|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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»\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № 2**

|  |  |
| --- | --- |
| **Тема** Программно-алгоритмическая реализация методов Рунге-Кутта 2-го и 4-го порядков точности при решении системы ОДУ в задаче Коши  **Студент** Белоусова Ю.С.  **Группа** ИУ7-61Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Градов В.М. |  |

Москва.

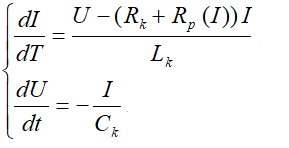
2020 г

**Цель работы**. Получение навыков разработки алгоритмов решения задачи Коши при реализации моделей, построенных на системе ОДУ, с использованием методов Рунге-Кутта 2-го и 4-го порядков точности.

**Исходные данные**

Задана система электротехнических уравнений, описывающих разрядный контур, включающий постоянное активное сопротивление , нелинейное сопротивление , зависящее от тока , индуктивность и емкость .

|  |
| --- |
|  |



Начальные условия: .

- ток и напряжение на конденсаторе.

Сопротивление рассчитать по формуле

Для функции применить выражение .

Параметры находятся интерполяцией из табл.1 при известном токе .

Коэффициент электропроводности зависит от и рассчитывается интерполяцией из табл.2.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | I, A | To, 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 1 | |  |  | | --- | --- | | T, K | , 1/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 |   Таблица 2 |

Параметры разрядного контура:

R=0.35 см

lэ=12 см

Lk=187\*10-6 Гн

Ck=268\*10-6 Ф

Rk=0.25 Ом

Uco=1400 В

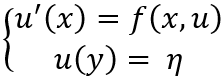
Io=0..3 A

Tw*=*2000 K

# Задача Коши

# Рассмотрим задачу с заданным начальным условием для дифференциального уравнения (Задачу Коши).

Для системы уравнений вида

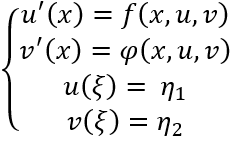


метод Рунге-Кутта второго порядка описывается следующим уравнением:

Где α – произвольный параметр, α ∈ [0, 1] (обычно α = 0.5 или α = 1)

При α = 0.5 получается неявный метод трапеций, при α = 1 – метод средних

Дана система уравнений вида:



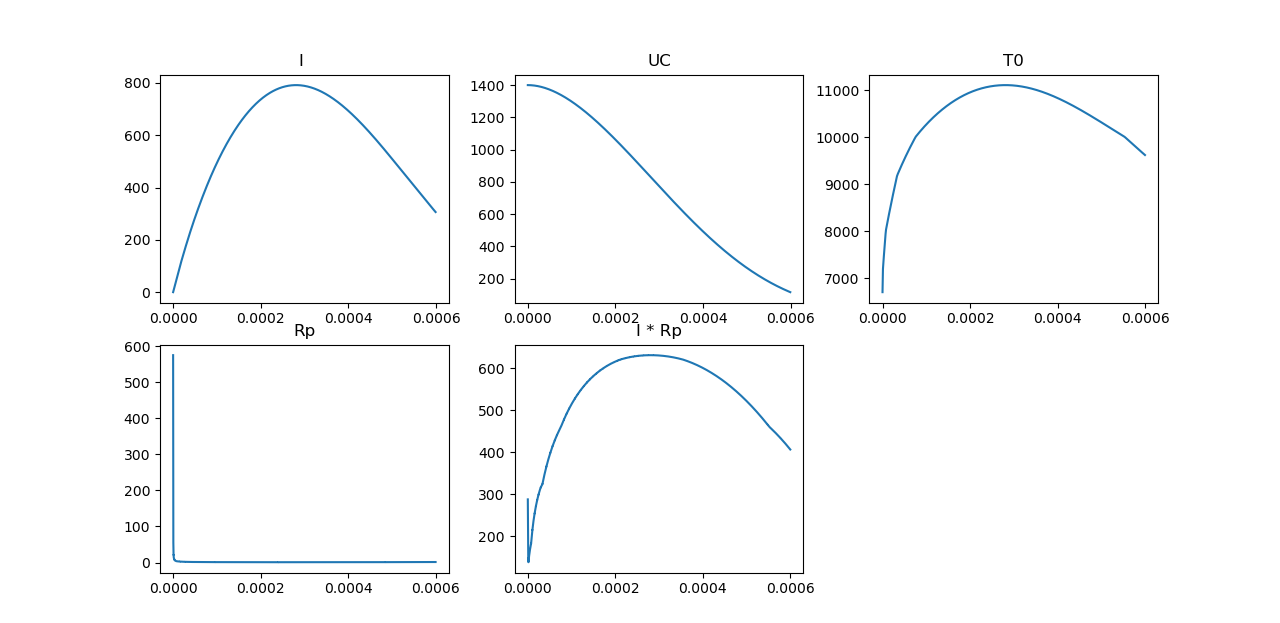
Тогда

, где

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

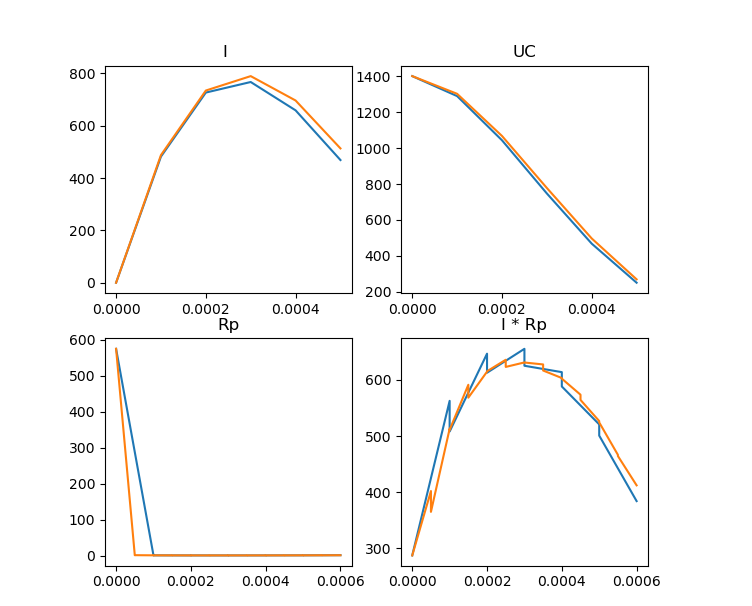
**Результаты работы программы.**

1. Графики зависимости от времени импульса : при заданных выше параметрах:

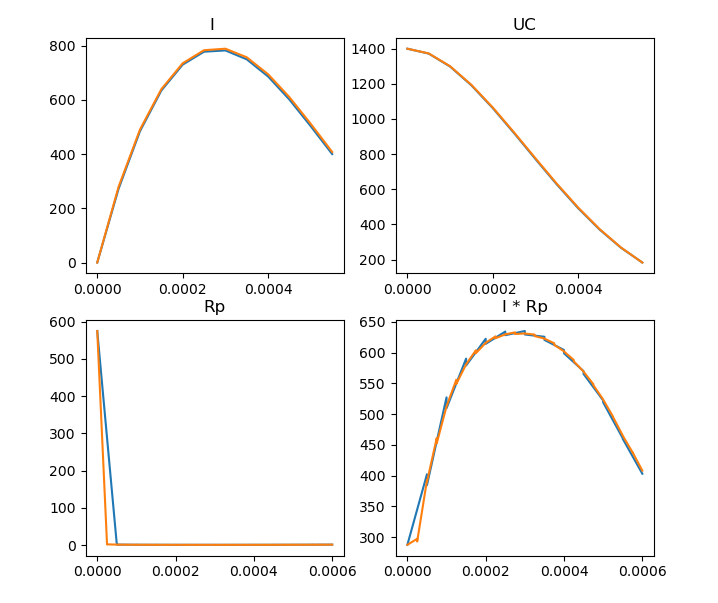


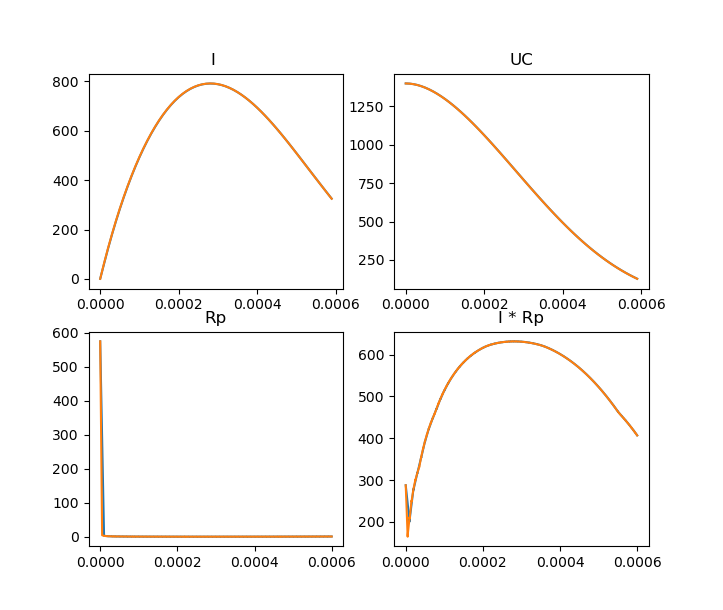
2. Сравним методы разных порядков точности. Для этого проведем вычисления с изменением шага.

Шаг 1e-4:



Шаг 5e-5:

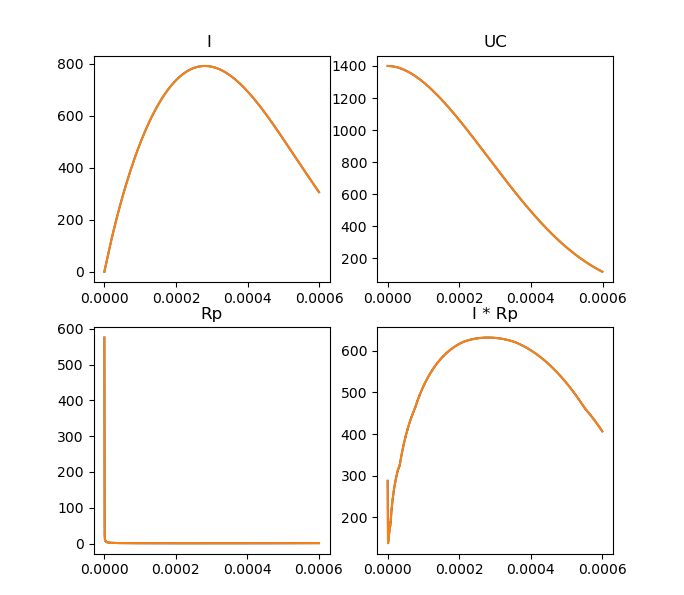


Шаг 1e-5:

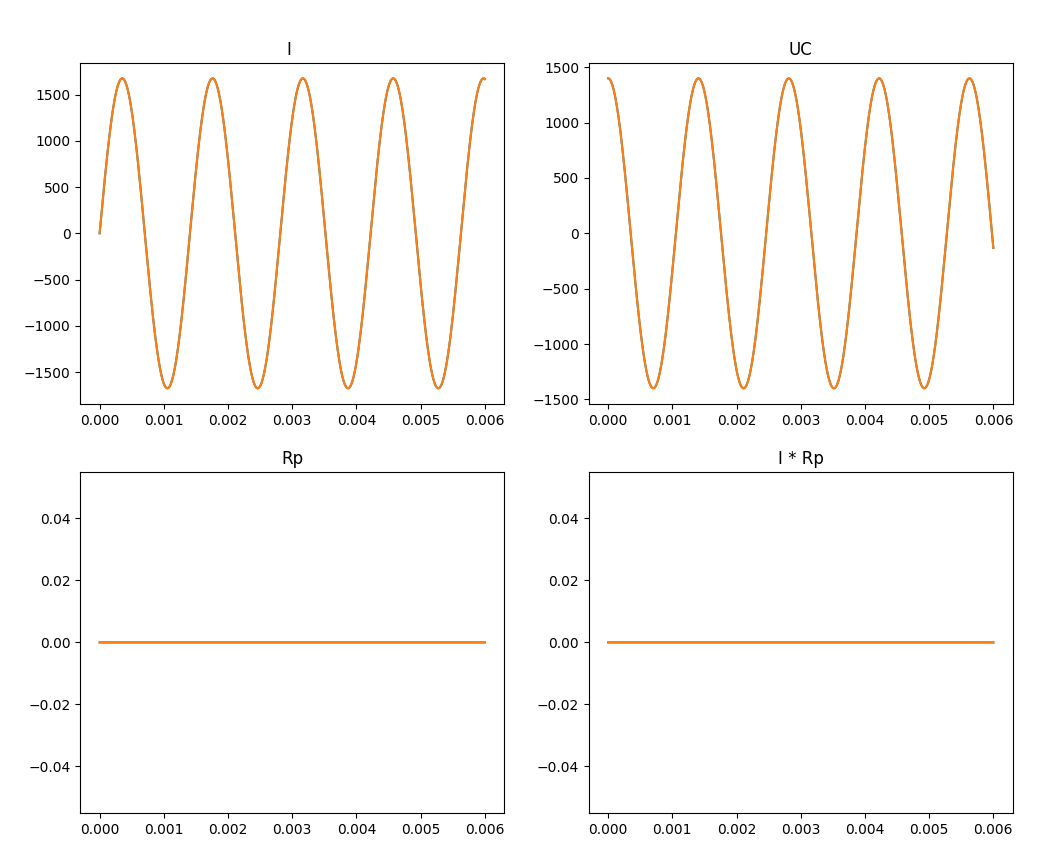
Синие графики построены с использованием метода Рунге-Кутта второго порядка, оранжевые - четвертого.

Мы видим, что при большем шаге (1e-4) разница между значениями, полученными разными методами достаточно велика. При этом метод четвертого порядка дает более точные результаты. При уменьшении шага разница значений также становится меньше.

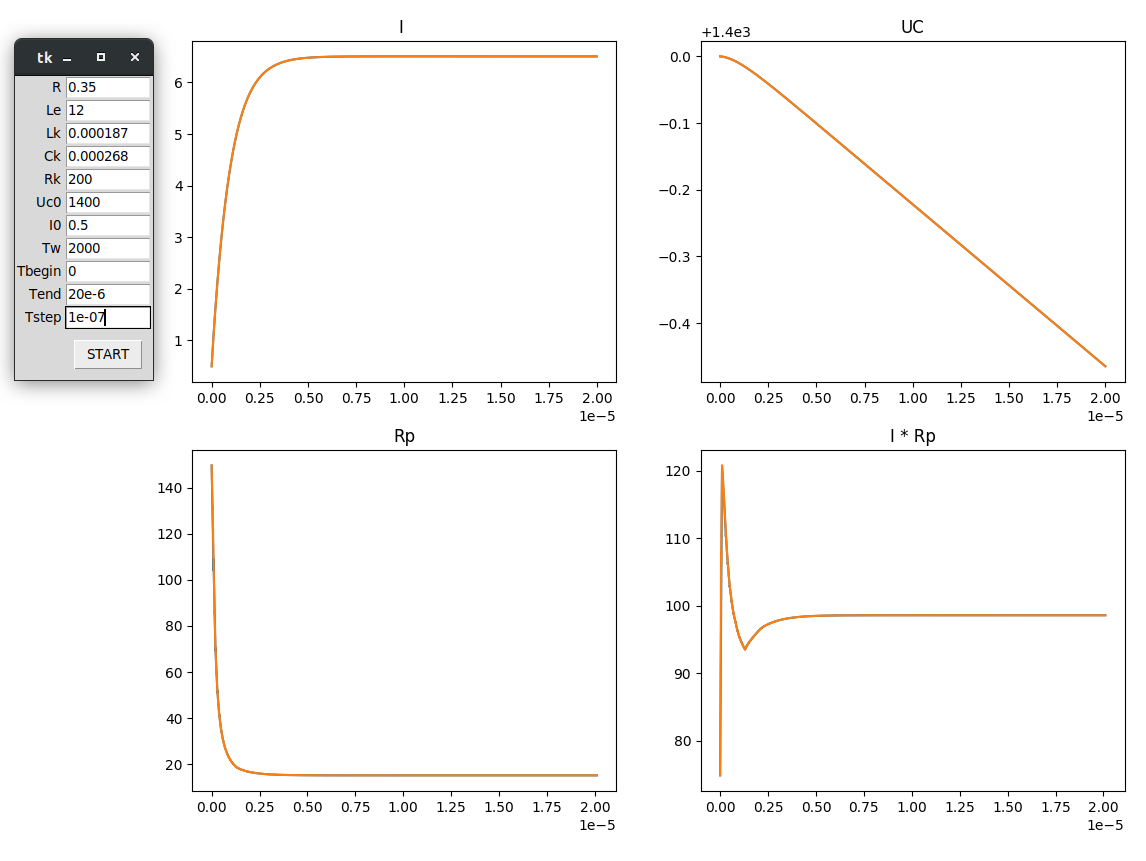
При достаточно маленьком шаге (1e-6) графики совпали:



3. График зависимости при .

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

4. График зависимости при Ом в интервале значений 0-20 мкс.



**Текст программы**

|  |
| --- |
| *Листинг 1. Метод Рунге-Кутта второго порядка точности.*   1. def RungeKutta2(x0, y0, z0, h): 2. alpha = 0.5 3. nh = h / (2 \* alpha) 4. k1 = functionF\_2(x0, y0, z0) 5. q1 = functionPHI(x0, y0, z0) 6. k2 = functionF\_2(x0 + nh, y0 + nh \* k1, z0 + nh \* q1) 7. q2 = functionPHI(x0 + nh, y0 + nh \* k1, z0 + nh \* q1) 8. y1 = y0 + h \* ((1 - alpha) \* k1 + alpha \* k2) 9. z1 = z0 + h \* ((1 - alpha) \* q1 + alpha \* q2) 10. return y1, z1 |
| *Листинг 2. Метод Рунге-Кутта четвертого порядка точности.*   1. def RungeKutta4(xn, yn, zn, hn): 2. hn2 = hn / 2 3. k1 = hn \* functionF\_4(xn, yn, zn) 4. q1 = hn \* functionPHI(xn, yn, zn) 5. k2 = hn \* functionF\_4(xn + hn2, yn + k1 / 2, zn + q1 / 2) 6. q2 = hn \* functionPHI(xn + hn2, yn + k1 / 2, zn + q1 / 2) 7. k3 = hn \* functionF\_4(xn + hn2, yn + k2 / 2, zn + q2 / 2) 8. q3 = hn \* functionPHI(xn + hn2, yn + k2 / 2, zn + q2 / 2) 9. k4 = hn \* functionF\_4(xn + hn, yn + k3, zn + q3) 10. q4 = hn \* functionPHI(xn + hn, yn + k3, zn + q3) 11. yn\_1 = yn + (k1 + 2 \* k2 + 2 \* k3 + k4) / 6 12. zn\_1 = zn + (q1 + 2 \* q2 + 2 \* q3 + q4) / 6 13. return yn\_1, zn\_1 |
| *Листинг 3. Функции F и PHI*   1. def functionF\_2(t, I, U): 2. Rp = calculateRp(I) 3. return ((U - (data['Rk'] + Rp) \* I) / data['Lk']) 4. def functionPHI(t, I, U): 5. return -1 / data['Ck'] \* I |
| *Листинг 4. Вычисление Rp, метод Симпсона и интерполяция*   1. def calculateRp(I): 2. R = data['R'] 3. integral = integrateSimpson(I) 4. return data['Le'] / (2 \* math.pi \* R \* R \* integral) 5. def integrateSimpson(I): 6. n = 40 7. begin = 0 8. end = 1 9. width = (end - begin) / n 10. result = 0 11. for step in range(n): 12. x1 = begin + step \* width 13. x2 = begin + (step + 1) \* width 14. result += (x2 - x1) / 6.0 \* (siqmaFunc(I, x1) + 4.0 \* siqmaFunc(I, 0.5 \* (x1 + x2)) + siqmaFunc(I, x2)) 15. return result 16. def siqmaFunc(I, z): 17. m = interpolate(ItK, I, 0, 2) 18. T0 = interpolate(ItK, I, 0, 1) 19. Tz = getTz(T0, m, z) 20. siqma = interpolate(Tsigma, Tz, 0, 1) 21. return siqma 22. def getTz(T0, m, r): 23. z = r / data['R'] 24. return (data['Tw'] - T0) \* math.pow(z, m) + T0 |
| *Листинг 5. Интерполяция*   1. def interpolate(table, xValue, xIndex, yIndex): 2. interpolateIndexFound = False 3. x1 = 0 4. x2 = 0 5. y1 = 0 6. y2 = 0 7. yResult = 0 8. for i in range(len(table) - 1): 9. if (table[i][xIndex] <= xValue and table[i + 1][xIndex] >= xValue): 10. y1 = table[i][yIndex] 11. y2 = table[i + 1][yIndex] 12. x1 = table[i][xIndex] 13. x2 = table[i + 1][xIndex] 14. interpolateIndexFound = True 15. if (interpolateIndexFound): 16. yResult = y1 + ((xValue - x1) / (x2 - x1)) \* (y2 - y1) 17. else: 18. if (xValue < table[0][xIndex]): 19. yResult = table[0][yIndex] 20. if (xValue > table[len(table) - 1][xIndex]): 21. yResult = table[len(table) - 1][yIndex] 22. return yResult |

**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.

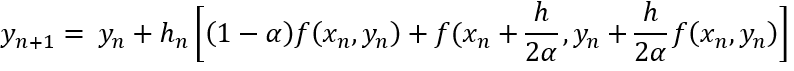
Также при тестировании нужно учесть, что работа моделируемого электрического контура описывается теоретически законами физики. Поэтому проверяются ситуации, когда сопротивление в контуре нулевое ,и он становится колебательным, а также когда сопротивление наоборот велико.

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

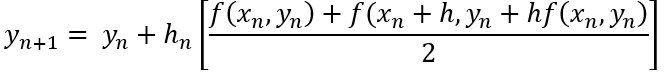
Неявный метод трапеций – это метод Рунге-Кутта второго порядка точности с

α = 0.5

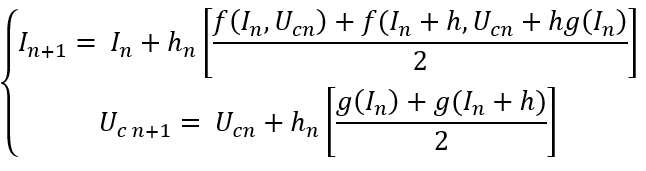
Уравнение



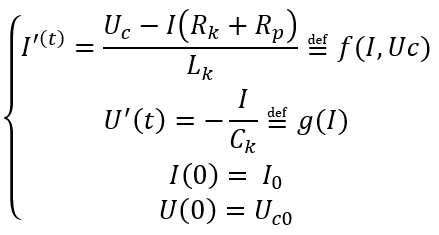
сводится к уравнению



Получаем систему разностных уравнений:



Имеется исходная система:



Подставляя уравнения производных в разностные уравнения, можно найти решение итерационно, так как на i-м ходу известны Ii, Uci, hi.

Подставив их в полученные формулы, находим Ii+1 и Uci+1, hi+1 известно заранее (например, шаг может быть постоянным).

**3. 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.