**МІНІСТЕРСТВО ОСВІТИ І НАУКИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**



**ІНТЕГРУВАННЯ СИСТЕМ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ. РОЗРАХУНОК ПЕРЕХІДНОГО ПРОЦЕСУ ДЛЯ RLC-ЛАНОК**

**Лабораторна робота №5**

**Виконав:**

**ст. гр. ІР-21**

**Левицький М. І.**

**Прийняла:**

**Доцент каф. КСА**

**Дзелендзяк У.Ю.**

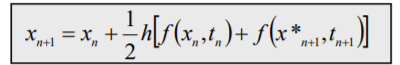
**Львів 2020**

**Мета роботи:** вивчити основні методи розв’язування систем диференціальних рівнянь першого порядку.

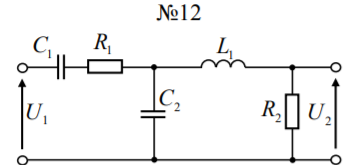
Завдання:

|  |  |
| --- | --- |
| № п/п |  |
| 12 | Модифікований метод Ейлера |

Розрахункова формула:



Задана схема:



**Код програми:**

from math import sin, pi

import matplotlib.pylab as plt

U\_max = 100

frequency = 50

R1 = 5

R2 = 4

R3 = 7

R4 = 2

L = 0.01

C1 = 300 \* 10 \*\* -6

C2 = 150 \* 10 \*\* -6

size\_of\_step = 0.00001

start\_values = [0, 0]

time\_to\_finish = 0.2

curr\_time = 0

input\_system = [lambda cur\_time, dynamic\_i3\_ul: ((R2 \* ((U\_max \* sin(2 \* pi \* frequency \* cur\_time)) - dynamic\_i3\_ul[0] \* R2) / R3 + R1)

                                                 - dynamic\_i3\_ul[0] \* R4 / L),

                lambda cur\_time, value: ((U\_max \* sin(2 \* pi \* frequency \* cur\_time) + value[0] \* R2) / (R1 + R2))]

def calculate\_dynamic\_values(time\_now, time\_next\_iteration, dynamic\_value, step):

    next\_value = dynamic\_value

    for i in range(len(dynamic\_value)):

        next\_value[i] = dynamic\_value[i] + step \* \

            input\_system[i](time\_now, dynamic\_value)

    return next\_value

def calculate\_output(time\_now, time\_final, dynamic\_i3\_ul, step):

    iterational\_result = dict()

    while time\_now < time\_final:

        next\_cur\_time = time\_now + step

        next\_value = calculate\_dynamic\_values(

            time\_now, next\_cur\_time, dynamic\_i3\_ul, step)

        time\_now = next\_cur\_time

        dynamic\_i3\_ul = next\_value

        iterational\_result[time\_now] = dynamic\_i3\_ul[0] \* L

    return iterational\_result

def main():

    iterational\_result = calculate\_output(

        curr\_time, time\_to\_finish, start\_values, size\_of\_step)

    time\_now\_list = []

    dynamic\_i3\_ul = []

    for time, i3\_ul in iterational\_result.items():

        time\_now\_list.append(time)

        dynamic\_i3\_ul.append(i3\_ul)

    plt.plot(time\_now\_list, dynamic\_i3\_ul)

    plt.show()

if \_\_name\_\_ == '\_\_main\_\_':

    main()

**Результат програми:**

