

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені І. І. МЕЧНИКОВА
ФАКУЛЬТЕТ МАТЕМАТИКИ, ФІЗИКИ та ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ОПТИМАЛЬНОГО КЕРУВАННЯ і ЕКОНОМІЧНОЇ
КІБЕРНЕТИКИ

МЕТОДИ ОБЧИСЛЕНЬ Частина II

Методичні вказівки до лабораторних робіт
для здобувачів першого (бакалаврського) рівня
вищої освіти спеціальності 113 Прикладна математика

ОДЕСА
2024

УДК 519.612.2:519.612.4(076.5)

М545

Укладач:

В. В. Вербіцький, кандидат фізико-математичних наук, доцент, доцент кафедри оптимального керування і економічної кібернетики Одеського національного університету імені І. І. Мечникова;

Рецензенти:

Є. М. Страхов, кандидат фізико-математичних наук, доцент, доцент кафедри оптимального керування і економічної кібернетики Одеського національного університету імені І. І. Мечникова;

З. Ю. Журавльова, кандидат фізико-математичних наук, доцент, доцент кафедри методів математичної фізики Одеського національного університету імені І. І. Мечникова;

*Рекомендовано вченою радою факультету математики,
фізики та інформаційних технологій ОНУ імені І. І. Мечникова.*

Протокол 1 від 11 вересня 2024 р.

Методи обчислень. Част. II.: метод. вказівки до лаб.
М545 робіт для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 113 Прикладна математика / уклад.: В. В. Вербіцький. – Одеса : Видавництво Олді+, 2024. – 44 с.

Методичні вказівки включають теми змістових модулів, методи знаходження власних значень та векторів симетричних дійсних матриць та методи розв'язання нелінійних рівнянь та систем, а саме: степеневий метод, метод зворотної ітерації, методи розв'язання повної задачі на власні значення, методи дихотомії, метод простої ітерації та метод Ньютона. До кожної теми додано необхідний теоретичний матеріал та завдання до лабораторних робіт.

УДК 519.612.2:519.612.4

ЗМІСТ

Вступ	3
1 Часткова проблема власних значень	5
1.1 Степеневий метод	5
1.2 Зворотня ітерація	7
1.3 Вичерпування відніманням	9
1.4 Використання зсувів	10
1.5 Завдання до лабораторної роботи	11
2 Повна проблема власних значень	13
2.1 QL -алгоритм	13
2.2 Метод обертань	17
2.3 Завдання до лабораторної роботи 1	22
2.4 Завдання до лабораторної роботи 2	23
3 Методи розв’язання нелінійних рівнянь	24
3.1 Методи дихотомії	25
3.2 Метод простої ітерації	26
3.3 Метод Ньютона	29
3.4 Завдання до лабораторної роботи	33
4 Методи розв’язання систем нелінійних рівнянь	35
4.1 Метод ітерацій	35
4.2 Метод Ньютона	36
4.3 Завдання до лабораторної роботи	38
Список рекомендованої літератури	40

ВСТУП

Методичні вказівки присвячені чисельним методам алгебри. Розглянуто такі теми: степеневий метод та метод зворотної ітерації розв’язання часткової проблеми власних значень, методи розв’язання повної проблеми на власні значення, методи дихотомії, метод простої ітерації та метод Ньютона розв’язання нелінійних рівнянь та систем.

До кожної теми додано необхідний теоретичний матеріал та завдання до лабораторних робіт.

На сьогоднішній день обсяг інформації в галузі методів обчислень практично неоглядний. Існують великі бібліотеки програм, такі, як бібліотека Netlib[14], (що включає пакети LINPACK, EISPACK, LAPACK, BLAS та ін.), бібліотеки чисельного аналізу Numerical Algorithms Group NAG[15], наукова бібліотека GSL асоціації [16]. Вказані бібліотеки широко застосовуються сучасними обчислювачами, однак у навчальних курсах використовуються досить рідко. До причин цього слід віднести деяку їхню "громіздкість". Досить складними є умови щодо параметрів функцій. Вихідний код зазвичай написаний на Фортрані. Ці властивості часто ускладнюють можливість використання вихідного коду цих пакетів у студентських лабораторних роботах. Тому, у вказівках приведені псевдокоди алгоритмів розглянутих числових методів, щоб у учбових цілях їх можна було легко реалізувати мовою програмування високого рівня.

Для роботи зі вказівками необхідні знання лінійної алгебри та математичного аналізу обсягом, що відповідає університетським курсам. Передбачається, що лабораторні роботи виконуються паралельно з вивченням теоретичного матеріалу за дисципліною "Методи обчислень". Як підручники з сучасних чисельних методів лінійної алгебри ми рекомендуємо[3–5, 9].

У методичних вказівках наводяться докази лише деяких ключових тверджень, що, як ми вважаємо, цілком виправдовується поставленими задачами.

1. ЧАСТКОВА ПРОБЛЕМА ВЛАСНИХ ЗНАЧЕНЬ

1.1. Степеневий метод

Припустимо, що максимальне за модулем власне значення симетричної дійсної матриці $A \in R^{n \times n}$ відокремлено, тобто,

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|.$$

У цьому випадку степеневий метод дозволяє знайти максимальне за модулем власне значення λ_1 та відповідний власний вектор q_1 .

Побудуємо наступний ітераційний процес. Нехай y_0 — початкове наближення до власного вектора q_1 . За теоремою про спектральне розкладання симетричної матриці власні вектори матриці A утворюють ортонормований базис простору $R^{n \times n}$. Розкладемо y_0 за цим базисом,

$$y_0 = \sum_{i=1}^n \alpha_i q_i, \quad (1.1)$$

де $\alpha_i = q_i^T y_0$. Припустимо, що $\alpha_1 \neq 0$. Нормуємо вектор y_0 ,

$$x_0 = \frac{y_0}{\|y_0\|_2}.$$

Помножимо матрицю A на вектор x_0 ,

$$y_1 = Ax_0 = \frac{\sum_{i=1}^n \alpha_i \lambda_i q_i}{\left\| \sum_{i=1}^n \alpha_i q_i \right\|_2}.$$

Нормуємо вектор y_1 ,

$$x_1 = \frac{y_1}{\|y_1\|_2} = \frac{\sum_{i=1}^n \alpha_i \lambda_i q_i}{\left\| \sum_{i=1}^n \alpha_i \lambda_i q_i \right\|_2}.$$

Після k ітерацій маємо,

$$x_k = \frac{\sum_{i=1}^n \alpha_i \lambda_i^k q_i}{\left\| \sum_{i=1}^n \alpha_i \lambda_i^k q_i \right\|_2}, \quad (1.2)$$

$$y_{k+1} = Ax_k = \frac{\sum_{i=1}^n \alpha_i \lambda_i^{k+1} q_i}{\left\| \sum_{i=1}^n \alpha_i \lambda_i^k q_i \right\|_2}. \quad (1.3)$$

Покладемо,

$$\lambda^{(k+1)} = y_{k+1}^T x_k.$$

Покажемо, що $\lambda^{(k+1)} \rightarrow \lambda_1$ при $k \rightarrow \infty$. З (1.2) та (1.3) маємо,

$$\begin{aligned} \lambda^{(k+1)} &= y_{k+1}^T x_k = \frac{\left(\sum_{i=1}^n \alpha_i \lambda_i^{k+1} q_i \right)^T \left(\sum_{i=1}^n \alpha_i \lambda_i^k q_i \right)}{\left\| \sum_{i=1}^n \alpha_i \lambda_i^k q_i \right\|_2^2} = \frac{\sum_{i=1}^n \alpha_i^2 \lambda_i^{2k+1}}{\sum_{i=1}^n \alpha_i^2 \lambda_i^{2k}} = \\ &= \frac{\lambda_1 + \frac{\alpha_2^2 \lambda_2}{\alpha_1^2} \left(\frac{\lambda_2}{\lambda_1} \right)^{2k} + \dots + \frac{\alpha_n^2 \lambda_n}{\alpha_1^2} \left(\frac{\lambda_n}{\lambda_1} \right)^{2k}}{1 + \frac{\alpha_2^2}{\alpha_1^2} \left(\frac{\lambda_2}{\lambda_1} \right)^{2k} + \dots + \frac{\alpha_n^2}{\alpha_1^2} \left(\frac{\lambda_n}{\lambda_1} \right)^{2k}} = \frac{\lambda_1 + O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}\right)}{1 + O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}\right)}. \end{aligned}$$

Звідси,

$$\lambda^{(k+1)} - \lambda_1 = O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}\right). \quad (1.4)$$

Покажемо тепер, що $x_k \rightarrow q_1$ при $k \rightarrow \infty$. Дійсно,

$$x_k = \frac{\sum_{i=1}^n \alpha_i \lambda_i^k q_i}{\left\| \sum_{i=1}^n \alpha_i \lambda_i^k q_i \right\|_2} = \frac{\alpha_1 \lambda_1^k q_1 + \alpha_2 \lambda_2^k q_2 + \dots + \alpha_n \lambda_n^k q_n}{|\alpha_1| |\lambda_1|^k \sqrt{1 + O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}\right)}} =$$

$$= \frac{\pm q_1 + O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^k\right)}{\sqrt{1 + O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}\right)}}.$$

Звідси,

$$x_k - q_1 = O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^k\right). \quad (1.5)$$

Зауважимо, що порядок швидкості збіжності власного значення у 2 рази більше, ніж власного вектор.

Запишемо алгоритм степеневого методу.

Алгоритм 1.1 *Степенной метод*

1. $x_0 = y_0 / \|y_0\|_2$
2. *For* $k = 0, 1, \dots$ *Until Convergence Do*
3. $y_{k+1} = Ax_k$
4. $\lambda^{(k+1)} = y_{k+1}^T x_k$
5. $x_{k+1} = y_{k+1} / \|y_{k+1}\|_2$
6. *End Do*

Тестом на збіжність може бути одна з наступних умов.

$$\left| \lambda^{(k+1)} - \lambda^{(k)} \right| \leq \varepsilon \left| \lambda^{(k+1)} \right|, \quad (1.6)$$

або

$$\left| \max_{(x_k)_i \neq 0} \frac{(y_{k+1})_i}{(x_k)_i} - \min_{(x_k)_i \neq 0} \frac{(y_{k+1})_i}{(x_k)_i} \right| \leq \varepsilon \left| \max_{(x_k)_i \neq 0} \frac{(y_{k+1})_i}{(x_k)_i} \right|. \quad (1.7)$$

Відмітимо, що крок 3. алгоритма 1.1 вимагає $O(n^2)$ арифметичних дій, решта кроків — $O(n)$ арифметичних дій.

1.2. Зворотня ітерація

Розглянемо тепер метод зворотної ітерації. Припустимо, що найменше за модулем власне значення дійсної симетричної матриці $A \in R^{n \times n}$ відокрем-

лено, тобто,

$$|\lambda_1| \geq \dots \geq |\lambda_{n-1}| > |\lambda_n|. \quad (1.8)$$

Зауважимо, що якщо $\{\lambda, x\}$ — власна пара матриці A , то $\{1/\lambda, x\}$ — власна пара зворотної матриці A^{-1} . Очевидно, що $1/\lambda_n$ — максимальне за модулем власне значення матриці A^{-1} . Тому, для знаходження $1/\lambda_n$ можна застосувати степеневий метод для матриці A^{-1} . У цьому випадку при $k \rightarrow \infty$

$$\lambda^{(k+1)} \rightarrow \frac{1}{\lambda_n}, \quad x_k \rightarrow q_n.$$

Таким чином, зворотна ітерація дозволяє обчислити найменше за модулем власне значення матриці (і відповідний власний вектор), якщо воно відокремлено від інших власних значень.

Запишемо алгоритм зворотної ітерації.

Алгоритм 1.2 Зворотна ітерація

1. $x_0 = y_0 / \|y_0\|_2$
2. *QuadFor* $k = 0, 1, \dots$ *Until Convergence Do*
3. $y_{k+1} = A^{-1}x_k$
4. $\lambda^{(k+1)} = y_{k+1}^T x_k$
5. $x_{k+1} = y_{k+1} / \|y_{k+1}\|_2$
6. *End Do*

Крок 3. алгоритма зворотної ітерації зводиться до розв'язання системи $Ay_{k+1} = x_k$. Для розв'язання цієї системи потрібно $O(n^3)$ арифметичних дій. Однак, якщо відомо деяке розкладання матриці A (наприклад, LU -розкладання), то цей крок вимагатиме лише $O(n^2)$ арифметичних дій. Інші кроки алгоритму вимагають лише $O(n)$ дій.

Зауваження 1.1 Встановлюючи збіжність степеневого методу, ми припускали (див. (1.1)), що коефіцієнт α_1 у розкладанні за власними векторами початкового наближення y_0 відмінний від нуля. Навіть якщо це не так, то, внаслідок округлень при виконанні арифметичних дій, на деякій ітерації k коефіцієнт α_1 у розкладанні y_k буде відмінний від нуля.

1.3. Вичерпування відніманням

Нехай власні значення дійсної симетричної матриці $A \in R^{n \times n}$ задовольняють умовам

$$|\lambda_1| > |\lambda_2| > |\lambda_3| \geq \dots \geq |\lambda_n|.$$

Припустимо, що власну пару $\{\lambda_1, q_1\}$ вже знайдено. Необхідно знайти $\{\lambda_2, q_2\}$.

Нехай $\tilde{y}_0$ — початкове наближення та

$$\tilde{y}_0 = \alpha_1 q_1 + \alpha_2 q_2 + \dots + \alpha_n q_n,$$

причому $\alpha_2 \neq 0$. Покладемо

$$y_0 = \tilde{y}_0 - \alpha_1 q_1 = \tilde{y}_0 - \tilde{y}_0^T q_1 q_1.$$

Повторюючи доказ збіжності степеневому методу для обраного таким чином початкового наближення y_0 , переконаємося, що при $k \rightarrow \infty$ в точній арифметиці

$$\lambda^{(k+1)} \rightarrow \lambda_2, \quad x_k \rightarrow q_2.$$

Однак, через похибки округлень коефіцієнт при q_1 знову може з'явитися у розкладанні y_k . Тому цей коефіцієнт видаляють на кожній ітерації або через кілька ітерацій. Таким чином, для знаходження власної пари $\{\lambda_2, q_2\}$ можна застосувати степеневий метод з вичерпуванням відніманням.

Алгоритм 1.3 *Степеневий метод з вичерпуванням відніманням*

1. $y_0 = \tilde{y}_0 - \tilde{y}_0^T q_1 q_1$
2. $x_0 = y_0 / \|y_0\|_2$
3. *For* $k = 0, 1, \dots$ *Until Convergence Do*
4. $\tilde{y}_{k+1} = Ax_k$
5. $y_{k+1} = \tilde{y}_{k+1} - \tilde{y}_{k+1}^T q_1 q_1$
6. $\lambda^{(k+1)} = y_{k+1}^T x_k$
7. $x_{k+1} = y_{k+1} / \|y_{k+1}\|_2$
8. *End Do*

Так по черзі можна знайти кілька максимальних за модулем власних

значень, якщо вони відокремлені один від одного і від решти спектра.

Очевидно, що для знаходження кількох відокремлених мінімальних за модулем власних значень можна застосувати зворотну ітерацію з вичерпуванням відніманням.

1.4. Використання зсувів

Нехай $A \in R^{n \times n}$ і $\sigma \in R$. Якщо $\{\lambda, x\}$ — власна пара матриці A , то $\{\lambda - \sigma, x\}$ є власною парою матриці $A - \sigma I$. Дійсне число σ називається зсувом.

Спочатку розглянемо використання зсуву задля збільшення швидкості збіжності зворотної ітерації. Оскільки зворотна ітерація — це степеневий метод для зворотної матриці A^{-1} , то

$$\lambda^{(k+1)} - \frac{1}{\lambda_n} = O \left(\left(\frac{\frac{1}{\lambda_{n-1}}}{\frac{1}{\lambda_n}} \right)^{2k} \right) = O \left(\left(\frac{\lambda_n}{\lambda_{n-1}} \right)^{2k} \right).$$

Звідси випливає, що швидкість збіжності залежить від малості величини $|\frac{\lambda_n}{\lambda_{n-1}}|$. Остання залежить від відстані між $|\lambda_{n-1}|$ і $|\lambda_n|$, тобто, від відокремлення найменшого за модулем власного значення, і від малості $|\lambda_n|$. Тому зсув σ вибирають так, щоб найменше за модулем власне значення $\tilde{\lambda}_n = \lambda_n - \sigma$ матриці $A - \sigma I$ було якомога ближче до нуля. Таким чином, для збільшення швидкості збіжності зворотної ітерації зсув σ треба вибирати якомога ближче до шуканого власного значення λ_n .

Відомо, що якщо $\tilde{x}$ — наближення до деякого власного вектора симетричної матриці, то найкращим наближенням до відповідного власного значення дає відношення Релея

$$\rho(\tilde{x}) = \frac{\tilde{x}^T A \tilde{x}}{\tilde{x}^T \tilde{x}}.$$

Вищесказане лежить в основі наступного алгоритму.

Алгоритм 1.4 Зворотна ітерація зі зсувом Релея

1. $x_0 = y_0 / \|y_0\|_2$
2. For $k = 0, 1, \dots$ Until Convergence Do

3. $\sigma_{k+1} = x_k^T A x_k$
4. $y_{k+1} = (A - \sigma_{k+1} I)^{-1} x_k$
5. $x_{k+1} = y_{k+1} / \|y_{k+1}\|_2$
6. *End Do*

Тестом на збіжність може бути умова (1.7).

Зауважимо, що крок 3. алгоритму вимагає $O(n^3)$ арифметичних дій, бо щоразу треба розв'язувати систему лінійних алгебраїчних рівнянь із новою матрицею.

Розглянемо тепер використання зсуву для обчислення власних значень матриці степеневим методом (зворотною ітерацією) відмінних від максимальних (мінімальних) за модулем. Очевидно, що зворотна ітерація, обчислюючи мінімальне за модулем власне значення матриці $A - \sigma I$, фактично буде обчислювати власне значення матриці A найближче до σ .

Визначивши межі спектру матриці A (наприклад, за лемою Гершгоріна), можна так вибрати зсув σ , щоб степеневий метод чи зворотна ітерація фактично обчислювали найменше (найбільше) власне значення матриці A . Дійсно, якщо $\sigma \leq \lambda_{\min}(A)$, то всі власні значення матриці $A - \sigma I$ додатні, найменшому власному значенню матриці A відповідає мінімальне (мінімальне за модулем) власне значення матриці $A - \sigma I$, а максимальному власному значенню матриці A — максимальне (максимальне за модулем) власне значення матриці $A - \sigma I$. Якщо ж $\sigma \geq \lambda_{\max}(A)$, то всі власні значення матриці $A - \sigma I$ від'ємні, найменшому власному значенню матриці A відповідає мінімальне (максимальне за модулем) власне значення матриці $A - \sigma I$, а максимального власного значення матриці A — максимальне (мінімальне за модулем) власне значення матриці $A - \sigma I$.

1.5. Завдання до лабораторної роботи

Мовою пакету Octave написати програми степеневого методу та методу зворотної ітерації.

Провести наступний обчислювальний експеримент. Для заданої матриці A із заданою точністю ε :

1. знайти найбільше і найменше власні значення степеневим методом;
2. знайти найбільше та найменше власні значення методом зворотної ітерації;
3. порівняти кількість ітерацій, витрачених методами для обчислення найбільшого (найменшого) власного значення;
4. знайти друге мінімальне за модулем власне значення;
5. знайти найближче до заданого числа α власне значення;
6. порівняти швидкість збіжності зворотної ітерації та зворотної ітерації з зсувом Релея для знаходження мінімального за модулем власного значення.

Варіанти завдань

Матриця A визначається так:

$$\begin{aligned} a_{11} &= \alpha, \\ a_{i+1,i+1} &= -a_{i,i}, \quad i = \overline{1, n-1}, \\ a_{ij} &= (i+j)/\alpha, \quad i \neq j, \end{aligned}$$

де $n = 20$, $\alpha = 10+k/10$, k — номер прізвища студента у журналі студентської групи.

До звіту з виконання лабораторної роботи включити:

1. опис алгоритмів, що використовуються;
2. текст програми;
3. результати обчислювального експерименту.

2. ПОВНА ПРОБЛЕМА ВЛАСНИХ ЗНАЧЕНЬ

2.1. QL -алгоритм

QL -алгоритм розв'язання повної проблеми власних значень дійсної симетричної матриці $A \in R^{n \times n}$ будує послідовність ортогонально подібних матриць $\{A_k\}_{k=1}^{\infty}$ так, що $A_k \rightarrow \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ при $k \rightarrow \infty$. Покладемо $A_1 = A$. Перехід від k -ї до $(k+1)$ -ї ітерації полягає в наступному. Використовуючи обертання Якобі, будуємо QL -розкладання матриці A_k :

$$A_k = Q_k L_k. \quad (2.1)$$

Обчислюємо

$$A_{k+1} = L_k Q_k. \quad (2.2)$$

Тут Q_k — ортогональна матриця, L_k — нижня трикутна матриця.

Спочатку зауважимо, що

$$A_{k+1} = L_k Q_k = Q_k^T A_k Q_k = \dots = Q_k^T Q_{k-1}^T \dots Q_1^T A_1 Q_1 \dots Q_{k-1} Q_k.$$

Таким чином,

$$A_{k+1} = P_k^T A P_k, \quad (2.3)$$

де $P_k = Q_1 \dots Q_{k-1} Q_k$ — ортогональна матриця. Значить, матриці A_{k+1} і A ортогонально подібні, тому їх власні значення співпадають.

Встановимо зв'язок між QL -алгоритмом, степеневим методом і зворотною ітерацією. QL -алгоритм породжує послідовність матриць $\{A_k\}_{k=1}^{\infty}$,

$$A_{k+1} = Q_k^T A_k Q_k = P_k^T A P_k, \quad (2.4)$$

де $A_k = Q_k L_k$, $A_1 = A$, $P_k = Q_1 \dots Q_k$. Степеневий метод породжує послідовність векторів $\{v_k\}_{k=1}^{\infty}$,

$$v_{k+1} = \frac{A v_k}{\mu_k}, \quad (2.5)$$

де v_1 ($\|v_1\|_2 = 1$) — початкове наближення, μ_k — нормуючий множник, який

вибирають так, щоб $\|v_k\|_2 = 1$ для будь-якого k . Зворотна ітерація породжує послідовність векторів $\{u_k\}_{k=1}^\infty$,

$$Au_{k+1} = \tau_k u_k, \quad (2.6)$$

де u_1 ($\|u_1\|_2 = 1$) — початкове наближення, τ_k — нормуючий множник, який вибирають так, щоб $\|u_k\|_2 = 1$ для будь-якого k .

Теорема 2.1 *Нехай $u_1 = e_1$, $v_1 = e_n$, $P_0 = I$. Тоді послідовності векторів, що породжуються степеневим методом та зворотною ітерацією, пов'язані з послідовністю матриць QL -алгоритму наступним чином. Для будь-якого $k \geq 1$*

$$u_k = P_{k-1}e_1, \quad v_k = P_{k-1}e_n. \quad (2.7)$$

Доведення. Використовуючи (2.4), отримуємо

$$P_k L_k = P_{k-1} Q_k L_k = P_{k-1} A_k = P_{k-1} P_{k-1}^T A P_{k-1} = A P_{k-1},$$

тобто,

$$P_k L_k = A P_{k-1}. \quad (2.8)$$

Прирівнюючи останні стовпці в матричній рівності (2.8), отримуємо

$$P_k e_n e_n^T L_k e_n = A P_{k-1} e_n.$$

Порівнюючи цю рекурентну формулу з (2.5), знаходимо, що $v_k = P_{k-1}e_n$, $\mu_k = e_n^T L_k e_n$.

З (2.8)

$$L_k^T P_k^T = P_{k-1}^T A.$$

Домножуючи останню рівність зліва на P_{k-1} і праворуч на P_k , отримуємо

$$P_{k-1} L_k^T = A P_k.$$

Прирівнюючи перші стовпці у цій матричній рівності, отримуємо

$$P_{k-1}e_1e_1^TL_k e_1 = AP_k e_1.$$

Порівнюючи цю рекурентну формулу з (2.7), знаходимо, що $u_k = P_{k-1}e_1$, $\tau_k = e_1^TL_k e_1$. Теорему доведено.

Доведемо тепер збіжність QL -алгоритму, використовуючи його зв'язок з степеневим методом та зворотною ітерацією.

Теорема 2.2 *Нехай власні пари $\{\lambda_i, q_i\}$ ($i = \overline{1, n}$) симетричної дійсної матриці $A \in R^{n \times n}$ задовольняють умовам*

$$|\lambda_1| < |\lambda_2| \leq \dots \leq |\lambda_{n-1}| < |\lambda_n|, \quad (2.9)$$

$$e_1^T q_1 \neq 0, \quad e_n^T q_n \neq 0. \quad (2.10)$$

$\{A_k\}_{k=1}^\infty$ — послідовність QL -алгоритму. Тоді

$$A_k e_1 \rightarrow \lambda_1 e_1, \quad A_k e_n \rightarrow \lambda_n e_n \quad \text{при } k \rightarrow \infty.$$

Доведення. За теоремою 2.1 про зв'язок QL -алгоритму зі зворотним ітерацією, QL -алгоритм формує послідовність векторів $\{u_k\}_{k=1}^\infty$ зворотної ітерації:

$$u_1 = e_1, \quad u_k = P_{k-1}e_1.$$

З умов (2.9), (2.10) випливає, що ця послідовність збігається і

$$u_k = q_1 + O\left(\left(\frac{\lambda_1}{\lambda_2}\right)^k\right).$$

Значить,

$$e_1 = P_{k-1}^T P_{k-1} e_1 = P_{k-1}^T u_k = P_{k-1}^T q_1 + O\left(\left(\frac{\lambda_1}{\lambda_2}\right)^k\right),$$

i

$$\begin{aligned} A_k e_1 &= P_{k-1}^T A P_{k-1} e_1 = P_{k-1}^T A u_k = P_{k-1}^T (\lambda_1 q_1 + O\left(\left(\frac{\lambda_1}{\lambda_2}\right)^k\right)) = \\ &= \lambda_1 P_{k-1}^T q_1 + O\left(\left(\frac{\lambda_1}{\lambda_2}\right)^k\right) = \lambda_1 e_1 + O\left(\left(\frac{\lambda_1}{\lambda_2}\right)^k\right), \end{aligned}$$

тобто,

$$A_k e_1 = \lambda_1 e_1 + O\left(\left(\frac{\lambda_1}{\lambda_2}\right)^k\right). \quad (2.11)$$

Аналогічно можна показати, що

$$A_k e_n = \lambda_n e_n + O\left(\left(\frac{\lambda_{n-1}}{\lambda_n}\right)^k\right).$$

Теорему доведено. $\square$

QL -алгоритм можна застосовувати для будь-якої дійсної симетричної матриці. Однак, найбільша ефективність методу досягається для тридіагональних симетричних матриць. Тому симетричну матрицю спочатку приводять до тридіагонального вигляду ортогональними перетвореннями, а потім застосовують QL -алгоритм.

Для підвищення швидкості збіжності QL -алгоритму використовують зсуви. Перехід від k -й до $(k+1)$ -й ітерації QL -алгоритму зі зсувами полягає в наступному. Вибираємо зсув σ_k . Використовуючи обертання Якобі, будуємо QL -розкладання матриці $A_k - \sigma_k I$:

$$A_k - \sigma_k I = Q_k L_k.$$

Обчислюємо

$$A_{k+1} = L_k Q_k + \sigma_k I.$$

Оскільки

$$A_{k+1} = L_k Q_k + \sigma_k I = Q_k^T (A_k - \sigma_k I) Q_k + \sigma_k I = Q_k^T A_k Q_k,$$

то

$$A_{k+1} = Q_k^T A_k Q_k = \dots = Q_k^T Q_{k-1}^T \dots Q_1^T A_1 Q_1 \dots Q_{k-1} Q_k = P_k^T A P_k,$$

де $P_k = Q_1 \dots Q_{k-1} Q_k$ — ортогональна матриця. Значить матриці A_k знову ортогонально подібні до матриці A , тобто, їхні власні значення співпадають з власними значеннями матриці A . З (2.11) випливає, що чим ближче λ_1 до нуля, тим швидше збігається QL -алгоритм. Оскільки $A_k e_1 \rightarrow \lambda_1 e_1$ при $k \rightarrow \infty$, то є сенс в якості зсуву σ_k брати $a_{11}^{(k)}$. Такий зсув називають зсувом за Релеєм. Зсув за Вілкінсоном — це найближче до $a_{11}^{(k)}$ власне значення матриці

$$\begin{bmatrix} a_{11}^{(k)} & a_{12}^{(k)} \\ a_{21}^{(k)} & a_{22}^{(k)} \end{bmatrix}.$$

2.2. Метод обертань

Розглянемо ще один метод розв'язання повної проблеми власних значень для симетричної дійсної матриці $A \in R^{n \times n}$, який називають методом обертань або методом Якобі.

Метод будує послідовність ортогонально подібних матриць

$$\{A_k\}_{k=1}^{\infty} \quad (A_1 = A)$$

так, що

$$A_k \rightarrow \Lambda = \text{diag}(\lambda_1, \dots, \lambda_n) \text{ при } k \rightarrow \infty.$$

Опишемо перехід від k -ї до $(k+1)$ -ї ітерації. У матриці A_k знайдемо максимальний за модулем позадіагональний елемент

$$|a_{pq}^{(k)}| = \max_{i \neq j} |a_{ij}^{(k)}|. \quad (2.12)$$

Побудуємо матрицю обертання $Q_{pq}^{(k)}$, яка відрізняється від одиничної матриці

елементами

$$q_{pp}^{(k)} = \cos \varphi_k, \quad q_{pq}^{(k)} = -\sin \varphi_k, \quad q_{qp}^{(k)} = \sin \varphi_k, \quad q_{qq}^{(k)} = \cos \varphi_k.$$

φ_k виберемо так, щоб у матриці

$$A_{k+1} = (Q_{pq}^{(k)})^T A_k Q_{qp}^{(k)}$$

у позиціях (p, q) і (q, p) були нулі. Зазначимо, що в наступних ітераціях ці нулі не зберігаються.

Позначимо

$$B = A_k Q_{pq}^{(k)},$$

тоді,

$$A_{k+1} = (Q_{pq}^{(k)})^T B.$$

При переході від A_k до B перераховуються лише p -й та q -й стовпці:

$$\begin{aligned} b_{ip} &= a_{ip}^{(k)} c + a_{iq}^{(k)} s, \\ b_{iq} &= -a_{ip}^{(k)} s + a_{iq}^{(k)} c, \quad i = \overline{1, n}, \end{aligned} \tag{2.13}$$

де $c = \cos \varphi_k$, $s = \sin \varphi_k$. При переході від B до A_{k+1} перераховуються лише p -й та q -й рядки:

$$\begin{aligned} a_{pj}^{(k+1)} &= b_{pj} c + b_{qj} s, \\ a_{qj}^{(k+1)} &= -b_{pj} s + b_{qj} c, \quad j = \overline{1, n}. \end{aligned} \tag{2.14}$$

З (2.13), (2.14) отримуємо

$$\begin{aligned} a_{pq}^{(k+1)} &= b_{pq} c + b_{qq} s = (-a_{pp}^{(k)} s + a_{pq}^{(k)} c) c + (-a_{qp}^{(k)} s + a_{qq}^{(k)} c) c = \\ &= (a_{qq}^{(k)} - a_{pp}^{(k)}) s c + a_{pq} (c^2 - s^2). \end{aligned}$$

Тепер умова,

$$a_{pq}^{(k+1)} = 0$$

для визначення матриці обертання $Q_{pq}^{(k)}$, набуває вигляду

$$(a_{qq}^{(k)} - a_{pp}^{(k)})sc + a_{pq}(c^2 - s^2) = 0.$$

Розділивши це рівняння на $-c^2$ і ввівши позначення

$$t = \frac{s}{c}, \quad \tau = \frac{a_{pp}^{(k)} - a_{qq}^{(k)}}{2a_{pq}},$$

приходимо до квадратного рівняння

$$t^2 + 2\tau t - 1 = 0.$$

В якості t виберемо менший за модулем корінь квадратного рівняння

$$t = \text{sign}(\tau)(\sqrt{1 + \tau^2} - |\tau|) = \frac{\text{sign}(\tau)}{|\tau| + \sqrt{1 + \tau^2}}. \quad (2.15)$$

Тепер $\sin \varphi_k$ і $\cos \varphi_k$ можна знайти за формулами

$$\cos \varphi_k = \frac{1}{\sqrt{1 + t^2}}, \quad \sin \varphi_k = t \cos \varphi_k. \quad (2.16)$$

Таким чином, перехід від k -ї до $(k + 1)$ -ї ітерації здійснюється так:

1. знаходимо максимальний за модулем позадіагональний елемент матриці A_k ;
2. за формулами (2.15), (2.16) обчислюємо $\cos \varphi_k$ і $\sin \varphi_k$;
3. елементи матриці A_{k+1} обчислюємо за формулами (2.13), (2.14).

Дослідимо збіжність методу обертань. Позначимо через σ_k^2 суму квадратів позадіагональних елементів матриці A_k . Доведемо, що

$$\sigma_k^2 \rightarrow 0 \text{ при } k \rightarrow \infty.$$

Оскільки ортогональні перетворення зберігають норму Фробеніуса, то

$$\sigma_{k+1}^2 = \|A_{k+1}\|_F^2 - \sum_{i=1}^n (a_{ii}^{(k+1)})^2 = \|A_k\|_F^2 - \sum_{i=1, i \neq p, q}^n (a_{ii}^{(k)})^2 - (a_{pp}^{(k+1)})^2 - (a_{qq}^{(k+1)})^2.$$

Очевидно, що

$$\begin{bmatrix} a_{pp}^{(k+1)} & a_{pq}^{(k+1)} \\ a_{qp}^{(k+1)} & a_{qq}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \cos \varphi_k & -\sin \varphi_k \\ \sin \varphi_k & \cos \varphi_k \end{bmatrix}^T \begin{bmatrix} a_{pp}^{(k)} & a_{pq}^{(k)} \\ a_{qp}^{(k)} & a_{qq}^{(k)} \end{bmatrix} \begin{bmatrix} \cos \varphi_k & -\sin \varphi_k \\ \sin \varphi_k & \cos \varphi_k \end{bmatrix}.$$

Оскільки $a_{pq}^{(k+1)} = a_{qp}^{(k+1)} = 0$, прирівнюючи норми Фробеніуса цих ортогонально подібних матриць, отримуємо

$$(a_{pp}^{(k+1)})^2 + (a_{qq}^{(k+1)})^2 = (a_{pp}^{(k)})^2 + 2(a_{pq}^{(k)})^2 + (a_{qq}^{(k)})^2.$$

Тому,

$$\begin{aligned} \sigma_{k+1}^2 &= \|A_k\|_F^2 - \sum_{i=1, i \neq p, q}^n (a_{ii}^{(k)})^2 - (a_{pp}^{(k+1)})^2 - (a_{qq}^{(k+1)})^2 = \\ &= \|A_k\|_F^2 - \sum_{i=1}^n (a_{ii}^{(k)})^2 - 2(a_{pq}^{(k)})^2 = \sigma_k^2 - 2(a_{pq}^{(k)})^2. \end{aligned}$$

Очевидно, що

$$\sigma_k^2 \leq n(n-1)(a_{pq}^{(k)})^2.$$

Тому,

$$\sigma_{k+1}^2 \leq \left(1 - \frac{2}{n(n-1)}\right) \sigma_k^2.$$

Застосовуючи отриману нерівність послідовно, отримуємо

$$\sigma_{k+1}^2 \leq \left(1 - \frac{2}{n(n-1)}\right)^k \sigma_1^2.$$

Значить,

$$\sigma_{k+1}^2 \rightarrow 0 \text{ при } k \rightarrow \infty,$$

та збіжність методу лінійна. Однак асимптотична швидкість збіжності значно

вища за лінійну. Так, для великих k вона квадратична[8].

Метод Якобі історично є найстарішим для проблеми власних значень. Маючи трудомісткість $O(n^3)$ арифметичних операцій, він набагато повільніше сучасних методів, наприклад, QL -алгоритму. Однак, інтерес до методу зберігається тому, що часом він дає набагато точніші результати, ніж інші методи [8], крім того метод Якобі дозволяти розпаралелювання.

Наведемо алгоритм методу Якобі.

Алгоритм 2.1 Метод Якобі

```

1; 1.  $Q = I$ 
2. While True Do
3.    $|a_{pq}| = \max_{i \neq j} |a_{ij}|$ 
4;   If  $|a_{pq}| \leq \varepsilon ||A||$  Then Break
5.    $\tau = \frac{a_{pp} - a_{qq}}{2a_{pq}}$ 
6.    $t = \frac{\text{sign}(\tau)}{|\tau| + \sqrt{1 + \tau^2}}$ 
7.    $c = 1/\sqrt{1 + t^2}; s = tc$ 
8.   For  $i = 1 : n$ 
9.      $b_p(i) = a_{ip}c + a_{iq}s$ 
10.     $b_q(i) = -a_{ip}s + a_{iq}c$ 
11.   Next i
12.    $a(:, p) = b_p; a(:, q) = b_q$ 
13.   For  $j = 1 : n$ 
14.      $b_p(j) = a_{pj}c + a_{qj}s$ 
15.      $b_q(j) = -a_{pj}s + a_{qj}c$ 
16.   Next j
17.    $a(p, :) = b_p; a(q, :) = b_q$ 
18.   For  $i = 1 : n$ 
19.      $b_p(i) = Q_{ip}c + Q_{iq}s$ 
20.      $b_q(i) = -Q_{ip}s + Q_{iq}c$ 
21.   Next i
22.    $Q(:, p) = b_p; Q(:, q) = b_q$ 
23. End Do

```

2.3. Завдання до лабораторної роботи 1

Мовою пакета Octave написати програму, що реалізує QR -алгоритм обчислення всіх власних значень симетричної матриці. Для побудови QR -розкладання матриці використовувати Octave-функцію $qr()$. Передбачити використання зсувів для прискорення швидкості збіжності алгоритму.

Використовувати зсуви двох типів:

- $\sigma = a_{nn}^{(k)}$;
- зсув за Вілкінсоном.

Обчислити всі власні значення заданої матриці QR -алгоритмом з точністю $\varepsilon = 10^{-6}$. Для визначення точності обчислення використати таку оцінку. Якщо

$$\| [a_{n,1}^{(k)}, \dots, a_{n,n-1}^{(k)}]^T \|_2 / (n-1) < \varepsilon * |a_{n,n}^{(k)}|,$$

$a_{n,n}^{(k)}$ — мінімальне за модулем власне значення матриці $A \in R^{n \times n}$.

Визначити кількість ітерацій QR -алгоритму для обчислення власних значень без зсуву та зі зсувами обох типів.

Обчислити всі власні значення заданої матриці Octave-функцією $eig()$.

Варіанти завдань

Варіант завдання визначається номером $i = n + g$, де n — номер ПІБ студента у списку групи, g — номер групи.

$$A = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\ \alpha_2 & \alpha_1 & \alpha_2 & \alpha_3 \\ \alpha_3 & \alpha_2 & \alpha_1 & \alpha_2 \\ \alpha_4 & \alpha_3 & \alpha_2 & \alpha_1 \end{bmatrix},$$

де $\alpha_1 = i \bmod 10 + i * 0.1$, $\alpha_2 = \alpha_1/2$, $\alpha_3 = \alpha_1/4$, $\alpha_4 = -\alpha_1/4$.

Звіт з виконання лабораторної роботи повинен містити:

1. опис $QR(QL)$ -алгоритму;
2. текст програми мовою пакету Octave;
3. результати обчислень;
4. висновки.

2.4. Завдання до лабораторної роботи 2

Завдання

Мовою пакета Octave написати програму алгоритму Якобі обчислення всіх власних значень симетричної матриці. Ітерації алгоритму виконувати до виконання умови:

$$\max_{i \neq j} |a_{ij}^{(k)}| < \varepsilon * \max_i |a_{ii}^{(k)}|.$$

Обчислити всі власні значення заданої матриці алгоритмом Якобі з точністю $\varepsilon = 10^{-6}$.

За результатами обчислювального експеримента знайти кількість ітерацій алгоритму Якобі, що затрачені для обчислення власних значень заданої матриці та побудувати графік залежності величини σ_k^2 від номера ітерації;

Обчислити всі власні значення заданої матриці Octave-функцією $eig()$.

Варіанти завдань

Варіант завдання визначається номером $i = n + g$, де n — номер ПІБ студента у списку групи, g — номер групи.

$$A = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\ \alpha_2 & \alpha_1 & \alpha_2 & \alpha_3 \\ \alpha_3 & \alpha_2 & \alpha_1 & \alpha_2 \\ \alpha_4 & \alpha_3 & \alpha_2 & \alpha_1 \end{bmatrix},$$

де $\alpha_1 = i \bmod 10 + i * 0.1$, $\alpha_2 = \alpha_1/2$, $\alpha_3 = \alpha_1/4$, $\alpha_4 = -\alpha_1/4$.

До звіту включити:

1. текст програми мовою пакету Octave;
2. результати обчислень;
3. висновки.

3. МЕТОДИ РОЗВ'ЯЗАННЯ НЕЛІНІЙНИХ РІВНЯНЬ

Розглянемо задачу обчислення коренів рівняння

$$f(x) = 0 \quad (3.1)$$

або, що те саме, нулів функції $f(x)$, де $f : R \rightarrow R$ — алгебраїчна або трансцендентна функція. Нелінійна функція $f(x)$ у своїй області визначення $D(f) \subseteq R$ може мати скінченну/нескінченну кількість нулів або не мати їх зовсім. Більшість числових методів знаходження нулів функції потребує знання проміжків, де явно є єдиний нуль функції. Задачу щодо існування та локалізації коренів нелінійної функції зазвичай вирішують засобами математичного аналізу. Розглянемо, наприклад, графічний метод локалізації коренів. Припустимо, що рівняння (3.1) можна подати у вигляді

$$f_1(x) = f_2(x), \quad (3.2)$$

де функції $f_1(x)$ і $f_2(x)$ такі, що можна побудувати графіки $y = f_1(x)$ і $y = f_2(x)$. Тоді корені рівняння (3.2) — абсциси точок перетину цих графіків.

Приклад 3.1 Рівняння

$$x^2 - \sin x - 1 = 0$$

запишемо у вигляді

$$x^2 - 1 = \sin x.$$

Побудувавши графіки функцій $y = x^2 - 1$ і $y = \sin x$, встановлюємо, що рівняння має два корені: $\xi_1 \in [-1, 0]$, $\xi_2 \in [1, \pi/2]$.

Надалі будемо вважати, що проміжок $[a, b]$ містить єдиний корінь рівняння (3.1).

3.1. Методи дихотомії

Нехай функція $f(x)$ визначена і неперервна на проміжку $[a, b]$ і $f(a)f(b) < 0$, тоді, згідно з теоремою Больцано-Коші, на інтервалі (a, b) вона має хоча б один корінь. Візьмемо довільну точку $c \in (a, b)$. Обчислимо $f(c)$. Можливі випадки: $f(c) = 0$ — корінь знайдено; $f(a)f(c) < 0$ — корінь знаходиться на проміжку (a, c) ; $f(c)f(b) < 0$ — корінь знаходиться на проміжку (c, b) . Якщо корінь не знайдено, процес повторюємо для нового проміжку. Найбільш поширеним випадком методу дихотомії є метод половинного поділу, за яким точка $c = (a + b)/2$. За один крок методу половинного поділу проміжок, що містить корінь, зменшується вдвічі. Через k кроків шуканий корінь ξ буде на проміжку $[a_k, b_k]$ довжиною $(b - a)/2^k$ і

$$|\xi - x_k| < \frac{b - a}{2^k}, \quad (3.3)$$

де $x_k \in (a_k, b_k)$. Значить,

$$\lim_{k \rightarrow \infty} x_k = \xi.$$

З іншого боку, нерівність (3.3) дозволяє визначити кількість кроків, яка достатня для обчислення кореня ξ із заданою точністю ε .

Можна припустити, що метод дихотомії буде збігатися швидше, якщо відрізок $[a, b]$ ділити на частини точкою c не навпіл, а пропорційно величинам ординат $f(a)$ і $f(b)$. Це означає, що точку c є сенс знаходити як абсцису точки перетину осі Ox з прямою, що проходить через точки $(a, f(a))$, $(b, f(b))$:

$$\frac{y - f(a)}{f(b) - f(a)} = \frac{x - a}{b - a}.$$

Поклавши $y = 0$, маємо

$$c = a - \frac{f(a)(b - a)}{f(b) - f(a)}.$$

Метод дихотомії з таким вибором пробної точки c називають методом хорд.

Зазначимо, що в загальному випадку, якщо на функцію $f(x)$ не накладати жодних додаткових обмежень, може виявитися, що метод хорд збігається повільніше за метод половинного поділу.

3.2. Метод простої ітерації

Рівняння,

$$f(x) = 0,$$

запишемо у еквівалентному вигляді,

$$x = \varphi(x). \quad (3.4)$$

Для знаходження кореня рівняння (3.4) побудуємо ітераційний процес, який називають методом простої ітерації:

$$x_{k+1} = \varphi(x_k), \quad k = 0, 1, 2, \dots, \quad (3.5)$$

де x_0 — деяке початкове наближення.

Теорема 3.1 *Нехай функція $\varphi(x)$ визначена та диференційована на відрізку $[a, b]$. Якщо виконуються умови:*

$$\varphi(x) \in [a, b], \quad \forall x \in [a, b], \quad (3.6)$$

$$\exists q : |\varphi'(x)| \leq q < 1 \quad \forall x \in (a, b), \quad (3.7)$$

то для будь-якого початкового наближення $x_0 \in [a, b]$ послідовність $\{x_k\}$, яка визначається методом простої ітерації (3.5), збігається до єдиного кореня $\xi \in [a, b]$ рівняння (3.4). При цьому справедливі такі оцінки:

$$|\xi - x_k| \leq \frac{q}{1 - q} |x_k - x_{k-1}|, \quad (3.8)$$

$$|\xi - x_k| \leq \frac{q^k}{1 - q} |x_1 - x_0|. \quad (3.9)$$

Доведення. Нехай $x_0 \in [a, b]$. З (3.6) випливає, що $x_1 = \varphi(x_0) \in [a, b]$. За індукцією отримуємо, що всі члени послідовності $\{x_k\}$ належать відрізку $[a, b]$. З рівності (3.5) віднімемо рівність $x_k = \varphi(x_{k-1})$ і до правої частини отриманої рівності

$$x_{k+1} - x_k = \varphi(x_k) - \varphi(x_{k-1})$$

застосуємо формулу Лагранжа, згідно з якою на інтервалі (x_{k-1}, x_k) , знайдеться така точка ϑ_k , що

$$\varphi(x_k) - \varphi(x_{k-1}) = \varphi'(\vartheta_k)(x_k - x_{k-1}).$$

Отже,

$$x_{k+1} - x_k = \varphi'(\vartheta_k)(x_k - x_{k-1}),$$

і, в силу умови (3.7), справедлива нерівність

$$|x_{k+1} - x_k| \leq q|x_k - x_{k-1}|. \quad (3.10)$$

На основі (3.10) отримуємо нерівність

$$|x_{k+i} - x_{k+i-1}| \leq q^i |x_k - x_{k-1}| \quad \forall i, k \in N,$$

зокрема

$$|x_k - x_{k-1}| \leq q^{k-1} |x_1 - x_0|. \quad (3.11)$$

Тому,

$$|x_{k+m} - x_k| \leq |x_{k+m} - x_{k+m-1}| + |x_{k+m-1} - x_{k+m-2}| + \dots$$

$$\begin{aligned} \dots + |x_{k+2} - x_{k+1}| + |x_{k+1} - x_k| &\leq (q^m + q^{m-1} + \dots + q^2 + q) |x_k - x_{k-1}| = \\ &= \frac{q - q^{m+1}}{1 - q} |x_k - x_{k-1}|, \end{aligned}$$

тобто,

$$|x_{k+m} - x_k| \leq \frac{q - q^{m+1}}{1 - q} |x_k - x_{k-1}|. \quad (3.12)$$

З (3.11) та (3.12) отримуємо

$$|x_{k+m} - x_k| \leq \frac{q^k}{1 - q} (1 - q^m) |x_1 - x_0|. \quad (3.13)$$

Оскільки права частина нерівності (3.13) при фіксованому m і $k \rightarrow \infty$ збігається до нуля, послідовність $\{x_k\}$ є фундаментальною, отже, збігається до

певної межі ξ , причому $\xi \in [a, b]$, бо всі члени послідовності належать замкненому проміжку $[a, b]$. Так як функція, що диференціюється безперервна, то, розглядаючи межу рівності (3.5), знаходимо, що ξ — корінь рівняння (3.4).

Припустимо, що є ще один корінь $\tilde{\xi} \in [a, b]$ рівняння (3.4). Тоді

$$\xi - \tilde{\xi} = \varphi(\xi) - \varphi(\tilde{\xi}),$$

і, за формулою Лагранжа,

$$\xi - \tilde{\xi} = \varphi'(\vartheta)(\xi - \tilde{\xi}).$$

Остання ж рівність можлива лише за $\xi = \tilde{\xi}$, адже за умовою теореми $\varphi'(\vartheta)$ не може дорівнювати одиниці.

Переходячи до межі при $m \rightarrow \infty$ в нерівностях (3.12) і (3.13), отримуємо оцінки (3.8) і (3.9), відповідно. $\square$

У практичних обчисленнях оцінку (3.8) використовують для завершення ітераційного процесу (3.5), а оцінку (3.9) — для попереднього визначення числа ітерацій, достатніх для обчислення кореня із заданою точністю.

Наприкінці зазначимо, що перехід від рівняння,

$$f(x) = 0,$$

до рівняння,

$$x = \varphi(x),$$

можна здійснити так:

$$x = x - \lambda f(x),$$

де параметр λ вибираємо таким чином, щоб у потрібній області похідна від функції $\varphi(x)$,

$$\varphi'(x) = 1 - \lambda f'(x),$$

була за модулем менше 1, тобто, щоб виконувалася умова (3.7) збіжності простої ітерації.

Приклад 3.2 Нехай

$$0 < \alpha \leq f'(x) \leq \beta < \infty \quad \forall x \in [a, b].$$

Тоді

$$1 - \lambda\beta \leq \varphi'(x) \leq 1 - \lambda\alpha$$

і

$$|\varphi'(x)| \leq q(\lambda) = \max\{|1 - \lambda\alpha|, |1 - \lambda\beta|\}$$

для будь-якого $x \in [a, b]$. Легко встановити, що $q(\lambda) < 1$ при $\lambda \in (0, 2/\beta)$.
Оптимальне значення параметра λ обчислимо з рівняння

$$-(1 - \lambda\beta) = 1 - \lambda\alpha.$$

Тоді

$$\lambda_{opt} = \frac{2}{\alpha + \beta}$$

і

$$q(\lambda_{opt}) = \frac{\beta - \alpha}{\alpha + \beta}.$$

3.3. Метод Ньютона

Припустимо, що в деякій околиці кореня функція $f(x)$ строго зростає і випукла вниз. Виберемо початкове наближення x_0 досить близько до кореня ξ . Запишемо рівняння щодо графіку функції $y = f(x)$ у точці x_0 :

$$y = f(x_0) + f'(x_0)(x - x_0).$$

В якості наступного наближення до кореня візьмемо точку перетину цієї дотичної з віссю Ox :

$$0 = f(x_0) + f'(x_0)(x_1 - x_0).$$

Тоді

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

За наближенням x_1 обчислюємо наближення x_2 , і т.д. Отриманий ітераційний процес називається методом Ньютона:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots, \quad (3.14)$$

де x_0 — початкове наближення.

Теорема 3.2 *Нехай на відрізку $[a, b]$ функція $f(x)$ має першу та другу похідні постійного знака та*

$$f(a)f(b) < 0. \quad (3.15)$$

Якщо початкове наближення x_0 вибрано $[a, b]$ так, що

$$f(x_0)f''(x_0) > 0, \quad (3.16)$$

то послідовність $\{x_k\}$, що визначається методом Ньютона (3.14), монотонно збігається до кореня $\xi \in (a, b)$ рівняння $f(x) = 0$. Крім того,

$$|\xi - x_{k+1}| \leq \frac{\beta}{2\alpha} |\xi - x_k|^2, \quad (3.17)$$

$$|\xi - x_{k+1}| \leq \frac{\beta}{2\alpha} |x_{k+1} - x_k|^2, \quad (3.18)$$

де α і β такі константи, що

$$0 < \alpha \leq |f'(x)|, \quad |f''(x)| \leq \beta < \infty \quad \forall x \in [a, b].$$

Доведення. Спочатку зауважимо, що умови теореми забезпечують існування єдиного кореня $\xi \in (a, b)$. Припустимо для визначеності, що $f'(x) > 0$, $f''(x) > 0$ для будь-якого $x \in [a, b]$ і $f(a) < 0$, $f(b) > 0$. У цьому випадку за x_0 можна взяти будь-яку точку з $(\xi, b]$. Очевидно, що $f(x_0) > 0$.

Запишемо формулу Тейлора для функції $f(\xi)$ у точці x_0 :

$$f(\xi) = f(x_0) + f'(x_0)(\xi - x_0) + \frac{f''(\xi_0)}{2}(\xi - x_0)^2 = 0,$$

де $\xi_0 \in (a, b)$. Звідси слідує, що

$$f(x_0) + f'(x_0)(\xi - x_0) < 0.$$

Значить,

$$\xi < x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} < x_0.$$

Припустимо, що $\xi < x_k$ і $f(x_k) > 0$. Запишемо формулу Тейлора функції $f(\xi)$ у точці x_k :

$$f(\xi) = f(x_k) + f'(x_k)(\xi - x_k) + \frac{f''(\xi_k)}{2}(\xi - x_k)^2 = 0, \quad (3.19)$$

де $\xi_k \in (a, b)$. Звідси слідує що

$$f(x_k) + f'(x_k)(\xi - x_k) < 0.$$

Значить,

$$\xi < x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} < x_k \quad \forall k \in N.$$

Таким чином, послідовність $\{x_k\}$, що формується методом Ньютона (3.14), обмежена знизу і монотонно зменшується. Значить,

$$\lim_{k \rightarrow \infty} x_k = \xi.$$

З (3.14) отримуємо

$$f(x_k) + f'(x_k)(x_{k+1} - x_k) = 0. \quad (3.20)$$

Прирівнюючи ліві частини рівностей (3.19) та (3.20), отримуємо

$$f(x_k) + f'(x_k)(\xi - x_k) + \frac{f''(\xi_k)}{2}(\xi - x_k)^2 = f(x_k) + f'(x_k)(x_{k+1} - x_k).$$

Значить,

$$\xi - x_{k+1} = -\frac{f''(\xi_k)}{2f'(x_k)}(\xi - x_k)^2.$$

Звідси відразу випливає оцінка (3.17).

Запишемо формулу Тейлора для функції $f(x_{k+1})$ у точці x_k

$$f(x_{k+1}) = f(x_k) + f'(x_k)(x_{k+1} - x_k) + \frac{f''(\tilde{\xi}_k)}{2}(x_{k+1} - x_k)^2,$$

де $\tilde{\xi}_k \in (a, b)$. Звідси, враховуючи (3.20), отримуємо

$$f(x_{k+1}) = \frac{f''(\tilde{\xi}_k)}{2}(x_{k+1} - x_k)^2.$$

Тому,

$$|f(x_{k+1})| \leq \frac{\beta}{2}(x_{k+1} - x_k)^2. \quad (3.21)$$

За формулою Лагранжа

$$f(\xi) - f(x_{k+1}) = f'(\tau_{k+1})(\xi - x_{k+1}).$$

Значить,

$$|\xi - x_{k+1}| \leq \frac{1}{\alpha}|f(x_{k+1})|. \quad (3.22)$$

Нарешті, з (3.21) і (3.22) маємо оцінку (3.18). $\square$

Щоб об'єктивніше судити про швидкість збіжності ітераційних процесів, вводять такі поняття.

Говорять, що послідовність $\{x_k\}$ збігається до ξ принаймні з p -м порядком (ітераційний процес має принаймні p -й порядок швидкості збіжності), якщо існують такі константи $C > 0$, $p \geq 1$, $\nu \in (0, 1)$ і $k_0 \in N$, що

$$|\xi - x_{k+1}| \leq C|\xi - x_k|^p \quad (3.23)$$

або

$$|x_{k+1} - x_k| \leq C|x_k - x_{k-1}|^p \quad (3.24)$$

або

$$|\xi - x_k| \leq C\nu^{kp} \quad (3.25)$$

для всіх $k \geq k_0$.

Якщо $p = 1$, то кажуть, що ітераційний процес збігається лінійно, чи має перший порядок швидкості збіжності. З (3.10) випливає, що проста ітерація збігається лінійно. До лінійної збіжності застосовують також термін "збіжність зі швидкістю геометричної прогресії".

Якщо $p = 2$, то кажуть, що ітераційний процес збігається квадратично, або має другий порядок швидкості збіжності. З (3.17) випливає, що метод Ньютона має другий порядок швидкості збіжності.

3.4. Завдання до лабораторної роботи

Завдання

1. Визначити кількість коренів заданого нелінійного рівняння.
2. Знайти з точністю $\varepsilon = 10^{-6}$ найменший за модулем корінь заданого нелінійного рівняння методом простої ітерації, методом Ньютона (порівняти кількість ітерацій витрачене методами) та за допомогою функції `fsolve()` пакета Octave.
3. За результатами обчислювального експерименту знайти фактичні показники збіжності методу ітерацій та методу Ньютона.

Варіанти завдань

Вар.	Рівняння
1	$a_i x^2 - \sin(b_j x) - 1 = 0$
2	$e^{a_i x} - b_j x^2 - 2.5 = 0$
3	$\cos(a_i x) - \ln(b_j x) = 0$
4	$\tan(a_i x) - \ln(b_j x) = 0$
5	$\cos(a_i x) - e^{b_j x} = 0$
6	$\sin(a_i x) - b_j^{-x} = 0$
7	$\tan(a_i x) - b_j^x = 0$
8	$a_i x^3 - b_j^x = 0$
9	$a_i x \ln x - \cos(b_j x) = 0$
10	$a_i x^2 \ln x - \sin(b_j x) = 0$

Варіант — номер прізвища студента в журналі академічної групи. $a_i = 1 + 0.1 * i$, $b_j = 2 + 0.1 * j$, де i і j — дві останні цифри в номері залікової книжки.

Звіт з виконання лабораторної роботи повинен містити:

1. опис алгоритмів;
2. текст програм мовою пакету Octave;
3. результати обчислень;
4. висновки.

4. МЕТОДИ РОЗВ'ЯЗАННЯ СИСТЕМ НЕЛІНІЙНИХ РІВНЯНЬ

4.1. Метод ітерацій

Нехай потрібно розв'язати систему рівнянь,

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0, \\ f_2(x_1, x_2, \dots, x_n) = 0, \\ \dots \\ f_n(x_1, x_2, \dots, x_n) = 0, \end{cases} \quad (4.1)$$

де f_i ($i = \overline{1, n}$) — задані нелінійні дійсні функції дійсних змінних. Приймаючи позначення,

$$x = [x_1, x_2, \dots, x_n]^T, F(x) = [f_1(x), f_2(x), \dots, f_n(x)]^T,$$

систему (4.1) можна записати у вигляді:

$$F(x) = 0, \quad (4.2)$$

де $F : R^n \rightarrow R^n$ — векторна функція векторного аргументу.

Приведемо систему (4.1) до еквівалентної вигляду,

$$\begin{cases} x_1 = \varphi_1(x_1, x_2, \dots, x_n) = 0, \\ x_2 = \varphi_2(x_1, x_2, \dots, x_n) = 0, \\ \dots \\ x_n = \varphi_n(x_1, x_2, \dots, x_n) = 0, \end{cases} \quad (4.3)$$

або, у компактній формі:

$$x = \Phi(x), \quad (4.4)$$

де $\Phi(x) = [\varphi_1(x), \varphi_2(x), \dots, \varphi_n(x)]^T$.

Метод простої ітерації розв'язання задачі (4.4) визначається рекурентною рівністю

$$x_{k+1} = \Phi(x_k), \quad k = 0, 1, 2, \dots, \quad (4.5)$$

де x_0 — початкове наближення. Справедлива наступна теорема.

Теорема 4.1 Нехай функція $\Phi(x)$ і замкнута множина $M \subseteq D(\Phi) \subseteq R^n$ такі, що:

$$\Phi(x) \in M, \quad \forall x \in M; \quad (4.6)$$

$$\exists q < 1 : \|\Phi(x) - \Phi(\tilde{x})\| \leq q\|x - \tilde{x}\| \quad \forall x, \tilde{x} \in M. \quad (4.7)$$

Тоді, для будь-якого початкового наближення $x_0 \in M$ послідовність $\{x_k\}$, що визначається методом простої ітерації (4.5), збігається до єдиного кореня $\xi \in M$ рівняння (4.4) та мають місце оцінки:

$$\|\xi - x_k\| \leq \frac{q}{1-q} \|x_k - x_{k-1}\| \leq \frac{q^k}{1-q} \|x_1 - x_0\|. \quad (4.8)$$

Доведення. Доказ цієї теореми майже повністю повторює доказ теореми 3.1 (див., наприклад, [?]). $\square$

4.2. Метод Ньютона

Щоб застосувати метод простої ітерації (4.5) до системи (4.2), її треба привести до вигляду (4.4), наприклад, так:

$$x = x - AF(x), \quad (4.9)$$

де $A \in R^{n \times n}$ — невироджена матриця, яку треба підібрати так, щоб вектор-функція $\Phi(x) = x - AF(x)$ мала потрібні властивості.

Нехай $\{A_k\}$ — деяка послідовність невироджених $n \times n$ -матриць. Очевидно, що задачі

$$x = x - A_k F(x), \quad k = 0, 1, 2, \dots \quad (4.10)$$

мають ті самі розв'язки, як і задача (4.2). Для наближеного знаходження розв'язку задачі (4.10) формально визначимо ітераційний процес:

$$x_k = x_k - A_k F(x_k), \quad k = 0, 1, 2, \dots \quad (4.11)$$

Зауважимо, що якщо $A_k \equiv A$, то ітераційний процес (4.11) — це метод ітерацій для рівняння (4.9). Якщо ж A_k різні за різних k , то формула (4.11)

визначає велике сімейство ітераційних методів.

Покладемо $A_k = (F'(x_k))^{-1}$, де

$$F'(x) = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \cdots & \frac{\partial f_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(x)}{\partial x_1} & \frac{\partial f_n(x)}{\partial x_2} & \cdots & \frac{\partial f_n(x)}{\partial x_n} \end{bmatrix}$$

— матриця Якобі для векторної функції $F(x)$. Підставивши цю матрицю A_k в (4.11), отримаємо явну формулу методу Ньютона,

$$x_{k+1} = x_k - (F'(x_k))^{-1} F(x_k), \quad (4.12)$$

яка узагальнює на багатовимірний випадок скалярний метод Ньютона (3.14). Формулу (4.12) можна записати так

$$F'(x_k)(x_{k+1} - x_k) = -F(x_k). \quad (4.13)$$

Позначивши $p_k = x_{k+1} - x_k$, отримуємо неявну формулу методу Ньютона:

$$\begin{aligned} F'(x_k)p_k &= -F(x_k), \\ x_{k+1} &= x_k + p_k, \quad k = 0, 1, 2, \dots \end{aligned} \quad (4.14)$$

Порівнюючи формулу (4.13) з формальним розкладанням функції $F(x)$ у ряд Тейлора в точці x_k ,

$$F(x) = F(x_k) + F'(x_k)(x - x_k) + \frac{1}{2!}F''(x_k)(x - x_k)^2 + \dots,$$

бачимо, що послідовність $\{x_k\}$ у методі Ньютона отримуємо в результаті заміни на кожному кроці нелінійного рівняння $F(x) = 0$ лінійним рівнянням,

$$F(x_k) + F'(x_k)(x - x_k) = 0,$$

тобто, покроковою лінеаризацією. Як наслідок цього факту, можна розраховувати, що при достатній гладкості $F(x)$ і досить хорошему початковому наближенні x_0 , збіжність методу Ньютона буде квадратичною і в багатови-

мірному випадку. Є ряд теорем, що встановлюють це за тих чи інших припущень (див., наприклад, [4? ?]).

4.3. Завдання до лабораторної роботи

Нехай $\{x^0, \lambda^0\}$ — початкове наближення до деякої власної пари матриці $A = A^T \in R^{n \times n}$.

Для уточнення цієї власної пари застосуємо метод Ньютона.

Для цього, задачу на власні значення, знайти такі $x \in R^n$ ($x \neq 0$) та $\lambda \in R$, що

$$Ax = \lambda x, \quad (4.15)$$

запишемо у вигляді системи нелінійних рівнянь

$$F(y) = 0, \quad (4.16)$$

де

$$y = \begin{bmatrix} x \\ \lambda \end{bmatrix},$$

$$F \left(\begin{bmatrix} x \\ \lambda \end{bmatrix} \right) = \begin{bmatrix} Ax - \lambda x \\ \frac{1}{2}(1 - x^T x) \end{bmatrix}. \quad (4.17)$$

Тоді,

$$F' \left(\begin{bmatrix} x \\ \lambda \end{bmatrix} \right) = \begin{bmatrix} A - \lambda I & -x \\ -x^T & 0 \end{bmatrix}. \quad (4.18)$$

Тепер метод Ньютона для уточнення власної пари $\{x^0, \lambda^0\}$ визначимо як наступний ітераційний процес:

$$\begin{bmatrix} x^{k+1} \\ \lambda^{k+1} \end{bmatrix} = \begin{bmatrix} x^k \\ \lambda^k \end{bmatrix} - \left(F' \left(\begin{bmatrix} x^k \\ \lambda^k \end{bmatrix} \right) \right)^{-1} F \left(\begin{bmatrix} x^k \\ \lambda^k \end{bmatrix} \right), k = 0, 1, \dots \quad (4.19)$$

k -й крок методу Ньютона (4.19) зручно виконувати так:

1. знаходимо розв'язок $[y^k, \mu^k]^T$ системи

$$\begin{bmatrix} A - \lambda^k I & -x^k \\ -(x^k)^T & 0 \end{bmatrix} \begin{bmatrix} y^k \\ \mu^k \end{bmatrix} = \begin{bmatrix} Ax^k - \lambda^k x^k \\ \frac{1}{2}(1 - (x^k)^T x^k) \end{bmatrix}; \quad (4.20)$$

2. припускаємо

$$\begin{cases} x^{k+1} = x^k - y^k, \\ \lambda^{k+1} = \lambda^k - \mu^k. \end{cases} \quad (4.21)$$

Завдання

1. Для заданої матриці A степеневим методом знайти найбільше за модулем власне значення λ^0 та відповідний власний вектор x^0 з точністю ε_0 ;
2. для уточнення власної пари $\{x^0, \lambda^0\}$ застосувати метод Ньютона. Ітерації методу Ньютона продовжувати, поки не виконується нерівність,

$$\|Ax^{k+1} - \lambda^{k+1}x^{k+1}\|_2 < \varepsilon.$$

Варіанти завдань

$$\varepsilon_0 = 10^{-4}, \quad \varepsilon = 10^{-6}.$$

Матриця A визначається так:

$$\begin{aligned} a_{11} &= \alpha, \\ a_{i+1,i+1} &= -a_{i,i}, \quad i = \overline{1, n-1}, \\ a_{ij} &= (i+j)/\alpha, \quad i \neq j, \end{aligned}$$

де $n = 20$, $\alpha = 10+k/10$, k — номер прізвища студента у журналі студентської групи.

До звіту з виконання лабораторної роботи включити:

1. опис алгоритмів, що використовуються;
2. текст програми;
3. результати обчислювального експерименту.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Гаврилюк І.П., Макаров В.Л. Методи обчислень. Підручник. У 2ч. — К.: Вища шк., 1995. — 367 с.
2. Мазманішвілі О.С. Чисельні методи. Навчальний посібник до лабораторних занять. — Харків: НТУ „ХП“, 2010. — 219 с.
3. Попов В.В. Методи обчислень. — К.: Видавничо-поліграфічний центр „Київський університет“, 2012. — 303 с.
4. Цегелик Г. Г. Чисельні методи. — Львів: Світ, 2005. — 407 с.
5. Шахно С. Чисельні методи лінійної алгебри. — Львів: Видавничий центр ЛНУ ім. І. Франка, 2007. — 245 стр.
6. Bai Z., Demmel J., Dongarra J., Ruhe A., and H. van der Vorst. Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide. — SIAM, Philadelphia, PA, USA, 2000. — 410 p.
7. Beilina L., Karchevskii E., Karchevskii M. Numerical linear algebra: theory and applications / Springer, 2017. 457 p.
8. Demmel James W. Applied numerical linear algebra. — SIAM, 1997. — 425 p.
9. Epperson J. F. An Introduction to Numerical Methods and Analysis : Third edition / John Wiley & Sons Inc., 2021. 286 p.
10. Gupta R. K. Numerical methods. Fundamentals and applications / Cambridge University Press, 2019. 830 p.
11. Lyche T. Numerical linear algebra and matrix factorizations / Springer, 2020. 376 p.
12. Saad Y. Iterative Methods for Sparse Linear Systems, Second Edition. — SIAM, 2003. — 350 p.
13. Vorst H. A. Iterative Krylov Methods for Large Linear systems. — Cambridge University Press, Cambridge, 2003. — 221 p.
14. <http://netlib.org> — Netlib is a collection of mathematical software, papers, and databases.
15. <http://www.nag.com/> — Numerical Algorithms Group.
16. <http://www.gnu.org/software/gsl/> — GNU Association GSL Sci. Library.

ДЛЯ НОТАТОК

ДЛЯ НОТАТОК

ДЛЯ НОТАТОК

Навчальне видання

МЕТОДИ ОБЧИСЛЕНЬ

Частина II

Методичні вказівки до лабораторних робіт
для здобувачів першого (бакалаврського) рівня
вищої освіти спеціальності 113 Прикладна математика

Укладач

Вербіцький Віктор Васильович

В авторській редакції

Підписано до друку 13.09.2024 р. Формат 60x84/16.

Папір офсетний. Цифровий друк.

Ум. друк. арк. 2,56. Наклад 30. Зам. № 1024-29.

Віддруковано з готового оригінал-макета.

Видавництво та друк: ОЛДІ+

65101, Україна, м. Одеса, вул. Інглезі, 6/1

Свідоцтво ДК № 7642 від 29.07.2022 р.

Тел.: +38 (098) 559-45-45,

+38 (095) 559-45-45, +38 (093) 559-45-45

Для листування: 65101, Україна, м. Одеса, вул. Інглезі, 6/1

E-mail: office@oldiplus.ua