



Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Фізико-технічний інститут

## Комп'ютерний практикум №3

### Розв'язання систем лінійних алгебраїчних рівнянь ітераційними методами

предмет «Методи обчислень»

**Роботу виконав:**

Студент 3 курсу ФТІ, групи ФІ-91  
Цибульник Антон Владиславович

**Приймала:**

Стьопочкіна Ірина Валеріївна

## Завдання

Якщо матриця не є матрицею із діагональною перевагою, привести систему до еквівалентної, у якій є діагональна перевага (письмово). Написати програму розв'язання за ітераційним методом, який відповідає заданому варіантові (метод простої ітерації для парних варіантів та метод Зейделя для непарних варіантів).

Для методу простої ітерації навести письмовий етап приведення матриці до діагональної переваги, лістинг програми та результати її дії (на кожній ітерації вивести номер ітерації, наближене значення розв'язку, значення критерію).

Для методу Зейделя навести письмовий етап приведення системи до діагональної переваги та до виду  $A^T A x = A^T b$ . Виконати розрахунки для цих двох варіантів вхідних матриць. Порівняти швидкість збіжності. Вміст звіту такий самий, як і для методу простої ітерації.

Обчислення проводити з  $\varepsilon = 10^{-4}$ . Для кожної ітерації розраховувати вектор нев'язки  $r = |b - Ax|$ , де  $x$  – отриманий розв'язок.

## Варіант завдання

Розглядається така система рівнянь:

$$\begin{cases} 5,5x_1 + 7x_2 + 6x_3 + 5,5x_4 = 23 \\ 7x_1 + 10,5x_2 + 8x_3 + 7x_4 = 32 \\ 6x_1 + 8x_2 + 10,5x_3 + 9x_4 = 33 \\ 5,5x_1 + 7x_2 + 9x_3 + 10,5x_4 = 31 \end{cases} \quad (1)$$

З цієї системи запишемо матрицю коефіцієнтів  $A$  та матрицю вільних членів  $B$ :

$$A = \begin{pmatrix} 5.5 & 7 & 6 & 5.5 \\ 7 & 10.5 & 8 & 7 \\ 6 & 8 & 10.5 & 9 \\ 5.5 & 7 & 9 & 10.5 \end{pmatrix}, \quad B = \begin{pmatrix} 23 \\ 32 \\ 33 \\ 31 \end{pmatrix}$$

Розв'яжемо систему рівнянь (1) за допомогою методу простої ітерації.

## Теоретичні відомості

Суть методу простої ітерації полягає у тому, щоб звести початкове матричне рівняння  $Ax = B$  до вигляду

$$x_{k+1} = Cx_k + D, \quad (2)$$

де для матриці  $C$  мусить виконуватися умова

$$\forall i \quad \sum_{j=1}^n |C_{ij}| \leq q < 1 \quad (3)$$

Для виконання умови (3) матриця  $A$  має бути матрицею із діагональною перевагою, тобто такою, для якої справедлива нерівність

$$\forall i, j \quad |a_{ii}| > \sum_{i \neq j} |a_{ij}| \quad (4)$$

Отже, для справноного застосування ітераційного процесу (2) обов'язковою умовою є діагональний вид матриці  $A$ . Тоді, знайшовши елементи матриць  $C$  та  $D$  за формулами

$$c_{ij} = \begin{cases} 0, & i = j \\ -\frac{a_{ij}}{a_{ii}}, & i \neq j \end{cases}$$

$$d_i = \frac{b_i}{a_{ii}}$$

й задавши довільний вектор  $x_0$  (наприклад,  $x_0 = (0, 0, 0, 0)$ ), можна розпочинати пошук розв'язку із заданою точністю  $\varepsilon$  доти, доки

$$\frac{q}{1-q} \max_j |x_{k+1} - x_k| < \varepsilon \quad (5)$$

## Зведення матриці до вигляду із діагональною перевагою

Крок за кроком зведемо початкові матриці

$$A = \begin{pmatrix} 5.5 & 7 & 6 & 5.5 \\ 7 & 10.5 & 8 & 7 \\ 6 & 8 & 10.5 & 9 \\ 5.5 & 7 & 9 & 10.5 \end{pmatrix}, \quad B = \begin{pmatrix} 23 \\ 32 \\ 33 \\ 31 \end{pmatrix} \quad (6)$$

до виду із діагональною перевагою. Для зручності введемо ряд позначок:

- незмінений рядок,
- щойно змінений рядок,
- ⊗ рядок, зведений до вигляду діагональної переваги.

Наведений нижче покроковий алгоритм не є уніфікованим, узагальненим. Тобто ці кроки застосовні лише до конкретного прикладу.

Крок 1:  $A, B \xrightarrow{\text{IV-I}} A', B'$

$$A' = \begin{pmatrix} 5.5 & 7 & 6 & 5.5 & \circ \\ 7 & 10.5 & 8 & 7 & \circ \\ 6 & 8 & 10.5 & 9 & \circ \\ 0 & 0 & 3 & 5 & \bullet \end{pmatrix}, \quad B' = \begin{pmatrix} 23 & \circ \\ 32 & \circ \\ 33 & \circ \\ 8 & \bullet \end{pmatrix}$$

Крок 2:  $A', B' \xrightarrow{\Pi - \frac{7}{5.5} \cdot I} A'', B''$

$$A'' = \begin{pmatrix} 5.5 & 7 & 6 & 5.5 & \circ \\ 0 & 35/22 & 4/11 & 0 & \bullet \\ 6 & 8 & 10.5 & 9 & \circ \\ 0 & 0 & 3 & 5 & \circ \end{pmatrix}, B'' = \begin{pmatrix} 23 & \circ \\ 30/11 & \bullet \\ 33 & \circ \\ 8 & \circ \end{pmatrix}$$

Крок 3:  $A'', B'' \xrightarrow{I - IV - 4 \cdot II} A^*, B^*$

$$A^* = \begin{pmatrix} 5.5 & 7/11 & 17/11 & 0.5 & \bullet \\ 0 & 35/22 & 4/11 & 0 & \circ \\ 6 & 8 & 10.5 & 9 & \circ \\ 0 & 0 & 3 & 5 & \circ \end{pmatrix}, B^* = \begin{pmatrix} 45/11 & \bullet \\ 30/11 & \circ \\ 33 & \circ \\ 8 & \circ \end{pmatrix}$$

Крок 4: по чергово обнулимо елементи  $a_{31}, a_{32}$  та  $a_{34}$ :

(а)  $A^*, B^* \xrightarrow{III - \frac{6}{5.5} \cdot I} A^{**}, B^{**}$

$$A^{**} = \begin{pmatrix} 5.5 & 7/11 & 17/11 & 0.5 & \circ \\ 0 & 35/22 & 4/11 & 0 & \circ \\ 0 & 884/121 & 2133/242 & 93/11 & \bullet \\ 0 & 0 & 3 & 5 & \circ \end{pmatrix}, B^{**} = \begin{pmatrix} 45/11 & \circ \\ 30/11 & \circ \\ 3453/121 & \bullet \\ 8 & \circ \end{pmatrix}$$

(б)  $A^{**}, B^{**} \xrightarrow{III - \frac{22}{35} \cdot \frac{884}{121} \cdot II} A^*, B^*$

$$A^* = \begin{pmatrix} 5.5 & 7/11 & 17/11 & 0.5 & \circ \\ 0 & 35/22 & 4/11 & 0 & \circ \\ 0 & 0 & 5501/770 & 93/11 & \bullet \\ 0 & 0 & 3 & 5 & \circ \end{pmatrix}, B^* = \begin{pmatrix} 45/11 & \circ \\ 30/11 & \circ \\ 1233/77 & \bullet \\ 8 & \circ \end{pmatrix}$$

(в)  $A^*, B^* \xrightarrow{III - \frac{1}{5} \cdot \frac{93}{11} \cdot IV} A^{**}, B^{**}$

$$A^{**} = \begin{pmatrix} 5.5 & 7/11 & 17/11 & 0.5 & \circ \\ 0 & 35/22 & 4/11 & 0 & \circ \\ 0 & 0 & 29/14 & 0 & \bullet \\ 0 & 0 & 3 & 5 & \circ \end{pmatrix}, B^{**} = \begin{pmatrix} 45/11 & \circ \\ 30/11 & \circ \\ 87/35 & \bullet \\ 8 & \circ \end{pmatrix}$$

Таким чином зведені до діагонального виду матриці виглядатимуть так:

$$A^\diamond = \begin{pmatrix} 5.5 & 7/11 & 17/11 & 0.5 \\ 0 & 35/22 & 4/11 & 0 \\ 0 & 0 & 29/14 & 0 \\ 0 & 0 & 3 & 5 \end{pmatrix}, B^\diamond = \begin{pmatrix} 45/11 \\ 30/11 \\ 87/35 \\ 8 \end{pmatrix} \quad (7)$$

# Программный код

```
1 import numpy as np
2
3 a = np.array([[5.5, 7.0, 6.0, 5.5],
4               [7.0, 10.5, 8.0, 7.0],
5               [6.0, 8.0, 10.5, 9.0],
6               [5.5, 7.0, 9.0, 10.5]])
7
8 length = a[0].size
9
10 a_diagonal = np.array([[5.5, 7/11, 17/11, 0.5],
11                        [0.0, 35/22, 4/11, 0.0],
12                        [0.0, 0.0, 29/14, 0.0],
13                        [0.0, 0.0, 3.0, 5.0]])
14
15 b = np.array([23.0, 32.0, 33.0, 31.0])
16
17 b_diagonal = np.array([45/11, 30/11, 87/35, 8])
18
19 c = np.array([[0.0, 0.0, 0.0, 0.0],
20               [0.0, 0.0, 0.0, 0.0],
21               [0.0, 0.0, 0.0, 0.0],
22               [0.0, 0.0, 0.0, 0.0]])
23
24 d = np.array([0.0, 0.0, 0.0, 0.0])
25
26 for i in range(length):
27     for j in range(length):
28         if (i == j): c[i][j] = 0.0
29         else: c[i][j] = -a_diagonal[i][j]/a_diagonal[i][i]
30
31 for i in range(length):
32     d[i] = b_diagonal[i]/a_diagonal[i][i]
33
34 q = 0
35 for i in range(length):
36     sum = 0
37     for j in range(length):
38         sum = sum + abs(c[i][j])
39     if (sum > q): q = sum
40
41 e = 0.0001
42
43 x = []
44 x.append([0.0, 0.0, 0.0, 0.0])
45
46 stop = False
47 i = 0
48
49 while (stop != True):
50     x.append(np.dot(c,x[i]) + d)
51     max = 0
```

```

52     for j in range(length):
53         subtraction = abs(x[i+1][j] - x[i][j])
54         if (subtraction > max): max = subtraction
55
56     i = i + 1
57     if (q*max/(1-q) < e): stop = True
58
59 r = abs(b - np.dot(a,x[len(x)-1]))

```

## Результати й проміжні кроки

Ітерація 1:  $x = [0.74380165, 1.71428571, 1.2, 1.6]$   
 $\text{criterion} = 2.571428571428571$

Ітерація 2:  $x = [0.06280992, 1.44, 1.2, 0.88]$   
 $\text{criterion} = 1.0799999999999998$

Ітерація 3:  $x = [0.16, 1.44, 1.2, 0.88]$   
 $\text{criterion} = 0.1457851239669422$

Ітерація 4:  $x = [0.16, 1.44, 1.2, 0.88]$   
 $\text{criterion} = 0.0$

Крім того, зазначимо проміжні матриці  $C$  та  $D$ :

$$C = \begin{pmatrix} 0 & -0.11570248 & -0.28099174 & -0.09090909 \\ 0 & 0 & -0.22857143 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -0.6 & 0 \end{pmatrix}, \quad D = \begin{pmatrix} 0.74380165 \\ 1.71428571 \\ 1.2 \\ 1.6 \end{pmatrix}$$

І наостанок був обрахований вектор нев'язки  $r = |B^\diamond - A^\diamond x|$ , який вказує похибку обчислень (коротше кажучи, це перевірка правильності розв'язку):

$$r = (0, 0, 0, 0)$$

Судячи з вектора нев'язки  $r$ , отриманий розв'язок  $x = (0.16, 1.44, 1.2, 0.88)$  і є розв'язком початкової системи рівнянь  $Ax = B$ , зображеної у виразі (1). Отже, остаточний розв'язок:

$$x_1 = 0.16, \quad x_2 = 1.44, \quad x_3 = 1.2, \quad x_4 = 0.88$$

## Контрольні запитання

1. В чому полягає основна відмінність прямих та ітераційних методів розв'язання СЛАР?

Основна відмінність полягає у припущенні, що навіть якщо в ітераційних методах (зокрема у методі простої ітерації та у методі Зейделя) обчислення ведуться без округлення, то це дозволяє отримати розв'язок заданої системи лише із заданою точністю.

2. Який метод буде збігатись швидше при однакових вихідних даних – метод Зейделя чи метод простої ітерації?

Ітераційні методи можуть повільно збігатися. До того ж випадки поганої збіжності складно ідентифікуються. На противагу прямим методам, в ітераційних кількість операцій заздалегідь невідома. Припускаю, що за однакових вихідних даних неможливо визначити напевне, який з методів буде збігатися швидше.

3. Чи необхідний для методу Зейделя етап приведення до діагональної переваги? Чому?

Для методу Зейделя етап приведення до діагональної переваги не є обов'язковим (проте іноді все ж бажаний). Вся справа у тому, що зазначений метод справедливий для довільної симетричної додатньо визначеної матриці. Саме тому часто достатньо буває віднайти саме таку матрицю, а не зводити наявну до вигляду діагональної переваги.

4. Виведіть оцінку точності для метода простої ітерації.

Згідно теореми, яка доводилася на лекції

$$\|C\| \leq \sum_{j=1}^n |C_{ij}| \leq q < 1 \quad (8)$$

Застосовуючи вираз (8) покажемо, що  $|x_{k+1} - x_k| \leq q^k |x_1 - x_0|$ :

Нехай	$\begin{cases} \delta_k = x_k - x_{k-1} \\ \delta_{k+1} = x_{k+1} - x_k \end{cases} \xrightarrow{\text{підставимо в}} x_{k+1} = Cx_k + D$
Отримаємо	$\delta_{k+1} + x_k = C\delta_k + Cx_{k-1} + D$
З (2) маємо	$\delta_{k+1} + \cancel{x_k} = C\delta_k + \cancel{Cx_{k-1}} + D \implies \delta_{k+1} = C\delta_k$
Тоді	$ \delta_{k+1}  =  C\delta_k  \leq \ C\   \delta_k  \leq \ C\ ^2  \delta_{k-1}  \leq \dots \leq \ C\ ^k  \delta_1 $
Таким чином	$ \delta_{k+1}  \leq \ C\ ^k  \delta_1  \xrightarrow{(8)}  x_{k+1} - x_k  \leq q^k  x_1 - x_0 $

Занотуємо цей результат:

$$|x_{k+1} - x_k| \leq q^k |x_1 - x_0| \quad (9)$$

Зазначимо, що  $x^* = \lim_{k \rightarrow \infty} x^k$ , тоді можемо дещо видозмінити таку різницю:

$$\begin{aligned} |x^* - x_{k+1}| &= |x_{k+2} - x_{k+1} + x_{k+3} - x_{k+2} + x_{k+4} - x_{k+3} + \dots| \leq \\ &\leq |x_{k+2} - x_{k+1}| + |x_{k+3} - x_{k+2}| + |x_{k+4} - x_{k+3}| + \dots \leq \\ &\leq_{(9)} q^{k+1}|x_1 - x_0| + q^{k+2}|x_1 - x_0| + q^{k+3}|x_1 - x_0| + \dots \end{aligned}$$

Маємо нескінченну геометричну прогресію з кроком  $q < 1$  й початковим членом  $q^{k+1}$ , а тому:

$$|x^* - x_{k+1}| \leq \frac{q^{k+1}}{1 - q} |x_1 - x_0| \quad (10)$$

Перепозначивши деякі значення  $x_0 = x^k$ ,  $x_1 = x^{k+1}$  й скориставшись нерівністю  $q^{k+1} \leq q$  й обмеженням точності  $|x^* - x_{k+1}| < \varepsilon$ , отримаємо остаточний результат:

$$|x^* - x_{k+1}| \leq \frac{q}{1 - q} |x_{k+1} - x_k| < \varepsilon \quad (11)$$

Таким чином наведений нижче вираз

$$\frac{q}{1 - q} \max_j |x_{k+1} - x_k| < \varepsilon$$

і є оцінкою точності для методу простої ітерації.