Лабораторна робота № 7.

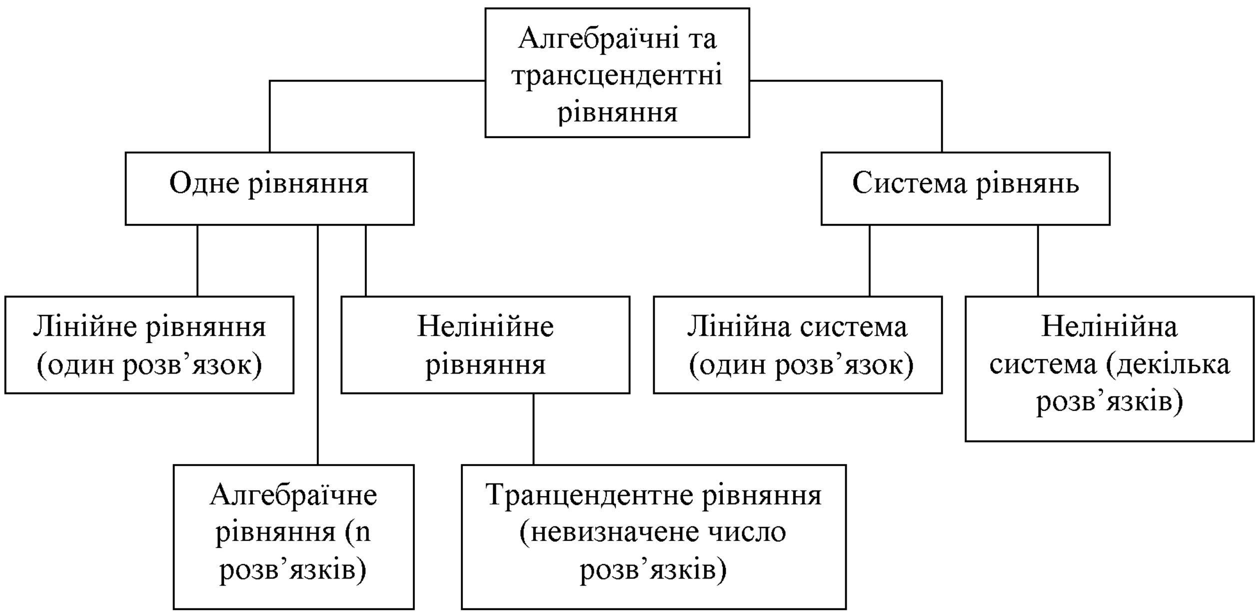
Тема: Розв’язання лінійних рівнянь та систем лінійних рівнянь у системі **MATLAB.**

Мета роботи: ознайомитись з методами розв’язання лінійних рівнянь та лінійних систем рівнянь; навчитись користуватися спеціальними командами системи **MATLAB**, які використовуються в алгоритмах розв’язання систем рівнянь

Теоретичний мінімум

Інженеру часто доводиться вирішувати алгебраїчні та трансцендентні рівняння, що може розглядатися самостійно як задача або бути частиною складніших задач. В обох випадках практична цінність методу значною мірою визначається швидкістю і ефективністю отриманого рішення.

Вибір відповідного методу розв’язання рівнянь залежить від характеру даної задачі. Задачі, що зводяться до вирішення алгебраїчних та трансцендентних рівнянь, можна класифікувати по числу рівнянь, залежно від запропонованого характеру рівняння і числа можливих розв’язків та інших критеріїв (див. мал. 7.1).



Мал. 7.1 – Класіфікація рівнянь

Будемо називати рівняння алгебраїчним або трансцендентним залежно від того, чи має воно n розв’язків, або має невизначене число розв’язків. У випадку, коли $n=1$, алгебраїчне рівняння називається лінійним.

Систему рівнянь будемо називати лінійною або нелінійною в залежності від виду рівнянь (лінійних або нелінійних), які формують систему.

Вправа 1. Розв'язування алгебраїчних рівнянь та найпростіших систем рівнянь. Символьний розв'язок рівнянь на основі використання підсистеми **Symbolyc Math**.

Алгебраїчне рівняння має представлення у вигляді $P_n(x) = 0$, де $P_n(x)$ – многочлен n-ї степені від декількох незалежних змінних.

Алгебраїчним рівнянням з однією невідомою називається рівняння, яке приводиться до виду $a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + \ldots + a_{n-1} x + a_n = 0$, де n – додатне ціле число (порядок многочлену).

Значення невідомого x, яке перетворює алгебраїчне рівняння у тотожність, називаються коренем (розв'язком) алгебраїчного рівняння.

Для аналітичного розв'язання алгебраїчного рівняння у системі **MATLAB** використовується команда **solve** (з підсистеми комп'ютерної алгебри **Symbolyc Math,** [20,с.45]), яка формається у наступному вигляді: **solve('eqn1','eqn2',...,'eqnN')** або **solve('eqn1','eqn2',...,'eqnN','var1','var2',..., 'varN')**, де **eqn1, eqn2**,… – одне або декілька рівнянь з символьними змінними, які використовуються для представлення рівнянь, **var1, var2**,… – шукані невідомі (символьні) змінні.

Приклад 1. Знайти розв'язок рівняння $p\sin x = r$.

Ця проста задача розв'язується наступним чином:

```
> syms p r x
>> solve('p*sin(x)=r')
ans =
asin(r/p)
```

Необхідно мати на увазі, що при розв'язані системи рівнянь для зберігання результату створюється структура з полями, яка формується за допомогою квадратних дужок, в яких виписані імена шуканих невідомих (змінних)**.**

Приклад 2. Знайти розв'язок системи рівнянь $\begin{cases} x_1 + 2x_2 = 12 \\ x_1 + x_2 = 18 \end{cases}$ та призначити отримані результати змінним x_1, x_2.

Розв'язання прикладу можливе наступними командами:

```
>> [x1,x2]=solve('x1+2*x2=12','x1+x2=18')
x1 =
24
x2 =
–6
```

Можна також використовувати інший варіант формування команди розв'язання рівняння або системи рівнянь, а саме **s = solve(eqn1,eqn2,…)**.

Приклад 3. Розв'язати систему рівнянь $\begin{cases} x+y=2 \\ x-5y=1 \end{cases}$ відносно невідомих x, y.

Використовуємо згадану вище команду:

```
>> s=solve('x+y=2','x–5*y=1')
s =
   x: [1x1 sym]
   y: [1x1 sym]
```

Для того, щоб отримати лише один з можливих розв'язків системи (наприклад, x), треба скористатися такою командою:

```
>> s.x
ans =
11/6
>> s.y
ans =
1/6
```

Зауважимо, що команда **solve** у системі **MATLAB** не розв'язує нерівностей. Аналогічна команда, яка входить до складу математичної системи **Maple,** дозволяє здійснювати цю процедуру.

Вправа 2. Ознайомитись з методами розв'язання систем лінійних рівнянь. Розв'язання систем лінійних рівнянь прямими методами.

Підходи до розв'язання систем лінійних рівнянь [24] складаються з двох напрямків:

- прямі методи, які використовуються для обчислення коренів системи; до таких методів відносять: розв'язання систем за допомогою оберненої матриці, на основі використання правила Крамера, методу Гауса та інші)
- ітераційні методи, що дозволяють отримати розв'язок системи з заданою точністю шляхом ітераційних процесів, які збігаються (метод ітерації, метод Зейделя та інших).

Метод розв'язання задачі називають прямим, якщо він дозволяє отримати розв'язок після виконання скінченного числа елементарних операцій. До прямих методів розв'язання відносять метод Гауса і його модифікації (наприклад, метод Холецького (дивись далі)).

Зауважимо, що внаслідок округлень, що мають місце при розрахунках на комп'ютері, результати навіть точних методів розглядаються як наближені.

Нехай маємо алгебраїчну неоднорідну систему з n лінійних рівнянь з n невідомими $x_1, x_2, ..., x_n$:

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

Дана система лінійних рівнянь може бути записана у матричному вигляді:

$$AX = B, \qquad (2)$$

де

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix}, B = \begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{bmatrix}. \qquad (3)$$

Матриця **A**, стовпцями якої є коефіцієнти при відповідних невідомих, а рядками – коефіцієнти при невідомих у відповідному рівнянні, називається матрицею системи; матриця-стовпець B, елементами якої є праві частини рівнянь системи, називається матрицею правої.

Матриця-стовпець **X**, елементи якої є шуканими невідомими, називається розв'язком вихідної лінійної системи. Розширена матриця системи – це матриця **D**, що утворюється з елементів матриці A і стовпця B**:** **D=[A B].**

Основними чинниками системи, від яких залежить існування і єдиність розв'язання, є:

- n – число невідомих змінних (число стовпців матриці коефіцієнтів A);
- r – ранг матриці коефіцієнтів**: r=rank(A);**
- R – ранг розширеної матриці: **R=rank([A B]).**

Нагадаємо твердження:

- система сумісна (має розв'язок) тоді і тільки тоді, коли **r=R**; у випадку **r<R** система несумісна (тобто розв'язку не існує);
- розв'язок системи єдиний тоді і тільки тоді, коли **r=R=n**; у випадку **r=R<n** система є невизначеною (тобто має нескінченно багато розв'язків).

Розглянемо матричну систему **AX=B**, у який ранг матриці коефіцієнтів **r=R=m=n**, **det(A)≠0**.

Зауважимо, що однією матричною операцією можна вирішити відразу декілька систем лінійних рівнянь з однією і тією ж матрицею коефіцієнтів системи, але з різними наборами правих частин – матриці **B** та **A**.

Якщо матриця **А** – неособлива (несингулярна, невироджена), тобто $\det A \neq 0$, то система (1) або еквівалентне їй матричне рівняння (2), має єдиний розв'язок. Дійсно, якщо $\det A \neq 0$, існує обернена матриця A^{-1}.

Множення обох частин рівняння (2) на матрицю A^{-1} дозволяє отримати наступне рівняння: $\mathbf{A^{-1}AX=A^{-1}B}$. Звідки

$$X=A^{-1}B, \qquad (4)$$

Так як $A^{-1}A=E$, $EX=X$, Е т – одинична матриця. Формула (4) дає єдиний розв'язок рівняння (2).

У системі **MATLAB** для розв'язання систем лінійних рявнянь існують декілька методів [20, с.336]. Простішим методом розв'язання є метод вирішувачів систем лінійних рівнянь:

- оператор **X=B\A** знаходить розв'язок системи рівнянь вигляду **AX=B**, де **А** – прямокутна матриця розміром **m×n, B** – матриця розміром **m×1**;
- оператор **X=B/A** знаходить розв'язок системи рівнянь вигляду **XA=B**, де **А** – прямокутна матриця розміром **n×m**, **B** – матриця розміром n×**1.**

Приклад 1. Виконати розв'язання алгебраїчної системи

$$\begin{cases} x_1 + x_2 + x_3 + x_4 = 1 \\ x_1 + x_2 + x_3 + x_4 = 1 \\ x_1 + x_2 + x_3 + x_4 = 1 \end{cases}.$$

При розв'язуванні задачі необхідно звернути увагу, що розмір матриці А дорівнює 3×4, а розмір вектору В дорівнює 3×1. В цьому випадку можна скористатися оператором **X=B\A:**

```
>> a=[1 1 1 1;1 1 1 1;1 1 1 1]; b=[1; 1; 1]; x1=b\a
x1 =
    1    1    1    1
```

У разі, коли ранг **r=R<n,** алгебраїчна система є невизначеною (має нескінченно багато розв'зків). Загальне розв'язання такої системи може бути виписане у вигляді $x = x_0 + y$, де x_0 – частинний розв'язок неоднорідної системи **AX=B**, $B \neq 0$; y – загальний розв'язок однорідної системи **AX=0**.

Можливий частинний розв'язок неоднорідної системи **AX=B** може бути отриманий за допомогою лівого матричного ділення **X=A\B**. При виконанні операції **A\B** система **MATLAB** видає попередження, яке означає що «матриця близька до сингулярної (виродженої) матриці або погано масштабована». В цьому випадку результати розв'язання можуть бути неточними.

Приклад 2. Знайти розв’язок системи лінійних алгебраїчних рівнянь

$$\begin{cases} 2x_1 + 7x_2 + 3x_3 + x_4 = 5 \\ x_1 + 3x_2 + 5x_3 - 2x_4 = 3 \\ x_1 + 5x_2 - 9x_3 + 9x_4 = 1 \\ 5x_1 + 18x_2 + 4x_3 + 5x_4 = 12 \end{cases}.$$

Перелік команд, що наведений нижче, дозволяє здійснити процедуру розв’язання вихідної системи:

```
>> A=[2 7 3 1;1 3 5 –2;1 5 –9 8;5 18 4 5]; b=[5;3;1;12]; x=A\b

Warning: Matrix is close to singular or badly scaled.

        Results may be inaccurate. RCOND = 4.077954e-019.

(Type "warning off MATLAB:nearlySingularMatrix" to suppress this warning.)

x=

  –5.5000

   1.2500

   1.7500

   2.0000
```

(Зауваження: матриця близька до сингулярної або погано масштабована. Результати можуть бути неточними).

Найбільший інтерес серед частинних розв’язків неоднорідної системи має так званий *нормальний розв’язок*. У системі **MATLAB** він може бути отриманий за допомогою псевдооберненої матриці згідно формули: **x=pinv(A)*B.**

Команда **pinv(A)** виконує операцію обчислення матриці, яка є псевдооберненою до матриці **A**. Вона має тіж самі розміри, що і матриця $\mathbf{A^{-1}}$**.** Зміст команди витікає з наступних тверджень [44, с.147]: A*P*A=A, P*A*P=P.

Приклад 3. Знайти корені системи рівнянь приклада 2 за допомогою команди **pinv(A).**

Розв’язання здійснюється наступними командами:

```
>> pA=pinv(A)
pA =
```

```
   0.0045    0.0030     0.0001    0.0105
   0.0147    0.0071     0.0081    0.0371
   0.0138    0.0275    –0.0549    0.0139
  –0.0028   –0.0150     0.0393    0.0066
>> x=pA*b
x =
   0.1576
   0.5478
   0.2636
   0.0594
```

Існують алгоритми розв’язків систем лінійних алгебраїчних рівнянь засновані на операції факторизації матриць за однією з схем: Холецького, методу Гауса, QR-розкладення.

У випадку вирішення системи з однією матрицею **A** коефіцієнтів та різними матрицями правих частин системи, для прискорення обчислення краще один раз провести операцію факторизації матриці та визначити обернену матрицю. Команди факторизації наступні:

- **chol** – здійснює розкладення за схемою Холецького для симетричних або позитивно визначених матриць;
- **cholnc** – здійснює неповне розкладання за схемою Холецького (представлення симетричної матриці у вигляді добутку матриць верхньої трикутної та транспонованої);
- **lu** – здійснює **LU**-розкладання для квадратних матриць;
- **luinc** – здійснює неповне **LU**-розкладання;
- **hess** – виконує приведення до форми Хесенбергу;
- **rref** – реалізує приведення вихідної матриці до трикутної форми;
- **qr** – здійснює **QR**-розкладання (представлення матриці у вигляді добутку ортогональної та верхньої трикутної матриць).

Вправа 3. Розв’язування системи лінійних алгебраїчних рівнянь за методом Крамера.

Про особливості методу Крамера радимо звернутися до [1].

Приклад 1. Розв’язати методом Крамера систему лінійних алгебраїчних рівнянь

$$\begin{cases} 10x_1 + \quad x_2 + \quad x_3 = 12 \\ 2x_1 + 10x_2 + \quad x_3 = 13 \\ 2x_1 + \ 2x_2 + 10x_3 = 14 \end{cases}.$$

Команди, що допомагають розв'язати задачу, наступні:

```
A = [10 1 1; 2 10 1; 2 2 10];b = [12;13;14];
```

Перевірка невиродженості системи:

```
>> rank(A)
ans = 4
```

Описуємо допоміжні матриці згідно правилу Крамера

```
A1 = A;A2 = A;A3 = A;
A1(:,1) = b;A2(:,2) = b;A3(:,3) = b;
>> x1 = det(A1) / det(A);x2 = det(A2) / det(A);
x3 = det(A3) / det(A); x=[x1;x2;x3]
```

Перевірка розв'язку:

```
A*x–b
ans =
    0
    0
    0
```

Таким чином, шуканий розв'язок має вигляд: $x_1 = 1, x_2 = 1, x_3 = 1$.

Вправа 4. Розв'язування системи лінійних алгебраїчних рівнянь методом Гауса.

У методі Гауса для обчислення масштабуючих множників вимагається ділити відповідні елементи матриці на провідні елементи системи на кожному кроці. Якщо елемент дорівнює нулю або близький до нуля, то можливе неконтрольоване зростання похибки. Це обумовлює застосування модифікації методу Гауса, що мають кращі обчислювальні властивості.

Розглянемо метод Гауса (метод виключення провідного елементу) з вибором провідного елементу по стовпцю (схема часткового вибору). Провідний елемент вибірається на k-му кроці прямого ходу як максимальний по модулю коефіцієнт $a_{i_1 k_i}$ при невідомому x_k в рівняннях з номерами $i = k+1, \ldots, m$. Рівняння, яке відповідає вибраному коефіцієнту з номером i_k, міняють місцями з k-им рівнянням системи з метою, щоб провідний елемент зайняв місце коефіцієнту $a_{kk}^{(k-1)}$. Після цієї перестановки проводять виключення згідно схеми єдиного розподілу. В результаті всі масштабуючі множники будуть по модулю меньші одиниці і схема отримує обчислювальну стійкість.

Іншими словами, метод полягає в тому, що систему (1) приводять послідовним виключенням невідомих до еквівалентної системи з верхньою трикутною матрицею:

$$\begin{cases} x_1 + \alpha_{12}x_2 + \ldots + \alpha_{1n}x_n = \beta_1, \\ \qquad x_2 + \ldots + \alpha_{2n}x_n = \beta_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \qquad\qquad x_n = \beta_n, \end{cases} \tag{5}$$

розв’язання якої відбувається за рекурентними формулами:

$$x_n = \beta_n,\ x_i = \beta_i - \sum_{j=i+1}^{n} \alpha_{ij}x_j,\ (i = n-1, n-2, \ldots, 1) \tag{6}$$

В матричній формі це відповідає наступному: спочатку елементарними операціями над рядками розширену матрицю системи приводять до спеціального вигляду (прямий хід методу Гауса):

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ a_{21} & a_{22} & \ldots & a_{2n} & b_2 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} & b_n \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & \alpha_{12} & \ldots & \alpha_{12} & \beta_1 \\ 0 & 1 & \ldots & \alpha_{2n} & \beta_2 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & \beta_n \end{bmatrix}, \tag{7}$$

а потім (зворотний хід методу Гауса) цю матрицю перетворюють до вигляду:

$$\begin{bmatrix} 1 & 0 & \ldots & 0 & x_1 \\ 0 & 1 & \ldots & 0 & x_2 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & x_n \end{bmatrix} \tag{8}$$

Останній, $(n+1)$-й, стовпець цієї матриці містить в собі шуканий розв’язок вихідної системи (1).

У термінах системи **MATLAB** рівняння (4) еквівалентне виразу **x=inv(A)*B**. Зверніть увагу на те, щоб розмірності масивів **A** і **B** були погоджені**.** Для квадратних матриць відшукується точний розв’язок; у випадку перевизначених систем використовується метод найменших квадратів.

Приклад 1 Здійснити розв'язання системи лінійних алгебраїчних рівнянь

$$\begin{cases} x_1 + 4x_2 = \ 12 \\ 3x_1 - 2x_2 = - \ 6 \end{cases}.$$

Для цього скористаємося наступним переліком команд:

```
>> A=[1 4;3 –2]; B=[12; –6]; X=A\B

X =

    0

    3
```

Вправа 5. Розв'язування системи лінійних алгебраїчних рівнянь за допомогою **LU**-розкладання.

У цьому методі матриця A представляється у вигляді добутку нижньої трикутної матриці L і верхньої трикутної U: $A = LU$, де

$$L = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ \mu_{21} & 1 & \dots & 0 & 0 \\ \mu_{31} & \mu_{32} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \mu_{m1} & \mu_{32} & \mu_{33} & \dots & 1 \end{bmatrix} \text{ та } U = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m-1} & a_{1m} \\ 0 & a_{22}^{(1)} & \dots & a_{2m-1}^{(1)} & a_{2m}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{3m-1}^{(2)} & a_{3m}^{(2)} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & a_{mm}^{(m-1)} \end{bmatrix}.$$

Нехай дана матриця $A = \begin{bmatrix} 3 & 4 & -9 & 5 \\ -15 & -12 & 50 & -16 \\ -27 & -36 & 73 & 8 \\ 9 & 12 & -10 & -16 \end{bmatrix}$. Розкладемо матрицю A на множники, для чого обчислимо масштабні множники:

$$\mu_{21} = -\frac{15}{3} = -5, \quad \mu_{31} = -\frac{27}{3} = -9, \quad \mu_{41} = \frac{9}{3} = 3.$$

Щоб отримати верхню трикутну матрицю, здійснемо наступне перетворення вихідної матриці A: $\begin{bmatrix} 3 & 4 & -9 & 5 \\ 0 & 8 & 5 & 9 \\ 0 & 0 & -8 & 53 \\ 0 & 0 & 17 & -31 \end{bmatrix}$, для чого обчислюємо масштабні множники:

$$\mu_{32} = \frac{0}{8} = 0, \quad \mu_{42} = \frac{0}{8} = 0 .$$

На останньому кроці треба обчислити лише один масштабний множник: $\mu_{43} = \dfrac{17}{8}$.

В результаті отримуємо шуканий вигляд матриці U (верхньої трикутньої) і матриці L (нижня трикутна), де

$$U = \begin{bmatrix} 3 & 4 & -9 & 5 \\ 0 & 8 & 5 & 9 \\ 0 & 0 & -8 & 53 \\ 0 & 0 & 0 & \dfrac{653}{8} \end{bmatrix} L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -5 & 1 & 0 & 0 \\ -9 & 0 & 1 & 0 \\ 3 & 0 & -\dfrac{17}{8} & 1 \end{bmatrix}.$$

Приклад 1. Розв’язати систему лінійних алгебраїчних рівнянь

$$\begin{cases} 3x_1 + 4x_2 - 9x_3 + 5x_4 = -14 \\ -15x_1 - 12x_2 + 50x_3 - 16x_4 = 44 \\ -27x_1 - 36x_2 + 73x_3 + 8x_4 = 142 \\ 9x_1 + 12x_2 - 10x_3 - 16x_4 = -76 \end{cases}.$$

Розв’язання вихідної системи на мові системи **MATLAB** буде мати наступний вигляд:

```
>> A=[3 4 –9 5;–15 –12 50 –16;–27 –36 73 8;9 12 –10 –16];
>> b=[–14;44;142;–76]; [L,U]=lu(A); x=U\(L\b)
x =
  –8.0000
  –2.0000
  –2.0000
   0.0000
```

Вправа 6. Знаходження розв’язку системи лінійних алгебраїчних рівнянь методом Холецького.

Якщо матриця системи є симетричною або додатньо визначеною, то зручно для розв’язання системи використовувати метод Холецького (метод квадратних коренів). В основі цього методу покладений алгоритм спеціального **LU**-розкладання матриці **A**, внаслідок чого вона приводиться до вигляду $\mathbf{A=LL^T}$.

В методі **LU**-розкладання, розв’язання системи відбувається послідовним розв’язанням двох систем з трикутними матрицями: **LY=B** і $\mathbf{L^TX=Y}$. З метою знаходження коефіцієнтів матриці **L** вихідні коефіцієнти матриці $\mathbf{LL^T}$ прирівнюють відповідним елементам матриці **A**. Потім знаходять послідовно необхідні коефіцієнти за формулами:

$$l_{11} = \sqrt{a_{11}}, ..., l_{i1} = a_{i1} / l_{i1}, \quad i = 2, 3, ..., m;$$

$$l_{22} = \sqrt{a_{22} - l_{21}^2}, ..., l_{i2} = (a_{i2} - l_{i1} l_{21}) / l_{22}, \quad i = 3, 4, ..., m;$$

$$\dots\ \dots\ \dots\ \dots\ \dots\ \dots$$

$$l_{kk} = \sqrt{a_{kk} - l_{k1}^2 - ... - l_{k,k-1}^2}, ..., l_{ik} = (a_{ik} - l_{i1} l_{k1} - l_{i2} l_{k2} - ... - l_{i,k-1} l_{k,k-1}) / l_{kk},\ i = k+1, ..., m;$$

Приклад 1. Виконати розв’язання методом Холецького системи лінійних алгебраїчних рівнянь $\begin{cases} 81x_1 - 45x_2 + 45x_3 = \ \ 81 \\ -45x_1 + 50x_2 - 15x_3 = -460 \\ 45x_1 - 15x_2 + 38x_3 = \ \ 193 \end{cases}$.

Для знаходження розв’язку системи рівнянь за допомогою метода Холецького скористуємося наступними командами:

```
>> A=[81 –45 45;–45 50 –15;45 –15 38]
A =
   81  –45   45
  –45   50  –15
   45  –15   38
>> b=[531;–460;193]; L=chol(A)
L =
    9   –5    5
    0    5    2
    0    0    3
>> %L'*L*x=b
>> x=L\(L'\b)
x =
    6
   –5
```

–4

Перевірка:

```
>> A*x–b
ans =
    0
    0
    0
```

Таким чином, легко пересвідчитися, що розв'язання системи виконане правильно.

Вправа 7. Розв'язання систем лінійних алгебраїчних рівнянь за допомогою ітераційних методів. Метод простої ітерації.

Метод розв'язання задачі називають *ітераційним*, якщо в результаті одержують нескінчену послідовність наближень до точного розв'язку. Якщо ця послідовність збігається до розв'язку задачи, то кажуть, що ітераційний процес збігається.

Використання ітераційних методів пов'язане з накопиченням похібки методу. Ефективне застосування ітераційних методів істотно залежить від вдалого вибору початкового наближення і швидкості збігання процесу.

Нехай розглядається лінійна алгебраїчна система $AX = B$. Для застосування ітераційного методу система повинна бути приведена до еквівалентного вигляду $X = BX + D$. Потім вибирається початкове наближення до шуканого розв'язку системи $x^{(0)} = \left(x_1^0,\ x_2^0,\ ...,\ x_m^0\right)$ і знаходиться необхідна кількість (послідовностей) наближень. Для збіжності ітераційного процесу достатньо, щоб була виконана умова $\| B \| < 1$.

Критерій, згідно якому завершується ітераційна процедура, залежить від вибраного ітераційного методу. На практиці застосовують декілька ітераційних методів, а саме: метод Якобі, метод Зейделя, метод простої ітерації. Звернемося до методу простої ітерації. Якщо матриця А – симетрична і додатньо визначена матриця, то систему вихідних рівнянь часто приводять до еквівалентного вигляду: $X = X - \tau(AX - B)$, де τ –параметр ітерації. Розрахункова формула методу простої ітерації в цьому випадку отримає вигляд: $X^{n+1} = X^n - \tau(AX^n - B)$ при $\tau>0$.

Передбачимо, що $\lambda_{\min}$ і $\lambda_{\max}$ – відповідно мінімальне й максимальне власні значення матриці А. Тоді оптимальним є такий вибір параметра $\tau=2/(\lambda_{\min}+\lambda_{\max})$.

Звернемо увагу на те, що для методу простої ітерації необхідно використовувати оператор циклу з невизначеним числом операцій та деякі команди обчислення матриць, а саме:

- **while…end** – команди оператора циклу, в якому процедура виконується то тих пір, доки масив логічного виразу не стає нульовим;
- **eig** – команда обчислює власні значення матриці;
- **norm** – команда обчислює норми векторів та матриць.

Приклад 1. Розв’язати методом простої ітерації лінійну алгебраїчну систему

$$\begin{cases} 3x_1 - 2x_2 \qquad = -\ 21 \\ -2x_1 + 3x_2 \qquad = \ \ 24\,. \\ \qquad\qquad 3x_3 = \ 193. \end{cases}$$

Забезпечимо виконання наступної послідовністі команд:

```
>> A = [3 –2 0; –2 3 0; 0 0 3];b = [–21;24;15];
>> e = eig(A);
```

Обчислимо ітераційний параметр:

```
>> tau = 2 / (min(e) + max(e));r = 1;
```

Виконаємо метод простої ітерації з початковим наближенням **x0**=(0,0,0):

```
>> x0 = [0;0;0];y = x0;
>> while r > 100 * eps
x = y – tau * (A * y – b);
r = norm(x–y);
y = x;
end
```

Перевіримо отриманий результат:

```
>> A * x – b
ans =
  1.0e-013 *
  –0.2842
   0.2487
      0
```

Вправа 8. Здійснити розв’язування системи алгебраїчних рівнянь методом найменших квадратів.

Одним з точних методів розв’язання системи лінійних алгебраїчних рівнянь за схемою ітераційного методу є метод найменших квадратів [24].

Розглянемо команди, що можуть бути використані для розв’язання, зокрема, перевизначених систем:

- **lsqr(A,B) – команда** повертає точний розв’язок системи алгебраїчних лінійних рівнянь **A*X=B**. Матриця коефіцієнтів **A** повинна бути прямокутною за розміром **m×n**, а вектор-стовпець правих частин рівнянь, вектор B, повинен мати довжину **m**. Умова **m**$\geq$**n** може бути зайвою.

Команда **lsqr** починає процес ітерації від нульовоговихідного вектору (за замовчанням – вектора з **n** компонентами). Ітерації проводяться до збіжності з шуканим розв'язком або до появи помилки, або до досягнення максимального числа ітерацій (за замовчанням таке число ітерацій дорівнює **min(20, m, n)** , тобто або 20, або числу рівнянь в системі, або числу невідомих). Збіжність забезпечується, коли відношення норм векторів **norm(B-Ax)/norm(B)** меньше або дорівнює величині **tol** похибці методу (за замовчанням **tol =1e-6**);

- **lsqr(A.B,tol)** – команда повертає розв'язок з заданою похибкою, що визначається порогом обчислення **tol**;
- **lsqr(A,b.tol,maxit)** – команда повертає розв'язок при заданому максимальному числі (**maxit)** ітерацій замість визначеного машиною числа за замовчанням.

Приклад. Розв'язати лінійну алгебраїчну систему рівнянь

$$\begin{cases} x_3 + 2x_4 = 11 \\ x_1 + 3x_2 = 7 \\ x_2 + x_4 = 6 \\ x_1 + x_3 = 4 \end{cases}.$$

Необхідний для розв'язання перелік команд:

```
>> A=[0 0 1 2; 1 3 0 0; 0 1 0 1; 1 0 1 0]; B=[11; 7; 6; 4];
>> lsqr(A,B,.1e–6)
lsqr stopped at iteration 4 without converging to the desired tolerance 1e-007
because the maximum number of iterations was reached.
The iterate returned (number 4) has relative residual 1.9e-013
ans =
   1.0000
   2.0000
   3.0000
   4.0000
```

В прикладі процес ітерацій збігається на четвертому кроці з відносною похибкою **tol =1.9e–13**.

Вправа 9. Розв'язання системи лінійних алгебраїчних рівнянь за допомогою методу найменших квадратів з обмеженнями.

Розглянемо команди, які використовуються при розв'язування систем лінійних рівнянь методом найменших квадратів з обмеженнями:

- **X =lscov(A,B.V)** – структура розв'язання команди повертає точний розв'язок системи лінійних алгебраїчних рівнянь **A*X=B+e**, де – є вектор шумів, який характеризується коваріаційною матрицею **V** (**в**икористовується метод найменших квадратів у присутності шумів з відомою коваріацією). Прямокутна матриця **A** повинна мати розмір **m×n**, де **m>n**. Розв'язання відшукується у вигляді **X=inv(A'*inv(V) *A) *A'* inv(V)*B**;

- **[X, dX] = lscov(A,B,V)** – структура розв'язання команди повертає крім шуканого результату (вектора **X**) також повертає стандартну похибку вектора **X**, що міститься в змінній **dX**;
- **X =isqnonneg(A,B)** – структура розв'язання системи лінійних алгебраїчних рявнянь **AX=B** методом найменших квадратів з додатковими обмеженнями, де **A** – дійсна прямокутна матриця, **B** – дійсний вектор. Вектор **X** містить невід'ємні елементи, **X**$\geq$**0**, де j = 1, 2... n;
- **X=Isqnonneg(A.B.X0)** – структура розв'язання системи лінійних алгебраїчних рявнянь з явно заданим невід'ємним початковим значенням **X0** для ітерацій.

Вправа 10. Визначення точності розв'язування лінійних алгебраїчних систем. Обумовленість матриць.

Метод Гауса є точним настільки, наскільки точно виконуються арифметичні операції, які використовуються в методі. Оскільки в системі **MATLAB** арифметичні операції в більшості випадків виконуються неточно, то розв'язання систем лінійних рівнянь в реальності є наближеним.

Стійкість розв'язання лінійної системи суттєво залежить від числа обумовленості матриці системи. Число обумовленості – важливий чинник, що характеризує точність розв'язування задачі.

Нехай задана матриця А. Числом обумовленості матриці А називається число $\mu(A) = \| A \| \| A^{-1} \|$.

Розглянемо лінійну систему $AX = B$. Природньо, що і задача розв'язання системи може бути як добре обумовленою, так і погано обумовленою. Припустимо, система рівнянь розв'язується з погрішністю на вхідних даних. Число обумовленості $\mu(A)$ визначає, наскільки погрішність вихідних даних може вплинути на розв'язування системи рявнянь. системи.

Якщо число обумовленості більше одиниці, то система вважається погано обумовленою, оскільки можливе сильне зростання погрішності результату.

Якщо відносне число обумовленості близьке до одиниці, то система рівнянь добре обумовлена.

Перед розв'язанням системи рівнянь варто обчислити відносне число обумовленністі матриці коефіцієнтів. У системі **MATLAB** ця процедура здійснюється за допомогою команди **cond(A).**

З'ясуємо, як впливають на розв’язання системи лінійних рівнянь (**AX=B**) невеликі зміни в матриці **B** або в елементах матриці **A**. Якщо такі перетворення приводять до незначної зміни розв’язку, можна з деякою упевненістю вважати, що і погрішності округлення не дуже вплинуть на остаточний результат розв’язання системи.

Приклад 1. Виконати розв’язання системи алгебраїчних рівнянь

$$\begin{cases} x_1 + 0.99x_2 = 1.99 \\ 0.99x_1 + 0.98x_2 = 1.97 \end{cases}.$$

Знайти розв’язок системи, якщо внести збурення у вектор **B**. Одержати відносну та абсолютну похибки та проаналізувати отримані результати.

Розв’язання цієї системи за допомогою команди лівого ділення (**X=A\B**) відповідає наступним операціям:

```
> A=[1 .99;.99 .98];B=[1.99;1.97]; X=A\B
X =
   1.0000
   1.0000
```

Таким чином, розв’язок системи: $x_1 = 1,\ x_2 = 1$.

Тепер незначно змінимо величину вільних членів системи:

$$\begin{cases} x_1 + 0.99x_2 = 1.989903 \\ 0.99x_1 + 0.98x_2 = 1.970106 \end{cases}.$$

Розв’язання цієї системи за допомогою тієї ж посілдовності:

```
>> B1=[1.989903;1.970106]; X1=A\B1
X1 =
   3.0000
  –1.0203
```

Проаналізуємо результат втручання в вихідні дані системи. Визначимо вектор приростів вільних членів $\Delta B = B - B1$:

```
>> Format long
DeltaB=Bи – B1
DeltaB =
  1.0e–003 *
   0.09700000000001
  –0.10599999999994
```

Будем використовувати поняття норми вектору. Нагадаємо, що норма вектора V визначається наступним чином: $\| V \|_p = (\sum | V |^p)^{1/p}$. Норму цього вектора можна обчислити за допомогою відповідної команди системи **MATLAB**:

```
>> norm(DeltaB)
ans =
   1.436836803537195e-004
```

Тобто, $\|\Delta B\| = norm(\Delta B) = 1.4368e-004$. Відносний приріст норми вільних членів $\delta B = \dfrac{\|\Delta B\|}{\|B\|}$:

PRB=norm(DeltaB)/norm(B)
PRB =
5.131232776079074e–005

Згідно даного результату можна констатувати, що відносний приріст вектора вільних членів відносно малий.

Визначимо вектор приростів вектора розв'язання δX:

DeltaX=X–X1
DeltaX =
– 1.99999999999877
2.02029999999875

Та відносний приріст вектора розв'язання $\delta X = \dfrac{\|\Delta X\|}{\|X\|}$:

>> PRX=norm(DeltaX)/norm(X)
PRX =
2.01017562541063

Таким чином, відношення відносних приростів, $\dfrac{\delta X}{\delta B}$:

>> PRX/PRB
ans =
3.917529594022168e+004

Очевидно, відношення відносних приростів, $\dfrac{\delta X}{\delta B} = 3.9175e+004$, значне за величиною. Такий результат обумовлений величиною визначника матриці **A**, хоча він відносно і невеликий:

>> disp(det(A))
–9.999999999998899e–005,

тобто, (det (A) =–le–004), але det (A) можна дещо збільшити (зробити рівним –1), для цього помножимо на 100 обидва боки рівнянь, що входять в систему (зауважимо, що ні розв'язок системи, ні відношення відносних приростів $\dfrac{\delta X}{\delta B}$ при цьому не змінюються).

Набагато важливішим показником є чинник обумовленості матриці А. Її число обумовленості можна знайти наступним чином:

>> cond(A)
ans =
3.920599997447899e+004,

що лише незначно перевищує знайдену вище величину відношення відносних приростів $\frac{\delta X}{\delta B}$ = 3. 9175e+004.

Практичні завдання лабораторної роботи № 7.

Виконати наступне завдання :

Визначити обумовленість матриці системи та розв’язати систему лінійних алгебраїчних рівнянь (варіанти систем задані в таблиці 7.1):

- за допомогою вирішувача;
- за допомогою підпакету **Symbolyc Math** системи **MATLAB**;
- методом Крамера;
- за допомогою **LU**-розкладання.

Таблиця 7.1 – Варіанти до завдання

№ п/п	Система рівнянь	№ п/п	Система рівнянь
1	$\begin{cases} 2x_1 + x_2 + 2x_3 + 3x_4 = 8 \\ 3x_1 + 3x_3 = 6 \\ 2x_1 - x_2 + 3x_4 = 4 \\ x_1 + 2x_2 - x_3 + 2x_4 = 4 \end{cases}$	9	$\begin{cases} 2x_1 + x_2 - 5x_3 + x_4 = -4 \\ x_1 - 3x_2 - 6x_4 = -7 \\ 2x_2 - x_3 + 2x_4 = 2 \\ x_1 + 4x_2 - 7x_3 + 6x_4 = -2 \end{cases}$
2	$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 22 \\ 2x_1 + 3x_2 + x_3 + 2x_4 = 17 \\ x_1 + x_2 + x_3 - x_4 = 8 \\ x_1 - 2x_3 - 3x_4 = -7 \end{cases}$	10	$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 26 \\ 2x_1 + 3x_2 + 4x_3 + x_4 = 34 \\ 3x_1 + 4x_2 + x_3 + 2x_4 = 26 \\ 4x_1 + x_2 + 2x_3 + 3x_4 = 26 \end{cases}$
3	$\begin{cases} 9x_1 + 10x_2 - 7x_3 - x_4 = 23 \\ 7x_1 - x_3 - 5x_4 = 37 \\ 5x_1 - 2x_3 + x_4 = 22 \\ 4x_1 + x_2 + 2x_3 + 3x_4 = 26 \end{cases}$	11	$\begin{cases} 2x_1 - 8x_2 - 3x_3 - 2x_4 = -18 \\ x_1 - 2x_2 + 3x_3 - 2x_4 = 28 \\ x_2 + x_3 + x_4 = 10 \\ 11x_2 + x_3 + 2x_4 = 21 \end{cases}$
4	$\begin{cases} 6x_1 - x_2 + 10x_3 - x_4 = 158 \\ 2x_1 + x_2 + 10x_3 + 7x_4 = 128 \\ 3x_1 - 2x_2 - 2x_3 - x_4 = 7 \\ x_1 - 12x_2 + 2x_3 - x_4 = 17 \end{cases}$	12	$\begin{cases} 2x_1 - x_2 + 4x_3 + x_4 = 66 \\ 2x_2 - 6x_3 + x_4 = -63 \\ 8x_1 - 3x_2 + 6x_3 - 5x_4 = 146 \\ 2x_1 - 7x_2 + 6x_3 - x_4 = 80 \end{cases}$

5	$\begin{cases} x_1 - 2x_2 + 6x_3 + x_4 = 88 \\ 5x_1 + 2x_3 - 3x_4 = 88 \\ 7x_1 - 3x_2 + 7x_3 + 2x_4 = 181 \\ 3x_1 - 7x_2 + 5x_3 + 2x_4 = 99 \end{cases}$	13	$\begin{cases} 2x_1 - 3x_3 - 2x_4 = -16 \\ 2x_1 - x_2 + 13x_3 + 4x_4 = 213 \\ 3x_1 + x_2 + 2x_3 + x_4 = 72 \\ x_1 - 12x_3 - 5x_4 = -159 \end{cases}$

Закінчення таблиці 5.1

6	$\begin{cases} x_1 - 2x_2 - 8x_4 = -7 \\ x_1 + 4x_2 - 7x_3 + 6x_4 = -8 \\ x_1 + x_2 - 5x_3 + x_4 = -10 \\ 2x_1 - x_2 + 2x_4 = 7 \end{cases}$	14	$\begin{cases} 7x_1 + 7x_2 - 7x_3 - 2x_4 = 5 \\ 3x_1 + 4x_2 + 5x_3 + 8x_4 = 60 \\ 2x_1 + 2x_2 + 2x_3 + x_4 = 27 \\ 2x_1 - 2x_3 - x_4 = -1 \end{cases}$
7	$\begin{cases} 2x_1 + 2x_2 + 6x_3 + x_4 = 15 \\ -x_2 + 2x_3 + x_4 = 18 \\ 4x_1 - 3x_2 + x_3 - 5x_4 = 37 \\ 3x_1 - 5x_2 + x_3 - x_4 = 30 \end{cases}$	15	$\begin{cases} 6x_1 - 9x_2 + 5x_3 + x_4 = 124 \\ 7x_1 - 5x_3 - x_4 = -54 \\ 5x_1 - 5x_2 + 2x_3 + 4x_4 = 83 \\ 3x_1 - 9x_2 + x_3 + 6x_4 = 45 \end{cases}$
8	$\begin{cases} 4x_1 - 5x_2 + 7x_3 + 5x_4 = 165 \\ 2x_1 + x_2 - 3x_3 - x_4 = -15 \\ 9x_1 + 4x_3 - x_4 = 194 \\ x_1 - x_2 - 2x_3 - 3x_4 = 19 \end{cases}$		

Контрольні питання

1. Визначити поняття прямих і ітераційних методів?
2. Що означає процедура символьного розв’язання рівняння або системи рівнянь?
3. Назвіть точні методи розв’язання систем лінійних рівнянь.
4. Для яких систем рівнянь застосовують метод Холецького? Який синтаксис такої відповідної команди у системі у **MATLAB?**
5. У чому зміст методу Гауса. Як цей метод реалізован у системі **MATLAB**?
6. Які форми ітераційних процесів вам відомі?
7. За допомогою якої команди у системі **MATLAB** можна розв’язувати перевизначені системи лінійних рівнянь? Який метод лежить в основі виконання цієї команди?
8. Як вирішити алгебраїчну систему рівнянь з обмеженнями за допомогою засобів **MATLAB**?

Лабораторна робота № 8.

Тема: Розв’язання нелінійних рівнянь та систем нелінійних рівнянь у системі **MATLAB**.

Мета роботи: ознайомитись з деякими методами розв’язування нелінійних рівнянь та систем нелінійних рівнянь.

Теоретичний мінімум

Нелінійні рівняння можна розділити на два класи — алгебраїчні та трансцендентні [24].

Визначимо дві групи методів розв’язання нелінійних рівнянь:

- точні методи;
- ітераційні методи.

Точні методи дозволяють записати корені у вигляді деякого скінченого співвідношення (формули). В курсі алгебри з такими методами ми зустрічалися при розв’язуванні тригонометричних, логарифмічних, показникових, а також алгебраїчних нелінійних рівнянь.

Вправа 1. Розв’язування нелінійних рівнянь на основі методу бісекції.

Більшість трансцендентних рівнянь, також як і систем цих рівнянь, не мають аналітичних розв’язків. Доведено, що не можна побудувати формулу, за якою можна знайти корені довільного алгебраїчного рівняння степені вище ніж четверта. Крім того, в деяких випадках рівняння містять коефіцієнти, які відомі лише наближено, і, отже, сама задача про точне визначення коренів рівняння втрачає сенс. Для розв’язування таких задач використовуються ітераційні методи із заданим ступенем точності.

Нагадаємо, що всяке значення $x=\xi$, що обертає функцію $f(x)$ в нуль, тобто $f(\xi)=0$, називається коренем рівняння

$$f(x)=0 \qquad (1)$$

або нулем функції $f(x)$.

Ітераційний процес полягає в послідовному уточненні початкового наближення $x(0)$ можливого значення кореня рівняння (1). Кожний такий крок називається ітерацією. В результаті ітерацій знаходиться послідовність наближених значень кореня $x(1), x(2), ..., x(n)$. Якщо ці значення із збільшенням числа ітерацій n наближаються до точного значення кореня, то кажуть, що ітераційний процес збігається.

Передбачаємо, що функція $f(x)$ в рівнянні (1) задовільняє наступним вимогам:

1) функція $f(x)$ неперервна на відрізку $[a,b]$ разом з своїми похідними 1-го і 2-го порядку;

2) значення функції $f(x)$ на кінцях відрізку мають різні знаки, тобто $f(a)*f(b)<0$;

3) перші та другі похідні функції, $f'(x)$ та $f''(x)$, зберігають певний знак на всьому відрізку $[a, b]$.

Умови 1) і 2) гарантують, що на інтервалі $[a,b]$ знаходиться хоча б один корінь, а з умови 3) витікає, що функція $f(x)$ на даному інтервалі є монотоною і тому корінь рівняння (1) буде єдиним.

Задача знаходження кореня рівняння $f(x)=0$ ітераційним методом вирішується в два етапи:

1. Визначаються корені, тобто відшукується наближене значення кореня або відрізка, що його містить;
2. Уточнюється наближене значення коренів, тобто наближені величини коренів доводяться до заданої степені точності.

Передбачимо, що наближене значення кореня $\bar{x}$ необхідно отримати з точністю $\boldsymbol{\varepsilon}$, тобто знайти таке значення $\bar{x}$, щоб для точного значення кореня x мала місце наступна нерівність: $|x-\bar{x}|<\varepsilon$.

Процес визначення коренів починається зі встановлення знаків функції $f(x)$ в граничних точках області її існування: $x=a$ та $x=b$.

Вирішення рівняння (1) ітераційним методом передбачає встановлення відповідей на питання: чи має рівняння (1) корені і скільки їх існує, а також виявлення процедури знаходження значень вказаних коренів з потрібною точністю.

Для наближеного знаходження коренів рівняння (1) можна застосовувати метод бісекції.

Нехай $[a,b]$ — відрізок, що містить шуканий корінь. Допустимо, що функція $f(x)$ неперервна на відрізку $[a,b]$ і на кінцях відрізка приймає значення різних знаків: $f(a)\cdot f(b)<0$. Алгоритм методу бісекції полягає в побудові послідовності вкладених відрізків, на кінцях яких функція приймає значення різних знаків. Кожен наступний відрізок отримують в результаті ділення попереднього відрізку навпіл.

Нехай на k-ому кроці даного алгоритму знайдений такий відрізок $[a^{(k)}, b^{(k)}]$, що $f(a^{(k)})\cdot f(b^{(k)})<0$. Знайдемо середину цього відрізку $c^{(k)}=\dfrac{a^{(k)}+b^{(k)}}{2}$.

Якщо $f(c^{(k)})=0$, то $c^{(k)}$ буде шуканим коренем.

Якщо ця умова не виконується, то з двох частин відрізку вибираємо ту, на кінцях якої функція $f(x)$ має протилежні знаки. Вибираємо $[a^{(k+1)},c^{(k)}]$, коли $f(a^{(k)})*f(c^{(k)})<0$, де $a^{(k+1)}=a^{(k)}$, $b^{(k+1)}=c^{(k)}$, або відрізок $[c^{(k)},b^{(k+1)}]$, коли $f(c^{(k)})*f(b^{(k)})<0$, де $a^{(k+1)}=c^{(k)}$, $b^{(k+1)}=b^{(k)}$.

Наближені значення коренів можуть бути встановлені з фізичного формування задачі, з процедури розв'язування аналогічної задачі при інших початкових даних або за допомогою графічного аналізу, що застосовують для визначення наближених коренів.

Приклад 1. Знайти корені рівняння $f(x)=x^3-10x-10=0$.

Множину існування функції $f(x)$ розбиваємо на підмножини:

$$(-\infty,-4],\ (-4,-2],\ (-2,0],\ (0,2],\ (2,4],\ (4,+\infty).$$

Встановимо знаки функції $f(x)$ в граничних точках підмножин у випадках, коли $x\to-\infty$ та $x\to+\infty$:

x	$-\infty$	–4	–2	0	2	4	$+\infty$
$f(x)$	–	–	+	–	–	+	+

З даної таблиці витікає, що корені функції можуть міститися в настуних відрізках [–4, –2], [–2, 0] і [2, 4].

Зауважимо, що дійсні корені рівняння (1) — це точки перетину графіка функції $f(x)$ з віссю абсцис. Процедура встановлення координат коренів передбачає побудову графіка функції $f(x)$, визначення точок перетину графіка функції $f(x)$ з віссю Ох або визначення на вісі Ох відрізків, що містять один нуль функції.

Тепер розглянемо розв'язання рівняння $f(x)=0$: $x^3-10x-10=0$ за допомогою команд системи **MATLAB**. Для завдання інформації про функцію $f(x)$ використовуємо команду **inline**. Локалізуємо корні рівняння:

```
>> f = inline('x.^3–10.*x–10');
```

Побудуємо графік функції на відрізку [–10,10]:

```
>> x=linspace(–10,10,100);
>> plot(x,f(x));grid
```

Визначаємо відрізки, що вміщують можливі нулі функції і будуємо графікі функції $f(x)$ послідовно на визначених відрізках [–4, –2], [–2, 0] і [2, 4]. Визначаємо корінь на відрізку [–4, –2]:

```
>> x=linspace(–4,–2,100);
>> plot(x,f(x));grid
```

Визначаємо корінь на більш меньшому відрізку [–2.6,–2.4]:

```
>> x=linspace(–2.6,–2.4,1000); plot(x,f(x));grid
```

Шуканий корінь відрізку [–4, –2]:

x1=–2.425.

Робимо перевірку знайденого кореня:

```
>> f(–2.425)
ans =
  –0.0105
```

У такій же спосіб знайдемо інший корінь. Визначаємо корінь на відрізку [–2, 0]:

```
>> f = inline('x.^3–10.*x–10');
x=linspace(–4,–2,100);
plot(x,f(x));grid
>> x=linspace(–2,0,100);
>> plot(x,f(x));grid
>> x=linspace(–1.2,–1,100);
>> plot(x,f(x));grid
>> x=linspace(–1.16,–1.15,100);
>> plot(x,f(x));grid;
>> f(–1.155)
```

Перевіримо знайдений корень:

```
ans =
   0.0092
```

Знаходження третього кореня. Визначаємо корінь на відрізку [2, 4]:

```
>> x=linspace(2,4,100);
>> plot(x,f(x));grid
>> x=linspace(3.4,3.6,100);
>> plot(x,f(x));grid
```

Перевіряємо знайдений корінь:

```
>> f(3.5771)
ans =
  2.9962e–004
```

Побудову графіків часто вдається спростити, якщо замінити ліву частину рівняння (1) рівнозначною функцією $f(x) \equiv f_1(x) + f_2(x)$. Тоді рівняння (1) набуває вигляду

$$f_1(x) + f_2(x) = 0,$$

де функції $f_1(x)$ і $f_2(x)$ – значно простіші, ніж функція $f(x)$. Якщо побудувати графіки функцій $y = f_1(x)$ і $y = -f_2(x)$, то шукані корені визначаються як абсциси точок перетину вказаних графіків.

Приклад 2. Треба визначити корені рівняння: $x \lg x = 1$.

Рівняння зручно переписати у вигляді рівності: $\lg x = \dfrac{1}{x}$. Звідси ясно, що корені рівняння можуть бути знайдени як абсциси точок перетину логарифмічної кривої $y = \lg x$ і гіперболи $y = \dfrac{1}{x}$. Після побудови графіків, можна наближено встановити координату точки перетину, тобто координату шуканого кореня вихідного рівняння $x \lg x = 1$, $x \approx 1.76$ або визначити відрізок [1.75, 1.78], який містить в собі цей корінь.

Реалізуємо цю процедуру знаходження кореня за допомогою команд системи **MATLAB**:

```
>> x=1.5:.01:2;
>> f1=log(x);f2=1./x;plot(x,f1,x,f2);grid
```

Локалізуємо корінь:

```
>> x=1.75:.001:1.78;
>> f1=log(x);f2=1./x;plot(x,f1,x,f2);grid
```

Приклад 3. Знайти корені рівняння $f(x)=0$, де $f(x)=\dfrac{x^2-6}{x+6}$.

Пошук кореня за процедурою з застосуванням команд системи **MATLAB:**

```
>> x=– 5:1:5; f1=x.^2–6;f2=x+6;
>> plot(x,f1,x,f2);grid
```

Приклад 4. Знайти корені рівняння $x^2+6=e^{-x}$.

Пошук кореня за процедурою з застосуванням команд системи **MATLAB:**

```
>> x=–3:.1:–1; f1=exp(–x);f2=x.^2+6;
>> plot(x,f1,x,f2);grid
```

Вправа 2. Розв’язування нелінійних рівнянь за допомогою пошуку нулів функції.

Для розв’язання рівнянь виду $f(x)=0$ (тобто для знаходження нулів функції) існує команда **fzero**, яка дозволяє знаходити наближене значення кореня за заданим початковим значенням [44]:

- **z=fzero(fun ',x0),** структура команди дозволяє знаходить нуль функції в околі точки **x0**;
- **z=fzero(fun,x0,tol),** структура команди **fzero** повертає результат з відносною похибкою **tol**, величина якої задається користувачем (у випадку замовчування **tol=eps**);
- **z=fzero(fun,x0,tol,trace)** структура команди **fzero** дозволює видавати на екран поточні послідовні значення процедури пошуку нуля функції.

За допомогою цієї команди **fzero** реалізуються метод ділення відрізка навпіл, метод січних та зворотньої квадратичної інтерполяції.

Параметр **fun** команди **fzero** може бути заданий одним з наступних способів:

- як формула з невідомим **x**, яка ув’язнена в апострофи;
- як ім'я **m**-файлу (у апострофах і без розширення);
- як показник на функцію (наприклад **@fun**_name);

Формула, ув’язнена в апострофи, розглядається як незалежна змінна і може містити тільки **x**. Використання незалежної змінної з іншим ім'ям приведе до появи інформації про помилку.

Параметр **x0** команди **fzero** може бути заданий одним з двох способів:

• у вигляді вектора **[a b]**, що задає інтервал **(a<b)**, на кінцях якого параметр **fun** має різні знаки, що гарантує знаходження, принаймні одного кореня на цьому інтервалі;

• у вигляді скалярного значення, в околі якого передбачається знаходження шуканого кореня; в цьому випадку команда **fzero** сама намагається знайти відрізок з центром в заданій точці **x0**, на кінцях якого параметр **fun** має різні знаки.

Щоб полегшити роботу по вибору початкового наближення, рекомендується побудувати графік функції **y=fun(x).**

Команда **fzero** може повертати ще два вихідних параметра **e_flag, inform: [x,f,e_flag,inform] = fzero(fun,x0).**

Додатні значення **e_flag** (зазвичай це 1) означає, що вдалося знайти інтервал, на кінцях якого параметр **fun** має різні знаки. Якщо такий інтервал не виявлений, то **e_flag= –1**.

Структура **inform** містить три поля з іменами **iterations, funcCount** і **algorithm**.

У першому полі записується кількість ітерацій, що були виконані під час пошуку шуканого кореня; у другому полі — кількість звернень до параметру **fun**, в третьому полі — найменування алгоритму, що був використаний для знаходження кореня.

Приклад 1. Знайти корінь рівняння $5xe^{-x}+5\cos x=0$ на відрізку $[-1,1]$.

Застосуємо команду **fzero** з додатковими параметрами виходу:

```
>> [x,f,e_flag,inform]=fzero('5.*x.*exp(–x)+5*cos(x)',[–1,1])
x =
  –0.5178
f =
    0
e_flag =
    1
inform =
   iterations: 11
    funcCount: 11
    algorithm: 'bisection, interpolation'
```

Приклад 2. Знайти корені рівняння $x^3-10x-10=0$.

Скористаємось командою **fplot** для побудови графіку вказанної функції на відрізку [–4 4]:

```
>> fplot('x.^3–10.*x–10',[–4 4]);
>> grid
```

```
>> x1=fzero('x.^3–10.*x–10', –2.5)
x1 =
  –2.4236
```

Обчислення точності знайденого рішення:

```
>> disp(eps)
 2.2204e–016
```

Задана машинна точність:

```
>> f(x1–eps)*f(x1+eps)
ans =
    0
>> [x2,f]=fzero('x.^3–10.*x-10',-1.5)
x2 =
  –1.1535
f =
 1.7764e-015
>> [x3,f,e_flag,inform]=fzero('x.^3–10.*x–10',3.5)
x3 =
   3.5771
f =
–1.4211e–014
e_flag =
    1
inform =
   iterations: 8
    funcCount: 8
    algorithm: 'bisection, interpolation'
```

Зауважимо, що точність знаходження коренів у системі **MATLAB** достатньо висока. Користувачеві надається можливість формувати різні умови припинення ітераційного процесу – згідно точності обчислення координати **x**, по модулю значення параметра **fun**, згідно кількості звернень до параметру **fun** і т.д.

Умова виявлення інтервалу, на кінцях якого функція приймає значення різних знаків, є принциповим для алгоритму, який використовується командою **fzero**. Навіть у таких тривіальних рівняннях, як $x^4 = 0$, зазвичай не вдається знайти шуканий розв’язок:

```
>> [x,f,e_flag,inform]=fzero('x.^4',[–1,1])
??? Error using ==> fzero
The function values at the interval endpoints must differ in sign.
```

Відмова щодо виконання команди відбулася через те, що на обох кінцях заданого інтервалу функція приймає додатні значення. Якщо початкове наближення "випадково" співпадає з коренем, то команда **fzero** повертає це випадкове значення (яке може збігатись з розв’язком):

```
>> [x,f]=fzero('x.^4',0)
x =
    0
f =
    0
```

Проте зміна стартової точки знову приводить алгоритм до невизначеності. Спроба розширити інтервал з центром в точці **x=1** приводить до появи нескінченно великих значень функції, але інтервал, на якому відбулася би зміна знаку, так може і не бути виявлений (звернить увагу на значення вихідного параметра **e_flag**):

```
>> [x,f,e_flag]=fzero('x.^4',1)
Exiting fzero: aborting search for an interval containing a sign change
    because NaN or Inf function value encountered during search
(Function value at –1.482139e+077 is Inf)
Check function or try again with a different starting value.
x =
   NaN
f =
   NaN
e_flag =
    –1
```

Існує ще один спосіб вирішення рівнянь – це графічне знаходження коренів рівняння за допомогою команди **ginput**.

Приклад 3. Розв'язати нелінійне рівняння $x \sin x^2 = 0$.

Процедура знаходження коренів визначається наступними командами:

```
>> x=0:.01:3;
>> f=x.*sin(x.^2);
>> plot(x,[f;0*f]); grid
>> ginput
```

Після виконання цих команд з'являється графічне вікно з двома рухомими лініями, які перетинаються під кутом 90 градусів. За допомогою цих ліній можна визначити точки перетину. Як тільки усі точки були визначені, треба ініціювати на клавіатурі клавішу **ENTER**, після чого наближено обчислюються шукані корені:

```
ans =
    2.5127   –0.0088
    1.7730   –0.0088
    0.0035   –0.0088
```

Вправа 3 Розв’язання системи нелінійних рівнянь за допомогою команди **fsolve.**

Формально задача пошуку розв’язку системи нелінійних рівнянь

$$\begin{cases} f_1(x_1, x_2, \ldots, x_n) = 0 \\ f_2(x_1, x_2, \ldots, x_n) = 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ f_n(x_1, x_2, \ldots, x_n) = 0 \end{cases}$$

може бути представлена в еквівалентній форми, як задача пошуку кореня одного рівняння, де $f(x) = (f(x)_1, f_2(x), \ldots, f_n(x))$ і $x = (x_1, x_2, \ldots, x_n)$.

Команда **fsolve** призначена для розв’язання систем нелінійних рівнянь виду $f(x) = 0$, де x — вектор невідомих, а f — функція, значенням якої є вектор або матриця. Алгоритм роботи команди **fsolve** використовує початкове значення x_0 і базується на мінімізації суми квадратів складових функції $f(x) = (f(x)_1, f_2(x), \ldots, f_n(x))$ методами Гауса-Ньютона і Льовенберга-Марквардта [46]. У простому випадку структура звернення до команди **fsolve** має вигляд**: X = fsolve(F,X0).**

Команду **fsolve** можна використовувати і як альтернативу функції **fzero** при знаходженні кореня нелінійного рівняння, наприклад, $\cos x = 0$:

```
>> x=fsolve(@cos,1)
Optimization terminated successfully:
 First-order optimality is less than options.TolFun.
x =
   1.5708
```

Другий параметр команди **fsolve** може бути заданий у вигляді вектора початкових значень, і для кожного компоненту цього вектора будуть знайдені наближені розв’язки. Наприклад:

```
>> x=fsolve(@cos,[1 2 3 4 5],optimset('fsolve'))
Optimization terminated successfully:
 First-order optimality is less than options.TolFun.
x =
   1.5708    1.5708    1.5708    4.7124    4.7124
```

На відміну від **fzero** команда **fsolve** може знайти наближений розв’язок для вище розглянутого рівняння $x^4 = 0$:

```
>> x=fsolve('x.^4',1,optimset('Display','off'))
x =
   0.1001
```

У випадку розв’язування рівняння $tgx = 0$ команда **fsolve** може знайти наближений розв’язок і не схибити в околі точки розриву функції:

```
>> x=fsolve(@tan,1,optimset('Display','off'))
x =
  2.3205e–010
```

Сутність параметра **'Display'** та **'off'** полягає в викресленні "зайвих" повідомлень.

Головним призначенням команди **fsolve** є розв’язок систем нелінійних рівнянь.

Приклад 1. Розв’язати систему рівнянь: $\begin{cases} x_1 = -x_2 + \sin(\pi x_1) \\ x_1 = \ \ x_2 + \cos(\pi x_2) \end{cases}$.

Випишимо **m**-файл **funsc**, значення якого сформуємо у вигляді вектора-стовпця:

```
function y = funsc(x)
y=[x(1)+x(2)–sin(pi*x(1)); x(1)–x(2)–cos(pi*x(1))];
```

Використовуємо команду **fsolve.** Зауважимо, що кожне звернення до цієї команди передбачає завдання різних початкових значень **(xl, x2)**. Оскільки команда **fsolve** подібно команді **fzero** повертає і вектор-стовпець значень функцій в знайденій точці, то можна сформувати два вихідних параметра, щоб оцінити точність розв’язання. Звернемо увагу на те, що координати початкової точки теж представлені у вигляді вектора-стовпця:

```
>> [x,f] = fsolve(@funsc,[0.2;1],optimset('Display','off'))
x =
   0.5000
   0.5000
f =
  1.0e–007 *
   0.2139
  –0.0000
```

Результати подальших звернень до команди **fsolve** з різними початковими значеннями (x1, x2) наведені в таблиці 8.1:

Таблиця 8.1 – Результати обчислень

№ п/п	x0[x1;x2]		Розв'язок системи		Значення функції (з поточною точністю)	
	x1	x2	x1	x2	y1	y2
1	0.2	1	0.5	0.5	0.2139e-007	0
2	1	0.5	0.5	0.5	0.4694e-008	0
3	1	–0.2	0.5	0.5	0.9686e-012	–0.0012e-0.12
4	–0.2	0	–0.5	–0.5	–0.4223e-007	0.0001e-007
5	–0.5	–1	–0.5	–0.5	0	–0.6123e-016
6	–1	–2	–0.5	–0.5	–0.1604e-006	0.0001e-006

Розібратися з виписаними ситуаціями допоможуть графіки функцій, які отримані шляхом розв’язування рівнянь вихідної системи відносно відповідних змінних:

```
>>axes('Xlim', [–1 1.5] , 'Ylim', [–1 1.5]);axis equal; grid on; hold on
x1=–1:0.1:1.5; y=sin(pi*x1)–x1; plot(x1,y,'g-');
x2=–1:0.1:1.5; y=cos(pi*x2)+x2; plot(y,x2,'r:');
xlabel('x1'); ylabel('x2');
title('Розв’язок нелінійної системи','FontName','Courier');
```

Розв’язок системи – точки перетинання двох графіків.

Серед вихідних параметрів команди **fsolve** можуть з’явитися такі ж за змістом параметри, які дозволяють використовувати структуру команди **fzero**: **[x,f,exitflag,output] = fsolve(fun,xO).**

Значення параметру **exitflag=1** свідчить про те, що розв’язок системи знайдений. При **exitfiag=0** розв’язок не був знайдений, оскільки ітераційний процес був перерваний у зв'язку з досягненням максимальної кількості ітерацій або звернень до параметру **fun**. При **exitfiag= –1** досягнутий мінімум не є розв’язком системи.

У структурі **output** з'являються додаткові поля з іменами:

- **stepsize**, в якому видається величина кроку при завершенні пошуку;
- **firstorderopt**, в якому видається точність, досягнута згідно значенням функції.

Приклад 2. Розв’язати систему приклада 1 $\begin{cases} x_1 = -x_2 + \sin(\pi x_1) \\ x_1 = \ \ x_2 + \cos(\pi x_2) \end{cases}$ з додатковіими вихідними параметрами e_flag,inform.

Процедура розв’язання використовує наступні команди:

```
>> [x,f,e_flag,inform] = ...
fsolve(@funsc,[1;0.5],optimset('Display','off'))
x =
    0.5000
    0.5000
f =
  1.0e–008 *
    0.4694
   –0.0000
e_flag =
     1
inform =
       iterations: 5
        funcCount: 15
        algorithm: 'trust-region dogleg'
    firstorderopt: 4.6948e–009
```

Практичні завдання лабораторної роботи № 8.

Виконати наступні завдання :

Завдання 1. Розв'язати символьним методом систему рівнянь (див табл. 8.2):

Таблиця 8.2 – Варіанти до завдання № 1

№ п/п	Система нелінійних рівнянь	№ п/п	Система нелінійних рівнянь
1	$\begin{cases} \cos(x-1)+y=0.5 \\ x-\cos y=3 \end{cases}$	9	$\begin{cases} x+\cos(y-1)=0.8 \\ y-\cos x=2 \end{cases}$
2	$\begin{cases} \sin(x-0.6)-y=1.6 \\ 3x-\cos y=3 \end{cases}$	10	$\begin{cases} 5y=6x+3 \\ \sin(x-y)-xy+1=0 \end{cases}$
3	$\begin{cases} 3ey=5x+2 \\ \sin(x+y)-1.5xy=0 \end{cases}$	11	$\begin{cases} 5x+3y=0 \\ \sin(x-y)-xy=0 \end{cases}$
4	$\begin{cases} \ln x+(x+1)^3=0 \\ \sin(x+1)-y=1.2 \end{cases}$	12	$\begin{cases} \cos(x-1)+y=0.5 \\ y-\cos x=3 \end{cases}$
5	$\begin{cases} y+\lg(1+x)=1.5 \\ \sin x+2y=2 \end{cases}$	13	$\begin{cases} \cos(x-1)+y=1 \\ \sin y+2x=1.6 \end{cases}$
6	$\begin{cases} 2arctgx-x+3=0 \\ y=1+e-x \end{cases}$	14	$\begin{cases} y-3\cos= \\ y=1-2e-xy \end{cases}$
7	$\begin{cases} x+4y=0 \\ \sin x-y^2=0 \end{cases}$	15	$\begin{cases} 2ex+5x-6=0 \\ y=1/(x^2+1) \end{cases}$
8	$\begin{cases} x^2+y^2=5 \\ y=e-xy \end{cases}$		

Завдання 2. Відшукати найменший за модулем дійсний корінь рівняння $f(x)=0$ з припустимою похибкою не більшою ніж $\varepsilon=10^{-4}$ (див. □ абл 8.3).

Таблиця 8.3 – Варіанти до завдання № 2

№ п/п	Рівняння завдання	№ п/п	Рівняння завдання
1	$x^2-2x+\ln(x)=0;$	9	$x^4-6x^2+12x-8=0;$
2	$x^2-2\ln(x+2)=0;$	10	$2^x+2x^2-3=0;$

Закінчення таблиці 8.3

3	$x^3+2x-13=0;$	11	$x^2+arctg(x)-0.5=0;$
4	$x\cdot e^{2x}-4=0;$	12	$ctg(0{,}8x)-2x^2=0;$
5	$x^5+5x+1=0;$	13	$x^5+18x^3-34=0;$
6	$(x-2)^2-e^x=0;$	14	$2x\cdot e^{x^2}-5=0;$
7	$x^3+2x^2-11=0;$	15	$2e^{-x^2}-3x+4=0;$
8	$x^2-1-\cos(1{,}2x)=0;$	16	$2x-3\sin(2x)-1=0;$

Завдання 3. Розв'язати систему нелінійних рівнянь за допомогою команд для побудови графіків (див. табл. 8.4).

Таблиця 8.4 – Варіанти до завдання № 3

№ п/п	Система нелінійних рівнянь	№ п/п	Система нелінійних рівнянь
1	$\begin{cases}\sin x+2y=2\\ \cos(y-1)+x=0.7\end{cases}$	9	$\begin{cases}\sin y+x=-0{,}4\\ 2y-\cos(x+1)=0\end{cases}$
2	$\begin{cases}\sin(x+0{,}5)-y=1\\ \cos(y-2)+x=0\end{cases}$	10	$\sin(x+2)-y=1{,}5$ $\cos(y-2)+x=0{,}5$
3	$\begin{cases}\cos x+y=1{,}5\\ 2x-\sin(y-0{,}5)=1\end{cases}$	11	$\begin{cases}\cos(x+0{,}5)-y=2\\ \sin y-2x=1\end{cases}$
4	$\begin{cases}\cos(x+0{,}5)+y=0{,}8\\ \sin y-2x=1{,}6\end{cases}$	12	$\begin{cases}\cos(x-2)+y=0\\ \sin(y+0{,}5)-x=1\end{cases}$
5	$\begin{cases}\sin(x-1)=1{,}3-y\\ x-\sin(y+1)=0{,}8\end{cases}$	13	$\begin{cases}\cos(x+0{,}5)+y=1\\ \sin(y+0{,}5)-x=1\end{cases}$
6	$\begin{cases}\cos(x+0{,}5)+y=1\\ \sin y-2x=2\end{cases}$	14	$\begin{cases}\sin x-2y=1\\ \cos(y+0{,}5)-x=2\end{cases}$
7	$\begin{cases}-\sin(x+1)+y=0{,}8\\ \sin(y-1)+x=1{,}3\end{cases}$	15	$\begin{cases}2y-\sin(x-0{,}5)=1\\ \cos y+x=1{,}5\end{cases}$
8	$\begin{cases}\sin(x)-2y=1\\ \sin(y-1)+x=1{,}3\end{cases}$		

Контрольні питання

1. Дати визначення поняття нелінійного алгебраїчного рівняння. Що означає: відшукати розв'язок нелінійного рівняння $f(x)=0$? Яким умовам повинна відповідати функція $f(x)$ і що вони гарантують?
2. На які етапи поділяється процес розв’язування нелінійного рівняння?
3. У чому полягає сутність методу ітерацій при розв'язуванні нелінійного рівняння? Які переваги і недоліки у цьому методі?
4. Наведіть процедуру відшукування кореня нелінійного алгебричного рівняння у системі **MATLAB**?
5. Які методи розв’язування нелінійних рівнянь у системі **MATLAB** Вам відомі?
6. З яких етапів складається задача знаходження нулів функції $f(x)$ у системі **MATLAB?**
7. Які функції для розв’язування нелінійного рівняння в системі **MATLAB** Ви знаєте?
8. Назвіть функції для розв’язування системи нелінійних рівнянь в **MATLAB** і особливості їх застосування.
9. Як розвязується нелінійне рівняння або система нелінійних рівнянь у символьному вигляді в системі **MATLAB**?

Лабораторна робота № 9.

Тема. Розв’язання задач двовимірної евклідової геометрії у системі **MATLAB**

Мета роботи: навчитись вирішувати завдання з планіметрії за допомогою команд системи **MATLAB**.

Теоретичний мінімум

Лінія на площині розглядається (задається) як множина точок, які мають деякі тільки їм характерні геометричні властивості [30]. Наприклад, коло радіусу R є множина всіх точок площини, видалених на відстань R від деякої фіксованої точки O (центру кола).

Введення на площині системи координат дозволяє визначати положення точки площини завданням двох чисел, які визначають її координати, а положення лінії на площині потребує завдання визначеного рівняння (тобто закономірності, що зв'язує координати точок лінії).

Рівнянням лінії (або кривої) на площині xOy називається таке рівняння $F(x, y) = 0$ з двома змінними, якому задовольняють координати x і y кожної точки лінії і не задовольняють координати будь-якої точки, що не належить цій лінії. Змінні x і y в рівнянні лінії називаються *поточними координатами* точок лінії. Рівняння $F(x, y) = 0$ визначає змінну y як функцію від змінної x. Таким чином визначена функція називається *неявно заданою*.

Дослідження геометричних властивостей ліній здійснюється в результаті аналізу рівнянь ліній, що є звичайною процедурою для систем типу **MATLAB.**

Завдання про знаходження точок перетину двох ліній, які описуються рівняннями $F_1(x, y) = 0$ і $F_2(x, y) = 0$, зводиться до відшукання точок, координати яких задовольняють рівнянням обох ліній, тобто зводиться до розв’язання системи двох рівнянь з двома невідомими:

$$\begin{cases} F_1(x,y) = 0 \\ F_2(x,y) = 0 \end{cases}$$

Якщо ця система не має дійсних розв’язків, то лінії не перетинаються.

Аналогічним чином вводиться поняття рівняння лінії в полярній системі координат. Рівняння $F(r, \varphi) = 0$ називається рівнянням даної лінії в полярній системі координат, якщо координати будь-якої точки, яка належить цій лінії, і лише вони, задовільняють цьому рівнянню.

Лінію на площині можна також задати за допомогою двох рівнянь:

$$\begin{cases} x = x(t) \\ y = y(t) \end{cases}, \qquad (1)$$

де x і y – координати довільної точки $M(x, y)$, яка належить даній лінії, а параметр t визначає положення цієї точки $M(x, y)$ на площині. Такий спосіб завдання лінії називається *параметричним*, а вищевказана система рівнянь — *параметричними рівняннями лінії*.

Лінію на площині можна також визначити за допомогою радіуса-вектора $\vec{r} = x(t)\vec{i} + y(t)\vec{j}$, де t – скалярний змінний параметр, $\vec{i}, \vec{j}$ – базісні вектори. Кожному значенню t відповідає певний вектор площини. При зміні параметра t кінець вектора $\vec{r}$ описує деяку лінію. Векторному рівнянню лінії в системі координат Oxy відповідають два скалярні рівняння (1), тобто проекції на вісі координат векторного рівняння лінії.

Векторне рівняння і параметричні рівняння лінії мають механічний зміст. Якщо точка переміщується на площині, то вказані рівняння називаються *рівняннями руху*, а лінія – *траєкторією руху* точки, параметр t визначає час.

Отже, у аналітичній геометрії на площині виникають два основні завдання:

- за інформацією про геометричні властивості кривої здійснити опис кривої за допомогою рівняння;
- по заданому рівнянню кривої проаналізувати її геометричні властивості і відтворити її графік.

Будь-яка пряма лінія на площині може бути задана рівнянням першого порядку

$$Ax + By + C = 0,$$

причому сталі A, B не дорівнюють одночасно нулю, тобто $A^2 + B^2 \neq 0$. Це рівняння першого порядку називають *загальним рівнянням прямої*.

Рівняння

$$y = kx + b. \qquad (2)$$

є рівнянням прямої з кутовим коефіцієнтом k.

Рівняння

$$y - y_0 = k(x - x_0) \qquad (3)$$

є рівнянням прямої, що проходить через задану точку у визначеному за допомогою кутового коефіцієнта напрямку.

Рівняння прямої, що проходить через дві задані точки (x_1, y_1) та (x_2, y_2) має вигляд:

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1} \qquad (4)$$

Рівняння $\frac{x}{a}+\frac{y}{b}=1$ називається *рівнянням прямої* у відрізках, де точки $(a, 0)$ і $(0, b)$ є точками перетину прямої з вісями координат.

Якщо дві прямі $y=k_1x+b_1,\ y=k_2x+b_2$ перетинають одна іншу, то гострий кут α між цими прямими визначатиметься за допомогою $tg\alpha=\left|\frac{k_2-k_1}{1+k_1k_2}\right|$. Дві прямі паралельні, якщо $k_1=k_2$. Дві прямі перпендикулярні, якщо $k_1=-1/k_2$.

Координати точки перетину двох прямих знаходяться шляхом розв'язання системи двох рівнянь, описують кожну пряму.

Пряма, що проходить через точку (x_1, y_1) і перпендикулярна до прямій $y=kx+b$ представляється рівнянням:

$$y-y_1=-\frac{1}{k}(x-x_1)$$

Відстань d від точки з координатами (x_0, y_0) до прямої $Ax+By+C=0$ визначається по формулі:

$$d=\frac{|Ax_0+By_0+C|}{\sqrt{A^2+B^2}}$$

Вправа 1. Використовування команди **fzero** для розв'язування задач аналітичної геометрії.

Нагадаемо рівняння деяких кривих на площини, що вивчалися в курсі вищої математики (дивись таблицю 9.1).

Таблиця 9.1 – Основні криві на площині

№ п/п	Назва	Канонічне рівняння
1	Коло	$(x-x_0)^2+(y-y_0)^2=R^2$, де точка (x_0, y_0) визначає координати центру, R –радіус кола.
2	Парабола	$(y-y_0)^2=2p(x-x_0)^2$, де точка (x_0, y_0) визначає координати вершини параболи
3	Еліпс	$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$, де a і b – сталі, що визначають довжину піввісей.
4	Гіпербола	$\frac{x^2}{a^2}-\frac{y^2}{b^2}=1$, де a і b – сталі, що визначають довжину піввісей.

Як вже відомо з попередніх лабораторних робіт для розв'язку рівнянь або систем рівнянь в системі **MATLAB** використовуються команди **fzero** і **fsolve**.

Приклад 1. Розглянемо коло з радіусом $r=2$, рівняння якого $x^2+y^2=4$. Треба знайти значення x, коли $y=1$, якщо заздалегідь відомо, що розв'язок знаходиться поблизу точки, абсциса якої $x=2$.

Зауважимо, якщо функція має складний вигляд, то має сенс створити окремий **m**-файл (файл-функцію):

```
function a=fun2(x,y)
a=x^2+y^2−4;
```

Для розв'язування задачі використаємо команду **fzero**. В цьому випадку формат виклику команди **fzero** буде наступним:

```
x0=fzero(@fun2,2,optimset,1)
x0 =
1.7321,
```

де **@fun2** – виклик файл-функції під назвою **fun2** з поточного (активного) каталога **MATLAB**; 2 – початкове значення аргументу x; 1 – значення аргументу y, який входить у процедуру **fzero** як додатковий параметр.

Іншими словами, даний варіант виклику команди **fzero** означає наступне: при заданому значенні x знайти значення y таке, що буде виконуватись умова: $x^2+y^2=4$.

Приклад 2. Треба знайти найменшу відстань від точки $C(3,1)$ до параболи $y=x^2-x$ у випадку, коли $x\in[1, 2]$.

Задамо координати шуканої точки та опишемо криву (параболу):

```
>> x=1:.001:2;
>> y=x.^2–x;
>> c=[3 1];
```

Тепер обчислюємо мінімальну відстань від точки до кривої:

```
>> s=min(sqrt((c(1) – x).^2+(c(2) – y).^2))
s =
   1.2780
```

Вправа 2. Визначення точок перетину за допомогою команди **fsolve**.

Виконання вправи зводиться до розв'язку системи рівнянь, які описують вихідні криві. В загальному випадку ці рівняння є нелінійними.

Нагадаємо, що для розв'язання систем нелінійних рівнянь в системі **MATLAB** використовується команда **fsolve**.

Приклад 1. Знайти точки перетину кола і прямої: $x^2+y^2=4, 3x+y-1=0$.

Необхідно скласти **m**-функцію для цієї системи рівнянь:

```
function f=myfyn2(x)
f(1)=x(1)^2+x(2)^2–4^2;
f(2)=3*x(1)+x(2)–1;
```

Викликаємо процедуру команди **fsolve** з наступними параметрами для пошуку кординати першої точки перетину:

```
x=fsolve(@myf1,[–1 4],optimset('Display','off'))
x =
–0.9610 3.8829
```

та визначаємо координати другої точки перетину:

```
x=fsolve(@myf1,[1 –4],optimset('Display','off'))
x =
1.5610 –3.6829
```

Недоліки команди **fsolve** проявляються в тому, що неможливо отримати всі розв’язки системи рівнянь відразу, а бажано попередньо знайти наближене місцезнаходження точок перетину. Іншими словами, варіант розв’язання залежить від координат початкової точки для ітераційного процесу.

Задача пошуку точок перетину можна розв’язувати за допомогою команд побудови графіків функцій, що описують вихідні криві.

Приклад 2. Знайти точки перетину кола $x^2 + y^2 = 6$ та еліпсу $\frac{(x-2)^2}{16} + \frac{(y-2)^2}{25} = 1$.

Складаємо **m**-файл:

```
function f=toch2(x)
f(1)=((x(1)–2)^2)/16+((x(2)–2)^2)/25–1;
f(2)=x(1)^2+x(2)^2–6;
```

Побудуємо графікі вихідних функцій за допомогою команди **ezplot** з підсистеми **Simbolic Math**:

```
>> ezplot('x^2+y^2=6');
axis equal;grid on;hold on;
>> ezplot('((x–2)^2)/16+((y–2)^2)/25=1');
```

Визначаємо точки перетину:

```
>> x=fsolve(@toch2,[–2 2],optimset('Display','off'))
x =
  –1.9756    1.4482
>> x=fsolve(@toch2,[0 –1],optimset('Display','off'))
x =
  0.1665   –2.4438
```

Приклад 3. Знайти точки перетину двох кривих: $y = e^x - 2$ та $x^2 + y^2 = 16$.

Побудуємо графікі вихідних функцій. Графік кола відтворюється на основі використання рівняння кола у полярній системі координат:

```
>> t=0:2*pi/100:2*pi;
>> x=4.*cos(t);y=4.*sin(t);
>> plot(x,y); grid on
>> hold on
>> x1=–4:.01:4; y1=exp(x1)–2;
>> plot(x1,y1); axis([–5 5 –5 5])
```

Один з можливих підходів побудови функцій, які визначені у неявному вигляді, полягає в наступній процедурі:

```
>> h=.02; x=–4:h:4; [X,Y]=meshgrid(x); f=X.^2+Y.^2–16;
>> v=[0,0]; contour(x,x,f,v);
>> axis equal
>> hold on
>> Y1=exp(x)–2;
>> plot(x,Y1);
```

Рекомендуємо самостійно знайти точки перетину вихідних кривих згідно процедур вправи 1.

Вправа 3 Побудова перпендикуляру до плоскої лінії за допомогою команди **fminbnd**.

Задача вирішується за допомогою команди пошуку мінімального значення функції (**fmin** або **fminbnd**). Формат виклику процедури даних команд: **fminbnd(@Func1,x0,x1)**, де **Func1** – задана як файл-функція, **[x0;x1]** – діапазон пошуку локального мінімуму.

Розглянемо задачу: встановити перпендикуляр із заданої точки на замкнуту лінію. Цю задачу можна вирішити, склавши рівняння перпендикулярності відрізка, що належить перпендикуляру, і дотичної до кривої. Інший підхід до розв'язання цієї задачи може бути зв'язаний з знаходженням мінімальної величини відрізка, що сполучає задану точку і точку на лінії, але в цьому випадку треба переконатись, що відрізок є шуканим перпендикуляром.

Нехай нам задана точка $C(0,\ 5)$ і еліпс $y = \sin t,\ x = 3\cos t$, вісі якого повернені на кут $\pi / 6$ проти годинникової стрілки відносно координатних вісей. Центр еліпса перенесений в точку (2, 3).

Створимо **m**-файл **myfy1**.**m**, яка відтворює заданий еліпс:

```
function [a,b]=myfy1(t)
alfa=pi/6;
x=3*cos(t); y=sin(t);
```

```
a=cos(alfa)*x–sin(alfa)*y+2;
b=sin(alfa)*x+cos(alfa)*y+3;
```

Тепер можна побудувати графік еліпса:

```
t=0:2*pi/100:2*pi;[x,y]=myfy1(t);plot(x,y)
```

Підготуємо **m**-файл, який розглядатимемо як вхідний параметр команди **fminbnd**:

```
function s=myfy2(t)
[x,y]=myfy1(t);
c=[0 5];
s=sqrt((c(1) –x)^2+(c(2) –y)^2);
```

Застосуємо команду **fminbnd** і визначимо значення **tP**, яке відповідає умові існування перпендикуляра:

```
tP=fminbnd(@myfy2,0,2*pi)
tP =
1.7758
```

Повний програмний алгоритм побудови і візуалізації перпендикуляра наведений нижче:

```
>> tP=fminbnd(@myfy2,0,2*pi)
tP =
   1.7758
>> [x,y]=myfy1(tP)
x =
   0.9817
y =
   3.5426
>> line([0 x],[5 y]); hold on; t=0:2*pi/100:2*pi; [x,y]=myfy1(t);
>> plot(x,y); axis equal;grid
```

Зауважимо, що для розв'язування задачи побудови перпендикуляра був застосований неповний формат команди:

fminbnd(@FunName,Tbeg,Tfin,options,P1,P2,…).

Не слід забувати, що для виклику довідки про вхідні параметри команди **fminbnd** можна скористатися опцією **help** (тобто^ **help optimset**, **help fminbnd, help fmin**).

Вправа 4. Обчислення площі плоскої фігури.

Площа криволінійної трапеції, що обмежена лініями $x = a$, $x = b(a < b)$, $y = f(x)$, де функція $f(x) > 0$, неперервна на інтервалі (a,b), обчислюється за допомогою поняття визначеного інтегралу [35]:

$$S = \int_a^b f(x)dx\,.$$

Відрізок (a,b) варто розділити на частини, у кожній з якіх знак величини функції $f(x)$ не змінюється. При цьому будемо дотримуватись наступного правила: площі, які знаходяться над віссю Ох, отримують знак плюс, а площі, які розташовані під віссю Ох, отримають знак мінус.

Площа фігури, що обмежена лініями $x=a, x=b, y=f_1(x), y=f_2(x)$, де $f_2(x) \geq f_1(x)$ при $a \leq x \leq b$ обчислюється за формулою:

$$S = \int_a^b (f_2(x) - f_1(x))dx.$$

Якщо рівняння неперервної лінії задано в полярній системі координат: $\rho = \rho(\varphi),\ \varphi_1 < \varphi < \varphi_2$, то площа сектору визначається за формулою:

$$S = \frac{1}{2}\int_{\varphi_1}^{\varphi_2} \rho(\varphi)d\varphi.$$

Якщо рівняння ліній дано в параметричній формі $x=\varphi(t),\ y=\psi(t)$, то площа фігури, що обмежена цією лінією та $x=a, x=b, y=0$, може бути обчислена за формулою:

$$S = \int_{t_1}^{t_2} \psi(t)\varphi'(t)dt,$$

де величини границь інтегралу, t_1, t_2, визначаються з рівнянь

$$a = \varphi(t_1) \text{ та } b = \varphi(t_2).$$

Приклад 1. Визначити площу області, яка описується рівнянням $\frac{x^2}{2} + \frac{y^2}{4} = 1$ (рівняння еліпсу).

Побудуємо за допомогою команди **contour** графік вихідної функції, що описує заданий еліпс і потім визначимо площу, яка обмежена еліпсом. Для перевірки знайденого значення площі порівняємо її з відомим точним значенням площі еліпсу $S = \pi ab$:

```
>> h=.02; x=–2:h:2; [X,Y]=meshgrid(x);
>> f=(X.^2)./2+(Y.^2)./4–1;
>> v=[0,0]; contour(x,x,f,v), grid
```

Визначимо координати точок перетину графіка еліпса з вісями координат та знайдемо шукану площу елипсу за допомогою команди інтегрування **quad(fun,a,b)**, де **fun** – параметр, що визначає підінтегральну функцію, **a,b** – границі інтегрування:

```
>> p=quad('sqrt(4–2.*(x.^2))',–1.4,1.4)
p =
   4.4375
```

Отримана половина площі еліпса, що знаходиться вище вісі Ox. Тоді вся шукана площа визначається наступною командою:

```
>> S=2*p
S =
   8.8750
```

За допомогою формули для обчислювання площі, що обмежена кривою елипса, тобто: $S = \pi ab$, зробимо перевірку отриманого результату:

```
>> Sf=pi*2*sqrt(2)
Sf =
   8.8858
```

Зверніть увагу, що у системі **MATLAB** для обчислення площі багатокутника із відомими координатам існує спеціальна команда **polyarea([x1 x2 x3 x4…],[y1 y2 y3 y4…])**, де **[x1 x2 x3 x4…]**,**[y1 y2 y3 y4…]** – координати вершин багатокутника. Наприклад, знайдемо площу багатокутника з координатами вершин $(1,-1), (2, 3), (3, 2), (4, 3), (5, 0)$:

```
>> disp(polyarea ([1 2 3 4 5],[–1 3 2 3 0]))
   9.5000
```

Приклад 2. Знайти площу фігури, що обмежена лініями: $y = -\frac{3}{2}x$ та $y = x^2 - x$, якщо відомо, що точки перетину цих ліній: $(0, 0)$, $\left(-\frac{3}{2}, \frac{15}{4}\right)$.

З вихідних даних задачі визначаємо інтервал інтегрування: $\left(-\frac{3}{2}, 0\right)$.

Для знаходження площі фігури використовуємо команду **quad:**

```
>> quad('–15/6.*x– (x.^2–x)', –3/2,0)
ans =
   0.5625
```

Практичні завдання лабораторної роботи № 9

Виконати наступні завдання :

Завдання 1. За допомогою команди **fzero** розв’язати задачу: знайти координати вершин трикутника, сторони якого описані рівняннями (див. табл. 9.2).

Таблиця 9.2 – Варіанти до завдання № 1

№ п/п	Рівняння сторон трикутника
1	$(AB) 3x - y - 4 = 0,$ $(BC) 3x + 5y - 34 = 0,$ $(AC) 3x + 2y - 1 = 0$
2	$(AB) 2x + 4y + 1 = 0,$ $(AC) x - y + 2 = 0,$ $(BC) 3x + 4y - 12 = 0$
3	$(AB) x + y - 5 = 0,$ $(BC) 2x - y + 4 = 0,$ $(AC) 5x - 8y + 14 = 0$
4	$(AB) 2x + y - 5 = 0,$ $(BC) 2x - y + 4 = 0,$ $(AC) 5x - 8y + 14 = 0$

Закінчення таблиці 9.2

5	$(AB)3x-5y-1=0,\quad (BC)5x+y+12=0,\quad (AC)2x+5y-12=0$
6	$(AB)5x-y-6=0,\quad (BC)x-7y-8=0,\quad (AC)6x-8y+20=0$
7	$(AB)11x+2y-25=0,\ (BC)3x+5y+11=0,\quad (AC)8x-3y-29=0$
8	$(AB)2x-4y-6=0,\quad (BC)4x+4y=0,\quad (AC)2x+8y-18=0$
9	$(AB)x-7y-18=0,\quad (BC)10x-y+27=0,\ (AC)9x+6y-24=0$
10	$(AB)11x+2y-25=0,\ (BC)3x+5y+11=0,\quad (AC)8x-3y-29=0$
11	$(AB)2x-4y-6=0,\quad (BC)4x+4y=0,\quad (AC)2x+8y-18=0$
12	$(AB)x-7y-18=0,\quad (BC)10x-y+27=0,\quad (AC)9x+6y-24$
13	$(AB)3x-2y-8=0,\quad (BC)6x+4y-8=0,\quad (AC)3x+6y-24=0$
14	$(AB)7x-3y-13=0,\ (BC)x+4y+7=0,\quad (AC)6x-7y+11=0$

Завдання 2. За допомогою команди **fsolve** вирішити задачу: знайти точки перетину заданих плоских кривих (див. табл. 9.3).

Таблиця 9.3 – Варіанти до завдання № 2

№ п/п	Рівняння кривих
1	$(x-1)^2+(y-1)^2=4, y=2x$
2	$x^2+(y-5)^2=25, y=x^2$
3	$\dfrac{x^2}{36}+\dfrac{y^2}{16}=1,(x+4)^2+y^2=9$
4	$x^2/36+y^2/4=1, y=6x-3x^3$
5	$x^2/9+y^2/4=1,\ \ 4x-12y=0$
6	$y=\dfrac{1}{x}, x^2+y^2=9$
7	$(x+1)^2/25+(y-2)^2/9=1,\ y=x^2/6$
8	$y=\dfrac{1}{x},\ \dfrac{x^2}{9}-\dfrac{y^2}{4}=1$
9	$(x+1)^2+y^2=16, y=\dfrac{1}{x-1}$
10	$\dfrac{(x+2)^2}{25}+\dfrac{(y-3)^2}{36}=1, y=\dfrac{x^2}{x-1}$
11	$x^2/25+(y+2)^2/16=1, y=3-x^2$
12	$y=x^2+3,5y+2x-3=0$
13	$x/2+y/4=1, y=3x^2-6x$
14	$\dfrac{x^2}{4}+\dfrac{(y-1)^2}{25}=1, \dfrac{y}{2}-\dfrac{x}{4}=1$

Завдання 3. Обчислити площу плоскої фігури, що обмежена заданими лініями (див. табл. 9.4)

Таблиця 9.4 – Варіанти до завдання № 3

№№	Рівняння ліній
1	$y = x^2/2,\ x = 1,\ x = 3,\ y = 0$
2	$y - \sin x,\ [0\pi],\ y = 0$
3	$y = 4x - x^2,\ y = 0$
4	$x = a\cos t,\ y = b\sin t,\ a = 3,\ b = 4$
5	$4y = 8x - x^2,\ 4y = x + 6$
6	$y = 3x^2 - 6x,\ x = 4, [0\ 4],\ y = 0$
7	$y = x^3 - 6x^2 + 11x - 6,\ y = 0$
8	$y = -x^2,\ x + y + 2 = 0$
9	$y = x^3,\ y = 8,\ x = 0$
10	$y = 2x - x^2,\ y = -x$
11	$xy = 3,\ x + y = 4$
12	$y = x^2,\ y = (x-1)^2$
13	$y = (x+2)^2,\ y = 4 - x$
14	$y = \dfrac{8}{4 + x^2},\ y = \dfrac{x^2}{4}$
15	$y = 1/(1 + x^2),\ y = x^2/2$

Методичні вказівки:

- в деких випадках при обчисленні площі треба знайти точки перетину з віссю Ox, тобто вирішити рівняння $f(x) = 0$ за допомогою вже відомих (раніше введених) команд.

Контрольні питання

1. Як знайти точки перетину двох плоских кривих за допомогою команд системи **MATLAB**?
2. Як можна описати задану криву на площині? Який спосіб найзручніший для завдання інформації про криву в системі **MATLAB**?
3. Перекажить процедуру пощуку найменьшої відстані від точки до кривої у системи **MATLAB.**
4. За допомогою яких команд у системі **MATLAB** вирішується задача пошуку перпендикуляра до лінії?
5. Як обчислити площу фігури, що обмежена лініями**?**
6. Розкрийте зміст команди **polyarea([x1 x2 x3 x4…],[y1 y2 y3 y4…])**.

Лабораторна робота № 10.

Тема. Розв’язання задач тривимірної геометрії у системі **MATLAB**.

Мета роботи: навчитись розв’язувати вправи зі стереометрії за допомогою команд системи **MATLAB**.

Теоретичний мінімум

Довільна лінія як на площині, так і в просторі, може бути задана множиною точок, координати яких в визначеній системі координат задовільняють рівнянню $F(x,y,z)=0$ [30, с. 159]. Лінію у просторі часто розглядають як лінію перетину двох поверхонь, кожна з яких задана рівнянням:

$$\begin{cases} F_1(x,y,z)=0 \\ F_2(x,y,z)=0 \end{cases}.$$

Зокрема, просторова пряма лінія може бути заданою:

$$\begin{cases} A_1x+B_1y+C_1z+D_1=0 \\ A_2x+B_2y+C_2z+D_2=0 \end{cases}$$

Нагадаємо, що рівняння, які визначають пряму в 3D-просторі:

$$\begin{cases} x=x_0+mt \\ y=y_0+nt \\ z=z_0+pt \end{cases}$$

називаються *параметричними рівняннями* просторової прямої. Використовується також і інша форма представлення прямої в просторі у вигляді наступного канонічного рівняння:

$$\frac{x-x_0}{m}=\frac{y-y_0}{n}=\frac{z-z_0}{p}.$$

Направляючими косинусами такої прямої є величини, що обчислюються за формулами:

$$\cos\alpha=\frac{m}{\sqrt{m^2+n^2+p^2}},\cos\beta=\frac{n}{\sqrt{m^2+n^2+p^2}},\cos\gamma=\frac{p}{\sqrt{m^2+n^2+p^2}},$$

де m, n, p – кутові коефіцієнти прямої, для яких має місце співвідношення: $m:n:p=\cos\alpha:\cos\beta:\cos\gamma$.

Гострий кут між прямою та площиною у просторі (даний кут формується між направляючим вектором прямої та нормальним вектором до плоскості) визначається за формулою:

$$\sin\varphi = \left| \frac{Am + Bn + Cp}{\sqrt{A^2 + B^2 + C^2}\sqrt{m^2 + n^2 + p^2}} \right|.$$

Вправа 1. Розв’язання задач побудови рівнянь і знаходження площин.

З попередніх лабораторних робіт відомо, що для побудови функцій в системі **MATLAB** існує визначена кількість команд. В даній роботі для представлення функцій, що описують площини, використовуються деякі спеціальні команди з підпакету **Simbolic Math** [20,с.453].

Приклад 1. Побудувати пряму, яка визначається двома площинами, рівняння яких задані:

$$\begin{cases} \dfrac{x}{2} + \dfrac{y}{2} + \dfrac{z}{2} = 1 \\ \dfrac{x}{2} + \dfrac{y}{2} - \dfrac{z}{6} = 1 \end{cases}.$$

Команда **ezsurfc** відтворює поверхні з нанесенням ліній рівня на координатну площину. Отримаємо з вихідних рівнянь залежність $z = z(x, y)$. З першого рівняння отримуємо: $z = 2 - x - y$, з другого рівняння; $z = \dfrac{x}{3} + \dfrac{y}{3} - \dfrac{1}{6}$.

Представлення шуканої прямої за допомогою команд системи **MATLAB** здійснюється у наступній формі:

```
>> ezsurfc('2–x–y')
>> hold on
>> ezsurfc('x/3+y/3–1/6')
```

Зауважимо, що для більш зручного зображення прямої у графічному вікні (тобто при необхідності здійснення розвороту площин, які задають вихідну пряму) треба скористатися кнопкою **Rotate 3D.** Команда, що реалізує цю операцію розташована у графічному вікні на панелі інструментів.

Приклад 2. Знайти рівняння площини, яка паралельна вісі Oz і проходить через задані точки $P_1(2, 3, -1)$ та $P_2(-1, 2, 4)$.

Рівняння площини, що паралельна вісі Oz має вигляд:

$$Ax + By + D = 0.$$

Підставляючи координати точок P_1 і P_2 в рівняння, отримаємо наступну систему рівнянь:

$$\begin{cases} 2A + 3B + D = 0 \\ -A + 2B + D = 0 \end{cases}.$$

Розв’язок такої системи однорідних лінійних рівнянь має вигляд:

$A = \begin{vmatrix} b_1 & d_1 \\ b_2 & d_2 \end{vmatrix} t,\ B = \begin{vmatrix} d_1 & a_1 \\ d_2 & a_2 \end{vmatrix} t,\ D = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} t$, t – параметр; важливо, принаймні, хоча б один з визначників розв'язку не дорівнює нулю. Позначимо визначники $M1 = \begin{vmatrix} b_1 & d_1 \\ b_2 & d_2 \end{vmatrix},\ M2 = \begin{vmatrix} d_1 & a_1 \\ d_2 & a_2 \end{vmatrix},\ M3 = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}$.

На основі вище виписаного розв'язку реалізуємо алгоритм вирішення вихідної вправи.

Складаємо головну матрицю коефіцієнтів системи M :

```
>> M=[2 3 1;–1 2 1];
```

Знаходимо з матриці M системи визначники M1, M3, вибираючи необхідні рядки й стовпці:

```
>> M1=M(1:2,2:3);
>> A=det(D1)
A =
    1
M3=M(1:2,1:2)
M3 =
    2    3
   –1    2
>> D=det(D3)
D =
    7
```

Виводимо з матриці M другий стовпець та змінюємо місцями стовпці, що залишаються:

```
>> M(:,2)=[]
M =
    2    1
   –1    1
>> M2=fliplr(M)
M2 =
    1    2
    1   –1
>> B=det(M2)
B =
   –3
```

Знайдені коефіцієнті $A = 1, B = -3, D = 7$ підставляємо у рівняння площини: $x - 3y + 7 = 0$. Рекомендуємо зробити перевірку результата за допомогою підстановки координат відомих точок A і B в отримане рівняння площини.

Вправа 2. Задачі на пряму лінію у просторі.

Приклад 1. Побудувати сукупність паралельних прямих, які належать однієї площині.

Цю задачу можна розв'язати за допомогою команди **ribbon([x1,x2],[y1,y2]),** яка дозволяє представити на графіку площину у вигляді сукупності прямих. Наприклад**:**

```
>> ribbon([4 –1],[2 –2]);grid on;hold on
```

Команда для побудови прямої лінії має структуру **line([x1,x2],[y1,y2],[z1,z2]), яка** дозволяє задати просторову пряму у графічному вікні:

```
>> line([0 1],[0 3],[–5 6]);grid on.
```

Багатокутник або багатогранник у просторі, який задається точками (координатами вершин), може бути визначен структурою:

fill3([x1,x2,x3, …],[y1,y2,y3, …],[z1,z2,z3, …]).

Приклад 2. Побудувати піраміду, яка визначається чотирма точками: $A(0,0,0)$, $B(0,4,0)$, $D(2, 2, 4)$. $C(4,0,0)$, $E(4, 4, 0)$.

Сукупність команд для побудови піраміди має вигляд**:**

```
>> fill3([4 0 0 4],[0 0 4 4],[0 0 0 0],'c'); grid on;hold on
>> fill3([4 0 2],[0 0 2],[0 0 4],'y'); fill3([0 0 2],[0 4 2],[0 0 4],'m');
>> fill3([4 4 2],[0 4 2],[0 0 4],'g'); fill3([4 0 2],[4 4 2],[0 0 4],'r');
```

Приклад 3. Знайти точки перетину прямої $y = x$ та площини

$$\frac{x}{3} + \frac{y}{3} + \frac{z}{3} = 0$$

Результати розв'язання отримуємо при застосуванні наступної сукупності команд:

```
>> ezsurfc('3–x–y'); hold on
>> ezplot('x–y'); [x y]=solve('3–x–y','x–y')
x =
3/2
y =
3/2
```

Вправа 3. Візуалізація перетинів поверхонь та знаходження ліній перетину поверхонь.

Візуалізувати перетин поверхонь можна за допомогою команд з пакету **Symbolic Math** та деяких спеціальних команд тривимірної графіки, наприклад, **fill3(x,y,z,c)**.

Приклад 1. Візуалізувати перетин поверхні, що описується рівнянням $z = x^2 + y^2 - 1$, з площиною, яка задається трьома точками: $A(-6,0,0), B(6,0,80), C(7,0,30)$.

Ця задача вирішується наступною послідовністю команд:

```
>>ezsurfc('Y^2+X^2–1');hold on
>>X=[–6 6 7];Y=[0 0 0];Z=[0 80 30];
>>fill3(X,Y,Z,'g')
```

Приклад 2. Знайти лінію перетину двох параболоїдів: $z = y^2 + x^2 - 25$ та $z = 25 - x^2 - y^2$.

Для розв’язування використовуємо наступну послідовність команд:

```
>> ezsurfc('Y^2+X^2–25');hold on
>> ezsurfc('–Y^2–X^2+25');hold on
>> [x,y]=solve('Y^2+X^2=25','–Y^2–X^2=-25')
x =
[  (25–Y^2)^(1/2)]
[ –(25–Y^2)^(1/2)]
```

Лінія перетину має форму кола з радіусом $r = 5$.

Приклад 3. Візуалізувати перетин двох поверхонь: $z = y^2 + x^2 - 64$ та $z = y^2 - x^2 - 49$ і знайти точки проекцій перетину визначених поверхонь на площині xOy.

Для розв’язку задачи використовуємо наступні команди:

```
>> ezsurfc('Y^2+X^2–64');hold on
>> ezsurfc('Y^2–X^2–49');hold on
>> [x y]=solve('Y^2+X^2=64','Y^2–X^2=49')
 x =
[  1/2*30^(1/2)]
[ –1/2*30^(1/2)]
[  1/2*30^(1/2)]
[ –1/2*30^(1/2)]
y =
[  1/2*226^(1/2)]
[  1/2*226^(1/2)]
[ –1/2*226^(1/2)]
[ –1/2*226^(1/2)]
```

Вправа 4. Пошук линії перетину сфери $x^2 + y^2 + z^2 = 25$ та параболоїда обертання $z = x^2 + y^2$.

Для розв’язання використовуємо команду **fsolve.** Звернемо увагу, що при розв’язанні задач такого типу, в більшості випадків, трудомісткість і успіх процесу розв’язання залежать від вдалого вибору процедури параметризації. Виконаємо параметризацію цих поверхонь.

Рівняння сфери $x^2 + y^2 + z^2 = 25$ може бути приведено до наступної форми:

$$\begin{cases} x(u,v) = 5\sin(u)\cos(v) \\ y(u,v) = 5\sin(u)\sin(v) \\ z(u,v) = 5\cos(u) \end{cases}.$$

Рівняння параболоїду $z = x^2 + y^2$ може бути приведено до наступної форми:

$$\begin{cases} x(r,t) = r\cos(t) \\ y(r,t) = r\sin(t) \\ z(r,t) = r^2 \end{cases}.$$

Для формування команди **fsolve** необхідно скласти **m-**файл, що описує вихідні поверхні в параметричній формі представлення:

```
function f=myfyn3(p,t)
f(1)=5*sin(p(1))*cos(p(2))–p(3)*cos(t);
f(2)=5*sin(p(1))*sin(p(2))–p(3)*sin(t);
f(3)=5*cos(p(1))–p(3)^2;
```

Тепер випишимо **m-**файл, який буде використаний в команді **fsolve.** Цей файл містить в собі розв’язок вихідної задачи і задає команду на побудову геометричних фигур з лінією перетину в одному графічному вікні:

```
function myfyn4(i, j)
T=0:2*pi/100:2*pi;
M=length(T);
for i=1:M
c=T(i);
Vector=fsolve(@myfyn3,[1,1,1],optimset('Display','off'),c);
U(i)=Vector(1);
V(i)=Vector(2);
R(i)=Vector(3);
end;
for i=1:M
X(i)=5*sin(U(i))*cos(V(i));
Y(i)=5*sin(U(i))*sin(V(i));
Z(i)=5*cos(U(i));
end;
u=0:pi/20:pi; m=length(u);
v=0:2*pi/20:2*pi; n=length(v);
for i=1:m
for j=1:n
Sx(i,j)=5*sin(u(i))*cos(v(j));
Sy(i,j)=5*sin(u(i))*sin(v(j));
Sz(i,j)=5*cos(u(i));
end;
```

```
end;
r=0:5/20:sqrt(10); t=0:2*pi/20:2*pi; mP=length(r); nP=length(t);
for i=1:mP
for j=1:nP
Px(i,j)=r(i)*cos(t(j));
Py(i,j)=r(i)*sin(t(j));
Pz(i,j)=r(i)^2;
end;
end;
mesh(Sx,Sy,Sz);
hold on
mesh(Px,Py,Pz);
hold on
plot3(X,Y,Z,'bo-');
```

Після цього залишається викликати цей **m**-файл з командного рядка **myfyn4**.

Таким чином, команда **myfyn4** побудує сферу, параболоїд і лінію перетину даних поверхонь. Слід зазначити, що масштаб по координатних вісях різний, тому сфера виглядає як еліпсоїд. Всі параметри графіка регулюються або програмно, або в інтерактивному режимі в системі **MATLAB**. У відповідному графічному вікні можна задати діапазони значень, які управляють вибіром кольору, товщиною ліній та іншими параметрами. Рекомендуємо розібратися з особливостями такого управління

Вправа 5. Обчислення об’ємів геометричних тіл.

Як відомо, об’єм циліндричного тіла, обмеженого з боків циліндричною поверхнею, утворююча якої паралельна вісі Oz, може бути знайдений за допомогою подвійного інтегралу

$$V = \iint_{(\sigma)} f(x,y)dxdy\,.$$

Напрямною цього циліндричного тіла є контур, який задає область інтегрування (σ), що знаходиться у площині xOy і є нижньою основою цілиндричного тіла. Зверху тіло обмежено поверхнею, яка описується рівнянням $z = f(x,y)$. Припускається, що функція $f(x,y)$ – неперервна та однозначна в області (σ) [31, с.375].

Таким чином, об’єм такого тіла знаходиться за формулою:

$$V = \iint_{(\sigma)} z\,dx\,dy\,.$$

У разі переходу до полярної системи координат дана формула отримує вигляд:

$$V = \iint\limits_{(\sigma)} f(r\cos\varphi, r\sin\varphi) r dr d\varphi .$$

У випадку, коли відома площа поперечного перерізу, об’єм тіла може бути знайдений за формулою $V = \int_c^d S(y)dy$.

Приклад 1. Знайдемо об’єм сфери, радіус якої дорівнює 2, а центр сфери знаходиться в точці $C(0, 2, 0)$.

Область інтегрування представляє собою круг, що розташован на площині xOy з центром в точці (0, 2) і радіусом $r = 2$. Площа поперечного перерізу у цьому випадку $S(y) = \pi(y-2)^2$, відповідні команди, що дозволяють визначити цю площину:

```
>> V=2*int('pi*(y–2)^2',0,2)
V =
16/3*pi
```

Приклад 2. Знайти об’єм тіла, який відсікає площина $y = 4$ від еліптичного параболоїду $y = \frac{x^2}{36} + \frac{z^2}{25}$.

В результаті перетинання параболоїд площиною, отримуємо фігуру, яка є еліпсом. Рівняння цього еліпсу визначається з рівняння параболоїду, якщо обидві частині рівняння розділити на y: $1 = \frac{x^2}{36y} + \frac{z^2}{25y}$. Полувісі еліпсу: $6\sqrt{y}, 5\sqrt{y}$. Площа поперечного перерізу: $S(y) = \pi \cdot 6\sqrt{y} \cdot 5\sqrt{y}$. Об’єм тіла знаходимо за допомогою наступних команд:

```
>> V=int(pi*30*y,0,4)
V =
180*pi
```

Приклад 3. Обчислимо об’єм тіла, обмеженого поверхнями: $z = 4x^2 + 2y^2 + 1;\ x + y - 3 = 0;\ x = 0;\ y = 0;\ z = 0$.

Перша поверхня – це еліптичний параболоїд, у якого вісь сіметрії є координатна вісь Oz.

Друга поверхня – площина, яка паралельна вісі Oz, інші поверхні – координатні площини yOz, xOz, xOy.

На площині xOy тіло вирізає трикутник, який обмежений координатними вісями Ox, Oy та прямою $x + y - 3 = 0$.

Об’єм тіла обчислюється за формулою:

$$V = \iint\limits_{(\sigma)} (4x^2 + 2y^2 + 1) dx dy .$$

Перетворюємо подвійний інтеграл у повторний. Внутрішнє інтегрування будемо вести по змінній x, зовнішнє – по змінній y. При постійному y

змінна x змінюється від 0 до $x = 3 - y$, а змінна y змінюється в діапазоні від 0 до 3.

Визначимо символьні змінні та виконаємо інтегрування за допомогою команди підсистеми **Simbolic Math**:

```
>> syms x y
```

Внутрішній інтеграл:

```
>> int('4*x^2+2*y^2+1',0,3–y)
ans =
4/3*(3–y)^3+2*y^2*(3–y)+3-y
```

Зовнішній інтеграл:

```
>> int('4/3*(3–y)^3+2*y^2*(3-y)+3–y',0,3)
ans =
45
```

Приклад 4. Знайти об'єм поверхні, що описується рівнянням $f(x,y) = x^3 y - 2xy^2 + y - 0.2 = 0$ при $x, y = [0.1]$.

Побудуємо графік неявно заданої функції $f(x,y) = 0$ за допомогою команди **contour (**для функціонування цієї команди треба задати крок дискретизації h та діапазон зміни аргументу **x**):

```
>> h=.02; x=0:h:1;
>> [X,Y]=meshgrid(x);
>> f=X.^3.*Y–2*X.*Y.^2+Y–.2;
>> v=[0,0];
>> contour(x,x,f,v), grid
```

В графічному вікні зелена лінія представляє графік перетину поверхні з $f(x,y) = 0$ з координатною площиною xOy. Область в першому квадранті між цими кривими позначимо через G. З'ясуємо, який знак має функція $f(x,y)$ в області G, для чого використовуємо команду **mesh 3D**-графіків:

```
>> mesh(x,x,f.*(f>0))
```

Обчислимо площу S області G:

```
>> S=h^2*sum(f(:) >=0)
    (S=0.7296).
```

Для порівняння результатів обчислимо площу при різних h. Тобто при h=0.01 площа буде дорівнювати S=0.7204, а при h=0.005 площа S=0.7152.

З'ясуємо, який об'єм утворюється між поверхнею $f(x,y)$ і областю G, де $f(x, y) \geq 0$. Для цього задамо крок h=0.02 і обчислимо об'єм:

```
>> V=h^2*sum(f(f>=0))
     (V=0.1268)
```

У випадку, коли $h = 0.01$ отримуємо об'єм V=0.1235, а коли $h = 0.005$ отримуємо об'єм V=0.1219.

Практичні завдання лабораторної роботи № 10

Виконати наступні завдання :

Завдання 1. Знайти рівняння площини, яка буде паралельна вказаній вісі і проходитиме через вказані точки (див. табл. 10.1):

Таблиця 10.1 – Варіанти зо завдання №1

№ п/п	Вісь	Точки
1	Ox	$A(2,-3,2)$, $B(7,1,0)$
2	Oy	$A(2,1,-2)$, $B(-7,-2,1)$
3	Oz	$A(1,2,-4)$, $B(3,1,-1)$
4	Ox	$A(1,-5,2)$, $B(6,1,3)$
5	Oy	$A(3,1,-4)$, $B(-6,-1,2)$
6	Oz	$A(0,7,-4)$, $B(5,-1,-1)$
7	Ox	$A(-4,-3,2)$, $B(3,1,7)$
8	Oy	$A(1,2,-3)$, $B(-7,-2,1)$
9	Oz	$A(-5,7,-4)$, $B(9,1,-6)$
10	Ox	$A(6,-6,7)$, $B(7,1,2)$
11	Oy	$A(4,1,-5)$, $B(7,-2,1)$
12	Oz	$A(5,3,-4)$, $B(2,4,-9)$
13	Ox	$A(4,-6,5)$, $B(8,2,0)$
14	Oy	$A(0,1,-2)$, $B(-7,2,-5)$

Завдання 2. Розв’язати наступні задачі (див. табл. 10.2):*)

Таблиця 10.2 – Варіанти до завдання №2

№ п/п	Зміст завдання
1	Знайти точки перетину еліпсу $\frac{x^2}{4}+\frac{y^2}{36}=1$ і площини $2x-y+3z+3=0$.
2	Знайти мінімальну відстань від точки (1,3,2) до площини $3x+y+z-6=0$.
3	Знайти точки перетину прямої $\frac{x-1}{3}=\frac{y+1}{-1}=\frac{z-2}{5}$ з площиною $x+y-2z-4=0$ (рівняння прямої привести до каноничного вигляду з використанням параметру t).
4	Знайти гострий кут між прямою $\begin{cases}x+y+z-4=0\\2x-y+4z+5=0\end{cases}$ та площиною $x+y+3z+5=0$ (вказівка: скласти матрицю кутових коефіцієнтів).

Закінчення таблиці 10.2

5	Знайти точку перетину сфери радіусу 5 з центром на початку координат і прямої, що проходить через точки A(0;0;3) та B(3;3;0). Отримати всі розв'язки.
6	Знайти точку перетину поверхонь: сфери радіусу 5 з центром в початку координат і площини $\frac{x}{3}+\frac{y}{3}+\frac{z}{3}=1$.
7	Побудувати лінію перетину поверхонь $\frac{x^2}{4}+\frac{y^2}{9}=1$ та $x^2+y^2=9$. Знайти точки перетину проекцій поверхонь на xOy
8	Визначити сліди прямої $\begin{cases}5x+3y-4z+8=0\\x-y+z+5=0\end{cases}$ на координатних площинах (слідом прямої на площині називається точка перетину прямої з площиною)
9	З точки A(-4;4;4) опустити перпендикуляр на поверхню, що описується рівнянням $x+\frac{y}{5}+z+5=0$.
10	Опустити перпендикуляр з точки A(10;5;8) на поверхню сфери радіусу 4 з центром в початку координат.
11	Знайти точку перетину поверхонь: сфери радіусу 10 з центром в точці (1, 1, 1) та площиною $\frac{x}{5}+\frac{y}{5}+\frac{z}{5}=1$.
12	Знайти точку перетину сфери радіусу 4 з центром на початку координат і прямої, що проходить через точки A(0;0;2) та B(2;2;0). Отримати всі розв'язки.
13	З точки A(0;0;-10) опустити перпендикуляр на поверхню, що описується рівнянням $2x+3y-5z+30=0$.
14	Знайти точки перетину відповідних ліній рівня на площині xOy двох поверхонь $x^2+\left(\frac{9}{4}\right)^2 y^2+z^2=9$ та $4-x^2-y^2=0$.

***)** Методичні вказівки:

- задачі №№ 5–8 вирішувати за допомогою команд **fsolve** або **solve**;
- задачі №№ 9–13 вирішувати за допомогою команди **fzero**.

Завдання 3. Знайти об'єм тіла, що обмежено заданими поверхнями (див. табл. 10.3):

Таблиця 10.3 – Варіанти до завдання №3

№ п/п	Рівняння поверхонь	№ п/п	Рівняння поверхонь
1	$z=16-x^2,\ x+y=4,$ $y=2x,\ z=0,\ y=0$	8	$10(x^2+y^2)+9z=90,\ z=0$

Закінчення таблиці 10.3

2	$12(x^2 + y^2) + 4z = 48,\ z = 0$	9	$z = 36 - x^2,\ x^2 + y^2 = 36,$ $x = 1,\ y = 1,\ z = 1$
3	$z = 16 - x^2,\ x^2 + y^2 = 16,$ $x = 0,\ y = 0,\ z = 0$	10	$z = 25 - x^2,\ x + y = 5,$ $y = 2x,\ z = 0,\ y = 0$
4	$z = 9 - x^2,\ x + y = 3,$ $y = 2x,\ z = 0,\ y = 0$	11	$6(x^2 + y^2) + 4z = 24,\ z = 0$
5	$8(x^2 + y^2) + 4z = 32,\ z = 0$	12	$z = 36 - x^2,\ x + y = 6,$ $y = 2x,\ z = 0,\ y = 0$
6	$z = 49 - x^2,\ x + y = 7,$ $y = 2x,\ z = 0,\ y = 0$	13	$z = 25 - x^2,\ x^2 + y^2 = 25,$ $x = 0,\ y = 0,\ z = 0$
7	$z = 64 - x^2,\ x^2 + y^2 = 64,$ $x = 0,\ y = 0,\ z = 0$	14	$15(x^2 + y^2) + 9z = 135,\ z = 0$

Контрольні питання

1. За допомогою якої команди с підпакету **Simbolic Math** можна побудувати графічне зображення площини?
2. Як побудувати графічне зображення площини, якщо вона проходить через три задані точки?
3. За допомогою якої функції можна побудувати лінію у просторі, що проходить через три задані точки (через дві задані точки)? Поясніть синтаксис написання команд.
4. Яка команда виконує побудову сукупності паралельних прямих, що належать одній площині?
5. Який спосіб побудови графіку неявно заданої функції можна використовувати в системі **MATLAB**?
6. Визначити етапи візуалізації ліній перетину поверхонь.
7. Як знайти об’єм тіла за допомогою команд у системі **MATLAB**?