



**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 (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.03 ПРОГРАММНАЯ ИНЖЕНЕРИЯ

О Т Ч Е Т

По лабораторной работе № 2

Дисциплина: Моделирование

Студент

ИУ7-62Б

(Группа)

Н.А. Гарасев

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

В.М. Градов

(Подпись, дата)

(И.О. Фамилия)

Москва, 2021

Тема: Программно-алгоритмическая реализация метода Рунге-Кутты 4-го порядка точности при решении системы ОДУ в задаче Коши.

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

Исходные данные:

Задана система электротехнических уравнений, описывающих разрядный контур, включающий постоянное активное сопротивление R_k , нелинейное сопротивление $R(I)_p$, зависящее от тока I , индуктивность L_k и 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: $t=0, I=I_0, U=U_0$. Здесь I, U - ток и напряжение на конденсаторе. Сопротивление R_p рассчитать по формуле

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

, где

$$T(z) = T_0 + (T_w - T_0) z^m.$$

Параметры разрядного контура:

$R=0.35$ см

$l_p=12$ см

$L_k = 187 \cdot 10^{-6} \text{ Гн}$

$C_k = 268 \cdot 10^{-6} \text{ Ф}$

$R_k = 0.25 \text{ Ом}$

$U_{co} = 1400 \text{ В}$

$I_o = 0..3 \text{ А}$

$T_w = 2000 \text{ К}$

Ход работы:

Для вычисления I и U , заданных системой электротехнических уравнений, используется метод Рунге-Кутты 4-го порядка точности, описанный в листинге 1.

Листинг 1. Метод Рунге-Кутты 4-го порядка точности.

```
def RungeKutta4(I, U, h):  
    k1 = f1(I, U)  
    q1 = f2(I)  
  
    k2 = f1(I + h * k1 / 2, U + h * q1 / 2)  
    q2 = f2(I + h * k1 / 2)  
  
    k3 = f1(I + h * k2 / 2, U + h * q2 / 2)  
    q3 = f2(I + h * k2 / 2)  
  
    k4 = f1(I + h * k3, U + h * q3)  
    q4 = f2(I + h * k3)  
  
    I_res = I + h * (k1 + 2 * k2 + 2 * k3 + k4) / 6  
    U_res = U + h * (q1 + 2 * q2 + 2 * q3 + q4) / 6  
    return I_res, U_res
```

Где функции $f1$ и $f2$ – функции системы электротехнических уравнений из условия, описанные в листинге 2.

Листинг 2. Система электротехнических уравнений.

```
def f1(I, U):  
    return (U - (Rk + calcR(I)) * I) / Lk  
def f2(I):  
    return -1 * I / Ck
```

R_p – рассчитывается по формуле из условия, реализация представлена в листинге 3.

Листинг 3. Вычисление R_p .

```
def calcR(I):  
    r = 1 / (2 * math.pi * R * R * integral(I))  
    return r
```

Вычисление численного значения интеграла, зависящего от силы тока, реализован в листинге 4.

Листинг 4. Вычисление интеграла.

```
def integral(I):  
    return trapezoid(T, 0, 1, 10 ** -2, I)  
def trapezoid(f, a, b, h, i):  
    p = round((b - a) // h)  
    s = 0  
    for _ in range(p):  
        s += (f(a, i) + f(a + h, i)) / 2 * h  
        a += h  
    return s
```

Интеграл вычисляется по методу трапеции с шагом 10^{-2} (такой шаг выбран как, что-то среднее между временем вычислением и точностью результата). Значение в точках вычисляются с помощью функции T, представленную в листинге 5.

Листинг 5. Функция T().

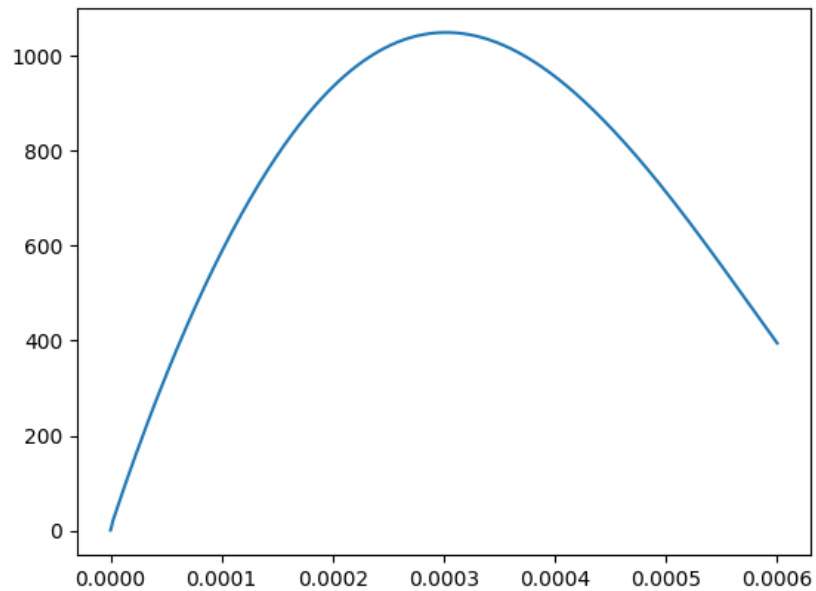
```
def T(z, i):  
    temp = newton(tab2[0], tab2[1])  
    f_t = linear(tab1[0], tab1[1])  
    f_m = linear(tab1[1], tab1[2])  
    return temp(f_t(i) + (Tw - f_t(i)) * z**f_m(f_t(i)))
```

Первым делом в функции T() определяются функции temp(), f_t(), f_m(). С помощью разных методов интерполяции, заданные функции вычисляют значения при заданном i. Функция temp() - используя интерполяцию по Ньютону, вычисляет значение сигмы по заданной температуре. Для функций f_t() и f_m() был выбран метод линейной интерполяции, так как у узлов в таблице переменный шаг. Функция f_t() – вычисляет значение T при заданном I. Функция f_m() – вычисляет значение m при заданном T.

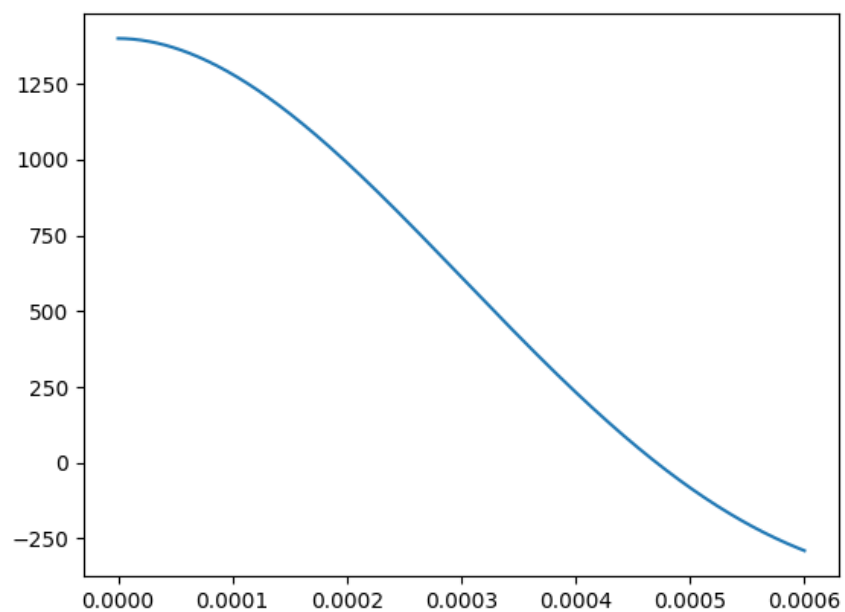
Результат программы:

1. Графики зависимости от времени импульса t .

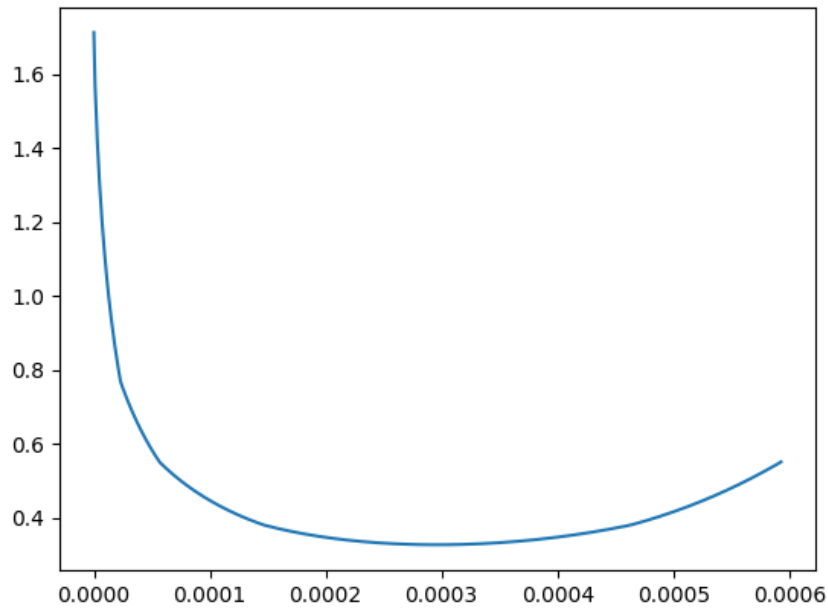
$I(t)$, где на горизонтальной оси время импульса в секундах, а на вертикальной – сила тока в А.



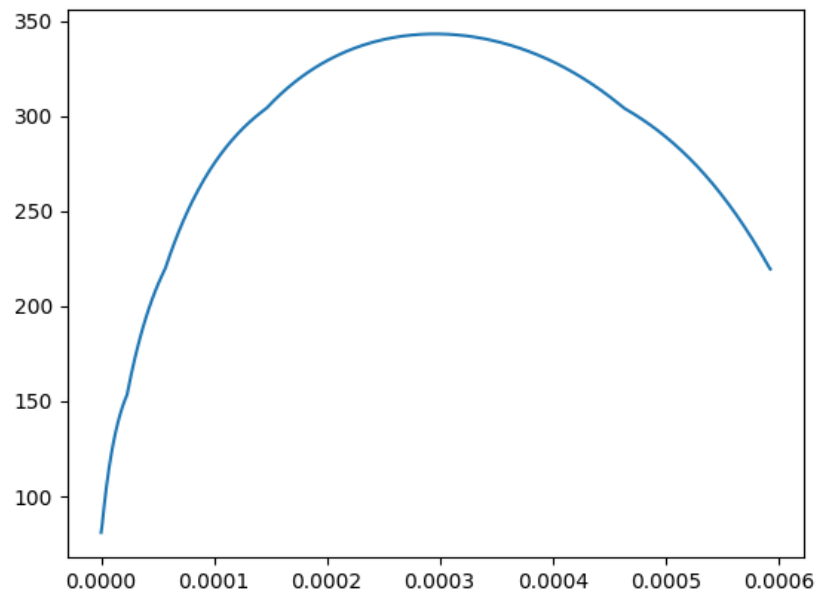
$U(t)$, где на горизонтальной оси время импульса в секундах, а на вертикальной – напряжение в В.



$R_p(t)$, где на горизонтальной оси время импульса в секундах, а на вертикальной – сопротивление в Ом.



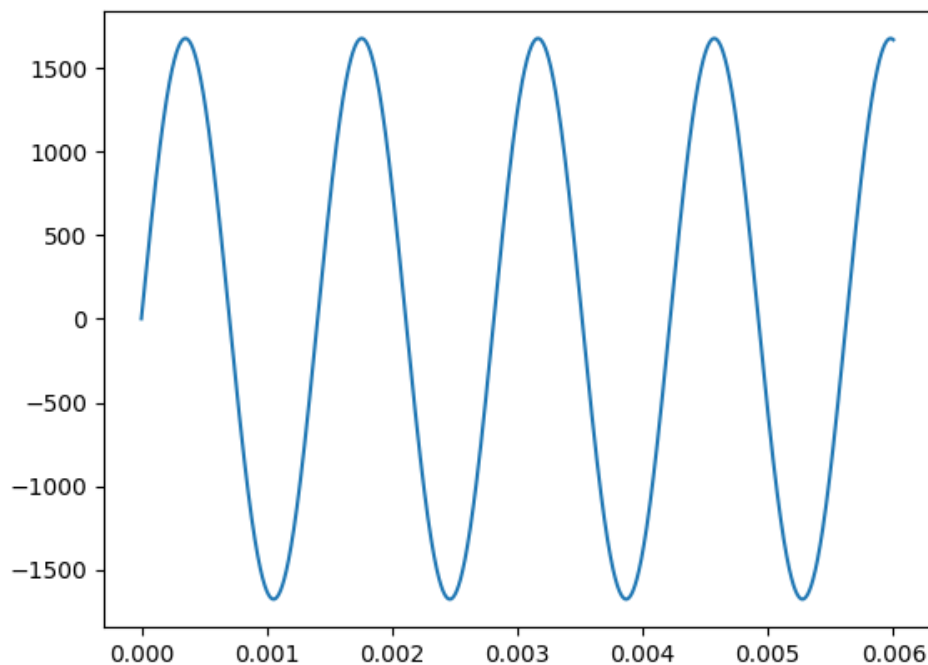
Произведение $I(t) * R_p(t)$, где на горизонтальной оси время импульса в секундах, а на вертикальной – произведение силы тока на сопротивление в А * Ом.



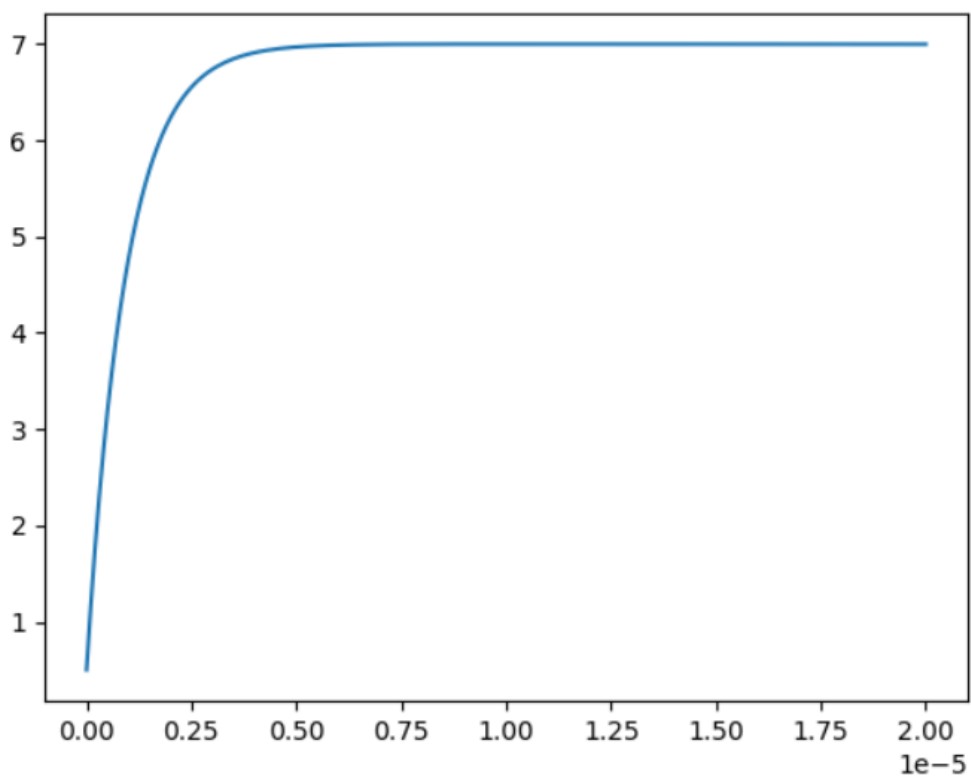
Все вычисления происходили при $dt = 10^{-6}$ на временном интервале от 0 до 600 мкс.

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

$I(t)$, где на горизонтальной оси время импульса в секундах, а на вертикальной – сила тока в А.



3. График зависимости $I(t)$ при $R_k + R_p = \text{const} = 200 \text{ Ом}$.



4. Результаты исследования влияния параметров контура C_k , L_k , R_k на длительность импульса тмп. аperiodической формы

Длительность импульса определяется по кривой зависимости тока от времени на высоте 35 max 0. I , max I - значение тока в максимуме.

При исходных данных:

$$I_{\max} = 1047.02 \text{ A}$$

$$0.35 * I_{\max} = 366.46 \text{ A}$$

$$T_{\text{imp}} = 551 \text{ мкс}$$

Составим таблицу с измененными параметрами.

Таблица 1. Изменение C_k .

C_k	T_{imp} мкс	T_{start} , мкс	T_{end} , мкс	I_{\max}	$0.35 * I_{\max}$
$0.25 * C_k$	273	33	306	616.34	215.72
$0.5 * C_k$	387	44	431	810,78	283,77
C_k	551	58	609	1047.02	366.46
$1.25 * C_k$	618	63	681	1132,27	396,29
$1,5 * C_k$	675	67	742	1199,79	419,92

Как можно заметить, длительность импульса и максимальная сила тока, прямо-пропорциональна параметру C_k .

Таблица 2. Изменение L_k .

L_k	T_{imp} мкс	T_{start} , мкс	T_{end} , мкс	I_{\max}	$0.35 * I_{\max}$
$0.75 * L_k$	479	49	528	1157.77	405.22
L_k	551	58	609	1047.02	366.46
$1.25 * L_k$	615	64	679	966,43	338,25
$1,5 * L_k$	673	73	756	903,73	316,30

Параметр L_k прямо-пропорционален длительности импульса и обратно-пропорционален максимальной силе тока.

Таблица 3. Изменение R_k .

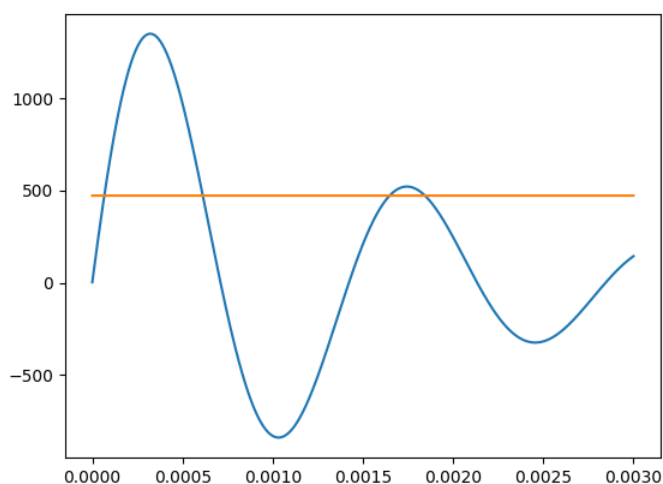
R_k	T_{imp} мкс	T_{start} , мкс	T_{end} , мкс	I_{max}	$0.35 * I_{max}$
$0.5 * R_k$	547	63	610	1149.96	402.48
$0.75 * R_k$	549	60	609	1096.15	383.65
R_k	551	58	609	1047.02	366.46
$1.25 * R_k$	554	56	610	1001.97	350.69
$1,5 * R_k$	557	54	611	690,54	336,19
$2 * R_k$	566	50	616	886,87	310,40

Параметр R_k прямо-пропорционален длительности импульса и
обратно-пропорционален максимальной силе тока.

Вопросы.

1. Какие способы тестирования программы, кроме указанного в п.2, можете провести еще?

Если убрать R_p , то по законам физики сила тока будет совершать затухающие колебания.



При уменьшении шага, значения практически не изменяются, это означает, что найден точный результат.

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

Возьмем первоначальные уравнения:

$$\frac{dI}{dt} = \frac{U_c - (R_k + R_p(I))I}{L_k} \equiv f(I, U_c)$$

$$\frac{dU_c}{dt} = -\frac{1}{C_k}I \equiv g(I)$$

Далее запишем выражения для метода трапеций:

$$I_{n+1} = I_n + \Delta t \frac{f(I_n, U_{cn}) + f(I_{n+1}, U_{cn+1})}{2}$$

$$U_{cn+1} = U_{cn} + \Delta t \frac{g(I_n) + g(I_{n+1})}{2}$$

Затем необходимо подставить выражения f и g .

$$I_{n+1} = I_n + \Delta t \frac{U_{cn} - (R_k + R_p(I_n))I_n + U_{cn+1} - (R_k + R_p(I_{n+1}))I_{n+1}}{2L_k}$$

$$U_{cn+1} = U_{cn} - \Delta t \frac{I_n + I_{n+1}}{2C_k}$$

Имеем два уравнения и две неизвестных величины. Подставляем U_{cn+1} из второго уравнения в первое, находим I_{n+1} , затем находим U_{cn+1} .

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?

Для того, чтобы провести вычисления методом Рунге-Кутты четвертого порядка точности, необходимо чтобы: функция в правой части была ограничена и непрерывна, а также ее четвертые производные. Если функция не имеет четвертых производных, то целесообразно использовать другой метод, например Рунге-Кутта второго порядка точности, чтобы не тратить вычислительные мощности системы.

4. Можно ли метод Рунге - Кутта применить для решения задачи, в которой часть условий задана на одной границе, а часть на другой?

Например, напряжение по-прежнему задано при $t = 0$, т.е. $U = U_0$, а ток задан в другой момент времени, к примеру, в конце импульса, т.е. при $t = T$, $I = I_0$. Какой можете предложить алгоритм вычислений?

Каждое последующее вычисление требует значения в предыдущий момент времени, т.е. метод Рунге-Кутты не подходит для такого условия, так как известен лишь одно значение. Для решения необходим алгоритм, способный решать краевые задачи.