МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Информатика»

**Отчет по лабораторной работе №2.4**

**по дисциплине**

**«Базовые средства математических пакетов»**

Выполнил: студент гр. БЭИ2202

Кулешов А. С.

Вариант 14.

Проверил: доцент Шакин В. Н.

Москва, 2024г.

1. **Цель работы:** Изучение возможностей вычисления значений определенных интегралов, а также значений производной в точке с использованием функций Scilab.
2. **Индивидуальное задание**
3. ***Изучить материал учебника*** [1] (*п. 2.5).*
4. ***Выбрать*** *индивидуальный вариант задания из* ***табл. 2.5-1****.*
5. ***Создать сценарии*** *для выполнения практического задания.*
6. ***Найти*** *решение ОДУ на отрезке* **[x0;b]** *с шагом* **h** *с использованием функции* **ode**.
7. ***Создать*** *матрицу решений, записав в первый столбец аргумент, а во второй - решение, полученное с использованием функций* **ode***.*
8. ***Вывести*** *полученную таблицу по столбцам.*
9. ***Построить*** *график полученного решения ОДУ.*
10. ***Выбрать*** *индивидуальный вариант задания из* ***табл. 2.5-2****.*
11. ***Создать сценарий*** *для решения системы ОДУ или решения ОДУ 2-го порядка согласно вашему варианту.*
12. ***Найти*** *решение системы ОДУ, вывести матрицу решения и построить график решения системы ОДУ.*
13. ***Спроектировать и реализовать приложение****:* ***«Решение обыкновенных дифференциальных уравнений»*** *для ввода данных и отображения результатов (по требованию преподавателя).*
14. ***Предоставить*** *результаты работы преподавателю и* ***ответить*** *на поставленные вопросы.*





***Оформить отчет*** *по выполненной работе.*

1. **Программный код**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from scipy.integrate import odeint

# Определение функции правой части ОДУ

def model(y, x):

    return np.cos(y) + x\*\*2

# Начальное условие

y0 = 1

# Отрезок [x0; b] и шаг h

x0 = 1

b = 6

h = 0.5

# Создание массива точек x на отрезке [x0; b] с шагом h

x\_values = np.arange(x0, b + h, h)

# Решение ОДУ с помощью odeint

solution = odeint(model, y0, x\_values)

# Создание DataFrame для хранения решения

solution\_df = pd.DataFrame({'x': x\_values, 'y': solution[:, 0]})

# Вывод таблицы с решением

print(solution\_df)

# Построение графика

plt.plot(x\_values, solution, label='Решение ОДУ')

plt.xlabel('x')

plt.ylabel('y')

plt.title('Решение ОДУ y` = cos(y) + x^2')

plt.legend()

plt.grid(True)

def system\_model(y, x):

    y1, y2 = y

    return [y2, 2 \* np.exp(x) + 2 \* y2 - y1]

# Начальные условия

y0 = [2, 5]

# Отрезок [x0; b] и шаг h

x0 = 0

b = 1

h = 0.1

# Создание массива точек x на отрезке [x0; b] с шагом h

x\_values = np.arange(x0, b + h, h)

# Решение системы ОДУ с помощью odeint

solution\_system = odeint(system\_model, y0, x\_values)

# Создание DataFrame для хранения решения

solution\_df\_system = pd.DataFrame({'x': x\_values, 'y1': solution\_system[:, 0], 'y2': solution\_system[:, 1]})

# Вывод таблицы с решением

print(solution\_df\_system)

# Построение графика

plt.plot(x\_values, solution\_system[:, 0], label='y1')

plt.plot(x\_values, solution\_system[:, 1], label='y2')

plt.xlabel('x')

plt.ylabel('y')

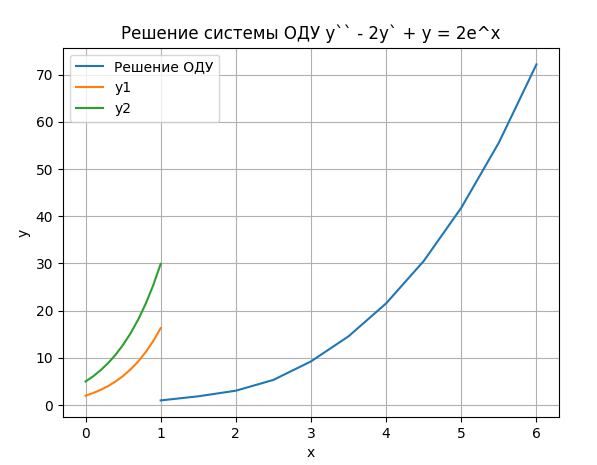
plt.title('Решение системы ОДУ y`` - 2y` + y = 2e^x')

plt.legend()

plt.grid(True)

plt.show()

1. **Полученные графики**

  
  
Рисунок 1 – Полученный график

1. **Полученные результаты**

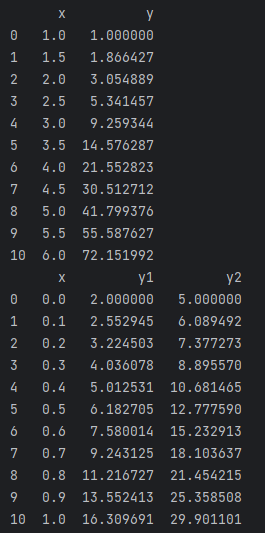
****

Рисунок 2 – Полученный результат