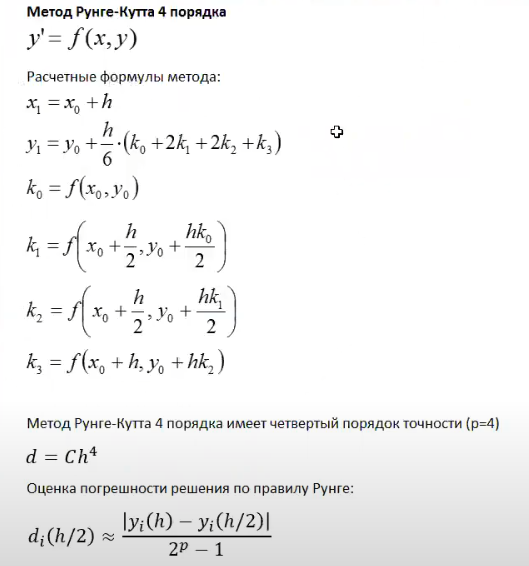


**Контрольные вопросы**

1. **Напишите формулу метода Эйлера. Какова погрешность аппроксимаций этого метода?**



1. **Напишите формулу какого-либо метода Рунге-Кутта. Какова точность этого метода?**



1. **Как численно вычисляются правая, левая и центральная производные? Какая схема вычислений даёт наименьшую погрешность аппроксимаций?**

Формулы для приближенного вычисления первой производной функции:

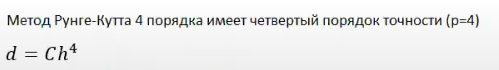
* левая разностная производная ,
* правая разностная производная ,
* центральная разностная производная , эта формула точнее предыдущих.

Таким образом, формулы Л и П имеют первый порядок точности по  . Иначе говоря, правая и левая разностные производные аппроксимируют  с первым порядком точности относительно  , а центральная разностная производная – со вторым порядком.

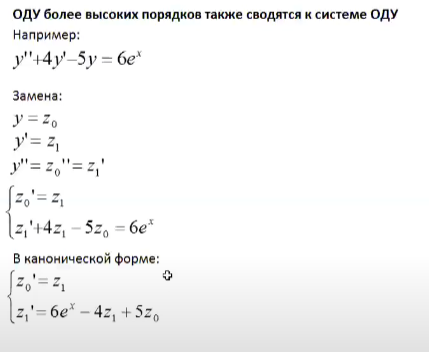
1. **Как погрешность численного дифференцирования зависит от шага дифференцирования?**

Это зависит от метода ЧД.

 уменьшение шага прямо пропорционально погрешности. Например если уменьшить шаг с 0,4 до 0,2 погрешность уменьшится в 2раза.

т.епогрешность пропорциональная шагу в 4степени. Если шаг уменьшить в 10раз то погрешность уменьш. в 10^4 раз.

1. **Как численно можно найти вторую производную?**



1. **Что такое явные и неявные конечно-разностные схемы решения дифференциальных уравнений? Что в них хорошего и плохого?**

Разностная схема — это конечная система алгебраических уравнений, поставленная в соответствие какой-либо дифференциальной задаче, содержащей дифференциальное уравнение и дополнительные условия (например, краевые условия и/или начальное распределение). Таким образом, разностные схемы применяются для сведения дифференциальной задачи, имеющей континуальный характер, к конечной системе уравнений, численное решение которых принципиально возможно на вычислительных машинах.

**Явные** схемы вычисляют значение сеточной функции через данные соседних точек. Явные схемы часто оказываются неустойчивыми.

**Неявные** схемы используют уравнения, которые выражают данные через несколько соседних точек результата. Для нахождения результата решается система линейных уравнений. Неявные схемы обычно являются устойчивыми.

1. **Что такое жёсткость системы обыкновенных дифференциальных уравнений? Физический смысл жёсткости?**

Жёсткой системой обыкновенных дифференциальных уравнений (ОДУ) называется (нестрого говоря) такая система ОДУ, численное решение которой явными методами (например, методами Рунге — Кутты или Адамса) является неудовлетворительным из-за резкого увеличения числа вычислений (при малом шаге интегрирования) или из-за резкого возрастания погрешности (так называемого, взрыва погрешности) при недостаточно малом шаге. Для жёстких систем характерно то, что для них неявные методы дают лучший результат, обычно несравненно более хороший, чем явные методы.

**Физ. смысл**

При изучении уравнений химической кинетики с одновременным присутствием очень медленно и очень быстро протекающих химических реакций. Тогда неожиданно оказалось, что считавшиеся исключительно надежными методы Рунге-Кутты стали давать сбой при расчете этих задач.

1. **Как решают задачу Коши в случае дифференциального уравнения высокого порядка?**

