

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування
імені адмірала Макарова

О. М. ДУДЧЕНКО, С. О. КАРПОВА

**ЧИСЕЛЬНІ МЕТОДИ
В ІНЖЕНЕРНИХ РОЗРАХУНКАХ.
РОЗВ'ЯЗУВАННЯ СИСТЕМ
ЛІНІЙНИХ РІВНЯНЬ ЗА ДОПОМОГОЮ ПК**

Методичні вказівки
до виконання лабораторних робіт

Рекомендовано Методичною радою НУК



2026

УДК 519.6(076)

Д81

Автори: О. М. Дудченко, канд. техн. наук, професор НУК;
С. О. Карпова, старший викладач

Рецензенти: П. Й. Гучек, д-р техн. наук, доцент;
М. Б. Літвінова, д-р пед. наук, канд. фіз.-мат. наук, професор

Рекомендовано Методичною радою НУК імені адмірала Макарова

Дудченко О. М.

Д81 Чисельні методи в інженерних розрахунках. Розв'язування систем лінійних рівнянь за допомогою ПК : методичні вказівки до виконання лабораторних робіт / О. М. Дудченко, С. О. Карпова. – Миколаїв : НУК, 2026. – 32 с.

У вказівках наведені короткі відомості з теорії, тексти програм для розв'язання систем лінійних рівнянь, написаних на мові програмування C++ з використанням компілятора Microsoft Visual Studio та на мові Fortran з використанням компіляторів Compaq Visual Fortran та Fortran Power Station (у вільному та фіксованому форматах), а також для online-компіляторів. Наведені варіанти завдань для виконання лабораторних робіт.

Призначені для студентів технічних закладів вищої освіти, які вивчають курси «Чисельні методи», «Обчислювальна техніка, програмування та математичне моделювання», або споріднені курси, пов'язані з чисельними методами.

УДК 519.6(076)

© О. М. Дудченко, С. О. Карпова, 2026
© Національний університет кораблебудування
імені адмірала Макарова, 2026

ЗМІСТ

Вступ.....	4
Розв’язування систем лінійних алгебраїчних рівнянь.....	5
Обумовленість систем рівнянь.....	5
Прямі методи розв’язування систем лінійних рівнянь.....	7
Ітераційні методи розв’язування систем лінійних рівнянь.....	10
Тексти програм на мові FORTRAN.....	18
Тексти програм на мові C++.....	23
Послідовність виконання лабораторних робіт.....	27
Список літератури.....	30

ВСТУП

Сучасний розвиток науки та техніки тісно пов'язаний з використанням обчислювальної техніки. Комп'ютери давно вже стали робочим інструментом для вченого, інженера, конструктора та студента. Вони використовуються для дослідження реальних складних об'єктів, процесів та явищ, при створенні систем управління цими об'єктами і процесами. І тут на перший план виступає їх математична модель, тобто опис об'єктів, явищ і процесів на математичній мові за допомогою функцій, рівнянь (диференціальних, інтегральних тощо), нерівностей, логічних відношень. Для таких задач розроблені алгоритми та методи рішень, існує цілий ряд програм для їх рішення на ЕОМ. Однак постійний розвиток обчислювальної техніки та застосування нових мов програмування в інженерній практиці потребує вдосконалення діючих і створення нових програм. Особливо актуальна ця задача для студентів, які в період воєнного стану навчаються в режимі online і використовують для виконання лабораторних робіт online-компілятори.

У даних методичних вказівках наведені тексти програм для розв'язання систем лінійних рівнянь. Ця задача є однією з найважливіших і найбільш поширених задач обчислювальної математики. Програми написані на найбільш поширених в інженерній практиці мовах програмування FORTRAN та C++. Стил написання програм обрано таким чином, щоб користувач ПК, який володіє однією з вказаних мов програмування, зміг без особливих зусиль зрозуміти тексти програм, написані іншою мовою. В методичних вказівках наведені також варіанти завдань та вказівки до виконання лабораторних робіт.

Щоб система (1) мала єдиний розв'язок, усі n рівнянь, що входять до неї, повинні бути лінійно незалежними. Необхідною та достатньою умовою для цього є умова нерівності нулю визначника даної системи, тобто $\Delta \neq 0$.

Якщо $\Delta = 0$, а серед визначників $\Delta x_1, \Delta x_2, \dots, \Delta x_n$ є не рівні нулю, то система не має розв'язків.

Нехай $\Delta = \Delta x_1 = \Delta x_2 = \dots = \Delta x_n = 0$, причому один з мінорів $(n-1)$ -го порядку визначника не дорівнює нулю. (Нагадаємо, що мінором якогось елемента називається визначник, отриманий з даного визначника викреслюванням того рядка і того стовпчика, на перехресті яких стоїть елемент). Тоді система зводиться до $n-1$ рівняння; одне з рівнянь, що відповідає номеру викресленого рядка, є наслідком інших. Одному з невідомих, що відповідає номеру викресленого стовпчика, можна дати довільне значення. Решта $n-1$ невідомих визначаються єдиним чином із системи $n-1$ рівнянь.

Якщо всі визначники $(n-1)$ -го порядку, що є мінорами визначника Δ , дорівнюють нулю, то система може не мати рішень, а може зводитись до $n-2$ рівняння або до меншої їх кількості.

Алгоритми розв'язування систем лінійних рівнянь поділяються на прямі та ітераційні.

Прямі методи розв'язування систем лінійних рівнянь

Найбільш поширені прямі методи основані на зведенні систем рівнянь до «трикутного» вигляду. При цьому одне з рівнянь містить лише одну невідому, а в кожному наступному додається ще по одній невідомій. Однією з найбільш вдалих схем є схема, розроблена Гаусом [8; 9; 10].

За схемою Гауса потрібно виконати лише $N_2 = n / 3(n^2 + 3n - 1)$ множення та ділення, тобто кількість витраченої праці, грубо кажучи, буде зростати як n^3 .

При $n = 10$ кількість операцій буде дорівнювати лише

$$N_2 = 10 / 3 (100 + 30 - 1) = 430,$$

тобто майже у 8400 разів менше, ніж N_1 .

Застосовуючи вказаний метод, спочатку нормують перше рівняння, ділячи його коефіцієнти на a_{11} . Потім перше рівняння множать на перші коефіцієнти $a_{i,1}$ усіх других рівнянь і послідовно віднімають від решти рівнянь. У результаті перша змінна буде виключена з усіх рівнянь, крім першого. На наступному етапі розв'язання така сама процедура застосовується до решти $n-1$ рівнянь. В результаті з $n-2$ рівнянь, що залишилися, вилучається друга невідома. Уся процедура повторюється доти, поки після n кроків уся система не буде зведена до трикутного вигляду [6; 7; 9]. Математично цю процедуру можна описати таким чином.

На k -му кроці процесу вилучення нормовані коефіцієнти кожного k -го рівняння мають вигляд:

$$b_{k,j} = \frac{a_{k,j}}{a_{k,k}},$$

а нові коефіцієнти в наступних рівняннях будуть мати такий вигляд:

$$b_{i,j} = a_{i,j} - a_{i,k} b_{k,j}, \quad i > k.$$

При цьому слід пам'ятати, що коефіцієнти нижче розташованих рівнянь a_{ij} змінюються місцями на кожному кроці. Наприклад, коефіцієнти b_{ij} на наступному кроці перетворюються на коефіцієнти a_{ij} .

Приклад. Нехай треба розв'язати методом Гауса таку систему рівнянь:

$$\begin{cases} x_1 + x_2 + x_3 = 2 \\ x_1 - x_2 - x_3 = 0 \\ 2x_1 + x_2 - x_3 = 9 \end{cases}.$$

Розв'язок. Позначимо рівняння буквами та выпишемо коефіцієнти при невідомих і вільні члени. Тоді вихідна система набуде такого вигляду:

$$\begin{array}{l} A_1 \quad 1 \quad 1 \quad 1 \quad 2 \\ A_1 \quad 1 \quad -1 \quad -1 \quad 0 \\ A_1 \quad 2 \quad 1 \quad -1 \quad 9 \end{array}$$

Вилучимо члени, які містять x_1 :

$$\begin{array}{l} B_1 = A_1 / 1 \quad 1 \quad 1 \quad 1 \quad 2 \\ B_2 = A_2 - B_1 \quad 0 \quad -2 \quad -2 \quad -2 \\ B_3 = A_3 - 2B_1 \quad 0 \quad -1 \quad -3 \quad 5 \end{array}$$

Вилучимо члени, які містять x_2 :

$$\begin{array}{l} B_1 \quad 1 \quad 1 \quad 1 \quad 2 \\ C_2 = B_2 / (-2) \quad 0 \quad 1 \quad 1 \quad 1 \\ C_3 = B_3 + C_2 \quad 0 \quad 0 \quad -2 \quad 6 \end{array}$$

Повертаючись до форми рівнянь, одержимо:

$$\begin{cases} x_1 + x_2 + x_3 = 2 \\ x_2 + x_3 = 1 \\ -2x_3 = 6 \end{cases}.$$

Розв'язок системи легко знаходиться підстановкою значення x_3 в друге рівняння, потім значень x_2 та x_3 у перше рівняння. Розв'язання системи таке: $x_1 = 1$, $x_2 = 4$, $x_3 = -3$.

При великих n схема Гауса перестає бути ефективною. В результаті великої кількості математичних операцій виникає накопичування помилок округлення, що в кінцевому підсумку призводить до неправильних результатів.

Серед методів виключення, що мають широке розповсюдження, слід відзначити додатково метод Гауса–Жордана [3; 9] і метод Холецького, відомий також як метод Кроута [5].

Як сказано вище, звичайно прямі методи достатньо ефективні, однак у випадку великих матриць ($n > 500$) вони поступаються ітераційним. Їх використання із застосуванням ЕОМ призводить до труднощів, пов'язаних зі зберіганням матриці коефіцієнтів у пам'яті ЕОМ і з втратою точності результатів.

У методі Якобі початкові значення змінних використовуються для розрахунку нових значень $x_1, x_2, \dots, x_n$ за допомогою наведених вище рівнянь (2). Процес завершується, коли всі нові значення виявляються достатньо близькими до початкових. У протилежному випадку нові значення використовуються замість початкових. Цей процес повторюється доки не буде досягнута збіжність або не стане зрозумілим, що процес розбіжний. У цьому методі заміна значень усіх змінних здійснюється водночас.

В методі Гауса–Зейделя уточнені значення x_1 відразу ж використовуються для обчислювання x_2 . Потім за новими значеннями x_1 і x_2 обчислюється x_3 і т. п. Це невелике вдосконалення ітераційної процедури дозволяє суттєво збільшити швидкість збіжності.

Якщо система містить багато нульових коефіцієнтів a_{ij} , то цей метод має перевагу. Метод Гауса–Зейделя, як і всі ітераційні методи, використовує існування нулів, у той час як методи виключення – ні.

У методі верхньої релаксації нові значення кожної змінної представляються у такому вигляді:

$$x_i^{(n+1)} = x_i^{(n)} + w(\bar{x}_i^{(n+1)} - x_i^{(n)}),$$

де $\bar{x}_i^{n+1}$ – уточнене значення змінної за методом Гауса–Зейделя, а w – параметр релаксації, замкнутий в інтервалі значень $1 \leq w \leq 2$. При $w = 1$ цей метод тотожний методу Гауса–Зейделя. Швидкість збіжності залежить від обраної величини w .

Зауваження. Ітераційні методи збігаються, якщо рівняння розташовуються таким чином, що більші коефіцієнти знаходяться на головній діагоналі. Якщо ці члени головної діагоналі достатньо сильно перевищують інші коефіцієнти рівняння, то збіжність гарантована, в протилежному випадку – ні.

Існує ще одна достатня умова:

$$|a_{ii}| > |a_{i1}| + |a_{i2}| + \dots + |a_{i,j-1}| + |a_{i,j+1}| + \dots + |a_{in}| \quad (3)$$

для всіх $i, i = 1 \dots n$.

У тому випадку, коли умова (3) не виконується, а кількість невідомих достатньо велика, розв'язувати систему можна за допомогою блочних методів. Найбільш відомим серед блочних методів є метод Зейделя.

Блочні методи, по суті, є комбінацією прямих та ітераційних методів. У початковій системі (1) виділяються блоки, розміри яких залежать від можливостей комп'ютера. Наприклад, у системі з 1000 невідомими можна виділити два блоки з 500 невідомими, або два блоки з 400 невідомими та один блок з 200 невідомими тощо. Тобто замість системи з 1000 невідомими ми отримуємо дві системи з 500 невідомими, або дві системи з 400 невідомими та одну з 200.

Виділення блоків розглянемо на прикладі системи з 6-ма невідомими.

У системі з 6-ма невідомими є три варіанти виділення блоків. Наприклад, можна виділити три блоки, в кожному з яких буде 2 невідомих, або два блоки, в одному з яких буде 2 невідомих, а в іншому 4, або два блоки, в кожному з яких буде 3 невідомих.

Останній варіант виділення двох блоків виконується таким чином:

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + a_{15}x_5 + a_{16}x_6 = c_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 + a_{25}x_5 + a_{26}x_6 = c_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 + a_{35}x_5 + a_{36}x_6 = c_3, \\ a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 + a_{45}x_5 + a_{46}x_6 = c_4, \\ a_{51}x_1 + a_{52}x_2 + a_{53}x_3 + a_{54}x_4 + a_{55}x_5 + a_{56}x_6 = c_5, \\ a_{61}x_1 + a_{62}x_2 + a_{63}x_3 + a_{64}x_4 + a_{65}x_5 + a_{66}x_6 = c_6. \end{array} \right. \quad (4)$$

Далі система (4) може бути приведена до вигляду (5):

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = c_1 - (a_{14}x_4 + a_{15}x_5 + a_{16}x_6), \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = c_2 - (a_{24}x_4 + a_{25}x_5 + a_{26}x_6), \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = c_3 - (a_{34}x_4 + a_{35}x_5 + a_{36}x_6), \\ a_{44}x_4 + a_{45}x_5 + a_{46}x_6 = c_4 - (a_{41}x_1 + a_{42}x_2 + a_{43}x_3), \\ a_{54}x_4 + a_{55}x_5 + a_{56}x_6 = c_5 - (a_{51}x_1 + a_{52}x_2 + a_{53}x_3), \\ a_{64}x_4 + a_{65}x_5 + a_{66}x_6 = c_6 - (a_{61}x_1 + a_{62}x_2 + a_{63}x_3). \end{cases} \quad (5)$$

У свою чергу, з (5) ми отримуємо дві системи (6) та (7):

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = c_1 - (a_{14}x_4 + a_{15}x_5 + a_{16}x_6), \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = c_2 - (a_{24}x_4 + a_{25}x_5 + a_{26}x_6), \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = c_3 - (a_{34}x_4 + a_{35}x_5 + a_{36}x_6). \end{cases} \quad (6)$$

$$\begin{cases} a_{44}x_4 + a_{45}x_5 + a_{46}x_6 = c_4 - (a_{41}x_1 + a_{42}x_2 + a_{43}x_3), \\ a_{54}x_4 + a_{55}x_5 + a_{56}x_6 = c_5 - (a_{51}x_1 + a_{52}x_2 + a_{53}x_3), \\ a_{64}x_4 + a_{65}x_5 + a_{66}x_6 = c_6 - (a_{61}x_1 + a_{62}x_2 + a_{63}x_3). \end{cases} \quad (7)$$

Спочатку невідомим присвоюються початкові значення так само, як і в точечних методах. Виділений блок залишається у лівій частині, а все інше переноситься до правої частини. Утворені системи розв'язуються за допомогою прямих методів (наприклад, за допомогою метода Гауса). Знайдені невідомі порівнюються з початковими. Якщо різниця між усіма невідомими задовольняє заданій точності, то процес закінчується, якщо ні, то за початкові дані беруть значення знайдених, і процес продовжується.

Подібно до методів Якобі та Гауса–Зейделя, заміна всіх змінних може здійснюватися водночас після розв'язування всіх систем рівнянь, або послідовно після розв'язування кожної системи.

Головним недоліком блочних методів є їх громіздкість та значне збільшення потрібного часу на розв'язування систем у порівнянні з іншими методами.

Зауваження щодо вибору початкових значень невідомих. Зазвичай початкові значення невідомих вибирають, виходячи з фізики задачі, що розв'язується. У випадку абстрактної системи ці значення можуть бути будь-якими, наприклад, дорівнювати одиниці або нулю.

Приклад. Дано систему лінійних алгебраїчних рівнянь:

$$\begin{cases} 4x_1 + 0,24x_2 - 0,08x_3 = 8, \\ 0,09x_1 + 3x_2 - 0,15x_3 = 9, \\ 0,04x_1 - 0,08x_2 + 4x_3 = 20. \end{cases}$$

Розв'язати систему методом Якобі та методом Гауса–Зейделя, якщо задана точність становить $\varepsilon = 10^{-4}$ [2].

Розв'язок. Виберемо як початкове наближення $x_{10} = x_{20} = x_{30} = 1$. Розрахунок наведемо у вигляді таблиці 1.

З наведеної таблиці видно, що збіжність метода Гауса–Зейделя настала на одну ітерацію скоріше, ніж у методі Якобі.

Кількість ітерацій може змінюватися, якщо змінюється задана точність розрахунку.

Таблиця 1

Метод Якобі	Метод Гауса–Зейделя
$x_{11} = \frac{8 - 0,24 + 0,08}{4} = 1,9600$ $x_{21} = \frac{9 - 0,09 + 0,15}{3} = 3,0200$ $x_{31} = \frac{20 - 0,04 + 0,08}{4} = 5,0100$	$x_{11} = \frac{8 - 0,24 + 0,08}{4} = 1,9600$ $x_{21} = \frac{9 - 0,09 * 1,96 + 0,15}{3} = 2,9912$ $x_{31} = \frac{20 - 0,04 * 1,96 + 0,08 * 2,9912}{4} = 5,0400$
$x_{12} = \frac{8 - 0,24 * 3,02 + 0,08 * 5,01}{4} = 1,9190$ $x_{22} = \frac{9 - 0,09 * 1,96 + 0,15 * 5,01}{3} = 3,1917$ $x_{32} = \frac{20 - 0,04 * 1,96 + 0,08 * 3,02}{4} = 5,0408$	$x_{12} = \frac{8 - 0,24 * 2,9912 + 0,08 * 5,04}{4} = 1,9213$ $x_{22} = \frac{9 - 0,09 * 1,9213 + 0,15 * 5,04}{3} = 3,1943$ $x_{32} = \frac{20 - 0,04 * 1,9213 + 0,08 * 3,1943}{4} = 5,0447$
$x_{13} = \frac{8 - 0,24 * 3,1917 + 0,08 * 5,0408}{4} = 1,9093$ $x_{23} = \frac{9 - 0,09 * 1,919 + 0,15 * 5,0408}{3} = 3,19447$ $x_{33} = \frac{20 - 0,04 * 1,919 + 0,08 * 3,1917}{4} = 5,0446$	$x_{13} = \frac{8 - 0,24 * 3,1943 + 0,08 * 5,0447}{4} = 1,9092$ $x_{23} = \frac{9 - 0,09 * 1,9092 + 0,15 * 5,0447}{3} = 3,1949$ $x_{33} = \frac{20 - 0,04 * 1,9092 + 0,08 * 3,1949}{4} = 5,0448$
$x_{14} = \frac{8 - 0,24 * 3,19447 + 0,08 * 5,0446}{4} = 1,9092$ $x_{24} = \frac{9 - 0,09 * 1,9093 + 0,15 * 5,0446}{3} = 3,1949$ $x_{34} = \frac{20 - 0,04 * 1,9093 + 0,08 * 3,19447}{4} = 5,0447$	$x_{14} = \frac{8 - 0,24 * 3,1949 + 0,08 * 5,0448}{4} = 1,9091$ $x_{24} = \frac{9 - 0,09 * 1,9091 + 0,15 * 5,0448}{3} = 3,1949$ $x_{34} = \frac{20 - 0,04 * 1,9091 + 0,08 * 3,1949}{4} = 5,0448$
$x_{15} = \frac{8 - 0,24 * 3,1949 + 0,08 * 5,0447}{4} = 1,9091$ $x_{25} = \frac{9 - 0,09 * 1,9092 + 0,15 * 5,0447}{3} = 3,1949$ $x_{35} = \frac{20 - 0,04 * 1,9092 + 0,08 * 3,1949}{4} = 5,0448$	

Зауваження. При дослідженнях збіжності систем краще порівнювати із заданою точністю ε не абсолютні, а відносні різниці сусідніх величин, розглядаючи нерівність

$$\max \left| \frac{x_i^{(k)} - x_i^{(k-1)}}{x_i^k} \right| < \varepsilon.$$

Нижче наведені тексти програм, в яких реалізовано алгоритми розв’язування систем лінійних рівнянь за методами Якобі та Гауса–Зейделя. Програми з реалізованим алгоритмом Гауса достатньо поширені, їх можна знайти у стандартному програмному забезпеченні або в спеціальній технічній літературі, наприклад, [1; 4; 5; 10].

У програмах використовуються динамічні масиви для зберігання матриць коефіцієнтів, стовпців вільних членів та невідомих двох наближень, тому кількість невідомих може бути довільною та обмежується лише технічними характеристиками комп’ютера.

Програми, написані на Фортрані, можуть бути використані у середовищі MS Developer Studio.

Результати роботи програм виводяться на екран монітора і записуються у стандартний файл fort.7, з якого вони можуть бути переглянуті та роздруковані. При використанні інших компіляторів Фортрану, ім’я стандартного файлу з результатами може бути іншим.

Сучасний Фортран має можливості створювати файли у фіксованому та вільному форматах (Fortran Fixed Format Source File та Fortran Free Format Source File відповідно). Тому серед наведених програм є варіанти програм, написаних для фіксованого та вільного форматів. При використанні online-компіляторів можна використовувати спрощені варіанти програм. Як приклад наведений текст програми для розв’язування системи з трьома невідомими методом Гауса–Зейделя.

У програмах, написаних на мові C++, використовуються стандартні функції обробки даних, за допомогою яких результати можуть бути переглянуті та роздруковані.

У наведених програмах введення даних здійснюється у діалоговому режимі в такій послідовності: кількість невідомих, матриця коефіцієнтів, стовпець вільних членів. Результатами розрахунків є корені системи. Для контролю правильності результатів, а також порівняння ефективності методів виводиться лічильник ітерацій.

Тексти програм на мові FORTRAN Фіксований формат

Метод Якобі

PROGRAM RSPI

```
* Розв'язування СЛАР методом Якобі
REAL, ALLOCATABLE::Q(:,,:),W(:,),X(:,,:)
PRINT *, 'Input the number of unknowns'
READ(*,*)N
ALLOCATE (Q(N,N),W(N),X(N,2))
PRINT *, 'Input the matrix coefficients row by row'
READ(*,*)((Q(I,J),J=1,N),I=1,N)
PRINT *, 'Input the right part of the system'
READ(*,*)(W(I),I=1,N)
      DO LK=6,7
      WRITE(LK,3)
3    FORMAT(10X,' Jacobi method'/
*12X,'Output data'/10X,'Matrix coefficients')
      DO I=1,N
      WRITE(LK,4)I,(Q(I,J),J=1,N)
4    FORMAT(14X,'Line N',I3/(8F10.3))
      ENDDO
      WRITE(LK,5)(W(I),I=1,N)
5    FORMAT(10X,'Right part of the system'/(8F10.3))
      ENDDO
      DO 1 I=1,N
1    X(I,1)=1
      K=0
2    PR=0
      DO 15 I=1,N
      S=0
      DO 7 J=1,N
      IF(I.EQ.J)GO TO 7
      S=S+Q(I,J)*X(J,1)
7    CONTINUE
      X(I,2)=(W(I)-S)/Q(I,I)
      IF(ABS((X(I,2)-X(I,1))/X(I,2)).LE.0.00001)GO TO 15
      PR=1
```

```

15  CONTINUE
    DO I=1,N
      X(I,1)=X(I,2)
    ENDDO
    K=K+1
    IF(K.GT.30)GO TO 8
    IF(PR.EQ.1)GO TO 2
8    DO LK=6,7
      WRITE(LK,10)
10     FORMAT(15X,'Calculation results')
      WRITE(LK,18)(I,X(I,2),I=1,N)
18     FORMAT(5X,(10(' X',I1,'=',F10.5,:)))
      WRITE(LK,20)K
20     FORMAT(10X,'Number of iterations K=',I3)
    ENDDO
    DEALLOCATE(Q,W,X)
  END

```

Метод Гауса–Зейделя

```

PROGRAM RSGZ
* Розв'язування СЛАР методом Гаусса-Зейделя
  REAL, ALLOCATABLE::Q(:,:),W(:),X(:,:)
  PRINT *, 'Input the number of unknowns'
  READ(*,*)N
  ALLOCATE (Q(N,N),W(N),X(N,2))
  PRINT *, ' Input the matrix coefficients row by row'
  READ(*,*)((Q(I,J),J=1,N),I=1,N)
  PRINT *, 'Enter the right part of the system'
  READ(*,*)(W(I),I=1,N)
  DO LK=6,7
    WRITE(LK,3)
3    FORMAT(10X,'Gauss-Seidel method'/
*12X,'Output data'/10X,'Matrix coefficients')
    DO I=1,N
      WRITE(LK,4)I,(Q(I,J),J=1,N)
4      FORMAT(14X,'Line N',I3/(8F10.3))
    ENDDO
    WRITE(LK,5)(W(I),I=1,N)

```

```

5  FORMAT(10X,'Right part of the system'/(8F10.3))
   ENDDO
   DO 1 I=1,N
1   X(I,1)=1
   K=0
2   PR=0
   DO 15 I=1,N
   S=0
   DO 7 J=1,N
   IF(I.EQ.J)GO TO 7
   S=S+Q(I,J)*X(J,1)
7   CONTINUE
   X(I,2)=(W(I)-S)/Q(I,I)
   IF(ABS((X(I,2)-X(I,1))/X(I,2)).LE.0.00001)GO TO 12
   PR=1
12  X(I,1)=X(I,2)
15  CONTINUE
   K=K+1
   IF(K.GT.30)GO TO 8
   IF(PR.EQ.1)GO TO 2
8   DO LK=6,7
   WRITE(LK,10)
10  FORMAT(15X,'Calculation results')
   WRITE(LK,18)(I,X(I,2),I=1,N)
18  FORMAT(5X,(10(' X',I1,'=',F10.5,:)))
   WRITE(LK,20)K
20  FORMAT(10X,'Number of iterations K=',I3)
   ENDDO
   DEALLOCATE(Q,W,X)
   END

```

Вільний формат

Метод Гауса–Зейделя

```
PROGRAM RSGZ
! Розв'язування СЛАП методом Гаусса-Зейделя
REAL, ALLOCATABLE::Q(:,:),W(:,:),X(:,:)
PRINT *, 'Input the number of unknowns'
READ(*,*)N
ALLOCATE (Q(N,N),W(N),X(N,2))
PRINT *, 'Input the matrix coefficients row by row'
READ(*,*)((Q(I,J),J=1,N),I=1,N)
PRINT *, 'Input the right part of the system'
READ(*,*)(W(I),I=1,N)
DO LK=6,7
WRITE(LK,3)
3  FORMAT(10X,'Gauss-Seidel method'/ &
12X,'Output data'/10X,'Matrix coefficients')
DO I=1,N
WRITE(LK,4)I,(Q(I,J),J=1,N)
4  FORMAT(14X,'Line N',I3/(8F10.3))
ENDDO
WRITE(LK,5)(W(I),I=1,N)
5  FORMAT(10X,'Right part of the system'/(8F10.3))
ENDDO
DO 1 I=1,N
1  X(I,1)=1
K=0
2  PR=0
DO 15 I=1,N
S=0
DO 7 J=1,N
IF(I=J)S=S+Q(I,J)*X(J,1)
7  CONTINUE
X(I,2)=(W(I)-S)/Q(I,I)
IF(ABS((X(I,2)-X(I,1))/X(I,2))<=0.00001)GO TO 12
PR=1
12  X(I,1)=X(I,2)
15  CONTINUE
K=K+1; IF(K.GT.30)GO TO 8; IF(PR.EQ.1)GO TO 2
```

```

8   DO LK=6,7
   WRITE(LK,10)
10   FORMAT(15X,'Calculation results')
   WRITE(LK,18)(I,X(I,2),I=1,N)
18   FORMAT(5X,(10(' X',I1,'=',F10.5,:)))
      WRITE(LK,20)K
20   FORMAT(10X,'Number of iterations K=',I3)
      ENDDO
      DEALLOCATE(Q,W,X)
      END

```

Спрощений варіант програми для метода Гауса–Зейделя

```

      program p5
! Розв'язування СЛАР методом Гаусса-Зейделя
      Dimension A(3,3),B(3),X(3,2)
      print*,' Input the matrix coefficients row by row'
      read(5,*)((A(i,j),j=1,3),i=1,3)
      print*,'Enter the right part of the system'
      read(5,*)(B(i),i=1,3)
      do i=1,3
        X(i,1)=1
      enddo
      k=0
5     pr=0
      do 20 i=1,3
        S=0
        do 10 j=1,3
          if (i/=j)S=S+A(i,j)*X(j,1)
10      continue
        x(i,2)=(B(i)-S)/A(i,i)
        if (abs((x(i,2)-x(i,1))/x(i,2))<=0.00001) goto 15
        pr=1
15     x(i,1)=x(i,2)
20     continue
        k=k+1
        if (pr==1.and.k<=50)goto 5
        print 40,(x(i,1),i=1,3),k
40    format ('Calculation results'/x(i)=' ',3E12.4,' k=',i2)
      end

```

Тексти програм на мові C++

Метод Якобі

```
// Метод Якобі
#include "stdafx.h"
#include <iostream>
using namespace std;
#define M(x,y) *(m+n*x+y)
#define X(c,b) *(xp+n*c+b)
#define FORI for(i=0;i<n;i++)
#define FORJ for(j=0;j<n;j++)
float *m,*pr,*xp;
int n,x,y,d,c,b;

int main()
{
    int iv,k,i,j;
    float prr,s;
    cout<<" Jacobi method "<<endl;
    cout<<" Input the number of unknowns "; cin>>n;
    cout<<" Input the matrix coefficients row by row "<<endl;
    float *m=(float*)calloc(n*n,sizeof(float));

    FORI { cout<<"          Line N"<<i+1<<endl;
        FORJ {cin>>M(i,j);}
    }
    float *pr=(float*)calloc(n,sizeof(float));
    cout<<" Input the right part of the system"<<endl;
    FORI { cin>>pr[i]; }
    printf("\n          Output data \n      Matrix coefficients");
    FORJ {
        FORI { if((i%10)==0)printf("\n");
            printf("%f ",M(j,i)); }
    }
    printf("\n      Right part of the system\n");
    FORI {printf("%f ",pr[i]); }
    float *xp=(float*)calloc(2*n,sizeof(float));
    FORI { X(0,i)=1;}
    k=0;
    m2:prr=0;
```

```

FORI { s=0;
        FORJ { if(i!=j)s=s+M(i,j)*X(0,j); }
                X(1,i)=(pr[i]-s)/M(i,i);
                }
FORI { if(fabs((X(1,i)-X(0,i))/X(1,i))<=0.00001)goto m1;
        X(0,i)=X(1,i); prr=1;
        m1:;
    }
k++; if(k>30)goto m8;
    if(prr==1) goto m2;
m8: cout<<endl;
cout<<"\n Calculation results"<<endl;
    iv=n+1; if(n>10) iv=11;
FORI {if((i%iv)==0)printf("\n");cout<<"X"<<i+1<<"="<<X(0,i)<<" ";}
        cout<<endl;
        cout<<" Number of iterations " <<k<<endl;
free(m); free(pr); free(xp);
system ("pause");
return 0;
}

```

Метод Гауса–Зейделя

```

// Метод Гаусса–Зейделя
#include "stdafx.h"
#include <iostream>
using namespace std;
#define M(x,y) *(m+n*x+y)
#define X(c,b) *(xp+n*c+b)
#define FORI for(i=0;i<n;i++)
#define FORJ for(j=0;j<n;j++)
float *m,*pr,*xp;
int n,x,y,d,c,b;
    int main()
{
    int iv,k,i,j;
    float prr,s;
    cout<<" Gauss-Seidel method "<<endl;
    cout<<" Input the number of unknowns "; cin>>n;
}

```



```

    cout<< " Input the matrix coefficients row by row "<<endl;
    float *m=(float*)calloc(n*n,sizeof(float));

FORI { cout<<"          Line N"<<i+1<<endl;
        FORJ {cin>>M(i,j);}
        }
        float *pr=(float*)calloc(n,sizeof(float));
    cout<<" Input the right part of the system"<<endl;
FORI { cin>>pr[i]; }
    printf("\n          Output data \n      Matrix coefficients");
FORJ {
    FORI { if((i%10)==0)printf("\n");
            printf("%f ",M(j,i)); }
        }
    printf("\n      Right part of the system\n");
    FORI {printf("%f ",pr[i]); }
    float *xp=(float*)calloc(2*n,sizeof(float));
FORI { X(0,i)=1;}
    k=0;
    m2:pr=0;
FORI { s=0;
        FORJ { if(i!=j)s=s+M(i,j)*X(0,j); }
                X(1,i)=(pr[i]-s)/M(i,i);
                if(fabs((X(1,i)-X(0,i))/X(1,i))<=0.00001)goto m1;
                X(0,i)=X(1,i); pr=1;
                m1:;
            }
    k++; if(k>30)goto m8;
        if(pr==1) goto m2;
m8: cout<<endl;
    cout<<" \n Calculation results"<<endl;
    iv=n+1; if(n>10) iv=11;
FORI {if((i%iv)==0)printf("\n");cout<<"X"<<i+1<<"="<<X(0,i)<<" ";}
        cout<<endl;
        cout<<" Number of iterations  "<<k<<endl;
    free(m); free(pr); free(xp);
    system ("pause");
    return 0;
}

```

Нижче надані результати розрахунків з використанням наведених програм, виконані за методами Якобі та Гауса–Зейделя.

Метод Якобі	Метод Гауса–Зейделя
Jacobi method	Gauss-Seidel method
Output data	Output data
Matrix coefficients	Matrix coefficients
Line N 1	Line N 1
4.000 0.240 -0.080	4.000 0.240 -0.080
Line N 2	Line N 2
0.090 3.000 -0.150	0.090 3.000 -0.150
Line N 3	Line N 3
0.040 -0.080 4.000	0.040 -0.080 4.000
Right part of the system	Right part of the system
8.000 9.000 20.000	8.000 9.000 20.000
Calculation results	Calculation results
X1= 1.90920 X2= 3.19496 X3= 5.04481	X1= 1.90920 X2= 3.19496 X3= 5.04481
Number of iterations K= 6	Number of iterations K= 5

Результати відповідають наведеним вище контрольним розрахункам, які виконані без використання комп'ютера.

Послідовність виконання лабораторних робіт

1. З таблиці 2 вибрати варіант, що відповідає № за списком журналу відвідування групи.
2. За наведеними зразками скласти програму та зробити розрахунки за завданням викладача.
3. Виконати перевірку отриманих результатів.
4. Скласти звіт про виконану роботу, зробити висновки про застосування та ефективність методів.

Таблиця 2

№ варіанта	Матриця коефіцієнтів			Стовпець вільних членів
1	2			3
1	1,840	0,225	0,253	-6,090
	0,232	2,600	0,282	-6,980
	0,183	0,206	2,240	-5,520
2	2,580	0,293	0,313	-6,660
	0,132	1,550	0,158	-3,580
	0,209	0,225	2,340	-5,010
3	2,180	0,244	0,249	-4,340
	2,170	2,310	0,236	-3,915
	0,315	0,322	3,170	-5,278
4	1,542	0,178	0,162	-1,973
	0,369	3,745	0,358	-3,748
	0,245	0,243	2,246	-2,245
5	1,547	0,163	0,143	-5,135
	0,235	2,342	0,207	-3,798
	0,384	0,373	3,456	-5,962
6	2,364	0,237	0,213	1,484
	0,251	2,456	0,214	1,925
	0,259	0,245	2,063	2,168
7	3,435	0,338	0,396	0,552
	0,417	4,000	0,365	0,693
	0,435	0,432	3,672	7,296

Продовж. табл. 2

1	2			3
8	3,887	0,378	0,345	9,457
	0,321	2,794	0,239	8,346
	0,267	0,237	1,968	7,620
9	3,435	0,326	0,291	9,056
	0,264	2,393	0,196	8,045
	0,464	0,432	3,848	7,836
10	2,533	0,236	0,193	2,678
	0,395	4,121	0,368	9,973
	0,278	0,243	1,945	3,935
11	2,167	0,196	0,156	3,168
	0,355	3,235	0,278	7,178
	0,485	0,447	3,978	9,753
12	2,689	0,247	0,237	9,377
	0,273	2,398	0,192	9,436
	0,293	0,253	2,025	5,572
13	3,745	0,347	0,306	8,768
	0,447	4,193	0,367	6,857
	0,496	0,457	4,012	7,352
14	4,356	0,439	0,367	3,126
	0,404	3,658	0,317	6,848
	0,314	0,693	2,176	8,934
15	4,075	0,379	0,337	6,778
	0,284	2,448	0,195	7,893
	0,499	0,452	3,978	9,375
16	3,193	0,289	0,247	3,935
	0,443	4,124	0,353	7,212
	0,347	0,292	2,456	2,948
17	2,576	0,226	0,184	8,667
	0,447	4,035	0,357	5,279
	0,489	0,442	3,878	5,035
18	2,833	0,256	0,208	3,287
	0,300	2,055	0,207	4,596
	0,372	0,321	2,689	3,448

Продовж. табл. 2

1	2			3
19	3,782	0,344	0,302	6,821
	0,433	3,878	0,339	3,457
	0,406	0,434	3,710	8,756
20	4,598	0,453	0,382	9,542
	0,483	4,376	0,388	2,333
	0,406	0,353	3,012	5,116
21	4,567	0,421	0,378	3,843
	0,321	2,732	0,225	2,398
	0,458	0,404	3,525	6,672
22	3,757	0,339	0,297	3,387
	0,418	3,708	0,322	4,548
	0,445	0,388	4,785	6,262
23	2,976	0,258	0,216	4,161
	0,511	4,625	0,414	6,684
	0,438	0,382	3,306	5,345
24	2,935	0,255	0,214	6,432
	0,347	2,982	0,256	4,748
	0,478	0,422	3,748	5,826
25	3,748	0,336	0,294	3,267
	0,402	3,573	0,304	7,548
	0,418	0,361	3,095	4,567

СПИСОК ЛІТЕРАТУРИ

1. Економіко-математичні методи та моделі : навчальний посібник / Н. Л. Воропай, Т. В. Герасименко, Л. О. Кирилова, Л. М. Корсун, М. В. Мацкул, Є. В. Мальцева, А. В. Михайленко, Є. В. Орлов, В. Г. Чернишев, О. Є. Чепурна, В. М. Шинкаренко (за заг. ред. В. М. Мацкул). – Одеса : ОНЕУ, 2018. – 404 с.
2. Дудченко О. М. Чисельні методи в інженерних розрахунках : у 3 ч. Частина 3. Розв'язання систем лінійних алгебраїчних рівнянь за допомогою ПК : методичні вказівки. – Миколаїв : УДМТУ, 2003. – 27 с.
3. Литвинов А. Л. Чисельні методи: теорія і практика : навч. посіб. / А. Л. Литвинов; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2022. – 166 с.
4. Маликов В. Т., Кветний Р. Н. Вычислительные методы и применение ЭВМ. – Киев : Вища школа, 1989. – 213 с.
5. Полтораченко Н. І. Практикум із чисельних методів : навч. посіб. / Н. І. Полтораченко, С. А. Теренчук, Ю. Н. Убайдуллаєв. – Київ : КНУБА, 2023. – 160 с.
6. Чисельні методи : конспект лекцій / О. В. Шебанина, С. І. Тищенко, І. І. Хилько, В. О. Крайній. – Миколаїв : МНАУ, 2023. – 100 с.
7. Чисельні методи в інформатиці : методичні вказівки до виконання практичних робіт / уклад. В. В. Ключова. – Київ : КНУБА, 2022. – 28 с.
8. Шахно С. М., Дудикевич А. Т., Левицька С. М. Практична реалізація чисельних методів лінійної алгебри : навч. посібник. – Львів : Вид. центр ЛНУ імені Івана Франка, 2009. – 137 с.

9. Chapra, Steven C. Numerical methods for engineers / Steven C. Chapra, Berger chair in computing and engineering, Tufts University, Raymond P. Canale, professor emeritus of civil engineering. University of Michigan. – Seventh edition, 2020. 987 p.

10. Shoup T. E. Applied Numerical Methods for Microcomputers. Prentice Hall. 1984. 262 p.

Навчальне видання

ДУДЧЕНКО Олег Миколайович
КАРПОВА Світлана Олегівна

**ЧИСЕЛЬНІ МЕТОДИ
В ІНЖЕНЕРНИХ РОЗРАХУНКАХ.
РОЗВ'ЯЗУВАННЯ СИСТЕМ
ЛІНІЙНИХ РІВНЯНЬ ЗА ДОПОМОГОЮ ПК**

Методичні вказівки
до виконання лабораторних робіт

Комп'ютерна правка й коректура *О. Є. Вакула*
Комп'ютерне верстання *Ю. С. Семенченко*

Формат 60×84/16. Ум. друк. арк. 1,86. Вид. № 16. Зам. № 1007-69.
Видавець і виготівник Національний університет кораблебудування
імені адмірала Макарова
просп. Героїв України, 9, м. Миколаїв, 54007
E-mail : publishing@nuos.edu.ua
Свідоцтво суб'єкта видавничої справи ДК № 6402 від 19.09.2018 р.