Министерство образования и науки 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»

Отчет по лабораторной работе № 2

По курсу: "Математическое моделирование"

Программно-алгоритмическая реализация методаРунге-Кутта 4-го порядка точности при решении системы ОДУ в задаче Коши.

Студент: Турсунов Жасурбек Рустамович

Группа: ИУ7-66Б

Оценка(баллы):

Преподователь: Градов Владимир Михайлович

Москва, 2021 г.

Содержание

Введение

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

1 Аналитическая часть

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_kи емкость C_k.

$$\begin{cases} \frac{dI}{dT} = \frac{U-(R_k+R_p(I))I}{L_k} \\ \frac{dU}{dt} = -\frac{I}{C_k} \end{cases}$$

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

I, U- ток и напряжение на конденсаторе.

Сопротивление R_p рассчитать по формуле

$$R_p = \frac{l_p}{2\pi R^2 \int_0^1 \sigma(T(z))zdz}$$

Для функции $T(z)$ применить выражение $T(z) = T_0 + (T_w - T_0)z^m$.

Параметры T_0, m находятся интерполяцией из таблицы 1 при известном токе I.

Коэффициент электропроводности $\sigma(T)$ зависит от T и рассчитывается интерполяцией из таблицы 2.

Таблица 1:

I, A	T_0	**m**
0.5	6730	0.5
1	6790	0.55
5	7150	1.7
10	7270	3
50	8010	11
200	9185	32
400	10010	40
800	11140	41
1200	12010	39

Таблица 2

T, K	$\sigma, \frac{1}{Omcm}$
4000	0.031
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 = 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

2 Технологическая часть

2.1 Листинг кода

```
def interpolate(x, masX, masY):
        order = 1
        s = InterpolatedUnivariateSpline(masX, masY, k=order)
        return float(s(x))

def T(z):
    return (Tw - T0) * z**m + T0

def sigma(T):
    return interpolate(T, masT, masSigm)

def Rp(I):
    global m
    global T0
    m = interpolate(I, masI, masm)
    T0 = interpolate(I, masI, masT0)

    def func(z): return sigma(T(z)) * z
    integral = integrate.quad(func, 0, 1)
    Rp = le/(2 * numpy.pi * R**2 * integral[0])

    return Rp

def f(xn, yn, zn):
    return -((Rk + m_Rp_global) * yn - zn)/Lk

def phi(xn, yn, zn):
    return -yn/Ck

def fourth_order(xn, yn, zn, hn, m_Rp):
    global m_Rp_global
    m_Rp_global = m_Rp

    k1 = hn * f(xn, yn, zn)
    q1 = hn * phi(xn, yn, zn)
```

```

    k2 = hn * f(xn + hn/2, yn + k1/2, zn + q1/2)
    q2 = hn * phi(xn + hn/2, yn + k1/2, zn + q1/2)

    k3 = hn * f(xn + hn/2, yn + k2/2, zn + q2/2)
    q3 = hn * phi(xn + hn/2, yn + k2/2, zn + q2/2)

    k4 = hn * f(xn + hn, yn + k3, zn + q3)
    q4 = hn * phi(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

```

3 Исследовательская часть

3.1 Примеры работы

Графики зависимости от времени импльса t: $I(t), U(t), R_p(t), I(t)*R_p(t), T_0(t)$ при заданных выше параметрах:

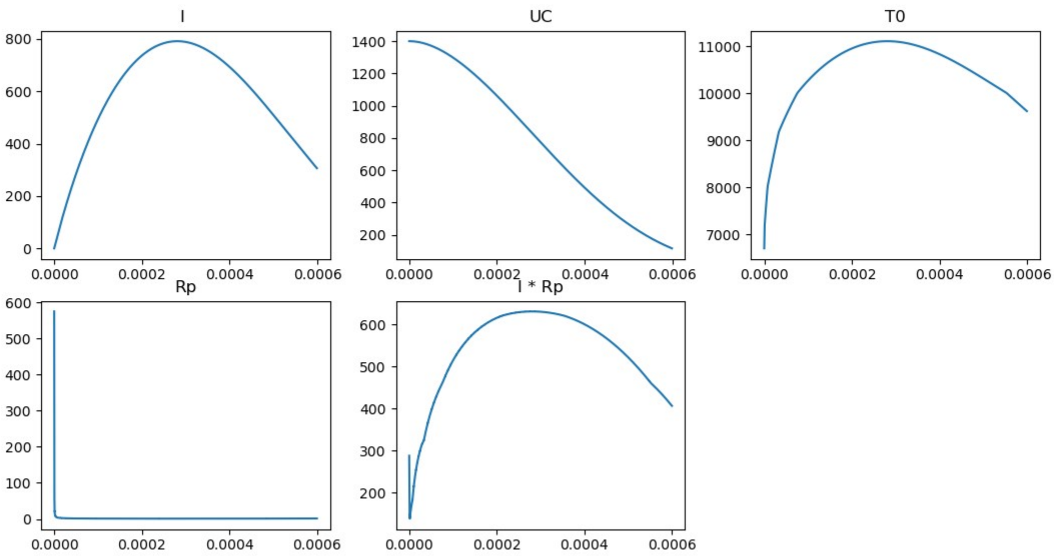
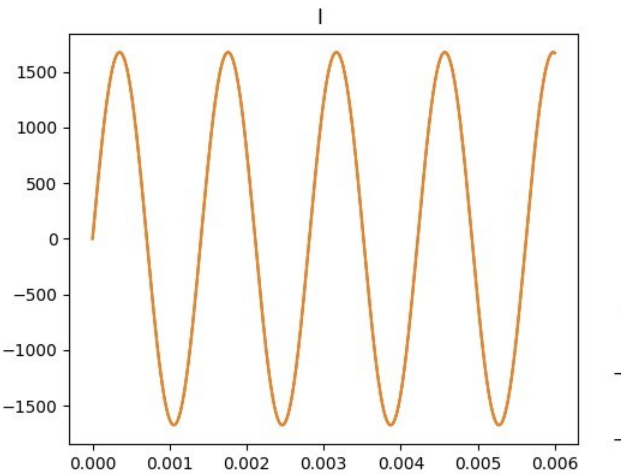
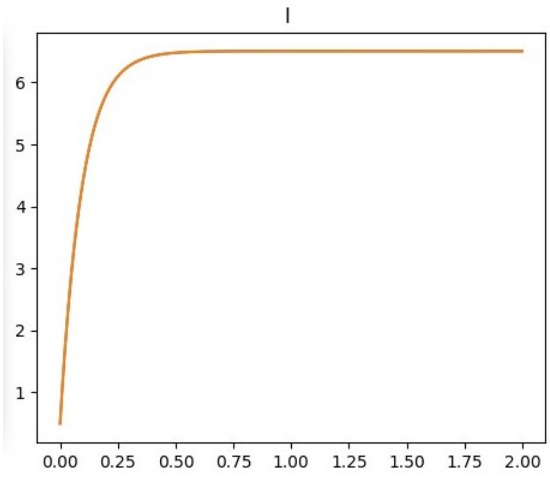


График зависимости $I(t)$при $R_k + R_p = 0$.



Так как сопротивление в контуре нулевое, контур - колебательный, и колебания тока не затухают.

График зависимости $I(t)$ при $R_k + R_p = 200$ Ом в интервале значений t0-20 мкс.



4 Ответы на вопросы

1) Какие способы тестирования программы можно предложить?

При тестировании программы изменять шаг. Уменьшая шаг, мы дойдем до момента, когда новое уменьшение шага никак не изменит полученный результат. Отсюда следует, что полученный результат является точным. Также при тестировании нужно учесть, что работа моделируемого электрического контура описывается теоретически законами физики. Поэтому проверяются ситуации, когда сопротивление в контуре нулевое ,и он становится колебательным, а также когда сопротивление наоборот велико.

2) Из каких соображений проводится выбор того или иного метода, учитывая, что чем выше порядок точности метода, тем он более сложен?

Выбор метода проводится с учетом точности и шага. При большом шаге для получения достаточно точного результата лучше использовать методы более высокого порядка точности. Сложность вычислений на каждой итерации в таком случае будет компенсирована тем, что самих итераций при большом шаге будет меньше. При маленьком шаге методы менее высокого порядка точности дают такой же результат, как и методы более высокого порядка, для которых количество вычислений значительно больше. Также стоит учесть, что количество итераций при маленьком шаге возрастает. Поэтому в этом случае лучше выбрать методы менее высокого порядка точности.

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

Неявный метод трапеций – это метод Рунге-Кутта второго порядка точности с $\alpha = 0.5$

Уравнение

$$y_{n+1} = y_n + h_n[(1-\alpha)f(x_n, y_n) + f(x_n + \frac{h}{2\alpha}, y_n + \frac{h}{2\alpha}f(x_n, y_n))]$$

сводится к уравнению $y_{n+1} = y_n + h_n[\frac{f(x_n, y_n) + f(x_n, h, y_n + hf(x_n, y_n))}{2}]$

Получаем систему разностных уравнений:

$$\begin{cases} I_{n+1} = I_n + h_n[\frac{f(I_n, U_{cn} + f(I_n + h, U_{cn} + hg(I_n)))}{den}] \\ U_{c_{n+1}} = U_{cn} + h_n[\frac{g(I_n) + g(I_n + h)}{2}] \end{cases}$$

Имеется исходная система:

$$\begin{cases} I'^{(t)} = \frac{U_c - I(R_k + R_p)}{L_k} \stackrel{\text{def}}{=} f(I, U_c) \\ U'(t) = -\frac{I}{C_k} \stackrel{\text{def}}{=} (g(I)) \\ I(0) = I_0 \\ U(0) = U_{c0} \end{cases}$$

Подставляя уравнения производных в разностные уравнения, можно найти решение итерационно, так как на i-м ходу известны I_i, U_{ci}, h_i. Подставив их в полученные формулы, находим $I_{i+1} U_{c_{i+1}}, h_{i+1}$ известно заранее (например, шаг может быть постоянным).