Лабораторна робота № 6.

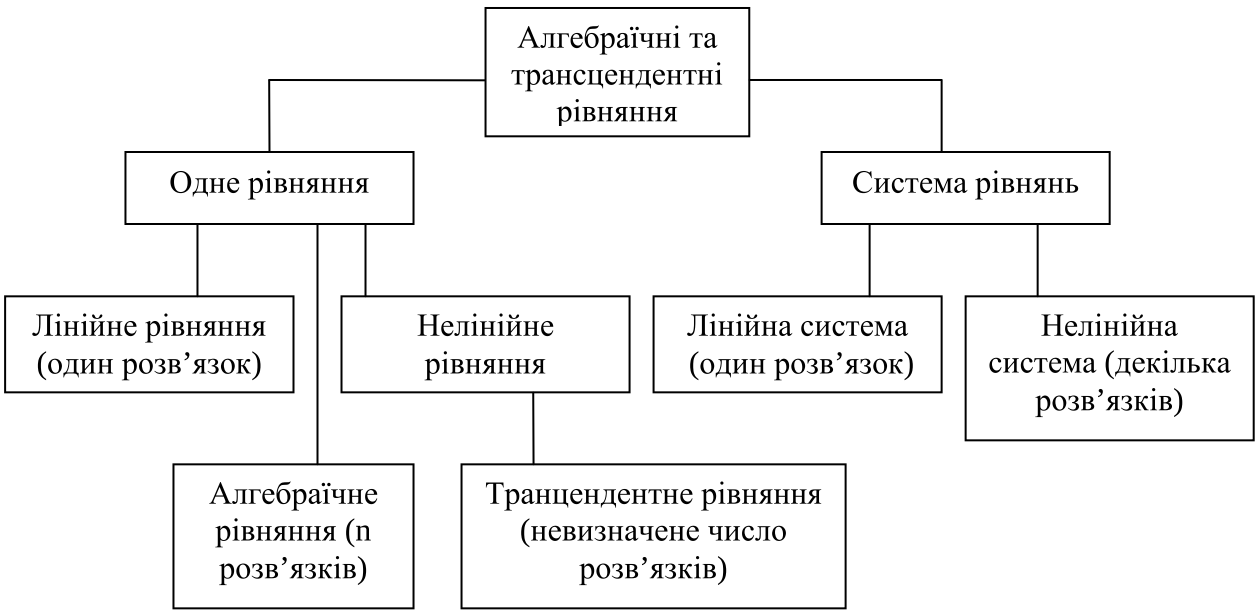
Тема: Розв'язання лінійних рівнянь та систем лінійних рівнянь у системі **MATLAB.**

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. рис. 6.1).



Мал. 6.1 – 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 $a_0x^n + a_1x^{n-1} + a_2x^{n-2} + ... + a_{n-1}x + a_n = 0$, 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 $x_1, x_2, ..., x_n$:

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

Дана система лінійних рівнянь може бути записана у матричному вигляді:

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

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

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 **A** – 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

$$X=A^{-1}B, \tag{4}$$

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

У системі **MATLAB** для розв'язання систем лінійних рявнянь існують декілька методів. Простішим методом розв'язання є метод вирішувачів систем лінійних рівнянь:

- оператор **X=B\A** знаходить розв'язок системи рівнянь вигляду **AX=B**, де **A** – прямокутна матриця розміром **m×n, B** – матриця розміром **m×1**;
- оператор **X=B/A** знаходить розв'язок системи рівнянь вигляду **XA=B**, де **A** – 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 A дорівнює 3×4, а розмір вектору B 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: 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

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

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 $i = k+1,\ ...,\ m$. 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:

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

розв’язання якої відбувається за рекурентними формулами:

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

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

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

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

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

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: $\mu_{43} = \frac{17}{8}$.

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

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

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 $x^{(0)} = \left(x_1^0,\ x_2^0,\ \ldots,\ x_m^0\right)$ 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 A – 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 τ>0.

Передбачимо, що $\lambda_{\min}$ і $\lambda_{\max}$ – відповідно мінімальне й максимальне власні значення матриці A. 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**≥**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**≥**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=Ви – 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 $\delta B = \frac{\|\Delta B\|}{\|B\|}$:

```
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 $\dfrac{\delta X}{\delta B}$ = 3. 9175e+004.