



**Визначення.** Нев'язкою системи (3.2), що відповідає довільному вектору  $x$ , називається вектор  $r = Ax - b$ . Очевидно, що для розв'язку  $x^* \in R^n$  невязка  $(Ax - b)$  дорівнює нулю.

Для того щоб система (3.2) мала єдиний розв'язок, необхідно й достатньо, щоб  $\det A \neq 0$  [Ошибка! Источник ссылки не найден.]. У цьому випадку розв'язок системи (3.2) може бути отриманий за формулами Крамера:

$$x_j = \frac{\det(A^{(j)})}{\det(A)}, \quad j = \overline{1, n},$$

де матриця  $A^{(j)}$  утворюється з матриці  $A$  заміною її  $j$ -го стовпця стовпцем вільних членів  $b$ .

Але такий спосіб розв'язання системи лінійних рівнянь з  $n$  невідомими призводить до обчислення  $(n+1)$ -го визначника порядку  $n$ , що є дуже трудомісткою операцією при великих  $n$ .

Чисельні методи розв'язання систем лінійних алгебраїчних рівнянь (3.2) можна розбити на дві групи: прямі (точні) й ітераційні (наближені).

**Визначення. Точними** (прямими) **методами** називаються методи, які в припущенні, що обчислення ведуться точно (без округлень), приводять за скінчене число кроків до точних значень  $x_j^*$ . Оскільки обчислення на комп'ютері ведуться з округленнями, то розв'язок неминуче міститиме погрішності. До прямих методів відносяться, наприклад, метод Гауса, метод Холецького та ін. [Ошибка! Источник ссылки не найден.; Ошибка! Источник ссылки не найден.; Ошибка! Источник ссылки не найден.; Ошибка! Источник ссылки не найден.].

**Визначення. Наближеними** (ітераційними) **методами** називаються такі методи, які навіть у припущенні, що обчислення ведуться без округлень, дозволяють отримати розв'язок  $x^*$  тільки із зазначеною точністю. Точний розв'язок системи в цьому випадку може бути отриманий теоретично як результат нескінченного (ітераційного) процесу. До наближених методів відносяться, наприклад, метод ітерацій, метод Зейделя та ін. [Ошибка! Источник ссылки не найден.; Ошибка! Источник ссылки не найден.; Ошибка! Источник ссылки не найден.]. Кожний із цих методів не завжди є збіжним у застосуванні до конкретного класу систем лінійних алгебраїчних рівнянь.

## 1.1. Метод виключення Гауса

Найбільш відомим із прямих методів розв'язання системи (3.2) є **метод виключення Гауса**, ідея якого полягає в послідовному виключенні невідомих із рівнянь. Спочатку відправна система (3.2) приводиться до системи трикутного вигляду (прямий хід), а потім невідомі визначаються за простими формулами (зворотний хід).

Розрахункові формули методу Гауса:

Нехай  $a_{ij}^{(0)} = a_{ij}$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, n}$ ;  $b_i^{(0)} = b_i$ ,  $i = \overline{1, n}$ .

**Прямий хід:**  $k$ -й крок

при  $k = \overline{1, n-1}$ :

$$a_{kj}^{(k)} = \frac{a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}, \quad j = \overline{k+1, n}; \quad b_k^{(k)} = \frac{b_k^{(k-1)}}{a_{kk}^{(k-1)}};$$

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{ik}^{(k-1)} a_{kj}^{(k)}, \quad j = \overline{k+1, n},$$

$$b_i^{(k)} = b_i^{(k-1)} - a_{ik}^{(k-1)} b_k^{(k)}, \quad i = \overline{k+1, n};$$

при  $k = n$ :

$$b_k^{(k)} = \frac{b_k^{(k-1)}}{a_{kk}^{(k-1)}}.$$

**Зворотний хід:**

$$x_n = b_n^{(n)};$$

$$x_k = b_k^{(k)} - \sum_{j=k+1}^n a_{kj}^{(k)} x_j, \quad k = \overline{n-1, 1}. \quad (3.3)$$

Тут  $A^{(k)} = (a_{ij}^{(k)})$  – це проміжна матриця, а  $b^{(k)} = (b_i^{(k)})$  – проміжний вектор на  $k$ -му кроці прямого ходу методу Гауса, причому  $A^{(0)} = A$ ,  $b^{(0)} = b$ . Елементи  $a_{kk}^{(k)} = 1$  для усіх  $k$  формально, але виконувати це обчислювальня фактично не потрібно. Також не потрібно обчислювати і піддіагональні (нульові) елементи матриці.

Рядки, що містять одиницю на діагоналі, називаються **виділеними рядками**. Процес отримання виділених рядків (приведення системи до трикутного вигляду) називається **прямим ходом**, а процес знаходження невідомих шляхом використання виділених рядків – **зворотним ходом** методу Гауса.

**Приклад 3.1.** Розв'язати систему рівнянь методом Гауса.

$$\begin{cases} 5x_1 + x_2 + 2x_3 = 10, \\ 6x_1 + 18x_2 + 6x_3 = 54, \\ 10x_1 + 20x_2 + 40x_3 = 160. \end{cases}$$

### Розв'язання в математичному пакеті R

Процедуру розв'язання системи лінійних рівнянь методом Гауса можна записати так:

```
MGaussSLE = function(A,b,n)
{
  for(k in 1:(n-1))
  {
    for(j in (k+1):n)
      A[k,j] = A[k,j]/A[k,k]
    b[k] = b[k]/A[k,k]
    for(i in (k+1):n)
    {
      for(j in (k+1):n)
        A[i,j] = A[i,j] - A[i,k]*A[k,j]
      b[i] = b[i] - A[i,k]*b[k]
    }
  }
  b[n] = b[n]/A[n,n]
  for(k in (n-1):1)
  {
    for(j in (k+1):n)
      b[k] = b[k] - A[k,j]*b[j]
  }
  return(b)
}
```

Відправними даними задачі (3.1) є матриця  $A$  коефіцієнтів при невідомих, вектор-стовпець  $b$  вільних членів і кількість невідомих  $n$ :

```
> A = cbind(c(5,6,10), c(1,18,20), c(2,6,40))
> b = c(10,54,160)
> n = 3
```

```

> A
      [,1] [,2] [,3]
[1,]    5    1    2
[2,]    6   18    6
[3,]   10   20   40
> b
[1] 10 54 160
> n
[1] 3

```

Результат розв'язання заданої системи лінійних рівнянь з використанням записаної процедури:

```

> x = MGaussSLE(A,b,n)
> x
[1] 0.4444444 1.8666667 2.9555556

```

Таким чином, розв'язком заданої системи лінійних рівнянь є знайдені значення невідомих:  $x_1 = 0.444$ ,  $x_2 = 1.867$ ,  $x_3 = 2.956$ . Для перевірки отриманих результатів знайдені значення невідомих були підставлені у систему:

```

> y = A**x
> y
      [,1]
[1,]   10
[2,]   54
[3,]  160

```

Кількість арифметичних операцій, необхідних для реалізації методу Гауса, визначається формулою [Ошибка! Источник ссылки не найден.]

$$K(n) = \frac{2n(n+1)(n+2)}{2} + n(n-1),$$

де  $n$  – розмірність системи (3.2), тобто пропорційна кубу числа невідомих ( $O(n^3)$ ).

Слід зазначити, що з логічної точки зору кращою є модифікація методу Гауса, що зветься **метод Гауса – Жордана**. Його суть полягає в тому, що зворотний хід виконується не за формулами (3.3), а проводиться виключення невідомих у зворотному порядку з рівнянь, отриманих після виконання прямого ходу методу Гауса. При цьому система приводиться до діагонального вигляду з одиницями на діагоналі, а розв'язок системи виявляється на місці вектора  $b$ . Таким чином, зворотний хід дійсно є

зворотним, тобто виконуються операції симетричні відносно прямого ходу.

### 3.2. Метод Гауса з вибором головного елемента

Стандартний метод Гауса може стати чисельно нестійким, якщо серед  $a_{kk}^{(k-1)}$ , на які проводиться ділення, виявляються дуже малі числа по абсолютній величині, хоча і відмінні від нуля (не говорячи вже про нульові значення). Тоді при діленні на них отримуються великі числа з великими абсолютними погрішностями. В результаті цього отриманий розв'язок сильно відрізнятиметься від дійсного розв'язку, тобто метод Гауса буде нестійким до помилок обчислень.

Щоб цього уникнути, на практиці застосовують **метод виключення Гауса з вибором головного елемента**. Діагональний елемент  $a_{kk}^{(k-1)}$ , на який проводиться ділення на  $k$ -му кроці, називається **головним** (ведучим) елементом. Якщо головний елемент близький до нуля по абсолютній величині, то можна знайти у відповідному ( $k$ -му) стовпці максимальний за модулем елемент і переставити рядки місцями так, щоб цей елемент став головним. Але така перестановка теж не завжди забезпечує стійкість. Тому при програмній реалізації зазвичай вибирають максимальний за модулем елемент не в  $k$ -му стовпці, а в усій матриці, що залишилася. Потрібно зазначити, що якщо доводиться переставляти місцями стовпці матриці, то ці перестановки потрібно запам'ятовувати, а потім (після отримання розв'язку) проводити їх у зворотному порядку вже у векторі отриманого розв'язку.

Сенс вибору головного елемента полягає в тому, щоб зробити якомога меншими по абсолютній величині числа  $a_{kj}^{(k)} = \frac{a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}$  і, тим самим, зменшити погрішність обчислень і округлень. Тому для реалізації методу Гауса на комп'ютері зазвичай використовують саме схему з вибором головного елемента.

### 3.3. LU-розкладання матриці, метод Холецького

Метод LU-розкладання в принципі еквівалентний методу Гауса, відмінність полягає тільки в порядку дій.

У методі LU-розкладання матриця  $A$  системи (3.2) спочатку подається у вигляді **LU-розкладання**, тобто у вигляді добутку двох матриць

$$A = LU, \quad (3.4)$$

де  $L$  – нижньотрикутна матриця;

$U$  – верхньотрикутна матриця з одиницями на діагоналі.

Тоді розв'язання системи (3.2) проводиться в два етапи: спочатку розв'язується система

$$Ly = b, \quad (3.5)$$

відносно  $y \in R^n$ , а потім вже знаходиться шуканий розв'язок  $x^*$  шляхом розв'язання системи

$$Ux = y. \quad (3.6)$$

Оскільки матриці  $L$ ,  $U$  – трикутні, то знаходження розв'язків систем (3.5) та (3.6) проводиться за простими формулами, аналогічними формулам зворотного ходу методу Гауса.

Можна ввести позначення:

$$L = \begin{pmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{pmatrix}, \quad U = \begin{pmatrix} 1 & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & 1 & u_{23} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}.$$

Тоді зі співвідношення (3.4) будуть отримані формули для визначення елементів матриць  $L$  і  $U$ :

$$l_{j1} = a_{j,1},$$

$$l_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}, \quad i \geq j > 1$$

і

$$u_{1j} = \frac{a_{1j}}{l_{11}},$$

$$u_{ij} = \frac{1}{l_{ii}} \left( a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj} \right), \quad 1 < i < j.$$

**Приклад 3.2.** Розв'язати систему рівнянь з прикладу 3.1 методом LU-розкладання.

### Розв'язання в математичному пакеті R

Процедуру розв'язання системи лінійних рівнянь методом LU-розкладання можна записати так:

```
MetLUSLE = function(A,b,n)
{
  L = matrix(0, nrow=n, ncol=n)
  U = matrix(0, nrow=n, ncol=n)

  for(i in 1:n)
    L[i,1] = A[i,1]
  for(j in 1:n)
    U[1, j] = A[1, j]/L[1,1]
  for(i in 1:n)
  {
    for(j in 1:n)
    {
      s = 0
      for(k in 1:i)
        s = s + L[i,k]*U[k, j]
      if( (i >= j) & (j > 1) )
        L[i, j] = A[i, j] - s
      if( (j > i) & (i > 1) )
        U[i, j] = (A[i, j] - s)/L[i, i]
    }
    U[i, i] = 1
  }
  y = solve(L, b)
  x = solve(U, y)
  return(x)
}
```

Відправними даними задачі (3.1) є матриця  $A$  коефіцієнтів при невідомих, вектор-стовпець  $b$  вільних членів і кількість невідомих  $n$ :

```
> A = cbind(c(5,6,10), c(1,18,20), c(2,6,40))
> b = c(10,54,160)
> n = 3
> A
      [,1] [,2] [,3]
[1,]    5    1    2
[2,]    6   18    6
[3,]   10   20   40
> b
[1] 10 54 160
```



Результат розв'язання заданої системи лінійних рівнянь з використанням записаної процедури:

```
> x = MetLUSLE(A,b,3)
> x
[1] 0.4444444 1.8666667 2.9555556
```

Таким чином, розв'язком заданої системи лінійних рівнянь є знайдені значення невідомих:  $x_1 = 0.444$ ,  $x_2 = 1.867$ ,  $x_3 = 2.956$ .

Слід зазначити, що якщо  $A$  – симетрична знакопевна матриця [Ошибка! Источник ссылки не найден.], то розкладання (3.4) може бути записане у вигляді:

$$A = U^T D U, \quad (3.7)$$

де  $U^T$  – нижньотрикутна матриця, транспонована до  $U$ ;

$$D - \text{діагональна матриця } D = \begin{pmatrix} d_{11} & 0 & 0 & \dots & 0 \\ 0 & d_{22} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & d_{nn} \end{pmatrix}, \text{ всі } d_{ij} \text{ мають}$$

однаковий знак.

Застосування розкладання (3.7) для розв'язання системи (3.2) має назву **методу Холецького**.

Оскільки проблема розв'язання системи лінійних алгебраїчних рівнянь виду (3.2) з'являється, як правило, як проміжна задача при розв'язанні більш складних задач (розв'язання систем нелінійних рівнянь, оптимізації та ін.), то використання розкладань (3.4) та (3.7) часто буває корисним для отримання додаткової інформації про відповідну задачу. Наприклад, коли матриця  $A$  обчислюється як матриця других похідних деякої функції  $f(x)$   $n$  змінних у деякій точці  $x^{(0)} \in R^n$ , то якщо всі діагональні елементи матриці  $D$  з розкладання (3.7) додатні, то це означає, що функція  $f(x)$  опукла в околі точки  $x^{(0)}$ .

Розкладання (3.4) може бути використане і для обчислення визначника ( $\det A$ ) матриці  $A$ . Так  $\det A = \det(L)\det(U)$  [Ошибка! Источник ссылки не найден.], але визначник трикутної матриці дорівнює добутку діагональних елементів. Тому  $\det(U) = 1$ , оскільки матриця  $U$  –

трикутна і має одиниці на діагоналі, а  $\det(L) = \prod_{i=1}^n l_{ij}$ . Тоді

$$\det(A) = \det(L) = \prod_{i=1}^n l_{ij}.$$

### 3.4. Метод ітерацій

Як приклад ітераційного (наближеного) методу розв'язання системи (3.2) варто розглянути **метод ітерацій**.

Для застосування методу ітерацій систему (3.2) необхідно подати у вигляді

$$x = C x + d, \quad (3.8)$$

де  $C \in R^{n \times n}$ ,  $d \in R^n$ .

Алгоритм методу ітерацій.

1. Задається  $\varepsilon > 0$  – точність розв'язку задачі і початкове наближення розв'язку  $x^{(0)} \in R^n$  (наприклад,  $x^{(0)} = d$ ).

2. На  $k$ -й ітерації методу ( $k = 0, 1, 2, \dots$ ) обчислюється наступне наближення

$$x^{(k+1)} = C x^{(k)} + d. \quad (3.9)$$

3. Перевіряється критерій останову  $\frac{q}{1-q} \|x^{(k+1)} - x^{(k)}\| \leq \varepsilon$ , де  $0 < q < 1$  (визначається з умови збіжності).

Метод простої ітерації збігається тільки при виконанні умови:

$$\|C\| \leq q < 1, \quad (3.10)$$

де  $0 < q < 1$  [Ошибка! Источник ссылки не найден.; Ошибка! Источник ссылки не найден.; Ошибка! Источник ссылки не найден.; Ошибка! Источник ссылки не найден.; Ошибка! Источник ссылки не найден.; Ошибка! Источник ссылки не найден.].

При цьому як норму матриці  $C$  можна розглядати величину

$$\|C\| \equiv \max_j \sum_{i=1}^n |c_{ij}| \quad \text{або} \quad \|C\| \equiv \max_i \sum_{j=1}^n |c_{ij}|. \quad (3.11)$$



$$\sum_{\substack{i=1 \\ i \neq j}}^n \frac{|a_{ij}|}{|a_{jj}|} \leq q < 1, \quad j = \overline{1, n} \quad \text{або} \quad \sum_{\substack{j=1 \\ j \neq i}}^n \frac{|a_{ij}|}{|a_{ii}|} \leq q < 1, \quad i = \overline{1, n}. \quad (3.12)$$

У свою чергу, ці нерівності будуть справедливими, якщо діагональні елементи матриці  $A$  задовольняють умову  $|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|$ ,  $i = \overline{1, n}$ , тобто

модулі діагональних коефіцієнтів для кожного рівняння системи (3.1) більше суми модулів решти всіх коефіцієнтів.

Слід зазначити, що описана процедура приведення системи (3.2) до виду (3.8) з подальшим застосуванням методу ітерацій називається **методом Якобі**.

**Приклад 3.3.** Розв'язати систему рівнянь методом Якобі з заданою точністю  $\varepsilon = 10^{-5}$

$$\begin{cases} 5x_1 + x_2 + 2x_3 = 10, \\ 6x_1 + 18x_2 + 6x_3 = 54, \\ 10x_1 + 20x_2 + 40x_3 = 160. \end{cases}$$

### Розв'язання в математичному пакеті R

Відправними даними задачі є матриця коефіцієнтів при невідомих  $A$ , вектор-стовпець вільних членів  $b$ , кількість невідомих  $n$ , точність обчислення  $\varepsilon$ . Для запобігання закцилювання ітераційного процесу слід задати додатково максимальну кількість ітерацій методу  $k_{\max}$ :

```
> A = cbind(c(5, 6, 10), c(1, 18, 20), c(2, 6, 40))
> b = c(10, 54, 160)
> n = 3
> eps = 0.0001
> kmax = 100
```

```

> A
      [,1] [,2] [,3]
[1,]    5    1    2
[2,]    6   18    6
[3,]   10   20   40
> b
[1] 10 54 160
> n
[1] 3
> eps
[1] 1e-04
> kmax
[1] 100

```

Процедуру розв'язання системи лінійних алгебраїчних рівнянь виду (3.8) методом ітерацій можна записати так:

```

MetIterSLE = function(C,d,n,eps,kmax)
{
  x0 = d
  for (k in 1:kmax)
  {
    x = C*x0 + d
    if( norm(x-x0) <= eps )
      break
    x0 = x
  }
  return(x)
}

```

Тоді процедуру розв'язання системи лінійних алгебраїчних рівнянь виду (3.8) методом Якобі можна записати так:

```

MetJakobiSLE = function(A,b,n,eps,kmax)
{
  for (i in 1:n)
  {
    for (j in 1:n)
    {
      C[i,j] = -A[i,j]/A[i,i]
    }
    C[i,i] = 0
    d[i] = b[i]/A[i,i]
  }
  x = MetIterSLE(C,d,n,eps,kmax)
  return (x)
}

```

Результат розв'язання заданої системи лінійних рівнянь із використанням записаних процедур:

```

> x = MetJakobiSLE (A,b,n,eps,kmax)
> x
          [,1]
[1,] 0.4444534
[2,] 1.8666762
[3,] 2.9555659

```

Перевірка розв'язку – обчислення нев'язки  $Ax - b$ :

```

> y = A%*%x -b
> y
          [,1]
[1,] 7.486361e-05
[2,] 2.867975e-04
[3,] 6.931325e-04

```

Таким чином, розв'язком поставленої задачі є знайдені значення невідомих:  $x_1 = 0.444$ ,  $x_2 = 1.867$ ,  $x_3 = 2.956$ . Для перевірки отриманих результатів обчислена нев'язка  $(Ax - b)$  для системи (3.2) при знайдених значеннях невідомих, тобто розв'язок отримано з точністю  $10^{-4}$ , як і було задано.

Метод Гауса для розв'язання системи (3.2) застосовують на практиці при  $n \leq 10^3$ , метод простої ітерації – при  $10^3 < n \leq 10^6$ . Метод простої ітерації також застосовують для уточнення розв'язку, отриманого методом Гауса.

### 3.5. Метод Гауса – Зейделя

Метод Гауса – Зейделя є модифікацією метода Якобі, який було описано у розділі 3.5 для розв'язання системи (3.2). Він також застосовується для матриць  $A$ , для яких виконується умова  $a_{ii} \neq 0$  для всіх  $i$ . По суті це той самий ітераційний процес, що і (3.9), але в ньому нове наближення  $x^{(k+1)}$  застосовується одразу ж, як змінився його  $i$ -й елемент. Ітераційна формула методу Гауса – Зейделя має вигляд:

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

Умови збіжності методу Гауса – Зейделя ті ж самі, що й у методу Якобі, тобто (3.12), але на практиці швидкість його збігу дещо вища. Тому саме цей метод застосовують найчастіше.

### 3.6. Обчислення оберненої матриці

**Визначення.** Оберненою до матриці  $A$  називається така матриця  $B$ , для якої виконується рівність

$$AB = BA = E, \quad (3.13)$$

де  $E$  – одинична матриця ( $E \in R^{n \times n}$ ), тобто

$$E = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}.$$

Обернену матрицю до  $A$  прийнято позначати через  $A^{-1}$ .

З умови (3.13) виходить, що обернена матриця  $A^{-1}$  вводиться тільки для квадратних матриць  $A$ , при цьому  $A^{-1}$  також буде квадратною тієї ж розмірності.

Квадратна матриця  $A$  називається **невиродженою** або **неособливою (несингулярною)**, якщо її визначник  $\det A$  відмінний від нуля. Будь-яка невивроджена матриця має обернену матрицю.

Якщо відома обернена матриця  $A^{-1}$ , то розв'язок системи (3.2) записується у вигляді  $x = A^{-1}b$ . Слід зазначити, що з практичної точки зору все ж таки ефективнішим буде розв'язання системи рівнянь, ніж обчислення оберненої матриці з подальшим множенням на вектор  $b$ .

Метод Гауса може бути застосований для обчислення оберненої матриці. Нехай дана невивроджена матриця  $A = (a_{ij})$  розмірності  $n \times n$ .

Елементи шуканої оберненої матриці  $A^{-1}$  можна позначити через  $(x_{ij})$ , а також ввести позначення  $x^j$  –  $j$ -й стовпець ( $x^j \in R^n$ ,  $j = \overline{1, n}$ ) матриці  $X = A^{-1}$ ,  $e^j$  –  $j$ -й стовпець ( $e^j \in R^n$ ,  $j = \overline{1, n}$ ) матриці  $E$ , тобто

$$x^j = \begin{pmatrix} x_{1j} \\ x_{2j} \\ \dots \\ x_{nj} \end{pmatrix}, \quad e^j = \begin{pmatrix} 0 \\ \dots \\ 1 \\ \dots \\ 0 \end{pmatrix}, \quad (e_j^j = 1).$$

Тоді рівність  $AX = E$  можна записати у вигляді

$$Ax^j = e^j, (j = \overline{1, n}), \quad (3.14)$$

тобто стовпці  $x^j$  ( $j = \overline{1, n}$ ) можуть бути знайдені методом Гауса як розв'язок  $n$  систем лінійних рівнянь з однією і тією ж матрицею  $A$ , але з різними векторами правих частин  $e^j$ . Звідси випливає, що прямий хід методу Гауса при розв'язанні систем (3.14) буде загальним, тобто його можна проводити одночасно для всіх  $j$ , а потім вже визначити вектори  $x^j$  ( $j = \overline{1, n}$ ), виконавши  $n$  разів зворотний хід методу Гауса.

Треба нагадати (див. розділ 3.2), що виконання зворотного ходу методу Гауса – Жордана еквівалентне приведенню трикутної системи, отриманої після прямого ходу, до системи з діагональною одиничною матрицею. Таким чином, приводячи початкову матрицю  $A$  до одиничної матриці і виконуючи аналогічні дії з елементами матриці  $E$ , ця матриця буде перетворена в матрицю  $A^{-1}$ . Звідси випливає, що обчислення оберненої матриці ефективно проводити саме за методом Гауса – Жордана.

**Приклад 3.4.** Знайти обернену матрицю до матриці  $A$  з прикладу 3.1 методом Гауса – Жордана.

### Розв'язання в математичному пакеті R

Відправними даними задачі є матриця  $A$ :

```
> A = cbind(c(5,6,10), c(1,18,20), c(2,6,40))
> A
      [,1] [,2] [,3]
[1,]    5    1    2
[2,]    6   18    6
[3,]   10   20   40
```

Процедуру обчислення оберненої матриці методом Гауса – Жордана можна записати так:

```
MGaussJord = function(A,n)
{
  # створюємо одиничну матрицю E
  E = matrix(0, nrow=n, ncol=n)
  for(i in 1:n)
    E[i,i] = 1
```



```

# приводимо матрицю A до трикутного виду (метод Гауса)
for(k in 1:n)
{
  for(j in 1:n)
    E[k,j] = E[k,j]/A[k,k]
  if(k==n)
    break
  for(j in (k+1):n)
    A[k,j] = A[k,j]/A[k,k]
  for(i in (k+1):n)
  {
    for(j in (k+1):n)
      A[i,j] = A[i,j] - A[i,k]*A[k,j]
    for(j in 1:n)
      E[i,j] = E[i,j] - A[i,k]*E[k,j]
  }
}
# приводимо матрицю A до діагонального виду (метод Жордана)
for(k in (n-1):1)
{
  for(i in (k+1):n)
    for(j in 1:n)
      E[k,j] = E[k,j] - A[k,i]*E[i,j]
}
return(E)
}

```

Результат обчислення оберненої матриці з використанням записаної процедури:

```

> A = cbind(c(5,6,10), c(1,18,20), c(2,6,40))
> A
      [,1] [,2] [,3]
[1,]    5    1    2
[2,]    6   18    6
[3,]   10   20   40
> A1 = MGaussJord(A,3)
> A1
           [,1]           [,2]           [,3]
[1,]  0.22222222 -3.469447e-18 -0.011111111
[2,] -0.06666667  6.666667e-02 -0.006666667
[3,] -0.02222222 -3.333333e-02  0.031111111
> E1 = A%*%A1
> E1
           [,1]           [,2]           [,3]
[1,] 1.000000e+00 0.000000e+00 -6.938894e-18
[2,] 0.000000e+00 1.000000e+00  2.775558e-17
[3,] 2.220446e-16 2.220446e-16  1.000000e+00

```

```

> A2 = solve(A)
> A2
           [,1]           [,2]           [,3]
[1,] 0.22222222 0.00000000 -0.01111111
[2,] -0.06666667 0.06666667 -0.00666667
[3,] -0.02222222 -0.03333333 0.03111111

```

Для перевірки отриманих результатів перемножені матриці  $A$  і  $A1$ . Добуток  $E1$  дорівнює одиничній матриці, що підтверджує правильність результату. Також обчислена обернена матриця  $A2$  з використанням процедури *solve* пакета  $R$ . Як видно, результати збігаються з точністю до помилки обчислень.

### 3.7. Висновки

1. Розв'язання систем лінійних алгебраїчних рівнянь є базовою процедурою в математичних пакетах чисельного аналізу, оскільки в багатьох чисельних методах вона використовується як допоміжна. Тому питанню оптимізації базової процедури приділяється особлива увага.

2. Якщо розв'язання системи лінійних алгебраїчних рівнянь є проміжною задачею при розв'язанні більш складної задачі, то найчастіше для цього застосовують метод LU-розкладання матриць.

3. Обчислення оберненої матриці ефективно проводити за методом Гауса – Жордана.

### 3.8. Контрольні запитання та завдання

1. Сформулюйте постановку задачі розв'язання системи лінійних алгебраїчних рівнянь.

2. Назвіть групи методів розв'язання систем лінійних алгебраїчних рівнянь, вкажіть їх принципів відмінності.

3. У чому полягає ідея методу Гауса?

4. У чому полягає суть методу виключення Гауса з вибором головного елемента? В яких випадках його застосовують на практиці?

5. Що називається LU-розкладанням матриці?

6. Як застосовується LU-розкладання матриць для розв'язання систем лінійних алгебраїчних рівнянь?

7. Як можна використати LU-розкладання матриці для обчислення її оберненої матриці?

8. Опишіть схему обчислення оберненої матриці методом Гауса – Жордана.

9. При яких умовах збігається метод простої ітерації? Наведіть оцінки швидкості збіжності цього методу.

10. Який порядок арифметичних обчислень потрібен для розв'язання системи лінійних алгебраїчних рівнянь з  $n$  невідомими?

11. Знайдіть коефіцієнти параболи  $y = ax^2 + bx + c$ , яка проходить через точки (1; 8.6), (3; 30.8), (5; 65).

12. Знайдіть обернену матрицю для матриці  $A$ :

$$A = \begin{pmatrix} 1.8 & -3.8 & 0.7 & -3.7 \\ 0.7 & 2.1 & -2.6 & -2.8 \\ 7.3 & 8.1 & 1.7 & -4.9 \\ 1.9 & -4.3 & -4.9 & -4.7 \end{pmatrix}.$$

13. Знайдіть точку перетину двох функцій:

$$y = 1 - 0.3x \text{ і } x = 3 - 2.2y.$$