Київський національний університет імені Тараса Шевченка
Факультет комп’ютерних наук і кібернетики

Звіт
з лабораторної роботи №2
«Розв’язування систем лінійних алгебраїчних рівнянь»

Виконав:
Студент групи ІПС-32
Гончаренко Ілля Сергійович

Київ
2024

Мета:

Метою роботи є розв'язок системи лінійних алгебраїчних рівнянь (СЛАР) за допомогою таких методів:

- Метод Гаусса
- Метод прогонки
- Метод Якобі
- Крім того, необхідно обчислити визначник і обернену матрицю для заданої системи.

Теорія:

Метод Гаусса

Метод Гаусса полягає в послідовному виключенні змінних із системи лінійних рівнянь з метою приведення матриці коефіцієнтів до верхньотрикутної форми, після чого здійснюється обчислення змінних зворотним ходом.

Розглянемо систему лінійних алгебраїчних рівнянь:

$$Ax = b$$

де:

- A — матриця коефіцієнтів розміром n×n
- x — вектор невідомих,
- b — вектор вільних членів.

1. Прямий хід (елімінація)

Метою є перетворення системи таким чином, щоб зменшити матрицю **A** до верхньотрикутного вигляду. Це досягається за допомогою елементарних перетворень над рядками матриці **A** та відповідних змін у векторі **b**.

Для кожного кроку k (де k=1,2,...,n−1) виконуємо такі дії:

1. Обираємо головний елемент A_{kk} , який повинен бути найбільшим за модулем серед елементів стовпця k (щоб уникнути помилок у разі малих чисел).
2. Після цього виконуємо операції для рядків, щоб зробити всі елементи під A_{kk} рівними нулю. Для цього використовуємо формули:

$$A_{ij} \leftarrow A_{ij} - \frac{A_{ik}}{A_{kk}} A_{kj} \text{ для } i > k, j = k + 1,...,n$$

Вектор вільних членів також змінюється за правилом:

$$b_i \leftarrow b_i, \frac{A_{ik}}{A_{kk}} b_k \text{ для } i > k$$

Після цього матриця A перетворюється на верхньотрикутну:

$$A' = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ 0 & A'_{22} & \dots & A_{2n} \\ 0 & 0 & \dots & A'_{nn} \end{pmatrix}$$

2. Зворотний хід

Після того, як матриця стала верхньотрикутною, невідомі можна обчислювати з кінця системи за допомогою підстановки. Останнє рівняння дає значення x_n, потім обчислюємо $x_{n-1}, x_{n-2}, \ldots, x_1$ за формулою:

$$x = \frac{b_i - \sum_{j=i+1}^{n} A_{ij}x_j}{A_{ii}}, i = n, n-1,...,1$$

Умова застосування

Метод Гаусса застосовується лише в тому випадку, якщо визначник матриці **A** відмінний від нуля:

$$det(A) \neq 0$$

Це означає, що система має єдиний розв’язок.

Метод Якобі:

Метод Якобі - це ітераційний метод для розв’язування систем лінійних рівнянь. Він полягає в тому, що на кожній ітерації нові наближені значення змінних обчислюється на основі попередніх наближень.

1. Формуло ітерації

Нехай маємо систему:

$$Ax = b$$

де $A = [a_{ij}]$ - квадратна матриця. На кожному кроці k+1 - та ітерація змінною x_i визначається за формулою

$$x_i^{(K+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j\neq i} a_{ij}x_j^{(k)})$$

де $a_{ii} \neq 0$

2. Умова збіжності

Метод Якобі збігається, якщо матриця A є діагонально домінуючою, тобто для кожного і

$$|a_{ii}| > \sum_{j\neq i} |a_{ij}|$$

Якщо ця умова не виконується, метод може не збігатися або збігатися повільно.

Метод прогонки:

Цей метод спеціально розроблений для розв'язування систем рівнянь з трьохдіагональною матрицею. Він значно ефективніший для таких систем, оскільки дозволяє зменшити кількість операцій.

1. Форма системи

Трьохдіагональна система має вигляд:

$$\begin{pmatrix} b_1 & c_1 & 0 & \dots & 0 \\ a_2 & b_2 & c_2 & \dots & 0 \\ 0 & a_3 & b_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & a_n & b_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \dots \\ d_n \end{pmatrix}$$

2. Алгоритм прогонки
   1. Прямий хід: Спочатку обчислюються проміжні коефіцієнти:

$$\alpha_i = \frac{-c_i}{b_i + a_i\alpha_{i-1}}, \beta_i = \frac{d_i - a_i\beta_{i-1}}{b_i + a_i\alpha_{i-1}}$$

для $i = 2,3,...,n$, де $\alpha_1 = \frac{-c_1}{b_1}, \beta_1 = \frac{d_1}{b_1}$

   2. Зворотній хід: Після обчислення α та β, розв'язок знаходиться з кінця системи:

$$x_n = \beta_n, x_i = \alpha_i x_{i+1} + \beta_i \text{ для } i = n-1, n-2, ..., 1$$

Обчислення оберенної матриці

Обернена матриця A^{-1} до квадратної матриці. A обчислюється з умови:

$$AA^{-1} = I$$

Де I - одинична матриця

Алгоритм обчислення оберненої матриці :

1. Приєднуємо до матриці **A** одиничну матрицю I і застосовуємо метод Гаусса для приведення A до одиничної матриці
2. У результаті, права частина стане оберненою матрацею A^{-1}

Формули для оберненої матриці:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \text{ тоді } A^{-1} = \frac{1}{ad - bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Для більшого розміру матриць використовуються алгоритми, такі як метод Гаусса або розклади LU

Для структурування звіту на основі коду, який ви надали, можна використовувати наступну структуру:

Хід роботи:

2. Опис вихідних даних

У даній роботі матриця системи генерується випадковим чином з дотриманням діагональної переваги для забезпечення збіжності в методі Якобі та симетрії для застосування методу прогонки. Вектор правої частини також генерується випадковим чином.

Генерована матриця A (4x4) і вектор правих частин b:

Приклад:

$$A = \begin{pmatrix} 10 & 2 & 0 & 0 \\ 2 & 10 & 2 & 0 \\ 0 & 2 & 10 & 2 \\ 0 & 0 & 2 & 10 \end{pmatrix} \quad b = \begin{pmatrix} 7 \\ 1 \\ 9 \\ 4 \end{pmatrix}$$

3. Реалізація методу Гаусса

Метод Гаусса реалізований для приведення системи до верхньої трикутної форми з подальшим зворотним ходом. Перед початком розв'язку перевіряється, чи можливо застосувати цей метод через обчислення визначника.

$$\textbf{Крок 1:Поточна матриця } A : A = \begin{pmatrix} 10 & 2 & 0 & 0 \\ 2 & 10 & 2 & 0 \\ 0 & 2 & 10 & 2 \\ 0 & 0 & 2 & 10 \end{pmatrix} \textbf{Поточний вектор } b : b = \begin{pmatrix} 7 \\ 1 \\ 9 \\ 4 \end{pmatrix}$$

$$\textbf{Матриця перестановок (P)} : P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \textbf{Матриця усунення (M)} : M = \begin{pmatrix} 0.1 & 0 & 0 & 0 \\ -0.2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Після першого кроку ми починаємо перетворювати елементи матриці A та вектора b. Для цього вибирається перший елемент головної діагоналі і нормалізується (ділиться на нього сам вектор b, і всі рядки матриці обробляються так, щоб на позиціях поза діагоналлю були нулі або потрібні співвідношення).

- Матриця A змінюється так, що перший рядок стає нормалізованим з першого елемента, тобто перший елемент діагоналі стає 1.
- Вектор b відповідно змінюється для першого рівняння. Замість $b_1=7$, він стає $b_1=0.7$
- Решта рядків матриці модифікуються з використанням матриці усунення M

$$\textbf{Крок 2:Поточна матриця } A : A = \begin{pmatrix} 1 & 0.2 & 0 & 0 \\ 0 & 9.6 & 2 & 0 \\ 0 & 2 & 10 & 2 \\ 0 & 0 & 2 & 10 \end{pmatrix} \textbf{Поточний вектор } b : b = \begin{pmatrix} 0.7 \\ -0.4 \\ 9 \\ 4 \end{pmatrix}$$

$$\textbf{Матриця перестановок (P)} : P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \textbf{Матриця усунення (M)} : M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.1042 & 0 & 0 \\ 0 & -0.2083 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

На другому кроці ми продовжуємо процес нормалізації для другого рядка. Ми видаляємо коефіцієнти, що стоять нижче діагоналі.

- Другий рядок матриці А нормалізується так, що головний елемент діагоналі (2-й) стає 1. Це досягається поділом другого рівняння на 9.6, і результат стає $A_{2,2} = 1$.
- Вектор b модифікується відповідно до нової матриці A, тепер $b_2 = -0.4$.
- Матриця усунення M оновлюється для збереження зв'язків між рядками.

Крок 3:Поточна матриця $A : A = \begin{pmatrix} 1 & 0.2 & 0 & 0 \\ 0 & 1 & 0.2083 & 0 \\ 0 & 0 & 9.5833 & 2 \\ 0 & 0 & 2 & 10 \end{pmatrix}$ **Поточний вектор** $b : b = \begin{pmatrix} 0.7 \\ -0.0417 \\ 9.0833 \\ 4 \end{pmatrix}$

Матриця перестановок (P) $: P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ **Матриця усунення (M)** $: M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.1043 & 0 \\ 0 & 0 & -0.2087 & 1 \end{pmatrix}$

Третій крок передбачає обробку третього рядка.

- Третій елемент головної діагоналі (2-й рядок) стає нормалізованим на 1, тобто $A_{3,3} = 1$, після поділу рядка на 9.5833.
- Відповідно, b_3 стає рівним $b_3 = 9.0833$.
- Матриця А оновлюється, зменшуючи вплив елементів за межами діагоналі.

Крок 4:Поточна матриця $A : A = \begin{pmatrix} 1 & 0.2 & 0 & 0 \\ 0 & 1 & 0.2083 & 0 \\ 0 & 0 & 1 & 0.2087 \\ 0 & 0 & 0 & 9.5826 \end{pmatrix}$ **Поточний вектор** $b : b = \begin{pmatrix} 0.7 \\ -0.0417 \\ 0.9478 \\ 2.1043 \end{pmatrix}$

Матриця перестановок (P) $: P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ **Матриця усунення (M)** $: M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.1044 \end{pmatrix}$

На цьому кроці нормалізується останній рядок.

- Четвертий рядок нормалізується так, що $A_{4,4} = 1$, шляхом поділу на 9.5826.
- Вектор b_4 оновлюється до $b_4 = 2.1043$, відповідно до нової матриці.
- Матриця усунення M гарантує, що в нижньому лівому куті залишаються нулі.

Фінальна матриця $: A = \begin{pmatrix} 1 & 0.2 & 0 & 0 \\ 0 & 1 & 0.2083 & 0 \\ 0 & 0 & 1 & 0.2087 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ **Фінальний вектор** $b : b = \begin{pmatrix} 0.7 \\ -0.0417 \\ 0.9478 \\ 0.2196 \end{pmatrix}$

Після завершення всіх кроків маємо діагональну матрицю A із одиницями на діагоналі, а вектор b перетворений у відповідні значення, які можна використовувати для вирішення системи рівнянь.

Результати:

$$\text{Solution using Gauss method:} \begin{pmatrix} 0.74591652 \\ -0.22958258 \\ 0.90199637 \\ 0.21960073 \end{pmatrix}$$

Перевірка:

1: 10*0.7459 + 2*(-0.2296) = 7.0 ≈ 7
2: 2*0.7459 + 10*(-0.2296) + 2*0.9020 = 1.0 ≈ 1
3: 2*(-0.2296) + 10*0.9020 +2*0.2196 = 9.0 ≈ 9
4: 2*0.9020 + 10*0.2196 = 4.0 ≈ 4

4. Обчислення визначника та оберненої матриці
Детермінант обчислюється за Гауссом і дорівнює 8816.0000

Обчислення оберненої матриці:
Прямий хід:

$$\text{Початкова розширена матриця } [A\,|\,I]: \begin{pmatrix} 10 & 2 & 0 & 0 & | & 1 & 0 & 0 & 0 \\ 2 & 10 & 2 & 0 & | & 0 & 1 & 0 & 0 \\ 0 & 2 & 10 & 2 & | & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 10 & | & 0 & 0 & 0 & 1 \end{pmatrix}$$

Побудована розширена матриця `[A | I]`, де `A` — це початкова матриця, а `I` — одинична матриця.

Крок 1
Нормалізуємо перший рядок, роблячи перший елемент рівним 1:

$$\begin{pmatrix} 10 & 2 & 0 & 0 & | & 1 & 0 & 0 & 0 \\ 2 & 10 & 2 & 0 & | & 0 & 1 & 0 & 0 \\ 0 & 2 & 10 & 2 & | & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 10 & | & 0 & 0 & 0 & 1 \end{pmatrix}$$

Крок 2
Виконуємо віднімання другого рядка, щоб нуль на другій позиції першого рядка залишився:

$$\begin{pmatrix} 1 & 0.2 & 0 & 0 & | & 0.1 & 0 & 0 & 0 \\ 0 & 1 & 0.208333 & 0 & | & -0.020833 & 0.104167 & 0 & 0 \\ 0 & 2 & 10 & 2 & | & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 10 & | & 0 & 0 & 0 & 1 \end{pmatrix}$$

Крок 3
Нормалізуємо третій рядок та приводимо його до одиничного вигляду:

$$\left(\begin{array}{cccc|cccc} 1 & 0.2 & 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 1 & 0.208333 & 0 & -0.020833 & 0.104167 & 0 & 0 \\ 0 & 0 & 1 & 0.208696 & 0.004348 & -0.02174 & 0.104348 & 0 \\ 0 & 0 & 2 & 10 & 0 & 0 & 0 & 1 \end{array}\right)$$

Крок 4
Приводимо останній рядок до потрібного вигляду, роблячи останній елемент 1:

$$\left(\begin{array}{cccc|cccc} 1 & 0.2 & 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 1 & 0.208333 & 0 & -0.020833 & 0.104167 & 0 & 0 \\ 0 & 0 & 1 & 0.208696 & 0.004348 & -0.02174 & 0.104348 & 0 \\ 0 & 0 & 0 & 1 & -0.000907 & 0.004537 & -0.021778 & 0.104356 \end{array}\right)$$

Зворотний хід

Крок 1
Нормалізуємо перший рядок і отримуємо нову матрицю:

$$\left(\begin{array}{cccc|cccc} 1 & 0.2 & 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 1 & 0.208333 & 0 & -0.020833 & 0.104167 & 0 & 0 \\ 0 & 0 & 1 & 0.208696 & 0.004348 & -0.02174 & 0.104348 & 0 \\ 0 & 0 & 0 & 1 & -0.000907 & 0.004537 & -0.021778 & 0.104356 \end{array}\right)$$

Крок 2
Проводимо перетворення другого та третього рядків, щоб позбутися елементів нижче головної діагоналі:

$$\left(\begin{array}{cccc|cccc} 1 & 0.2 & 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -0.021778 & 0.110889 & -0.022686 & 0.004537 \\ 0 & 0 & 1 & 0 & 0.004537 & -0.022686 & 0.110889 & -0.021778 \\ 0 & 0 & 0 & 1 & -0.000907 & 0.004537 & -0.021778 & 0.104356 \end{array}\right)$$

Крок 3
Далі нормалізуємо перші два стовпці та отримуємо останні елементи:

$$\left(\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0.104356 & -0.021778 & 0.004537 & -0.000907 \\ 0 & 1 & 0 & 0 & -0.021778 & 0.110889 & -0.022686 & 0.004537 \\ 0 & 0 & 1 & 0 & 0.004537 & -0.022686 & 0.110889 & -0.021778 \\ 0 & 0 & 0 & 1 & -0.000907 & 0.004537 & -0.021778 & 0.104356 \end{array}\right)$$

Крок 4

Перевіряємо і підтверджуємо, що отримали остаточні значення:

$$\left(\begin{array}{cccccccccc} 1 & 0 & 0 & 0 & | & 0.104356 & -0.021778 & 0.004537 & -0.000907 \\ 0 & 1 & 0 & 0 & | & -0.021778 & 0.110889 & -0.022686 & 0.004537 \\ 0 & 0 & 1 & 0 & | & 0.004537 & -0.022686 & 0.110889 & -0.021778 \\ 0 & 0 & 0 & 1 & | & -0.000907 & 0.004537 & -0.021778 & 0.104356 \end{array}\right)$$

Обернена матриця для матриці A виглядає наступним чином:

$$A^{-1} = \begin{pmatrix} 0.10435572 & -0.02177858 & 0.00453721 & -0.00090744 \\ -0.02177858 & 0.10889292 & -0.02268603 & 0.00453721 \\ 0.00453721 & -0.02268603 & 0.10889292 & -0.02177858 \\ -0.00090744 & 0.00453721 & -0.02177858 & 0.10435572 \end{pmatrix}$$

5. Метод Якобі

Ітерація 1: $\mathbf{x}^{(1)} = [0.7 \quad 0.1 \quad 0.9 \quad 0.4]$

Ітерація 2: $\mathbf{x}^{(2)} = [0.68 \quad -0.22 \quad 0.8 \quad 0.22]$

Ітерація 3: $\mathbf{x}^{(3)} = [0.744 \quad -0.196 \quad 0.9 \quad 0.24]$

Ітерація 4: $\mathbf{x}^{(4)} = [0.7392 \quad -0.2288 \quad 0.8912 \quad 0.22]$

Ітерація 5: $\mathbf{x}^{(5)} = [0.74576 \quad -0.22608 \quad 0.90176 \quad 0.22176]$

Ітерація 6: $\mathbf{x}^{(6)} = [0.74522 \quad -0.2295 \quad 0.90086 \quad 0.21965]$

Ітерація 7: $\mathbf{x}^{(7)} = [0.7459 \quad -0.22922 \quad 0.90197 \quad 0.21983]$

Ітерація 8: $\mathbf{x}^{(8)} = [0.74584 \quad -0.22957 \quad 0.90188 \quad 0.21961]$

Ітерація 9: $\mathbf{x}^{(9)} = [0.74591 \quad -0.22954 \quad 0.90199 \quad 0.21962]$

Ітерація 10: $\mathbf{x}^{(10)} = [0.74591 \quad -0.22958 \quad 0.90198 \quad 0.2196]$

Ітерація 11: $\mathbf{x}^{(11)} = [0.74592 \quad -0.22958 \quad 0.902 \quad 0.2196]$

Ітерація 12: $\mathbf{x}^{(12)} = [0.74592 \quad -0.22958 \quad 0.902 \quad 0.2196]$

Ітерація 13: $\mathbf{x}^{(13)} = [0.74592 \quad -0.22958 \quad 0.902 \quad 0.2196]$

Ітерація 14: $\mathbf{x}^{(14)} = [0.74592 \quad -0.22958 \quad 0.902 \quad 0.2196]$

Ітерація 15: $\mathbf{x}^{(15)} = [0.74592 \quad -0.22958 \quad 0.902 \quad 0.2196]$

Ітерація 16: $\mathbf{x}^{(16)} = [0.74592 \quad -0.22958 \quad 0.902 \quad 0.2196]$

Ітерація 17: $\mathbf{x}^{(17)} = [0.74592 \quad -0.22958 \quad 0.902 \quad 0.2196]$

Ітерація 18: $\mathbf{x}^{(18)} = [0.74592 \quad -0.22958 \quad 0.902 \quad 0.2196]$

Ітерація 19: $\mathbf{x}^{(19)} = [0.74592 \quad -0.22958 \quad 0.902 \quad 0.2196]$

Ітерація 20: $\mathbf{x}^{(20)} = [0.74592 \quad -0.22958 \quad 0.902 \quad 0.2196]$

Ітерація 21: $\mathbf{x}^{(21)} = [0.74592 \quad -0.22958 \quad 0.902 \quad 0.2196]$

Ітерація 22: $\mathbf{x}^{(22)} = [0.74592 \quad -0.22958 \quad 0.902 \quad 0.2196]$

Ітерація 23: $\mathbf{x}^{(23)} = [0.74592 \quad -0.22958 \quad 0.902 \quad 0.2196]$

Ітерація 24: $\mathbf{x}^{(24)} = [0.74592 \quad -0.22958 \quad 0.902 \quad 0.2196]$

Метод Якобі збігся на 24-й ітерації з розв'язком:

$$x = [0.74591652 \quad -0.22958258 \quad 0.90199637 \quad 0.21960073]$$

Перевірка:

1: 10*0.7459 + 2*(-0.2296) = 7.0 ≈ 7
2: 2*0.7459 + 10*(-0.2296) + 2*0.9020 = 1.0 ≈ 1
3: 2*(-0.2296) + 10*0.9020 +2*0.2196 = 9.0 ≈ 9
4: 2*0.9020 + 10*0.2196 = 4.0 ≈ 4

6. Метод прогонки (для тридіагональних матриць)
Метод прогонки застосовується для розв'язку систем із тридіагональними матрицями. У цьому методі проводиться прямий та зворотний хід для знаходження розв'язку.

Результат:

$$\text{Матриця } A = \begin{bmatrix} 10 & 2 & 0 & 0 \\ 2 & 9.6 & 2 & 0 \\ 0 & 2 & 10 & 2 \\ 0 & 0 & 2 & 10 \end{bmatrix} \text{Вектор } b = \begin{bmatrix} 7 \\ 0 \\ 9 \\ 4 \end{bmatrix}$$

$$\text{Крок 2:} A = \begin{bmatrix} 10 & 2 & 0 & 0 \\ 2 & 9.6 & 2 & 0 \\ 0 & 2 & 9.58333 & 2 \\ 0 & 0 & 2 & 10 \end{bmatrix} b = \begin{bmatrix} 7 \\ 0 \\ 9 \\ 4 \end{bmatrix}$$

$$\text{Крок 3:} A = \begin{bmatrix} 10 & 2 & 0 & 0 \\ 2 & 9.6 & 2 & 0 \\ 0 & 2 & 9.58333 & 2 \\ 0 & 0 & 2 & 9.58261 \end{bmatrix} b = \begin{bmatrix} 7 \\ 0 \\ 9 \\ 2 \end{bmatrix}$$

$$\text{Зворотний хід, крок 1:} d = \begin{bmatrix} 0 \\ 0 \\ 0.89557 \\ 0.20871 \end{bmatrix}$$

$$\text{Зворотний хід, крок 2:} d = \begin{bmatrix} 0 \\ -0.18658 \\ 0.89557 \\ 0.20871 \end{bmatrix}$$

$$\text{Зворотний хід, крок 3:} d = \begin{bmatrix} 0.73732 \\ -0.18658 \\ 0.89557 \\ 0.20871 \end{bmatrix}$$

$$\text{Розв'язок за методом прогонки:} \begin{bmatrix} 0.73731555 \\ -0.18657776 \\ 0.89557327 \\ 0.20871143 \end{bmatrix}$$

Перевірка:

1: 10*0.7373 + 2*(-0.1866) = 7.0 ≈ 7
2: 2*0.7373 + 10*(-0.1866) + 2*0.8956 = 1.0 ≈ 1
3: 2*(-0.1866) + 10*0.8956 +2*0.2087 = 9.0 ≈ 9
4: 2*0.8956 + 10*0.2087 = 4.0 ≈ 4

Висновки

Для аналізу та вибору оптимального методу розглянемо особливості кожного з трьох методів:

1. Метод Гаусса:
   - Показав хорошу ефективність у розв’язанні систем і при обчисленні обернених матриць.
   - Має універсальне застосування, оскільки підходить для будь-якої квадратної матриці без спеціальних вимог до її структури.
   - Результат збігається з методом Якобі, що свідчить про його точність у цьому випадку.

2. Метод Якобі:
   - Дає результат, який збігається з методом Гаусса, отже, є надійним.
   - Однак для його збіжності потрібна діагональна перевага матриці, що обмежує його застосування у випадках, коли цієї умови не дотримано.
   - Якобі є ітераційним методом, і для великих систем або слабкозбіжних випадків може знадобитися більше ітерацій, ніж для методу Гаусса.

3. Метод прогонки:
   - Є найефективнішим для тридіагональних систем, оскільки спеціально оптимізований для таких структур, забезпечуючи швидкість і точність.
   - Метод прогонки менш універсальний, адже його можна застосовувати лише для систем із тридіагональною структурою.

- Відмінності в отриманих результатах порівняно з іншими методами можуть бути зумовлені обчислювальними похибками, але в цьому випадку вони незначні.

- Якщо система має тридіагональну структуру, найкраще використовувати метод прогонки завдяки його швидкості й оптимізації для таких структур.
- У випадках загальної системи рівнянь, коли структура матриці довільна, найефективніше рішення — метод Гаусса через його універсальність і точність.
- Метод Якобі є альтернативою для великих систем із діагональною перевагою, хоча поступається Гауссу за швидкістю збіжності в системах без такої властивості.