Лабораторна робота №3 (4 години)

Чисельне розв'язання систем лінійних алгебраїчних рівнянь

<u>*Мета роботи*</u>*:* отримання практичних навичок побудови алгоритмів розв'язання систем лінійних алгебраїчних рівнянь, їх програмної реалізації на комп'ютері, порівняння різних методів.

Короткі теоретичні відомості

Система лінійних алгебраїчних рівнянь (СЛАР) в найбільш загальному випадку має вигляд:

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

де a_{ij} — задані коефіцієнти системи, b_j — відомі вільні члени системи, x_j — невідомі, що підлягають визначенню.

Ввівши матрицю коефіцієнтів системи:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix},$$

вектор-стовпчик вільних членів $\mathbf{B} = \left[b_1, b_2, ..., b_n\right]^T$ и вектор-стовпчик невідомих $\mathbf{X} = \left[x_1, x_2, ..., x_n\right]^T$, систему (1) можна подати у матричній формі:

$$\mathbf{AX}=\mathbf{B}. \qquad (2)$$

Найбільш ефективні прямі методи рішення СЛАР використовують розкладення матриці **A** на множники (факторизацію).

В методі LU-факторизації матриця коефіцієнтів A подається у вигляді добутку матриць **L** і **U**.

$$\mathbf{A}=\mathbf{LU} \qquad (3)$$

де $\mathbf{L} = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix}$ — нижньотрикутна матриця,

$$\mathbf{U}=\begin{bmatrix} 1 & u_{12} & \cdots & u_{1n} \\ 0 & 1 & \cdots & u_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$ — верхньотрикутна матриця, всі діагональні елементи якої дорівнюють 1.

Вектор **B** під час факторизації не змінюється.

Метод LU-факторизації складається із двох етапів:

1) факторизація (розкладення) матриці **A**;
2) отримання розв’язку **X**.

Факторизація матриці **A** виконується за n стадій. На кожній j-тій стадії послідовно обчислюються елементи l_{ij} чергового j-го стовпчика матриці **L** за формулами:

$$l_{ij}=a_{ij}-\sum_{k=1}^{j-1} l_{ik}u_{kj},\ i=\overline{j,n} \qquad (4)$$

і елементи u_{ji} чергового j-го рядка матриці **U** за формулами:

$$u_{ji}=\frac{a_{ji}-\sum_{k=1}^{j-1} l_{jk}u_{ki}}{l_{jj}},\ i=\overline{j+1,n} \qquad (5)$$

Отримання розв’язку **X** системи (2) відбувається таким чином: спочатку розв’язується система з нижньотрикутною матрицею **L** вигляду

LY=B

за допомогою алгоритму прямого ходу:

$$y_i=\frac{b_i-\sum_{j=1}^{i-1} l_{ij}y_j}{l_{ii}},\ i=\overline{1,n} \qquad (6)$$

Потім розв’язується система **UX=Y** з верхньотрикутною матрицею **U** за формулою зворотного ходу

$$x_i=y_i-\sum_{j=i+1}^{n} u_{ij}x_j,\ i=\overline{n,1} \qquad (7)$$

Описана схема LU-факторизації може бути реалізована тільки у випадку, коли її елементи $l_{jj}\neq 0$. Крім того, близькість цих елементів до нуля може привести до великої втрати точності обчисленого на машині розв’язку. Щоб уникнути цього, розв’язок системи лінійних алгебраїчних рівнянь методом LU-факторизації слід виконувати з вибором найбільшого за модулем (ведучого) елемента l_{jj}. При цьому крім матриць **L** і **U** необхідно зберігати матрицю перестановок **P**. Матриця перестановок отримується із одиничної матриці **E** перестановкою рядків або стовпчиків. Факторизована матриця **A** в цьому випадку має вигляд:

A=PLU

Зазвичай **P** зберігається у вигляді вектора $\mathbf{P}=\begin{bmatrix} p_1, & p_2, & \ldots, & p_n \end{bmatrix}^T$, отриманого в результаті перестановки елементів $\begin{bmatrix} 1, & 2, & \cdots, & n \end{bmatrix}^T$, причому i-ий рядок в **P** співпадає з p_i рядком в **E**. Легко помітити, що система **PX=B** має розв’язок $x_{pi}=b_i$, $i=\overline{1,n}$.

При виборі ведучого елемента по стовпчику на кожному кроці k виконуються такі дії. Обчислюються елементи l_{ik} за (4) і розміщуються на місцях a_{ik}, $i=\overline{k,n}$. Якщо $\max\limits_{i\ge k}|l_{ik}|=|l_{mk}|$ і $m\ne k$, то рядки k і m міняються місцями, тобто у всіх елементів індекс рядка m замінюють індексом k і навпаки. При цьому запам'ятовуються номери переставлених рядків (якщо матриця перестановок **P** зберігається як вектор із поточними номерами рядків, то в ньому також переставляються елементи з індексами k і m). Обчислюються елементи u_{ki} за формулами (5) і розміщуються на місці елементів a_{ki}, $i=\overline{k+1,n}$.

Під час LU-факторизації вектор **B** не змінюються, тому отримання матриць **L** і **U** можна використовувати для розв'язання СЛАР із однаковою матрицею **A** і різними векторами **B**.

При програмній реалізації алгоритму LU-факторизації масиви для зберігання матриць **L, U** і вектора **Y** не створюються. Елементи матриць **L** і **U** розміщуються під час розкладу на місці елементів матриці **A**, причому елементи $u_{jj}=1, j=\overline{1,n}$ не запам'ятовуються. Таким чином, факторизована матриця **A** має вигляд

$$\mathbf{A}_{\mathrm{LU}}=\begin{bmatrix} l_{11} & u_{12} & \cdots & u_{1n} \\ l_{21} & l_{22} & \cdots & u_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix}.$$

Елементи вектора **Y** розміщують безпосередньо на місцях вектора **X**.

Метод факторизації Холеського призначений для розв'язання СЛАР (2) із симетричною матрицею **A**. Він ґрунтується на розкладенні матриці **A** в добуток

$$\mathbf{A}=\mathbf{LDL}^{\mathrm{T}} \tag{8}$$

де **L** — нижньотрикутна матриця, **D** — діагональна матриця.

Факторизація матриці **A** виконується за n стадій. На кожній j-й стадії елементи l_{jj} і d_{jj} обчислюються за формулами

$$l_{jj}=1, j=\overline{1,n}$$

$$d_{jj}=a_{jj}-\sum_{k=1}^{j-1} l_{jk}^2 d_{kk},\ j=\overline{1,n} \tag{9}$$

Потім обчислюються елементи l_{ij} чергового j-го стовпчика за формулами

$$l_{ij}=\frac{a_{ij}-\sum\limits_{k=1}^{j-1} l_{ik}d_{kk}l_{jk}}{d_{jj}},\ j=\overline{1,n},\ i=\overline{j+1,n}$$

Після факторизації матриці **A** спочатку розв'язується СЛАР

$$\mathbf{LY}=\mathbf{B}$$

за допомогою прямої підстановки:

$$y_i=b_i-\sum_{j=1}^{i-1} l_{ij}y_j,\quad i=\overline{1,n} \tag{10}$$

Потім розв'язується СЛАР

$$\mathbf{DZ}=\mathbf{Y}$$

як

$$z_i = \frac{y_i}{d_{ii}} \qquad (11)$$

Розв'язок **X** знаходимо із системи

$$\mathbf{L}^{\mathrm{T}}\mathbf{X}=\mathbf{Z}$$

$$x_i = z_i - \sum_{j=i+1}^{n} l_{ji} x_j ,\ i = \overline{n,1} \qquad (12)$$

При програмній реалізації алгоритму факторизації Холеського масиви для зберігання матриць **L, D** і векторів **Y, Z** не створюються. Елементи матриць **L** і **D** розміщуються під час факторизації на місцях елементів матриці **A**, причому елементи $l_{jj} = 1, j = \overline{1,n}$ не запам'ятовуються. Таким чином, факторизовану матрицю зберігають у вигляді

$$\mathbf{A}_{\mathrm{LDL^T}} = \begin{bmatrix} d_{11} & a_{12} & \dots & a_{1n} \\ l_{21} & d_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & \dots & d_{nn} \end{bmatrix}.$$

Елементи векторів **Y** і **Z** розміщують безпосередньо на місцях вектора **X**.

Якщо матриця **A** симетрична і додатно визначена, то її факторизують у вигляді:

$$\mathbf{A}=\mathbf{L}\mathbf{L}^{\mathrm{T}}$$

Для цього на кожній j-й стадії факторизації матриці **A** обчислюють елементи l_{ij} чергового j-го стовпчика за формулами:

$$l_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2}, \quad j = \overline{1,n}$$

$$l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}}{l_{jj}},\ j = \overline{1,n}, \quad i = \overline{j+1,n},$$

Як видно із наведених виразів, для факторизації матриці **A** необхідно добувати квадратний корінь. Тому цей метод ще називають методом квадратного кореня.

Після факторизації матриці **A** розв'язок задачі (2) зводиться до розв'язання двох СЛАР із трикутними матрицями

$$\mathbf{LY}=\mathbf{B},\ \mathbf{L}^{\mathrm{T}}\mathbf{X}=\mathbf{Y}$$

Розв'язок цих систем знаходять за рекурентним формулами

$$y_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij} y_j}{l_{ii}}, \quad i = \overline{1,n}$$

$$x_i = \frac{y_i - \sum_{j=i+1}^{n} l_{ji} x_j}{l_{ii}}, i = \overline{n,1}$$

Із обчислювальних формул методу випливає, що для додатно визначеної симетричної матриці **A** процес факторизації завжди виконується, і ніяких перестановок не потребує.

При програмній реалізації методу квадратного кореня масиви для зберігання матриці **L** і вектора **Y** не створюються. Під час розкладу, елементи матриці **L** розміщують на місцях елементів матриці **A**, і факторизовану матрицю зберігають у вигляді

$$\mathbf{A}_{\mathrm{LL^T}} = \begin{bmatrix} l_{11} & a_{12} & \dots & a_{1n} \\ l_{21} & l_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix}.$$

Елементи вектора **Y** розміщують безпосередньо на місцях вектора **X**.

В методі QR-факторизації матрицю **A** подають у вигляді добутку ортогональної матриці **Q** і трикутної матриці **R**

$$\mathbf{A}=\mathbf{QR} \qquad (13)$$

В якості матриці **Q** частіше за все використовують матрицю відбивання и матрицю обертання. В першому випадку метод QR-*факторизації* називають методом відбиття, в другому - методом обертання.

В методі відбиття матрицю **R** отримують множенням матриці **A** зліва на n-1 матрицю відбиття $\mathbf{Q}_j$:

$$\mathbf{R} = \mathbf{Q}_{n-1}\, \mathbf{Q}_{n-2} \dots \mathbf{Q}_1\, \mathbf{A} \qquad (14)$$

Кожна матриця $\mathbf{Q}_j$ обнуляє елементи j-го стовпчика матриці

$$\mathbf{A}_{j-1} = \mathbf{Q}_{j-1}\, \mathbf{Q}_{j-2} \dots \mathbf{Q}_1\, \mathbf{A},$$

які лежать нижче головної діагоналі, і залишає без змін перші j-1 стовпчики. Для цього матриця $\mathbf{Q}_j$ подається у вигляді

$$\mathbf{Q}_j = \mathbf{E} - k_j\, \mathbf{S}_j\, \mathbf{S}_j^T \qquad (15)$$

де

$$\mathbf{S}_j = [0,\ 0, \dots,\ sign(a_{jj}^{(j-1)})\left(\left|\beta_j\right| + \left|a_{jj}^{(j-1)}\right|\right),\ a_{j+1j}^{(j-1)}, \dots, a_{nj}^{(j-1)}]^T; \qquad (16)$$

$$k_j = \frac{1}{(\beta_j^2 - \beta_j a_{jj}^{(j-1)})}; \qquad (17)$$

$$\beta_j = \begin{cases} +\sqrt{\sum\limits_{i=j}^{n} \left(a_{ij}^{(j-1)}\right)^2}, & a_{jj}^{(j-1)} < 0 \\ -\sqrt{\sum\limits_{i=j}^{n} \left(a_{ij}^{(j-1)}\right)^2}, & a_{jj}^{(j-1)} > 0 \end{cases}, \qquad (18)$$

$a_{km}^{(l)}$ — елементи матриці $\mathbf{A}_l$, отримані на l-й стадії.

Матриця **Q** в (13) може бути знайдена як транспонована до матриці добутків елементарних матриць $\mathbf{Q}_j$. Однак на практиці матрицю **Q** явно не обчислюють, а отримують перетворенням системи (2) до вигляду

$$\mathbf{R\,X} = \mathbf{B}_{n-1} \qquad (19)$$

по схемі

$$\begin{aligned} &\mathbf{A}_1 = \mathbf{Q}_1\, \mathbf{A}, && \mathbf{B}_1 = \mathbf{Q}_1\, \mathbf{B} \\ &\mathbf{A}_2 = \mathbf{Q}_2\, \mathbf{A}_1, && \mathbf{B}_2 = \mathbf{Q}_2\, \mathbf{B}_1 \qquad (20) \\ &\quad \cdots\cdots \\ &\mathbf{R} = \mathbf{Q}_{n-1}\, \mathbf{A}_{n-2}, && \mathbf{B}_{n-1} = \mathbf{Q}_{n-1}\, \mathbf{B}_{n-2} \end{aligned}$$

Система (19) розв'язується оберненою підстановкою за формулами

$$x_i = \frac{b_i^{(n-1)} - \sum_{j=i+1}^{n} r_{ij} x_j}{r_{ii}}, \quad i = \overline{n,1} \tag{21}$$

Слід відмітити, що на практиці не слід будувати матрицю $\mathbf{Q}_j$, $j = \overline{1, n-1}$ в явному вигляді, а вигідно залишити її в формі (15). Тоді, в (20) будемо мати

$$\mathbf{A}_j = \mathbf{A}_{j-1} - k_j\, \mathbf{S}_j\, (\mathbf{S}_j^{\mathrm{T}}\, \mathbf{A}_{j-1}) \tag{22}$$

$$\mathbf{B}_j = \mathbf{B}_{j-1} - k_j\, \mathbf{S}_j\, (\mathbf{S}_j^{\mathrm{T}}\, \mathbf{B}_{j-1}) \tag{23}$$

При програмній реалізації алгоритму QR-факторизації, масив для зберігання вектора $\mathbf{S}_j$ не створюється, а в якості останніх $n\text{-}j$ елементів цього вектора використовують безпосередньо елементи j-го стовпчика матриці $\mathbf{A}_{j-1}$: $a_{ij}^{(j-1)}, i = \overline{j+1, n}$. Замість j-го елемента вектора $\mathbf{S}_j$ використовують значення $sign(a_{jj}^{(j-1)})\left(\left|\beta_j\right| + \left|a_{jj}^{(j-1)}\right|\right)$. Крім того, елементи j-го стовпчика матриці $\mathbf{A}_j$, які розташовані нижче головної діагоналі, на практиці, не обнуляють, що дозволяє, при необхідності, відновити матриці $\mathbf{Q}_j$. При цьому факторизована матриця має вигляд:

$$\mathbf{A}_{\mathrm{QR}} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ a_{21}^{(2)} & r_{22} & \cdots & r_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1}^{(1)} & a_{n2}^{(2)} & \cdots & r_{nn} \end{bmatrix}. \tag{24}$$

У випадку, коли необхідно розв'язувати СЛАР із однаковими матрицями **A** і різними векторами **B**, задачу розв'язують в два етапи. На першому етапі за формулами (22) вихідну матрицю **A** факторизують до форми (24). На другому етапі кожні вектора **B** перераховують за формулами (23). При цьому враховують, що на місцях нульових елементів j-го стовпчика матриці **R** зберігаються $n\text{-}j$ елементи вектора $\mathbf{S}_j$: ($a_{j+1j}^{(j-1)}, a_{j+2j}^{(j-1)}, \ldots, a_{nj}^{(j-1)}$) . Для відновлення j-го елемента вектора $\mathbf{S}_j$ використовують (18) із врахуванням, того, що

$$r_{jj} \equiv \beta_j \tag{25}$$

Значення k_j відновлюють за (17). Розв'язок СЛАР x_i для кожного вектора **B** знаходять за (21).

Деякі алгоритми при факторизації використовують перестановку стовпчиків так, що на j-му кроці стовпчик із найбільшим значенням суми квадратів його елементів, що лежать нижче (j-1)-го рядка переставляється на місце j-го стовпчика. При цьому крім факторизованої матриці **A** зберігають вектор перестановок **P**. При цьому при використанні (21) замість x_i необхідно підставити x_{pi}.

Перевага методу QR-факторизації полягає в тому, що оскільки

$$\|\, \mathbf{A} \,\| = \|\, \mathbf{QR} \,\| = \|\, \mathbf{R} \,\|,$$

при формуванні **R** він не приведе до суттєвого збільшення в цілому перетворюваної матриці **A**. Це робить його чисельно дуже стійким.

Робоче завдання

1. Для лінійної схеми із розрахунково-графічної роботи по курсу "Теорія електронних і електричних кіл" отримати СЛАР відносно вузлових напруг вузловим методом.
2. Побудувати алгоритми розв'язання отриманої СЛАР методами LU-факторизації, LL^T факторизації Холеського, QR-факторизації.
3. Скласти робочу програму для розв'язання СЛАР кожним із методів. Кожен метод повинен бути оформлений у вигляді окремої універсальної функції. Формальні параметри функції повинні включати: **A** - матрицю коефіцієнтів, **B** - вектор вільних членів, *n* – кількість рівнянь СЛАР, **X** - вектор розв'язків (параметр, що повертається).
4. Набрати і налаштувати програму на комп'ютері. При налаштуванні необхідно використовувати СЛАР, для якої попередньо аналітично отримані факторизованні форми матриці **A** і вектор розв'язків.
5. Для СЛАР, отриманої в пункті 1, кожним із методів отримати і записати факторизовану форму матриці **A**, вектор розв'язків **X**, і вектор нев'язки **R**=**AX**-**B**.
6. Порівняти різні методи за надійністю, потрібним машинним ресурсам (об'єм оперативної пам'яті, кількість арифметичних операцій, час виконання).

Зміст звіту

1. Назва роботи.
2. Мета.
3. Робоче завдання.
4. Математичне формулювання алгоритмів розв'язання СЛАР кожним із методів.
5. Вихідна СЛАР.
6. Робоча програма.
7. Результати розрахунків.
8. Висновки.