Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Методы численного анализа

**ОТЧЁТ**

к лабораторной работе

на тему

Методы Эйлера и Рунге-Кутта

Выполнил: студент группы 253502

Шишко Виктор Викторович

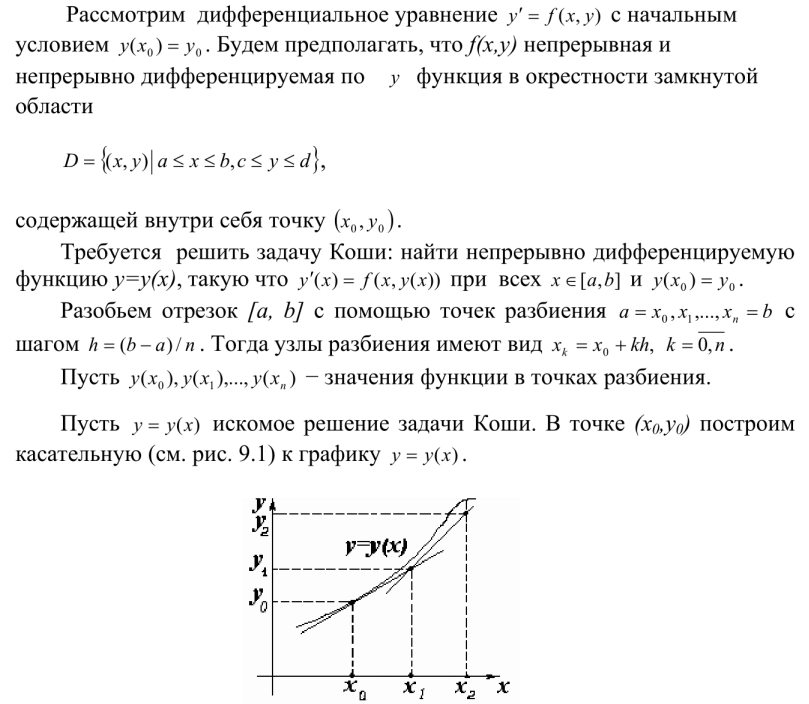
Проверил: Анисимов Владимир Яковлевич

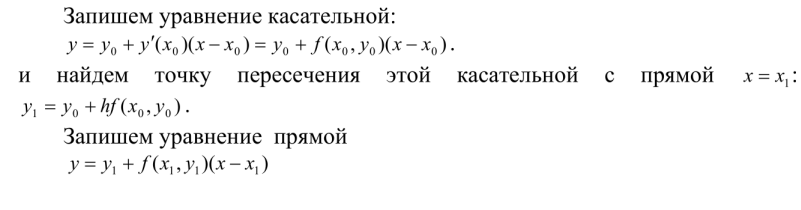
Минск 2023

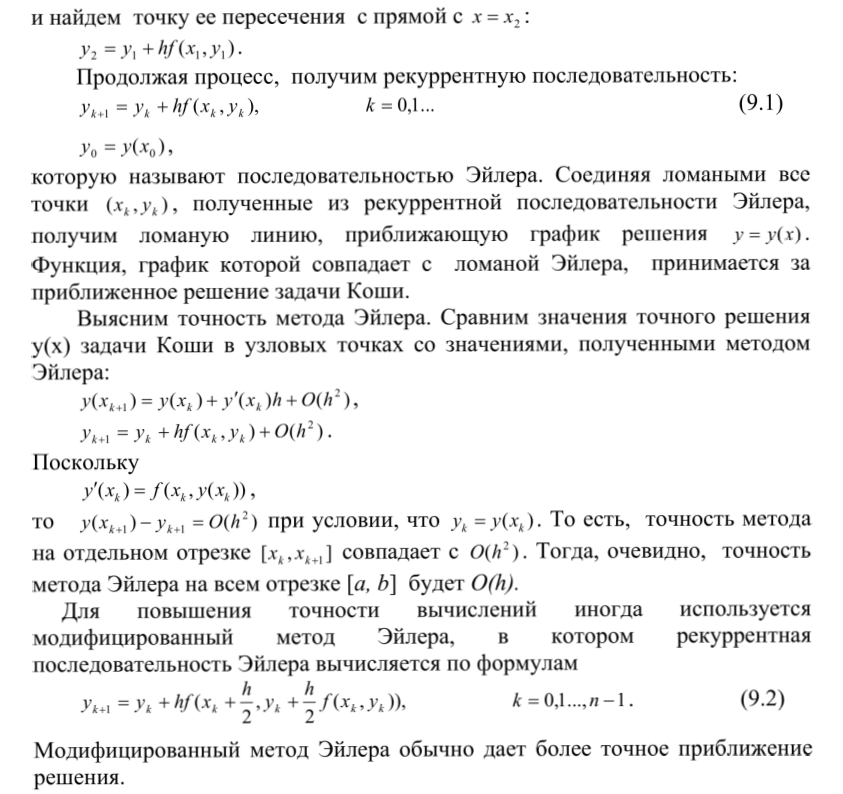
Цели выполнения задания

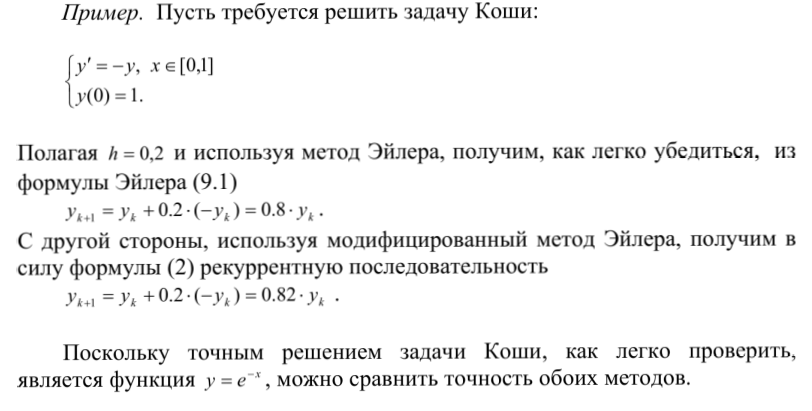
1. Изучить численное решение задачи Коши для обыкновенных дифференциальных уравнений методами Эйлера и Рунге-Кутта.
2. Реализовать численное решение задачи Коши для обыкновенных дифференциальных уравнений методами Эйлера и Рунге-Кутта.

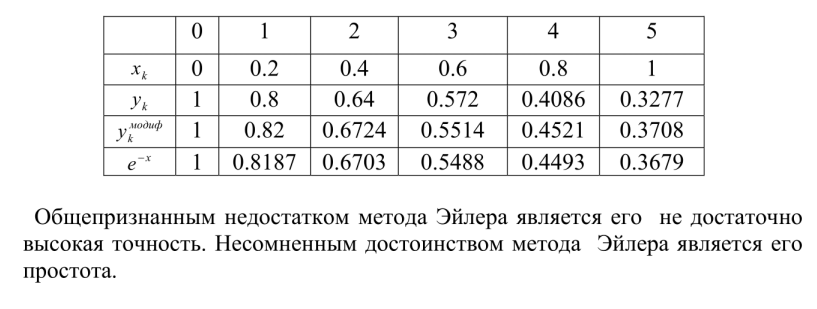
**Краткие теоретические сведения**

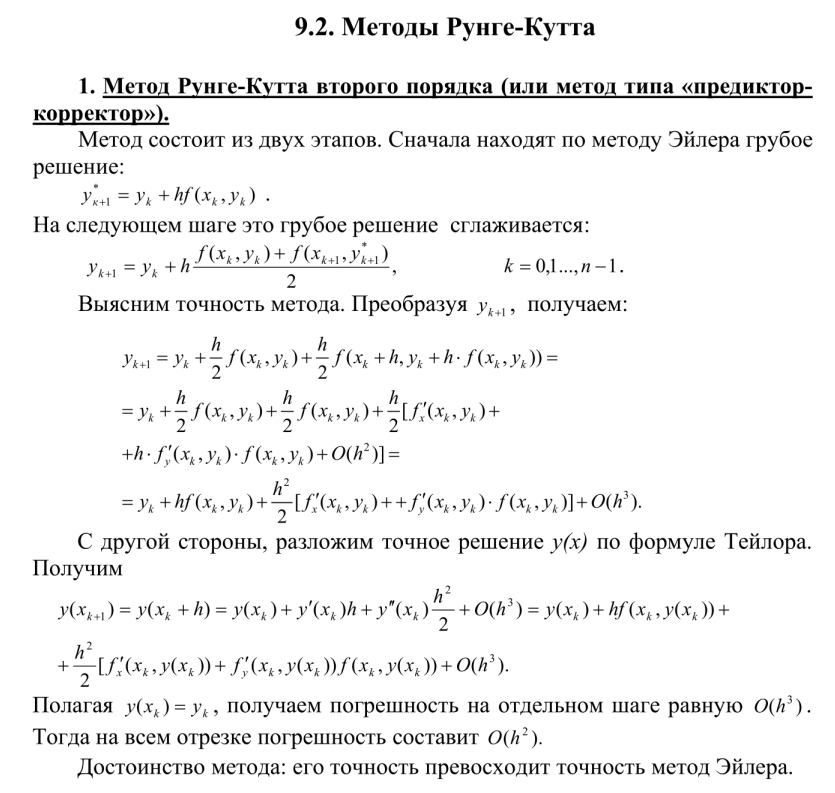
****

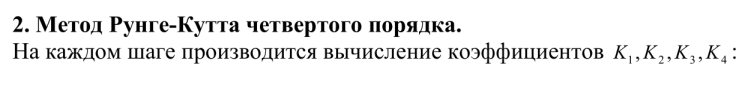
****

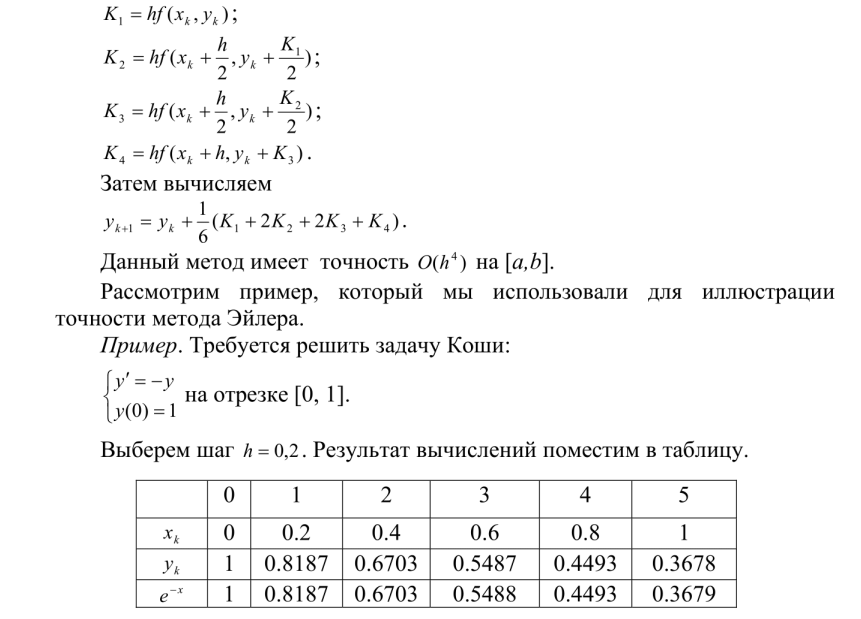
****

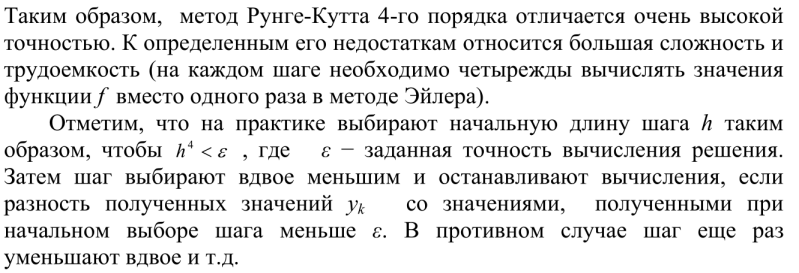
****

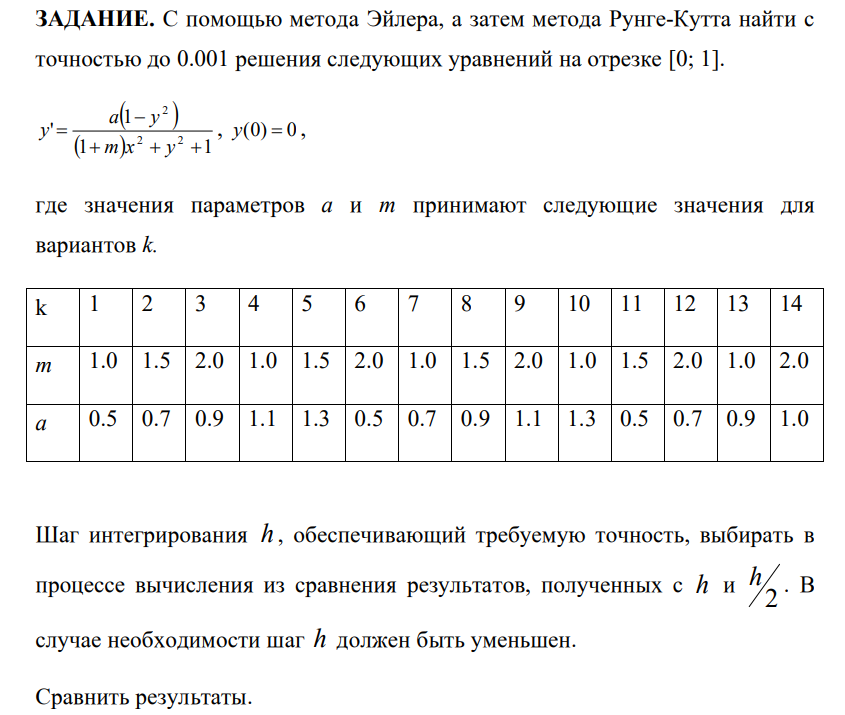
****

****

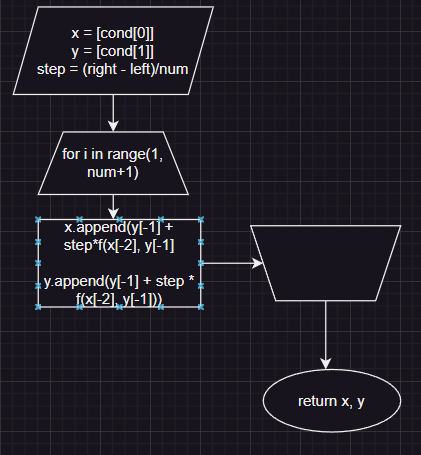
****

****

****

****

**Программная реализация**

****

def f(x, y):

    return a \* (1 - y \*\* 2) / ((1 + m) \* x \*\* 2 + y \*\* 2 + 1)

def euler(num):

    x = [cond[0]]

    y = [cond[1]]

    step = (right - left) / num

    for i in range(1, num + 1):

        x.append(left + step \* i)

        y.append(y[-1] + step \* f(x[-2], y[-1]))

    return x, y

def euler\_modified(num):

    x = [cond[0]]

    y = [cond[1]]

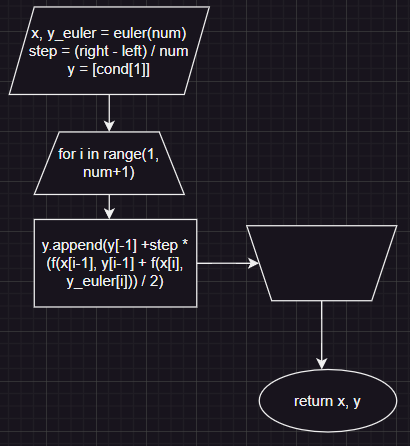
    step = (right - left) / num

    for i in range(1, num + 1):

        x.append(left + step \* i)

        y.append(y[-1] + step \* f(x[-2] + step / 2, y[-1] + step / 2 \* f(x[-2], y[-1])))

    return x, y



def runge\_kutta2(num):

    x, y\_euler = euler(num)

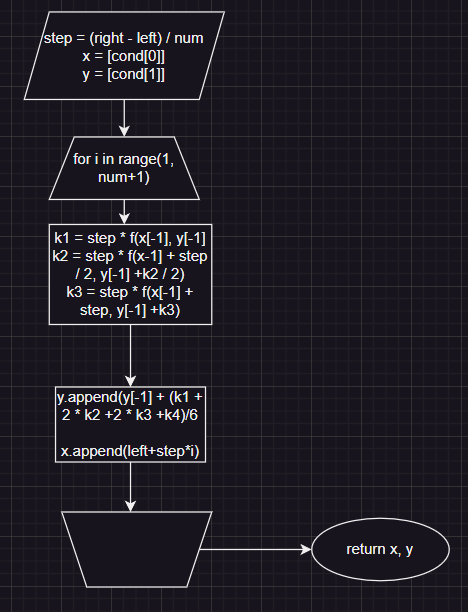
    step = (right - left) / num

    y = [cond[1]]

    for i in range(1, num + 1):

        y.append(y[-1] + step \* (f(x[i-1], y[-1]) + f(x[i], y\_euler[i])) / 2)

    return x, y



def runge\_kutta4(num):

    step = (right - left) / num

    x = [cond[0]]

    y = [cond[1]]

    for i in range(1, num + 1):

        k1 = step \* f(x[-1], y[-1])

        k2 = step \* f(x[-1] + step / 2, y[-1] + k1 / 2)

        k3 = step \* f(x[-1] + step / 2, y[-1] + k2 / 2)

        k4 = step \* f(x[-1] + step, y[-1] + k3)

        y.append(y[-1] + (k1 + 2 \* k2 + 2 \* k3 + k4) / 6)

        x.append(left + step \* i)

    return x, y

def get\_difference(sol1, sol2):

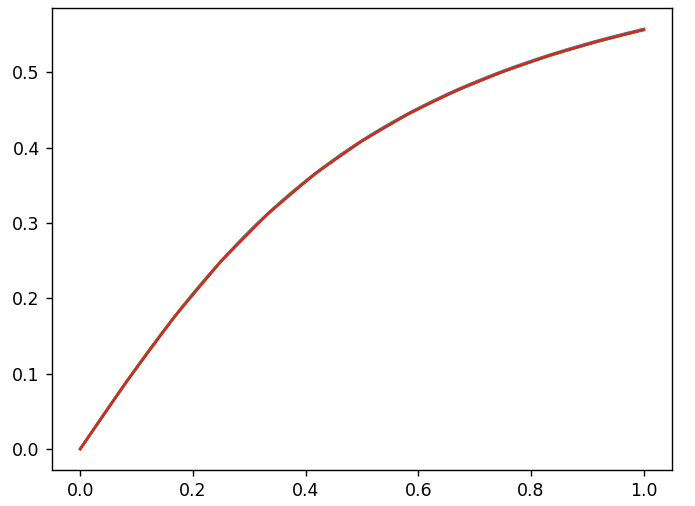
    max\_ = 0.0

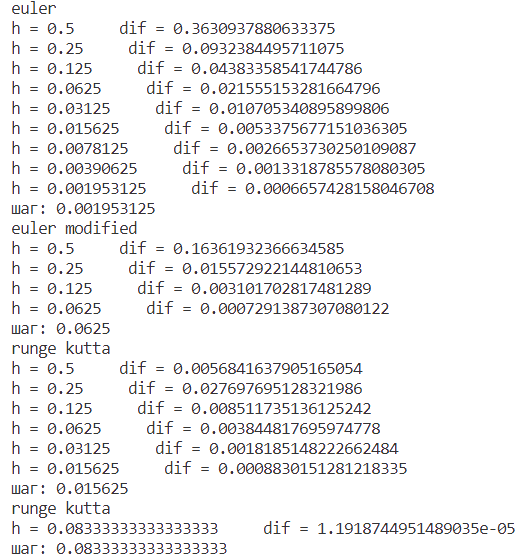
    for i in range(len(sol1)):

        max\_ = max(max\_, abs(sol1[i] - sol2[2\*i]))

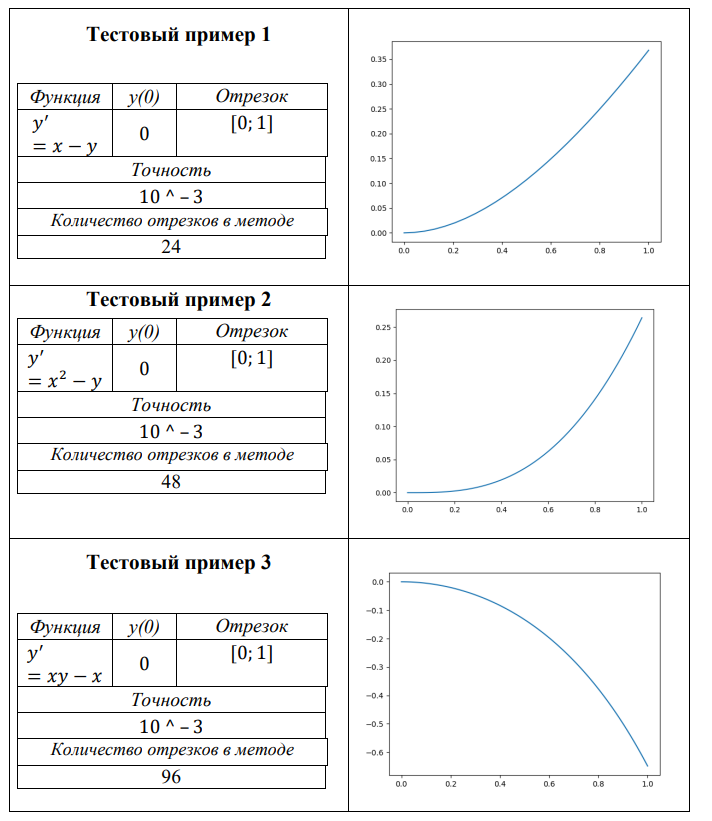
    return max\_

**Результаты программной реализации**

****

****

**Реализация тестовых примеров**

****

**Выводы**

Таким образом, в ходе выполнения лабораторной работы был освоен метод Адамса для решения задачи Коши для обыкновенных дифференциальных уравнений. Составлена компьютерная программа, на тестовых примерах проверена правильность её работы, с заданной точностью построен график решения дифференциального уравнения заданного варианта, по количеству необходимых для этого отрезков оценена трудоёмкость метода.