МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Д.О. Півторак, Ю.Ф. Лазарєв, С.Л. Лакоза

Комп'ютерне моделювання процесів і систем

Практикум

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра за освітньою програмою «Комп'ютерно-інтегровані технології та системи навігації і керування»
спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»,*

Київ
КПІ ім. Ігоря Сікорського
2021

Рецензенти:	*Семінський О.О.* — канд. техн. наук, доцент, доцент кафедри машин та апаратів хімічних та нафтопереробних виробництв КПІ ім. Ігоря Сікорського
	Вислоух С.П.. — канд. техн. наук, доцент, доцент кафедри виробництва приладів КПІ ім. Ігоря Сікорського
Відповідальний редактор:	*Бурау Н. І.,* док. техн. наук, професор кафедри приладів і систем орієнтації і навігації КПІ ім. Ігоря Сікорського

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 5 від 14.01.2021 р.) за поданням Вченої ради Приладобудівного факультету (протокол № 10/20 від 28.12.2020 р.)

Електронне мережне навчальне видання

Півторак Діана Олександрівна, канд. техн. наук, доц.
Лазарєв Юрій Федорович, канд. техн. наук, доц.
Лакоза Сергій Леонідович, канд. техн. наук

КОМП’ЮТЕРНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ І СИСТЕМ

Практикум

Комп’ютерне моделювання процесів і систем. Практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 151 «Автоматизація та комп’ютерно-інтегровані технології» / Д.О. Півторак, Ю.Ф. Лазарєв, С.Л. Лакоза ; КПІ ім. Ігоря Сікорського, 2021. - 207 с.

Навчальний посібник містить стислі теоретичні відомості з векторних та матричних операцій у MatLab, методів розв’язування лінійних та нелінійних алгебраїчних рівнянь. У посібнику детально розглянуті можливості побудови графіків функцій та їх візуального оформлення, чисельного інтегрування функцій та звичайних диференціальних рівнянь, приведено опис базових бібліотек візуального середовища моделювання Simulink. Теоретичні положення продемонстровані на прикладах коду програм та блок-схем у Simulink. Кожен комп’ютерний практикум завершується переліком завдань для самостійної роботи студента та питаннями для контролю вивченого матеріалу. Виконання практичних завдань, пов’язаних з застосуванням чисельних методів та засобів системи MatLab для розв’язку прикладних завдань комп’ютерного моделювання сприятиме закріпленню, поглибленню та узагальненню теоретичних основ курсу, а також сприятиме розвитку навичок самостійної творчої роботи студентів у процесі їх навчання, зокрема при виконанні контрольних, комп’ютерних практикумів, розрахункових та дипломних робіт, при розробці алгоритмів, для моделювання та дослідження комп’ютерно-інтегрованих систем.

ЗМІСТ

ВСТУП

Інженеру та досліднику у своїй роботі потрібно працювати з об’єктами та системами різної складності, кожна з яких описується багатьма характеристиками: від опису зовнішнього вигляду до підбору оптимальних параметрів алгоритмів та чутливих елементів. У цьому випадку доводиться працювати з моделями різного роду, який описують реальний об’єкт з точки зору різних підходів та областей застосування. Дані описи називаються моделі і в техніці найбільше поширення здобули математичні моделі, які дозволяють описати об’єкт не тільки якісно, а й кількісно. Маючи ці моделі, дослідник може проводити аналіз об’єкта чи алгоритму та виконати підбір найкращих параметрів для функціонування процесів і систем.

Сучасні моделі процесів і систем вже давно мають високий рівень складності та деталізації. Виконання їх аналізу та вдосконалення з використанням аналітичних підходів носить проблематичний характер, коли отримати аналітичний розв’язок буває неможливо. У цьому випадку застосовуються різноманітні чисельно-аналітичні та чисельні методи. Для реалізації цих методів застосовують автоматизовані комп’ютерні системи та комплекси. Тому для сучасного інженера та дослідника необхідно майстерно користуватися сучасними пакетами для комп’ютерного моделювання процесів і систем, а також розуміти підходи та алгоритми в них закладені.

У даному посібнику для вирішення вищеописаних задач застосовується комп’ютерна система MatLab. Мільйони інженерів і вчених в усьому світі використовують MatLab, щоб аналізувати і проектувати складні системи. Комп’ютерна система Matlab наразі широко розповсюджена в інженерних і університетських колах завдяки багатьом перевагам, до яких відносяться:

1) простота опановування і доступність текстів практично усіх програмних засобів, окрім вбудованих;

2) велика бібліотека досяжних математичних програм, яка містить майже всі сучасні чисельні методи і функції;

3) можливість утворювати власні програмні засоби і навіть корегувати існуючі;

4) вельми зручний і пристосований для практичних потреб інженерів і науковців апарат графічного оформлення результатів обчислень;

5) наявність інтегрованого пакету програм Simulink візуального програмування, який дозволяє суттєво спростити процес утворення складних

програм і автоматизувати процес організації чисельного інтегрування диференційних рівнянь досліджуваної системи.

Важливо відмітити, що код, розроблений в MatLab, може бути інтегрований з іншими мовами та системами програмування, дозволяючи розгортати алгоритми та програми в мережі на підприємствах і в промислових системах.

КОМП’ЮТЕРНИЙ ПРАКТИКУМ № 1
ПОЧАТОК ОБЧИСЛЕНЬ У MATLAB

Мета роботи: ознайомитися з програмним забезпеченням MatLab та навчитися працювати в режимі безпосередніх обчислень (режимі наукового калькулятора).

1.1. Теоретичні відомості

1.1.1. Обчислення з дійсними числами. Введення дійсних чисел

Введення чисел з клавіатури здійснюється за загальними правилами, що прийняті для мов програмування високого рівня:

- для виділення дробової частини мантиси числа застосовується десяткова крапка (замість коми у звичайному записі);
- десятковий показник числа записується у вигляді цілого числа після попереднього запису символу «е»;
- між записом мантиси числа та символом «е» (який відділяє її від показника) не повинно бути ніяких символів, включаючи і символ пропуску.

Наприклад, якщо ввести у командне вікно MatLab рядок

2.4567e-12,

то після натискання клавіші <Enter> в цьому вікні з’явиться запис (рис.1.1):

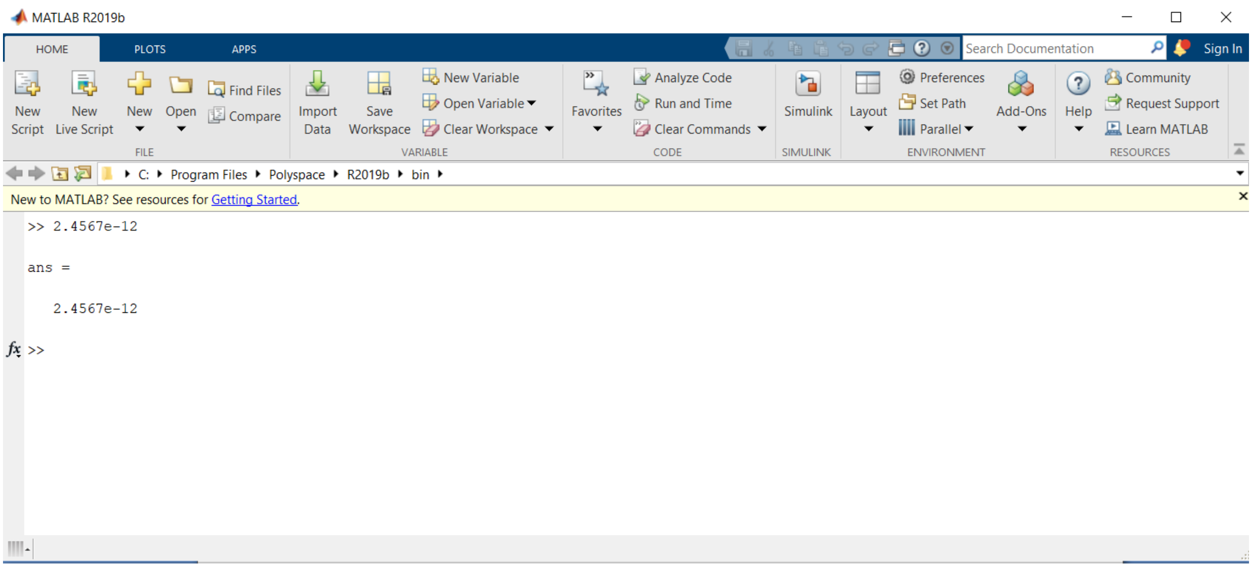


Рис. 1.1 Командне вікно MATLAB

Результат виводиться у командне вікно (**Command Winodw**) у вигляді (форматі), що визначається попередньо встановленими налаштуваннями (за допомогою розділу «Options» меню командного вікна).

Введені числа і результати усіх обчислень у системі MatLab зберігаються у пам’яті ПК з відносною похибкою близько 1e-17 (тобто з точними значеннями у 16 десяткових розрядах).

Діапазон подання модуля дійсних чисел лежить у проміжку між 1e-308 і 1e308.

1.1.2. Найпростіші арифметичні дії

У арифметичних виразах мови MatLab використовуються наступні знаки арифметичних операцій: «+» – додавання; «-» – віднімання; «*» – множення; «/» – ділення зліва направо; «\» – ділення справа наліво; «^» – піднесення до степеня.

Застосування MatLab у режимі калькулятора може здійснюватися шляхом простого запису у командний рядок послідовності арифметичних дій з числами, тобто звичайного арифметичного виразу.

Наприклад, вираз

$$(10+5\cdot 10^2)\cdot\frac{4}{5}+(54\cdot 2-35)^2$$

в командному вікні буде мати наступну послідовність арифметичних дій

(10+5.3e2)*4/5+(54*2-35)^2.

Якщо після введення з клавіатури цієї послідовності натиснути клавішу <Enter>, у командному вікні з’явиться запис (рис.1.2):

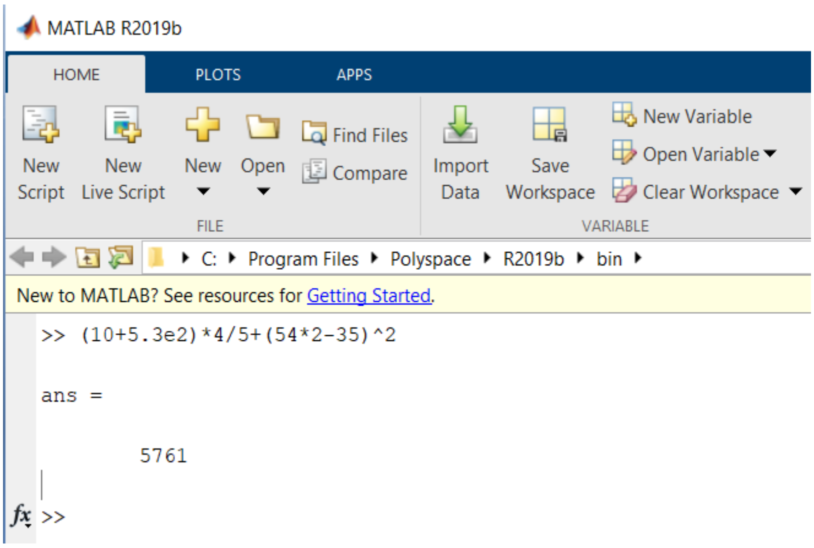


Рис. 1.2 Приклад використання режиму калькулятора

Тобто на екрані виводиться результат дії останнього виконаного оператора під іменем системної змінної ***ans*** (рис. 1.3).

Взагалі виведення проміжної інформації у командне вікно **<u>підпорядковується таким правилам</u>**:

- якщо запис останнього оператора не закінчується символом «;», то результат дії цього оператора одразу ж виводиться до командного вікна;

- якщо оператор закінчується символом «;», то результат його дії не відображається у командному вікні (рис.1.4);

- якщо оператор не містить знака присвоєння, тобто є просто записом деякої послідовності дій з числами і змінними, то значення результату надається спеціальній системній змінній, що має ім’я «***ans***»;

- отримане значення можна використовувати у наступних операторах обчислень під цим ім’ям ***ans***; при цьому слід пам’ятати, що значення системної змінної ***ans*** змінюється після дії чергового оператора без знака присвоєння;

- загальний формат виведення до командного вікна має вигляд:

<ім’я змінної>=<результат>.

Особливістю роботи MatLab, як калькулятора, є можливість використання імен змінних для запису у пам'ять ПК проміжних результатів. Для цього використовується операція присвоєння, яка вводиться знаком рівності «=» згідно з виразом:

<ім’я змінної>=<вираз>[;]

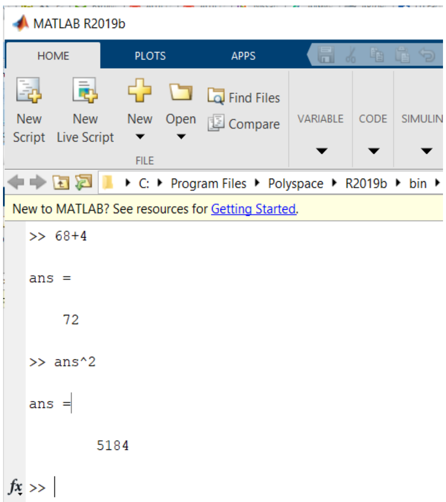


Рис. 1.3 Приклад використання змінної ***ans***

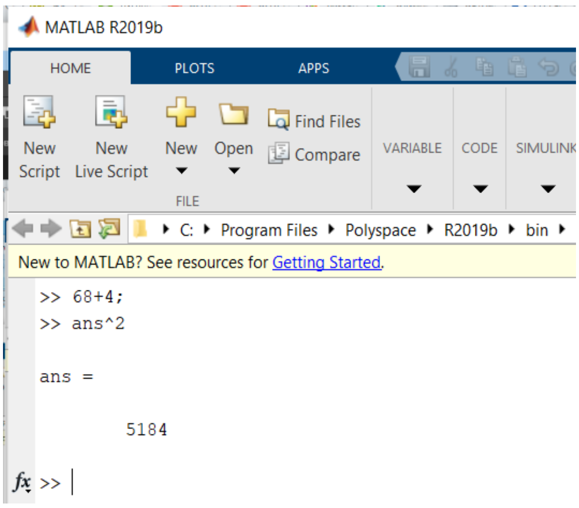


Рис. 1.4 Приклад використання символу «;»

Ім’я змінної може містити до 19 символів і не повинно збігатися з іменами функцій і процедур систем та системних змінних. **При цьому система розрізняє змінні з великими та малими літерами!!!**

Вираз праворуч від символу присвоювання може бути просто числом, арифметичним виразом, рядком символів (тоді ці символи треба взяти в апострофи) або символьним виразом. Якщо вираз не закінчується символом

«;», після натискання клавіші <Enter> у командному вікні з’явиться результат виконання у вигляді:

<ім’я змінної>=<результат>.

Наприклад, якщо ввести у командне вікно рядок «a = 57+24», на екрані з’явиться запис (рис. 1.5):

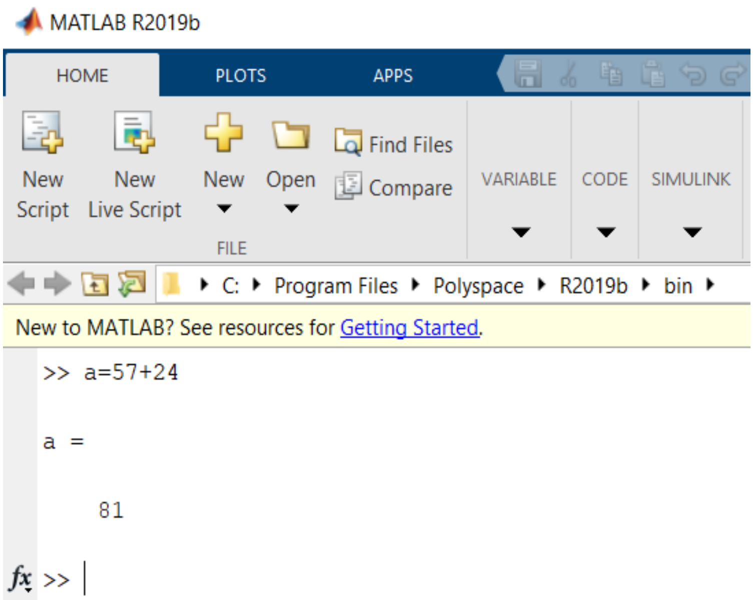


Рис. 1.5 Приклад виконання операції присвоєння («=»)

Система MatLab має декілька імен змінних, що використовуються самою системою і входять до складу зарезервованих:

pi – число π (зберігається у вигляді 3.141592653589793);

inf – значення машинної нескінченності;

NaN – позначення операції невизначеності (наприклад, типу 0/0 або inf/inf);

ans – результат останньої операції без операції присвоєння.

1.1.3. Елементарні математичні функції

Загальна форма використання функції є такою:

<ім’я результату> = <ім’я функції> (<список імен аргументів або їх значень>).

У мові MatLab передбачені такі елементарні арифметичні функції.

Тригонометричні і гіперболічні функції:

sin(Z) – синус числа Z;

sinh(Z) – гіперболічний синус;

asin(Z) – арксинус (у радіанах, у діапазоні від -π /2 до +π /2);

asinh(Z) – обернений гіперболічний синус;

cos(Z) – косинус;

cosh (Z) – гіперболічний косинус;

acos(Z) – арккосинус (у диапазоні від 0 до π);

acosh (Z) – обернений гіперболічний косинус;

tan(Z) – тангенс;

tanh (Z) – гіперболічний тангенс;

atan(Z) – арктангенс (у діапазоні від -π /2 до +π /2);

atan2(X,Y) – чотири квадрантний арктангенс (кут у діапазоні від -π до +π між горизонтальним променем і променем, що проходить крізь точку з координатами X і Y);

atanh (Z) – обернений гіперболічний тангенс;

sec(Z) – секанс;

sech(Z) – гіперболічний секанс;

asec(Z) – арксеканс;

asech(Z) – обернений гіперболічний секанс;

cscd(Z) – косеканс;

csch(Z) – гіперболічний косеканс;

acsc(Z) – арккосеканс;

acsch(Z) – обернений гіперболічний косеканс;

cot(Z) – котангенс;

coth(Z) – гіперболічний котангенс;

acot(Z) – арккотангенс;

acoth(Z) – обернений гіперболічний котангенс.

Слід зауважити, що значення змінних в тригонометричних і гіперболічних функціях вводяться в радіанах.

Експоненціальні функції:

exp(Z) – експонента числа Z;

log(Z) – натуральний логарифм;

log10(Z) – десятковий логарифм;

sqrt(Z) – обчислення квадратного кореня з числа Z;

abs(Z) – обчислення модуля числа Z.

Цілочислові функції:

fix(Z) – округлення до найближчого цілого у бік нуля;

floor(Z) – округлення до найближчого цілого у бік від’ємної нескінченності;

ceil(Z) – округлення до найближчого цілого у бік додатної нескінченності;

round(Z) – звичайне округлення числа Z до найближчого цілого;

rem(X,Y) – обчислення остачі від ділення X на Y;

sign(Z) – обчислення сігнум-функції числа Z (0 при Z=0, -1 при Z<0, 1 при Z>0).

1.1.4. Формати відображення даних

У MatLab можна працювати з різними форматами представлення чисел. За замовчуванням MatLab виводить числові значення з чотирма цифрами після коми та однією до, в так званій нормалізованій формі. Для встановлення відповідного формату використовується функція

format *<службова назва формату>.*

Назва форматів з коротким описом форматів представлена в табл. 1.1.

Таблиця 1.1

Формати представлення даних

short	З фіксованою крапкою та 4 знаками після крапки (за замовченням)
short e	Наукова нотація з 4 знаками після коми
Long	З фіксованою крапкою та 14 знаками після крапки
Long e	Наукова нотація з 15 десятковими знаками
Short g *Long g*	Гібрид між відповідними форматами виводу з фіксованою та плаваючою комою
Hex	Вивід числових даних в шістнадцятковому форматі
Bank	Зберігає в дробовій частині числа 2 знаки, відповідно дрібним грошовим одиницям
+	Режим обов’язкового виводу числа перед знаком, навіть якщо це +
rational	Число відображається у вигляді відношення цілих чисел

1.2. Завдання до виконання комп’ютерного практикуму

Завдання 1. Обчисліть арифметичний вираз відповідно до варіанту згідно з табл.1.2.

Таблиця 1.2

Варіанти індивідуальних завдань

Варіант	**Вираз**
1	2
1.	$\dfrac{\left(15\frac{5}{8}+14{,}025\right)\cdot\frac{8}{11}}{13\frac{5}{6}-12\frac{3}{5}\div 0{,}12}\cdot 5\frac{4}{5}$
2.	$\dfrac{5\cdot 4^3}{14\frac{5}{8}+12\frac{2}{3}\cdot 0{,}18-12{,}18}$
3.	$\dfrac{0{,}04+18\div 0{,}1-13\frac{2}{3}}{12\frac{8}{11}\cdot\left(\frac{4}{7}\div\frac{8}{11}+\left(3\frac{1}{3}\right)^2\right)}$
4.	$\dfrac{\left(18\frac{4}{7}-4\frac{5}{8}\right)\cdot 0{,}25}{(0{,}15\cdot 0{,}8+0{,}23\cdot 3^3)\cdot 0{,}5}$
5.	$\dfrac{(25-15^2)\cdot(0{,}75+14\cdot 0{,}13^2)}{8\frac{6}{9}+25\frac{4}{11}-0{,}18\cdot 5}$
6.	$\dfrac{18{,}75+14\frac{3}{5}-\left(14\frac{1}{3}-10{,}8\right)\cdot 0{,}2}{19\div\left(25\frac{1}{3}-13\frac{5}{6}+0{,}25\cdot 3^5\right)}$
7.	$\dfrac{14\frac{8}{11}-25\cdot 3^2+\frac{1}{4}\left(\frac{8}{9}-\frac{3}{4}\right)}{(18{,}75-16{,}15+14)\div 0{,}15}$
8.	$\dfrac{2500+(14\cdot 28+13{,}25)^3}{281\frac{1}{3}+\left(12\frac{1}{4}+125\frac{2}{3}\right)\cdot 0{,}2}$

1	2
9.	$\dfrac{268 \cdot 0{,}25 \div 0{,}1 + 15^2}{\left(17\frac{2}{3} - 12\frac{5}{8} \cdot 14\right) \div 0{,}2} \cdot \dfrac{1}{8}$
10.	$\dfrac{\left(19\frac{1}{3} + 12\frac{1}{6}\right) \cdot 4{,}1}{25\frac{6}{11} + 18\frac{6}{8} - 28\frac{2}{3}} \div \dfrac{2}{3}$
11.	$\dfrac{28\frac{4}{11} + 19\frac{2}{5} + 13{,}89}{(0{,}18^2 + 2500 \div 19{,}8) \cdot 14{,}1}$
12.	$\dfrac{444{,}4 \div 0{,}15 + 13{,}5}{\left(18\frac{2}{5} - 16{,}025 + 12\right) \cdot 8{,}2} \div 12{,}5$
13.	$\dfrac{8 \cdot 10^3 - 14 \cdot 10^2 + 13\frac{2}{5}}{\left(28\frac{4}{7} - 8\frac{6}{11} \div \frac{8}{15}\right) \cdot 0{,}15}$
14.	$\dfrac{26 \cdot 0{,}1 \div \frac{5}{6} - 14}{\left(-0{,}11 + 4\frac{3}{7} \cdot 2^2\right) \cdot \frac{6}{9} + 0{,}25}$
15.	$\dfrac{0{,}015 \cdot 10^5 + 28\frac{1}{3} - 8\frac{7}{8}}{\left(14\frac{9}{11} + 0{,}28 - \frac{4}{9}\right)^2}$
16.	$\dfrac{\left(15\frac{7}{9} - \frac{1}{3} + 0{,}14\right) \cdot \frac{6}{7}}{27\frac{8}{11} - \left(0{,}125 + 18\frac{6}{11} \cdot \frac{3}{7}\right)} \div 0{,}1$
17.	$\dfrac{\left(19\frac{8}{9} + 11\frac{4}{5} \cdot 0{,}29\right) \div 0{,}14}{\left(14\frac{8}{11} \cdot 3^3 - 10{,}75\right) \cdot 0{,}15}$
18.	$\dfrac{0{,}15 \cdot 10^3 + 9\frac{9}{11} - 5{,}05}{\left(19\frac{3}{7} - 14{,}01 \div 2^2\right) \cdot 0{,}1 + 17}$

1	2
19.	$\dfrac{28{,}75-\left(17\frac{8}{11}-\frac{4}{15}\cdot 2\right)\div 0{,}1}{\left(28\frac{4}{9}-18\cdot 3^2\right)\cdot 0{,}19+18\frac{4}{9}}$
20.	$\dfrac{39\frac{9}{14}+7\frac{4}{9}-6\cdot 0{,}28^2}{16\frac{8}{15}+3\frac{3}{14}-0{,}05}\cdot 1{,}47$
21.	$\dfrac{(89{,}98-15{,}78)\cdot 0{,}2^2+12}{14\frac{6}{9}-8\frac{5}{11}+29\cdot 0{,}1^2}$
22.	$\dfrac{29\frac{4}{11}-2{,}75+14{,}5^2}{84\frac{2}{5}\cdot 0{,}25-28\frac{7}{11}}\cdot\dfrac{2}{3}$
23.	$\dfrac{\left(14\frac{8}{9}-4{,}87+12{,}3\right)\cdot\frac{2}{3}}{\left(4\frac{8}{9}-14\frac{4}{9}\cdot 0{,}025\right)\div 0{,}11}$
24.	$\dfrac{(1086+0{,}14)\cdot 0{,}11^2-11\frac{6}{11}}{\left(29\frac{8}{11}-25\cdot 2^2\right)\cdot 0{,}18+14\frac{6}{7}}$
25.	$\dfrac{\left(29\frac{6}{7}-18\frac{3}{7}+5\right)\cdot\frac{4}{6}}{12\frac{3}{8}+4\frac{2}{7}-0{,}15^2}$
26.	$\dfrac{\left(43\frac{6}{7}-15\frac{1}{7}\right)\cdot\frac{9}{11}}{\left(29\frac{8}{11}-14\frac{7}{12}\right)\div 0{,}14}$
27.	$\dfrac{38\frac{5}{9}+19\frac{4}{8}-2{,}2}{19\frac{4}{7}-10{,}84\cdot 4^2+158\frac{3}{7}}\cdot\dfrac{2}{5}$
28.	$\dfrac{(47{,}5-29{,}45)\cdot 1{,}3^2}{\left(57\frac{4}{9}+12\frac{8}{11}\right)\cdot 0{,}27}\div 0{,}2$

1	2
29.	$\dfrac{(35758{,}75 - 21512{,}898) \cdot 0{,}1^2}{\left(89\frac{8}{11} + 14\frac{9}{13} - 3{,}75 \cdot 5{,}2\right)}$
30.	$\dfrac{\left(39\frac{8}{9} - 15\frac{9}{11} \cdot 0{,}5\right) \cdot 0{,}4^2}{\left(0{,}3 \cdot 0{,}9 + 15\frac{6}{7}\right) \cdot 26{,}5}$

Завдання 2. Проведіть обчислення по заданій формулі при заданих значеннях параметрів відповідно до варіанту згідно з табл.1.3.

Таблиця 1.3

Варіанти індивідуальних завдань

Варіант	**Вираз**	**Значення змінних**	
1	2	3	4
1.	$\dfrac{a \cdot \sqrt{b}}{2ab} \cdot (a + b^2)$	$a = 5 \cdot 10^3$ $b = 3{,}2 \cdot 10^2$	$a = -59{,}6$ $b = 29{,}6$
2.	$\dfrac{c + d}{\sqrt{c - \sqrt{d}}} \cdot (5c + 3d)$	$c = 159$ $d = 253{,}4$	$c = 34{,}9$ $d = 2 \cdot 10^2$
3.	$\dfrac{x^2 + 4x^3y^2 + y^5}{x}$	$x = 44{,}57$ $y = 0{,}4 \cdot 10^2$	$x = 18{,}5$ $y = -54{,}4$
4.	$\dfrac{14a\sqrt{b}}{5a + 4b + \sqrt{ab}}$	$a = 15 \cdot 10^2$ $b = 1{,}4 \cdot 10^3$	$a = 6 \cdot 10^{-2}$ $b = 17{,}5$
5.	$m^{\frac{1}{2}} + 2mn + \dfrac{4mn}{2m + 3n}$	$m = 254{,}6$ $n = 179{,}7$	$m = 85{,}3$ $n = -67{,}8$
6.	$\left(\sqrt{x^3 + 5} - 3\right) \cdot (x^4 + 3xy + 2y^3)$	$x = 147{,}8$ $y = 5{,}58$	$x = 57{,}9$ $y = -35{,}8$

1	2	3	4
7.	$\left(x^5y+\frac{y^3}{3}-1\right)\cdot\sqrt{xy}$	$x=26{,}8$ $y=69{,}5$	$x=15{,}82$ $y=27{,}9$
8.	$\frac{28\sqrt{a^3+3b^2}}{14\sqrt{ab}}$	$a=12\cdot10^3$ $b=7\cdot10^2$	$a=14{,}6$ $b=27{,}65$
9.	$(c^3-5d+c^5d^3)\cdot\frac{\sqrt{c^2+d^3}}{5}$	$c=15{,}9$ $d=34{,}6$	$c=-2{,}9$ $d=4{,}7$
10.	$\sqrt{m}+n^2\left(5\sqrt{mn}-14n\right)$	$m=25{,}6\cdot10^{-2}$ $n=24{,}96$	$m=78{,}3$ $n=35{,}9$
11.	$\left(x^4-\frac{\sqrt{xy}}{2}+y^3\right)\frac{\sqrt{x^2+3xy-4}}{18\sqrt[3]{xy}}$	$x=13{,}88$ $y=24{,}4\cdot10^{-2}$	$x=254$ $y=362$
12.	$\frac{\sqrt[5]{4x^5+8xy-y^3}}{\sqrt{2x+3xy+y^2}}$	$x=54{,}9$ $y=36{,}9$	$x=78{,}4$ $y=14{,}83$
13.	$14\sqrt{\frac{mn+2m^3-3n^2}{\sqrt[3]{4m+2n}}}$	$m=15{,}36$ $n=37{,}63$	$m=23{,}7$ $n=59{,}4$
14.	$\sqrt{\frac{4xy+x^2+2y^3}{\sqrt[5]{18x+12y-3xy}}}$	$x=54{,}7$ $y=26{,}42$	$x=14{,}8$ $y=17{,}9$
15.	$\frac{8(8m^3+12mn^2+2n^2)}{9m^5-\sqrt[5]{mn}+2m^2}$	$m=54{,}34$ $n=17{,}51$	$m=47{,}4$ $n=36{,}9$
16.	$\frac{18\div(0{,}05^a+b\cdot15{,}26)}{35\cdot a^4+\sqrt{ab}-b}$	$a=0{,}4\cdot10^4$ $b=2{,}5\cdot10^2$	$a=0{,}6$ $b=7\cdot10^{-2}$
17.	$\sqrt{\frac{89(15m+n\cdot3^m)}{46\sqrt{mn}\cdot(315+244{,}4)}}$	$m=36{,}8$ $n=24{,}9$	$m=47{,}9$ $n=53{,}7$

1	2	3	4
18.	$\dfrac{\left(14ab^{a}+28\sqrt[3]{a}+b^{6}\right)}{\sqrt{ab}}$	$a=124\cdot 10^{-3}$ $b=0{,}8\cdot 10^{2}$	$a=158{,}5$ $b=34{,}8$
19.	$\left(\dfrac{5}{a^{2}+5b+2}+\dfrac{3}{a^{2}+3b+8}\right)\sqrt{ab}$	$a=15\cdot 10^{2}$ $b=1{,}6\cdot 10^{3}$	$a=26{,}9$ $b=47{,}4$
20.	$\dfrac{\sqrt{2x^{3}+8\sqrt{xy}-y}}{\sqrt{3x+y^{2}}-3xy}$	$x=540{,}57$ $y=7{,}4\cdot 10^{2}$	$x=36{,}4$ $y=57{,}8$
21.	$\left(x^{5}+\dfrac{\sqrt{xy}}{4}-y^{2}\right)\dfrac{\sqrt{x^{2}+2xy-3}}{10\sqrt[5]{xy}}$	$x=368$ $y=1{,}9\cdot 10^{2}$	$x=16{,}8$ $y=25{,}9$
22.	$\left(\sqrt{x^{2}+10}-4\right)(x^{5}+4xy+2y^{2})$	$x=-64{,}76$ $y=-14$	$x=78{,}14$ $y=57{,}8$
23.	$\left(a^{3}-5\sqrt[3]{b}+a^{\frac{2}{3}}\cdot b^{3}\right)\dfrac{\sqrt{a^{2}+b^{3}}}{5}$	$a=-27{,}9$ $b=45{,}93$	$a=75{,}98$ $b=151{,}4$
24.	$\left(c^{5}d^{2}+\sqrt{\dfrac{d}{5}}-1\right)\dfrac{\sqrt{cd}}{0{,}1}$	$c=75{,}4$ $d=46{,}84$	$c=96{,}4$ $d=87{,}3$
25.	$\dfrac{17\sqrt{a^{3}+3{,}07b^{5}}}{25\sqrt[3]{ab}}$	$a=94{,}7$ $b=785\cdot 10^{-2}$	$a=49{,}5$ $b=95{,}6$
26.	$\dfrac{25\cdot 10^{-2}\cdot\sqrt{a+b^{2}}}{0{,}8\left(25a^{2}+\sqrt{ab}+b^{2}\right)}$	$a=57\cdot 10^{-3}$ $b=3{,}73$	$a=87{,}5$ $b=53{,}6$
27.	$\dfrac{49\sqrt{x^{2}+3xy+y^{2}}}{24\left(\sqrt{x}+3x^{2}-y^{3}\right)}$	$x=59{,}7$ $y=26{,}9$	$x=14{,}7$ $y=27{,}6$
28.	$(a+b)^{2}+\dfrac{24\sqrt{a-b}}{\sqrt[5]{ab}}$	$a=74$ $b=694\cdot 10^{-2}$	$a=56{,}7$ $b=54{,}8$

1	2	3	4
29.	$\frac{14-\sqrt{a+b}-a^2}{96+(a^2+ab-2)^2}$	$a=0{,}6\cdot10^3$ $b=0{,}89\cdot10^2$	$a=58{,}3$ $b=42{,}8$
30.	$\sqrt{\frac{x+y+10}{x+\sqrt{xy}-4}}(x+y)$	$x=58{,}8$ $y=484\cdot10^{-2}$	$x=47{,}9$ $y=27{,}5$

Завдання 3. Проведіть обчислення по заданій формулі при заданих значеннях параметрів відповідно до варіанту згідно з табл.1.4.

Таблиця 1.4

Варіанти індивідуальних завдань

Варіант	**Вираз**	**Значення змінних**
1	2	3
1.	$\frac{14\cdot sin(m)\cdot cos(m)}{sin(n)}$	$m=84°; n=105°$
2.	$\frac{15\cdot\left(sin^2(m)-cos^2(m)\right)}{cos(n)}$	$m=15°; n=25°$
3.	$\frac{64cos(a)}{sin(b)}+\frac{32sin(c)}{sin(d)}$	$a=75°; b=35°;$ $c=15°;\ d=25°$
4.	$\frac{19cos(a)}{cos(b)}-\frac{14sin(c)}{sin(d)}$	$a=127°; b=27°;$ $c=24°; d=45°$
5.	$\frac{21cos(x)}{sin^2(y)+cos^2(z)}$	$x=18°;\ y=57°; z=237°$
6.	$\frac{14sin(a)cos(a)}{sin(b)}$	$a=33°; b=67°$
7.	$\frac{15tg(a)}{tg(b)}+\frac{19sin(c)}{sin(d)}$	$a=184°;\ b=24°;$ $c=405°;\ d=71°$

1	2	3
8.	$\frac{28cos(x)}{sin^2(y)+sin^2(z)}$	$x=15°;\ y=27°;z=117°$
9.	$\frac{tg(a)-tg(b)}{1-tg(a)tg(b)}$	$a=24°;b=11°$
10.	$\big(ctg(a)-tg(a)\big)tg(b)$	$a=60°;\ b=45°$
11.	$\big(sin(a)-cos(b)\big)^2$	$a=60°;\ b=30°$
12.	$\sqrt{15}cos(a)+\frac{\sqrt{3}}{2}sin^2(b)$	$a=45°;\ b=47°$
13.	$\sqrt{\frac{sin(a)+tg(b)}{1+cos(c)}}$	$a=55°;\ b=15°;c=30°$
14.	$\sqrt{sin^2(a)+cos^2(b)}$	$a=68°;b=118°$
15.	$arccos(a)+arcsin(b)$	$a=\frac{\sqrt{3}}{3};\ b=\frac{1}{4}$
16.	$\frac{\sqrt{1-cos^2\big(arcsin(x)\big)}}{sin(y)}$	$x=\frac{1}{8};\ y=45°$
17.	$\frac{\big[cos\big(2arcsin(a)\big)\big]^2}{sin(b)}$	$a=\frac{1}{10};\ b=115°$
18.	$ctg\big(arccos(k)\big)+cos\big(arctg(m)\big)$	$k=\frac{1}{4};\ m=\frac{1}{3}$
19.	$\frac{cos\big(arctg(a)\big)}{sin\big(arccos(b)\big)}$	$a=\frac{1}{2};\ b=\frac{2}{3}$
20.	$\sqrt[3]{\frac{1+ctg^2(m)}{sin(n)}}$	$m=60°;\ n=15°$

1	2	3
21.	$\dfrac{tg(k)}{\sqrt{1+cos^2(l)}}$	$k = 15°;\ l = 30°$
22.	$\dfrac{tg\big(arcsin(c)\big)}{\sqrt{1+tg^2\big(arccos(d)\big)}}$	$c = \dfrac{1}{2};\ d = \dfrac{2}{3}$
23.	$sin\big(arcctg(a)\big) + cos\big(arctg(b)\big)$	$a = \dfrac{4}{5};\ b = \dfrac{2}{3}$
24.	$tg\big(arcsin(c)\big) + cos\big(arcsin(d)\big)$	$c = \dfrac{1}{2};\ d = \dfrac{2}{3}$
25.	$\dfrac{\sqrt{1-sin^2\big(arccos(x)\big)}}{sin(y)}$	$x = \dfrac{1}{8};\ y = 45°$
26.	$\dfrac{15\cdot\big(sin^2(a)-cos^2(b)\big)}{cos(a)}$	$a = 45°;\ b = 55°$
27.	$\sqrt{\dfrac{cos(a)+tg(b)}{1+sin(c)}}$	$a = 55°;\ b = 15°;\ c = 30°$
28.	$\dfrac{45sin(x)}{sin^2(y)+cos^2(z)}$	$x = 16°; y = 48°; z = 187°$
29.	$\dfrac{\big(14sin(a)+cos(a)\big)^2}{sin(b)}$	$a = 33°;\ b = 67°$
30.	$\dfrac{\sqrt{1-cos\big(arctg(x)\big)}}{cos(y)}$	$x = \dfrac{1}{4};\ y = 48°$

1.3. Контрольні запитання

1. Які системні формати подання чисел ви знаєте? Як подаються дійсні числа при обчисленнях в системі MatLab?

2. Як змінити формат подання дійсних чисел у командному вікні?

3. Як зробити, щоб MatLab компактно відображав введені дані в командному вікні без додаткових пустих рядків?

4. Яким чином оголошуються змінні в мові MatLab?

5. Як зробити так, щоб результат дій, записаних у черговому рядку: а) виводився в командне вікно; б) не виводився на екран?

6. Яку роль грає системна змінна ***ans***?

7. Які є системні змінні у MatLab? Що вони позначають і як їх використовувати ?

8. За що відповідають системні змінні ***realmax*** і ***realmin, eps***?

9. Як повернути в командний рядок раніше введену команду? Як продивитися історію команд?

10. Які вбудовані елементарні математичні функції є у MatLab? Які з них відсутні у всіх інших мовах програмування високого рівня?

11. В яких одиницях має бути поданий аргумент тригонометричних функцій? В яких одиницях виводиться результат застосування обернених тригонометричних функцій?

12. Як виразити значення кута у градусах через радіани і навпаки?

13. Чи існують в MatLab тригонометричні функції, які сприймають введені аргументи у градусах, а не в радіанах?

14. Що таке **NaN?**

15. Як очистити командне вікно від попередніх записів?

КОМП'ЮТЕРНИЙ ПРАКТИКУМ №2
ОПЕРАЦІЇ З КОМПЛЕКСНИМИ ЧИСЛАМИ

Мета роботи: навчитися операціям з комплексними числами у командному вікні MatLab.

2.1. Теоретичні відомості

2.1.1. Введення комплексних чисел

Комплексні числа можуть бути записані в алгебраїчній, експоненціальній та тригонометричній формах.

Алгебраїчна форма комплексного числа має вигляд

$$z = a + ib, \tag{2.1}$$

де a – дійсна частина, b – уявна частина комплексного числа.

Експоненціальна форма запису комплексного числа (2.1)

$$z = |z|e^{i\varphi}, \tag{2.2}$$

де $|z|$ – модуль комплексного числа і визначається

$$|z| = \sqrt{a^2 + b^2}\,, \tag{2.3}$$

φ – аргумент комплексного числа:

$$\varphi = arctg\,\frac{b}{a}. \tag{2.4}$$

На рис. 2.1 зображено комплексне число $z = a + ib$ на комплексній площині.

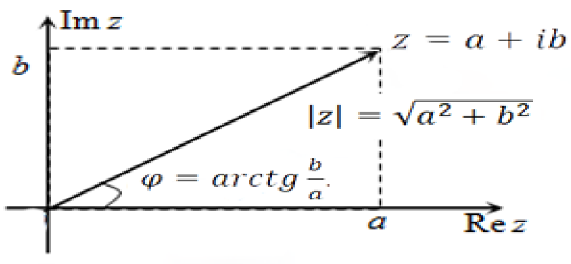


Рис. 2.1. Графічне представлення комплексного числа $z = a + ib$ на комплексній площині

Експоненціальна та тригонометрична форми пов'язані між собою формою Ейлера:

$$e^{i\varphi} = \cos\varphi + i\sin\varphi. \tag{2.5}$$

З врахуванням (2.2) та (2.5) тригонометрична форма має вигляд:

$$z = |z|(cos\,\varphi + i\,sin\,\varphi). \tag{2.6}$$

Для позначення уявної одиниці у мові MatLab зарезервовано два імені – «*i*» та «*j*». Введення з клавіатури значення комплексного числа здійснюється шляхом запису у командне вікно рядка типу:

<ім’я комплексної змінної> = <значення ДЧ>+i[j] <значення УЧ>,*

де ДЧ – дійсна частина, УЧ – уявна частина.

На рис. 2.2 представлено командне вікно з прикладом введення комплексного числа.

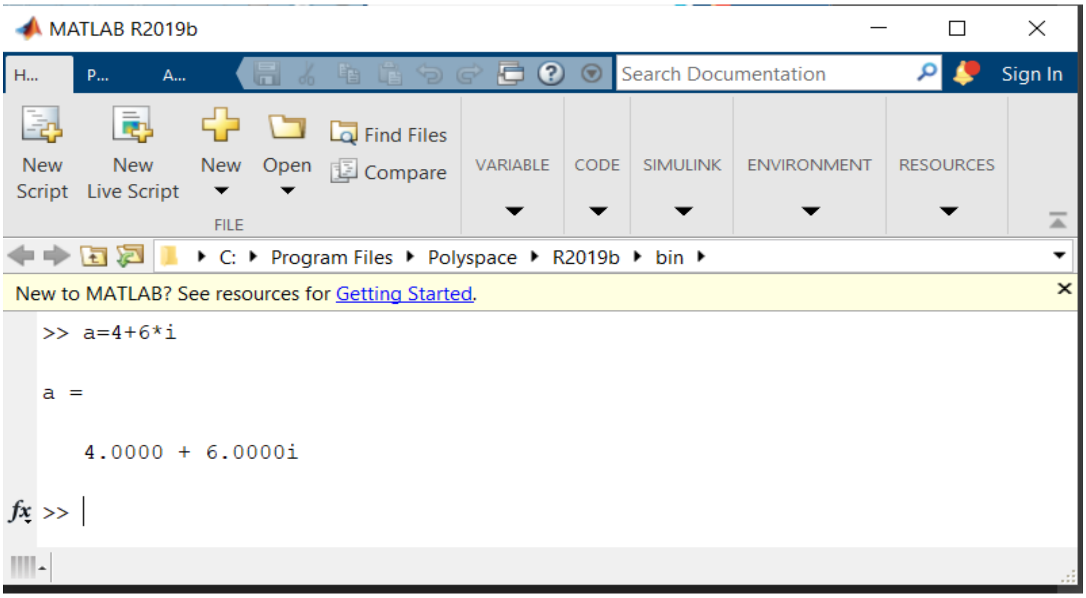


Рис. 2.2. Приклад введення комплексного числа

Також для введення комплексного числа можна використовувати вбудовану функцію ***complex*(a,b)**, де a – дійсна частина, b – уявна частина комплексного числа. Приклад застосування функції представлений на рис. 2.3.

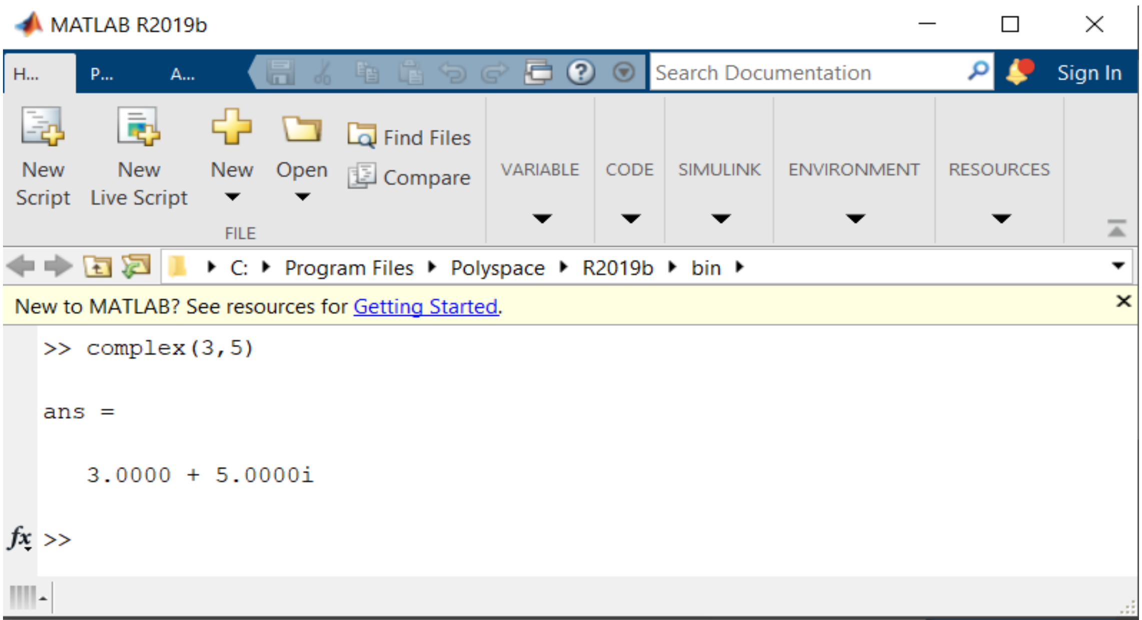


Рис. 2.3 Приклад введення комплексного числа за допомогою функції ***complex***

Дії додавання, віднімання, множення, ділення та піднесення до степеня комплексних чисел здійснюється за допомогою звичайних арифметичних знаків, що й для аналогічних операцій над дійсними числами.

Приклади елементарних дій з комплексними числами наведені на рис. 2.4.

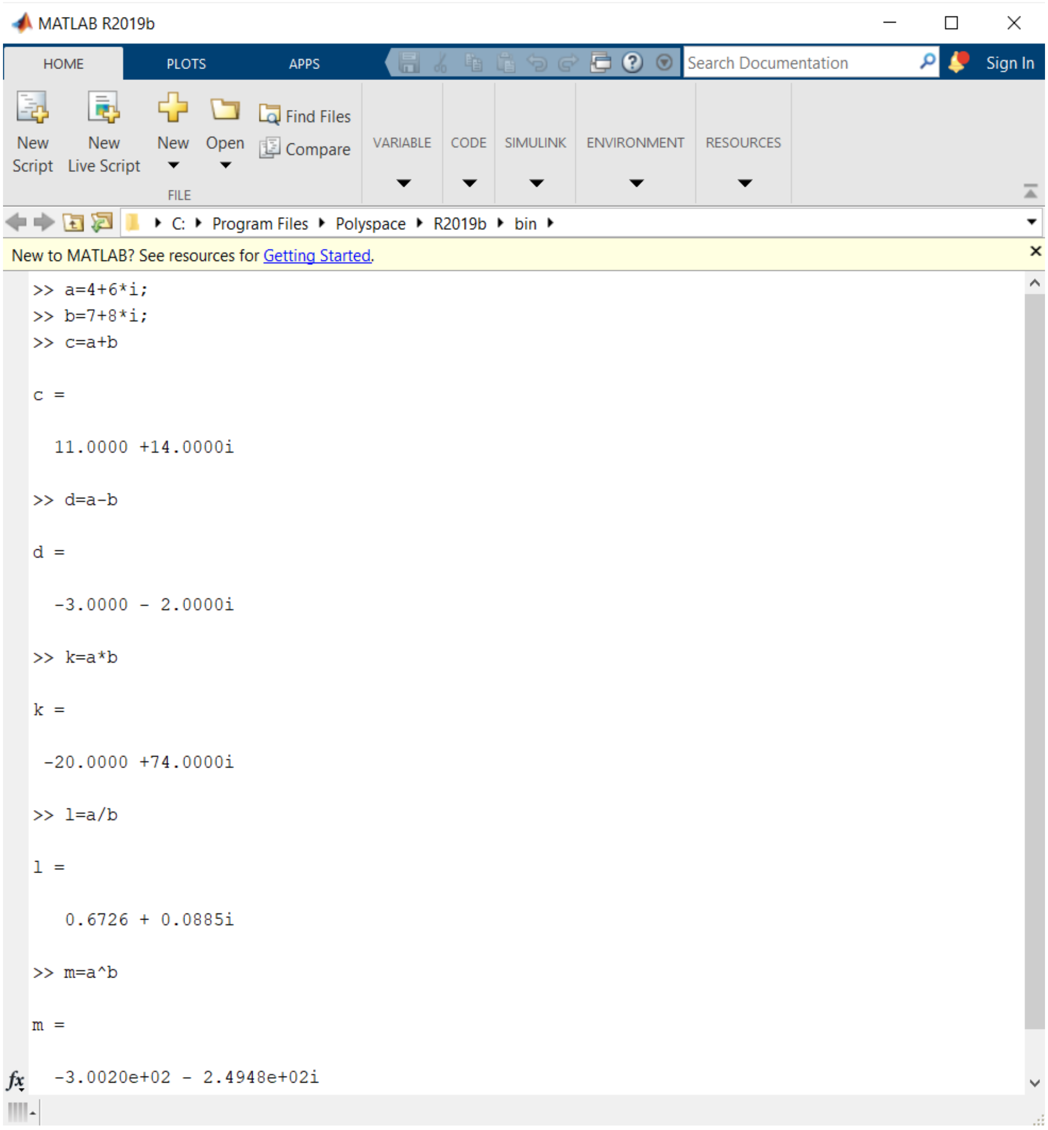


Рис. 2.4 Приклади елементарних дій з комплексними числами

Майже всі елементарні математичні функції, наведені у теоретичних відомостях до комп'ютерного практикуму №1, обчислюються при комплексному значенні аргументу і отримують внаслідок цього комплексні значення результату.

Приклади застосування деяких тригонометричних та експоненціальних функцій наведено на рис. 2.5.

Спеціально для роботи з комплексними числами використовуються наступні функції:

real(Z) – відокремлює дійсну частину комплексного аргументу Z;

imag(Z) – відокремлює уявну частину комплексного аргументу;

abs (Z) – обчислює модуль комплексного числа;

angle(Z) – обчислює значення аргументу комплексного числа Z (у радіанах від –π до +π);

conj(Z) – видає число, комплексно-спряжене відносно Z.

Приклади застосування спеціальних функцій наведені на рис.2.6.

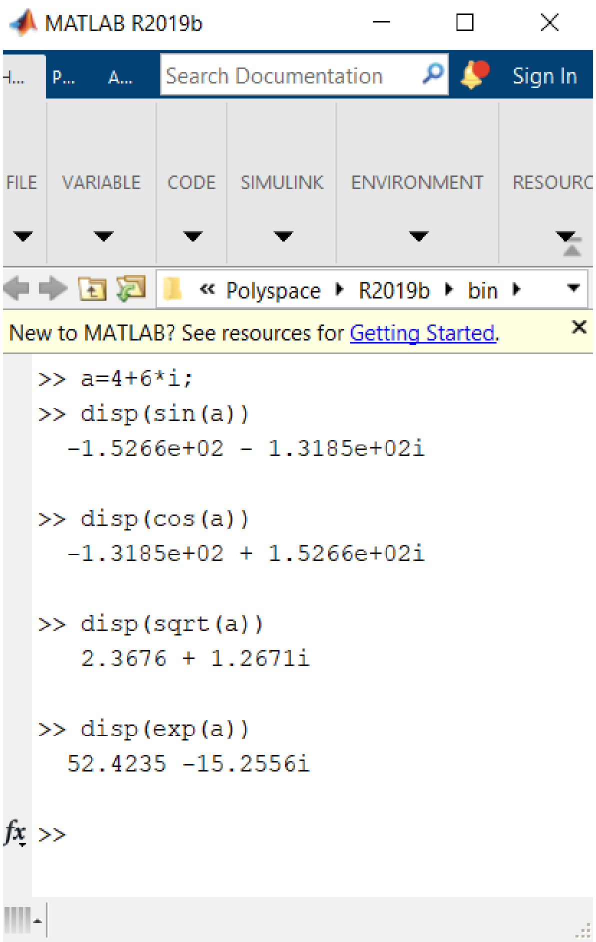


Рис. 2.5 Приклади використання стандартних математичних функцій

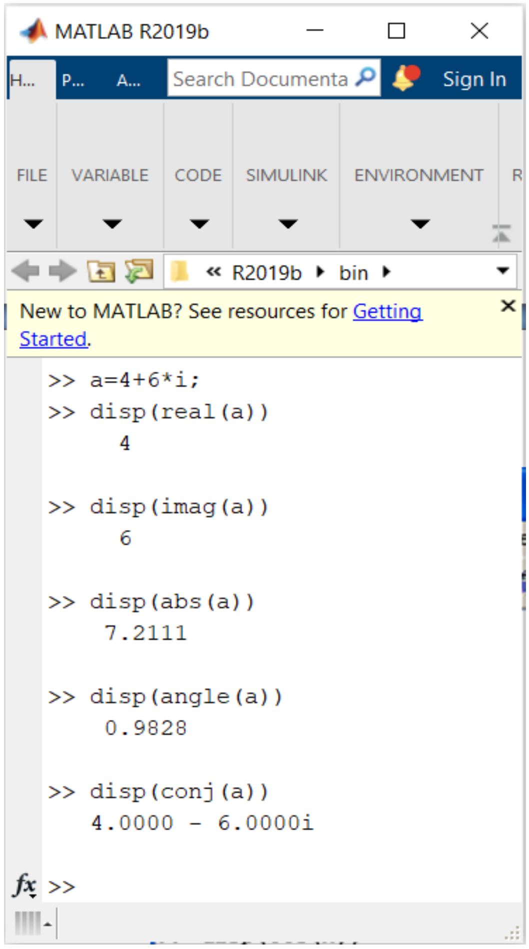


Рис. 2.6 Приклади використання функцій для роботи з комплексними числами

На рис. 2.7 представлено приклад обчислення модуля та аргументу комплексного числа за допомогою виразів (2.3) і (2.4) та вбудованих функцій в MatLab.

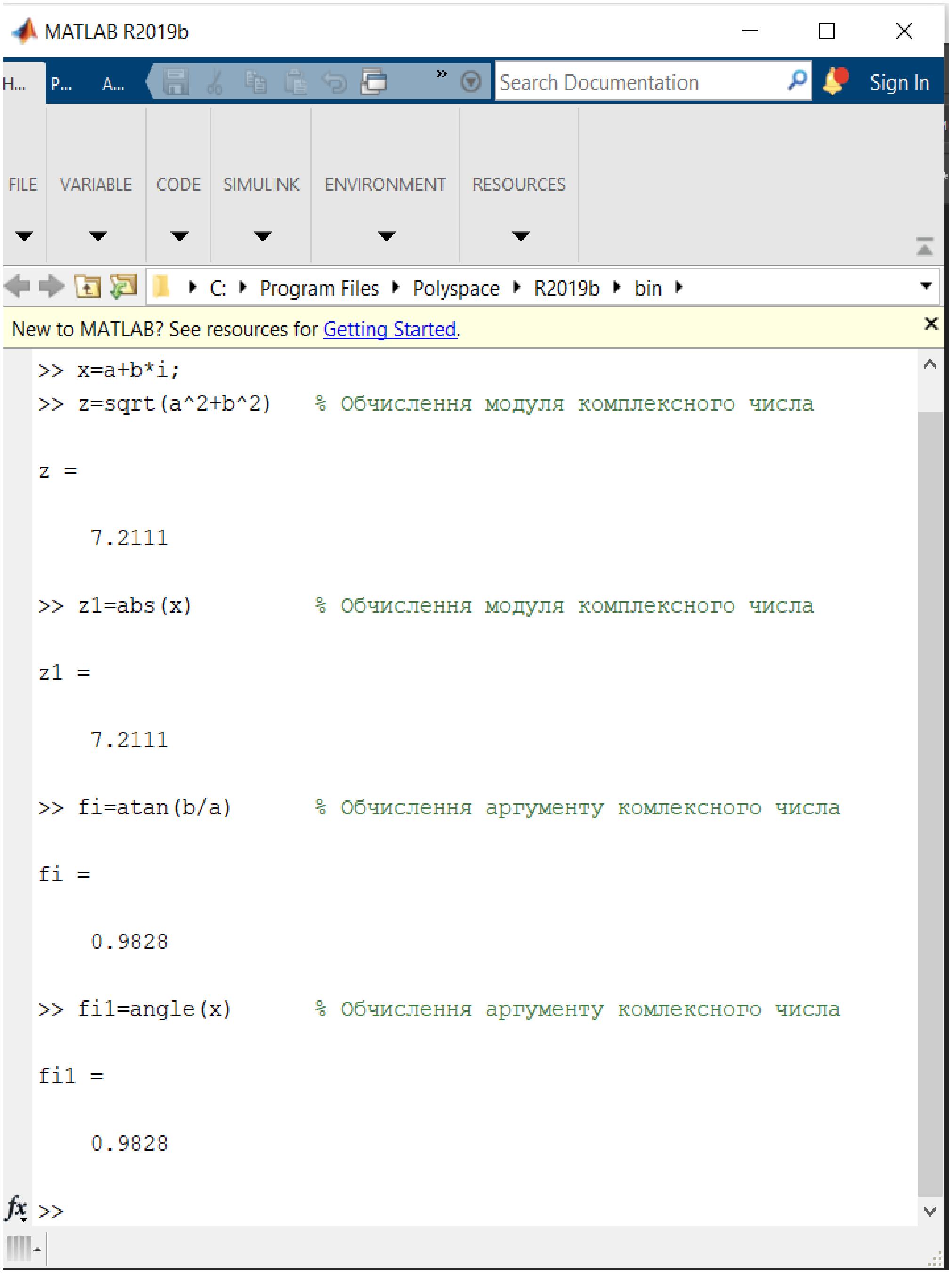


Рис. 2.7. Приклади обчислення модуля та аргументу комплексного числа за допомогою виразів (2.3), (2.4) та вбудованих функцій в MatLab

Приклад переведення з алгебраїчної форми в експоненціальну та тригонометричну форму запису комплексного числа:

```
clc
a=4; b=6;
disp('Алгебраїчна форма запису комлексного числа')
disp(a+b*i)
z=abs(x);    % обчислення модуля комплесного числа
fi=angle(x);% обчислення аргументу комплексного числа
```

```
disp('Експоненціальна форма запису комплексного числа')
disp([num2str(z), '*exp(i*', num2str(fi), ')'])
disp('Тригонометрична форма запису комплексного числа')
disp([num2str(z), '*(cos(', num2str(fi), ')+i*sin(',
...
      num2str(fi), '))'])
```

Результат, що відображається в командному вікні, представлений на рис. 2.8.

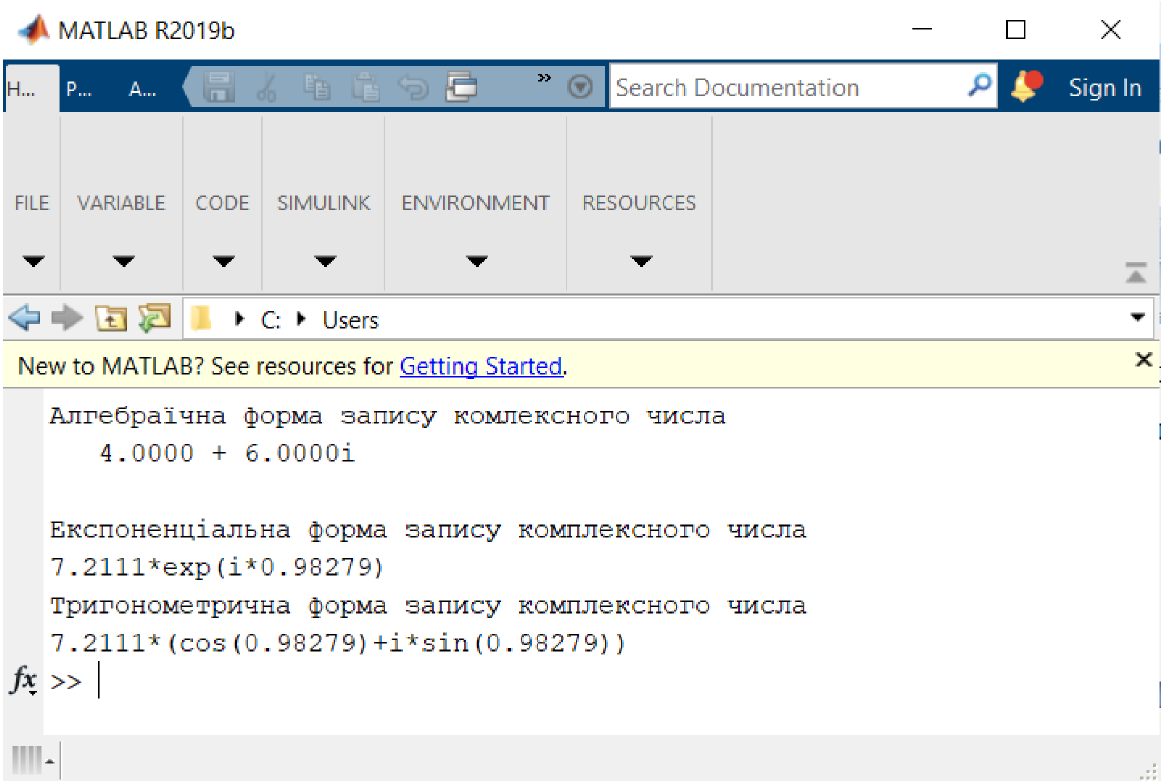


Рис. 2.8. Командне вікно з результатом представлення комплексного числа в алгебраїчній, експоненціальній та тригонометричній формах

Для оформлення отриманих даних використовується функція ***num2str***, яка перетворює числове значення параметра в символьний рядок (*str* = num2str(*A*)).

Символ % – використовується для вказівки логічного кінця рядка. Текст, що знаходиться після знаку відсотка, сприймається як коментар і ігнорується (за винятком коментарів кирилицею, які нерідко ведуть до виникнення помилок). Якщо необхідно закоментувати декілька рядків, тоді виділяються рядки та одночасно нажимають кнопки «CTRL+R», щоб розкоментувати – «CTRL+T».

2.2. Завдання до виконання комп'ютерного практикуму

Завдання 1. Визначити за допомогою вбудованих функцій дійсну та уявну частини, модуль та аргумент комплексного числа, комплексне-спряжене число ***conj*** відповідно до варіанту (табл. 2.1)**.**

Таблиця 2.1

Варіанти індивідуальних завдань

Варіант	**Комплексне число**	
1	2	3
1.	$1{,}5 + 3i$	$-14{,}9 + 9i$
2.	$-5 + 15i$	$3{,}75 + 2{,}67i$
3.	$-5{,}4 - 43i$	$0{,}85 - 23{,}4i$
4.	$54\frac{2}{3} - 3{,}05i$	$2{,}45 + 3{,}15i$
5.	$3{,}87 - 2{,}8i$	$-0{,}5 + 2{,}7i$
6.	$-2{,}7 + 11i$	$3{,}5 + 4{,}5i$
7.	$0{,}14 - 0{,}45i$	$78{,}5 + 67i$
8.	$\frac{1}{2} - 0{,}3i$	$11{,}5 - 5{,}6i$
9.	$2{,}5 + 1{,}5i$	$4{,}5 - 5i$
10.	$1{,}7 + 2{,}1i$	$-1{,}8 + 3i$
11.	$3{,}7 - 2{,}3i$	$2{,}6 - 5{,}4i$
12.	$-4{,}5 + 1{,}3i$	$5{,}5 - 4{,}9i$
13.	$5{,}4 + 3{,}3i$	$-1{,}65 + 4{,}2i$
14.	$-6{,}3 + 4{,}1i$	$15{,}9 - 34i$
15.	$-2{,}7 + 3{,}4i$	$5{,}8 + 2i$
16.	$2{,}5 + 1{,}3i$	$5{,}89 - 8{,}9i$

1	2	3
17.	$-3{,}5-2{,}3i$	$4{,}9-5i$
18.	$4{,}5+2{,}7i$	$4{,}9+5i$
19.	$3{,}5+2i$	$-9{,}4+6i$
20.	$0{,}55+1{,}34i$	$157{,}5-145{,}3i$
21.	$-1{,}5-4{,}2i$	$4{,}9+2{,}1i$
22.	$-4{,}5+2{,}7i$	$6{,}78+2{,}75i$
23.	$5{,}5-6{,}5i$	$57+38i$
24.	$7{,}8-1{,}9i$	$2{,}45+3i$
25.	$2{,}7+3{,}5i$	$-1{,}9-2{,}6i$
26.	$-7{,}185+5{,}9i$	$5{,}24+5{,}7i$
27.	$2{,}45-5{,}6i$	$3{,}8-4{,}7i$
28.	$2{,}75+7{,}9i$	$1{,}98-1{,}98i$
29.	$-0{,}65+2{,}7i$	$-4{,}8-7{,}5i$
30.	$1{,}86+3{,}79i$	$-6{,}7-9{,}4i$

Завдання 2. Представити комплексні числа (табл.2.1) у тригонометричному та в експоненціальному вигляді.

Завдання 3. Виконайте такі дії (див. таблицю 2.2):

а) число z1, задане в алгебричній (експоненціальній) формі, переведіть в експоненціальну (алгебричну) форму, запишіть результат та перевірте його;

б) число z2, задане в експоненціальній (алгебричній) формі, переведіть в алгебричну (експоненціальну) форму, запишіть результат та перевірте його;

в) обчисліть заданий вираз; запишіть результат в алгебричній і експоненціальній та тригонометричній формах, причому аргумент результату забезпечте в межах між (-π) і +π.

Таблиця 2.2

Варіанти індивідуальних завдань

Варіант	**Комплексне число**				**Вираз**
	z_1	z_2	z_3	z_4	
1	2	3	4	5	6
1	$-1+\sqrt{5}i$	$2{,}1e^{211^0 i}$	$0{,}4e^{32^0 i}$	$4+3i$	$\sqrt[3]{z_1}\cdot z_2 : z_3 + z_4$
2	$-0{,}8+2{,}7i$	$-2e^{\sqrt{3}i}$	$0{,}9e^{i5\pi/7}$	$3{,}1-2{,}1i$	$\sqrt{(z_1+z_2)\cdot z_3 : z_4}$
3	$2{,}1-3{,}2i$	$0{,}68e^{148^0 i}$	$\sqrt{5}+\sqrt{2}i$	$2{,}73e^{23^0 i}$	$\sqrt{z_1-z_2} : z_3\cdot z_4$
4	$1{,}1e^{-0{,}8i}$	$\sqrt{5}-2i$	$-1{,}7+i$	$0{,}97e^{\sqrt{2}i}$	$((z_1+z_2)^2-z_3) : z_4$
5	$0{,}3e^{-97^0 i}$	$-1+\sqrt{5}i$	$-0{,}7+4i$	$5{,}2e^{71^0 i}$	$(\sqrt{z_1\cdot z_2}-z_3) : z_4$
6	$2{,}1e^{73^0 i}$	$1+i\pi/2$	$\sqrt{2}+\sqrt{3}i$	$1{,}93e^{192^0 i}$	$\sqrt{(z_1\cdot z_2-z_3) : z_4}$
7	$1{,}05e^{-0{,}4i}$	$\sqrt{5}-i$	$2{,}7e^{0{,}8i}$	$-0{,}7+4i$	$\sqrt{(z_1 : z_2+z_3)\cdot z_4}$
8	$0{,}8-2i$	$3{,}08e^{i7\pi/12}$	$8{,}01e^{2i}$	$-5+\sqrt{2}i$	$z_1^2 : z_2+z_3-z_4$
9	$0{,}187-3{,}94i$	$0{,}3e^{-107^0 i}$	$-0{,}7+4i$	$1{,}5e^{23^0 i}$	$\sqrt[3]{z_1+z_2-z_3}\cdot z_4$
10	$\sqrt{5}-i$	$0{,}7e^{1{,}7i}$	$1{,}2e^{0{,}9i}$	$-3-2i$	$(z_1 : z_2-z_3)^2-z_4$
11	$1{,}1e^{-2{,}1i}$	$\pi/8-2{,}1i$	$2{,}71+0{,}4i$	$1{,}71e^{-\sqrt{3}i}$	$(z_1-z_2)^3 : z_3+z_4$
12	$3{,}21e^{15i}$	$4+3i$	$\sqrt{3}-4i$	$1{,}23e^{111^0 i}$	$(z_1-z_2)^3\cdot z_3+z_4$
13	$5+3{,}87i$	$0{,}9e^{65°i}$	$3{,}6-5i$	$2{,}3e^{34°i}$	$\sqrt{{z_1}^2+z_2+z_3}-z_4$
14	$2{,}5e^{-3{,}4i}$	$-2{,}5+4i$	$0{,}8e^{98°i}$	$3+2{,}6i$	$(z_1+z_2)^2\cdot(z_3+z_4)$
15	$2{,}7+0{,}8i$	$2e^{-\sqrt{3}i}$	$0{,}81e^{i\pi/7}$	$-\sqrt{2}-\sqrt{3}i$	$\sqrt{(z_1+z_2) : z_3\cdot z_4}$
16	$2{,}71e^{i\pi/12}$	$-0{,}7+4i$	$1{,}31e^{-i5\pi/12}$	$-8-3i$	$\sqrt{z_1 : z_2}\cdot z_3-z_4$
17	$1{,}74e^{0{,}3i\pi}$	$0{,}8-2i$	$3{,}28e^{-1{,}2i}$	$\sqrt{3}-\sqrt{2}i$	$(\sqrt{z_1}+z_2)\cdot z_3 : z_4$
18	$3{,}4e^{45°i}$	$7{,}3+2{,}6i$	$1{,}9e^{1{,}2i}$	$3+6i$	$(z_1-z_2)^3\cdot(z_3-z_4)$
19	$-3-2i$	$3{,}21e^{15^0 i}$	$1{,}2+3i$	$2{,}71e^{-78^0 i}$	$\sqrt{z_1\cdot z_2} : z_3+z_4$
20	$1{,}25e^{0{,}6i}$	$-\sqrt{3}-4i$	$4+3i$	$2{,}5e^{3{,}8i}$	$(\sqrt{z_1 : z_2}-z_3)\cdot z_4$
21	$3{,}08e^{i7\pi/12}$	$-3-2i$	$2.03e^{i14/13}$	$\sqrt{3}+\sqrt{2}i$	$(z_1+z_2)^4\cdot z_3 : z_4$
22	$-0{,}7+4i$	$1{,}74e^{i0{,}3\pi}$	$3+4i$	$2{,}1e^{-2{,}3i}$	$\sqrt{z_1 : z_2}\cdot z_3+z_4$

1	2	3	4	5	6
23	$1+i\pi/2$	$1,2e^{107^0 i}$	$0,8-2i$	$2,5e^{14^0 i}$	$(z_1 : z_2 + z_3)^3 \cdot z_4$
24	$2,1e^{0,8i}$	$-\sqrt{5}+2i$	$1,7e^{\sqrt{3}i}$	$0,8e^{2,5i}$	$((z_1 - z_2)^2 + z_3) : z_4$
25	$-\sqrt{3}-4i$	$1,25e^{-0,8i}$	$-3-2i$	$0,75e^{0,7i}$	$(\sqrt[3]{z_1 \cdot z_2} + z_3) : z_4$
26	$3,7e^{-3,4i}$	$3,5+4i$	$1,8e^{68° i}$	$5+1,6i$	$(z_1 - z_2)^2 \cdot (z_3 + z_4)$
27	$4+3i$	$2,71e^{i\pi/12}$	$1,82e^{-1,2i}$	$\sqrt{3}-2i$	$z_1^2 \cdot z_2 : z_3 + z_4$
28	$1,9e^{1,4i}$	$2,5+4i$	$2,4e^{98° i}$	$-4,5+5i$	$(z_1 + z_2)^2 \cdot z_3 + z_4$
29	$-1,1-3,2i$	$0,33e^{-1,9i}$	$2e^{\sqrt{2}i}$	$2,08+i$	$\sqrt{z_1 - z_2} \cdot z_3 : z_4$
30	$1,2e^{1,7i}$	$0,18-3,9i$	$0,71e^{4i}$	$0,8-2i$	$(\sqrt[3]{z_1 : z_2} - z_3) \cdot z_4$

2.3. Контрольні запитання

1. Що таке комплексна одиниця? Чому вона дорівнює?

2. Зобразіть на площині комплексне число 2+2j. Чому дорівнює модуль такого комплексного числа? Аргумент комплексного числа?

3. Як знайти комплексно-спряжене число до заданого?

4. Як додати два комплексні числа?

5.Як перемножити два комплексні числа? Чому дорівнює результат множення (1 + 2j)*(1 - j)?

6. Як поділити на комплексне число? Чому дорівнює 5/(1 + 2j) ?

7. Як піднести до степеню комплексне число? Запишіть аналітичний вираз, використовуючи експоненціальну та тригонометричну форму подання комплексного числа.

8. За допомогою якої функції можна виділити з комплексного числа лише дійсну частину?

9. За допомогою якої функції можна виділити з комплексного числа лише уявну частину?

10. Як знайти модуль комплексного числа? Аналітично та за допомогою функцій MatLab?

11. Як знайти аргумент комплексного числа? Аналітично та за допомогою функцій MatLab? Назвіть мінімум дві функції, що дозволяють виконати поставлене завдання.

12. Яким чином можна закоментувати будь-який рядок у скрипт-файлі? Які гарячі клавіші використовують для коментування/розкоментування рядків коду?

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 3
ОПЕРАЦІЇ З ВЕКТОРАМИ

Мета роботи: навчитися використовувати вбудовані функцій MatLab, створювати вектори та здійснювати з ними елементарні дії.

3.1. Теоретичні відомості

Вектор – це одновимірний масив чисел. Вектора бувають: вектор-рядок та вектор-стовпець. За замовчуванням у системі MatLab вектор вважається вектором-стовпцем.

У системі MatLab початкові значення векторів задаються з клавіатури за допомогою поелементного введення. Для цього у рядку слід спочатку вказати ім'я вектора, потім поставити знак присвоєння «=», далі відкрити квадратну дужку, а за нею ввести задані значення елементів вектора, відділяючи їх пропусками або комами. Завершується рядок записом закриваючої квадратної дужки.

Наприклад, вектор-рядок

$$V = [1\ 2\ 3\ 4]$$

задає вектор V, що містить чотири елементи зі значеннями 1, 2, 3 та 4.

Довгий вектор можна вводити частинами, які потім об'єднувати за допомогою операції поєднання векторів. У кінець першого вектора дописуються елементи другого вектора, таким чином створюється новий вектор.

$$V = [V_1 \quad V_2].$$

Результат поєднання двох векторів представлений на рис. 3.1.

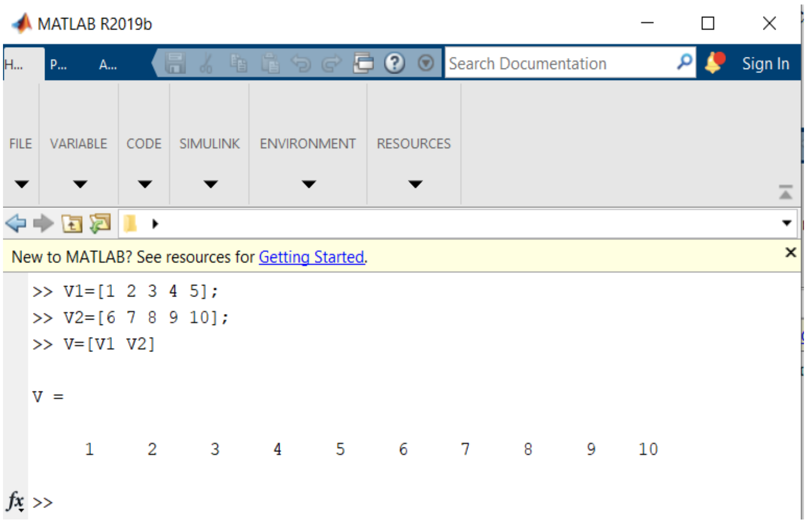


Рис. 3.1 Приклад поєднання двох векторів у командному вікні

У системі MatLab вектор, який містить елементи значення якого є арифметичною прогресією може бути записаний у вигляді:

$$V = pz: h: kz\ ,$$

де pz – початкове значення прогресії, kz – кінцеве значення прогресії, h – різниця прогресії (крок).

Вектор-стовпець вводиться аналогічно, але значення елементів у їх переліку відділяються знаком «;».

На рис. 3.2 представлені приклади запису векторів-рядків та векторів-стовпців.

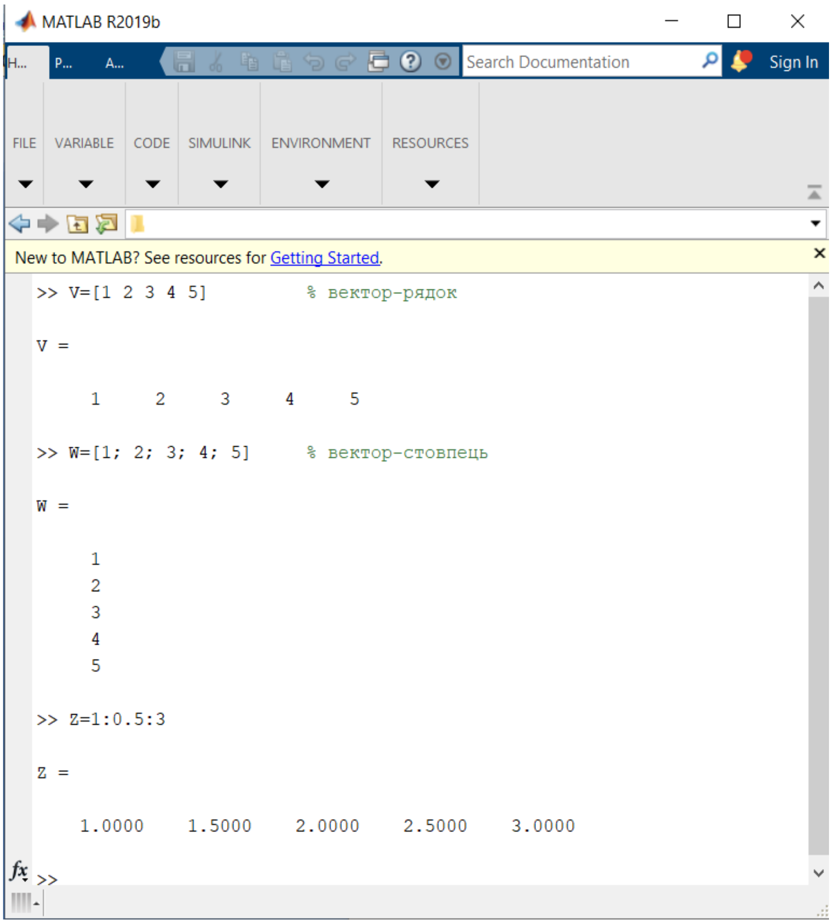


Рис. 3.2 Приклади запису вектора-рядка та вектора-стовпця у командному вікні

Розрізняють дві суттєві різні групи дій над векторами:

– векторні дії – такі, що дозволяються векторним численням у математиці;

– дії по перетворюванню – це дії, що перетворюють елементи вектора, але не є операціями, що дозволені математикою з векторами.

До векторних дій належать дії додавання, віднімання, множення та транспонування векторів.

Додавання векторів може здійснюватися лише, якщо вектори однакового типу (обидва є або векторами-рядками, або векторами-стовпцями) та мають однакову довжину (тобто мають однакову кількість елементів).

Транспонування вектора здійснюється за допомогою апострофа, який записується одразу за записом початкового вектора (який транспонується).

На рис. 3.3 представлені приклади додавання та віднімання векторів.

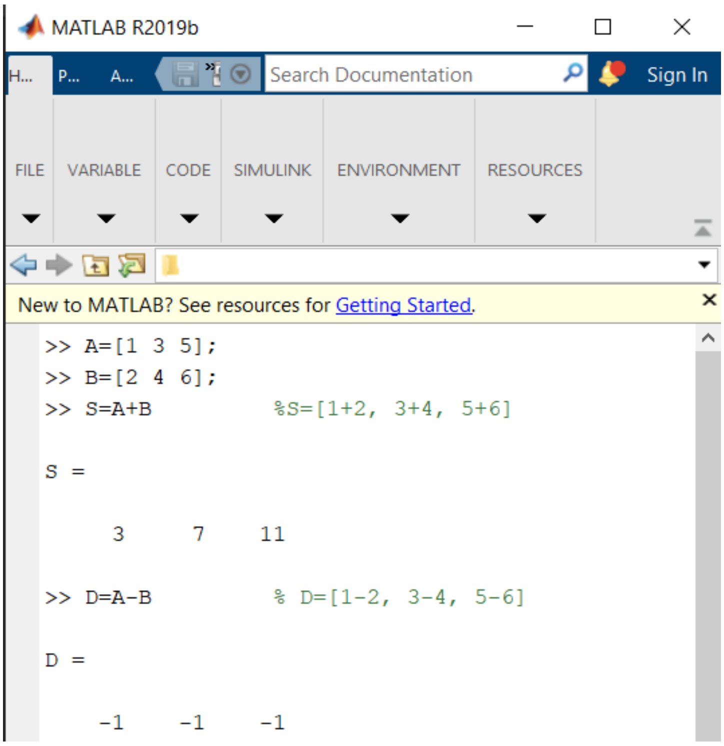


Рис.3.3 Приклади додавання та віднімання векторів

Додавання, віднімання та множення векторів здійснюється за допомогою знака арифметичного додавання «+», віднімання «-» та множення «*», відповідно.

Множення двох векторів визначено у математиці лише для векторів однакового розміру і лише тоді, коли один з векторів-співмножників є рядком, а другий – стовпцем. Тобто якщо вектори X і Y є рядками, то математичний зміст мають лише дві форми множення цих векторів: $C = A \cdot B'$ та $D = B' \cdot A$. Причому у першому випадку результатом є число, а у другому – квадратна матриця.

На рис. 3.4 представлено командне вікно з прикладами дій множення векторів та транспонування.

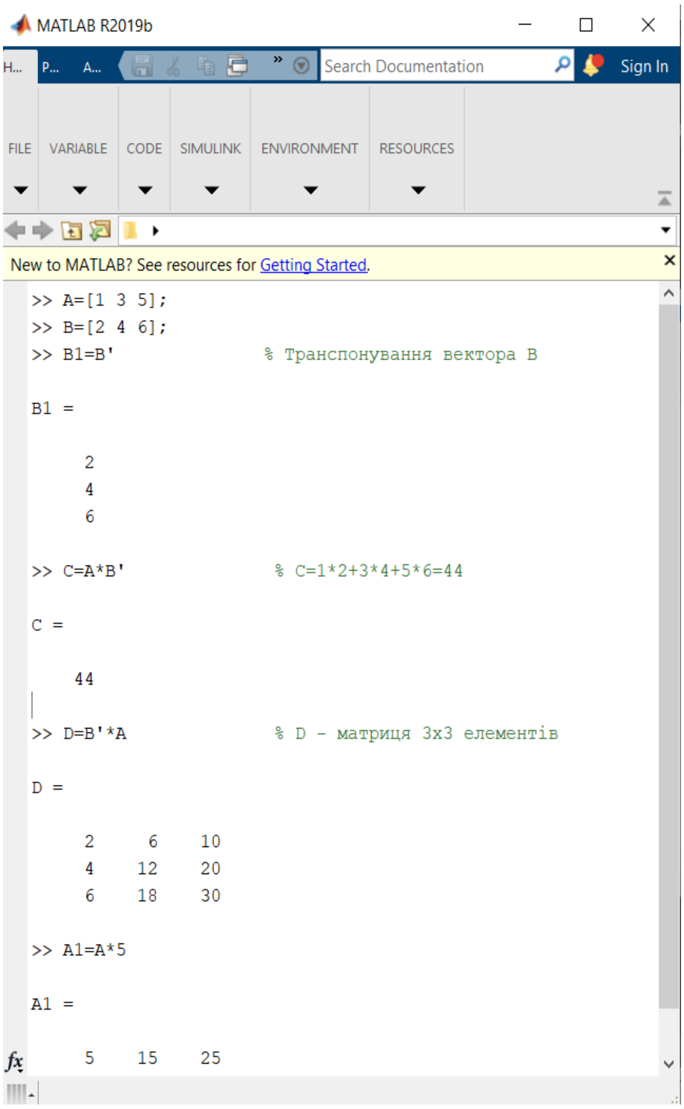


Рис. 3.4 Приклади множення векторів та транспонування вектора

Дії з перетворюванням векторів

У мові MatLab є цілий ряд операцій, що перетворюють заданий вектор на іншій того ж розміру і типу, але які не є математичними операціями з вектором, як математичним об’єктом. Усі ці операції перетворюють елементи вектора, як елементи звичайного одномірного масиву чисел. До таких операцій належать, наприклад, усі елементарні математичні функції, що наведені у теоретичних відомостях до комп’ютерного практикуму №1.

У MatLab передбачені кілька операцій поелементного перетворення, що здійснюються за допомогою звичайних арифметичних дій:

– додавання (віднімання) до кожного елемента вектора числа здійснюються за допомогою знаку «+» («-»);

– поелементне множення векторів здійснюється за допомогою сполучення знаків «.*», що записується між іменами векторів, які перемножуються;

– поелементне ділення векторів здійснюється за допомогою сполучення знаків «./»;

– поелементне ділення векторів у зворотному напрямку здійснюється за допомогою сполучення знаків «.\»;

– поелементне піднесення до степеня здійснюється за допомогою сполучення знаків «.^».

На рис. 3.5 представлено командне вікно з прикладами операцій поелементного перетворення векторів: додавання, віднімання, множення, ділення та піднесення до степеня.

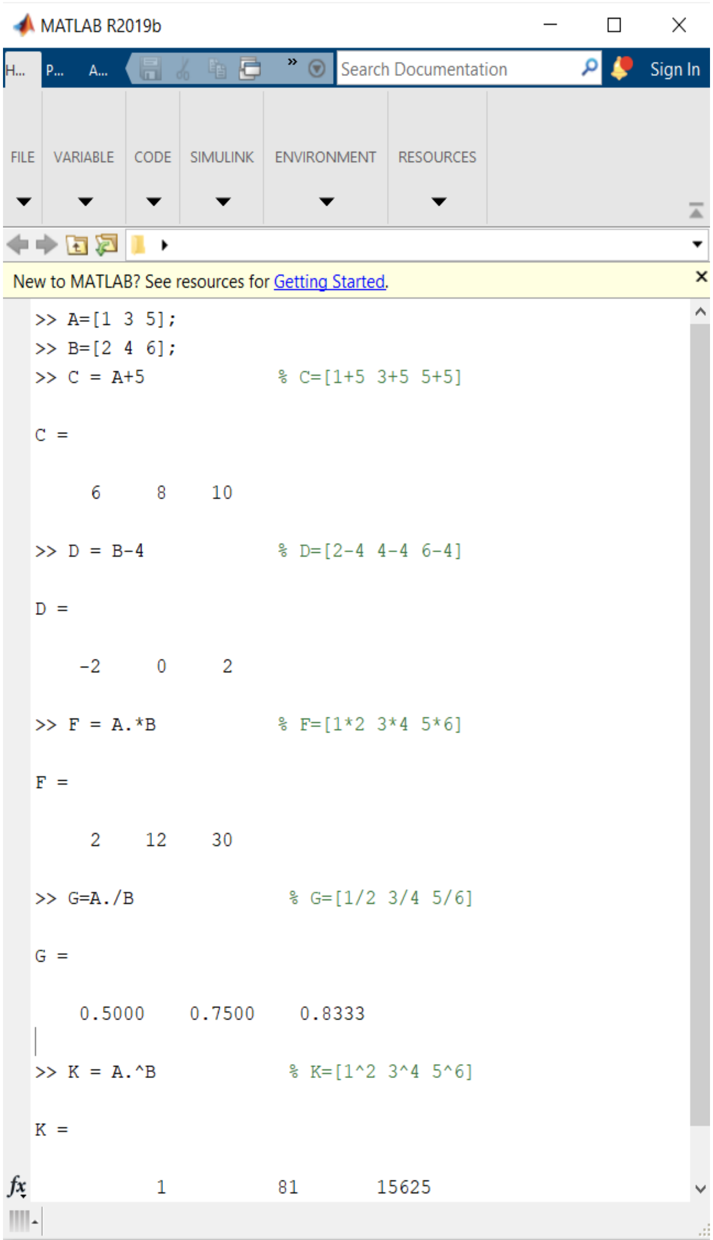


Рис.3.5 Приклади з операціями поелементного перетворення векторів

У табл. 3.1 приведені функції, які формують вектори та призначені для загального оперування з об’єктами типу вектори та матриці.

Таблиця 3.1

Перелік функцій, які формують вектори та призначені для загального оперування з об'єктами вектори та матриці

Функція	Опис
linspace	Створює вектор з лінійно розподіленими значеннями
logspace	Створює вектор з логарифмічно розподіленими значеннями
length	Повертає довжину самого великого виміру масиву
size	Повертає значення (довжину) кожної розмірності масиву
ndims	Визначає розмірність багатовимірного масиву
numel	Повертає кількість елементів масиву
isscalar	Функція, яка визначає чи являється скалярним значенням поданий до неї об'єкт
isvector	Функція, яка визначає чи являється вектором поданий до неї об'єкт
ismatrix	Функція, яка визначає чи являється матрицею поданий до неї об'єкт
isrow	Функція, яка визначає чи являється вектором-рядком поданий до неї об'єкт
iscolumn	Функція, яка визначає чи являється вектором-стовпцем поданий до неї об'єкт
isempty	Функція, яка визначає чи порожній об'єкт

3.2. Завдання до виконання комп'ютерного практикуму

Завдання 1. Знайдіть корені квадратного рівняння

$$ax^2 + bx + c = 0$$

при заданих значеннях коефіцієнтів a, b і c (див. табл. 3.2) за точними формулами і користуючись функцією ***roots***. Результати порівняйте.

Таблиця 3.2

Варіанти індивідуальних завдань

Варіант	**a**	**b**	**c**		**Варіант**	**a**	**b**	**c**
1.	15	12	1		16.	0,13	0,67	-17
2.	23	26	1,5		17.	15	25,7	1,8
3.	2e-2	14	28		18.	0,15	3	1,5
4.	6,5	2,7	-5		19.	15e-2	18	24
5.	4,3e2	7,05e-3	-1,5		20.	2.5	2,6	0,15
6.	5.2	475	3.45		21.	8,5	6,45	-7
7.	-8,5	535	58		22.	5,4	-1,9	-0,65
8.	15	35	2,4		23.	15	1,3	-3,4
9.	1,2	0,45	-16e-2		24.	0,97	2,6	5,4e-1
10.	17	12	-8,75		25.	15	35	14
11.	3,3	-27,7	-0,45		26.	35	1,2	-1,5
12.	28	28	-1,3		27.	1,3	1,66	0,15
13.	1,3	0,87	-2,17		28.	4,6	-23	4,6
14.	75	11	1,3e-2		29.	0,23	2e2	3,1
15.	2,3	89	234		30.	0,23	2,6	-0,15

Завдання 2. За заданими векторами A і B знайти (див. табл. 3.3):

1. Суму векторів A і B та сформувати вектор C.
2. Різницю векторів A і B та сформувати вектор D.
3. Помножити вектор D на значення k (k – номер варіанту)
4. Відсортувати вектор C у порядку зростання.
5. Відсортувати вектор C у порядку зменшення.
6. Вивести значення третього та п’ятого елементів.
7. Замінити всі парні елементи вектора C на значення k, а непарні – на $k-1$ (k – номер варіанту).
8. Об’єднати вектора C та D та сформувати вектор F.
9. Перетворити вектор-рядок F в вектор-стовбець G.
10. Визначити суму всіх елементів вектора F.
11. Перемножити поелементно вектор A на вектор B та сформувати вектор H.
12. Помножити вектор-рядок A на вектор-стовпець B та сформувати вектор K. Порівняйте отримані вектори H та K. Поясніть отриманий результат.
13. Помножити вектор-стовпець A на вектор-рядок B та сформувати новий об’єкт M. Що Ви отримали в результаті даних дій, як називається

отриманий об’єкт? Чому при перемноженні двох векторів (п.11 та п.12) були отримані зовсім різні результати?

14. Поділити поелементно вектор *A* на вектор *B* та сформувати вектор *L*. Виконайте ділення векторів за допомогою операції «/» та операції «\»? Поясніть отримані результати, чому вони відрізняються між собою?

15. Чи вірно A=(дорівнює) L*B’? Поясніть отриманий результат.

16. Перевірте результат виконання функцій ***isscalar, isvector, ismatrix***, ***isrow, iscolumn, isempty*** на створених векторах *A, B, C, D, F, G, H, K, M, L*.

Таблиця 3.3

Варіанти індивідуальних завдань

Варіант	**Вектор A**	**Вектор B**
1	2	3
1.	$[2.6\ \ 2.5\ \ -2.7\ \ 0.9\ \ 1.1]$	-3 : 1
2.	$[-2\ \ 5.5\ \ 3.2\ \ -3\ \ 1]$	-2 : 2
3.	$[1.2\ \ 2.5\ \ -0.7\ \ -1.9\ \ 0.1]$	-4 : 0.5 : -2
4.	$[25\ \ 57\ \ -7\ \ 1.9\ \ 14]$	2 : 0.4 : 3.6
5.	$[28\ \ -5\ \ 1.7\ \ 4.9\ \ 19]$	-2 : 0.5 : 0
6.	$[5.2\ \ -\ \ 3.5\ \ 4.9\ \ 2.9\ \ 3.7]$	-1.5 : 2.5
7.	$[5\ \ -8.5\ \ 9.9\ \ 4.7\ \ 1.9]$	0 : 0.5 : 2
8.	$[3.3\ \ -3.5\ \ 2.7\ \ 3.9\ \ 15]$	-1 : 3
9.	$[2.5\ \ 5.7\ \ 3.7\ \ 3.9\ \ 5.8]$	-0.6 : 0.6 : 1.8
10.	$[-4.7\ \ 3.8\ \ 2.8\ \ 1.5\ \ 3.6]$	-1.6 : 0.8 : 1.6
11.	$[6.6\ \ -6.9\ \ 7.2\ \ 5.4\ \ 2.1]$	-1.4 : 0.7 : 1.4
12.	$[2.7\ \ 9.7\ \ 4.7\ \ 2.7\ \ 14]$	-1.5 : 0.75 : 1.5
13.	$[4.3\ \ 7.8\ \ 1.6\ \ -4.9\ \ 6.1]$	-0.8 : 0.4 : 0.8
1	2	3

14.	$[2.5\ \ 12\ \ -9\ \ 5.9\ \ 1.8]$	-1.2 : 0.4 : 0.4
15.	$[2.8\ \ 3.6\ -4.8\ \ 3.8\ \ 1.9]$	-1 : 0.5 : 1
16.	$[4.6\ \ -3.7\ \ 5.8\ \ 3.9\ \ 0.6]$	-3 : 1
17.	$[2.7\ \ 7.5\ \ 5.2\ \ 5.7\ \ -15]$	-2 : 2
18.	$[1.5\ \ -4.5\ \ 3.9\ \ 4.8\ \ 9]$	-4 : 0.5 : -2
19.	$[3.5\ \ -5.7\ \ 4.6\ \ 3.7\ \ 1.4]$	2 : 0.4 : 3.6
20.	$[12\ \ 5.5\ \ -4.7\ \ 8.9\ \ 1.9]$	-2 : 0.5 : 0
21.	$[7.6\ \ 9.5\ \ -2.8\ \ 8.7\ \ 3.1]$	-1.5 : 2.5
22.	$[-5.2\ \ 3.5\ \ 8.5\ -2.4\ \ 16.5]$	0 : 0.5 : 2
23.	$[4.2\ \ -1.5\ \ 2.7\ \ -4.8\ \ 6.1]$	-1 : 3
24.	$[2.5\ \ 3.7\ \ -3.6\ \ 3.9\ \ 1.9]$	-0.6 : 0.6 : 1.8
25.	$[2.8\ \ -2.3\ \ 5.7\ \ 8.9\ \ 9]$	-1.6 : 0.8 : 1.6
26.	$[2.6\ \ 15\ \ -27\ \ 9\ \ 11]$	-1.4 : 0.7 : 1.4
27.	$[25\ \ 9.5\ -5.9\ -4.9\ \ 9.3]$	-1.5 : 0.75 : 1.5
28.	$[12\ \ 45\ -9.7\ \ -7.9\ \ 9.1]$	-0.8 : 0.4 : 0.8
29.	$[15\ \ 18\ \ -8.9\ \ 9.9\ \ 7.4]$	-1.2 : 0.4 : 0.4
30.	$[8.8\ \ -12.5\ \ 17.7\ \ 14.9\ \ 19.6]$	-1 : 0.5 : 1

Завдання 3. а) Обчисліть значення функції $f(x)$ на відрізку $[a;\ b]$ із кроком h у відповідності з табл. 3.4.

б) Обчисліть значення функції $f(x)$ на відрізку [a; b], задавши логарифмічну сітку розбиття аргументу x із кількістю точок, аналогічній у пункті a.

Таблиця 3.4

Варіанти індивідуальних завдань

Варіант	$f(x)$	**a**	**b**	**h**
1	2	3	4	0,5
1	$\arcsin\ e^{-x^2/5}$	8	13	0,5
2	$\dfrac{1+e^{-x/2}}{\sqrt{3x^2+1}}$	3	5	0,2
3	$\left(x^4-\dfrac{\sqrt{x}}{2}\right)\dfrac{\sqrt{x^2+3x-4}}{18\sqrt[3]{x}}$	0,1	4	0,1
4	$\sqrt{3+2x}\cdot tg\,\dfrac{\pi x^3}{2}$	0,1	0,9	0,08
5	$\sqrt{1+2x^2}\cdot\sin\dfrac{3x}{2}$	2	4	0,2
6	$\dfrac{x^2}{1+0,25\sqrt{x}}$	1,1	3,1	0,2
7	$\dfrac{15\cdot\left(sin^2(x)-cos^2(x)\right)}{cos(x)}$	0,1	2	0,2
8	$3x^3+\dfrac{1}{x}+e^{-2x^2}$	1,2	2,2	0,1
9	$x^{2x+1}+x^3-2x$	1	5	0,4
10	$\dfrac{8(8x^3+12x^2)}{9x^5-\sqrt[5]{x}+2x^2}$	0,1	3	0,3
11	$\sqrt{3x^2+5}\cdot\cos\dfrac{\pi x}{2}$	0,5	1,5	0,1
12	$\sqrt{\dfrac{4x+x^2+2x^3}{\sqrt[5]{18x}}}$	0,1	2	0,2
13	$\dfrac{2e^{-x}}{2\pi+x^3}$	0	1,6	0,16
14	$\sqrt{1+4x}\sin\pi x$	0,1	0,8	0,07
15	$e^{-2x}+x^2-1$	0,25	2,25	0,2
16	$\arccos\ e^{-\sqrt[3]{3x}}$	0,2	0,5	0,03

1	2	3	4	5
17	$\sqrt[3]{x^2+3}\cdot\cos\frac{\pi x}{2}$	1	2,5	0,15
18	$\left(x^3-5\sqrt[3]{x}+x^{\frac{2}{3}}\right)\frac{\sqrt{x^2+x^3}}{5}$	0,1	2	0,1
19	$\frac{x^3-0{,}3x}{\sqrt{1+2x}}$	2,05	3,05	0,1
20	$\frac{\cos\pi x^2}{\sqrt{1-3x}}$	-1	0	0,1
21	$\frac{17\sqrt{x^3+3{,}07x^5}}{25\sqrt[3]{x}}$	0,1	2	0,2
22	$\frac{e^{x/3}}{1+x^2}$	1,4	2,4	0,1
23	$e^{-x^2}(1+3x-x^2)$	0	2	0,2
24	$\frac{\sqrt{1-sin^2(arccos(x))}}{sin(y)}$	0,1	2	0,2
25	$\frac{15\cdot\left(sin^2(x)-cos^2(x)\right)}{cos(x)}$	0,1	2	0,1
26	$\frac{28cos(x)}{sin^2(x)+sin^2(x)}$	0,2	3	0,4
27	$\frac{tg(x)-sin(x)}{1-tg(x)sin(x)}$	0,1	2	0,1
28	$x^3-3x+\frac{8}{\sqrt{1+x^2}}$	0	1,7	0,17
29	$(4+7x)\sin(\pi^3\sqrt{1+x})$	0	7	0,7
30	$\frac{x^3+2x}{\sqrt{1+e^x}}$	-0,2	0,8	0,1

3.3. Контрольні запитання

1. Що таке вектор у MatLab?
2. Як задати вектор-рядок? Як задати вектор-стовпець?
3. Як поєднати два вектори-стовпці? Як поєднати два вектори-рядки?
4. Що таке векторні дії? Дії по перетворенню векторів?

5. Як транспонувати вектор?

6. Як виконується поелементне перемноження двох векторів? Як помножити вектор-рядок на вектор-стовпець? Яка умова має виконуватися, щоб можна було виконати останню дію?

7. Чи можна помножити вектор-стовпець на вектор-рядок? Що отримаєте в результаті?

8. Які дії по перетворенню векторів ви знаєте? Як поелементно піднести до степеня вектор?

9. Як визначити чи розглядуваний об’єкт є вектором? Як знайти довжину вектора?

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 4
ОПЕРАЦІЇ З МАТРИЦЯМИ

Мета роботи: навчитися створювати матриці із заданими властивостями і здійснювати елементарні дії над ними.

4.1. Теоретичні відомості

Матриця – це багатовимірний масив чисел.

У математиці використовується велика кількість спеціальних матриць. Деякими найпоширенішими видами матриць є:

- Нульова матриця – це матриця, в якій всі елементи дорівнюють нулю. В MatLab нульова матриця розміром $(M \times N)$ формується за допомогою функції ***zeros(M, N)*** (рис. 4.1).
- Матриця з усіма одиничними елементами розміром $(M \times N)$ формується за допомогою функції ***ones(M, N)*** (рис. 4.1).
- Одиничні матриці – це матриці, в яких всі елементи у головній діагоналі дорівнюють одиницям, а всі решта – нульові елементи. Одинична матриця розміром $(M \times N)$ в MatLab формується за допомогою функції ***eye(M, N)*** (рис. 4.2).
- Випадкова матриця – це матриця з випадковими числами, які рівномірно розподілені у діапазоні від 0 до 1. Випадкова матриця розміром $(M \times N)$ з числами розподіленими за ***рівномірним законом*** розподілу формується за допомогою функції ***rand(M, N)*** (рис. 4.2)**,** а функція ***randn(M, N)*** – створює матрицю розміром $(M \times N)$ з випадковими числами, які рівномірно розподілені за ***нормальним законом***.
- Нижня трикутна матриця – це матриця, яка формується на основі матриці A шляхом обнулювання її елементів вище головної діагоналі. Нижня трикутна матриця створюється за допомогою функції ***tril(A)*** (рис. 4.3).
- Верхня трикутна матриця – це матриця, яка формується на основі матриці A шляхом обнулювання її елементів нижче головної діагоналі. Нижня трикутна матриця створюється за допомогою функції ***triu(A)*** (рис. 4.3).

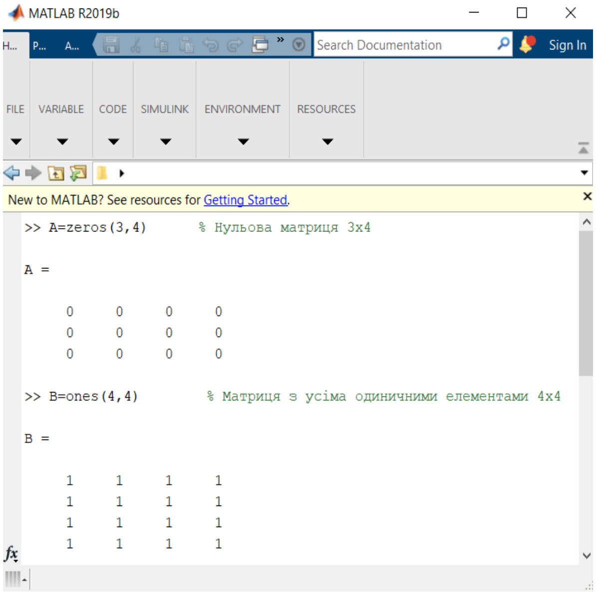


Рис. 4.1 Приклади використання функцій ***zeros*** та ***ones***

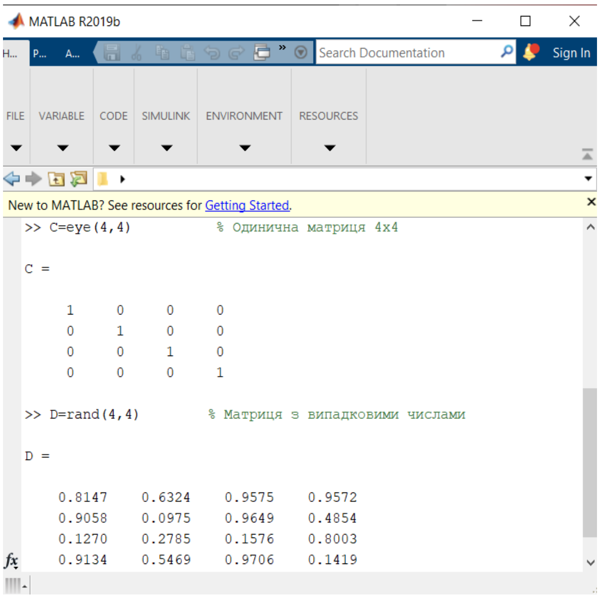


Рис. 4.2 Приклади використання функцій ***eye*** та ***rand***

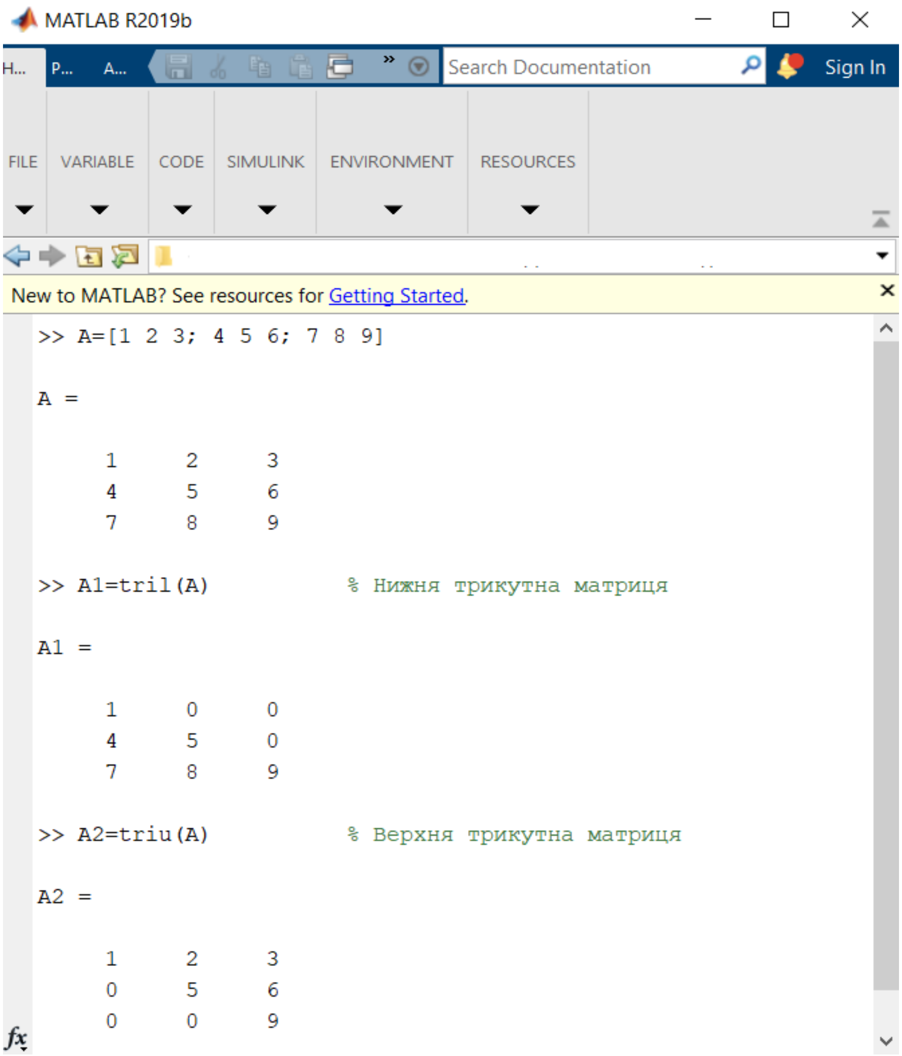


Рис. 4.3 Приклади використання функцій ***tril(A)*** та ***triu(A)***

У MatLab вбудовані функції для формування масиву елементів:

• ***randn(M,N)*** – утворює матрицю розміром $(M*N)$ з випадковими числами, які рівномірно розподілені за нормальним законом з нульовим математичним очікуванням та стандартним (середньоквадратичним) відхиленням, що дорівнює одиниці.

• ***hadamard(N)*** – утворює матрицю Адамара розміром $(N*N)$;

• ***hilb(N)*** – утворює матрицю Гільберта розміром $(N*N)$;

• ***invhilb(N)*** – утворює обернену матрицю Гільберта розміром $(N*N)$;

• ***fliplr(A)*** – утворює матрицю, переставляючи стовпці відомої матриці A відносно вертикальної осі;

• ***flipud(A)*** – утворює матрицю, переставляючи рядки відомої матриці A відносно горизонтальної осі;

• ***rot90(A)*** – утворює матрицю шляхом «повороту» відомої матриці A на 90 градусів проти годинникової стрілки;

• ***rechape(A,m,n)*** – утворює матрицю розміром $(m*n)$ шляхом послідовної вибірки елементів заданої матриці A по стовпцям; при цьому кількість елементів матриці A повинна дорівнювати $m*n$;

До дій з матрицями належать такі операції, які випливають з матричного числення у математиці і не протирічать йому.

Базовими операціями над матрицями є додавання, віднімання, транспонування, множення матриці на число, множення матриці на матрицю, піднесення матриці до цілого степеня. Вони здійснюється у мові MatLab за допомогою звичайних знаків арифметичних операцій. При застосуванні цих операцій важливо пам'ятати умови, за яких ці операції є можливими:

- при додаванні або відніманні матриць обидві матриці повинні мати однакові розміри;
- при множенні матриць кількість стовпців першого множника повинна збігатися з кількістю рядків другого множника (матриці мають бути узгодженими).

На рис. 4.4 та рис. 4.5 представлені командні вікна з прикладами додавання, віднімання матриць, транспонування, множення матриць та піднесення матриці до степеня.

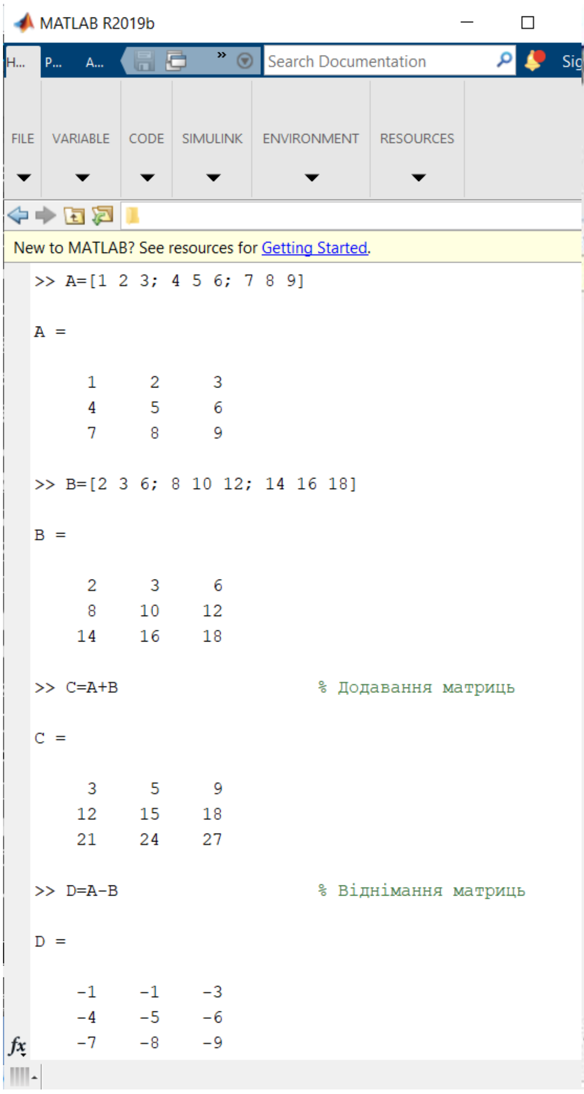


Рис. 4.4 Приклади додавання та віднімання матриць

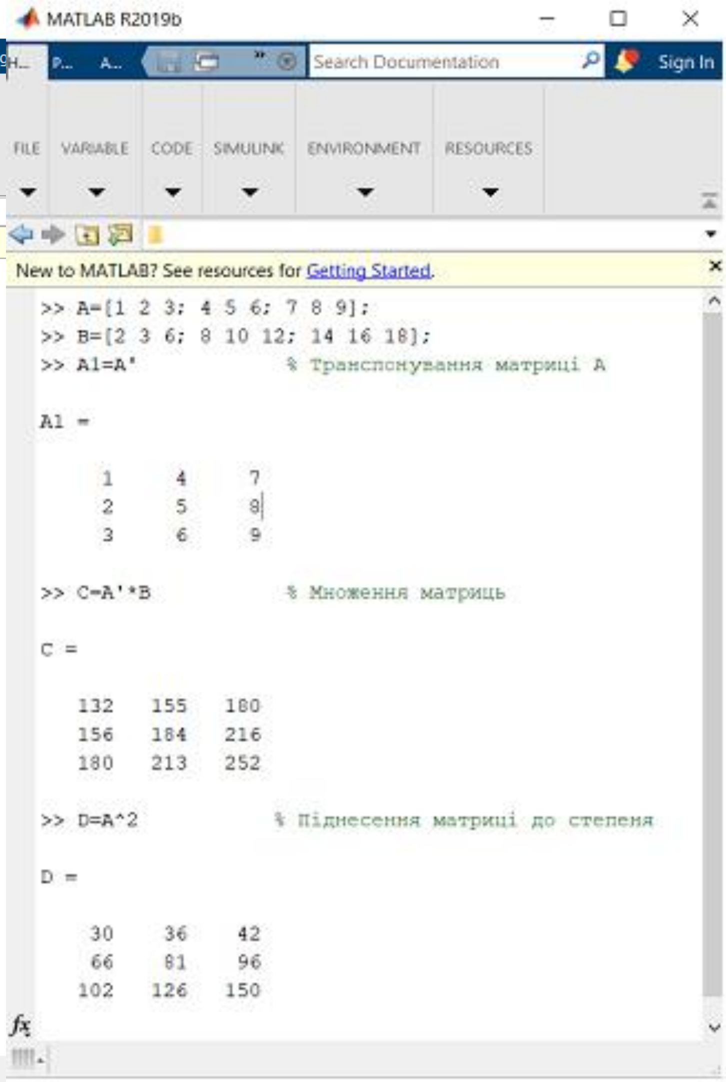


Рис. 4.5 Приклади транспонування матриці, множення матриць та піднесення матриці до степеня

У MatLab можна виконувати складні операції з матрицями:

- **Обчислення визначника матриці** здійснюється за допомогою функції ***det***(A). Матриця повинна бути квадратна (рис. 4.6).
- **Функція обернення матриці** ***inv***(A) – обчислює матрицю, обернену до заданої матриці A. Для застосування цієї функції початкова матриця A повинна бути квадратною і її визначник не повинен дорівнювати нулю (рис. 4.6).

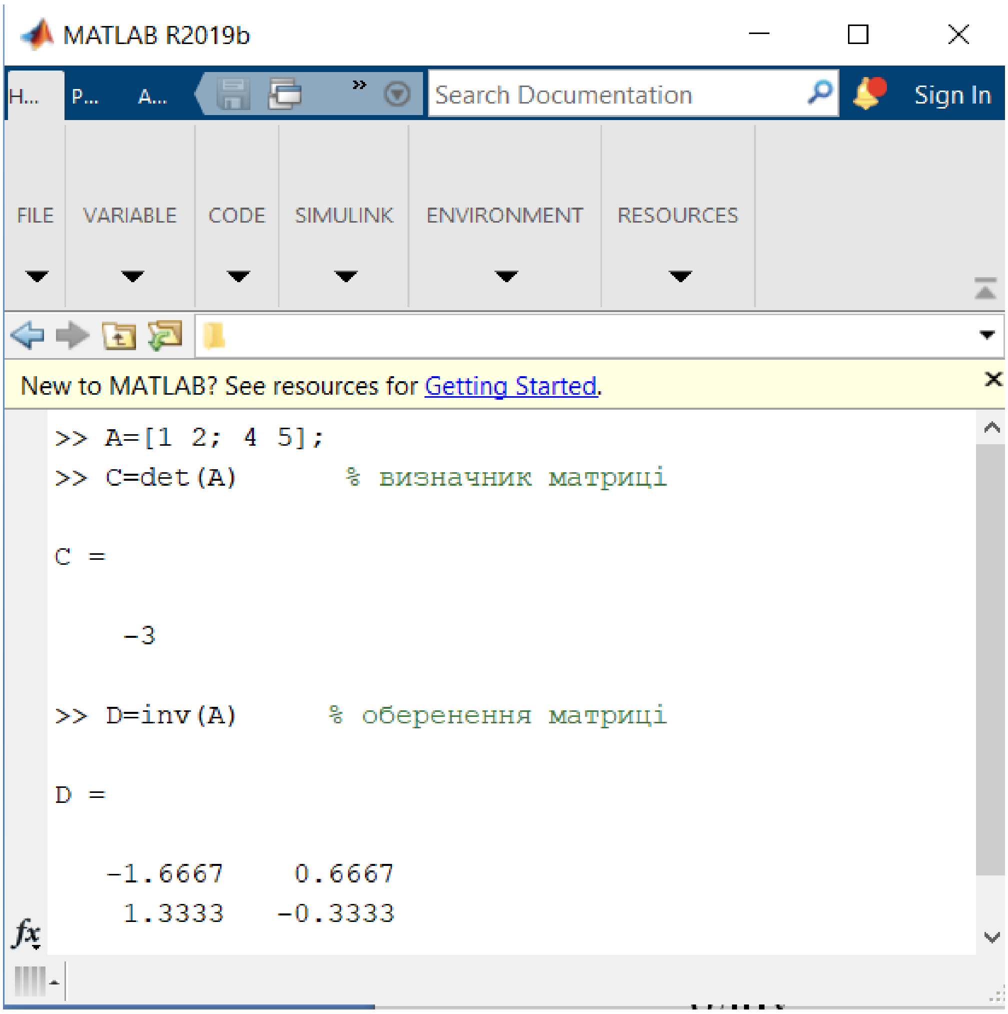


Рис. 4.6 Приклади обчислення визначника матриці та знаходження оберненої матриці

- **Ділення матриць зліва направо** та **ділення матриць справа наліво.** Перша операції записується за допомогою знака «/», а друга – за допомогою «\», що вміщується між іменами матриць, що діляться одна на одну (рис. 4.7). Також для матриць може застосовуватися операція поелементного ділення.
- **Обчислення матричної експоненти** $\left(e^{A}\right)$ здійснюється за допомогою функцій ***expm***, ***expm1***, ***expm2***, ***expm3***. Ці функції слід відрізняти від раніше розглянутої функції ***exp(A)***, яка формує матрицю, кожний елемент якої дорівнює e у степені, що дорівнює відповідному елементу матриці A.
- Функція ***logm(A)*** здійснює зворотну операції – логарифмування матриці за натуральною основою.

Приклад застосування функцій ***expm(A)*** та ***logm(A)*** представлений на рис. 4.8.

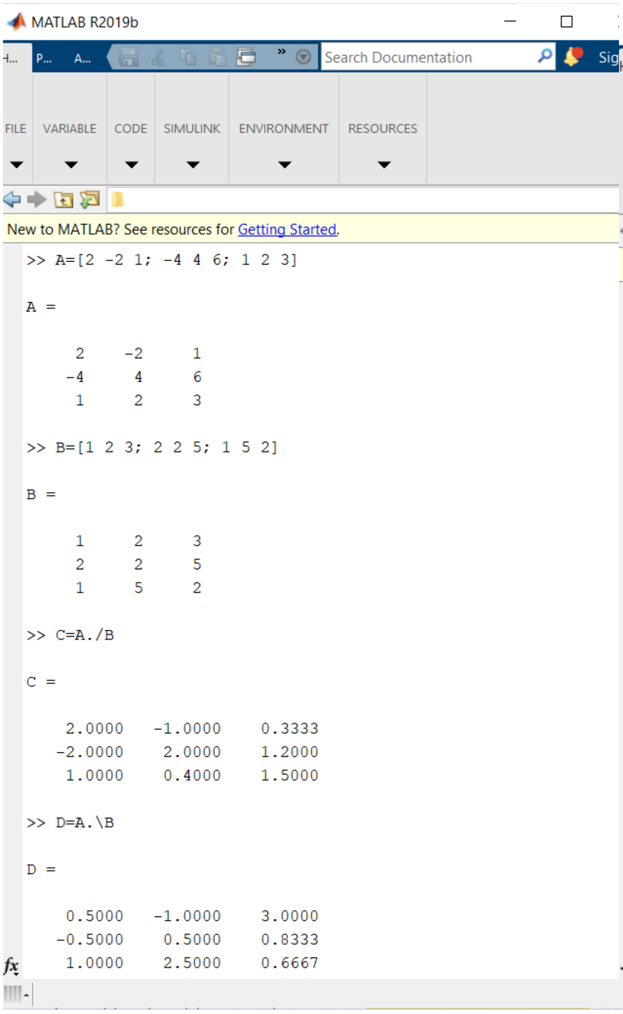


Рис. 4.7 Приклади поелементного ділення матриць зліва направо та ділення матриць справа наліво

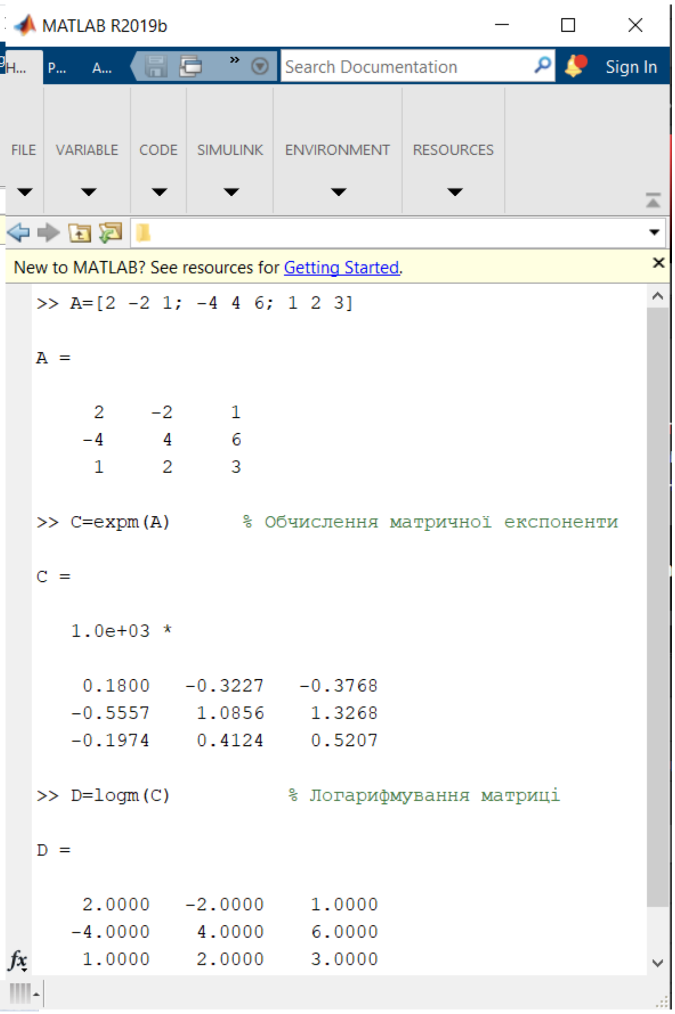


Рис. 4.8 Приклади обчислення матричної експоненти та логарифму

- Функція ***sqrtm(A)*** обчислює таку матрицю Y, що $Y*Y = A$ (рис. 4.9).
- Функція ***diag(x)*** формує або витягає діагональ матриці: якщо ***x*** – вектор, то функція утворює квадратну матрицю з вектором ***x*** на головній діагоналі (рис. 4.10).
- Функція ***trace(A)*** – знаходить слід матриці (сума діагональних елементів матриці A) (рис. 4.10).

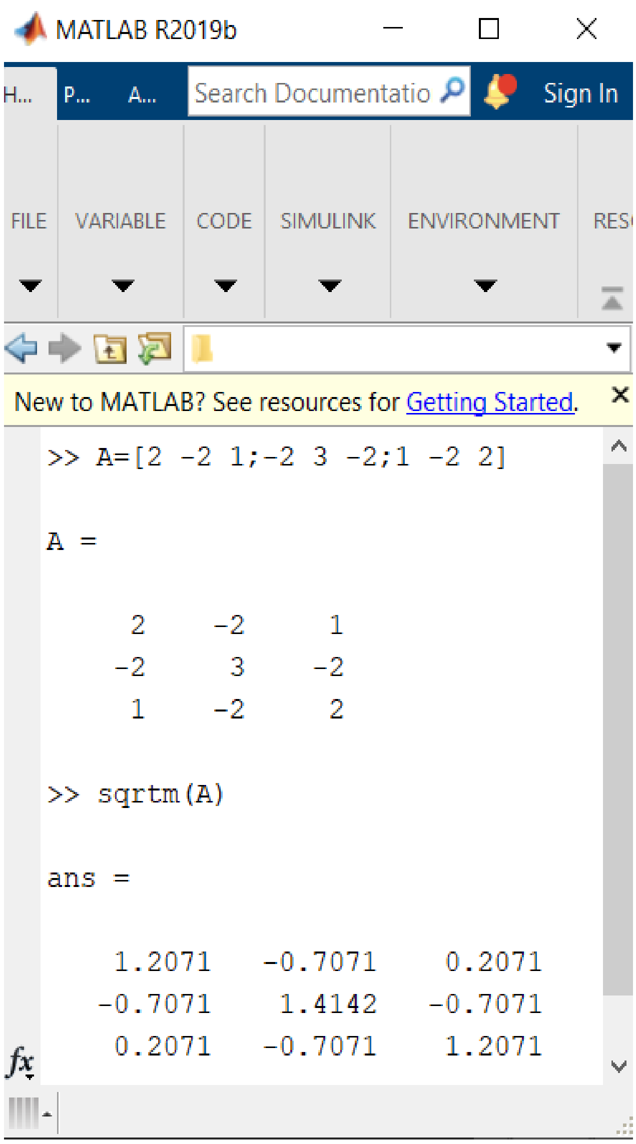


Рис. 4.9 Приклад використання функції ***sqrtm***

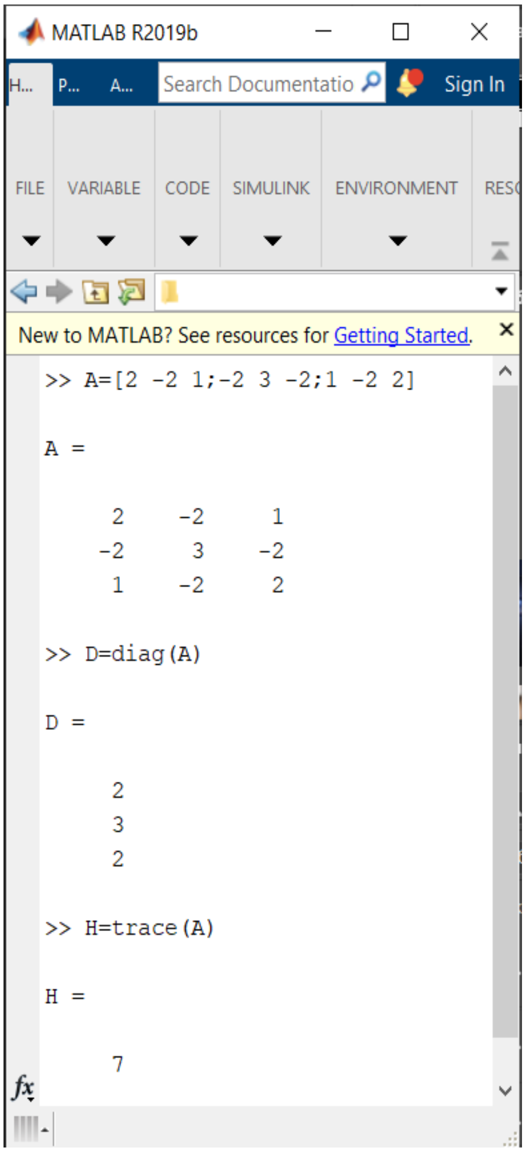


Рис. 4.10 Приклади використання функцій ***diag*** та ***trace***

- **Обчислення власних значень і власних векторів матриці** здійснює процедура ***eig***(A);
- Функція ***size***(A) – визначає розмір матриці A (рис. 4.11).
- **Основні функції математичної статистики**. В результаті застосування функцій ***max***, ***min***, ***mean***, ***std*** отримуються вектори-рядки з кількістю елементів, що дорівнює кількості стовпців заданої матриці. Кожен елемент містить, відповідно, максимальні, мінімальні, середні або середньоквадратичні елементи відповідного стовпця заданої матриці (рис. 4.11-4.12).

В усіх прикладах використовується матриця A:

$$A = \begin{bmatrix} 2 & -2 & 1 \\ -2 & 3 & -2 \end{bmatrix}.$$

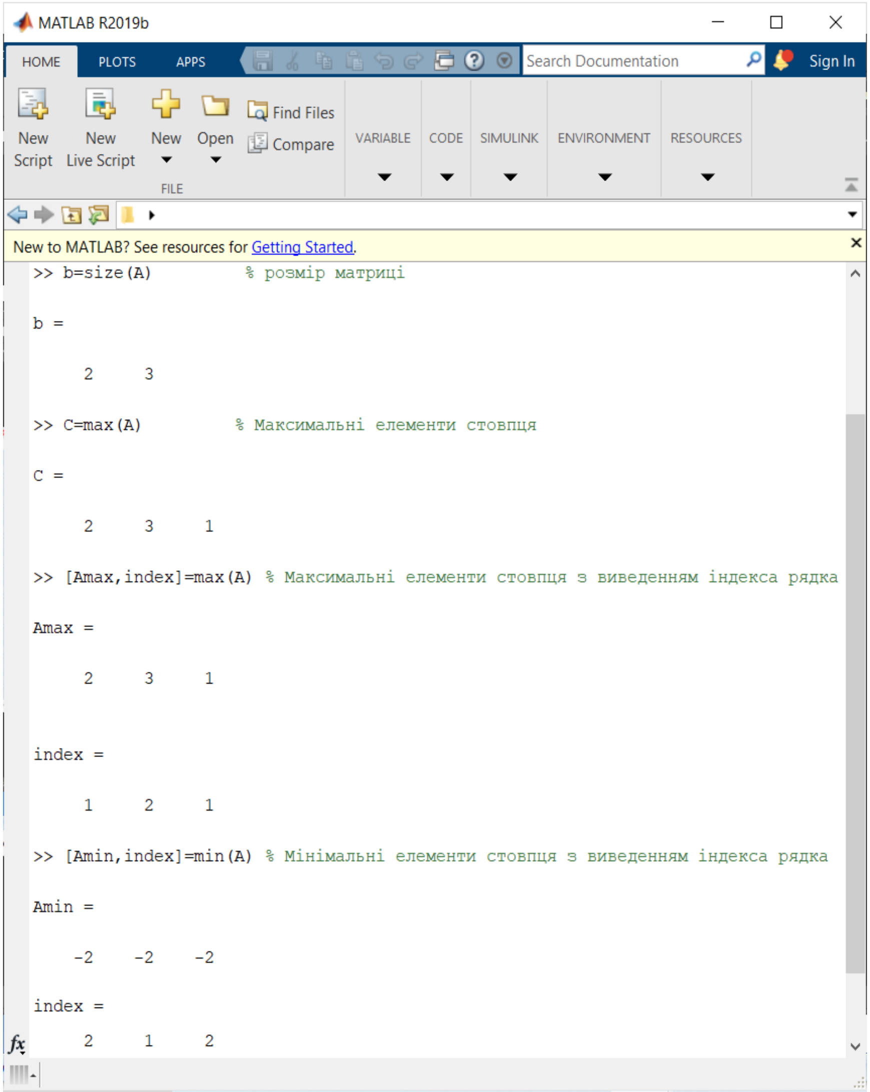


Рис. 4.11 Приклади використання функцій ***size, max*** та ***min***

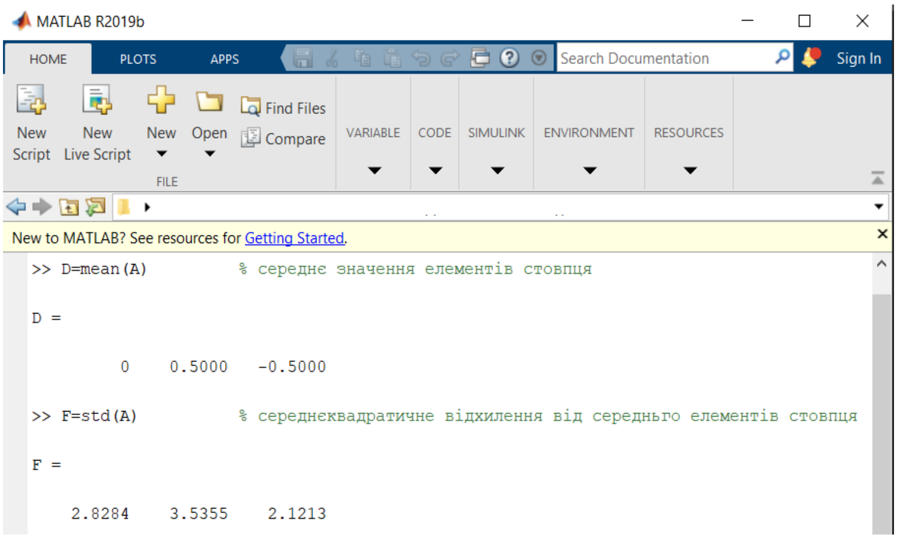


Рис. 4.12 Приклади використання функцій ***mean*** та ***std***

- Функція ***sort*** – сортує елементи кожного з стовпців початкової матриці. Результатом є матриця того ж розміру (рис. 4.13).

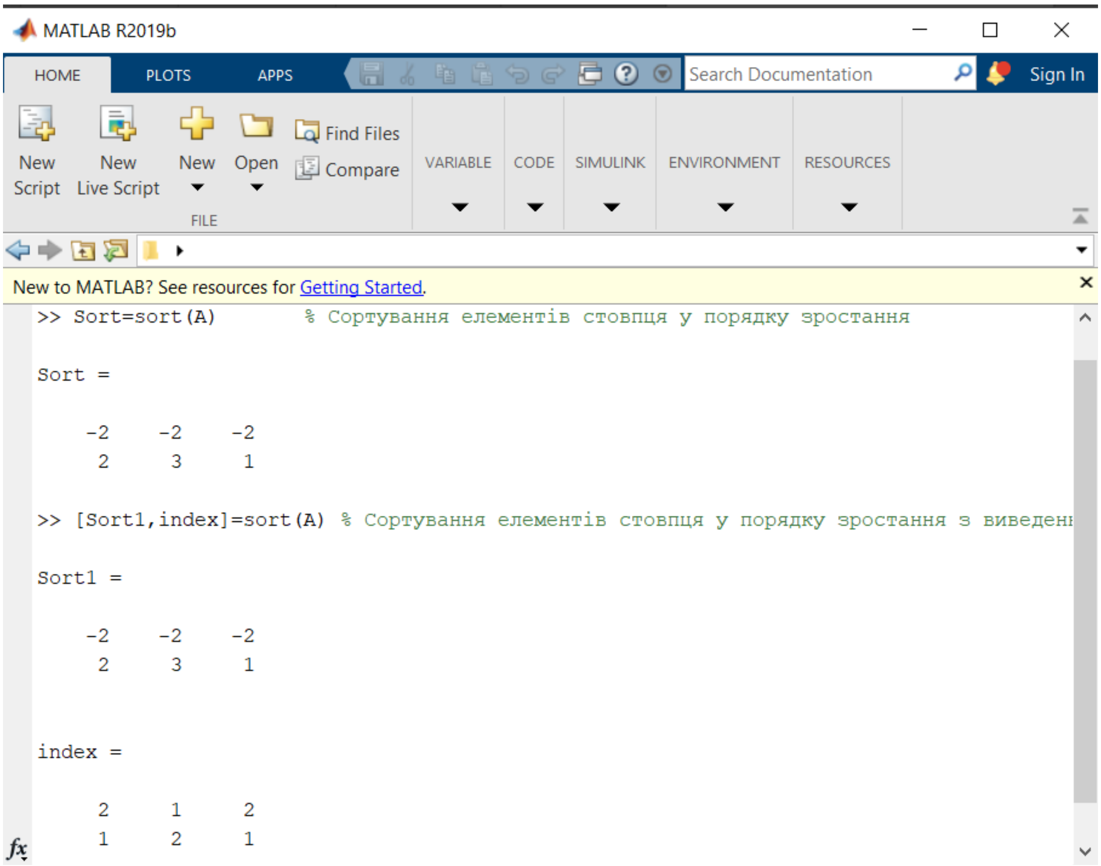


Рис. 4.13 Приклад використання функції ***sort***

- Функції ***sum*** і ***prod*** – формують вектор, кожний елемент якого є сумою або добутком елементів відповідного стовпця початкової матриці (рис. 4.14).

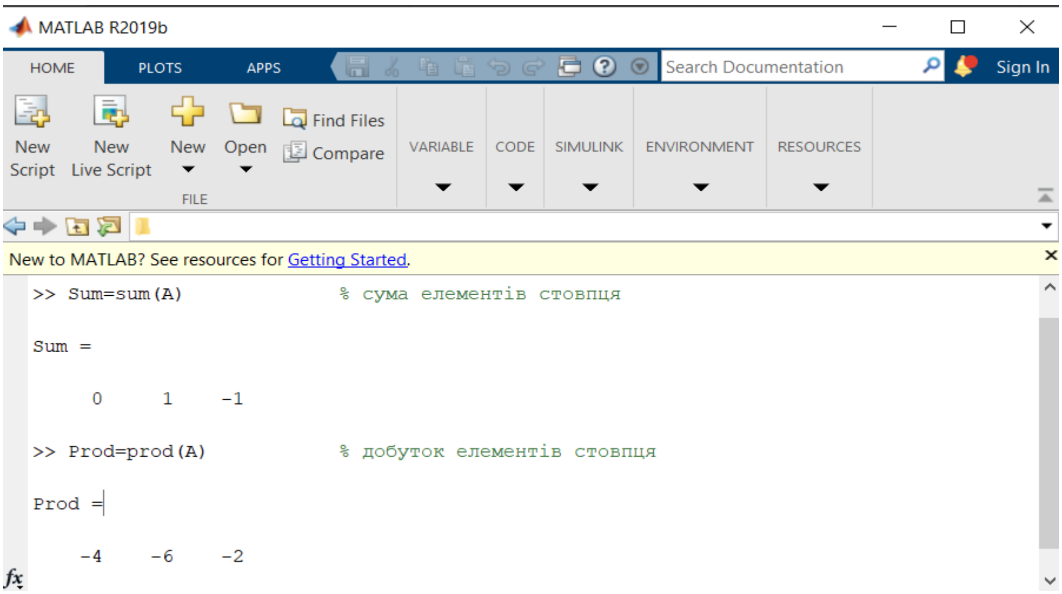


Рис. 4.14 Приклади використання функцій ***sum*** та ***prod***

- Функції ***cumsum*** і ***cumprod*** – утворюють матриці того ж розміру, елементи кожного стовпця яких утворюють суми або добутки елементів відповідного стовпця початкової матриці, які розташовані вище за відповідний елемент, включаючи і сам цей елемент (рис. 4.15).

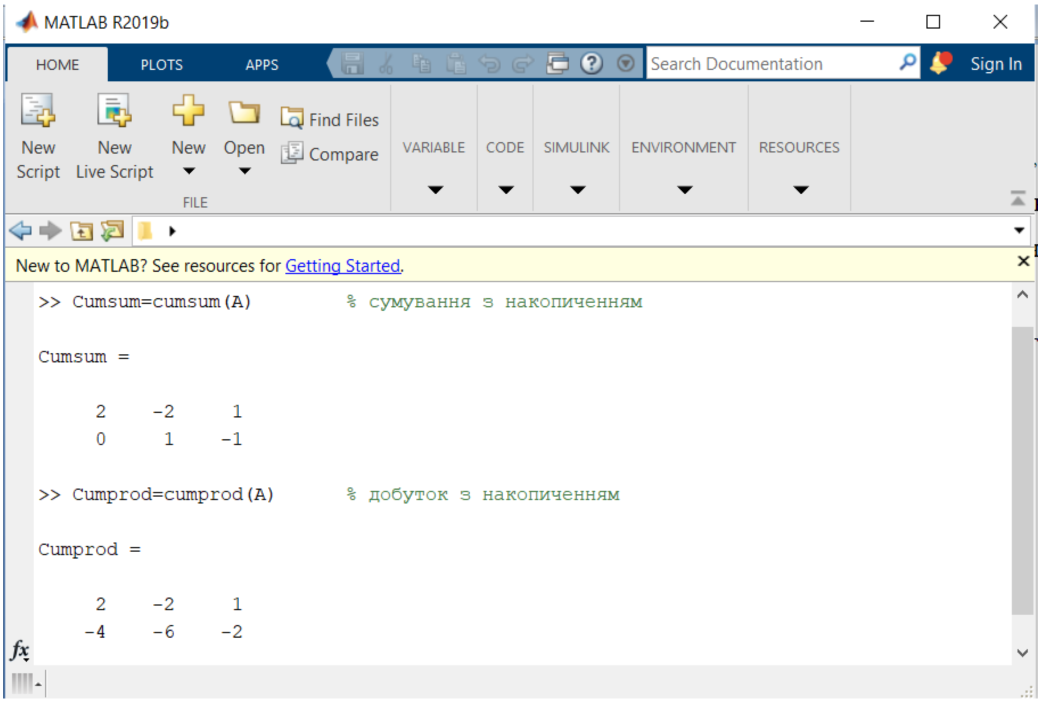


Рис. 4.15 Приклади використання функцій ***cumsum*** та ***cumprod***

- Функція ***diff*** – утворює з заданої матриці розміром $(m*n)$ матрицю розміром $((m-z)*n)$, елементи якої утворюються як різниця між суміжними рядками початкової матриці (рис. 4.16).

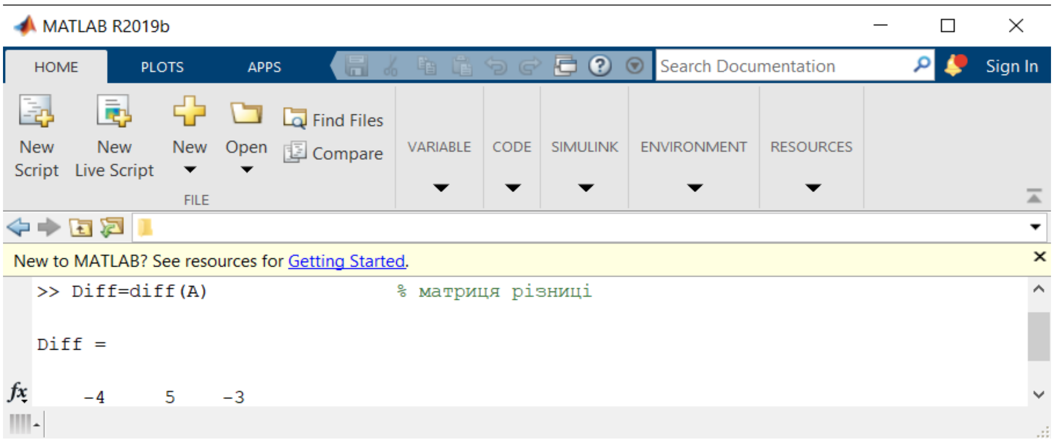


Рис. 4.16 Приклади використання функції ***diff***

Вище описані операції здійснюються відносно кожного з стовпців заданої матриці (за винятком ***size***). Тобто кожен стовпець матриці A розглядається як змінна, а кожний рядок – як спостереження.

Функція r=***rank***(A) обчислює ранг матриці, який визначається як кількість сингулярних чисел матриці. що перевищують поріг ***max(size***(A)***)*norm***(A)****eps.*** Де ***eps*** ($2{,}2204 \cdot 10^{-16}$) – похибка операцій над числами з плаваючою комою.

Формування векторів та підматриць в MatLab виконується наступним чином:

- Виділення вектор-рядка: $A(M,:)$, де M – номер рядка (рис. 4.17).
- Виділення вектор-стовпця: $A(:,N)$, де N – номер стовпця (рис. 4.17).
- Виділення підматриці з зазначенням граничних індексів: $A(M1\colon M2, N1\colon N2)$, де $M1\colon M2$ – номер рядків з $M1$ по $M2$ включно, $N1\colon N2$ – номер стовпців з $N1$ по $N2$ включно (рис. 4.18).
- Виділення підматриці з зазначенням початкових індексів: $A(M1\colon end, N1\colon end)$, де $M1\colon end$ – номер рядків з $M1$ по кінцевий елемент включно, $N1\colon end$ – номер стовпців з $N1$ по кінцевий елемент включно (рис. 4.18).
- Розтягування матриці у вектор-стовпець: $A(:)$ (рис. 4.19).
- Розтягування матриці у вектор-рядок: $(A(:))'$ (рис. 4.19).

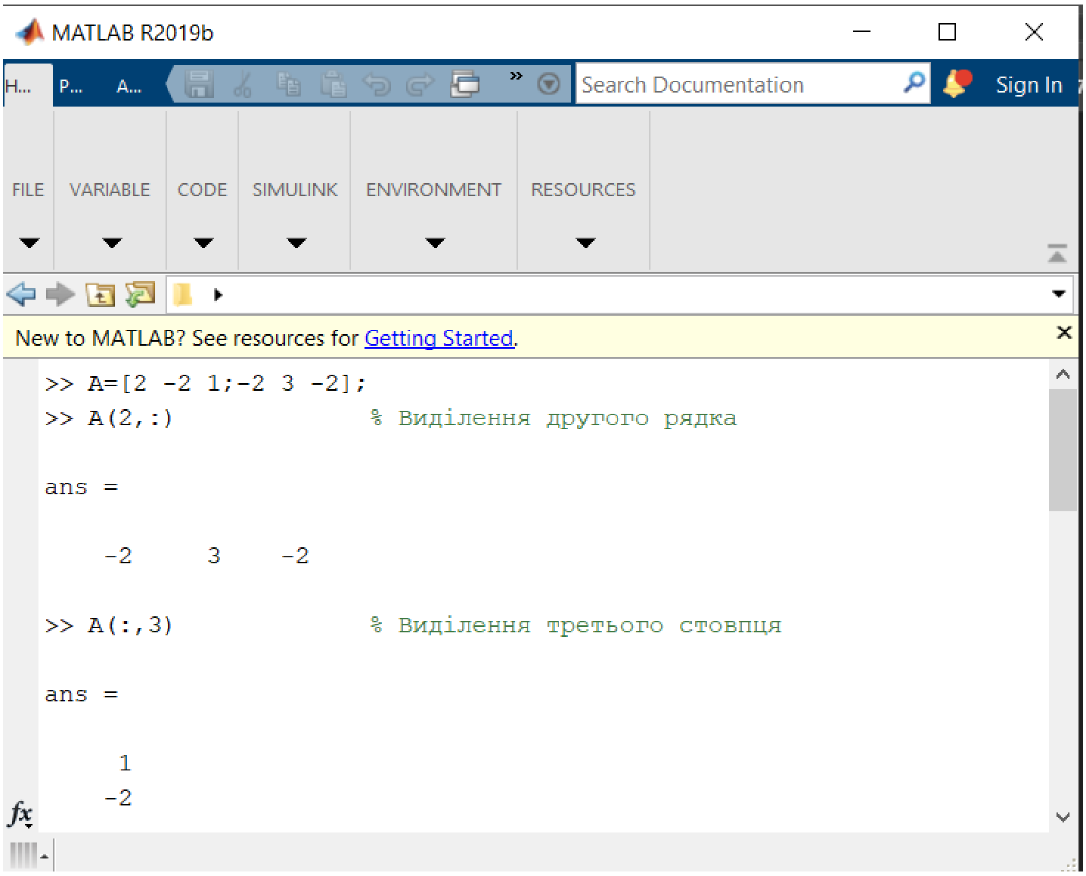


Рис. 4.17 Приклади виділення вектора-рядка та вектора-стовпця

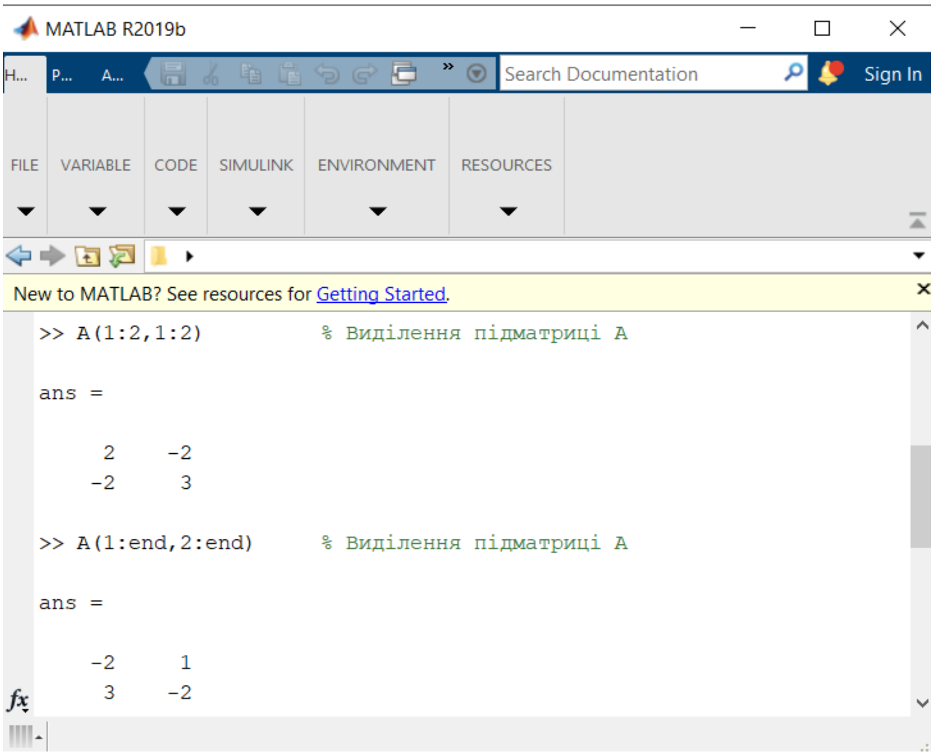


Рис. 4.18 Приклад виділення підматриці

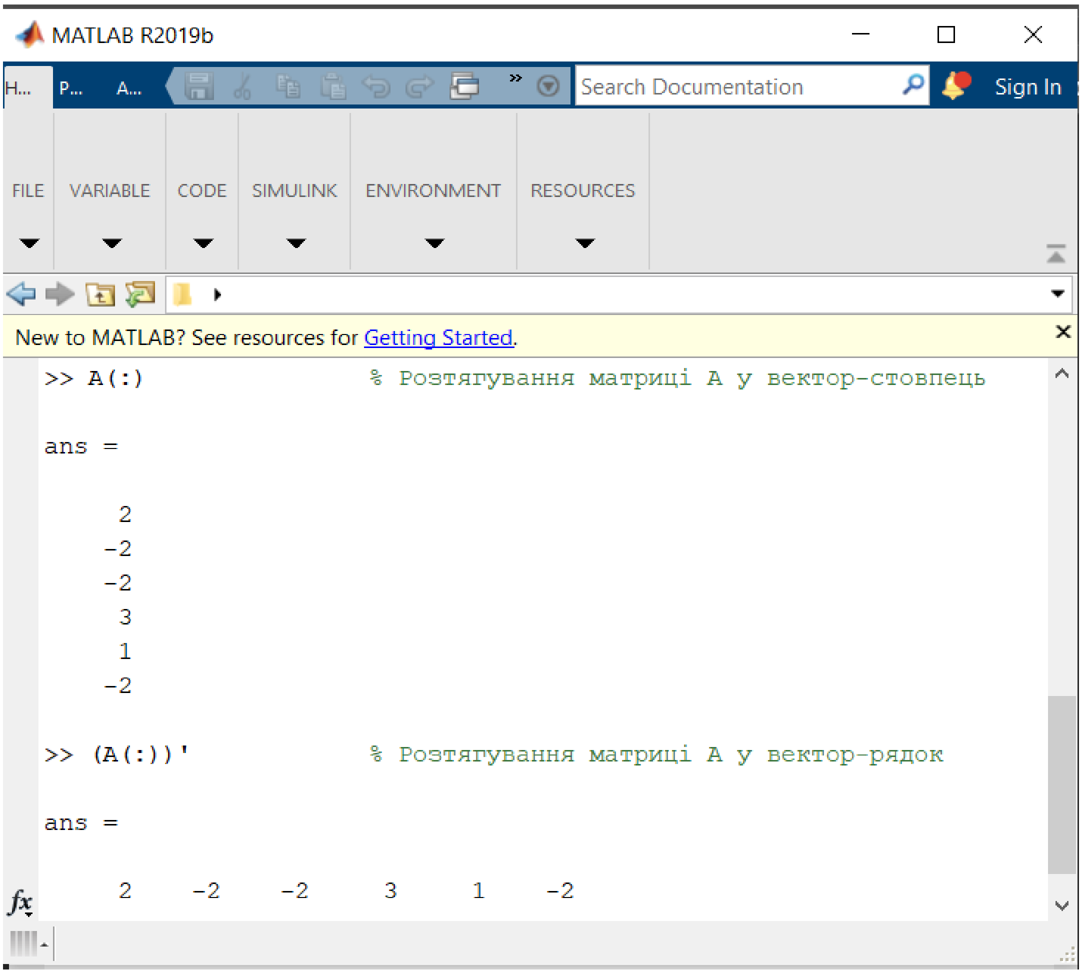


Рис. 4.19 Приклади розтягування матриці у вектор-рядок та вектор-стовпець

4.2. Завдання до виконання комп'ютерного практикуму

Завдання 1. Сформувати матрицю (масив значень) розміром $\boldsymbol{M \times N}$ елементами, якої будуть випадкові величини, які розподілені за рівномірним законом в інтервалі (0, v), де v – номер варіанту (див. табл.4.1).

1. Замінити значення елементу ***(M1,N1)*** на значення ***К***.
2. Заповнити перший рядок значеннями v.
3. Заповнити останній стовбець значеннями $v \cdot 5$.
4. Вивести останній рядок масиву елементів, отриманого в п.2.
5. Вивести перший стовбець масиву елементів, отриманого в п.3.

Таблиця 4.1

Варіанти індивідуальних завдань

Варіант	$\boldsymbol{M \times N}$	**(M1, N1)**	**К**
1	2	3	4
1.	3×5	(2, 4)	0,25
2.	7×4	(5, 2)	0,5
3.	4×6	(2, 5)	0,75
4.	2×5	(1, 3)	1
5.	3×7	(2, 5)	1,25
6.	4×8	(3, 6)	1,5
7.	5×6	(4, 5)	1,75
8.	6×5	(2, 4)	2
9.	7×4	(5, 2)	2,25
10.	6×7	(3, 5)	2,5
11.	5×6	(3, 4)	2,75
12.	4×8	(4, 7)	3
13.	6×7	(5, 3)	3,25
14.	5×8	(3, 7)	3,5

1	2	3	4
15.	6×4	(2, 2)	3,75
16.	5×9	(3, 8)	4
17.	7×3	(2, 1)	4,25
18.	5×7	(4, 6)	4,5
19.	4×8	(3, 7)	4,75
20.	3×6	(3, 5)	5
21.	6×8	(5, 7)	5,25
22.	5×7	(2, 6)	5,5
23.	4×6	(3, 5)	5,75
24.	5×8	(4, 7)	6
25.	6×7	(3, 7)	6,25
26.	5×8	(2, 6)	6,5
27.	7×5	(6, 2)	6,75
28.	6×8	(3, 3)	7
29.	4×5	(3, 4)	7,25
30.	6×7	(2, 5)	7,5

Завдання 2. Здійснити наступні операції над матрицями A і B.

1. Створити матриці A і B розміром (5x5) з випадковими значеннями в діапазоні від 0 до v (v – номер варіанту).
2. Знайти визначник матриць A та B.
3. Обчислити добуток матриць A та B. Перемножити поелементно матриці A та B, сформувавши матрицю C.
4. Поелементно поділити матрицю A на B.
5. Визначити максимальні та мінімальні значення стовпців матриць A та B з виведенням індексів. Для цього ретельніше вивчіть синтаксис виклику відповідних функцій. Користуйтесь записом у командному вікні типу – *help Function_Name*.

6. Знайти суму максимальних значень елементів рядків матриць A та B.

7. Обчислити суму найбільших елементів рядків матриць A та B.

8. Побудувати вектори, елементами якого є середньоквадратичне відхилення елементів відповідного стовпців матриць A та B від їхнього середнього значення.

9. Побудувати вектори, елементами якого є середньоарифметичне значення елементів відповідних стовпців матриць A та B.

10. Визначити розмір матриць A та B.

11. Побудувати вектор, елементами якого є елементи головної діагоналі матриць A та B.

12. Транспонувати матриць A та B.

13. Обчислити суму всіх елементів матриць A та B.

14. Знайти матричну та звичайну експоненту від матриць A та B.

15. Знайти логарифм від експоненти матриць, порівняти з вихідними матрицями A та B.

Завдання 3. Виконати наступні дії з формування векторів та підматриць:

1. Створити матриці A і B розміром (10x10) з випадковими значеннями в діапазоні від 10 до 100+v (v – номер варіанту).

2. Зробити перших 5 рядків матриці А рівними числу 21* v (v – номер варіанту)

3. Зробити останніх 3 стовпчики матриці В рівними числу першим 3 стовпцям матриці А.

4. Знайти суму підматриць $A(M1{:}\,M2, N1{:}\,N2)$ та $B(M1{:}\,M2, N1{:}\,N2)$, де $M1 = 2, M2 = 5, N1 = 3, N2 = 7$.

5. Перемножити матрицю D, що складається з $M3$ та $M4$ рядочка матриці A на матрицю F, що складається з $N3$ та $N4$ стовпчика матриці В. Номери рядків та стовпців визначаються за формулами

$M3 = 2 * ((42 + Year * \ v) \ mod \, 5)$,

$M4 = 3 * ((13 + Year * \ v) \, mod \, 3) + 1$,

$N3 = (42 + Year * (v - 1)) \ mod \, 7$,

$N4 = 3 * ((88 + Year * \ v) \, mod \, 3) + 1$, де ***mod*** – операція знаходження залишку після цілочисельного ділення.

6. Розтягніть матрицю D у вектор-стовпець, а матрицю F у вектор-рядок.

7. Поверніть на 90 градусів матрицю, яка складається з поелементного перемноження поелементної суми та поелементної різниці матриці *D* та матриці *F*.

8. Знайти суму діагональних елементів матриці *A* та присвоїти це значення змінній *Diag_Sum_A*. Знайти суму діагональних елементів матриці *B* та присвоїти це значення змінній *Diag_Sum_B*.

9. Замінити елементи головної діагоналі матриці *A* на значення змінної *Diag_Sum_A (використайте функції **tril**(A), **triu**(A), **diag()**, **diag**(**diag**(A)) та не забудьте сформувати відповідний вектор для нової головної діагоналі*). Замінити елементи побічної діагоналі матриці *B* на значення змінної *Diag_Sum_B (використайте прийом аналогічний для головної діагоналі, тільки перед цим поверніть матрицю B на 90 градусів проти годинникової стрілки. А після обнулення та задання нового значення для нової головної діагоналі, виконайте поворот ново отриманої матриці назад за допомогою функції* ***rot90(New_Matrix_Name, 3)*** *).*

4.3. Контрольні запитання

1. Що таке матриця? Чим матриця відрізняється від вектора?
2. Чи є результат виконання команди *a=1* у командному вікні матрицею?
3. Як сформувати вектор розміром 13x1, що складається лише з нулів? Лише з одиниць?
4. Яка матриця називається верхньотрикутною? Нижньотрикутною?
5. Назвіть базові операції над матрицями.
6. Які операції називаються матричними?
7. Що значить транспонувати матрицю? Чи можна транспонувати вектор-рядок? Що отримаємо в результаті?
8. Чи можна транспонувати вектор-стовпець? Що отримаємо в результаті?
9. Як визначити розмір матриці?
10. Що таке визначник? Як його можна обчислити за допомогою MatLab?
11. Чи кожна матриця має визначник? Чи можна знайти матрицю обернену до матриці розміром 3x2?
12. Як витягнути з матриці вектор, що складається з її діагональних елементів?
13. Як сформувати діагональну матрицю, маючи вектор?

14. Функції ***max***, ***min***, ***mean***, ***std*** визначають значення по стовпцях чи по рядках матриці?

15. За що відповідають отримані два числа, в результаті виклику функції **size** до матриці?

16. Як виділити вектор-рядок з матриці? Для чого застосовують символ «:»?

17. Поясніть як сформувати підматрицю? Дайте загальний формат та покажіть на прикладі матриці G розміром 6х6.

18. За що відповідає службове слово «*end*» при формуванні підматриць та векторів?

19. Як розтягнути матрицю у вектор-рядок?

20. Як розтягнути матрицю у вектор-стовпець? Що значить символ «′»?

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 5
РОЗВ'ЯЗУВАННЯ СИСТЕМ ЛІНІЙНИХ АЛГЕБРАЇЧНИХ РІВНЯНЬ

Мета роботи: навчитися розв'язувати системи лінійних рівнянь за допомогою вбудованих функцій MatLab.

5.1. Теоретичні відомості

Система лінійних алгебричних рівнянь (СЛАР) *n*-го порядку має вигляд:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases}, \tag{5.1}$$

де x_i (i=1,2,..., n) – значення змінних, a_{ij} $(i,\ j = 1,2,\ldots,n)$ – коефіцієнти при змінних, b_{ij} $(i,\ j = 1,2,\ldots,n)$ – вільні члени.

Під розв'язуванням СЛАР розуміється відшукування таких значень змінних x_i, підстановка яких у кожне з n рівнянь, перетворює їх одночасно у тотожності.

Якщо використати матричні позначення:

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix}; \qquad B = \begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{bmatrix}; \qquad X = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix}, \tag{5.2}$$

то система рівнянь (5.1) може бути поданою у матричній формі у такий спосіб:

$$A \cdot X = B. \tag{5.3}$$

Розв'язок у матричній формі записується в системі MatLab наступним чином:

$$X = A \setminus B. \tag{5.4}$$

Якщо ж СЛАР записана у наступній матричній формі

$$X \cdot A = B,$$

то розв'язок відшукується записом команди в системі MatLab

X = A /B.

Приклад. Застосуємо засоби системи MatLab для розв'язку СЛАР 3-ого порядку матричним методом*:*

$$\begin{cases} 4x_1 + 2x_2 + 2x_3 = 4 \\ 2x_1 - 2x_2 = -4 \\ 6x_1 - 2x_2 + 4x_3 = 4 \end{cases} \tag{5.5}$$

Використовуючи матричні позначення відповідно до системи (5.2):

$$A = \begin{bmatrix} 4 & 2 & 2 \\ 2 & -2 & 0 \\ 6 & -2 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 4 \\ -4 \\ 4 \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{5.6}$$

У командному вікні MatLab матричний запис системи (5.6) матиме вигляд:

```
>> A = [4 2 2; 2 -2 0; 6 -2 4]    % Матриця коефіцієнтів при змінних x1,x2,x3

A =

     4     2     2
     2    -2     0
     6    -2     4

>> B = [4; -4; 4]                 % Матриця вільних членів

B =

     4
    -4
     4
```

З урахуванням виразу (5.4) обчислюються розв’язки системи (5.5) та в командному вікні відображається відповідь:

```
>> x=A\B

x =

    -1
     1
     3
```

Для перевірки, помножимо матрицю *A* на отриманий вектор розв'язків *x*:

```
>> b=A*x

b =

     4
    -4
     4
```

Як видно, отриманий вектор b – це вектор вільних членів, що підтверджує правильність виконаних обчислень.

До прямих методів ще відносяться: метод Крамера; метод Гаусса; метод Гаусса-Жордана, LU-розкладання та ін.

Метод Крамера застосовується до СЛАР, в яких кількість рівнянь дорівнює кількості невідомих. Даний метод використовує визначники.

СЛАР 2-ого порядку має вигляд:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases}. \tag{5.7}$$

Визначники системи (5.7) визначаються як

$$\Delta = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, \qquad \Delta_{x_1} = \begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{vmatrix}, \qquad \Delta_{x_2} = \begin{vmatrix} a_{11} & b_1 \\ a_{21} & b_2 \end{vmatrix}, \tag{5.8}$$

де Δ – визначник системи (5.7), який складається з коефіцієнтів при невідомих; Δ_{x_1} – визначник, який одержаний з визначника Δ шляхом заміни стовпця при x_1 на стовпець вільних членів; Δ_{x_2} – визначник, який одержаний з визначника Δ шляхом заміни стовпця при x_2 на стовпець вільних членів.

Якщо $\Delta \neq 0$, тоді система сумісна і визначена, тобто має один розв’язок, який знаходиться за методом Крамера:

$$x_1 = \frac{\Delta_{x_1}}{\Delta}, \quad x_2 = \frac{\Delta_{x_2}}{\Delta} \ . \tag{5.9}$$

Якщо $\Delta = 0$, а хоча б один з визначників Δ_{x_1}, Δ_{x_2} відмінний від нуля, тоді система не має розв’язків (несумісна).

Якщо $\Delta = \Delta_{x_1} = \Delta_{x_2} = 0$, тоді система має нескінченно багато розв’язків.

Приклад. Застосуємо засоби системи MatLab для розв'язку СЛАР 3-го порядку методом Крамера, використавши систему (5.5) та матричне позначення (5.6).

У командному вікні MatLab матричний запис системи (5.6) матиме вигляд:

```
>> A = [4 2 2; 2 -2 0; 6 -2 4]    % Матриця коефіцієнтів при змінних x1,x2,x3

A =

     4     2     2
     2    -2     0
     6    -2     4

>> B = [4; -4; 4]                      % Матриця вільних членів

B =

     4
    -4
     4
```

Далі в командному вікні задаються матриці для визначення визначників Δ_{x_1}, Δ_{x_2} та Δ_{x_3}, які одержані з визначника Δ матриці $\boldsymbol{A}$ шляхом заміни стовпця при x_1, x_2 та x_3 на стовпець вільних членів, відповідно.

```
>> A1 = A;
A2 = A;
A3 = A;
A1(:,1) = B
A2(:,2) = B
A3(:,3) = B

A1 =

     4     2     2
    -4    -2     0
     4    -2     4


A2 =

     4     4     2
     2    -4     0
     6     4     4

A3 =

     4     2     4
     2    -2    -4
     6    -2     4
```

Для обчислення визначників використовується вбудована функція ***det:***

```
>> disp('Визначник матриці A')
disp(det(A))
Визначник матриці A
  -32.0000

>>  disp('Визначник матриці A1')
disp(det(A1))
Визначник матриці A1
    32

>>  disp('Визначник матриці A2')
disp(det(A2))
Визначник матриці A2
  -32.0000

>>  disp('Визначник матриці A3')
disp(det(A3))
Визначник матриці A3
   -96
```

Якщо $\Delta \neq 0$ (див. ***det(A)***), тоді система сумісна і визначена, тобто має один розв'язок, який знаходиться за виразами (5.9). З урахуванням (5.9) в командному вікні відображається розв'язок системи (5.5):

```
>> x1 = det(A1) / det(A);
x2 = det(A2) / det(A);
x3 = det(A3) / det(A);
X=[x1;x2;x3]

X =

   -1.0000
    1.0000
    3.0000
```

Для підтвердження правильності обчислень проводиться перевірка:

```
>> disp(A*X - B)
     0
     0
     0
```

Метод Гаусса на відміну від метода Крамера застосовується до СЛАР з довільним числом рівнянь та невідомих, таким чином є універсальним методом. Метод полягає в послідовному виключені невідомих за допомогою елементарних перетворень над рівняннями СЛАР та складається з двох етапів:

1. Перетворення розширеної матриці.
2. Зворотній хід.

На першому етапі поділивши перше рівняння системи (5.1) на a_{11} (вважаючи, що $a_{11} \neq 0$) отримується

$$x_1 + a'_{12}x_2 + \cdots + a'_{1n}x_n = b'_1. \tag{5.10}$$

де $a'_{1i} = \frac{a_{1i}}{a_{11}}, b'_1 = \frac{b_i}{a_{11}},\ i = 2,3, \dots, n$.

Множачи рівняння (5.10) послідовно на $a_{21}, a_{31}, \dots, a_{m1}$ та віднімаючи отримане рівняння послідовно від другого, третього та т.д. m-ого рівняння, таким чином, виключаються з наступних рівнянь невідома x_1.

У результату отримується система рівнянь

$$\begin{cases} x_1 + a'_{12}x_2 + \cdots + a'_{1n}x_n = b'_1 \\ \qquad a'_{22}\, x_2 + \cdots + a'_{2n}x_n = b'_2 \\ \qquad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \qquad a'_{m2}x_2 + \cdots + a'_{mn}x_n = b'_m \end{cases} \tag{5.11}$$

Таким чином в СЛАР (5.11) послідовно виключаються невідомі x_2 з третього, четвертого та т.д.; x_3 з четвертого, п'ятого і т.д. У результаті виключень отримується СЛАР

$$\begin{cases} x_1 + a'_{12}x_2 + \cdots + a'_{1n}x_n = b'_1 \\ \quad x_2 + \cdots + a^{(2)}{}_{2n}x_n = b^{(2)}{}_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \qquad x_n = b^{(n)}{}_n \end{cases} \tag{5.12}$$

де $a_{2i}^{(2)} = \frac{a'_{2i}}{a'_{22}}, b_2^{(2)} = \frac{b'_i}{a'_{22}}, \; i = 3,4, \dots, n$,

$a_{mn}^{(n)} = \frac{a_{mn}^{(n-1)}}{a_{mn}^{(n-1)}}, b_n^{(n)} = \frac{b_n^{(n-1)}}{a_{mn}^{(n-1)}}.$

На другому етапі (зворотньому) вирішується система (5.12), тобто зворотній хід полягає у рішенні трикутної системи.

Приклад. Застосуємо засоби системи MatLab для розв'язку СЛАР 3-ого порядку методом Гаусса, використавши систему (5.5) та матричне позначення (5.6).

У командному вікні MatLab матричний запис системи (5.5) матиме вигляд:

```
>> A = [4 2 2; 2 -2 0; 6 -2 4]    % Матриця коефіцієнтів при змінних x1,x2,x3

A =

     4     2     2
     2    -2     0
     6    -2     4

>> B = [4; -4; 4]                 % Матриця вільних членів

B =

     4
    -4
     4
```

Далі записується в командному вікні розширена матриця *A*, яка матиме вигляд:

```
>> C=[A B]              % Формується розширина матриця

C =

     4     2     2     4
     2    -2     0    -4
     6    -2     4     4
```

За допомогою вбудованої функції ***rref***, яка повертає наведену до ступінчатого по рядках виду матриці, використовуючи метод виключення Гаусса з частковим вибором ведучого елемента, одержується розв’язок СЛАР (5.5).

```
>> G = rref(C)          % приведення до ступенчатого по рядках виду матриці C

G =

     1     0     0    -1
     0     1     0     1
     0     0     1     3

>> x = G(:,4)            % Останній стовпець є розв'язком системи рівнянь

x =

    -1
     1
     3
```

Таким чином, використання матричного методу і методів Крамера та Гаусса дають однакові результати.

Розглянуті методи є точними. Точні (або прямі) методи працюють достатньо швидко і широко застосовуються на практиці, якщо є достатніми обсяги пам'яті для їхньої реалізації.

Але існують й інші шляхи знаходження коренів СЛАР, які відносяться до наближених. Наприклад, метод ітерацій, метод Якобі, метод сполучених градієнтів, метод біспряжених градієнтів, метод Гаусса-Зейделя, стабілізований метод біспряжених градієнтів та ін. Вони полягають в знаходженні за скінченну кількість кроків (ітерацій) лише наближених розв'язків із заданою припустимою відносною похибкою.

Метод ітерацій. Нехай дана система алгебраїчних рівнянь (5.1). Виразивши x_1 – через перше рівняння системи (5.1), x_2 – через друге,…, x_n – через n-е рівняння, отримаємо наступну систему:

$$\begin{cases} x_1 = \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}} x_2 - \cdots - \frac{a_{1n}}{a_{11}} x_n \\ x_2 = \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}} x_1 - \cdots - \frac{a_{2n}}{a_{22}} x_n \\ \dots\dots\dots\dots\dots\dots \\ x_n = \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}} x_1 - \frac{a_{n2}}{a_{nn}} x_2 - \cdots - \frac{a_{n,n-1}}{a_{nn}} x_{n-1} \end{cases} \tag{5.13}$$

де $a_{ii} \neq 0,\ i = 1,2, \dots, n$.

Виконавши заміну $\beta_i = \frac{b_i}{a_{ii}}, \alpha_{ij} = -\frac{a_{ij}}{a_{ii}}, i = 1,2, \dots, n;\ j = 1,2, \dots, n$ система (5.12) приводиться до вигляду (**метод Якобі**):

$$\begin{cases} x_1 = \beta_1 + \alpha_{12} x_2 + \cdots + \alpha_{1n} x_n \\ x_2 = \beta_2 + \alpha_{21} x_1 + \cdots + \alpha_{2n} x_n \\ \dots\dots\dots\dots\dots\dots \\ x_n = \beta_n + \alpha_{n1} x_1 + \alpha_{n2} x_2 + \dots + \alpha_{n,n-1} x_{n-1} \end{cases} \tag{5.14}$$

СЛАР (5.13) в матричному вигляді має вигляд:

$$x = \beta + \alpha x$$

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}, \beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_n \end{pmatrix}, \alpha = \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1n} \\ \vdots & \ddots & \vdots \\ \alpha_{n1} & \cdots & \alpha_{nn} \end{pmatrix}. \qquad (5.15)$$

За нульове наближенні $x^{(0)}$ вектора невідомих приймається вектор правих частин $x^{(0)} = \beta$, тобто

$$\begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ \dots \\ x_n^{(0)} \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_n \end{bmatrix} \qquad (5.16)$$

Перше наближення$x^{(1)}$:

$$\begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ \dots \\ x_n^{(1)} \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_n \end{bmatrix} + \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1n} \\ \vdots & \ddots & \vdots \\ \alpha_{n1} & \cdots & \alpha_{nn} \end{pmatrix} \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ \dots \\ x_n^{(0)} \end{bmatrix} \qquad (5.17)$$

Відповідно записуються друге, третє наближення і т. д.

У результаті отримується ітераційна процедура знаходження розв’язку:

$$x^{(k)} = \beta + \alpha x^{(k-1)}, k = 1,2, \dots \qquad (5.18)$$

Метод простих ітерацій матиме вигляд:

$$\begin{cases} x^{(0)} = \beta \\ x^{(1)} = \beta + \alpha x^{(0)} \\ x^{(2)} = \beta + \alpha x^{(2)} \\ \quad\quad \dots \dots .. \\ x^{(k)} = \beta + \alpha x^{(k-1)} \end{cases} \qquad (5.19)$$

Критерієм закінчення пошуку рішення системи в методі Якобі :

$$\left| x^{(k)} - x^{(k-1)} \right| < \varepsilon , \qquad (5.20)$$

де $\varepsilon -$ задана точність.

Приклад. Застосуємо засоби системи MatLab для розв'язку СЛАР 3-го порядку методом Якобі, використавши систему (5.5) та матричне позначення (5.6).

У створеному m-файлі записується текст основної програми. В основній програмі задаються вектори *A* та *B*, точність *tol*, максимальна кількість ітерацій *max*.

Основна програма матиме вигляд:

```
clear
clc
A = [4 2 2; 2 -2 0; 6 -2 4];
B = [4; -4; 4];
tol=10^(-3);
max=100;
n = length (B); % Розрахунок довжини вектора B
x0 = zeros (n,1); % Обнулення початкового наближення
x=yakobi (A,B,tol,max,x0) % Застосування методу Якобі
```

В окремому M-файлі створюється підпрограма, яка матиме вигляд:

```
function [x,flag,relres,iter,resvec] = yakobi (A,B,tol,max,x0)
x = x0;
flag = 0;
N=length(B);
nB=norm(B,inf);          % Норма вектора B
r = B - A * x;           % Розрахунок невязки початкового наближення
nr = norm (r,inf);       % Норма невязки
resvec (1) = nr;         % результуючий вектор невязок
for iter = 1 : max       % iter - кількість ітерацій
 xk=x;
 for j=1:N
 x(j)=(B(j)-A(j,[1:j-1,j+1:N])*xk([1:j-1,j+1:N]))/A(j,j);
 end
 r = B - A * x;          % Розрахунок невязки нового наближення
 nr = norm (r,inf);      % Норма невязки
 resvec (iter+1) = nr;   % результуючий вектор невязок
 relres = nr / nB;       % Досягнута відносна норма невязки
 if relres <= tol        % Перевірка умови кінцевого процесу
 flag = 1;
 break;
 end
end
```

У командному вікні MatLab отримується результат:

```
x =

  -9.998326934874058e-01
   9.997796192765236e-01
   3.000474732369184e+00
```

Далі розраховується похибка методу Якобі:

```
disp('ПОХИБКА')
X=A\B;              % Точний метод
dX=x-X
```

```
ПОХИБКА

dX =

   1.673065125942230e-04
  -2.203807234764099e-04
   4.747323691844940e-04
```

MatLab має декілька вбудованих процедур, які дозволяють розв'язувати систему лінійних алгебричних рівнянь виду (5.1) наближеними ітераційними методами. Їх застосовують при обчисленнях тоді, коли матриці коефіцієнтів СЛАР є розрідженими і великими за розмірами. До них відносяться:

bicg – метод біспряжених градієнтів;

bicgstab – стабілізований метод біспряжених градієнтів;

cgs – квадратичний метод спряжених градієнтів;

gmres – узагальнений метод мінімального відхилення;

qmr – метод квазімінімального відхилення;

pcg – передобумовлений метод спряжених градієнтів (застосовується лише для симетричних матриць A).

Загальне звернення до цих процедур має вигляд

$$x = \boldsymbol{bicg}(A, B, tol, maxit),$$

де A – квадратна матриця розміром $(n \times n)$ коефіцієнтів при аргументах системи рівнянь (5.1), B – матриця-стовпець розміром $(n \times 1)$ вільних членів, tol - припустима гранична відносна похибка визначення коренів, $maxit$ – гранична припустима кількість ітерацій, x – вектор одержаних наближених значень коренів рівняння (5.1).

За початкове наближення обирається вектор $x0$ з нульових елементів.

Приклад. Застосуємо засоби системи MatLab для розв'язку СЛАР 3-го порядку різними методами, використавши систему (5.5) та матричне позначення (5.6). Методами, що розглядаються:

- метод біспряжених градієнтів;
- стабілізований метод біспряжених градієнтів;
- квадратичний метод спряжених градієнтів;
- узагальнений метод мінімального відхилу;
- метод квазімінімального відхилу, використовуючи систему (5.5) та матричне позначення (5.6).

У командному вікні MatLab матричний запис системи (5.9) матиме вигляд:

```
>> A = [4 2 2; 2 -2 0; 6 -2 4]    % Матриця коефіцієнтів при змінних x1,x2,x3

A =

     4     2     2
     2    -2     0
     6    -2     4

>> B = [4; -4; 4]                    % Матриця вільних членів

B =

     4
    -4
     4
```

За допомогою вбудованих функцій ***bicg, bicgstab, cgs, gmres, qmr*** одержуються наближенні розв'язки СЛАР (5.5).

```
>> format long g
X = A\B;                    % точний метод
X1 = bicg(A,B);             % метод біспряжених градієнтів
X2 = bicgstab(A,B);         % стабілізований метод біспряжених градієнтів
X3 = cgs(A,B);              % квадратичний метод спряжених градієнтів
X4 = gmres(A,B);            % узагальнений метод мінімального відхилу
X5 = qmr(A,B);              % метод квазімінімального відхилу
```

```
>> disp(['       Точне      ','      BICG   ','     BICGSATB    ' ,'    CGS    ','     GMRES    ','    QMR   '])
Y = [X X1 X2 X3 X4 X5]
disp('ПОХИБКИ')
disp(['       BICG    ','      BICGSATB    ' ,'    CGS     ','     GMRES    ','    QMR   '])
dY = [X1-X X2-X X3-X X4-X X5-X]
```

Система MatLab у командному вікні виведе результат у наступному вигляді:

```
                 Точне                   BICG                BICGSATB                   CGS                   GMRES                    QMR

Y =

   -1.000000000000000e+00    -1.000000000000000e+00                        0    -9.999999999999990e-01    -9.999999999999996e-01    -1.000000000000002e+00
    1.000000000000000e+00     1.000000000000000e+00                        0     9.999999999999912e-01     1.000000000000001e+00     1.000000000000003e+00
    3.000000000000000e+00     3.000000000000002e+00                        0     2.999999999999988e+00     2.999999999999999e+00     3.000000000000000e+00

ПОХИБКИ
                 BICG                 BICGSATB                  CGS                   GMRES                    QMR

dY =

   -4.440892098500626e-16     1.000000000000000e+00     9.992007221626409e-16     4.440892098500626e-16    -1.554312234475219e-15
    4.440892098500626e-16    -1.000000000000000e+00    -8.770761894538737e-15     6.661338147750939e-16     2.664535259100376e-15
    1.776356839400250e-15    -3.000000000000000e+00    -1.154631945610163e-14    -8.881784197001252e-16     4.440892098500626e-16
```

5.2. Завдання до виконання комп'ютерного практикуму

Завдання. Відшукати розв'язки системи лінійних алгебричних рівнянь, заданих у таблиці 5.1:

а) точними методами розв’язування систем лінійних алгебричних рівнянь (методом Крамера, Гаусса та матричним методом);

б) одним із ітераційних методів, розрахувати похибки;

в) порівняти результати.

Таблиця 5.1

Варіанти індивідуальних завдань

Варіант	**СЛАР**	**Варіант**	**СЛАР**
1	2	3	4
1.	$\begin{cases} 2x_1 + x_2 - x_3 = 3 \\ 3x_1 + 2x_2 + 2x_3 = -7 \\ x_1 + x_3 = -2 \end{cases}$	14	$\begin{cases} x_1 + 3x_2 - 11x_3 = -18 \\ 7x_1 + 2x_2 + 9x_3 = 16 \\ 3x_1 - 2x_2 + 7x_3 = 24 \end{cases}$
1	2	3	4
2.	$\begin{cases} 4x_1 + 3x_2 + 6x_3 = 10 \\ 5x_1 + 3x_2 + 6x_3 = 15 \\ 12x_1 + 7x_2 + 4x_3 = 2 \end{cases}$	15.	$\begin{cases} 4x_1 + 5x_2 + 7x_3 + x_4 = 3 \\ 5x_1 - 5x_2 - 2x_3 + x_4 = 15 \\ x_1 + 9x_2 + 13x_3 - 2x_4 = 2 \\ x_1 - 2x_2 + 14x_3 + 8x_4 = -8 \end{cases}$
3.	$\begin{cases} 11x_1 + 6x_2 = 21 \\ 4x_1 + 8x_2 + 3x_3 = -15 \\ 13x_1 + 9x_2 + 6x_3 = 25 \end{cases}$	16.	$\begin{cases} 4x_1 + 5x_2 + 4x_3 + x_4 = 4 \\ 5x_1 - 5x_2 + 2x_4 = 17 \\ 2x_1 + 5x_2 + 13x_3 - 2x_4 = 6 \\ x_1 - 2x_2 + 14x_3 + 8x_4 = -7 \end{cases}$
4.	$\begin{cases} 12x_1 + 9x_2 + 5x_3 = -21 \\ 12x_1 - 8x_3 = -15 \\ x_1 + 5x_2 + 8x_3 = 3 \end{cases}$	17.	$\begin{cases} 4x_1 + 3x_2 - 5x_3 + x_4 = 14 \\ -3x_1 + 8x_2 + 4x_3 + 7x_4 = -1 \\ 2x_1 - 2x_2 + 7x_4 = 9 \\ 6x_1 + 3x_2 + 2x_3 + 9x_4 = 25 \end{cases}$
5.	$\begin{cases} 13x_1 + 5x_2 - 3x_3 = 12 \\ 3x_1 - 2x_2 + x_3 = 5 \\ 7x_1 + 9x_2 - 8x_3 = 3 \end{cases}$	18.	$\begin{cases} 1{,}2x_1 - 2{,}4x_2 + 2{,}6x_3 = 2 \\ 2{,}2x_1 - 2{,}8x_2 = -3 \\ 5{,}6x_1 - 2{,}5x_2 + 4{,}3x_3 = 4 \end{cases}$
6.	$\begin{cases} x_1 - 2x_2 + 8x_3 = -5 \\ 6x_1 + 7x_3 = 21 \\ -3x_1 + 7x_2 + 5x_3 = -18 \end{cases}$	19.	$\begin{cases} 12x_1 - 53x_2 + 45x_3 = 45 \\ 17x_1 - 28x_2 + 6x_3 = -45 \\ 56x_1 - 17x_2 + 47x_3 = 87 \end{cases}$
7.	$\begin{cases} 17x_1 + 5x_2 - 7x_3 = 12 \\ 7x_1 - 2x_2 + 9x_3 = 15 \\ 5x_1 - 8x_2 + 2x_3 = -25 \end{cases}$	20.	$\begin{cases} 2{,}5x_1 - 4{,}2x_2 + 3{,}7x_3 = 5{,}5 \\ 3{,}6x_1 - 4{,}6x_2 + 2{,}6x_3 = 7{,}7 \\ 0{,}7x_1 + 4{,}8x_2 + 3{,}5x_3 = -5{,}8 \end{cases}$
8.	$\begin{cases} 26x_1 + 7x_2 - 4x_3 = -18 \\ 8x_1 - 3x_2 + 11x_3 = 42 \\ 9x_1 + 17x_3 = 3 \end{cases}$	21	$\begin{cases} 0{,}8x_1 + 5{,}9x_3 = 3 \\ 4{,}3x_1 + 1{,}6x_2 + 0{,}7x_3 = 6{,}8 \\ 4{,}7x_1 - 5{,}8x_2 + 6{,}5x_3 = 4{,}7 \end{cases}$

9.	$\begin{cases} 7x_1 - 12x_2 + 7x_3 = 26 \\ x_1 + 5x_2 + 9x_3 = 5 \\ 2x_1 - 6x_2 + x_3 = -12 \end{cases}$	22.	$\begin{cases} 2{,}8x_1 - 4{,}5x_2 + 7{,}9x_3 = 2{,}2 \\ 3{,}5x_1 - 5{,}7x_2 + 2{,}7x_3 = 4{,}8 \\ 7{,}7x_1 + 0{,}8x_2 = 7{,}9 \end{cases}$
10	$\begin{cases} 8x_1 - 5x_2 + 9x_3 = -5 \\ 9x_1 + 2x_2 + 11x_3 = 35 \\ x_1 - 6x_2 + x_3 = -4 \end{cases}$	23.	$\begin{cases} 1{,}9x_1 - 3{,}7x_2 + 3{,}8x_3 = 2{,}6 \\ 5{,}6x_1 + 3{,}5x_2 - 2{,}7x_3 = 3{,}6 \\ 5{,}4x_1 + 0{,}6x_3 = 3{,}8 \end{cases}$
11.	$\begin{cases} 5x_1 + x_2 - 4x_3 = 8 \\ x_1 + 7x_2 + 6x_3 = -12 \\ 4x_1 + 8x_2 + 12x_3 = 7 \end{cases}$	24.	$\begin{cases} 15x_1 - 37x_2 + 28x_3 = 23 \\ 12x_1 + 23x_2 - 15x_3 = 27 \\ 54x_1 + 16x_3 = 28 \end{cases}$
12.	$\begin{cases} 5x_1 - 8x_2 = 5 \\ 4x_1 + 7x_2 + 5x_3 = -9 \\ 4x_1 + 8x_2 - 4x_3 = 17 \end{cases}$	25.	$\begin{cases} 1{,}7x_1 + 4{,}7x_2 - 1{,}8x_3 = 4{,}7 \\ 1{,}8x_1 - 1{,}6x_2 + 2{,}7x_3 = -2{,}8 \\ -4{,}4x_1 - 7{,}4x_2 + 2{,}6x_3 = 3{,}6 \end{cases}$
13.	$\begin{cases} 4x_1 + 6x_2 - 3x_3 = 11 \\ 7x_1 - 2x_2 + 9x_3 = 1 \\ -6x_1 + 2x_2 + x_3 = -5 \end{cases}$	26.	$\begin{cases} 3{,}9x_1 + 2{,}4x_2 - 2{,}3x_3 = 5{,}2 \\ 2{,}5x_1 - 2{,}9x_2 + 3{,}8x_3 = 8{,}4 \\ 7{,}4x_1 + 3{,}4x_2 - 2{,}6x_3 = 2{,}7 \end{cases}$
1	2	3	4
27.	$\begin{cases} 3x_1 - 8x_2 + 6x_3 = 16 \\ 11x_1 + 4x_2 + 8x_3 = 8 \\ x_1 + 4x_3 = -15 \end{cases}$	29.	$\begin{cases} 2{,}4x_1 + 6{,}4x_2 - 3{,}2x_3 = 4{,}2 \\ 7{,}4x_1 - 1{,}5x_2 + 7{,}3x_3 = 4{,}6 \\ 5{,}2x_1 + 5{,}6x_2 - 7{,}9x_3 = -2{,}7 \end{cases}$
28.	$\begin{cases} 7x_1 - 7x_2 = 23 \\ 5x_1 + 9x_2 + 4x_3 = -14 \\ 6x_1 - 5x_2 + 13x_3 = 21 \end{cases}$	30	$\begin{cases} 3{,}2x_1 + 7{,}1x_2 - 4{,}2x_3 = 2{,}4 \\ 2{,}9x_1 - 3{,}6x_2 + 3{,}9x_3 = 3{,}8 \\ 4{,}2x_1 - 6{,}6x_2 + 4{,}8x_3 = -4{,}3 \end{cases}$

5.3. Контрольні запитання

1. Що таке система лінійних алгебричних рівнянь?

2. Що значить розв’язати СЛАР? Чи може СЛАР мати декілька розв’язків?

3. Приведіть приклад запису СЛАР у матричній формі.

4. Які методи розв’язку СЛАР називаються прямими? Перечисліть основні з них.

5. Як виконується розв’язок СЛАР методом Крамера? Що таке визначник?

6. Як обчислити визначник матриці? Чи кожна матриця має визначник? Навести приклади.

7. У якому випадку СЛАР вважається виродженою (не сумісна)? Скільки розв’язків має така СЛАР?

8. У чому полягає метод Гаусса при розв’язуванні СЛАР? Скільки етапів він має? Назвіть їх.

9. Яка функція MatLab реалізує метод Гаусса? Що вона приймає в якості аргументів? У якому вигляді отримується розв'язок СЛАР?

10. Які методи розв'язку СЛАР називаються наближеними? У чому вони полягають і як відрізняються від прямих методів?

11. Які наближені методи розв'язку СЛАР ви знаєте?

12. Охарактеризуйте розв'язування СЛАР з використанням методу ітерацій.

13. Як вигляд має загальний формат звернення до процедур наближеного розв'язку СЛАР в MatLab? Що вибирається за нульове приближення шуканого розв'язку.

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 6
РОЗВ'ЯЗУВАННЯ НЕЛІНІЙНИХ АЛГЕБРАЇЧНИХ РІВНЯНЬ

Мета роботи: навчитися розв'язувати нелінійні алгебраїчні рівняння за допомогою вбудованих функцій MatLab.

6.1. Теоретичні відомості

Нелінійним рівнянням називають алгебричне рівняння виду

$$f(x)=0, \tag{6.1}$$

де x – аргумент, а функція

$$y=f(x) \tag{6.2}$$

є деякою довільною неперервною алгебричною функцією цього аргументу.

Під *розв'язком* нелінійного рівняння (*нулем відповідної нелінійної функції* $f(x)$) розуміють таке значення аргументу, підстановка якого в рівняння (6.1) перетворює його у тотожність.

У даному комп'ютерному практикумі мова буде йти про відшукування *дійсних* нулів нелінійних рівнянь.

Загальних способів відшукання дійсних нулів довільних нелінійних рівнянь не існує. Нелінійна функція може не мати взагалі дійсних нулів, може мати один, два і навіть нескінченну їх кількість.

У загальному випадку процес відшукування нулів розподіляється на два етапи. Перший етап, дії за яким не можна формалізувати, пошук інтервалу змінювання аргументу, всередині якого існує єдиний нуль. Немає загальних правил, за якими можна було б установлювати такий (такі) інтервал. Перш за все це має бути аналітичне дослідження поводження заданої функції у різних інтервалах змінювання аргументу. Наприклад, якщо задана нелінійна функція є аналітичною, тобто необмежено диференційованою, то умови, за яких функція $y=f(x)$ має єдиний нуль всередині інтервалу $[a,b]$ змінювання аргументу x, мають бути наступними:

- значення функції на кінцях цього інтервалу повинні бути протилежного знаку

$$f(a)\cdot f(b)<0; \tag{6.3}$$

- перша похідна функції $f'(x)=\dfrac{df(x)}{dx}$ не змінює знаку на цьому інтервалі;

• друга похідна функції $f''(x) = \frac{d^2 f(x)}{dx^2}$ також не змінює знака на цьому інтервалі.

Другий етап – процес <u>*уточнення значення нуля*</u>. Відомі чисельні алгоритми відшукування нулів нелінійного рівняння, які будуть розглядатися у подальшому, насправді, є лише алгоритмами уточнення значення нуля, бо зводяться до послідовного звуження інтервалу $[a,b]$, всередині якого знаходиться нуль. Слід узяти до уваги, що встановлення інтервалу, всередині якого знаходиться єдиний нуль функції, фактично є визначенням наближеного значення цього нуля

$$x^* = \frac{a+b}{2} \quad . \tag{6.4}$$

При цьому похибкою такого значення може вважатися величина

$$\Delta x = \frac{|a-b|}{2} . \tag{6.5}$$

6.1.1. Метод дихотомії (ділення навпіл)

Метод полягає в наступному(рис.6.1). Заданий інтервал $[a,b]$ ділиться навпіл. Потім знаходиться наближене значення $x^* = \frac{a+b}{2}$ нуля функції. Обчислюється значення $y^* = f(x^*)$ функції при цьому значенні аргументу. Якщо воно дорівнює нулю, значить x^* є точним значенням нуля й процес закінчується. Якщо ні, то визначається знак значення y^*. Обирається той інтервал, на межах якого задана функція набуває значень протилежного знаку. Наприклад, якщо виявиться, що $f(x^*) \cdot f(a) < 0$, то як нове значення верхньої межі інтервалу приймається $b = x^*$. У протилежному випадку змінюється нижня межа інтервалу $a = x^*$. Далі процес повторюється для нового звуженого удвічі інтервалу $[a,b]$ доти, поки значення похибки (6.5) не стане меншою за задане припустиме її значення

$$\Delta x < \Delta x_{\text{доп}} \quad . \tag{6.6}$$

За остаточне значення нуля при цьому слід узяти значення (6.4).

Якщо обчислення потрібно проводити з максимальною точністю, процес звуження інтервалу слід продовжувати доти, поки нижня й верхня межі інтервалу $[a,b]$ не збіжаться у машинному поданні.

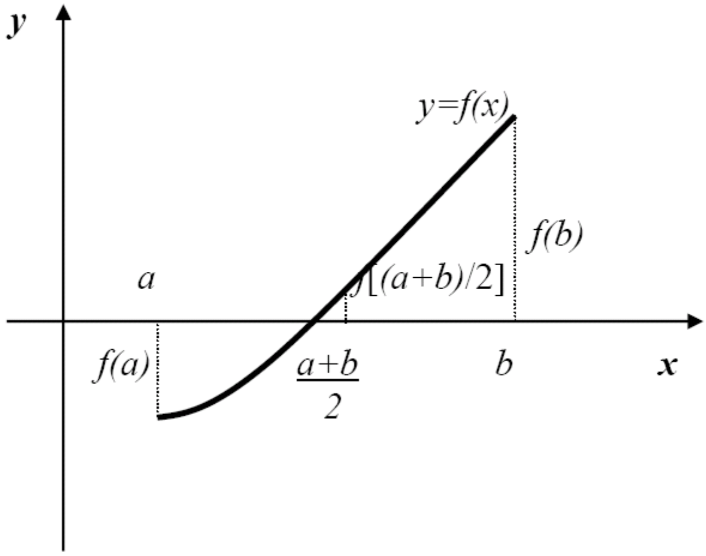


Рис. 6.1 Графічне представлення метода дихотомії

До переваг метода дихотомії слід віднести те, що він може бути застосований навіть до тих неперервних функцій, що є недиференційованими у деяких точках усередині заданого інтервалу визначення кореня.

6.1.2. Метод хорд

Метод полягає в наступному (рис.6.2). З’єднуються точки графіка функції, що відповідають кінцям заданого інтервалу $[a,b]$ прямою лінією. Відшукується точка перетинання цієї прямої з віссю аргументу – наближене значення $x*$ нуля функції. Обчислюється значення функції у цій точці $y* = f(x*)$. Далі, як і у методі дихотомії, визначається, у якому з двох інтервалів $[a, x*]$ або $[x*, b]$ міститься нуль і відповідно до того змінюються межі інтервалу, в першому випадку $b = x*$, у другому – $a = x*$.

Як випливає з рис. 6.2, на відміну від методу дихотомії, у методі хорд звуження інтервалу може бути не необмеженим (не збіжним до нуля). Одна з меж інтервалу може не змінюватися. Тому умова закінчення процесу тут має бути дещо іншою, а саме: на першому кроці методу слід запам'ятати перше наближене значення нуля x_1; здійснити другий крок і знайти друге наближене значення нуля x_2; відшукати різницю між цими нулями

$$\Delta x = |x_1 - x_2|. \tag{6.7}$$

Перевірити умову (6.6). Якщо її виконано, процес завершується. За значення нуля слід узяти останнє наближене значення x_2*. Якщо ні, слід замінити межі інтервалу, як вказано раніше, і замінити $x_1{*} = x_2{*}$. На наступному кроці наближення відшукується лише нове значення $x_2{*}$.

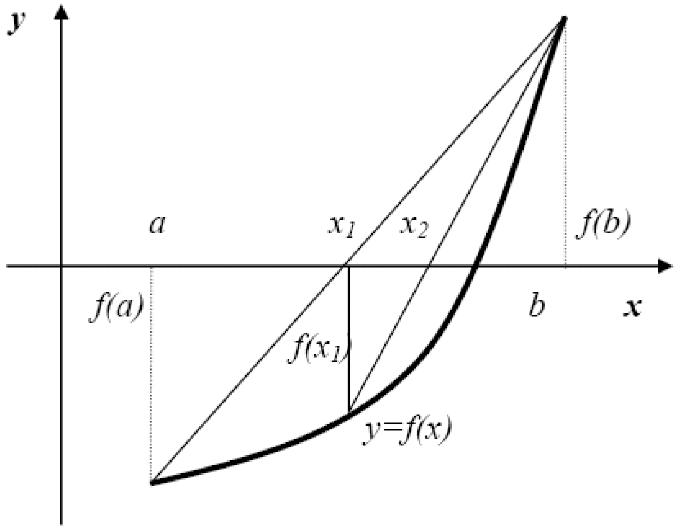


Рис. 6.2 Графічне представлення метода хорд

Отже, внаслідок однобічного наближення до кореня, критерієм близькості до кореня є не довжина поточного інтервалу, а близькість двох послідовних наближених значень кореня.

Сформулюємо математичнийо опис вищеприведеного алгоритму. У першу чергу необхідно обчислити значення функції $f(a)$ і $f(b)$ по краях інтервалу. Рівняння прямої, що проходить через точки $(a,\ f(a))$ і $(b,\ f(b))$ має вигляд:

$$\frac{y - f(b)}{f(a) - f(b)} = \frac{x - b}{a - b}.$$

Припускаючи у цьому рівнянні прямої, що $y = 0$, відшукаємо значення аргументу точки перетинання цієї прямої з віссю аргументу

$$x = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}\ . \qquad (6.8)$$

Метод хорд, як і метод дихотомії, пристосований і для недиференційованих функцій. Його недоліком є однобічність наближення до кореня.

6.1.3. Метод дотичних (Ньютона)

Наступний метод може бути застосований лише для диференційованих функцій і потребує не лише заданої функції *f(x)*, а й функції *f'(x)* її похідної.

Суть метода полягає в наступному (рис. 6.3). Проводиться пряма, що є дотичною до графіка заданої функції у точці одній з меж заданого інтервалу $[a,b]$. Відшукується значення x_1 аргументу точки перетинання цією прямою осі абсцис (аргументу). Обчислюється значення функції у цій точці $y_1 = f(x_1)$. Далі будується нова пряма, дотична до графіка функції у точці y_1, x_1. Відшукується точка перетину x_2, і процес повторюється.

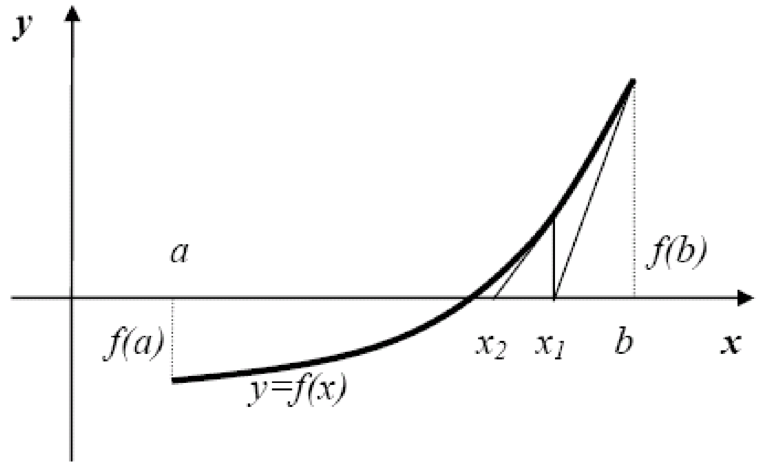


Рис. 6.3 Графічне представлення метода дотичних

Як і в методі хорд, процес наближення нулів переважно є однобічним. Тому, як і у попередньому випадку, наближення нулів слід оцінювати через близькість двох сусідніх послідовних наближених значень нулів.

Рівняння прямої, дотичної до кривої *f(x)* у точці $x=b$, має вигляд:

$$y - f(b) = f'(b) \cdot (x - b).$$

Звідси точка перетину цією прямою осі абсцис визначиться формулою:

$$x_1 = b - \frac{f(b)}{f'(b)}. \qquad (6.9)$$

Перевагою метода Ньютона є велика швидкість збіжності наближеного значення кореня до істинного. Недоліками є пристосовуваність лише для диференційованих функцій, необхідність задавати додаткової функції – похідної від заданої, однобічність наближення коренів.

6.1.4. Комбінований метод (хорд і дотичних)

Як неважко впевнитися однобічні методи хорд і дотичних наближують до істинного значення нуля із протилежних боків. Тому можна суттєво поліпшити чисельне обчислення нулів, якщо поєднати ці два методи у єдиний метод.

Дійсно, якщо на одному кроці застосувати обидва методи, одержується вже звужений із двох боків новий інтервал змінювання аргументу, всередині якого міститься корінь рівняння.

Метод полягає в наступному (рис. 6.4). На кожному кроці метода здійснюються дві операції: обчислення наближеного значення x_1* кореня за формулою (6.8) метода хорд і обчислення другого наближеного значення x_2* того самого кореня за формулою (6.9) метода Ньютона. Установлюються нові межі інтервалу аргументу: за межу a обирається менше з цих двох значень, за межу b – більше. Далі процес продовжується до виконання умови (6.6).

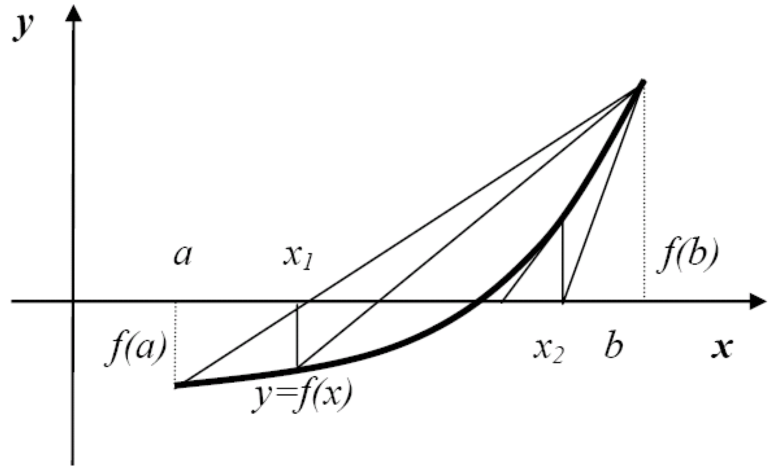


Рис. 6.4 Графічне представлення методу хорд і дотичних

До переваг метода відноситься велика швидкість збіжності значень кореня до істинного значення.

Недолік – пристосовуваність лише до диференційованих функцій, необхідність знання функції-похідної.

6.1.5. Метод ітерацій

У методі ітерацій для уточнення значення нуля використовується ітераційна формула, яка пов'язує попереднє наближене значення нуля з його подальшим, більш точним значенням.

Найбільш просто одержати ітераційну формулу, якщо попередньо подати вихідне нелінійне рівняння (6.1) у вигляді

$$x = \varphi(x) \qquad (6.10)$$

Перетворення рівняння (6.1) до форми (6.10) не є однозначним і визначається досвідченістю дослідника.

Щоб отримати графічне уявлення про сутність метода ітерацій, введемо позначення двох функцій

$$y_1(x) = x; \qquad y_2(x) = \varphi(x). \tag{6.11}$$

Тоді розв'язок рівняння (6.10) можна інтерпретувати як аргумент точки перетину графіків цих двох функцій, а саме – прямої $y_1(x) = x$ (бісектриси координатного кута) і нелінійної функції $y_2(x) = \varphi(x)$ (див. рис. 6.5).

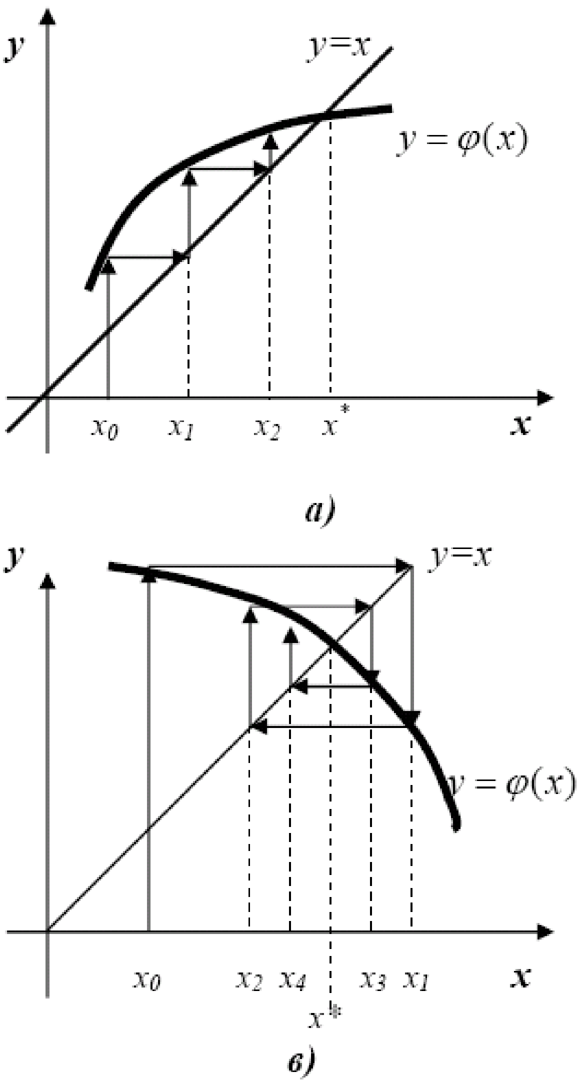


Рис. 6.5 Графічне представлення методу ітерацій

Власне відшукування нуля методом ітерацій у цьому випадку полягає у застосовуванні такої ітераційної формули, яка безпосередньо випливає з (6.10):

$$x_{i+1}(x) = \varphi(x_i), \tag{6.12}$$

де i – номер ітерації (наближення).

Метод полягає в наступному. Для використання ітераційної формули спочатку задаються певним значенням кореня x_0 усередині встановленого інтервалу існування єдиного кореня (точка на рис. 6.5). Обчислюють значення нелінійної функції при цьому значенні аргументу (рух удовж вертикальної лінії до перетинання із графіком кривої $y_2(x)=\varphi(x)$). Одержується точка 1 на графіку. У відповідності до ітераційної формули (6.12) за нове наближене значення кореня приймають одержане значення функції $x_1=y_{20}$ (рух удовж горизонтальної прямої до перетинання із прямою) $y_1(x)=x$. Одержується точка 2 графіка. Далі увесь процес повторюється доти, поки різниця між послідовними наближеними значеннями кореня не зменшиться до заданого рівня.

Як видно з рис. 6.5, успішність такого алгоритму забезпечується далеко не завжди. Лише за умови, що різниця між двома послідовними значеннями аргументу, одержаними при таких діях, зменшується за модулем, можна бути впевненим, що відбувається дійсно наближення до істинного кореня. Неважко впевнитися, що останнє можливе, якщо виконується співвідношення

$$\left|\frac{d\varphi(x)}{dx}\right|<1 \qquad (6.13)$$

у всіх точках інтервалу. Це й є *умова стійкості (збіжності) ітераційного процесу.*

Впливати на виконання цієї умови можна, по-різному формуючи нелінійну функцію $\varphi(x)$.

6.1.6. Реалізація процесу відшукування коренів у середовищі MatLab

Корені нелінійного рівняння (6.1) у середовищі MatLab можна відшукати за допомогою вбудованих функцій:

- ***fzero('ім'я функції', x0, tol, trace)*** – дозволяє знайти нуль функції в околиці точки *x0* або всередині інтервалу, що задається вектором [*a b*] із двох значень – нижньої й верхньої меж початкового інтервалу існування єдиного кореня; аргументи *tol* і *trace* можуть не вказуватися. Параметр *tol* задає значення максимальної припустимої похибки при обчисленні значення *x* шуканого кореня. Параметр *trace* вказує, що проміжні результати слід виводити на екран дисплея.
- ***fsolve('ім'я функції', x0)*** – розв'язок системи нелінійних рівнянь.

Приклад. Розв'язати нелінійне рівняння

$$x=-2e^{x}\,. \qquad (6.13)$$

Рівняння (6.13) в зручному вигляді буде мати вид:

$$x+2e^{x}=0\,. \qquad (6.13)$$

Створюємо m-файл, в якому за допомогою вбудованої функції ***plot(x, f)*** виводиться графік функції на екран (використовується графічний метод):

```
x=-2:0.01:1;
f=x+2*exp(x)
plot(x,f), grid
```

Більш детальне ознайомлення з можливостями побудови графіків в комп'ютерному практикумі №7.

На рис. 6.6 представлений графік функції $f(x) = x + 2e^x$.

З графіка видно, що корінь рівняння $f(x) = x + e^x$ знаходиться в межах $[-1, 0.5]$ (де значення функції $f(x) = 0$).

За допомогою функції ***fzero*** знаходиться корінь нелінійного рівняння на зазначеному відрізку

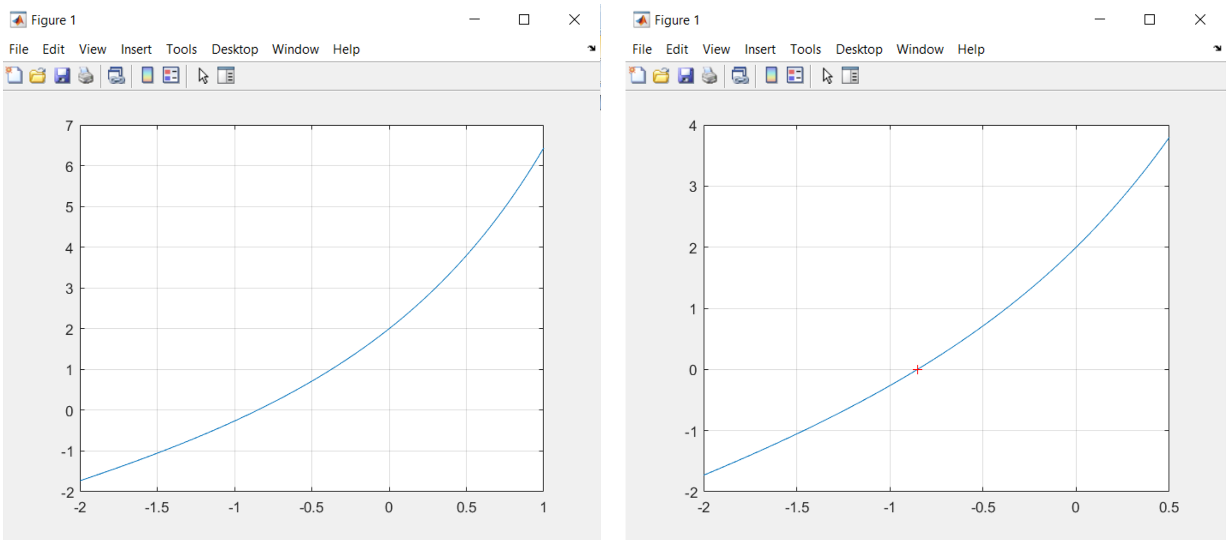
```
X1=fzero('x+2*exp(x)', [-1,0.5])
```

У командному вікні MatLab відображається результат:

```
X1=
-0.8526
```

Для перевірки на графік (рис. 6.6,а) наноситься значення отриманого коріння рівняння (рис. 6.6, б).

```
plot(x,f,X1,0,'r+'), grid
```



а) б)

Рис.6.6 Приклади побудови графіка функції $f(x) = x + 2e^x$

Для отримання проміжних результатів й інформації про точність результатів, треба вказати замість параметра *trace* будь-яке ціле додатне число:

```
options=optimset('Display', 'iter');
X1=fzero('x+2*exp(x)', [-1,0.5], options)
```

Тоді у командному вікні створиться таблиця

```
 Func-count     x          f(x)             Procedure
    2              -1       -0.264241        initial
    3       -0.902414       -0.091236        interpolation
    4       -0.852159     0.000827092        interpolation
    5       -0.852611    -9.43007e-06        interpolation
    6       -0.852606    -9.68716e-10        interpolation
    7       -0.852606     1.11022e-16        interpolation
    8       -0.852606     1.11022e-16        interpolation

Zero found in the interval [-1, 0.5]

X1 =

   -0.8526
```

Якщо для знаходження кореня рівняння вказати не інтервал, а наближене значення, наприклад, $x0 = -1$, тоді в m-файл записується

```
options=optimset('Display', 'iter');
X1=fzero('x+2*exp(x)', -1, options)
```

У результаті в командному вікні створюється наступна таблиця:

```
Search for an interval around -1 containing a sign change:
 Func-count    a          f(a)             b             f(b)        Procedure
    1             -1      -0.264241             -1      -0.264241    initial interval
    3      -0.971716      -0.214849       -1.02828      -0.313044    search
    5          -0.96      -0.194214          -1.04      -0.333091    search
    7      -0.943431      -0.164852       -1.05657      -0.361275    search
    9          -0.92      -0.122962          -1.08      -0.400809    search
   11      -0.886863     -0.0629708       -1.11314      -0.456084    search
   12          -0.84       0.023421       -1.11314      -0.456084    search

Search for a zero in the interval [-0.84, -1.11314]:
 Func-count    x          f(x)             Procedure
   12          -0.84       0.023421        initial
   13      -0.853341    -0.00136268        interpolation
   14      -0.852608   -3.95817e-06        interpolation
   15      -0.852606    1.07836e-12        interpolation
   16      -0.852606   -1.11022e-16        interpolation
   17      -0.852606   -1.11022e-16        interpolation

Zero found in the interval [-0.84, -1.11314]

X1 =

   -0.8526
```

Таким чином, загальна процедура відшукування кореня у випадку, коли задається не інтервал існування кореня, а його наближене значення, містить два етапи – пошук інтервалу, всередині якого є корінь, і потім уточнення значення кореня усередині знайденого інтервалу.

Число операцій значно скорочується, якщо замість одного наближеного значення кореня, як другий аргумент функції, вказати вектор із двох меж інтервалу, всередині якого міститься шуканий корінь. У цьому випадку необхідність у першому етапі – пошуку границь інтервалу – зникає.

Неважко впевнитися, що у випадку, коли відносна допустима похибка *tol* у явному вигляді при зверненні не вказується, система MatLab виконує обчислення нуля з максимально досяжною (машинною) точністю.

За допомогою функції ***fzero*** можна знайти корінь лише одного нелінійного рівняння.

Якщо ж потрібно відшукати корені системи з алгебричних рівнянь, доцільно користуватися процедурами ***solve*** і ***fsolve***.

Приклад. Відшукати корені системи нелінійних рівнянь *n*

$$\begin{cases} x_1 + 2x_2 + x_3 = 200 \\ 3x_1 + x_2 = -40 \\ x_1 + 2x_2 + 5x_3 = 2300 \end{cases}$$

Якщо тепер звернутися до процедури ***solve*** у такий спосіб:

```
syms x1 x2 x3
[x1,x2,x3]=solve(x1+2*x2+x3==200,3*x1+x2==-40,
x1+2*x2+5*x3==2300)
x1 =
 49
x2 =
-187
x3 =
 525
```

Щоб скористатися функцією ***fsolve***, потрібно створити m-файл процедури, яка обчислює вектор-стовпець $y=[y_1;y_2;y_3]$ з функцій, корені яких потрібно відшукати, за заданим (вхідним) вектором-стовпцем $x=[x_1;x_2;x_3]$. Назвемо процедуру ***example***:

```
function F=example(x)
F=[x(1)+2*x(2)+x(3)-200; 3*x(1)+x(2)+40;
x(1)+2*x(2)+5*x(3)-2300];
end
```

Тепер можна звернутися до процедури ***fsolve*** у такий спосіб (попередньо ввівши вектор стовпець початкових значень шуканих коренів x0):

```
fun=@example;
x0=[-1; -1; -1];
X=fsolve(fun,x0)
```

Результатом буде поява у командному вікні вектора коренів

```
X =
   49.0000
 -187.0000
  525.0000
```

Як бачимо, процедура ***fsolve*** реалізує ітераційний процес і відшукує лише дійсні корені, а ***solve*** визначає усі корені, у тому числі й комплексні.

6.2. Завдання до виконання комп'ютерного практикуму

Завдання 1. Відшукати найменший за модулем дійсний корінь рівняння ***f(x)=0*** за допомогою процедур ***fzero*** відповідно до варіанту (табл. 6.1). Результати порівняти.

Таблиця 6.1

Варіанти індивідуальних завдань

Варіант	Рівняння	Варіант	Рівняння
1.	$x^2 - 2x + \ln(x) = 0$	16.	$x^5 + 18x^3 - 34 = 0$
2.	$x^2 + arctg(x) - 0{,}5 = 0$	17.	$x^3 - 3x - 23 = 0$
3.	$x^3 - 2\cos(\pi x) = 0$	18.	$tg(1{,}2x) + 3x - 2 = 0$
4.	$tg(0{,}8x) - x - 2 = 0$	19.	$x^3 + 2x^2 - 11 = 0$
5.	$ctg(0{,}8x) - 2x^2 = 0$	20.	$2^x - 2x^2 - 3 = 0$
6.	$x^3 + 3x^2 - 6x - 1 = 0$	21.	$3 - x^3 + sin\left(\frac{\pi}{2}x\right) = 0$
7.	$x^2 - 2\ln(x+2) = 0$	22.	$x^2 - 1 - cos(1{,}2x) = 0$
8.	$2e^{-x^2} - 3x + 4 = 0$	23.	$x^3 - 2x - 15 = 0$
9.	$x^4 + 3x - 3 = 0$	24.	$x \cdot e^{2x} - 4 = 0$
10.	$x^4 - 6x^2 + 12x - 8 = 0$	25.	$x^3 + 2x^2 - 3x + 3 = 0$
11.	$(x-3)^2 - e^x - 2 = 0$	26.	$2x \cdot e^{x^2} - 5 = 0$
12.	$2x - 3\sin(2x) - 1 = 0$	27.	$x + 2\sin(5x) - 3x + 3 = 0$
13.	$5^x - 7x^2 + 8 = 0$	28.	$(x - 0{,}5)^2 - sin(\pi x) = 0$
14.	$(x-1)^2 - 0{,}5e^x = 0$	29.	$x^3 + 5x^2 + 5x - 8 = 0$
15.	$x + e^x + 5 = 0$	30.	$(x-2)^2 - e^x = 0$

Завдання 2. Відшукати рішення системи нелінійних рівнянь за допомогою процедур ***solve*** і ***fsolve*** відповідно до варіанту (табл. 6.2). Результати порівняти.

Таблиця 6.2

Варіанти індивідуальних завдань

Варіант	**Система рівнянь**	**Початкові наближення**
1	2	3
1.	$\begin{cases} 2x_1^4 + x_2^2 = 8 \\ x_1^3 + x_2^2 = 14 \end{cases}$	$x_1 = -1;$ $x_2=3$
2.	$\begin{cases} 2\sin x_1 + x_2 = -5{,}3 \\ 3\cos x_2 + 5x_1 = -3{,}5 \end{cases}$	$x_1 = -0{,}3;$ $x_2=-5$
3.	$\begin{cases} 3x_1^5 + 2x_2^3 - 8x_1 = -15 \\ 3x_1^5 - 2x_2^5 - 7x_2 = 23 \end{cases}$	$x_1 = 1;$ $x_2 = -1$
4.	$\begin{cases} (x_1^2 - 1)x_2 = 9 \\ x_1x_2 + x_1^2 + 10 = 0 \end{cases}$	$x_1 = -2;$ $x_2=3$
5.	$\begin{cases} \sin(x_1 + x_2) - 2{,}4x_1 = -3{,}2 \\ 2x_1^2 + x_2^2 - x_1 = 10 \end{cases}$	$x_1 = 1;$ $x_2 = -3$
6.	$\begin{cases} tg(x_1 - x_2) - 4x_1 = 0 \\ x_1^2 + 2x_2^2 = 1 \end{cases}$	$x_1 = -0{,}5;$ $x_2=0{,}6$
7.	$\begin{cases} tg(-x_2) - 4x_1 = -3 \\ x_1^2 + 2x_2^2 = 4 \end{cases}$	$x_1 = 0{,}2;$ $x_2 = -1{,}2$
8.	$\begin{cases} 2x_1^2x_2 - 8x_1 = -6 \\ x_1x_2 + 2x_2^2 = 14 \end{cases}$	$x_1 = 0{,}7;$ $x_2 = 1$
9.	$\begin{cases} tg(x_1x_2 + 0{,}2) = x_1^2 \\ 0{,}6x_1^2 + 2x_2^2 = 1 \end{cases}$	$x_1 = 0{,}88;$ $x_2=0{,}52$
10.	$\begin{cases} 5\sin(x_1 + 2x_2) - 5{,}4x_1 = -5{,}2 \\ 5x_1^3 + x_2^3 - x_1 = 25 \end{cases}$	$x_1 = 2;$ $x_2 = -3$
11.	$\begin{cases} \sin x_1 + 2\sin x_2 = 1 \\ 2\sin 3x_1 + 3\sin 3x_2 = 0{,}3 \end{cases}$	$x_1 = 1{,}08;$ $x_2=0{,}06$
12.	$\begin{cases} 5\sin x_1 + 4x_2 = -7{,}2 \\ \cos x_2 + 8x_1 = -15 \end{cases}$	$x_1 = 0;$ $x_2=0$
13.	$\begin{cases} (x_1^3 + 3)x_2^2 = 3 \\ 3x_1x_2 + 5x_1^2 - 6 = 0 \end{cases}$	$x_1 = -1;$ $x_2 = -1$
14.	$\begin{cases} x_1^6 + 3x_2^3 - 2x_1 = -23 \\ 3x_1^3 - 5x_2^3 - 8x_2 = 26 \end{cases}$	$x_1 = 0;$ $x_2 = 0$

1	2	3
15.	$\begin{cases}\sin(x_1+x_2)-1{,}24x_1=0{,}1\\ {x_1}^2+{x_2}^2=1\end{cases}$	$x_1=0{,}74$; x_2=0,67
16.	$\begin{cases}5{x_1}^5+{x_2}^3=-14\\ 3{x_1}^2+1{,}5{x_2}^3=-25\end{cases}$	$x_1=-1$; x_2=2
17.	$\begin{cases}x_1^2\sin x_2+{x_2}^2\sin x_1=-1\\ 2x_1+e^{x_1+x_2}=5\end{cases}$	$x_1=-0{,}3$; $x_2=1{,}5$
18.	$\begin{cases}2x_1{x_2}^2-4{x_2}^2=7{,}5\\ {x_1}^2-3x_1x_2=-4{,}5\end{cases}$	$x_1=3$; $x_2=1{,}5$
19.	$\begin{cases}3\sin x_1-2x_2=15\\ 5\cos x_2-5x_1=25\end{cases}$	$x_1=0$; x_2=0
20.	$\begin{cases}\sin(x_1+x_2)-2{,}4x_1=-3{,}2\\ {x_1}^2+{x_2}^2-x_1x_2=10\end{cases}$	$x_1=0{,}92$; $x_2=-2{,}60$
21.	$\begin{cases}5{x_1}^4+5{x_2}^3-12x_1=-5\\ 5{x_1}^4-6{x_2}^4-5x_2=3\end{cases}$	$x_1=0$; $x_2=0$
22.	$\begin{cases}(5{x_1}^2+2){x_2}^3=6\\ 2x_1x_2\quad+7{x_1}^3-10=0\end{cases}$	$x_1=-2$; $x_2=0$
23.	$\begin{cases}5x_1{x_2}^2+4{x_2}^2=15\\ 2{x_1}^2-5x_1x_2=9{,}5\end{cases}$	$x_1=3$; $x_2=1{,}5$
24.	$\begin{cases}{x_1}^4+{x_2}^2=3\\ {x_1}^3+{x_2}^3=4\end{cases}$	$x_1=0{,}95$; x_2=1,4
25.	$\begin{cases}\sin x_1+4\sin x_2=1\\ 2\sin 3x_1+3\sin 2x_2=0{,}4\end{cases}$	$x_1=-0{,}34$; $x_2=0{,}38$
26.	$\begin{cases}\sin(x_1+1)-x_1=0{,}1\\ \cos x_1+2x_2=2\end{cases}$	$x_1=0{,}86$; $x_2=0{,}67$
27.	$\begin{cases}{x_1}^3+{x_2}^3-6x_1=-3\\ {x_1}^3-{x_2}^3-6x_2=4\end{cases}$	$x_1=0{,}52$; $x_2=-0{,}37$
28.	$\begin{cases}\sin x_1+2\cos x_2=0{,}8\\ x_1{x_1}^2+3x_1=4{,}5\end{cases}$	$x_1=0{,}01$; $x_2=-24$
29.	$\begin{cases}\cos(x_2-1)+x_1=0{,}5\\ x_2-\cos x_1=3\end{cases}$	$x_1=1{,}20$; $x_2=3{,}36$
30.	$\begin{cases}\sin x_1-x_2=1{,}3\\ \cos x_2-x_1=-0{,}82\end{cases}$	$x_1=0{,}8$; x_2=0,35

6.3. Контрольні запитання

1. Що таке нелінійне алгебричне рівняння? Що означає відшукати розв'язок нелінійного рівняння?

2. На які етапи поділяється процес відшукування розв'язків нелінійного рівняння?

3. У чому логіка відшукування розв'язку нелінійного рівняння методом ділення навпіл? Які переваги і недоліки цього метода?

4. У чому сутність метода хорд? Які його переваги і недоліки?

5. Яка ідея методу дотичних? Які переваги і недоліки цього метода?

6. Які особливості комбінованого методу? У чому полягають його переваги і недоліки у порівнянні з методом хорд? З методом Ньютона?

7. У чому полягає сутність метода ітерацій розв'язування нелінійного рівняння? Які в нього переваги і недоліки у порівнянні з іншими методами?

8. Які засоби розв'язування нелінійних рівнянь є у сучасних мовах програмування?

9. Як відшукати корінь нелінійного алгебричного рівняння у системі MatLab?

10. Як можна відшукати комплексні нулі полінома? Які функції MatLab з тих, що призначені для відшукання нулів функцій, не дозволяють відшукати комплексні корені?

11. Які засоби відшукання комплексних нулів поліномів є у сучасних мовах програмування?

12. Як відшукати усі нулі заданого полінома у системі MatLab?

13. Чим відрізняються функції ***solve*** і ***fsolve***?

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 7
ОСНОВИ ГРАФІЧНОЇ ВІЗУАЛІЗАЦІЇ РОЗРАХУНКІВ. ОСНОВИ ДВОВИМІРНОЇ ГРАФІКИ

Мета роботи: навчитися побудові графіків за допомогою вбудованих процедур у середовищі MatLab.

7.1. Теоретичні відомості

7.1.1. Побудова графіків в декартовій системі координат

Для відображення функцій однієї змінної використовуються графіки в декартовій системі координат. Виведення графіків в декартовій системі координат у системі MatLab здійснюються за допомогою команди ***plot***.

- ***plot(****X, Y)* – команда, яка будує графік функції $y(x)$, у якому одновимірний масив *X* відповідає значенням аргументу, а одновимірний масив Y – значенням функції. При цьому вважається, що значення аргументу відкладаються вдовж горизонтальної осі графіка, а значення функції – вздовж вертикальної осі.

Графіки у MatLab завжди виводяться в окреме (графічне) вікно, яке називається фігурою.

Приклад. Побудувати графік функція $y(x) = \cos(x)$ на відрізку $[-2\pi, 2\pi]$ з кроком $\pi/100$.

В командному вікні MatLab записується

```
>> x=-2*pi:pi/100:2*pi;
>> y=cos(x);
>> plot(x,y)
```

У результаті на екрані з'явиться додаткове вікно з графіком функції (рис. 7.1).

- ***plot(***Y) – команда, яка будує графік функції $y(k)$, де значення y з вектора *Y*, а k – індекс відповідного елементу. Якщо y містить комплексні елементи, то виконується команда ***plot(***real***(***Y), imag(Y)).

Якщо в командному вікні записати

```
>> plot(y)
```

У графічному вікні відобразиться графік (рис. 7.2)

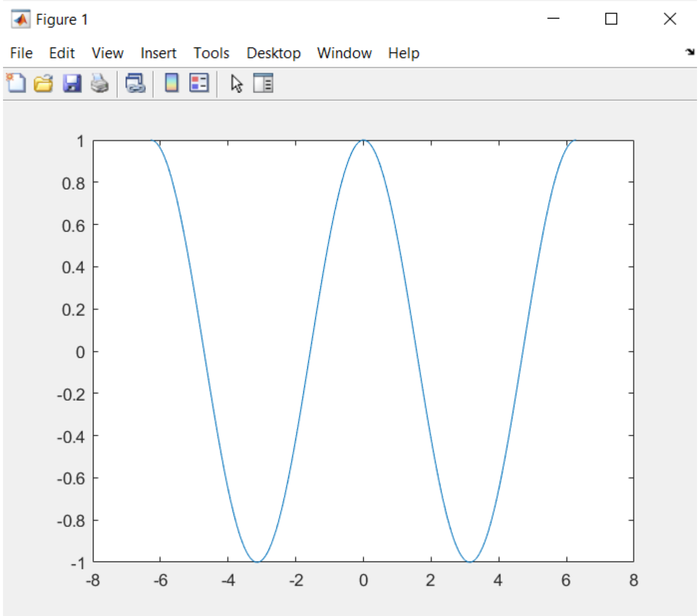


Рис.7.1 Графік функції $y(x) = \cos(x)$

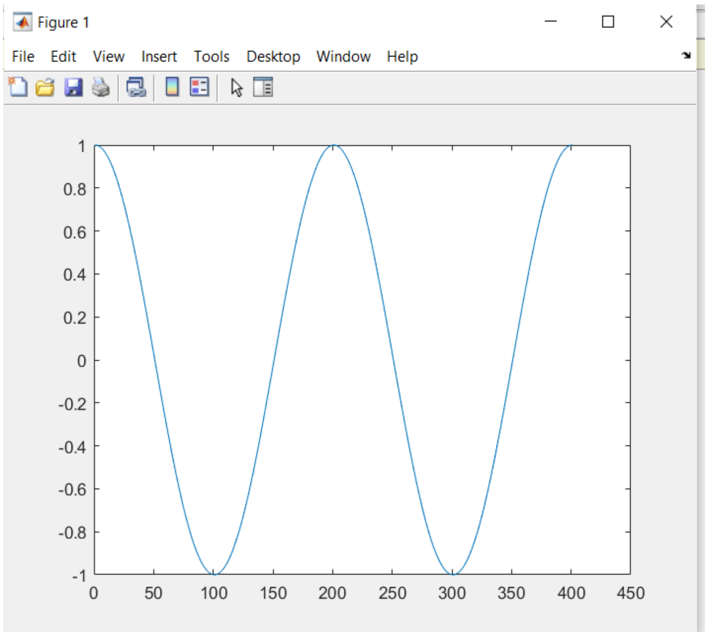


Рис.7.2 Приклад побудови графіка функції $y(x) = \cos(x)$, використовуючи функцію ***plot(***Y)

- ***plot(***X, Y, S) – аналогічно функції ***plot(***X, Y), але в даному випадку задається символьна змінна *S*, яка може містити до трьох спеціальних символів, які визначають відповідно: тип лінії, що з'єднує окремі точки графіка; тип точки графіка; колір лінії.

Якщо змінна *S* не вказана, то тип лінії за замовчуванням – відрізок прямої, тип точки – піксель, а колір встановлюється за чергуванням, прописаним у властивості вісей ColorOrder. Один з можливих варіантів послідовності кольорів наступний: жовтий, зелений, фіолетовий, блакитний, червоний, зелений, синій та чорний – залежно від того, яка по черзі лінія виводиться на графік.

У таблиці 7.1 представлені дані про види символів.

Таблиця 7.1.

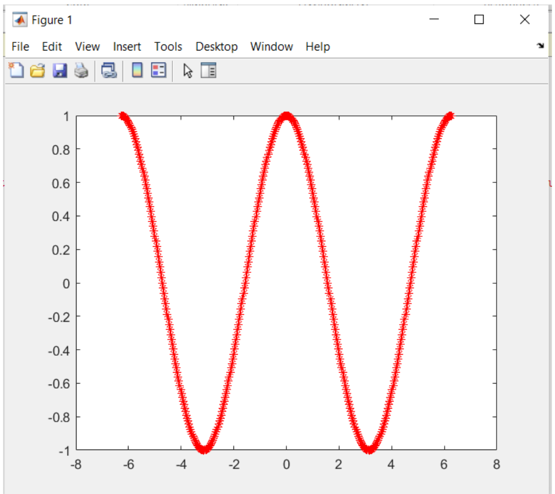
Характеристики лінії *plot(*X, Y, S)

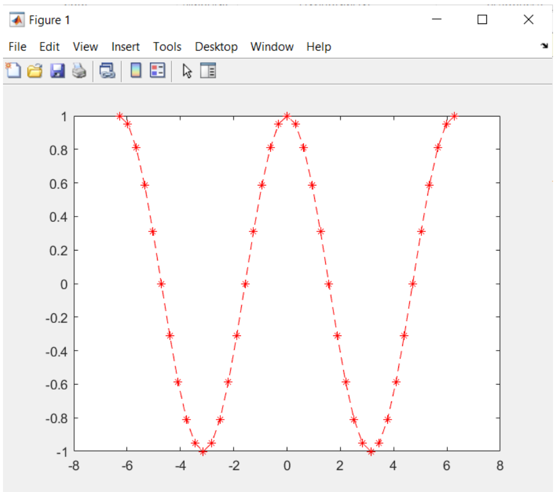
Колір	Тип лінії	Тип маркера
y – жовтий	- – суцільна	. – крапка
m – рожевий	: – пунктирна	o – коло
c – блакитний	-. – штрих-пунктирна	x – хрестик
r – червоний	-- – штрихова	+ – знак плюс
g – зелений	(none) – без лінії	* – зірочка
b – синій		s – квадрат
w – білий		d – ромб
k – чорний		v – трикутник вершиною вниз
		^ – трикутник вершиною вгору
		< – трикутник вершиною вліво
		> – трикутник вершиною вправо
		p – п’ятикутна зірка
		h – шестикутна зірка

Наприклад, якщо в командному вікні вказати

```
>> plot(x,y,'r--*')
```

Тоді в графічному вікні побудується графік функції $y(x) = \cos(x)$ штриховою лінією, червоного кольору із зірочками (рис. 7.3).





а) б)

Рис.7.3 Приклади побудови графіка функції $y(x) = \cos(x)$ з використанням функції ***plot***(x,y,'r--*'):

а) з кроком $\pi/100$; б) з кроком $\pi/10$

- ***plot(***X1, Y1, S1, X2, Y2, S2, …) – команда, яка дозволяє побудувати декілька графіків в одному графічному вікні.

Приклад. Побудувати графіки функцій на відрізку $[-2\pi, 2\pi]$ з кроком $\pi/100$:

- $y1(x) = \cos(x + \pi/2)$ пунктирною лінією рожевого кольору з колами;
- $y2(x) = \sin(x + \pi/3)$ штриховою лінією зеленого кольору з ромбами;
- $y3(x) = \sin(x + \pi/4)$ штрих-пунктирною лінією синього кольору зі знакам плюс.

В командному вікні прописується

```
>> x=-2*pi:pi/10:2*pi;
>> y1=cos(x+pi/2);
>> y2=sin(x+pi/3);
>> y3=sin(x+pi/4);
>> plot(x,y1,'m:o',x,y2,'g--d',x,y3,'b-.+')
```

У графічному вікні відображаються графіки функцій $y1(x) = \cos(x + \pi/2)$, $y2(x) = \sin(x + \pi/3)$, $y3(x) = \sin(x + \pi/4)$ (рис. 7.4).

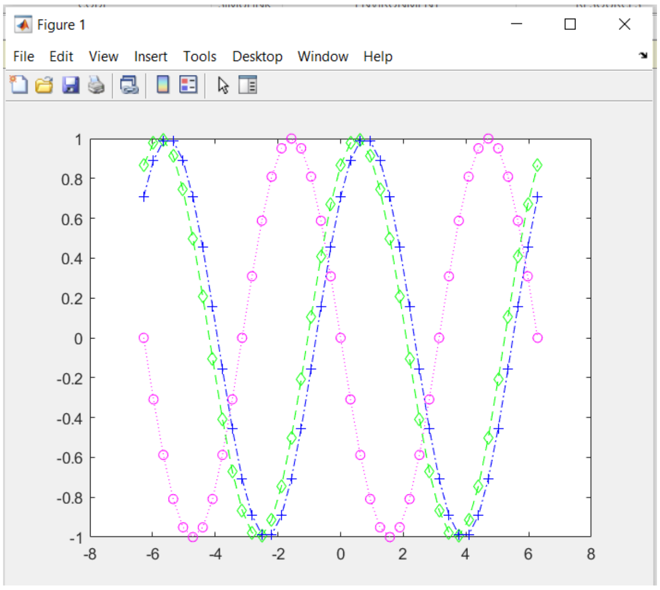


Рис.7.4 Приклад побудови графіків функцій, використовуючи команду ***plot(***X1, Y1, S1, X2, Y2, S2, X3, Y3, S3)

- ***plot(***X, Y, S, ‘LineWidth’, n) – аналогічно функції ***plot(***X, Y, S), але в даному випадку задається значення *n* властивості *‘LineWidth’*, що визначає товщину лінії.

Приклад. Побудувати графіки функцій $y(x) = \cos(x + \pi/2)$ на відрізку $[-2\pi, 2\pi]$ з кроком $\pi/100$ суцільною лінією червоного кольору, товщина лінії 7.

```
>> x=-2*pi:pi/100:2*pi;
>> y=cos(x+pi/2);
>> plot(x,y,'r-','LineWidth',7),grid,
```

Результат виконання коду показано в графічному вікні (рис. 7.5).

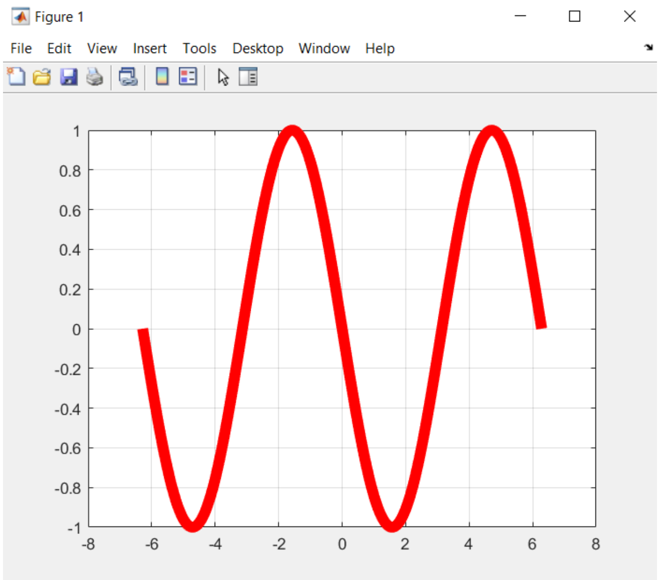


Рис.7.5 Приклад побудови графіків функцій, використовуючи команду ***plot(***X, Y, S, ‘LineWidth’, n)

7.1.2. Основи форматування двовимірних графіків

Графіки представлені на рис. 7.1-7.4 мають кілька недоліків:

- на них не нанесено сітку з координатних ліній, що ускладнює «читання» графіків;
- немає загальної інформації про криві графіка (заголовки);
- невідомо, які величини відкладено по осях графіка.

Перший недолік усувається за допомогою функції ***grid***. Якщо ***grid*** записати одразу після звернення до функції ***plot*** :

```
>> x=-2*pi:pi/10:2*pi;
y=cos(x+pi/2);
plot(x,y,'m:'), grid,
```

то графік буде споряджений координатною сіткою (рис.7.6, а).

Цінною особливістю графіків, побудованих у системі MatLab, є те, що сітка координат завжди відповідає «цілим» крокам змінювання величини, що робить графіки «читабельними». Тобто за графіком можна відлічувати значення функції за будь-яким заданим значенням аргументу і навпаки. Такої властивості не має жоден з графічних пакетів-додатків до мов програмування високого рівня.

Заголовок графіка виводиться за допомогою процедури ***title.*** Якщо після звернення до процедури ***plot*** звернутися до ***title*** таким чином:

title(‘<текст>’),

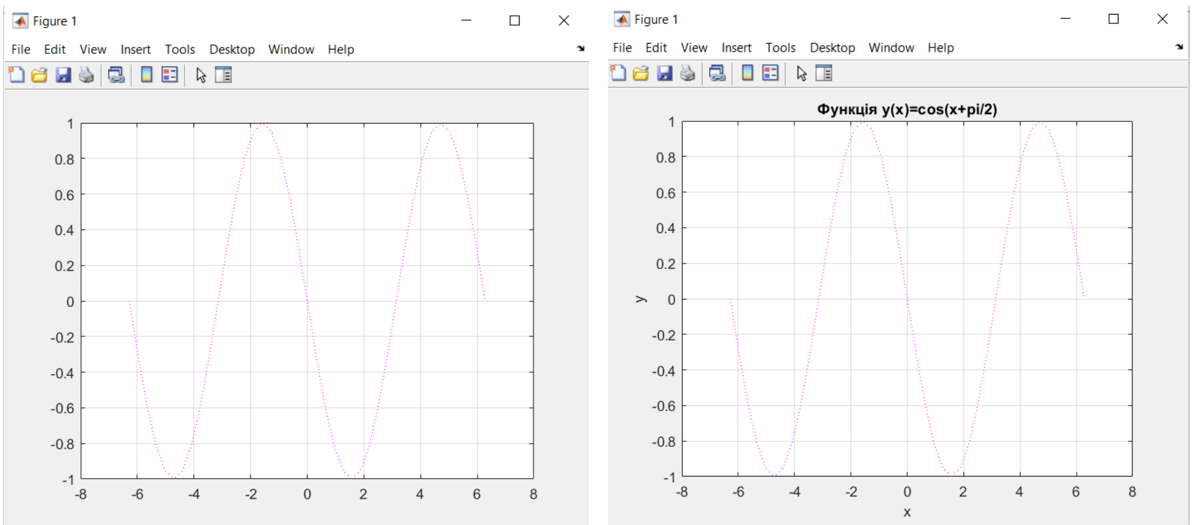
то зверху у полі фігури з’явиться текст, який записано між дужками. При цьому слід пам’ятати, що текст завжди повинен записуватися між апострофами.

Аналогічно можна створити пояснення до графіка, що виводиться вдовж горизонтальної осі (функція ***xlabel***) та вдовж вертикальної осі (функція ***ylabel***).

Наприклад:

```
>> x=-2*pi:pi/10:2*pi;
>> y=cos(x+pi/2);
>> plot(x,y,'m:'), grid,
>> title('Функція y(x)=cos(x+pi/2)')
>> xlabel('x')
>> ylabel('y')
```

Результат показано в графічному вікні (рис. 7.6, б).



а) б)

Рис.7.6 Приклади побудови графіка функції

$y(x) = \sin(x + \pi/3)$: а) з використанням функції ***grid***;

б) з використанням функцій ***title***, ***xlabel*** та ***ylabel***

Щоб додати пояснення до графіка у вигляді текстових рядків у MatLab використовується команда

legend(‘<текст1>’, ‘<текст2>’, ‘<текст3>’,…).

Наприклад,

```
>> x=-2*pi:pi/10:2*pi;
y1=cos(x+pi/2);
y2=sin(x+pi/3);
y3=sin(x+pi/4);
plot(x,y1,'m:o',x,y2,'g--d',x,y3,'b-.+'),grid,
xlabel('x')
ylabel('y')
legend('y1=cos(x+pi/2)','y2=sin(x+pi/3)','y3=sin(x+pi/4)')
```

Результат показано в графічному вікні (рис. 7.7).

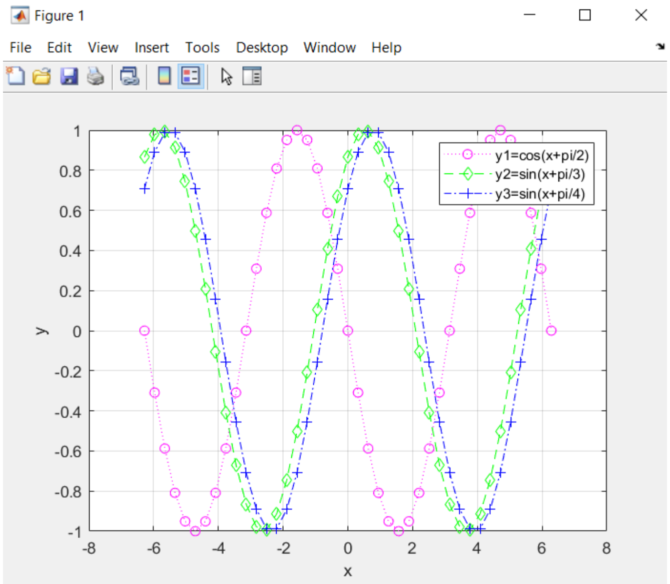


Рис.7.7 Приклад побудови графіка, використовуючи команду ***legend***

Для розташування легенди в точно визначене місце використовується наступний формат команди

legend(…, 'Location',lcn),

де ***lcn*** може приймати наступні ***значення 'north', 'south', 'east', 'west', 'northeast', 'northwest', 'southeast', 'southwest', 'northoutside', 'southoutside', 'eastoutside', 'westoutside', 'northeastoutside', 'northwestoutside', 'southeastoutside', 'southwestoutside', 'best', 'bestoutside', 'none'***. Основні з цих значень розміщують легенду наступним чином:

'best' – найкраще місце, яке вибирається автоматично;

'north' – зверху посередині фігури;

'south' – знизу посередині фігури;

'west' – зліва посередині фігури;

'east' – справа посередині фігури;

'northeast' – верхній правий кут (застосовується за замовчуванням);

'southwest' – нижній лівий кут;

'northoutside' – зверху посередині, проте легенда поміщається поза осі фігури.

Встановлення діапазонів координат по вісях x та y для поточного графіка здійснюється за допомогою команди ***axis:***

axis([Xmin, Xmax, Ymin, Ymax]).

Функція ***figure*** створює об'єкт графічного вікна. Наприклад, якщо необхідно відобразити декілька графіків в різних графічних вікнах, тоді необхідно в тексті програми вказати ***figure(1), figure(2),…***

Приклад. Побудувати графіки функцій $y1(x) = \cos(x + \pi/2)$, $y2(x) = \sin(x + \pi/3)$ в різних графічних вікнах.

```
>> x=-2*pi:pi/10:2*pi;
>> y1=cos(x+pi/2);
>> y2=sin(x+pi/3);
>> figure(1)
>> plot(x,y1,'m:o'),grid,
>> figure(2)
>> plot(x,y2,'g--d'),grid,
```

Результат застосовування функції ***figure*** показано на рис. 7.8.

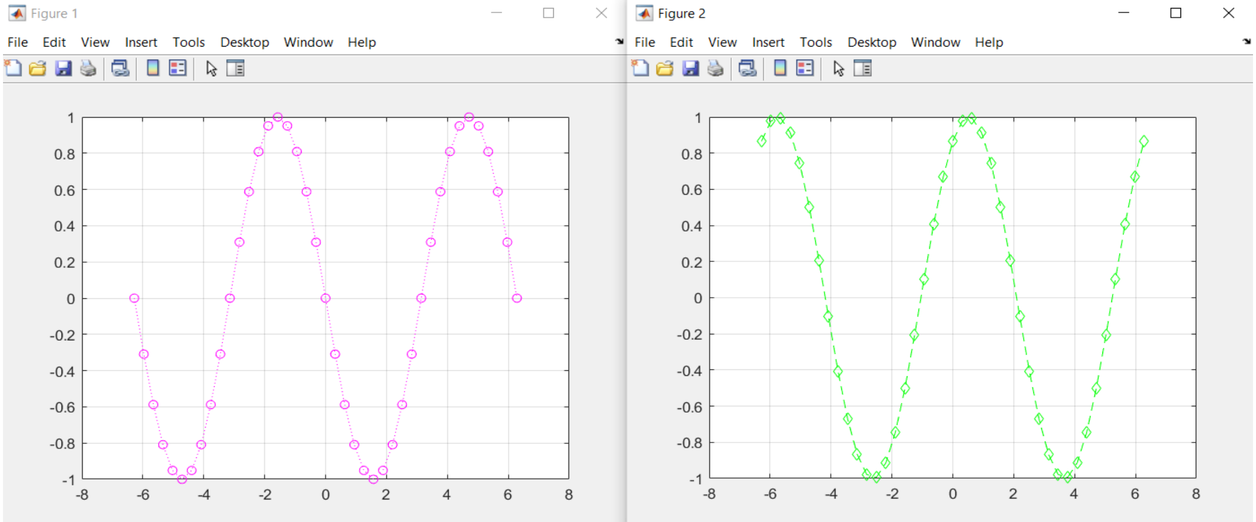


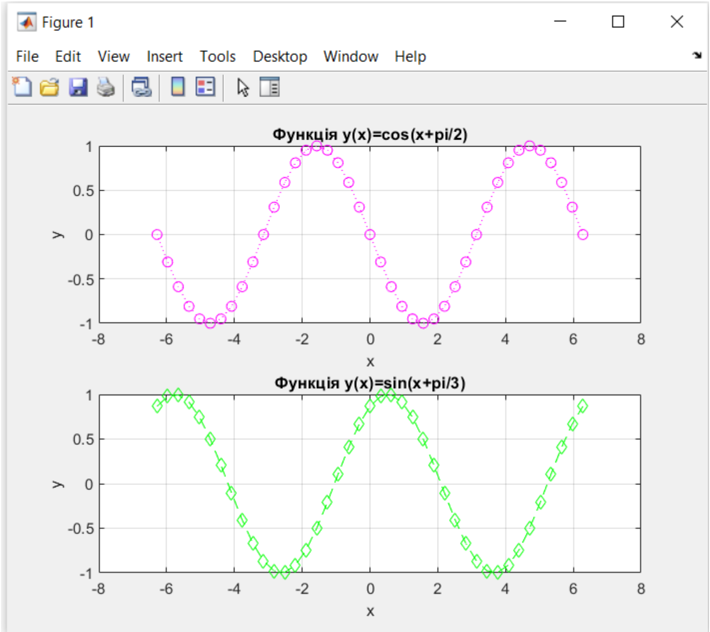
Рис. 7.8. Приклад побудови графіків в різних графічних вікнах

Щоб розбити графічне вікно для розміщення в ньому декількох графіків без накладання їх друг на друга, використовується команда ***subplot(m,n,p)***, де ***m*** – кількість підвікон по горизонталі; ***n*** – кількість підвікон по вертикалі; ***p*** – номер підвікна, в якому буде виводитись поточний графік.

Приклад. Побудувати графіки, використовуючи команду ***subplot*** (рис. 7.9-7.11).

```
>> x=-2*pi:pi/10:2*pi;
y1=cos(x+pi/2);
y2=sin(x+pi/3);
subplot(2,1,1)
plot(x,y1,'m:o'),grid,
title('Функція y(x)=cos(x+pi/2)')
xlabel('x')
ylabel('y')
subplot(2,1,2)
plot(x,y2,'g--d'),grid,
title('Функція y(x)=sin(x+pi/3)')
xlabel('x')
ylabel('y')
```

а)

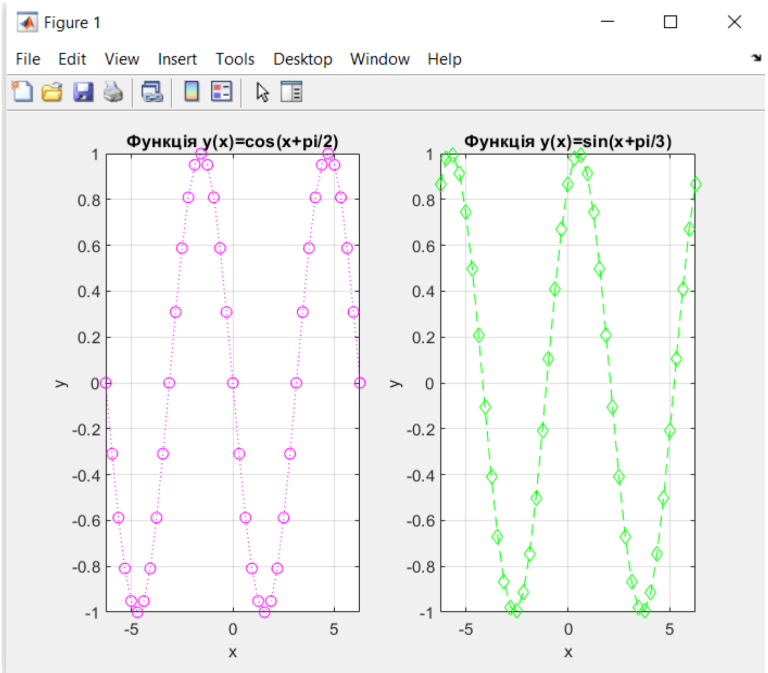


б)

Рис.7.9 Приклад побудови графіка функцій, використовуючи команду ***subplot***

```
>> x=-2*pi:pi/10:2*pi;
y1=cos(x+pi/2);
y2=sin(x+pi/3);
subplot(1,2,1)
plot(x,y1,'m:o'),grid,
title('Функція y(x)=cos(x+pi/2)')
xlabel('x')
ylabel('y')
subplot(1,2,2)
plot(x,y2,'g--d'),grid,
title('Функція y(x)=sin(x+pi/3)')
xlabel('x')
ylabel('y')
```

а)

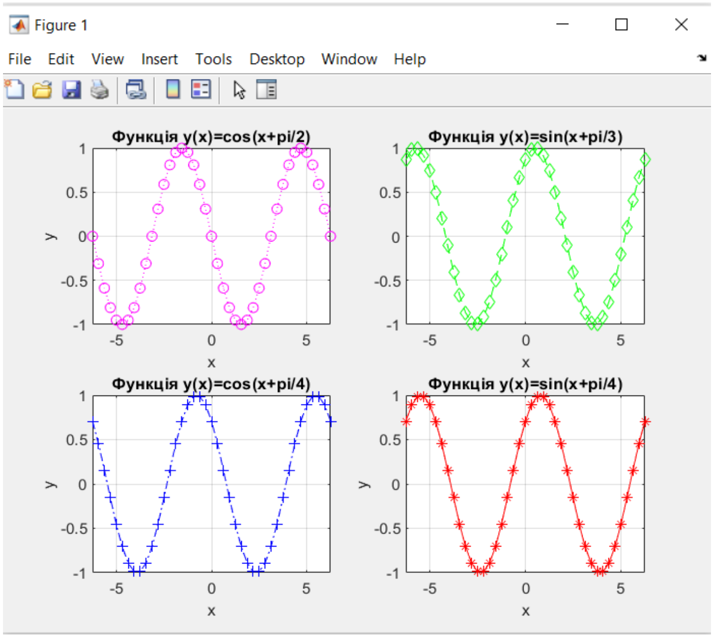


б)

Рис.7.10 Приклад побудови графіка функцій, використовуючи команду ***subplot***

```
>> x=-2*pi:pi/10:2*pi;
y1=cos(x+pi/2);
y2=sin(x+pi/3);
y3=cos(x+pi/4);
y4=sin(x+pi/4);
subplot(2,2,1)
plot(x,y1,'m:o'),grid,
title('Функція y(x)=cos(x+pi/2)')
xlabel('x')
ylabel('y')
subplot(2,2,2)
plot(x,y2,'g--d'),grid,
title('Функція y(x)=sin(x+pi/3)')
xlabel('x')
subplot(2,2,3)
plot(x,y3,'b-.+'),grid,
title('Функція y(x)=cos(x+pi/4)')
xlabel('x')
ylabel('y')
subplot(2,2,4)
plot(x,y4,'r-*'),grid,
title('Функція y(x)=sin(x+pi/4)')
xlabel('x')
ylabel('y')
```

а)



б)

Рис.7.11 Приклад побудови графіка функцій, використовуючи команду ***subplot***

7.1.3. Деякі можливості двовимірної графіки

Побудова графіків в полярній системі координат здійснюється за допомогою функції ***polar***:

polar(phi,r),

де ***phi*** – масив значень полярного кута; ***r*** – масив значень радіус-вектора.

Приклад. Побудувати графік $r = \cos(5 \cdot phi)$ в полярних координатах, де phi змінюється в діапазоні від 0 до 2π з кроком 0,01.

```
>> phi=0:0.01:2*pi;
>> r=cos(5*phi);
>> polar(phi,r), grid,
```

Результат застосування функції ***polar*** представлений на рис. 7.12.

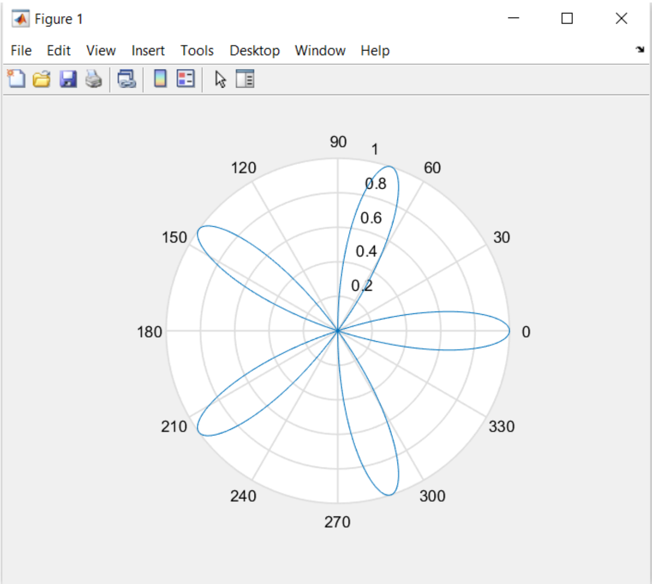


Рис.7.12 Приклад побудови графіка функцій, використовуючи функцію ***polar***

Для побудови графіків у логарифмічному масштабі використовується функція ***loglog,*** а для побудови графіків в напівлогарифмічному масштабі – ***semilogx, semilogy***.

- ***semilogx(…)*** – будує графік функції в логарифмічному масштабі по вісі ***X*** та лінійному по вісі ***Y***;
- ***semilogy(…)*** – будує графік функції в логарифмічному масштабі по вісі ***Y*** та лінійному по вісі ***X***;
- ***loglog(…)***– будує графік функції в логарифмічному масштабі по вісям ***X*** та ***Y***.

Для формування вузлів логарифмічної сітки в MatLab застосовується функція ***logspace:***

- ***logspace***(d1,d2) – функція, яка формує вектор-рядок, який складається з 50 рівновіддалених в логарифмічному масштабі точок, які покривають діапазон від 10^{d1} до 10^{d2};
- ***logspace***(d1,d2,n) – функція, яка формує вектор-рядок, який складається з n рівновіддалених в логарифмічному масштабі точок, які покривають діапазон від 10^{d1} до 10^{d2}.

Приклад. Побудувати графік функції $y(x) = \exp(x)$ в напівлогарифмічному масштабі по вісі *y*.

```
>> x=0:0.01:10;
>> semilogy(x, exp(x)), grid,
```

Результат показано на рис. 7.13.

Приклад. Побудувати графік функції $y(x) = \exp(x)$ в логарифмічному масштабі.

```
>> x=logspace(-1,2,45);
>> loglog(x,exp(x)),grid,
```

Результат представлений на рис. 7.14.

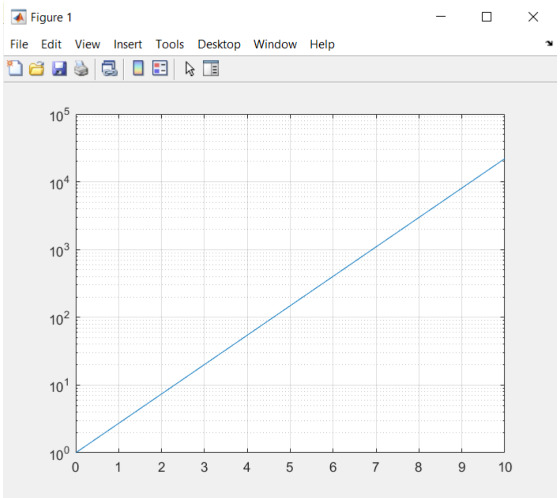


Рис.7.13 Приклад побудови графіка в напівлогарифмічному масштабі

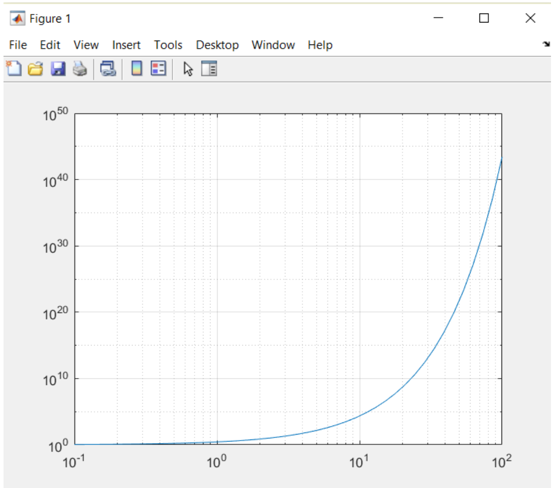


Рис.7.14 Приклад побудови графіка в логарифмічному масштабі

7.2. Завдання до виконання комп'ютерного практикуму

Завдання 1. Побудувати в графічному вікні MatLab графіки функції на відрізку $[a, b]$ з кроком h_1 та h_2 відповідно до варіанту згідно табл.7.2. Порівняти результати. Виконати оформлення графіків відповідно до варіанту згідно табл.7.3. Зробити текстове оформлення графіків.

Таблиця 7.2

Варіанти індивідуальних завдань

Варіант	$f(x)$	**a**	**b**	$\boldsymbol{h_1}$	$\boldsymbol{h_2}$
1	2	3	4	5	6
1	$\sqrt[3]{x^2+3} \cdot cos\left(\frac{\pi}{4}x\right)$	0,8	2,4	0,3	0,03
2	$\frac{e^{x/2}}{1+x^3}$	0,4	3,4	0,2	0,02
3	$\sqrt[4]{x^2+7} \cdot \sin\left(\frac{\pi}{3}x\right)$	0,5	3	0,1	0,01
4	$\left(5+2\sqrt{x}\right) \cdot \cos(\pi\sqrt{1+x})$	0,1	2,5	0,4	0,04

5	$\dfrac{x^3 - 0{,}5x}{\sqrt{1+3x}}$	1,05	3,05	0,1	0,01
1	2	3	4	5	6
6	$x^{3x+3} + x^2 - 3x$	1	5	0,4	0,01
7	$\dfrac{x^4}{1 + 0{,}5\sqrt[3]{x}}$	0,1	2,1	0,3	0,03
8	$\sqrt{4+2x} \cdot tg\left(\dfrac{\pi x}{3}\right)$	0,1	1,9	0,3	0,02
9	$(3+6x) sin\left(\pi^2\sqrt{1+x}\right)$	0	5	0,5	0,05
10	$\sqrt{2+3x^2} \cdot sin\left(\dfrac{2\pi x}{3}\right)$	1,2	3,6	0,3	0,03
11	$\dfrac{x^2}{x - 0{,}25\sqrt{x}}$	0,1	2,2	0,3	0,03
12	$\dfrac{x^3 + 2x}{1 + e^{2x}}$	0,2	3,2	0,6	0,01
13	$x^{x+2} + x^3 - 5x$	1	5	0,4	0,02
14	$\sqrt{5x+2} \cdot sin\left(\dfrac{\pi}{3}x\right)$	0,2	3,2	0,3	0,02
15	$e^{x^2}(1 + 3x - x^2)$	0	2,4	0,2	0,02
16	$3x^3 + \dfrac{1}{x} + e^{2x^2}$	0,2	2,2	0,1	0,01
17	$\sqrt{1+4x} \cdot sin(\pi x)$	0,1	3,6	0,7	0,01
18	$\dfrac{sin^2(\pi x)}{\sqrt{1+2x}}$	0,1	8,1	0,8	0,02
19	$\dfrac{x^4 - 0{,}8x}{\sqrt{1+2x}}$	0,5	6,5	0,6	0,05
20	$x^3 - 5x + \dfrac{6}{\sqrt{1+x^2}}$	0	5	0,5	0,01
21	$\sqrt{1+5x^2} \cdot cos\left(\dfrac{2\pi}{3}x\right)$	1,2	3,6	0,3	0,03
22	$\dfrac{e^{2x/3}}{3+4x^2}$	0,4	3,4	0,2	0,02
23	$\sqrt{x^2+2} \cdot sin\left(\dfrac{\pi}{3}x\right)$	0,8	2,4	0,3	0,03
24	$\sqrt{3+5x} \cdot \sin\left(\dfrac{\pi}{2}x\right)$	0,1	5	0,7	0,02
25	$e^{-2x} + x^2 - 2$	0,2	5,8	0,8	0,02

1	2	3	4	5	6
26	$\frac{cos^2(\pi x)}{\sqrt{1+4x}}$	0	5,6	0,7	0,01
27	$arcsin\left(e^{-x^2/3}\right)$	4	9,6	0,7	0,02
28	$\frac{1+e^{-x/2}}{\sqrt{3x^2+1}}$	0,5	5,3	0,6	0,01
29	$\frac{x^3-0{,}3x}{\sqrt{1+2x}}$	1,2	4	0,6	0,03
30	$\sqrt{3x^2+5}\cdot cos\left(\frac{\pi}{3}x\right)$	0,5	2,5	0,5	0,01

Таблиця 7.3

Варіанти індивідуальних завдань

Варіант	**Тип лінії**	**Колір**	**Тип маркера**	**Товщина лінії**
1	2	3	4	5
1	суцільна	жовтий	коло	5
2	пунктирна	рожевий	хрестик	6
3	штрих-пунктирна	блакитний	знак плюс	7
4	штрихова	червоний	зірочка	8
5	суцільна	зелений	квадрат	4
6	пунктирна	синій	ромб	5
7	штрих-пунктирна	чорний	трикутник вершиною вниз	6
8	штрихова	жовтий	трикутник вершиною вгору	7
9	суцільна	рожевий	трикутник вершиною вліво	8
10	пунктирна	блакитний	трикутник вершиною вправо	4
11	штрих-пунктирна	червоний	п’ятикутна зірка	5
12	штрихова	зелений	шестикутна зірка	6
13	суцільна	синій	коло	7

1	2	3	4	5
14	пунктирна	чорний	хрестик	8
15	штрихова	рожевий	зірочка	5
16	суцільна	блакитний	квадрат	6
17	пунктирна	червоний	ромб	7
18	суцільна	червоний	коло	4
19	штрих-пунктирна	жовтий	знак плюс	4
20	штрих-пунктирна	зелений	трикутник вершиною вниз	8
21	штрихова	синій	трикутник вершиною вгору	4
22	суцільна	чорний	трикутник вершиною вліво	5
23	пунктирна	жовтий	трикутник вершиною вправо	6
24	штрих-пунктирна	рожевий	п’ятикутна зірка	7
25	штрихова	блакитний	шестикутна зірка	8
26	пунктирна	зелений	хрестик	5
27	штрих-пунктирна	синій	знак плюс	6
28	штрихова	чорний	зірочка	7
29	суцільна	жовтий	квадрат	8
30	пунктирна	рожевий	ромб	5

Завдання 2. Побудувати графіки функцій $y_1(x)$, $y_2(x)$ та $y_3(x)$ на відрізку $[a, b]$ з кроком h відповідно до варіанту згідно табл.7.4. Розмістити графіки: накладаючи друг на друга в одному графічному вікні MatLab; у різних графічних вікнах; в одному графічному вікні три графіка без накладання їх друг на друга (3 підвікна по горизонталі та 3 підвікна по вертикалі). Виконати оформлення графіків. Зробити текстове оформлення графіка.

Таблиця 7.4

Варіанти індивідуальних завдань

Варіант	$f(x)$	**a**	**b**	***h***
1	2	3	4	5
1	$\frac{x^3-0{,}8x}{\sqrt{1+2{,}5x}}$ $x^{x+1}+2x^3-6x$ $\sqrt{1+5x}\cdot sin(0{,}2\pi x)$	0,2	3,2	0,02
2	$\sqrt{5+2{,}5x}\cdot tg\left(\frac{\pi x}{4}\right)$ $\frac{x^2}{x-0{,}5\sqrt{x}}$ $\sqrt{4+3{,}5x}\cdot \sin\left(\frac{\pi}{3}x\right)$	0,1	3,6	0,01
3	$x^{2x+1}+x^2+4x$ $\sqrt{1+2{,}4x}\cdot sin(\pi x)$ $\frac{sin^2(0{,}3\pi x)}{\sqrt{1{,}2x-1}}$	1,5	5,6	0,01
4	$(3+2x)sin\left(\pi\quad\sqrt{1+x}\right)$ $\frac{x^3+2{,}5x}{1+3{,}2e^x}$ $\frac{1+e^{-x}/_2}{\sqrt{4{,}5x^2+1}}$	0,2	3,2	0,02
5	$\frac{e^{x}/_2}{2+x^3}$ $\sqrt{4x+2{,}2}\cdot sin\left(\frac{\pi}{3}x\right)$ $arcsin\left(e^{-2x^2}/_3\right)$	0,1	3,6	0,01

1	2	3	4	5
6	$\sqrt[4]{x^2+5}\cdot\sin\left(\frac{\pi}{4}x\right)$ $\frac{x^2}{x+0{,}27\sqrt{x}}$ $\sqrt{2{,}5x^3+3{,}8}\cdot cos\left(\frac{\pi}{3}x\right)$	1,5	5,6	0,01
7	$\sqrt[3]{4{,}5x^2+3}\cdot cos\left(\frac{\pi}{4}x\right)$ $3x^3+\frac{3{,}4}{x}+e^{x^2}$ $\frac{x^4-5{,}8x}{\sqrt{1+3x}}$	0,1	4,5	0,02
8	$\left(4+5\sqrt{x}\right)\cdot\cos(\pi\sqrt{1+x})$ $x^3+3x+\frac{4}{\sqrt{1+x^2}}$ $\frac{x^3-1{,}3x}{\sqrt{1+2x}}$	0,2	3,2	0,02
9	$\frac{2x^3-1{,}5x}{\sqrt{1+2{,}3x}}$ $e^{2x^2}(5+3x-x^2)$ $\sqrt{1+1{,}2x}\cdot sin(\pi x)$	0,1	3,6	0,01
10	$\frac{e^{x/2}}{2+1{,}5x^3}$ $x^{x+3}+1{,}5x^3-3x$ $\sqrt{1+3x^2}\cdot cos\left(\frac{\pi}{3}x\right)$	1,5	5,6	0,01

1	2	3	4	5
11	$\sqrt{2+3x}\cdot tg\left(\frac{\pi x}{4}\right)$ $\frac{x^4-2x}{\sqrt{1+3x}}$ $\sqrt{1+2x}\cdot\sin\left(\frac{\pi}{3}x\right)$	0,1	4,5	0,02
12	$\sqrt[3]{x^2+3}\cdot\sin\left(\frac{\pi}{4}x\right)$ $\frac{e^{2x/3}}{2+5x^2}$ $arcsin\left(e^{-x^2/5}\right)$	0,2	3,2	0,02
13	$\sqrt{1+4x^2}\cdot sin\left(\frac{\pi x}{4}\right)$ $\frac{x^2}{x-2\sqrt{x}}$ $\frac{1+e^{-x/2}}{\sqrt{4x^2+1}}$	0,1	3,6	0,01
14	$\sqrt[3]{x^2+2}\cdot cos\left(\frac{\pi}{8}x\right)$ $\frac{x^3+3x}{1+e^{3x}}$ $e^{-2x}+2x^2-1,2$	1,5	5,6	0,01

1	2	3	4	5
15	$\frac{e^{x/2}}{1+2x^3}$ $\sqrt{6x+1}\cdot sin\left(\frac{\pi}{3}x\right)$ $\frac{cos^2(\pi x)}{\sqrt{1+2x}}$	0,1	4,5	0,02
16	$\left(1+3\sqrt{x}\right)\cdot\cos(\pi\sqrt{2+x})$ $\frac{x^4-1{,}2x}{\sqrt{1+3x}}$ $\sqrt{3+6x}\cdot\sin\left(\frac{\pi}{4}x\right)$	0,2	3,2	0,02
17	$\frac{x^3-1{,}5x}{\sqrt{1+2{,}5x}}$ $e^{x^2}(1+4x-x^2)$ $\sqrt{x^2+3}\cdot sin\left(\frac{\pi}{4}x\right)$	0,1	3,6	0,01
18	$\sqrt[4]{x^2+7}\cdot\sin\left(\frac{\pi}{4}x\right)$ $3x^3+\frac{1}{2x}+e^{x^2}$ $\frac{x^3-2{,}3x}{\sqrt{1+3x}}$	1,5	5,6	0,01

1	2	3	4	5
19	$\frac{x^4}{1+0{,}5\sqrt{x}}$ $\sqrt{4x+1}\cdot sin\left(\frac{\pi}{6}x\right)$ $\frac{1+e^{-x/2}}{\sqrt{3x^2+1}}$	0,1	4,5	0,02
20	$x^{x+1}+2x^2-3x$ $\frac{sin^2(\pi x)}{\sqrt{1+3x}}$ $e^{-2x}+2x^3-2$	0,2	3,2	0,02
21	$(3+4x)sin\left(\pi^2\sqrt{1+x}\right)$ $\frac{x^3+3x}{1+e^{3x}}$ $\sqrt{x^2+3}\cdot sin\left(\frac{\pi}{5}x\right)$	0,1	3,6	0,01
22	$\sqrt{2+3x^2}\cdot sin\left(\frac{2\pi x}{3}\right)$ $\frac{e^{2x/3}}{4+5x^2}$ $\frac{cos^2(\pi x)}{\sqrt{1+3x}}$	1,5	5,6	0,01
23	$\frac{3x^4}{1+5\sqrt[3]{x}}$ $x^{x+2}+x^3-5x$ $\sqrt{2x^2+4}\cdot cos\left(\frac{\pi}{4}x\right)$	0,1	4,5	0,02

1	2	3	4	5
24	$\left(5+3\sqrt{x}\right)\cdot\cos(\pi\sqrt{1+x})$ $5x^3+\frac{3}{x}+e^{3x^2}$ $arcsin\left(e^{-x^2/3}\right)$	0,2	3,2	0,02
25	$(2+4x)sin\left(\pi^2\sqrt{1+x}\right)$ $x^3-5x+\frac{6}{\sqrt{1+x^2}}$ $\sqrt{3x^2+4}\cdot cos\left(\frac{\pi}{3}x\right)$	0,1	3,6	0,01
26	$\sqrt{2+3x^2}\cdot sin\left(\frac{2\pi x}{3}\right)$ $\frac{e^{2x/3}}{4+2{,}5x^2}$ $e^{-2x}+4x^2+3$	0,1	4,5	0,02
27	$\sqrt[3]{x^2+4}\cdot cos\left(\frac{\pi}{4}x\right)$ $\frac{x^3-0{,}3x}{\sqrt{1+2{,}3x}}$ $arcsin\left(e^{-x^2/4}\right)$	1,5	5,6	0,01
28	$x^{3x+3}+x^2-5x$ $\sqrt{1+5x^2}\cdot cos\left(\frac{\pi}{3}x\right)$ $\frac{cos^2(\pi x)}{\sqrt{1+2{,}4x}}$	0,2	3,2	0,02

1	2	3	4	5
29	$\sqrt{4+2x}\cdot tg\left(\frac{\pi x}{6}\right)$ $x^3-4x+\frac{6}{\sqrt{2+x^2}}$ $\sqrt{x^2+3}\cdot sin\left(\frac{\pi}{6}x\right)$	0,1	3,6	0,01
30	$\frac{x^4}{1+1{,}5\sqrt[3]{x}}$ $e^{x^2}(1+5x-3x^2)$ $\sqrt{1+5x^3}\cdot cos\left(\frac{\pi}{3}x\right)$	1,5	5,6	0,01

Завдання 3. Побудувати в графічному вікні MatLab графіки $r(phi)$ в полярній системі координат, де phi змінюється в діапазоні від 0 до 2π з кроком 0,01 відповідно до варіанту згідно табл.7.5. Зробити текстове оформлення графіків.

Таблиця 7.5

Варіанти індивідуальних завдань

Варіант	$\boldsymbol{r(phi)}$	**Варіант**	$\boldsymbol{r(phi)}$
1	2	3	4
1	$2cos(phi)$	7	$2sin(3phi)$
2	$cos(2phi+3)$	8	$2cos(5phi+1)$
3	$sin(5phi+3)$	9	$cos(6phi)$
4	$3cos(2phi)$	10	$5sin(0{,}5phi)$
5	$2sin(phi+5)$	11	$2cos(4phi+4)$
6	$2cos(4phi)$	12	$2cos(5phi+3)$

1	2	3	4
13	$2sin(2phi + 3)$	22	$3cos(5phi)$
14	$2cos(6phi + 1)$	23	$2sin(2phi + 1)$
15	$cos(5phi)$	24	$2cos(phi)$
16	$2sin(4phi + 4)$	25	$3cos(5phi + 0,5)$
17	$2sin(phi)$	26	$2sin(2phi + 4)$
18	$6cos(6phi + 10)$	27	$5cos(8phi)$
19	$2cos(4phi + 1)$	28	$3sin(7phi + 1)$
20	$0,5sin(4phi + 4)$	29	$2cos(3phi + 2)$
21	$2sin(3p\square i)$	30	$3sin(5phi + 7)$

7.3. Контрольні питання

1.Яка основна команда використовується для виведення графіків у системі MatLab? Що таке фігура? Як побудувати графік у полярній системі координат?

2. Як задати тип лінії при побудові графіку за допомогою ***plot***? Як задати вид окремих точок на графікові? Як визначити колір лінії?

3. Як задати чорну штрих-пунктирну лінію, у якої окремі точки виділяються у вигляді ромбів?

4. Як побудувати на одній фігурі відразу декілька графіків в одних осях?

5. Як задати товщину лінії графіка?

6. Як добавити сітку координат на графіки?

7. Як добавити довідкові дані для опису інформації на графіках? Як формується заголовок графіка? Як добавити підписи до вісей?

8. Як добавити легенду до побудованих графіків?

9. Покажіть як розмістити легенду у лівому верхньому куті? У правому верхньому куті? Як розмістити знизу посередині?

10. Як розмістити легенду до графіків поза осями фігури?

11. Що виконує команда ***axis([-20, 30, 0, 125]*** ?

12. Як створити декілька фігур для виведення на них різних графіків?

13. Як зробити, щоб одні фігура містила декілька графічних вісей (дає змогу розмістити в графічному вікні декілька графіків без накладання їх друг на друга) ?

14. Напишіть команду, що розіб'є графічне вікно на 3 підвікна по горизонталі та 2 вікна по вертикалі. Як встановити поточним для виводу графіків праве нижнє вікно? Праве верхнє?

15. Побудуйте графік функції $y=2x^3+5x^2+0.5+1/x$ у логарифмічному масштабі для аргументу від -100 до 1000.

16. Як побудувати попередній графік в логарифмічному масштабі по вісі ***X*** та лінійному по вісі ***Y***? У логарифмічному масштабі по вісі ***Y*** та лінійному по вісі ***X***?

17. Як сформувати вектор аргументів функції, що заданий рівномірно у логарифмічому масштабі? Скільки точок містить такий вектор?

18. Як сформувати вектор аргументів функції, що заданий рівномірно у логарифмічому масштабі, щоб вектор містив 150 точок?

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 8
ЧИСЕЛЬНІ МЕТОДИ НАБЛИЖЕННЯ. ІНТЕРПОЛЯЦІЯ ТА АПРОКСИМАЦІЯ

Мета роботи: навчитися використовувати функції апроксимації та інтерполяції, які вбудовані у MatLab.

8.1. Теоретичні відомості
8.1.1. Короткі відомості щодо інтерполяції та апроксимації функцій

Інтерполяцією називається метод знаходження проміжних значень за наявним дискретним набором відомих значень (наприклад, таблично-задана функція (табл. 8.1)).

Таблиця 8.1

Таблично-задана функція

x	x_0	x_1	x_2	x_3	…	x_n
$y(x)$	$y(x_0)$	$y(x_1)$	$y(x_2)$	$y(x_3)$	…	$y(x_n)$

Інтерполяцію застосують у випадках, коли необхідно відшукати значення $y(x)$ в точці x_k, яка знаходиться на проміжку $[x_0, \ldots, x_n]$, але не співпадає з жодним цих значенням.

Точки $x_0, x_1, x_2, \ldots, x_n$ називаються *вузлами інтерполяції*.

Інтерполяційна функція обов'язково повинна проходити через всі вузли інтерполяції та має вигляд поліному n-го порядку.

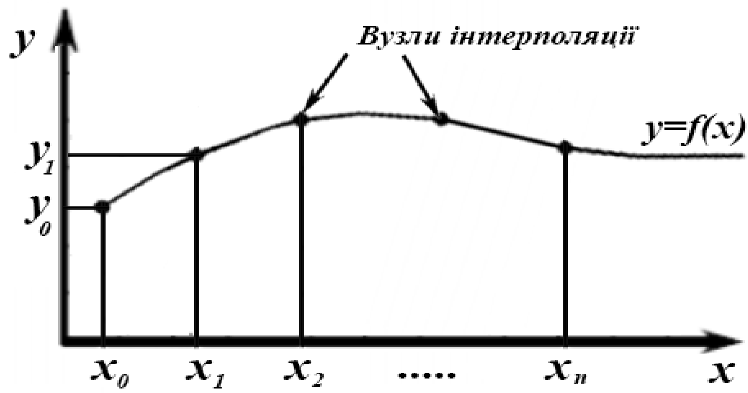


Рис.8.1. Графічне представлення інтерполяційної функції

До відомих методів інтерполяції відносяться: лінійна інтерполяція, інтерполяційна формула Ньютона, метод скінченних різниць, поліном Лагранжа, сплайн-функція, кубічний сплайн та ін.

Лінійна інтерполяція полягає у відшуканні проміжних значень функції за допомогою лінійного рівняння. Наприклад, якщо необхідно відшукати значення функції $y(x)$ в точці x_k, яка знаходиться на проміжку $[x_0, x_1]$. Тоді можна знайти наближення значень функції, замінивши функцію на даному проміжку лінійною функцією – така заміна називається лінійною інтерполяцією (рис. 8.2).

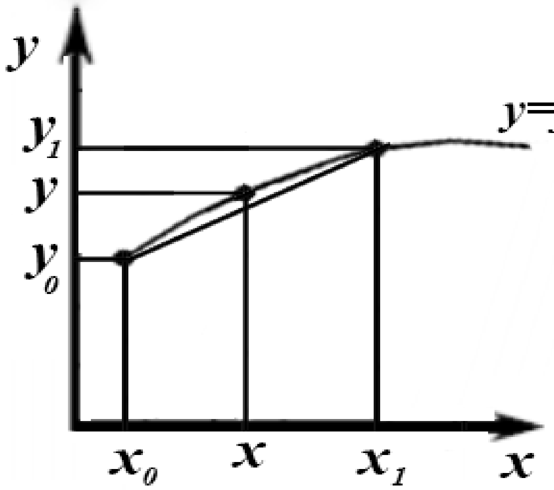


Рис. 8.2. Графічне представлення лінійної інтерполяції

Рівняння прямої, яка проходить через точки (x_0, y_0) та (x_1, y_1) має вигляд

$$\frac{y - y_0}{y_1 - y_0} = \frac{x - x_0}{x_1 - x_0}$$

У результаті отримується формула для знаходження наближеного значення функції:

$$y \approx y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0)$$

При застосуванні лінійної інтерполяції на всіх проміжках значень отримується наближена функція $y(x)$ у вигляді ламаної лінії.

Апроксимацією (наближенням) називають науковий метод, який полягає в заміні одних об’єктів іншими, більш простими. На відмінну від інтерполяції, апроксимуюча функція проходить не через вузли, а поміж них (рис.8.3).

Апроксимація буває:

- *точковою* або *дискретною* – якщо наближена функція будується на основі дискретного набору точок;
- *неперервною* або *інтегральною* – якщо апроксимація проводиться на неперервній множені точок (відрізку).

8.1.2. Реалізація апроксимації та інтерполяції засобами MatLab

Система MatLab надає зручні процедури для апроксимування та інтерполювання даних вимірювань.

Поліноміальне апроксимування даних вимірювань, які сформовані як деякий вектор ***y***, за деяких значень аргументу, котрі утворюють вектор ***x*** тієї ж довжини, що й вектор ***y***, здійснюється процедурою ***polyfit***(x,y,n), де ***n*** – порядок апроксимуючого полінома. Результатом дії цієї процедури є вектор довжиною (n+1) коефіцієнтів апроксимуючого полінома.

Нехай масив значень аргументу є таким:

$$x = [-2, -1,\ 0, 1,\ 2,\ 3,\ 4\ ],$$

а масив відповідних значень вимірювальної величини – наступним:

$$y = [-0.6,\ 1.1,\ 2.7, 5.3, -1.8, -\ 2.4, -2.9],$$

Тоді, застосовуючи функцію ***polyfit*** при різних значень порядку апроксимуючого полінома,

```
x=[-2 -1 0 1 2 3 4];
y=[-0.6 1.1 2.7 5.3 -1.8 -2.4 -2.9];
x1=-2:0.1:4;
p1=polyfit(x,y,1)
p2=polyfit(x,y,2)
p3=polyfit(x,y,3)
p4=polyfit(x,y,4)
```

отримуємо вектори, що містять коефіцієнти апроксимуючих поліномів:

```
p1 =
   -0.6571    0.8571


p2 =
   -0.4929    0.3286    2.3357


p3 =
    0.1583   -0.9679   -0.3048    3.2857


p4 =
    0.1186   -0.3159   -1.3913    1.4906    3.4890
```

Це означає, що дану залежність можна апроксимувати або прямою

$$y(x) = -0{,}6571x + 0{,}8571,$$

або квадратною параболою

$$y(x) = -0{,}4929x^2 + 0{,}3286x + 2{,}3357,$$

або кубічною параболою

$$y(x) = 0{,}1583x^3 - 0{,}9679x^2 - 0{,}3048x + 3{,}2857,$$

або параболою четвертого ступеня

$$y(x) = 0{,}1186x^4 - 0{,}3159x^3 - 1{,}3913x^2 + 1{,}4906x + 3{,}4890.$$

Для побудови графіків в М-файлі записується

```
y1=-0.6571*x+0.8571;
y2=-0.4929*x.^2+0.3286*x+2.3357;
y3=0.1583*x.^3-0.9679*x.^2-0.3048*x+3.2857;
y4=0.1186*x.^4-0.3159*x.^3-1.3913*x.^2+1.4906*x+3.4890;
figure(1)
plot(x,y, '*', x,y1, x,y2, x,y3, x,y4), grid
xlabel('x');
ylabel('y');
legend('табличні значення','1-порядок апроксимуючого
поліному', ...
    '2-порядок апроксимуючого поліному',...
    '3-порядок апроксимуючого поліному',...
    '4-порядок апроксимуючого поліному')
```

На рис.8.3 представлено графічне вікно з прикладом застосування апроксимації.

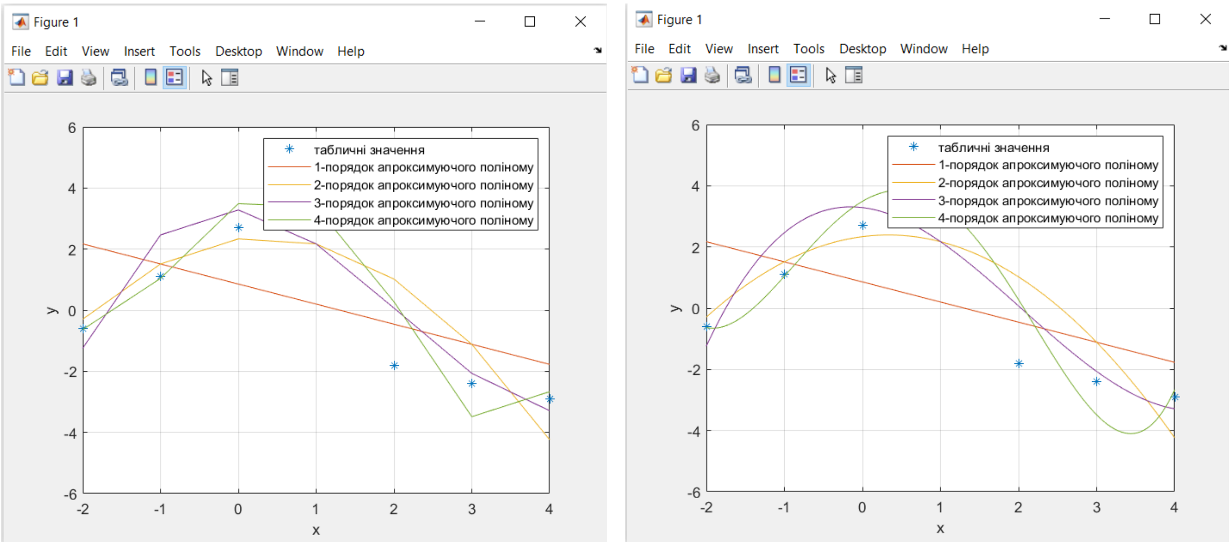


Рис.8.3. Приклади застосування апроксимації

Лінійна інтерполяція в MatLab здійснюється за допомогою команди ***interp***1(X, Y, Xi), яка здійснює одновимірну табличну інтерполяцію. Звернення до неї у загальному випадку має вигляд:

interp1(X, Y, Xi, ’<метод>’)

і дозволяє додатково вказати метод інтерполяції:

‘linear’ – лінійна,

‘cubic’ – кубічна,

‘spline’ – кубічні сплайни.

Якщо метод не вказано, здійснюється за замовчуванням лінійна інтерполяція. Наприклад, (для того ж вектора):

```
x=[-2 -1 0 1 2 3 4];
y=[-0.6 1.1 2.7 5.3 -1.8 -2.4 -2.9];
x1=-2:0.01:4;
y1=interp1(x,y,x1);
y2=interp1(x,y,x1,'cubic');
y3=interp1(x,y,x1,'spline');
plot(x,y,'*',x1,y1,x1,y2,x1,y3); grid;
xlabel('x'); ylabel('y');
legend('Табличні значення','Лінійна інтерполяція', ...
    'Кубічна інтерполяція', 'Інтерполяція сплайнами')
```

Результати роботи наведено на рис. 8.4.

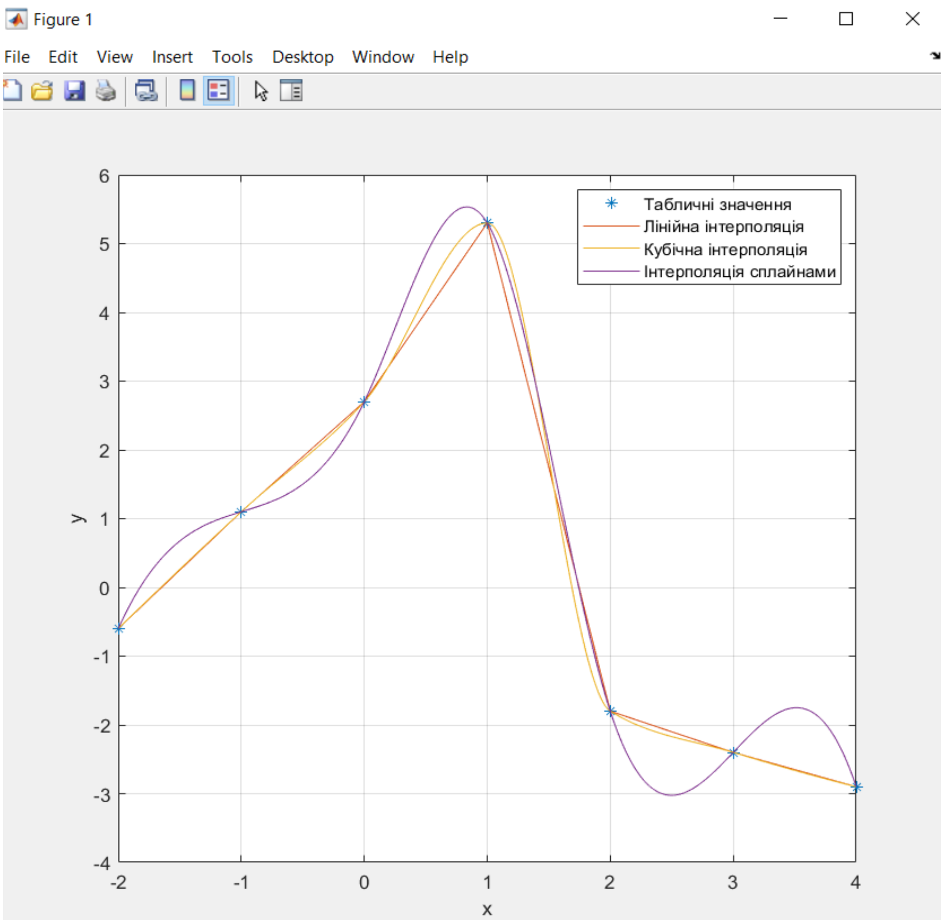


Рис.8.4 Приклад застосування процедури ***interp1***

8.2. Завдання на виконання комп’ютерного практикуму

Завдання 1. За допомогою функції ***polyfit*** провести поліноміальне апроксимування даних, значення векторів X та Y обираються з табл. 8.1 відповідно до варіанту.

Таблиця 8.1

Варіанти індивідуальних завдань

Варіант	**Вектор X**	**Вектор Y**
1	2	3
1	$[1\ 3\ 4\ 7\ 8\ 12\ 15\ 17]$	$[-0.5\ -0.2\ 0\ 0.2\ 0.6\ 0.8\ 1\ 1.8]$
2	$[1\ 2\ 3\ 4\ 7\ 8\ 9\ 12]$	$[0\ 0.1\ 0.8\ 1\ 1.8\ 1.9\ 2\ 2.2]$
3	$[1\ 2\ 6\ 8\ 10\ 14\ 15\ 18]$	$[-1\ -0.5\ -0.3\ -0.2\ 0\ 0.2\ 0.5\ 1]$
4	$[1\ 2\ 3\ 4\ 7\ 8\ 9\ 10]$	$[-0.9\ -0.5\ 0\ 0.1\ 0.2\ 1\ 2\ 3]$
5	$[1\ 2\ 4\ 6\ 7\ 11\ 14\ 16]$	$[-3\ -2\ -1.2\ -0.8\ 0\ 0.2\ 0.3\ 1]$
6	$[1\ 3\ 5\ 8\ 11\ 12\ 15\ 18]$	$[0\ 0.2\ 0.4\ 0.6\ 0.8\ 0.9\ 1\ 1.2]$
7	$[-0.2\ 0\ 0.2\ 0.4\ 0.8\ 1\ 1.8\ 2.1]$	$[-3\ -2\ 0\ 1\ 1.5\ 2\ 1.4\ 0.8]$
8	$[1\ 3\ 5\ 7\ 8\ 9\ 10\ 13]$	$[0\ 1\ 1.5\ 2.1\ 4.2\ 3.1\ 2.5\ 3]$
9	$[-0.6\ 0\ 0.6\ 1\ 1.8\ 2.1\ 2.4\ 3]$	$[-5\ -2\ 0\ 2\ 3.5\ 4\ 3.4\ 1.8]$
10	$[1\ 2\ 3\ 7\ 10\ 13\ 15\ 17]$	$[-2\ -1.8\ -1.4\ 0\ 0.3\ 0.5\ 0.7\ 1]$
11	$[1\ 2\ 4\ 6\ 7\ 9\ 13\ 18]$	$[-4\ -3.5\ -1.2\ -0.8\ -0.4\ 0\ 0.5\ 1]$
12	$[1\ 2\ 5\ 7\ 11\ 12\ 15\ 18]$	$[-2\ -1.5\ -0.3\ -0.1\ 0\ 0.4\ 0.5\ 1]$
13	$[1\ 2\ 4\ 7\ 9\ 12\ 15\ 21]$	$[-0.8\ -0.3\ 0\ 0.3\ 0.5\ 1.8\ 1\ 0.8]$
14	$[-1.2\ 0\ 1.2\ 2.4\ 2.8\ 4\ 6.8\ 8.1]$	$[-5\ -3\ 0.1\ 2\ 3.5\ 2\ 1.8\ 0.5]$
15	$[1\ 3\ 5\ 8\ 9\ 12\ 15\ 19]$	$[0\ 0.2\ 1.4\ 2.6\ 2.9\ 1\ 0.2\ -1.5]$
16	$[1\ 2\ 6\ 7\ 9\ 11\ 12\ 13]$	$[-3\ -2.5\ -1\ -0.6\ -0.2\ 0.5\ 0.8\ 1]$
17	$[-1.5\ 0\ 0.1\ 0.7\ 1.8\ 3\ 2.8\ 2.1]$	$[-7\ -3\ -2\ 1\ 2.5\ 3\ 4.4\ 3.8]$
18	$[-5\ -2\ 5\ 6\ 7\ 8\ 12\ 15]$	$[0\ 0.2\ 0.4\ 1.6\ 3.9\ 5\ 4.2\ 2.5]$
19	$[1\ 3\ 6\ 8\ 9\ 12\ 15\ 17]$	$[-0.5\ -0.01\ 0\ 1.2\ 1.6\ 1.8\ 2.2\ 2.8]$
20	$[1\ 3\ 6\ 7\ 9\ 12\ 14\ 21]$	$[-1\ -0.8\ -0.4\ 0\ 0.2\ 0.6\ 0.8\ 1.5]$
21	$[-2\ 1\ 3\ 5\ 7\ 8\ 9\ 15]$	$[-1.3\ -1.2\ -0.1\ 0\ 0.2\ 2.4\ 1.6\ 0.8]$
22	$[1\ 2\ 3\ 5\ 7\ 8\ 10\ 12]$	$[-2\ -1.5\ -1\ -0.8\ 0\ 0.2\ 0.3\ 1]$
23	$[-0.4\ 0\ 0.4\ 0.8\ 1.8\ 1\ 1.8\ 2.1]$	$[-3\ -1\ 0\ 1\ 2.5\ 4\ 2.4\ 0.8]$
24	$[1\ 2\ 3\ 5\ 7\ 8\ 9\ 10]$	$[-0.3\ -0.2\ -0.1\ 0\ 0.2\ 0.4\ 0.6\ 0.8]$
25	$[1\ 2\ 3\ 6\ 9\ 10\ 11\ 12]$	$[-1\ -0.8\ -0.4\ 0\ 0.2\ 0.6\ 0.8\ 1.2]$
1	2	3
26	$[1\ 2\ 3\ 4\ 5\ 8\ 9\ 12]$	$[0\ 0.2\ 0.4\ 0.6\ 0.9\ 1\ 1.2\ 1.5]$

27	$[1\ \ 3\ \ 6\ \ 8\ \ 9\ \ \ 12\ \ 15\ \ 17]$	$[-0.5 - 0.01\ \ \ 0\ \ 1.2\ \ \ 1.6\ \ \ 1.8\ \ 2.2\ \ 2.8]$
28	$[1\ \ 2\ \ 6\ \ 7\ \ 9\ \ \ 11\ \ 12\ \ 13]$	$[-2 - 1.5\ \ - 1\ \ - 0.8 - 0.2\ \ \ 0.3\ \ 0.5\ \ 1]$
29	$[1\ \ 2\ \ 3\ \ 5\ \ 6\ \ 8\ \ 9\ \ 12]$	$[-3\ \ - 2\ \ - 1.5\ \ - 1\ \ - 0.8\ \ 0\ \ 0.8\ \ 1]$
30	$[1\ \ 3\ \ 4\ \ 6\ \ 8\ \ 9\ \ 11\ \ 15]$	$[-2 - 1.5 - 1.2\ \ - 0.8\ \ - 0.2\ \ 0\ \ 0.3\ \ 1]$

Завдання 2. За допомогою функції ***interp1*** провести інтерполяцію даних трьома методами (лінійною, кубічною та сплайнами), значення векторів X та Y обираються з табл. 8.1, значення X1 - табл. 8.2 відповідно до варіанту.

Таблиця 8.2

Варіанти індивідуальних завдань

Варіант	**Вектор X1**	**Варіант**	**Вектор X1**
1	$1{:}\,0.1{:}\,17$	16	$1{:}\,0.01{:}\,13$
2	$1{:}\,0.2{:}\,12$	17	$-1.5{:}\,0.02{:}\,2.1$
3	$1{:}\,0.05{:}\,18$	18	$-5{:}\,0.1{:}\,15$
4	$1{:}\,0.1{:}\,10$	19	$1{:}\,0.01{:}\,17$
5	$1{:}\,0.2{:}\,16$	20	$1{:}\,0.01{:}\,21$
6	$1{:}\,0.05{:}\,18$	21	$-2{:}\,0.1{:}\,15$
7	$-0.2{:}\,0.01{:}\,2.1$	22	$1{:}\,0.01{:}\,12$
8	$1{:}\,0.05{:}\,13$	23	$-0.4{:}\,0.02{:}\,2.1$
9	$-0.6{:}\,0.01{:}\,3$	24	$1{:}\,0.05{:}\,10$
10	$1{:}\,0.02{:}\,17$	25	$1{:}\,0.1{:}\,12$
11	$1{:}\,0.05{:}\,18$	26	$1{:}\,0.05{:}\,12$
12	$1{:}\,0.1{:}\,18$	27	$1{:}\,0.1{:}\,17$
13	$1{:}\,0.02{:}\,21$	28	$1{:}\,0.01{:}\,13$
14	$-1.2{:}\,0.1{:}\,8.1$	29	$1{:}\,0.02{:}\,12$
15	$1{:}\,0.1{:}\,19$	30	$1{:}\,0.01{:}\,15$

8.3. Контрольні запитання

1. Що таке інтерполяція? Що таке апроксимація?
2. У чому полягає різниця між інтерполяцією та апроксимацією?
3. Які методи інтерполяції ви знаєте? Як виконати інтерполяцію в системі MatLab?
4. Які методи апроксимації ви знаєте? Як виконати апроксимацію в системі MatLab?

5. Як задати порядок полінома при апроксимації?

6. Як вибрати тип інтерполянта (метод інтерполяції)?

7. Чим відрізняється кубічна інтерполяція від інтерполяції сплайнами?

8. Який метод найчастіше застосовується при визначенні коефіцієнтів апроксимуючих поліномів?

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 9
ЧИСЕЛЬНЕ ІНТЕГРУВАННЯ ТА ДИФЕРЕНЦІЮВАННЯ ФУНКЦІЙ. ЗНАХОДЖЕННЯ МІНІМУМІВ ФУНКЦІЙ

Мета: ознайомитися з методами чисельного інтегрування функцій, вивчити можливості системи MatLab для чисельного інтегрування та знаходження мінімумів функцій.

9.1. Теоретичні відомості

При розв'язуванні багатьох інженерних задач доводиться обчислювати визначені інтеграли від функцій, у тому числі і від заданих таблично окремими значеннями аргументу (на рис. 9.1 наведено графічне подання такої функції). Методи чисельного інтегрування широко застосовуються при автоматизації розв'язку наукових та інженерних задач, тому що їх простіше реалізувати програмно ніж аналітичні методи.

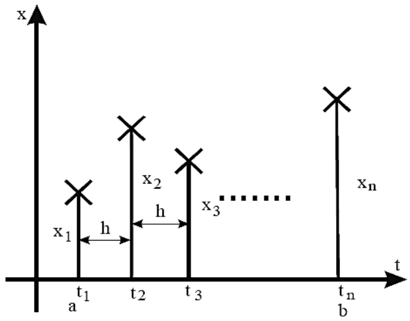


Рис. 9.1 Графічне подання табличної функції

Методи чисельного інтегрування функцій спираються на ідею заміни підінтегральної функції $f(x)$ деякою наближаючою її функцією $G(x_i)$, інтеграл від якої обчислюється досить просто. Наближення, зазвичай, здійснюється інтерполюванням у межах заданого діапазону змінювання аргументу. Також існує підхід, у якому інтерполяційною формулою наближається не підінтегральна функція $f(x)$, а будується наближення інтеграла на основі апроксимації розрахованих раніше наближень шуканого інтеграла. Всі методи чисельного інтегрування можна поділити на дві великі групи:

1) з постійним кроком сітки інтегрування;

2) зі змінним кроком сітки інтегрування;

Більш детальна класифікація даних методів показана на рис.9.2.

Методи чисельного інтегрування

з постійним кроком сітки

1) Методи засновані на визначенні інтеграла:

— метод односторонніх прямокутників (лівих або правих);

— метод середніх прямокутників.

2) Методи Ньютона-Котеса (інтерполяційні квадратурні формули)

— метод трапецій;

— метод Сімпсона;

— метод Сімпсона правило $3/8$;

— метод Буля;

— методи вищих порядків.

3) Рекурсивні алгоритми:

— метод Ромберга;

— метод прямокутників-трапецій.

зі змінним кроком сітки

— квадратурні формули Чебишева;

— квадратурні формули Гауса.

Рис. 9.2 Класифікація методів чисельного інтегрування функцій

Найбільш загальні вимоги, які пред’являють до методів чисельного інтегрування, наступні:

- *Універсальність*. Функція $f(x)$ може бути задана у вигляді «чорного ящика».
- *Економічність*. Кількість обчислень функції $f(x)$ має бути зведено до можливого мінімуму.
- *Хороша обумовленість*. Чисельні похибки Δf при розрахунку значень функції $f(x)$ не мають приводити до значної підсумкової похибки.

Опишемо найбільш відомі методи чисельного інтегрування функцій. Всі формули для розрахунку квадратур зведено до таблиці 9.1.

1. Метод правих прямокутників. Цей найпростіший метод чисельного інтегрування полягає у тому, що на кожному кроці $x_i \le x \le x_{i+1}$ інтегрування функція $f(x)$ замінюється на сталу величину, рівну значенню $f(x_i)$ на лівому кінці відповідного інтервалу (рис.9.3).

Отже, для визначення повного інтегралу за методом правих прямокутників достатньо просумувати усі задані значення функції, окрім останнього, і результат домножити на крок інтегрування $h = b-a/_{n-1}$. При цьому похибка методу визначається середнім значенням похідної на інтервалі визначення інтегралу і пропорційна другому степеню кроку. Зменшуючи величину кроку інтегрування, можна зменшити похибку розрахунку інтегралу.

2. Метод лівих прямокутників. Сутність цього методу аналогічна, за винятком того, що апроксимуючі прямокутники тепер примикають зліва (рис. 9.4) до заданих точок.

Властивості цього методу майже не відрізняються від попереднього. Слід звернути увагу лише на те, що похибка його має зворотний знак. З цього одразу можна зробити висновок, що комбінуючи ці два методи, можна суттєво збільшити точність чисельного інтегрування. У такий спосіб приходимо до методу трапецій.

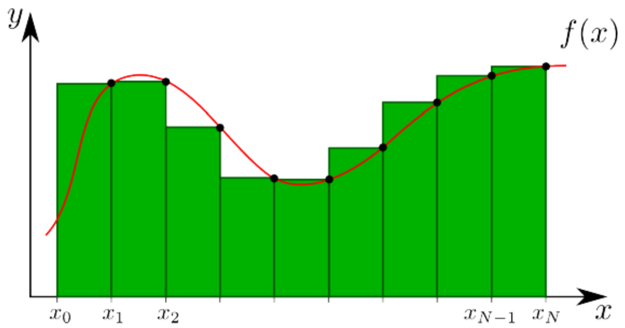


Рис. 9.3. Графічне подання метода правих прямокутників

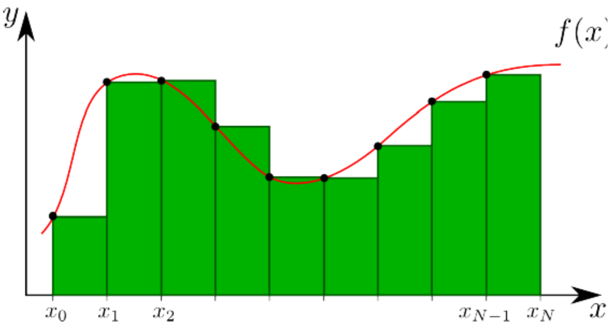


Рис. 9.4 Графічне подання метода лівих прямокутників

3. Метод середніх прямокутників. Якщо висоту прямокутників розраховувати по значенням функції посередині між вузлами, то будуть отримані формули методу середніх прямокутників. Даний метод має другий порядок сходимості відносно кроку інтегрування h (рис. 9.5).

4. Метод трапецій. Метод трапецій відноситься до сімейства методів Ньютона-Котеса. У даних методах відрізок інтегрування розбивається на декілька рівних відрізків точками $x_1, x_2, x_3 \dots x_n$. Квадратурні формули цих методів отримуються шляхом заміни підінтегральної функції її кусково-поліноміальним інтерполянтом (многочленом Лагранжа різної степені).

Графічна інтерпретація методу трапецій полягає в тому, що цим методом обчислюються площі трапецій (рис. 9.6), які утворюються при з'єднанні окремих заданих точок відрізками прямих — тобто за лінійної інтерполяції.

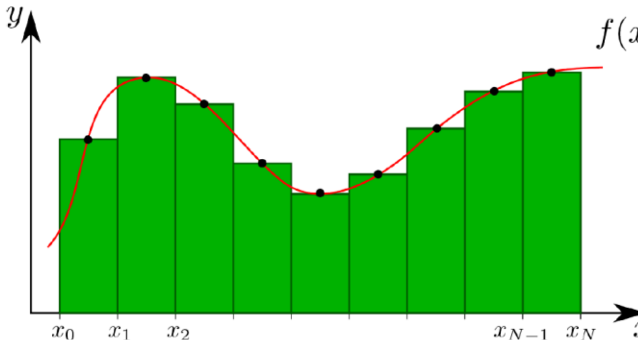


Рис. 9.5. Графічне подання метода середніх прямокутників

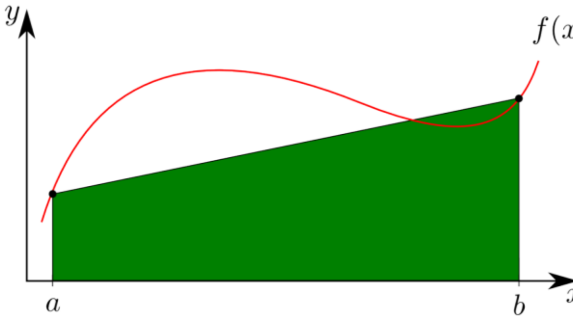


Рис. 9.6. Графічне подання метода трапецій

5. Метод Сімпсона. У даному методі (рис. 9.7 та 9.8) інтервал інтегрування *[a,b]* необхідно поділити на парну кількість рівних частин так, що загальна кількість точок буде непарною, а крок інтегрування дорівнюватиме

$$h=(b-a)/(n-1)=(b-a)/2(k-1).$$

На кожному з інтервалів виконується інтерполяція заданих точок квадратною параболою. Інтегруючи цю функцію в інтервалі $x_i \le x \le x_{i+2}$, можна одержати формулу Сімпсона для окремої ділянки . Підсумовуючи цей результат по усіх частинних відрізках $(i=1,3,5,...,n-2)$, отримують *квадратурну формулу Сімпсона.*

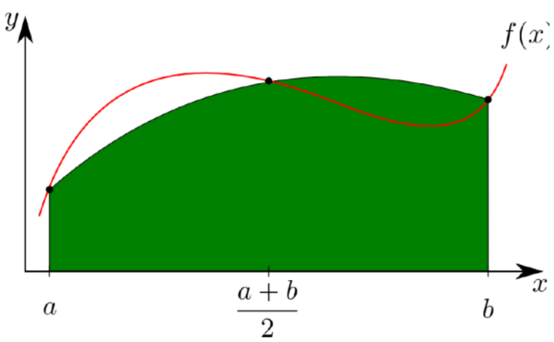


Рис. 9.7. Графічне подання метода Сімпсона

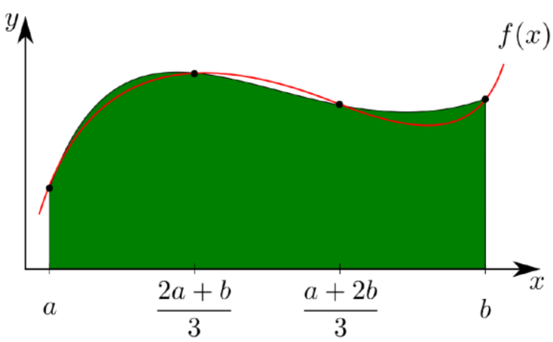


Рис. 9.8. Графічне подання метода Сімпсона-правило 3/8

4. Методи Ньютона-Котеса вищих порядків. Для сімейства методів Ньютона-Котеса відомий загальний вираз:

$$\int_a^b f(x)dx \approx \frac{r \cdot h}{C_n} \sum_{j=1}^{N} \sum_{i=0}^{r} c_{ir} f(x_i) \qquad (9.1)$$

де r – порядок метода Ньютона-Котеса, N – кількість часткових відрізків, $h = \dfrac{x_j - x_{j-1}}{n}$, $C_n = \sum\limits_{i=0}^{r} c_{ir}$, $x_j = x_j + i \cdot h$.

З виразу (9.1) легко отримуються формули методу трапецій для r=1, методу Сімпсона для r=2. Коефіцієнти c_{ir} можуть бути задані таблично (табл. 9.1).

Таблиця 9.1

Вагові коефіцієнти для квадратурних формул Ньютона-Котеса

r	C_n	c_{0r}	c_{1r}	c_{2r}	c_{3r}	c_{4r}	c_{5r}	c_{6r}	c_{7r}	c_{8r}
0	1	1								
1	2	1	1							
2	6	1	4	1						
3	8	1	3	3	1					
4	90	7	32	12	32	7				
5	288	19	75	50	50	75	19			
6	840	41	216	27	272	27	216	41		
7	17280	751	3577	1223	2989	2989	1323	3577	751	
8	28350	989	5888	-928	10496	-4540	10496	-928	5888	989

Похибку метода оцінюють з розгляду залишкових членів (див. останній стовпчик табл. 9.2). Їх формула є степеневою функцією від кроку інтегрування. *Показник степеня у цій залежності похибки від кроку прийнято називати порядком метода інтегрування*. Таким чином, методи прямокутників є методами першого порядку, метод трапеції – другого порядку, а метод Сімпсона є методом четвертого порядку.

На відміну від чисельного диференціювання, чисельне інтегрування є *обчислювально стійкою операцією*. Похибка будь-якого методу зменшується зі збільшенням кількості заданих точок початкової функції у заданому інтервалі, причому степінь зменшення залежить від методу.

Чисельне диференціювання необхідно для знаходження похідних функцій першого та вищих порядків, у випадку коли знаходження похідних аналітично ускладнено або взагалі неможливо, або функція задана набором точок. Існують наступні методи чисельного диференціювання:

1. диференціювання з використанням інтерполяційних поліномів:
   1.1. з використанням інтерполяційних поліномів Ньютона;
   1.2. з використанням інтерполяційних поліномів Лагранжа;
   1.3. безрізницеві формули чисельного диференціювання;
2. метод невизначених коефіцієнтів;
3. чисельне диференціювання за допомогою сплайнів;
4. сіткові методи:
   4.1. метод односторонньої різниці;
   4.2. метод двосторонньої різниці.

Коротко охарактеризуємо сіткові методи. Математично похідна функції визначається наступним чином

$$f'(x_0) = \frac{\partial f}{\partial x} = \lim_{dx \to 0} \frac{f(x_0 + dx) - f(x_0)}{dx} \qquad (9.2)$$

Замінюючи в (9.2) приріст dx на скінченну величину h, яка називається кроком диференціювання, отримується наближена формула для розрахунку похідної

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} \qquad (9.3)$$

Якщо $f(x)$ неперервна, то для обчислення похідної необхідно розрахувати значення функції у точках x_0 та $(x_0 + h)$, та згідно формули (9.3) чисельно розрахувати значеннях похідної в точці x_0. Якщо функція задана вибіркою (рис.9.1), то вираз для чисельного диференціювання буде

$$f'(x_0) = f_i' = \frac{f(x_0 + h) - f(x_0)}{h} = \frac{f_{i+1} - f_i}{h} \qquad (9.4)$$

Формула (9.4) називається **правосторонньою різницею**, тому що похідна оцінюється по значенням функції в даній x_i та наступній точці x_{i+1}. Аналогічно можна записати вираз для **лівосторонньої різниці**

$$f'(x_0) = f_i' = \frac{f(x_0) - f(x_0 - h)}{h} = \frac{f_i - f_{i-1}}{h} \qquad (9.5)$$

Похибки методу у формулах правосторонньої (9.4) та лівосторонньої різниці (9.5) рівноцінні. Більш точне значення похідної можна отримати використовуючи **метод двохсторонньої різниці** (особливо для гладких функцій)

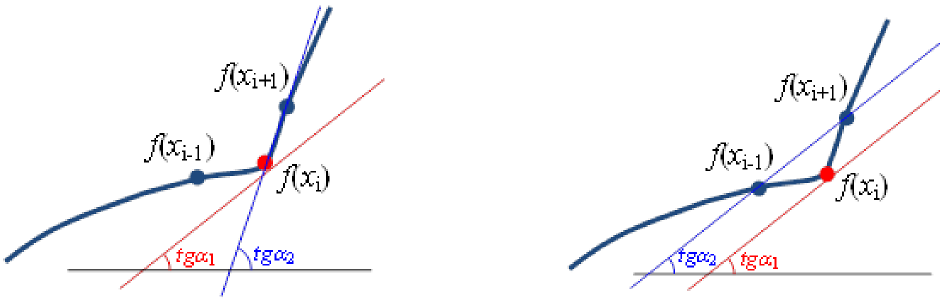
$$f'(x_0) = f_i' = \frac{f(x_0 + h) - f(x_0 - h)}{2h} = \frac{f_{i+1} - f_{i-1}}{h} \qquad (9.6)$$

Таблиця 9.2

Квадратурні формули основних методів чисельного інтегрування функцій

Назва методу	**Порядок**	**Квадратурна формула**	**Залишковий член квадратурної формули** $E[f]$
Метод правих прямокутників	1	$\int_a^b f(x)dx = h\sum_{i=1}^{n} f(x_{i-1})$	$-\frac{(b-a)^2}{2} f'(x)$
Метод лівих прямокутників	1	$\int_a^b f(x)dx = h\sum_{i=1}^{n} f(x_i)$	$\frac{(b-a)^2}{2} f'(x)$
Метод середніх прямокутників	2	$\int_a^b f(x)dx = h\sum_{i=1}^{n} f(x_{i-1}+h/2)$	$\frac{h^2(b-a)}{24} f''(x)$
Метод трапецій	2	$\int_a^b f(x)dx = h\left(\frac{f(x_1)+f(x_n)}{2}+\sum_{i=1}^{n-1} f(x_i)\right)$	$-\frac{h^2\cdot(b-a)}{12} f''(x)$
Метод Сімпсона	4	$\int_a^b f(x)dx = \frac{h}{3}\left(f(x_1)+4f(x_2)+2f(x_3)+4f(x_4)+2f(x_5)+\ldots\right.$ $\left.+2f(x_{n-2})+4f(x_{n-1})+f(x_n)\right)$	$-\frac{h^4(b-a)}{180} f^{(4)}(x)$
Метод Сімпсона правило $3/8$	4	$\int_a^b f(x)dx = h\left(\frac{3}{8}f(x_1)+\frac{7}{6}f(x_2)+\frac{23}{24}f(x_3)+f(x_4)+f(x_5)+\ldots\right.$ $\left.+f(x_{n-4})+f(x_{n-3})+\frac{23}{24}f(x_{n-2})+\frac{7}{6}f(x_{n-1})+\frac{3}{8}f(x_n)\right)$	$-\frac{h^4(b-a)}{405} f^{(4)}(x)$
Методи Ньютона-Котеса вищих порядків	r	$\int_a^b f(x)dx \approx \frac{r\cdot h}{C_n}\sum_{j=1}^{N}\sum_{i=0}^{r} c_{ir} f(x_i)$, $C_n = \sum_{i=0}^{r} c_{ir}$, $x_j = x_j + i\cdot h$.	$E[f] \le M\cdot C\left(\frac{b-a}{2}\right)^{n+1}$, $M = \max_{[a,b]}\left\|f^{(n)}(x)\right\|$, $C = \frac{1}{n!}\int_a^b \prod_{i=1}^{n}\|x-x_i\|dx$

Для візуального порівняння односторонньої та двохсторонньої різниці зручно скористатися графічним представленням похідної – тангенс кута нахилу дотичної до функції в точці x_i. На рис. 9.9 точному значенню похідної відповідає $tg\alpha_{іст}$. У методі правосторонньої різниці (рис. 9.9, а) це значення похідної оцінюється тангенсом кута нахилу $tg\alpha_{оцін}$ прямої, що проходить через точки x_i та x_{i+1}. Якщо в околицях точки x_i функція не гладка, то значення оцінки похідної $tg\alpha_{оцін}$ буде значно відрізнятися від точного $tg\alpha_{іст}$. Використовуючи метод двохсторонньої різниці, провівши пряму через точки x_{i-1} і x_{i+1} (рис. 9.9, б), можна отримати оцінку похідної, що практично співпадає з точним.



а) одностороння різниця б) двохстороння різниця

Рис.9.9. Графічна візуалізація похідної

Похідні вищих порядків. При розрахунку похідних вищих порядків похідна (n)-ого порядку вважається першою похідною від похідної $(n-1)$-ого порядку. Таким чином друга похідна функції є першою похідною від першої похідної і чисельно може бути розрахована

$$\frac{d^2 f}{dx^2} = \frac{d}{dx}\left(\frac{df}{dx}\right) = \frac{f'_{i+1} - f'_{i-1}}{2h} = \frac{\dfrac{f_{i+2} - f_i}{2h} - \dfrac{f_i - f_{i-2}}{2h}}{2h} = \frac{f_{i+2} - 2f_i + f_{i-2}}{(2h)^2} \qquad (9.7)$$

Засоби чисельного диференціювання і інтегрування на ЕОМ

Практично в усіх мовах програмування високого порядку відсутні вбудовані функції, які б дозволяли здійснювати чисельне диференціювання і інтегрування функцій. Деякі вбудовані функції (процедури) передбачені лише у новітніх комп‘ютерних математичних системах.

У системі MatLab існує декілька вбудованих функції, які здійснюють чисельне інтегрування функцій – ***trapz, integral, quad, quadl, quadgk*** і ***quadv***. Функції ***quad, quadl, quadgk*** і ***quadv*** не рекомендується використовувати. Система MatLab рекомендує, всюди де це можливо, замінити ці функції використанням функції ***integral***. У даній роботі більш детально зупинимося на використанні функції ***trapz*** та ***integral,*** проте далі коротко опишемо всі вище приведені функції.

Q = ***trapz***(X, Y) – виконує чисельне інтегрування методом трапецій функції *Y* заданої таблично вздовж аргументу, заданого у векторі *X*. Якщо вектор аргументів *X* відсутній — ***trapz***(Y), то система вважає, що значення функції *Y* отримані для аргументів з кроком рівним 1. Значення інтегралу присвоюється змінній *Q*.

Обчислення інтегралів методами квадратур за допомогою функцій ***integral, quad, quadl, quadgk*** і ***quadv*** здійснюється у випадку, коли функція задається у m-файлі або за допомогою запису функції в inline-форматі. Виклик даних функцій в основному має однаковий формат, відрізняючись набором опційних налаштувань:

Q = ***quad***(*FUN, a, b, tol, trace*) або

[Q, *fcnt*] = ***quad***(*FUN, a, b, tol, trace*) – використовується рекурсивна адаптивна формула Сімпсона.

Q = ***quadl***(*FUN, a, b, tol, trace*) або

[Q, *fcnt*] = ***quadl***(*FUN, a, b, tol, trace*) – використовується рекурсивна адаптивна формула Лобатто.

Q = ***quadgk***(*FUN, a, b, Name, Value*) або

[Q, *errbnd*]= ***quadgk***(*FUN, a, b, Name, Value*) – використовуються адаптивні формули Гаусса-Кронрода вищих порядків..

Q = ***integral***(*FUN, a, b, Name, Value*) використовуються глобальні адаптивні формули.

Обов'язковими являються параметри *FUN, a, b*, а виклик функцій у форматі з квадратними дужками є опційним. Він надає додаткову інформацію щодо роботи функцій чисельного інтегрування.

a – початкове значення аргументу функції, яку потрібно проінтегрувати.

b – кінцеве значення аргументу функції, яку потрібно проінтегрувати.

Fcnt — кількість раз розрахунку підінтегральної функції.

Errbnd — скаляр, який визначає приблизне значення верхньої границі абсолютної похибки.

FUN – задає функцію, яка буде проінтегрована. Задається у форматі посилання на функцію. Існує три варіанти задання підінтегральної функції:

1) зовнішній m-файл. Тоді *FUN* є рядком, у якому записано ім’я m-файлу. У даному файлі підінтегральна функція записується відповідно до правил MatLab, наприклад, для функції $y = \frac{1}{x^2 + 2x + 3}$:

```
function y = example_function(x)
y = 1./(x.^2+2*x+3);
```

У цьому випадку значення змінної FUN має наступний вигляд

```
FUN = 'example_function';
```

2) функція задається аналогічно п.1, проте у якості змінної *FUN* використовується не строкова змінна з її ім’ям, а посилання на цю функцію

```
FUN = @example_function
```

3) використання анонімних посилань на функцію

```
FUN = @(x) 1./(x.^3-2*x-5);
```

tol – точність інтегрування, яка задається максимально допустимою абсолютною похибкою. За замовчуванням, тобто у випадку коли користувач не вказує цього значення, значення абсолютної похибки становить 10^{-6}. Цей параметр задається тільки у функціях ***quad*** і ***quadl***.

Name, Value – іменований параметр опцій та його значення відповідно. Значення *Name* можуть бути наступними '*AbsTol*' — абсолютна похибка. Наприклад, запис

integral(*FUN, a, b, 'AbsTol', 1e-10*)

встановлює абсолютну точність інтегрування на рівні 10 знаків після коми.

'RelTol' — відносна похибка. Формат задання аналогічний попередній опції.

Приклад. Знайти значення інтеграла від функції $y(x) = 2\sin(2x) + 3x^2$ у діапазоні від 0 до $\frac{\pi}{2}$. Точне значення дорівнює $2 + \frac{\pi^3}{8} \simeq 5.87578458503748$.

Розв’язок

1) інтегрування методом трапецій. Використаємо рівномірну сітку з 100 точок.

```
>> x = 0:(pi/2)/100:pi/2;
>> y = 2*sin(x) + 3*x.^2;
>> Q = trapz(x, y)
Q =
          5.87593725074594
>> delta = 2+pi^3/8-Q
```

```
delta =
      -0.00015266570846606
```

2) інтегрування методом Сімпсона з використання анонімних посилань на функцію (параметри точності методу – за замовчуванням)

```
>> FUN = @(x) 2*sin(x) + 3*x.^2;
>> Q = quad(FUN, 0, pi/2)
Q =
           5.87578458078372
 >> delta = 2+pi^3/8-Q
delta =
       4.25376178725401e-09
```

3) інтегрування за допомогою рекурсивної адаптивна формули Лобатто з використання анонімних посилань на функцію (параметри точності методу – за замовчуванням)

```
>> FUN = @(x) 2*sin(x) + 3*x.^2;
>> Q = quadl(FUN, 0, pi/2)
Q =
           5.87578458502097
 >> delta = 2+pi^3/8-Q
delta =
       1.65059077517071e-11
```

4) інтегрування за допомогою глобальних адаптивних формули (параметри точності методу – за замовчуванням). Підінтегральну функцію запишемо у окремому m-файлі. Текст цього файл приведений нижче

```
function y = myfun(x)
y = 2*sin(2*x)+3*x.^2;
Інтегрування функції в консолі:
>> FUN = 'myfun';
>> Q = integral(FUN, 0, pi/2)
Q =
           5.87578458503748
>> delta = 2+pi^3/8-Q
delta =
       8.88178419700125e-16
5) інтегрування з вказанням абсолютної точності
>> Q = quad(FUN, 0, pi/2, 1e-14)

Q =
```

```
            5.87578458503748
>> delta = 2+pi^3/8-Q
delta =
      8.88178419700125e-16
>> Q = integral(FUN, 0, pi/2, 'AbsTol', 1e-14)
Q =
            5.87578458503748
>> delta = 2+pi^3/8-Q
delta =
      8.88178419700125e-16
6) інтегрування з вказанням відносної точності
>> Q = integral(FUN, 0, pi/2, 'RelTol', 1e-5)
Q =
            5.87578458503748
>> delta = 2+pi^3/8-Q
delta =
      8.88178419700125e-16
```

Чисельне диференціювання можна виконати з використанням функцій ***diff*** та ***gradient.*** Спочатку розглянемо функцію ***diff***, що дозволяє знаходити скінченні різниці:

Y = ***diff***(*X, n, dim*)

Параметри *n, dim* є необов'язковими.

X — вектор значень функцій, для якої потрібно обчислити скінченні різниці.

n — порядок скінченної різниці. ***diff***(*X, n)* рекурсивно *n* раз застосовує ***diff***(*X*) до попереднього результату виконання функції. Тобто ***diff***(*X, 2)* аналогічно до ***diff***(***diff***(*X*)).

dim — розмірність масиву вздовж якої треба виконати розрахунок скінченних різниць.

Використовуючи дану функцію, першу похідну можна розрахувати по формулах односторонньої різниці наступним чином

```
h = 0.001;         % крок табулювання функції
X = 0:h:pi;        % аргумент
f = sin(X);        % значення функції
Y = diff(f)/h;     % перша похідна
```

Функція ***gradient*** призначена для розрахунку градієнту функції, заданої таблично. Загальний виклик цієї функції має наступний вигляд

[FX, FY, FZ, ..., FN] = ***gradient***(*F*, *hx*, *hy*, ..., *hN*)

Проте нас цікавить виклик даної функції для розрахунку похідної

FX = ***gradient***(*F*, *h*),

що повертає одновимірний чисельний градієнт вектора *F*. Вихідний вектор *FX* відповідає $\partial F/\partial x$ з кроком *h*. Якщо крок не вказано, то система за замовчуванням прирівнює його 1. При цьому для чисельного розрахунку функція використовує формулу двохсторонньої різниці

FX(j) = (*F*(j+1) - *F*(j-1)/(2*h).

Для знаходження екстремумів функції (точок локального максимуму чи мінімуму) потрібно знайти розв'язки рівняння

$$\partial F/\partial x=0.$$

Дану процедуру можна виконати чисельно за допомогою відповідних функцій MatLab або графічно. У даному комп'ютерному практикумі рекомендується використати графічний спосіб.

9.2. Завдання на виконання комп'ютерного практикуму

Завдання 1. Чисельно розрахувати значення інтеграла від функцій в табл.9.3 в діапазоні аргументу від x_1 до x_2. Діапазон аргументу функції рівномірно розбити на m точок. Для чисельного інтегрування використати метод трапецій та метод вказаний у завданні. Інтегрування виконати з функцією, записаною у окремому m-файлі, так і використовуючи анонімне посилання на функцію. Побудувати графік підінтегральної функції.

Таблиця 9.2

Варіанти індивідуальних завдань

Варіант	x_1	x_2	m	$f(x)$	**Метод**
1	2	3	4	5	6
1	2	5	120	$5\sin(x)\cos(2x)$	***quad***
1	-1	4	70	$tg(x)\cos(2x)$	***quadl***
2	3	10	50	$e^{-0.1x}\cos(x)$	***quadgk***
3	0.3	3	60	$\cos(x)/(x^2+3x)$	***integral***
4	0.4	4	50	$x^2\ln(1+x)$	***quadl***

1	2	3	4	5	6
5	0.5	5	30	$\ln(x\cos(x)+10)$	***integral***
6	0.6	6	220	$x\cos(x)/e^{x}$	***quadl***
7	100	120	250	$arctg(x)\cdot\cos(5x)$	***integral***
8	0.8	8	45	$\cos(x)\cdot 5^{x}$	***quad***
9	5.1	11	110	$ctg(x^3+2x^2)$	***quadgk***
10	11.2	12	50	$\cos ec(10+\frac{1}{x^2})$	***integral***
11	1.3	13	300	$arctg(\frac{1}{5x^4+x^2})$	***quadl***
12	1.4	14	75	$\arccos(\frac{1}{5x^5+10x^3+2})$	***integral***
13	1.5	15	80	$arctg(e^{x})$	***quad***
14	1.6	10.6	70	$8x^3+1/x^2$	***quadl***
15	0.9	9	50	$\sin(x^3+5/3x^2+1)$	***quadgk***
16	-2	10	50	$\cos(\ln(x+5))$	***quadgk***
17	0.6	5	50	$\cos(x)+1/x^2$	***integral***
18	-0.9	0.9	45	$arcctg(\sin(x))$	***quadl***
19	1	20	60	$sh(x^2)$	***integral***
20	1	20	50	$ch(x)/(0.4x^2+6)$	***quad***
21	-0.9	0.9	100	$arctg(x+\cos(x))$	***quad***
22	1	20	300	$arctg(0.07x^3+1/x)$	***integral***
23	-0.8	0.8	50	$\arcsin(x^4+3x^2)$	***quad***
24	-0.8	0.8	50	$\arccos(5^{x})$	***quadgk***
25	-5	5	50	$e^{5\cos(x)}$	***integral***
26	1	20	300	x^4+3x^2	***quad***
27	3	7.8	50	$x^2+3\cos x^2$	***quadl***
28	-0.8	0.8	150	$x^2+3\cos 0.1x^3$	***quadgk***
29	-5	5	50	$e^{-5tg(x)}$	***integral***
30	5	9	120	$\arcsin(x+2^{x})$	***quadl***

Завдання 2. Чисельно розрахувати значення похідної методом односторонньої різниці від функцій в табл. 9.3 в діапазоні аргументу від x_1 до x_2. Діапазон аргументу функції рівномірно розбити на m точок. Побудувати графік похідної.

Завдання 3. Чисельно розрахувати значення похідної методом двохсторонньої різниці від функцій в табл. 9.3 в діапазоні аргументу від x_1 до x_2 . Діапазон аргументу функції рівномірно розбити на m точок. Побудувати графік похідної. Порівняти з результатами отриманими у завданні 2. Знайти значення аргументу, при якому функція $f\left(x\right)$ має екстремум. Розрахувати значення $f(x)$ в точці екстремуму при його наявності.

9.3. Контрольні запитання

1. У чому полягає задача чисельного інтегрування функції?
2. Які існують найпростіші методи чисельного інтегрування функції?
3. Що являють собою методи Ньютона-Котеса?
4. Які Ви знаєте рекурсивні методи чисельного інтегрування?
5. Що таке порядок методу інтегрування? Який порядок має метод трапецій? Метод Сімпсона правило 3/8?
6. Як можна оцінити похибку методу чисельного інтегрування?
7. Які вбудовані функції, що здійснюють чисельне диференціювання і чисельне інтегрування функції, існують у мовах високого рівня, математичних комп‘ютерних системах, у MatLab?
8. Чим відрізняється обчислення інтегралу у функціях ***trapz***, ***quad*** і ***integral*** системи MatLab?
9. Чим відрізняються методи ***quad, quadl*** та ***quadgk.*** У яких умовах краще використовувати кожен з цих методів? формули використовуються для інтегрування у функціях ***quad*** і ***quadl*** системи MatLab?
10. Які Ви знаєте методи чисельного диференціювання?
11. У чому різниця між функціями ***diff*** і ***gradient***?
12. Як чисельно розрахувати похідну в MatLab?
13. Що таке екстремум? Як визначити екстремуми функцій?

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 10
ЧИСЕЛЬНЕ ІНТЕГРУВАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

Мета: навчитися створювати власні програмні засоби у системі MatLab для виконання чисельного інтегрування звичайних диференціальних рівнянь заданим методом. Навчитися використовувати вбудовані в систему MatLab функції чисельного інтегрування диференціальних рівнянь.

10.1. Теоретичні відомості

Багато технічних задач, пов'язаних з розробкою і дослідженням складних технічних систем та їх компонентів, описуються з використанням диференціальних рівнянь та їх систем. Будь-яка задача проектування, яка зв'язана з розрахунком потоків енергії чи руху тіл, в кінцевому рахунку зводиться до розв'язку диференціальних рівнянь. Розв'язок більшості цих задач дуже важко отримати у аналітичному вигляді, а інколи швидше і раціональніше мати певний чисельний розв'язок, отриманий на ЕОМ. У такому випадку використовуються спеціальні чисельні методи розв'язку диференціальних рівнянь. Більшість інженерних задач з досить високим ступенем наближення дуже добре описуються звичайними диференціальними рівняннями.

Спочатку нагадаємо, що диференціальними рівняннями називають рівняння, у якому невідома функція входить під знаком похідної чи диференціала:

$$y''_x + z'_t = x^2 + \ln t\,,\ \frac{\partial y}{\partial x} = x^3 - 1,\ \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = x + y\,,\ xdy = y^3 dx\,.$$

Якщо невідома функція, що входить у диференціальне рівняння, залежить тільки від однієї незалежної змінної, то диференціальне рівняння називається звичайним, наприклад:

$$x\frac{d^2 y}{dx^2} = 5,\quad 5V\,dt = V\,dt\,,\ 5x'' + x' - 2 = 10\sin\omega t\,.$$

Для чисельного розв'язку звичайного диференціального рівняння (ЗДР) найчастіше використовується постановка у формі задачі Коші. Задачею Коші називають задачу розв'язання диференціального рівняння:

$$\frac{dy}{dt} = f\left(t, y\right) \tag{10.1}$$

де $f\left(t, y\right)$ — задана неперервна функція двох аргументів, із початковою умовою:

$$y(t=0)=y_0 \tag{10.2}$$

При чисельному розв’язку задачі Коші по змінній t вводять сітку $0<t_1<t_2<....<t_n$ і шукають значення невідомої функції у вузлах сітки $0, y_1, y_2, ..., y_n$.

Відповідно необхідно приводити ЗДР чи систему ЗДР до форми Коші.

Форма Коші – це форма запису ЗДР, розв’язаних виключно відносно перших похідних:

$$\frac{dx_1}{dt}=a_{11}x_1+a_{12}x_2+a_{12}x_3+f_1(x_1,x_2,x_3,t)$$
$$\frac{dx_2}{dt}=a_{21}x_1+a_{22}x_2+a_{22}x_3+f_2(x_1,x_2,x_3,t)$$
$$\frac{dx_3}{dt}=a_{31}x_1+a_{32}x_2+a_{32}x_3+f_3(x_1,x_2,x_3,t)$$

Чисельні методи розв'язання задачі Коші ділять на 3 групи (див. рис.10.1):

- однокрокові;
- багаткокрокові (методи прогнозу та корекції);
- методи з автоматичним вибором кроку інтегрування.

Чисельні методи розв’язання задачі Коші

Однокрокові методи

1) Ейлера
2) Модифікований Ейлера
3) Удосконалений Ейлера
4) Розенберга
5) Хойне
6) Рунге-Кутти

Багатокрокові методи

1) Адамса
2) Хеммінга
3) Мілна

Методи з автоматичним вибором кроку

1) Рунге-Кутти з автоматичним вибором крока (схема Дорманда-Принса)
2) Рунге-Кутти-Мерсона
3) Формули Адамса-Башфорта-Моултона
4) Формули чисельного диференціювання

Рис.10.1. Класифікація методів чисельного інтегрування ЗДР

Чисельні методи розв'язку ЗДР базуються на розкладі шуканої функції $y(t)$в ряд Тейлора в околі певної точки, найчастіше початкової,

$$y(t_0+h)=y(t_0)+hy'(t_0)+\frac{1}{2}h^2y''(t_0)+\ldots+\frac{1}{n!}h^ny^{(n)}(t_0)+\ldots, \quad (10.3)$$

де h – крок інтегрування, тобто відстань між поточною точкою x_i та наступною $x_{i+1}=x_i+h$, в якій обчислюють розв'язок ЗДР.

Безпосереднє використання формули (10.3) зводиться до обчислення n похідних (якщо вони існують) і знаходження суми усіченого ряду Тейлора:

$$y(t)\approx\sum_{i=0}^{n}\frac{y^{(i)}(t_0)}{i!}(t-t_0)^i \quad (10.4)$$

Проте на практиці формула (10.4) потребує багато обчислень і не завжди сходиться до істинного розв'язку ЗДР. Більш поширеним є використанням класу методів Рунге-Кутти, які теж базуються на розкладі (10.3), проте при побудові інтегрального розв'язку використовують різні інтерполяційні підходи.

У загальному випадку обчислення за методами Рунге-Кутти має форму

$$y(t+h)\approx z(h)=y(t)+\sum_{i=1}^{q}p_ik_i(h)+\mathrm{O}(h), \quad (10.5)$$

де коефіцієнти k_i є функціями, які розраховуються на кожному кроці інтегрування на базі правих частин ЗДР, а p_i — фіксовані коефіцієнти, що залежать від типу методу і його порядку, q — порядок методу інтегрування, $\mathrm{O}(h)$ — залишковий член, який визначає похибку методу інтегрування.

Функції k_i з (10.5) мають наступний вид

$$\begin{aligned} &k_1(h)=h\,f(t,y)\\ &k_2(h)=h\,f\left(t+\alpha_2h,y+\beta_{21}k_1(h)\right),\\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ &k_q(h)=hf\left(t+\alpha_qh,y+\beta_{q1}k_1(h)+\ldots+\beta_{q,q-1}k_{q-1}(h)\right)\end{aligned} \quad (10.6)$$

Коефіцієнти $\alpha_2\ldots\alpha_q,\beta_{i,j}\,(0<j<i\le q)$ залежать від методу інтегрування та вимог, що до нього ставляться. По суті в (10.6) обчислюється функція правих частин ЗДР $f(t,y)$ із запису (10.1) в різних точках, поступово приближуючи кожну з вищих похідних. Формулу (10.5) зручно подати у наступному виді

$$\begin{aligned} &y(t+h)\approx y(t)+h*\mathbf{F}(y,t)\\ &\mathbf{F}(y,t)=\sum_{i=1}^{q}b_ik_i(h)\end{aligned} \quad (10.7)$$

У табл. 10.1 приведено найбільш відомі та вживані методи Рунге-Кутти. Для їх реалізації потрібно обчислювати в MatLab значення правих частин ЗДР, які можуть описуватися різними функціями. При цьому бажано використовувати інструменти, що дозволяють гнучко змінювати функцію для обчислень, не вносячи багато рутинних і не потрібних правок у коді програми.

Таблиця 10.1

Методи Рунге-Кутти, $y_{m+1} = y_m + h \cdot \mathbf{F}(t_m, y_m)$

№ п/п	Формула метода	Допоміжні величини	Порядок	Назва методу
1	2	3	4	5
1	$F = k_1$	$k_1 = Z(t_m, y_m)$	1	Ейлера
2	$F = (k_1 + k_2)/2$	$k_1 = Z(t_m, y_m)$ $k_2 = Z(t_m + h, y_m + hk_1)$	2	Модифіко-ваний Ейлера-1
3	$F = Z(t_m + \frac{h}{2}, y_m + h\frac{k_1}{2})$	$k_1 = Z(t_m, y_m)$	2	Модифіко-ваний Ейлера-2
4	$F = \frac{k_1 + 4k_2 + k_3}{6}$	$k_1 = Z(t_m, y_m)$ $k_2 = Z(t_m + \frac{h}{2}, y_m + h\frac{k_1}{2})$ $k_3 = Z(t_m + h, y_m + h(2k_2 - k_1))$	3	Хойне-1
5	$F = (k_1 + 3k_3)/4$	$k_1 = Z(t_m, y_m)$ $k_2 = Z(t_m + \frac{h}{3}, y_m + h\frac{k_1}{3})$ $k_3 = Z(t_m + 2\frac{h}{3}, y_m + 2h\frac{k_2}{3})$	3	Хойне-2
6	$F = \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$	$k_1 = Z(t_m, y_m)$ $k_2 = Z(t_m + h/2, y_m + hk_1/2)$ $k_3 = Z(t_m + h/2, y_m + hk_2/2)$ $k_4 = Z(t_m + h, y_m + hk_3)$	4	Рунге-Кутти-1
1	2	3	4	5

7	$F=\frac{k_1+3k_2+3k_3+k_4}{8}$	$k_1=Z(t_m,y_m)$ $k_2=Z(t_m+\frac{h}{3},y_m+h\frac{k_1}{3})$ $k_3=Z(t_m+2\frac{h}{3},y_m+h\frac{k_2-k_1}{3})$ $k_4=Z\left(t_m+h,y_m+h\left(k_1-k_2+k_3\right)\right)$	4	Рунге-Кутти-2

Деякі алгоритми є загальними для усіх видів функцій певного вигляду. Тому їх програмна реалізація є єдиною, але потребує знання алгоритму обчислення функції, який може суттєво змінюватися при переході від однієї функції до іншої. Останній алгоритм може бути зафіксований у вигляді деякої файл-функції. Щоб перший, більш загальний алгоритм був пристосований для будь-якої функції, що використовується в ньому, потрібно, щоб ім'я цієї функції набувало будь-якого вигляду, тобто було деякою змінною, що набуває певного значення (текстового імені файл-функції) лише при зверненні до основного алгоритму.

До таких функцій належать процедури:

- обчислення інтегралів від функції, які потребують вказання імені m-файлу, що містить обчислення значення підінтегральної функції;
- чисельного інтегрування диференційних рівнянь, які потребують вказання імені m-файлу, в якому обчислюються праві частини рівнянь у формі Коші;
- алгоритмів чисельного обчислення коренів нелінійних алгебричних рівнянь, які потребують вказання файл-функції, корінь якої відшукується.

У MatLab є зручна функція, що викликає інші функції по їх іменах, які передаються їй як аргумент, у вигляді строкової (текстової) змінної. Це функція ***feval***. У MatLab будь-яка функція (процедура), наприклад за ім'ям *FUN1,* може бути виконана не лише за допомогою звичайного звернення вигляду:

[y1,y2….,yk] = ***FUN1***(x1, x2, …., xn),

а й за допомогою функції ***feval*** таким чином:

[yl ,y2,...,yk] = ***feval***(*'FUN1'*, xl, x2, ..., xn),

де ім'я функції FUN1 є вже однією з вхідних змінних – текстовою змінною, і тому її поміщено у апострофи.

Перевагою виклику функції у другій формі є те, що він має той самий вигляд при змінюванні імені функції, наприклад, на *FUN2*. Це дозволяє уніфікувати звернення до усіх функцій певного вигляду, тобто таких, що мають однакову кількість вхідних і вихідних параметрів відповідного типу. При цьому ім’я функції, яка використовується, може бути довільним і змінюватися протягом повторних обчислень.

Через те, що при виклику функції за допомогою процедури ***feval*** ім’я функції розглядається як один з вхідних параметрів процедури, ім’я функції можна використовувати як змінну і оформляти у m-файлі звернення до неї, не маючи ще конкретного імені функції.

Приклад 1. Реалізувати користувацьку функцію, яка розраховує значення функції $y(t)=5e^{-0.4t}\sin(10t+0.2)+y_0$ для аргументу $t\in[0;2]$ для довільного значення константи y_0, яка визначається користувачем. Виконати виклик цієї функції за допомогою функції ***feval.***

Створимо m-файл, який реалізує функцію $y(t)$. Дамо функції ім’я AttenuateExp. Не забувайте (**!!!**), що ім’я файлу-функції має співпадати з ім’ям функції. Лістинг приведено нижче

```
function y=AttenuateExp(t, y0)
y=5*exp(-0.4*t)*sin(10*t+0.2) +y0;
end
```

Створимо скрипт файл, який спочатку викликає дану користувацьку функцію безпосередньо по імені та за допомогою функції ***feval.*** Результати розрахунку виведено на один графік, для наглядності змінено значення константи y_0. Лістинг основного скрипта приведено нижче, а результати його роботи показано на рис.10.2.

```
Clc, clear all
format short g, format compact
time=0:0.01:2;
N=length(time);

for i=1:N
    y1(i) = AttenuateExp(time(i), 1);
    y2(i) = feval('AttenuateExp', time(i), 3);
end

figure(1)
plot(time, y1, 'b-', time, y2, 'r-o', 'LineWidth',
2),grid
xlabel('Час')
legend('y1 - виклик по імені функції', 'y2 - виклик через
```

```
feval')
```

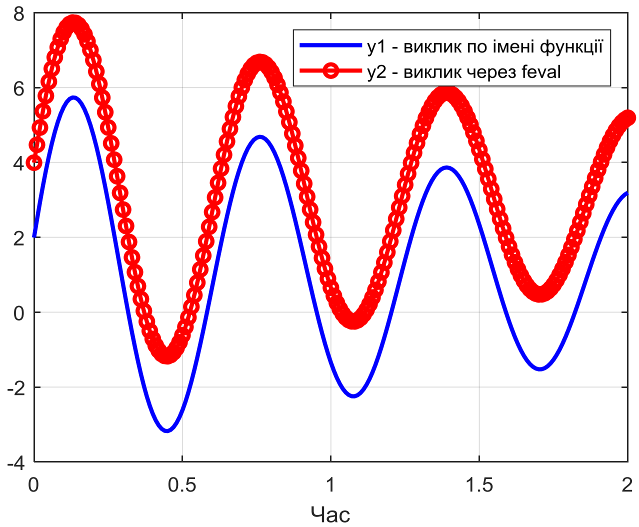


Рис.10.2. Результати виклику функції з використанням імені функції та за допомогою ***feval***

Далі покажемо, як за допомогою функції ***feval*** зручно реалізувати метод Рунге-Кутти. Це класичний приклад створення ***процедур від функцій.***

*Приклад 2****.*** *Процедура метода чисельного інтегрування Рунге-Кутти 4-го порядку.*

Нехай маємо ЗДР у вигляді Коші:

$$\frac{dy}{dt} = Z(y,t),$$

де y – вектор змінних стану системи; t – аргумент (час); Z – вектор заданих нелінійних функцій, який визначає систему ЗДР.

Для метода Рунге-Кутти 4-го порядку оберемо наступну форму функції $\mathbf{F}(y,t)$ з (11.7):

$$F = (k_1 + 3\cdot k_2 + 3\cdot k_3 + k_4)/8,$$

де

$$k_1 = Z(y,t);$$
$$k_2 = Z(y + h\cdot k_1/3, t + h/3);$$
$$k_3 = Z(y + h\cdot k_2 - h\cdot k_1/3, t + 2\cdot h/3);$$
$$k_4 = Z(y + h\cdot k_3 - h\cdot k_2 - h\cdot k_1, t + h).$$

Створимо m-файл функції, що здійснює ці обчислення, назвавши його ***rko43****:*

```
function [tout, yout]=rko43(Zpfun, h, t, y)
%RK041 Інтегрування ЗДР методом Рунге-Кутта 4-го порядку.
%     праві частини яких задані процедурою Zpfun.
```

```
%
%     Вхідні змінні:
% Zpfun - рядок, що містить ім'я процедури обчислення
правих частин ЗДР
%     звернення: z = fun(t,y), де Zpfun = 'fun'
%     t         поточний момент часу
%     y         вектор поточних значень змінних стану
%     z      — обчислені значення похідних z(i) =
dy(i)/dt.
%     h - крок інтегрування
%     t - попередній момент часу
%     y - попереднє значення вектора змінних стану.
% -----
%     Вихідні змінні:
% tout - новий момент часу
% yout - обчислене (проінтегроване) значення вектора
змінних стану
%        через крок інтегрування
% Ю.Ф. Лазарев, С.Л. Лакоза, Д.О. Півторак, 274-67-31.
% Copyright (c) 2020 by The PSON, HTU of Ukraine "KPI".

% Розрахунок проміжних значень похідних
kl = feval(Zpfun, t,        y);
k2 = feval(Zpfun, t+h/3,    y+h/3*kl);
k3 = feval(Zpfun, t+2*h/3, y+h*k2-h/3*kl);
k4 = feval(Zpfun, t+h,      y+h*(k3+kl-k2));

% Розрахунок нових значень вектора змінних стану
tout = t + h;
yout = y + h* (kl + 3*k2 + 3*k3 + k4)/8;
end %Кінець процедури RK043
```

<u>Зверніть увагу на такі обставини:</u>

- звернення до процедури обчислення правих частин не конкретизовано; ім’я цієї процедури входить до вхідних змінних процедури інтегрування і конкретизується при зверненні до останньої;
- проміжні змінні *k* є векторами-рядками (те саме стосується і змінних

y, а також змінних z, що обчислюються у процедурі правих частин).

Приклад 3. Інтегрування ЗДР механічного гіроскопічного датчика кутової швидкості, який призначений для вимірювання абсолютної кутової швидкості об'єкта, на якому його встановлено (точніше – проекції кутової швидкості на напрямок осі чутливості).

Розглянемо процес приведення ЗДР до форми Коші та створення відповідної процедури правих частин на прикладі диференціальних рівнянь, що відображають основні особливості гіротахометрів (так часто називають механічні гіроскопічні датчики кутової швидкості.) Спочатку розробимо програмний комплекс інтегрування ЗДР для обертання об'єкта з постійною кутовою швидкістю, а потім ускладнимо рівняння для коливального руху об'єкта, наприклад, при хитавиці корабля на хвилях. Диференціальне рівняння, що описує роботу гіротахометра має вигляд

$$J \cdot \ddot{\beta} + h\dot{\beta} + c\beta = H\omega_\xi \cos\beta - H\omega_\eta \sin\beta \tag{10.8}$$

де β – кут повороту рухомої частини датчики, по якому можна визначити кутову швидкість об'єкта ($\omega_\xi = \frac{c}{H}\beta$); J – момент інерції гірокамери; h – коефіцієнт демпфування; c – жорсткість механічної чи електричної пружини; H – власний кінетичний момент (момент кількості руху); ω_ξ, ω_η – проекції кутової швидкості основи. Складова ω_ξ являється інформаційною складовою, яку вимірює гіротахометр, а ω_η – це вплив перехресного каналу, що вносить похибки у вимірювання кутової швидкості ω_ξ.

Перше, що треба виконати – це привести початкове диференціальне рівняння (10.8) до форми Коші. Для цього необхідно понизити порядок ЗДР (10.8) шляхом введення нових змінних:

$$y_1 = \beta; \quad y_2 = \dot{\beta}.$$

Тоді (10.8) матиме наступну форму

$$y_1 = \beta; \quad y_2 = \dot{\beta}; \qquad J\dot{y}_2 + h y_2 + cy_1 = H\omega_\xi \cos y_1 - H\omega_\eta \sin y_1.$$

Приведемо отриману проміжну форму до вигляді двох диференційних рівнянь 1-го порядку, розв'язаних відносно похідних:

$$\begin{cases} \dfrac{dy_1}{dt} = y_2; \\ \dfrac{dy_2}{dt} = \dfrac{1}{J}\left(-h y_2 - cy_1 + H\omega_\xi \cos y_1 - H\omega_\eta \sin y_1\right). \end{cases} \tag{10.9}$$

Порівнюючи отриману систему з загальною формою рівнянь Коші, яка використовувалася у прикладі 2, можна зробити висновок, що

$$z1 = \frac{dy_1}{dt} = y_2;$$

$$z2 = \frac{dy_2}{dt} = \frac{1}{J}\left(-h\,y_2 - cy_1 + H\omega_\xi \cos y_1 - H\omega_\eta \sin y_1\right).$$

<u>**Саме обчислення цих двох функцій і повинно відбуватися у процедурі правих частин**</u>. Назвемо майбутню процедуру правих частин гіротахометра *GyroTahometr_RP*. Вихідною змінною у ній буде вектор

z = [zl, z2],

а вхідними – момент часу t та вектор проінтегрованих на попередньому кроці значень функції

y = [y1, y2].

Деякою складністю є те, що сталі коефіцієнти, які є у правих частинах у (10.9), не можна передавати у функцію через її заголовок. Виходом є об'єднання цих коефіцієнтів у вектор коефіцієнтів K=[J,h, c, H]. Для того, щоб функція правих частин могла бачити ці змінні їх необхідно оголосити глобальними (***global*** K) і прописати це у головній програмі, і у підпрограмі правих частин.

Тоді m-файл матиме такий вигляд:

```
function z = GyroTahometr_RP(t, y)
% Процедура правих частин ЗДР Гіроскопічного датчика
кутової швидкості
%                            (Гіротахометра)
% Дана функція здійснює розрахунок вектора "z" похідних
від вектора "Y"
% змінних стану. ЗДР записано у формі Коші
% Коефіцієнти рівняння передаються у функцію через
глобальний вектор K:
%      K = [ J, h, c, H, om_ksi, om_eta]

global K
J = K(1);
h = K(2);
c = K(3);
H = K(4);
om_ksi = K(5);
om_eta = K(6);
```

```
z(1) = y(2);
z(2) = 1/J*(-h*y(2) - c*y(1) + H*om_ksi*cos(y(1) - 
H*om_eta*sin(y(1))));
end
```

При використанні цієї процедури слід пам’ятати, що у тексті програми попередньо має бути визначений вектор K з 6 елементів і потім він має бути перетворений на глобальний за допомогою службового слова ***global*.**

Для організації чисельного інтегрування ЗДР необхідно створити головний m-файл (скрипт-файл), у якому задати необхідні коефіцієнти ЗДР та організувати циклічне обчислення протягом необхідного часу для досліджень. У прикладі взято час дослідження в 5 секунд, крок інтегрування h=0.01с (спробуйте самостійно змінити крок на h=0.1с і побачите, що для заданого ЗДР метод зривається). Лістинг М-файлу приведений нижче. А результати роботи показано на рис.10.3:

```
% Головна програма для виконання чисельного інтегрування
% ЗДР гіроскопічного датчика кут. швидкості 
(гіротахометра)
clc
clear all

format short g, format compact

global K

%% Параметри гіротахометра
% K = [ J, h, c, H, om_ksi, om_eta]
K(1) = 5e-5;% J - момент інерції рухомої частини 
гіротахометра [кг м/с^2]
K(2) = 0.0003;% h - коефіцієнт демпфірування [Нм с]
K(3) = 0.025;% c - жорсткість механ. або електрично 
пружини [Нм/рад]
K(4) = 0.02;% H - власний кінетичний момент в[Нм с].
            % Щоб перевести в [Г см с] необхідно 
домножити на 10^4
```

```
K(5) = 0.2;% om_ksi [1/c]
K(6) = 0*0.01;% om_eta  [1/c]

%% Параметри інтегрування
t_bgn = 0; % Початковий момент часовго відрізку
t_end = 5; % Кінцевий момент часовго відрізку
h = 0.01; % Крок інтегрування

time = t_bgn:h:t_end; % Час інтегрування

y0 = [3*pi/180, 0]; % Початкові умови
    % y0(1) - початкове значення кута повороту Beta
гіротахометра [рад]
    % y0(2) - початкове значення кут. швидкості dBeta/dt
гіротахометра [рад/с]

N = length(time);% Кількість точок для розрахунку

y = zeros(N,2);
y(1,:) = y0;

for i=2:N
    Zpfun = 'GyroTahometr_RP';
    t = time(i);

[tout, yout] = rko43(Zpfun, h, t, y(i-1, :));
%Інтегрування методом Рунге-Кутти 4-го порядку
 y(i, :) = yout; % накопичення проінтегрованих вихідних
змінних

end


figure(1)
subplot(2,1,1)
plot(time, y(:, 1)*180/pi,'b-','LineWidth', 2), grid
set(gca,'FontSize',11)
title('Інтегрування ЗДР гіротахометра методом rko43')
```

```
ylabel('Градус')
xlabel('Час, с')
legend('\beta - куто поворотурухомої частини')

subplot(2,1,2)
plot(time, y(:, 2)*180/pi,'b-','LineWidth', 2), grid
set(gca,'FontSize',11)
ylabel('Градус / с')
xlabel('Час, с')
legend('d\beta/dt - кутова швидк. повороту рухомої
частини')
```

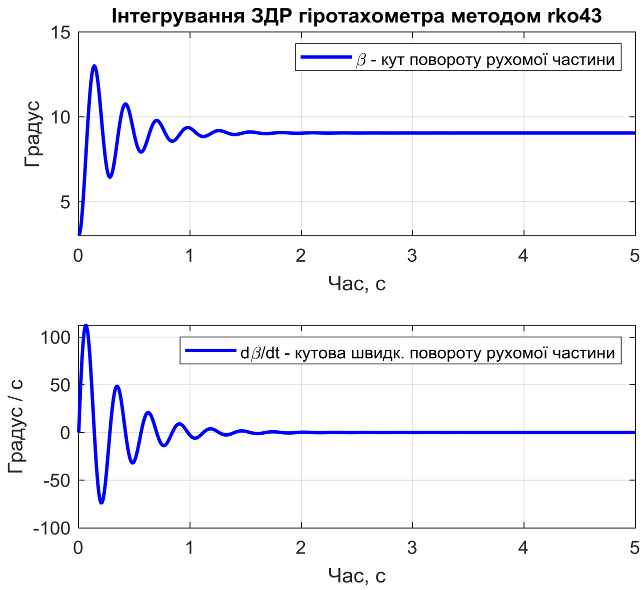


Рис.10.3. Графіки проінтегрованих змінних відповідно до ЗДР гіротахометра (10.9)

Розглянемо як задавати в підінтегральній функції складові, що змінюються в часі. Для цього візьмемо один з найпоширеніших динамічних рухів об'єкта – хитавицю. Використаємо найпростішу форму подання хитавиці та задамо її через гармонічну функцію у вигляді

$$\omega_{\xi} = \omega_{AMP} \sin\left(2\pi f_{xum} t + \varepsilon\right), \tag{10.10}$$

де ω_{AMP} – амплітуда гармонічної складової кутової швидкості хитавиці в [Гц], f_{xum} – циклічна частота хитавиці в [Гц], ε – певний випадковий зсув фаз. Для реалізації такої залежності нам треба розширити вектор коефіцієнтів $K=[\,J, h, c, H, \omega_{AMP}, f_{xum}, \varepsilon, \omega_{\eta}]$. Далі приведено тільки частину лістингу головного m-файлу, де враховані ці зміни

```
%% Параметри гіротахометра
```

```
% K = [ J, h, c, H, om_ksi_AMP,f_hut, eps, om_eta]
K(1) = 5e-5;% J - момент інерції рухомої частини
гіротахометра [кг м/с^2]
K(2) = 0.0003;% h - коефіцієнт демпфірування [Нм с]
K(3) = 0.025;% c - жорсткість механ. або електрично
пружини [Нм/рад]
K(4) = 0.02;% H - власний кінетичний момент в[Нм с].
            % Щоб перевести в [Г см с] необхідно
домножити на 10^4
K(5) = 0.2;% om_ksi_AMP - амплітуда кут.швидк.
гармонічної хитавиці [1/с]
K(6) = 0.5;% f_hut - циклічна частота хитавиці в [Гц],
K(7) = 0.2;% eps – певний випадковий зсув фаз.
K(8) = 0*0.01;% om_eta  [1/с]
```

Підпрограма правих частин *GyroTahometr_Hutav_RP* для гармонічної хитавиці (11.10) набуде вигляду

```
function z = GyroTahometr_Hutav_RP (t, y)
% Процедура правих частин ЗДР Гіроскопічного датчика
кутової швидкості
%                            (Гіротахометра)
% Дана функція здійснює розрахунок вектора "z" похідних
від вектора "Y"
% змінних стану. ЗДР записано у формі Коші
% Дана підпрограма призначена для дослідження гарм.
хитавиці.
%    om_ksi = om_ksi_AMP*sin(2*pi*f_hut*t + eps)
%
% Коефіцієнти рівняння передаються у функцію через
глобальний вектор K:
%     K = [ J, h, c, H, om_ksi_AMP,f_hut, eps, om_eta]

% Параметри гіротахометра
global K
J = K(1);
h = K(2);
c = K(3);
H = K(4);
```

```
om_ksi_AMP = K(5);
f_hut = K(6);
eps = K(7);
om_eta = K(8);
z(1) = y(2);
z(2) = 1/J*(-h*y(2) - c*y(1) ...
      + H*om_ksi_AMP*sin(2*pi*f_hut*t + eps)*cos(y(1) ...
      - H*om_eta*sin(y(1))));
end
```

Результати роботи оновленої програми для ЗДР, що враховують гармонічну хитавицю об'єкта, показано на рис.10.4

Обчислення умов руху та зовнішніх впливів можна винести в окрему функцію, яку теж передавати як параметр всім функціям, що були вищеописані. Проте це значно ускладнює програмну реалізацію і не дозволяє уніфікувати вид підпрограми правих частин для можливості використання у штатних інструментах системи MatLab.

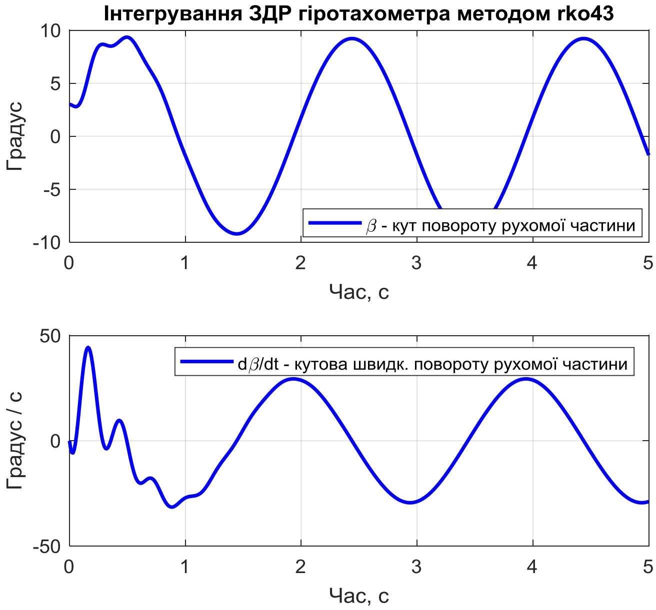


Рис.10.4 . Графіки проінтегрованих змінних ЗДР гіротахометра (10.9) для гармонічної хитавиці

У системі MatLab, як і в багатьох інших системах чисельного та символьного аналізу, реалізовані процедури, що дозволяють інтегрувати дуже широкий клас задач, включаючи ЗДР. До цього переліку відноситься, так зване, сімейство методів «ode» (скорочено від англ. *ordinary differential equations*):

1) ***ode45*** – даний метод використовує явні формули методів Рунге-Кутти 4, 5-го порядку, так звана пара рівнянь зі схеми Дорманда-Принса. Це перший

метод, який потрібно використовувати для розв'язку ЗДР. Якщо даний метод зривається чи дає незадовільний результат, тоді необхідно звернути увагу на приведені далі методи, котрі дають змогу добре інтегрувати жорсткі системи.

2) ***ode113*** — процедура, що використовується для інтегрування не жорстких ЗДР. Використовує методи різного порядку, а також автоматично підбирає крок інтегрування. Дана функція використовує розв'язувач, який реалізує формули прогнозу-корекції методів Адамса-Башфорта-Моултона з порядком від 1 до 13. Найвищий використовуваний порядок методу – 12, а метод 13-го порядку використовується для оцінки похибки інтегрування. Процедура ***ode113*** може бути ефективнішою за ***ode45*** при високих вимогах до точності інтегрування.

3) ***ode15s*** – метод, який ефективно використовувати для розв'язування жорстких ЗДР. У цій функції використовується розв'язувач зі змінним порядком та кроком, що базується на формулах чисельного диференціювання від 1 до 5-го порядку. Даний метод використовують у випадку зриву методу ***ode45*** або коли використання ***ode45*** є малоефективним (наприклад, для інтегрування жорстких ЗДР). Метод ***ode15s*** можна застосовувати для розв'язування диференціально-алгебраїчних рівнянь.

4) ***ode23*** – однокроковий метод другого порядку, який застосовують для розв'язування не жорстких ЗДР. Дана функція використовує явні формули Рунге-Кутти 2-го та 3-го порядку – формули Богацкі та Шемпайна. Метод ***ode23*** може бути ефективнішим у порівнянні з ***ode45*** при невисоких точностях та при наявності помірних жорсткостей у ЗДР.

5) ***ode23s*** – однокроковий метод другого порядку, який застосовують для розв'язування жорстких ЗДР. Дана функція використовує модифіковані формули Розенброка 2-го порядку. Метод ***ode23s*** може бути ефективнішим у порівнянні з ***ode15s*** при невисоких точностях та у випадку наявності швидкозмінних функцій.

6) ***ode23t*** –метод другого порядку, який раціонально використовувати для помірно жорстких ЗДР та диференціально-алгебраїчних рівнянь. Даний метод реалізує метод трапецій з вільною інтерполяцією, Цей метод дає хороші результати при розв'язуванні ЗДР, які описують коливальні системи з майже гармонічним вихідним сигналом.

Процедури ***ode45***, ***ode113***, ***od15s***, ***ode23***, ***ode23s***, ***ode23t*** (у формі виклику вони мають загальну назву «**odeNAME**») мають наступні форми виклику

[tout, yout] = **odeNAME** (odefun, tspan,y0);

[tout, yout] = **odeNAME** (odefun, tspan, y0, options);

[tout, yout, te, ye, ie] = **odeNAME** (odefun, tspan, y0, options),

де *yout* – чисельні значення розв'язку ЗДР у моменти часу *tout*; *odefun* – строкова змінна, що містить ім'я підпрограми правих частин (дивись далі *GyroTahometr_RP_ode*(t, y)); *tspan* – вектор, що вказує початковий та кінцевий момент часу інтегрування ЗДР або може містити моменти часу, для яких треба отримати розв'язки ЗДР; *y0* – вектор початкових умов, має містити таку кількість елементів, скільки змінних стану у ЗДРР; *options* – перелік опцій, що задається за допомогою функції ***odeset***. Дозволяє задати відносну та абсолютну точність, а також ще ряд важливих налаштувань. З можливостями налаштування методів «*ode*» більш детально можна познайомитися у довідці системи MatLab.

Як видно з приведеної вище форми виклику процедур інтегрування, то вони ідентичні для кожного з методів. Перелік параметрів теж збігається, а деякі відмінності можуть існувати лише в параметрі «*options*», який відповідає за налаштування методу інтегрування (проте перелік таких особливих налаштувань не великий). **Варто наголосити, що для процедур «*ode*» підпрограми правих частин мають повертати результати у вигляді вектора-стовпця!!!** Тобто потрібно транспонувати вектор вихідних значень. Відповідним чином модифікована підпрограма правих частин для гіротахометра, яку можна використовувати у процедурах ***ode*** набуде виду:

```
function z = GyroTahometr_RP_ode(t, y)
% Процедура правих частин ЗДР Гіроскопічного датчика кутової швидкості
%                              (Гіротахометра)
% Дана функція здійснює розрахунок вектора "z" похідних від вектора "Y"
% змінних стану. ЗДР записано у формі Коші
% Коефіцієнти рівняння передаються у функцію через глобальний вектор K:
%      K = [ J, h, c, H, om_ksi, om_eta]

global K
J = K(1);
h = K(2);
c = K(3);
H = K(4);
om_ksi = K(5);
om_eta = K(6);
```

```
z(1) = y(2);
z(2) = 1/J*(-h*y(2) - c*y(1) + H*om_ksi*cos(y(1) -
H*om_eta*sin(y(1))));
z=z'; % ТРАНСПОНУВАННЯ вектора "z" похідних від вектора
"Y"
end
```

Лістинг головного m-файлу, у якому організовано чисельне інтегрування з використанням вбудованих в MatLab процедур приведено нижче, а результати роботи показані на рис.10.5. Як видно, приведені результати співпадають з результатами на рис.10.3:

```
% Головна програма для виконання чисельного інтегрування
% ЗДР гіроскопчіного датчика кут. швидкості
(гіротахометра),
% використовуючи вбудовані в MATLAB процедури.
clc
clear all
format short g, format compact

global K
%% Параметри гіротахометра
% K = [ J, h, c, H, om_ksi, om_eta]
K(1) = 5e-5;% J - момент інерції рухомої частини
гіротахометра [кг м/с^2]
K(2) = 0.0003;% h - коефіцієнт демпфірування [Нм с]
K(3) = 0.025;% c - жорсткість механ. або електрично
пружини [Нм/рад]
K(4) = 0.02;% H - власний кінетичний момент в[Нм с].
            % Щоб перевести в [Г см с] необхідно
домножити на 10^4
K(5) = 0.2;% om_ksi [1/c]
K(6) = 0*0.01;% om_eta  [1/c]


%% Параметри інтегрування
t_bgn = 0; % Початковий момент часовго відрізку
t_end = 5; % Кінцевий момент часовго відрізку
```

```
t_intgration = [t_bgn t_end];

y0 = [3*pi/180, 0]; % Початкові умови
    % y0(1) - початкове значення кута повороту Beta
гіротахометра [рад]
    % y0(2) - початкове значення кут. швидкості dBeta/dt
гіротахометра [рад/с]

Zpfun = 'GyroTahometr_RP_ode';
[tout, yout]=ode45(Zpfun, t_intgration,y0);%Інтегрування
методом Рунге-Кутти 4-го порядку
% [tout, yout]=ode113(Zpfun, t_intgration,y0);%
% [tout, yout]=ode23(Zpfun, t_intgration,y0);%

%% Результати чисельного інтегрування ЗДР
figure(1)
subplot(2,1,1)
plot(tout, yout(:, 1)*180/pi,'b-','LineWidth', 2), grid
set(gca,'FontSize',11)
title('Інтегрування ЗДР гіротахометра методом ode45')
ylabel('Градус')
xlabel('Час, с')
legend('\beta - кут повороту рухомої частини')

subplot(2,1,2)
plot(tout, yout(:, 2)*180/pi,'b-','LineWidth', 2), grid
set(gca,'FontSize',11)
ylabel('Градус / с')
xlabel('Час, с')
legend('d\beta/dt - кутова швидк. повороту рухомої
частини')
```

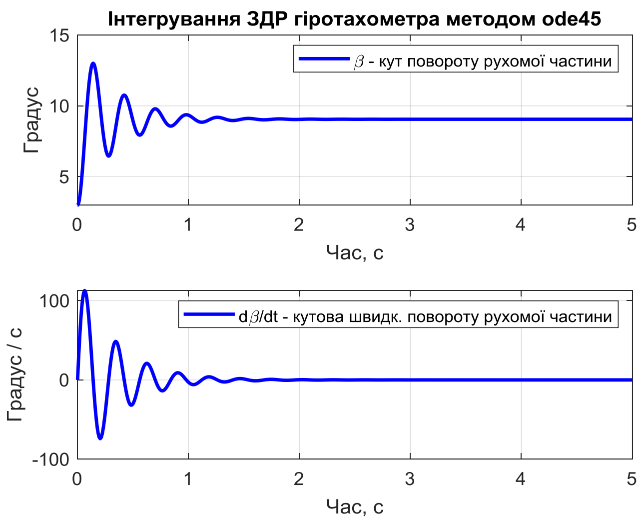


Рис.10.5. Графіки проінтегрованих змінних відповідно до ЗДР гіротахометра (10.9) процедурою ***ode45***

10.2. Завдання на виконання комп'ютерного практикуму

Завдання 1. Створити m-файл правих частин для диференціальних рівнянь відповідно до варіанту (табл. 10.2), реалізуйте інтегрування за допомогою вказаного методу Рунге-Кутти з таблиці 10.1. Початкові умови необхідно обрати самостійно (умови мають бути не нульові).

Таблиця10.2

Варіанти індивідуальних завдань

Варіант	Диференційне рівняння	Метод
1	2	3
1.	$y'' - y' - 12y = 5e^{-0.3t}t$	Хойне-2
2.	$y'' + 7y' + 6y = 5\sin 0.3x$	Ейлера
3.	$y'' + 4y' + 4y = 5\sin 0.1x - 2\cos 0.3x$	Рунге-Кутти-1
4.	$y'' - 4y' + 13y = 5e^{-5t}\sin 0.1t$	Рунге-Кутти-2
5.	$y'' + 8y' = 5^{-5t} + 0.1t$	Хойне-1
6.	$y'' + 25y = 0.1t$	Модифікований Ейлера-2
7.	$y''' - 2y'' - y' + y = e^{0.03t}$	Хойне-1
8.	$y''' - 2y'' - 15y' = 1/e^{3x}$	Рунге-Кутти-1
9.	$y''' - 2y'' + 16y' = \dfrac{1}{2 + 0.5\sin t}$	Модифікований Ейлера-1

1	2	3
10.	$y''' - 3y' - 2y = \dfrac{1}{2 + 0.5\cos t}$	Хойне-2
11.	$y''' - 3y'' + 3y' - y = 2 \cdot 5^{-0.08x}$	Рунге-Кутти-2
12.	$y''' + 64y' = 2 \cdot tg0.5x$	Хойне-1
13.	$y'' + y = \dfrac{1}{\cos 2t}$	Рунге-Кутти-1
14.	$y'' + 0.2y' + y = ctgt$	Рунге-Кутти-2
15.	$y'' - y = \dfrac{e^x}{e^x + 1}$	Хойне-2
16.	$y'' - 2y' + y = \dfrac{e^{-0.3t}}{t}$	Модифікований Ейлера-2
17.	$y'' + 9y = \dfrac{1}{\sin 3x}$	Хойне-1
18.	$y'' + 2y' + y = \dfrac{1}{xe^x}$	Рунге-Кутти-2
19.	$y'' + 4y = \dfrac{1}{\sin^2 x}$	Рунге-Кутти-1
20.	$y'' + 2y' + y = \dfrac{1}{\cos^3 3x}$	Модифікований Ейлера-2
21.	$y'' + y = tg1.5t$	Ейлера
22.	$y'' + 0.1y = \dfrac{1}{\cos 5t}$	Рунге-Кутти-1
23.	$y''' + 8y'' - 15y' + 0.1y = \dfrac{1 - arctgx}{1 + arctgx}$	Хойне-2
24.	$y''' + 8y'' + 3y' + y = \dfrac{1 - 2e^{5t}}{1 + e^{5t}}$	Рунге-Кутти-1
25.	$y''' + 8y = 3\sin e^{-0.7t}$	Рунге-Кутти-2
26.	$y'' + y' - 4y = 3e^{-0.7t} + 0.1\sin 2t$	Модифікований Ейлера-2
27.	$y'' + 0.1y' + 0.1y = \dfrac{e^{-t}}{2 - \cos 5t}$	Хойне-2
28.	$y''' - 2y'' + 16y' - 2y = \dfrac{1}{2 + 0.5e^{3t}}$	Модифікований Ейлера-1

1	2	3
29.	$y''' + 16y' - 2y = \dfrac{1}{2 + 0.5e^{-3t}}$	Рунге-Кутти-2
30	$y'' + 4y' + 2y = \ln 5x$	Хойне-1

Завдання 2. Проінтегруйте диференціальні рівняння із завдання (1) за допомогою методу ***ode45*** та одного з методів для розв’язування жорстких систем. Порівняйте результати цих методів. Порівняйте отримані результати із результатами із завдання 1. Виконайте побудову відповідних графіків.

10.3. Контрольні запитання

1. Які рівняння називають диференціальними? Як з диференціальних рівнянь відносяться до класу звичайних диференційних рівнянь?

2. Що таке задача Коші? Що таке нормальна форма Коші для ЗДР? Як привести ЗДР до форми Коші?

3. На чому базуються методи чисельного інтегрування ЗДР? У чому полягає задача чисельного інтегрування системи диференційних рівнянь?

4. Яку загальну форму мають методи Рунге –Кутти?

5. Як класифікують методи чисельного інтегрування ЗДР?

6. Що значить поняття «порядок методу»? Який порядок має метод класичний метод Рунге-Кутти?

7. Для чого прислуговується функція ***feval***? Як її застосовують?

8. Що таке процедура правих частин диференціальних рівнянь? Які дії мають передувати складанню цієї процедури? Що обчислюється внаслідок звернення до неї?

9. Як забезпечити передавання у процедуру правих частин диференціальних рівнянь потрібних сталих коефіцієнтів?

10. Що називають змінними стану системи? Чи збігаються вони з шуканими змінними заданих диференціальних рівнянь? Чим визначається кількість змінних стану?

11. Що робить процедура, яка реалізує той чи інший метод чисельного інтегрування диференціальних рівнянь? Які аргументи у такої процедури? Що обчислюється в результаті застосування цієї процедури?

12. До якої процедури обов’язково звертається процедура методу інтегрування? Як забезпечується звернення до потрібної процедури, якщо заздалегідь невідомо, які саме рівняння інтегруватимуться?

13. Чим відрізняються однокрокові методи різних порядків?

14. Чим відрізняються однокрокові методи від багатокрокових? Що таке формули прогнозу та корекції?

15. Що визначає величина порядку методу чисельного інтегрування?

16. Які функції MatLab здійснюють чисельне інтегрування диференціальних рівнянь і у який спосіб?

17. Як визначається крок інтегрування у вбудованих в систему процедурах інтегрування сімейства «*ode*»? Припустима похибка інтегрування? Як забезпечується інтегрування саме тих рівнянь, розв'язок яких потрібно знайти?

18. Як для вбудованих процедур MATLAB отримати вектор моментів часу, в яких отримані розв'язки ЗДР (тобто чисельно проінтегровано ЗДР) ?

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 11
РОЗВ'ЯЗОК РІВНЯНЬ У СИМВОЛЬНОМУ ВИГЛЯДІ

Мета роботи: ознайомлення з можливостями пакету MatLab для розв'язку рівнянь у символьному вигляді.

11.1. Теоретичні відомості

У системі MatLab для виконання символьних обчислень використовують символьні об'єкти. Для оголошення символьних змінних використовується команду ***syms***. Наприклад,

```
>> syms x y
```

Таким чином оголошується і використовується дві символьні змінні ***x*** та ***y***. Символьні вирази будуються з символьних змінних

```
>> a=5*sin(x+2*y)*cos(3*x+y)/(5*x-y)
```

У командному вікні відображається символьний вираз ***a***

```
a =
    (5*cos(3*x + y)*sin(x + 2*y))/(5*x - y)
```

Для відображення виразу у більш зручному вигляді використовується команда ***pretty***

```
>> pretty(a)

cos(3 x + y) sin(x + 2 y) 5
---------------------------
          5 x - y
```

Дії додавання, віднімання, множення, ділення та піднесення до ступеня для символьних змінних здійснюється за допомогою звичайних арифметичних знаків, що і для аналогічних операцій над дійсними числами. Наприклад,

```
syms x y z
a=5*x+3*y^2-4*(2.5*x-2*y*z)
pretty(a)
```

У командному вікні відображається результат

```
a =
 3*y^2 + 8*z*y - 5*x
    2
3 y  + z y 8 - 5 x
```

Функція, які здійснюють деякі алгебраїчні маніпуляції над символьними змінними:

- ***expand(f)*** – розкриває дужки у символьному виразі ***f***.

```
>> syms a b
>> c=(a+4*b)
c =
a + 4*b
>> d=(3*a-b)
d =
3*a - b
>> f=c*d
f =
(a + 4*b)*(3*a - b)
>> g=expand(f)
g =
3*a^2 + 11*a*b - 4*b^2
```

- ***factor(f)*** – розкладує на множники вираз ***f***.

```
>> syms a
> b=10*a^4+10*a^3+10*a+10;
>> c=factor(b)
c =
[ 10, a^2 - a + 1, a + 1, a + 1]
```

Для перевірки коректності розкладу на множники застосовується команда ***expand:***

```
>> d=10*(a^2-a+1)*(a+1)^2;
>> k=expand(d)
 k =
 10*a^4 + 10*a^3 + 10*a + 10
```

- ***simplify(f)*** – спрощує кожний елемент символьної матриці ***f***.

```
>> syms a
>> c=(15*a^3 - 23*a^2 - 46*a + 42)/(3*a-7);
>> c1=simplify(c)
c1 =
5*a^2 + 4*a - 6
```

Для перевірки застосовується функція розкладу на множники ***factor:***

```
>> factor(15*a^3 - 23*a^2 - 46*a + 42)
ans =
```

```
[ 3*a - 7, 5*a^2 + 4*a - 6]
```

Для побудови символьного рівняння застосовується функція ***ezplot***:

- ***ezplot(f)*** – будує графіки на інтервалі $[-2\pi, 2\pi]$;
- ***ezplot(f, xmin, xmax)*** – будує графіки на заданому інтервалі $[x_{min}, x_{max}]$.

Приклад. Побудувати графік функції

$$f(x) = 5x^3 + 8x^2 + 4x - 10$$

```
>> syms x
>> f=5*x^3+8*x^2+4*x-10;
>> ezplot(f), grid
```

На рис. 11.1 представлено графічне вікно з графіком функції $f(x) = 5x^3 + 8x^2 + 4x - 10$.

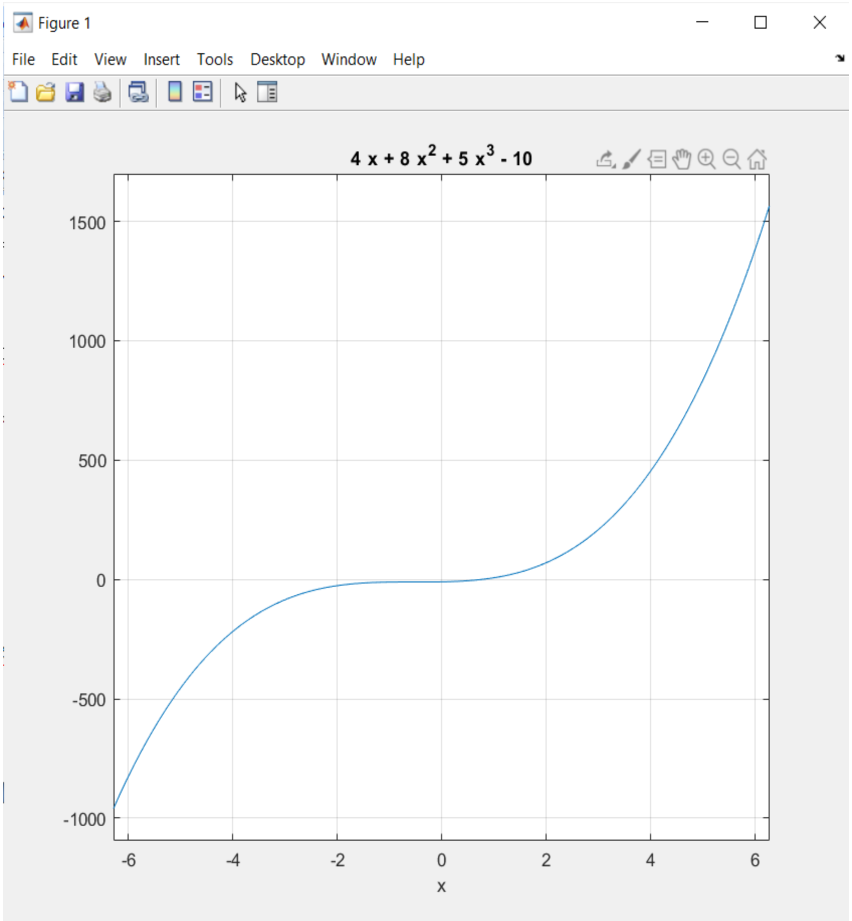


Рис. 11.1 Приклад побудови графіка з застосуванням функції ***ezplot***

У середовищі MatLab можна розв’язувати алгебраїчні рівняння з однією символьною змінною, з декількома змінними, а також системи алгебраїчних рівнянь. Для розв’язку рівнянь та систем рівнянь застосовується функція ***solve***.

Наприклад, для того, щоб розв’язати рівняння $x + 5 = 10$, необхідно:

```
>> syms x
>> solve(x+5==10)
ans =
5
```

Використання подвійного дорівнює «==» використовується для означення рівняння. Якщо не вказувати праву частину рівняння, тоді функція ***solve*** приймає, що права частина рівняння дорівнює нулю.

```
>> solve(x+5)
ans =
-5
```

Якщо розв'язок рівняння має декілька коренів, тоді

```
>> syms x
>> X=solve(25*x^2-10*x+1)
X =
1/5
1/5
```

Функцію ***solve*** також можна застосовувати для розв'язку системи рівнянь. Наприклад, розв'язати систему рівнянь:

$$\begin{cases} 4x_1 + 2x_2 + 2x_3 = 4 \\ 2x_1 - 2x_2 = -4 \\ 6x_1 - 2x_2 + 4x_3 = 4 \end{cases}$$

```
>> syms x1 x2 x3
>> [x1, x2, x3]=solve(4*x1+2*x2+2*x3==4, 2*x1-2*x2==-4,
6*x1-2*x2+4*x3==4)
x1 =
-1
x2 =
1
x3 =
3
```

Диференціювання символьного виразу здійснюється за допомогою функції ***diff :***

- ***diff(f)*** – диференціює символьний вираз ***f*** за незалежною змінною;
- ***diff(f,t)*** – диференціює символьний вираз ***f*** за змінною ***t***;
- ***diff(f,n)*** – ***n*** раз диференціює символьний вираз ***f;***
- ***diff(f,t,n)*** – ***n*** раз диференціює символьний вираз ***f*** за змінною ***t***.

Приклад. Визначити

- першу похідну символьного рівняння:

$$y = \cos(2x + 5).$$

```
>> syms x
>> f=diff(cos(2*x+5))
 f =
 -2*sin(2*x + 5)
```

- другу похідну символьного рівняння:

$$y = \cos(2x + 5).$$

```
>> syms x
>> f=diff(cos(2*x+5), 2)
 f =
 -4*cos(2*x + 5)
```

Для обчислення інтегралів (визначених та невизначених) у символьному вигляді в системі MatLab використовується функція ***int***. За замовчуванням інтегрування відбувається за змінною ***x***.

- ***int(f)*** – інтегрування символьного виразу ***f*** за незалежною змінною;
- ***int(f,t)*** – інтегрування символьного виразу ***f*** за змінною ***t***;
- ***int(f,a,b)*** – обчислює визначений інтеграл на інтервалі $[a, b]$***;***
- ***int(f,t,ab)*** – обчислює визначений інтеграл на інтервалі $[a, b]$ за вказаною змінною.

Приклад. Знайти невизначений інтеграл $\int -2x\sin(2x + 5)$ за допомогою вбудованої функції MatLab.

```
>> syms x
>> int(-2*sin(2*x + 5))
ans =
cos(2*x + 5)
```

Приклад. Знайти визначений інтеграл функції $f(x) = -2x\sin(2x + 5)$ на проміжку $\left[0, \frac{\pi}{2}\right]$

```
>> syms x
>> int(-2*sin(2*x + 5), 0, pi/2)
ans =
-2*cos(5).
```

11.2. Завдання до виконання комп'ютерного практикуму

Завдання 1. Представити розв'язок рівняння $f(x)\cdot g(x)$ (табл. 11.1) в символьному вигляді у зручному вигляді за допомогою функції ***pretty***.

Таблиця 11.1

Варіанти індивідуальних завдань

Варіант	$f(x)$	$g(x)$
1	2	3
1	$\dfrac{x^2}{2+0{,}5\sqrt{x}}$	$\sqrt{1+3x^2}\cdot\cos\dfrac{\pi x}{2}$
2	$\dfrac{x^3-0{,}8x}{1{,}5\sqrt{1+3x}}$	$\dfrac{x^2+2}{5+4\sqrt{x}}$
3	$\dfrac{2e^{-x}}{2\pi+x^2}$	$\sqrt{1+2x^2}\cdot\sin\dfrac{3x}{2}$
4	$\dfrac{\cos\pi x}{\sqrt{1-2x}}$	$\sqrt{3x^2+7}\cdot\cos\dfrac{\pi x}{2}$
5	$\sqrt{1+4x}\cdot\sin\pi x$	$\arccos e^{-\sqrt[3]{5x}}$
6	$\dfrac{e^{x/4}}{1+x^2}$	$\arcsin e^{-x^2/4}$
7	$e^{-3x}+x^2-4$	$\sqrt{5x^2+2}\cdot\cos\dfrac{\pi x}{3}$
8	$(1+x)\sin\left(\pi\sqrt{x-3}\right)$	$\dfrac{1+e^{-x/2}}{5x^2+1}$
9	$\sqrt{5+3x}\cdot\mathrm{tg}\dfrac{\pi x^3}{2}$	$3x^3+\dfrac{1}{x}+e^{-2x^2}$
10	$\sqrt{2+3x}\cdot\cos\dfrac{\pi x}{3}$	$x^{2x+1}+x^3-2x$
11	$\sqrt[3]{x^2+3}\cdot\cos\dfrac{\pi x}{2}$	$\dfrac{\cos\dfrac{\pi x}{4}}{\sqrt{2-5x}}$
12	$(3+5x)\cdot\sin\left(\pi^2\sqrt{2+x}\right)$	$\sqrt{1+4x}\cdot\sin\pi x$
13	$x^2+\left(y\quad-\sqrt[3]{x^2}\right)^2$	$\dfrac{x^2-5x+2}{\sqrt{1+5x}}$
14	$x^3-3x+\dfrac{5}{\sqrt{1+x^2}}$	$(x^3-3)^2+5x^2-10$
15	$(x^2-1)^3-2x^2$	$\dfrac{2x^2-7x+3}{(5x+3)}$
16	$(1+3x)^2\cdot\cos\left(\pi\sqrt{1+x^2}\right)$	$\sqrt{1+4x}\cdot\sin\dfrac{\pi x}{4}$
17	$\dfrac{2}{5}(1+3x)^3\cdot(3x+4x^2-10)$	$\sqrt{1+4x}\cdot(4x^2-10x+7)$

1	2	3
18	$\sqrt[3]{5x^3-2}\cdot\cos\frac{\pi x}{4}$	$\frac{\cos\frac{\pi x}{3}}{\sqrt{1-4x}}$
19	$x^4-4x+\frac{2}{\sqrt{5+x^3}}$	$(x^2+3)^2+7x^2-1$
20	$(1+2x)\cdot\sin\left(\sqrt{3+4x}\right)$	$\sqrt{x+6}\cdot\sin\frac{\pi x}{3}$
21	$5x^3+3x^2-5\sqrt{2+5x}$	$3x^5+2x^2+\frac{8}{5x+3}$
22	$(2x^2+1)^4+\frac{2x^2}{1+x}$	$\frac{3x^3+9x-10}{(2x+1)}$
23	$\frac{\cos\frac{\pi x}{3}}{\sqrt{4-7x}}$	$\sqrt[3]{5x^2-1}\cdot\cos\frac{\pi x}{4}$
24	$\sqrt{1+5x}\cdot\cos\frac{\pi x}{4}$	$\frac{1}{2}\arccos e^{-\sqrt[3]{4x+1}}$
25	$2x^3-5x^2+3\sqrt{4-8x}$	$3x^4+4x^2+\frac{1}{5x+3}$
26	$\sqrt[3]{(1+2x)}\cdot\sin\left(\frac{3}{5x+1}\right)$	$\frac{\sqrt{x+6}}{5x^3+3}$
27	$5x^3-3x+\frac{1}{\sqrt{4+x^3}}$	$(x^2+3)^3+5x^2-10$
28	$(5-3x)^3\cdot(4x+x^2+5)$	$\sqrt{2+5x}\cdot(6x^2-3x+5)$
29	$\sqrt{3+5x}\cdot\cos\frac{\pi x}{4}$	$x^{3x+2}+2x^5-x$
30	$(1+2x)^3\cdot\sin\left(\sqrt{2+5x}\right)$	$\frac{\sqrt{2x+5}}{5x+1}\cdot\sin\frac{\pi x}{4}$

Завдання 2. Побудувати в графічному вікні MatLab графік функції відповідно до варіанту (табл. 11.2) за допомогою функції ***ezplot***. Зробіть текстове оформлення графіка.

Таблиця 11.2

Варіанти індивідуальних завдань

Варіант	$f(x)$	**a**	**b**
1	2	3	4
1	$(3+2x)sin\left(\pi \quad \sqrt{1+x}\right)$	0,2	3,2

1	2	3	4
2	$\frac{e^{x/2}}{2+x^3}$ $arcsin\left(e^{-2x^2/3}\right)$	0,1	3,6
3	$\sqrt[4]{x^2+5}\cdot\sin\left(\frac{\pi}{4}x\right)$	1,5	5,6
4	$\frac{x^3+2{,}5x}{1+3{,}2e^x}$	0,2	3,2
5	$\frac{1+e^{-x/2}}{\sqrt{4{,}5x^2+1}}$	0,1	3,6
6	$\sqrt{4x+2{,}2}\cdot sin\left(\frac{\pi}{3}x\right)$	1,5	5,6
7	$\sqrt{2{,}5x^3+3{,}8}\cdot cos\left(\frac{\pi}{3}x\right)$	0,2	3,2
8	$\frac{x^2}{x+0{,}27\sqrt{x}}$	0,1	3,6
9	$\sqrt[3]{4{,}5x^2+3}\cdot cos\left(\frac{\pi}{4}x\right)$	1,5	5,6
10	$3x^3+\frac{3{,}4}{x}+e^{x^2}$	0,1	4,5
11	$\frac{x^4-5{,}8x}{\sqrt{1+3x}}$	0,2	3,2
12	$\left(4+5\sqrt{x}\right)\cdot\cos(\pi\sqrt{1+x})$	0,1	4,5
13	$x^3+3x+\frac{4}{\sqrt{1+x^2}}$	0,2	3,2
14	$\frac{x^3-1{,}3x}{\sqrt{1+2x}}$	0,1	4,5
15	$e^{2x^2}(5+3x-x^2)$	0,2	3,2
16	$x^{x+1}+2x^3-6x$	0,2	4,2
17	$\sqrt{1+5x}\cdot sin(0{,}2\pi x)$	0,1	3,6
18	$\frac{x^3-0{,}8x}{\sqrt{1+2{,}5x}}$	1,5	5,6
19	$\sqrt{5+2{,}5x}\cdot tg\left(\frac{\pi x}{4}\right)$	0,2	3,2
20	$x^{2x+1}+x^2+4x$	0,1	3,6

1	2	3	4
21	$\dfrac{x^2}{x-0{,}5\sqrt{x}}$	1,5	5,6
22	$\sqrt{4+3{,}5x}\cdot\sin\left(\dfrac{\pi}{3}x\right)$	0,2	3,2
23	$\dfrac{sin^2(0{,}3\pi x)}{\sqrt{1{,}2x-1}}$	0,1	3,6
24	$\sqrt{1+2{,}4x}\cdot sin(\pi x)$	1,5	5,6
25	$\dfrac{2x^3-1{,}5x}{\sqrt{1+2{,}3x}}$	0,1	3,6
26	$\dfrac{e^{x}/_2}{2+1{,}5x^3}$	1,5	5,6
27	$e^{2x^2}(5+3x-x^2)$	0,1	3,6
28	$x^{x+3}+1{,}5x^3-3x$	1,5	5,6
29	$\sqrt{1+1{,}2x}\cdot sin(\pi x)$	0,1	3,6
30	$\sqrt{1+3x^2}\cdot cos\left(\dfrac{\pi}{3}x\right)$	1,5	5,6

Завдання 3. Обчислити інтеграли за допомогою функції ***int*** відповідно до варіанту (табл. 11.3).

Таблиця 11.3

Варіанти індивідуальних завдань

Варіант	**Інтеграл 1**	**Інтеграл 2**
1	2	3
1	$\int arctg(\sqrt[3]{5x-2})dx$	$\int \dfrac{dx}{5\cos x}$
2	$\int \sqrt{4x^2+1}dx$	$\int \dfrac{(5x+1)dx}{\sin x\cos x}$
3	$\int \sqrt{6-x^2}dx$	$\int \dfrac{sin^2x}{cos^6x}dx$
4	$\int \sqrt{x^2+7x+5}dx$	$\int \dfrac{dx}{2sin^3x\cdot cos^3x}$
5	$\int e^{3x}\sin x\,dx$	$\int \dfrac{(3tgx+5)dx}{sin^2x+4cos^2x}$
6	$\int e^{5x}\cos 4x\,dx$	$\int \dfrac{dx}{sin^2x-sinx\cdot cosx+4cos^2x}$

1	2	3
7	$\int \frac{dx}{x\sqrt{x^2+2x+7}}$	$\int \frac{(5x+12)dx}{(x+5)(x^2+3x+4)}$
8	$\int \frac{dx}{(x-5)\sqrt{x^2+2x-5}}$	$\int \frac{(5x^2+4x+9)dx}{x^2+6x+10}$
9	$\int \frac{dx}{(x+1)\sqrt{x^2-5x+9}}$	$\int \frac{(5x^2-4x+8)dx}{(x+5)^2(x^2+16)}$
10	$\int \frac{dx}{(x+4)\sqrt{x^2-6x+16}}$	$\int \frac{(3x^3+2x^2-7x+9)dx}{(x+2)^2(4x^2+25)}$
11	$\int \frac{(3x^3-2x+9)dx}{(x^2+16x+25)}$	$\int \frac{(3x^2-8x+9)dx}{x^3+3x^2+7x+15}$
12	$\int \frac{dx}{(x+5)^3(2x^2+10x+25)}$	$\int \frac{(65x-87)dx}{(x+4)(x^2-5x+16)}$
13	$\int (5x+14)tg(4x+7)dx$	$\int \frac{(3\sin 2x+5)dx}{sin^4x+5cos^4x}$
14	$\int (x+1)ctg(2x+5)dx$	$\int \frac{(x^2-7x+6)dx}{(x-4)(x^2-5x+9)}$
15	$\int \frac{(x^2-4x+8)dx}{\sin(x+5)}$	$\int \frac{(3x^3-5)dx}{(x-1)(x^2-9)}$
16	$\int arctg(\sqrt[5]{7x-1})dx$	$\int \frac{dx}{(\mathrm{x}+15)\cos x}$
17	$\int (4x+6)\sqrt{15x^2+4}dx$	$\int \frac{(6x^2+7)dx}{3\sin \mathrm{x} \cos x}$
18	$\int (2x^2+3x+9)\sqrt{5-x^2}dx$	$\int \frac{sin^2(4x+7)}{cos^6x}dx$
19	$\int \sqrt{3x^2+5x-11}dx$	$\int \frac{(5x+3)dx}{2sin^3x\cdot cos^3x}$
20	$\int e^{5x}\sin(4x+5)\,dx$	$\int \frac{(7tgx+15)dx}{sin^2x+5cos^2x}$
21	$\int e^{(5x+1)}\cos 4x\,dx$	$\int \frac{dx}{sin^2x-sinx\cdot cosx+5cos^2x}$
22	$\int \frac{dx}{(x-1)\sqrt{x^2+5x+8}}$	$\int \frac{(2x+7)dx}{(x-4)(2x^2+4x+9)}$
23	$\int \frac{dx}{(x-4)\sqrt{4x^2+8x-25}}$	$\int \frac{(5x^3+7x+11)dx}{2x^2+9x+12}$

1	2	3
24	$\int \frac{dx}{(x+5)\sqrt{3x^2-8x+13}}$	$\int \frac{(7x^2-5x+12)dx}{(x-5)^2(x^2+9)}$
25	$\int \frac{dx}{(x+3)\sqrt{2x^2-5x+15}}$	$\int \frac{(x^3+2x^2-5x+12)dx}{(x-2)^2(x^2+25)}$
26	$\int \frac{(2x^3-3x+16)dx}{(x^2+4x+9)}$	$\int \frac{(2x^2-4x+12)dx}{2x^3+3x^2+5x+18}$
27	$\int \frac{(x+5)^3dx}{(8x^3+15x+125)}$	$\int \frac{(5x-8)dx}{(x-4)(x^2-10x+26)}$
28	$\int (x+14)^2 tg(5x+3)dx$	$\int \frac{(5\sin 3x+5)dx}{sin^3x+5cos^3x}$
29	$\int (x-1)ctg(5x+7)dx$	$\int \frac{(2x^2-5x+8)dx}{(x+4)(2x^2-6x+9)}$
30	$\int \frac{(x^2-3x+9)dx}{\sin(2x+5)}$	$\int \frac{(x^3-5)dx}{(x+1)(x^2+9)}$

Завдання 4. Обчислити вираз та систему рівнянь за допомогою функції ***solve*** відповідно до варіанту згідно з табл. 6.1 та 6.2.

11.3. Контрольні запитання

1. Що таке символьна змінна? Чим вона відрізняється від звичайної?
2. Як показати системі Matlab, що змінна *ZW* є символьною?
3. Як розкрити дужки у символьному виразі?
4. Чи можна розкласти вираз на множники? Як це зробити в системі MatLab?
5. Як спростити вираз в MatLab?
6. Для чого у символьному виразі використовується подвійне дорівнює «==» ?
7. Як побудувати графік для функції, заданої чисельним виразом? Як вказати межі змінювання аргументу?
8. Чи можна продиференціювати у символьному вигляді у MatLab? Яку функцію необхідно використовувати?
9. Як знайти похідну n-го порядку для символьної функції?
10. Як обчислити у символьному виразі інтеграл у MatLab?
11. На чому базується пакет символьної математики системи MatLab?

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 12
МОДЕЛЮВАННЯ ДИНАМІЧНИХ СИСТЕМ З ВИКОРИСТАННЯМ ПАКЕТА SIMULINK. СТВОРЕННЯ S-МОДЕЛЕЙ, ВИКОРИСТОВУЮЧИ РОЗДІЛИ SOURCES І SINK

Мета: ознайомитися з бібліотекою Simulink та набути навичок створення S-моделей для генерації вхідних сигналів, використовуючи розділ Sources і Sink.

12.1. Теоретичні відомості

Для запуску Simulink необхідно на панелі інструмента командного вікна системи MATLAB натиснути на кнопку 

(рис.12.1) та на екрані з'явиться вікно (рис. 12.2).

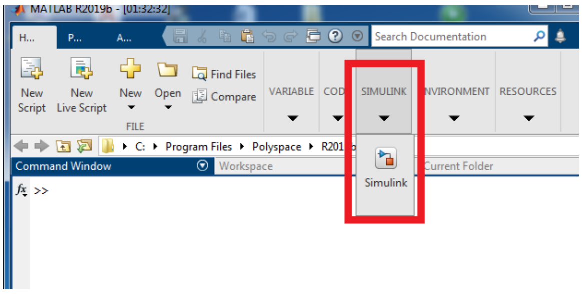


Рис. 12.1. Запуск Simulink

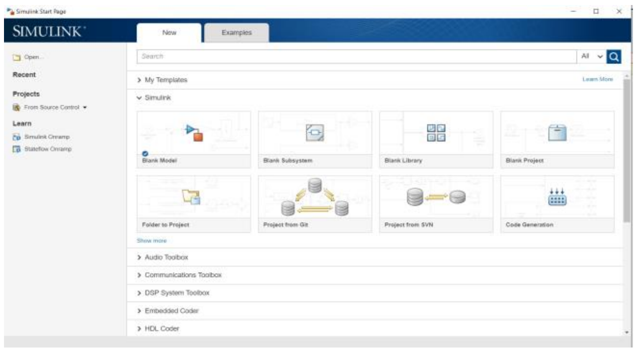


Рис. 12.2. Стартове вікно Simulink

Для створення нової Simulink-моделі у стартовому вікні необхідно натиснути на бланк порожньої моделі Blank Model (рис. 12.3).

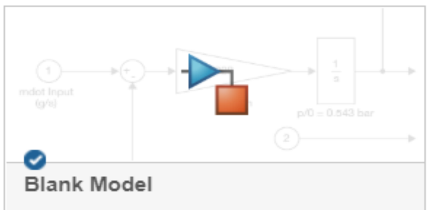


Рис. 12.3. Створення нової порожньої Simulink-моделі

Після чого відкривається редактор Simulink (рис. 12.4).

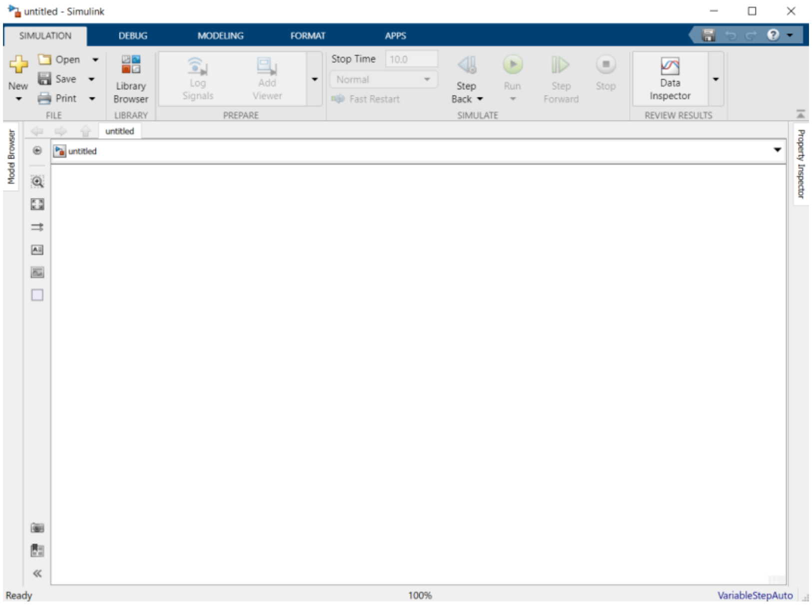


Рис. 12.4. Вікно редактора Simulink

Для збереження нової S-моделі на панелі інструментів необхідно натиснути на кнопку 

 та обрати Save as… (рис. 12.5).

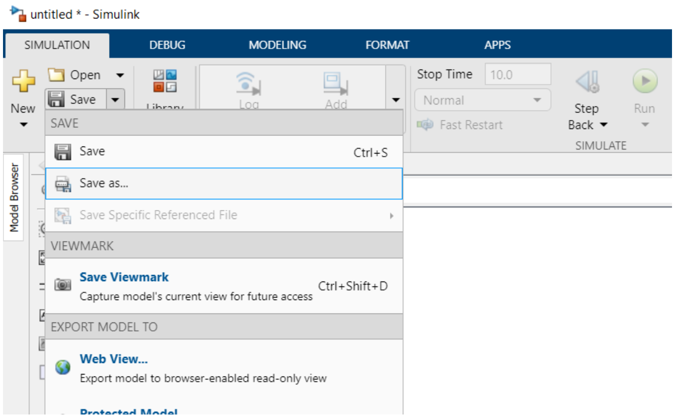
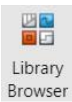


Рис. 12.5. Вікно збереження S-моделі

У полі File name написати назву моделі та натиснути кнопку Save (наприклад, назва моделі *model_1*). Файл зі збереженою моделлю має розширення ***.slx***.

Браузер бібліотеки Simulink відкривається натисненням на піктограму 

, після чого на екрані комп'ютера з'явиться вікно перегляду бібліотек – Simulink Library Browser (рис. 12.6).

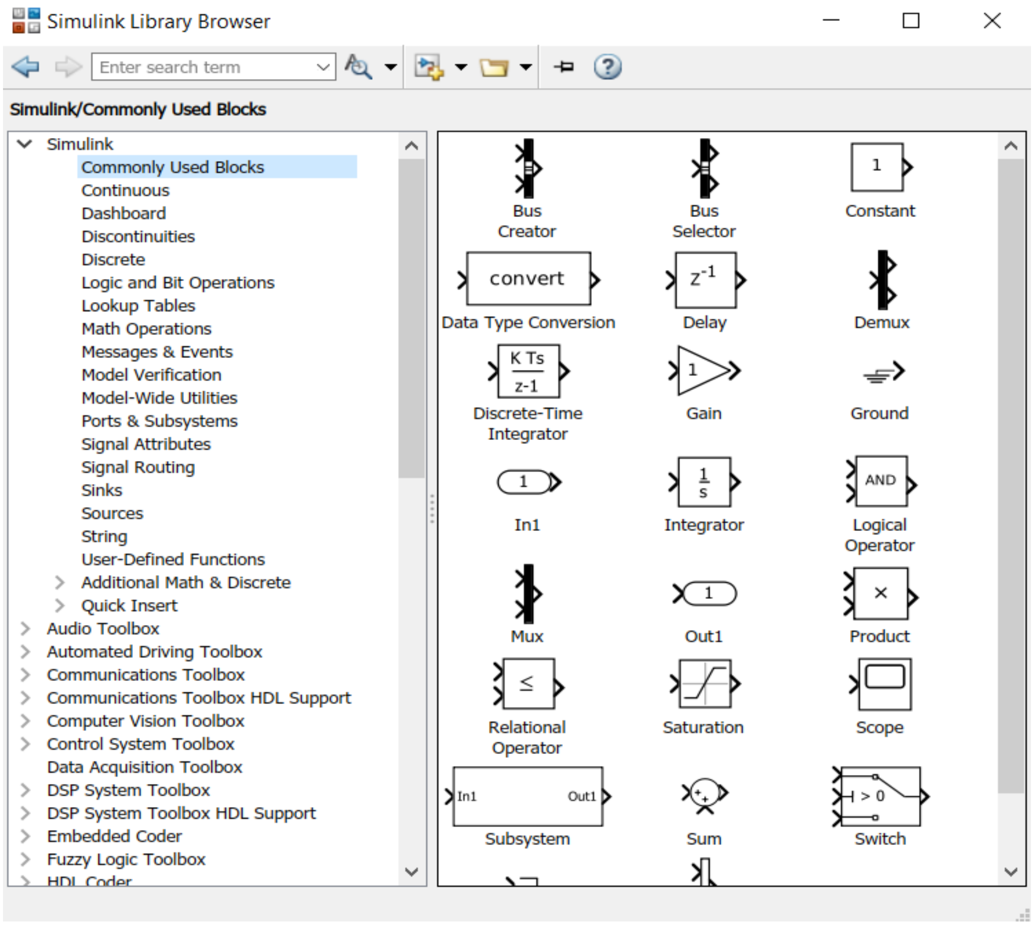


Рис. 12.6. Вікно перегляду бібліотеки Simulink

Бібліотека елементів організована у блоки за їх функціональністю у браузері бібліотеки. Бібліотека Simulink системи MATLAB складається з наступних розділів, які є загальними для більшості робочих процесів:

12.1.1. Розділ Sources (джерела)

Блоки, що входять в розділ Sources (джерела сигналів та впливів), призначені для формування сигналів, які при моделюванні забезпечують роботу S-моделі в цілому або окремих її частин. Всі блоки-джерела мають по одному виходу і не мають входів. На рис. 12.7 представлено вікно розділу Sources бібліотеки з джерелами сигналів.

Для відображення стислого опису блоків та переліку його параметрів необхідно навести мишку на блок, який цікавить, та натиснути правою кнопкою на даний блок, а потім обрати *Block parameters* – в результаті відобразиться опис блоку. У даному випадку параметри блока доступні тільки для читання. Приклад відображення параметрів блока Constant представлений на рис. 12.8.

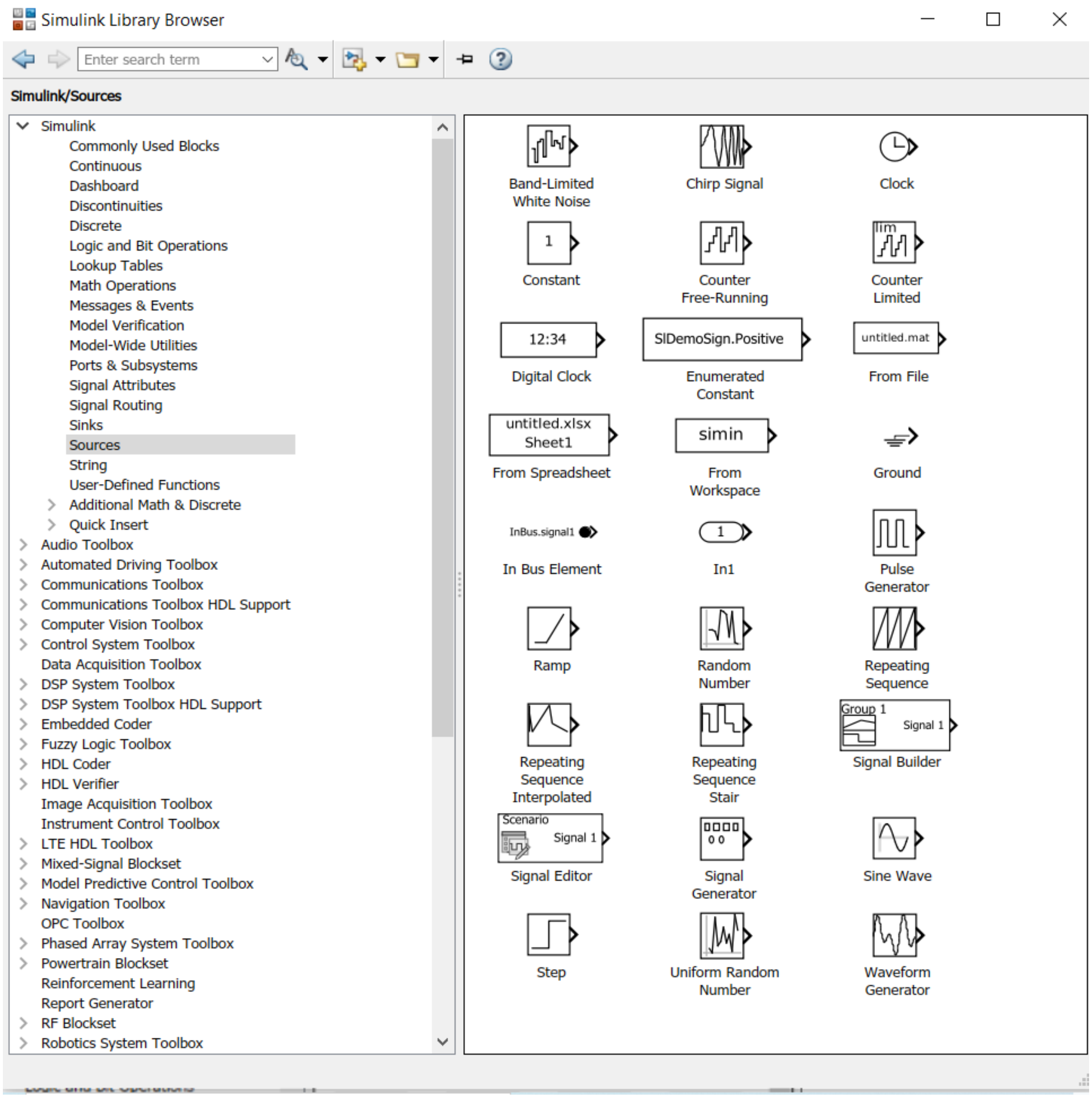
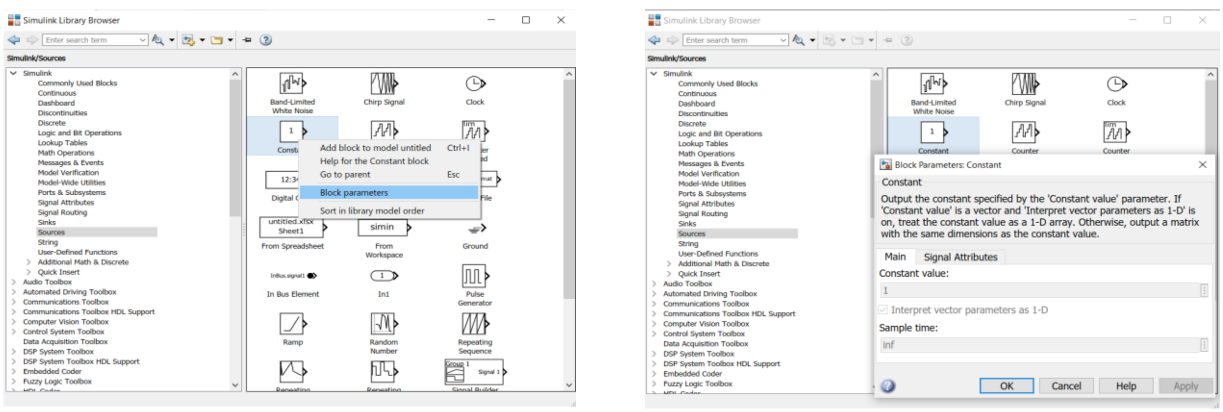


Рис. 12.7. Розділу Sources

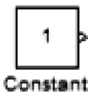


а) б)

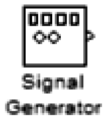
Рис. 12.8. Приклад відображення параметрів блоку Constant розділу Sources: а) вибір Block parameters; б) вікно параметрів блоку Constant

Для того, щоб почати використовувати блоки бібліотеки Simulink їх просто необхідно перетягнути з бібліотеки у робоче поле редактора Simulink вашої моделі. Потік сигналів між блоками визначається стрілочками, що з’єднують необхідні входи і виходи блоків.

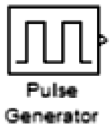
Піктограми блоків, що входять до розділу Sources бібліотеки Simulink:



– формує на виході сигнал постійної величини. На рис. 12.9 представлено застосування цього джерела і контроль рівня його впливу за допомогою осцилографа і цифрового індикатора-дисплея.

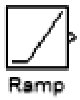


– створює неперервний коливальний сигнал однієї з хвильових форм (синусоїдальної, прямокутної, трикутної) або випадковий сигнал.



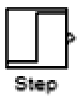
– генератор нескінченних прямокутних імпульсів (рис.12.10).

– створює один або декілька процесів, що апроксимується відрізками прямих (до п'яти відрізків у кожному).



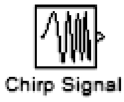
– створює лінійний висхідний (або спадний) сигнал. На рис. 12.11 представлено джерело лінійно наростаючого впливу виду $F(t) = k * t + i$ та вікно його параметрів. До параметрів цього джерела відносяться: *Slope* – кутовий коефіцієнт нахилу часової залежності k; *Start time* – час, починаючи з якого вплив наростає; *Initial value* – початковий рівень сигналу.

– генерує гармонійний сигнал. Він характеризується амплітудою *Amplitude*, зміщенням по вертикалі *Bias*, частотою *Frequence*, фазою *Phase* і еталонним часом *Sample time*. Останнє використовується для узгодження роботи джерела. На рис. 12.12 представлено застосування цього джерела і контроль рівня його впливу за допомогою осцилографа.



– генерує сигнал у вигляді сходинки (ступінчастий сигнал) із заданими параметрами (час початку сходинки і її висота). На рис.12.13. представлено джерело впливу у вигляді одиночної сходинки. До параметрів джерела відносяться: *Step time* – час появи перепаду (стрибка); *Initial value* – початкове значення впливу (до перепаду); *Final value* – кінцеве значення впливу (після перепаду); *Sample time* – еталонний час.

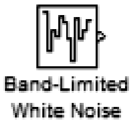
– генерує періодичну послідовність.



– генератор гармонійних коливань, частота яких лінійно змінюється у часі.

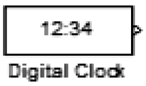
– джерело дискретного сигналу, значення якого є випадковою величиною, розподіленою за нормальним законом.

– джерело дискретного сигналу, значення якого є випадковою рівномірно розподіленою величиною.

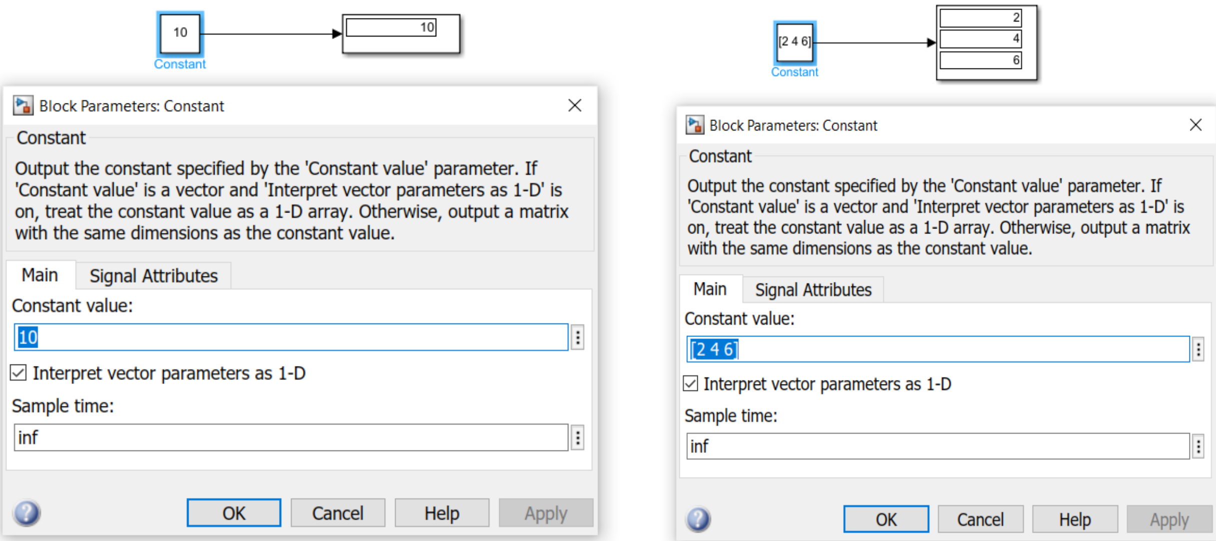


– генератор білого шуму з обмеженою смугою частот.

– джерело неперервного сигналу, пропорційного модельному часу (рис.12.14).



– джерело дискретного сигналу, пропорційного часу.



а) б)

Рис.12.9 Джерело постійного впливу з:

а) одним значенням; б) джерелом постійного впливу з багатьма значеннями

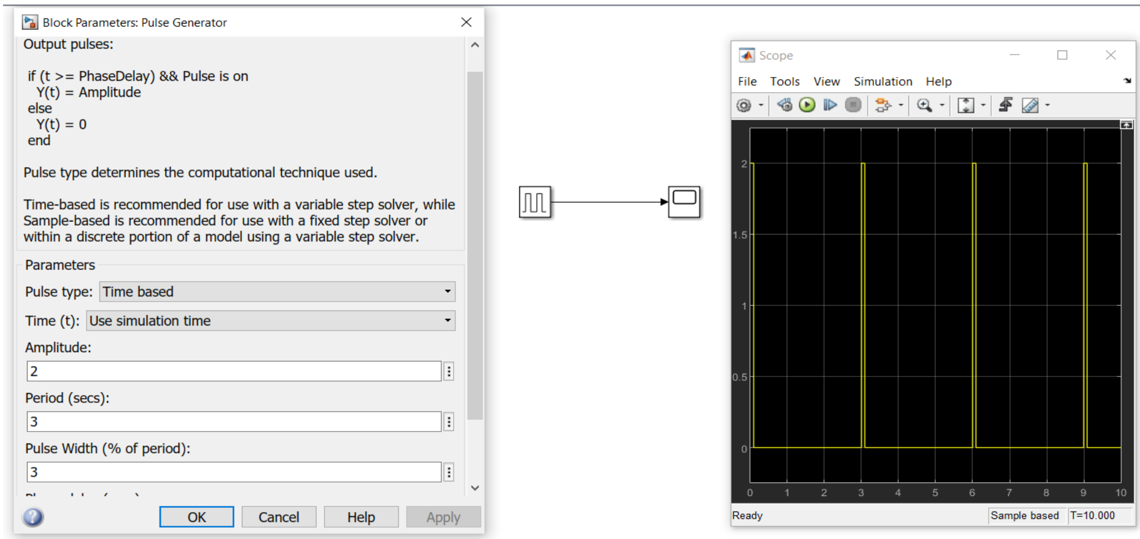


Рис. 12.10 Джерело прямокутних імпульсів

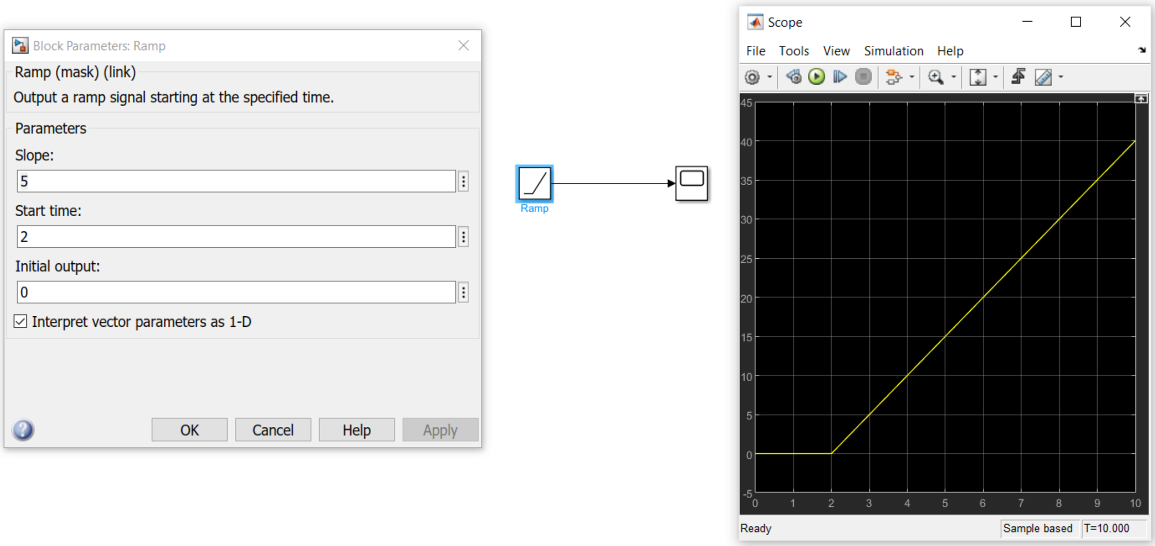


Рис.12.11 Джерело лінійно-наростаючого впливу

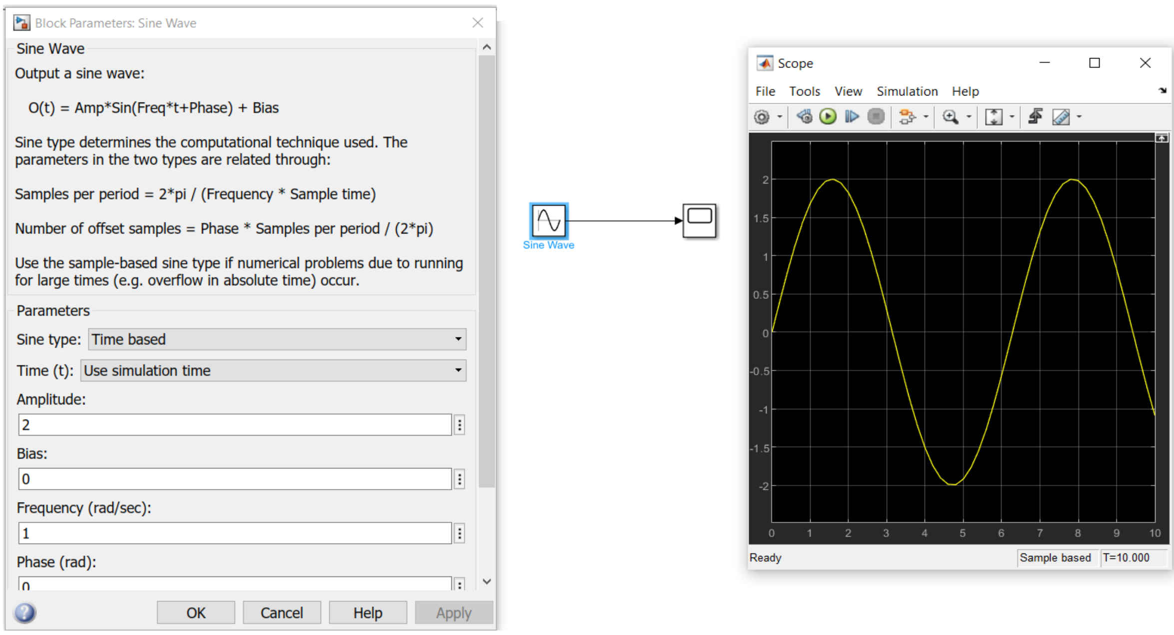


Рис. 12.12 Джерело синусоїдального впливу Sine Wave

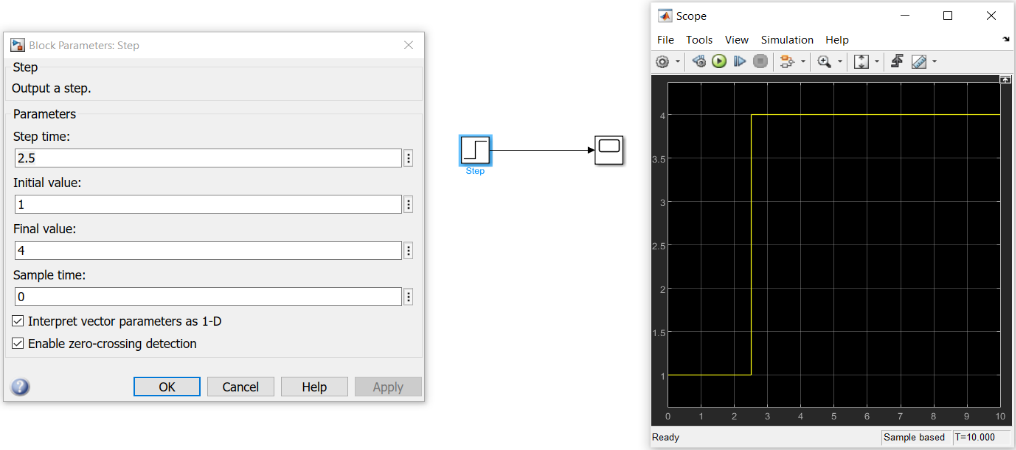


Рис. 12.13 Джерело ступінчастого сигналу (одиночного перепаду)

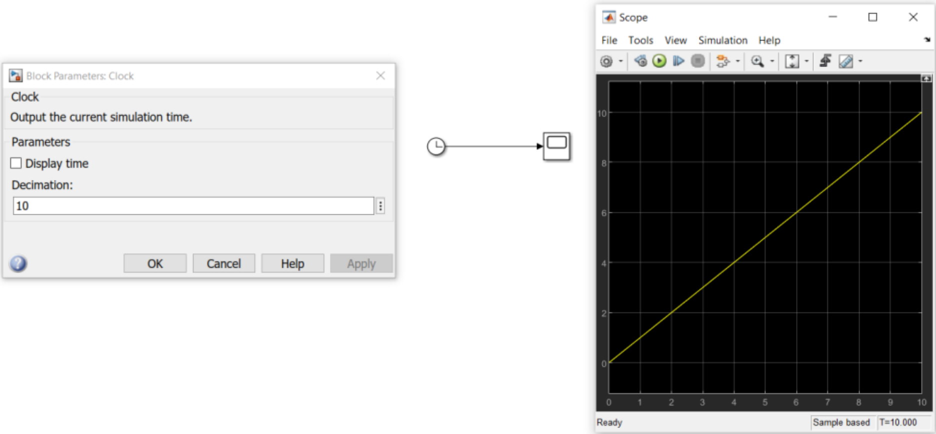


Рис.12.14 Джерело безперервного сигналу, пропорційного модельному часу

12.1.2. Розділ Sinks (приймачі)

Блоки, що входять в розділ Sinks, призначені для приймання сигналів. Всі блоки-приймачі мають входи і не мають виходів. На рис. 12.15 представлено вікно розділу бібліотеки з приймачами Sinks.

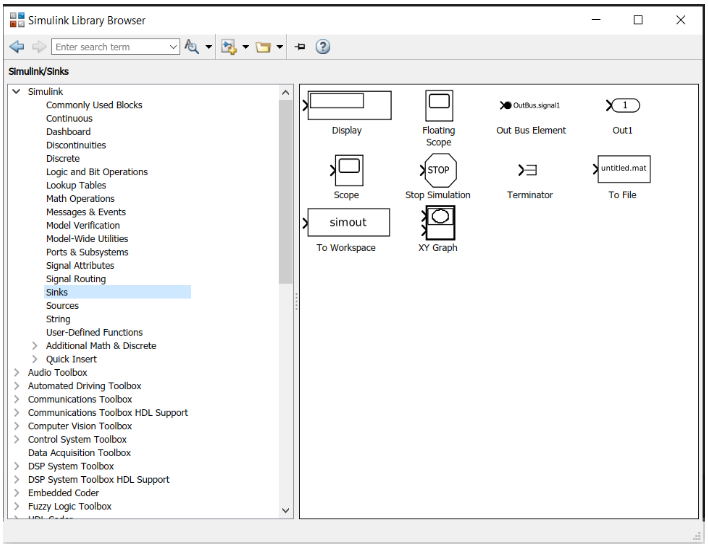


Рис.12.15. Розділ Sinks

12.2. Завдання на виконання комп’ютерного практикуму

Завдання 1. Створити S-моделі, використовуючи різні джерела, налаштувати параметри джерел (відповідно до варіанту) і зареєструвати сигнали на виходах моделей за допомогою осцилографів (Scope).

1.1. джерело постійного сигналу Constant (значення постійної дорівнює номеру варіанту);

1.2. джерело синусоїдального сигналу Sine Wave (табл. 12.1);

1.3. джерело наростаючого сигналу Ramp (табл. 12.2);

1.4. джерело одиночного перепаду Step (табл. 12.3);

1.5. джерело прямокутних імпульсів Pulse Generator (табл. 12.4).

Завдання 2. Змінити варіант та побудувати нові графіки для кожного типу **джерела.**

Завдання 3. Порівняти отримані графіки та зробити висновки.

Таблиця 12.1

Варіанти індивідуальних завдань

Варіант	**Амплітуда**	**Частота**	**Фаза**
1	2	3	4
1	0,4	0,4	20
2	0,5	0,2	10
3	0,2	1	15
4	1,5	0,5	100
5	2	0,35	40
6	2,5	1,2	50
7	1,2	1,4	30
8	0,5	1,6	20
9	1,4	0,8	10
10	1,6	0,4	1
11	0,8	0,6	20
12	0,5	1,2	50
13	0,7	1,1	40
14	1	1,8	100
15	1,2	2	30
16	1,5	1,2	40
17	1,7	0,5	20
18	0,7	0,6	10

1	2	3	4
19	1	1,8	5
20	2	0,4	15
21	1,5	0,5	20
22	1,8	0,2	15
23	1,4	1,2	30
24	0,8	0,5	40
25	1,5	1,4	20
26	2,5	0,2	10
27	1,2	1,2	15
28	2,5	1,5	10
29	3,5	0,5	40
30	4,5	1	50

Таблиця 12.2

Варіанти індивідуальних завдань

Варіант	**Нахил**	**Початковий час**	**Початкове значення**
1	2	3	4
1	0,5	0,4	20
2	0,8	1,0	25
3	1,2	1,3	10
4	1,5	1,0	15
5	1,8	1,35	10
6	2,0	1,25	10
7	2,2	0,48	25
8	2,0	2,4	20
9	1,8	1,6	35
10	1,6	1,4	35
11	1,5	1,6	25
12	1,9	1,2	25
13	2,1	1,1	30
14	3,0	1,8	30
15	3,2	0,5	30
16	1,4	1,2	30
17	1,5	0,5	25
18	2,0	0,6	25

1	2	3	4
19	2,2	1,8	25
20	0,8	0,4	25
21	0,6	0,5	30
22	0,4	0,2	20
23	0,5	1,2	20
24	1,0	0,5	20
25	1,5	2,4	10
26	2,8	2,0	15
27	2,2	0,3	20
28	3,0	1,0	25
29	2,0	1,5	30
30	2,5	1,5	35

Таблиця 12.3

Варіанти індивідуальних завдань

Варіант	Step time	Initial value	Final value
1	2	3	4
1	1,5	0,4	20
2	1,5	1,0	25
1	2	3	4
3	1,5	1,3	10
4	2,0	1,0	15
5	2,0	1,35	10
6	2,0	1,25	10
7	2,5	0,48	25
8	2,5	2,4	20
9	2,5	1,6	35
10	3,0	1,4	35
11	3,0	1,6	25
12	3,0	1,2	25
13	3,5	1,1	30
14	3,5	1,8	30
15	3,5	0,5	30
16	4,0	1,2	30
17	4,0	0,5	25

1	2	3	4
18	4,0	0,6	25
19	5,0	1,8	25
20	5,0	0,4	25
21	5,0	0,5	30
22	5,5	0,2	20
23	5,5	1,2	20
24	5,5	0,5	20
25	6,0	1,4	20
26	6,0	1,6	25
27	6,0	1,2	5
28	6,5	1,1	10
29	6,5	1,8	15
30	6,5	1,4	20

Таблиця 12.4

Варіанти індивідуальних завдань

Варіант	**Амплітуда**	**Період**	**Фаза**
1	2	3	4
1	0,5	1,2	0
2	0,5	1,4	0,5
3	0,5	1,6	1,0
4	1,0	2,2	0
5	1,0	2,4	1,5
6	1,0	2,6	2,5
7	1,5	1,5	1,5
8	1,5	2,5	2,0
9	2,5	3,5	2,5
10	1,5	3,6	2,2
11	2,0	0,5	1,0
12	2,0	1,5	1,0
13	2,5	0,5	1,0
14	2,5	1,5	1,0
15	2,5	3,5	1,5
16	3,0	1,2	0,5
17	3,0	1,4	1,5

1	2	3	4
18	3,0	1,6	0,5
19	3,5	2,2	1,5
20	3,5	2,4	2,5
21	3,5	2,6	3,5
22	4,0	4,2	2,5
23	4,0	4,4	3,5
24	4,0	4,6	4,5
25	4,5	2,5	0,5
26	4,5	3,5	1,5
27	4,5	4,5	2,5
28	5,0	1,2	0,5
29	5,0	1,6	1,5
30	5,0	1,8	2,5

Контрольні запитання

1. Як запустити пакет Simulink?
2. З яких розділів складається бібліотека Simulink?
3. Які блоки містяться у розділі Sources?
4. Які параметри джерела постійного впливу Constant?
5. Які параметри джерела синусоїдального впливу Sine Wave?
6. Які параметри джерела наростаючого впливу Ramp?
7. Які параметри джерела одиночного перепаду Step?
8. Які параметри джерела прямокутних імпульсів Pulse Generator?

КОМП’ЮТЕРНИЙ ПРАКТИКУМ № 13
СТВОРЕННЯ S-МОДЕЛІ, ВИКОРИСТОВУЮЧИ РОЗДІЛ MATH OPERATIONS

Мета: ознайомитися з розділом Math Operation бібліотеки Simulink та набути навичок створення S-моделей, використовуючи розділ Math Operations.

13.1. Теоретичні відомості

На рис. 13.1 представлено вікно розділу бібліотеки Math Operations з неперервними компонентами.

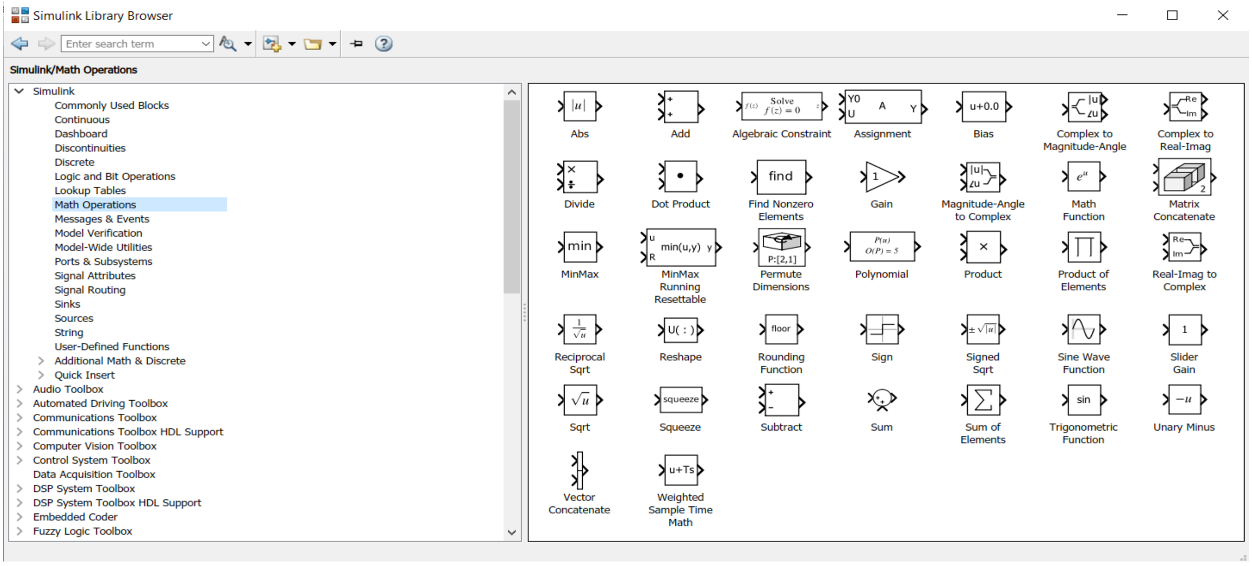
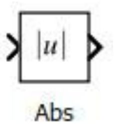


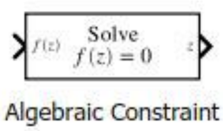
Рис. 13.1 Розділ бібліотеки Math Operations

Зображення блоків, що входять до розділу Math Operations бібліотеки Simulink:

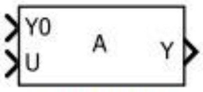


- обчислює абсолютне значення числа (модуль);

- обчислює суму або різницю вхідних сигналів;

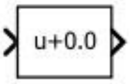


- обмежує вхідний сигнал;



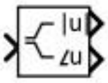
Assignment

- призначає значення заданим елементам сигналу;



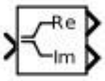
Bias

- додає зміщення до сигналу на постійне значення, тобто зміщує сигнал верх чи вниз по ординаті на певну сталу величину;



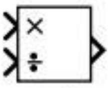
Complex to Magnitude-Angle

- обчислює модуль і (або) аргумент комплексного числа;



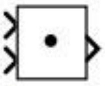
Complex to Real-Imag

- обчислює дійсну і (або) уявну частину комплексного числа;

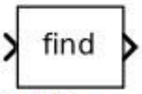


Divide

- видає результат ділення верхнього значення (підключається до входу «×») на нижнє (підключається до входу «÷»);

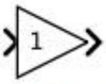

Dot Product

- обчислення скалярного добутку двох векторів;



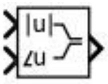
Find Nonzero Elements

- знаходить усі ненульові елементи вхідного сигналу та повертає лінійні індекси (номери по порядку вхідного одновимірного масиву зі значень сигналу) цих елементів;



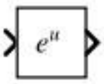
Gain

- виконує множення вхідного сигналу на постійний коефіцієнт;



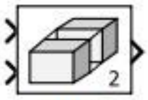
Magnitude-Angle to Complex

- перетворює модуль і (або) аргумент в комплексне число;



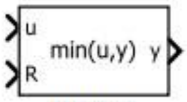
Math Function

- виконує обчислення математичної функції, яку можна обрати в налаштуваннях цього блоку;



Matrix Concatenate

- приєднує до однієї матриці одну або декілька матриць для створення нової матриці;

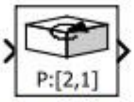


MinMax Running Resettable

- визначає мінімальне або максимальне значення сигналу протягом певного проміжку часу;

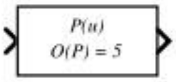
MinMax

- визначає максимальне або мінімальне значення з усіх сигналів, що надходять на його входи;



Permute Dimensions

- переставляє елементи багатовимірного вхідного сигналу, міняючи місцями виміри сигналу, наприклад, обмінюючи його перший і третій виміри;

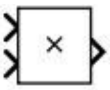


Polynomial

- виконує оцінку коефіцієнтів апроксимуючого поліному визначеного порядку на основі вхідних значень;

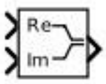
Product of Elements

- знаходить добуток всіх значень вхідного сигналу;



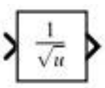
Product

- обчислює звичайний добуток поточних значень сигналів, що надходять на входи блоку. Блок Product призначений не тільки для множення, а й ділення;



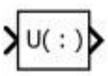
Real-Imag to Complex

- обчислює комплексне число за його дійсною та уявною частинами;



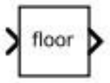
Reciprocal Sqrt

- обчислює значення, зворотне до квадратного кореня;



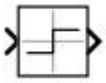
Reshape

- змінює розмірності сигналу;



Rounding Function

- округлення – дозволяє у вікні установки параметрів вибрати зі списку одну з чотирьох функцій округлення: *floor* – до найближчого меншого цілого; *ceil* – до найближчого більшого цілого; *round* – до найближчого цілого за звичайними математичними правилами; *fix* – до цілого, отриманого відкиданням цілої частини;


Sign

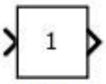
- визначає знак числа;


Signed Sqrt

- квадратний корінь від модуля вхідного сигналу, помножений на знак вхідного сигналу;


Sine Wave Function

- генерує синусоїдальний сигнал, використовуючи зовнішній сигнал як джерело часу;

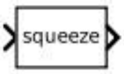


Slider Gain

- забезпечує зміну коефіцієнта підсилення в процесі моделювання;

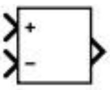

Sqrt

- обчислює квадратний корінь;



Squeeze

- видаляє одновимірні розмірі з багатовимірного сигналу;


Subtract

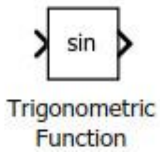
- обчислює різницю, виконує віднімання від значення верхнього входу «+ ») значення, що приходить на нижній вхід «- »;


Sum of Elements

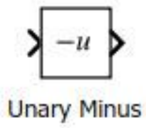
- обчислення суми або різниці елементів;


Sum

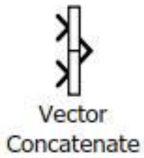
- обчислює суму або різницю поточних значень сигналів, що надходять на відповідні входи блоку;



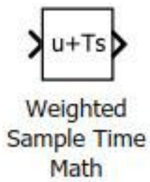
- виконує обчислення тригонометричної функції;



- встановлює негативне значення сигналу;



- об'єднує вхідні векторні сигнали одного типу даних, щоб створити багатовимірний вихідний сигнал (можна представити як матрицю);



- виконує додавання, віднімання, множення або ділення вхідного сигналу U на зважений час вибірки T_s

Для додавання та віднімання скалярних, векторних або матричних вхідних значень застосуються блоки Add, Subtract, Sum of Elements та Sum. Дані блоки є ідентичними.

На рис. 13.2 представлені приклади застосування блоку Add. У блоці параметрів можна змінювати знаки входів блоку, використовуючи поле List of signs. Наприклад,

- при введенні знаків «++» обчислюється вираз $5 + 2$;
- при введенні знаків «+-» обчислюється вираз $5 - 2$;
- при введенні знаків «-+» обчислюється вираз $-5 + 2$.

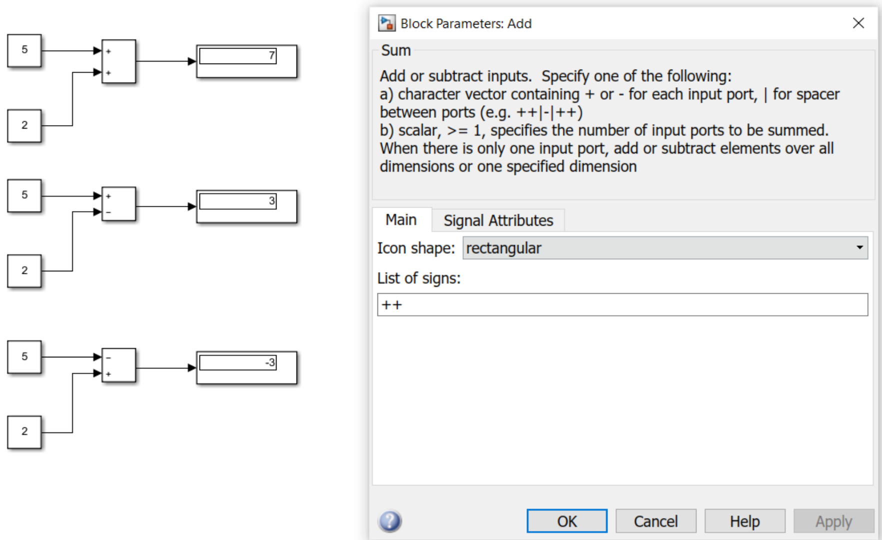


Рис. 13.2 Приклади застосування блоку Add

На рис. 13.3 представлені приклади застосування блоку Sum.

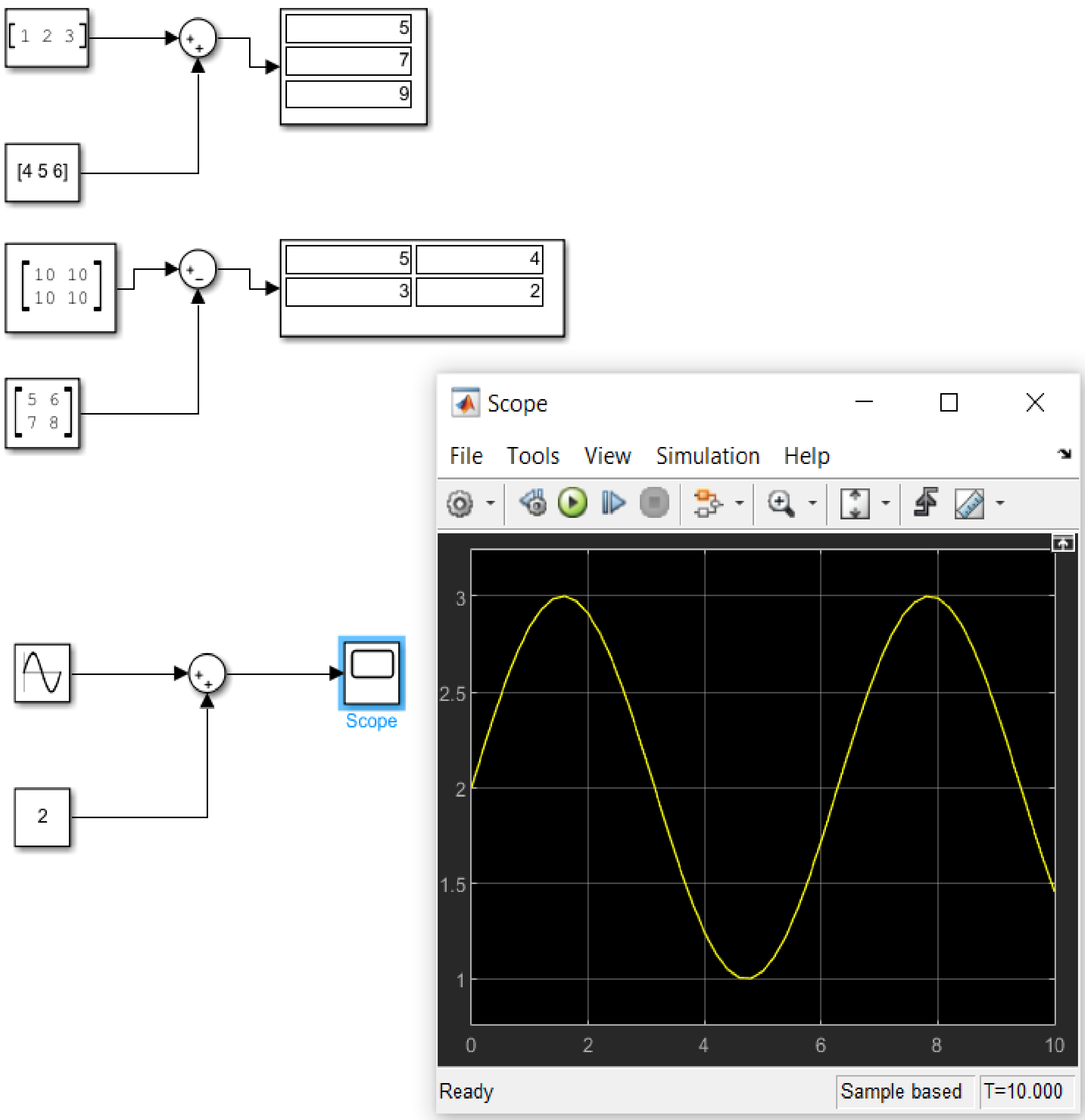


Рис. 13.3 Приклад застосування блоку Sum

Обчислити добуток або різницю значень можна, застосувавши блок Product. Множення можна здійснювати для скалярних, векторних та матричних вхідних значень. У параметрах блока можна змінити кількість входів (в полі Number of inputs). Наприклад, якщо вказано:

- «2», тоді виконується дія множення двох вхідних значень, тобто як представлено в прикладі (рис. 13.4) отримується вираз $5 \cdot 3$;
- «*/», тоді виконується ділення першого вхідного значення на друге - $5 \div 3$;
- «3», тоді виконується дія множення трьох вхідних значень.

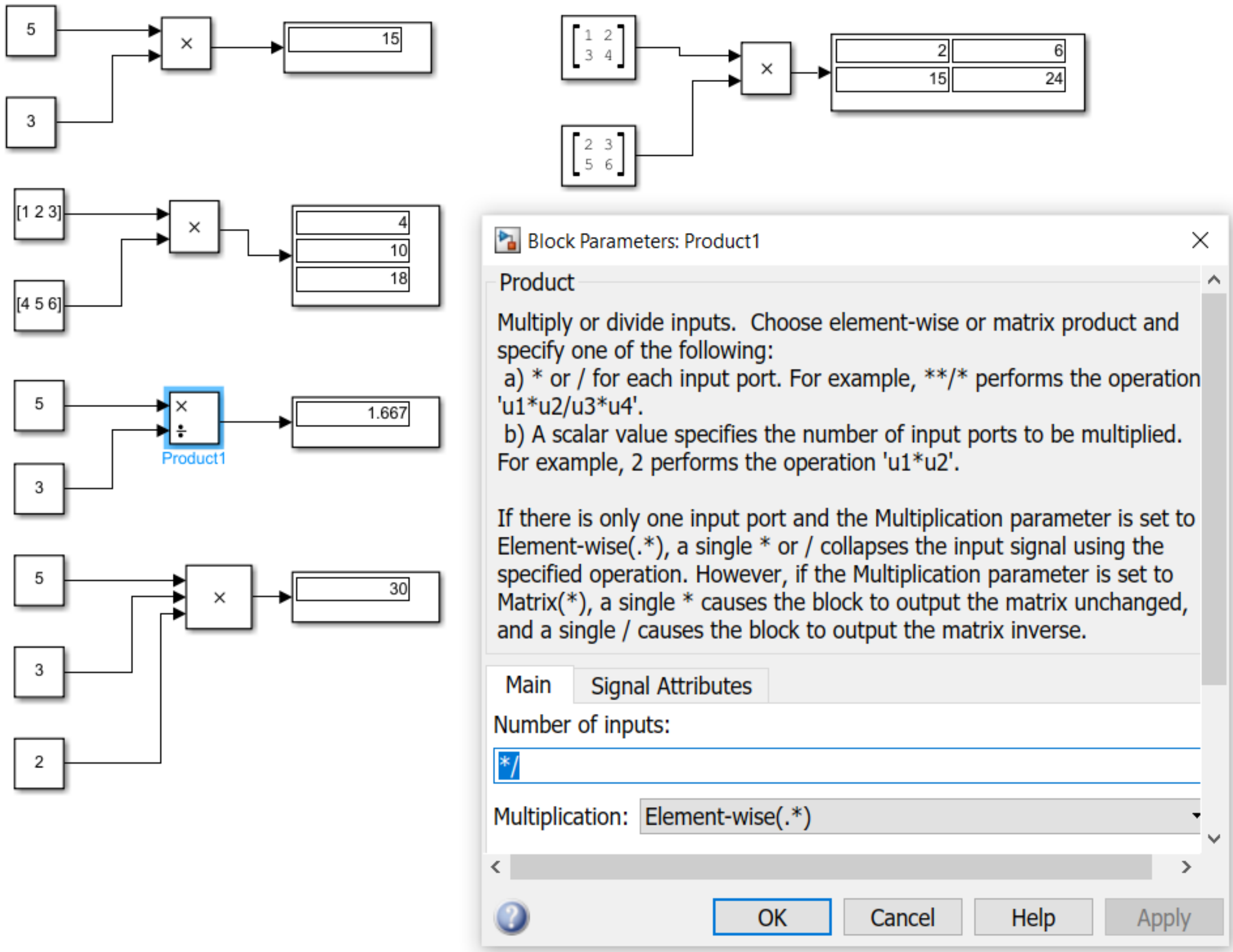


Рис. 13.4 Приклад застосування блоку Product

У розділі Math Operations розміщений блок Divide. Даний блок виконує ділення першого значення входу на другим. Блок може застосовуватись при діленні скалярних, векторних або матричних вхідних значень.

На рис. 13.5 показано приклад застосування блоку Abs та налаштування параметрів, який дозволяє виводити абсолютне значення входу, для вхідного сигналу у вигляді синусоїдального імпульсу.

Блок Gain застосовується для множення значень сигналу на вході блоку на постійне значення (підсилення). Вхідні значення і підсилення можуть бути скалярними, векторними або матричними. Приклади застосування блоку Gain представлені на рис. 13.6.

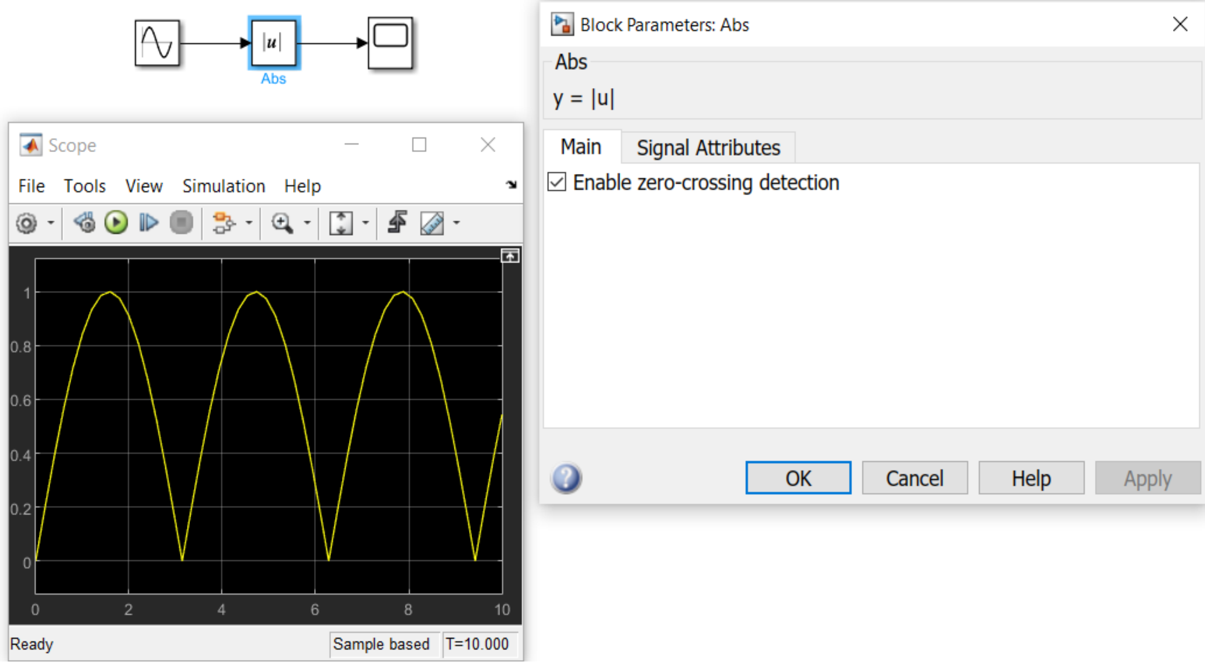


Рис. 13.5 Приклад застосування блоку Abs

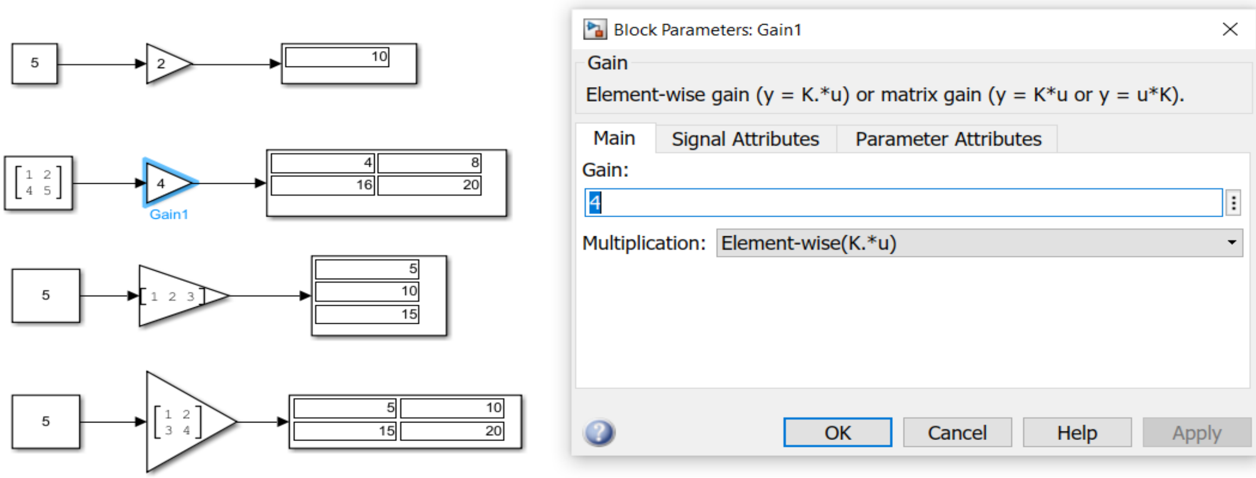


Рис. 13.5 Приклади застосування блоку Gain

Блок Math Function виконує обчислення математичної функції. До параметрів даного блоку відносяться:

1. ***Function*** – Вид функції, що обчислюється (вибирається зі списку):

- **exp** – експоненціальна функція;
- **log** –натуральний логарифм;
- **10 ^ u** – обчислення ступеня 10;
- **log10** – десятковий логарифм;
- **magnitude ^ 2** – обчислення квадрата модуля вхідного сигналу;
- **square** – обчислення квадрата вхідного сигналу;
- **sqrt** – квадратний корінь;

- **pow** – піднесення до степеня;
- **conj** – обчислення комплексно-спряженого числа;
- **reciprocal** – обчислення частки від розподілу вхідного сигналу на 1;
- **hypot** – обчислення кореня квадратного з суми квадратів вхідних сигналів (гіпотенузи прямокутного трикутника за значеннями катетів);
- **rem** – функція, що обчислює залишок від ділення першого вхідного сигналу на другий;
- **mod** – функція, що обчислює залишок від ділення з урахуванням знака;
- **transpose** – транспонування матриці;
- **hermitian** – обчислення ермітової матриці.

2. ***Output signal type*** – тип вихідного сигналу (вибирається зі списку): **auto** – автоматичне визначення типу; **real** – дійсний сигнал; **complex** – комплексний сигнал (тобто сигнал. що складається з комплексних чисел).

На рис. 13.7 представленні приклади застосування блоку Math Function з вибором різних видів функцій, які обчислює цей блок.

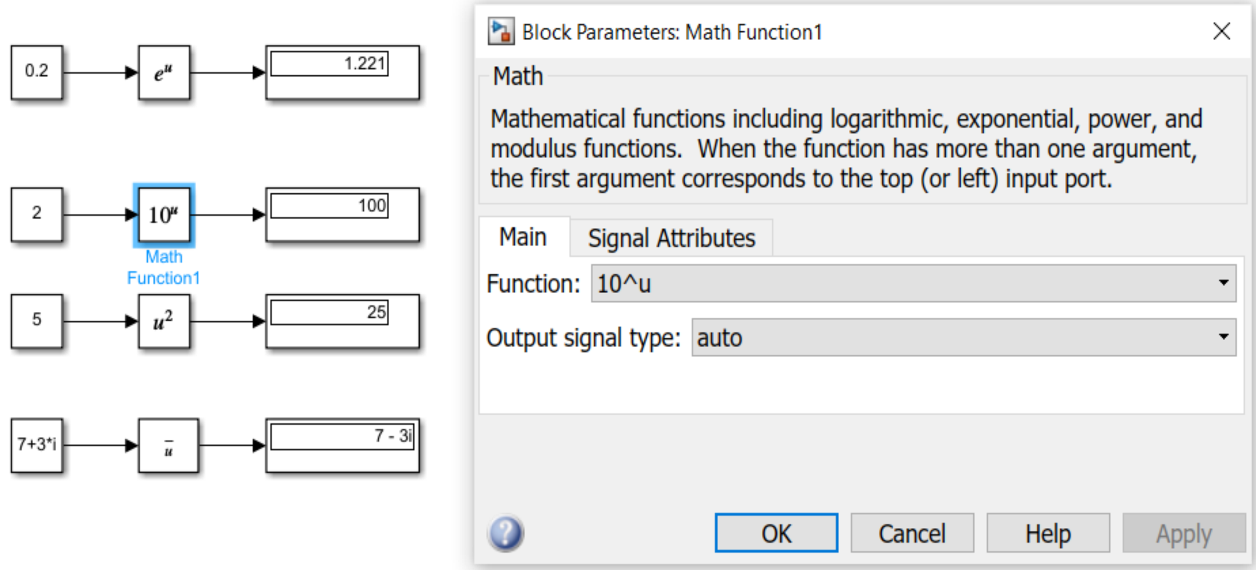


Рис. 13.7 Приклади застосування блоку Math Function

Блок Trigonometric Function виконує обчислення тригонометричної функції. До параметрів блоку відносяться налаштування:

1. ***Function*** – вид обчислюваної функції (вибирається зі списку): **sin**, **cos**, **tan**, **asin**, **acos**, **atan**, **atan2**, **sinh**, **cosh** і **tanh**.

2. ***Output signal type*** – тип вихідного сигналу (вибирається зі списку): **Auto** – автоматичне визначення типу, **Real** – дійсний сигнал; **Complex** – комплексний сигнал.

На рис. 13.8 представленні приклади застосування блоку Trigonometric Function з вибором різних видів тригонометричних функцій.

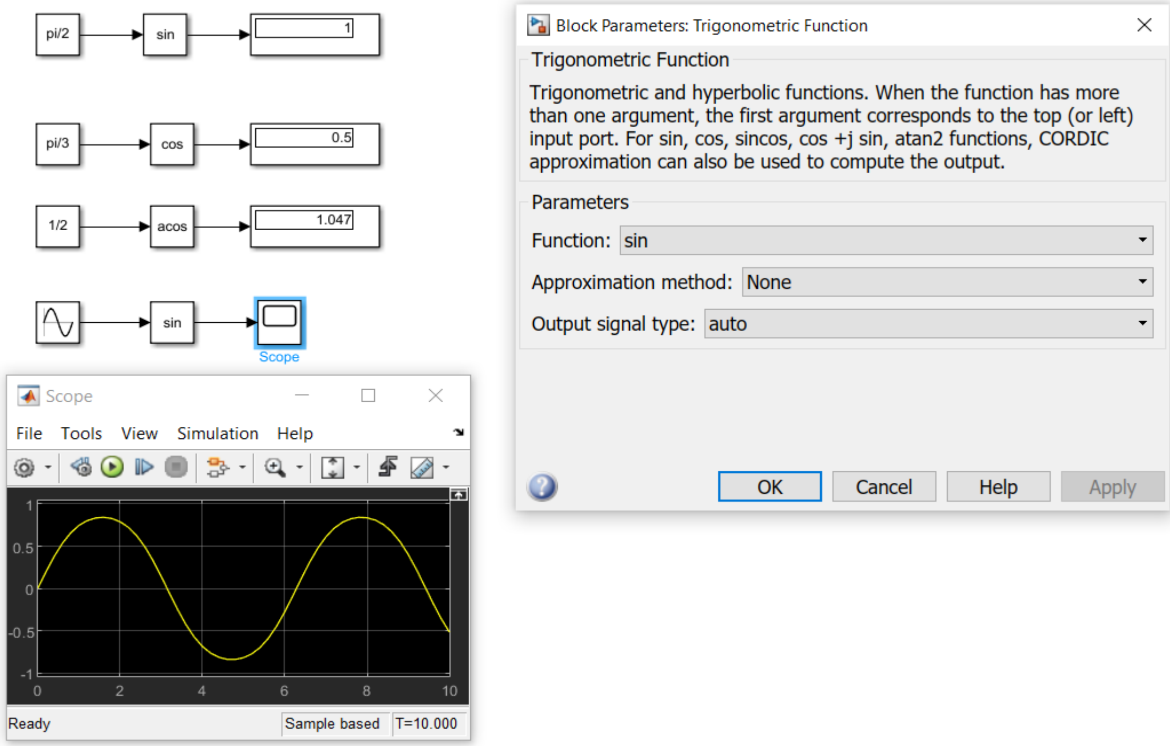


Рис.13.8 Приклад використання блоку Trigonometric Function

Блок Concatenate об'єднує вхідні сигнали для створення вихідного сигналу, елементи якого перебувають у суміжних областях пам’яті. На рис. 13.9 представлений приклад застосування блоку Matrix Concatenate, а на рис. 13.10 – застосування блоку Vector Concatenate.

Для дій над комплексними числами застосовуються блоки Magnitude-Angle to Complex, Complex to Magnitude-Angle, Complex to Real-Imag, Real-Imag to Complex (рис.13.11).

Блок Magnitude-Angle to Complex перетворює модуль і аргумент в комплексне число. Параметри блоку: **Output** – Вихідний сигнал (вибирається зі списку): *Magnitude* – модуль; *Angle* – аргумент, *MagnitudeAndAngle* – модуль і аргумент.

Блок Complex to Magnitude-Angle є протилежним до блоку Magnitude-Angle to Complex, тобто обчислює модуль і (або) аргумент комплексного числа.

Блок Complex to Real-Imag обчислює дійсну і (або) уявну частину комплексного числа. До параметрів блоку відноситься: **Output** – Вихідний сигнал (вибирається зі списку): *Real* – дійсна частина, *Image* – уявна частина, *RealAndImage* – дійсна і уявна частина.

Блок Real-Imag to Complex є протилежним до блоку Complex to Real-Imag, тобто обчислює комплексне число за його дійсною та уявною частинами.

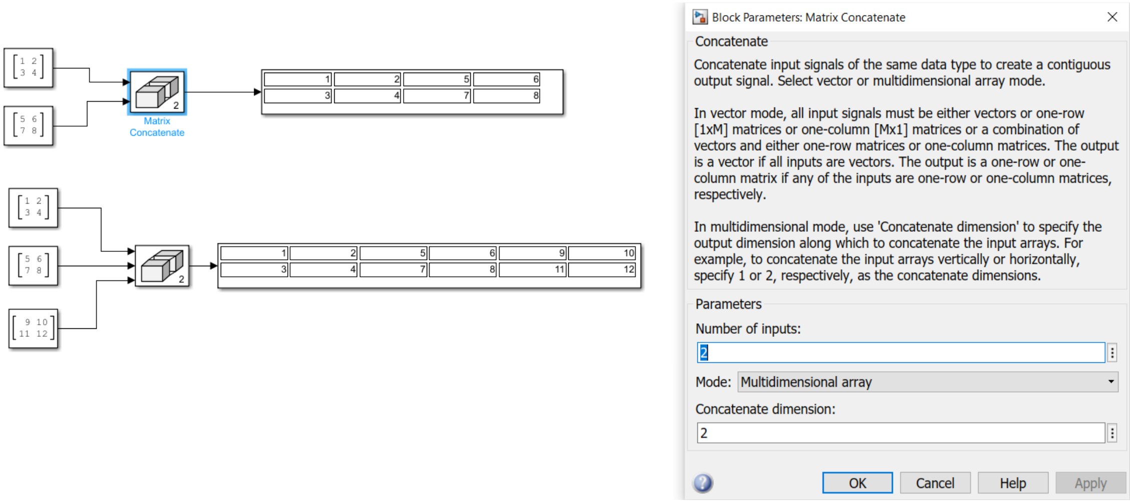


Рис.13.9 Приклад використання блоку Matrix Concatenate

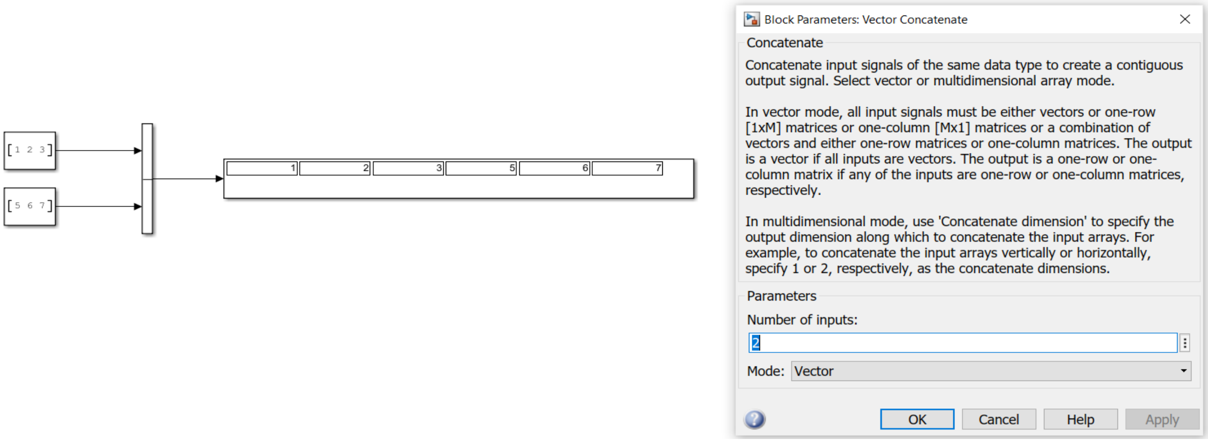


Рис.13.10 Приклад використання блоку Vector Concatenate

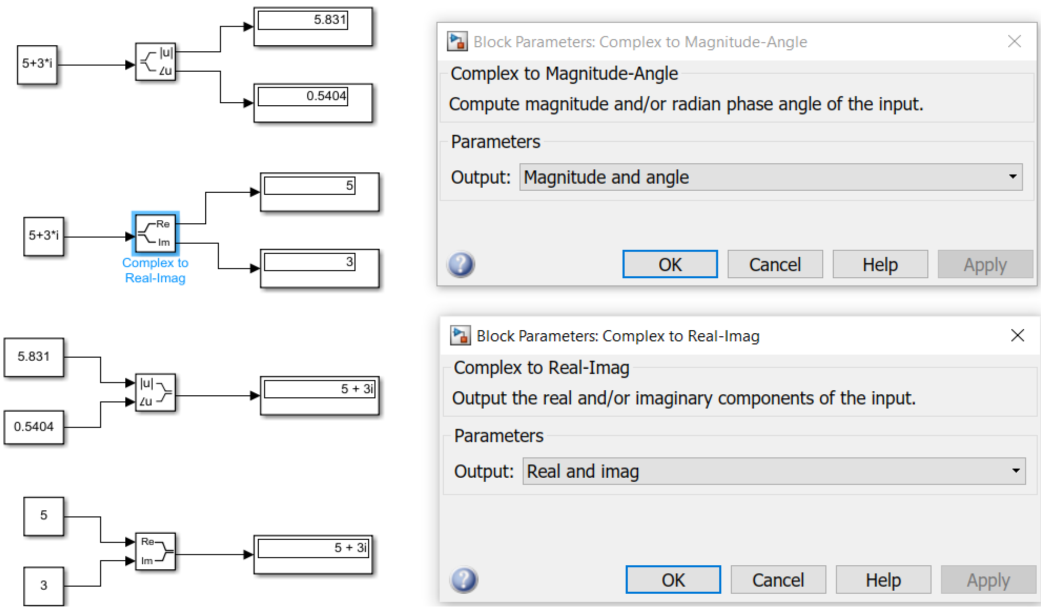


Рис.13.11 Приклад використання блоків Magnitude-Angle to Complex, Complex to Magnitude-Angle, Complex to Real-Imag, Real-Imag to Complex

13.2. Завдання на виконання комп’ютерного практикуму

Завдання 1. Створити S-модель, що генерує вказаний арифметичний вираз, налаштувати параметри блоків, відповідно до варіанту (табл. 1.1 та табл.1.3 з комп'ютерного практикуму №1), і зареєструвати сигнали на виходах кожного з логічно завершених блоків моделі за допомогою дисплею.

Завдання 2. Створити S-моделі, що генерують задані комплексні числа та дії з ними, налаштувати параметри блоків, відповідно до варіанту (табл. 2.1 з комп'ютерного практикуму №2), відобразити дійсну, уявну частини, обчислити модуль та аргумент комплексного числа, отримати комплексно-спряжене число conj і зареєструвати сигнали на виходах кожного з логічно завершених блоків моделі за допомогою дисплеїв.

Завдання 3. Створити S-модель, що генерує вказану функцію, налаштувати параметри блоків, відповідно до варіанту (табл. 13.1), і зареєструвати сигнали на виходах кожного з логічно завершених блоків моделі за допомогою осцилографів.

Таблиця 13.1

Варіанти індивідуальних завдань

Варіант	**Функція**	**Залежність аргументу від часу моделювання**
1	2	3
1	$y = 5 \cdot \sin\left(e^{2x^2 - ln\lvert x^3\rvert}\right)$	$x = \cos t + \operatorname{tg} t$
2	$y = 5 \cdot \cos\left(e^{2x^2 - ln\lvert x^{\sqrt{2}}\rvert}\right)$	$x = \operatorname{arctg} t$
3	$y = 5 \cdot log_5 \left\lvert\sqrt{e^{2x^2}}\right\rvert$	$x = e^{-t} \cos t$
4	$y = 5 \cdot \sin\left(e^{2x^2 - ln\lvert x^3 + 10\rvert}\right) + \sin\left(e^{2x^2}\right)$	$x = \cos t + \operatorname{tg} t$
5	$y = 5 \cdot \sin\left(e^{2x^2}\right)$	$x = floor\left(\frac{\cosh t}{t^3 + 1}\right)$
6	$y = 5 \cdot e^{e^{2x^2 - \sin\lvert x^3\rvert}}$	$x = t$
7	$y = 5 \cdot \sin(2x^2 - ln\lvert x^3\rvert)\ +$ $+3 \cdot \cos(ln\lvert x^3\rvert)$	$x = \cos t + \operatorname{tg} t$
1	2	3

8	$y=\sqrt{10+\dfrac{x^5}{\sin(25x+2)}}+5^x$	$x=2\cdot\sin t$
9	$y=\sqrt{\left\lvert 120-\dfrac{\sin(25x+2)+1.5}{x^5+0.1}\right\rvert}$	$x=e^t$
10	$y=5\cdot tg\left(e^{2x^2-\arcsin\lvert x^3\rvert}\right)$	$x=\cos e^{t+2}$
11	$y=\dfrac{3\cdot\sqrt[5]{x^3+20x}}{(x+1)^x+0.5}ln(\lvert x\rvert+0.05)$	$x=t+1$
12	$y=\dfrac{3\cdot\sqrt[5]{x^3}}{(x+1)^3+e^{0.5x}}\cos(20x+0.05)+$ $+\sin(20x-5)$	$x=1+\sin t$
13	$y=10\cdot e^{ln\left(x^5+\sin(1.5x)\right)}+\sin(2x^2)$	$x=e^{-t}\cos t$
14	$y=5\cdot\sin\left(e^{2x^2-ln\lvert x^3\rvert}\right)$	$x=\arccos t+\sqrt{t}$
15	$y=5\cdot\dfrac{\sin(e^x)}{10\cdot\cos(\sqrt{x}+15)+\left(\sin\left(\dfrac{x^3}{x+1}\right)\right)^2}$	$x=\sin t$
16	$y=12\cdot\dfrac{\sin\left(e^{x^2}\right)}{10\cdot\cos\left(\left\lvert x^{\frac{4}{9}}\right\rvert+15\right)+\sin\left(\dfrac{x}{x+1}\right)}+$ $+\arcsin(2x)$	$x=t$
17	$y=10\cdot\sin\left(e^{2x^2}\right)+tg\sqrt{\lvert x\rvert}+\dfrac{e^x}{x^3}$	$x=\cos t$
18	$y=\cosh\left(\sin\left(e^{2x^2}\right)\right)+tg(x)+x^3e^x$	$x=t^{2/3}$ $x=t^{2/3}$
19	$y=\sinh\left(\cos\left(e^{2x^2}\right)\right)+\mathrm{ctg}\,x+x^{-3}e^x$	$x=\cos t$
20	$y=\mathrm{arctg}\left(e^{2\sqrt[5]{x^2}-\sin\lvert x^3\rvert}\right)+\cos x$	$x=t$
21	$y=\mathrm{arctg}\left(10e^{2\sqrt[5]{x^2}-\sin\lvert x^3\rvert}+\cos x\right)$	$x=\sin(2\mathrm{t})$
22	$y=5^{log_{10}\frac{x^2+5x}{\sqrt{\sin(2x)}}}+e^{-x}$	$x=100\cos t$
23	$y=5^{\log_8\lvert x^2+5x\rvert}+e^{-\sin x}$ $y=5^{log_8\lvert x^2+5x\rvert}+e^{-\sin x}$	$x=round(120\sin t)$
1	2	3

24	$y = \sqrt{\operatorname{tg} 5x^2 + e^{ln(10x^3)+\operatorname{arctg}(10x)}} + \frac{x^3 - 1}{x^2 + 1}$	$x = \operatorname{arcctg} t$
25	$y = \sqrt{\operatorname{ctg}\frac{5x^2}{\sqrt{\lvert x\rvert + 1}} + e^{\operatorname{arctg} x}} + \frac{ln(\lvert x^3\rvert + 0.2) - 1}{x^2 + 1}$	$x = \sin t$
26	$y = \sqrt{5 + \frac{x^4}{\sin(15x + 4)}} + x^5$	$x = 2 * \sin t$
27	$y = \sqrt{\left\lvert 100 - \frac{\sin(28x + 4) + 2.5}{x^3 + 1.1}\right\rvert}$	$x = e^t$
28	$y = 4^{log_{10}\left(\frac{x^3+3x}{\sqrt{\cos(2x)}}\right)} + e^{-x}$	$x = \cos t$
1	2	3
29	$y = 5 \cdot e^{ln\left(x^3+sin(2x)\right)} + \sin(2x^3)$	$x = e^{-t} \cos t$
30	$y = 10 \cdot sin\left(e^{3x^2 - lnx^2}\right)$	$x = arccost + \sqrt[3]{t}$

13.3. Контрольні запитання

1. Які блоки містяться у розділі Math Operation?
2. За допомогою яких блоків можна здійснити елементарні арифметичні операції (множення, ділення, додавання, віднімання)?
3. Яку функцію виконує блок Gain?
4. Які параметри блока Math Function?
5. Які параметри блока Trigonometric Function?

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 14
СТВОРЕННЯ S-МОДЕЛЕЙ, ВИКОРИСТОВУЮЧИ РОЗДІЛ CONTINUOUS

Мета: ознайомлення з розділом Continuous бібліотеки Simulink та набуття навичок створення S-моделей, використовуючи розділ Continuous.

14.1. Теоретичні відомості

На рис. 14.1 представлено вікно розділу бібліотеки Continuous з неперервними компонентами.

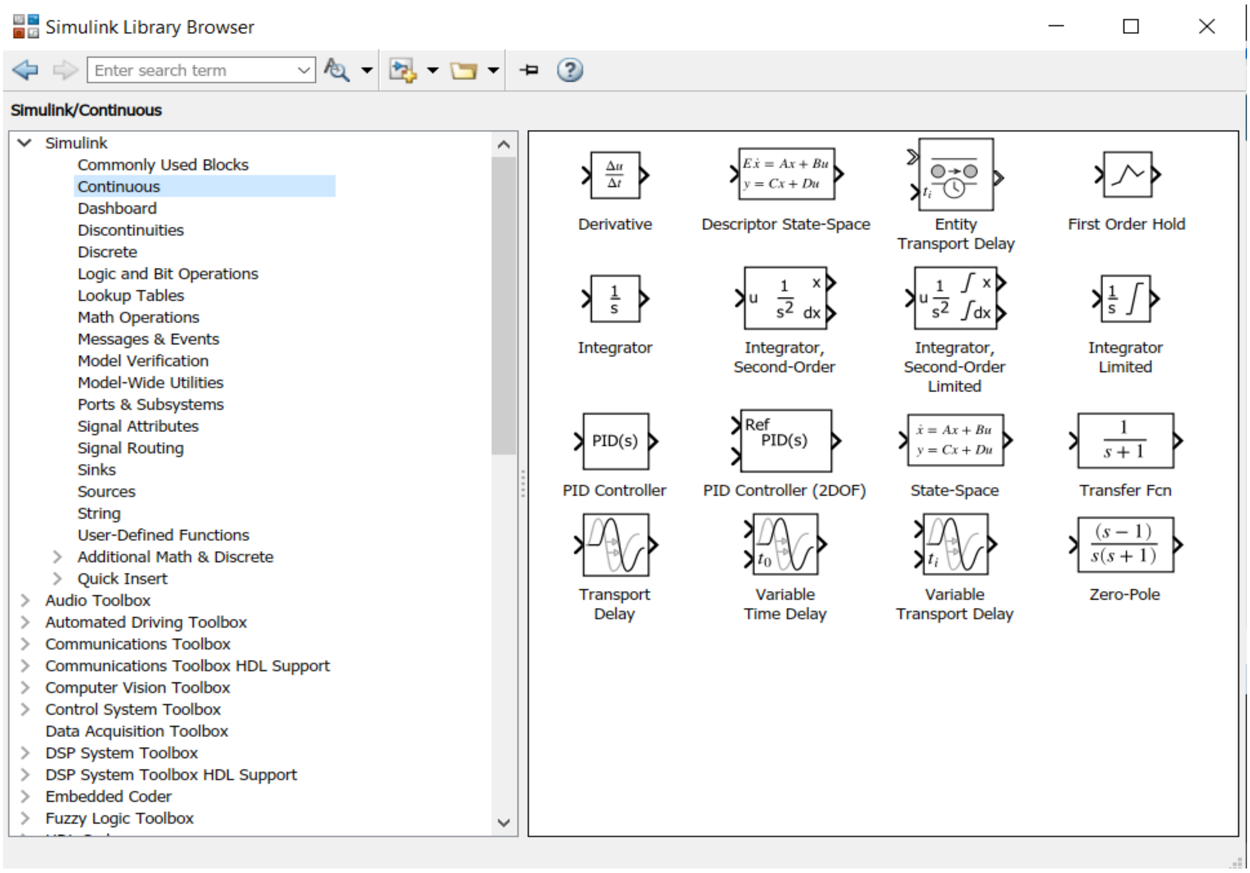
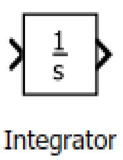


Рис. 14.1 Розділу Continuous

Основні блоки, що входять до розділу Continuous бібліотеки Simulink:



- ідеальна інтегруюча ланка. На рис. 14.2 наведено приклад застосування блоку інтегрування для вхідного сигналу у вигляді синусоїдального імпульсу та вікно налаштування параметрів інтегруючого блоку.

Вікно параметрів інтегруючого блоку містить такі елементи:

• *External reset* (зовнішнє скидання/обнулення) – тип зовнішнього керуючого сигналу, що обирається зі списку: *none* – ні, *rising* – наростаючий, *falling* – спадаючий, *either* – будь-який;

• *Initial condition Source* – джерело початкового значення вихідного сигналу при інтегруванні. У списку можна вибрати внутрішнє (internal) або зовнішнє (external) джерело;

• *Initial condition* (початкові умови) – установка початкового значення вихідного сигналу при інтегруванні (у вигляді числа, за умовчанням 0);

• *Limit output* – включення / відключення обмеження вихідного сигналу;

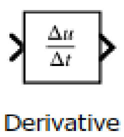
• *Wrap state* – включення / відключення переносу циклічного стану;

• *Show saturation port* – управляє відображенням порту, що виводить рівні обмеження вихідного сигналу;

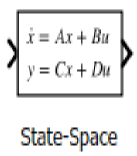
• *Show state port* – управляє відображенням порту стану системи;

• *Absolute tolerance* – абсолютна похибка (за замовчуванням автоматичний вибір – auto).

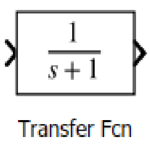
• *Enable zero crossing detections* – включення перевірки наявності переходів через нуль.



- ідеальна диференціююча ланка. На рис. 14.3 приведено приклад послідовного диференціювання прямокутних імпульсів та вікно налаштування параметрів блоку диференціювання.



- дозволяє задати лінійну ланку шляхом введення чотирьох матриць простору станів.



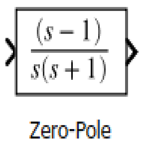
- дозволяє задати лінійну ланку шляхом введення її передатної функції. На рис. 14.4 показано приклад застосування блоку передатної функції. Вид блоку показаний після налаштування його параметрів у вікні параметрів.

Блок Transfer Fcn має два параметри - вектори коефіцієнтів поліномів чисельника Numerator і знаменника Denominator.

$$W(p) = \frac{b_n p^n + b_{n-1} p^{n-1} + \ldots + b_1 p + b_0}{a_m p^m + a_{m-1} p^{m-1} + \ldots + a_1 p + a_0}.$$

Numerator – значення коефіцієнтів чисельника b_i; їх вводять через пробіл, починаючи з коефіцієнта b_n при старшій похідній;

Denominator – значення коефіцієнтів знаменника a_i; їх вводять через пробіл, починаючи з коефіцієнта a_m при старшій похідній.



- використовується для визначення динамічної ланки за допомогою введення векторів її полюсів і нулів, а також значення коефіцієнта передачі.

- забезпечує затримку сигналу на задану кількість кроків модельного часу (не обов'язково ціле число).

- дозволяє задавати керовану ззовні величину затримки.

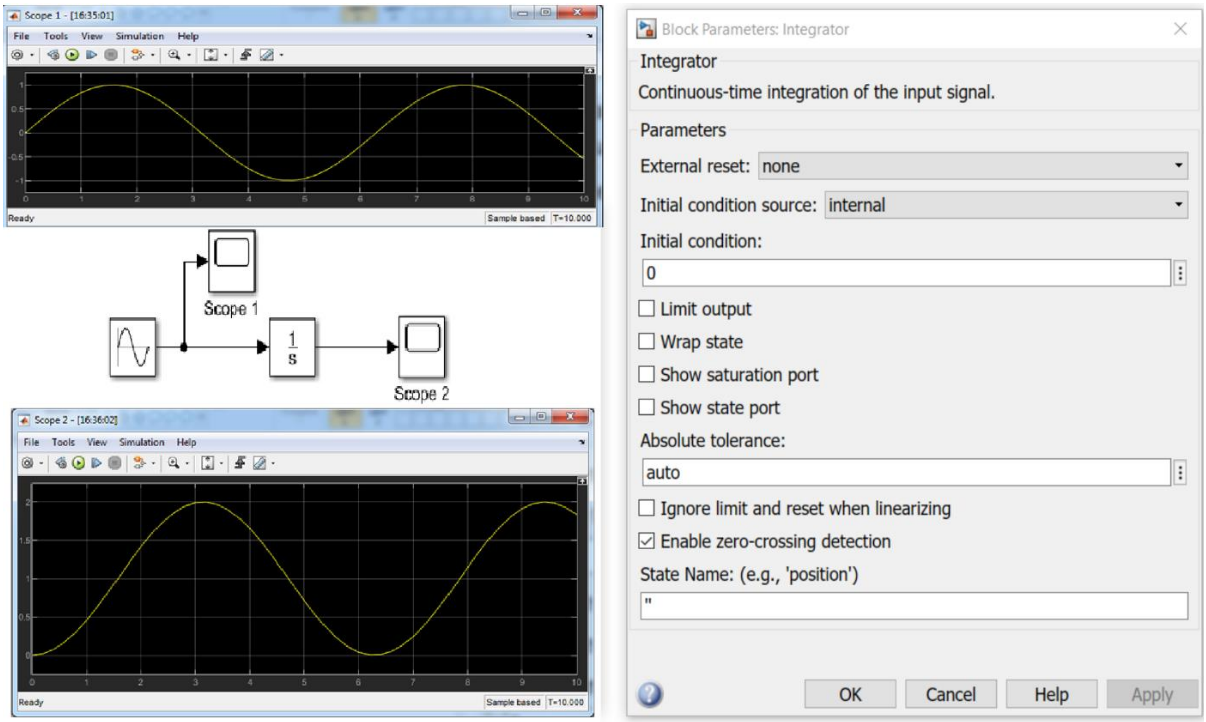


Рис. 14.2 Приклад інтегрування синусоїдальних імпульсів та вікно налаштування параметрів блоку інтегратора

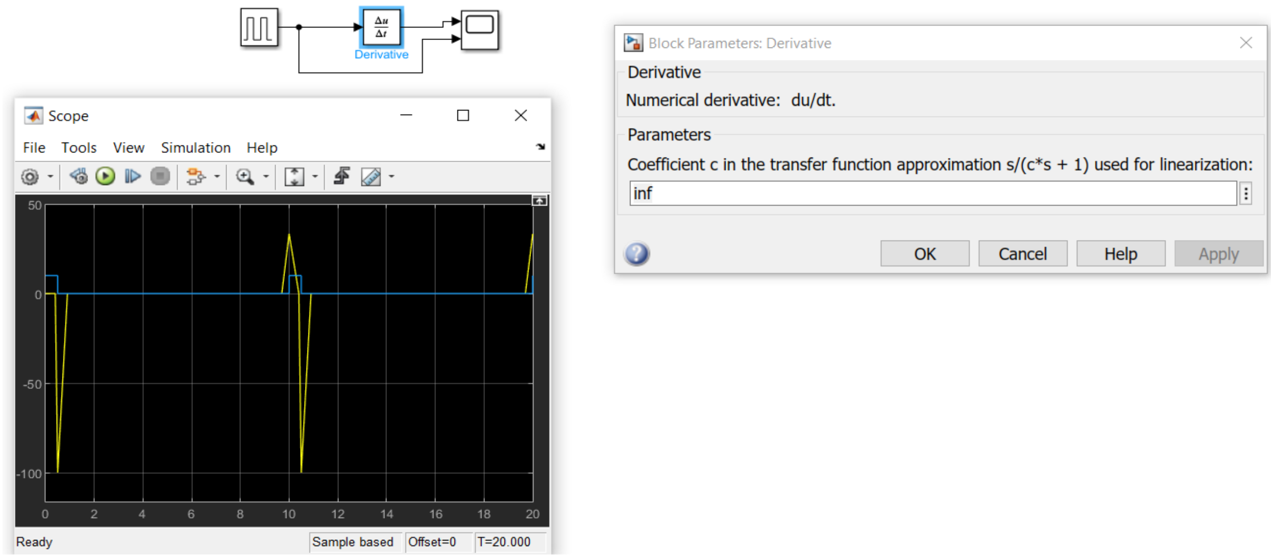


Рис. 14.3 Приклад диференціювання прямокутних імпульсів та вікно налаштування параметрів блоку диференціювання

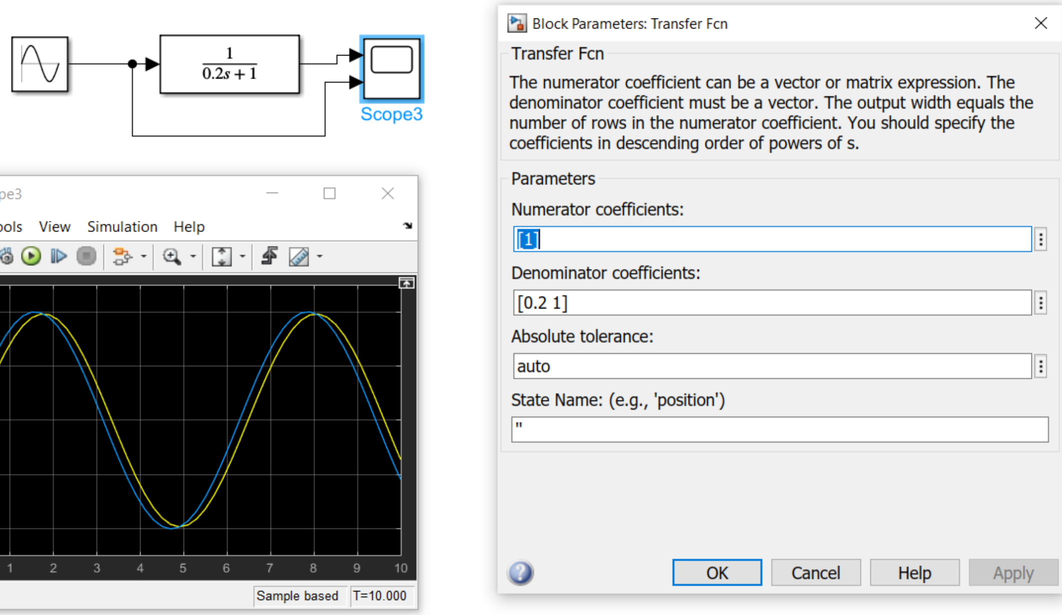


Рис.14.4 Приклад застосування блоку Transfer Fcn

Продемонструємо далі, як за допомогою раніше вивчених блоків можна виконати моделювання роботи динамічної системи. Для цього використаємо найпростіші диференціальні рівняння, що описують роботу гіротахометра:

$$J \cdot \ddot{\beta} + h\dot{\beta} + c\beta = H\omega_{\xi} \cos\beta, \qquad (14.1)$$

де β – кут повороту рухомої частини датчики, по якому можна визначити кутову швидкість об'єкта ($\omega_{\xi} = \dfrac{c}{H}\beta$); J – момент інерції гірокамери; h – коефіцієнт демпфірування; c – жорсткість механічної чи електричної пружини; H – власний кінетичний момент (момент кількості руху); ω_{ξ} – проекції кутової швидкості основи. Складова ω_ξ является інформаційною складовою, яку вимірює гіротахометр.

Щоб знайти розв'язок (14.1) їх потрібно проінтегрувати, і для цього (14.1) треба записати відносно старшої похідної:

$$\ddot{\beta} = \frac{1}{J}\left(-h\dot{\beta} - c\beta + H\omega_\xi \cos\beta\right) \tag{14.2}$$

Логіка розв'язку (14.2) полягає у тому, що потрібно створити сигнал, який являє собою другу похідну від β, а потім послідовно проінтегрувати необхідну кількість раз за допомогою блока ідеального інтегратора. Відповідні складові сигналу β та $\dot{\beta}$ потрібно чи домножити на константу за допомогою блоку Gain, чи виконати більш складні математичні операції за допомогою відповідних блоків з розділу Math Operations. Окремі складові сигналу, що входять до $\ddot{\beta}$ підсумовуються за допомогою блоку Add. Проте це не означає, що вхідний сигнал для інтегрування має формуватися тільки шляхом підсумовування, для його формування, за необхідності, можна включати весь спектр доступних математичних операцій. Модель, що дозволяє проінтегрувати (14.2) показана на рис. 14.5 (кутова швидкість ω_ξ задана у вигляді синусоїдального сигналу). Зверніть увагу, що значення коефіцієнтів, що визначають властивості частини блоків (Gain) вводилися не явно, а у вигляді змінних. Це дуже зручно при моделюванні складних взаємопов'язаних систем. Задати значення цих змінних можна в командному вікні або аналогічно, записавши значення, зберегти скрипт-файл:

```
>> H=0.02, h=0.001, c=0.004, J=2e-5
H =
    0.0200
h =
   1.0000e-03
c =
    0.0040
J =
   2.0000e-05
>>
```

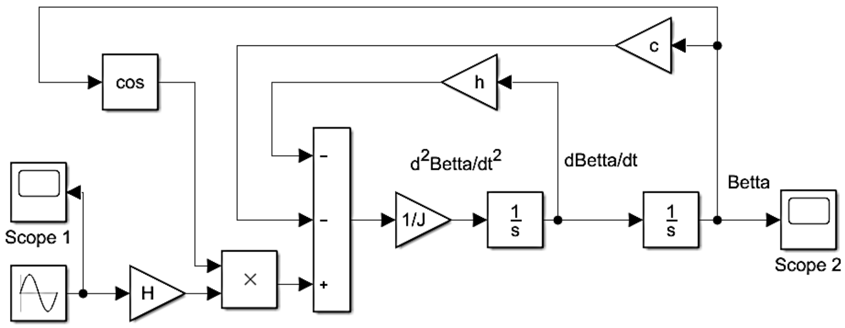


Рис. 14.5. Simulink-модель для моделювання динамічної системи з урахуванням невеликої нелінійності

Кутову швидкість задамо у вигляді $\omega_\xi = 0.1\,\text{рад/с}$ (рис. 14.6), результат запуску моделі у вигляді проінтегрованого сигналу гіротахометра показано на рис. 14.7

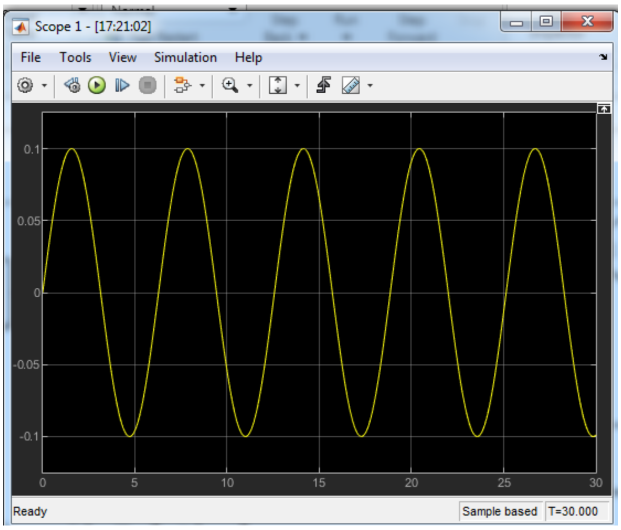


Рис.14.6. Вхідна кутова швидкість, що діє на гіротахометр

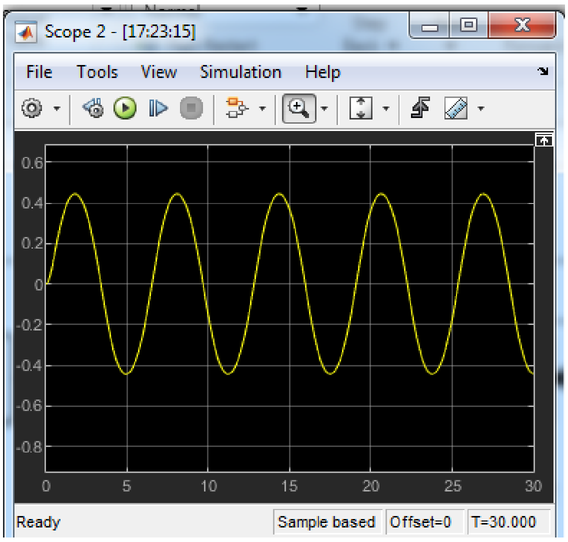


Рис.14.7. Вихідний сигнал гіротахометра (двічі проінтегрований сигнал $\ddot{\beta}$)

Систему, що представлена на рис. 14.5, можна ліанеризувати, замінивши $\cos\beta$ на 1, враховуючи $\beta \ll 1$:

$$J \cdot \ddot{\beta} + h\dot{\beta} + c\beta = H\omega_\xi, \tag{14.3}$$

Для лінеаризованих рівнянь (14.3) можна записати передатну функцію:

$$W_{ГТ}(p) = \frac{\beta(p)}{\omega_\xi(p)} = \frac{H}{Jp^2 + hp + c}. \tag{14.4}$$

Simulink-модель, що реалізує лінеаризовану модель гіротахометра показана на рис. 14.8, а результат її роботи на рис. 14.9 (вхідний сигнал аналогічний як на рис. 14.6)

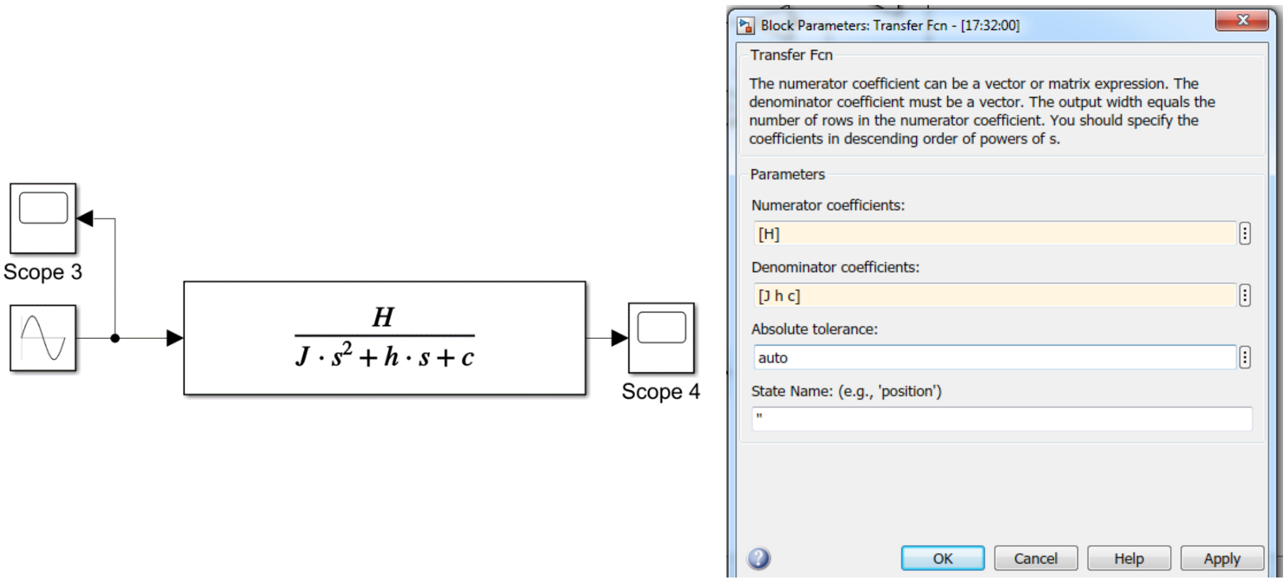


Рис. 14.8. Реалізація в середовищі Simulink ліанеризованої моделі гіротахометра (14.3) за допомогою блоку Transfer Fcn

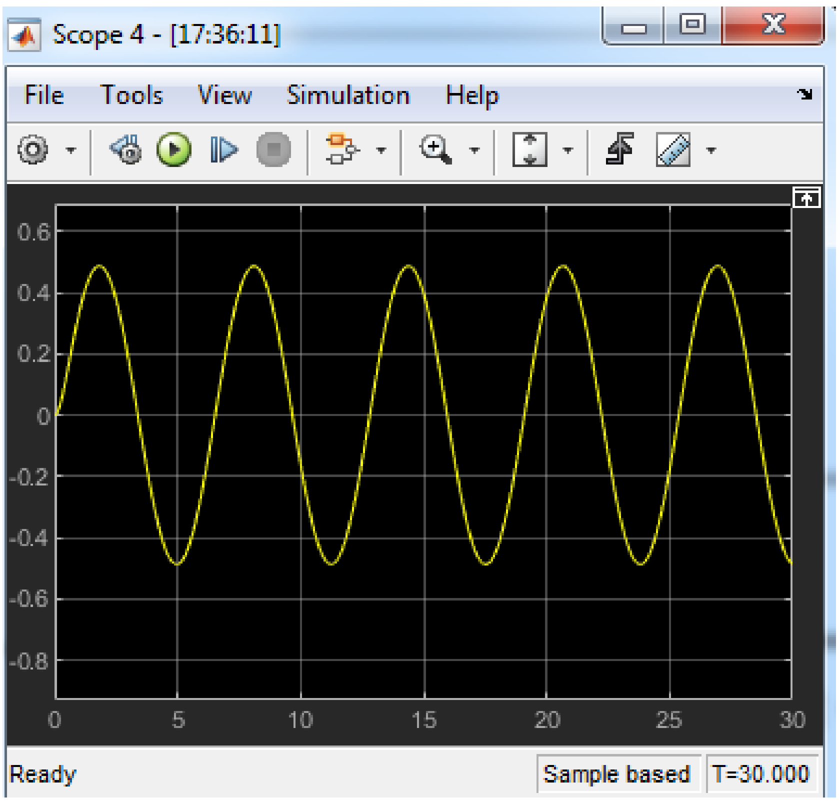


Рис. 14.9. Вихідний сигнал ліанеризованої моделі гіротахометра

14.2. Завдання на виконання комп'ютерного практикуму

Завдання 1. Створити S-моделі, використовуючи різні блоки з розділу бібліотеки Continuous та різні джерела, налаштувати параметри блоків (відповідно до варіанту) і зареєструвати сигнали на виходах моделей за допомогою осцилографів.

1.1. Блок інтегрування Integrator

\+ джерело постійного впливу Constant (відповідно до номеру варіанту);

\+ джерело синусоїдального впливу Sine Wave (табл.12.1);

\+ джерело наростаючого впливу Ramp (табл. 12.2);

\+ джерело одиночного перепаду Step (табл. 12.3);

\+ джерело прямокутних імпульсів Pulse Generator (табл. 12.4).

1.2. Блок диференціювання Derivative

\+ джерело постійного впливу Constant (відповідно до номеру варіанту);

\+ джерело синусоїдального впливу Sine Wave (табл.12.1);

\+ джерело наростаючого впливу Ramp (табл. 12.2);

\+ джерело одиночного перепаду Step (табл. 12.3);

\+ джерело прямокутних імпульсів Pulse Generator (табл. 12.4).

1.3. Блок передавальної функції Transfer Fcn (табл. 13.1)

\+ джерело постійного впливу Constant (відповідно до номеру варіанту);

\+ джерело синусоїдального впливу Sine Wave (табл.12.1);

\+ джерело наростаючого впливу Ramp (табл. 12.2);

\+ джерело одиночного перепаду Step (табл. 12.3);

\+ джерело прямокутних імпульсів Pulse Generator (табл. 12.4).

Завдання 2. Змінити варіант та побудувати нові графіки для кожного типу джерела.

Завдання 3. Порівняти отримані графіки та зробити висновки.

14.3. Контрольні запитання

1. Які блоки містяться у розділі Continuous**?**
2. Які параметри блоку інтегрування?
3. Які параметри блоку Transfer Fcn?

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Авдєєва Т.В. Звичайні диференціальні рівняння вищих порядків: Практикум/ Т.В. Авдєєва, О.Б. Качаєнко .- К.: НТУУ «КПІ», 2016. – 67 с.

2. Барановская Г.Г. Микрокалькуляторы в курсе высшей математики : Практикум : Учеб. пособие для вузов / Г. Г. Барановская, И. Н. Любченко. – К : Вышая шк. Головное изд-во, 1987. – 288 с.

2. Бахвалов Н. С. Численные методы / Н. С. Бахвалов. – М.: Наука, 1973. – 632с.

3. Бахвалов Н. С. Численные методы / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. — 6-е изд. — М. : БИНОМ. Лаборатория знаний, 2008. — 636 с.: ил.

4. Березин М.С. Методы вычислений / М.С. Березин, Н.П. Жидков. – М.: Наука, 1966. т.1,2. – 620с.

5. Васильев В.В., Симак Л.А., Рыбникова А.М. Математическое и компьютерное моделирование процессов и систем в среде MATLAB/SIMULINK. Учебное пособие. — Киев: Национальный авиационный университет, 2008. — 91 с.

6. Вержбицкий, В.М. Основы численных методов / В.М. Вержбицкий. – М.: Высшая школа, 2002. – 840 с.

7. Горбаченко В. И., Убиенных Г. Ф. Вычислительные методы линейной алгебры: лабораторный практикум в системе MATLAB. – Пенза: ПГУ, 2010. – 93 с.

8. Демидович Б.П. Основы вычислительной математики / Б.П. Демидович, И.А. Марон. – М.: Наука, 1970. – 664с. Дьяконов В. П. MATLAB. Полный самоучитель / В.П. Дьяконов. – М.: ДМК Пресс, 2012. – 768 с.

9. Документація MatLab [Електронний ресурс]. — Режим доступа: https://docs.exponenta.ru/

10. Задачин В. М. Чисельні методи: навчальний посібник / В. М. Задачин, І. Г. Конюшенко. – Х.: Вид. ХНЕУ ім. С. Кузнеця, 2014. – 180 с.

11. Лазарєв Ю.Ф. Моделювання на ЕОМ: Навчальний посібник / Ю.Ф. Лазарєв. – Київ: Корнійчук, 2007. – 290 с.

12. Лазарєв Ю.Ф. Початки програмування у середовищі MatLab: Навчальний посібник / Ю.Ф. Лазарєв. – Київ: Корнійчук, 1999. – 160 с.

13. Солонина А.И. Цифровая обработка сигналов. Моделирование в MATLAB/А.И. Солонина, С.М. Арбузов. – СПб.: БХВ-Петербург, 2008. – 816 с.

14. Черных И.В. Simulink: Инструмент моделирования динамических систем / И.В.Черных. – М.: Диалог-МИФИ. 2003. – 252 с. 4. Дьяконов В. П. Simulink 5/6/7: Самоучитель. – М.: ДМК-Пресс, 2008. – 784 с.

15. Simulink Documentation - MathWorks [Електронний ресурс]. — Режим доступа: https://www.mathworks.com/help/simulink/

16. Математичне моделювання на ЕОМ (частина 1) [Текст] : метод. вказівки до викон. комп. практикумів для студентів напряму підгот. 6.051003 «Приладобудування» / Уклад.: Ю.Ф. Лазарєв, Д.О. Півторак, Д.В. Шевчук. – К.: НТУУ «КПІ», 2015. – 70 с.

17. Математичне моделювання на ЕОМ (частина 2) [Текст] : метод. вказівки до викон. комп. практикумів для студентів напряму підгот. 6.051003 «Приладобудування» / Уклад.: Ю.Ф. Лазарєв, Д.О. Півторак, С.Л. Лакоза. – К.: НТУУ «КПІ», 2015. – 86 с.