



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»

Отчёт

по лабораторной работе № 2

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

Дисциплина: Моделирование

Студент

ИУ7-65Б

(Группа)

Д.В. Сусликов

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

В.М. Градов

(Подпись, дата)

(И.О. Фамилия)

Москва, 2021

Введение

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 $R_p(I)$, 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

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

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 $\sigma(T)$ зависит от T и рассчитывается интерполяцией из табл.2.

Таблица 1

I, A	T_0, K	m
0.5	6730	0.50
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	$\sigma, 1/\text{Ом см}$
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,

$$L_e = 12 \text{ см},$$

$$L_k = 187 * 10^{-6} \text{ Гн},$$

$$C_k = 268 * 10^{-6} \text{ Ф},$$

$$R_k = 0.25 \text{ Ом},$$

$$U_c = 1400 \text{ В},$$

$$I_0 = 0.3 \text{ А},$$

$$T_w = 2000 \text{ К}$$

Теоритические сведения:

Метод Рунге-Кутты 4-го порядка точности.

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

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

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

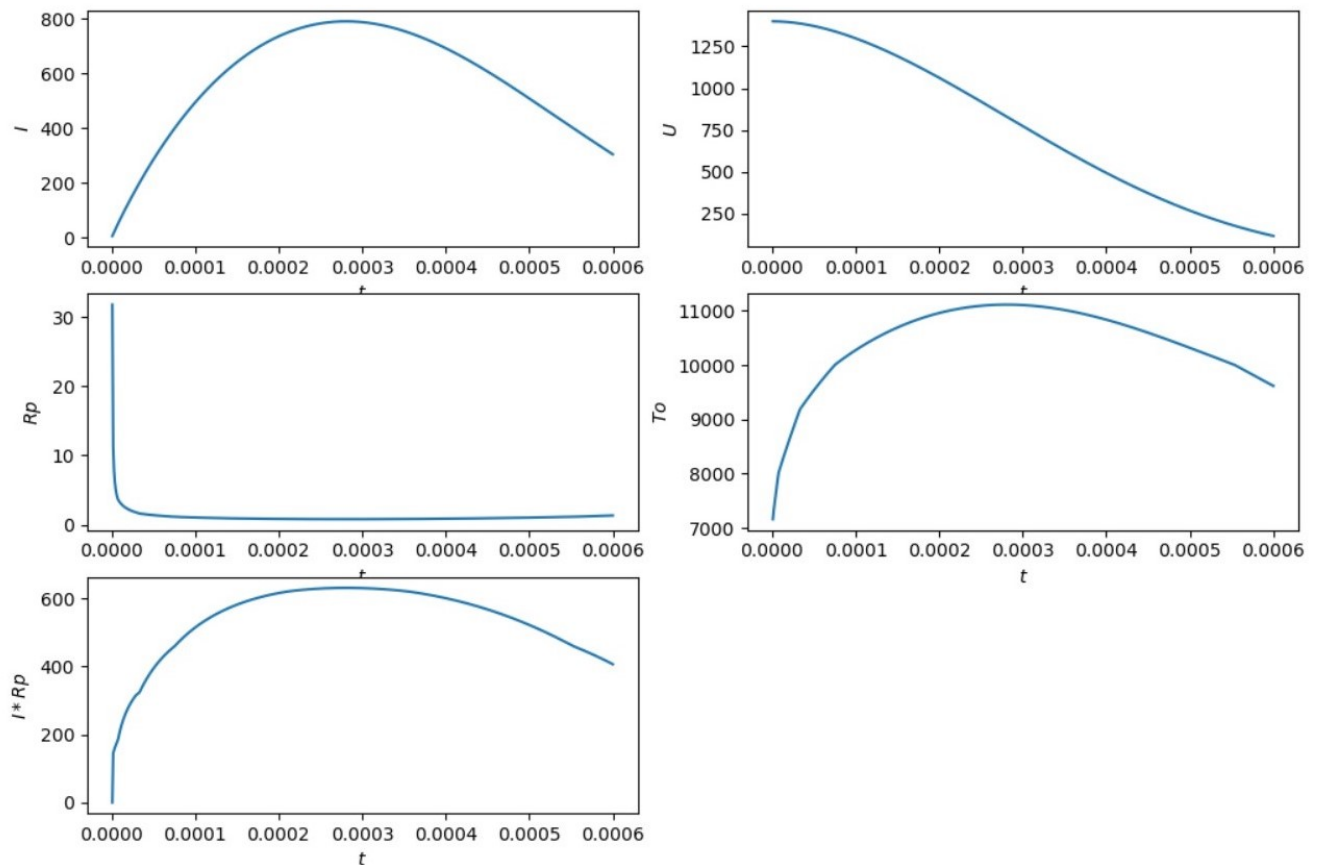
$$k_2 = h_n f(y_n + \frac{k_1}{2}, z_n + \frac{q_1}{2}), \quad q_2 = h_n \varphi(y_n + \frac{k_1}{2})$$

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

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

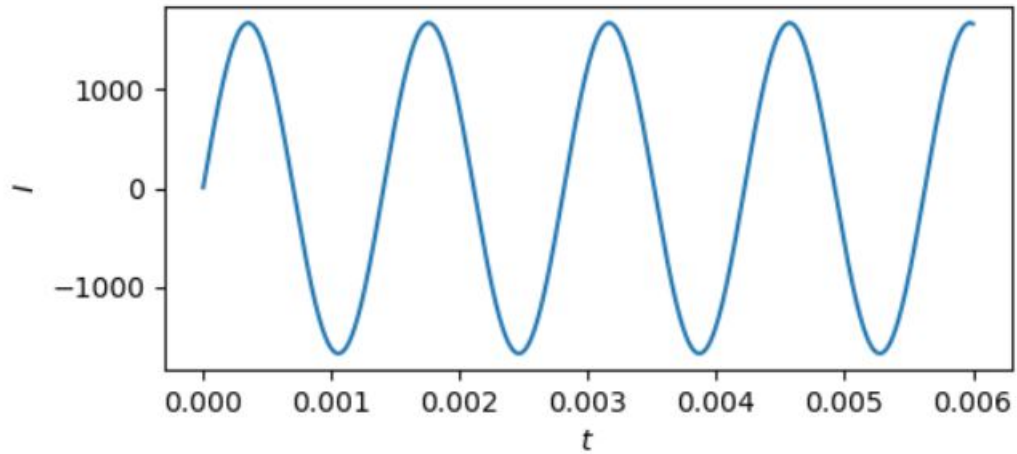
Результат работы

1. Графики зависимости от времени импульса t : $I(t)$, $U(t)$, $R_p(t)$, произведения $I(t)R_p(t)$, $T_0(t)$ при заданных выше параметрах. Указать шаг сетки.

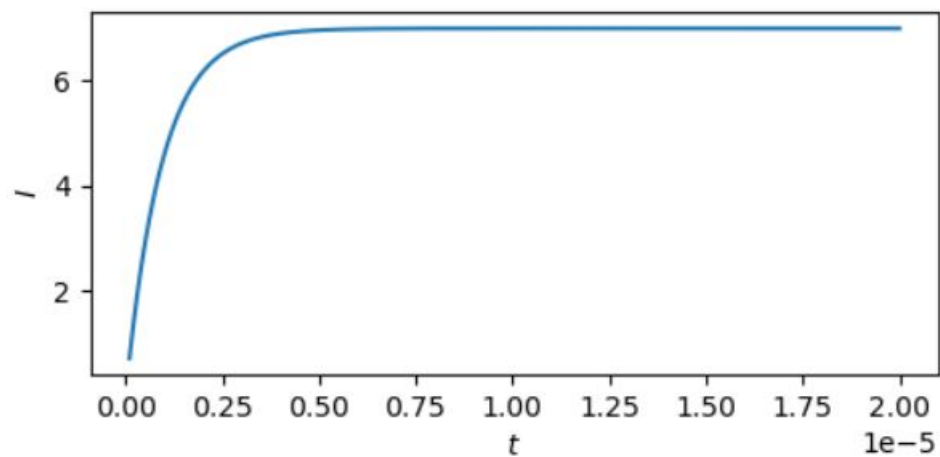


Шаг - 10^{-6}

2. График зависимости $I(t)$ при $R_k + R_p = 0$. Обратите внимание на то, что в этом случае колебания тока будут незатухающими.

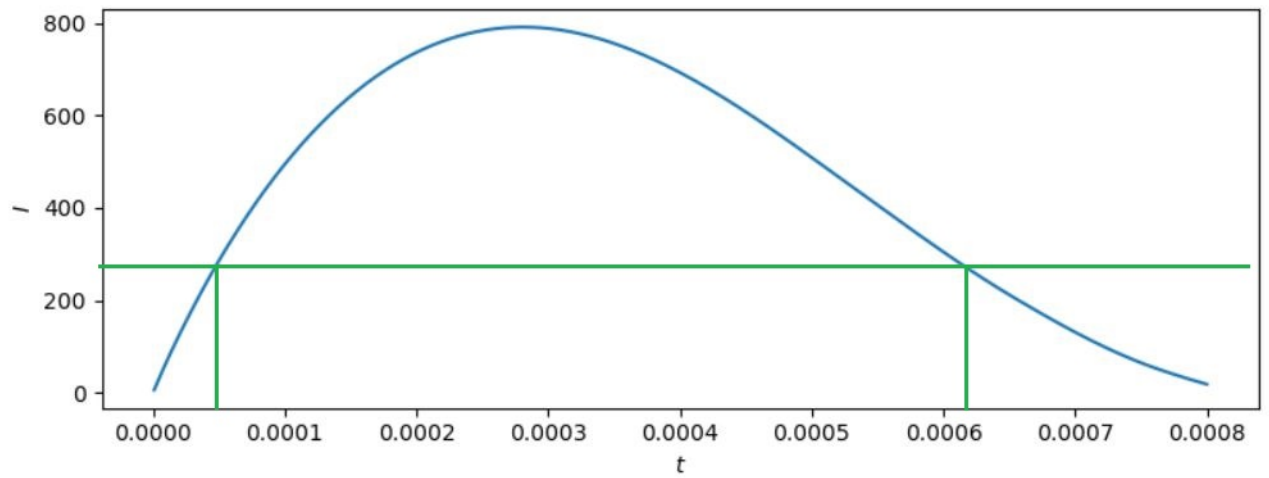


3. График зависимости $I(t)$ при $R_k + R_p = \text{const} = 200$ Ом в интервале значений 0 - 20 мкс.



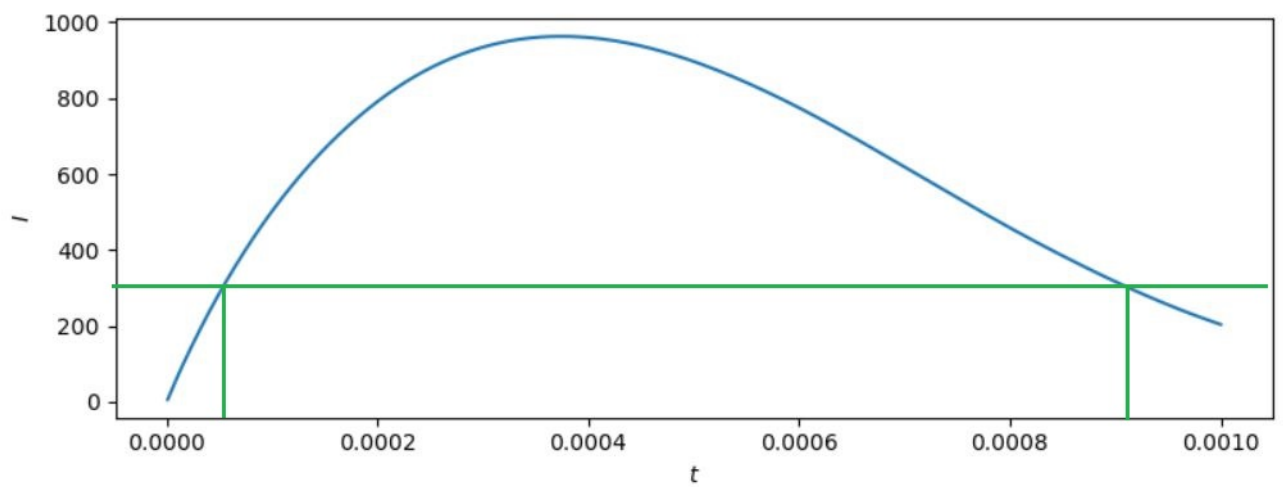
4. Результаты исследования влияния параметров контура C_k, L_k, R_k на длительность импульса $t_{\text{имп}}$ апериодической формы. Длительность импульса определяется по кривой зависимости тока от времени на высоте $0.35I_{\text{max}}, I_{\text{max}}$ - значение тока в максимуме.

График при начальных значениях:



Длительность импульса чуть меньше 6 мкс.

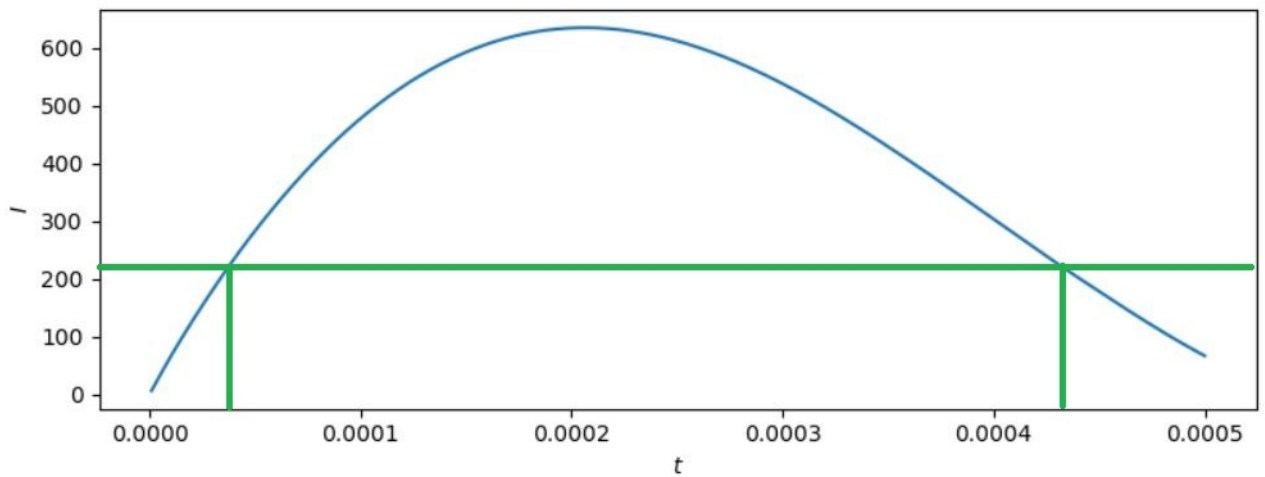
График при увеличении C_k в 2 раза:



Длительность импульса около 9 мкс.

При увеличении C_k , $t_{\text{имп}}$ увеличивается.

График при уменьшении C_k в 2 раза:

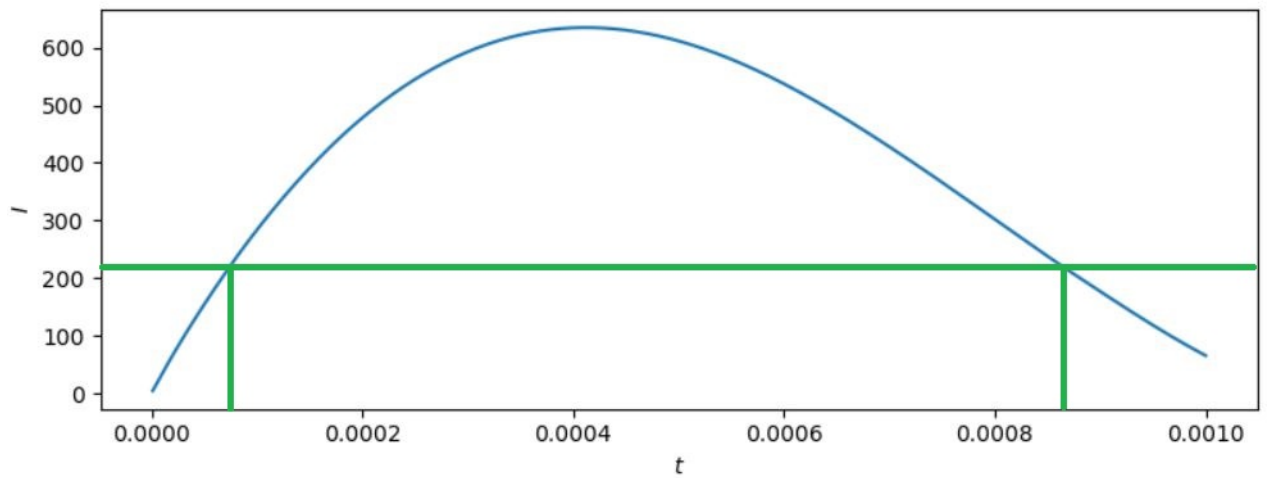


Длительность импульса около 4 мкс.

При уменьшении C_k , $t_{\text{имп}}$ уменьшается.

Можно сделать вывод, что C_k , $t_{\text{имп}}$ пропорционально зависимы.

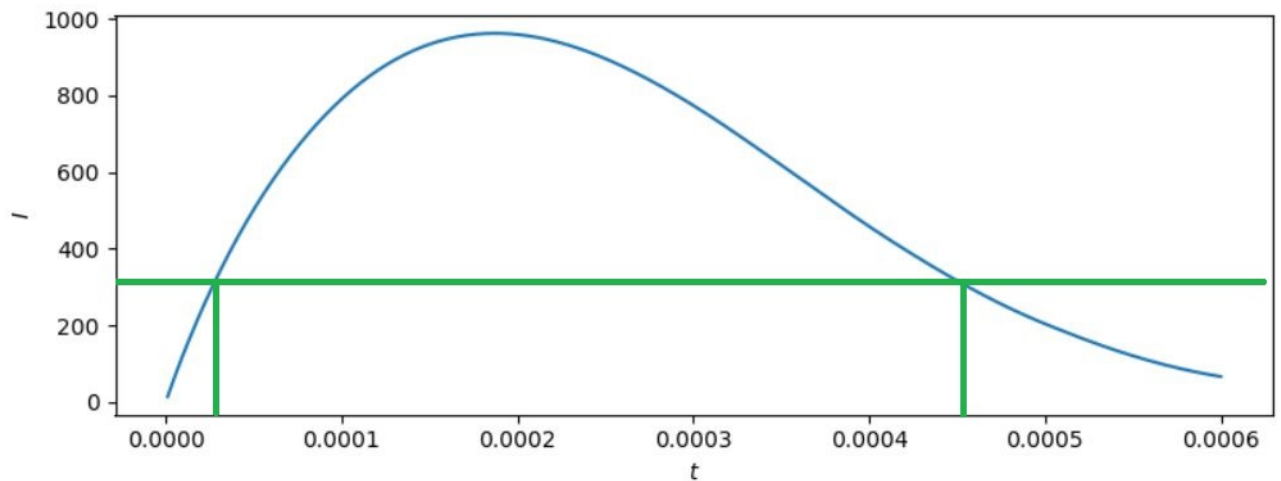
График при увеличении L_k в 2 раза:



Длительность импульса около 8 мкс.

При увеличении L_k , $t_{\text{имп}}$ увеличивается.

График при уменьшении L_k в 2 раза:

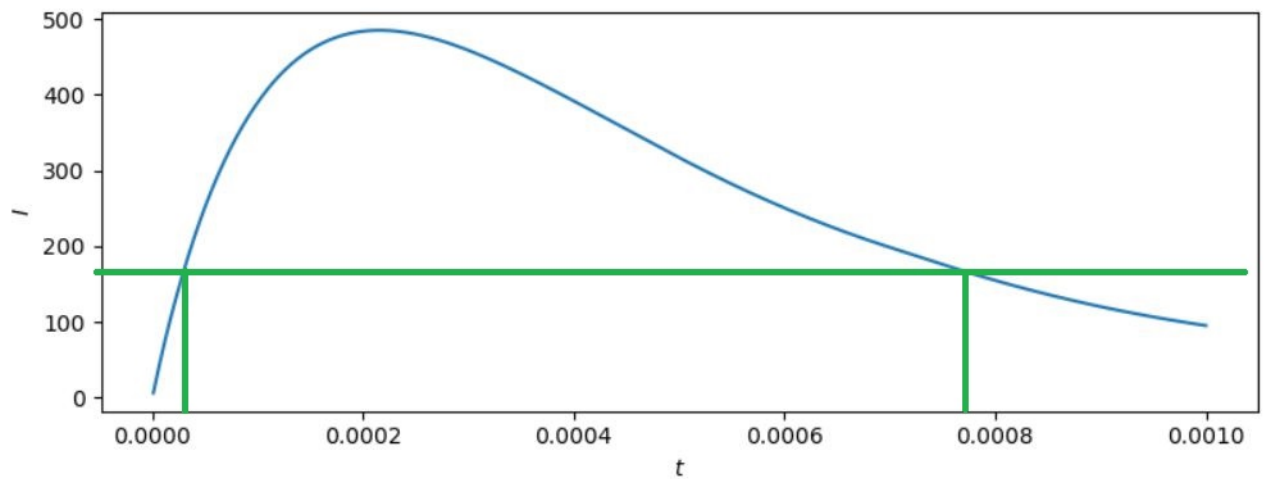


Длительность импульса около 4 мкс.

При уменьшении L_k , $t_{\text{имп}}$ уменьшается.

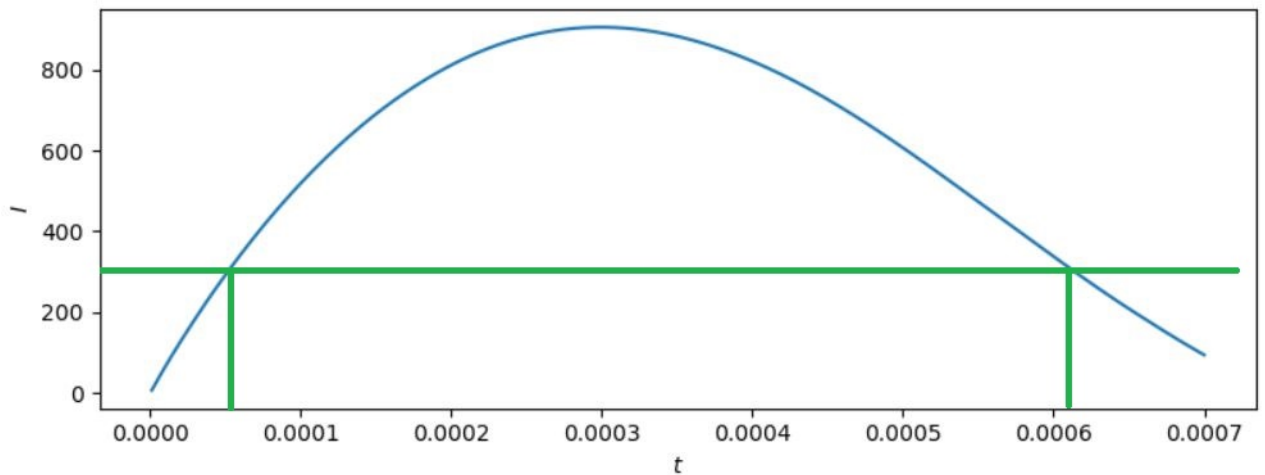
Можно сделать вывод, что L_k , $t_{\text{имп}}$ пропорционально зависимы.

График при увеличении R_k в 5 раза:



Длительность импульса чуть меньше 8 мкс. При увеличении R_k , $t_{\text{имп}}$ увеличивается.

График при уменьшении R_k в 5 раза:



При уменьшении R_k , $t_{\text{имп}}$ уменьшается.

Можно сделать вывод, что R_k , $t_{\text{имп}}$ пропорционально зависимы.

Ответы на вопросы

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

Можно изменять параметры I_0, U_{c0}, R_k при $R_p(I) = 0$. Если значение R_k будет велико, то будет наблюдаться апериодическое затухание, при малых значениях - затухающие колебания.

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

$$u_{n+1} = u_n + \int_{x_n}^{x_{n+1}} f(x, u(x)) dx$$

$$u_{n+1} = u_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, u_{n+1})] + O(h^2)$$

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

$$I_{n+1} = I_n + \frac{h}{2} \left[\frac{U_n - (R_k + R_p(I_n))I_n}{L_k} + \frac{U_{n+1} - (R_k + R_p(I_{n+1}))I_{n+1}}{L_k} \right]$$

$$U_{n+1} = U_n + \frac{h}{2} \left[-\frac{I_n}{C_k} - \frac{I_{n+1}}{C_k} \right] = U_n - \frac{h}{2} \left[\frac{I_n + I_{n+1}}{C_k} \right]$$

Подставляя U_{n+1} в выражение для I_{n+1}

$$I_{n+1} = I_n + \frac{h}{2L_k} \left[2U_n - (R_k + R_p(I_n) + \frac{h}{2C_k})I_n - (R_k + R_p(I_{n+1}) + \frac{h}{2C_k})I_{n+1} \right]$$

$$I_{n+1} = a(I_n) + bI_{n+1} + cR_p(I_{n+1})I_{n+1}$$

$$\text{где } a(I_n) = I_n + \frac{h}{2L_k} \left[2U_n - (R_k + R_p(I_n) + \frac{h}{2C_k})I_n \right]$$

$$b = -\frac{h}{2L_k} \left(R_k + \frac{h}{2C_k} \right)$$

$$c = -\frac{h}{2L_k}$$

Получим уравнение вида

$$x = f(x)$$

Его можно решить методом простых итераций или методом Ньютона, после этого определить U_{n+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?

Выбор порядка точности численного метода зависит от вида правой части дифференциального уравнения и требуемой точности вычислений.

$$\phi(x, u) \equiv \phi(x)$$

Если правая часть непрерывна и ограничена, и её четвёртые производные тоже, то использование метода Рунге-Кутты четвёртого порядка имеет смысл. В противном случае предельный (четвертый) порядок схемы Рунге-Кутты не может быть достигнут, и стоит использовать более простые схемы.

Реализация

Листинг 1 – Интерполяция

```
1  def interpolate(value, table_v, table):
2      end = 1
3      start = 0
4      i = 0
5
6      while ((i < (len(table_v))) and (value > table_v[i])):
7          i += 1
8          end = i
9
10     start = end - 1
11
12     return table[start] + (table[end] - table[start]) / (table_v[
        end] - table_v[start]) * (value - table_v[start])
```

Листинг 2 – Интегрирование

```
1  def f_integr(l, z):
2      t0 = interpolate(l, table_l, table_T0)
3      m = interpolate(l, table_l, table_m)
4      t = t0 + (tw - t0) * (z ** m)
5      sigma = interpolate(t, table_T, table_sigma)
6
7      return sigma * z
8
9  def i_integr(l):
10     a = 0
11     b = 1
12     n = 100
13     h = (b - a) / n
14     s = (f_integr(l, a) + f_integr(l, b)) / 2
15     x = 0
16
17     for _ in range(n - 1):
```

```

18     x = x + h
19     s = s + f_integr(l, x)
20     s = s * h
21
22     return s

```

Листинг 3 – Вычисление сопротивления Rp

```

1     def Rp(le, R, l):
2         return le / (2 * pi * R * R * i_integr(l))

```

Листинг 4 – Решение системы уравнений методом Рунге-Кутты

```

1     def f(l, U, le, R, Lk, Rk):
2         global global_rp
3         global_rp = Rp(le, R, fabs(l))
4         return (U - (Rk + global_rp) * l) / Lk
5
6     def g(l, Ck):
7         return -l / Ck
8
9     def runge_kutta_l_U(l, U, le, R, Lk, hn, Rk, Ck):
10        k1 = f(l, U, le, R, Lk, Rk)
11        q1 = g(l, Ck)
12
13        k2 = f(l + hn * k1 / 2, U + hn * q1 / 2, le, R, Lk, Rk)
14        q2 = g(l + hn * k1 / 2, Ck)
15
16        k3 = f(l + hn * k2 / 2, U + hn * q2 / 2, le, R, Lk, Rk)
17        q3 = g(l + hn * k2 / 2, Ck)
18
19        k4 = f(l + hn * k3, U + hn * q3, le, R, Lk, Rk)
20        q4 = g(l + hn * k3, Ck)
21
22        return l + hn * (k1 + 2 * k2 + 2 * k3 + k4) / 6, \
23        U + hn * (q1 + 2 * q2 + 2 * q3 + q4) / 6

```