

**УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КЕРУЮЧИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

**Кафедра автоматики та комп'ютерного телекерування
рухом поїздів**

ВИКОРИСТАННЯ MATLAB ДЛЯ ІНЖЕНЕРНИХ РОЗРАХУНКІВ

МЕТОДИЧНІ ВКАЗІВКИ

**для виконання практичних робіт, самостійної роботи
з освітньої компоненти**

***«МЕТОДИ ТА ПРОГРАМНО-ТЕХНІЧНІ ЗАСОБИ ІНЖЕНЕРНИХ
РОЗРАХУНКІВ»***

**для здобувачів вищої освіти першого (бакалаврського) рівня
спеціальностей G9 (131) «Прикладна механіка»,
G11.03 (133) «Машинобудування (Технологічні машини та
обладнання)», G18 (193) «Геодезія та землеустрій»,
G19 (192) «Будівництво та цивільна інженерія» і
J7 (273) «Залізничний транспорт»**

Харків 2026

Методичні вказівки розглянуто і рекомендовано до друку на засіданні кафедри автоматики та комп'ютерного телекерування рухом поїздів 29 грудня 2025 р., протокол № 4.

Методичні вказівки призначені для здобувачів спеціальностей G9 (131) «Прикладна механіка», G11.03 (133) «Машинобудування (Технологічні машини та обладнання)», G18 (193) «Геодезія та землеустрій», G19 (192) «Будівництво та цивільна інженерія» і J7 (273) «Залізничний транспорт», які вивчають освітню компоненту «Методи та програмно-технічні засоби інженерних розрахунків будівельного факультету, усіх форм здобуття освіти.

Укладачі:

доценти С. О. Змій,
Д. П. Лабенко

Рецензент

доц. О. А. Дудін

Зміст

Практична робота 1. Ознайомлення з середовищем Matlab. Найпростіші обчислення та робота зі змінними	4
Практична робота 2. Операції з векторами та матрицями. Лінійна алгебра в Matlab.....	13
Практична робота 3. Візуалізація даних: побудова 2D- і 3D-графіків	20
Практична робота 4. Обробка експериментальних даних. Інтерполяція та апроксимація	25
Практична робота 5. Символьні обчислення (symbolic math toolbox)	30
Список літератури.....	34

Практична робота 1. Ознайомлення з середовищем Matlab.

Найпростіші обчислення та робота зі змінними

Мета роботи

Метою роботи є формування у здобувачів базових навичок роботи в інтегрованому середовищі MATLAB (Matrix Laboratory). Здобувачі повинні ознайомитися з архітектурою системи, інтерфейсом користувача, навчитися керувати вікнами, опанувати синтаксис найпростіших команд, вивчити фундаментальні типи даних, правила створення змінних і специфіку машинної арифметики з рухомою комою. Головним завданням є набуття компетенцій у використанні вбудованих математичних функцій для виконання скалярних інженерних обчислень.

1.1 Теоретичні відомості

MATLAB (скорочення від *Matrix Laboratory*) – це не просто мова програмування, а високопродуктивне інтерактивне середовище для технічних обчислень, розроблене компанією MathWorks. Його унікальність полягає в інтеграції обчислювальних потужностей, засобів візуалізації та інструментів програмування в єдину оболонку, де задачі та їхні розв'язки виражені у формі, максимально наближеній до математичної нотації.

Історія MATLAB сягає кінця 1970-х років, коли Клів Молер (Cleve Moler), професор університету Нью-Мексико, розробив першу версію програми як простий інтерфейс для бібліотек LINPACK (Linear System Package) і EISPACK (Eigen System Package), написаних мовою Fortran. Метою було дати здобувачам можливість працювати з лінійною алгеброю без необхідності вивчення складного синтаксису Fortran. Згодом, у 1984 році, була заснована компанія MathWorks, і MATLAB був переписаний мовою C, що забезпечило йому високу швидкість та переносимість.

Сьогодні ядром MATLAB є бібліотеки BLAS (Basic Linear Algebra Subprograms) і LAPACK, які забезпечують високу ефективність векторних і матричних операцій.

Сучасна архітектура MATLAB складається з п'яти основних частин:

1 **Мова MATLAB:** високорівнева мова матриць і масивів із функціями керування потоком, структурами даних, введенням/виведенням та об'єктно-орієнтованим програмуванням.

2 **Робоче середовище (Desktop):** набір інструментів і засобів, що полегшують роботу користувача або програміста (Command Window, Editor, Workspace, Profiler).

3 **Графічна система (Handle Graphics)** включає високорівневі команди для 2D- і 3D-візуалізації, обробки зображень і створення графічних інтерфейсів.

4 **Бібліотека математичних функцій:** величезна колекція алгоритмів, від елементарних (синус, косинус) до складних (обернення матриць, швидке перетворення Фур'є, функції Бесселя).

5 **API (Application Program Interface):** бібліотека, що допомагає писати програми мовами C/C++ і Fortran, які взаємодіють із MATLAB.

Інтерфейс користувача та налаштування середовища

Запускаючи MATLAB, користувач потрапляє в робочий стіл (MATLAB Desktop), який є гнучким і налаштовуваним середовищем. За замовчуванням він містить такі критично важливі компоненти, кожен із яких виконує специфічну роль у процесі розроблення:

– **Command Window (вікно команд).** Це основний інтерфейс для діалогу з системою. Користувач вводить команди після символу запрошення >> (prompt). Це вікно працює в режимі REPL (Read-Eval-Print Loop): система зчитує команду, обчислює її та одразу виводить результат, що ідеально для швидких розрахунків і налагодження коду;

- **Workspace (робочий простір).** Це візуалізація оперативної пам'яті поточного сеансу. Тут відображені всі змінні, які були створені або імпортовані. Для кожної змінної вказують її ім'я (Name), значення (Value), розмірність (Size), тип (Class) та обсяг зайнятої пам'яті. Можливість візуального контролю стану змінних є однією з головних переваг MATLAB порівняно з компільованими мовами;

- **Current Folder (поточна папка).** Файловий менеджер, інтегрований у середовище. Він відображає вміст поточної робочої директорії. Це важливо, оскільки MATLAB може запускати скрипти і функції лише тоді, коли вони знаходяться в «полі зору» системи, тобто в поточній папці або папках, прописаних у шляхах пошуку (Search Path);

- **Editor/Debugger (редактор).** Інструмент для написання, редагування та налаштування М-файлів (скриптів і функцій). Він підтримує підсвічування синтаксису, автодоповнення коду та інтерактивне налагодження (breakpoints).

Основи синтаксису: типи даних і змінні

MATLAB є мовою з динамічною типізацією. Це означає, що користувачеві не потрібно явно оголошувати тип змінної (як `int` або `float` у C++) перед її використанням. Тип даних визначений автоматично із присвоєнням значення.

Базовим елементом даних у MATLAB є **масив (array)**. Навіть одиничне число (скаляр) система розглядає як матрицю розміром 1×1 . Це спрощує синтаксис, оскільки одні й ті самі функції можуть працювати як зі скалярами, так і векторами та матрицями (поліморфізм).

Основні числові типи:

- *double* – основний тип даних за замовчуванням. Це числа з рухомою комою подвійної точності згідно зі стандартом IEEE 754. Вони займають 64 біти (8 байтів) пам'яті. Забезпечують точність приблизно 15-16 значущих десяткових цифр. Діапазон значень: від 10^{-308} до 10^{308} ;

- *single* – числа з рухомою комою одинарної точності (32 біти), використовувані для економії пам'яті, працюючи з величезними масивами, але мають меншу точність;

- *int8, int16, int32, int64* (і відповідні беззнакові *uint*) – цілочисельні типи. Використовують рідше, переважно для обробки зображень або роботи з даними від мікроконтролерів.

Спеціальні константи

Система має деякі зарезервовані змінні, які мають спеціальне значення:

- *ans (answer)* – зберігає результат останньої операції, якщо він не був присвоєний іншій змінній;

- *pi* - число π (3.14159...);

- *eps (floating-point relative accuracy)* – відстань від 1.0 до наступного найбільшого числа подвійної точності. Це міра машинної точності ($2.22 \cdot 10^{-16}$). Використовують для оцінювання похибок округлення;

- *Inf (Infinity)* – нескінченність. Виникає з діленням на нуль ($1/0$) або переповненням;

- *NaN (Not a Number)* – не число. Результат невизначених операцій, таких як $0/0$ або $\inf - \inf$. Операції з NaN завжди дають NaN, що дає змогу відслідковувати помилки в ланцюжку обчислень;

- *i, j* уявна одиниця (корінь квадратний від мінус одиниці).

Правила іменування змінних:

- 1 Ім'я має починатися з літери.

- 2 Може містити літери латинського алфавіту, цифри та символ підкреслення.

- 3 Регістрозалежність: змінні *Var*, *var* і *VAR* є різними.

- 4 Довжина імені обмежена (зазвичай 63 символи, що перевіряє функція `namelengthmax`).

5 Заборонено використовувати зарезервовані слова (keywords), такі як if, while, break, end. Список ключових слів можна отримати командою iskeyword.

Вбудовані математичні функції та формати виведення

MATLAB надає доступ до широкого спектра елементарних і спеціальних функцій. Важливо пам'ятати, що аргументи тригонометричних функцій (*sin*, *cos*, *tan*) за замовчуванням у радіанах (таблиця 1.1).

Таблиця 1.1 – Основні математичні функції

Функція	Математичний зміст	Приклад	Результат
abs(x)	Модуль x	abs(-10)	10
sqrt(x)	Квадратний корінь x	sqrt(16)	4
exp(x)	Експонента	exp(1)	2.7183
log(x)	Натуральний логарифм	log(exp(2))	2
log10(x)	Десятковий логарифм	log10(100)	2
sin(x)	Синус (радіани)	sin(pi/2)	1
cosd(x)	Косинус (градуси)	cosd(60)	0.5
mod(x,y)	Залишок від ділення	mod(13, 5)	3
round(x)	Округлення до найближчого	round(3.6)	4
floor(x)	Округлення вниз	floor(3.9)	3
ceil(x)	Округлення вгору	ceil(3.1)	4

Керування форматом виведення

Команда *format* змінює спосіб відображення чисел у Command Window, але не впливає на точність внутрішніх обчислень:

- *format short*: (за замовчуванням) чотири знаки після коми (3.1416);
- *format long*: 15 знаків після коми (3.141592653589793);
- *format short e*: експоненціальний запис (3.1416e+00);
- *format rat*: наближення раціональним дробом (355/113).

1.2 Програма роботи

- 1 Запустити середовище MATLAB і налаштувати робочий стіл.
 - 2 Використати Command Window як інженерний калькулятор для виконання арифметичних дій.
 - 3 Створити змінні різних типів, перевірити роботу з пам'яттю (Workspace).
 - 4 Дослідити вплив команд форматування (*format*) на відображення результатів.
 - 5 Виконати індивідуальне завдання: обчислити значення складного математичного виразу для заданих параметрів.
- Оформити звіт, що містить скриншоти виконання команд і аналіз результатів.

1.3 Порядок виконання

Крок 1. Запуск і діагностика.

Запустіть MATLAB. У командному рядку введіть *ver* для перегляду встановлених тулбоксів. Введіть *bench*, щоб запустити вбудований тест продуктивності комп'ютера. Це допоможе оцінити відносну швидкість вашої машини порівняно з еталонними системами.

Крок 2. Робота в режимі калькулятора.

Виконайте такі обчислення, аналізуючи пріоритет операцій:

Matlab

```
>> 2 + 3 * 4^2  
>> (2 + 3) * 4^2  
>> 10 / 0 % Зверніть увагу на результат Inf  
>> 0 / 0 % Зверніть увагу на результат NaN
```

Зверніть увагу, що символ ; (крапка з комою) наприкінці команди пригнічує виведення результату на екран, хоча обчислення виконуване, і змінна *ans* оновлювана. Це корисно для виконання проміжних розрахунків.

Крок 3. Робота зі змінними.

Створіть змінні для радіуса та висоти циліндра, обчисліть його об'єм.

Matlab

```
>> R = 5;  
>> H = 10;  
>> V = pi * R^2 * H
```

Перевірте вікно Workspace. Спробуйте видалити змінну *R* командою *clear R*. Очистіть весь простір командою *clear all* та очистіть екран командою *clc*.

Крок 4. Дослідження точності та форматів.

Введіть число π у різних форматах.

Matlab

```
>> format short; pi  
>> format long; pi  
>> format bank; pi % Формат для валюти (2 знаки)  
>> format rat; pi
```

Крок 5. Виконання індивідуального завдання.

Відповідно до вашого варіанта (таблиця 1.2) запишіть математичний вираз мовою MATLAB. Зверніть увагу на розстановку дужок для збереження математичного порядку дій. Використовуйте змінні для вхідних параметрів.

Таблиця 1.2 – Основні математичні функції

Варіант	Вираз для обчислення Z	Вхідні дані
1	$Z = \frac{\sqrt{x^2 + a^2} - \cos(x)}{e^a + \ln(x)}$	x = 2.5, a = 1.2
2	$Z = \tan(x) \cdot \sin^2(y) + \sqrt[3]{x - y}$	x = 0.8, y = 0.5
3	$Z = \frac{e^{-2x} + \sin(3x)}{\log_{10}(x) + 5}$	x = 4.2
4	$Z = \arcsin\left(\frac{x}{\sqrt{1 + x^2}}\right) + x \cdot e^x$	x = 0.6
5	$Z = \frac{a^b - b^a}{\sin(a) + \cos(b)} + \sqrt{ab}$	a = 2.1, b = 3.5
6	$Z = \sqrt[4]{x^4 + y^4} + \log_2(x + 5)$	x = 3, y = 4
7	$Z = \frac{\cos(x + y)}{1 + x^2 y^2} + e^{-0.5x}$	x = 1.0, y = 2.0
8	$Z = \sqrt{1 + \sin^2(x)} - \frac{1}{2} \tan(x/2)$	x = \pi/3
9	$Z = \frac{x^2 + 2x + 1}{\sqrt{x} + \ln(x)} \cdot \sin(2x)$	x = 2.8
10	$Z = 2^{-x} + (x^2 + 1)^{-1} - \cos^2(x)$	x = 1.7
11	$Z = \frac{\sin(x) - \cos(x)}{\sin(x) + \cos(x)} \cdot e^x$	x = 0.4
12	$Z = \sqrt[4]{x^4 + y^4} + \log_2(x + 5)$	x = 3, y = 4
13	$Z = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	x = 2, μ = 0, σ = 1
14	$Z = \ln\left(\sqrt{\frac{1+x}{1-x}}\right) + \arctan(x)$	x = 0.3
15	$Z = \frac{x \sin(y) - y \cos(x)}{x^2 + y^2 + 1}$	x = 1.2, y = 0.8

1.4 Індивідуальні завдання

Скласти програму (скрипт) мовою MATLAB для обчислення значення виразу Z за заданих вхідних даних. Забезпечити виведення результату у зручному форматі.

Контрольні запитання

- 1 Охарактеризуйте призначення та основні складові інтерфейсу системи MATLAB.
- 2 Поясніть поняття робочого простору (Workspace) і шляхів пошуку (Search Path).
- 3 Які існують типи даних у MATLAB? У чому особливість типу double?
- 4 Яка різниця між результатом ділення $1/0$ і $0/0$? Що таке eps?
- 5 Як організовано пріоритет арифметичних операцій?
- 6 Як приховати виведення результату виконання команди?
- 7 Які команди використовують для очищення пам'яті та екрана консолі?
- 8 Як обчислити $\sin^2(x)$ і $\cot(x)$ засобами MATLAB?

Практична робота 2. Операції з векторами та матрицями. Лінійна алгебра в Matlab

Мета роботи

Глибоке вивчення методів роботи з масивами – основним типом даних у MATLAB. Здобувачі повинні навчитися створювати і маніпулювати векторами і матрицями, розуміти принципи індексації та адресації пам'яті. Особливу увагу приділено розрізненню матричних (лінійна алгебра) і поелементних (array math) операцій. Практична частина передбачає застосування цих знань для розв'язання систем лінійних алгебраїчних рівнянь (СЛАР).

2.1 Теоретичні відомості

Матриці як фундаментальний об'єкт

У середовищі MATLAB будь-яку змінну розглядають як матрицю. Навіть якщо ми працюємо з зображеннями, звуком або текстом, на низькому рівні це масиви чисел. Матриці в MATLAB зберігаються в пам'яті за стовпцями (column-major order), як у мові Fortran, на відміну від C/C++, де зберігання відбувається за рядками. Це важливо пам'ятати для оптимізації коду: прохід по елементах матриці вздовж стовпця буде швидшим, ніж уздовж рядка, через особливості кешу процесора.

Способи створення матриць

Масиви задані у квадратних дужках []:

- вектор-рядок: елементи відокремлені пробілами або комами

$$V = \text{або } V = [1\ 2\ 3]. \text{ Розмірність: } 1 \times 3;$$

- вектор-стовпець: елементи відокремлені крапкою з комою

$$C = [1; 2; 3]. \text{ Розмірність: } 3 \times 1;$$

- двовимірна матриця: рядки відокремлені ;

$$M = [1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9].$$

Оператор двокрапка (:)

Це один із найпотужніших інструментів MATLAB для генерації числових послідовностей та індексації.

Синтаксис: *Start : Step : End*: якщо *Step* пропущено, він дорівнює 1;

- приклад: $x = 1:5$ створює вектор [1 2 3 4 5];
- приклад: $y = 10:-2:4$ створює [10 8 6 4].

Це надзвичайно ефективний спосіб створення векторів аргументів для побудови графіків.

Генерація спеціальних матриць

Для прискорення роботи існують функції, що створюють матриці заданого розміру $m \times n$:

- *zeros(m, n)*: нульова матриця, часто використовувана для попереднього виділення пам'яті (preallocation) перед циклами, що значно прискорює виконання коду;
- *ones(m, n)*: матриця з одиниць;
- *eye(n)*: одинична матриця *I* (Identity matrix), де на головній діагоналі одиниці, а решта – нулі;
- *rand(m, n)*: матриця псевдовипадкових чисел, рівномірно розподілених на інтервалі (0, 1);
- *randn(m, n)*: матриця випадкових чисел із нормальним (гаусовим) розподілом (середнє 0, дисперсія 1);
- *magic(n)*: магічний квадрат (суми по рядках, стовпцях і діагоналях рівні);
- *diag(V)*: створення діагональної матриці з вектора *V* або вилучення головної діагоналі з існуючої матриці.

Індексація та доступ до даних

Індексація в MATLAB починається з 1 (one-based indexing). Це відрізняє її від Python або C, де індексація починається з 0.

Доступ до елементів - через круглі дужки: $M(\text{row}, \text{col})$.

Типи індексації:

1 **Індексація за підписом (Subscript indexing):** $A(i, j)$ – елемент на перетині i -го рядка та j -го стовпця.

2 **Лінійна індексація (Linear indexing):** оскільки матриця зберігається як довгий вектор стовпців, до елемента можна звернутися одним індексом. Наприклад, для матриці 3×3 елемент $A(1, 2)$ (перший рядок, другий стовець) буде четвертим за рахунком у пам'яті, тому $A(4)$ поверне те саме значення.

3 **Логічна індексація (Logical indexing):** використання масиву булевих значень (true/false) для вибору елементів, що задовольняють певну умову. Наприклад, $A(A > 5)$ поверне вектор усіх елементів, більших за 5.

Вирізка підматриць (Slicing)

Використання двокрапки дає змогу вибирати діапазони:

- $A(:, 1)$: усі рядки, перший стовець;
- $A(2, :)$: другий рядок, усі стовпці;
- $A(1:2, 2:3)$: підматриця з перетину 1-2 рядків і 2-3 стовпців;
- $A(\text{end}, :)$: останній рядок. Ключове слово `end` автоматично визначає останній індекс.

Матричні операції проти поелементних

Це фундаментальна концепція, нерозуміння якої веде до більшості помилок початківців.

1 **Матричні операції (Linear Algebra):** виконувані за правилами вищої математики.

* (множення): $A * B$. Кількість стовпців A має дорівнювати кількості рядків B . Це скалярний добуток рядка на стовець.

^ (піднесення до степеня): A^2 означає $A * A$. Матриця має бути квадратною.

' (транспонування): A' – ермітове спряжене транспонування (для комплексних чисел змінює знак уявної частини). $A.'$ – звичайне транспонування.

$\backslash$ (ліве ділення): $A \backslash B$. Унікальний оператор MATLAB для розв'язання рівнянь $Ax = B$.

$/$ (праве ділення): B / A . Розв'язання рівняння $xA = B$.

2 Поелементні операції (Array Operations): виконувані з відповідними елементами масивів однакового розміру. Позначають **крапкою** перед оператором.

$.*$ (множення): $C = A.* B$ означає $c_{ij} = a_{ij} b_{ij}$.

$./$ (ділення): $C = A./ B$.

$.^$ (ступінь): $A.^ 2$ підносить кожен елемент до квадрата.

Приклад. Якщо x – вектор значень часу, то вираз $y = x^2$ викличе помилку, бо не можна помножити вектор на вектор матрично. Правильний запис: $y = x.^2$.

Системи лінійних алгебраїчних рівнянь (СЛАР)

Розв'язання систем вигляду $Ax = b$ є однією з найчастіших задач в інженерії (розрахунок електричних кіл, механічних конструкцій, баланс мас).

Математично розв'язок записують як $x = A^{-1}b$.

У MATLAB це можна реалізувати як $x = \text{inv}(A) * b$. Проте так робити не рекомендовано з двох причин:

1 Обчислення оберненої матриці є обчислювально витратним ($O(n^3)$).

2 Це призводить до значних похибок округлення для погано обумовлених матриць.

Правильний метод: Оператор $\backslash$ (mldivide).

Команда $x = A \backslash b$ використовує розумний алгоритм. Вона аналізує матрицю A :

- якщо A трикутна, використано просту підстановку;
- якщо A симетрична і додатно визначена, використано розклад Холецького;

– у загальному випадку використано LU-розкладання із частковим вибором ведучого елемента (Gaussian elimination). Це забезпечує найвищу швидкість і точність.

2.2 Порядок виконання

- 1 Створити вектори-рядки та вектори-стовпці різними методами.
- 2 Сформувати матриці за допомогою функцій *rand*, *ones*, *zeros*.
- 3 Виконати операції конкатенації (склеювання) матриць.
- 4 Навчитися виділяти підматриці та окремі елементи.
- 5 Дослідити різницю між $A*B$ та $A.*B$.
- 6 Розв'язати СЛАР методом лівого ділення та перевірити результат.
- 7 Виконати індивідуальне завдання з обробки матричних виразів.

Крок 1. Маніпуляції з векторами.

Створіть вектор v від 0 до 10 з кроком 2. Перетворіть його на стовпець.

Matlab

```
v = 0:2:10  
v_col = v' % Транспонування  
len = length(v) % Довжина вектору
```

Крок 2. Матричні операції.

Створіть матриці.

Matlab

```
A = [1 2; 3 4];  
B = [5 6; 7 8];  
C_mult = A * B % Матричне множення  
C_dot = A.*B % Поелементне множення
```

Проаналізуйте різницю результатів.

Крок 3. Розв'язання СЛАР.

Дано систему

$$\{(4x_1 + x_2 - x_3 = 8 @ x_1 - 5x_2 + 2x_3 = -1 - 2x_1 + x_2 + 3x_3 = 5).$$

Сформууйте матрицю коефіцієнтів A і вектор вільних членів b .

Matlab

```
A = [4 1 -1; 1 -5 2; -2 1 3];  
b = [8; -1; 5];
```

Перевірте визначник матриці ($\det(A)$). Якщо він не нуль, система має єдиний розв'язок.

Знайдіть розв'язок.

Matlab

```
x = A \ b
```

Виконайте перевірку ($A*x$ має дорівнювати b).

Matlab

```
check = A*x - b % Результат має бути дуже близьким до нуля (наприклад 1e-15)
```

2.3 Індивідуальні завдання

- 1 Задати матриці A і B і вектор b відповідного розміру.
- 2 Обчислити матричний вираз C згідно з варіантом (таблиця 2.1).
- 3 Знайти розв'язок системи лінійних рівнянь $Ax = b$ (або $Bx = b$, якщо вказано) методом лівого ділення.
- 4 Виконати перевірку розв'язку.

Контрольні запитання

- 1 Чому індексація $A(1,2)$ і $A(4)$ може повертати однакове значення для матриці 3×3 ?
- 2 Як створити одиничну матрицю довільного розміру?
- 3 Поясніть фізичний зміст оператора $:$.
- 4 Яка принципова відмінність між операціями $*$ та $.*$?
- 5 Чому використання $\text{inv}(A)$ для розв'язання СЛАР вважають поганою практикою?

- 6 Як транспонувати матрицю? Чим відрізняється ' від ' !?
- 7 Як витягнути з матриці останній стовпець, не знаючи її розміру?

Таблиця 2.1

Варіант	Матриці A та B (MATLAB syntax)	Вираз для C	Вектор b
1	A = [2 1 3; -1 4 0; 2 2 -1] B = [1 0 0; 0 1 0; 0 0 1]	$C = A^2 - 3B$	[5; -2; 1]
2	A = [1 2 -1; 3 0 2; 4 -2 5] B = [1 1 1; 2 2 2; 3 3 3]	$C = (A + B)^T \cdot A$	[10; 5; -1]
3	A = [3 -1 2; 1 5 -1; -2 3 4] B = eye(3)	$C = A \cdot B - B \cdot A$	[4; 2; 6]
4	A = [0.5 1 2; 1 0.5 3; 2 3 0.5] B = ones(3)	$C = A .* B + A$	[1; 1; 1]
5	A = [4 1 1; 1 4 1; 1 1 4] B = [2 0 -1; 0 2 0; -1 0 2]	$C = \text{inv}(A) \cdot B$	[12; 0; -4]
6	A = [1 2 3; 2 3 1; 3 1 2] B = [1 -1 1; -1 1 -1; 1 -1 1]	$C = A^3 + B^2$	[6; 6; 6]
7	A = [5 2 0; 2 5 2; 0 2 5] B = magic(3)	$C = A \cdot B^T$	[1; 2; 3]
8	A = [1 0 -2; 3 1 0; -1 2 4] B = [0 1 2; 1 0 3; 2 3 0]	$C = (A - B) .* (A + B)$	[2; -1; 4]
9	A = [2 -1 0; -1 2 -1; 0 -1 2] B = [3 1 1; 1 3 1; 1 1 3]	$C = \det(A) \cdot B$	[0; 0; 0]
10	A = [10 2 1; 2 10 2; 1 2 10] B = [1 2 3; 4 5 6; 7 8 9]	$C = A \setminus B$	[10; 20; 30]
11	A = [1 1 1; 1 2 3; 1 3 6] B = [1 0 1; 0 1 0; 1 0 1]	$C = A^2 - 2A + I$	[1; 2; 1]
12	A = [3 2 1; 2 3 2; 1 2 3] B = rand(3)	$C = A .* B - B .* A$	[5; 4; 3]
13	A = [2 3 4; 5 6 7; 8 9 1] B = [1 0 0; 0 2 0; 0 0 3]	$C = B \cdot A \cdot B$	[1; 1; 1]
14	A = [0 1 2; -1 0 3; -2 -3 0] B = ones(3) 5	$C = A^T + A$	[2; 0; -2]
15	A = [4 -2 1; -2 4 -2; 1 -2 4] B = [0.5 0 0; 0 0.5 0; 0 0 0.5]	$C = \text{inv}(B) \cdot A$	[8; -4; 2]

Практична робота 3. Візуалізація даних: побудова 2D- і 3D-графіків

Мета роботи

Метою роботи є опанування засобів графічної візуалізації системи MATLAB. Здобувачі повинні навчитися будувати графіки функцій однієї змінної (2D), параметричні криві, а також тривимірні поверхні (3D). Робота передбачає вивчення архітектури Handle Graphics для тонкого налаштування властивостей графічних об'єктів, додавання анотацій, легенд і збереження результатів у графічних форматах.

3.1 Теоретичні відомості

Концепція візуалізації в MATLAB

Візуалізація є критично важливим етапом інженерного аналізу. MATLAB використовує підхід, заснований на об'єктах. Кожен елемент графіка (лінія, вісь, заголовок, сама фігура) є об'єктом, що має набір властивостей (Properties). Цю систему називають Handle Graphics. Кожен об'єкт має унікальний ідентифікатор (Handle), через який можна змінювати його вигляд.

Ієрархія об'єктів:

Root (Екран комп'ютера) -> Figure (Вікно) -> Axes (Система координат) -> Line, Surface, Text (Примітиви).

Двовимірна графіка (2D)

Основною командою є `plot(x, y)`. Вона будує ламану лінію, що з'єднує точки з координатами (x_i, y_i) . Чим менший крок зміни аргумента x , тим гладшою виглядає крива.

Стилізація:

`plot(x, y, 'StringSpec')`, де *StringSpec* – рядок форматування, що визначає колір, тип маркера і стиль лінії.

Кольори: r (red), g (green), b (blue), k (black), w (white), y (yellow), m (magenta), c (cyan).

Маркери: o (коло), * (зірочка), . (точка), x (хрестик), s (квадрат), d (ромб).

Лінії: - (суцільна), -- (штрихова), : (точкова), -. (штрих-пунктирна).

Приклад: `plot(x, y, 'r--o')` побудує червону штрихову лінію з круглими маркерами у вузлах.

Анотації:

- `title('Text')`: заголовок графіка;
- `xlabel('Text')`, `ylabel('Text')`: підписи осей. Підтримується LaTeX-форматування;
- `legend('Line1', 'Line2')`: легенда;
- `grid on/off`: керування сіткою;
- `axis([xmin xmax ymin ymax])`: жорстке задавання меж відображення.

Кілька графіків

Для відображення кількох кривих на одних осях використовують команду *hold on*, яка «заморожує» поточний графік, даючи змогу додавати нові. Команда *figure* створює нове вікно.

Команда `subplot(m, n, p)` розбиває вікно на матрицю графіків $m \times n$ і робить активним p -ту область.

Тривимірний графік (3D)

Для побудови поверхонь $z = f(x, y)$ необхідно спочатку створити координатну сітку на площині XY. Для цього служить функція *meshgrid*.

- 1 Створюємо вектори діапазонів x та y .
- 2 `= meshgrid(x, y)`: створює матриці координат вузлів сітки.
- 3 `Z = f(X, Y)`: обчислюємо висоту поверхні в кожному вузлі (використовуючи поелементні операції!).
- 4 Візуалізація:
 - `plot3(x, y, z)`: 3D-лінія;

- mesh(X, Y, Z): каркасна поверхня («сітка рабиця»). Колір ліній залежить від висоти Z;
- surf(X, Y, Z): поверхня, залита кольором. Виглядає як тверде тіло;
- contour(X, Y, Z): карта ліній рівня (топографічна карта).

3.2 Програма роботи

Крок 1. 2D-графік.

Побудуємо графік затухаючих коливань $y = e^{-0.5t} \sin(5t)$.

Matlab

```
t = 0:0.05:10; % Час від 0 до 10 с
y = exp(-0.5*t).* sin(5*t); % Увага на.*

figure(1);
plot(t, y, 'b-o', 'LineWidth', 1.5, 'MarkerSize', 4);
grid on;
title('Затухаючий процес');
xlabel('Час t, c');
ylabel('Амплітуда Y(t)');
```

Крок 2. Робота з Subplot.

Matlab

```
x = -pi:0.1:pi;
figure(2);
subplot(2, 1, 1); % Верхній графік
plot(x, sin(x)); title('Синус');

subplot(2, 1, 2); % Нижній графік
plot(x, cos(x), 'r'); title('Косинус');
```

Крок 3. 3D-поверхня.

Побудуємо «мексиканський капелюх» (функція sinc).

Matlab

```
= meshgrid(-8:0.5:8);
```

```

R = sqrt(X.^2 + Y.^2) + eps; % +eps щоб уникнути ділення на 0
Z = sin(R)./ R;

figure(3);
surf(X, Y, Z);
shading interp; % Згладжування кольорів (прибирає сітку)
colormap jet; % Колірна схема
colorbar; % Шкала праворуч
title('Функція sinc(r)');

```

3.3 Індивідуальні завдання

1 Побудувати графік функції $y = f(x)$ на вказаному інтервалі (2D).
Додати сітку, заголовок і підписи осей.

2 Побудувати поверхню $z = f(x, y)$ у вказаній області (3D).
Використати кольорову карту (colormap) і панель кольорів (colorbar)
(таблиця 3.1).

Контрольні запитання

- 1 Яке призначення функції meshgrid?
- 2 У чому різниця між функціями plot і plot3?
- 3 Як змінити товщину лінії та розмір маркера на графіку?
- 4 Як побудувати декілька графіків в одному вікні, не використовуючи subplot?
- 5 Що робить команда hold on?
- 6 Як експортувати графік у файл із високою роздільною здатністю?
- 7 У чому візуальна різниця між mesh і surf?

Таблиця 3.1

Варіант	Функція 2D (інтервал x)	Функція 3D (область x, y)
1	$y = x^2 \sin(x), x \in [-2\pi, 2\pi]$	$z = \sin(x) \cos(y), x, y \in [-3, 3]$
2	$y = e^{-0.5x} \cos(2x), x \in [0, 10]$	$z = x^2 - y^2, x, y \in [-2, 2]$
3	$y = \frac{1}{1 + x^2}, x \in [-5, 5]$	$z = x e^{-x^2 - y^2}, x, y \in [-2, 2]$
4	$y = \arctan(x) + x, x \in [-10, 10]$	$z = \cos(\sqrt{x^2 + y^2}), x, y \in [-10, 10]$
5	$y = \sqrt{x} \ln(x), x \in [0.1, 5]$	$z = \frac{\sin(r)}{r}, r = \sqrt{x^2 + y^2}, x, y \in [-8, 8]$
6	$y = x^3 - 3x + 2, x \in [-3, 3]$	$z = \frac{1}{1 + x^2 + y^2}, x, y \in [-3, 3]$
7	$y = \sin(x) + \sin(3x), x \in [0, 2\pi]$	$z = \arctan(x \cdot y), x, y \in [-2, 2]$
8	$y = e^{-x^2}, x \in [-3, 3]$	$z = \cos(x^2 + y^2), x, y \in [-2, 2]$
9	$y = \frac{\sin(x)}{x}, x \in [-15, 15]$	$z = \sin(x) + \sin(y), x, y \in [-5, 5]$
10	$y = \cosh(x), x \in [-2, 2]$	$z = \sqrt{16 - x^2 - y^2}, x, y \in [-3, 3]$
11	$y = \ln(x^2 + 1) \cos(x), x \in [-5, 5]$	$z = (1 - x^2)(1 - y^2), x, y \in [-1.5, 1.5]$
12	$y = x - \frac{x^3}{6} + \frac{x^5}{120}, x \in [-4, 4]$	$z = x^2 + y^2 + \sin(x + y), x, y \in [-3, 3]$
13	$y = e^{-0.5x} \cos(2x), x \in [1, 15]$	$z = \frac{1}{1 + x^2 + y^2}, x, y \in [-4, 4]$
14	$y = 2^{-x} \sin(10x), x \in [0, 5]$	$z = \cos(x^2 + y^2), x, y \in [-2, 2]$
15	$y = \frac{x}{x^2 + 1}, x \in [-5, 5]$	$z = x^2 + y^2 + \sin(x + y), x, y \in [-3, 3]$

Практична робота 4. Обробка експериментальних даних. Інтерполяція та апроксимація

Мета роботи

Ознайомитися з методами чисельного аналізу дискретних даних. Навчитися виконувати інтерполяцію (знаходження проміжних значень) і апроксимацію (пошук аналітичної залежності) експериментальних даних засобами MATLAB. Вивчити роботу з поліномами та метод найменших квадратів.

4.1 Теоретичні відомості

Поліноми в MATLAB

У системі MATLAB поліном $P(x) = a_n x^n + \dots + a_1 x + a_0$ подано як вектор його коефіцієнтів: $p = [a_n \dots a_1 a_0]$.

Наприклад, $y = 2x^2 - 4x + 5$ записують як $p = [2 \ -4 \ 5]$.

Функції для роботи з поліномами:

- `polyval(p, x)`: обчислює значення полінома в точках x (за схемою Горнера);
- `roots(p)`: знаходить корені рівняння $P(x) = 0$;
- `polyfit(x, y, N)`: апроксимує дані поліномом степеня N методом найменших квадратів (МНК). Повертає коефіцієнти полінома.

Інтерполяція

Інтерполяція передбачає, що шукана крива проходить точно через задані точки даних.

Функція `interp1(x, y, xi, 'method')` виконує одновимірну інтерполяцію для нових точок xi .

Основні методи:

- `'linear'` з'єднує точки прямими лініями (ламаний графік);

- 'spline' кубічний сплайн. Забезпечує гладкість першої та другої похідних;
- 'pchip' кубічна інтерполяція, що зберігає монотонність (не дає викидів, як сплайн).

Приклад: інтерполяція даних.

Завдання. Маємо набір експериментальних точок (температура двигуна залежно від часу). Необхідно дізнатися температуру в проміжні моменти часу, використовуючи лінійну та сплайн-інтерполяцію.

Код програми:

```
% 1. Задання вихідних дискретних даних
t = [0, 10, 20, 30, 40, 50, 60]; % Час (с)
T = [20, 45, 70, 85, 92, 95, 98]; % Температура (град. С)

% 2. Створення густої сітки для плавного відображення
ti = 0 : 0.5 : 60; % Крок 0.5 с

% 3. Виконання інтерполяції різними методами
Ti_linear = interp1(t, T, ti, 'linear'); % Лінійна (ламані лінії)
Ti_spline = interp1(t, T, ti, 'spline'); % Кубічний сплайн (гладка крива)

% 4. Візуалізація результатів
figure('Name', 'Порівняння методів інтерполяції');
plot(t, T, 'ko', 'MarkerSize', 8, 'MarkerFaceColor', 'k'); % Вихідні точки
hold on;
plot(ti, Ti_linear, 'b--', 'LineWidth', 1.5); % Лінійна
plot(ti, Ti_spline, 'r-', 'LineWidth', 2); % Сплайн
grid on;
legend('Експериментальні точки', 'Лінійна інтерполяція', 'Сплайн-інтерполяція');
xlabel('Час t, с');
ylabel('Температура T, ^oC');
title('Порівняння методів інтерполяції');
```

Аналіз. Лінійна інтерполяція просто з'єднує точки відрізками. Сплайн будує гладку криву, що проходить через усі точки, і краще моделює фізичні процеси (наприклад нагрівання), де немає різких зламів.

Апроксимація (Regression)

Апроксимацію використовують, коли дані містять шум (похибки вимірювань). Задача полягає не в тому, щоб пройти через кожну точку, а в

тому, щоб провести криву, яка мінімізує суму квадратів відхилень від точок даних. Це роблять за допомогою функції `polyfit`.

Приклад: апроксимація (регресія).

Завдання. Отримано дані з шумом (похибками вимірювання). Побудувати лінію тренду (лінійна регресія) і квадратичну залежність, що найкраще описують дані.

Код програми:

```
% 1. Дані з випадковим шумом
x = 1:10;
y = [2.1, 3.8, 6.5, 9.2, 11.5, 14.8, 16.5, 19.8, 22.1, 24.5];

% 2. Знаходження коефіцієнтів поліномів (Метод найменших квадратів)
% Поліном 1-го степеня (пряма):  $y = ax + b$ 
p1 = polyfit(x, y, 1);

% Поліном 2-го степеня (парабола):  $y = ax^2 + bx + c$ 
p2 = polyfit(x, y, 2);

% 3. Обчислення значень поліномів для відображення
xi = 1 : 0.1 : 10;
y_fit1 = polyval(p1, xi); % Значення прямої
y_fit2 = polyval(p2, xi); % Значення параболи

% 4. Візуалізація
figure('Name', 'Апроксимація даних');
plot(x, y, 'ko', 'MarkerSize', 8, 'LineWidth', 2); % Точки
hold on;
plot(xi, y_fit1, 'b--', 'LineWidth', 2); % Пряма
plot(xi, y_fit2, 'r-', 'LineWidth', 2); % Парабола
grid on;
legend('Дані', ['Лінійна:  $y =$ ' num2str(p1(1)) '  $x +$ ' num2str(p1(2))], ...
      'Квадратична');
title('Апроксимація методом найменших квадратів');

% 5. Оцінка похибки (для лінійної моделі)
y_calc = polyval(p1, x);
MSE = mean((y - y_calc).^2); % Середньоквадратична похибка
disp(['Середньоквадратична похибка (MSE) лінійної моделі: ', num2str(MSE)]);
```

4.2 Порядок виконання

- 1 Задати вектори X (аргумент) і Y (виміряні значення).
- 2 Побудувати графік вихідних точок маркерами (без ліній).
- 3 Створити густу сітку точок x_i для відображення гладкої кривої.

- 4 Виконати сплайн-інтерполяцію (`interp1`) і відобразити її на графіку.
- 5 Виконати лінійну ($N = 1$) і квадратичну ($N = 2$) апроксимації (`polyfit`). Побудувати отримані поліноми.
- 6 Порівняти результати візуально та зробити висновки.

4.3 Індивідуальні завдання

- 1 Згенерувати вектори X та Y за вказаною формулою на інтервалі $[a, b]$ із кроком $h = 1.0$ (кількість точок буде невеликою, 7-12 точок).
- 2 Виконати інтерполяцію (методом сплайну) для «густої» сітки (крок 0.1).
- 3 Виконати апроксимацію поліномом першого степеня (лінійна регресія) і поліномом того степеня N , що вказана в завданні (таблиця 4.1).
- 4 Побудувати на одному графіку: вихідні точки (маркерами), криву інтерполяції (лінією одного кольору) і криві апроксимації (лініями інших кольорів).

Контрольні запитання

- 1 Поясніть принципову **різницю** між задачами інтерполяції та апроксимації.
- 2 Опишіть призначення функції **polyfit** і зміст її вхідних і вихідних параметрів.
- 3 Як впливає **ступінь полінома** на точність апроксимації та ризик виникнення осциляцій?
- 4 У чому полягає перевага **сплайн-інтерполяції** порівняно з лінійною інтерполяцією?
- 5 Поясніть сутність **методу найменших квадратів** для обробки експериментальних даних.

Таблиця 4.1

Варіант	Діапазон $x \in [a, b]$	Формула для генерації даних Y (додати шум можна як $+ 0.5 * \text{rand}(\text{size}(x))$)	Степінь полінома апроксимації N (крім 1-го)
1	[0, 10]	$y = 0.5x + 2 + \sin(x)$	3
2	[1, 8]	$y = \ln(x) + 0.2x^2$	2
3	[-5, 5]	$y = \tanh(x) + 0.1x$	3
4	[0, 6]	$y = e^{0.5x} - x$	2
5	[0, 3π]	$y = \cos(x) + 0.5x$	3
6	[-2, 4]	$y = x^2 - 3x + \text{rand}$	2
7	[-2, 5]	$y = 0.5x^3 - 2x^2 + 1$	3
8	[0, 5]	$y = 10 / (1 + x^2)$	3
9	[-3, 3]	$y = x^3 - 4x$	3
10	[0, 10]	$y = 2 - e^{-0.5x}$	2
11	[2, 12]	$y = 0.1x^2 + \sin(2x)$	2
12	[0, 2π]	$y = \sin(x) \cos(x)$	4
13	[-4, 4]	$y = x + 0.2x^2$	2
14	[1, 7]	$y = 1/x + x$	3
15	[0, 4]	$y = x e^{-x}$	3

Практична робота 5. Символьні обчислення (symbolic math toolbox)

Мета роботи

Ознайомитися з можливостями аналітичних (символьних) обчислень у MATLAB. На відміну від чисельних методів, що дають наближений результат, символьні методи дають змогу отримувати точні формули.

5.1 Теоретичні відомості

Symbolic Math Toolbox базований на ядрі MuPAD. Для роботи з символами необхідно оголосити символьні змінні командою `syms`.

Основні функції:

- `diff(f)`: диференціювання;
- `int(f)`: інтегрування (невизначене та визначене);
- `limit(f)`: границі;
- `solve(eq)`: розв'язання алгебраїчних рівнянь;
- `dsolve(eq)`: розв'язання диференціальних рівнянь;
- `subs(f, old, new)`: підстановка значень у формулу.

Приклад 1. Дослідження функції (похідні та границі).

Завдання. Знайти першу та другу похідні функції $f(x) = x \sin(x^2)$, а також границю за умови $x \rightarrow 0$.

Код програми:

```
syms x          % Оголошення символьної змінної
f = x * sin(x^2); % Задання функції

% 1. Диференціювання
df = diff(f, x); % Перша похідна
d2f = diff(f, x, 2); % Друга похідна

disp('Перша похідна:');
pretty(df)        % pretty робить вивід схожим на математичний запис

disp('Друга похідна:');
```

```
pretty(d2f)
```

```
% 2. Обчислення границі (наприклад, sin(x)/x при x->0)
g = sin(x)/x;
lim_g = limit(g, x, 0);
disp(['Границя sin(x)/x при x->0: ', char(lim_g)]);
```

Приклад 2. Інтегрування та розв'язання рівнянь.

Завдання. Знайти невизначений інтеграл $\int x e^x dx$, визначений інтеграл $\int_0^\pi \sin(x) dx$ і розв'язати квадратне рівняння $ax^2 + bx + c = 0$ відносно x .

Код програми:

```
syms x a b c
% 1. Інтегрування
int_indef = int(x * exp(x), x); % Невизначений інтеграл
int_def = int(sin(x), x, 0, pi); % Визначений інтеграл від 0 до Pi

disp('Невизначений інтеграл:');
pretty(int_indef)
disp(['Визначений інтеграл: ', char(int_def)]);

% 2. Розв'язання рівнянь
eqn = a*x^2 + b*x + c == 0; % Рівняння (зверніть увагу на ==)
roots_eq = solve(eqn, x); % Розв'язати відносно x

disp('Корені квадратного рівняння:');
pretty(roots_eq)
```

Приклад 3. Розв'язання диференціальних рівнянь (ODE).

Завдання. Знайти розв'язок рівняння $y' = -y + t$ з початковою умовою $y(0) = 1$.

Код програми:

```
syms y(t)
eqn = diff(y, t) == -y + t; % Рівняння y' = -y + t
cond = y(0) == 1; % Початкова умова

ySol(t) = dsolve(eqn, cond); % Знаходження розв'язку
disp('Розв'язок ДР:');
pretty(ySol)
```

```
% Побудова графіка розв'язку  
fplot(ySol, [0, 10]);  
grid on;  
title('Розв'язок  $y'' = -y + t$ ');
```

5.2 Порядок виконання

1 Оголошення змінних; використати команду `syms` для створення символьних змінних (наприклад x , y , a , b).

2 Диференціювання; знайти похідну заданої функції $f(x)$ за допомогою функції `diff`, результат спростити командою `simplify` або `pretty` (за потреби).

3 Інтегрування; обчислити інтеграл (визначений або невизначений) від вказаної функції, використовуючи команду `int` і вказуючи межі інтегрування для визначених інтегралів.

4 Розв'язання рівнянь; знайти корені алгебраїчного рівняння або розв'язок диференціального рівняння за допомогою `solve` або `dsolve` відповідно.

5 Візуалізація та аналіз; побудувати графіки отриманих функцій за допомогою `fplot` і проаналізувати результати; оформити звіт.

5.3 Індивідуальні завдання

Знайти похідну заданої функції:

- 1 Обчислити визначений або невизначений інтеграл.
- 2 Розв'язати алгебраїчне або диференціальне рівняння (таблиця 5.1).

Таблиця 5.1

Варіант	1 Знайти похідну функції $f(x)$	2 Знайти інтеграл	3 Розв'язати рівняння
1	$\frac{1+x}{1-x}$	$\int x \cos(x) dx$	$x^2 - 5x + 6 = 0$
2	$\sin^3(x) + \cos^3(x)$	$\int_0^1 e^{-x} dx$	$y' = 2y$
3	$e^x \cdot \ln(x)$	$\int \frac{dx}{x^2 + 1}$	$\sin(x) = 0.5$
4	$\sqrt{1+x^4}$	$\int_0^\pi \sin^2(x) dx$	$y'' + y = 0$
5	$\frac{\sin(x)}{x}$	$\int x^2 e^x dx$	$x^3 - 8 = 0$
6	$\arctan(e^x)$	$\int \ln(x) dx$	$y' + y = x$
7	$x^{\sin(x)}$	$\int_0^1 \sqrt{1-x^2} dx$	$e^x = 10$
8	$\ln(\sin(x))$	$\int \tan(x) dx$	$y'' - 4y = 0$
9	e^{-x^2}	$\int_1^2 \frac{1}{x} dx$	$x^2 + 2x + 5 = 0$
10	$\frac{x^2}{1+x}$	$\int x \sin(x^2) dx$	$y' = y x$
11	$2^x + x^2$	$\int_0^{\pi/2} \cos^3(x) dx$	$\ln(x) = 1$
12	$\arcsin(\sqrt{x})$	$\int \frac{x}{1+x^2} dx$	$y'' - 2y' + y = 0$
13	$(x+1)^5 \cdot (x-1)^3$	$\int e^{2x} \sin(x) dx$	$x^4 - 16 = 0$
14	$\ln(x + \sqrt{x^2 + 1})$	$\int_0^1 x e^{-x^2} dx$	$y' + \frac{y}{x} = 0$
15	$\cos(2x) \cdot e^x$	$\int \frac{1}{\sqrt{x}} dx$	$\tan(x) = 1$

Контрольні запитання

- 1 У чому полягає відмінність між **символьними** (аналітичними) і **чисельними** обчисленнями в MATLAB?
- 2 Яку команду використовують для **оголошення символічних змінних**?
- 3 Опишіть синтаксис функцій для знаходження **похідної** та **інтеграла** в символічному вигляді.
- 4 Як розв'язати **диференціальне рівняння** з початковими умовами за допомогою Symbolic Math Toolbox?
- 5 У який спосіб можна перетворити отриманий **символьний вираз** у функцію для чисельних розрахунків або побудови графіка?

Список літератури

- 1 Литвинов А. Л. Чисельні методи: теорія і практика: навч. посіб. Харків: ХНУМГ ім. О. М. Бекетова, 2022. 166 с.
- 2 Клочко Н. В. MATLAB. Практикум: навч. посіб. Харків: НТУ «ХПІ», 2020. 250 с.
- 3 Chapra S. C. Applied Numerical Methods with MATLAB for Engineers and Scientists. 4th Edition. New York: McGraw-Hill Education, 2018. 688 p.
- 4 Дорф Р., Бішоп Р. Сучасні системи керування. Вид. 13-те. Київ, 2020. 960 с.
- 5 Трохимчук С. М. Основи роботи в системі MATLAB: навч. посіб. Київ: КПІ ім. Ігоря Сікорського, 2019. 180 с.

ВИКОРИСТАННЯ MATLAB ДЛЯ ІНЖЕНЕРНИХ РОЗРАХУНКІВ

Методичні вказівки
для виконання практичних робіт, самостійної роботи
з освітньої компоненти
*«МЕТОДИ ТА ПРОГРАМНО-ТЕХНІЧНІ ЗАСОБИ
ІНЖЕНЕРНИХ РОЗРАХУНКІВ»*

Відповідальний за випуск Щебликіна О. В.

Редактор Ібрагімова Н. В.

Підписано до друку 16.01.2026 р.

Умовн. друк. арк. 2,0. Тираж . Замовлення № .

Видавець та виготовлювач Український державний університет
залізничного транспорту,
61050, Харків-50, майдан Фейєрбаха, 7.

Свідоцтво суб'єкта видавничої справи ДК № 6100 від 21.03.2018 р.