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

*ФАКУЛЬТЕТ «Информатика и системы управления»*

*КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»*

# О Т Ч Е Т

**по лабораторной работе №** 0 2

**Дисциплина:** ***Моделирование***

| | | | |
|---|---|---|---|
| Студент | ***ИУ7И-66Б*** | | **Нгуен Ф. С.** |
| | (Группа) | (Подпись, дата) | (И.О. Фамилия) |
| Преподаватель | | | **Градов В. М.** |
| | | (Подпись, дата) | (И.О. Фамилия) |

*Москва, 2021*

**Цель работы**: Получение навыков разработки алгоритмов решения задачи Коши при реализации моделей, построенных на системе ОДУ, с использованием метода Рунге-Кутта 4-го порядка точности.

## I. **Исходные данные**.

Задана система электротехнических уравнений, описывающих разрядный контур, включающий постоянное активное сопротивление $R_k$ , нелинейное сопротивление $R_p(I)$ , зависящее от тока I , индуктивность $L_k$ и емкость $C_k$ .

$$\begin{cases} \dfrac{dI}{dT} = \dfrac{U - \left(R_k + R_p(I)\right) I}{L_k}, \\ \dfrac{dU}{dI} = -\dfrac{I}{C_k} \end{cases} \tag{1}$$

Начальные условия: $t = 0$, $I = I_0$, $U = U_0$.

Здесь I, U - ток и напряжение на конденсаторе. Сопротивление $R_p$ рассчитать по формуле

$$R_p = \frac{l_p}{2\pi R^2 \int_0^1 \sigma\left(T(z)\right) z dz} \tag{2}$$

Для функции T(z) применить выражение $T(z) = T_0 + (T_w - T_0)z^m$ . Параметры $T_0$ , m находятся интерполяцией из табл.1 при известном токе I . Коэффициент электропроводности σ (T) зависит от T и рассчитывается интерполяцией из табл.2.

Таблица 1

| **I, A** | **To, K** | **m** |
|---|---|---|
| 0.5 | 6730 | 0.50 |
| 1 | 6790 | 0.5 |
| 5 | 7150 | 1.7 |
| 10 | 7270 | 3 |
| 50 | 8010 | 11 |
| 200 | 9180 | 32 |
| 400 | 10010 | 40 |

| | | |
|---|---|---|
| 800 | 11140 | 41 |
| 1200 | 12010 | 39 |

Таблица 2

| **T, K** | **σ, 1/Om cm** |
|---|---|
| 4000 | 0.03 |
| 5000 | 0.27 |
| 6000 | 2.05 |
| 7000 | 6.06 |
| 8000 | 12.0 |
| 9000 | 19.9 |
| 10000 | 29.6 |
| 11000 | 41.1 |
| 12000 | 54.1 |
| 13000 | 67.7 |
| 14000 | 81.5 |

Параметры разрядного контура:

$R$=0.35 см

$L_э$=12 см

$L_k$=187 10-6 Гн

$C_k$=268 10-6 Ф

$R_k$=0.25 Ом

$U_{co}$=1400 В

$I_o$=0..3 А

$T_w$=2000 К

II. Метод Рунге-Кутта четвертого порядка точности:

Дана система уравнений вида

$$\begin{cases} u'(x) = f(x,u,v) \\ v'(x) = \varphi(x,u,v) \\ u(\xi) = \eta_1 \\ v(\xi) = \eta_2 \end{cases} \tag{3}$$

Тогда

$$y_{n+1} = y_n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \tag{4}$$

$$z_{n+1} = z_n + \frac{g_1 + 2g_2 + 2g_3 + g_4}{6} \tag{5}$$

, где

$$k_1 = h_n f(x_n, y_n, z_n)$$

$$k_2 = h_n f\left(x_n + \frac{h_n}{2}, y_n + \frac{k_1}{2}, z_n + \frac{g_1}{2}\right)$$

$$k_3 = h_n f\left(x_n + \frac{h_n}{2}, y_n + \frac{k_2}{2}, z_n + \frac{g_2}{2}\right)$$

$$k_4 = h_n f(x_n + h_n, y_n + k_3, z_n + g_3)$$

$$g_1 = h_n \varphi(x_n, y_n, z_n)$$

$$g_2 = h_n \varphi\left(x_n + \frac{h_n}{2}, y_n + \frac{k_1}{2}, z_n + \frac{g_1}{2}\right)$$

$$g_3 = h_n \varphi\left(x_n + \frac{h_n}{2}, y_n + \frac{k_2}{2}, z_n + \frac{g_2}{2}\right)$$

$$g_4 = h_n \varphi(x_n + h_n, y_n + k_3, z_n + g_3)$$

III. Код программы:

1) Вычисление Rp

```
def calculateRp(I):
    R = data['R']
    Le = data['Le']
    integral = integrateSimpson(I)
    return Le / (2 * math.pi * R * R * integral)

def integrateSimpson(I):
    n = 40
    begin = 0
    end = 1
    width = (end - begin) / n
    result = 0
    for step in range(n):
        x1 = begin + step * width
        x2 = begin + (step + 1) * width
        result += (x2 - x1) / 6.0 * (siqmaFunc(I, x1) + 4.0 * siqmaFunc(I, 0.5 *
(x1 + x2)) + siqmaFunc(I, x2))
    return result
```

```
def siqmaFunc(I, z):
    m = interpolate(ItK, I, 0, 2)
    T0 = interpolate(ItK, I, 0, 1)
    Tz = getTz(T0, m, z)
    siqma = interpolate(Tsigma, Tz, 0, 1)

def getTz(T0, m, r):
    z = r
    Tw = data['Tw']
    return (Tw - T0) * math.pow(z, m) + T0

def interpolate(table, xValue, xIndex, yIndex):
    interpolateIndexFound = False
    x1 = 0
    x2 = 0
    y1 = 0
    y2 = 0
    yResult = 0
    for i in range(len(table) - 1):
        if (table[i][xIndex] <= xValue and table[i + 1][xIndex] >= xValue):
            y1 = table[i][yIndex]
            y2 = table[i + 1][yIndex]
            x1 = table[i][xIndex]
            x2 = table[i + 1][xIndex]
            interpolateIndexFound = True
    if (interpolateIndexFound):
        yResult = y1 + ((xValue - x1) / (x2 - x1)) * (y2 - y1)
    else:
        if (xValue < table[0][xIndex]):
            yResult = table[0][yIndex]
        if (xValue > table[len(table) - 1][xIndex]):
            yResult = table[len(table) - 1][yIndex]

    return yResult
```

2) Метод Рунге-Кутта четвертого порядка точности.

```
def functionPHI(t, I, U):
    Ck = data['Ck']
    return -1 / Ck * I


def functionF(t, I, U):
    Rp = calculateRp(I)
    Rk = data['Rk']
    Lk = data['Lk']
    return ((U - (Rk + Rp) * I) / Lk)


def RungeKutta4(xn, yn, zn, hn):

    hn2 = hn / 2

    k1 = hn * functionF(xn, yn, zn)
    q1 = hn * functionPHI(xn, yn, zn)

    k2 = hn * functionF(xn + hn2, yn + k1 / 2, zn + q1 / 2)
    q2 = hn * functionPHI(xn + hn2, yn + k1 / 2, zn + q1 / 2)

    k3 = hn * functionF(xn + hn2, yn + k2 / 2, zn + q2 / 2)
    q3 = hn * functionPHI(xn + hn2, yn + k2 / 2, zn + q2 / 2)
```

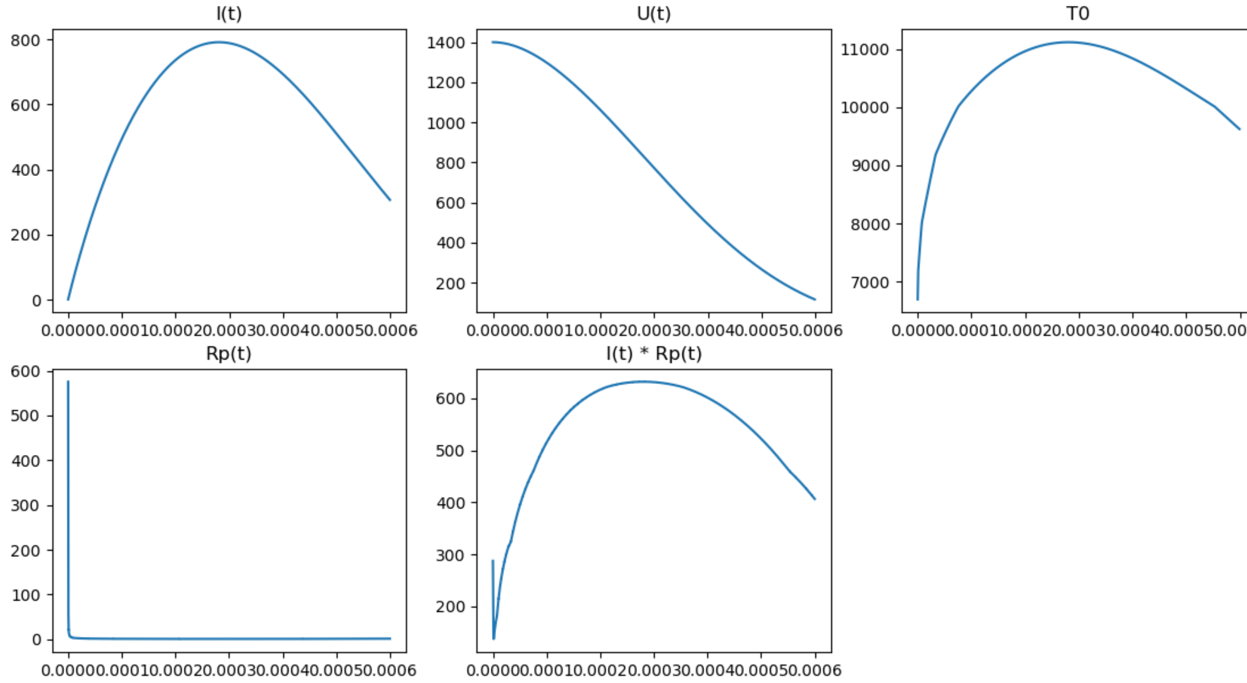
```
    k4 = hn * functionF(xn + hn, yn + k3, zn + q3)
    q4 = hn * functionPHI(xn + hn, yn + k3, zn + q3)

    yn_1 = yn + (k1 + 2 * k2 + 2 * k3 + k4) / 6
    zn_1 = zn + (q1 + 2 * q2 + 2 * q3 + q4) / 6

    return yn_1, zn_1
```

IV. Результаты работы программы:

1) Графики зависимости от времени импульса t : I(t), U(t), $R_p(t)$, произведения $I(t)*R_p(t)$ при заданных выше параметрах. Указать шаг сетки.




2) График зависимости I(t) при Rk + Rp = 0 . Обратить внимание на то, что в этом случае колебания тока будут незатухающими.

Для достижения условия установим следующие параметры:

R=0.35 см

lэ=0

Lk=187 10-6 Гн

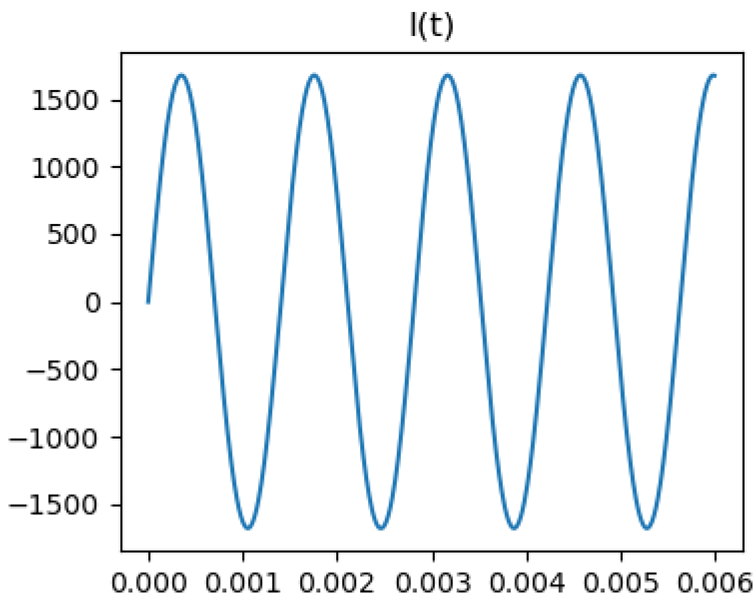
Ck=268 10-6 Ф

Rk=0

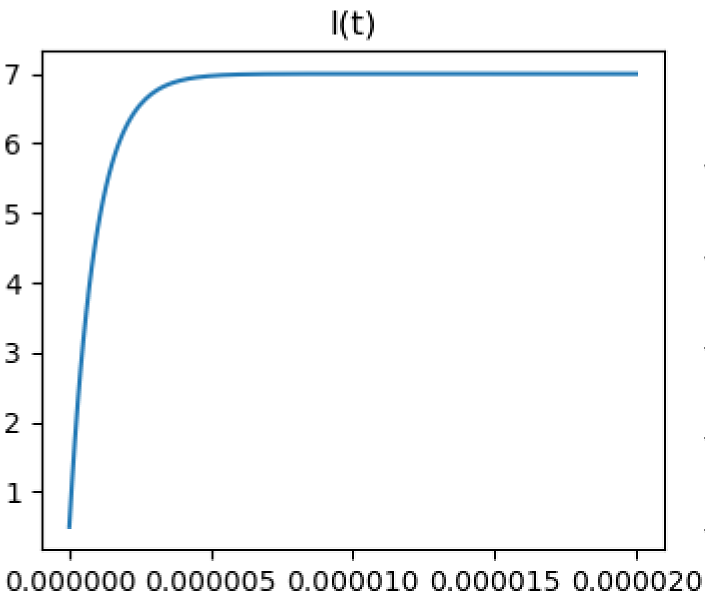
Uco=1400 В

Io=0..3 A

Tw=2000 K




3) График зависимости I(t) при Rk + Rp = const = 200 Ом в интервале значений t 0-20 мкс.



## V. Ответы на вопросы:

1) ***Какие способы тестирования программы, кроме указанного в п.2, можете предложить ещё?***

Программа должна выводить результаты, соответствующие законам физики. Это можно определить по виду графиков при определенных значениях входных параметров. Также, можно проверить, что программа правильно ведет себя при вводе большого значения сопротивления или в случае, когда сумма сопротивлений контура равна нулю (контур обращается в колебательный, колебания не затухают).

Также программу можно тестировать при разных значениях шага. При определенном введем значении шага, дальнейшее уменьшение шага не меняет результат. Это означает, что найден точный результат.

Помимо этого можно сравнить результаты работы двух методов разной точности. При маленьком значении шага они должны совпадать. Это объясняется тем, что при маленьком шаге результат перестает меняться, а результаты двух методов решения одной и той же системы уравнений должны совпадать.

2) ***Получите систему разностных уравнений для решения сформулированной задачи неявным методом трапеций. Опишите алгоритм реализации полученных уравнений.***

$$\begin{cases} u'(x) = f(x,u,v) \\ v'(x) = \varphi(x,u,v) \\ u(\xi) = \eta_1 \\ v(\xi) = \eta_2 \end{cases} => \begin{cases} \frac{dI}{dT} = \frac{U-\left(R_k+R_p(I)\right)I}{L_k} = u'(t) \\ \frac{dU}{dt} = -\frac{I}{C_k} = v'(t) \end{cases}$$

Подставим наши уравнения ($\alpha = 1/2$)

$$I_{n+1} = I_n + h_n * \left(\frac{f(t_n, I_n, Uc_n) + f\left(t_n + h_n, I_n + h_n * f(t_n, I_n, Uc_n), Uc_n + h_n * \phi(t_n, I_n)\right)}{2}\right)$$

$$Uc_{n+1} = Uc + h_n * \left(\frac{\phi(t_n, I_n) + \phi(t_n + h_n,)}{2}\right)$$

3) ***Из каких соображений проводится выбор численного метода того или иного порядка точности, учитывая, что чем выше порядок точности метода, тем он более сложен и требует, как правило, больших ресурсов вычислительной системы?***

Чтобы добиться точного результата на методах низкого уровня точности, нужно использовать очень маленький шаг, а следовательно производить очень много итераций. Методы более высокого порядка сложнее, однако требуют меньше итераций для поиска точного значения. Выбор метода должен основываться на возможностях системы, а также точности, с которой необходимо получить удовлетворительный результат. Для получения результата более высокой точности следует использовать методы более высокого порядка точности. Для менее точного результата целесообразнее использовать методы меньшего порядка точности, так как удовлетворительный результат можно получить быстро и просто и не требуется производить сложные вычисления.