Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический
университет имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»
КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 2

Дисциплина Моделирование.
Тема ОДУ. Задача Коши

Студент Куприй А. А.
Группа ИУ7-63Б
Преподаватель Градов В. М.

Москва, 2020 г.

ВВЕДЕНИЕ

Цель: изучить методы Рунге-Кутта 2-го и 4-го порядка для решения системы дифференциальных уравнений.

1 Теоретические сведения

Дан разрядный контур:

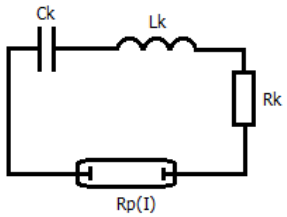

Рисунок 1.1 — Схема контура.

Получена система дифференциальных уравнений:

$$\begin{cases} L_{\text{к}}\frac{dI}{dt} + (R_{\text{к}} + R_p)I - U_c = 0 \\ C_{\text{к}}\frac{dU_c}{dt} = -I \end{cases}$$

Необходимо решить систему и построить графики $I(t), U_c(t), I \cdot R_p(t), R_p(t), T_0(t)$.

Сопротивление газоразрядной трубки, находится в зависимости от силы тока:

$$R_p(I) = \frac{l_{\text{э}}}{2\pi R^2 \int_0^1 \sigma(T(z))zdz}$$

Даны 2 таблицы, для нахождения T_0, m, σ

Система уравнений решается методом Рунге-Кутта 4 порядка для системы ОДУ.

$$y_{n+1} = y_n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6},$$

$$z_{n+1} = z_n + \frac{q_1 + 2q_2 + 2q_3 + q_4}{6}$$

, где

$$k_1 = h_n f(y_n, z_n), \quad q_1 = h_n \varphi(y_n)$$

$$k_2 = h_n f(y_n + \frac{k_1}{2}, z_n + \frac{q_1}{2}), \;\; q_2 = h_n \varphi(y_n + \frac{k_1}{2})$$

$$k_3 = h_n f(y_n + \frac{k_2}{2}, z_n + \frac{q_2}{2}), \;\; q_3 = h_n \varphi(y_n + \frac{k_2}{2})$$

$$k_4 = h_n f(y_n + k_3, z_n + q_3, \;\; q_4 = h_n \varphi(y_n + k_3)$$

2 Практическая часть

Ниже представленны листинги программы:

Листинг 2.1 — Интерполяция

```
def interpolation(val, tableVal, table):
    min_i = 0
    max_i = 0

    for i in range(len(tableVal)):
        if (val > tableVal[i]):
            max_i = i
        else:
            max_i = i
            break

    if (0 == max_i):
        max_i = 1

    min_i = max_i - 1

    return table[min_i] + (table[max_i] - table[min_i]) / (tableVal[max_i]
        - tableVal[min_i]) * (val - tableVal[min_i])
```

Листинг 2.2 — Интегрирование методом трапеций

```
def funcInt(I, z):
    t0 = interpolation(I, table_I, table_T0)
    global gt0
    gt0 = t0
    m = interpolation(I, table_I, table_M)
    t = t0 + (tw - t0) * (z ** m)

    sigma = interpolation(t, table_T, table_Sigma)

    return sigma * z

def trapezoidInt(I):
    a = 0
    b = 1
    n = 100
    h = (b - a) / n

    s = (funcInt(I, a) + funcInt(I, b)) / 2

    x = 0
    for i in range(n - 1):
```

```
22            x = x + h
23            s = s + funcInt(I, x)
24
25        s = s * h
26        return s
```

Листинг 2.3 — Нахождение сопротивления

```
1  def Rp(le, R, I):
2      return le / (2 * pi * R ** 2 * trapezoidInt(I))
```

Листинг 2.4 — Решение системы уравнений методом Рунге-Кутта 2-го и 4-го порядка

```
1  def f(I, U, le, R, Lk, Rk):
2      global grp
3      grp = Rp(le, R, fabs(I))
4
5      return (U - (Rk + grp) * I) / Lk
6
7  def g(I, Ck):
8      return -I / Ck
9
10 def stepOrder4(I, U, le, R, Lk, hn, Rk, Ck):
11     k1 = f(I, U, le, R, Lk, Rk)
12     q1 = g(I, Ck)
13
14     k2 = f(I + hn * k1 / 2, U + hn * q1 / 2, le, R, Lk, Rk)
15     q2 = g(I + hn * k1 / 2, Ck)
16
17     k3 = f(I + hn * k2 / 2, U + hn * q2 / 2, le, R, Lk, Rk)
18     q3 = g(I + hn * k2 / 2, Ck)
19
20     k4 = f(I + hn * k3, U + hn * q3, le, R, Lk, Rk)
21     q4 = g(I + hn * k3, Ck)
22
23     return I + hn * (k1 + 2 * k2 + 2 * k3 + k4) / 6, U + hn * (q1 + 2 * q2
           + 2 * q3 + q4) / 6
24
25 def stepOrder2(I, U, le, R, Lk, hn, Rk, Ck):
26     k1 = f(I, U, le, R, Lk, Rk)
27     q1 = g(I, Ck)
28
29     k2 = f(I + hn * k1 / 2, U + hn * q1 / 2, le, R, Lk, Rk)
30     q2 = g(I + hn * k1 / 2, Ck)
31
```

```
32        return I + hn * ((1 - alpha) * k1 + alpha * k2), U + hn * ((1 - alpha)
              * q1 + alpha * q2)
```

На изображениях ниже представлены скриншот работы программы:

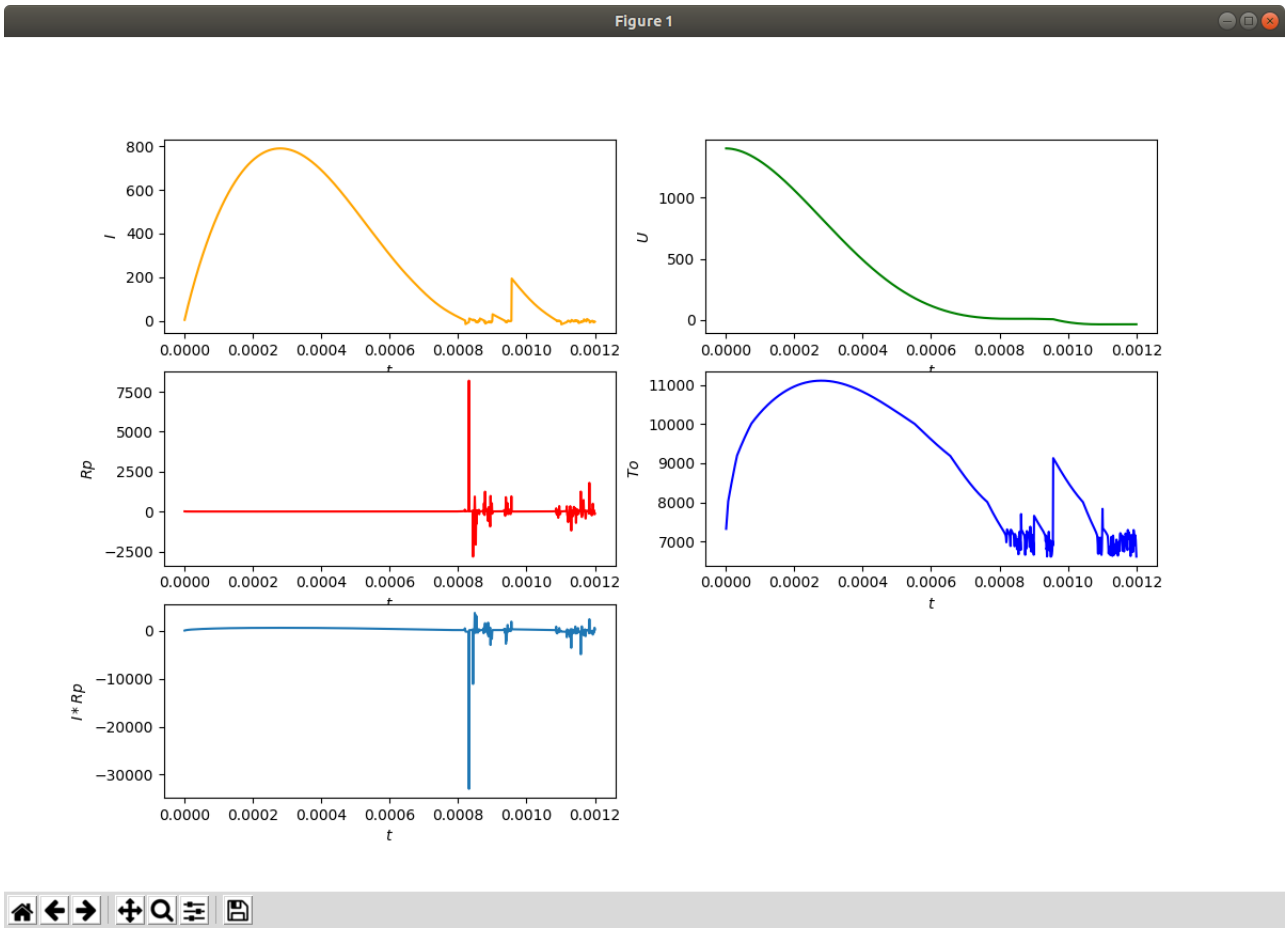

Рисунок 2.1 — Результат работы метода Рунге-Кутта 2-го порядка.

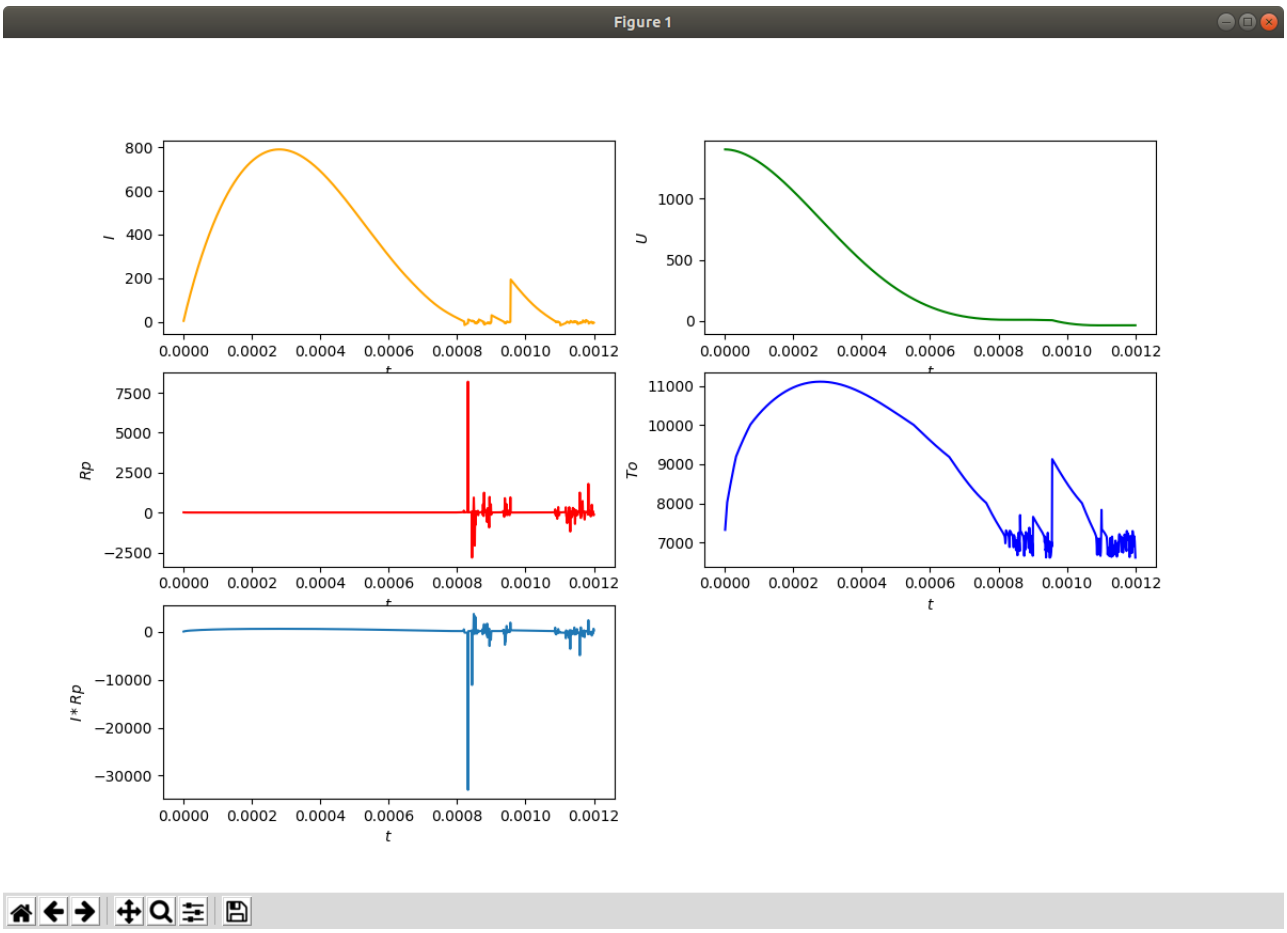

Рисунок 2.2 — Результат работы метода Рунге-Кутта 4-го порядка.

3 Заключение

В результате выполнения данной лабораторной работы были получены навыки применения численного метода Рунге-Кутта для решения системы ОДУ.