Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Саратовский государственный технический университет

имени Гагарина Ю.А.»

Институт электронной техники и приборостроения

Кафедра Информационная безопасность автоматизированных систем

Специальность 10.05.03 Информационная безопасность автоматизированных систем

Практическая работа №2

Тема «Метод Рунге-Кутта для решения задачи Коши»

|  |  |
| --- | --- |
|  | Выполнил: студент 3 курса  учебной группы с-ИБС32  очной формы обучения  Солодилов В.В.  Проверил: доцент кафедры ИБС Кожанова Е.Р. |

Саратов 2021

**Цель работы**: сформировать практические навыки решения задачи Коши для линейных обыкновенных дифференциальных уравнения первого и второго порядков методом Эйлера с их программной реализацией.

**Краткие теоретические сведения**

*Методы Рунге-Кутта* — большой класс численных методов решения задачи Коши для обыкновенных дифференциальных уравнений и их систем. К классу методов Рунге-Кутта относятся явный метод Эйлера и модифицированный метод Эйлера с пересчётом, которые представляют собой соответственно методы первого и второго порядка точности. Существуют стандартные явные методы третьего порядка точности, не получившие широкого распространения. Наиболее часто используется и реализован в различных математических пакетах классический метод Рунге-Кутта, имеющий четвёртый порядок точности.

Пусть дано дифференциальное уравнение первого порядка

с начальным условием

В окрестностях точки функцию разложим в ряд Тейлора:

который можно применить для приближенного определения искомой функции . Для уменьшения погрешности метода интегрирования дифференциального уравнения необходимо учитывать большее количество членов ряда. Однако при этом возникает необходимость аппроксимации производных от правых частей дифференциального уравнения.

Основная идея методов Рунге-Кутты заключается в том, что производные аппроксимируются через значения функции в точках на интервале которые выбираются из условия наибольшей близости алгоритма к ряду Тейлора. В зависимости от старшей степени *h*, с которой учитываются члены ряда, построены вычислительные схемы Рунге-Кутты разных порядков точности.

Так, например, общая форма записи метода Рунге-Кутты второго порядка следующая:

(1)

где .

Решение ОДУ, полученное по этой схеме, имеет погрешность Для параметра а наиболее часто используют значения

Рассмотрим первый вариант метода Рунге-Кутта второго порядка. При формула (1) примет вид:

Формулу (2) можно представить в виде следующей схемы:

Это метод Рунге-Кутты второго порядка (1-й вариант), или исправленный метод Эйлера.

Геометрически процесс нахождения точки можно проследить по рис.1.

По методу Эйлера находится точка  , лежащая на прямой . В этой точке снова вычисляется тангенс угла наклона касательной (прямая )*.*

Усреднение двух тангенсов дает прямую . Проводим через точку прямую *L,* параллельную *.* Точка, в которой прямая *L* пересечется с ординатой  будет искомой точкой *.*

Тангенс угла наклона прямой равен

Уравнение прямой L запишется в виде:

(5)

тогда в точке с учетом (4) получим решение:

(6)

Формула описывает метод Рунге-Кутта второго порядка при а = 0,5.

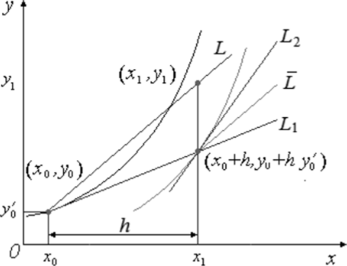


Рис.1

В случае второго варианта метода Рунге-Кутта второго порядка принимают при Тогда формула (1) принимает вид:

Представим формулу (7) в виде схемы:

Это метод Рунге-Кутта второго порядка (2-й вариант), или модифицированный метод Эйлера. Геометрическая интерпретация метода Рунге-Кутта при представлена на рис.2.

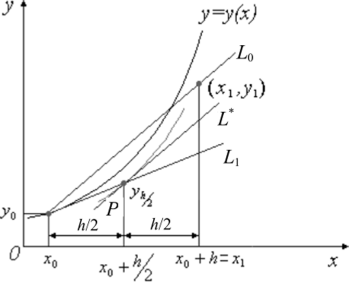


Рис.2

Формула описывает метод Рунге-Кутта второго порядка при .

Метод Рунге-Кутта четвертого порядка описывается системой следующих соотношений:

где

Геометрическая интерпретация метода представлена на рис.3.

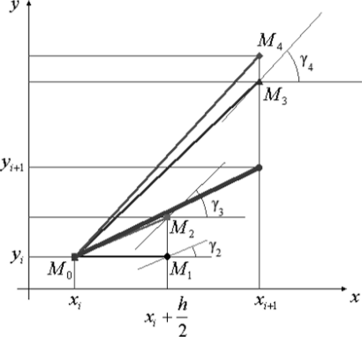


Рис.3

Задача Коши для дифференциального уравнения второго порядка сводится к системе двух уравнений первого порядка.

Получается, метод Рунге-Кутта второго порядка для дифференциального уравнения второго порядка примет вид:

(11)

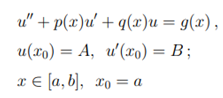
Метод Рунге-Кутта четвертого порядка для дифференциального уравнения второго порядка примет вид:

(12)

*Задание 1*. Решить задачу Коши для обыкновенного дифференциального уравнения первого порядка методом Рунге – Кутта второго и четвертого порядков точности. Сравнить результаты между собой и с результатами, полученными методом Эйлера:

*Задание 2*. Решить задачу Коши для обыкновенного дифференциального уравнения второго порядка (таблица 2) методом Рунге – Кутта второго и четвертого порядков точности. Сравнить результаты между собой и с результатами, полученными методом Эйлера:

Метод решения:

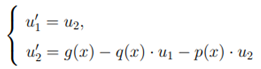


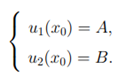
Пусть





Тогда задачу Коши запишем:

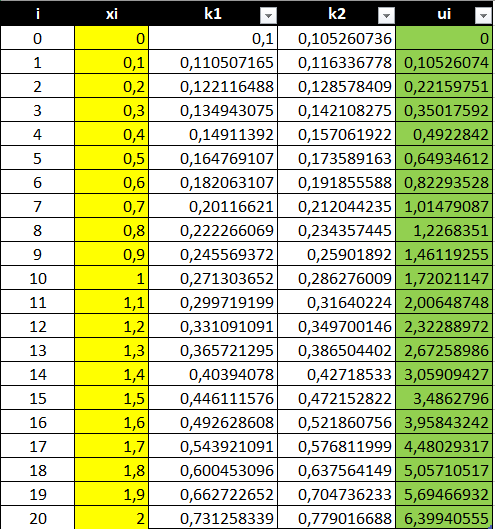


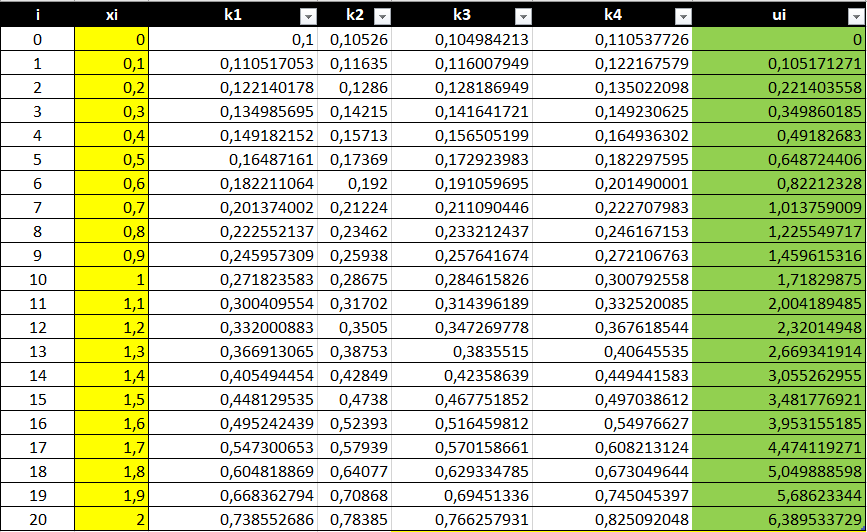


**Расчёт заданий**

*Расчёт Задания 1*

Для решения ДУ методом Рунге-Кутта второго и четвертого порядков на основе заданных значений аргумента были найдены значения функции, которые были оформлены в виде таблицы:





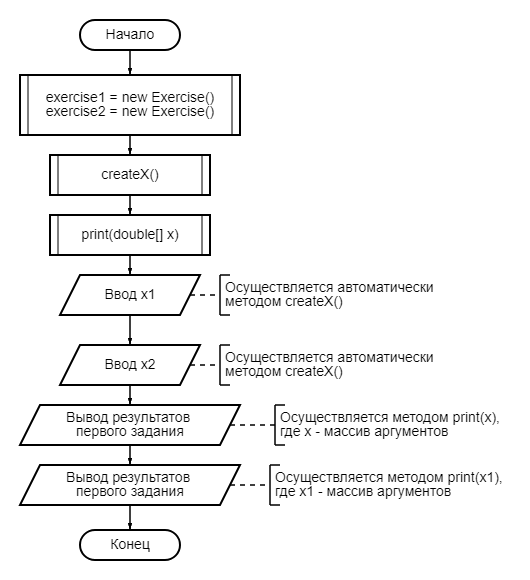
По полученным и ранее известным значениям функции и аргумента были построены графики функции:

На основе полученных результатов можно сказать, что наиболее точным является метод Рунге-Кутта четвертого порядка точности.

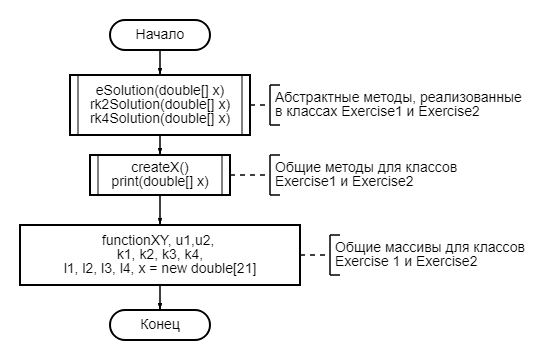
*Задание 2.*

//

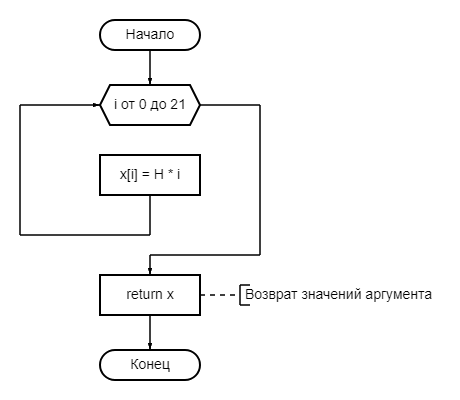
**Блок-схемы**



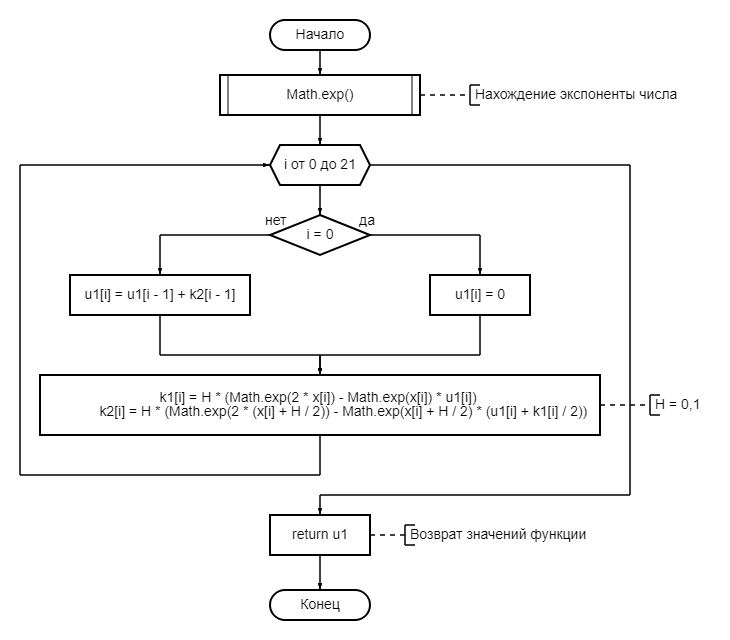
Класс Main()



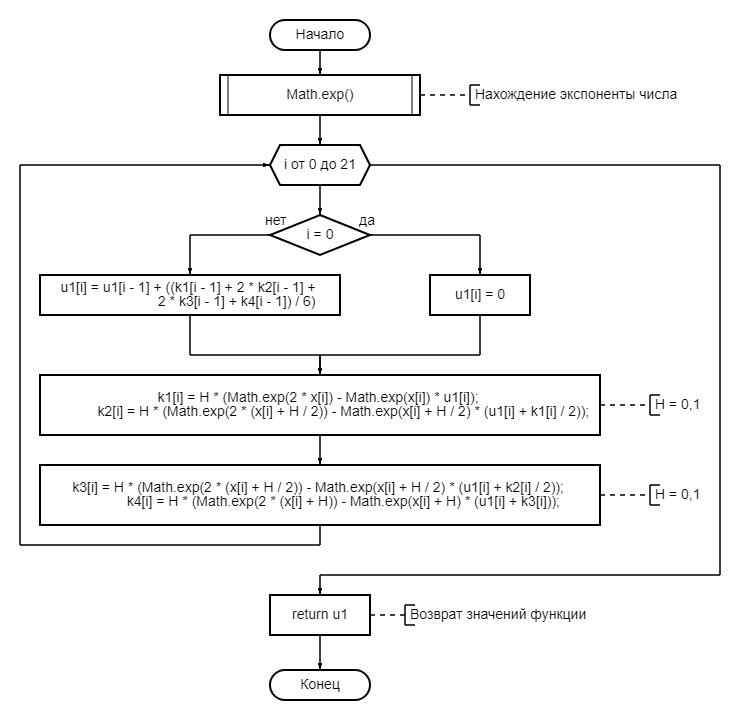
Класс Exercise()



Метод createX()



Метод rk2Solution() класса Exercise1()



Метод rk4Solution() класса Exercise1()

**Листинг программы**

*//Main.java*

package com.company;  
  
public class Main {  
 public static void main(String[] args) {  
 Exercise exercise1 = new Exercise1();  
 double[] x = exercise1.createX();  
 Exercise exercise2 = new Exercise2();  
 double[] x1 = exercise2.createX();  
  
 System.out.println("Задание 1");  
 exercise1.print(x);  
 System.out.println("\nЗадание 2");  
 exercise2.print(x1);  
 }  
}

*//Exercise.java*

package com.company;  
  
public abstract class Exercise {  
 protected final double H = 0.1;  
 protected double[] functionXY = new double[21];  
 protected double[] u1 = new double[21];  
 protected double[] u2 = new double[21];  
 protected double[] k1 = new double[21];  
 protected double[] k2 = new double[21];  
 protected double[] k3 = new double[21];  
 protected double[] k4 = new double[21];  
 protected double[] l1 = new double[21];  
 protected double[] l2 = new double[21];  
 protected double[] l3 = new double[21];  
 protected double[] l4 = new double[21];  
 protected double[] x = new double[21];  
  
 abstract double[] rk2Solution(double[] x);  
 abstract double[] rk4Solution(double[] x);  
  
 public double[] createX() {  
 for (int i = 0; i < 21; i++) {  
 x[i] = H \* i;  
 }  
 return x;  
 }  
  
 public void print(double[] x) {  
 System.out.println("\tX\t\t\tМетод Эйлера \t\t\tМетод Рунге-Кутта 2-го порядка\t\tМетод Рунге-Кутта 4-го порядка");  
 for (int i = 0; i < 21; i++) {  
 System.out.format("%5f\t\t ", x[i]);  
 System.out.format("%5f\t\t\t\t\t\t\t", rk2Solution(x)[i]);  
 System.out.format("%5f\t\t\t\t\n", rk4Solution(x)[i]);  
 }  
 }  
}

*//Exercise1.java*

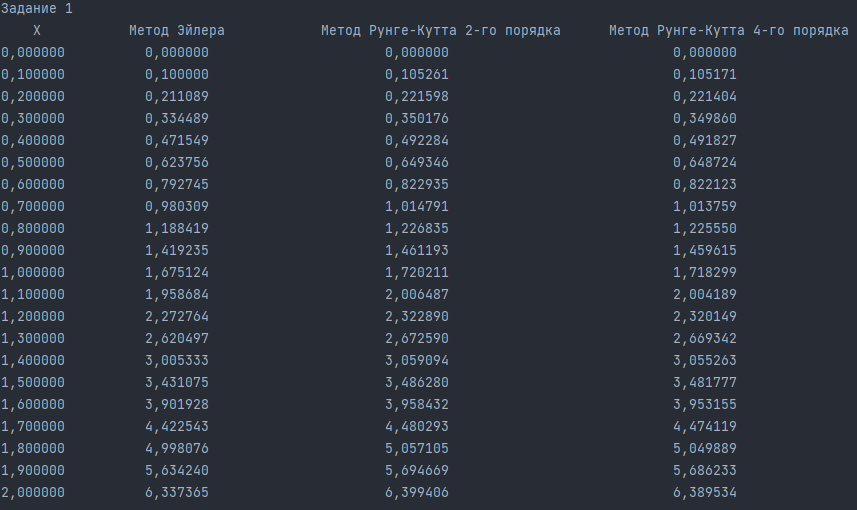
package com.company;  
  
public class Exercise1 extends Exercise {  
 @Override  
 public double[] rk2Solution(double[] x) {  
 for (int i = 0; i < 21; i++) {  
 if (i == 0) {  
 u1[i] = 0;  
 } else {  
 u1[i] = u1[i - 1] + k2[i - 1];  
 }  
  
 k1[i] = H \* (Math.exp(2 \* x[i]) - Math.exp(x[i]) \* u1[i]);  
 k2[i] = H \* (Math.exp(2 \* (x[i] + H / 2)) -  
 Math.exp(x[i] + H / 2) \* (u1[i] + k1[i] / 2));  
 }  
 return u1;  
 }  
  
 @Override  
 public double[] rk4Solution(double[] x) {  
 for (int i = 0; i < 21; i++) {  
 if (i == 0) {  
 u1[i] = 0;  
 } else {  
 u1[i] = u1[i - 1] + ((k1[i - 1] + 2 \* k2[i - 1] +  
 2 \* k3[i - 1] + k4[i - 1]) / 6);  
 }  
  
 k1[i] = H \* (Math.exp(2 \* x[i]) - Math.exp(x[i]) \* u1[i]);  
 k2[i] = H \* (Math.exp(2 \* (x[i] + H / 2)) -  
 Math.exp(x[i] + H / 2) \* (u1[i] + k1[i] / 2));  
 k3[i] = H \* (Math.exp(2 \* (x[i] + H / 2)) -  
 Math.exp(x[i] + H / 2) \* (u1[i] + k2[i] / 2));  
 k4[i] = H \* (Math.exp(2 \* (x[i] + H)) -  
 Math.exp(x[i] + H) \* (u1[i] + k3[i]));  
 }  
 return u1;  
 }  
}

*//Exercise2.java*

package com.company;  
  
public class Exercise2 extends Exercise {  
 @Override  
 double[] rk2Solution(double[] x) {  
 for (int i = 0; i < 21; i++) {  
 if (i == 0) {  
 u1[i] = 3;  
 u2[i] = 5;  
 } else {  
 u1[i] = u1[i - 1] + ((k1[i - 1] + k2[i - 1]) / 2);  
 u2[i] = u2[i - 1] + (H \* (6 \* Math.exp(x[i - 1]) +  
 3 \* u1[i - 1]));  
 }  
 k1[i] = H \* u2[i];  
 l1[i] = H \* (6 \* Math.exp(x[i]) + 3 \* u1[i]);  
 l2[i] = H \* (6 \* Math.exp(x[i] + H) + 3 \* (u1[i] + k1[i]));  
 k2[i] = H \* (((l1[i] + l2[i]) / 2) + l1[i]);  
 }  
 return u2;  
 }  
  
 @Override  
 double[] rk4Solution(double[] x) {  
  
 return u1;  
 }  
}

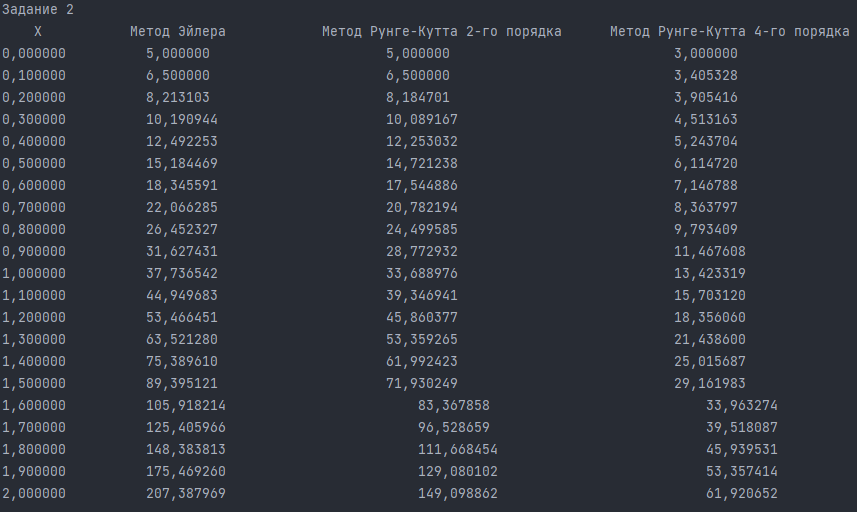
**Результат программы**

В результате выполнения программы были получены следующие результаты для *задания 1:*



Полученные значения полностью совпали с значениями, полученными при выполнении задания в Microsoft Office Excel.

Для задания 2 результат выполнения программы имеет вид:



**Выводы**

В результате выполнения практической работы были изучены способы решения дифференциальных уравнений первого и второго порядков численными методами Эйлера и Рунге-Кутта. Все вычисления были получены с использованием Microsoft Office Excel, а затем была написана программа на языке программирования Java, которая подтвердила полученные результаты. Итого, все задачи практической работы были выполнены, результат достигнут.