

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Запорізький національний технічний університет

## **МЕТОДИЧНІ ВКАЗІВКИ**

**до виконання лабораторних робіт з дисципліни**  
**"Моделювання електромеханічних систем"**  
для студентів спеціальності 141 «Електроенергетика, електротехніка  
та електромеханіка» (освітня програма - Електричні машини та  
апарати) усіх форм навчання

Методичні вказівки до виконання лабораторних робіт з дисципліни «Моделювання електромеханічних систем» для студентів спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» (освітня програма - Електричні машини та апарати) усіх форм навчання /Укл.: Т.М. Корнус, О.О. Каплієнко, С.І.Шило., В.В. Василевський – Запоріжжя: ЗНТУ, 2016. - 72с.

Укладачі: Т.М. Корнус, старший викладач кафедри Е та ЕА  
О.О. Каплієнко, старший викладач кафедри Е та ЕА  
С.І. Шило, старший викладач кафедри Е та ЕА  
В.В. Василевський, інженер кафедри Е та ЕА

Рецензент: Л.Б. Жорняк, к.т.н., доцент кафедри Е та ЕА

Відповідальний  
за випуск: Андрієнко П.Д., д.т.н., професор, зав. каф. Е та ЕА.

Затверджено  
на засіданні кафедри «Е та ЕА»  
протокол №1  
від 16 серпня 2016р.

Рекомендовано до видання НМК  
Електротехнічного факультету  
протокол №1  
від 20 вересня 2016р.

**ЗМІСТ**

1 ЛАБОРАТОРНА РОБОТА №1.....	4
2 ЛАБОРАТОРНА РОБОТА №2.....	14
3 ЛАБОРАТОРНА РОБОТА №3.....	23
4 ЛАБОРАТОРНА РОБОТА №4.....	31
5 ЛАБОРАТОРНА РОБОТА №5.....	36
6 ЛАБОРАТОРНА РОБОТА №6.....	43
7 ЛАБОРАТОРНА РОБОТА №7.....	48
8 ЛАБОРАТОРНА РОБОТА №8.....	55
9 ЛАБОРАТОРНА РОБОТА №9.....	63
10 ЛАБОРАТОРНА РОБОТА №10.....	69
ПЕРЕЛІК ПОСИЛАНЬ .....	72

## 1 ЛАБОРАТОРНА РОБОТА №1

ТЕМА: Моделювання сталих електричних процесів у середі програмування MATLAB. Частина 1.

МЕТА РОБОТИ: Вивчити основні оператори MATLAB; принципи побудови скрипт-файлів і підпрограм; організації введення і виведення інформації; обробки результатів моделювання.

---

### Математична модель

Для виконання роботи необхідно ретельно розглянути приклад складання математичної моделі електричних процесів і їх аналізу в простому електричному колі (рис.1.1) за допомогою системи MATLAB.

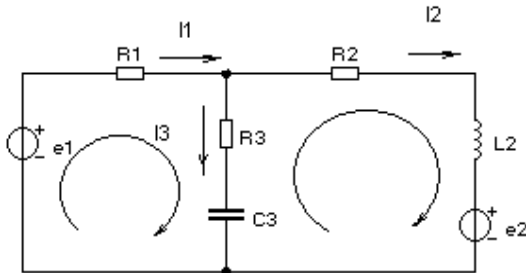


Рисунок 1.1 – Електрична схема, що моделюється.

Задані такі параметри елементів схеми:

$$e1(t)=100\sin(\omega t+\pi/6) \text{ В}; \quad e2(t)=200\sin((\omega t+\pi/3) \text{ В};$$

$$\omega=2\pi f; \quad f=50 \text{ Гц};$$

$$R1=1 \text{ Ом}; \quad R2=2 \text{ Ом}; \quad R3=3 \text{ Ом};$$

$$C3=1 \cdot 10^{-4} \text{ Ф}; \quad L2=1 \cdot 10^{-3} \text{ Гн}.$$

Потрібно визначити сталі струми в гілках.

Перш, ніж складати програму, запишемо систему рівнянь Кірхгофа в комплексній формі.

Тоді комплексне зображення синусоїдальних ЕДС

$$\underline{E1} = \frac{100}{\sqrt{2}} e^{j\frac{\pi}{6}}; \quad \underline{E2} = \frac{200}{\sqrt{2}} e^{j\frac{\pi}{3}}.$$

Реактивні опори

$$X_L = \omega L2; \quad X_C = \frac{1}{\omega C3}.$$

Комплексне зображення опорів гілок

$$\underline{Z1} = R1; \quad \underline{Z2} = R2 + jX_L; \quad \underline{Z3} = R3 - jX_C.$$

Система рівнянь Кірхгофа для визначення струмів в гілках

$$\begin{cases} \underline{I1} - \underline{I2} - \underline{I3} = 0 \\ \underline{Z1} \cdot \underline{I1} + \underline{Z3} \cdot \underline{I3} = \underline{E1} \\ \underline{Z2} \cdot \underline{I2} - \underline{Z3} \cdot \underline{I3} = -\underline{E2} \end{cases}$$

Перепишемо систему рівнянь як

$$\begin{cases} \underline{I1} & -\underline{I2} & -\underline{I3} = 0 \\ \underline{Z1} \cdot \underline{I1} & +0 \cdot \underline{I2} & +\underline{Z3} \cdot \underline{I3} = \underline{E1} \\ 0 \cdot \underline{I1} & +\underline{Z2} \cdot \underline{I2} & -\underline{Z3} \cdot \underline{I3} = -\underline{E2} \end{cases}$$

Представимо коефіцієнти перед невідомими у вигляді матриці

$$M = \begin{bmatrix} 1 & -1 & -1 \\ \underline{Z1} & 0 & \underline{Z3} \\ 0 & \underline{Z2} & -\underline{Z3} \end{bmatrix};$$

Вектор правих частин

$$F = \begin{bmatrix} 0 \\ \underline{E1} \\ -\underline{E2} \end{bmatrix}.$$

Тоді наведена вище система рівнянь, записана в матричному вигляді, виглядатиме як

$$M \cdot I = F,$$

де

$$I = \begin{bmatrix} \underline{I1} \\ \underline{I2} \\ \underline{I3} \end{bmatrix} \text{ – вектор-стовпець шуканих струмів.}$$

Вирішення цього матричного рівняння дає вектор шуканих струмів в гілках електричного кола.

### **Етапи розробки програми**

Щоб розробити програму, що реалізує наведену вище математичну модель в середовищі MATLAB, необхідно виконати наступні завдання:

- організувати введення початкових даних;
- скласти матриці системи лінійних рівнянь;
- вирішити систему лінійних рівнянь;
- визначити похибку обчислень;
- організувати виведення результатів розрахунку в числовому та\або графічному вигляді.

### **Виклик DeskTop MATLAB**

Роботу в середовищі MATLAB починають з виклику пакету в меню «Пуск».

Введення програми можна здійснювати безпосередньо з командного вікна, послідовно викликаючи до виконання кожного оператора. Проте доцільніше користуватися для цієї мети вікном редактора програм. Для цього викличемо File⇒New з головного меню

або відповідну ікону на панелі інструментів. З'явиться вікно редактора тексту, в якому необхідно набрати оператори програми.

### **Файли-сценарії (Script-files) та файли-функції (Function-files)**

Файли, що містять команди і оператори MATLAB, називаються М-файлами (M-files).

Існують два типи М-файлів (M-files):

- файли-сценарії (Script-files), що не мають вхідних параметрів;
- файли-функції (Function-files), що мають вхідні параметри.

### **Введення початкових даних методом привласнення. Коментарі**

На першому етапі знайомства з пакетом MATLAB виконаємо введення початкових даних, використовуючи оператор привласнення безпосередньо в тексті програми.

Почнемо програму з введення коментарів.

Кожен рядок коментарів починається із знаку %. Слід пам'ятати, що коментарі російською мовою можна вводити тільки заголовними буквами.

Можна записувати кожний оператор MATLAB як в окремому рядку, так і послідовні декілька операторів в одному рядку. Якщо оператори записані в стовпець, тобто поодиночі в рядку, після кожного з них можна ставити знак ;. Це означає, що результат обчислень автоматично не буде виводитись на екран при виконанні програми.

Якщо знак ; не ставити – то після виконання оператора буде виводитись результат його виконання. Оператори, які записані в одному рядку необхідно відділяти знаком ;.

Оператори введення даних з відповідними коментарями наведені в тексті програми Circuit 1 (см.п.1.10).

### **Організація обчислень. Верифікація результатів**

У тексті програми Circuit 1 (див. п.1.10) знаходиться код, який реалізує етап обчислень E1, E2, X1,xс,z1,z2,z3, а також формування матриць M і F.

Примітка: число  $\pi$  є константою MATLAB і викликається знаком pi.

Для перевірки правильності організації обчислень розрахуємо  $\varepsilon$  - похибку обчислень по зовнішньому контуру схеми рис.1.1.

### Виведення результатів обчислень

Для виведення результатів скористаємося оператором `disp`, який дозволяє вивести на друк символічні результати. Оскільки MATLAB не дозволяє одним оператором виводити на друк і числа, і символи, застосуємо стандартну функцію `num2str(X)`, що перетворює числову змінну  $X$  в змінну типа «строка» (див. текст програми `Circuit 1` в п.1.10).

### Збереження програми і створення шляху пошуку `Search path`

Для збереження програми викличте з меню `File` опцію `Save as` і збережіть створений `M`-файл в особистій директорії під ім'ям `Circuit 1`.

Для того, щоб з MATLAB можна було викликати створений вами файл, вкажемо йому шлях пошуку. Для цього виберемо `Set Path` з меню `File` на робочому столі (`desktop`) MATLAB і використаємо діалогове вікно.

### Виконання програми

Щоб виконати програму, досить в командному вікні (`Command Window`) після знаку запрошення `>>` ввести ім'я `M`-файлу, в нашому випадку це `Circuit 1`, і натиснути `ENTER`. Результати обчислень будуть виведені в командному вікні.

### Текст програми і лістинг її виконання

```
%ПРИКЛАД РОЗРАХУНКУ КОЛА СИНУСОЇДАЛЬНОГО СТРУМУ
% ПРОГРАМА Circuit 1
f=50;
R1=1;
R2=2;
R3=3;
L2=1e-3;
C3=1e-4;
```



```
ph1=30;
ph2=60;
```

```
XL=2*pi*f*L2;
XC=1/(2*pi*f*C3);
E1=100/sqrt(2)*exp(ph1*pi/180*i);
E2=200/sqrt(2)*exp(ph2*pi/180*i);
Z1=R1; Z2=R2+i*XL; Z3=R3-i*XC;
M=[1 -1 -1;
   Z1 0 Z3;
   0 Z2 -Z3];
F=[0; E1; -E2];
I=M\F;
```

```
disp(['I1=',num2str(I(1,1))]);
disp(['I2=',num2str(I(2,1))]);
disp(['I3=',num2str(I(3,1))]);
```

*% ПЕРЕВІРКА ПО 2 ЗАКОНУ КІРХГОФА*

```
eps=E1-E2-Z1*I(1,1)-Z2*I(2,1);
disp([' error calculation eps=',num2str(eps)]);
```

*% ОБЧИСЛЕННЯ МОДУЛІВ І АРГУМЕНТІВ (У РАДІАНАХ)  
СТРУМІВ, ЩО ОБЧИСЛЮЮТЬСЯ*

```
I1M=abs(I(1,1));
I2M=abs(I(2,1));
I3M=abs(I(3,1));
```

```
phI1=angle(I(1,1));
phI2=angle(I(2,1));
phI3=angle(I(3,1));
disp(['I1M=',num2str(I1M)]); disp(['phI1=',num2str(phI1)]);
disp(['I2M=',num2str(I2M)]); disp(['phI2=',num2str(phI2)]);
disp(['I3M=',num2str(I3M)]); disp(['phI3=',num2str(phI3)]);
```

Результати обчислень:

```
>> I1=-7.37618-26.903i
```

```

I2=-5.63886-29.2223i
I3=-1.7373+2.3193i
error calculation eps=2.6645e-015-1.4211e-014i
I1M=27.8959
phI1=-1.8384
I2M=29.7614
phI2=-1.7614
I3M=2.8978
phI3=2.2137
>>

```

Висновки: похибка розрахунку складає  $10^{-15}$ , отже, розрахунок виконано вірно.

### Організація введення початкових даних в діалоговому режимі

Введемо початкові дані за допомогою оператора **input**:

```

R1=input('input value R1=');
R2=input('input value R2=');
R3=input('input value R3=');
L2=input('input value L2=');
C3=input('input value C3=');
ph1=input('input value ph1=');
ph2=input('input value ph2=');

```

Для цього:

- викличемо М-файл Circuit 1 і проведемо відповідні зміни введення даних;
- переведемо при виведенні результатів розрахунку аргументи phI1, phI2, phI3 струмів I1, I2, I3 з радіан в градуси:

```

disp(['I1M=', num2str(I1M), ' phI1=', num2str(phI1/pi*180)]);
disp(['I2M=', num2str(I2M), ' phI2=', num2str(phI2/pi*180)]);
disp(['I3M=', num2str(I3M), ' phI3=', num2str(phI3/pi*180)]);

```

- замінимо відповідні оператори виводу;

- збережемо програму як Circuit 2 й «пропишемо» в MATLAB (см.п.1.8);
- з Command Window викличемо Circuit 2 на виконання (см.п.1.9);
- введемо чисельні значення початкових даних в діалоговому режимі. Після кожного числа, що вводиться, натискуйте ENTER.

Результати розрахунку за програмою Circuit 2 повинні повністю збігатися з результатами розрахунку за програмою Circuit 1, за винятком аргументів  $ph1, ph2, ph3$  комплексних чисел  $I1, I2, I3$ , значення яких при виводі переводяться з виміру в радіанах у вимір в градусах.

```
>> circuit 2
input value R1=1
input value R2=2
input value R3=3
input value L2=1e-3
input value C3=1e-4
input value ph1=30
input value ph2=60
I1=1.31391+30.2501i
I2=-0.904861+28.5767i
I3=2.2188+1.6734i
error calculation eps=1.2434e-014+2.1316e-014i
I1M=30.2786 phI1=87.5129
I2M=28.591 phI2=91.8136
I3M=2.7791 phI3=37.0242
>>
```

### Створення підпрограми-функції (Function-file)

Для подальшого виводу на графік залежностей струмів в гілках розрахуємо масиви цих функцій за допомогою підпрограми-функції MATLAB.

Математично залежність струму в гілці від часу виражається

$$i(t) = \sqrt{2} \cdot I \cdot \sin(2 \cdot \pi \cdot f \cdot t + \varphi),$$

де  $I$  – модуль комплексу струму (значення діючого струму)

$\varphi$  - аргумент комплексу струму..

Реалізуємо підпрограму-функцію обчислення  $i(t)$  як `current(IM,f,phI,t)`,

де значення формальних параметрів

$IM$  - модуль комплексу струму  $I$ ;

$f$  - частота;

$phI$  - аргумент комплексу струму  $\varphi$ ;

$t$  - поточний час.

Відкрисмо новий файл в редакторі тексту і наберемо наступну програму-функцію:

```
function im=current(IM,f,phI,t)
im=sqrt(2)*IM*sin(2*pi*f*t+phI);
```

Збережемо створену підпрограму в особистій теці як `current` і «пропишемо» в MATLAB.

### Графічний вивід

Послідовність дій:

- викличемо М-файл `Circuit 2`;
- запишемо в кінці програми фрагмент, що реалізує заповнення масивів миттєвих значень струмів в гілках і вивід їх в графічному вигляді:

```
% GRAPH
h=0.04/200;
t(1)=0;
i1(1)=current(I1M,f,phI1,t(1));
i2(1)=current(I2M,f,phI2,t(1));
i3(1)=current(I3M,f,phI3,t(1));
for k=2:201
    t(k)=t(k-1)+h;
    i1(k)=current(I1M,f,phI1,t(k));
```

```
i2(k)=current(I2M,f,phI2,t(k));
```

```
i3(k)=current(I3M,f,phI3,t(k));
```

```
end
```

```
plot(t,i1,t,i2,t,i3);
```

- збережемо програму як Circuit 3 і «пропишемо» її в MATLAB (см.п.1.8);

- з Command Window викличемо Circuit 3 на виконання (см.п.1.9).

Отримані графічні залежності струмів в гілках від поточного часу наведені на рис.1.2.

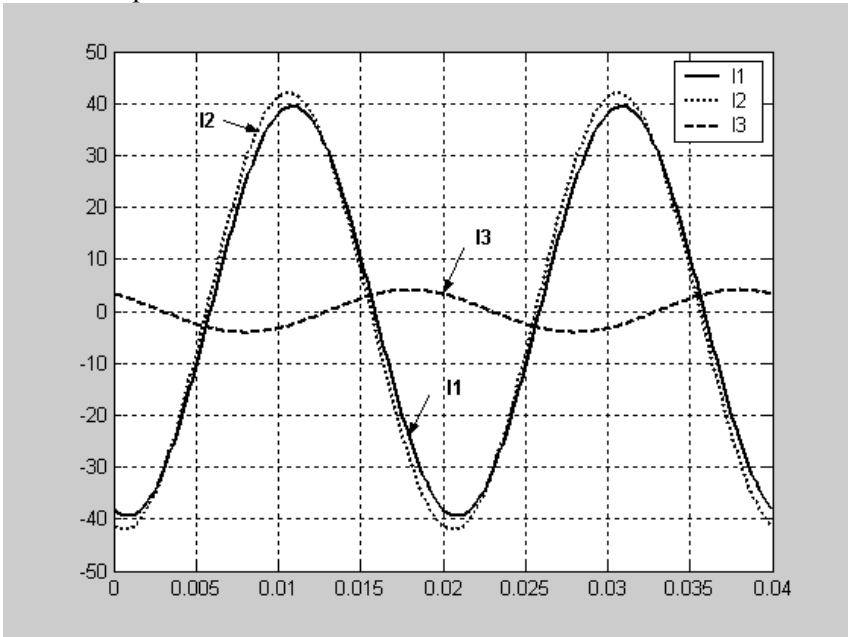


Рисунок 1.2 – Зависимости токов в ветвях от текущего времени.

- Оформіть графіки, як показано на рис.1.2;
- Виконайте програму ще раз, змінивши початкові дані:  
 $e1(t)=200\sin(\omega t+\pi/2)$  В;      $e2(t)=80\sin((\omega t+\pi/3)$  В;  
 $R1=0.5$  Ом;      $R2=1$  Ом;      $R3=4$  Ом;  
 $C3=2\cdot 10^{-4}$  Ф;      $L2=4\cdot 10^{-3}$  Гн.
- Подивіться, як це вплине на результат розрахунку.
- У полі графіка запишіть значення початкових даних.

## 2 ЛАБОРАТОРНА РОБОТА №2

ТЕМА: Моделювання сталих електричних процесів в у середі програмування MATLAB. Частина 2.

МЕТА РОБОТИ: Вивчити основні принципи відладки MATLAB-програм; верифікація результатів розрахунку.

---

Як правило, всі програми містять помилки і вимагають відладки, причому практика показує, що відладка і верифікація результатів займає, приблизно, 80% часу роботи над програмою. Помилки бувають двох типів: синтаксичні і логічні.

Оскільки мова MATLAB передбачає чітку формалізацію написання операторів і інших компонентів програми і чітке виконання математичних операцій, то порушення цих правил у багатьох випадках контролюється самою системою MATLAB. В результаті система видає діагностичне повідомлення про передбачувану помилку.

Логічні помилки пов'язані з наступними чинниками:

- невідповідність вербального формулювання поставленого завдання реальному об'єкту
- невідповідність математичного і вербального формулювань;
- помилки в алгоритмі математичної постановки завдання;
- помилки в програмі, що реалізовує даний алгоритм
- помилки введення тесту програми
- помилки введення початкових даних.

У ряді випадків логічні помилки призводять до порушення синтаксису програми і виявляються на етапі трансляції. Проте, в більшості випадків, сама система логічні помилки визначити не може. Виявлення логічних помилок є обов'язком самого програміста. Тому програміст повинен добре представляти всю логіку вирішуемого завдання в цілому і критерії перевірки адекватності обчислення проміжних даних і остаточних результатів. Взагалі, пошук логічних помилок не має єдиних правил і є творчим завданням.

Проте існують спільні рекомендації по відладці. Наприклад, на етапі програмування або алгоритмізації необхідно продумувати методику майбутньої відладки. У програму вводять деяку кількість

операторів, необхідних тільки на етапі відладки, наприклад, виведення деяких особливо значущих величин; перевірка деяких співвідношень, які повинні виконуватися завжди або в конкретному режимі. Програміст повинен прагнути до локалізації області помилок. Для цієї мети розроблений спеціальний підхід і модульне і структурне програмування. Про це можна прочитати в спеціальній літературі [6].

### Пошук синтаксичних помилок

Для того, щоб ознайомитися з методикою пошуку синтаксичних помилок, штучно вноситимемо їх до тексту програми і спостерігатимемо реакцію транслятора MATLAB на їх появу.

Викличемо текст програми Circuit 1. Збережемо його в тій же директорії як Circuit 1\_debug. Тепер внесемо синтаксичну помилку до тексту програми: у рядку 15 обчислень E1 приберемо останню дужку:

$$E1=100/\text{sqrt}(2)*\text{exp}(ph1*\text{pi}/180*i;$$

Збережемо отриманий файл. Запустимо програму.

Отримаємо наступне повідомлення в Command Window:

```
>> Circuit 1_debug
??? Error: File: E:\..\Circuit 1_debug.m Line: 15 Column: 31
")" expected, ";" found.
```

Це повідомлення точно локалізує внесену помилку. Виправимо її.

Тепер внесемо іншу помилку: у рядку 21 змінимо розмірність матриці F таким чином:

$$F=[0; \quad E1; \quad -E2; 0];$$

Збережемо отриманий файл і запустимо програму.

Отримаємо наступне повідомлення в Command Window:

```
>>Circuit 1 _debug
??? Error using ==> \
Matrix dimensions must agree.
```

Error in ==> E:\.\Circuit 1\_debug.m  
 On line 22 ==> I=M\F;

Це повідомлення попереджає про невідповідність розміру матриць при виконанні оператора  $I=M\F$  в рядку 22. Тепер виправимо цю помилку і збережемо файл.

Далі, внесіть помилки до написання стандартних функцій (наприклад, sqrt). Збережіть зміни, запустіть її і глянете повідомлення про помилку. Можете спробувати внести інші помилки. Не забувайте перед запуском зберігати програму, а потім виправляти внесені помилки.

### **Відладка за допомогою редактора програм Editor/Debugger**

Одним із засобів інструментального пошуку помилок є вживання вбудованої системи редагування і відладки програм Editor/Debugger.

Ознайомимося з опціями відладчика на прикладі програми Circuit1\_debug.

Перед початком сеансу доцільно очистити Workspace за допомогою команди Clear і Command Window за допомогою команди Clc ( можна використовувати з цією метою Головне меню⇒Edit⇒Clear Workspace і Головне меню⇒Edit⇒Clear Command Window ). Далі:

- запустіть відладчик програм. Для цього в головному меню Editor/Debugger натискуйте Debug⇒Run. Після цього, підводячи курсор до будь-якої змінної або масиву в тексті програми, можна прослідити їх чисельні (або символльні) значення;
- очистіть Workspace і Command Window . Встановіть курсор перед яким-небудь оператором. Хай це буде оператор в рядку 13

$$XL=2*\pi*f*L2;$$

Натискуйте на ікону Set/clear breakpoint на панелі інструментів. Червона крапка зліва від вказаного оператора позначає точку переривання при виконанні програми. Натискуйте Debug⇒Run.



Зелена стрілка, що з'явилася в тексті програми, вказує на оператор, перед яким зупинені обчислення. У Command Window ви побачите запрошення

K>>

до введення подальших вказівок. Перш, ніж ви зробите це, підведіть курсор до значень змінних, обчислених до переривання. Система покаже ці значення. Але значень змінних і масивів після переривання ви не побачите, оскільки на момент переривання вони не визначені.

Далі необхідно виконати наступні дії:

- натискуйте на ікону Step на панелі інструментів. Зелена стрілка переміститься на один рядок вниз. Програма виконала один оператор і знову зупинилася. Таким чином, можна виконувати програму по кроках, контролюючи як послідовність обчислень, так і значення обчислених змінних і масивів. Доведіть до кінця виконання програми, використовуючи покрокове обчислення. Спостерігайте зміну і появу нової інформації;

- знову запустіть програму на розрахунок (натискуйте Debug⇒Run). Для того, щоб після переривання виконати програму до кінця, натискуйте на панелі інструментів кнопку Continue. Результати обчислень ви побачите в Command Window;

- за допомогою кнопки Set/clear breakpoint можна зняти вказану курсором точку переривання. Кнопкою Clear all breakpoints можна зняти всі точки переривання в програмі;

- для проведення покрокової відладки із заходом в підпрограму виклинемо з редактора програм М-файл Circuit 3. Збережемо його в тій же директорії як Circuit 3\_debug. З метою заощадження часу замінимо операторів діалогового введення на операторів введення даних методом привласнення (використовуйте для цієї мети операторів введення М-файлу Circuit 1).

Програма повинна виглядати таким чином:

```
% ПРИКЛАД РОЗРАХУНКУ КОЛА СИНУСОЇДАЛЬНОГО СТРУМУ
% Circuit 3_debug
f=50;
R1=1;
R2=2;
```

```

R3=3;
L2=1e-3;
C3=1e-4;
ph1=30;
ph2=60;
XL=2*pi*f*L2;
XC=1/(2*pi*f*C3);
E1=100/sqrt(2)*exp(ph1*pi/180*i);
E2=200/sqrt(2)*exp(ph2*pi/180*i);
Z1=R1; Z2=R2+i*XL; Z3=R3-i*XC;
M=[1    -1    -1;
   Z1    0    Z3;
   0    Z2   -Z3];
F=[0; E1; -E2];
I=M\F;

disp(['I1=',num2str(I(1,1))]);
disp(['I2=',num2str(I(2,1))]);
disp(['I3=',num2str(I(3,1))]);
% ПЕРЕВІРКА ЗА 2 ЗАКОНОМ КІРХГОФА
eps=E1-E2-Z1*I(1,1)-Z2*I(2,1);
disp([' error calculation eps=',num2str(eps)]);
% ОБЧИСЛЕННЯ МОДУЛІВ І АРГУМЕНТІВ (У РАДІАНАХ)
ОБЧИСЛЕНИХ СТРУМІВ
I1M=abs(I(1,1));
I2M=abs(I(2,1));
I3M=abs(I(3,1));

phI1=angle(I(1,1));
phI2=angle(I(2,1));
phI3=angle(I(3,1));
disp(['I1M=',num2str(I1M),' phI1=',num2str(phI1/pi*180)]);
disp(['I2M=',num2str(I2M),' phI2=',num2str(phI2/pi*180)]);
disp(['I3M=',num2str(I3M),' phI3=',num2str(phI3/pi*180)]);
% GRAPH
h=0.04/200;
t(1)=0;

```

```

i1(1)=current(I1M,f,phI1,t(1));
i2(1)=current(I2M,f,phI2,t(1));
i3(1)=current(I3M,f,phI3,t(1));
for k=2:201
    t(k)=t(k-1)+h;
    i1(k)=current(I1M,f,phI1,t(k));
    i2(k)=current(I2M,f,phI2,t(k));
    i3(k)=current(I3M,f,phI3,t(k));
end
plot(t,i1,t,i2,t,i3);

```

Для того, щоб досліджувати роботу програми в циклі, як правило, досить зробити 2 – 3 заходи в нього. Тому замінимо верхню межу оператора циклу таким чином

```

for k=2:5
    t(k)=t(k-1)+h;
    i1(k)=current(I1M,f,phI1,t(k));
    i2(k)=current(I2M,f,phI2,t(k));
    i3(k)=current(I3M,f,phI3,t(k));
end

```

Збережіть змінену програму.

Встановіть курсор перед оператором визначення кроку обчислення значень функціям струмів в гілках

$$h=0.04/200;$$

та встановіть точку переривання.

Для контролю виконання підпрограм скористаємося опцією Step in на панелі інструментів, яка дає можливість дослідити покрокове виконання операторів підпрограми. Виконаєте програму до кінця за допомогою цієї опції.

Опція Step out дозволяє вийти з підпрограми на будь-якому етапі її виконання.

Опція Exit Debug Mode реалізує вихід з режиму відладки.

З Main menu  $\Rightarrow$  Breakpoints menu можна викликати додаткові опції відладки:

- Stop If Error – зупинення виконання програми в разі помилки;
- Stop If Warning – зупинення виконання програми в разі попередження системи про можливу помилку;
- Stop If NAN Or Inf – зупинення виконання в разі нечислового або безконечного значення обчислюваної величини.

Вживання тих або інших способів і опцій відладки програм визначається самим програмістом залежно від його досвіду, кваліфікації, ступеня складності вирішуваного завдання і індивідуального стилю програмування.

### Верифікація результатів розрахунку

Як сказано вище, система може зафіксувати логічні помилки тільки непрямым чином, але і це трапляється далеко не завжди. Самий незручний випадок для програміста, коли результати розрахунку схожі на правду. Тому завдання доказу правильності рішення має бути поставлена і реалізована на етапі розробки програми.

Зробимо це на прикладі програми розрахунку струмів в ланцюзі, реалізованому М-файлом Circuit 3\_debug.

На етапі написання програми ми здійснили перевірку за 2-им законом Кірхгофа по контуру, не включеному в систему рівнянь для знаходження струмів в гілках. Для електричних ланцюгів такого роду перевірка необхідна. Тепер перевіримо, чи достатня вона.

Для цього виконаємо наступні дії:

- Внесемо логічну помилку до програми. При заповненні матриці М замінимо  $M(1,2)=-1$  на  $M(1,2)=1$ . Тепер матриця повинна виглядати таким чином:

$$M = \begin{bmatrix} 1 & 1 & -1; \\ Z1 & 0 & Z3; \\ 0 & Z2 & -Z3; \end{bmatrix}$$

- Збережемо програму і запустимо її на розрахунок. Результати розрахунку:

$$I1 = -17.5739 - 76.7793i$$

$$I2 = 16.5153 + 79.355i$$

```

I3=-1.0586+2.5757i
  error calculation eps=0
I1M=78.7649  phI1=-102.8923
I2M=81.0554  phI2=78.2435
I3M=2.7848   phI3=112.3428
>>

```

Таким чином ми бачимо, що криві струмів на графіку схожі на реальні, а помилка розрахунку  $\text{eps}=0$ . Отже, проведеної перевірки недостатньо для виявлення даної логічної помилки.

Для електричної схеми рекомендується проводити перевірку також по 1-у закону Кірхгофа для вузла, не охопленого системою рівнянь для визначення невідомих. Організуємо розрахунок перевірки по 1-у закону Кірхгофа і по 2-у закону Кірхгофа таким чином:

```

%ПЕРЕВІРКА ПО 1 ЗАКОНУ КІРХГОФА
eps1=I(1,1)-I(2,1)-I(3,1);
%ПРОВЕРКА ПО 2 ЗАКОНУ КІРХГОФА
eps2=E1-E2-Z1*I(1,1)-Z2*I(2,1);
disp([' error calculation eps1=',num2str(eps1)]);
disp([' error calculation eps2=',num2str(eps2)]);

```

і проведемо відповідні заміни в програмі.

Збережемо програму і запусимо її на рахунок. Отримаємо наступні результати:

```

I1=-17.5739-76.7793i
I2=16.5153+79.355i
I3=-1.0586+2.5757i
  error calculation eps1=-33.0306-158.71i
  error calculation eps2=0
I1M=78.7649  phI1=-102.8923
I2M=81.0554  phI2=78.2435
I3M=2.7848   phI3=112.3428

```

Таким чином, перевірка за 1-им законом Кірхгофа дає похибку  $\text{eps1} = -33.0306 - 158.71i$ . Очевидно, що, оскільки 1-й закон Кірхгофа не виконується, то розрахунок виконано невірно.

Виправимо внесену помилку і знову запусимо програму. Отримаємо наступні значення похибок

error calculation  $\text{eps1} = 0$

error calculation  $\text{eps2} = 3.5527e-015 + 7.1054e-015i$

Величина похибки свідчить про правильність розрахунку.

### 3 ЛАБОРАТОРНА РОБОТА №3

ТЕМА: Моделювання сталих електричних процесів в середі MATLAB. Частина 3.

МЕТА РОБОТИ: Розробити і відладити програму розрахунку струмів і напруги в заданому електричному ланцюзі і довести правильність розрахунку.

---

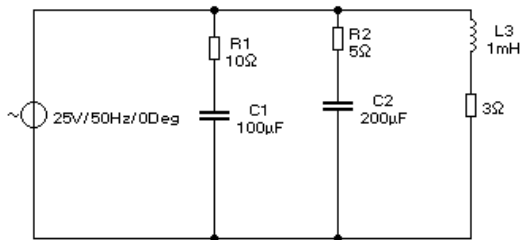
#### Завдання на виконання роботи

Для заданого варіанту електричної схеми необхідно:

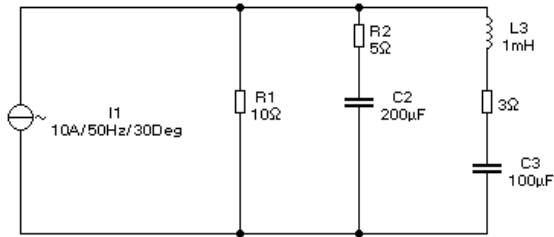
- розробити математичну модель розрахунку струмів в гілках і напруги на реактивних елементах;
- скласти програму, що реалізує розроблену модель;
- включити в програму операторів, що дозволяють провести верифікацію результатів розрахунку;
- організувати графічне виведення вказаних функцій струмів і напруги;
- відладити програму;
- результати роботи ( програму, лістинг розрахунку, графіки) зберегти в особистій директорії на жорсткому диску.

#### Індивідуальні завдання

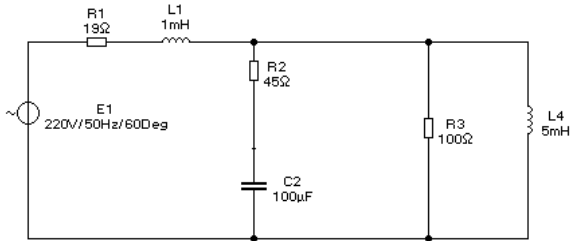
Варіант 1:



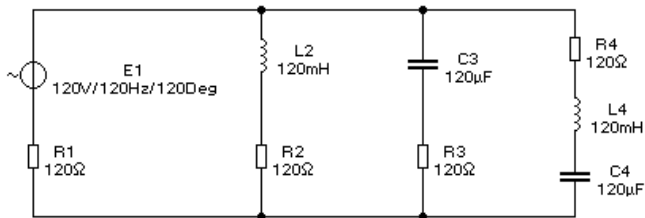
Вариант 2:



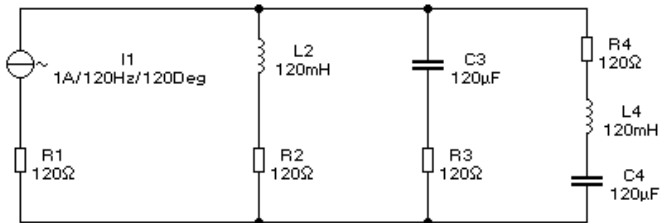
Вариант 3:



Вариант 4:

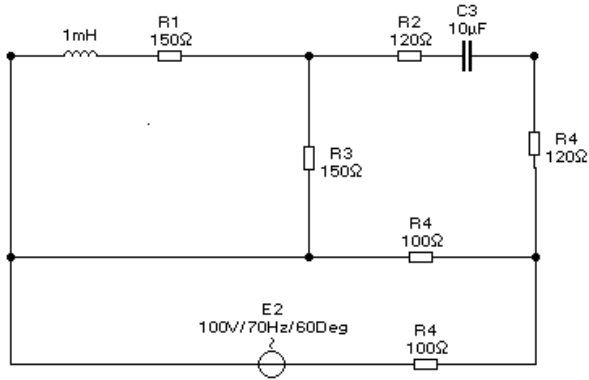


Вариант 5:

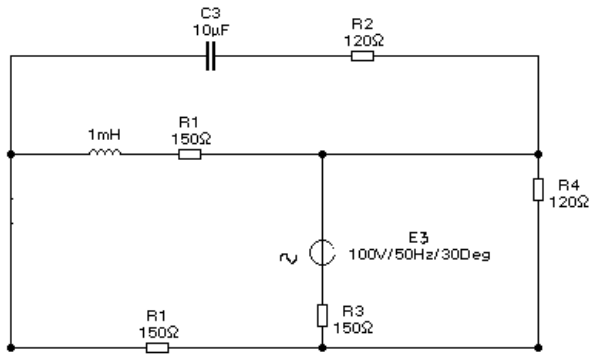




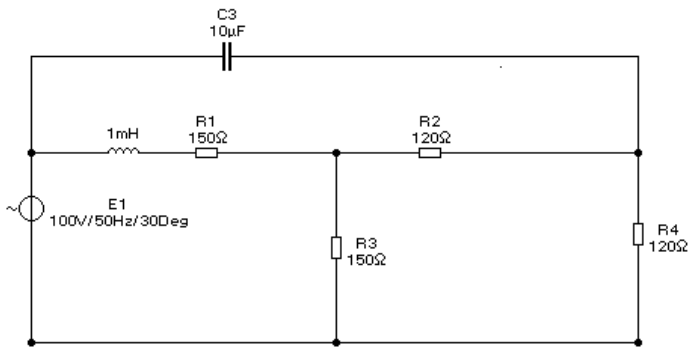
Вариант 6:



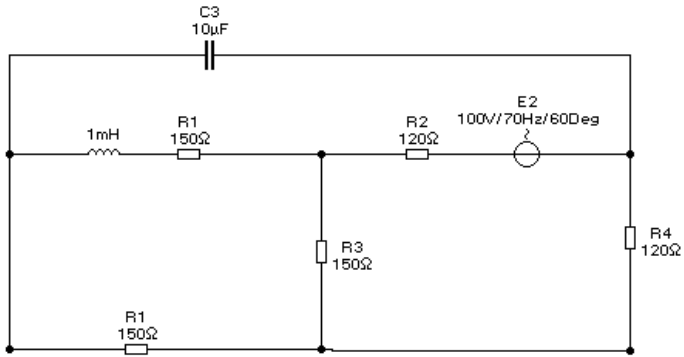
Вариант 7:



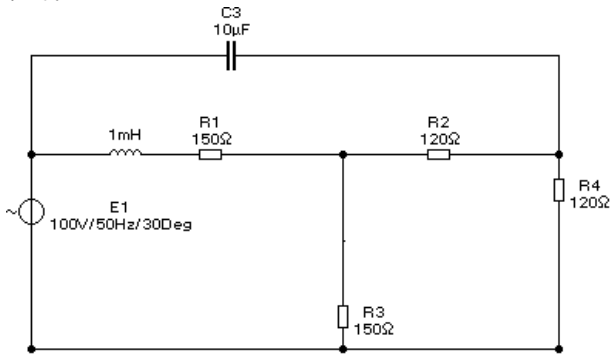
Вариант 8:



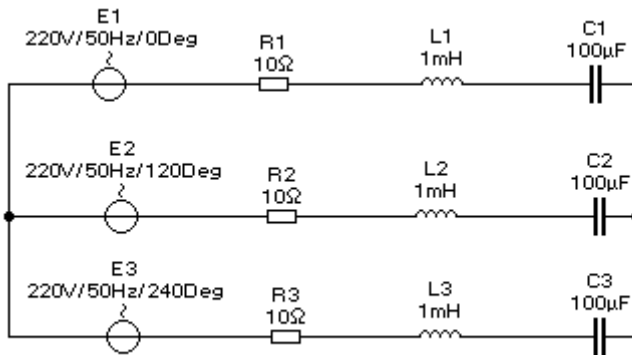
Варіант 9:



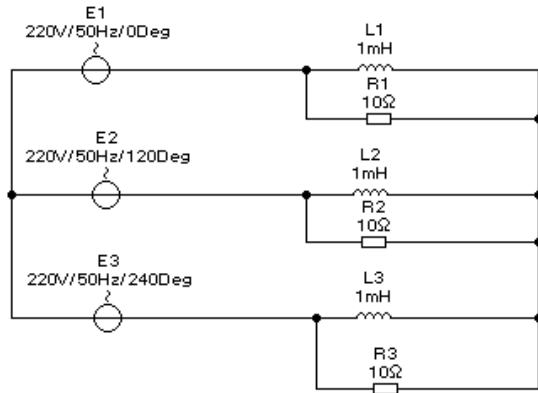
Варіант 10:



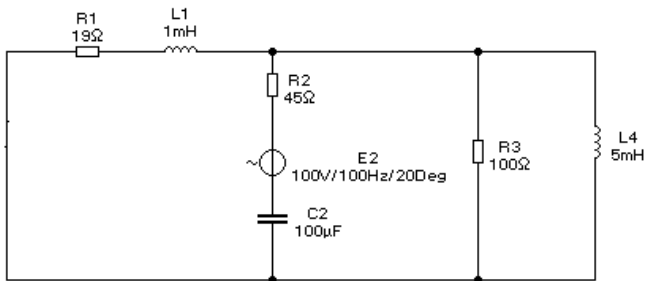
Варіант 11:



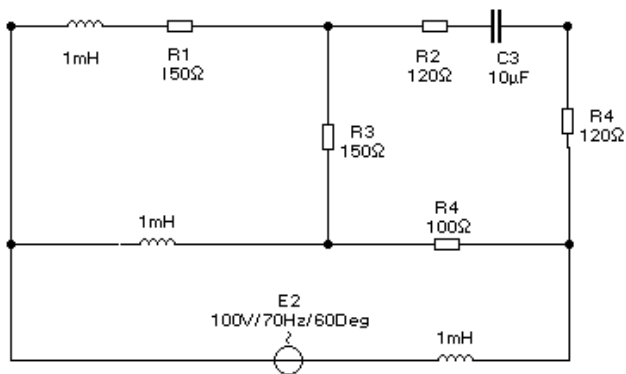
Варіант 12:



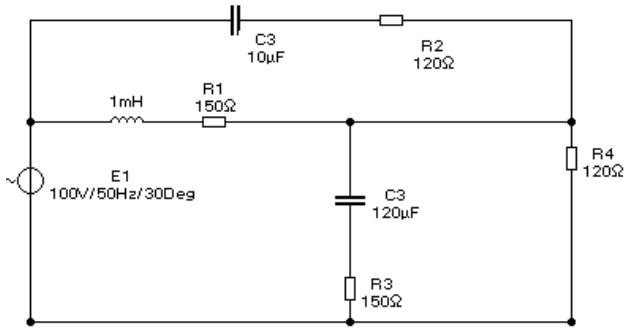
Варіант 13:



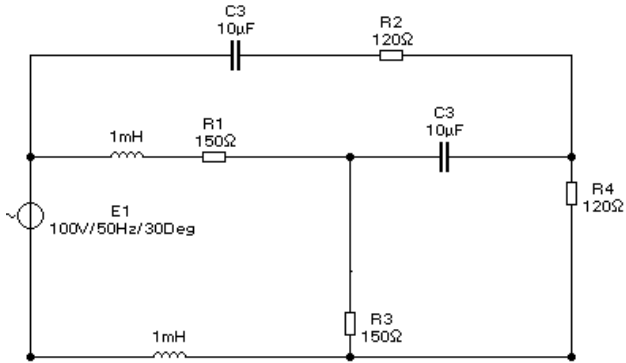
Варіант 14:



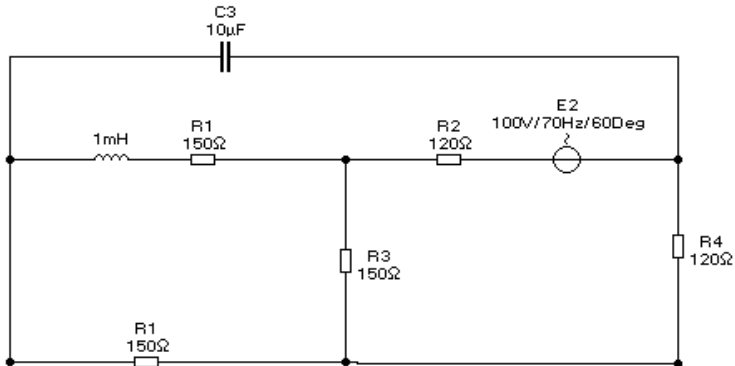
Варіант 15:



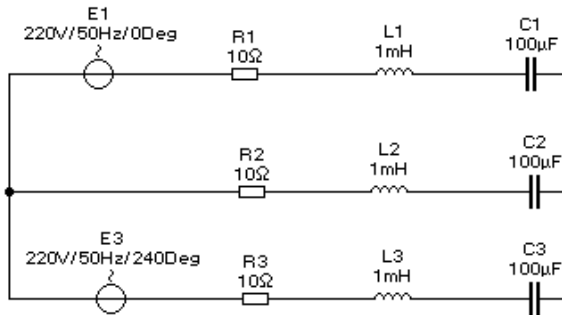
Варіант 16:



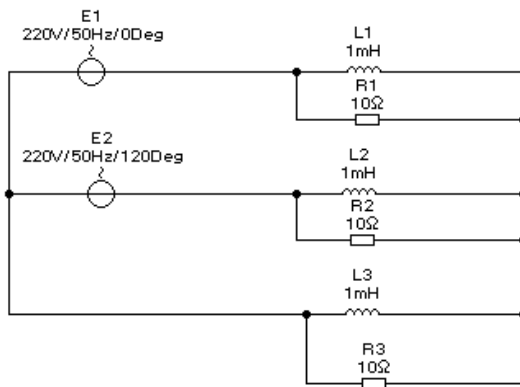
Варіант 17:



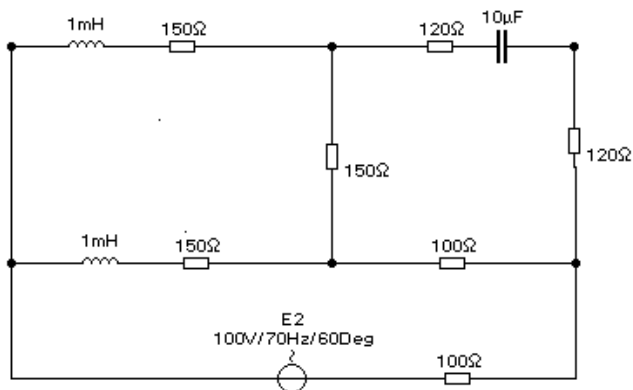
Вариант 18:



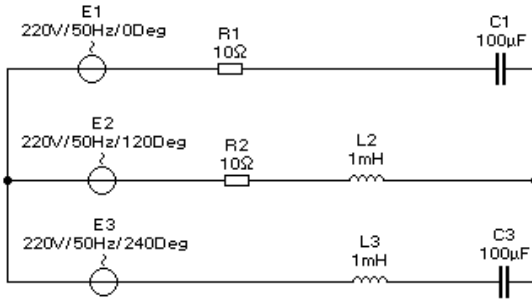
Вариант 19:



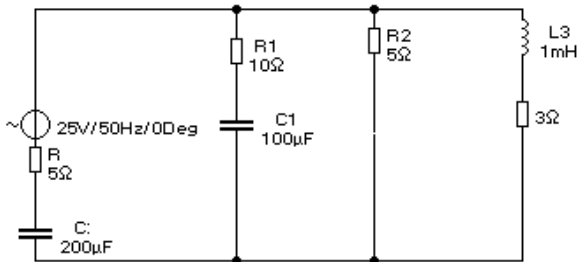
Вариант 20:



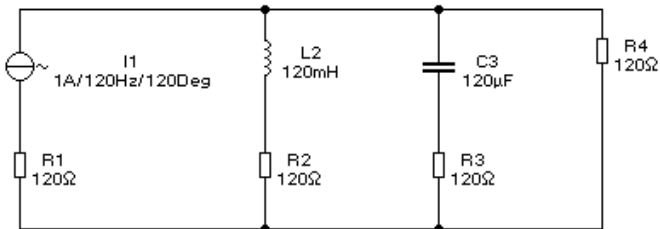
Вариант 21:



Вариант 22:



Вариант 23:



## 4 ЛАБОРАТОРНА РОБОТА №4

ТЕМА: Розрахунок перехідних процесів чисельним методом в лінійних електричних колах. Частина 1.

МЕТА РОБОТИ: Використання методу змінних станів, чисельних методів інтегрування диференціальних рівнянь і дискретних струмових моделей індуктивних і ємкісних елементів в неявному методі Ейлера для розрахунку перехідних процесів.

---

### Математична модель

Розглянемо електричну схему, яка показана на рисунку 4.1.

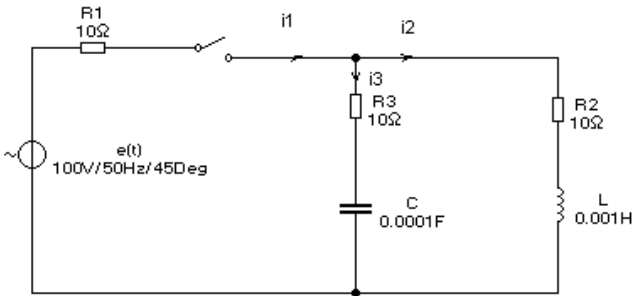


Рисунок 4.1 – Електричне коло з параметрами елементів, що завдані

У момент  $t=0$  замикається ключ і починається перехідний процес в наведеному електричному колі.

Замінімо індуктивний і ємкісний елементи їх дискретними струмовими моделями. Тоді коло набуде вигляд, показаний на рисунку 4.2. У цьому колі немає жодного ємкісного чи індуктивного елементу – це ланцюг постійного струму.

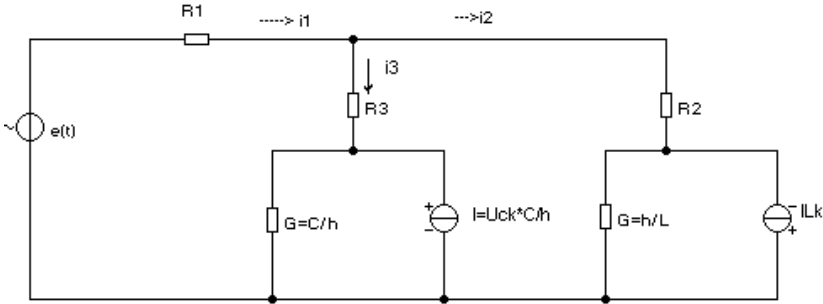


Рисунок 4.2 – Еквівалентна схема постійного струму для кожного кроку інтегрування

В початковий момент  $t=0$  відомі струм  $i_2^k$  і напруга  $u_c^k$  ( $k=1$ ). Це незалежні початкові умови. Для моменту  $t=h$  можна визначити всі струми та напруги, виконавши розрахунок кола постійного струму. Виконаємо цей розрахунок методом вузлових потенціалів, для цього перетворимо джерело ЕРС  $e(t)$  в джерело струму й отримаємо схему, яка наведена на рисунку 4.3.

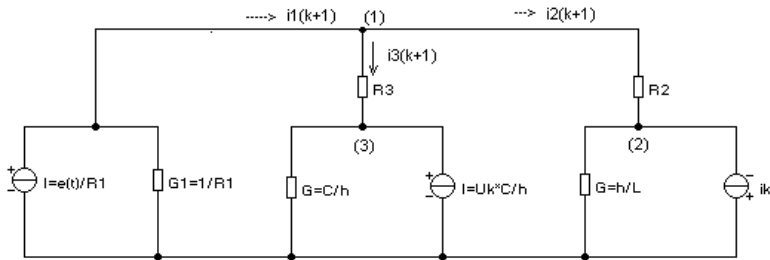


Рисунок 4.3 – Ланцюг з джерелом ЕРС  $e(t)$ , який перетворено на джерело струму.

Система потенціальних рівнянь має вигляд:

$$\begin{aligned}
 +G_{11}\varphi_1 - G_{12}\varphi_{21} - G_{13}\varphi_3 &= I_{11} \\
 -G_{21}\varphi_1 + G_{22}\varphi_{21} - G_{23}\varphi_3 &= I_{22} \\
 -G_{31}\varphi_1 - G_{32}\varphi_{21} + G_{33}\varphi_3 &= I_{33}
 \end{aligned}
 \tag{4.1}$$



де

$G_{11}$  - сума провідностей гілок, що сходяться до вузла 1.

$$G_{11} = \frac{1}{R_1} + \frac{1}{R_3} + \frac{1}{R_2};$$

$G_{22}$  - сума провідностей гілок, що сходяться до вузла 2.

$$G_{22} = \frac{1}{R_2} + \frac{h}{L};$$

$G_{33}$  - сума провідностей гілок, що сходяться до вузла 3.

$$G_{33} = \frac{1}{R_3} + \frac{C}{h};$$

$G_{12} = G_{21}$  - провідність гілки між вузлами 1 і 2.

$$G_{12} = \frac{1}{R_2};$$

$G_{13} = G_{31}$  - провідність гілки між вузлами 1 і 3.

$$G_{13} = \frac{1}{R_3};$$

$G_{23} = G_{32}$  - провідність гілки між вузлами 1 і 3.

$$G_{23} = 0;$$

$I_{11}, I_{22}, I_{33}$  - вузлові струми.

$I_{11}$  - алгебраїчна сума струмів джерел, що сходяться до вузла 1.

$$I_{11} = \frac{e(t)}{R_1};$$

$I_{22}$  - алгебраїчна сума струмів джерел, що сходяться до вузла 2.

$$I_{22} = -i_2^k;$$

$I_{33}$  - алгебраїчна сума струмів джерел, що сходяться до вузла 3.

$$I_{33} = U_C^k;$$

Таким чином, отримуємо систему лінійних рівнянь з постійними коефіцієнтами.

Рішення системи (4.1) дозволить отримати потенціали вузлів, тобто  $\varphi_1, \varphi_2, \varphi_3$ .

Якщо відомі потенціали вузлів визначимо всі струми й напруги на ємності

$$i_1 = \frac{e(t)}{R} - \varphi_1 \frac{1}{R};$$

$$i_2 = \frac{\varphi_1 - \varphi_2}{R_2};$$

$$i_3 = \frac{\varphi_1 - \varphi_3}{R_3};$$

$$u_C = \varphi_3;$$

Після визначення  $i_2$  і  $u_C$  уточнюємо параметри дискретних струмових моделей, час збільшуємо на величину кроку  $h$  і все обчислюється заново, поки кількість кроків інтеграції не досягне заданої величини.

#### **Блок-схема алгоритму розрахунку згідно з математичною моделлю**

Блок-схема алгоритму розрахунку приведена на малюнку 4.4.

#### **Завдання на виконання лабораторної роботи**

- розробіть програму на основі даного алгоритму;
- введіть до програми оператори, що дозволять виконати перевірку результатів обчислення;
- організуйте графічне виведення розрахованих функцій струмів і напруг;
- проведіть відладку програми;
- збережіть результати роботи (програма, лістинг обчислень, графіки) в особистому файлі.

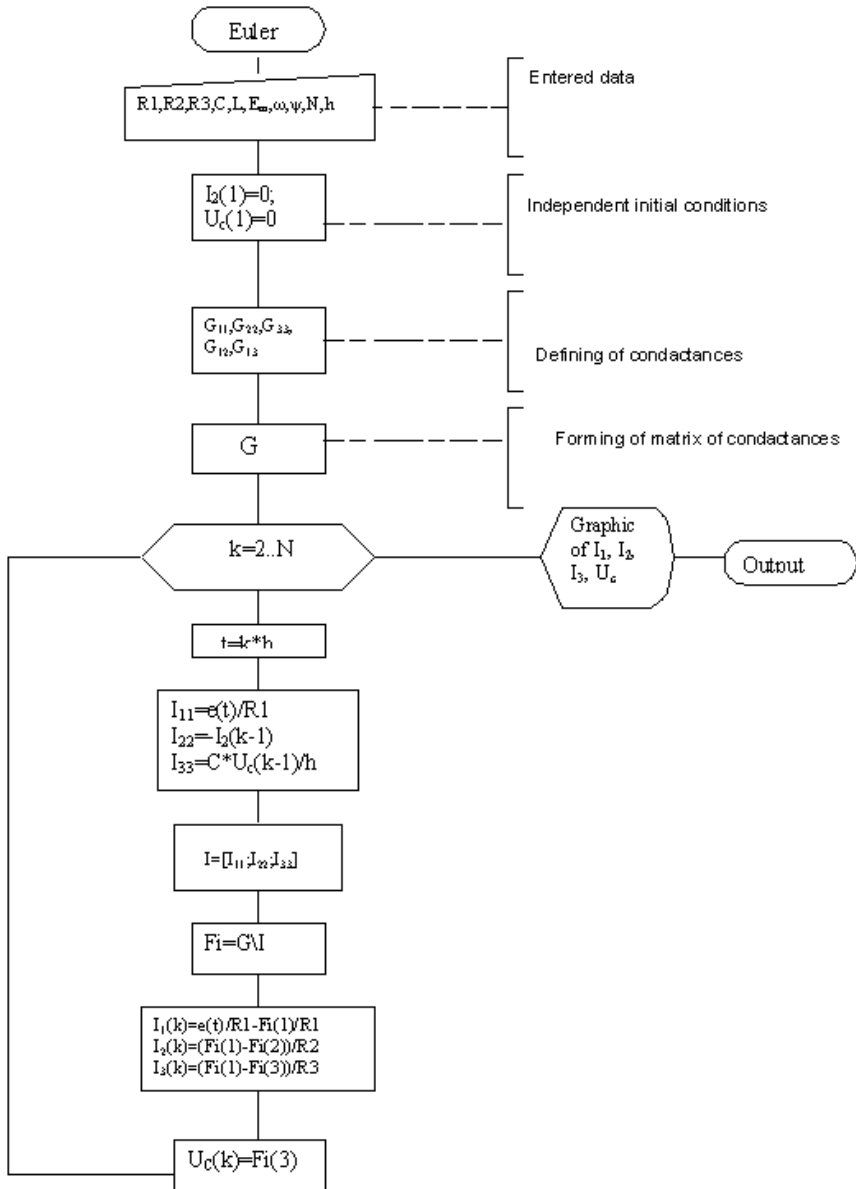


Рисунок 4.4 – Блок-схема алгоритму обчислень перехідного процесу

## 5 ЛАБОРАТОРНА РОБОТА №5

ТЕМА: Обчислення перехідних процесів в лінійних електричних колах чисельним методом. Частина 2.

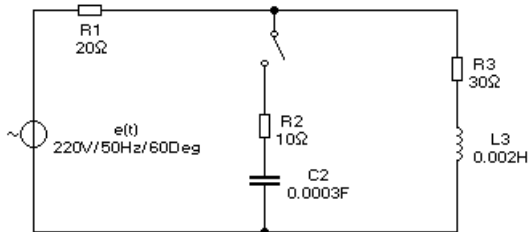
МЕТА РОБОТИ: Практичне вживання методу змінних станів, чисельних методів інтеграції диференціальних рівнянь і дискретних струмових моделей індуктивних і ємкісних елементів в неявному методі Ейлера для розрахунку перехідних процесів.

### Завдання на виконання роботи:

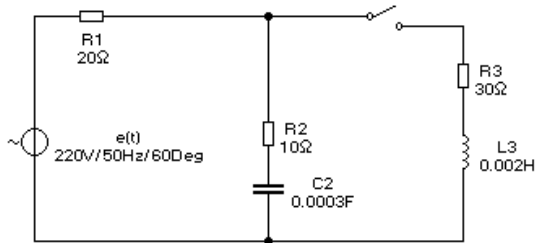
- створіть математичну модель, що реалізує розрахунок перехідних струмів в гілках схеми для вашого індивідуального завдання;
- розробіть алгоритм згідно створеної математичної моделі, що реалізує неявний метод Ейлера;
- розробіть програму, що реалізує розроблений алгоритм;
- введіть до програми оператори, що дозволяють виконати перевірку результатів обчислень;
- організуйте графічне виведення розрахованих функцій струмів і напруги;
- проведіть відладку програми;
- збережіть результати роботи (програма, лістинг обчислень, графіки) у вашому особистому файлі.

### Індивідуальні завдання

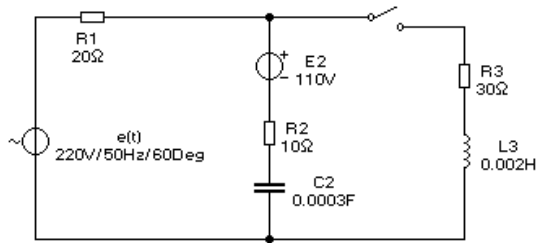
Варіант 1:



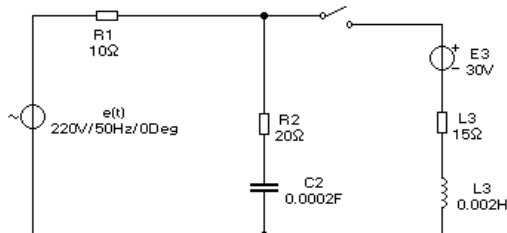
Вариант 2:



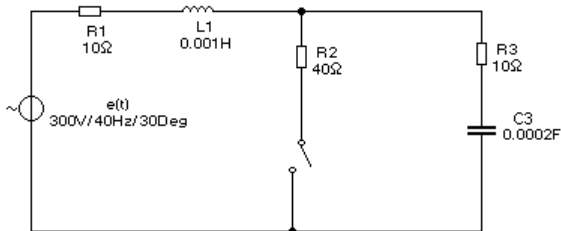
Вариант 3:



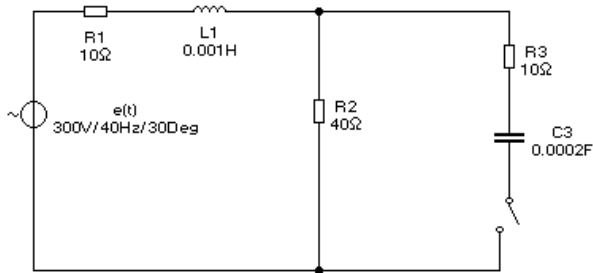
Вариант 4:



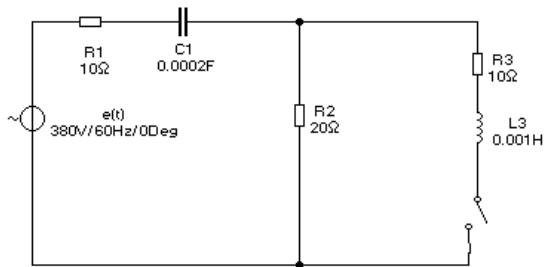
Вариант 5:



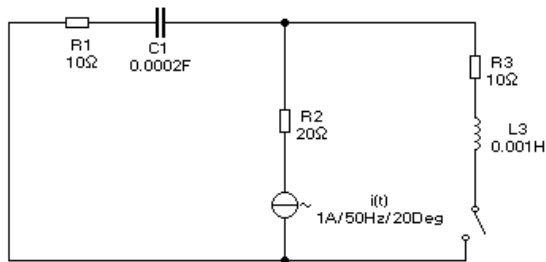
Вариант 6:



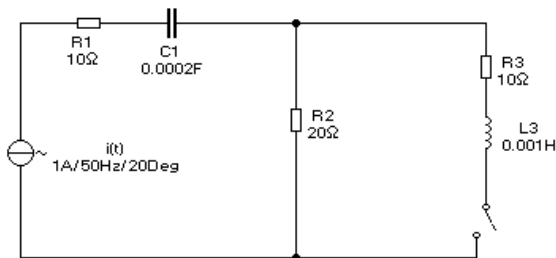
Вариант 7:



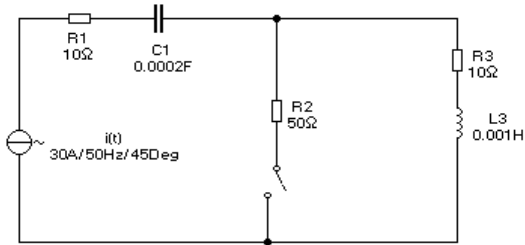
Вариант 8:



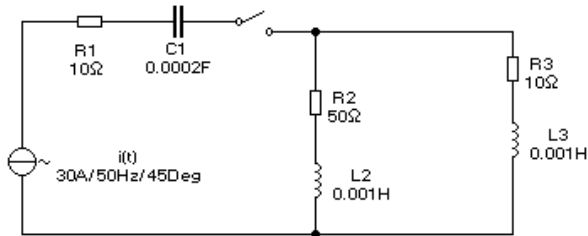
Вариант 9:



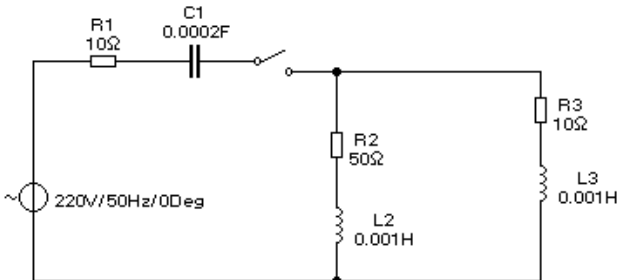
Вариант 10:



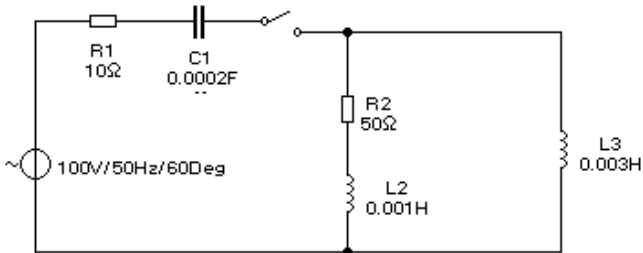
Вариант 11:



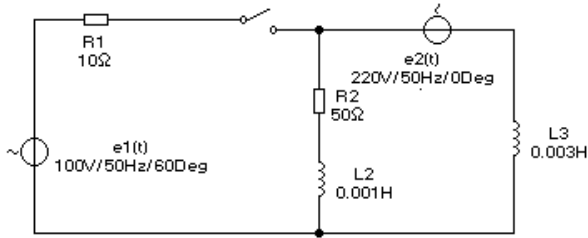
Вариант 12:



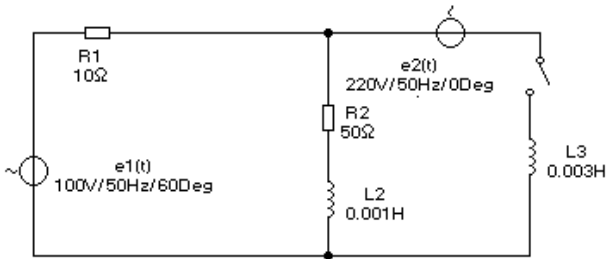
Вариант 13:



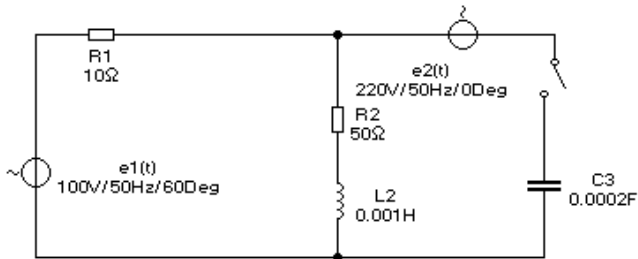
Вариант 14:



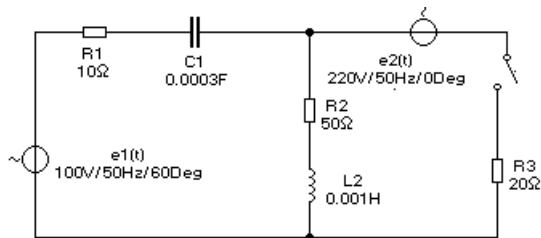
Вариант 15:



Вариант 16:

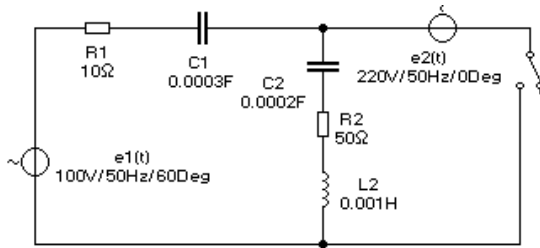


Вариант 17:

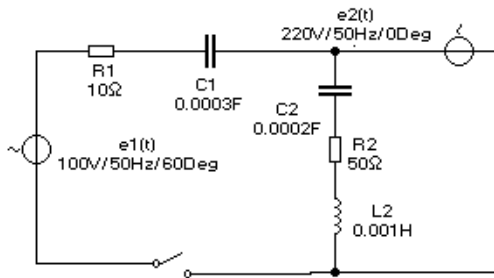




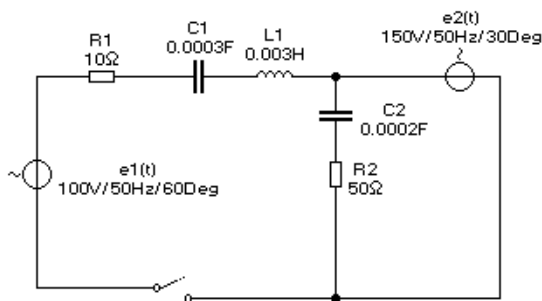
Вариант 18:



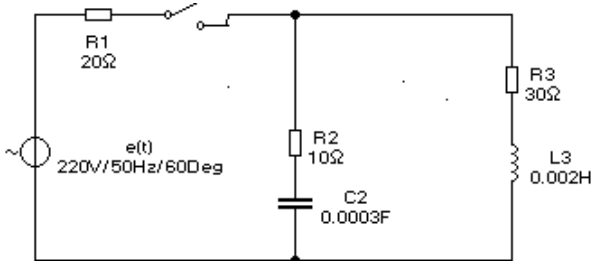
Вариант 19:



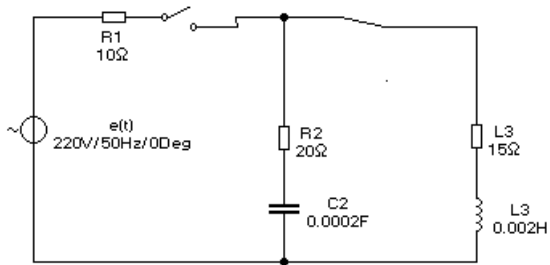
Вариант 20:



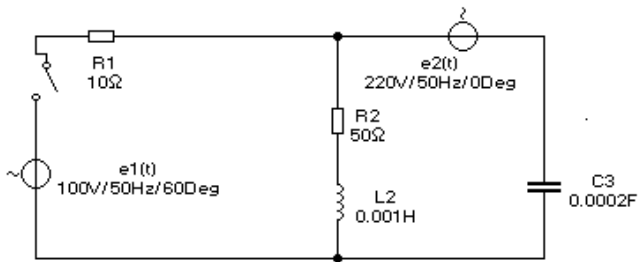
Вариант 21:



Вариант 22:



Вариант 23:



## 6 ЛАБОРАТОРНА РОБОТА №6

ТЕМА: Моделювання сталих електричних процесів в нелінійних електричних колах постійного струму у MATLAB. Частина 1.

МЕТА РОБОТИ: Постановка завдання, створення програми розрахунку електричного стану простого нелінійного ланцюга постійного струму методом Ньютона.

---

### Математична модель

Наведемо приклад складання математичної моделі електричних процесів і їх аналізу в простому електричному колі (рис.6.1) в MATLAB.

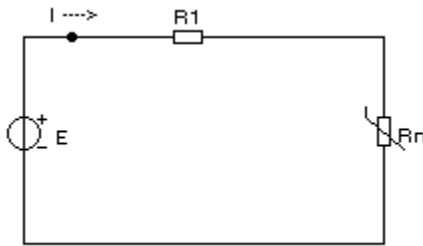


Рисунок 6.1 – Електрична схема, що моделюється.

Параметри елементів схеми:

$$E=10 \text{ В}; \quad R_1=0.5 \text{ Ом};$$

Нелінійний елемент заданий вольт-амперною характеристикою (ВАХ) в аналітичному вигляді:

$$U_n = a \cdot I^{1/3}, \quad (6.1)$$

де  $U_n$  – напруга на нелінійному елементі;  $a=4^{(1/3)}$ .

Потрібно визначити струм в схемі методом Ньютона.  
На підставі другого закону Кирхгофа можна скласти рівняння

$$R_1 \cdot I + U_n = E. \quad (6.2)$$

Підставимо вирази для ВАХ нелінійного опору (6.1) в рівняння (6.2) і отримаємо нелінійне алгебраїчне рівняння відносно струму I:

$$R_1 \cdot I + a \cdot I^{1/3} = E. \quad (6.3)$$

Рівняння (6.3) можна записати як  $f(I) = 0$ , де

$$f(I) = R_1 \cdot I + a \cdot I^{1/3} - E. \quad (6.4)$$

Відповідно до алгоритму Ньютона  $k+1$  наближення для струму можна знайти як

$$I_{k+1} = I_k - \frac{f(I_k)}{f'(I_k)} \quad (6.5)$$

Виходячи з (6.4), знайдемо вираз для  $f'(I)$ :

$$f'(I) = R_1 + a/(3 \cdot I^{2/3}).$$

#### **Визначення початкового наближення**

Прийmemo початкове наближення струму в ланцюзі  $I_0=0$ , тоді на першому кроці ітерації різниця струмів початкового і наступного наближень

$$I_1 - I_0 = f(I_0) / f'(I_0) = -10/\infty = 0,$$

т.ч., при заданому наближенні процес розходиться.  
Візьmemo за початкове наближення  $I_0=1$ , тоді:

$$I_1 - I_0 = f(I_0) / f'(I_0) = -7.91/1.03 = -7.68.$$

Отже,  $I_0=1$  можна прийняти як початкове наближення.

## Програма розрахунку в середі MATLAB

Головна програма(script-file)::

```
% Calculation of the simple circuit by the Newton method
% Newt1
% Initial data
E=10; R=0.5; a=4^(1/3);
% Number of iterations
N=20;
% Initial approach
k=1;
I(k)=1;
% Iterative process
for k=2:N
    fk_1=fun1(I(k-1),R,a,E); % call of f(i)
    dfdIk=dfdI(I(k-1),R,a); % call of f'(i)
    I(k)=I(k-1)-0.5*fk_1/dfdIk; % calculation of the next approach
end
% display of iterative process
plot(I);
```

Підпрограма –функція розрахунку (M-file) f(i):

```
% calculation of f(i)
function f=fun1(I,R,a,E)
f=R*I+a*I^0.333-E;
```

Підпрограма – функція розрахунку(M-file) f'(i):

```
% calculation of f'(i)
function der1=dfdI(I,R,a)
der1=R+a/(3*I^(2/3));
```

### Результати розрахунку

На рисунку 6.2 наведена графічна залежність розрахункового значення струму в схемі (рис. 6.1) на кожному кроці ітерації від номера цього кроку.

З рисунка видно, що практично достатньо 12-ти кроків ітерації для задовільної точності рішення.

### Індивідуальні завдання

Розрахувати струм в схемі (рис.6.1), якщо ВАХ нелінійного елемента задається аналітичною залежністю відповідно до таблиці 6.1.

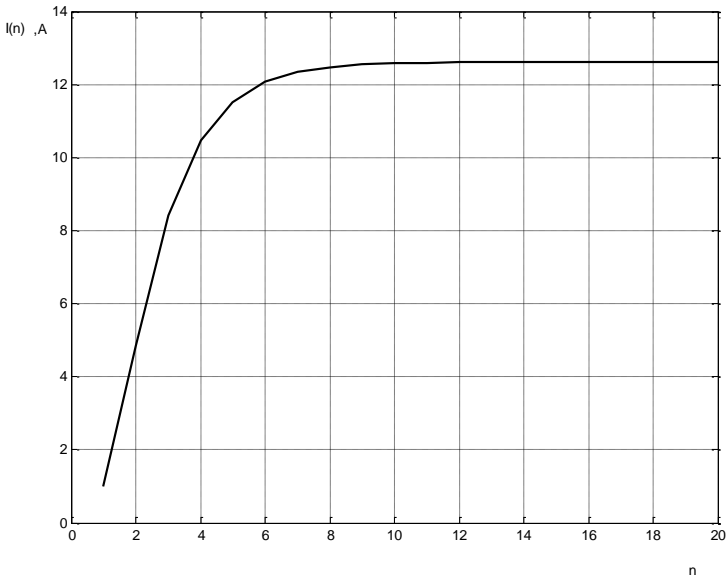


Рисунок 6.2 – Залежність I(n).

### Завдання на виконання роботи

- Вивчити пп.6.1 – 6.3.
- Повторити програму, наведену в п.6.4. і отримати залежність I(n) (рис.6.2);

Для заданого варіанту ВАХ нелінійного елемента  $R_n$ :

- Розробити математичну модель розрахунку струму в колі (рис.6.1);
- Скласти програму, що реалізовує розроблену модель;

- Організувати графічне виведення розрахункового значення струму в схемі (рис. 6.1) на кожному кроці ітерації від номера цього кроку  $I(n)$ ;
- Відладати програму;
- Результати роботи (програму, лістинг розрахунку, графіки) зберегти в особистій просторі на жорсткому диску.
- Оформити звіт по лабораторній роботі.

Таблиця 6.1

№ варіанта	ВАХ нелінійного елемента $R_n$	№ варіанта	ВАХ нелінійного елемента $R_n$
1	$I=2.5U^3$	16	$I=3U^2 \text{ sign}(U)$
2	$I=6U^2 \text{ sign}(U)$	17	$I=0.5U^4 \text{ sign}(U)$
3	$I=2\sqrt{ U } \text{ sign}(U)$	18	$I=5U \text{ arctg}(U)$
4	$I=3U^3$	19	$I=0.2 \text{ arctg}(U)$
5	$I=3[\log_a U-1]$	20	$I=8.6U^2 \text{ sign}(U)$
6	$I=3U^3$	21	$I=3.5[\exp(bU)-1]$
7	$I=\text{argtg}(U/5)$	22	$I=2 \text{ arctg}(U)$
8	$I=5U^2 \text{ sign}(U)$	23	$I=9.4U^2 \text{ sign}(U)$
9	$I=8 U ^{4/3} \text{ sign}(U)$	24	$I=1.7U^3$
10	$I=7U^3$	25	$I=8U^3$
11	$I=6.1U^2 \text{ sign}(U)$	26	$I=6.9[\exp(U)-1]$
12	$I=5.2U^3$	27	$I=6.9U^3$
13	$I=9.3[\ln U-1]$	28	$I=2[\exp(U/2)-1]$
14	$I=0.5U^4 \text{ sign}(U)$	29	$I=7[\log_a U-1]$
15	$I=8 U ^{3/2} \text{ sign}(U)$	30	$I=5.5[\ln U-1]$

## 7 ЛАБОРАТОРНА РОБОТА №7

ТЕМА: Моделювання сталих процесів в нелінійних електричних колах постійного струму чисельними методами в середі MATLAB. Частина 2.

МЕТА РОБОТИ: Постановка завдання, розробка програми розрахунку нелінійних електричних кіл постійного струму методом Ньютона, при використанні дискретних струмових моделей.

---

### Математична модель

Розглянемо приклад складання математичної моделі електричних процесів в колі, наведеному на рисунку 7.1, і проаналізуємо їх в середі MATLAB.

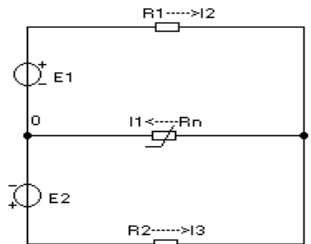


Рисунок 7.1 – Задане електричне коло

Значення параметрів елементів:

ЕРС:

$$E_1 = 10 \text{ V}; \quad E_2 = 5 \text{ V};$$

Лінійні опори :

$$R_1 = 2 \text{ Ом}; \quad R_2 = 3 \text{ Ом};$$

Нелінійний опір  $R_n$  задається аналітичним вираженням ВАХ:

$$U_f(I) = a \cdot I^{1/3}, \quad (7.1)$$

де  $U_f(I)$  – напруга на нелінійному резисторі,



$$a=10^{1/3}.$$

Потрібно визначити струми у всіх гілках даного ланцюга.

Для вирішення потрібно обчислювати похідну  $\frac{dI}{dU}$  при конкретних значеннях напруги. Для виконання цієї мети вирішимо рівняння (7.1) щодо струму  $I$ , і візьмемо похідну:

$$I_f(U) = 0.3U^2 \quad (7.2)$$

Вираз (7.2) дозволяє обчислити провідність дискретної струмової моделі нелінійного резистивного елемента. Тому в програмі визначимо вираз (7.2) функцією користувача. Початкове значення (нульове наближення) струму в нелінійному опорі має бути задане. Задано його рівним  $I_0 = 0,1$  А (середньо-стельова величина). Нульове наближення повинне відповідати двом вимогам: похідна  $dI/dU$  при цьому значенні струму не має бути рівною нулю; значення  $I_0$  не повинно перевищувати максимально можливого значення струму в нелінійному елементі. Попередній розрахунок  $I_0$  можна виконати таким чином: грубо замінити ВАХ прямою лінією і виконати розрахунок струмів в отриманому лінійному ланцюзі. Проте в більшості випадків виконувати попередній розрахунок  $I_0$  не потрібно. Простіше визначити  $I_0$  методом пристрілки. Якщо при якомусь значенні  $I_0$  розрахунок упевнено сходиться, то вибране значення  $I_0$  – те, що треба.

Обчислені струми у всіх гілках всіх кроків ітерації зберігаємо в двовимірному масиві  $I$ . Перший індекс задає номер струму гілки, другої, – номер ітерації.

Далі йде цикл обчислень, в якому номер ітерації до змінюється від одиниці до заданого номера  $N$ . У циклі обчислюються параметри дискретної струмової моделі  $G$ ,  $J$ , а нелінійний ланцюг (рис.7.1) замінюється еквівалентною схемою заміщення (рис.7.2).

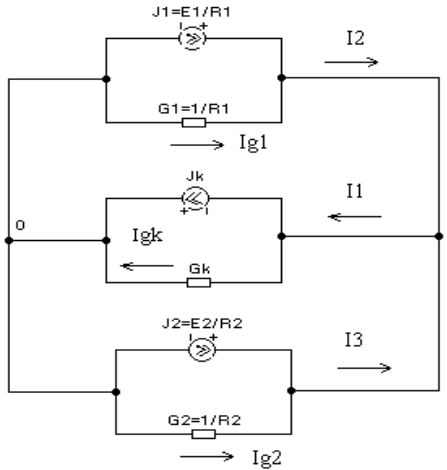


Рисунок 7.2 – Еквівалентна схема

Розрахунок струмів в лінійному колі, показаному на рис.3.9, проведемо методом двох вузлів:

$$\varphi_1 = \frac{\frac{E_1}{R_1} + \frac{E_2}{R_2} - J}{\frac{1}{R_1} + \frac{1}{R_2} + G},$$

де  $J$  та  $G$  - параметри дискретної струмової моделі. Тоді струми гілок обчислюються по формулах

$$I_{1,k} = \frac{E_1 - \varphi_1}{R_1};$$

$$I_{2,k} = \frac{E_2 - \varphi_1}{R_2};$$

$$I_{0,k} = J - \varphi_1 \cdot G.$$

## Програма розрахунку в середі MATLAB

### Головна програма (script-file):

```

% Calculation of the direct current circuit by the Newton method
% with application of discrete current models of nonlinear resistance
% Newt2
% Initial data

global a
R1=2; R2=3; E1=10; E2=5; I0=0.8; N=10;
a=10^(1/3);

% Initial approximation
I(1,1)=I0;

% Iterative process
for k=2:N
    U=Uf(I(1,k-1)); % call function, calculating Uf(I) accord to previous
% iterative step K-1
    G=dIf(U); % call function, calculating dIf(U)(conductance)
% accord to previous iterative step K-1
    J=I(1,k-1)-U*G; % equivalent source current according to previous
% iterative step K-1
    phi1=(E1/R1+E2/R2-J)/(1/R1+1/R2+G); % potential of the node 1
    I(1,k)=J+phi1*G; % current through the nonlinear resistor Rn
    I(2,k)=(E1-phi1)/R1; % current through the resistor R1
    I(3,k)=(E2-phi1)/R2; % current through the resistor R2
End
% Display of iterative process
p=1:10;
plot (p,I(1,p),p,I(2,p),p,I(3,p));

```

### Підпрограма-функція (M-file) розрахунку $U_f(I)$ :

```

% Calculation the voltage across Rn as current function
function f=Uf(I)
global a
f=a*I^(1/3);

```

Підпрограма-функція (M-file) розрахунку  $dI/dU$ :

% Calculation the derivative I with respect to U

function dI=dIf(U)

dI=0.3\*U^2;

**Результати обчислень**

Графічні залежності розрахункових струмів в гілках даного електричного кола (мал. 7.1.) на кожному кроці ітерації наведені на рисунку 7.3.

Графічні залежності струмів гілок від номера ітерації показує упевнену збіжність методу.

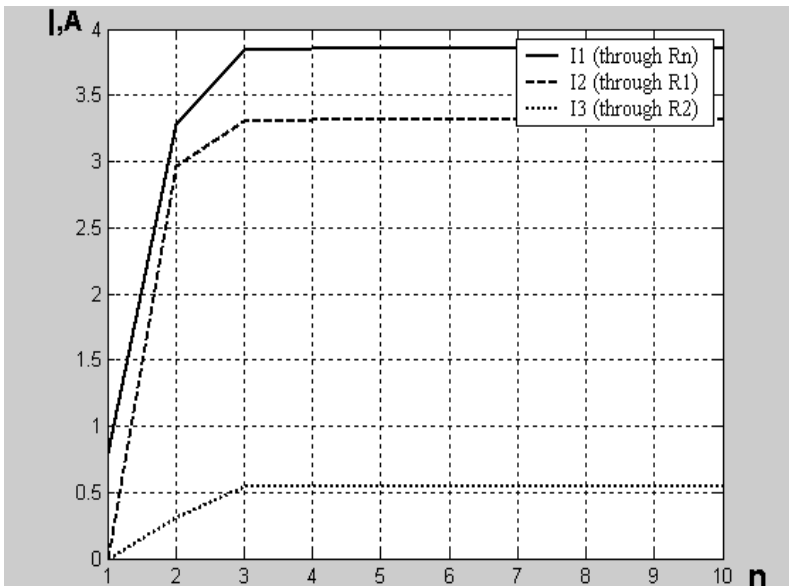


Рисунок 7.3 – Графічні залежності розрахункових струмів гілок від номера кроку ітерації

### Індивідуальні завдання

Обчисліть струми в гілках даного електричного кола (рис. 7.1), якщо ВАХ нелінійного елемента задається таблицею 7.1.

#### Завдання на виконання роботи

- Вивчити п.7.1.
- Скласти програму, наведену в пункті 7.2 і отримати залежність як на рис.7.3.

Для заданої в таблиці 7.1 ВАХ і значень параметрів  $R_1$  і  $R_2$  виконати:

- Розробити математичну модель розрахунку струму в електричному колі (рис.7.1);
- Скласти програму, що реалізовує розроблену модель;
- Організувати графічне виведення розрахункового значення струмів в схемі (рис. 7.1) на кожному кроці ітерації від номера цього кроку  $I_1(n)$ ;  $I_2(n)$ ;  $I_3(n)$ ;
- Відладити програму;
- Результати роботи (програму, лістинг розрахунку, графіки) зберегти в особистому просторі жорсткого диску.
- Оформити звіт по лабораторній роботі.

Таблиця 7.1

Номер варіанта	ВАХ нелінійного елемента $R_n$	$R_{1,}$ $Ом$	$R_{2,}$ $Ом$	$E_{1,}$ $В$	$E_{2,}$ $В$
1	$I=5.5[\ln U-1]$	5	5	40	7
2	$I= 7[\log_a U-1]$	6	8	15	20
3	$I=2[\exp(U/2)-1]$	7	7.1	5	13
4	$I=6.9U^3$	3	8	8	10
5	$I=6.9[\exp(U)-1]$	2	8.5	13	14
6	$I=8U^3$	8	4.1	5.5	6.6
7	$I=0.5U^4 \text{ sign}(U)$	7	8	8.9	8.5
8	$I=5U^2 \text{ sign}(U)$	5	3	20	3.8
9	$I=9.3[\ln U-1]$	9	9.9	13	8.9
10	$I=7U^3$	2	10	6.9	8.3
11	$I=0.5U^4 \text{ sign}(U)$	5	7.4	3.8	19
12	$I=3 U ^{4/3} \text{ sign}(U)$	1	1.9	7	8.6
13	$I=5.2U^3$	6	25	9	6.8
14	$I=6.1U^2 \text{ sign}(U)$	7.1	8.8	8.7	4.9
15	$I=8 U ^{4/3} \text{ sign}(U)$	9.7	8.3	4.8	3.9
16	$I=3U^2 \text{ sign}(U)$	5.9	7.1	20	25
17	$I=5U \text{ arctg}(U)$	3.7	8.2	30	40
18	$I=\text{arctg}(U/5)$	7	9.5	8	71
19	$I=0.2 \text{ arctg}(U)$	20	1.8	5.9	9
20	$I=8.6U^2 \text{ sign}(U)$	5.4	7.4	50	28
21	$I=3.5[\exp(bU)-1]$	4.0	8.5	41	52
22	$I=2 \text{ arctg}(U)$	22	4.8	17	7
23	$I= 9.4U^2 \text{ sign}(U)$	13	5.9	5.9	12
24	$I=1.7bU^3$	3.5	6	8.9	35
25	$I=3U^3$	2.1	8	26	23
26	$I=3[\log_a U-1]$	4.8	4	13	9
27	$I=3U^3$	6.1	9	4	10
28	$I=2 \sqrt{ U } \text{ sign}(U)$	8.2	5	46	13
29	$I=6U^2 \text{ sign}(U)$	5	3.9	11	12
30	$I=2.5U^3$	9	8.2	45	50

## 8 ЛАБОРАТОРНА РОБОТА №8

ТЕМА: Моделювання сталих електричних процесів в нелінійних електричних колах постійного струму в середі MATLAB. Частина 3.

МЕТА РОБОТИ: Постановка завдання, створення програми розрахунку електричного стану нелінійного електричного кола постійного струму методом Ньютона із застосуванням сплайнової інтерполяції ВАХ.

---

### Математична модель

Складемо математичну модель електричних процесів в електричному колі (рис.8.1) в середі MATLAB, якщо ВАХ нелінійних елементів задана в графічному вигляді.

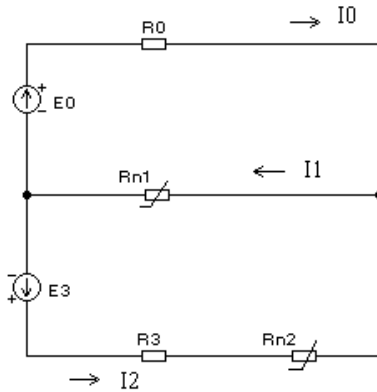
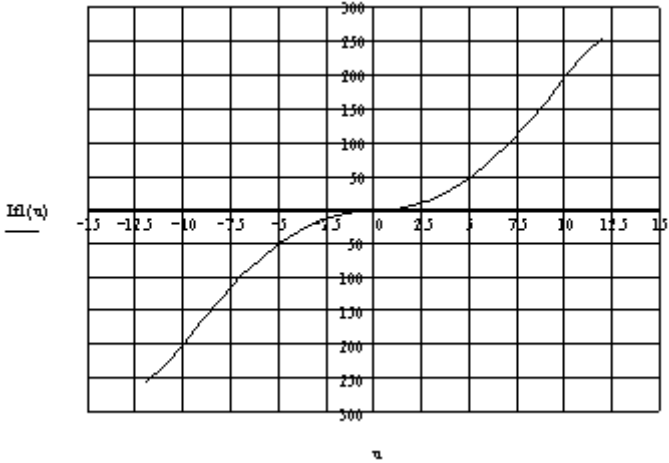


Рисунок 8.1 – Моделюєма електрична схема.

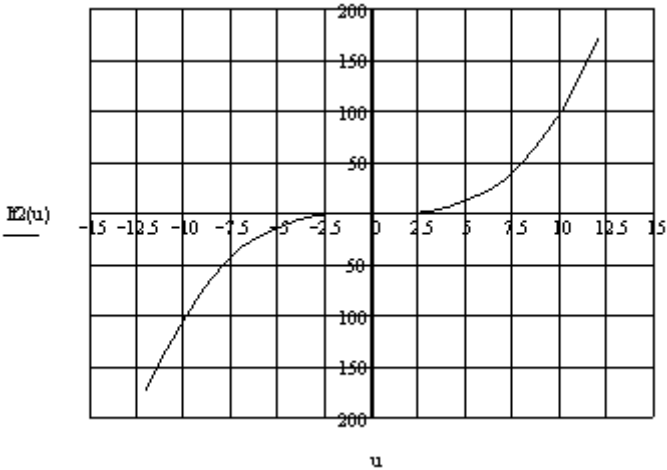
Параметри елементів схеми:

$$E_0=5 \text{ В}; \quad E_3=4 \text{ В}; \quad R_0=2 \text{ Ом}; \quad R_3=5 \text{ Ом} .$$

Нелінійні елементи  $R_{n1}$  й  $R_{n2}$  задані вольт-амперною характеристикою (ВАХ) у графічному вигляді у відповідності з рис. 8.2:



а)



б)

а) – для  $R_{n1}$ ;б) – для  $R_{n2}$ .

Рисунок 8.2 – ВАХ нелінійних опорів.



Схема заміщення, в якій нелінійні опори замінені дискретними струмовими моделями, наведена на рис.8.3.

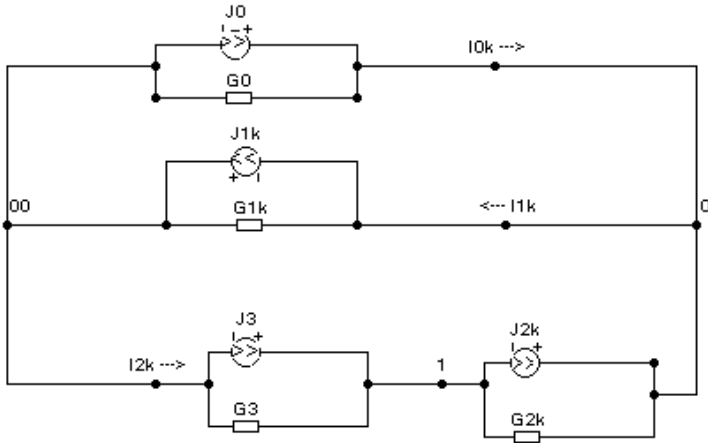


Рисунок 8.3 – Схема заміщення.

Алгоритм вирішення цієї задачі в цілому збігається з алгоритмом рішення задачі, що розглядується в лабораторній роботі №7. Особливістю є графічний спосіб завдання ВАХ нелінійних елементів.

У математичній моделі визначимо ВАХ вузлами інтерполяції так, щоб абсциси вузлів інтерполяції ВАХ для  $Rn1$  і  $Rn2$  збігалися, і їх, таким чином, можна було б задати одним вектором.

### Програма розрахунку в середі MATLAB

Головна програма:

```
% Newt3
```

```
%-----
```

```
% Calculation of the direct current branched nonlinear circuit by the  
Newton method
```

```
% with application of discrete current models of nonlinear resistance
```

*% VACH of nonlinear resistances given as arrays*

*% VACH is interpolated by spline-method*

*% derivatives are calculated by numerical seven-point method*

*%-----*

*% Initial data:*

*%-----*

*%MU - vector of voltage coordinates of reference points for*

*% the VACH of Rn1 and Rn2;*

*%MII - vector of current coordinates of reference points for*

*% the VACH of Rn1;*

*%MI2 - vector of current coordinates of reference points for*

*% the VACH of Rn2;*

MU=[-12, -10, -8, -6, -4, -3, -2, -1, -0.5, -0.2, -0.1, 0,...

0.1, 0.2,0.5, 1, 2, 3, 4, 6, 8, 10, 12];

MI1=[-256, -200, -128, -72, -32, -18, -8, -2, -0.5, -0.08, -0.02, 0,...

0.02, 0.08, 0.5, 2, 8, 18, 32, 72, 128, 200, 256];

MI2=[-172.8,-100,-51.2,-21.6,-6.4,-2.7,-0.8,-0.1,-0.013,-0.0008,-

0.0001,...

0, 0.0001,0.0008,0.013,0.1,0.8,2.7,6.4,21.6,51.2,100,172.8];

R0=2; R3=5; E0=5; E3=4; hU=0.1; hI=0.1;

*%-----*

*% Parameters of equivalent current sources*

*%-----*

G0=1/R0; G3=1/R3;

J0=E0\*G0;

J3=E3\*G3;

*%-----*

*% Initial approximation*

*%-----*

I\_11=2.5;

I\_21=1;

*%-----*

*% Number of approximation steps*

*%-----*

N=40;

*%-----*

*% Iterative process*

```

%-----
I(1,1)=I_11;
I(2,1)=I_21;
for k=2:N
U1k=spline(MI1,MU,I(1,k-1)); % calculation of the voltage drop U1k
across Rn1
                                % according to VACH by the spline-method

G1k=der7(MU,MI1,U1k,hU);      % calculation of the derivative of
VACH-function I_1(U)

U2k=spline(MI2,MU,I(2,k-1));   % calculation of the voltage drop
U2k across Rn2
                                % according to VACH by the spline-method

G2k=der7(MU,MI2,U2k,hU);      % calculation of the derivative of
VACH-function I_2(U)

J1k=I(1,k-1)-U1k*G1k;         % current through the nonlinear resistor
Rn1

J2k=I(2,k-1)-U2k*G2k;         % current through the nonlinear resistor
Rn2

% account of conductances
G11=G0+G1k+G2k;
G22=G2k+G3;
G12=G2k;

% node currents
I11=J0-J1k+J2k;
I22=J3-J2k;

% matrix of conductances
M=[ G11, -G12;
   + -G12, G22];
% matrix of node currents

```

```

Iuz=[I1 1; I22];

% potential of node 1
phi=M\Iuz;

% branch currents
I(3,k)=J0-phi(1)*G0;
I(1,k)=J1k+phi(1)*G1k;
I(2,k)=J2k+(phi(2)-phi(1))*G2k;
end
%-----
% display of iterative process
%-----
p=1:N;
plot(p,I(1,p),p,I(2,p),p,I(3,p));

```

Підпрограма –функція розрахунку похідної ВАХ в заданій точці:

```

% der7
% Calculation of derivative of function given by table using
% seven-point method
% MX, MY - vectors of values of argument & function
% x - value of argument at which function is calculated
% h- increment of argument

function df7=der7(MX,MY,x,h)
yk_3=spline(MX,MY,x-3*h);
yk_2=spline(MX,MY,x-2*h);
yk_1=spline(MX,MY,x-h);
yk1=spline(MX,MY,x+h);
yk2=spline(MX,MY,x+2*h);
yk3=spline(MX,MY,x+3*h);
df7=(-yk_3+9*yk_2-45*yk_1+45*yk1-9*yk2+yk3)/(60*h);

```

### Результати розрахунку

На рисунку 8.4. наведена графічна залежність розрахункових значень струмів в схемі (рис. 8.1) на кожному кроці ітерації від номера цього кроку.

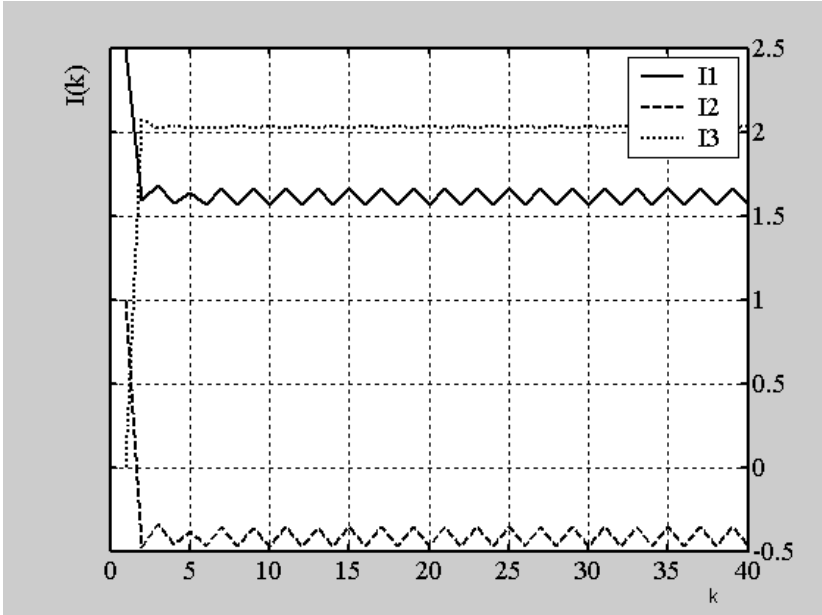


Рисунок 8.4 – Розрахункові залежності струмів в гілках.

Як видно з графіка, при ітераціях наближені рішення здійснюють циклічні коливання довкола шуканого рішення. Очевидно, що процес розходиться.

### Поліпшення збіжності методу Ньютона

Для поліпшення збіжності введемо ваговий коефіцієнт в значення  $K+1$ , що розраховується, наближення струмів в гілках. Внесемо наступні зміни до головної програми:

*% branch currents*

$$I(3,k)=(J0-\text{phi}(1)*G0+I(3,k-1))*0.5;$$

$$I(1,k)=(J1k+\text{phi}(1)*G1k+I(1,k-1))*0.5;$$

$$I(2,k)=(J2k+(\text{phi}(2)-\text{phi}(1))*G2k+I(2,k-1))*0.5;$$

Отримайте розрахункові залежності струмів в гілках. Якщо збіжність незадовільна, необхідно зменшити вагове значення нового наближення таким чином:

*% branch currents*

$$I(3,k)=(J0-\text{phi}(1)*G0+3*I(3,k-1))*0.25;$$

$$I(1,k)=(J1k+\text{phi}(1)*G1k+3*I(1,k-1))*0.25;$$

$$I(2,k)=(J2k+(\text{phi}(2)-\text{phi}(1))*G2k+3*I(2,k-1))*0.25;$$

Отримайте розрахункові залежності струмів в гілках на кожному кроці ітерації від номера цього кроку. Переконайтеся, що процес сходиться.

### **Завдання на виконання роботи:**

- вивчити п.8.1.
- повторити програму, приведену в п.8.2. і отримати залежності  $I(n)$  (рис.8.4);
- відповідно до п.8.3. добитися збіжності процесу обчислень та отримати відповідні залежності збіжності процесу обчислень струмів в гілках;
- результати роботи (програму, лістинг розрахунку, графіки) зберегти в особистому просторі жорсткого диску.
- оформити звіт по лабораторній роботі. У звіті відобразити математичну модель досліджуваної схеми, тексти програм, результати розрахунку, висновки.

## 9 ЛАБОРАТОРНА РОБОТА №9

ТЕМА: Моделювання електромагнітних процесів в магнітних колах постійного струму в середі MATLAB. Частина 1.

МЕТА РОБОТИ: Постановка завдання, створення програми розрахунку нелінійного магнітного ланцюга постійного струму методом Ньютона із застосуванням дискретних моделей магнітних опорів і їх інтерполяції сплайнами.

---

### Математична модель

Складемо математичну модель магнітного ланцюга (рис.9.1) в середі MATLAB, якщо всі нелінійні елементи задані дискретними моделями, а їх криві намагнічення у вигляді опорних точок.

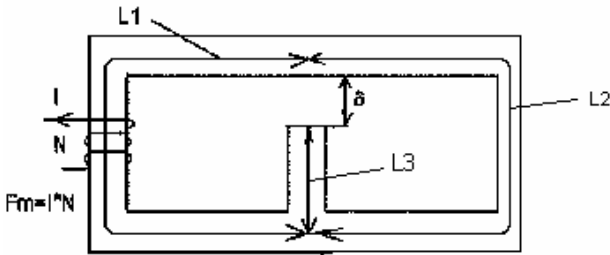


Рисунок 9.1 – Моделюєме магнітне коло.

Параметри магнітного кола:

Довжина середніх магнітних ліній відповідно до рис.9.1:

$l_1=0.3$  м;  $l_2=0.3$  м;  $l_3=0.1$  м;

Ширина магнітних гілок:

$D_1=0.1$  м;  $D_2=0.1$  м;  $D_3=0.1$  м;

Товщина сердечника:

$d_{st}=0.1$  м;

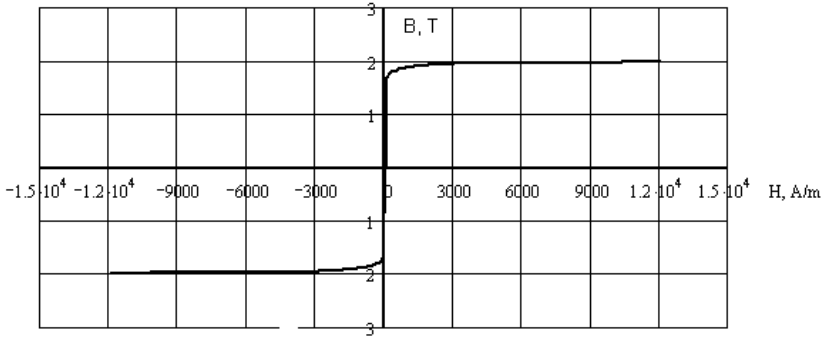
Величина повітряного зазору:

$l_{uft}=0.00002$  м;

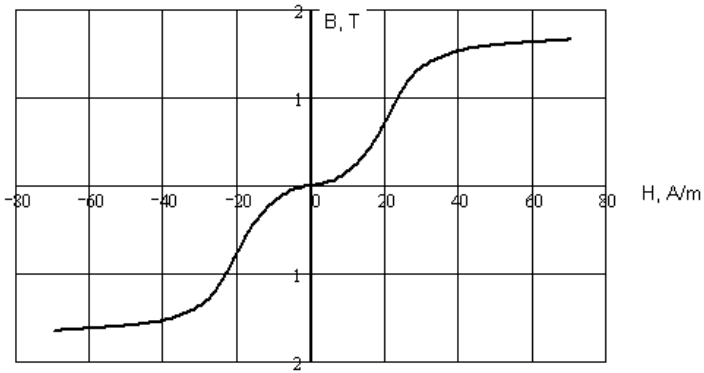
Струм котушки і кількість витків:

$I=1$  A;  $N=20$ .

Крива намагнічення сталі сердечника  $B(H)$  показана на рисунку 9.2.



а).



б)

а) – загальний вигляд

б) – початкова ділянка кривої намагнічення.

Рисунок 9.2 – Крива намагнічення сталі сердечника  $B(H)$ ;



Складемо схему заміщення заданого магнітного ланцюга як на рис. 9.3

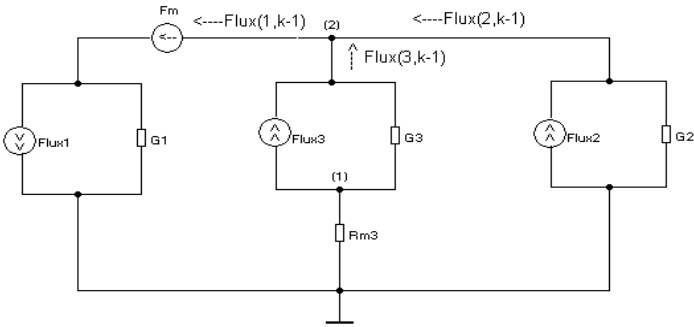


Рисунок 9.3 – Схема заміщення.

Відповідно до другого закону Кірхгофа джерело МДС  $F_m$  можна внести до гілки, в якій знаходиться провідність  $G_i$ , а потім перетворити його в джерело магнітного потоку  $F_m G_i$ . Схема заміщення, яка відповідає цим перетворенням, наведена на рис. 9.4

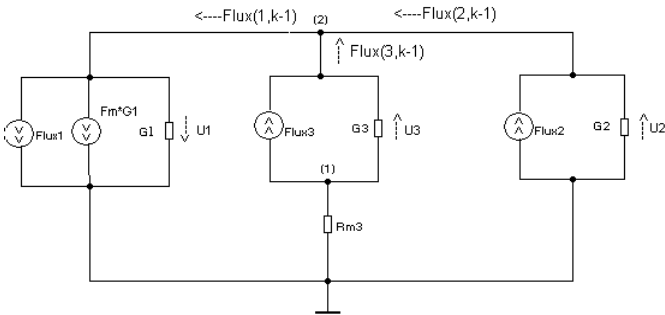


Рисунок 9.4 – Схема заміщення з джерелом МДС, який перетворено в джерело магнітного потоку.

### Програма розрахунку в середі MATLAB

Головна програма:

```
%      Newtб
%-----
% Розрахунок нелінійного магнітного кола методом Ньютона
%-----
n=20;                % кількість ітерацій методу Ньютона
h=0.001;  % крок для виконання чисельного диференціювання
I=5; N=50;          % струм котушки і кількість витків
Fm=N*I;            % MPC
mu0=4*pi*1e-7;
l1=0.3; l2=0.3; l3=0.1;          % довжини магнітних гілок
D1=0.1; D2=0.1; D3=0.1;        % ширина магнітних гілок
dlt=0.1 ;                    % товщина сердечника
luft=0.00002;                % товщина повітряного зазору
S1=D1*dlt; S2=D2*dlt; S3=D3*dlt; % площини перетинів магнітних
                                % гілок
Rm3=luft/(mu0*S3);%магнітний опір повітряного зазору
Gm3=1/Rm3;
% крива намагнічення стали
V=[-2,-1.98,-1.96,-1.94,-1.92,-1.9,-1.86,-1.84,-1.8,-1.7,-1.6,-1.4,...
-1.2,-1,-0.8,-0.4,-0.2,-0.1,0,0.1,0.2,0.4,0.8,1,1.2,1.4,1.6,1.7,1.8, 1.84,...
1.86, 1.9, 1.92, 1.94, 1.96, 1.98, 2];
H=[-11820,-5331,-3191,-2249,-1625,-1218,-660,-520,-295,-97,...
-51.4,-31.9,-26.2,-23.2,...
-20.8,15.2,10.8,7.58,10.8,15.2,20.8,23.2,26.2,31.9,...
51.4, 97, 295, 520, 660, 1218, 1625, 2249, 3197, 5331, 11820 ];
Flux(1,1)=0.02;    % початкове наближення магнітних потоків
Flux(2,1)=0.015;
Flux(3,1)=0.01;
%-----
% ітераційний розрахунок магнітних потоків в лінійному ланцюзі,
% що містить дискретні струмові моделі, методом вузлових
% потенціалів
%-----
for k=2:n
% розрахунок провідності дискретних струмових моделей для K-1
% ітерації
```

```

U3=I3*spline(B,H,Flux(3,k-1)/S3);
G3=S3/I3*der7(H,B,U3/I3,h);
U1=I1*spline(B,H,Flux(1,k-1)/S1);
G1=S1/I1*der7(H,B,U1/I1,h);
U2=I2*spline(B,H,Flux(2,k-1)/S2);
G2=S2/I2*der7(H,B,U2/I2,h);
% струми дискретних джерел для K-1 ітерації
Flux1=Flux(1,k-1)-U1*G1;
Flux2=Flux(2,k-1)-U2*G2;
Flux3=Flux(3,k-1)-U3*G3;
% вузлова провідність для K-1 ітерації
G22=G3+G1+G2;
G11=G3+Gm3;
G12=G3;
G=[ G11, -G12;
    -G12, G22];
% вузлові потоки для K-1 ітерації
Fluxnode=[-Flux3; Flux3+Flux2-Flux1-Fm*G1];
% расчет узловых потенциалов для K итерации
phi=G\Fluxnode;
% магнітні потоки для k-ой ітерації
Flux(3,k)=(-phi(1)*Gm3+Flux(3,k-1)*2)/3;
Flux(1,k)=(Flux1+Fm*G1+phi(2)*G1+Flux(1,k-1)*2)/3;
Flux(2,k)=(Flux2-phi(2)*G2+Flux(2,k-1)*2)/3;
End
%-----
% побудова графіка збіжності ітераційних процесів
% розрахунку магнітних потоків в стрижнях магнітопровода
%-----
p=1:n;
plot(p,Flux(1,p)/S1,p,Flux(2,p)/S2,p,Flux(3,p)/S3);

```

Підпрограма–функція `der7` розрахунку похідної в заданій точці наведена в методичних вказівках до лабораторної роботи № 8.

### Результати розрахунку

На рисунку 9.5 приведена графічна залежність розрахункових значень магнітної індукції в гілках магнітного кола (рис. 9.1) на кожному кроці ітерації від номера цього кроку.

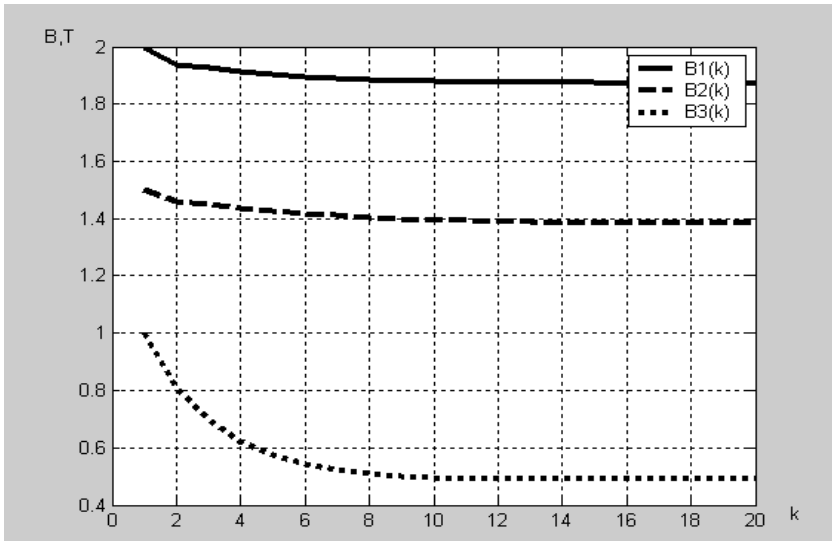


Рисунок 9.5 – Розрахункові залежності магнітних потоків в стрижнях сердечника від кроку ітерації.

### Завдання на виконання роботи:

- вивчити п.9.1;
- скласти програму, наведену в п.9.1. і отримати залежності магнітної індукції в стрижнях від номера ітерації (рис.9.5);
- вивести чисельні значення розрахованих величин індукції;
- довести правильність отриманого рішення;
- результати роботи (програму, лістинг розрахунку, графіки) зберегти в особистому просторі на жорсткому диску.

## 10 ЛАБОРАТОРНА РОБОТА №10

ТЕМА: Моделювання електромагнітних процесів в магнітних колах постійного струму в середі MATLAB. Частина 2.

МЕТА РОБОТИ: Постановка завдання, створення програми розрахунку нелінійного магнітного кола постійного струму методом Ньютона із застосуванням дискретних моделей магнітних опорів і їх інтерполяції сплайнами.

---

### Завдання на виконання роботи

Для заданого варіанту магнітного кола:

- розробити математичну модель розрахунку магнітних потоків в стрижнях;
- скласти програму, що реалізує розроблену модель;
- організувати графічне виведення розрахункових значень магнітних потоків в магнітному колі (рис. 10.1) на кожному кроці ітерації від номера цього кроку;
- відладити програму;
- результати роботи (програму, лістинг розрахунку, графіки) зберегти в особистому просторі на жорсткому диску;
- оформити звіт по лабораторній роботі. У звіті відобразити математичну модель досліджуваної схеми, тексти програм, результати розрахунку, висновки.

## Індивідуальні завдання

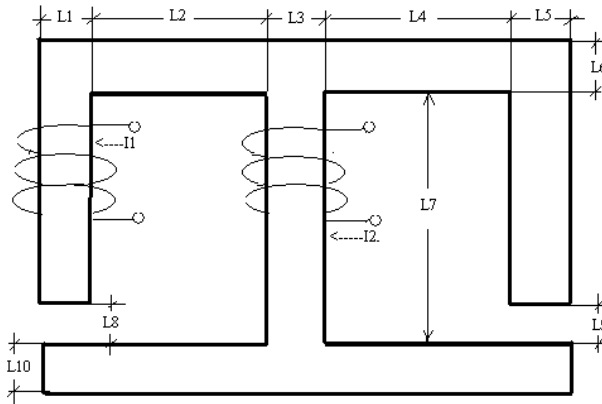


Рисунок 10.1 – Магнітне коло, що моделюється.

Параметри магнітної системи відповідно до заданого варіанту наведені в таблиці 10.1. Крива намагнічення сталі приведена в лабораторній роботі № 9.

Таблиця 10.1.

№	L1, ММ	L2, ММ	L3, ММ	L4, ММ	L5, ММ	L6, ММ	L7, ММ	L8, ММ	L9, ММ	L10, ММ	L11*, ММ	II, А	I2, А	N1	N2
1	20	80	25	80	20	25	80	0.02	0.02	25	25	10	10	60	60
2	25	80	20	80	20	20	80	0.01	0.02	20	20	20	10	30	60
3	20	100	20	100	20	20	100	0.03	0.02	20	20	40	10	10	60
4	30	80	30	80	30	30	80	0.04	0.01	30	30	40	30	14	20
5	20	90	20	90	20	20	90	0.02	0.03	20	20	50	5	40	50
6	15	80	15	80	15	15	80	0.05	0.03	15	15	15	10	15	10
7	20	70	20	70	20	20	70	0.06	0.04	20	20	35	30	50	20
8	40	180	40	180	40	40	180	0.02	0.02	40	40	50	7	10	80
9	50	200	50	200	50	50	200	0.07	0.07	50	50	70	20	20	60
10	20	90	20	90	20	20	90	0.01	0.1	20	20	30	10	40	40
11	30	100	30	100	30	30	100	0.05	0.02	30	40	55	15	5	15
12	50	100	50	100	50	50	300	0.08	0.07	50	50	70	40	20	10
13	40	150	40	150	40	40	140	0.05	0.05	60	40	50	60	10	80
14	40	100	40	100	30	30	100	0.05	0.02	30	40	55	15	50	45
15	10	60	10	60	10	10	60	0.03	0.05	10	20	5	15	50	10
16	80	200	80	200	80	80	200	0.1	0.2	80	80	70	60	10	60
17	50	200	50	200	50	50	200	0.1	0.2	50	50	50	60	20	60
18	50	300	50	300	50	50	300	0.4	0.5	50	50	60	60	30	60
19	30	250	30	250	30	30	250	0.05	0.02	30	40	55	15	30	45
20	40	150	40	150	40	40	150	0.05	0.04	40	40	50	30	50	80
21	60	280	60	280	60	60	280	0.02	0.02	60	60	60	50	10	60
22	20	70	20	80	20	10	80	0.03	0.02	10	20	10	10	60	30
23	20	190	20	190	20	20	190	0.2	0.08	20	20	60	30	40	80
24	25	80	25	80	25	25	80	0.02	0.02	25	20	10	10	65	20
25	30	150	30	150	30	30	150	0.04	0.03	30	30	40	35	18	19

-----  
 \* - L11- товщина стрижнів сердечника.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Чуа Л.О., Пен – Мин – Лин. Машинный анализ электронных схем. М.: Энергия 1980.
2. Ортега Дж., Пул У. Введение в численные методы решение дифференциальных уравнений. М.: Наука 1986.
3. Нерретер В. Расчет электрических цепей на персональной ЭВМ. М.: Энергоатомиздат 1991.
4. Лазарев Ю.Ф. MatLAB5.x. – К.:Издательская группа BNV, 2000.
5. Герман-Галкин С.Г. Компьютерное моделирование полупроводниковых систем в MATLAB 6.0: Учебное пособие. – СПб.: КОРОНА принт, 2001.
6. Дьяконов В., Круглов В. MATLAB. Анализ, идентификация и моделирование систем. Специальный справочник. – СПб.: Питер, 2002.