

Лабораторна робота 4

Ітераційні методи вирішення систем лінійних рівнянь

Мета роботи: вивчення ітераційних методів вирішення систем лінійних рівнянь та їх реалізація в пакеті *Mathematica*.

Короткі теоретичні відомості

4.1. Матричний запис ітераційних методів

Ітераційні методи вирішення систем лінійних рівнянь поліпшують початково обраний вектор рішення без обробки матриці коефіцієнтів (тобто зведення її до трикутної форми чи розкладу на доданок двох трикутних матриць), в результаті чого зберігається її ненульова структура.

Оскільки ітерації при рішенні є однією з головних особливостей чисельних методів, розглянемо детальніше ітераційний метод вирішення системи лінійних рівнянь типу $Ax = b$.

Для її вирішення обирається деяке початкове наближення $x^{(0)}$, яке використовується для обчислення наступних наближень $x^{(k)}$ за деякою рекурентною формулою обчислень:

$$x^{(k+1)} = Mx^{(k)} + C^{(k)}, \quad (4.1)$$

де k — номер ітерації, $M, C^{(k)}$ — ітераційні оператори.

У формулі (4.1) передбачається при обчисленні вектора невідомих на $(k+1)$ ітерації $x^{(k+1)}$ використання значення вектора невідомих з попередньої ітерації $x^{(k)}$, тому її називають *однокроковою* чи *одношаровою*. Але існують і більш складні рекурентні формули наближень, в яких, наприклад, при обчисленні $x^{(k+1)}$ використовуються дві попередні ітерації $x^{(k)}$ і $x^{(k-1)}$, тоді їх називають *двокроковими* або *двошаровими*.

При ітераційному вирішенні звичайно задають деяку похибку наближення $\varepsilon > 0$ і припиняють обчислення, якщо виконується умова

$$\|x^{(k+1)} - \xi\| \leq \varepsilon \|x^{(0)} - \xi\|, \quad (4.2)$$

де ξ — точне вирішення системи лінійних рівнянь.

4.2. Методи простої ітерації і верхньої релаксації

Найпростіша рекурентна формула наближень типу (4.1) відповідає *методу простої ітерації*, яка будується на основі використання нев'язку системи лінійних рівнянь, які вирішуються:

$$r^{(k)} = Ax^{(k)} - b. \quad (4.3)$$

Для цього метода формула наближень має вид:

$$x^{(k+1)} = x^{(k)} - r^{(k)} = x^{(k)} - Ax^{(k)} + b = (E - A)x^{(k)} + b. \quad (4.4)$$

Порівнюючи формули (4.1) і (4.4), знаходимо, що для методу простої ітерації:

$$M = E - A; C^{(k)} = b.$$

Тобто в якості початкового наближення $x^{(0)}$ для цього методу можливий вибір вектора правої частини системи рівнянь b .

Розглянемо умову збіжності обчислень ітераційними методами за рекурентною формулою (4.1). Для цього запишемо вираз, який визначає глобальну похибку (4.2) на кожній ітерації:

$$x^{(k+1)} - \xi = M(x^{(k)} - \xi) = \dots = M^{k+1}(x^{(0)} - \xi),$$

де, як і раніше, ξ – точний розв’язок системи лінійних рівнянь.

Оскільки $\xi = M\xi + C^{(k)}$ і $x^{(k+1)} = Mx^{(k)} + C^{(k)}$, то глобальні похибки на сусідніх ітераціях пов’язані лінійною залежністю:

$$x^{(k+1)} - \xi = M(x^{(k)} - \xi),$$

або

$$\|x^{(k+1)} - \xi\| \leq \|M\| \cdot \|x^{(k)} - \xi\|. \quad (4.5)$$

Для того, щоб обчислення збігалися (або глобальна похибка зменшувалася), необхідно, щоб норма матриці M була меншою за одиницю:

$$\|M\| < 1. \quad (4.6)$$

При обчисленнях оцінку глобальної похибки $\|x^{(k+1)} - \xi\|$ зручно контролювати за значеннями *локальної* похибки, тобто за значеннями, отриманими на сусідніх ітераціях. Для цього перепишемо (4.5), додавши і віднявши справа значення $x^{(k+1)}$:

$$\|x^{(k+1)} - \xi\| \leq \|M\| \cdot \|x^{(k)} - \xi + x^{(k+1)} - x^{(k+1)}\| = \|M\| \cdot \|x^{(k+1)} - \xi\| + \|M\| \cdot \|x^{(k+1)} - x^{(k)}\|, \quad (4.7)$$

звідки отримуємо:

$$\|x^{(k+1)} - \xi\| \leq \frac{\|M\|}{1 - \|M\|} \|x^{(k+1)} - x^{(k)}\|, \quad (4.8)$$

або

$$\|x^{(k+1)} - \xi\| \leq \frac{\|M\|^{(k+1)}}{1 - \|M\|} \|x^{(1)} - x^{(0)}\|. \quad (4.9)$$

На межі стійкості норма матриці M дорівнює одиниці, $\|M\| = 1$. У цьому випадку контроль глобальної похибки неможливий і проводиться контроль локальної похибки за формулою:

$$\|x^{(k+1)} - x^{(k)}\| \leq \varepsilon, \quad (4.9a)$$

задане значення локальної похибки.

Приклад 4.1.

Вирішити методом простої ітерації систему рівнянь:

$$A = \{\{5, 3, 1\}, \{3, 7, 3\}, \{1, 3, 5\}\};$$

$$b = \{11, 17, 19\};$$

Згідно з формулою (4.4) оцінимо норму матриці ітераційних обчислень:

$$\text{ones} = \{\{1, 0, 0\}, \{0, 1, 0\}, \{0, 0, 1\}\};$$

$$M = A - \text{ones}$$

```
norma=Norm[M, Infinity]
```

```
{{4, 3, 1}, {3, 6, 3}, {1, 3, 4}}
```

```
12
```

Для того, щоб обчислення збігалися (глобальна похибка зменшувалася), необхідно, щоб норма матриці M була меншою за одиницю:

$$\|M\| < 1. \quad (4.10)$$

У даному випадку $\|M\| = 12$, тому умова (4.10) не виконується. Отже, вирішити систему рівнянь методом простої ітерації неможливо. Накладання умови збіжності (4.10) обмежує використання цього методу для рішення більшості практичних математичних задач. Область збіжності рішення можна розширити, якщо в формулі (4.4) ввести демпфуючий коефіцієнт $\omega \leq 1$ для врахування нев'язки:

$$x^{(k+1)} = x^{(k)} - \omega r^{(k)}. \quad (4.11)$$

Такий метод отримав назву *метода верхньої релаксації*. Коли демпфуючий коефіцієнт ω не залежить від номера ітерації, то ітераційний метод вважається *стаціонарним*.

Приклад 4.2.

Вирішити методом верхньої релаксації систему рівнянь з прикладу 4.1:

```
A={{5, 3, 1}, {3, 7, 3}, {1, 3, 5}};
```

```
b={11, 17, 19};
```

Рішення методом верхньої релаксації

$$x^{(k+1)} = x^{(k)} - \omega(Ax^{(k)} - b) = (E - \omega A)x^{(k)} + \omega b,$$

на відміну від методу простої ітерації потребує введення демпфуючого коефіцієнту $\omega \leq 1$. Можна показати, що умовою нерівності

$$\|E - \omega A\| < 1.$$

є умова $\omega\|A\| < 1$, тому вибираємо $\omega = 0.05 < \frac{1}{\|A\|}$.

```
A={{5, 3, 1}, {3, 7, 3}, {1, 3, 5}};
```

```
b={11, 17, 19};
```

```
ones={{1, 0, 0}, {0, 1, 0}, {0, 0, 1}};
```

```
A1=0.05*A;
```

```
MN=ones-A1;
```

```
M6=Norm[MN, Infinity]
```

```
S1=M6/(1-M6)
```

```
0.95
```

```
19
```

Оскільки норма $M6$ менша за одиницю, забезпечена збіжність методу верхньої релаксації.

Примітка: у випадках, коли норма матриці $\|E - \omega A\|$ дорівнює одиниці, $\|E - \omega A\| = 1$, множник S_1 в оцінці похибки має невизнане значення і тому втрачається зв'язок між глобальною і локальною похибками (4.8). Тоді оцінку точності розв'язку виконують за локальною похибкою, формула (4.9 а).

Задамо необхідну точність обчислень і початкове наближення:

```
g = 0.05*b;
ep = 0.00001; x0 = g;
x1 = MN . x0 + g

{0.7875, 1.1775, 1.5075}
```

Виконаємо другий крок згідно з формулою

```
x2 = MN . x1 + g

{0.888625, 1.27113, 1.86463}
```

Виконаємо необхідну кількість кроків для досягнення заданої похибки

```
ek = 10; k = 0; x = g;
While[ek > ep, y = N[MN.x + g]; ek = Max[Abs[y - x]*S1]; x = y; k = k + 1]
```

Виведемо на екран отриманий розв'язок і число ітерацій:

```
Print["x = ", x, ", ", ek = ", ek, ", ", k = ", k]

x = {1.25, 0.500004, 3.25} , ek = 9.34413x10-6 , k = 107
```

Існує ще один різновид умови збіжності обчислень ітераційними методами, який базується на використанні оцінки власних значень матриці перетворення M . Згідно з цією умовою збіжності необхідно і достатньо, щоб

$$\max_i |\lambda_i(M)| = |\lambda_{\max}| < 1, \quad (4.12)$$

тобто всі власні значення цієї матриці за модулем повинні бути меншими за одиницю.

Доказ умови (4.12) базується на використанні ортогонального координатного базису, який утворюють власні вектори матриці M , і розкладі вектора початкової глобальної похибки у цьому базисі:

$$x_0 - \xi = \alpha_1 U_1 + \alpha_2 U_2 + \dots + \alpha_n U_n.$$

Вектори глобальної похибки на наступних ітераціях згідно (4.5) обчислюються з допомогою операції множення на матрицю M , якій у вибраному координатному базисі відповідає процедура множення окремих проекцій вектора поточної глобальної похибки на відповідне власне значення матриці M :

$$\begin{aligned} x^{(1)} - \xi &= M(x^{(0)} - \xi) = \alpha_1 \lambda_1 U_1 + \alpha_2 \lambda_2 U_2 + \dots + \alpha_n \lambda_n U_n, \\ x^{(k+1)} - \xi &= M^{k+1}(x^{(0)} - \xi) = \alpha_1 \lambda_1^{k+1} U_1 + \dots + \alpha_n \lambda_n^{k+1} U_n. \end{aligned}$$

Якщо власні числа матриці M суттєво відрізняються, умова збіжності, що співпадає з умовою зменшення глобальної похибки, набуває необхідний вид

$$R(M) = \max_i |\lambda_i(M)| = \lambda_{\max} < 1$$

Формула (4.12) має лише теоретичну цінність, тому що задача пошуку власних значень матриці – більш складна задача, ніж розв'язування систем рівнянь.

4.3. Метод Якобі

Ітераційний метод Якобі вирішення лінійної системи рівнянь $Ax = b$ при умові $a_{ii} \neq 0$ передбачає рішення кожного рівняння системи окремо відносно тільки однієї змінної за припущення, що всі інші змінні задані або початковим наближенням, або даними попередньої ітерації:

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1, j \neq i}^n a_{ij} x_j^{(k)}}{a_{ii}}, \quad i = \overline{1, n}. \quad (4.13)$$

Для цього розділимо матрицю A на діагональну D , нижню L і верхню U трикутні матриці :

$$A = D(L + I + U), \quad (4.14)$$

де

$$D = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix}, \quad L = \begin{bmatrix} 0 & & & \\ a_{21}/a_{22} & 0 & & \\ \dots & \dots & \dots & \\ a_{n1}/a_{nn} & a_{n2}/a_{nn} & \dots & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & a_{12}/a_{11} & \dots & a_{1n}/a_{11} \\ & 0 & \dots & a_{2n}/a_{22} \\ & & \dots & \dots \\ & & & 0 \end{bmatrix}.$$

Тоді рівняння (4.1) для методу Якобі можна записати у вигляді:

$$x^{(k+1)} = -(L + U)x^{(k)} + D^{-1}b,$$

з якого легко доводиться, що матриця перетворення

$$M = -(L + U), \quad (4.15)$$

при цьому

$$m_{i,j} = \begin{cases} -a_{ij}/a_{ii}, & i \neq j, \\ 0, & i = j. \end{cases}$$

Визначення. Матриця A розмірності n має домінуючу головну діагональ, якщо кожний її діагональний елемент за модулем більший за суму модулів інших елементів цього ж рядка:

$$|a_{ii}| > \sum_{\substack{k=1 \\ k \neq i}}^n |a_{ik}|, i = \overline{1, n}.$$

Покажемо, що для збіжності методу Якобі достатньо, щоб матриця A мала домінуючу головну діагональ. Дійсно, враховуючи вираз (4.15), отримуємо:

$$\|M\|_{\infty} = \max_i \sum_{\substack{k=1 \\ k \neq i}}^n \left| \frac{a_{ik}}{a_{ii}} \right| < 1. \quad (4.16)$$

В математичних пакетах типу *Mathematica*, які не призначені для вирішення великих за розмірністю систем лінійних рівнянь, відсутні підпрограми, що реалізують ітераційні методи Якобі чи Гаусса–Зейделя. Але скориставшись можливостями пакету, можна їх запрограмувати, як показано в прикладі 4.3.

Приклад 4.3

Вирішити методом Якобі систему рівнянь з матрицею A и вектором правої частини b , яка розглядалась у прикладі 4.2:

$A = \{\{5, 3, 1\}, \{3, 7, 3\}, \{1, 3, 5\}\};$

$b = \{11, 17, 19\};$

`MatrixForm[A]`

$$\begin{pmatrix} 5 & 3 & 1 \\ 3 & 7 & 3 \\ 1 & 3 & 5 \end{pmatrix}$$

Приведемо систему до вигляду (4.14), для цього сформуємо діагональну матрицю DI :

`n=3; DI=Table [If[i==j,A[[i,j]], 0], {i,n},{j,n}]`

`MatrixForm[DI]`

$$\begin{pmatrix} 5 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$

Знайдемо матрицю $M = -DI^{-1}(A - DI)$ і вектор $g = D^{-1}b$.

`M = -Inverse[DI] . (A - DI)`

`{{0., -0.6, -0.2}, {-0.428571, 0., -0.428571}, {-0.2, -0.6, 0.}}`

`g = Inverse[DI] . b`

`{2.2, 2.42857, 3.8}`

Приведемо систему до вигляду $x^{(k+1)} = Mx^{(k)} + g$. Використаємо процедури *Mathematica*, які визначають норму матриці і норму вектора. Визначимо норму матриці M :

`NM = Norm[M, Infinity]`

0.857142

Оскільки норма матриці M менше одиниці, умова збіжності метода Якобі задовольняється. Задамо необхідну точність обчислень і початкове наближення:

$\epsilon_p = 0.00001$; $x_0 = g$;

$x_1 = M \cdot x_0 + g$

{-0.0171429, -0.142857, 1.90286}

Виконаємо другий крок згідно з формулою:

$x_2 = M \cdot x_1 + g$

{1.90514, 1.62041, 3.88914}

$S = NM / (1 - NM)$

5.99996

Виконаємо необхідну кількість кроків для досягнення заданої похибки

$\epsilon_k = 10$; $k = 0$; $x = g$;

While[$\epsilon_k > \epsilon_p$, $y = N[M \cdot x + g]$; $\epsilon_k = \text{Max}[\text{Abs}[y - x] \cdot S]$; $x = y$; $k = k + 1$]

Виведемо на екран отриманий розв'язок і число ітерацій:

Print["x = ", x, ", ", $\epsilon_k =$, ϵ_k , ", ", $k =$, k]

x = {1.25, 0.499999, 3.25} , $\epsilon_k = 8.43882 \times 10^{-6}$, k = 75

Отримаємо рішення системи рівнянь стандартним оператором *Mathematica*

$N[\text{LinearSolve}[A, b]]$

{1.25, 0.5, 3.25}

Як бачимо, результати збігаються у межах заданої похибки.

Примітка: у випадках, коли норма матриці ітерацій M дорівнює одиниці, що трапляється, коли сума її недіагональних елементів дорівнює діагональному елементу ($NM=1$), множник S в оцінці похибки має невизнане значення і тому втрачається зв'язок між глобальною і локальною похибками (4.8). Тоді оцінку точності розв'язку виконують за локальною похибкою, формула (4.9 а).

Приклад 4.4.

Вирішити систему рівнянь

$A = \{\{3, 2, 1\}, \{2, 5, 2\}, \{3, 5, 9\}\}$;

$b = \{0, -3, 1\}$;

Оцінимо норму матриці ітерацій M :

$n = 3$; $D1 = \text{Table}[\text{If}[i == j, A[[i, j]], 0], \{i, n\}, \{j, n\}]$;

$M = -\text{Inverse}[D1] \cdot (A - D1)$;

```
g = Inverse[D1] . b;
NM = Norm[M, Infinity]
```

1

Оскільки $NM=1$, змінюється подальша програма рішення, контроль переводиться з глобальної похибки на локальну (тобто виключається показник S).

```
ep=0.00001; ek = 10; k = 0; x = g;
While[ek > ep, y = N[M.x + g]; ek = Max[Abs[y - x]]; x = y; k = k+1]
Print["x = ",x,"", ek = ",ek,", k = ",k]
```

```
x = {0.500004, -0.999996, 0.500004}, ek = 9.47769 *10-6, k = 83
```

4.4 Метод Гаусса–Зейделя

Ітераційний метод Гаусса–Зейделя рішення лінійної системи рівнянь $Ax = b$ за умови $a_{ii} \neq 0$ передбачає рішення кожного рівняння системи окремо відносно однієї змінної, але при пошуку компоненти вектора i рішень наближення $(k + 1)$ відразу на поточній $(k + 1)$ ітерації використовуються вже знайдені компоненти $(k + 1)$ наближення з меншими номерами:

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^i a_{ij}x_j^{(k+1)} - \sum_{j=i}^n a_{ij}x_j^{(k)}}{a_{ii}}, \quad i = \overline{1, n}. \quad (4.17)$$

Метод Гаусса–Зейделя потребує менше пам'яті ЕОМ, ніж метод Якобі, тому що після обчислення компоненти i вектора $x^{(k+1)}$ відповідна компонента вектора $x^{(k)}$ стає непотрібною. Оскільки дані $(k + 1)$ наближення використовуються для знаходження самого $(k + 1)$ наближення, то метод Гаусса–Зейделя іноді називають *неявним* ітераційним методом в протилежність методу Якобі, який вважається *явним* ітераційним методом. Рівняння (4.1) для методу Гаусса–Зейделя можна записати у вигляді:

$$x^{(k+1)} = -Lx^{(k+1)} - Ux^{(k)} + D^{-1}b, \quad (4.18)$$

звідки матриця перетворення

$$M = -(I + L)^{-1}U. \quad (4.19)$$

Таким чином, щоб метод Гаусса–Зейделя збігався, необхідно і достатньо, щоб усі власні значення цієї матриці були за модулем меншими одиниці або норма матриці була меншою за одиницю (4.6). Щоб запобігти обернення матриці у виразі (4.19), ще раз перепишемо рекурентну формулу методу через нормовані значення коефіцієнтів (4.18):

$$x_i^{(k+1)} = \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(k+1)} + \sum_{j=i+1}^n \frac{a_{ij}}{a_{ii}} x_j^{(k)} + \frac{b_i}{a_{ii}}. \quad (4.20)$$

Переходячи у матрично–векторному виразі (4.18) до норм, можна отримати:

$$\|x_i^{(k+1)}\| \leq \|s_i\| \cdot \|x_i^{(k+1)}\| + \|r_i\| \cdot \|x_i^{(k)}\| + \|c_i\|, \quad (4.21)$$

$$\text{де } \|r_i\| = \sum_{k=1+1}^n \left| \frac{a_{ik}}{a_{ii}} \right|, \|s_i\| = \max_i \sum_{k=1}^{i-1} \left| \frac{a_{ik}}{a_{ii}} \right|, \|c_i\| = \left| \frac{b_i}{a_{ii}} \right|,$$

Таким чином, якщо матриця A симетрична, позитивно визначена і має домінуючу головну діагональ, то умова збіжності обчислень має вид:

$$\|M\|_{\infty} = \max_i \left| \frac{r_i}{1 - s_i} \right| < 1. \quad (4.22)$$

Можна визначити необхідну кількість ітерацій при виборі за початкове наближення нормованої правої частини $C_i = b_i/a_{ii}$ з метою одержання наближеного розв'язку лінійної системи із заданою точністю Δ , скористувавшись формулою (4.8):

$$\Delta = \|x^{(k+1)} - \xi\| \leq \frac{\|M\|^{(k+1)}}{1 - \|M\|} \|C\|, \quad (4.23)$$

звідки

$$(k + 1) \lg \|M\| \leq \lg \Delta - \lg \|C\| + \lg(1 - \|M\|)$$

або

$$k \geq \frac{1}{\lg \|M\|} [\lg \|C\| - \lg(\Delta(1 - \|M\|))] - 1. \quad (4.24)$$

Приклад 4.5

Повторимо рішення системи рівнянь з матрицею A і вектором правої частини b з прикладу 4.3 методом Гаусса–Зейделя:

```
A={{5,3,1},{3,7,3},{1,3,5}};
b={11,17,19};
MatrixForm[A]
```

$$\begin{pmatrix} 5 & 3 & 1 \\ 3 & 7 & 3 \\ 1 & 3 & 5 \end{pmatrix}$$

Обчислення невідомих x^{k+1} виконується за формулою (4.15) послідовно за кожною компонентою x_j^{k+1} , $j = 1..n$, тому що нові значення, отримані на $k + 1$ ітерації, повинні використовуватися у наступних обчисленнях. Знайдемо матриці D , L і U векторної формули (4.18).

```
A={{5,3,1},{3,7,3},{1,3,5}};
b={11,17,19};
n = Length[A];
D1=Table [If[i==j,A[[i,j]], 0], {i,n},{j,n}];
MatrixForm[D1]
```

$$\begin{pmatrix} 5 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$

```
g = Inverse[D1] . b;
```

```
L = Table[If[i > j, l[i,j]=A[[i,j]]/A[[i,i]], l[i,j]=0], {i,n}, {j,n}];
```

```
MatrixForm[L]
```

$$\begin{pmatrix} 0 & 0 & 0 \\ \frac{3}{7} & 0 & 0 \\ \frac{1}{5} & \frac{3}{5} & 0 \end{pmatrix}$$

```
U = Table[If[i<j, u[i,j]=A[[i,j]]/A[[i,i]], u[i,j] = 0], {i,n}, {j,n}];
```

```
MatrixForm[U]
```

$$\begin{pmatrix} 0 & \frac{3}{5} & \frac{1}{5} \\ 0 & 0 & \frac{3}{7} \\ 0 & 0 & 0 \end{pmatrix}$$

Тепер можна знайти матрицю М (М_Г) і перевірити умову збіжності методу Гаусса–Зейделя:

```
N[MJ = -Inverse[(IdentityMatrix[n] + L)] . U]
```

```
{{0., -0.6, -0.2}, {0., 0.257143, -0.342857}, {0., -0.0342857, 0.245714}}
```

```
NMJ = Norm[MJ, Infinity]
```

```
4/5
```

```
S = NMJ/(1 - NMJ)
```

```
4
```

```
N[g1= -Inverse[IdentityMatrix[n]+L].g]
```

```
{-2.2, -1.48571, -2.46857}
```

Оскільки норма М_Г менша за одиницю, забезпечена збіжність методу Гаусса–Зейделя. Програма вирішення системи рівнянь:

```
ek = 10; ep = 0.00001; k = 0; x = g1;
```

```
While[ek>ep, y = N[MJ . x + g1]; ek = Max[Abs[y - x]*S]; x = y; k = k+1]
```

```
Print["x = ", x, ", ", ek = ", ek, ", ", k = ", k]
```

```
x = {-1.25, -0.5, -3.25} , ek = 4.1329*10-6 , k = 15
```

Таким чином, рішення з приблизно такою ж похибкою отримано за 15 ітерацій замість 75-ти при використанні метода Якобі. У розглянутих прикладах методи методами Якобі і Гаусса- Зейделя використовувалися для вирішення систем лінійних рівнянь, матриці яких мали домінуючу діагональ, тобто

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|.$$

Проте єдине рішення системи лінійних рівнянь існує і тоді, коли немає діагональної домінуючості, але матриця A є невиродженою, тобто $\det A \neq 0$. За допомогою елементарних перетворень можна з невиродженої матриці побудувати матрицю з діагональною домінуючістю, для якої ітераційні методи збігаються, як це було зроблено в методі верхньої релаксації. Проте існує більш простий спосіб, а саме нормалізувати систему лінійних рівнянь з невиродженою матрицею і застосувати до неї метод Гаусса- Зейделя.

Система лінійних рівнянь називається нормальною, якщо її матриця симетрична і позитивно визначена. Позитивно визначена матриця – це матриця, у якій всі власні значення позитивні. З будь-якої невиродженої матриці можна зробити нормальну, помножуючи її на транспоновану зліва. Помножимо $Ax = b$ зліва на, отримаємо

$$A^T Ax = A^T b \text{ або } Bx = b_1, \text{ де } B = A^T A, b_1 = A^T b. \quad (4.25)$$

Тому методи Якобі і Гаусса-Зейделя еквівалентні для нормальної системи лінійних рівнянь.

Приклад 4.6

Вирішити систему рівнянь

$$\begin{cases} 3.2x_1 + 8.9x_2 + 0.79x_3 = 6.1; \\ 3.5x_1 + 1.7x_2 + 2.9x_3 = 2.3; \\ 4.1x_1 + 5.7x_2 - 1.7x_3 = 0.6. \end{cases}$$

Введемо матрицю і вектор правих частин системи:

$A = \{\{3.2, 8.9, 0.79\}, \{3.5, 1.7, 2.9\}, \{4.1, 5.7, -1.7\}\};$
 $b = \{6.1, 2.3, 0.6\};$

Перевіримо невиродженість системи:

$\text{Det}[A]$

106.886

Нормалізуємо систему і вирішуємо її методом Якобі:

$A1 = \text{Transpose}[A] . A$

$\text{MatrixForm}[A1]$

$$\begin{pmatrix} 39.3 & 57.8 & 5.708 \\ 57.8 & 114.59 & 2.271 \\ 5.708 & 2.271 & 11.9241 \end{pmatrix}$$

$b1 = \text{Transpose}[A] . b;$

```
n=3;D1=Table[If[i==j,A1[[i,j]],0},{i,n},{j,n}];
M=-Inverse[D1].(A1-D1)
MatrixForm[M]
```

$$\begin{pmatrix} 0. & -1.47074 & -0.145242 \\ -0.504407 & 0. & -0.0198185 \\ -0.478694 & -0.190455 & 0. \end{pmatrix}$$

```
g=Inverse[D1].b1;
ep=0.00001;ek=10;k=0;x=g;
While[ek>ep,y=N[M.x+g];ek=Max[Abs[y-x]];x=y;k=k+1]
Print["x = ",x," , ek = ",ek," , k = ",k]
```

```
x = {-0.548484,0.794804,0.989158} , ek = 9.81054×10-6 , k = 136
```

Рішення із заданою точністю отримане за 136 ітерацій, але в формулі рішення вилучено множник S в виразі для оцінки похибки, як то було в прикладі 4.4. Приведемо нев'язку рішення

```
r= A.x-b
```

```
{0.0000408341,0.0000297194,0.0000293532}
```

Цікаво відмітити, що без процедури нормування задана система рівнянь методом Якобі не вирішується. Робиться лише перший прогін і далі похибка рішення не зменшується.

Примітка. Інколи вдається перетворити матрицю коефіцієнтів рівнянь в домінанту шляхом заміни окремих рівнянь системи композицією інших рівнянь. Наприклад, для розглянутого прикладу з недомінантною матрицею

```
A={{3.2,8.9,0.79},{3.5,1.7,2.9},{4.1,5.7,-1.7}}
```

при відніманні першого рівняння з другого, а другого з третього отримаємо домінантну матрицю

```
A2={{3.2,8.9,0.79},{0.3,-7.2,2.11},{0.6,4,-4.6}}.
```

Якщо відповідні перетворення виконати і з складовими вектора правих частин $b=\{6.1,2.3,0.6\}$, то отримаємо $b2=\{6.1,-3.8,-1.7\}$. Можна впевнитися, що рішення побудованої системи $A_2x_2=b_2$ дорівнює $x_2=\{-0.548489,0.794801,0.989155\}$.

Порядок виконання роботи

1. Обрати варіант завдання згідно з номером варіанту.
2. Вирішити задану систему рівнянь методами простої ітерації і верхньої релаксації.
3. Скласти програми ітераційного розв'язку системи рівнянь методами Якобі і Гауса-Зейделя, що реалізують співвідношення (4.13) і (4.17). При

використанні методів Якобі і Гауса-Зейделя можуть знадобитися еквівалентні перетворення системи рівнянь згідно з формулами (4.25).

4. Скласти звіт, що складається з програмного коду, отриманих результатів і математичних формул використаних методів для кожного пункту завдання, у висновках дати оцінку порівняльної точності отриманих рішень різними методами і кількості виконаних ітерацій.

Варіанти завдань

№1	A			b
	3	2	1	0
	2	5	2	-3
	3	5	9	1
№2	3	2	1	4.2
	1	9	4	17.8
	1	8	27	8.8
№3	1.42	0.43	0.27	-0.054
	0.43	-1.43	0.93	-1.022
	-0.27	0.43	0.93	-0.062
№4	0.64	0.40	0.24	0.784
	0.50	1.48	-0.93	1.755
	0.37	-0.43	1.24	-0.428
№5	1.77	0.5	0.39	0.689
	0.84	1.79	0.95	-2.152
	0.21	-0.81	1.03	1.992
№6	0.86	0.51	0.19	0.35
	0.51	0.95	0.32	0.42
	0.83	0.12	-0.95	0.45
№7	1.54	0.64	-0.33	0.3
	-0.92	1.54	0.24	0.2
	-0.33	0.24	0.78	0.1
№8	1.77	0.55	0.39	1.5
	0.84	1.79	0.95	2.5
	0.24	-0.41	1.03	3
№9	1.77	0.39	1.38	1.5

0.82	1.79	0.94	2.5
-0.21	1.02	1.24	3

№10	1.42	1.23	0.19	0.1
	-0.84	1.43	0.53	0.2
	0.27	-0.48	0.93	0.3

№11	-0.93	-0.08	0.11	1.18
	0.18	-0.48	0	0.21
	0.13	0.31	-1	0.21

№12	-1	-0.07	0.21	0.92
	0.03	1	-0.42	0.92
	-0.03	-0.04	1	1.2

№13	-1	-0.07	0.21	0.92
	0.07	-0.42	0.03	0.92
	0.21	-0.04	0.42	1.2

№14	0.08	-0.12	-0.77	0.32
	0.25	0.41	0.14	-1
	-0.77	-0.14	0.06	-0.12

№15	0.08	0.25	-0.77	0.32
	0.25	0.50	0.14	-1
	-0.77	0.14	0.06	-0.12

№16	1	1	3	0.55
	2	9	4	0.35
	15	9	5	0.55

№17	1	2	4	0.55
	1	16	4	1.35
	64	8	14	3.55

№18	0.86	0.53	-0.33	3
	0.53	-0.92	0.23	2
	-0.33	0.23	0.78	1

№19	6	3	3	11
	2	6	3	15

	1	3	5	21
№20	11	12	24	13
	12	28	15	4
	29	15	13	17
№21	1	2	4	0.55
	2	9	7	1.35
	14	6	3	3.55
№22	1.64	0.53	-0.33	3
	0.53	-0.92	0.23	2
	-0.33	0.23	1.78	1
№23	25	13	11	21
	13	25	11	27
	11	13	25	29
№24	0.08	0.25	-1.77	0.32
	0.25	0.42	0.14	-1
	-1.77	0.14	0.06	-0.12
№25	11	21	33	55
	21	91	41	176
	24	11	12	40