**ИНСТИТУТ ТРАНСПОРТА И СВЯЗИ**



ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ НАУК И ТЕЛЕКОММУНИКАЦИЙ

Лабораторная работа №4

по предмету

«Численные методы»

«Решение дифференциальных уравнений первого порядка»

Выполнили: Беляев Марк

Михаил Красильников

Группа: 4601BV

Проверил: Александр Граковский

Рига, 2020

Содержание

[1. Задание. 3](#_Toc29836809)

[2. Аппроксимация методом МНК 3](#_Toc29836810)

[3. Индивидуальный метод – заданных точек 5](#_Toc29836811)

[4. Графическое сравнение методов 6](#_Toc29836812)

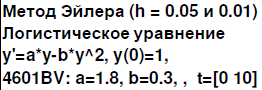
[5. Выводы 7](#_Toc29836813)

1. Задание.

Для выполнения лабораторной работы необходимо

1. реализовать метод Эйлера
2. реализовать индивидуальный метод – Метод Гира 2-го порядка ,
3. графически сравнить два реализованных метода, т.е. на один график нанести одну функцию, которая будет обработана двумя различными методами приближения.

Задачи для решения:



1. Метод Эйлера  
     
   public (List<float>, List<float>) Solve(float startXPoint, float endXPoint, float initialСondition, float partition)

{

int overalSteps = (int)(1 / partition);

var step = (endXPoint - startXPoint) \* partition;

step = (float)Math.Round(step \* 100f) / 100f; // Round

X = startXPoint;

Y = initialСondition;

XList.Add(X);

YList.Add(Y);

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

{

Y += step \* F(X, Y);

YList.Add(Y);

X += (float)Math.Round(step \* 100f) / 100f; // Round

XList.Add((float)Math.Round(X \* 100f) / 100f); // Round

}

return (XList, YList);

}

private float F(float x, float y)

{

return (1.8f \* y) + (0.3f \* (float)Math.Pow(x,2));

}

**Вызов для h = 0.01**

var result = euler.Solve(0, 10, 1, 0.01f);

**Вызов для h = 0.05**  
var result = euler.Solve(0, 10, 1, 0.05f);

1. Индивидуальный метод – Метод Гира 3-го порядка

public class Main {

private static final float LOW\_PRECISION = 0.05f;

private static final float HIGH\_PRECISION = 0.01f;

public static void main(String[] args) {

float end = 1f;

float y0 = 1.0f;

float step = Main.HIGH\_PRECISION;

int rank = 2;

int n = (int)(end / step);

float[] listOfY = new float[n];

listOfY[0] = y0;

for (int i = 1; i <= rank; i++) {

float y = listOfY[i - 1];

listOfY[i] = y + step \* f( step \* i, y);

}

for (int i = rank; i < n; i++) {

float y1 = listOfY[i - 1];

float y2 = listOfY[i - 2];

float predict = f(step \* i, y1 + step \* f(step \* i, y1));

listOfY[i] = ((4.f / 3.f) \* y1) - ((1.f/3.f) \* y2) + (step \* (2.f/3.f) \* predict);

}

for (int i = 0; i < listOfY.length; i++) {

float X = (i \* step);

float real = \_f(X, listOfY[i]);

System.out.println("X = " + X + " Y = " + listOfY[i] + " err: " + (Math.abs(listOfY[i] - real)));

}

}

public static float f(float x, float y) {

return (1.8f \* y) - (0.3f \* (float)Math.pow(y,2));

}

public static float \_f(float x, float y) {

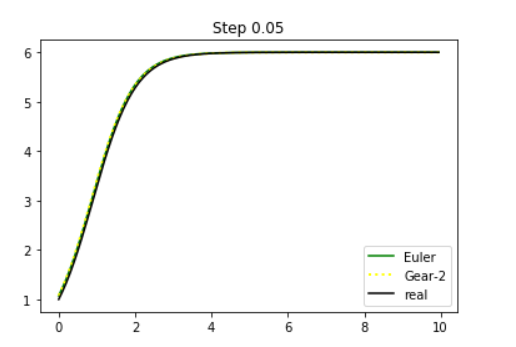
float tmp = (float)Math.exp(1.8f \* x);

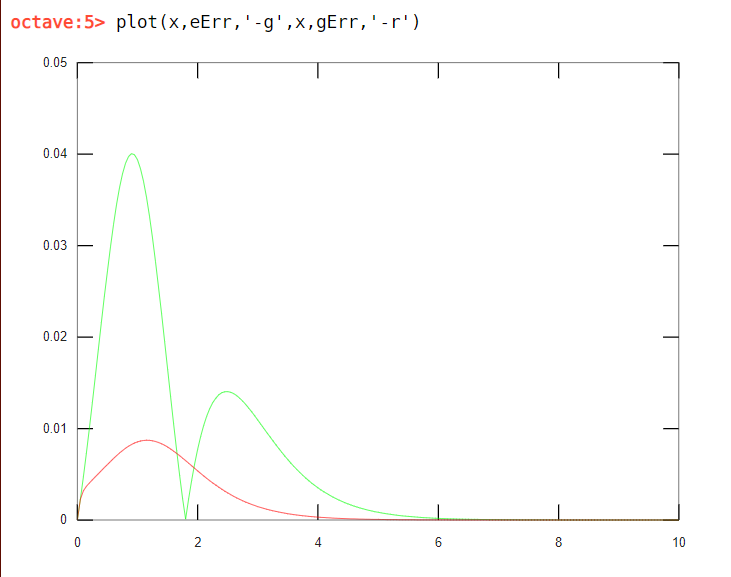
return 1.8f \* y \* tmp / (1.8f + 0.3f \* y \* (tmp - 1));

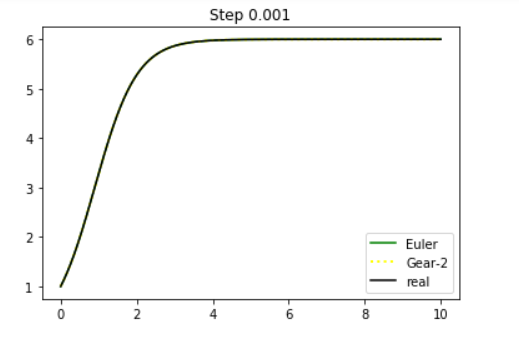
}

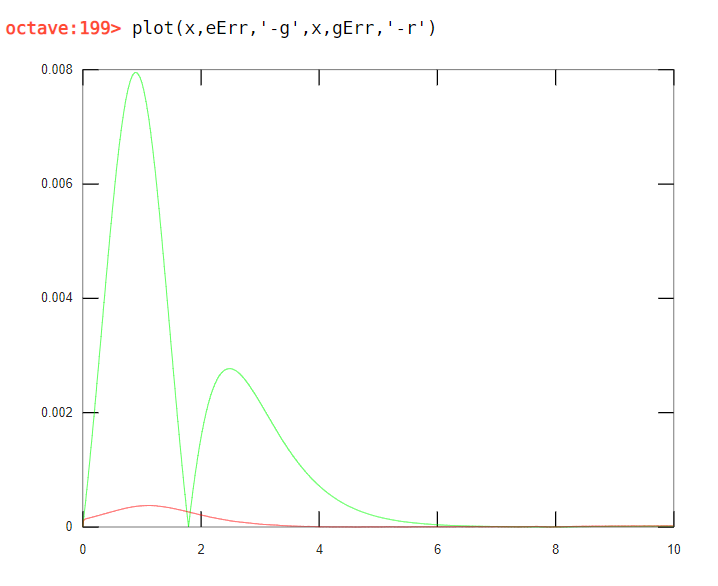
}

1. Графическое сравнение методов









1. Выводы

В данной лабораторной работе были использованы два метода решения дифференциальных уравнений явный и неявный

Метод Гира дал более точное решение. Его недостаток состоит в затрате времени на каждую итерацию. Это связано с попыткой совместить свойства явного метода и встроить в тело алгоритма цифровой сглаживающий фильтр 2-го порядка

Если же сравнивать полученные графики, видно, что эталонное и численное решение выданные явным методом совпадают. Это и есть доказательство его неустойчивости. При очень большом шаге кривые пойдут вразнос. Для борьбы с неустойчивостью используют схему "прогноз - коррекция". Аналитическое решение метода Гира 2-го порядка также сходится с графиком кривой, полученном предыдущим методом, тем не менее кривая для численного решения резко отличается и после тройки выгнута в противоположную сторону.

Увеличивая точность, мы получаем более реальных решения, но платим за это ресурсами процессора вычислительной машины и своим временем

Конкретно, после указания шага 0.001 машина зависла и заставила ожидать решение около 1 минуты. Это может серьезно повлиять на крупные технологические процессы. Не всегда выгодно рисковать временем вычислений, особенно если их много, поэтому точность должна соответствовать желаемому результату, однако не превосходить его