

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ВОЛОДИМИРА ДАЛЯ

Методичні рекомендації до лабораторних робіт

з дисципліни

«МОДЕЛЮВАННЯ ЕЛЕКТРОМЕХАНІЧНИХ СИСТЕМ»

Частина I

*(для здобувачів вищої освіти спеціальності 141 «Електроенергетика,
електротехніка та електромеханіка»)*

(Електронне видання)

ЗАТВЕРДЖЕНО

на засіданні кафедри

електричної інженерії

Протокол № 9 від 12.04.2023 р.

Київ – 2023

Методичні вказівки до лабораторних робіт з дисципліни «Моделювання електромеханічних систем» Частина I (для здобувачів вищої освіти спеціальності 141 «Електроенергетика, електротехніка та електромеханіка») // Укладачі: Є.С. Руднєв, Ю.А. Романченко – Київ: Вид-во СНУ ім. В. Даля, 2023. – 55с.

Методичні вказівки призначені для здобувачів вищої освіти спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» та можуть бути корисними для здобувачів вищої освіти інших інженерних спеціальностей.

Методичні вказівки містять рекомендації для виконання лабораторних робіт з дисципліни Моделювання електромеханічних систем з використанням пакету MATLAB (Simulink, Control System Toolbox, SimPowerSistem) за наступними темами: «Знайомство з режимом "прямих обчислень" пакета прикладних програм MATLAB», «Засоби програмування системи MATLAB», «Знайомство з графічними можливостями пакета прикладних програм MATLAB».

Укладачі

Є. С. Руднєв, к.т.н., доц.

Ю. А. Романченко, к.т.н., доц.

Рецензент

М.О. Морнева, к.т.н., доц.

ВСТУП

Пакет MATLAB є універсальним засобом дослідження систем. MATLAB являє собою набір додатків у галузях теорії керування, статистики, економіки, цифрового оброблення інформації та багатьох інших. Потужні засоби програмування, візуалізації та аналізу, вбудовані функції сучасних математичних методів, дають змогу використовувати MATLAB для дослідження систем практично будь-якої складності в будь-якій галузі знань.

Як і будь-який програмний продукт, додаток MATLAB, безперервно розвивається. У кожній новій версії збільшується кількість вбудованих функцій, пакетів розширення, і, як наслідок, сфер застосування додатка. Курс лабораторних робіт побудований на застосуванні пакета MATLAB версії 6.5 (MATLAB 6.5). Однак отримані навички та теоретичні положення можна застосовувати і в інших математичних пакетах (або наступних версіях MATLAB).

Лабораторні роботи спрямовані на формування єдиного сприйняття динамічних систем і вивчення методів їхнього аналізу. Запропоновані об'єкти дослідження орієнтовані на електромеханічні системи. Лабораторні роботи охоплюють такі напрями застосування пакета MATLAB: обчислення та програмування; графічне представлення результатів; структурне моделювання динамічних об'єктів; аналіз властивостей динамічних систем; візуальне моделювання електротехнічних систем.

У системі MATLAB можлива багатоваріантність при реалізації рішень. Наведені в методичних вказівках приклади є одним із варіантів розв'язання запропонованих задач і не виключають ініціативу здобувача вищої освіти до пошуку інших підходів і способів розв'язання поставленої задачі.

Лабораторні роботи побудовані таким чином, що розроблені моделі, блоки і програми аналізу динамічних систем здобувач вищої освіти може застосувати в подальших курсах при виконанні лабораторних робіт, курсових і дипломного проєктів.

ПРАВИЛА ТЕХНІКИ БЕЗПЕКИ ПІД ЧАС ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ

Виконання лабораторних робіт за курсом "Моделювання електромеханічних систем" вимагає засвоєння здобувачами вищої освіти основних навичок роботи на персональному комп'ютері, знання основних команд і правил роботи в операційній системі Windows.

Лабораторні роботи виконуються в спеціалізованих навчальних класах (комп'ютерних класах) за наявності до 25 комп'ютерів, об'єднаних у локальні обчислювальні мережі, з дотриманням основних правил техніки безпеки та пожежної безпеки. Невиконання цих правил може призвести до ураження електричним струмом працюючих, пошкодження дорогого обладнання, втрати важливої інформації, що зберігається на комп'ютерах локальної мережі або на сервері комп'ютерного класу.

Щоб уникнути нещасних випадків, псування обладнання здобувачі зобов'язані виконувати такі вимоги:

1) не торкатися частин обладнання, що з тих чи інших причин мають відкриті корпуси системних блоків і моніторів комп'ютерів;

2) не робити будь-які відключення і ввімкнення електрообладнання робочого місця без дозволу лаборанта або викладача, який проводить заняття, не знімати і не перевишувати попереджувальні плакати і написи;

3) у процесі виконання лабораторних робіт категорично забороняється використовувати неперевірені носії інформації, завантажувати і встановлювати в комп'ютер інше програмне забезпечення, крім використовуваного на цьому комп'ютері;

4) для тимчасового зберігання інформації при виконанні лабораторних робіт виділяється певний обсяг дискового простору на робочій машині або на сервері комп'ютерного класу. Правила доступу до цього обсягу (правила запису шляху в каталозі файлів) вказуються лаборантом навчального класу;

5) категорично забороняється видаляти інформацію з каталогів, які не належать здобувачам; будь-яке видалення інформації має бути погоджено з лаборантом навчального класу;

6) у разі помилкового видалення інформації з комп'ютера необхідно звернутися до лаборанта або викладача, який проводить лабораторні заняття, для надання допомоги у відновленні загубленої інформації. Самостійна спроба відновити видалені файли або каталоги може призвести до безповоротної втрати інформації;

7) не виконувати операцій запису в системні файли, що використовуються операційною системою Windows, не переміщати інформацію в каталогах, які не належать здобувачам, тому що це може призвести до втрати операційною системою працездатності;

8) у разі виникнення підозр у некоректній роботі комп'ютера, конфліктів програмного забезпечення, збоїв при виконанні команд операційної системи, непередбачуваній втраті оперативної інформації, появі попереджувальних повідомлень з боку антивірусного програмного забезпечення, неочікуваного перезавантаження комп'ютера необхідно повідомити про це керівника занять для перевірки комп'ютера та підтвердження його працездатності;

9) відповідальність за дотримання цих правил покладається на здобувачів вищої освіти, які працюють у комп'ютерному класі, а контроль за їх дотриманням ведеться керівником лабораторних робіт.

ЛАБОРАТОРНА РОБОТА №1

ЗНАЙОМСТВО З РЕЖИМОМ "ПРЯМИХ ОБЧИСЛЕНЬ" ПАКЕТА ПРИКЛАДНИХ ПРОГРАМ MATLAB

Мета роботи: навчання основним прийомам роботи з пакетом програм MATLAB (поняття про основні компоненти математичних виразів у MATLAB, типи і формати даних), ведення щоденника сесії, збереження робочої області сесії.

1.1 Теоретичні відомості

1.1.1 Основні відомості про правила роботи в режимі прямих обчислень

Правила роботи:

- для зазначення точки введення вихідних даних у MATLAB використовується символ >>;
- дані вводяться за допомогою найпростішого рядкового редактора;
- для блокування виведення результату обчислень деякого виразу після нього треба встановити знак " ; " (крапка з комою);
- якщо не вказано змінну для значення результату обчислень, то MATLAB призначає таку змінну з ім'ям "ans";
- знаком присвоювання є звичний математикам знак рівності "=", а не комбінований знак ":", як у багатьох інших мовах програмування і математичних системах;
- результат обчислень виводиться в рядках виведення (без знака >>).

1.1.2 Поняття про математичний вираз. Типи та формати даних

Математичні вирази будуються на основі чисел, констант, змінних, операторів, функцій і різних спецзнаків. Нижче даються короткі пояснення суті цих понять.

Число – найпростіший об'єкт мови MATLAB, що представляє кількісні дані. Числа використовуються в загальноприйнятому уявленні про них. Вони можуть бути цілими, дробовими, з фіксованою і плаваючою крапкою. Можливе подання чисел у добре відомому науковому форматі із зазначенням мантиси і порядку числа.

Правила запису чисел в MATLAB:

1) у мантисі чисел ціла частина відокремлюється від дробової не комою, а крапкою, як прийнято в більшості мов програмування;

2) для відокремлення порядку числа від мантиси використовується символ **e**;

3) знак "плюс" у чисел не проставляється, а знак "мінус" у числа називають *унарним мінусом*;

4) пропуски між символами в числах не допускаються.

*Числа можуть бути комплексними: $z = \text{Re}(x) + \text{Im}(x) * i$* . Такі числа містять дійсну $\text{Re}(z)$ і уявну $\text{Im}(z)$ частини. Уявна частина має множник i або j , що означає корінь квадратний із -1 :

$3i$, $2j$, $2+3i$, $-3.141i$, $-123.456+2.7e-3i$.

Константи та системні змінні.

Константа – це заздалегідь визначене числове або символічне значення, представлене унікальним ім'ям.

Інші види констант у MATLAB заведено називати системними змінними, оскільки, з одного боку, їх задає система під час її завантаження, а з іншого – вони можуть перевизначатися. Основні системні змінні, що застосовуються в системі MATLAB, зазначені нижче:

1) **i** або **j** – уявна одиниця (корінь квадратний з -1);

2) **pi** – число $\pi = 3.1415926\dots$;

3) **eps** – похибка операцій над числами з плаваючою крапкою (2^{-52});

4) **realmin** – найменше число з плаваючою крапкою (2^{-1022});

5) **realmax** – найбільше число з плаваючою крапкою (2^{1023});

- 6) **inf** – значення машинної нескінченності;
- 7) **ans** – змінна, що зберігає результат останньої операції і зазвичай викликає його відображення на екрані дисплея;
- 8) **NaN** – вказівка на нечисловий характер даних (Not-a-Number).

Текстові коментарі

Оскільки MATLAB використовується для досить складних обчислень, велике значення має наочність їхнього опису. Вона досягається, зокрема, за допомогою текстових коментарів. Текстові коментарі вводяться за допомогою символу %, наприклад, так:

% Функція обчислення сили намагнічування

1.1.3 Змінні та присвоювання їм значень

Змінні – це об'єкти, що мають імена і здатні зберігати деякі, зазвичай різні за значенням, дані. Залежно від цих даних змінні можуть бути числовими або символічними, векторними або матричними.

У системі MATLAB можна задавати змінним певні значення. Для цього використовується *операція присвоювання*, що вводиться знаком рівності "=".

Типи змінних заздалегідь не декларуються. Вони визначаються виразом, значення якого присвоюється змінній. Так, якщо цей вираз – вектор або матриця, то змінна буде векторною або матричною.

Ім'я змінної (її ідентифікатор) може містити скільки завгодно символів, але запам'ятовується та ідентифікується тільки 31 початковий символ. Ім'я будь-якої змінної не повинно збігатися з іменами інших змінних, функцій і процедур системи, тобто воно має бути унікальним. Ім'я має починатися тільки з літери, може містити літери, цифри та символ підкреслення "_". Також у назвах змінних у MATLAB можна використовувати великі та малі літери. Наприклад Ass і aSs MATLAB сприймає як різні змінні. Неприпустимо включати в імена змінних пропуски та спеціальні знаки, наприклад «+», «-», «*», «/» тощо, оскільки в цьому разі правильна інтерпретація виразів стає неможливою.

1.1.4 Оператори і функції

Оператор – це спеціальне позначення для певної операції над даними – операндами.

Наприклад, найпростішими арифметичними операторами є знаки суми "+", віднімання "-", множення "*" і ділення "/".

Арифметичні оператори є найпоширенішими та найвідомішими. На відміну від більшості мов програмування в системі MATLAB практично всі оператори є матричними, тобто призначені для виконання операцій над матрицями. У табл. 1.1 наводиться список арифметичних операторів і синтаксис їх застосування.

Таблиця 1.1 – Арифметичні оператори

Функція	Назва	Оператор	Синтаксис
Plus	Плюс	+	M1+M2
Uplus	Унарний плюс	+	+M
Minus	Мінус	-	M1-M2
Uminus	Унарний мінус	-	-M
Mtimes	Матричне множення	*	M1*M2
Times	Поелементне множення масивів	.*	A1.*A2
Mpower	Піднесення матриці до степеня	^	M1^x
Power	Поелементне піднесення масиву до степеня	.^	A1.^x
Mldivide	Зворотний (справа наліво) поділ матриць	\	M1\M2
Mrdivide	Поділ матриць зліва направо	/	M1/M2
Ldivide	Поелементний поділ масивів справа наліво	.\	A1.\A2
Rdivide	Поелементний поділ масивів зліва направо	./	A1./A2
Kron	Тензорне множення Кронекера	kron	kron(X,Y)
Ctranspose	Транспонування. У разі якщо елементи матриці A комплексні, то виконується і комплексне сполучення	'	A'
Transpose	Транспонування	.'	A.'

$A-B$ $A+B$

$A*B$ $A.*B$

A/B $A./B$

$A\backslash B$ $A.\backslash B$

A^B $A.^B$

A' $A.'$

Зверніть увагу на те, що кожен оператор має аналогічну за призначенням функцію. Наприклад, оператору матричного множення $*$ відповідає функція `mtimes(M1,M2)`.

Слід зазначити, що більшість операторів належить до матричних операцій, що може бути причиною серйозних непорозумінь. Наприклад, оператори множення $*$ і ділення $/$ обчислюють добуток і частку від ділення двох багатовимірних масивів, векторів або матриць. Є низка спеціальних операторів, наприклад, оператор \backslash означає ділення справа наліво, а оператори $.*$ і $./$ означають відповідно поелементне множення і поелементне ділення масивів.

Слід зазначити, що в математичних виразах оператори мають певний *пріоритет виконання*. Наприклад, у MATLAB пріоритет логічних операторів вищий, ніж арифметичних, пріоритет піднесення до степеня вищий за пріоритети множення і ділення, пріоритет множення і ділення вищий за пріоритет додавання і віднімання. Для зміни пріоритету операцій у математичних виразах використовуються круглі дужки. Ступінь вкладання дужок не обмежується.

Повний список операторів можна отримати, використовуючи команду **«help ops»**.

Роботу наведених у табл. 1.1 операторів можна зрозуміти з таких прикладів

Операція	Результат	Операція	Результат	Операція	Результат	Операція	Результат
x	1 2 3	y	4 5 6	$2 \setminus x$	$1/2$ 1 $3/2$	$2./x$	2 1 $2/3$
x'	1 2 3	y'	4 5 6	x/y	0 0 $1/6$ 0 0 $1/3$ 0 0 $1/2$	x./y	$1/4$ $2/5$ $1/2$
x+y	5 7 9	x-y	-3 -3 -3	x/2	$1/2$ 1 $3/2$	x./2	$1/2$ 1 $3/2$
x + 2	3 4 5	x-2	-1 0 1	x^y	Error	$x.^y$	1 32 729
x * y	Error	x.*y	4 10 18	x^2	Error	$x.^2$	1 4 9
x'*y	32	x'.*y	Error	2^x	Error	$2.^x$	2 4 8
x*y'	4 5 6 8 10 12 12 15 18	x.*y'	Error	$(x+i*y)'$	1 - 4i 2 - 5i 3 - 6i		
x*2	2 4 6	x.*2	2 4 6	$(x+i*y)'$	1 + 4i 2 + 5i 3 + 6i		
x\y	16/7	x.\y	4 $5/2$ 2				

Функції – це об'єкти, що мають унікальні імена, які виконують певні перетворення своїх аргументів і при цьому повертають результати цих перетворень. Повернення результату – відмінна риса функцій. При цьому результат обчислення функції з одним вихідним параметром підставляється на місце її виклику, що дозволяє використовувати функції в математичних виразах, наприклад, функцію \sin в $2*\sin(\pi/2)$.

Функції в загальному випадку мають список аргументів (параметрів), укладений у круглі дужки. Багато функцій допускають низку форм запису, що відрізняються списком параметрів. Якщо функція повертає кілька значень, то вона записується у вигляді $[Y1, Y2, \dots] = \text{func}(X1, X2, \dots)$

де $Y1, Y2, \dots$ – список вихідних параметрів і $X1, X2, \dots$ – список вхідних аргументів (параметрів).

Повний список функцій MATLAB досить широкий. Він охоплює всі розділи математики і налічує близько 3550 функцій. У табл. А1 (Додаток А) наведено список *елементарних* математичних функцій.

Зі списком елементарних функцій можна ознайомитися, виконавши команду **help elfun**, а зі списком спеціальних функцій – за допомогою команди **help specfun**.

Застосування оператора : (двокрапка)

Дуже часто необхідно провести формування впорядкованих числових послідовностей. Такі послідовності потрібні для створення векторів або значень абсциси при побудові графіків. Для цього в MATLAB використовується оператор ":" (двокрапка).

"Початкове_значення" : "Крок" : "Кінцеве_значення"

Дана конструкція породжує зростаючу послідовність чисел, яка починається з початкового значення, йде із заданим кроком і завершується кінцевим значенням. Якщо Крок не задано, то він набуває значення 1.

1.1.5 Формування векторів і матриць

Особливості завдання векторів і матриць

Якщо необхідно задати вектор із трьох елементів, то їхні значення слід перелічити у квадратних дужках, розділяючи пробілами або комами. Так, наприклад, присвоювання

```
>> V=[1 2 3]
```

```
V=
```

1 2 3

задає вектор V , що має три елементи зі значеннями 1, 2 і 3. Після введення вектора система виводить його на екран дисплея.

Слід зауважити, що в системі MATLAB, таким манером вводяться вектор-рядки.

Вектор-стовпець можна задати набравши елементи через « ; » Взагалі знаком « ; » у завданні матриці розділяються рядки

```
>> V=[1;2;3]
```

Задати вектор стовпець можна також із застосуванням оператора транспонування « ' », наприклад:

```
v=[1 2 3]'
```

```
v=
```

```
1
```

```
2
```

```
3
```

Завдання матриці потребує зазначення кількох рядків. Для розмежування рядків використовується знак ";" (крапка з комою). Так, введення

```
>> M=[1 2 3; 4 5 6; 7 8 9];
```

задає квадратну матрицю:

```
M =
```

```
1     2     3
```

```
4     5     6
```

```
7     8     9
```

Можливе введення елементів матриць і векторів у вигляді арифметичних виразів, що містять будь-які доступні системі функції, наприклад:

```
>> V = [2+2/(3+4) exp(5) sqrt(10)]
```

```
V =
```

```
2.2857 148.4132 3.1623
```

Для зазначення окремого елемента вектора або матриці використовуються вирази виду $V(n)$ або $M(n,k)$. Наприклад, якщо задати

```
>> m_32=M(3,2)
```

```
m_32= 8
```

Вираз $M(m)$ з одним індексом дає доступ до елементів матриці, розгорнутих в один стовпець. Така послідовність елементів утворюється, якщо поспіль виписати стовпці розглянутої матриці поспіль.

Можлива вибірка діапазону елементів матриці. Для цього використовують оператор «:». Наприклад, задавши

```
>> R=M(2:3, 1:2)
```

```
получим
```

```
R= 4 5
```

```
7 8
```

Нижче показано, як вибирати за допомогою оператора : рядки, стовпчики та елементи з векторів, матриць і багатовимірних масивів:

- $A(:,j)$ – це j -й стовпець з A ;
- $A(i,:)$ – це i -й рядок з A ;
- $A(:,:)$ – еквівалент двовимірного масиву. Для матриць це аналогічно A ;
- $A(j:k)$ – це $A(j), A(j+1), \dots, A(k)$. Для матриць – вектор-сторка з елементів з j по k за послідовної нумерації елементів матриці A зверху-вниз-зліва-направо;
- $A(:,j:k)$ – це матриця зі стовпців $A(:,j), A(:,j+1), \dots, A(:,k)$;
- $A(:,:,k)$ – це k -а сторінка тривимірного масиву A ;
- $A(k, :, :)$ – це матриця - "зріз" тривимірного масиву A по k -му рядку;
- $A(i, j, :)$ – вектор, виділений із тривимірного масиву A . Вектор включає елементи $A(i, j, 1), A(i, j, 2), A(i, j, 3)$ тощо;
- $A(:)$ – записує всі елементи масиву A у вигляді стовпчика.

Команда `disp` дає змогу "керовано" виводити на екран дані, не розділені порядково знаком `>>`. Цю команду зручно застосовувати для посторінкового виведення на екран елементів довгого вектора. Як аргумент цієї команди вказується вектор, елементи якого виводяться на екран. Наприклад:

`disp(V)` – виведення рядка з усіх елементів вектора V ;
`disp(V(1))` – виведення 1-го елемента вектора V ;
`disp([V(1) V(3)])` – виведення рядка з 1-го і 3-го елементів V .

Видалення стовпців і рядків матриць

Для формування матриць і виконання низки матричних операцій виникає необхідність видалення окремих стовпців і рядків матриці. Для цього використовуються порожні квадратні дужки "[]" та оператор двокрапка. Проробимо це з матрицею M :

```
>> M=[1 2 3; 4 5 6; 7 8 9];
```

Видалимо другий стовпець, використовуючи оператор : (двокрапка):

```
>> M(:,2)=[ ]
```

$M =$

```
1 3
4 6
7 9
```

А тепер, використовуючи оператор : (двокрапка), видалимо другий рядок:

```
>> M(2,:)= [ ]
```

$M =$

```
1 3
7 9
```

Об'єднання малих матриць у велику

Описаний спосіб завдання матриць дає змогу виконати операцію конкатенації – об'єднання малих матриць у велику.

Так, якщо є N матриць A1, A2...AN з однаковою кількістю рядків, то об'єднати їх в одну матрицю, розташовуючи блоки горизонтально можна командою

$$A=[A1,A2,\dots,AN].$$

Ця операція називається горизонтальною конкатенацією.

Вертикальна конкатенація (за умови, що складові блоки-матриці мають однакову кількість стовпців) реалізується аналогічно, шляхом відокремлення блоків оператором ";" замість ",", "

$$A=[A1;A2;\dots;AN]$$

Наприклад

$$A1 = [1\ 2\ 3;7\ 8\ 9]$$

$$A2 = [4\ 5;6\ 7]$$

$$A3=[-5\ -1\ -3]$$

$$A=[A1;A2;A3]$$

$$A = \begin{bmatrix} 1 & 2 & 3 & 7 & 8 & 9 \\ 4 & 5 & 6 & 7 & -5 & -1 \\ -3 & & & & & \end{bmatrix}$$

Операції над матрицями

inv(A) – повертає матрицю, зворотню квадратній матриці A.

У мові MATLAB є дві функції ділення матриць, які не визначені в математиці. При цьому вводяться поняття ділення матриць зліва направо "/" і ділення матриць справа наліво "\". Операція B/A еквівалентна операції

$B \cdot \text{inv}(A)$. Цю операцію зручно використовувати для розв'язання матричного рівняння $X \cdot A = B$. Аналогічно операція $A \setminus B$ рівносильна операції $\text{inv}(A) \cdot B$, яка являє собою розв'язання матричного рівняння $A \cdot X = B$. Однак для розв'язання цього матричного рівняння кращим з погляду мінімізації часу розрахунку і підвищення точності обчислень є використання оператора матричного ділення $x = A \setminus b$. Ця операція використовує метод виключення Гауса без явного формування оберненої матриці.

$\text{det}(A)$ – повертає визначник (детермінант) квадратної матриці X

$\text{rank}(A)$ – повертає ранг матриці (кількість сингулярних чисел)

$\text{eig}(A)$ – повертає вектор власних значень квадратної матриці A

$\text{diag}(A)$ – повертає вектор з елементами головної діагоналі матриці A .

$\text{trace}(A)$ – повертає слід (суму елементів головної діагоналі) матриці A .

$\text{rot90}(A)$ – здійснює поворот матриці A на 90° проти годинникової стрілки.

$\text{rot90}(A, k)$ – здійснює поворот матриці A на величину $90 \cdot k$ градусів, де k – ціле число.

Створити квадратну одиничну діагональну матрицю розміром $n \times n$ можна функцією $\text{eye}(n)$.

Функція $\text{squeeze}(A)$ повертає масив, у якому видалено всі одиничні розмірності

Для визначення розмірів матриць і довжин векторів застосовують 2 команди length і size .

$\text{size}(A)$ – розмір матриці. Для $m \times n$ матриці A повертає двоелементний вектор-рядок, у якому перша складова – число рядків m , а друга складова – число стовпців n .

$\text{length}(A)$ – повертає максимальну розмірність матриці A .

Роботу цих функцій можна зрозуміти з такого прикладу

Команда	Результат	Команда	Результат
$a = [1 \ 2 \ 3; 7 \ 8 \ 9]$	$a =$ <div style="display: flex; justify-content: space-around; width: 100%;"> 1 2 3 </div> <div style="display: flex; justify-content: space-around; width: 100%;"> 7 8 9 </div>	$b = [5 \ 6 \ 4 \ 7]$	$b =$ <div style="display: flex; justify-content: space-around; width: 100%;"> 5 6 4 </div> <div style="display: flex; justify-content: space-around; width: 100%;"> 7 </div>

size(a)	ans = 2 3	size(b)	ans = 1 4
length(a)	ans = 3	length(b)	ans = 4

1.1.6 Операції з робочою областю і текстом сесії

Збереження робочої області сесії

Змінні та визначення нових функцій у системі MATLAB зберігаються в особливій ділянці пам'яті, яку називають робочою ділянкою. MATLAB дає змогу зберігати значення змінних у вигляді бінарних файлів із розширенням .mat Для цього слугує ко-команда save, яку можна використовувати в низці форм:

save fname – записується робоча область усіх змінних у файлі бінарного формату з іменем fname.mat;

save fname X – записує тільки значення змінної X;

save fname X Y Z – записує значення змінних X, Y і Z.

Після цих параметрів можна вказати ключі, що уточнюють формат запису файлів:

-mat – двійковий MAT-формат, використовуваний за замовчуванням;

-ascii - AS СІІ-формат одиничної точності (8 цифр);

-ascii -double – ASCII-формат подвійної точності (16 цифр);

-ascii -double -tabs – формат із роздільником і мітками табуляції;

V4 – запис MAT-файлу у форматі версії MATLAB 4;

-append – додавання в наявний MAT-файл.

Можливе використання слова save і в форматі функції, а не команди, наприклад:

```
save ('fname', 'X', 'Y', ...)
```

У цьому разі імена файлів і змінних задаються строковими константами.

Слід зазначити, що можливості збереження всього тексту сесії, що формується в командному режимі, команда `save` не дає. Для цього служить команда `diary`, описана нижче.

Завантаження робочої області сесії

Для завантаження робочої області раніше проведеної сесії (якщо вона була збережена) можна використовувати команду **load**:

load fname ... – завантаження раніше збережених у файлі `fname.mat` визначень зі специфікаціями на місці три крапки, подібними до описаних для команди `save` (включно з ключем `-mat` для завантаження файлів із розширенням `.mat` звичайного бінарного формату, використовуваного за замовчуванням);

load('fname' ...) – завантаження файлу `fname.mat` у формі функції.

Якщо команда (або функція) `load` використовується під час проведення сесії, то відбудеться заміна поточних значень змінних тими значеннями, які були збережені в МАТ-файлі, що зчитується.

Ведення щоденника сесії

Раніше зазначалося, що сесії не записуються на диск стандартною командою `save`. Однак якщо така необхідність є, можна скористатися спеціальною командою для ведення так званого щоденника сесії:

diary file name – веде запис на диск усіх команд у рядках введення й отриманих результатів у вигляді текстового файлу із зазначеним ім'ям;

diary off – призупиняє запис у файл;

diary on – знову починає запис у файл.

Таким чином, чергуючи команди `diary off` і `diary on`, можна зберігати потрібні фрагменти сесії в їхньому формальному вигляді. Команду `diary` можна задати і у вигляді функції **diary('file')**, де рядок `'file'` задає ім'я файлу.

Завершення роботи із системою

Для завершення роботи із системою можна використовувати команди `exit`, `quit`. Якщо необхідно зберегти значення всіх змінних (векторів, матриць) системи, то перед введенням команди `exit` слід дати команду `save` потрібної форми. Команда `load` після завантаження системи зчитує значення цих змінних і дає змогу розпочати роботу із системою з того моменту, коли вона була перервана.

1.2 Порядок виконання роботи:

1.2.1 Відкрийте файл для ведення щоденника сесії (для цієї лабораторної роботи – основний документ звітності про виконання лабораторної роботи).

Для цього виконайте такі дії:

1) створіть директорию **D:\ml_lr\gr_№групи\br_№бригади**

де – *№групи* набуває значення 1 або 2 відповідно до номера Вашої групи (*це необхідно зробити поза програмою MatLab – у будь-якому файловому менеджері*);

2) запустіть пакет прикладних програм MatLab;

3) після появи у вікні Command символу ">>" наберіть *diary D:\ml_lr\gr_№групи\br_№бригади\ dn1.txt* (тим самим, ви відкриєте *щоденник сесії* – "запис усіх дій під час вашої роботи");

1.2.2 Ознайомтеся з константами MatLab (системні змінні), для чого введіть по черзі такі символи (завершення введення кожного символу завершується натисканням клавіші <Enter>) *i, j, pi, inf*, тощо (див. вище).

1.2.3. Розрахуйте значення функції $y(x)$ при значеннях x та ω зазначених у таблиці 1.2.

1.2.4 Створіть вектор x , що містить послідовність чисел від -1 до 1 з кроком 0.1. Розрахуйте значення функції $y(x)$ з таблиці 1.2, для кожного зі значень даного вектора.

1.2.5 Опануйте роботу з матрицями, виконавши завдання, наведені в таблиці 1.3.

Таблиця 1.2 – Функції до завдання з пункту 1.2.3

№ варіанта (бригади)	Функція
1	$y(x) = \frac{5}{\pi} x^2 \cdot \sin \frac{\omega}{x+2} + \frac{5}{7\pi} \cos(3\omega \cdot x), \quad \omega = 2, \quad x = 3$
2	$y(x) = x \cdot \cos \frac{4\omega}{\pi} + \frac{4}{9\pi} \cos(0.5 + \omega \cdot x^3) \quad \omega = 0.5, \quad x = 1/3$
3	$y(x) = \frac{4}{\pi} \sin^2(\omega \cdot x) - (x + \frac{7}{12\pi}) \cdot \cos 7\omega \quad \omega = 1/2, \quad x = 0.7$
4	$y(x) = \frac{4}{3\pi} \sin(\omega \cdot x^3) - \frac{5x}{\pi} \cos^2 3\omega \quad \omega = 2/7, \quad x = 3$
5	$y(x) = \frac{3}{\pi} \cdot x^2 \cdot \cos 5\omega + \frac{4}{7} \cos(2\omega + \frac{x}{\pi}) \quad \omega = 7/27, \quad x = 4$
6	$y(x) = \pi \cos^3(\omega \cdot x) + \frac{4}{\pi} (x^2 - 1) \cdot \cos 7\omega \quad \omega = 1/5, \quad x = 33$
7	$y(x) = \pi x \cdot \sin \left(\frac{\omega - x}{x^2 + 2} \right) + \frac{4}{\pi} \cos^2(7\omega \cdot x) \quad \omega = 0.8, \quad x = 3.65$
8	$y(x) = \pi \sin(\omega^2 - x^2) - \frac{3x^3}{2\pi} \cdot \cos(3\omega \cdot x) \quad \omega = 2, \quad x = 3/45$
9	$y(x) = 2\pi \sin^3(3\omega \cdot x) + \left(\frac{\omega + x}{\omega^2 + 1} \right) \cos x \quad \omega = 0.23, \quad x = 3/65$
10	$y(x) = \frac{x}{x^2 + 1} \cdot \cos(2\omega - 1) + \frac{\pi}{3} \sin^2(3\omega \cdot x) \quad \omega = 2.2, \quad x = 0.3$

№ варіанта (бригади)	Функція
11	$y(x) = \frac{x + \omega}{\pi} \cdot \sin^2 \omega + \frac{5}{3\pi} \cos(3\omega + x), \quad \omega = 2/15, x = 4.2$
12	$y(x) = (\omega - 2) \cdot \cos \frac{x}{\pi} + \frac{4}{9\pi} \sin^2(\omega \cdot x) \quad \omega = 0.54, x = 1/4$
13	$y(x) = 4\pi x \cos \frac{\omega^2}{\pi} - \frac{7}{12\pi} \cos(7\omega \cdot x + 2) \quad \omega = 1/6, x = 2.7$
14	$y(x) = x \cdot \sin(\omega + \frac{4\pi}{3}) - \frac{5}{\pi} \cos^2(\omega \cdot x^3) \quad \omega = 1/7, x = 4.25$

Таблиця 1.3 – Варіанти завдань

№ бригади	1	2	3	4	5
Скаляр	$x1=33+44$ $z=0.3\pi$	$c=35-77$ $d=0.4\pi$	$d=2.2/55$ $k=j2$	$y=77/4.4$ $d=2.2/\pi$	$z=4/3+3$ $s=i5$
Вектор	$C=2/x, 4, 7$	$X=4, 6-c, 7$	$Y=7, 1, 7-d$	$A=5, 6, y \cdot 8.8$	$Q=4.4, 5, z/44$
Матриця В=	1, 2, 3 $z \cdot C$ 4, 5, $x1-5$	5, $8+d, 9$ $\pi \cdot X$ $5+c, 7, 9 \cdot d$	$d, 5, 77+k$ $88, 6-k, 9$ $Y \cdot d$	$y \cdot 2, 55, 7/8$ 4, $y+5, 6-d$ $A \cdot y$	$z, 55, 88+s$ $z-1, 66, 88 \cdot s$ $Q \cdot z$
Виділіть з отриманої матриці В підматрицю G розмірності 2x2, у якій вирізано (це завдання може бути виконано кількома способами):					
-«>>-	останні рядок і стовпець	перші рядок і стовпець	перший рядок і другий стовпець	третій рядок і другий стовпець	перший рядок і третій стовпець
Збережіть набрані під час сесії змінні, див. вище "збереження робочої області"					
Визначте різницю між операторами "."*" і "**", використовуючи введені змінні. Для чого виконайте операції перемноження між двома скалярами, скаляром і вектором, двома векторами, вектором і матрицею, двома матрицями, використовуючи обидва оператори по черзі.					
№ бригади	6	7	8	9	10
Скаляр	$n=2-46$ $z=2.3\pi$	$c=5+45$ $d=1.4i$	$d=2.2/65$ $k=2\pi$	$y=73/4.4$ $d=5.2/\pi$	$z=7/3+8$ $s=j5$
Вектор	$C=3/n, 5, \pi$	$X=2, 3-c, 1$	$Y=2, 3, 4-d$	$A=3, 3, y \cdot 5.1$	$Q=4.5, 6, z/7$

Матриця В=	4, 5, 3 z·C 1, 2, x-3	5, 9+d, 3 π·X 4+c, 6, 8·d	d, 5, 7+2k 88, 6-k, 9 Y·k	y·2, 35, 7/6 4, y+5, 6-d A·d	z, 54, 38+s z-1, 14, 32·s Q·z
Виділіть з отриманої матриці В підматрицю G розмірності 2x2, у якій вирізано:					
-«»-	перші рядок і стовпець	останні рядок і стовпець	перший рядок і третій стовпець	третій рядок і перший стовпець	перший рядок і другий стовпець
Збережіть набрані під час сесії змінні, див. вище "збереження робочої області"					
Визначте різницю між операторами "." і "/", використовуючи введені змінні. Для чого виконайте операції ділення між двома скалярами, скаляром і вектором, двома векторами, вектором і матрицею, двома матрицями, використовуючи обидва оператори по черзі.					

1.3 Запитання для самоперевірки

1. Яким чином оголошуються змінні в мові MATLAB?
2. Як домогтися того, щоб результат дій, записаних у черговому рядку:
 - а) виводився в командне вікно; б) не виводився на екран?
3. Яку роль відіграє системна змінна ans?
4. Як ввести значення комплексного числа і в якому вигляді воно буде виведено на екран?
5. Як вводяться вектори та матриці в системі MATLAB?
6. Яким чином можна сформувати вектор з арифметичною послідовністю елементів?
7. Як у MATLAB здійснюються звичайні матричні операції?
8. Який результат виконання елементарних математичних функцій, у разі якщо їхнім аргументом є вектор або матриця?

ЛАБОРАТОРНА РОБОТА №2

ЗАСОБИ ПРОГРАМУВАННЯ СИСТЕМИ MATLAB

Мета роботи: навчання основним навичкам досліджень з використанням програмних засобів пакета MATLAB.

2.1 Теоретична підготовка

MATLAB підтримує завдання послідовності дій за програмою, записаною у вигляді m-файлу. Для створення m-файлів може використовуватися як вбудований редактор, так і будь-який текстовий редактор, що підтримує формат ASCII. Підготовлений і записаний на диск m-файл стає частиною системи, і його можна викликати як з командного рядка, так і з іншого m-файлу.

Є два типи m-файлів: файли-сценарії та файли-функції. Важливо, що в процесі свого створення вони проходять синтаксичний контроль за допомогою вбудованого в систему MATLAB редактора/налагоджувача m-файлів.

2.1.1 Структура та властивості файлів сценаріїв

Файл-сценарій, іменований також Script-файлом, є просто записом серії команд без вхідних і вихідних параметрів. Він має таку структуру:

%Основний коментар

%Додатковий коментар Тіло файлу з будь-якими виразами.

Важливими є такі властивості файлів-сценаріїв:

- вони не мають вхідних і вихідних аргументів;
- працюють із даними з робочої області;
- у процесі виконання не компілюються;
- являють собою зафіксовану у вигляді файлу послідовність операцій, повністю аналогічну тій, що використовується в сесії.
- основним коментарем є перший рядок текстових коментарів;

– додатковим коментарем є наступні рядки.

Основний коментар виводиться під час виконання команд `lookfor` і `help ім'я_каталогу`. Повний коментар виводиться під час виконання команди `help ім'я_файлу`.

Методичний приклад №1

Розглянемо наступний файл-сценарій *calculationdata.m* (пояснення до прикладу дано в коментарях самого файлу):

```
% Розрахунок значень функції y=sin(alpha*t)
% Значення alpha має бути задано в робочій області
% до виклику скрипта на виконання
% t=[0:0.05:1];
y=sin(alpha*t);
% виведення векторів t і y в командному вікні
t',y'
```

2.1.2 Керуючі структури

Крім програм з *лінійною структурою*, інструкції яких виконуються строго по порядку, існує безліч програм, структура яких *нелінійна*. При цьому гілки програм можуть виконуватися залежно від певних умов, іноді з кінцевим числом повторень – циклів, іноді у вигляді циклів, що завершуються при виконанні заданої умови. Для створення таких програм необхідні спеціальні керуючі структури. Вони є в будь-якій мові програмування, і зокрема, в MATLAB.

Умовний оператор

Варіанти використання умовного оператора `if` у загальному вигляді описуються однією з таких структур:

if Умова	if Умова	if Умова_1
Інструкції	Інструкції_1	Інструкції_1
end	else	elseif Умова_2
	Інструкції_2	Інструкції_2
	end	else
		Інструкції_3
		end

Робота цих конструкцій очевидна – при виконанні умови виконуються відповідні інструкції.

Умови записуються у вигляді:

Вираз_1 Оператор_відношення Вираз_2,

причому в якості Операторів_відношення використовуються такі оператори:

< менше;
 > більше;
 <= менше або дорівнює;
 >= більше або дорівнює;
 == дорівнює;
 ~= не дорівнює.

Усі ці оператори являють собою пари символів без пробілів між ними. Слід зазначити, що оператори <, <=, >, >= при комплексних операндах використовують для порівняння тільки дійсні частини операндів - уявні відкидаються. У той же час водночас оператори == і ~= ведуть порівняння з урахуванням як дійсної, так і уявної частин операндів. У загальному випадку оператори відношення порівнюють два масиви одного розміру і видають результат у вигляді масиву того самого розміру.

Умова може бути складеною, тобто складатися з кількох простих умов, об'єднаних знаками логічних операцій:

& операція І (AND)
| операція АБО (OR)
~ операція НЕ (NOT)

Цикли типу for...end

Цикли типу for...end зазвичай використовуються для організації обчислень із заданим числом повторюваних циклів. Синтаксис конструкцій такого циклу має такий вигляд:

```
for var=Вираз Інструкція1; ... Інструкція2, end
```

або

```
for var=Вираз Інструкція1; ... Інструкція2; end
```

або

```
for var=Вираз
```

```
Інструкції
```

```
end
```

Вираз найчастіше записується у вигляді s:d:e, де s – початкове значення змінної циклу var, d - приріст цієї змінної та e - кінцеве значення керівної змінної, при досягненні якого цикл завершується. Можливий і запис у вигляді s :e (у цьому разі d=1). Список виконуваних у циклі інструкцій завершується оператором end.

Наступні приклади пояснюють застосування циклу для отримання квадратів значень змінної циклу:

```
» for i=1:5 i^2, end;
```

```
» for X=0:.25:1 X^ 2, end
```

Оператор continue передає керування в наступну ітерацію циклу, пропускаючи оператори, які записані за ним, причому у вкладеному циклі він передає керування на наступну ітерацію основного циклу. Оператор

`break` може використовуватися для дострокового переривання виконання циклу. Як тільки він зустрічається в програмі, цикл переривається.

Можливі вкладені цикли, наприклад:

```
for i=1:3
for j=1:3
A(i,j)=i+j;
end
end
```

Слід зазначити, що робота з матрицями за допомогою вкладених циклів займає більше часу. Крім того, MATLAB уже містить практично всі функції обробки даних у матрицях. Однак застосування циклів нерідко виявляється більш наочним і зрозумілим.

Цикли типу while...end

Цикл типу `while` виконується доти, доки виконується Умова: `while`
Умова Інструкції `end`

Дострокове завершення циклів реалізується за допомогою операторів `break` або `continue`.

Також у MATLAB є можливість використовувати функцію перемикача – конструкція `switch – case`. Ознайомитися з цією конструкцією рекомендується самостійно.

2.1.3 Структура М-файлу-функції

М-файл-функція є типовим об'єктом мови програмування системи MATLAB. Водночас він є повноцінним модулем з точки зору структурного програмування, оскільки містить вхідні та вихідні параметри і використовує апарат локальних змінних. Структура такого модуля з одним вихідним параметром має такий вигляд:

```
function var=f_name(Список_параметрів)
%Основний коментар
```

```
%Додатковий коментар  
Тіло файлу з будь-якими виразами  
var=вираз
```

М-файл-функція має такі властивості:

– він починається з оголошення `function`, після якого вказується ім'я змінної `var` – вихідного параметра, ім'я самої функції та список її вхідних параметрів;

– функція повертає своє значення і може використовуватися у вигляді `name (Список_параметрів)` у математичних виразах;

– усі змінні, наявні в тілі файлу-функції, є локальними, тобто діють тільки в межах тіла функції;

– файл-функція є самостійним програмним модулем, який спілкується з іншими модулями через свої вхідні та вихідні параметри;

– правила виведення коментарів ті самі, що у файлів-сценаріїв;

– файл-функція слугує засобом розширення системи MATLAB;

– у разі виявлення файлу-функції його компілюють і потім виконують, а створені машинні коди зберігають у робочій області системи MATLAB;

– остання конструкція `var=вираз` вводиться, якщо потрібно, щоб функція повертала результат обчислень.

Наведена форма файлу-функції характерна для функції з одним вихідним параметром. Якщо вихідних параметрів більше, то вони вказуються у квадратних дужках після слова `function`. При цьому структура модуля має такий вигляд:

```
function [var1,var2 ] =f_name(Список_параметрів)  
%Основний коментар  
%Додатковий коментар  
Тіло файлу з будь-якими виразами  
var1=вираз  
var2=вираз
```

Така функція багато в чому нагадує процедуру. Її не можна сліпо використовувати безпосередньо в математичних виразах, оскільки вона повертає не єдиний результат, а безліч результатів - за кількістю вихідних параметрів. Якщо функція використовується як така, що має єдиний вихідний параметр, але має низку вихідних параметрів, то для повернення значення буде використовуватися перший з них. Це часто призводить до помилок у математичних обчисленнях. Тому, як зазначалося, ця функція використовується як окремий елемент програм виду:

```
[var1,var2, ] =f_name(Список_параметрів)
```

Методичний приклад №2

Створення файлу-функції *kolo.m*, який як вхідний параметр приймає значення радіуса кола, а як вихідні параметри видає довжину кола та площу кола, нею обмеженого:

```
function [dlina,ploschad]=kolo(r)
% Розрахунок довжини кола та площі кола
% Вхідним параметром є радіус
p=2*pi*r;
s=(r^2)*pi/2;
dlina=p;
ploschad=s;
```

Далі наведено приклад використання функції *kolo*:

```
>> x=5
x =
    5
>> [dl,pl]=kolo(x)
dl =
    31.4159
```

$p1 =$
39.2699

Якщо функція використовує велику кількість вхідних параметрів, то передати їх у функцію можна не як аргументи, а оголосивши їх як глобальні змінні. Значення глобальних змінних будуть визначені в зовнішньому описі та сприйняті функцією. Для визначення змінної як глобальної використовується команда `global`. Розібратися з роботою і синтаксисом цієї команди пропонується самостійно, набравши в командному вікні `help global`.

2.2 Порядок виконання роботи

2.2.1 Створіть файл-функцію, що реалізує розрахунок складної функції. Вхідним аргументом є значення x , а вихідним – вектор, перший елемент якого це значення функції (y), а другий – номер умови, за якою здійснювався вибір конкретної підфункції. Складові функції вибрати з табл. 2.1 відповідно до варіанта.

Створіть вектор значень аргументу x (10-13 значень), щоб він містив значення, які потрапляють у кожен діапазон, заданий умовами аргументу складової функції.

Створіть скрипт (m-файл сценарій), у якому використовується створена функція. Використовуючи оператори циклу отримайте вектор значень функції y , що відповідають елементам вектора x , і вектор nf номерів діапазону, що відповідають заданим аргументам.

Виведіть у командному вікні вектори x , y , nf , а також виведіть їх у вигляді таблиці з шапкою:

Argument	y	Nom fcn
----------	-----	---------

Виведення рядків у таблиці можна організувати, використовуючи в циклі команду `disp`.

Таблиця 2.1 – Складові функції

№ варіанта	Функція
1	$y = \begin{cases} 1,2 \cdot \sin(\arccos(1/x)) \cdot \frac{ch(\sqrt{3-x})}{e^{-0.3x^2+1}}, & \text{при } -\infty < x < -4 \\ \ln(3 \cdot \cos(0.6x^5)) + x \cdot \log^2 x, & \text{при } -4 \leq x < 2 \\ \frac{\sqrt[3]{3 \cdot x^4 - 1.9 \cdot x}}{x-1}, & \text{при } 2 \leq x < \infty \end{cases}$
2	$y = \begin{cases} sh^4(x/2) - \frac{\sin(0.5 \cdot \pi x)}{\cos(\pi \cdot x) + 1.2}, & \text{при } -\infty < x \leq -\pi/2 \\ \sqrt{ 1.5 \cdot x^3 + 0.4 } - 2.3 \cdot \operatorname{arctg}(x), & \text{при } -\pi/2 < x < \pi/2 \\ 4 \lg \frac{x^2}{x+1} + e^{-0.17 \cdot x}, & \text{при } \pi/2 \leq x < \infty \end{cases}$
3	$y = \begin{cases} x^2 \cdot \sin(0.3 \cdot x^3 + 2.1 \cdot x) + \cos(2/x), & \text{при } -\infty < x < 0 \\ 2 \cdot \sqrt{x} + ch \frac{e^{4x}}{x+0.5}, & \text{при } 0 \leq x < 3.5 \\ \sin(2.8 \cdot \ln(3^{\cos(-0.15 \cdot x)})), & \text{при } 3.5 \leq x < \infty \end{cases}$
4	$y = \begin{cases} \frac{\sqrt{3-x}}{e^{\sin(x)-2}} + th \frac{x}{x^2+1}, & \text{при } -\infty < x < -1 \\ \ln^3(\cos(x^4) + 2), & \text{при } -1 \leq x < 2 \\ \frac{ctg(x) - \sin(2 \cdot x - 1) }{(x-1) \cdot \sqrt{3x}}, & \text{при } 2 \leq x < \infty \end{cases}$
5	$y = \begin{cases} \exp(0.2 \cdot x^3) - \cos^2(\sqrt{ \sin(0.3 \cdot x) - x}), & \text{при } -\infty < x \leq -1 \\ \frac{2 \cdot (x+1) \cdot x^3 + \sin(0.7 \cdot \pi \cdot x + 2)}{x^2 + 1}, & \text{при } -1 < x \leq 0 \\ ch(x^{0.5} + 3x) \cdot \ln(\cos(x/\pi) + 1.2), & \text{при } 0 < x < \infty \end{cases}$

6	$y = \begin{cases} \frac{(x+3) \cdot e^{2.8x}}{\cos(3\pi \cdot x + \pi/4) + 1.1}, & \text{npu } -\infty < x < -2 \\ 5.1 \cdot x^2 \cdot \lg(\cos(\text{th}(\sqrt{x+3})) + 0.8), & \text{npu } -2 \leq x \leq 2 \\ \frac{1.7^{(3x-0.2)}}{(3.5 \cdot x^2 + 2)\sqrt[3]{\sin(x-2)+1}}, & \text{npu } 2 < x < \infty \end{cases}$
7	$y = \begin{cases} 2 \cdot x^3 \cdot \cos(x \cdot e^{(0.24 \cdot x + 1)}) \cdot \sqrt{1-x}, & \text{npu } -\infty < x < 0 \\ \frac{\sin(\pi \cdot x/3) \cdot \text{sh}^2(\sqrt{x+1})}{ x^2 - 2.6 \cdot x + 3 \cdot \ln(2 \cdot x + 0.3)}, & \text{npu } 0 \leq x < 3 \\ 1.6 \cdot \sin(2 \cdot \lg(x^{-3} + e^{-3.4x})), & \text{npu } 3 \leq x < \infty \end{cases}$
8	$y = \begin{cases} \frac{x^3 - ch^2(1/x) - sh(3/x) }{0.8 \cdot \cos(2.3 \cdot x + \pi) + 1}, & \text{npu } -\infty < x < 0 \\ 5 \cdot \cos(\lg(x + \sqrt{x+1})) \cdot \text{arctg}(e^{-3x}), & \text{npu } 0 \leq x < 2.5 \\ \frac{x^2 + 4}{(x+2) \cdot \ln(x+1)} + \frac{5.1 \cdot \sin^3(\pi \cdot x)}{e^{(0.3 \cdot x^2 - 2)}}, & \text{npu } 2.5 \leq x < \infty \end{cases}$
9	$y = \begin{cases} 0.7 \cos(x^2 + 2) + e^{0.8x} \cdot \lg^2(2.1 - x), & \text{npu } -\infty < x < -2 \\ \text{sh} \frac{\cos(2x) \cdot \sqrt{x^2 + 0.5}}{ 3 \cdot x^{-2} + 0.7 \cdot x + 2 }, & \text{npu } -2 \leq x < 1.5 \\ 0.3 \cdot \ln(\cos(\arcsin(1/x^2) + 1)), & \text{npu } 1.5 \leq x < \infty \end{cases}$
10	$y = \begin{cases} \exp\left(\frac{x}{\sqrt{x^2+1}}\right) \cdot \sin^2(\pi \cdot x/2 + 0.7), & \text{npu } -\infty < x < -1 \\ \cos(\arcsin(0.2x) + \lg(3-x)), & \text{npu } -1 \leq x < 2 \\ \frac{\sqrt{x^2+7} \cdot (3x-1)}{\sqrt[3]{\sin(x^2-1) + 2\cos(x)}}, & \text{npu } 2 \leq x < \infty \end{cases}$
11	$y = \begin{cases} \text{sh}^3(3x+7) + ch(\arccos(x^{-2}) - x), & \text{npu } -\infty < x \leq -1 \\ \frac{\text{tg}(0.5 \cdot x) \cdot e^{-\sin(x^3+1)}}{(x^2+2) \cdot \sqrt[3]{x-4}}, & \text{npu } -1 < x \leq 2 \\ \cos(2x + \lg(4x+2)), & \text{npu } 2 < x < \infty \end{cases}$

12	$y = \begin{cases} \sin\left(\arccos\left(\frac{1}{x^2+1}\right) + \lg 3x \right), & \text{npu } -\infty < x < 1 \\ ch^2(\ln x) + 1.4e^{(-x^2+4)}, & \text{npu } 1 \leq x \leq 4 \\ \frac{\sqrt{x^2+2} \cdot \sqrt[3]{x^2-3x}}{ \cos(\pi x + \pi/4) + 0.7}, & \text{npu } 4 < x < \infty \end{cases}$
13	$y = \begin{cases} \cos(x \cdot e^{\sin(x^3+7)} + \sin^2(\arctg(x))), & \text{npu } -\infty < x < 0 \\ \frac{ch(x^2+4) \cdot (x+2)^2}{\sqrt[3]{x-2}}, & \text{npu } 0 \leq x < 1.5 \\ \ln(sh(3x^2-2) + 1), & \text{npu } 1.5 \leq x < \infty \end{cases}$
14	$y = \begin{cases} 0,7 \cdot \cos(\arcsin(x^{-1})) \cdot \frac{ch(\sqrt{3+x^2})}{-0.3x^2+1}, & \text{npu } -\infty < x < -3 \\ \ln(\sin(0.23x^3) + 2) + x \cdot \log^2 x, & \text{npu } -3 \leq x < 1.5 \\ \frac{\sqrt[3]{3 \cdot x^3 - sh(x)}}{4x-1}, & \text{npu } 1.5 \leq x < \infty \end{cases}$
15	$y = \begin{cases} \frac{\sqrt{4-x^3}}{(x+1)} + th \frac{x}{x^2+1}, & \text{npu } -\infty < x < -1 \\ \lg^2(\cos(x^2)+2), & \text{npu } -1 \leq x < 2 \\ \frac{arcctg(x) - sh(2x-1) }{e^{\sin(x)-2} \cdot \sqrt{3x}}, & \text{npu } 2 \leq x < \infty \end{cases}$
16	$y = \begin{cases} \sin\left(\sqrt{ \cos(0.7x+5) } - x\right) + e^{0.4x^3}, & \text{npu } -\infty < x \leq -1 \\ \frac{(x+1) \cdot x^3 + 3.2\sin^2(x+2)}{x^2+2x+1}, & \text{npu } -1 < x \leq 0 \\ ch(3x+1) \cdot \lg(\sin(x+\pi)+2), & \text{npu } 0 < x < \infty \end{cases}$
17	$y = \begin{cases} \frac{(x^2+1) \cdot e^{2.8x}}{\sin^3(3x+\pi) + 2.8}, & \text{npu } -\infty < x < -2 \\ 1.1 \cdot \ln(\cos(\sqrt{x+3}) + 1.7), & \text{npu } -2 \leq x \leq 2 \\ \frac{sh(3x-0.2)}{(x^3+2) \cdot \sqrt[3]{\cos(4x+1)+2}}, & \text{npu } 2 < x < \infty \end{cases}$

18	$y = \begin{cases} 2 \cdot \sin(x^2 \cdot e^{(0.4 \cdot x + 1)} + \sqrt{1 - x^3}), & \text{npu } -\infty < x < 0 \\ \frac{ch(\sqrt{x} + 1) \cdot \cos^2(\pi \cdot x + \pi / 5)}{ x^2 + 1.6 \cdot x + 2 }, & \text{npu } 0 \leq x < 3 \\ 1.6 \cdot \cos(4 \cdot \ln(x^{-3} + e^{-3.4 \cdot x})), & \text{npu } 3 \leq x < \infty \end{cases}$
19	$y = \begin{cases} \frac{ x^3 - sh(1/x) - ch^2(3/x) }{0.8 \cdot \cos(x + \pi / 2) + 1}, & \text{npu } -\infty < x < 0 \\ \frac{\sin(\lg(\sqrt{x} + 1) + \pi / 5) + arctg(e^{-3 \cdot x})}{x^2 + 4}, & \text{npu } 0 \leq x < 3 \\ \frac{x^2 + 4}{\sqrt[3]{x - \sqrt{\lg(x - 1)}}}, & \text{npu } 3 \leq x < \infty \end{cases}$
20	$y = \begin{cases} 0.7 \cos(\sqrt{x^2 + 2}) + e^{0.8x} \cdot \lg^2(2.1 - x), & \text{npu } -\infty < x < -2 \\ sh \frac{2.3 \cdot \sin^2(\pi \cdot x)}{e^{(0.3 \cdot x^2 - 2)}}, & \text{npu } -2 \leq x < 1.5 \\ 0.3 \cdot \lg(\arcsin(x^{-2}) + \sqrt[3]{x - 5}), & \text{npu } 1.5 \leq x < \infty \end{cases}$
21	$y = \begin{cases} \frac{\sqrt{x^2 + 7} \cdot (3x - 1)}{\sqrt[3]{\sin(x^2 - 1) + 2 \cos(x)}}, & \text{npu } -\infty < x < -1 \\ \cos(\arcsin(0.3x) + \lg^2(4 - x)), & \text{npu } -1 \leq x < 2 \\ \exp\left(\frac{2x - 1}{\sqrt{x^3 + 1}}\right) \cdot \cos(x / 2 + 1.7), & \text{npu } 2 \leq x < \infty \end{cases}$
22	$y = \begin{cases} ch^3(3x + 7) \cdot sh(\arcsin(x^{-3}) - x), & \text{npu } -\infty < x < -1 \\ \frac{tg(0.3x) + e^{\cos(1 - x^2)}}{\sqrt{x^2 + 2} \cdot \sqrt[3]{x - 3}}, & \text{npu } -1 \leq x < 2 \\ 2 \sin(x^2 + \ln(4x - 3)), & \text{npu } 2 \leq x < \infty \end{cases}$
23	$y = \begin{cases} \frac{\sqrt{x^2 + 2} \cdot \sqrt[3]{x^2 - 3x}}{ \cos(\pi x + \pi / 4) + 0.7}, & \text{npu } -\infty < x < 1 \\ 1.2 \cdot ch^2(\ln x) + e^{(x^2 - 3)}, & \text{npu } 1 \leq x \leq 4 \\ \cos\left(\arcsin\left(\frac{x + 1}{x^2 - 1}\right) + 4 \cdot \ln(2x - 5)\right), & \text{npu } 4 < x < \infty \end{cases}$

24	$y = \begin{cases} \sin(x \cdot e^{sh(x^3+7)}) \cdot \cos^2(\arctg(x+1)), & \text{при } -\infty < x < 0 \\ \frac{sh(x^2 + 3x + 1) + \sqrt{x+2}}{\sqrt[3]{x-2}}, & \text{при } 0 \leq x < 2 \\ \lg(ch(x^2 - 3) + 2), & \text{при } 2 \leq x < \infty \end{cases}$
25	$y = \begin{cases} ch\left(\frac{x}{x^2+1}\right) - \frac{\sin^2(0.4 \cdot x)}{\cos(2x + \pi/4) + 1.2}, & \text{при } -\infty < x \leq -1 \\ \sqrt{ 1.2 \cdot x^2 - 4 } - 2 \cdot \arctg(0.3x), & \text{при } -1 < x < 3 \\ 2 \ln \frac{x^2}{x+1} + e^{-0.7 \cdot x^2 + 2}, & \text{при } 3 \leq x < \infty \end{cases}$

2.2.2 Створіть файл-функцію, що дає змогу визначити вектор вихідних значень функції $y(x) = 10 + 5e^{-x} \cos(\omega \cdot x + 0.5)$. Вхідними параметрами цієї функції мають бути значення частоти ω , початкове і кінцеве значення діапазону зміни x . Крок зміни змінної x має дорівнювати одній сотій діапазону її зміни.

2.2.3 Розробіть скрипт, який би використовував створену функцію, і задав змогу побудувати в одних осях $y(x)$ для трьох графіки значень ω (див. таблицю).

№ вар-та	ω рад/с	№ вар-та	ω рад/с
1, 25, 15	1, 5, 9	6, 20	5, 15, 20
2, 24	6, 8, 10	7, 19, 12	12, 5, 3
3, 23, 14	10, 3, 6	8, 18	3, 10, 15
4, 22	15, 5, 2	9, 17, 11	5, 10, 15
5, 21, 13	1, 3, 7	10, 16	4, 8, 16

Діапазон зміни вхідної змінної $0 \leq x \leq 5$ с. Графічне зображення повинно мати такі атрибути:

– заголовок* $y(x) = 10 + 5 \exp(-x) \cos(\omega \cdot x + 0.5)$

- позначення осі x time (с)
- позначення осі y $y(x)$
- типи ліній суцільна, штрихова, пунктирна
- сітка нанесена.

Збережіть отримані графіки.

*Вивести символ ω можна за допомогою такої символічної послідовності:
'\omega'.

2.2.4 Створіть текстовий файл, який би містив тему та мету роботи, лістинги створених m-файлів, отримані графіки.

2.3 Запитання для самоперевірки

1. Які є типи m-файлів?
2. Назвіть властивості файлів-сценаріїв.
3. Які оператори використовуються в якості Операторів_відношення?
4. Назвіть властивості m -файлу-функції.

ЛАБОРАТОРНА РОБОТА №3

ЗНАЙОМСТВО З ГРАФІЧНИМИ МОЖЛИВОСТЯМИ ПАКЕТА ПРИКЛАДНИХ ПРОГРАМ MATLAB

Мета роботи: навчання основним прийомам роботи з графікою в пакеті програм MATLAB.

3.1 Теоретичні відомості

Графіка є потужним *інтерактивним* засобом аналізу і синтезу систем управління. Повне розв'язання завдань аналізу та синтезу часто потребує детального розгляду безлічі даних у найрізноманітніших форматах і поданнях. MATLAB надає користувачеві широкі можливості для графічного опрацювання аналізованих даних.

На відміну від текстової інформації, що виводиться системою MATLAB у відповідь на запити користувача безпосередньо в командне вікно сесії, для подання графічних зображень MATLAB використовує спеціальне *графічне вікно (FIGURE)*. В одному сеансі роботи з MATLAB може бути створено кілька графічних вікон (figure 1, figure 2, ..., figure n) і, крім того, кожне з цих вікон може бути розбите на кілька графічних областей зі своєю окремою системою координат. Користувачеві надаються можливості з виведення графіків у будь-яке з відкритих графічних вікон або в будь-яку підобласть обраного вікна, з очищенням попередньої графічної інформації або накладенням декількох графіків один на інший. Під час виведення графіків користувач може задавати колір лінії, тип лінії і маркерних точок, створювати підписи координатних осей, заголовки графіків, розміщувати пояснювальні написи на поточному графіку, виводити або прибирати сітку тощо.

Найпростішою формою подання інформації в графічному вигляді є побудова двовимірного графіка. Існують три основні групи графічних функцій, що надаються системою MATLAB для цих цілей. Перша група призначена для активації та керування графічними вікнами, друга визначає формат графіка і

викликає процедуру побудови його побудови та третя - оформлення графічної інформації.

3.1.1 Функції активації та керування графічними вікнами

Для створення графічного вікна використовується функція **figure**. Якщо одне або кілька графічних вікон уже створено, то команда **figure** створить нове вікно з наступним за порядком зростання номером.

З огляду на те, що система MATLAB дає змогу працювати з кількома графічними вікнами, важливим є поняття *активного або поточного графічного вікна*, тобто вікна, готового в поточний момент до виведення графічної інформації. Для завдання поточного вікна виведення серед відкритих графічних вікон слід виконати команду **figure (n)**, де n номер вікна, що активується. У разі якщо серед відкритих вікон немає графічного вікна з даним номером, то воно буде створено, а потім активовано.

*Слід зауважити, що якщо немає відкритих графічних вікон, то перше графічне вікно автоматично буде створено й активізовано під час виконання будь-якої функції, результатом якої має бути деякий графік (наприклад, функції **plot**, детально описаної далі).*

У вихідному стані під час виведення нового графіка в активне вікно MATLAB очищає раніше виведену в це вікно графічну інформацію. Для побудови нових залежностей без знищення попередніх необхідно виконати команду **hold on**, а потім уже задати команду побудови графіка. При цьому якщо необхідно MATLAB автоматично підлаштує масштаби по осях координат. Повернутися до початкового порядку побудови можна командою **hold off**. Не виходячи з режиму **hold on** графічне зображення можна видалити з вікна за допомогою функції **clf**.

Якщо поточне графічне вікно перебуває у згорнутому стані, то побудова деякої залежності відбувається без його розгортання. Виконання функції **shg** командному рядку дає змогу розгорнути і вивести активне графічне вікно на перший план.

Графічне вікно можна розділити на кілька графічних областей меншого розміру. Функція **subplot (m, n, p)** ділить поточне вікно на **m x n** ділянок (**m** – кількість "рядків", **n** – кількість "стовпців"), а параметр **p** цієї функції визначає активну графічну ділянку. Нумерація йде зліва направо і зверху вниз. Параметр **p** може бути скаляром або вектором. Використання як параметра **p** вектора дає змогу об'єднувати суміжні графічні області в одну. Елементи цього вектора в цій ситуації мають містити номери об'єднаних графічних областей, що об'єднуються.

3.1.2 Функції побудови двовимірних графіків

Для побудови двовимірних графіків у середовищі MATLAB служать функції, наведені в таблиці 3.1. Синтаксис, наведених у таблиці 3.1 функцій, подібний до синтаксису функції **plot**, яка і буде розглянута більш детально.

Таблиця 3.1 – Функції побудови двовимірних графіків

Функція	Призначення
plot	побудова графіка в декартовій системі координат
loglog	побудова графіка з логарифмічними осями координат
semilogx	побудова графіка з логарифмічною віссю абсцис
semilogy	побудова графіка з логарифмічною віссю ординат
polar	побудова графіка в полярній системі координат
plotyy	побудова графіка в декартовій системі координат із виведенням осей праворуч і ліворуч від вікна графіка

Функція **plot** має різні форми запису залежно від вхідних аргументів:

- 1) **plot(y)** – побудова графіка залежності значення елемента вектора **y** від його порядкового номера;
- 2) **plot(x,y)** – побудова графіка функції **y=f(x)**. Тут **x** і **y** вектори, що мають однакову розмірність;

3) **plot(x1,y1,x2,y2,...,xn,yn)** – побудова графіків функції $y_1=f(x_1)$, $y_2=f(x_2)$, ... , $y_n=f(x_n)$. MATLAB автоматично виводить залежності різними кольорами, відповідно до заздалегідь визначеного списку кольорів;

4) **plot(x,y,'color_style_marker')** – побудова графіка функції $y=f(x)$ з визначеним користувачем видом графіка. Тут **color_style_marker** – 1, 2 або 3-значна символна змінна (укладена в одинарні лапки), яка утворюється зі знаків, що визначають відповідно колір, тип лінії та маркер:

– символи, що відносяться до кольору: 'c', 'm', 'y', 'r', 'g', 'b', 'w' і 'k'. Вони позначають блакитний, малиновий, жовтий, червоний, зелений, синій, білий і чорний кольори відповідно;

– символи, що належать до типу ліній:

'-' – для суцільної,

'--' – для розривної (широкий пунктир),

':' – для пунктирної,

'-.' – для штрих-пунктирної лінії;

'none' – для її відсутності.

– найчастіше зустрічаються маркери '+', 'o', '*' і 'x'. Повний перелік типів маркерів наведено в додатку А (табл. А.2)

Наприклад, вираз:

plot(x,y,'y:+')

будує **жовтий пунктирний** графік і поміщає маркери '+' у кожну точку даних. Якщо визначено тільки тип маркера, але не визначено тип стилю ліній, то MATLAB виведе тільки маркери.

Якщо аргумент функції **plot** комплексне число, то уявна частина ігнорується, за винятком випадку, коли комплексний аргумент один. Для цього спеціального випадку відбувається побудова графіка залежності реальної частини аргументу від уявної. Тому застосування функції **plot(Z)**, де **Z** – комплексний вектор, еквівалентне застосуванню команди **plot(real(Z), imag(Z))**.

Наведені вище приклади показують спрощене використання команди **plot**.
У загальному ж вигляді синтаксис цієї команди має такий вигляд

plot(x,y,'PropertyName',PropertyValue,...)

де **PropertyName** – ім'я властивості лінії, **PropertyValue** – значення цієї властивості.

Загалом є можливість задати близько 25 властивостей лінії: колір лінії; тип лінії; товщину лінії; тип, розмір, колір, заливку маркерів на лінії, створення тіні та багато іншого. У таблиці 3.2 наведено основні властивості ліній.

Таблиця 3.2 – Властивості ліній

Властивість	Опис	Значення
LineStyle	Тип лінії	-, --, :, -., none. За замовчуванням: -
LineWidth	Товщина лінії	Число. За замовчуванням: 0,5
Marker	Тип маркера	Знак, см. табл. А.3 За замовчуванням: none
MarkerEdgeColor	Колір контуру маркера	Колір, none, auto За замовчуванням: auto (як і колір лінії)
MarkerFaceColor	Колір заливки маркера	Колір, none, auto За замовчуванням: none
MarkerSize	Размер маркера	За замовчуванням: 6

Наприклад, команда

```
plot(x,y,'--rs','LineWidth',2,...
      'MarkerEdgeColor','k',...
      'MarkerFaceColor','g',...
      'MarkerSize',10)
```

виводить функцію $y(x)$ червоною пунктирною лінією завтовшки 2пт із маркерами у вигляді квадратів чорного кольору із зеленою заливкою. Розмір маркерів – 10пт.

3.1.3 Функції управління оформленням графіків

Графічні можливості MATLAB досить широкі. Під час виведення графічної інформації можна задати всі параметри осей і виду фігури та отримати бажаний вигляд графіка. Наприклад, MATLAB дає змогу: використовувати лінії сітки на одній з осей, наносити мітки, наносити підписи міток, задавати напрямки осей і багато іншого.

У таблиці 3.3 подано лише основні функції, що відповідають за оформлення графіків і графічних вікон.

Під час завдання виведеного тексту ('text' у командах заголовка, підписів осей, виведення тексту) можливе нанесення верхнього і нижнього індексів. Ідентифікатор верхнього індексу "^", нижнього - "_". Якщо в індексі більше двох символів, то їх необхідно вказувати у фігурних дужках {}. Наприклад, виведення тексту U_{AB}^2, V в підписі осі задається командою `ylabel('U_{AB}^2, V')`.

Виведення грецьких букв і спеціальних математичних символів здійснюється застосуванням так званих TeX символів. Таблиця TeX символів наведена в додатку А. Ідентифікатором TeX символу є оператор "\ ". Наприклад, виведення в підписі осі задається командою `ylabel('\omega_2, c^{1}')`.

Функція `axis` має кілька можливостей для налаштування масштабу, орієнтації та коефіцієнта стиснення координатних осей.

Зазвичай MATLAB автоматично знаходить максимальне і мінімальне значення змінних і обирає відповідний масштаб і маркування осей. Функція

$$\text{axis}([x_{\min} \ x_{\max} \ y_{\min} \ y_{\max}])$$

замінює значення за замовчуванням граничними значеннями, що вводяться користувачем.

У функції `axis` можна також використовувати ключові слова для керування зовнішнім виглядом осей. Наприклад, `axis square` створює осі абсцис і ординат

рівної довжини, а **axis equal** створює для цих осей позначки прирощень однакової довжини. Команда **axis auto** повертає значення за замовчуванням і переходить побудову осей в автоматичний режим. Функція **axis on** вмикає позначення осей і мітки проміжних поділок, а **axis off**, відповідно, вимикає позначення осей і мітки проміжних поділок.

Таблиця 3.3 – Функції управління оформленням графіків

Функція	Призначення
title ('text')	поміщає над графіком заголовок 'text'
legend (string1, string2,...)	поміщає на поточному графіку пояснення у вигляді заданих текстових рядків
xlabel ('text')	позначає вісь x написом 'text'
ylabel ('text')	позначає вісь y написом 'text'
text (p1, p2, 'text')	додає до поточного графіка напис 'text', починаючи з позиції (p1, p2), де (p1, p2) - координати точки по осях хуу
grid on	наносить лінії сітки на поточний графік
grid off	видаляє лінії сітки з поточного графіка
hold on	Дозволити додавання графіка при повторному зверненні до об'єкта командою plot. Команда корисна під час побудови серії графіків із циклу.
hold off	Заборонити додавання графіка в координатній системі. При повторному зверненні до об'єкта командою plot в поле буде виведено новий графік, при цьому попередній знищиться.
axis	керування виведенням координатних осей
Xlim([xmin xmax]) Ylim([ymin ymax])	Встановлення граничних значень за осями x і y

3.2 Методичний приклад

Розглянемо застосування описаних операторів на прикладі побудови механічних характеристик асинхронного двигуна за експериментальними даними і формулою Клосса:

```
>> %Результати експерименту
>> se=[0 0.1 0.21 0.3 0.39 0.5 0.6 0.7 0.81 0.91 1];
>> me=[0 34.1 56.3 66.1 63 58 53 49.2 47 42 37];
>> %Розрахунок за формулою Клосса
>> % Формуємо вектор ковзань
>> s=[0:0.01:1];
>> % Задаємо критичний момент двигуна
>> mk=65;
>> % Задаємо критичне ковзання
>> sk=0.34;
>> % Розраховуємо характеристику за формулою Клосса
>> m=2*mk./(s./sk+sk./s);
Warning: Divide by zero.
>> % Виконуємо побудову характеристик
>> plot(se,me,'ko')
>> hold on
>> plot(s,m,'k:')
>> ylabel('M, [Nm]')
>> xlabel('S')
```

Результат виконання програми наведено на малюнку 3.1.

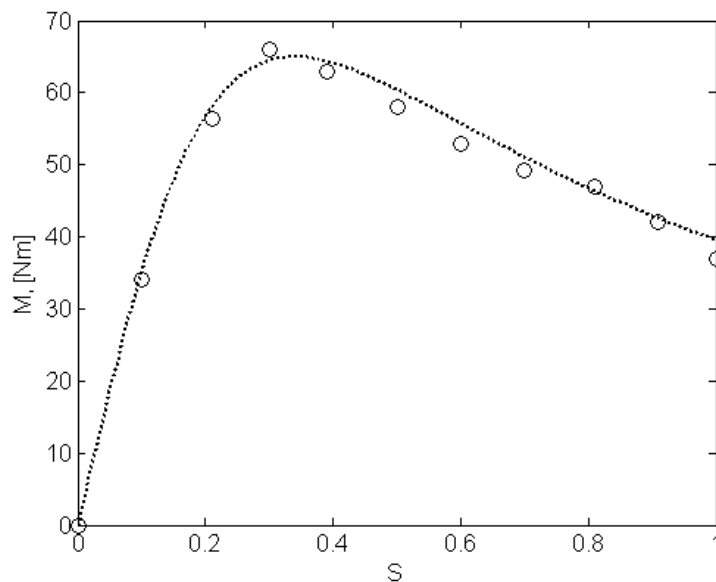


Рисунок 3.1 – Результат виконання програми

3.3 Порядок виконання роботи:

1. Відкрийте файл для ведення щоденника сесії **D:\ML_LR\GR_X\BR_Y\DN2.txt**. Виконайте методичний приклад 3.2.

2. Створіть скрипт-програму (m-файл) побудови графіків. Створіть нове графічне вікно і розбийте його на області відповідно до завдання (рис. 3.2). Пропорції розбивки вказано вздовж сторін областей графічного вікна, умовну нумерацію нанесено в центрі цих областей. В областях виконайте побудови відповідно до таблиці 3.4. Збережіть отриманий скрипт із назвою **D:\ML_LR\GR_X\BR_Y\lb3.m**. За необхідності виконайте масштабування графіків за допомогою команди `axis`. Підпишіть отримані графіки за допомогою функцій `title`, `xlabel`, `ylabel`

3. Перенесіть отримані фігури за допомогою команди з меню графічного вікна **Edit**→**Copy Figure** у буфер обміну і потім вставте як малюнок у документ **Word**.

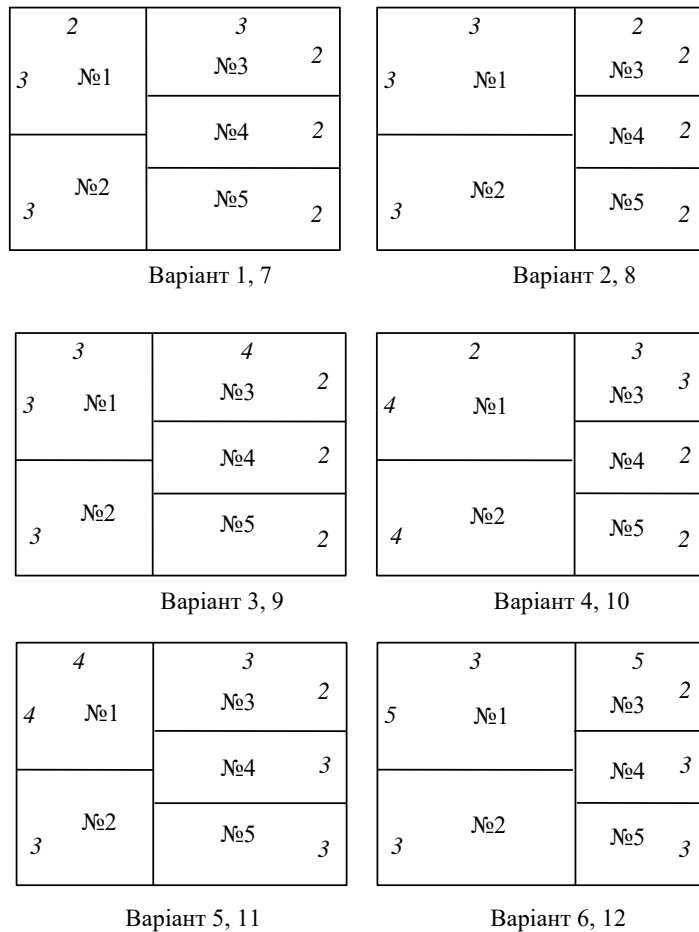


Рисунок 3.2 – Варіанти розбивки вікна фігури на області

Таблиця 3.4 – Варіанти завдань

№ варіанта (бригади)	Область	Функція
1,6	1	$y_1(t) = 5 \cdot e^{at} \cdot \sin(0.1t)$, $a = -10$, $0 \leq t \leq 1$, $\Delta t = 0.01$ суцільна синя
	2	$y_2(t) = 5 \cdot e^{at} \cdot \sin(0.1t)$, $a = 10$, $0 \leq t \leq 1$, $\Delta t = 0.01$ суцільна червона, на графіку рівномірно розмістити 20 маркерів типу '+'
	3	$y_3(x) = 2 \sin(x) \cdot \sin(2x)$, $y_4(x) = 2 \sin(x) \cdot \sin(2x + \pi / 6)$, $-2\pi \leq x \leq 2\pi$, $\Delta x = 1^\circ$ суцільна чорна, пунктирна чорна; легенда
	4	$y_5(x) = 2 \sin(x) + 0.3 \sin(5x + \pi/3)$, $-2\pi \leq x \leq 2\pi$, $\Delta x = 1^\circ$ штрих-пунктирна бузкова (товста)

	5	$y_6(x) = 2 \sin(x) / \cos(2x + \pi/6), \quad -\pi \leq x \leq \pi, \quad \Delta x = 1^\circ$ переривчаста жовта
2,7	1	$y_1(t) = 50 \cdot e^{at} \cdot \sin(0.2t), \quad a = -1, \quad 0 \leq t \leq 2, \quad \Delta t = 0.01$ суцільна бузкова
	2	$y_2(t) = 50 \cdot e^{at} \cdot \sin(0.2t), \quad a = 1, \quad 0 \leq t \leq 2, \quad \Delta t = 0.01$ суцільна червона
	3	$y_3(x) = \sin(x) \cdot \cos(5x), \quad -2\pi \leq x \leq 2\pi, \quad \Delta x = 1^\circ$ суцільна зелена на графіку рівномірно розмістити 20 маркерів типу 'o'
	4	$y_{41}(x) = 2 \sin(x) \cdot \cos(2x + \pi/3),$ $y_{42}(x) = \sin(x) \cdot \cos(2x + \pi/3), \quad -2\pi \leq x \leq 2\pi, \quad \Delta x = 1^\circ$ бузкова суцільна, штрих-пунктирна; легенда
	5	$y_5(x) = 2 \sin(x) + 2 / \cos(3x + \pi/4), \quad -\pi \leq x \leq \pi, \quad \Delta x = 1^\circ$ переривчаста синя товста
3,8	1	$y_1(t) = 2 \cdot e^{at} \cdot \sin(0.03t), \quad a = -1, \quad 0 \leq t \leq 2, \quad \Delta t = 0.01$ суцільна бузкова
	2	$y_2(t) = 2 \cdot e^{at} \cdot \sin(0.03t), \quad a = -3, \quad 0 \leq t \leq 2, \quad \Delta t = 0.01$ суцільна червона товста
	3	$y_3(x) = \sin(x) \cdot \cos(5x), \quad -2\pi \leq x \leq 2\pi, \quad \Delta x = 1^\circ$ суцільна жовта, на графіку рівномірно розмістити 10 маркерів типу '*'
	4	$y_4(x) = \sin(x),$ $y_5(x) = \sin(x) + 0.4 \cos(5x + \pi/7), \quad -2\pi \leq x \leq 2\pi, \quad \Delta x = 1^\circ$ синя суцільна, штрих-пунктирна; легенда
	5	$y_6(x) = 2 \sin(x) - 10 / \cos(5x + \pi/4), \quad -\pi \leq x \leq \pi, \quad \Delta x = 1^\circ$ переривчаста червона
4,9	1	$z_1(t) = 5 \cdot e^{at} \cdot \sin(0.05t), \quad a = -5, \quad 0 \leq t \leq 5, \quad \Delta t = 0.01$ суцільна зелена
	2	$z_2(t) = 5 \cdot e^{at} \cdot \sin(0.05t), \quad a = -3, \quad 0 \leq t \leq 1, \quad \Delta t = 0.01$ пунктирна синя
	3	$z_3(x) = 10 \cos(x) \cdot \sin(2x), \quad -\pi \leq x \leq \pi, \quad \Delta x = 1^\circ$ суцільна чорна, на графіку рівномірно розмістити 10 маркерів типу 'x'

	4	$v_4(x) = 4 \sin(x),$ $v_5(x) = 4 \sin(x) + \sin(3x + \pi/3), \quad -2\pi \leq x \leq 2\pi, \Delta x = 1^\circ$ штрих-пунктирна червона тонка, товста; легенда
	5	$z_6(x) = 2 \cos(x) / \sin(4x + \pi/10), \quad -4\pi \leq x \leq 4\pi, \Delta x = 1^\circ$ переривчаста жовта товста товста
5,10	1	$u_1(t) = 3 \cdot e^{at} \cdot \sin(0.02t), \quad a = -1, \quad 0 \leq t \leq 4, \quad \Delta t = 0.02$ суцільна бузкова
	2	$u_2(t) = 3 \cdot e^{at} \cdot \sin(0.02t), \quad a = 1, \quad 0 \leq t \leq 4, \quad \Delta t = 0.02$ суцільна червона
	3	$u_3(x) = 4 \sin(x),$ $u_4(x) = 4 \sin(x) \cdot \cos^2(2x), \quad -2\pi \leq x \leq 2\pi, \quad \Delta x = 1^\circ$ суцільна зелена товста, тонка; легенда
	4	$u_5(x) = \sin^2(x) + \cos(2x + \pi/3), \quad -2\pi \leq x \leq 2\pi, \Delta x = 1^\circ$ штрих-пунктирна бузкова, на графіку рівномірно розмістити 15 маркерів типу 's'
	5	$u_6(x) = 2 \sin(x) / \cos(x + \pi/4), \quad -\pi \leq x \leq \pi, \quad \Delta x = 1^\circ$ переривчаста синя

3.4 Запитання для самоперевірки

1. Які функції MATLAB дають змогу здійснити виведення графіків на екран?
2. Які функції дають змогу забезпечувати графіки координатними лініями та написами?
3. Чи можна побудувати кілька графіків в одній системі координат і одному графічному вікні?
4. Яким чином можна побудувати кілька окремих графіків у різних графічних вікнах?
5. Як побудувати кілька окремих графіків у різних графічних полях одного графічного вікна?
6. За допомогою якої команди можна задати масштаби по осях графіка?

7. Яким чином можна задати колір і стиль ліній графіка, нанести на них маркери?

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Гасв Є.О., Нестеренко Б.М. Універсальний математичний пакет MATLAB і типові задачі обчислювальної математики. Навчальний посібник.– К.: НАУ, 2004. – 176 с.
2. Ильин С.П. Вариационное исчисление с применением MATLAB. – Харьков: ХПИ, 2001. – 107 с.
3. Лазарєв Ю.Ф. Початки програмування в середовищі MATLAB. Навч. посібник. – К.: “Політехніка”, 2000. – 396 с.
4. Проведение математических расчетов с использованием системы MATLAB. Метод. пособие. – Харьков: НТУ "ХПИ", 2001. – 56 с.
5. Вірченко Н.О., Ляшко І.І. Графіки елементарних та спеціальних функцій. – К.: Наук. думка, 1996. – 582 с.

ДОДАТОК А

Таблиця А.1 – Елементарні математичні функції, які використовуються в
MATLAB

abs(X)	повертає абсолютну величину для кожного числового елемента вектора X. Якщо X містить комплексні числа, abs(X) обчислює модуль кожного числа
<p>Тригонометричні функції</p> <p>Функції обчислюються для кожного елемента масиву. Вхідний масив допускає комплексні значення. Усі кути у функціях задаються в радіанах</p>	
sin(X)	Синус числа X. Якщо X - вектор, то повертає синус для кожного елемента X. X задається в радіанах.
Asin(X)	Арксинус для кожного елемента X. Для дійсних значень X в області [-1, 1] asin(X) повертає дійсне число з діапазону $[-\pi/2, \pi/2]$, для дійсних значень X поза областю [-1, 1] asin(X) повертає комплексне число
cos(X)	Косинус
acos (X)	Арккосинус. Для дійсних значень X в області [-1, 1] acos(X) повертає дійсне значення з діапазону $[0, \pi]$, для дійсних значень X поза областю [-1, 1] acos(X) повертає комплексне число.
tan(X)	Тангенс
atan(X)	Арктангенс (в діапазоні от $-\pi/2$ до $\pi/2$)
atan2(X,Y)	Чотирьох квадрантний арктангенс
cot(X)	Котангенс
acot(X)	Арккотангенс
sec(X)	Секанс
asec(X)	Арксеканс
csc(X)	Косеканс
acsc(X)	Арккосеканс
<p>Гіперболічні та обернені їм функції</p> <p>Функції обчислюються для кожного елемента масиву.</p> <p>Вхідний масив допускає комплексні значення.</p> <p>Усі кути в тригонометричних функціях вимірюються в радіанах</p>	
sinh(X)	гіперболічний синус
cosh(X)	гіперболічний косинус
tanh(X)	гіперболічний тангенс
coth(X)	гіперболічний котангенс

sech(X)	гіперболічний секанс
csch(x)	гіперболічний косеканс
asinh(X)	гіперболічний арксинус
acosh(X)	гіперболічний арккосинус
atanh(X)	гіперболічний арктангенс
acoth(X)	гіперболічний арккотангенс
asech(X)	гіперболічний арксеканс
acsch(X)	гіперболічний арккосеканс
Експоненціальні функції	
exp(X)	Експонента числа X. Для комплексного числа $z = x + i*y$ функція $\exp(z)$ обчислює комплексну експоненту: $\exp(z) = \exp(x) * (\cos(y) + i * \sin(y))$
log (X)	Натуральний логарифм числа X. Для комплексного або від'ємного z , де $z = x + y*i$, обчислюється комплексний логарифм у вигляді $\log(z) = \log(\text{abs}(z)) + i * \text{atan2}(y, x)$.
log10(X)	Логарифм за основою 10 для числа X. Область функції включає комплексні числа.
sqrt(A)	повертає квадратний корінь кожного елемента масиву X. Для від'ємних і комплексних елементів X функція sqrt(X) обчислює комплексний результат
Функції комплексного аргумента	
real(Z)	Дійсна частина комплексного аргументу
imag(Z)	Уявна частина комплексного аргументу
angle(Z)	Обчислює значення аргументу комплексного числа Z у діапазоні від $-\pi$ до π
conj(Z)	Повертає число, комплексно-спряжене аргументу Z. Якщо Z комплексне, то $\text{conj}(Z) = \text{real}(Z) - i * \text{imag}(Z)$

Таблиця А.2 – Список TeX-символів

Рядок опис	СИМВОЛ	Рядок опис	СИМВОЛ	Рядок опис	СИМВОЛ
<code>\alpha</code>	α	<code>\upsilon</code>	υ	<code>\sim</code>	\sim
<code>\beta</code>	β	<code>\phi</code>	ϕ	<code>\leq</code>	\leq
<code>\gamma</code>	γ	<code>\chi</code>	χ	<code>\infty</code>	∞
<code>\delta</code>	δ	<code>\psi</code>	ψ	<code>\clubsuit</code>	\clubsuit
<code>\epsilon</code>	ϵ	<code>\omega</code>	ω	<code>\diamondsuit</code>	\diamondsuit
<code>\zeta</code>	ζ	<code>\Gamma</code>	Γ	<code>\heartsuit</code>	\heartsuit
<code>\eta</code>	η	<code>\Delta</code>	Δ	<code>\spadesuit</code>	\spadesuit
<code>\theta</code>	θ	<code>\Theta</code>	Θ	<code>\leftrightarrow</code>	\leftrightarrow
<code>\vartheta</code>	ϑ	<code>\Lambda</code>	Λ	<code>\leftarrow</code>	\leftarrow
<code>\iota</code>	ι	<code>\Xi</code>	Ξ	<code>\uparrow</code>	\uparrow
<code>\kappa</code>	κ	<code>\Pi</code>	Π	<code>\rightarrow</code>	\rightarrow
<code>\lambda</code>	λ	<code>\Sigma</code>	Σ	<code>\downarrow</code>	\downarrow
<code>\mu</code>	μ	<code>\Upsilon</code>	Υ	<code>\circ</code>	\circ
<code>\nu</code>	ν	<code>\Phi</code>	Φ	<code>\pm</code>	\pm
<code>\xi</code>	ξ	<code>\Psi</code>	Ψ	<code>\geq</code>	\geq
<code>\pi</code>	π	<code>\Omega</code>	Ω	<code>\propto</code>	\propto
<code>\rho</code>	ρ	<code>\forall</code>	\forall	<code>\partial</code>	∂
<code>\sigma</code>	σ	<code>\exists</code>	\exists	<code>\bullet</code>	\bullet
<code>\varsigma</code>	ς	<code>\ni</code>	\ni	<code>\div</code>	\div
<code>\tau</code>	τ	<code>\cong</code>	\cong	<code>\neq</code>	\neq
<code>\equiv</code>	\equiv	<code>\approx</code>	\approx	<code>\aleph</code>	\aleph
<code>\Im</code>	\Im	<code>\Re</code>	\Re	<code>\wp</code>	\wp
<code>\otimes</code>	\otimes	<code>\oplus</code>	\oplus	<code>\oslash</code>	\oslash
<code>\cap</code>	\cap	<code>\cup</code>	\cup	<code>\supseteq</code>	\supseteq
<code>\supset</code>	\supset	<code>\subseteq</code>	\subseteq	<code>\subset</code>	\subset
<code>\int</code>	\int	<code>\in</code>	\in	<code>\circ</code>	\circ
<code>\rfloor</code>	\rfloor	<code>\lceil</code>	\lceil	<code>\nabla</code>	∇
<code>\lfloor</code>	\lfloor	<code>\cdot</code>	\cdot	<code>\ldots</code>	\ldots
<code>\perp</code>	\perp	<code>\neg</code>	\neg	<code>\prime</code>	\prime
<code>\wedge</code>	\wedge	<code>\times</code>	\times	<code>\emptyset</code>	\emptyset
<code>\rceil</code>	\rceil	<code>\surd</code>	\surd	<code>\mid</code>	\mid
<code>\vee</code>	\vee	<code>\varpi</code>	ϖ	<code>\copyright</code>	\copyright
<code>\langle</code>	\langle	<code>\rangle</code>	\rangle		

Таблиця А.3 – Кодування типів маркерів на лініях графіків

Знак	Опис		Примітки
+	plus sign	Плюс	
o	circle	Окружність	
*	asterisk	Сніжинка	
.	point	Крапка	Круг
x	cross	Хрест	
s	square	Квадрат	
d	diamond	«Діамант»	◇
^	upward pointing triangle	Трикутник із відповідним поворотом вершини	▲
v	downward pointing triangle		▼
>	right pointing triangle		►
<	left pointing triangle		◄
p	five-pointed star	П'ятикутна зірка	
h	six-pointed star	Шестикутна зірка	
none	no marker (default)		

Навчальне видання

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ЛАБОРАТОРНИХ РОБІТ

з дисципліни

«МОДЕЛЮВАННЯ ЕЛЕКТРОМЕХАНІЧНИХ СИСТЕМ»

Частина I

(для здобувачів вищої освіти спеціальності 141 «Електроенергетика,
електротехніка та електромеханіка»)

(електронне видання)

Укладачі:

Є. С. РУДНЄВ

Ю. А. РОМАНЧЕНКО

Оригінал-макет *Ю.А. Романченко*

Підписано до друку 24.04.2023.

Формат 60x84 ¹/₁₆. Папір типогр. Гарнітура Times.

Друк офсетний. Умов. друк. арк. _____. Обл.-вид. арк. _____.

Тираж ____ екз. Вид. № _____. Замов. № _____. Ціна договірна.

**Видавництво Східноукраїнського національного університету
імені Володимира Даля**

Свідоцтво про реєстрацію: серія ДК № 1620 від 18.12.03 р.

Адреса університета: вул. Іоанна Павла 2, 17

м. Київ, 01042, Україна

e-mail: vidavnictvoSNU.ua@gmail.com.