МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний технічний університет України
«Київський політехнічний інститут»

Математичне моделювання на ЕОМ (частина 1)

Методичні вказівки
до виконання комп'ютерних практикумів
для студентів напряму підготовки
6.051003 «Приладобудування»

Рекомендовано Вченою радою
Приладобудівного факультету НТУУ «КПІ»

Київ
НТУУ «КПІ»
2015

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.

Рекомендовано Вченою радою ПБФ НТУУ «КПІ»
(Протокол № 8/15 від 28.09.2015 р.)

Навчально–методичне видання

Математичне моделювання на ЕОМ
(частина 1)
до виконання комп'ютерних практикумів
для студентів напряму підготовки
6.051003 «Приладобудування»

Укладачі: *Лазарєв Юрій Федорович*, канд. техн. наук, доц.
Півторак Діана Олександрівна, канд. техн. наук
Шевчук Дмитро Володимирович

Відповідальний редактор *Павловський О.М.*, канд. техн. наук.

Рецензенти: *Вислоух С.П.,* канд. техн. наук, доц.

ЗМІСТ

ВСТУП

Комп'ютерні практикуми з дисципліни «Математичне моделювання на ЕОМ» для студентів напряму підготовки 6.051003 «Приладобудування» сприятимуть закріпленню, поглибленню та узагальненню отриманих знань та умінь, а також сприятимуть розвитку навичок самостійної творчої роботи студентів у процесі їх навчання, при дипломному проектуванні, а також у професійній діяльності.

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 1.1

Початок обчислень у MATLAB

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

Теоретичні відомості

Обчислення з дійсними числами

Введення дійсних чисел

Введення чисел з клавіатури здійснюється за загальними правилами, що прийняті для мов 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

-1.234563e-17,

то після натискання клавіші <Enter> в цьому вікні з'явиться запис:

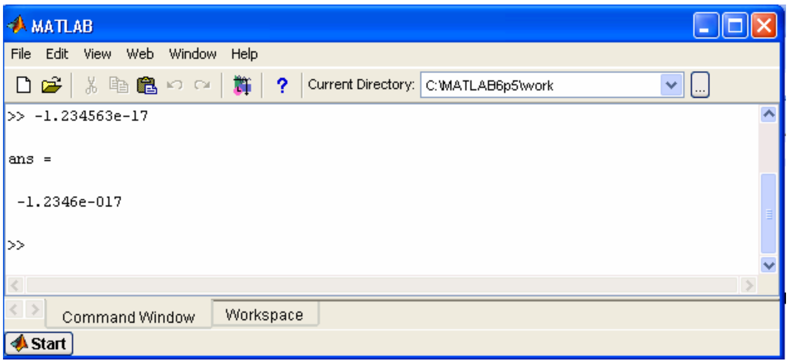

Рис. 1.1 Командне вікно MATLAB

Слід зазначити, що результат виводиться у вигляді (форматі), що визначається попередньо встановленими (за допомогою розділу «Options» меню командного вікна) форматом.

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

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

Найпростіші арифметичні дії

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

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

(5e-5)^3*7/2e5+5*13.5.

Якщо після введення з клавіатури цієї послідовності натиснути клавішу <Enter>, у командному вікні в цьому вікні з'явиться запис:

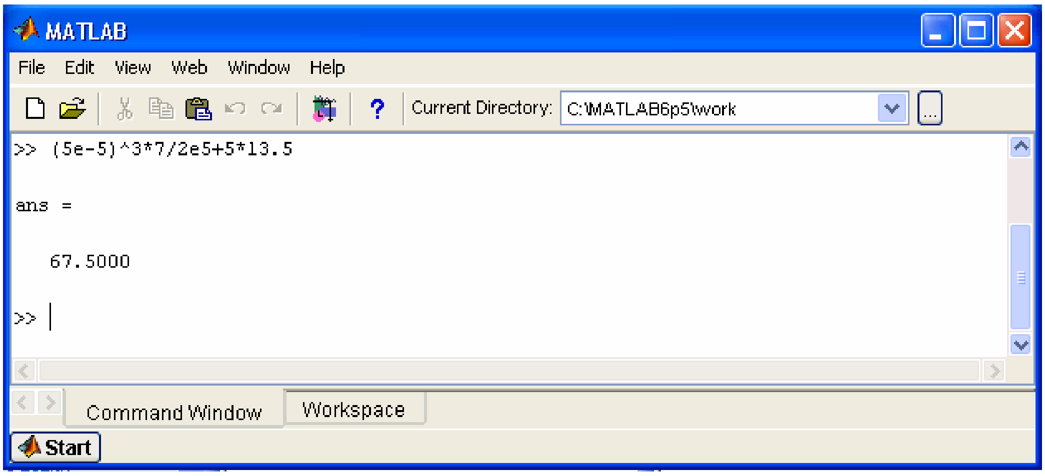

Рис. 1.2 Вікно з прикладом використання режима калькулятора

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

Взагалі виведення проміжної інформації у командне вікно підпорядковується таким правилам:

- 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;

- 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;

- взагалі формат виведення до командного вікна має вигляд:

<ім’я змінної>=<результат>.

Особливістю 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>[;]

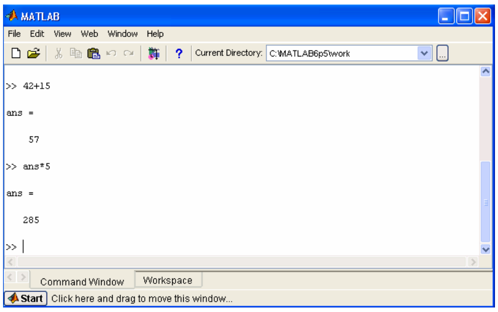

Рис. 1.3 Вікно з прикладом використання змінної ***ans***

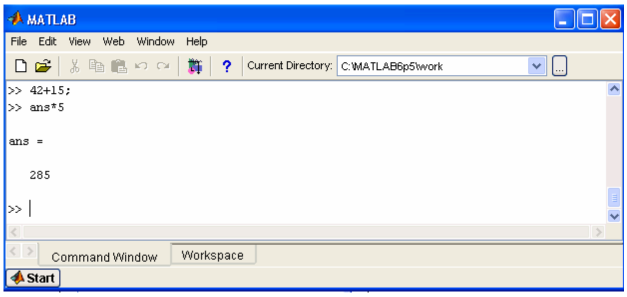

Рис. 1.4 Вікно з прикладом використання символу «;»

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 «x = 34+17», на екрані з’явиться запис:

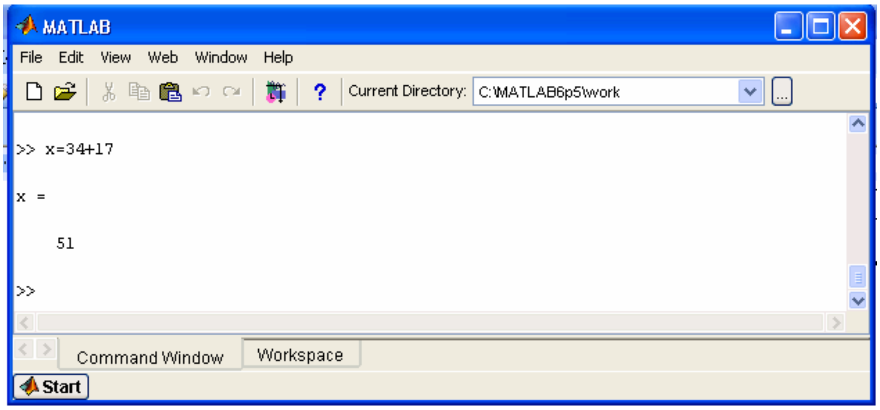

Рис. 1.5 Вікно з прикладом виконання операції присвоєння («=»)

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.

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).

Завдання до виконання комп'ютерного практикуму

Завдання 1. Обчисліть зазначений арифметичний вираз відповідно до варіанту згідно з табл.1.1. Укажіть послідовність натискання клавіш. Порівняйте отриманий результат із наведеною відповіддю.

Табл.1 Варіанти індивідуальних завдань

Варіант	Вираз	Відповідь
1	2	3
1.	$\dfrac{\left(12\frac{1}{6}-6\frac{1}{27}-5,2513,5\right)+0,111}{0,02}$	599,3
2.	$\dfrac{\left(1\frac{1}{12}+2\frac{5}{32}+\frac{1}{24}\right):9,6+2,13}{0,0004}$	$6{,}18\cdot10^{-3}$
3.	$\dfrac{\left(6,6-3\frac{3}{14}\right)5\frac{5}{6}}{(21-1,25):2,5}$	2,5
4.	$\dfrac{2,625-\frac{2}{3}\cdot2\frac{5}{14}}{\left(3\frac{1}{12}+4,375\right):19\frac{8}{9}}$	2,8095
5.	$\dfrac{0,134+0,05}{18\frac{1}{6}-1\frac{11}{14}-\frac{2}{15}\cdot2\frac{6}{7}}$	0,0115
6.	$\dfrac{\left(58\frac{4}{15}-56\frac{7}{24}\right):0,8+2\frac{1}{9}\cdot0,225}{8,75\cdot0,6}$	0,56071
7.	$\dfrac{\left(\frac{0,216}{0,15}+0,56\right):0,5}{\left(7,7:24,75+\frac{2}{15}\right)4,5}$	2
8.	$\dfrac{1\frac{4}{11}\cdot0,22:0,3-0,96}{\left(0,2-\frac{3}{40}\right)1,6}$	0,2
9.	$\dfrac{\left(\frac{3}{5}+0,425-0,005\right):0,12}{30,5+\frac{1}{6}+3\frac{1}{3}}$	0,25

1	2	3
10.	$\frac{3\frac{1}{3}+2,5}{2,5-1\frac{1}{3}}\cdot\frac{4,6-2\frac{1}{3}}{4,6+2\frac{1}{3}}:\left(\frac{0,05}{\frac{1}{7}-0,125}+5,7\right)$	0,19231
11.	$\frac{0.725+0,42(6)}{0,128-6,25-(0,0345:0,12)}\cdot 0,25$	- 0,04492
12.	$\frac{\left(4,5\cdot 1\frac{2}{3}-6.75\right)\cdot 0,(6)}{\left(3,333\cdot 0,3+0,222\cdot\frac{4}{9}\right)2\frac{2}{3}}$	0,27308
13.	$\frac{\left(5\frac{4}{45}-4\frac{1}{6}\right):5\frac{8}{15}}{\left(4\frac{2}{3}+0,75\right)3\frac{9}{13}}\cdot 34\frac{2}{7}.$	0,28571
14.	$\frac{1\frac{4}{11}\cdot 0,22:0,3-0.96}{\left(0,2-\frac{3}{40}\right)1,68}$	0,19048
15.	$\frac{\left(40\frac{7}{30}-38\frac{5}{12}\right):10,9+\left(0,875-\frac{7}{30}\right)\cdot\frac{20}{11}}{0,008}$	166,67
16.	$\frac{(68,023-66,028):6\frac{1}{9}+\frac{7}{40}\cdot 4,5}{0,042+0,086}$	8,7028
17.	$\frac{(2,1-1,965):(1,2\cdot 0,045)}{0,00325:0,013}-\frac{4}{0,2\cdot 0,73}$	-17,397
18.	$\frac{(1,88+2,127)\cdot 0,01875}{0,625-\frac{13}{18}:3,13}+8,29$	8,4805
19.	$\frac{3:0,4-0,009:(0,15:2,5)}{0,32\cdot 6+0,033-(5,3-3,88)}$	13,79
20.	$\frac{(34,06-33,81)\cdot 4}{6,84:(28,57-25,15)}+1,33:\frac{4}{21}$	7,4825

1	2	3
21.	$\dfrac{8,8077}{20-\left(28,2:\left(13,333\cdot 0,3+0,0125\right)\right)2,004}$	1,4889
22.	$\dfrac{\left(1,75:\dfrac{2}{3}-0,7\cdot 1,125\right):\dfrac{7}{12}}{\left(0,2012-0,0325\right):400}$	7468,9
23.	$\dfrac{\left(26\dfrac{1}{3}-18,02\cdot 0,75\right)\cdot 2,4:0,88}{1,37-23\dfrac{2}{3}:1,82}$	-3,005
24.	$26:\dfrac{3:(0,48+0,27)}{2,52\cdot(1,38+2.45)}+1,27$	18,8359

Завдання 2. Проведіть обчислення по заданій формулі при заданих значеннях параметрів. Зазначте необхідну послідовність дій. Порівняйте отриманий результат із наведеною відповіддю.

Вказівка. У системі MATLAB декілька останніх команд запам'ятовуються. Виклик цих команд у командне вікно здійснюється натисканням клавіш.

Використовуйте цю можливість для повторного звернення до набраної функції.

1. $3m^2+\sqrt[3]{2n^2}:m$; а) $m=-\dfrac{14}{5}$; $n=tg\dfrac{\pi}{8}$; б) $m=2,2\cdot 10^{-2}$; $n=\dfrac{1}{3,1}$

ВІДПОВІДЬ : а) 23,27; б)26,938.

2. $\dfrac{4}{3}l^3\sin^2\dfrac{\alpha}{2}\sqrt{\cos\alpha}$; $a)l=1,7\cdot 10^3$, $\alpha=18^\circ$; б) $l=\dfrac{16}{21},\alpha=\dfrac{\pi}{5}$.

ВІДПОВІДЬ : а) $1.5633e+0.008$; б) $5.0651e-0.002$.

3. $\sqrt{\dfrac{a\sqrt{b}}{\sqrt[3]{tg\alpha}}}$;	а) a=1,5, b=0,8, $\alpha = 61^{\circ}$;	б) $a = 3\cdot 10^{2}$, $b = 0,71$, $\alpha = \dfrac{3}{7}\pi$.
ВІДПОВІДЬ :	а) $1.0498e+000$;	б) $1.2429e-0.001$.
4. $\dfrac{3a^{2}\sqrt{6,8\cdot(a-b)}}{4(a+b)^{3}}$;	а) $a = 4,13\cdot 10^{-1}, b = \dfrac{1}{261}$;	б) $a = \sin\dfrac{5\pi}{8}, b = -tg12^{\circ}$.
ВІДПОВІДЬ :	а) $2.9464e+000$;	б) $4.9445e+000$.
5. $\dfrac{c^{3}}{6}\cos\dfrac{\alpha}{2}\sqrt{\sin\alpha}$;	а) $c = \lg 2,38, \alpha = \dfrac{\pi}{5}$;	б) $c = e^{-0,3}, \alpha = 65^{\circ}$.
ВІДПОВІДЬ :	а) $3.4657e-0.004$;	б) $2.2120e-002$.
6. $\sqrt{\dfrac{n^{3}}{16,\sin\alpha\sin 2\alpha}}$;	а) $n = 3,1516\cdot 10^{-2}, \alpha = 5^{\circ}$.	б) $n = e^{3,5}, \alpha = \dfrac{2\pi}{13}$.
ВІДПОВІДЬ :	а) $1.1265e-002$;	б) $7.6324e+001$.
7. $5\sin 35^{\circ}\sqrt{\dfrac{S^{3}\cos 36^{\circ}}{\pi^{3}tg\alpha}}$;	а) $S = \ln 3, \alpha = 44^{\circ}$;	б) $S = \dfrac{18}{25}, \alpha = \dfrac{7}{12}\pi$.
ВІДПОВІДЬ :	а) $5.4283e-001$;	б) $8.9703e-018+1.4650e-001i$.
8. $\left\lvert\lg\left(1+\sin\alpha\right)+\ln\left(1-\sin\beta\right)\right\rvert$;	а) $\alpha = \dfrac{3\pi}{7}, \beta = 83^{\circ}$.	б) $\alpha = \dfrac{2}{3}\pi, \beta = 16^{\circ}$.
ВІДПОВІДЬ :	: а) $4.6035e+000$;	б) $4.6035e+000$;
9. $\sqrt[3]{\sin^{2}(\alpha+\beta)-\sin^{2}(\alpha-\beta)}$;	а) $\alpha = \dfrac{5}{7}\pi, \beta = 0,3\pi$;	б) $\alpha = 12^{\circ}, \beta = 220^{\circ}$.
ВІДПОВІДЬ :	а) $4.8756e-001+8.448e-001i$	б) $7.3715e-001$.
10. $\left(\log_{a}\left(b+1,4\right)\right)^{-3/4}$;	а) $a = 3,56, \mathrm{b} = e^{0,316}$;	б) $a = 2, b = 2,1649\cdot 10^{-2}$.
ВІДПОВІДЬ :	а) $1.1790e+000$;	б) $6.6559e+001$.

11. $3\left(p^{-2/3}+q^{-1/2}\right)\sqrt[3]{pq}$; а) $p=\ln 3, q=\lg 3$; б) $p=0,013, \mathrm{q}=1,4\cdot 10^{2}$.

ВІДПОВІДЬ : а) $5.7737e+000$; б) $6.6559e+001$.

12. $\frac{2}{3}m\sqrt{m\sqrt[3]{m\sqrt[4]{m}}}$; а) $m=3,6485\cdot 10^{2}$; б) $m=\frac{24}{37}$.

ВІДПОВІДЬ : а) $1.5880e+004$; б) $5.4516e-001$.

13. $\frac{8}{3}S\sqrt{\frac{S}{\pi}}\sin^{6}\frac{\alpha}{2}$; $a)S=e^{1,11}, \alpha=\frac{7}{11}\pi$; б) $S=5,403, \alpha=28^{\circ}$.

ВІДПОВІДЬ : а) $2.8187e+000$; б) $3.7879e-003$.

14. $2\sqrt{\frac{F}{\pi}}tg\alpha\sin^{2}\frac{\alpha}{2}$; а) $F=\frac{1}{0,03}, \alpha=\frac{5}{7}\pi$; б) $F=\ln 7, \alpha=1,34^{\circ}$.

ВІДПОВІДЬ : а) $-6.6313e+000$; б) $5.0346e-006$.

15. $\frac{1}{12}\cdot\frac{m^{3}\cos\alpha}{(\sin\alpha+\cos\alpha)^{3}}$; а) $m=-20,1, \alpha=20^{\circ}$; б) $m=\lg 13,6, \alpha=1,48$.

ВІДПОВІДЬ : а) $-3.0201e+002$; б) $8.5792e-003$.

16.

$\frac{\sqrt{3h^{3}}}{\cos^{2}\alpha}\sin(\alpha+30^{\circ})\sin(\alpha-30^{\circ}$ а) $h=0,28; \alpha=41^{\circ}$; б) $h=e^{0,415}; \alpha=237^{\circ}$.

ВІДПОВІДЬ : а) $8.1284e-002$ б) $4.9334e+000$.

17. $\frac{\alpha}{3}\left(\lg(d+2)-tg\alpha\right)^{2}$; а) $d=10,6; \alpha=50^{\circ}$; б) $d=e^{2,3}; \alpha=1$.

ВІДПОВІДЬ : а) $3.5028e-002$; б) $1.4003\,\mathrm{e}+000$.

18. $d^{3}ctg\alpha\sqrt{\sin^{4}\alpha-\cos^{4}\alpha}$; а) $d=6,178; \alpha=20^{\circ}$; б) $d=2,2461\cdot 10^{-2}; \alpha=1.146$.

ВІДПОВІДЬ : а) $4.1645e+002$; б) $4.1645e+002$;

Контрольні запитання

1. Як подаються дійсні числа при обчисленнях в системі MATLAB?

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

3. Яким чином оголошуються змінні в мові MATLAB?

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

5. Яку роль грає системна змінна ans?

6. Які є системні змінні у MATLAB? Що вони позначають і як їх використовувати ?

7. Як повернути в командний рядок раніше введену команду?

8. Які вбудовані елементарні математичні функції є у MATLAB? Які з них відсутні у всіх інших мовах високого рівня?

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

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

КОМП’ЮТЕРНИЙ ПРАКТИКУМ № 1.2

Операції з комплексними числами

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

Теоретичні відомості

Введення комплексних чисел

Для позначення уявної одиниці у мові MATLAB зарезервовано два імені – «і» 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.

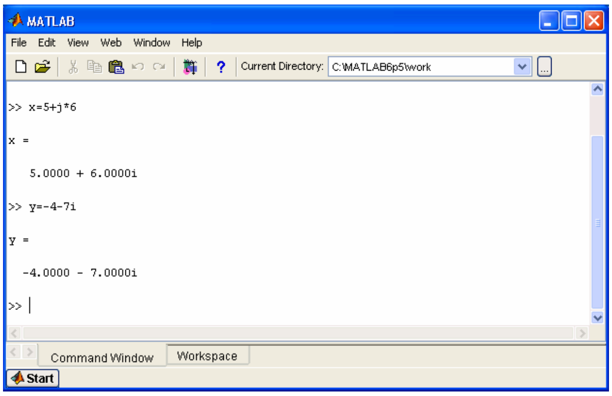

Рис. 2.1 Вікно з прикладом введення комплексного числа

Найпростіші дії з комплексними числами – додавання, віднімання, множення, ділення та піднесення до степеня – здійснюється за допомогою звичайних арифметичних знаків – «+», «-», «*», «/», «\» та «^», відповідно.

Приклади застосування наведені на рис. 2.2.

Примітка. У наведеному фрагменті використано функцію ***disp*** (від слова «дисплей»), яка дозволяє також виводити до командного вікна результати обчислень або деякий текст. При цього числовий результат, як видно, виводиться вже без зазначення імені змінної чи імені ***ans***.

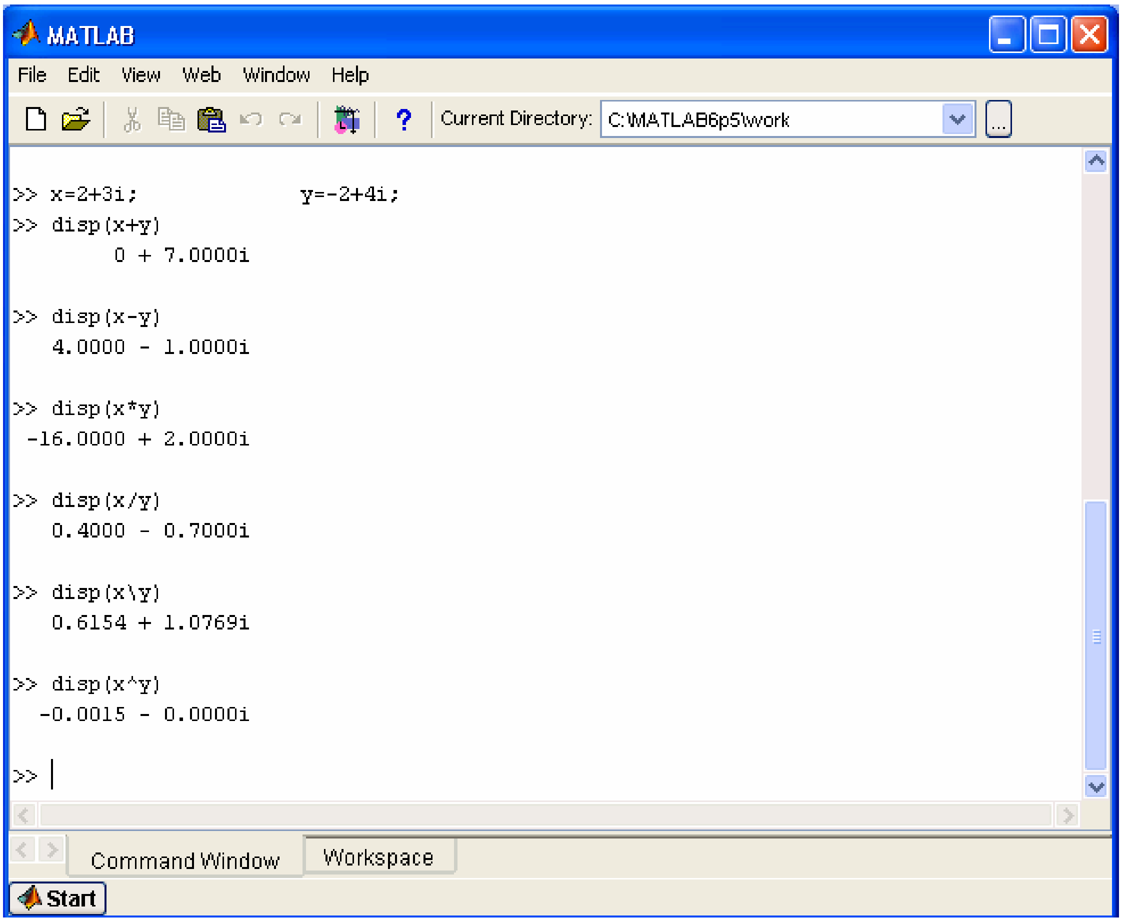

Рис. 2.2 Вікно з прикладом дії з комплексними числами

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.

Завдяки цьому, наприклад, функція ***sqrt*** обчислює, на відмінну від інших мов програмування, квадратний корінь з від'ємного аргументу, а функція abs при комплексному значенні аргументу обчислює модуль комплексного числа. Приклади наведені на рис.2.3.

Існують кілька додаткових функцій, які розраховані лише на комплексний аргумент:

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

imag(Z) – відокремлює уявну частину комплексного аргументу;

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

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

Для оформлення отриманих даних, доцільно використовувати функцію num2str, що перетворює числове значення параметра в символьний рядок (`str = num2str(A)`). Також часто використовується символ %.

% (Знак відсотка) – 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);

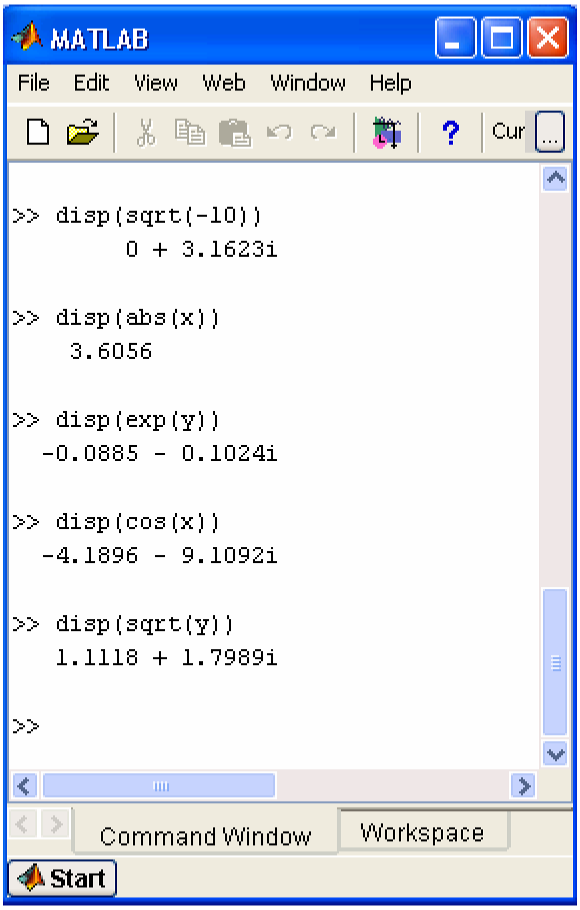

Рис. 2.3 Вікно з прикладом використання стандартних математичних функцій

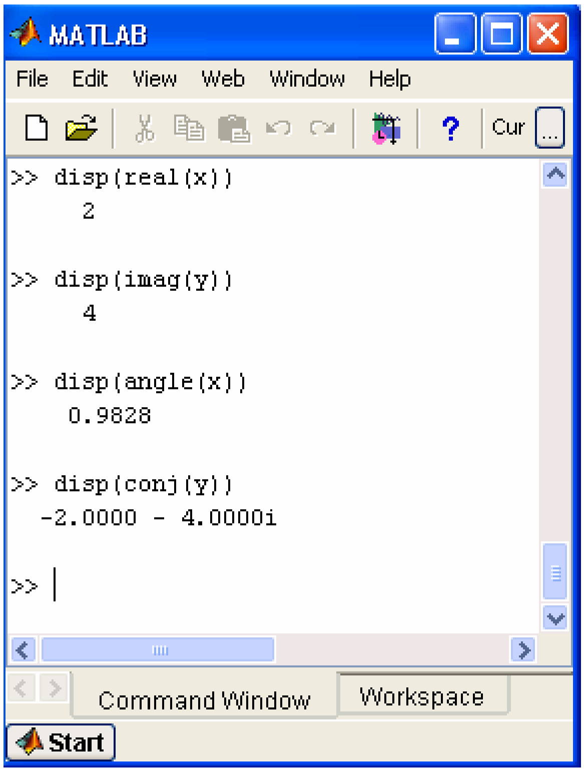

Рис. 2.4 Вікно з прикладом використання функцій для роботи з комплексними числами

Завдання до виконання комп'ютерного практикуму

Завдання. Виконайте такі дії (див. таблицю 2.1):

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 формах, причому аргумент результату забезпечте в межах між (-π) і +π.

Вказівка. Внаслідок складності і значної кількості дій для виконання цього завдання складіть програму у вигляді М-файлу.

Табл.2.1 Варіанти індивідуальних завдань

Варіант	Комплексне число				Вираз
	z_1	z_2	z_3	z_4	
1	2	3	4	5	6
1	$4+3i$	$2,71e^{i\pi/12}$	$1,82e^{-1,2i}$	$\sqrt{3}-2i$	$z_1^2 \cdot z_2 : z_3 + z_4$
2	$0,8-2i$	$3,08e^{i7\pi/12}$	$8,01e^{2i}$	$-5+\sqrt{2}i$	$z_1^2 : z_2 + z_3 - z_4$
3	$-0,7+4i$	$1,74e^{i0,3\pi}$	$3+4i$	$2,1e^{-2,3i}$	$\sqrt{z_1 : z_2} \cdot z_3 + z_4$
4	$-3-2i$	$3,21e^{15^0 i}$	$1,2+3i$	$2,71e^{-78^0 i}$	$\sqrt{z_1 \cdot z_2} : z_3 + z_4$
5	$2,71e^{i\pi/12}$	$-0,7+4i$	$1,31e^{-i5\pi/12}$	$-8-3i$	$\sqrt{z_1 : z_2} \cdot z_3 - z_4$
6	$3,08e^{i7\pi/12}$	$-3-2i$	$2.03e^{i14/13}$	$\sqrt{3}+\sqrt{2}i$	$\left(z_1 + z_2\right)^4 \cdot z_3 : z_4$
7	$1,74e^{0,3i\pi}$	$0,8-2i$	$3,28e^{-1,2i}$	$\sqrt{3}-\sqrt{2}i$	$(\sqrt{z_1} + z_2) \cdot z_3 : z_4$
8	$3,21e^{15i}$	$4+3i$	$\sqrt{3}-4i$	$1,23e^{111^0 i}$	$\left(z_1 - z_2\right)^3 \cdot z_3 + z_4$
9	$1+i\pi/2$	$1,2e^{107^0 i}$	$0,8-2i$	$2,5e^{14^0 i}$	$\left(z_1 : z_2 + z_3\right)^3 \cdot z_4$
10	$\sqrt{5}-i$	$0,7e^{1,7i}$	$1,2e^{0,9i}$	$-3-2i$	$(z_1 : z_2 - z_3)^2 - z_4$
11	$0,187-3,94i$	$0,3e^{-107^0 i}$	$-0,7+4i$	$1,5e^{23^0 i}$	$\sqrt[3]{z_1 + z_2 - z_3} \cdot z_4$
12	$-1+\sqrt{5}i$	$2,1e^{211^0 i}$	$0,4e^{32^0 i}$	$4+3i$	$\sqrt[3]{z_1} \cdot z_2 : z_3 + z_4$

1	2	3	4	5	6
13	$-\sqrt{3}-4i$	$1,25e^{-0,8i}$	$-3-2i$	$0,75e^{0,7i}$	$(\sqrt[3]{z_1\cdot z_2}+z_3):z_4$
14	$1,2e^{1,7i}$	$0,18-3,9i$	$0,71e^{4i}$	$0,8-2i$	$(\sqrt[3]{z_1:z_2}-z_3)\cdot z_4$
15	$0,3e^{-97^0 i}$	$-1+\sqrt{5}i$	$-0,7+4i$	$5,2e^{71^0 i}$	$(\sqrt{z_1\cdot z_2}-z_3):z_4$
16	$1,25e^{0,6i}$	$-\sqrt{3}-4i$	$4+3i$	$2,5e^{3,8i}$	$(\sqrt{z_1:z_2}-z_3)\cdot z_4$
17	$1,05e^{-0,4i}$	$\sqrt{5}-i$	$2,7e^{0,8i}$	$-0,7+4i$	$\sqrt{(z_1:z_2+z_3)\cdot z_4}$
18	$2,1e^{73^0 i}$	$1+i\pi/2$	$\sqrt{2}+\sqrt{3}i$	$1,93e^{192^0 i}$	$\sqrt{(z_1\cdot z_2-z_3):z_4}$
19	$2,7+0,8i$	$2e^{-\sqrt{3}i}$	$0,81e^{i\pi/7}$	$-\sqrt{2}-\sqrt{3}i$	$\sqrt{(z_1+z_2):z_3\cdot z_4}$
20	$-0,8+2,7i$	$-2e^{\sqrt{3}i}$	$0,9e^{i5\pi/7}$	$3,1-2,1i$	$\sqrt{(z_1+z_2)\cdot z_3:z_4}$
21	$-1,1-3,2i$	$0,33e^{-1,9i}$	$2e^{\sqrt{2}i}$	$2,08+i$	$\sqrt{z_1-z_2}\cdot z_3:z_4$
22	$2,1-3,2i$	$0,68e^{148^0 i}$	$\sqrt{5}+\sqrt{2}i$	$2,73e^{23^0 i}$	$\sqrt{z_1-z_2}:z_3\cdot z_4$
23	$1,1e^{-0,8i}$	$\sqrt{5}-2i$	$-1,7+i$	$0,97e^{\sqrt{2}i}$	$((z_1+z_2)^2-z_3):z_4$
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	$1,1e^{-2,1i}$	$\pi/8-2,1i$	$2,71+0,4i$	$1,71e^{-\sqrt{3}i}$	$(z_1-z_2)^3:z_3+z_4$

Контрольні запитання

1. Як увести значення комплексного числа й у якому виді воно виведеться на екран ?

2. Як мовою MATLAB забезпечити додавання, віднімання, множення, ділення й піднесення до степеня комплексних чисел ?

3. Які функції роботи з комплексними числами є в мові MATLAB?

4. Як складаються програми мовою MATLAB, як вони записуються, як викликаються до виконання?

5. Які правила написання тексту програми мовою MATLAB? Як ввести коментарі у текст програми MATLAB? Яку роль відіграють закоментовані перші рядки програми MATLAB?

6. Яку функцію виконує команда ***disp***? Як нею користуватися? В яких випадках нею доцільно користуватися?

7. Як за допомогою функції ***disp*** вивести у командне вікно значення визначених змінних?

8. Як за допомогою функції ***disp*** вивести у командне вікно одночасно і значення змінних, і деякий супроводжуючий їх текст?

9. За допомогою яких функцій можна перетворити числове значення деякої змінної у символьний рядок, що містить це числове значення?

10. За якою формулою визначається комплексне число, якщо задані значення його модуля і аргументу?

11. Як відображується на комплексній площині комплексне число? його дійсна частина? його уявна частина? його модуль? його аргумент? Як пов'язані між собою уявна і дійсна частина, з одного боку, і модуль і аргумент – з іншого боку?

12. З допомогою якої функції можна знайти модуль комплексного числа?

13. З допомогою якої функції можна знайти аргумент комплексного числа?

14. Які елементарні вбудовані математичні функції MATLAB можуть використовувати як аргументи комплексні числа?

15. Чи є у MATLAB функції, які призначені для роботи лише з комплексними числами? Назвіть їх і охарактеризуйте.

КОМП’ЮТЕРНИЙ ПРАКТИКУМ № 1.3

Операції з векторами

Мета роботи: навчитися здійснювати дії з векторами у командному вікні MATLAB.

Теоретичні відомості

Початкові значення векторів можна задавати 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, введення рядка «$V = \left[1.2 \quad -0.3 \quad 1.2e-2\right]$» задає вектора V, що містить три елементи зі значеннями 1.2, -0.3 та 1.2e-2:

Довгий вектор можна вводити частинами, які потім з’єднувати за допомогою операції поєднання векторів

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

Мова MATLAB надає користувачеві можливість скороченого введення вектора, значення елементів якого є арифметичною прогресією. Якщо позначити: nz – початкове значення цієї прогресії, kz – кінцеве значення цієї прогресії, h – різниця прогресії (крок), то вектор можна ввести за допомогою короткого рядка:

$$V = nz : h : kz .$$

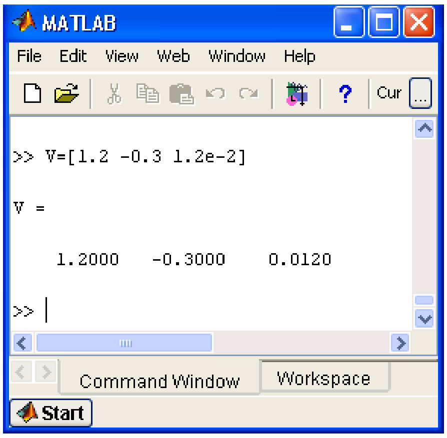

Рис. 3.1 Вікно з прикладом введення вектора у командному вікні MATLAB

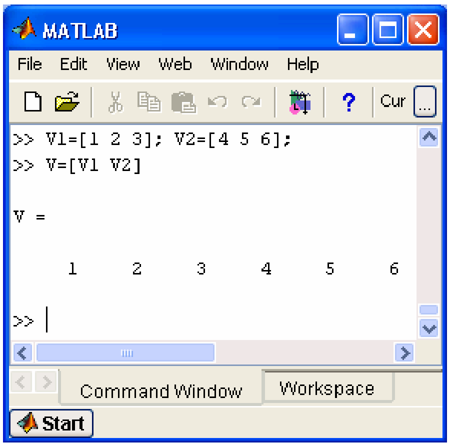

Рис. 3.2 Вікно з прикладом використання операції поєднання векторів

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

Дії над векторами

Розрізняють дві суттєві різні групи дій над векторами:

- векторні дії – такі, що дозволяють векторним обчисленням у математиці;

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

Векторні дії над векторами:

- Додавання векторів. Як відомо, додаватися вектори можуть лише вектори однакового типу (тобто обидва є або векторами-рядками, або векторами-стовпцями), що мають однакову довжину (тобто мають однакову кількість елементів).

- Віднімання векторів здійснюється за допомогою арифметичного оператора «-».

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

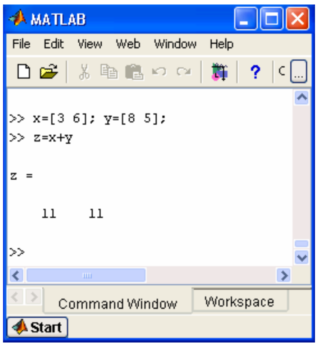

Рис.3.3 Вікно з прикладом додавання векторів

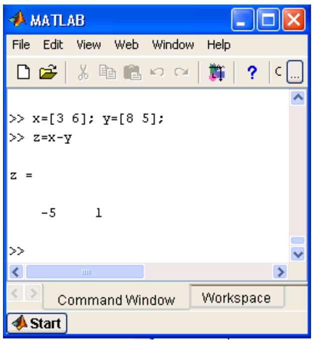

Рис. 3.4 Вікно з прикладом віднімання векторів

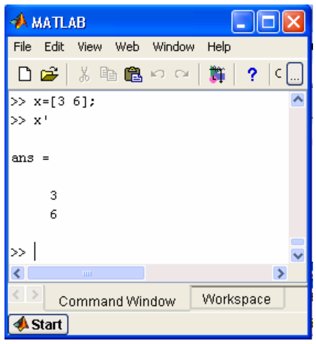

Рис. 3.5 Вікно з прикладом транспонування вектора

- Множення вектора на число здійснюється за допомогою знака арифметичного множення «*».

- 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: $Z = X'*Y$ та $Z = X*Y'$. Причому у першому випадку результатом є квадратна матриця, а у другому – число.

- Для трикомпонентних векторів у MATLAB існує функція ***cross***, яка дозволяє знайти векторний добуток двох векторів.

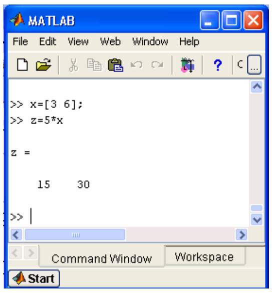

Рис. 3.6 Вікно з прикладом множення вектора на число

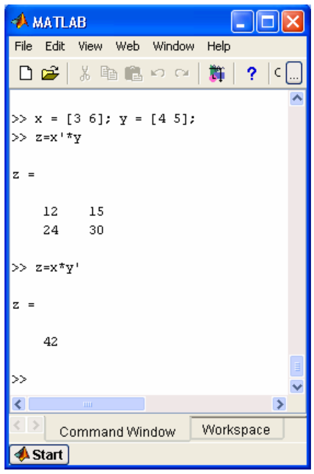

Рис. 3.7 Вікно з прикладом множення векторів

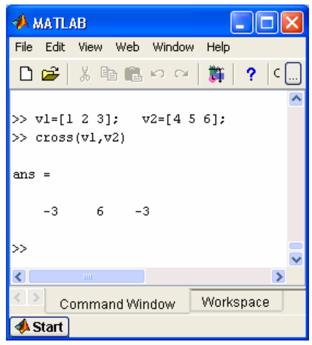

Рис. 3.8 Вікно з прикладом знаходження векторного добутку

<u>Дії з перетворюванням векторів:</u>

У мові MATLAB є припустимим 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 «.^».

Наприклад:

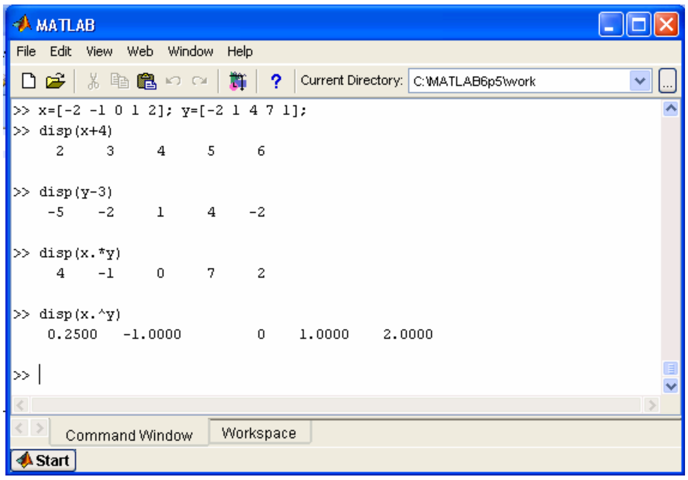

Рис.3.9 Вікно з прикладом з операцій поелементного перетворення векторів

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. 3.1) за точними формулами і користуючись функцією ***roots***. Результати порівняйте.

Табл. 3.1 Варіанти індивідуальних завдань

Вар.	a	b	c	Вар.	a	b	c
1.	0.56	1.2e-4	4.08	14.	2.3	7.9	324
2.	1	0.1	100	15.	1	0.02	16.57
3.	4.2e-3	8.03e-4	1.06	16.	1.3	0.56	18.8
4.	7.1e3	9.4e4	8.3e10	17.	0.13	0.056	18.8
5.	5.09	4.32	256	18.	17	12	956
6.	8.3	5.34	693	19.	0.085	1	1.3e3
7.	27	27	1276	20.	1.2	0.32	15
8.	3.08	0.2	30	21.	7.1	6.4	256
9.	5.3	10.6	876	22.	0.2	0.002	2.9
10.	0.45	0.034	121	23.	1.4e-3	3.9	2.6e2
11.	4.3	10.7	3.4e3	24.	0.86	3.2	5.4e2
12.	13	0.8	287	25.	7.3e3	8.2e2	3.5e8
13.	6.035	5.2	875				

Завдання 2. Обчисліть значення функції $f(x)$ на відрізку [a; b] із кроком h у відповідності з табл. 3.2.

Табл. 3.2 Варіанти індивідуальних завдань

Варіант	$f(x)$	a	b	h
1	2	3	4	5
1	$\dfrac{x^2}{1+0,25\sqrt{x}}$	1,1	3,1	0,2
2	$\dfrac{x^3-0,3x}{\sqrt{1+2x}}$	2,05	3,05	0,1
3	$\dfrac{2e^{-x}}{2\pi+x^3}$	0	1,6	0,16

1	2	3	4	5
4	$\dfrac{\cos \pi x^2}{\sqrt{1-3x}}$	-1	0	0,1
5	$\sqrt{1+4x}\sin \pi x$	0,1	0,8	0,07
6	$\dfrac{e^{x/3}}{1+x^2}$	1,4	2,4	0,1
7	$e^{-2x}+x^2-1$	0,25	2,25	0,2
8	$(e+x)\sin(\pi\sqrt{x-1})$	1,8	2,8	0,1
9	$\sqrt{3+2x}\cdot tg\dfrac{\pi x^3}{2}$	0,1	0,9	0,08
10	$\sqrt{2+3x}\cdot \ln(1+3x^2)$	-0,1	0,9	0,1
11	$\sqrt[3]{x^2+3}\cdot \cos\dfrac{\pi x}{2}$	1	2,5	0,15
12	$(4+7x)\sin(\pi^3\sqrt{1+x})$	0	7	0,7
13	$e^{-x^2}(1+3x-x^2)$	0	2	0,2
14	$x^3-3x+\dfrac{8}{\sqrt{1+x^2}}$	0	1,7	0,17
15	$\sqrt{sh\sqrt{2\pi x}},\ \left(sh\ x=\dfrac{e^x-e^{-x}}{2}\right)$	0	1,2	0,12
16	$\sqrt{ch\dfrac{x}{\sqrt{2x}}},\ \left(ch\ x=\dfrac{e^x+e^{-x}}{2}\right)$	0,5	1,5	0,1
17	$\dfrac{x^3+2x}{\sqrt{1+e^x}}$	-0,2	0,8	0,1
18	$\sqrt{1+2x^2}\cdot \sin\dfrac{3x}{2}$	2	4	0,2

1	2	3	4	5
19	$\sqrt{3x^2+5}\cdot\cos\frac{\pi x}{2}$	0,5	1,5	0,1
20	$\arccos\ e^{-\sqrt[3]{3x}}$	0,2	0,5	0,03
21	$\arcsin\ e^{-x^2/5}$	8	13	0,5
22	$x+\ln(x+\sqrt{1+x^2})$	-0,5	0,5	0,1
23	$\frac{1+e^{-x/2}}{\sqrt{3x^2+1}}$	3	5	0,2
24	$3x^3+\frac{1}{x}+e^{-2x^2}$	1,2	2,2	0,1
25	$x^{2x+1}+x^3-2x$	1	5	0,4

Контрольні запитання

1. Як вводяться вектори в мові MATLAB? Якими функціями можна формувати вектори в мові MATLAB?

2. Які функції MATLAB дозволяють перетворювати вектор поелементно?

3. За допомогою яких засобів у MATLAB здійснюються основні операції з векторами?

4. Як здійснити скалярний добуток двох векторів? Які вектори можуть бути скалярно перемножені?

5. Як здійснити векторний добуток двох векторів? Які вектори можуть бути векторно перемножені?

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 1.4

Операції з матрицями

Мета роботи: навчитися виконувати операції з матрицями у командному вікні MATLAB. Ознайомитись з вбудованими у MATLAB стандартними функціями для роботи з матрицями.

Теоретичні відомості

Формування матриць

Функції, які забезпечують утворення деяких найпоширеніших видів матриць:

- ***zeros(M,N)*** – утворює матрицю розміром $(M*N)$ з нульовими елементами;
- ***ones(M,N)*** – утворює матрицю розміром $(M*N)$ з одиничними елементами;
- ***eye(M,N)*** – утворює матрицю розміром $(M*N)$ з одиницями у головній діагоналі та рештою нульових елементів;
- ***rand(M,N)*** – утворює матрицю розміром $(M*N)$ з випадковими числами, які рівномірно розподілені у діапазоні від 0 до 1;
- ***randn(M,N)*** – утворює матрицю розміром $(M*N)$ з випадковими числами, які рівномірно розподілені за нормальним законом з нульовим математичним сподіванням 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 $m*n$;
- ***tril(A)*** – утворює нижню трикутну матрицю на основі матриці A шляхом обнулювання її елементів вище головної діагоналі;
- ***triu(A)*** – утворює верхню трикутну матрицю на основі матриці A шляхом обнулювання її елементів нижче головної діагоналі.

Дії з матрицями

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

Базові дії з матрицями – додавання, віднімання, транспортування, множення матриці на число, множення матриці на матрицю, піднесення матриці до цілого степеня – 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.

Приклади додавання та віднімання:

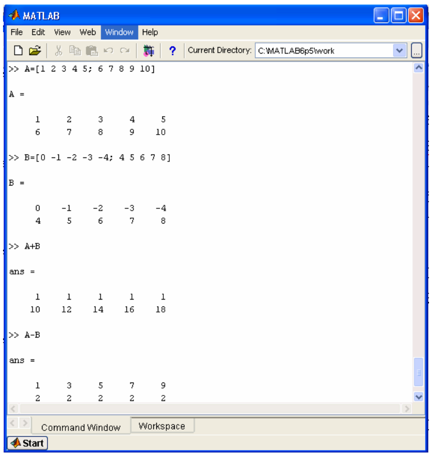

Рис. 4.1 Вікно з прикладом додавання та віднімання матриць

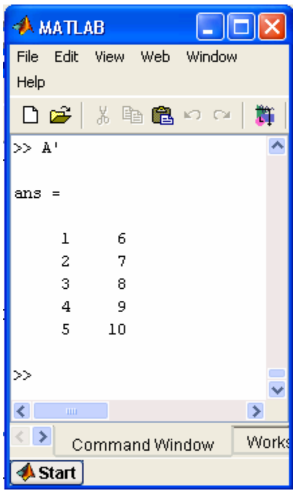

Рис. 4.2 Вікно з прикладом транспортування матриць

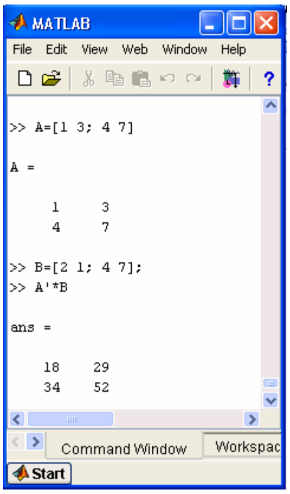

Рис. 4.3 Вікно з прикладом помноження матриці на число

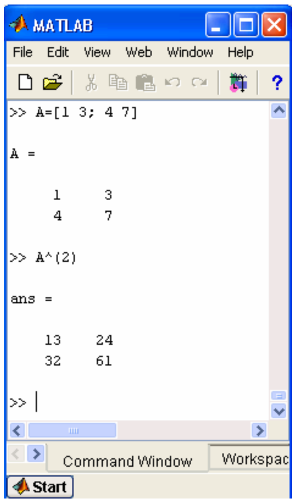

Рис. 4.4 Вікно з прикладом піднесення матриці до степеня

Розглянемо більш складні операції з матрицями:

- **Обчислення визначника матриці** здійснюється за допомогою функції ***det***(A). Матриця повинна бути квадратна.

- **Функції 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.

- **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.

Операція B/A є рівносильною послідовності дій $B*inv(A)$. Її зручно використовувати для розв'язання матричного рівняння:

$$X*A=B.$$

Аналогічна операція $A\backslash B$ є рівносильною сукупності операцій $inv(A)*B$, що є розв'язком матричного рівняння:

$$A*X=B.$$

- **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.

- Функція ***sqrtm(A)*** обчислює таку матрицю Y, що $Y*Y=A$;

- Функція ***diag(x)*** формує або витягає діагональ матриці: якщо ***x*** – вектор, то функція утворює квадратну матрицю з вектором ***x*** на головній діагоналі;

- Функція ***trace(A)*** – знаходить слід матриці (сума діагональних елементів матриці A);

- **Визначення характеристичного полінома матриці** A розміром m на n можна здійснити за допомогою процедури ***poly***(A);

- **Обчислення власних значень і власних векторів матриці** здійснює процедура ***eig***(A);

- **Сингулярне розкладання матриці** здійснює процедура ***svd***(A).

Операції з матрицею як таблицею даних (обробка даних вимірювань)

До таких операцій належать такі функції:

- ***size***(A) – визначає розмір матриці A;

- 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;

- функція ***sort*** – сортує елементи кожного з стовпців початкової матриці. Результатом є матриця того ж розміру;

- функції ***sum*** і ***prod*** – формують вектор, кожний елемент якого є сумою або добутком елементів відповідного стовпця початкової матриці;

- 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;

- функція ***diff*** – утворює з заданої матриці розміром $(m*n)$ матрицю розміром $\left((m-z)*n\right)$, елементи якої утворюються як різниця між суміжними рядками початкової матриці.

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

Функція k=***norm***(A,p) обчислює p-норму матриці A за формулою

k=***max***(***sum***(***ab***s(A).^p))^(1/p) ,

де p – ціле додатне число або inf або 'fro'. Якщо аргумент p не вказаний, обчислюється 2-норма. При цьому є слушними співвідношення:

norm(A,1)=***max***(***sum***(***abs***(A)))

norm(A,inf)=***max***(***sum***(***abs***(A')))

norm(A,'fro')=***sqrt***(***sum***(***diag***(A'*A)))

norm(A)=***norm***(A,2)= $\sigma_{\max}$ *(A)*

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

Процедура k=***cond***(A) обчислює число обумовленості матриці A по відношенню до операції обернення. Воно дорівнює відношенню максимального сингулярного числа матриці до мінімального.

k=***rcond***(A) обчислює величину, зворотну до значення числа обумовленості матриці A відносно її 1-норми. Якщо матриця A є добре обумовленою, значення k близько до одиниці, якщо ж вона є погано обумовленою, k наближається до нуля.

Завдання до виконання комп'ютерного практикуму

Завдання 1. Здійсніть наступні операції:

1. Введіть довільну матрицю розміром (4*6). Знайдіть суму найбільших елементів її рядків;

2. Введіть квадратну матрицю (5*5) з одним найменшим елементом. Знайдіть суму елементів рядка, у якому розміщений елемент із найменшим значенням.

3. Введіть матрицю (6*9), у якій є єдині найбільший і найменший елементи і вони розташовані в різних рядках. Поміняйте місцями рядок, що містить найбільший елемент, і рядок із найменшим елементом.

4. Введіть матрицю (5*6) із різними значеннями елементів. У кожному рядку виберіть елемент із найменшим значенням, з отриманих чисел виберіть найбільше. Знайдіть індекси отриманих елементів.

5. Введіть матрицю (5*6). Знайдіть вектор, елементами якого є найбільші елементи відповідного рядка матриці.

6. Введіть матрицю (5*6). Побудуйте вектор, елементами якого є суми найбільшого й найменшого елементів відповідного рядка матриці.

7. Введіть матрицю (5*6). Побудуйте вектор, елементами якого є середні значення елементів відповідного рядка матриці.

8. Введіть матрицю (5*6). Побудуйте вектор, елементами якого є середньоквадратичні відхилення елементів відповідного рядка матриці від їхнього середнього значення.

9. Введіть матрицю (5*6). Побудуйте вектор, елементами якого є середні арифметичні найбільшого й найменшого елементів відповідного рядка матриці.

10. Введіть матрицю (6*5). Побудуйте вектор, елементами якого є суми квадратів елементів відповідного стовпчика матриці.

11. Введіть матрицю (5*5). Побудуйте вектори, елементами яких є суми елементів стовпчиків матриці, добутки елементів стовпчиків і найменші значення елементів стовпчиків.

12. Введіть матрицю (5*6). Знайдіть середнє арифметичне найбільшого й найменшого її елементів.

13. Введіть матрицю (5*5). Побудуйте вектор, елементами якого є елементи головної діагоналі матриці. Знайдіть слід матриці.

14. Введіть дві матриці (4*4). Побудуйте нову матрицю розміром (4*8), включаючи в перші 4 стовпчика рядки першої матриці, а в інші – стовпчики другої матриці.

15. Знайдіть суму всіх елементів матриці розміром (4*3).

Завдання 2. Уведіть довільну матрицю розміром (5*5). Знайдіть:

1) визначник матриці; якщо визначник дорівнює нулю, або занадто малий, змініть деякі елементи матриці і повторіть обчислення;

2) обернену матрицю; перевірте слушність шляхом обернення оберненої матриці;

3) характеристичний поліном матриці;

4) корені характеристичного полінома матриці; упорядкуйте корені по комплексно-спряжених парах і за величиною;

5) власні значення матриці; порівняйте з раніше знайденими коренями характеристичного полінома;

6) LU-розкладання матриці; перевірте його слушність;

7) QR-розкладання матриці; перевірте його слушність;

8) сингулярні числа матриці; порівняєте їх з одержуваними при ***svd***-розкладанні;

9) слід матриці;

10) число обумовленості матриці;

11) експоненту від матриці;

12) логарифм від експоненти матриці; порівняйте з вихідною матрицею.

Контрольні запитання

1. Як вводяться матриці у системі MATLAB?

2. Які дії над матрицями передбачені у системі MATLAB?

3. Які функції є у MATLAB задля формування матриць визначеного виду?

4. Як сформувати матрицю:

а) по заданих векторах її рядків?

б) по заданих векторах її стовпців?

в) по заданих векторах її діагоналей?

5. Які функції поелементного перетворення матриці є у MATLAB?

6. Які функції забезпечують обробку експериментальних даних? Що потрібно зробити з експериментальними даними, щоб можна було застосувати ці функції і в якій формі виводяться результати обробки?

7. Як обчислити у MATLAB визначник матриці?

8. Як обчислити у MATLAB обернену матрицю?

КОМП’ЮТЕРНИЙ ПРАКТИКУМ № 1.5

Розв’язок системи лінійних алгебраїчних рівнянь

Мета роботи: навчитися розв’язувати систему лінійних рівнянь за допомогою вбудованих функцій MATLAB.

Теоретичні відомості

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

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

Тут позначено: x_i (i=1,2,..., n) – деякі змінні, a_{ij} $(i,\ j\ =\ 1,2,...,n)$ – коефіцієнти при змінних, b_{ij} $(i, j = 1,2,...,n)$ – так звані "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:

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

то система рівнянь (1) може бути поданою у матричній формі у такий спосіб:

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

Розв'язування СЛАР на ЕОМ потребує наявності ряду досить складних процедур з оперування й перетворення матриць. У більшості універсальних мов програмування такі процедури відсутні. Точніше, вони створені математиками й програмістами і знаходяться у спеціальних бібліотеках процедур, розроблених на відповідних мовах. Але зазвичай такі бібліотеки є недосяжними для пересічного користувача.

Тому, якщо у вашому розпорядженні є такі мови як Pascal, Delphi або Basic, то для розв'язування СЛАР потрібно утворити цілу низку підпрограм, які б реалізували основні операції з матрицями, починаючи з уведення, формування й виведення матриць і закінчуючи процедурами обчислення визначника матриці і зворотної матриці. Таку серію процедур зручніше за все розробляти мовою Pascal, яка дозволяє утворювати нові обчислювальні об'єкти на основі використання даних типу запису (***record***).

Приємним виключенням є система MATLAB. В ній уже передбачені усі необхідні процедури й функції матричної математики. Особливо зручним є те, що головні дії з матрицями здійснюються на мові MATLAB за допомогою відомих звичних знаків арифметичних дій: «+» – додавання; «-» – віднімання; «*» –множення; «^» – піднесення до ступеня.

Транспонування матриць (заміна рядків і стовпців місцями) здійснюється у MATLAB за допомогою знаку апострофа, розміщеного одразу після ймення тієї матриці, яку потрібно транспонувати.

Найбільш цікавими для нас є дві матричні операції, яких немає у матричній алгебрі - ділення матриць зліва направо (*A/B*) і справа наліво (*A\B*). Перша з них є рівносильною до сукупності операцій $A \cdot B^{-1}$, друга - $A^{-1} \cdot B$. Саме ці операції дозволяють відшукати розв'язки СЛАР.

Якщо задана СЛАР має вигляд (3):

$$A \cdot X = B,$$

то його розв'язок у матричній формі можна записати мовою MATLAB у такий спосіб:

$$X = A \setminus B. \qquad (5.4)$$

Якщо ж СЛАР записана у наступній матричній формі

$$X \cdot A = B,$$

то розв'язок її у MATLAB відшукується записом команди

X = A /B.

Приклад. Застосуємо засоби системи MATLAB для *розв'язку СЛАР:*

$$\begin{cases} x_1 + x_2 - x_3 = 6 \\ -2x_1 + 3x_2 = 4 \\ 4x_1 + 15x_2 + 2x_3 = 28 \end{cases}$$

Якщо, знаходячись у командному вікні MATLAB, ввести значення елементів матриці коефіцієнтів командою

```
A = [1 1 -1; -2 3 0; 4 15 2]
```

то отримуємо у тому ж вікні

```
A =
     1     1    -1
    -2     3     0
     4    15     2
```

Аналогічно введемо вектор-стовпець вільних членів

```
B = [6 4 28]'
```

Отримуємо

```
B =
     6
     4
    28
```

Тепер можна розпочати відшукання розв'язків системи (5.4):

X=A\B

У вікні виникне відповідь

```
X =
                              1
                              2
                             -3
```

Для перевірки, помножимо матрицю A на одержаний вектор X розв'язків

A*X

```
ans =
     6
     4
    28
```

Як бачимо, ми отримали вектор вільних членів, що підтверджує слушність проведених обчислень.

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 відшукування коренів СЛАР, які відносяться до наближених. Наприклад, різні модифікації методу ітерацій. Вони находять за скінченну кількість кроків (ітерацій) лише наближені розв’язки із заданою припустимою відносною похибкою.

Метод ітерацій взагалі полягає у тому, що попередньо первісне рівняння $f(x)=0$ перетворюється до виду

$$x=\varphi(x). \tag{5.5}$$

З області ізоляції $[a,b]$ шуканого кореня обирається деяке певне значення x_0 аргументу, яке приймається за початкове наближення кореня. Наближені значення кореня у наступних наближеннях визначаються з співвідношення

$$x_k=\varphi(x_{k-1}); \qquad (k=1,2,...,n), \tag{5.6}$$

при цьому k – це номер ітерації (наближення).

Очевидно, ітераційний процес може приводити до послідовності значень, які наближаються до деякого значення аргументу (шуканого кореня), і тоді цей процес є стійким. Але зазначений процес може приводити до послідовності значень аргументу, які все далі віддаляються від первісного значення, тобто бути нестійким. Стійкість або нестійкість ітераційного процесу суттєво залежить від виду залежності $\varphi(x)$, яка, як неважко впевнитися визначається неоднозначно, і тому можна підібрати з можливих її варіантів і такі, які забезпечують стійкість ітерацій. Саме це й є найбільш складним у побудові ітераційного процесу.

Методи ітерацій можуть бути застосовані і для відшукування коренів СЛАР, якщо попередньо систему (1) подати у виді

$$\begin{cases} x_1 = \varphi_1(x_1, x_2, ..., x_n) \\ x_2 = \varphi_2(x_1, x_2, ..., x_n) \\ .... \\ x_n = \varphi_n(x_1, x_2, ..., x_n) \end{cases} \tag{5.7}$$

MATLAB має кілька вбудованих процедур, які дозволяють розв'язувати систему лінійних алгебричних рівнянь виду (3) 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 відхилу;

qmr – метод квазімінімального відхилу;

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

Загальне звернення до цих процедур має вигляд

x = ***bicg***(A,B,tol,maxit),

де A – квадратна матриця розміром ($n \times n$) коефіцієнтів при аргументах системи рівнянь (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 (1).

За початкове наближення обирається вектор x0 з нульових елементів. Наведемо приклад. Введемо наступну послідовність операторів:

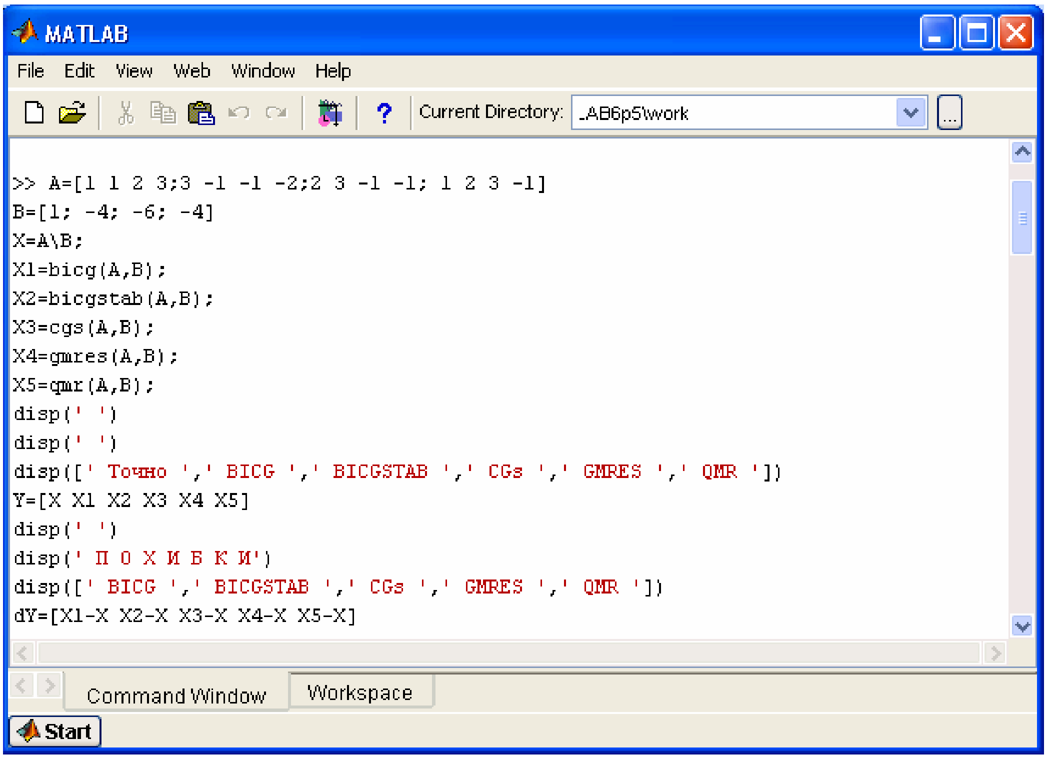

Рис. 5.1 Вікно з прикладом використання вбудованих процедур MATLAB для рішення СЛАР

Виконавши їх, система MATLAB виведе результат у наступному вигляді:

```
A =
     1     1     2     3
     3    -1    -1    -2
     2     3    -1    -1
     1     2     3    -1
B =
     1
    -4
    -6
    -4
```

```
 Точно  BICG  BICGSTAB  CGs  GMRES  QMR
Y =
                   -1        -0.99999999999999        -1.00000000000465        -1.00000000000007                       -1        -1.00000000000003
                   -1        -1.00000000000002        -0.999999999973917       -0.999999999999998                      -1        -0.999999999999992
                    0     3.56381590904675e-014    3.89563659108916e-011   -5.89528426075958e-014   -1.66533453693773e-016   -4.10782519111308e-015
                    1         1.00000000000005         1.0000000000201         0.999999999999972                        1         0.99999999999998

 П О Х И Б К И
 BICG  BICGSTAB  CGs  GMRES  QMR
dY =
    1.04360964314765e-014    -4.64850380410553e-012    -7.41628980449605e-014     2.22044604925031e-016    -3.15303338993544e-014
   -1.73194791841524e-014     2.60828025844262e-011     2.55351295663786e-015     4.44089209850063e-016     8.54871728961371e-015
    3.56381590904675e-014     3.89563659108916e-011    -5.89528426075958e-014    -1.66533453693773e-016    -4.10782519111308e-015
    4.84057238736568e-014     2.01032523960976e-011    -2.8421709430404e-014                         0    -2.04281036531029e-014
```

Завдання до виконання комп'ютерного практикуму

Завдання. Відшукайте розв'язки системи лінійних алгебричних рівнянь, заданої у таблиці 5.1: а) одним з методів точного розв'язування; б) методами ітерації. Порівняйте результати.

Табл.5.1 Варіанти індивідуальних завдань

Вар.	A	B	Вар.	A	B
1	2	3	4	5	6
1	$\begin{pmatrix} 4 & 0.86 & 0.3 \\ 0.05 & 2 & 0.14 \\ 0.07 & 0.5 & -3 \end{pmatrix}$	$\begin{pmatrix} 2 \\ -3 \\ 1 \end{pmatrix}$	10	$\begin{pmatrix} 5 & -1.2 & 1 \\ 0.23 & 3 & -0.08 \\ 0.92 & -0.33 & 2 \end{pmatrix}$	$\begin{pmatrix} 2 \\ -3 \\ 1 \end{pmatrix}$
2	$\begin{pmatrix} 2 & -0.1 & 0.3 \\ 0.3 & 3 & -0.01 \\ 0.04 & 0.1 & -4 \end{pmatrix}$	$\begin{pmatrix} -7 \\ 4 \\ 1 \end{pmatrix}$	11	$\begin{pmatrix} 7 & 0.52 & -0.08 \\ 1.3 & -4 & 0.2 \\ 0.1 & -0.02 & -2 \end{pmatrix}$	$\begin{pmatrix} 2 \\ -1 \\ 5 \end{pmatrix}$
3	$\begin{pmatrix} 1 & 0.02 & 0.03 \\ 0.02 & -4 & -0.01 \\ 0.4 & 1 & -3 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 4 \\ 1 \end{pmatrix}$	12	$\begin{pmatrix} 4 & -0.3 & -0.8 \\ 0.01 & 2 & -0.5 \\ 0.08 & -0.4 & -3 \end{pmatrix}$	$\begin{pmatrix} 3 \\ -2 \\ 4 \end{pmatrix}$
4	$\begin{pmatrix} 3 & -0.01 & 0.5 \\ 0.28 & 4 & 0.03 \\ 0.06 & -0.8 & -2 \end{pmatrix}$	$\begin{pmatrix} -2 \\ 3 \\ 0.1 \end{pmatrix}$	13	$\begin{pmatrix} 2 & 0.3 & -0.1 \\ 0.03 & -4 & -0.01 \\ 0.02 & 0.61 & -7 \end{pmatrix}$	$\begin{pmatrix} 7 \\ -4 \\ 1 \end{pmatrix}$

1	2	3	4	5	6
5	$\begin{pmatrix} 5 & 2 & 0.08 \\ 0.04 & -3 & 0.02 \\ 0.12 & 0.28 & -1 \end{pmatrix}$	$\begin{pmatrix} 4 \\ -3 \\ 7 \end{pmatrix}$	14	$\begin{pmatrix} 5 & -0.35 & -1 \\ 0.4 & 3 & -1.21 \\ 0.15 & 0.21 & -4 \end{pmatrix}$	$\begin{pmatrix} -1 \\ 3 \\ -4 \end{pmatrix}$
6	$\begin{pmatrix} 7 & -0.04 & -0.8 \\ 0.03 & -2 & 0.02 \\ 0.12 & 0.28 & -1 \end{pmatrix}$	$\begin{pmatrix} 3 \\ -2 \\ 7 \end{pmatrix}$	15	$\begin{pmatrix} 8 & -0.35 & 1 \\ 0.84 & 3 & -1.21 \\ 0.03 & -0.21 & -4 \end{pmatrix}$	$\begin{pmatrix} 8 \\ 2 \\ 7 \end{pmatrix}$
7	$\begin{pmatrix} 3 & 0.1 & 0.18 \\ 0.43 & -5 & 2 \\ 0.17 & 0.35 & -2 \end{pmatrix}$	$\begin{pmatrix} -1 \\ 4 \\ 3 \end{pmatrix}$	16	$\begin{pmatrix} 2 & -0.03 & -0.15 \\ 0.03 & -2 & 0.07 \\ 0.06 & 0.17 & -1 \end{pmatrix}$	$\begin{pmatrix} 2 \\ -4 \\ 8 \end{pmatrix}$
8	$\begin{pmatrix} 2 & -0.17 & 0.41 \\ 1.2 & 6 & -0.1 \\ 0.17 & 0.41 & -3 \end{pmatrix}$	$\begin{pmatrix} 1 \\ -2 \\ 4 \end{pmatrix}$	17	$\begin{pmatrix} 4 & 1.8 & -0.2 \\ 0.81 & -3 & 0.2 \\ 1.05 & 0.43 & -4 \end{pmatrix}$	$\begin{pmatrix} 1 \\ -3 \\ 2 \end{pmatrix}$
9	$\begin{pmatrix} 3 & -0.18 & 0.55 \\ 0.01 & 2 & -0.5 \\ 0.7 & 0.02 & -3 \end{pmatrix}$	$\begin{pmatrix} 7 \\ -3 \\ -2 \end{pmatrix}$	18	$\begin{pmatrix} 1.27 & -0.1 & -0.2 \\ 0.17 & -6 & 0.2 \\ 0.11 & 0.2 & -4 \end{pmatrix}$	$\begin{pmatrix} 2 \\ -3 \\ 4 \end{pmatrix}$
19	$\begin{pmatrix} 4 & 0.12 & 1.8 \\ 0.31 & 4 & 0.19 \\ 0.4 & -1 & 3 \end{pmatrix}$	$\begin{pmatrix} -1 \\ 2 \\ 5 \end{pmatrix}$	23	$\begin{pmatrix} 8 & 0.27 & 0.32 \\ 0.15 & -2 & 0.5 \\ 0.21 & 0.07 & -1 \end{pmatrix}$	$\begin{pmatrix} 0.2 \\ 2 \\ -3 \end{pmatrix}$
20	$\begin{pmatrix} 2.1 & 0.3 & 0.2 \\ 1.8 & -7 & 0.1 \\ 0.2 & 1.2 & -3 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 3 \\ -1 \end{pmatrix}$	24	$\begin{pmatrix} 9 & 1.1 & 0.2 \\ 0.4 & -1 & 0.1 \\ 0.7 & 0.2 & 3 \end{pmatrix}$	$\begin{pmatrix} 8.5 \\ -2 \\ -1 \end{pmatrix}$
21	$\begin{pmatrix} 8 & 0.2 & -1 \\ 0.1 & -2 & 0.5 \\ 0.6 & 1.3 & 3 \end{pmatrix}$	$\begin{pmatrix} -2 \\ 3 \\ -4 \end{pmatrix}$	25	$\begin{pmatrix} 1 & 0.4 & 0.1 \\ 0.9 & 9 & -0.1 \\ 0.2 & 0.7 & 2 \end{pmatrix}$	$\begin{pmatrix} 0.5 \\ -2 \\ 3 \end{pmatrix}$
22	$\begin{pmatrix} 8 & -0.02 & 1.1 \\ 0.83 & 3 & -1.65 \\ 0.08 & -0.21 & -4 \end{pmatrix}$	$\begin{pmatrix} 8 \\ 2 \\ 7 \end{pmatrix}$			

Контрольні запитання

1. Що називають СЛАР? Що означає «розв'язати СЛАР»?

2. Яким чином найпростіше розв'язати СЛАР у системі MATLAB?

3. Як вводяться вектори у мові MATLAB? Якими функціями можна формувати їх у мові MATLAB?

4. Які функції MATLAB дозволяють перетворювати вектор поелементно?

5. За допомогою яких засобів у MATLAB здійснюються основні операції з векторами?

6. Як вводяться матриці у системі MATLAB ?

7. Як сформувати матрицю: а) по заданих векторах її рядків? б) по заданих векторах її стовпців? в) по заданих векторах її діагоналей?

8. Які функції поелементного перетворення матриці є у MATLAB?

9. Як здійснюються у MATLAB звичайні матричні операції?

10. Як розв'язати у MATLAB систему лінійних алгебричних рівнянь?

11. Як обчислити у MATLAB визначник матриці?

14. Як обчислити у MATLAB обернену матрицю?

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 1.6

Розв'язування нелінійних алгебраїчних рівнянь

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

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 , підставляння якого в рівняння (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 $f''(x) = \dfrac{d^2 f(x)}{dx^2}$ також не змінює знака на цьому інтервалі.

Другий етап - процес *уточнення значення нуля*. Відомі чисельні алгоритми відшукування нулів нелінійного рівняння, розглядувані 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

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

При цьому похибкою такого значення може вважатися величина

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

Метод дихотомії (ділення навпіл)

Алгоритм методу легко зрозуміти з рис. 6.1.

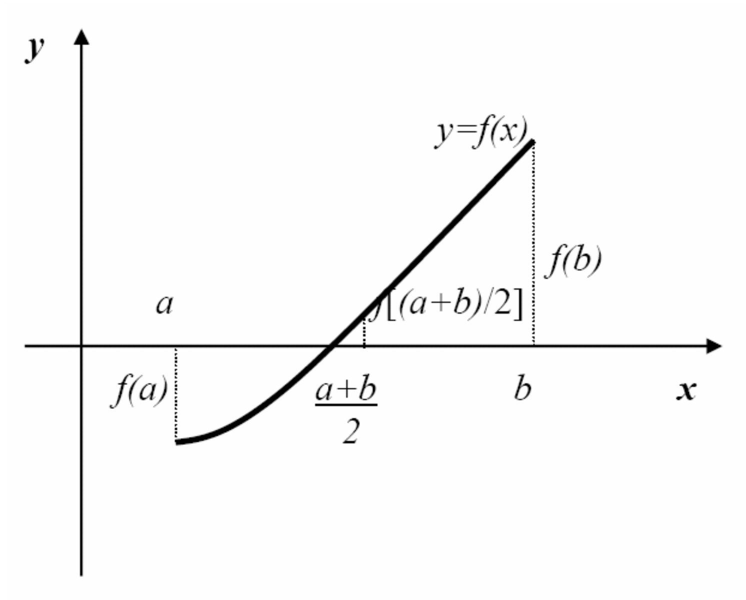

Рис. 6.1 Схема метода дихотомії

Заданий інтервал $[a,b]$ ділиться навпіл. Потім знаходиться наближене значення $x^* = \dfrac{a+b}{2}$ 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 (5) не стане меншою за задане припустиме її значення

$$\Delta x < \Delta_{доп} . \qquad (6.6)$$

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.

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.

Метод хорд

Графічне уявлення про ідею метода хорд дає рис. 6.2. З'єднуємо точки графіка функції, що відповідають кінцям заданого інтервалу $[a,b]$ прямою лінією. Відшукуємо точку перетинання цієї прямої з віссю аргументу - наближене значення x^* нуля функції. Обчислюємо значення функції у цій точці $y^* = f(x^*)$. Далі, як і у методі дихотомії, визначаємо, у якому з двох інтервалів $[a,x^*]$ або $[x^*,b]$ міститься нуль і відповідно до того змінюємо 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

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

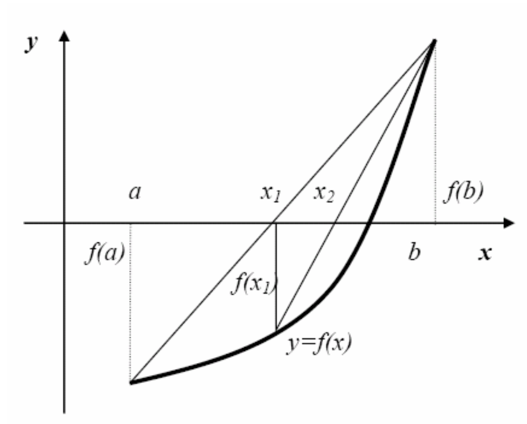

Рис. 6.2 Схема метода хорд

Перевірити умову (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*.

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

Перейдемо до математичного оформлення цього алгоритму.

Перш за все потрібно обчислити значення функції $f(a)$ і $f(b)$ по краях інтервалу. Відшукаємо рівняння прямої, що проходить крізь точки $f(a), a$ і $f(b), b$

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

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

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

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

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.

Сутність метода графічно подано на 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 перетинання x_2, і процес повторюється.

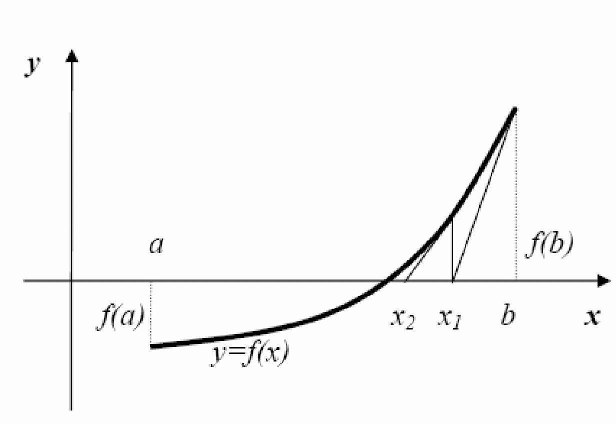

Рис. 6.3 Схема метода дотичних

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:

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

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

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.

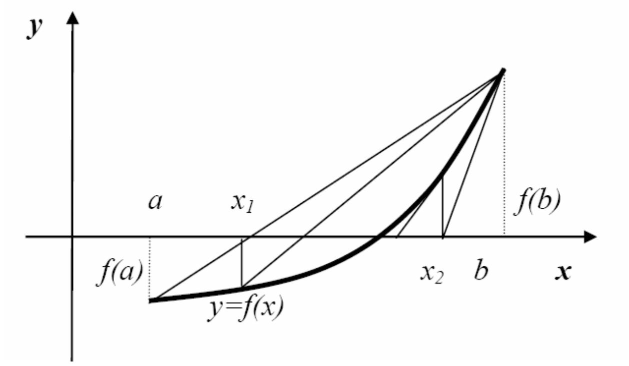

Рис. 6.4 Схема методу хорд і дотичних

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

Графічне уявлення про сутність метода можна одержати з рис. 6.4.

На кожному кроці метода здійснюються дві операції - обчислення наближеного значення x_1* кореня за формулою (8) метода хорд і обчислення другого наближеного значення x_2* того самого кореня за формулою (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).

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

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

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 (1) у вигляді

$$x = \varphi(x) \tag{6.10}$$

Перетворення рівняння (1) до форми (10) не є однозначним і визначається досвідченістю дослідника.

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

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

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.

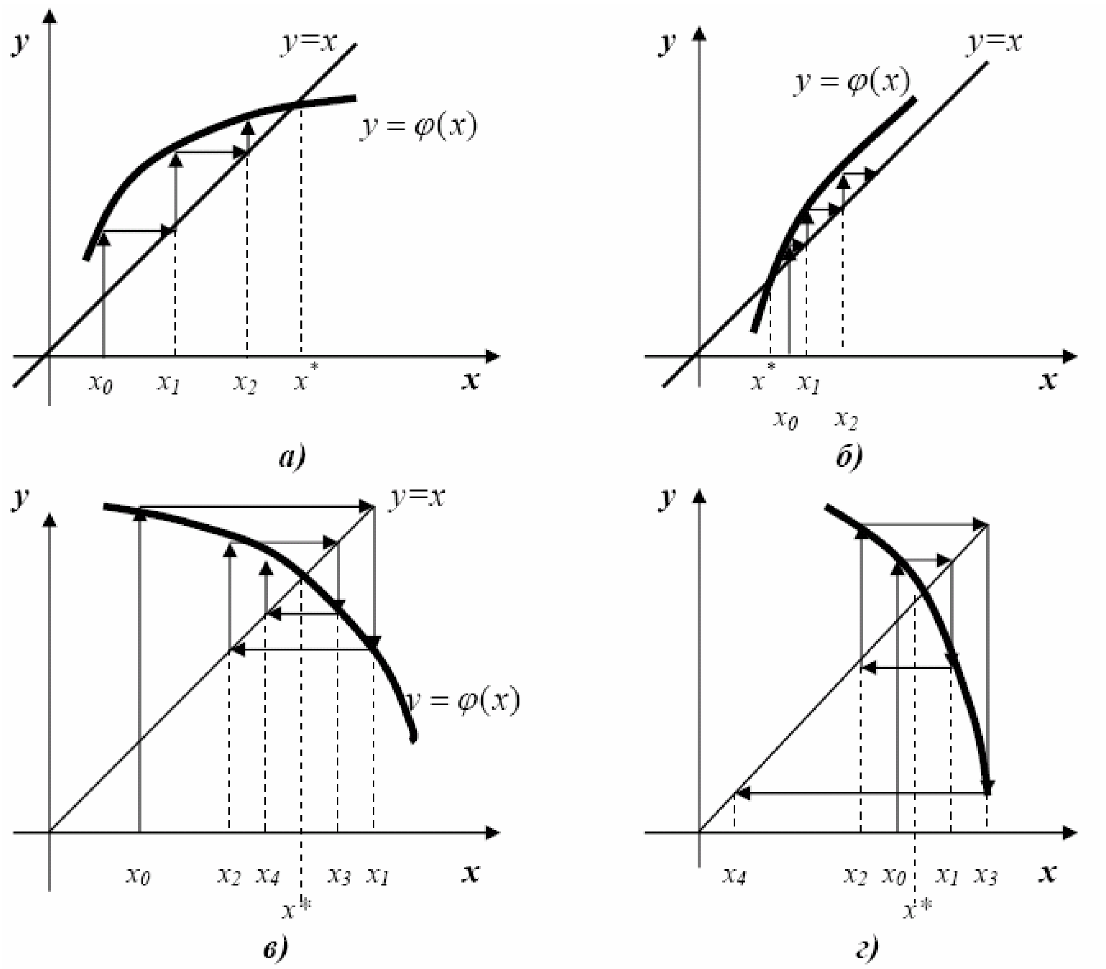

Рис. 6.5 Графічні схеми методу ітерацій

Власне відшукування нуля методом ітерацій у цьому випадку полягає у застосовуванні такої ітераційної формули, яка безпосередньо випливає з (10):

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

де i - номер ітерації (наближення).

Для її використання спочатку задаються певним значенням кореня x_o 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 $y_2(x) = \varphi(x)$). Одержуємо точку 1 на графіку. У відповідності до ітераційної формули (12) за нове наближене значення кореня приймають одержане значення функції $x_1 = y_{20}$ (рух удовж горизонтальної прямої до перетинання із прямою) $y_1(x) = x$. Одержуємо точку 2 графіка. Далі увесь процес повторюється доти, поки різниця між послідовними наближеними значеннями кореня не зменшиться до заданого рівня.

Як бачимо 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

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

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

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

Особливо просто відшукування коренів нелінійного рівняння (6.1) здійснюється у середовищі MATLAB. Покажемо, як це робиться. Утворимо окремий М-файл, в якому обчислюється значення функції

$$f(x) = x^3 + 6x^2 + 3x - 10,$$

яка має три корені $x_1 = -5; \quad x_2 = -2$ і $x_3 = +1$, і назвемо його "pol3fun":

```
function y=pol3fun(x)
% Обчислення функції y=x^3+6x^2+3x-10
v=[1 6 3 -10];
y=polyval(v,x)
```

Для відшукування коренів використовують процедуру ***fzero***, яка обчислює нуль заданої функції. Звичайне звернення до неї є таким

```
x=fzero('pol3fun',x0,tol,trace),
```

де у дужках першим указується (в апострофах) ім'я М-файлу, в якому обчислюється значення функції *f(x)* за заданим значенням аргументу; другим задається початкове значення аргументу *x0* або вектор [*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. Нарешті, параметр *trace* вказує, що проміжні результати слід виводити на екран дисплея.

Наведемо приклади. Задамо початкове значення кореня *x0* = - 6:

```
x0=-6.
```

Тепер звернемося до процедури ***fzero*** у найпростішому варіанті:

```
X=fzero('pol3fun',x0).
```

У командному вікні MATLAB одержимо результат

X = -5

Щоб отримати проміжні результати й інформацію про точність результатів, треба вказати замість параметра *trace* будь-яке ціле додатне число:

```
clc; clear all
x0=-6;
options = optimset('Display','iter');
x=fzero('pol3fun', x0, options)
```

Тоді у командному вікні виникне таблиця

```
Search for an interval around -6 containing a sign change:
 Func-count    a          f(a)             b          f(b)        Procedure
    1             -6           -28            -6           -28    initial interval
    3       -5.83029      -21.7222      -6.16971       -34.969    search
    5          -5.76      -19.3174         -6.24       -38.065    search
    7       -5.66059      -16.1063      -6.33941      -42.6585    search
    9          -5.52      -11.9342         -6.48      -49.5954    search
   11       -5.32118      -6.74272      -6.67882      -60.3165    search
   13          -5.04     -0.734464         -6.96      -77.3839    search
   14       -4.64235       5.33217         -6.96      -77.3839    search

Search for a zero in the interval [-4.64235, -6.96]:
 Func-count    x          f(x)             Procedure
   14       -4.64235       5.33217         initial
   15       -4.79176        3.3671         interpolation
   16       -5.03708     -0.679922         interpolation
   17       -4.99587     0.0742381         interpolation
   18       -4.99992    0.00136181         interpolation
   19             -5  -9.09834e-008        interpolation
   20             -5   3.43725e-012        interpolation
   21             -5             0         interpolation

Zero found in the interval [-4.64235, -6.96]

x =

    -5
```

Як бачимо, 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 меж інтервалу - зникає:

```
clc; clear all
x0=-6;
options = optimset('Display','iter');
x=fzero('pol3fun',[x0 -4], options)
```

```
 Func-count    x          f(x)             Procedure
    2              -4            10        initial
    3        -4.52632       6.61321        interpolation
    4        -4.52632       6.61321        interpolation
    5        -4.91218       1.51205        interpolation
    6        -5.00765     -0.138173        interpolation
    7        -4.99965    0.00623681        interpolation
    8              -5   2.3783e-005        interpolation
    9              -5 -2.78817e-011        interpolation
   10              -5             0        interpolation

Zero found in the interval [-6, -4]

x =

    -5
```

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

Значення відносної похибки обчислення нуля можна змінювати за власним бажанням, наприклад:

```
clc; clear all
x0=-6;
options = optimset('Display','iter', 'TolX',1e-3);
x=fzero('pol3fun',[x0 -4], options)
```

отримують із відповідною точністю:

```
 Func-count    x          f(x)             Procedure
    2              -4           10         initial
    3        -4.52632      6.61321         interpolation
    4        -4.52632      6.61321         interpolation
    5        -4.91218      1.51205         interpolation
    6        -5.00765    -0.138173         interpolation
    7        -4.99763    0.0425789         interpolation

Zero found in the interval [-6, -4]
```

```
x =

   -4.9976
```

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***. Наприклад, припустимо, що потрібно відшукати корені системи нелінійних рівнянь *n*

$$\begin{cases} x_1 x_2 + x_3 = 6{,}5 \\ x_1 x_2^{\ 2} + x_3 = 167 \\ x_1 x_2^{\ 6} + x_3 = 1470 \end{cases}$$

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

```
[x1,x2,x3]=solve('x1*x2+x3=6.5','x1*x2^4+x3=167','x1*x2^6+x3=1470')
```

```
x1 =

                                          1.7554934310605148014354104265164
                                          2.1511971917755071217289751079671
 762.52961873540473151333419519262 - 621.90701499914181279559780253477*i
 621.90701499914181279559780253477*i + 762.52961873540473151333419519262


x2 =

                                            -3.0659535021748359282404774061941
                                              2.967794475288655900039928054698
 - 0.82999197912837056252692361812036*i - 0.45092048655690998589972532425192
   0.82999197912837056252692361812036*i - 0.45092048655690998589972532425192
```

```
x3 =

                                        11.882261233004904252923256860344
                                       0.11568885899217876225427050184419
 352.46285360163186381239610138509*i + 866.51806090717871601749484835904
 866.51806090717871601749484835904 - 352.46285360163186381239610138509*i
```

Щоб скористатися функцією ***fsolve***, потрібно утворити М-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 ***fun33***:

```
function y=fun33(x)
y=[x(1)*x(2)+x(3)-6.5;
x(1)*x(2)^4+x(3)-167;
x(1)*x(2)^6+x(3)-1470];
```

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

```
clc; clear all
x0=[1 1 1];
x=fsolve('fun33',x0)
```

Результатом буде поява у командному вікні вектора коренів

```
x =

    2.1512    2.9678    0.1157
```

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

Завдання до виконання комп'ютерного практикуму

Завдання 1. Відшукайте найменший за модулем дійсний корінь рівняння ***f(x)=0*** за допомогою процедур ***fzero***, ***solve*** і ***fsolve***. Результати порівняйте.

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

Завдання 2. Відшукайте рішення системи нелінійних рівнянь за допомогою процедур ***solve*** і ***fsolve*** відповідно до варіанту (табл. 6.1). Результати порівняйте.

Табл.6.1 Варіанти індивідуальних завдань

Варіант	Система рівнянь	Початкові наближення
1	2	3
1.	$\begin{cases} \sin(x_1 + x_2) - 1,24x_1 = 0,1 \\ x_1^2 + x_2^2 = 1 \end{cases}$	$x_1 = 0,74; \quad x_2 = 0,67;$
2.	$\begin{cases} tg(x_1x_2 + 0.2) = x_1^2 \\ 0.6x_1^2 + 2x_2^2 = 1 \end{cases}$	$x_1 = 0,88; \quad x_2 = 0,52;$
3.	$\begin{cases} \sin x_1 + 2\sin x_2 = 1 \\ 2\sin 3x_1 + 3\sin 3x_2 = 0,3 \end{cases}$	$x_1 = 1,08; \quad x_2 = 0,06;$
4.	$\begin{cases} tg(x_1 - x_2) - 4x_1 = 0 \\ x_1^2 + 2x_2^2 = 1 \end{cases}$	$x_1 = -0,5; \quad x_2 = 0,6;$
5.	$\begin{cases} x_1^4 + x_2^2 = 3 \\ x_1^3 + x_2^3 = 4 \end{cases}$	$x_1 = 0,95; \quad x_2 = 1,4;$
6.	$\begin{cases} \sin x_1 - x_2 = 1,3 \\ \cos x_2 - x_1 = -0,82 \end{cases}$	$x_1 = 0,8; \quad x_2 = -0,35;$
7.	$\begin{cases} x_1^3 + x_2^3 - 6x_1 + 3 = 0 \\ x_1^3 - x_2^3 - 6x_2 = 2 \end{cases}$	$x_1 = 0,52; \quad x_2 = -0,37;$
8.	$\begin{cases} x_2 + e^{x_1 - x_2} = 0 \\ x_1 + e^{x_1 - x_2} = 0 \end{cases}$	$x_1 = -0,7; \quad x_2 = -0,35;$
9.	$\begin{cases} \sin(x_1 + 1) - x_1 = 0,1 \\ \cos x_1 + 2x_2 = 2 \end{cases}$	$x_1 = 0,86; \quad x_2 = 0,67;$
10.	$\begin{cases} \cos(x_2 - 1) + x_1 = 0,5 \\ x_2 - \cos x_1 = 3 \end{cases}$	$x_1 = 1,20; \quad x_2 = 3,36;$
11.	$\begin{cases} (x_1^2 - 1)x_2 = 9 \\ x_1x_2 - x_1^2 + 10 = 0 \end{cases}$	$x_1 = -2,0; \quad x_2 = 3,00;$
12.	$\begin{cases} \sin(x_1 + x_2) - 2,4x_1 = -3,2 \\ x_1^2 + x_2^2 - x_1x_2 = 10 \end{cases}$	$x_1 = 0,92; \quad x_2 = -2,60;$

1	2	3
13.	$\begin{cases} \sin x_1 + 2\cos x_2 = 0,8 \\ x_1 x_2^2 + 3x_1 = 4,5 \end{cases}$	$x_1 = 0,01; \quad x_2 = -23,97;$
14.	$\begin{cases} 2x_1 x_2^2 - 4x_2 = 7,5 \\ x_1^2 - 3x_1 x_2 = -4,5 \end{cases}$	$x_1 = 3; \quad x_2 = 1,5;$
15.	$\begin{cases} (x_1^2 - 1)x_2^2 = 18,75 \\ x_1 + x_2^2 = 8,25 \end{cases}$	$x_1 = 2; \quad x_2 = -2,50;$
16.	$\begin{cases} tg(-x_2) - 4x_1 = -3 \\ x_1^2 + 3x_2^2 = 4 \end{cases}$	$x_1 = 0,19; \quad x_2 = -1,15;$
17.	$\begin{cases} x_1^2 x_2 - 8x_1 = -5,5 \\ x_1 x_2 + 3x_2 = 10 \end{cases}$	$x_1 = 0,76; \quad x_2 = 1,03;$
18.	$\begin{cases} x_1^2 \sin x_2 + x_2^2 \sin x_1 + 1 = 0 \\ 2x_1 + e^{x_1 + x_2} = 4 \end{cases}$	$x_1 = -0,32; \quad x_2 = 1,86;$
19.	$\begin{cases} e^{x_1} + 2x_2 \ln x_1 = 3 \\ x_1^2 x_2 - 3x_1 + 5,4 = 0 \end{cases}$	$x_1 = 1,24; \quad x_2 = -1,08;$
20.	$\begin{cases} \sin x_1 + 3,5\sin x_2 = 1 \\ 2\sin 3x_1 + 3\sin 2x_2 = 0,4 \end{cases}$	$x_1 = -0,34; \quad x_2 = 0,39$

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***?

ЛІТЕРАТУРА

1. Лазарєв Ю.Ф. Моделювання на ЕОМ: Навчальний посібник / Ю.Ф. Лазарєв. – Київ: Корнійчук, 2007. – 290 с.
2. Лазарєв Ю.Ф. Початки програмування у середовищі MATLAB: Навчальний посібник / Ю.Ф. Лазарєв. – Київ: Корнійчук, 1999. – 160 с.
3. 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.
4. Барановская Г.Г., Любченко И.Н. Микрокалькуляторы в курсе высшей математики: Практикум / Г. Г. Барановская, И. Н. Любченко. – Киев, Вышая школа, 1987. – 288с.
5. Бахвалов Н. С. Численные методы / Н. С. Бахвалов. – М.: Наука, 1973. – 632с.
6. Демидович Б.П. Основы вычислительной математики / Б.П. Демидович, И.А. Марон. – М.: Наука, 1970. – 664с.
7. Березин М.С. Методы вычислений / М.С. Березин, Н.П. Жидков. – М.: Наука, 1966. т.1,2. – 620с.

8.