

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ
ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

МЕТОДИЧНІ ВКАЗІВКИ

до курсової роботи з дисципліни
«Математичні методи в електротехніці»

Донецьк – ДонНТУ – 2010

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ
до курсової роботи з дисципліни
« Математичні методи в електротехніці »
(для студентів спеціалізації 6.050702.03
“Електромеханічні системи автоматизації і електропривод”)

З а т в е р д ж е н о
На засіданні кафедри
«Електропривод і автоматизація
промислових установок»

Донецьк, ДонНТУ, 2010

Методичні вказівки до курсової роботи з дисципліни “ Математичні методи в електротехниці” (для студентів спеціалізації 6.050702.03 “Електромеханічні системи автоматизації і електропривод”) / Укл.: О.В.Песковатська – Донецьк: ДонНТУ, 2010. – 35 с.

Приведені завдання до курсової роботи, викладені основні принципи програмування у пакеті Матлаб, приведені теоретичні відомості методів Гауса, Крамера, зворотної матриці для рішення систем лінійних алгебраїчних рівнянь з дійсними коефіцієнтами, а також алгоритми цих методів, сформульовані вимоги до виконання курсової роботи та до форми її представлення.

Призначено для студентів електротехнічних спеціальностей.

Укладачі: О.В.Песковатська, асист.

Відповідальний
за випуск: Толочко О.І., проф.

ВСТУП

Сучасна інженерна і наукова діяльність органічно пов'язана з використанням обчислювальної техніки. Це пояснюється її високою ефективністю, що дозволяє істотно заощаджувати час і витрати, а також створювати принципово нові підходи при рішенні різних технічних, економічних і соціальних задач. Рішення будь-якої задачі на ЕОМ (електронно-обчислювальна машина) вимагає обов'язкового виконання наступних етапів:

- постановка або ж формулювання задачі;
- розробка математичної моделі;
- вибір методу рішення;
- розробка алгоритму рішення;
- складання програми рішення алгоритмічною мовою;
- налагодження програми на ЕОМ і її реалізація;
- аналіз отриманих результатів.

Практичне відпрацювання всіх цих етапів при рішенні конкретних задач і є *метою* курсової роботи. Для демонстрації переваг застосування ЕОМ при рішенні технічних задач взяті завдання до розрахункових робіт по дисципліні “Теоретичні основи електротехніки”(ТОЕ). Одночасно з цим переслідується задача відпрацювання навичок і прийомів об'єктно-орієнтованного програмування (ООП) алгоритмічною мовою C++ [1 – 14] та знайомство з програмуванням в середовищі пакету Matlab (ML) [22-26]. Окрім вище позначеної літератури, що стосується програмування, корисною може стати література з ТОЕ [19 – 21] та література з чисельних математичних методів [15 – 18].

1 ЗАВДАННЯ

Для схеми, що відповідає номеру варіанта, розробити алгоритм і скласти програму що передбачає:

- Розрахунок струмів у гілках за законами Кірхгофа або методом контурних струмів або методом вузлових потенціалів (за завданням викладача).
- Перевірку балансу потужностей.
- Побудування потенційної діаграми для будь-якого замкненого контуру, що включає джерело ЕРС.

Зробити перевірочний розрахунок:

- Розрахунок струмів у гілках матричним методом.
- Промоделювати процеси в електричній схемі в середовищі Simulink (бібліотека Power System Blockset).

Систему лінійних рівнянь з невідомими струмами, розв'язати методом Гауса (непарні варіанти) або методом Крамера (парні варіанти). Для цього створити універсальну власну функцію за відповідним алгоритмом.

Номер варіанту видає викладач. *Перша цифра* вказує на номер числових даних (табл.1.1), а *друга цифра* – на номер варіанту схеми (рис. 1.1). Якщо в таблиці відсутні $E1(E2)$ – у відповідній гілці джерела ЕРС не має, $J1(J2)$ – відсутня гілка з джерелом струму.

Таблиця 1.1

№ вар	r_1 , Ом	r_2 , Ом	r_3 , Ом	r_4 , Ом	r_5 , Ом	r_6 , Ом	r_7 , Ом	$E1$, В	$E2$, В	$J1$, А	$J2$, А
0	8	10	10	4	6	20	4	50	-	-	4
1	8	14	20	12	40	12	6	-	10	3	-
2	10	5	8	12	6	15	5	40	-	2	-
3	12	10	16	6	14	16	4	-	40	-	1
4	8	4	5	8	10	6	4	-	20	4	-
5	20	8	14	4	15	10	6	100	-	-	5

6	10	20	10	10	18	25	5	80	-	-	3
7	4	6	10	8	6	10	5	60	-	2	-
8	5	20	8	6	10	8	6	-	60	-	1
9	20	10	18	4	20	8	8	-	50	2	-

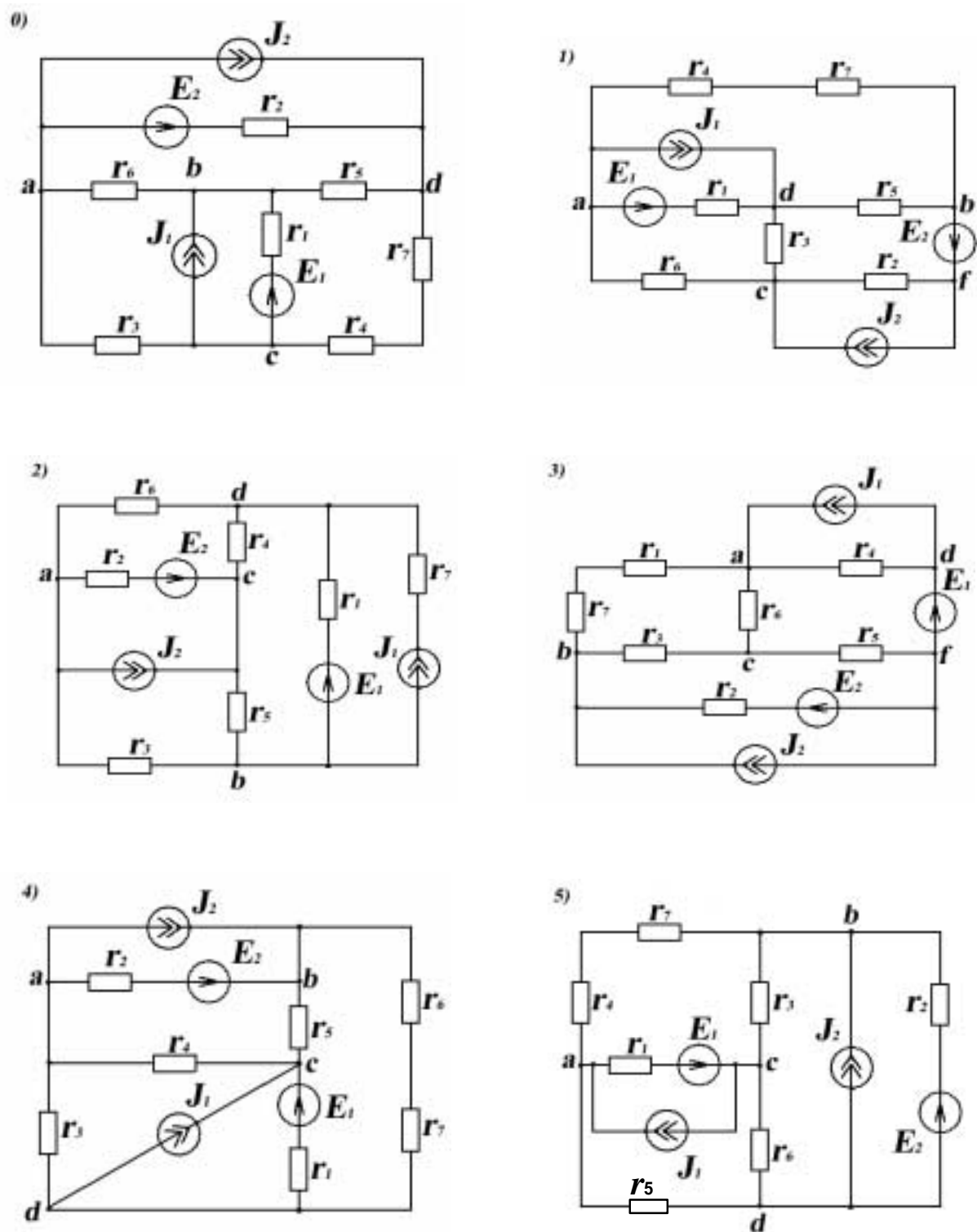


Рисунок 1.1

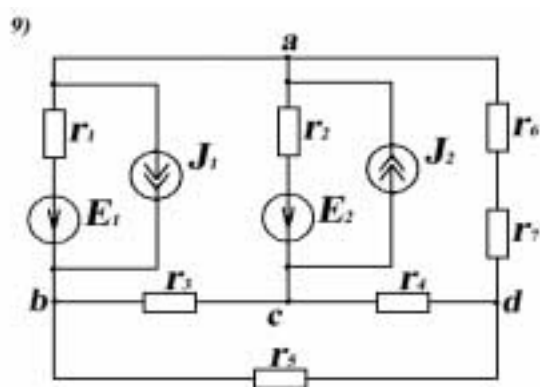
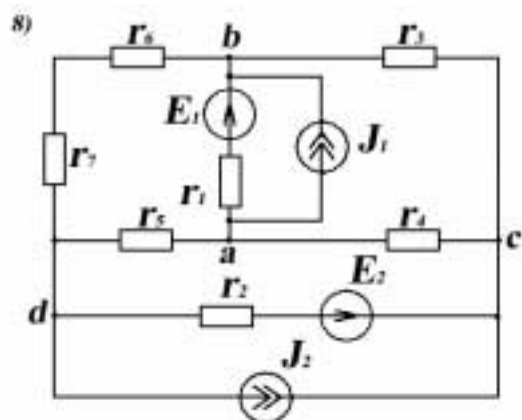
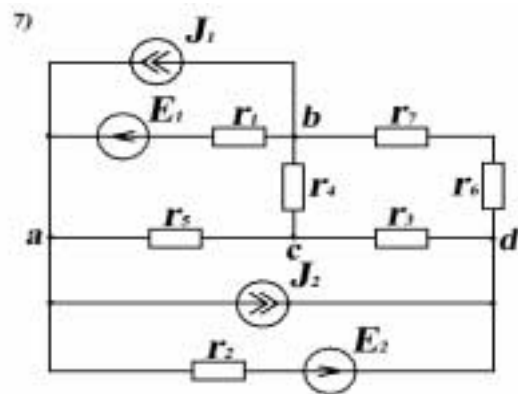
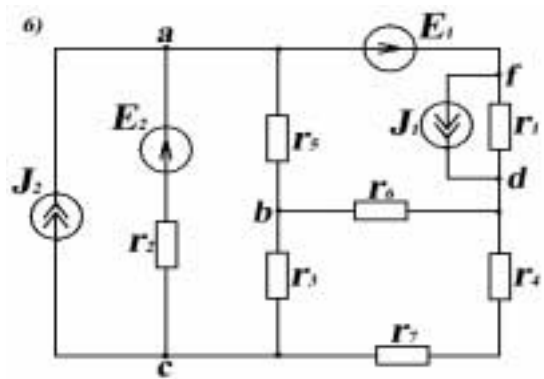


Рисунок 1.1

2 ВИМОГИ ДО ОФОРМЛЕННЯ КУРСОВОЇ РОБОТИ

Курсова робота повинна бути виконана у вигляді пояснювальної записки (30–35 сторінок), надрукованої на папері формату А4 з однієї сторони у відповідності до вимог Держстандарту України.

Звіт складається з титульного аркуша, аркуша завдання, реферату, змісту роботи, вступу, основної частини та висновків. Форма титульного аркуша наведена у додатку А.

Друга сторінка являє собою стандартний аркуш завдання до курсових робіт (проектів), заповнений студентом і підписаний викладачем.

Реферат повинен відбивати головний зміст проведеної роботи. Він будується по такій схемі: відомості про обсяг роботи (кількість сторінок, ілюстрацій, таблиць та джерел); текст реферату; перелік ключових слів. Текст реферату складається зі стислого опису наступних даних: об'єкт дослідження, ціль роботи, методи дослідження, результати і їхня новизна, стислі висновки, можливості й області застосування отриманих результатів. Перелік ключових слів (5–10 слів) характеризує головний зміст роботи. Обсяг реферату – одна сторінка. Приклад оформлення реферату наведений у додатку Б.

У вступі повинна бути відображена роль обчислювальної техніки в сучасному житті, сучасні тенденції в розвитку алгоритмічних мов і т.ін.

Основна частина роботи повинна містити наступні розділи:

1. завдання, в якому дається повний текст поставленої задачі згідно з установленим варіантом студента, початкові дані і електрична схема;

2. розрахунок електричних ланцюгів. В цьому розділі подається опис виводу рівнянь для розрахунку струмів у гілках за законами Кірхгофа або методом контурних струмів або методом вузлових потенціалів, рівняння для перевірки балансу потужностей, а також для побудування потенційної діаграми;

3. математичні методи, в якому приводиться опис математичних методів для розв'язання систем лінійних алгебраїчних рівнянь різними методами (метод Гауса, Крамера, матричний метод);

4. алгоритм обчислень повинен містити блок-схеми алгоритмів кожної процеду-

ри та функції, в тому числі і головної програми. Кожний алгоритм повинен супроводжуватися поясненнями, в яких потрібно вказати призначення алгоритму, а при необхідності і деякі коментарі до нього (наприклад опис параметрів);

5. *програма обчислень* являє собою розпечатку програм на Матлабі.

6. *результати обчислень*, в якому розташовується розпечатка одержаних результатів в заданій формі.

Висновки повинні мати аналіз результатів, підсумки досліджень та основні висновки по роботі.

3 ПРОГРАМУВАННЯ В СЕРЕДОВИЩІ ПАКЕТУ MATLAB

3.1 Загальні поняття

MatLab (Matrix Laboratory – матрична лабораторія) – найбільш розвинена система програмування для науково-технічних розрахунків та призначена перш за все задля програмування чисельних алгоритмів. ML включає в собі великий набір інструкцій (команд) для виконання самих різноманітних обчислень, завдання структур даних і графічного представлення інформації. Також є можливість для виводу графіки у дво- та тривимірному просторі.

Робота в пакеті MatLab можлива безпосередньо у командному вікні (ML Command Window) або за допомогою *m*-функцій. Знизу командного вікна знаходиться поточний командний рядок. Він починається запрошенням `>>`, після якого може йти набір однієї чи кількох (через розділитель) команд.

Присвоєння (тобто вираз типу `var = ...`) і окремі команди повинні відділятися одна від одної розділителем `,` або `;`. Крапка з комою (`;`) зачиняє можливий вивід на екран результатів виконання команди.

Змінна в ML задається своїм ім'ям, яке може мати не більше 31 символу (першою повинна бути латинська буква, потім можуть йти букви, цифри й знак підкреслення). Змінні представлені як чисельні (*double array*), текстові (*char array*), комірки (*cell array*) і структури (*struct array*).

Кожна чисельна змінна являє собою матрицю, тобто таблицю деяких розмірів. Найпростіші – числа: їх розмір 1×1 та вони допускають основні арифметичні дії, які виконуються поелементно; далі йдуть одновимірні масиви – вектор-рядок і вектор-стовпець з розмірами $1 \times m$ і $m \times 1$ (відповідно), де $m > 1$. Двовимірні масиви – матриці, мають розміри $m \times n$, де кількість рядків $m > 1$ та кількість стовпців $n > 1$.

Отримати розмір змінної можна за допомогою команди `[m,n]=size(var)`. Для векторів так звану довжину можна отримати командою `m=length(var)`. Більш детальну інформацію про усі змінні у робочому просторі ML видає команда `whos`. А саме ім'я (*name*), розмір (*size*), скільки займає байт пам'яті (*bytes*) та до якого типу відноситься (*class*).

Для побудування графіка функції $y(x)$ найпоширеною є функція $plot(x, y, color\&type_line)$, де x – вектор аргументу, y – вектор або матриця значень функції. Колір та тип лінії або маркера завдаються як текстова змінна (наприклад: ‘r-’ – точки функції з’єднуються червоною прямою лінією; ‘g*’ – точки функції у вигляді зелених зірок).

3.2 Операції над матрицями у ML

У системі ML реалізовано два типи арифметичних операцій: поелементні та матричні. Матричні операції виконуються у відповідності з правилами лінійної алгебри. Для того, щоб їх розрізнити, поелементним відповідає крапка перед символом операції. Наприклад: \cdot^* – поелементне множення, \cdot^{\wedge} – поелементне піднесення у ступінь.

Сумою або різницею двох матриць однакових розмірів: $C = A \pm B$ є матриця такого ж розміру, елементи якої дорівнюють сумах або різницям відповідних елементів матриць A і B :

Для цих дій поелементні та матричні операції співпадають.

Добутком матриці A розміром $m \times k$ на матрицю B розміром $k \times n$ є матриця C розміром $m \times n$, елементи якої вираховуються за формулою:

$$c_{ij} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + \dots + a_{ik} \cdot b_{kj} = \sum_{l=1}^k a_{il} \cdot b_{lj}.$$

Відмітимо, що матриця $C=A \cdot B$ визначена тільки тоді, коли число стовпців матриці A дорівнює числу рядків матриці B .

Для добутку матриць, у загальному випадку, несправедливий перемістительний закон, тобто $A \cdot B \neq B \cdot A$.

У відповідності з поняттям про добуток матриць, у цілу додаткову ступінь k можна звести тільки квадратну матрицю:

$$A^k = \underbrace{((A \cdot A) \cdot A) \cdot \dots \cdot A}_{k \text{ співмножників}}.$$

k співмножників

Ця операція у ML зображується символом \cdot^{\wedge} .

Крім того, у ML передбачені операції правостороннього та лівостороннього ділення матриць:

\mathbf{B}/\mathbf{A} відповідає $\mathbf{B}*\mathbf{A}^{-1}$;

$\mathbf{A}\backslash\mathbf{B}$ відповідає $\mathbf{A}^{-1}*\mathbf{B}$, де \mathbf{A}^{-1} – обернена матриця.

Якщо у матриці \mathbf{A} розміром $m*n$ замінити рядки відповідними стовпцями, то одержимо матрицю \mathbf{A}^T розміром $n*m$, яка має назву *транспонованої* у відношенні до матриці \mathbf{A} . У ML ця операція позначається символом $'$.

Для обчислення *визначника* матриці \mathbf{A} у ML використовують функцію $det(\mathbf{A})$. Для обчислення *оберненої* матриці використовують функцію $inv(\mathbf{A})$.

3.3 Рішення систем лінійних рівнянь у ML

Система n лінійних рівнянь з n у матричній формі має вигляд:

$$\mathbf{A}*\mathbf{X}=\mathbf{B},$$

$$\text{де } \mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} - \text{квадратна матриця коефіцієнтів; } \mathbf{B} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix} -$$

$$\text{вектор вільних членів; } \mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} - \text{шуканий вектор коренів.}$$

Якщо матриця \mathbf{A} неособлива, тобто її визначник не дорівнює нулю, то система має єдине рішення:

$$\mathbf{X}=\mathbf{A}^{-1}*\mathbf{B},$$

де \mathbf{A}^{-1} -матриця, обернена до матриці \mathbf{A} .

У ML рішення системи лінійних рівнянь може бути досягнуто за допомогою лівостороннього ділення або оберненої матриці:

$$X=A\backslash B,$$

$$X=inv(A)*B,$$

$$X=A^{-1}*B.$$

3.4 Програмування за допомогою власних функцій у ML

Функції (functions) повинні бути у окремому файлі і містити заголовок, коментар і текст функції. Заголовок має наступний формат:

```
function [список_вих_параметрів] = ім'я_функції (список_вх_параметрів)
```

Вхідні і вихідні параметри в заголовку функції є формальними.

Якщо функція не потребує наявності вхідних і (або) вихідних параметрів, вони в заголовку функції записуються у вигляді порожнього вектора [], або зовсім відсутні.

Після заголовку може бути записаний коментар (рядок чи декілька рядків символів після знаку %), який діє як довідка про функцію і виводиться на екран за допомогою команди

```
help ім'я_функції
```

Після коментаря записують послідовність операторів, що складають тіло функції. Запис функції потрібно виконати улюблений доступний каталог з ім'ям, яке співпадає з ім'ям функції і розширенням .m.

Для виконання функції до неї треба звернутися з програми, другої функції або з командного рядка, вказавши фактичні вхідні і вихідні параметри

```
[список_вих_параметрів]= ім'я (список_вх_параметрів)
```

Вхідні параметри (аргументи функції) обов'язково повинні бути на момент виклику визначені (тобто мати значення). Замість списку вхідних параметрів можна використовувати безпосередньо чисельні дані.

3.5 Програмування умов і циклів у ML

Формат умовного оператора **if**:

```
if вираз_1
```

```
оператори_1
```

```
elseif вираз_2
```

```
оператори_2
```

else

оператори_3

end

Число альтернативних гілок *elseif* не обмежено.

При записуванні виразів 1 і 2 використовуються логічні операції порівняння:

$=$, $<$, $>$, $<=$, $>=$, $\sim =$.

Формат оператора циклу **for**:

for і'мя_змінної = вираз

оператори

end

Змінна циклу може бути цілою, дійсною, вектором або матрицею.

Найчастіше використовують вектор у вигляді

поч._значення : шаг : кінц_значення

Формат оператора циклу **while**:

while вираз

оператори

end

Цикл триває доки виконується вираз.

4. ОГЛЯД МЕТОДІВ РІШЕННЯ ЛІНІЙНИХ РІВНЯНЬ

В практичній діяльності фахівців електротехнічної галузі дуже часто трапляється розраховувати різноманітні параметри електричних схем. Математичні моделі таких схем в своїй більшості описуються системами лінійних алгебраїчних рівнянь з дійсними коефіцієнтами. Отже дуже важливо знати основні прийоми рішення таких систем. Існують різноманітні методи рішення цих задач. Але найбільш поширеними і зручними при використанні обчислювальної техніки є декілька із них. До них відносяться методи Гауса, Крамера і зворотної матриці. Ці методи і рекомендуються для використання при виконанні курсової роботи. Приймаючи до уваги, що суть цих методів для студентів спеціальності “Електромеханічні системи автоматизації і електропривод” вже знайома з “Вищої математики”, далі приводяться лише їх основні положення з акцентуванням на особливостях їх використання при виконанні розрахунків на комп’ютері.

Способи рішення систем лінійних рівнянь поділяються на дві групи:

- точні методи (метод обернення матриці коефіцієнтів, правило Крамера, метод Гауса та інші);
- ітераційні методи (Ньютона, Зейделя, простих ітерацій та інші).

4.1 Метод Гауса

Цей метод є найбільш поширеним і зручним до застосування на ЕОМ. Ефективність його в порівнянні з “ручним” рішенням помітно виявляється в разі, якщо порядок системи алгебраїчних рівнянь більше трьох-чотирьох.

У загальному вигляді систему алгебраїчних рівнянь з дійсними коефіцієнтами можна записати так:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases} .$$

Метою рішення такої системи і є знаходження значень коренів x_1, x_2, \dots, x_n .

Метод Гауса передбачає рішення цієї задачі в два етапи:

- На першому етапі, який називається *прямий хід*, виконується послідовне виключення коренів, починаючи з першого (x_1) до передостаннього (x_{n-1}). При цьому матриця коефіцієнтів A перетворюється з *квадратної* до *трикутної*.

- На другому етапі, який називається *зворотний хід*, виконується послідовне обчислення коренів з перетвореної на першому етапі системи рівнянь у зворотному напрямку, починаючи з останнього (x_n) і до першого (x_1).

Після виконання першого етапу система рівнянь приймає вигляд:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ 0 + a_{22}^*x_2 + \dots + a_{2n}^*x_n = b_2^* \\ \dots \cdot \dots \cdot \dots \cdot \dots \cdot \dots \cdot \dots \\ 0 + 0 + \dots + a_{nn}^*x_n = b_n^* \end{cases}$$

В цьому випадку, отримані після перетворень коефіцієнти при невідомих коренях та вільні члени будуть мати значення, які відрізняються від початкових, у всіх рівняннях окрім першого. Вони позначаються символом (*).

У загальному випадку виключення k -го кореня ($k = 1, 2, 3 \dots, n-1$) з i -го рівняння ($i = k+1, k+2, \dots, n$) виконується шляхом заміни усіх коефіцієнтів i -го рівняння різницею між початковими коефіцієнтами цього рівняння (до перетворення) і відповідними коефіцієнтами k -го рівняння, помноженими на масштабний множник, який обчислюється таким чином:

$$p = \frac{a_{ik}}{a_{kk}}$$

Внаслідок перетворень коефіцієнти i -го рівняння приймуть наступні значення:

$$a_{ik}^* = 0, a_{ij}^* = a_{ij} - pa_{kj}$$

де $j = k+1, k+2, \dots, n$,

а вільний член:

$$b_i^* = b_i - pb_k$$

Обчислення коренів на етапі зворотного ходу виконується з останнього:

$$x_n = \frac{b_n^*}{a_{nn}^*}$$

Потім, його значення x_n підставляється у передостаннє рівняння і з нього обчислюється корінь x_{n-1} і так далі до першого. В загальному випадку формула для обчислення коренів має вигляд:

$$x_i = \frac{b_i^* - \sum_{j=i+1}^n a_{ij}^* x_j}{a_{ij}^*}.$$

Для того, щоб зменшити похибку обчислення при перетворенні коефіцієнтів на етапі прямого ходу, початкову систему рівнянь необхідно перетворити шляхом взаємної перестановки рівнянь, що не вплине на рішення системи. Ця перестановка повинна бути виконана таким чином, щоб модуль коефіцієнта a_{kk} при виключаємо-му корені x_k був максимальним.

Алгоритми методу Гауса можна спростити, якщо об'єднати квадратну матрицю коефіцієнтів $A(n \times n)$ та вектор-стовпець $B(n)$ у одну матрицю $AR(n \times (n + 1))$, яка називається *розширеною матрицею коефіцієнтів*. В ній $(n+1)$ -й стовпець складають елементи вектора вільних членів $B(n)$. Відповідно змінні b_n і b_i замінюються змінними $a_{n,n+1}$ і $a_{i,n+1}$.

Алгоритми прямого і зворотного ходів приведені у додатку В.

4.2 Метод Крамера

Суть методу зручно зрозуміти на прикладі рішення системи двох алгебраїчних рівнянь з двома невідомими.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases},$$

або у векторній формі $\mathbf{AX} = \mathbf{B}$,

$$\text{де } \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

Нехай відомо, що визначник матриці \mathbf{A} не є нульовий, тобто $\det A \neq 0$.

Якщо кожне з рівнянь системи помножити відповідно на алгебраїчні допов-

нення ad_{11} і ad_{21} , а потім отримані рівняння скласти, то для першого рядка отримаємо:

$$(a_{11}ad_{11} + a_{21}ad_{21})x_1 + (a_{12}ad_{11} + a_{22}ad_{21})x_2 = b_1ad_{11} + b_2ad_{21}.$$

За правилами математики

$$a_{11} = ad_{22},$$

$$a_{22} = ad_{11},$$

$$-a_{12} = ad_{21},$$

$$-a_{21} = ad_{12},$$

тому після підстановки одержимо:

$$(a_{11}a_{22} - a_{12}a_{21})x_1 + (a_{12}a_{22} - a_{22}a_{12})x_2 = b_1a_{22} - b_2a_{12}.$$

Так як

$$(a_{11}a_{22} - a_{12}a_{21}) = \det A,$$

$$(a_{12}a_{22} - a_{22}a_{12})x_2 = 0,$$

$$b_1a_{22} - b_2a_{12} = \begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{vmatrix} = \det B_1,$$

то:

$$\det A \cdot x_1 = \det B_1,$$

звідки:

$$x_1 = \frac{\det B_1}{\det A}.$$

Аналогічно можна отримати формулу обчислення x_2 з другого рядка.

Наведені положення та формули дійсні для матриць любого розміру. Якщо визначник основної матриці A системи n лінійних алгебраїчних рівнянь з n невідомими не дорівнює нулю ($\det A \neq 0$), то така система має єдине рішення, яке визначається за формулою

$$x_i = \frac{\det B_i}{\det A},$$

де $i = 1, 2, \dots, n$.

Наприклад $B_1 = \begin{bmatrix} b_1 & a_{12} & \dots & a_{1n} \\ b_2 & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ b_n & a_{n2} & \dots & a_{nn} \end{bmatrix}$.

Матриця B_i формується шляхом заміни i -го стовпця матриці \mathbf{A} на стовпець \mathbf{B} .

Алгоритм методу Крамера приведений у додатку Е. На цій блок-схемі присутні дві функції: $bufer()$ і $det()$. Функція $bufer()$ призначена для запам'ятовування матриці коефіцієнтів з метою подальшого її перетворення. Функція $det()$ призначена для обчислення визначника квадратної матриці будь-яким із відомих методів, наприклад, через алгебраїчні доповнення елементів матриці або за допомогою прямого ходу Гауса. Алгоритми обчислення алгебраїчного доповнення $ad_{k,l}$ k -ого рядка l -ого стовпця матриці \mathbf{A} розміром $n*n$ і визначника матриці \mathbf{A} наведені відповідно у додатках Д і Г.

5 РОЗРАХУНОК СТРУМІВ

5.1 Закони Кірхгофа

1-ий закон Кірхгофа – алгебраїчна сума струмів гілок, під'єднаних до одного вузла дорівнює нулю:

$$\sum I = 0.$$

Для кожного вузла приймаємо позитивний або негативний напрям струму. Число незалежних рівнянь, які складені за 1-им законом Кірхгофа, дорівнює $m-1$, де m – число вузлів у схемі.

2-ий закон Кірхгофа – алгебраїчна сума напруг на елементах довільного контуру дорівнює алгебраїчній сумі ЕРС цього контуру. Рівняння складаються для незалежних контурів. Кількість рівнянь за II законом Кірхгофа складає $(n-m)$ (де n – кількість гілок з невідомими струмами)

$$\sum IR = \sum E.$$

Таким чином отримаємо систему n лінійних рівнянь з n невідомими, яку можна представити у вигляді $\mathbf{A} \cdot \mathbf{I} = \mathbf{B}$, де \mathbf{A} – матриця коефіцієнтів, \mathbf{I} – вектор невідомих струмів, \mathbf{B} – вектор вільних членів.

5.2 Матричний метод

Для обчислення струмів та напруг у електричних колах матричним методом спершу необхідно визначити наступні параметри:

q – кількість вузлів у схемі;

$m=q-1$ – кількість незалежних вузлів;

p – кількість гілок з невідомими струмами;

$n=p-m$ – кількість незалежних контурів.

Після цього складаємо матриці незалежних контурів \mathbf{G} розміром $(p*n)$, з'єднань \mathbf{D} розміром $(m*p)$, а також вектори ЕРС \mathbf{E} розміром $(p*1)$, джерел струму \mathbf{J}_k

розміром $(m \times 1)$, квадратну діагональну матрицю опорів \mathbf{Z} розміром $(p \times p)$.

Елементи матриць \mathbf{G} і \mathbf{D} утворюються згідно з формулами:

$$G_{ij} = \begin{cases} 1, \text{ коли напрям } i - \text{ ої гілки} \\ \quad \text{співпадає з напрямом } j - \text{ ого контура;} \\ -1, \text{ навпаки;} \\ 0, \text{ коли гілка } i \text{ не належить контуру } j; \end{cases}$$

$$D_{ij} = \begin{cases} 1, \text{ коли напрям } j - \text{ ої гілки до вузла } i; \\ -1, \text{ навпаки;} \\ 0, \text{ коли гілка } j \text{ не підходить до вузла } i. \end{cases}$$

Запишемо рівняння у матричній формі за законом Ома та I і II законами Кірхгофа:

$$\mathbf{U}_{(p,1)} = \mathbf{E}_{(p,1)} - \mathbf{Z}_{(p,p)} * \mathbf{I}_{(p,1)}, \quad (1)$$

$$\mathbf{D}_{(m,p)} * \mathbf{I}_{(p,1)} + \mathbf{J}_{k(m,1)} = \mathbf{0}, \quad (2)$$

$$\mathbf{G}_{(n,p)}^T * \mathbf{U}_{(p,1)} = \mathbf{0},$$

Помноживши (1) на \mathbf{G}^T зліва, маємо:

$$\mathbf{G}^T * \mathbf{U} = \mathbf{G}^T * \mathbf{E} - \mathbf{G}^T * \mathbf{Z} * \mathbf{I} = \mathbf{0},$$

$$\mathbf{G}^T * \mathbf{Z} * \mathbf{I} = \mathbf{G}^T * \mathbf{E}. \quad (3)$$

Об'єднаємо рівняння (2) та (3) в одну систему:

$$\begin{cases} \mathbf{D} * \mathbf{I} = -\mathbf{J}_k \\ \mathbf{G}^T * \mathbf{Z} * \mathbf{I} = \mathbf{G}^T * \mathbf{E}. \end{cases}$$

Позначивши матрицю коефіцієнтів як \mathbf{A} і вектор вільних членів як \mathbf{B} :

$$\mathbf{A} = \begin{bmatrix} \mathbf{D} \\ \mathbf{G}^T * \mathbf{Z} \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} -\mathbf{J}_k \\ \mathbf{G}^T * \mathbf{E} \end{bmatrix},$$

отримаємо систему лінійних рівнянь $\mathbf{A} * \mathbf{I} = \mathbf{B}$, яку можна вирішити любим з відомих методів.

5.3 Перевірка балансу потужностей

Перевірка балансу потужностей необхідна для перевірки результатів обчислення струмів гілок електричної схеми. Для роботи схеми у нормальному режимі, необхідно щоб потужність, яка генерується джерелами живлення схеми, дорівнювала потужності споживачем.

$$P_{дж} = \sum EI + \sum U_k I_k = \sum I^2 R = P_{сп},$$

де E – ЕРС;

I_k – струм джерела струму;

U_k – напруга на джерелі струму;

I – струми гілок;

R – відповідні опори гілок.

Баланс потужностей не повинен перевищувати 5 %.

$$\delta P = \left| \frac{\sum P_{дж} - \sum P_{сп}}{\sum P_{дж}} \right| \cdot 100\% \leq 5\%.$$

5.4 Побудування потенціальної діаграми для контуру, який має ЕРС

Потенціальна діаграма (ПД) є однією з перевірок струмів у схемі. При обході якого-небудь замкненого контуру потенціал вузла, з якого починається обхід, приймаємо рівним нулю, і якщо після закінчення обходу останній потенціал (тобто потенціал з якого починали обхід) дорівнює нулю, то струми в контурі розраховані правильно.

Для побудови потенціальної діаграми $\vartheta(R)$ треба розрахувати вектор потенціалів і відповідний до нього вектор опорів.

Наприклад:

$$\vartheta = [0 \quad \varphi_1 \quad \varphi_2 \quad \dots \quad \varphi_k \quad 0],$$

$$R = [0 \quad r_1 \quad r_1 + r_2 \quad \dots \quad r_1 + r_2 + \dots + r_k],$$

та з'єднати відповідні точки прямими лініями.

5.5 Розрахунок струмів методом контурних струмів

МКС – це метод розрахунку, в якому за невідомі приймають контурні струми, а потім вже через них обчислюють струми в гілках електричної схеми.

Вздовж кожного незалежного контуру протікає свій струм, тому число струмів дорівнює числу незалежних контурів (числу гілок зв'язку).

Струми в гілках визначаються спільною дією контурних струмів, які протікають по цим гілкам. Струм контуру помножується на суму опорів контуру, потім опір, який належить двом контурам помножується на струм контуру, сполученого з даним контуром і знак визначається за напрямком контурних струмів. Якщо в контурі присутня ЕРС вона записується в правій частині рівняння як за 2-им законом Кірхгофа.

Гілка з джерелом струму виділяється як гілка зв'язку. Ця гілка входить тільки до одного контуру і його контурний струм дорівнює струму джерела струму. Для цього контуру складати рівняння не треба, але для інших контурів цей струм треба враховувати.

Потім струми в гілках електричної схеми обчислюються за допомогою обчислених контурних струмів.

5.6 Розрахунок струмів методом вузлових потенціалів

МВП – це метод розрахунку, в якому за невідомі приймають вузлові потенціали. Один з вузлів схеми заземлюють і його потенціал приймають рівним нулю.

Напруга на відрізку ланцюга:

$$U = \sum IR = \sum \left(I \frac{1}{G} \right),$$

де $G = \frac{1}{R}$ провідність гілки.

Для кожного незалежного вузла складається рівняння. Потенціал вузла помножується на суму провідностей безпосередньо під'єднаних до цього вузла, потім зі знаком “–” інші потенціали помножені на провідності, які з'єднують дані вузли. Все це дорівнює сумі джерел струмів та джерел напруг на провідність гілки в

яких воно знаходиться.

$$\begin{cases} \Phi_1 g_{11} - \Phi_2 g_{12} - \dots - \Phi_n g_{1n} = \sum E_g + \sum J \\ \Phi_1 g_{21} - \Phi_2 g_{22} - \dots - \Phi_n g_{2n} = \sum_2 E_g + \sum_2 J \\ \dots \end{cases}$$

Потім струми в гілках електричної схеми обчислюються за допомогою обчислених значень потенціалів за законом Ома.

6. МОДЕЛЮВАННЯ ЕЛЕКТРИЧНОЇ СХЕМИ В СЕРЕДОВИЩІ SIMULINK ПАКЕТА MATLAB

В пакеті Matlab існує спеціальний моделюючий додаток Simulink, в якому можна набирати різні схеми, наприклад структурні, електричні та ін., і проводити процес математичного моделювання.

Для того, щоб скласти електричну схему, треба скористатися спеціальною бібліотекою Power System Blockset (див. рис. 6.1), підбібліотеки якої містять різноманітні елементи принципів електричних схем. Перелік деяких потрібних блоків Power System наданий нижче.

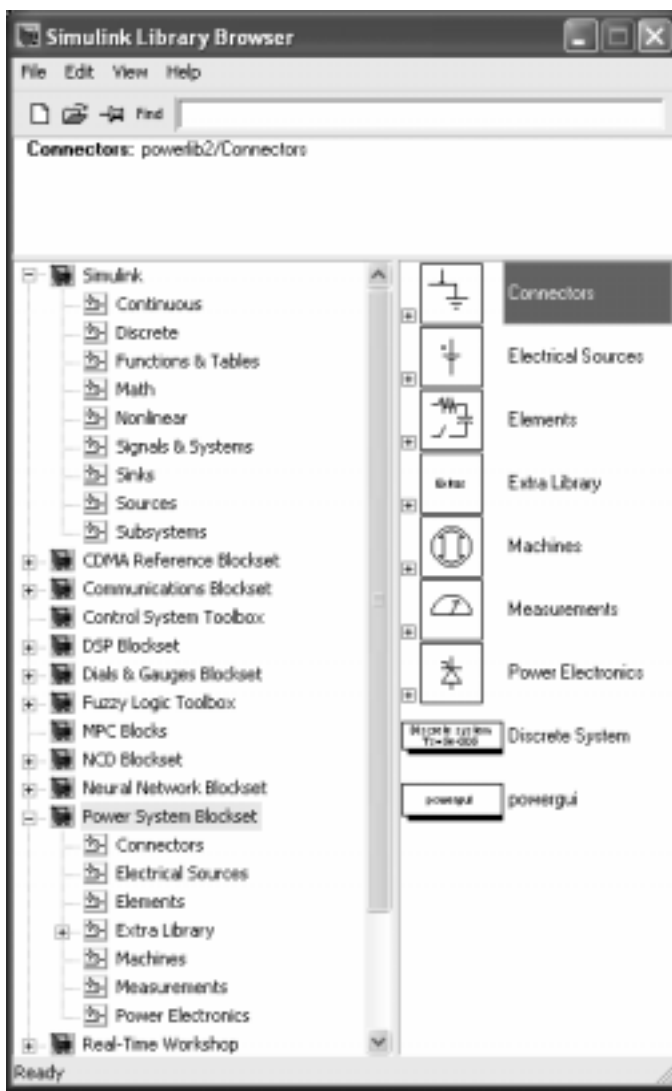


Рисунок 6.1

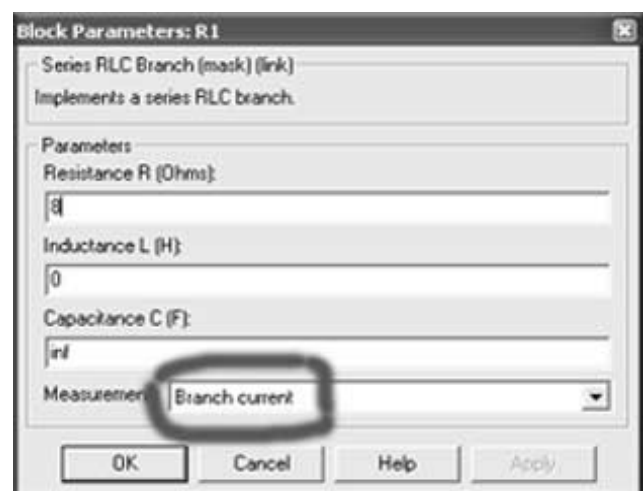


Рисунок 6.2

Connectors:

Ground – заземлення;

T-connector – Т-конектор;

L-connector – L- конектор;

Bas Bar – мультиплексор;

Electrical Sources:

DC Voltage Source – джерело постійної напруги;

AC Voltage Source – джерело синусоїдальної напруги;

AC Current Source – джерело синусоїдального струму;

Controlled Voltage Source – керуєме джерело напруги;

Controlled Current Source – керуєме джерело струму;

Elements:

Breaker – ключ;

Series RLC Branch – послідовний RLC ланцюг;

Parallel RLC Branch – паралельний RLC ланцюг;

Measurements:

Current measurements – амперметр;

Voltage measurements – вольтметр;

Multimeter – універсальний вимірювач;

Power Electronics:

Ideal Switch – ключ.


При використанні блоків “Parallel RLC Branch” для RL-ланцюга приймаємо $C=0$, для RC-ланцюга – $L=\infty$, для LC-ланцюга – $R=\infty$; при використанні блоків “Series RLC Branch” для RL-ланцюга приймаємо $C=\infty$, для RC-ланцюга – $L=0$, для LC-ланцюга – $R=0$.

Блоки “Breaker” і “Ideal Switch” використовують для замкнення та розімкнення електричного кола. Вхід “1” та вихід “2” приєднуються до електричної схеми. Початкові умови “initial state” завдають стан ключа (“0” – розімкнений, “1” – замкнений). Для зміни стану ключа на вхід “c” (“Breaker”) або “g” (“Ideal Switch”) треба подати керуючий сигнал Simulink. Особливості блока “Breaker” (зроблений у вигляді джерела струму) – не рекомендовано використовувати у послідовному зв’язку з джерелом струма та індуктивністю (в цих випадках треба зашунтувати вхід-вихід

резистором з дуже великим опір). Особливості блока “Ideal Switch” (зроблений у вигляді зашунтованого послідовного RC-ланцюга) – вихід “m” використовують для виведення струму і напруги ключа під час моделювання. Якщо це непотрібно до вихіду “m” під’єднують термінатор.

Для фіксування результатів можна скористатися блоками бібліотеки Simulink Sinks – *Display* (дисплей), *Scope* (осцилограф), *To Workspace* (в пам’ять).

Перед моделюванням усі значення елементів схеми повинні бути визначеними. Для під’єднання гілок у вузол використовуйте конектори або заземлення.

Для моделювання використовують кнопку  на панелі інструментів або запускають моделювання із меню Simulation / Start (Ctrl+T).

Параметри моделювання встановлюються в меню Simulation/Parameters:

Start time/Stop time – початковий і кінцевий час моделювання,

Solver option – вибір методу чисельного інтегрування.

При використанні блока *Multimeter* для вимірювання струму у гілці з опором треба в меню параметрів блоку *RLC Branch* встановити значення Branch current (див.рис.6.2).

В меню параметрів блоку *Multimeter* треба необхідні струми на опорах обрати для вимірювання (див.рис.6.3).

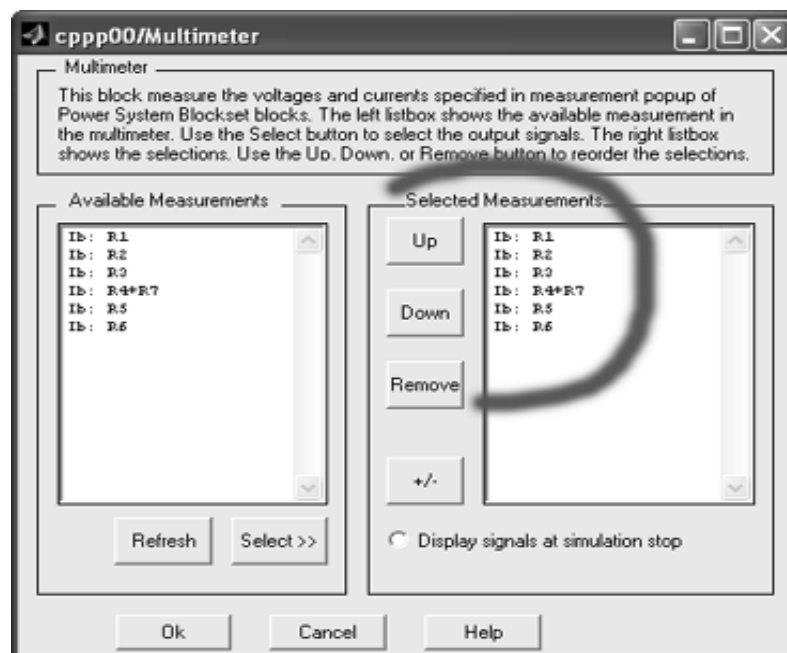


Рисунок 6.3

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Кафедра ЕАПУ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до курсової роботи з дисципліни
“Математичні методи в електротехніці”
на тему “*Розрахунок електричних розгалужених ланцюгів*”

Варіант № ____

Виконав:

ст. гр. ЕАПУ-

Іванов І.І.

Перевірив:

ас. Песковатська О.В.

РЕФЕРАТ

Пояснювальна записка:

___ сторінок, ___ рисунків, ___ додатків, ___ джерел

Об'єктом досліджень даної курсової роботи є розгалужена електрична схема.

Мета роботи – знайти струми в гілках електричної схеми вирішивши систему лінійних алгебраїчних рівнянь математичними методами.

Програма до курсової роботи розроблено та реалізовано мовою Матлаб.

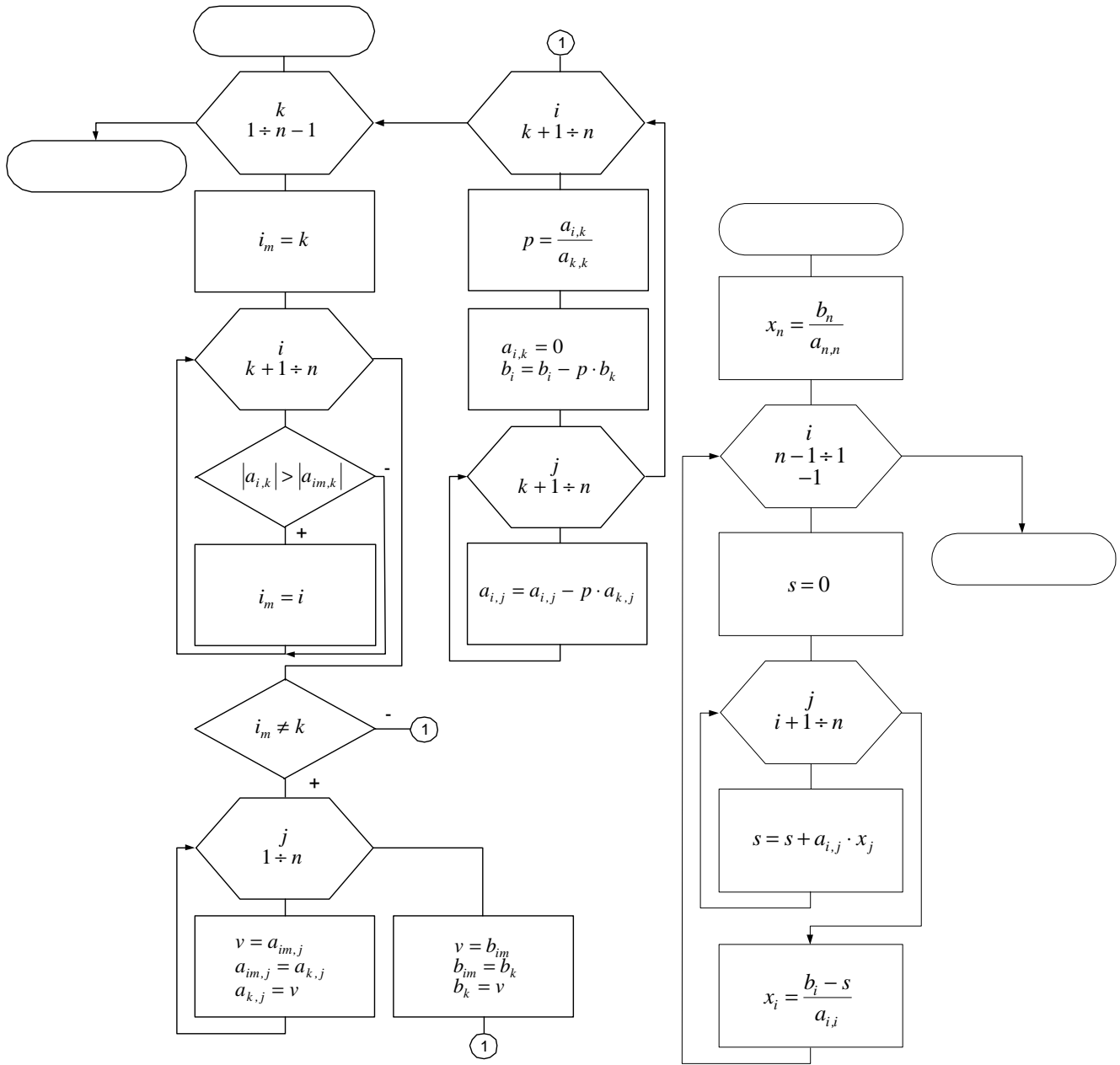
У пояснювальній записці наведені також блок-схеми розроблених алгоритмів програми, математичний опис знаходження струмів у електричному ланцюгу, математичний опис методів рішення систем лінійних рівнянь.

Актуальність роботи полягає в тому, що на прикладі розгалуженої електричної схеми показано використання ЕОМ, а саме програмування в середовищі пакету Матлаб, яке підвищує точність розрахунків та полегшує саме обчислення.

РОЗГАЛУЖЕНА ЕЛЕКТРИЧНА СХЕМА, СТРУМ,
ПОТЕНЦІАЛЬНА ДІАГРАМА, БАЛАНС ПОТУЖНОСТЕЙ,
МЕТОД ГАУСА, ЗАКОНИ КІРХГОФА

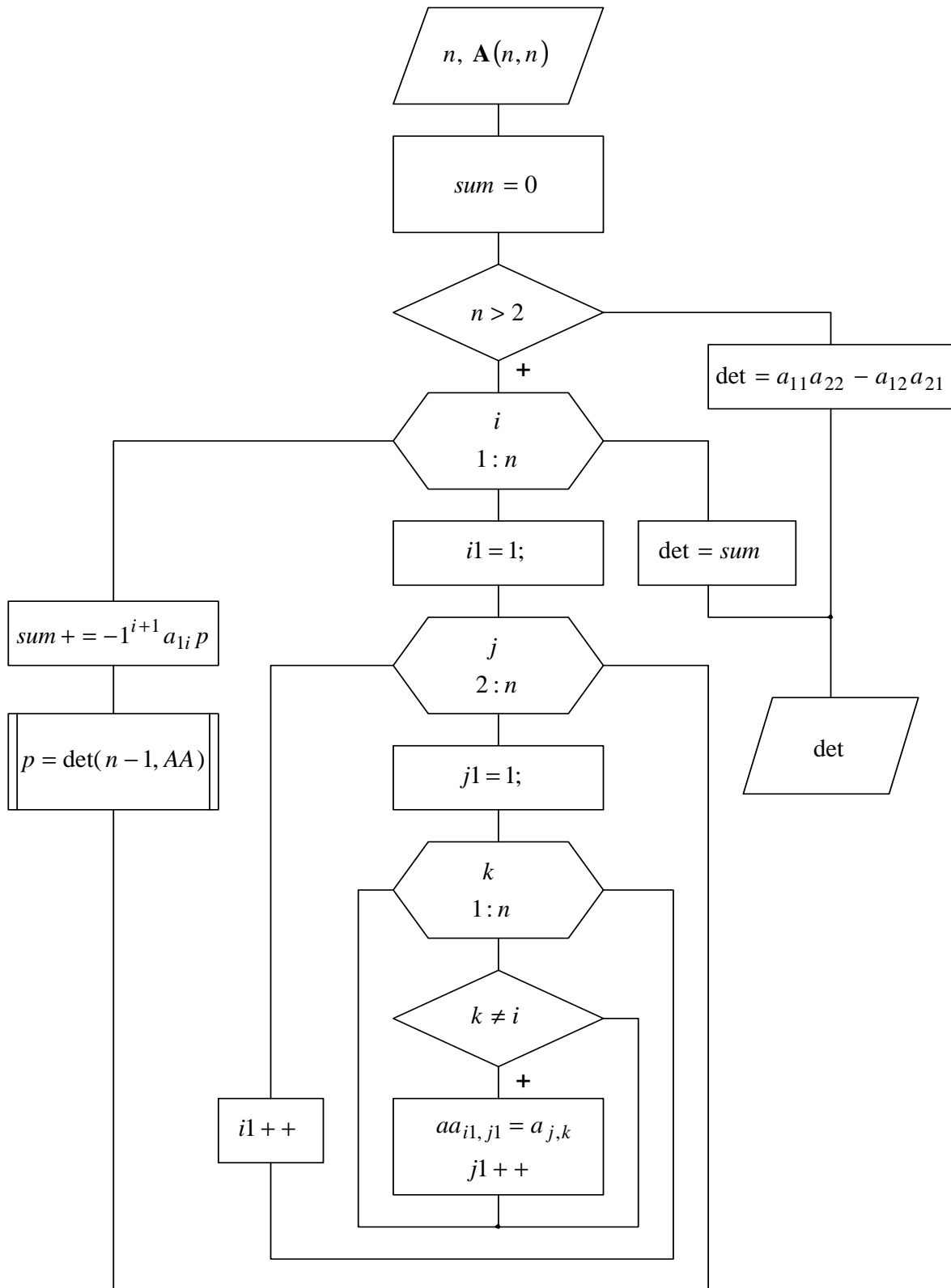
Додаток В

Блок-схема алгоритму прямого і зворотного ходу Гауса



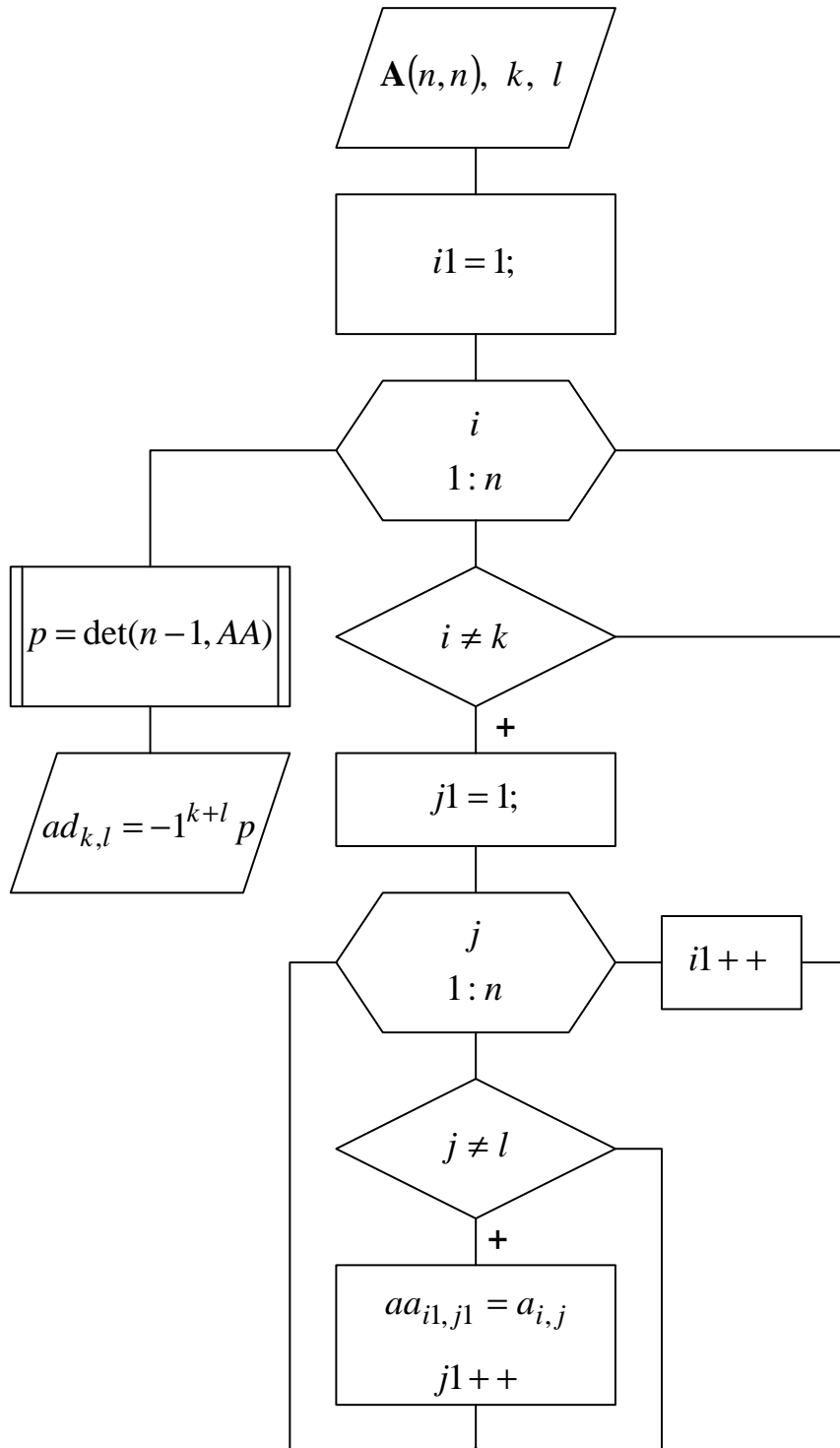
Додаток Г

Блок-схема алгоритму обчислення визначника матриці A



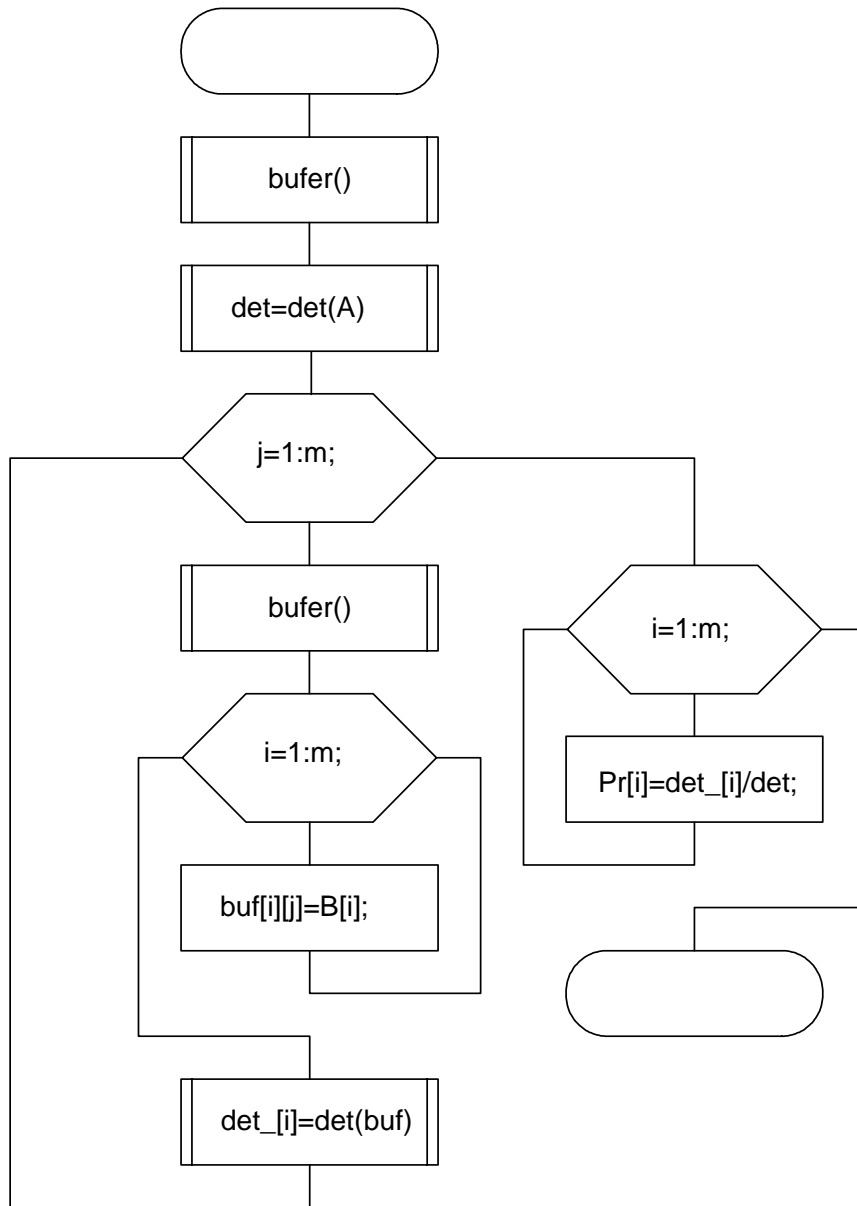
Додаток Д

Блок-схема алгоритму обчислення алгебраїчного доповнення
елемента (k,l) матриці A



Додаток E

Блок-схема алгоритму методу Крамера



ЛІТЕРАТУРА

1. Бабэ Б. Просто и ясно о Borland C++. – М.: БИНОМ, 1995. – 400 с.
2. Дейтел Х.М., Дейтел П.Дж. Как программировать на C++: Пер. с англ. – М.: ЗАО “Издательство Бином”, 1999 г.. –1024 с.: ил.
3. Дэвис С. C++ для «чайников». – К.: Диалектика, 1996. – 304 с.
4. Голуб А.И. С и C++. Правила программирования. – М.: БИНОМ, 1996. – 272 с.
5. Карпов Б., Баранова Т. C++: специальный справочник. – СПб: «Питер», 2000. – 480 с.
6. Луис Д. С и C++. Справочник. – М.: Восточная книжная компания, 1997. – 592 с.
7. Подбельский В.В. Язык C++: Учебное пособие. – М.: Финансы и статистика, 1996. – 560 с.
8. Пол А. Объектно-ориентированное программирование на C++. – 2-е изд. – Москва, Санкт-Петербург, 1999. – 461 с.
9. С/C++. Программирование на языке высокого уровня / Т.А. Павловская. – СПб.: «Питер», 2001. – 464 с.: ил.
10. Страуструп Б. Дизайн и эволюция языка C++. – М.: ДМК, 2000. – 448 с.
11. Страуструп Б. Язык программирования C++. – 3-е изд. – СПб.: «Невский диалект», 1999. – 991 с.
12. Франка П. C++. Учебный курс. – СПб.: «Питер», 1999. – 521 с.
13. Шилдт Г. Теория и практика C++. – 3-е изд., перераб. и доп. – СПб.: ВHV, 1998. – 683 с.
14. Элджер Д. C++: Библиотека программиста. Серия: Библиотека программиста. – СПб.: «Питер», 1999. – 320 с.
15. Дж. Форсайт, М. Малькольм, К. Моулер. Машинные методы математических вычислений. М.:Мир, 1980. – 279 с.
16. Мудров А.Е. Численные методы для ПЭВМ на языках Бейсик, Фортран, Паскаль. Томск: МП “Раско”. 1991. – 272 с.
17. Д. Мак-Кракен, У. Дорн. Численные методы и программирование на Фортра-

не. 2-е изд., пер. с англ.. под ред. Б.М. Наймерка. М.: Мир. 1977. – 584 с.

18. Мэтьюз Д.Г., Финк К.Д. Численные методы. Использование MATLAB, 3-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2001. – 720 с.: ил.

19. Л. А. Бессонов. Теоретические основы электротехники. М. ВШ. 1973. – 749 с.

20. Теоретические основы электротехники. / Под.ред. П.А.Ионкина. М.:ВШ. 1976. – 383 с.

21. Основы теории цепей / Г.В.Зевеке, П.А.Ионкин, А.В.Нетушил, С.В.Страхов. – М.:Энергоатомиздат. 1989. – 528 с.

22. Потемкин В.Г. Система инженерных и научных расчетов MATLAB 5.x: – В 2-х т. Том 1. – М.: ДИАЛОГ-МИФИ, 1999. – 366 с.

23. Потемкин В.Г. Система инженерных и научных расчетов MATLAB 5.x: – В 2-х т. Том 2. – М.: ДИАЛОГ-МИФИ, 1999. – 304 с.

24. Потемкин В.Г., Рудаков П.И. Система MATLAB 5 для студентов. – 2-е изд., испр. и дополн. – М.: ДИАЛОГ-МИФИ, 1999. – 448 с.

25. Дьяконов В.П., Абраменкова И.В. MATLAB 5.0/5.3. Система символьной математики. – М.: Нолидж, 1999. –640 с.

26. Лазарев Ю.Ф. MATLAB 5.x. – К.: Издательская группа ВНУ, 2000.– 384 с. (Серия “Библиотека студента”)