Национальный исследовательский университет ИТМО

Факультет ПИиКТ

Лабораторная работа №5
по вычислительной математике

Работу выполнила:

Тройникова Вероника

Группа: P3233

Преподаватель:

Перл О. В.

Санкт-Петербург

2022

1. Описание методов, расчетные формулы

Метод Эйлера – это одношаговый метод, позволяющий найти приближенное решение дифференциального уравнения первого порядка с заданным начальным условием (то есть решить задачу Коши). Формулы для данного метода получаются из разложения в ряд Тейлора искомой функции:

$$y(x) = y(x_0) + (x - x_0) * y'(x_0) + \cdots$$

При достаточно малых значениях шага h достаточно взять только первые два члена разложения:

$$y(x_0 + h) \approx y(x_0) + h * y'(x_0)$$

Из начальных условий $y'(x_0) = f(x_0, y_0)$ и $y(x_0) = y_0$ получается следующая формула:

$$y_1 = y_0 + h * f(x_0, y_0)$$

$$y_{i+1} = y_i + h * f(x_i, y_i)\,,\ x_{i+1} = x_i + h$$

Геометрически это можно интерпретировать следующим образом: искомая функция заменяется ломаной линией, состоящей из отрезков касательных к функции в точках x_i.

Порядок точности метода - O(h). Из-за низкой точности данный метод не часто используется на практике. Сложность данного алгоритма – O(n), где n – количество итераций (то есть $(x_n – x_0)/h$).

2. Блок-схема численного метода

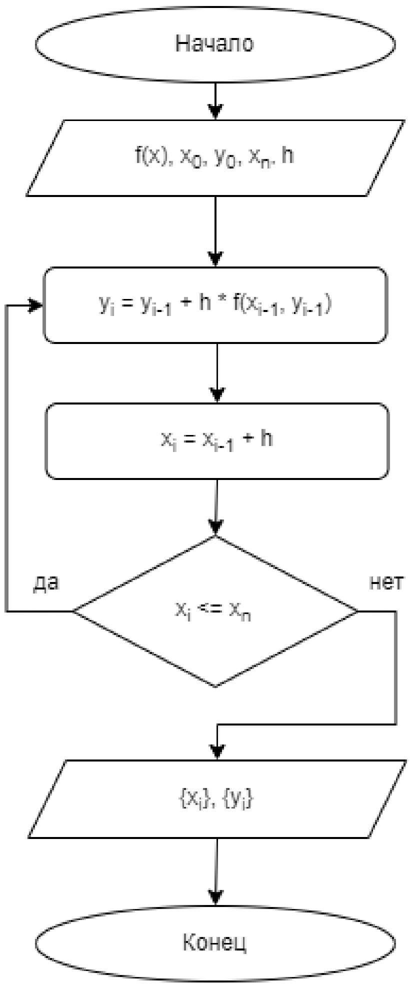


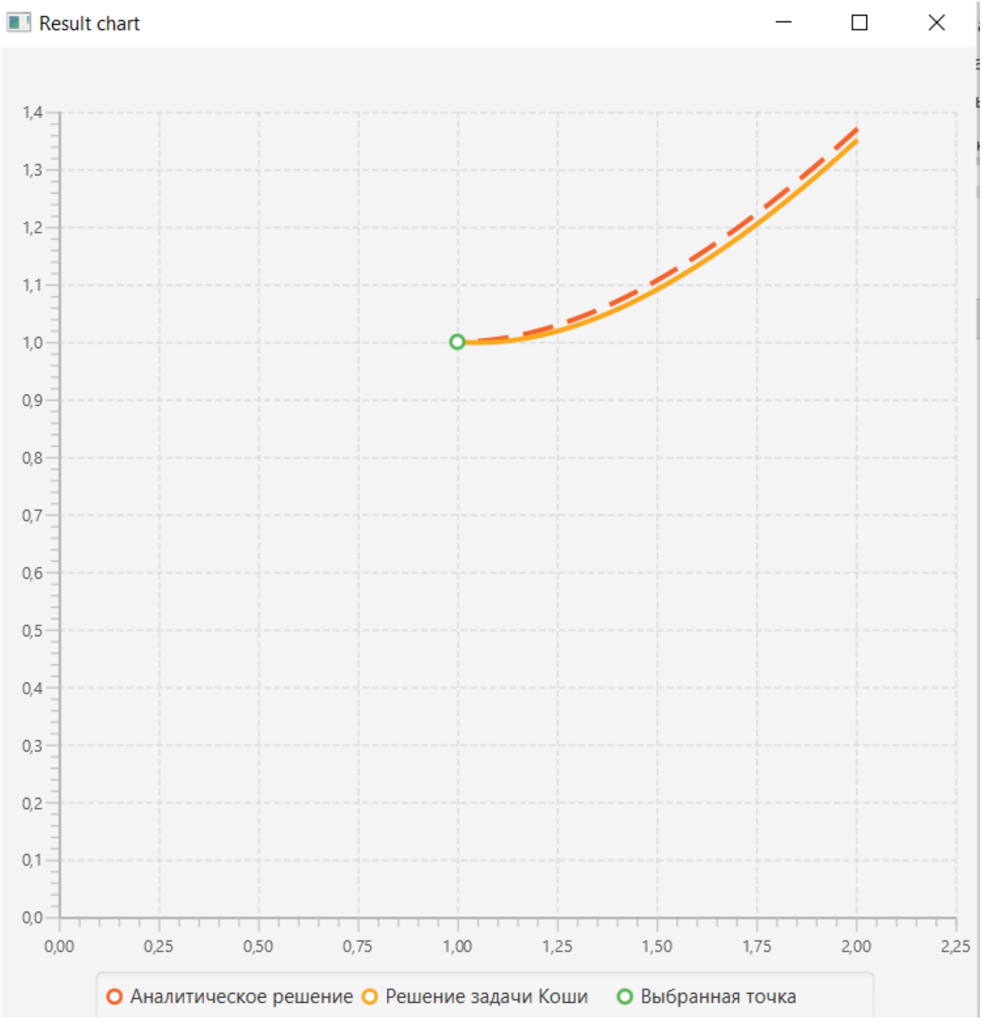
Рисунок 2 “Блок-схема метода Эйлера”

3. Листинг реализованного численного метода программы

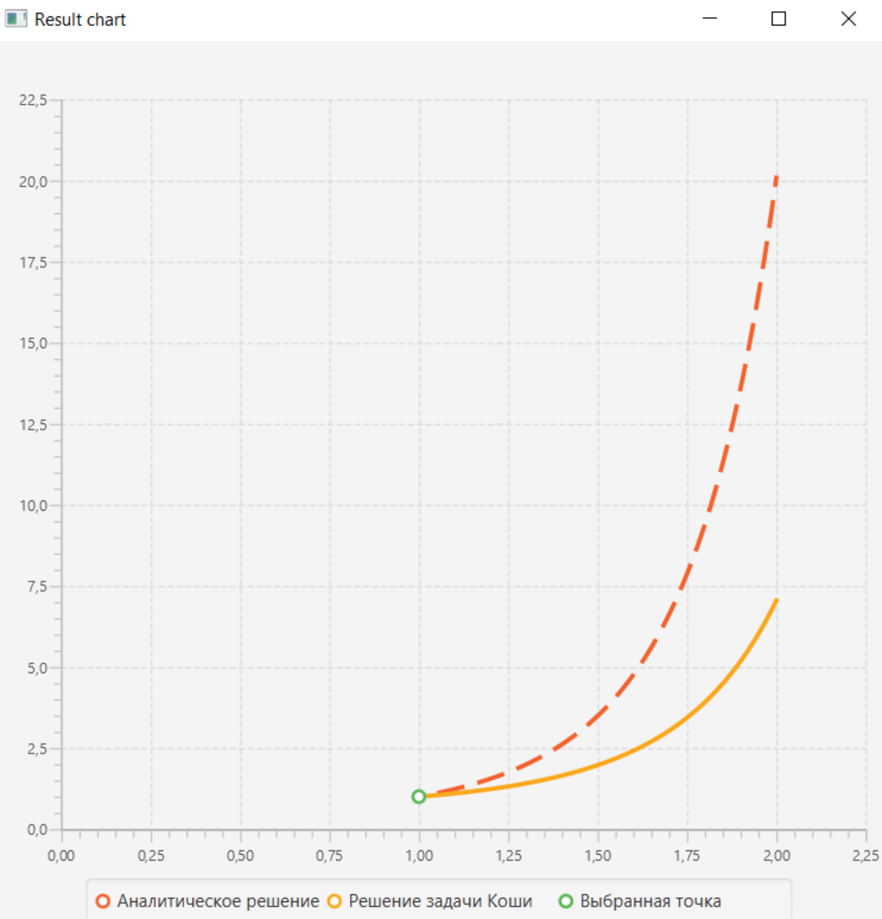
```java
public class MethodEuler {

    private final int maxDotsCount = 20;

    public DiscreteFunction findAnswer(double x0, double y0, double xn, Function function, double h) {
        int count = (int) Math.ceil((xn - x0) / h);
        count = Math.min(count, maxDotsCount);
        h = (xn - x0) / count;
        ArrayList<Double> y = new ArrayList<>();
        ArrayList<Double> x = new ArrayList<>();
        y.add(y0);
        x.add(x0);
        double currentX = x0;
        double currentY = y0;
        for (int i = 0; i < count; i++) {
            currentY = currentY + h * function.calculate(new double[]{currentX, currentY});
            y.add(currentY);
            currentX += h;
            x.add(currentX);
        }
        return new DiscreteFunction(x, y);
    }
}
```

4. Примеры и результаты работы программы на разных данных

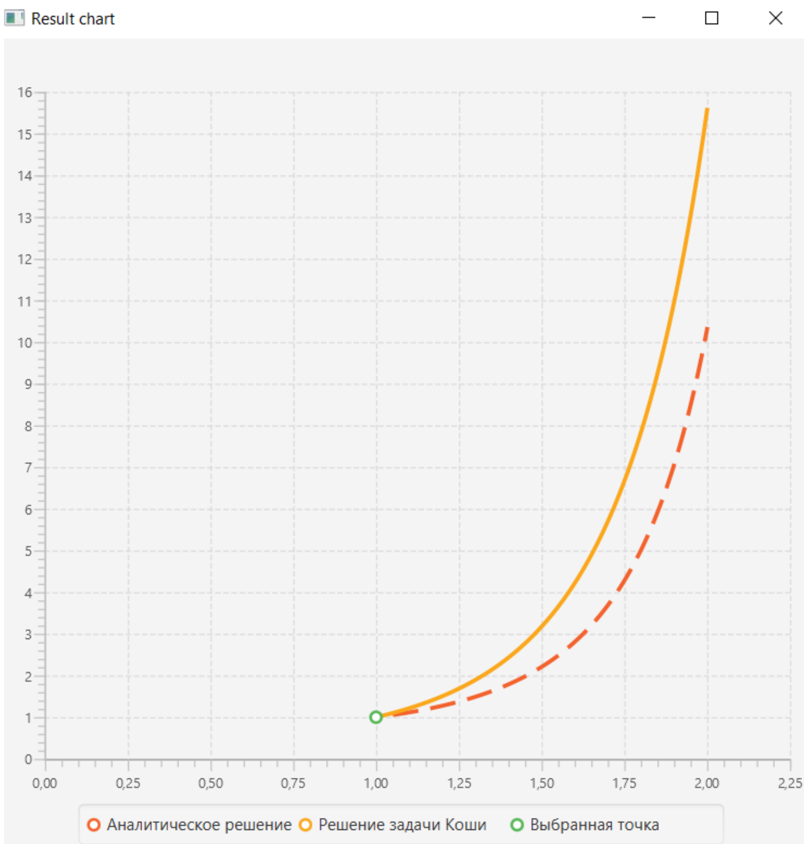
```
----Метод Эйлера----
Функции:
1) x - y
2) x^2 * y
3) 2xy
4) x^2 - 2y
Введите номер выбранной функции:
1
Введите x0:
1
Введите y0:
1
Введите xn:
2
Введите шаг:
0.1
Задача Коши решена
```



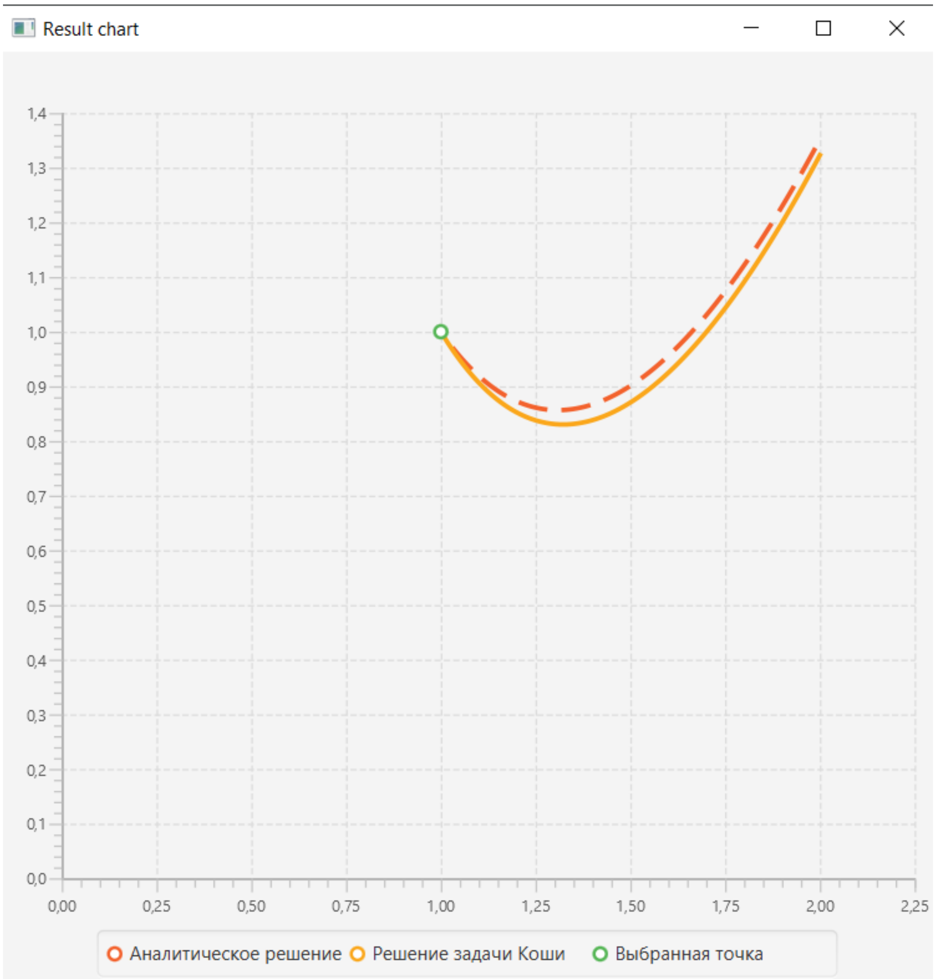
```
----Метод Эйлера----
Функции:
1) x - y
2) x^2 * y
3) 2xy
4) x^2 - 2y
Введите номер выбранной функции:
2
Введите x0:
1
Введите y0:
1
Введите xn:
2
Введите шаг:
0.1
Задача Коши решена
```



```
----Метод Эйлера----
Функции:
1) x - y
2) x^2 * y
3) 2xy
4) x^2 - 2y
Введите номер выбранной функции:
3
Введите x0:
1
Введите y0:
1
Введите xn:
2
Введите шаг:
0.05
Задача Коши решена
```



```
----Метод Эйлера----
Функции:
1) x - y
2) x^2 * y
3) 2xy
4) x^2 - 2y
Введите номер выбранной функции:
4
Введите x0:
1
Введите y0:
1
Введите xn:
2
Введите шаг:
0.07
Задача Коши решена
```



5. Вывод

В ходе выполнения данной лабораторной работы были рассмотрены одношаговые и многошаговые методы решения задачи Коши.

Суть одношаговых методов заключается в том, что они только по одной предыдущей точке находят следующую точку. Многошаговые методы, в свою очередь, для нахождения следующей точки анализирует несколько предыдущих точек. Важно отметить, что для вычисления начальных точек для многошаговых методов используются одношаговые.

Преимуществом метода Эйлера является высокая скорость нахождения ответа, однако данный метод имеет низкую точность (порядок точности – O(h)) и является неустойчивым (то есть имеет тенденцию к накоплению ошибок – данный факт можно увидеть на вышепредставленных примерах). Усовершенствованный метод Эйлера требует вычисления значения функции на каждом шаге дважды (метод Эйлера – только один), но при этом имеет большую точность (порядок точности – O(h^2)). Метод Рунге-Кутты (4-го порядка) является более устойчивым (так как на каждом этапе новое значение ‘y’ вычисляется на основе нескольких рассчитанных значений) и его точность выше (порядок точности –

$O(h^4)$). Однако, в методе Рунге-Кутты 4-го порядка на каждом шаге значение функции приходится вычислять 4 раза.