

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 I

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

ОДЕСА
2023

УДК 519.612.2:519.612.4(076.5)

М545

Укладачі:

В. В. Вербіцький, кандидат фізико-математичних наук, доцент, доцент кафедри оптимального керування і економічної кібернетики;

Є. В. Платонова, старший викладач кафедри оптимального керування і економічної кібернетики;

Рецензенти:

В. В. Мороз, кандидат технічних наук, професор кафедри оптимального керування і економічної кібернетики Одеського національного університету імені І. І. Мечникова;

І. М. Шпінарева, кандидат фізико-математичних наук, доцент, доцент кафедри математичного забезпечення комп'ютерних систем імені І. І. Мечникова.

*Рекомендовано вченою радою факультету математики,
фізики та інформаційних технологій ОНУ імені І. І. Мечникова.
Протокол 2 від 30 жовтня 2023 р.*

Методи обчислень. Ч. І.: метод. вказівки до лаб.
М545 робіт для здобувачів першого (бакалавр.) рівня вищої освіти спеціальності 113 Прикладна математика / уклад.:
В. В. Вербіцький, Є. В. Платонова. – Одеса : Видавництво Олді+, 2023. – 52 с.

Методичні вказівки включають теми змістових модулів, прямі методи розв'язання систем лінійних алгебраїчних рівнянь та ітераційні методи алгебри, а саме: прямі методи розв'язання систем лінійних алгебраїчних рівнянь, лінійна задача найменших квадратів, класичні ітераційні методи та методи підпросторів Крилова побудови наближених розв'язків лінійних систем. До кожної теми додано необхідний теоретичний матеріал та завдання до лабораторних робіт.

УДК 519.612.2:519.612.4

ЗМІСТ

Вступ	3
1 Прямі методи	5
1.1 LU -розкладання	5
1.2 PLU -розкладання	7
1.3 Метод Гауса з частковим вибором головного елемента	11
1.4 Розкладання Холевського	12
1.5 Ітераційне уточнення	13
1.6 Завдання до лабораторної роботи	14
2 Лінійна задача найменших квадратів	17
2.1 QR -розкладання	18
2.2 Розв'язання ЛЗНК за допомогою QR -розкладання	19
2.3 Розв'язання ЛЗНК за допомогою SVD -розкладання	20
2.4 Завдання до лабораторної роботи	21
3 Класичні ітераційні методи	23
3.1 Метод простої ітерації	23
3.2 Показник збіжності ітераційного процесу	26
3.3 Метод Річардсона	27
3.4 Завдання до лабораторної роботи	33
4 Ітераційні методи підростору Крилова	35
4.1 Метод Арнольді	36
4.2 Метод узагальненої мінімізації нев'язки	39
4.3 Метод Ланцоша	40
4.4 Метод спряжених градієнтів	44
4.5 Передобумовлення	47
4.6 Завдання до лабораторної роботи	48
Список рекомендованої літератури	50

ВСТУП

Методичні вказівки присвячені чисельним методам лінійної алгебри. Розглянуто такі теми: прямі методи розв’язання систем лінійних алгебраїчних рівнянь (СЛАР), лінійна задача найменших квадратів (ЛЗНК), класичні ітераційні методи та методи підпросторів Крилова побудови наближених розв’язків СЛАР. 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 угоди 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 всіх розглянутих числових методів, які у учбових цілях 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 чисельними методами. 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. ПРЯМІ МЕТОДИ

1.1. LU -розкладання

Головною провідною підматрицею $A^{(k)}$ матриці $A \in R^{n \times n}$ називається матриця, яка утворюється з матриці A викресленням всіх однойменних рядків та стовпців починаючи з $k + 1$. Так

$$\begin{aligned} A^{(1)} &= [a_{11}], \\ A^{(2)} &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \\ A^{(3)} &= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \\ &\vdots \\ A^{(n)} &= A. \end{aligned}$$

Теорема 1.1 Нехай усі головні провідні підматриці матриці $A \in R^{n \times n}$ не вироджені, тобто,

$$\det A^{(k)} \neq 0, \quad k = \overline{1, n}, \quad (1.1)$$

тоді матриця A єдино представима у вигляді

$$A = LU, \quad (1.2)$$

де L — нижня трикутна матриця з одиницями на головній діагоналі, U — верхня трикутна матриця, діагональні елементи якої обчислюються за формулою

$$u_{kk} = \frac{\det A^{(k)}}{\det A^{(k-1)}}, \quad k = \overline{1, n}, \quad (1.3)$$

де $\det A^{(0)} = 1$.

Доведення теореми можна знайти у навчальних посібниках[3–5].

Нижче наведено алгоритм обчислення LU -розкладання. В цьому алгоритмі множники L і U зберігаються на місці вихідної матриці A .

Алгоритм 1.1 LU -розкладання

1. For $k = 1$ To $n - 1$

2. *If* $a_{kk} = 0$ *Then*
3. $\text{print}(\text{"Деление на 0."}); \text{Break}$
4. *End If*
5. *For* $i = k + 1$ *To* n
6. $N = -\frac{a_{ik}}{a_{kk}}$
7. $a_{ik} = -N$
8. *For* $j = k + 1$ *To* n
9. $a_{ij} = a_{ij} + Na_{kj}$
10. *End For*
11. *End For*
12. *End For*

Якщо для матриці A знайдено LU -розкладання, то розв'язування системи

$$Ax = b \tag{1.4}$$

зводиться до пошуку розв'язків двох систем з трикутними матрицями:

$$Ux = y, \tag{1.5}$$

$$Ly = b. \tag{1.6}$$

Обчисливши розв'язок y системи (1.6), знаходимо розв'язок системи (1.5).

Алгоритми розв'язання систем з трикутними матрицями приведено нижче.

Алгоритм 1.2 *Алгоритм розв'язання системи $Ly = b$.*

1. $y_1 = b_1$
2. *For* $i = 2$ *To* n
3. $y_i = b_i - \sum_{j=1}^{i-1} l_{ij}y_j$
4. *End For*

Алгоритм 1.3 *Алгоритм розв'язання системи $Ux = y$.*

1. $x_n = y_n$

2. *For* $i = n - 1$ *To* 1 *Step* $- 1$
3. $x_i = (y_i - \sum_{j=i+1}^n u_{ij}x_j)/u_{ii}$
4. *End For*

1.2. PLU-розкладання

Нехай $x \in R^n$ и $x_k \neq 0$. Якщо

$$\tau = [\underbrace{0, \dots, 0}_k, \tau_{k+1}, \dots, \tau_n]^T, \quad \tau_i = -\frac{x_i}{x_k}, \quad i = \overline{k+1, n}$$

та

$$N_k = I + \tau e_k^T, \quad (1.7)$$

то

$$N_k x = \begin{bmatrix} 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & \tau_{k+1} & 1 & \dots & 0 \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \tau_n & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ x_{k+1} \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Матриця N_k називається *матрицею виключення* або матрицею *перетворення Гауса*.

Елементарною матрицею перестановок називається матриця P^{i, γ_i} , яка утворюється з одиничної матриці перестановкою i -го та γ_i -го рядків.

Множення матриці P^{i, γ_i} на вектор x міняє місцями i -у та γ_i -у компоненти вектора x .

Множення матриці P^{i, γ_i} на матрицю A зліва (праворуч) міняє місцями у матриці A i -й та γ_i -й рядки (стовпці).

Нехай $\gamma = (\gamma_1, \dots, \gamma_n)$ — деяка перестановка натуральних чисел $1, \dots, n$. *Матрицею перестановок* P^γ називається матриця у якій i -й рядок — це γ_i -й рядок одиничної матриці.

Будь-яка матриця перестановок може бути подана у вигляді добутку скінченного числа елементарних матриць перестановок. Очевидно, що добуток будь-якого числа матриць перестановок є матриця перестановок.

Якщо матрицю P^γ помножити на вектор x , то у векторі x i -тою компонентою стане γ_i -а компонента.

Якщо матрицю A помножити зліва на P^γ (справа на $(P^\gamma)^T$), то γ_i -й рядок стане i -м (γ_i -й стовпець стане i -м).

Приклад 1.1 Нехай $\gamma = (2, 3, 4, 1)$, тоді

$$P^\gamma = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

та

$$P^\gamma A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \\ a_{11} & a_{12} & a_{13} & a_{14} \end{bmatrix}.$$

Теорема 1.2 Будь-яка невідроджена матриця $A \in R^{n \times n}$ може бути представлена у вигляді

$$A = PLU, \quad (1.8)$$

де P — матриця перестановок, L — нижня трикутна матриця з одиницями на головній діагоналі, U — невідроджена верхня трикутна матриця.

Доведення. Покладемо

$$A_0 = A.$$

Знайдемо найбільший за модулем елемент у першому стовпці матриці A_0 :

$$|a_{\gamma_1 1}^{(0)}| = \max_{1 \leq i \leq n} |a_{i1}^{(0)}|.$$

Якщо таких елементів кілька, вибираємо будь-який із них. У матриці A_0 міняємо місцями перший рядок і рядок з номером γ_1 , який містить найбільший за модулем елемент. Для цього множимо зліва матрицю A_0 на елементарну матрицю перестановок $P^{1\gamma_1}$.

У першому стовпці отриманої матриці обнуляємо всі елементи крім пер-

шого:

$$A_1 = N_1 P^{1\gamma_1} A_0.$$

Тут N_1 — матриця виключення.

На k -му кроці знаходимо найбільший за модулем елемент серед $n - k + 1$ елементів k -го стовпця матриці A_{k-1} :

$$|a_{\gamma_k k}^{(k-1)}| = \max_{k \leq i \leq n} |a_{ik}^{(k-1)}|.$$

Якщо таких елементів кілька, вибираємо будь-який із них. У матриці A_{k-1} міняємо місцями k -й рядок і рядок з номером γ_k , який містить знайдений найбільший за модулем елемент. Для цього множимо зліва матрицю A_{k-1} на елементарну матрицю перестановок $P^{k\gamma_k}$. У k -му стовпці отриманої матриці обнулюємо всі елементи починаючи з $k + 1$ -го:

$$A_k = N_k P^{k\gamma_k} A_{k-1}.$$

Тут N_k — матриця виключення.

Після $n - 1$ кроку отримуємо верхню трикутну матрицю

$$U = N_{n-1} P^{n-1, \gamma_{n-1}} \dots N_2 P^{2\gamma_2} N_1 P^{1\gamma_1} A. \quad (1.9)$$

Зауважимо, що у точній арифметиці цей алгоритм перерватися не може. Справді, припустимо, що на k -му кроці елементи k -го стовпця матриці A_{k-1} починаючи з k -го елемента дорівнюють нулю (у цьому випадку неможливо визначити матрицю виключення N_k). Тоді

$$0 = \det(A_{k-1}) = \det(N_{k-1} P^{k-1, \gamma_{k-1}} \dots N_2 P^{2\gamma_2} N_1 P^{1\gamma_1} A) = \pm \det(A) \neq 0.$$

Отримали протиріччя.

Відмітимо, що

$$\begin{aligned} (N_{n-1} P^{n-1, \gamma_{n-1}} \dots N_2 P^{2\gamma_2} N_1 P^{1\gamma_1})^{-1} &= P^{1\gamma_1} N_1^{-1} P^{2\gamma_2} N_2^{-1} \dots P^{n-1, \gamma_{n-1}} N_{n-1}^{-1} = \\ &= (P^{1\gamma_1} P^{2\gamma_2} \dots P^{n-1, \gamma_{n-1}}) (P^{n-1, \gamma_{n-1}} \dots P^{2\gamma_2} N_1^{-1} P^{2\gamma_2} \dots P^{n-1, \gamma_{n-1}}) \times \end{aligned}$$

$$\times (P^{n-1, \gamma_{n-1}} \dots P^{3\gamma_3} N_2^{-1} P^{3\gamma_3} \dots P^{n-1, \gamma_{n-1}}) \dots N_{n-1}^{-1} = PL,$$

де $P = P^{1\gamma_1} P^{2\gamma_2} \dots P^{n-1, \gamma_{n-1}}$ — матриця перестановок, L — нижня трикутна матриця з одиницями на головній діагоналі, бо

$$L = \tilde{N}_1 \tilde{N}_2 \dots \tilde{N}_{n-2} N_{n-1}^{-1},$$

де

$$\tilde{N}_k = P^{n-1, \gamma_{n-1}} \dots P^{k+1, \gamma_{k+1}} N_k^{-1} P^{k+1, \gamma_{k+1}} \dots P^{n-1, \gamma_{n-1}}, \quad k = \overline{1, n-2},$$

— деякі матриці виключення. Нарешті, помножуючи ліворуч матричну рівність (1.9) на

$$(N_{n-1} P^{n-1, \gamma_{n-1}} \dots N_2 P^{2\gamma_2} N_1 P^{1\gamma_1})^{-1},$$

отримуємо розкладання (1.8). $\square$

Якщо отримано розкладання матриці $A = PLU$, систему $Ax = b$ можна замінити двома системами

$$\begin{cases} Ly = P^T b, \\ Ux = y, \end{cases}$$

які легко розв'язати, оскільки їх матриці трикутні.

Оскільки

$$P = P^{1\gamma_1} P^{2\gamma_2} \dots P^{n-1, \gamma_{n-1}},$$

то

$$P^T = P^{n-1, \gamma_{n-1}} \dots P^{2\gamma_2} P^{1\gamma_1}.$$

Тоді перестановку γ , якою визначається матриця P^T можна визначити так:

$$p = [1, 2, \dots, n]^T, \quad p = P^{k\gamma_k} p, \quad k = \overline{1, n-1}, \quad \gamma = p.$$

Таким чином, щоб визначити перестановку γ треба на k -му кроці розкладання поміняти місцями k -у та γ_k -у компоненти вектора p . Зауважимо, що $(P^T b)(i) = b(p(i))$, $i = \overline{1, n}$.

Наведемо алгоритм обчислення PLU -розкладання, в якому множники L і U зберігаються на місці вихідної матриці A ($A = L + U - I$, де I —

одинична матриця), а матриця перестановок P така, що

$$LU = PA.$$

Алгоритм 1.4 PLU -розкладання

1. $p = [1, 2, \dots, n]^T$
2. *For* $k = 1$ *To* $n - 1$
3. $|a_{\gamma_k k}| = \max_{k \leq i \leq n} |a_{ik}|.$
4. $p = P^{k\gamma_k} p$
5. $A = P^{k\gamma_k} A$
6. *If* $a_{kk} = 0$ *Then*
7. $\text{print}(\text{"Ділення на } 0\text{."}); \text{Break}$
8. *End If*
9. *For* $i = k + 1$ *To* n
10. $N = -\frac{a_{ik}}{a_{kk}}$
11. $a_{ik} = -N$
12. *For* $j = k + 1$ *To* n
13. $a_{ij} = a_{ij} + Na_{kj}$
14. *End For*
15. *End For*
16. *End For*

1.3. Метод Гауса з частковим вибором головного елемента

Метод Гауса з частковим вибором головного елемента (GEPP — Gaussian elimination with partial pivoting) розв'язання системи

$$Ax = b \tag{1.10}$$

з невідродженою матрицею A складається з двох етапів.

На першому етапі (прямий хід методу) система рівнянь (1.10) зводиться до системи з верхньотрикутнковою матрицею. Використовуючи наведені ви-

ще позначення, прямий хід методу можна описати так:

$$N_{n-1}P^{n-1,\gamma_{n-1}} \dots N_2P^{2\gamma_2}N_1P^{1\gamma_1}Ax = N_{n-1}P^{n-1,\gamma_{n-1}} \dots N_2P^{2\gamma_2}N_1P^{1\gamma_1}b. \quad (1.11)$$

Зворотний хід методу полягає у знаходженні розв'язку системи (1.11).

1.4. Розкладання Холеського

Теорема 1.3 *Будь-яку симетричну додатно визначену матрицю $A \in R^{n \times n}$ можна єдиним чином подати у вигляді*

$$A = LL^T, \quad (1.12)$$

де L — нижня трикутна матриця з позитивними діагональними елементами.

Різні доведення теореми можна знайти, наприклад, у навчальних посібниках[3–5].

Розглянемо алгоритм побудови розкладання Холеського симетричної додатно визначеної матриці A . Перший крок алгоритму можна отримати з наступних очевидних рівностей:

$$A = \begin{bmatrix} a_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} \sqrt{a_{11}} & 0^T \\ \frac{A_{21}}{\sqrt{a_{11}}} & I \end{bmatrix} \begin{bmatrix} 1 & 0^T \\ 0 & \tilde{A}_{22} \end{bmatrix} \begin{bmatrix} \sqrt{a_{11}} & \frac{A_{21}^T}{\sqrt{a_{11}}} \\ 0 & I \end{bmatrix},$$

де

$$\tilde{A}_{22} = A_{22} - \frac{A_{21}A_{21}^T}{a_{11}}.$$

Зауважимо, що матриця $\tilde{A}_{22}$ (активна підматриця першого кроку) — симетрична та позитивно визначена. Першим стовпцем множника Холеського є вектор $[\sqrt{a_{11}}, \frac{A_{21}^T}{\sqrt{a_{11}}}]^T$. Записуючи аналогічне розкладання для матриці $\tilde{A}_{22}$, отримаємо другий стовпець множника Холеського. Повторюючи n раз описаний вище процес, знайдемо всі n стовпців множника Холеського. Таким чином, алгоритм розкладання Холеського можна записати в такий спосіб.

Алгоритм 1.5 Розкладання Холевського

1. *For* $i = 1$ *To* n
2. *For* $j = 1$ *To* i
3. $l_{ij} = a_{ij}$
4. *End For*
5. *End For*
6. *For* $k = 1$ *To* n
7. $l_{kk} = \sqrt{l_{kk}}$
8. *For* $i = k + 1$ *To* n
9. $l_{ik} = l_{ik}/l_{kk}$
10. *For* $j = k + 1$ *To* i
11. $l_{ij} = l_{ij} - l_{ki}l_{kj}$
12. *End For*
13. *End For*
14. *End For*

Добре відомо, що алгоритм розкладання Холевського стійкий у арифметиці з плаваючою точкою [3–5, 9, 10].

1.5. Ітераційне уточнення

Через помилки округлення прямий метод знаходить наближений розв'язок x_0 системи

$$Ax = b. \quad (1.13)$$

Покладемо

$$x = x_0 + \Delta x,$$

де x — точний розв'язок системи (1.13). Тоді для визначення Δx отримуємо систему

$$A\Delta x = r_0, \quad (1.14)$$

де $r_0 = b - Ax_0$ — нев'язка наближеного розв'язка x_0 . Розв'язуючи рівняння (1.14) тим самим прямим методом, знаходимо наближене значення Δx , яке позначимо через Δx_0 . Покладемо, $x_1 = x_0 + \Delta x_0$. Продовжуючи цей процес,

отримуємо алгоритм, який називається ітераційним уточненням.

Алгоритм 1.6 *k*-а ітерація ітераційного уточнення

1. $r_k = b - Ax_k$
2. $A\Delta x_k = r_k$
3. $x_{k+1} = x_k + \Delta x_k$

Наступна теорема стверджує, що відносна покомпонентна зворотна помилка для x_1 має максимально можливий порядок малості [8].

Теорема 1.4 *Нехай нев'язка обчислюється зі звичайною точністю i*

$$\|A\|_\infty \|A^{-1}\|_\infty \frac{\max_i (|A| |x_0|)}{\min_i (|A| |x_0|)} \cdot \varepsilon < 1. \quad (1.15)$$

Тоді один крок ітераційного уточнення дає таке наближення $x_1 = x_0 + \Delta x_0$, що

$$(A + \delta A)x_1 = b + \delta b,$$

де

$$\begin{aligned} |\delta a_{ij}| &= O(\varepsilon) |a_{ij}|, \\ |\delta b_i| &= O(\varepsilon) |b_i|, \quad 1 \leq i, j \leq n. \end{aligned}$$

1.6. Завдання до лабораторної роботи

Згідно варіанта завдання мовою пакету MATLAB написати функції:

1. $[A] = \text{InitA}(ob, n)$ для ініціалізації матриці $A \in \mathbf{R}^{n \times n}$ ($ob = 1$ — матриця добре обумовлена, $ob = 0$ — матриця погано обумовлена);
2. $[x] = \text{Solve}(A, b)$ для знаходження наближеного розв'язку СЛАР

$$Ax = b.$$

Провести наступний обчислювальний експеримент. Для добре/погано обумовленої матриці різної розмірності n :

1. Довільно вибрати вектор $x \in R^n$, $x \neq 0$. Обчислити вектор $b = Ax$;
2. Знайти наближений розв'язок $\hat{x}$ модельної задачі

$$Ax = b. \quad (1.16)$$

за допомогою функції $[x] = \text{Solve}(A, b)$;

3. Якщо виконується оцінка,

$$\|A^{-1}\|_{\infty} \|A\|_{\infty} \frac{\max_i (|A| \cdot |\hat{x}|)_i}{\min_i (|A| \cdot |\hat{x}|)_i} \epsilon < 1,$$

виконати один крок ітераційного уточнення, обчислюючи нев'язку $r = b - A\hat{x}$ в машинній арифметиці з тією ж точністю, яка використовувалась для розв'язання СЛАР;

4. перевірити виконання апостеріорної оцінки:

$$\frac{\|\hat{x} - x\|_1}{\|\hat{x}\|_1} \leq \|A^{-1}\|_1 \frac{\|r\|_1}{\|\hat{x}\|_1}.$$

5. знайти наближений розв'язок $\tilde{x}$ модельної задачі (1.16) засобами пакету MATLAB.

Для добре/погано обумовленої матриці заповнити таблицю:

n	$\text{cond}_1(A)$	$\frac{\ \hat{x}-x\ _1}{\ \hat{x}\ _1}$	$\ A^{-1}\ _1 \frac{\ r\ _1}{\ \hat{x}\ _1}$	$\frac{\ \tilde{x}-x\ _1}{\ \tilde{x}\ _1}$
5				
10				
15				
20				
50				
75				
100				
150				
200				

Добре обумовлені матриці (ДОМ)

1. $a_{ii} = 1$, $a_{ij} = 1/n$ ($i \neq j$).
2. Симетрична додатно визначена матриця Кахана розмірності n , яку можна побудувати функцією MATLAB
 $A = \text{gallery}('kms', n, 0.5);$

3. Симетрична додатно визначена матриця розмірності n , що будеться функцією MATLAB

$A = \text{gallery}(\text{'gcdmat'}, n);$

4. Матриця розмірності n спектрального диференціювання Чебишева що будеться функцією MATLAB

$A = \text{gallery}(\text{'chebspec'}, n, 1);$

- 5.

$$a_{ij} = \begin{cases} 1/(i+j-1), & i \leq j, \\ -1/(i+j-1) & i > j. \end{cases}$$

- 6.

$$a_{ij} = \begin{cases} 1, & i = j, \\ 1/n, & i < j, \\ -1/n & i > j. \end{cases}$$

Погано обумовлені матриці (ПОМ)

1. Матриця Гільберта $a_{ij} = 1/(i+j-1)$.
2. Матриця Лоткіна розмірності n , що будеться функцією MATLAB
 $A = \text{gallery}(\text{'lotkin'}, n);$

Варіанти завдань

Варіант	Метод	ДОМ	ПОМ
1	GEPP	4	1
2	LL^T	1	1
3	GEPP	5	1
4	LL^T	2	1
5	GEPP	6	1
6	LL^T	3	1
7	GEPP	4	2
8	GEPP	5	2
9	GEPP	6	2
10	PLU	4	2
11	PLU	5	2
12	PLU	6	2

До звіту з виконання лабораторної роботи включити:

1. опис методу розв'язання СЛАР;
2. текст програми;
3. результати обчислювального експеримента;
4. Графіки залежності відносної похибки наближеного розв'язку від числа обумовленості матриці для добре та погано обумовлених матриць.
5. висновки.

2. ЛІНІЙНА ЗАДАЧА НАЙМЕНЬШИХ КВАДРАТІВ

Розглянемо систему лінійних алгебраїчних рівнянь

$$Ax = b, \quad (2.1)$$

де $A \in R^{m \times n}$ — матриця повного стовпцевого рангу, $b \in R^m$, $m > n$. Оскільки система (2.1) містить більше рівнянь, ніж невідомих, вона перевизначена. Точного розв'язка системи (2.1) може не існувати. Однак, серед наближених розв'язків системи (2.1) має сенс знайти таке, 2-норма нев'язки якого найменша. Приходимо до лінійної задачі найменших квадратів (ЛЗНК):

$$\min_{x \in R^n} \|Ax - b\|_2^2. \quad (2.2)$$

Розв'язок задачі мінімізації (2.2) називають *псевдорозв'язком* перевизначеної системи (2.1).

Оскільки функціонал

$$F(x) = \|Ax - b\|_2^2, \quad (2.3)$$

квадратичний, та його градієнт

$$\nabla F(x) = 2A^T(Ax - b),$$

то необхідна та достатня умова мінімуму функціоналу $F(x)$ має вигляд

$$\nabla F(x) = 2A^T(Ax - b) = 0,$$

тобто,

$$A^T Ax = A^T b. \quad (2.4)$$

Рівняння системи (2.4) називають *нормальними* рівняннями.

Оскільки матриця A системи (2.1) повного стовпцевого рангу, матриця $A^T A$ симетрична і додатно визначена. Отже, система (2.4) має єдиний розв'язок. Цей розв'язок є єдиним розв'язком ЛЗНК (2.2).

Систему (2.4) можна розв'язати, використовуючи, наприклад, розкладання Холевського матриці $A^T A$.

Загальна вартість обчислення $A^T A$, $A^T b$ та наступного розкладання Холевської матриці $A^T A$ складає $n^2 m + \frac{1}{3} n^3 + O(n^2)$ арифметичних дій. Так як $m > n$, то в загальній вартості переважає $n^2 m$ — ціна формування матриці $A^T A$.

Існують й інші методи розв'язання лінійної задачі найменших квадратів. Метод, який використовує нормальні рівняння, найшвидший, але й найменш точний. Метод, що використовує QR -розкладання матриці A , є найбільш уживаним, але його вартість може перевищувати вартість першого вдвічі. Метод, що використовує SVD -розкладання матриці A , найбільш корисний для погано обумовлених задач. Його вартість може у кілька разів перевищувати вартість згаданих методів [8].

2.1. QR -розкладання

Теорема 2.1 (QR -розкладання). *Для будь-якої дійсної матриці $A \in R^{m \times n}$ ($m \geq n$) повного стовпцевого рангу існують такі матриці $Q \in R^{m \times n}$ з ортонормованими стовпцями і верхня трикутна матриця $R \in R^{n \times n}$ з додатними діагональними елементами, що*

$$A = QR. \quad (2.5)$$

Доведення. Процес Грама-Шмідта ортогоналізації стовпців матриці $A = [a_1, \dots, a_n]$ у порядку зростання їх номерів обчислює ортонормовані вектори $q_1, \dots, q_n$. Нижче наведено два алгоритми процесу Грама-Шмідта.

Алгоритм 2.1 *Класичний (CGS) та модифікований (MGS) алгоритми Грама-Шмідта обчислення розкладання $A = QR$*

1. *For* $i = 1$ *To* n
2. $q_i = a_i$
3. *For* $j = 1$ *To* $i - 1$
4. $r_{ji} = q_j^T a_i$ (CGS) $\{r_{ji} = q_j^T q_i$ (MGS) $\}$
5. $q_i = q_i - r_{ji} q_j$

6. *End For*
7. $r_{ii} = \|q_i\|_2$
8. *If* $r_{ii} = 0$ *Then Stop*
9. $q_i = q_i/r_{ii}$
10. *End For*

Оскільки матриця A має повний стовпцевий ранг, то алгоритм у точній арифметиці перерватися не може. З рядків 5 та 9 алгоритму 2.1 отримуємо формули

$$r_{ii}q_i = a_i - \sum_{j=1}^{i-1} r_{ji}q_j, \quad i = \overline{1, n},$$

або

$$a_i = \sum_{j=1}^i r_{ji}q_j, \quad i = \overline{1, n}. \quad (2.6)$$

Формули (2.6) — це розкладання (2.5) матриці A , записане стовпцями. $\square$

На жаль, в арифметиці з плаваючою точкою алгоритм CGS чисельно нестійкий. Найчастіше відбувається втрата ортогональності обчислених векторів q_i . Модифікований метод Грама-Шмідта (MGS) стійкіший. Алгоритм MGS вимагає $2mn^2$ арифметичних операцій.

2.2. Розв'язання ЛЗНК за допомогою QR -розкладання

Нехай для матриці системи (2.1) побудовано QR -розкладання:

$$A = QR.$$

Покладемо

$$Ax - b = QRx - b = QRx - (QQ^T + I - QQ^T)b = Q(Rx - Q^Tb) + (QQ^T - I)b.$$

Оскільки вектори $Q(Rx - Q^Tb)$ і $(QQ^T - I)b$ ортогональні, бо

$$(Q(Rx - Q^Tb))^T(QQ^T - I)b = (Rx - Q^Tb)^T Q^T(QQ^T - I)b = 0,$$

то за теоремою Піфагора

$$\|Ax - b\|_2^2 = \|Q(Rx - Q^T b)\|_2^2 + \|(QQ^T - I)b\|_2^2 = \|Rx - Q^T b\|_2^2 + \|(QQ^T - I)b\|_2^2.$$

Остання сума буде мінімальною, якщо її перший доданок буде дорівнювати нулю, тобто, якщо

$$Rx = Q^T b.$$

Отже, розв'язок ЛЗНК (2.2) можна записати у такому вигляді:

$$x = R^{-1}Q^T b.$$

Вартість цього розв'язку визначається вартістю QR -розкладання.

2.3. Розв'язання ЛЗНК за допомогою SVD -розкладання

Нехай $A = U\Sigma V^T$ — сингулярне розкладання [3, 9, 10] матриці A та $[U, \tilde{U}] \in R^{m \times m}$ — ортогональна матриця. Тоді

$$\begin{aligned} \|Ax - b\|_2^2 &= \|U\Sigma V^T x - b\|_2^2 = \left\| \begin{bmatrix} U^T \\ \tilde{U}^T \end{bmatrix} (U\Sigma V^T x - b) \right\|_2^2 = \\ &= \left\| \begin{bmatrix} \Sigma V^T x - U^T b \\ -\tilde{U}^T b \end{bmatrix} \right\|_2^2 = \|\Sigma V^T x - U^T b\|_2^2 + \|\tilde{U}^T b\|_2^2. \end{aligned}$$

Остання сума буде мінімальною, якщо перший доданок дорівнює нулю, тобто, якщо

$$\Sigma V^T x = U^T b.$$

Звідси знаходимо розв'язок ЛЗНК

$$x = V\Sigma^{-1}U^T b.$$

Для $m \gg n$ вартість цього розв'язку така сама, як і розв'язку, що побудований за допомогою QR -розкладання, яке знайдено методом MGS . Для менших m вартість дорівнює $4n^2m - \frac{4}{3}n^3 + O(n^2)$.

Якщо матриця $A \in R^{m \times n}$ ($m \geq n$) повного стовпцевого рангу та

$$A = QR = U\Sigma V^T,$$

то матриця

$$A^+ = (A^T A)^{-1} A^T = R^{-1} Q^T = V \Sigma^{-1} U^T$$

називається *псевдозворотною* (Мура-Пенроуза) для матриці A .

Псевдозворотна матриця дозволяє записати розв'язок ЛЗНК (2.2) з матрицею A повного стовпцевого рангу у вигляді

$$x = A^+ b.$$

2.4. Завдання до лабораторної роботи

Студенти повинні знати:

- псевдорозв'язок перевизначеної СЛАР;
- число обумовленості перевизначеної СЛАР;
- QR -розкладання;
- SVD -розкладання;
- методи розв'язання ЛЗНК.

Завдання

1. Обчислити значення заданої функції в n довільних вузлах проміжку $[a, b]$.
2. До обчислених значень функції додати „шум“ інтенсивності $\varepsilon > 0$:

$$\tilde{f}(x_i) = f(x_i) + \varepsilon * rand(-1, 1) * |f(x_i)|, \quad i = \overline{1, n}.$$

3. Апроксимувати „зашумлену“ функцію багаточленом ступеня m ($m < n$). Коефіцієнти апроксимуючого багаточлена,

$$\alpha_m x^m + \alpha_{m-1} x^{m-1} + \dots + \alpha_1 x + \alpha_0,$$

знайти, розв'язавши методом найменших квадратів перевизначену СЛАР:

$$\alpha_m x_j^m + \alpha_{m-1} x_j^{m-1} + \dots + \alpha_1 x_j + \alpha_0 = \tilde{f}(x_j), \quad j = \overline{1, n}. \quad (2.7)$$

4. Обчислити число обумовленості перевизначеної системи (2.7).
5. Побудувати графіки апроксимуючого багаточлена, заданої та „зашумленої“ функцій.

Варіанти завдань

$\varepsilon = 0.1, n = 20, m = 3, 5, 10.$

Вар.	$f(x)$	$[a, b]$
1	$\sin(x)$	$[0, \pi]$
2	$\cos(x)$	$[0, \pi]$
3	$x + \cos(x)$	$[0, \pi]$
4	$x + \sin(x)$	$[0, \pi]$
5	$\cos(x)\sin(x)$	$[0, \pi]$
6	$x \cos(x)$	$[0, \pi]$
7	$x \sin(x)$	$[0, \pi]$
8	$\frac{1}{x^2 + 25}$	$[-1, 1]$
9	$\frac{1}{1.5 + \cos(x)}$	$[0, \pi]$
10	$\sin^2(x)$	$[0, \pi]$

До звіту включити:

1. текст програми мовою пакета MATLAB;
2. результати обчислювального експеримента;
3. висновки.

3. КЛАСИЧНІ ІТЕРАЦІЙНІ МЕТОДИ

Ітераційний метод розв'язання СЛАР

$$Ax = b, \quad A \in R^{n \times n}, \quad \det(A) \neq 0, \quad (3.1)$$

будує послідовність

$$x_0, x_1, \dots, x_k, \dots$$

наближених розв'язків системи (3.1).

Якщо

$$\lim_{k \rightarrow \infty} \|x_k - x_*\| = 0,$$

де x_* — розв'язок системи (3.1), ітераційний метод називається збіжним.

Для завершення ітераційного процесу, зазвичай, використовують умову

$$\|x_{k+1} - x_k\| \leq \varepsilon \|x_k\| \quad (3.2)$$

або

$$\|r_k\| \leq \varepsilon \|r_0\|, \quad (3.3)$$

де ε — точність шуканого розв'язку, $r_k = b - Ax_k$ — нев'язка k -го наближення.

Спочатку розглянемо класичні ітераційні методи, а потім методи підпростору Крилова, які зараз широко використовуються для розв'язання великих розріджених систем.

3.1. Метод простої ітерації

Спочатку систему,

$$Ax = b, \quad (3.4)$$

приведемо до еквівалентної системи,

$$x = Gx + f. \quad (3.5)$$

Це можна зробити багатьма способами. Наприклад, якщо покласти

$$x = x - \alpha(Ax - b), \quad \alpha \neq 0,$$

то $G = I - \alpha A$, $f = \alpha b$, де α — дійсний параметр, якщо ж

$$x = x - B^{-1}(Ax - b),$$

то

$$G = I - B^{-1}A, \quad f = B^{-1}b.$$

Як вибрати параметр α або невироджену матрицю B , яку називають передобумовлювачем, буде пояснено пізніше.

Для знаходження розв'язку системи (3.5) будують ітераційний процес,

$$x_{k+1} = Gx_k + f, \quad k = 0, 1, 2, \dots, \quad (3.6)$$

тут x_0 — початкове наближення, яке зазвичай вибирають довільно.

Ітераційний процес (3.6) називають методом простої ітерації розв'язання системи (3.5).

В подальшому нам знадобиться така лема [1, 9, 10].

Лема 3.1 *Для того, щоб послідовність матриць*

$$G, G^2, G^3, \dots$$

збігалася до нульової матриці необхідно і достатньо, щоб радіус спектра матриці G був меншим за одиницю:

$$\rho(G) = \max_i |\lambda_i(G)| < 1.$$

Наступна теорема визначає необхідну та достатню умову збіжності простої ітерації [1, 3–5, 9, 10].

Теорема 3.1 *Для того, щоб ітераційний процес (3.6) збігався до розв'язку рівняння (3.5) для будь-яких початкового наближення x_0 і правої частини f*

необхідно і достатньо, щоб радіус спектра матриці переходу G був меншим за одиницю:

$$\rho(G) = \max_i |\lambda_i(G)| < 1. \quad (3.7)$$

На практиці необхідну і достатню умову збіжності простої ітерації перевірити складно, оскільки треба визначати межі спектра матриці G , що є доволі складним завданням.

Розглянемо достатні умови збіжності простої ітерації, які легко перевіряються в практичних обчисленнях.

Теорема 3.2 Якщо $\|G\| < 1$, то метод простої ітерації (3.6) збігається.

Доведення. Нехай $\{\lambda, u\}$ ($\|u\| = 1$) — власна пара матриці переходу G , тобто $Gu = \lambda u$. Тоді

$$|\lambda| \|u\| = \|Gu\| \leq \|G\| \|u\|.$$

Значить,

$$|\lambda| \leq \|G\|.$$

Оскільки $\|G\| < 1$, то всі власні значення матриці G теж менше 1, тому за теоремою 3.1 ітераційний процес (3.6) збігається. $\square$

Отримаємо ще одну достатню умову збіжності простої ітерації.

Матрицю A системи

$$Ax = b \quad (3.8)$$

представимо у вигляді

$$A = L + D + U,$$

де $D = \text{diag}\{a_{11}, a_{22}, \dots, a_{nn}\}$, L — нижня трикутна матриця з нульовою діагоналлю, U — верхня трикутна матриця з нульовою діагоналлю. Тоді систему (3.8) можна записати так

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

За теоремою 3.2 ітераційний процес

$$x_{k+1} = -D^{-1}(L + U)x_k + D^{-1}b, \quad k = 0, 1, 2, \dots \quad (3.9)$$

збігається, якщо

$$\|D^{-1}(L + U)\|_{\infty} < 1,$$

тобто, якщо

$$\max_i \frac{1}{|a_{ii}|} \sum_{j=1, j \neq i}^n |a_{ij}| < 1.$$

Ця умова буде виконуватись, якщо:

$$\sum_{j=1, j \neq i}^n |a_{ij}| < |a_{ii}| \quad i = \overline{1, n}. \quad (3.10)$$

Матриця A , що задовольняє умові (3.10) називається матрицею зі строгою діагональною перевагою.

Таким чином, ми отримали ще одну достатню умову збіжності простої ітерації.

Теорема 3.3 *Якщо матриця A є матрицею зі строгою діагональною перевагою, то ітераційний процес (3.9) збігається.*

3.2. Показник збіжності ітераційного процесу

Теорема 3.4 [1, 3, 9, 10] *Для будь-якої матриці A*

$$\rho(A) = \lim_{k \rightarrow \infty} \|A^k\|^{\frac{1}{k}}. \quad (3.11)$$

Нехай x_* — розв’язок задачі (3.5), тобто,

$$x_* = Gx_* + f.$$

Віднімаючи з (3.6) останню рівність, отримуємо

$$x_k - x_* = G(x_{k-1} - x_*) = G^2(x_{k-2} - x_*) = \dots = G^k(x_0 - x_*).$$

Значить

$$\|x_k - x_*\| \leq \|G^k\| \|x_0 - x_*\|. \quad (3.12)$$

Оскільки за теоремою 3.4

$$\|G^k\| \approx \rho^k(G),$$

то з (3.12) випливає, що кількість ітерацій k ітераційного процесу (3.6), достатню для зменшення початкової помилки в e разів, можна знайти з рівняння

$$\rho^k(G) = \frac{1}{e}.$$

Звідси

$$k \ln \rho(G) = -1,$$

або

$$k = -\frac{1}{\ln \rho(G)}. \quad (3.13)$$

Величину $\nu = -\ln \rho(G)$ називають показником збіжності ітераційного процесу (3.6).

З (3.13) маємо, що величина, обернена до показника збіжності ітераційного процесу, дорівнює числу ітерацій, які необхідно зменшати початкову помилку $\|x_0 - x_*\|$ у e разів.

Отже, чим більше показник збіжності ітераційного процесу, тим швидше ітераційний процес збігається.

3.3. Метод Річардсона

Метод Річардсона розв'язання системи

$$Ax = b \quad (3.14)$$

з симетричною додатно визначеною матрицею $A \in R^{n \times n}$ задається наступним ітераційним процесом:

$$x_{k+1} = x_k - \alpha_k(Ax_k - b), \quad k = \overline{0, p-1}, \quad (3.15)$$

після кожних p кроків вважають $x_0 = x_p$.

Нехай x_* — розв'язок задачі (3.14). Тоді

$$x_* = x_* - \alpha_k(Ax_* - b), \quad k = \overline{0, p-1}. \quad (3.16)$$

Віднімаючи (3.16) з (3.15), отримаємо

$$\begin{aligned} x_{k+1} - x_* &= x_k - x_* - \alpha_k A(x_k - x_*) = (I - \alpha_k A)(x_k - x_*) = \\ &= (I - \alpha_k A)(I - \alpha_{k-1} A)(x_{k-1} - x_*) = \dots = \\ &= (I - \alpha_k A)(I - \alpha_{k-1} A) \dots (I - \alpha_0 A)(x_0 - x_*). \end{aligned}$$

Звідси, поклавши $k = p - 1$, маємо

$$x_p - x_* = G_p(x_0 - x_*),$$

де

$$G_p = (I - \alpha_{p-1} A)(I - \alpha_{p-2} A) \dots (I - \alpha_0 A).$$

Отже, якщо цикл (3.15) повторити m разів, то отримаємо наближення x_{mp} . Причому,

$$x_{mp} - x_* = G_p(x_{m(p-1)} - x_*) = G_p^2(x_{m(p-2)} - x_*) = \dots = G_p^m(x_0 - x_*).$$

Значить,

$$\|x_{mp} - x_*\| \leq \|G_p^m\| \|x_0 - x_*\|. \quad (3.17)$$

На підставі леми 3.1 звідси випливає, що метод Річардсона (3.15) збігається при фіксованому p і $m \rightarrow \infty$, якщо

$$\rho(G_p) < 1. \quad (3.18)$$

За теоремою 3.4 про радіус спектра матриці

$$\|G_p^m\| \approx \rho^m(G_p)$$

при великих m . Тоді з (3.17) слідує, що кількість ітерацій $k = mp$ достатніх для зменшення початкової помилки $\|x_0 - x\|$ у e разів можна знайти з рівняння

$$\rho^m(G_p) = \frac{1}{e}.$$

Звідси

$$mp \frac{\ln \rho(G_p)}{p} = -1,$$

$$k = mp = \frac{1}{-\ln \rho(G_p)/p},$$

тобто,

$$\nu = -\frac{\ln \rho(G_p)}{p} \quad (3.19)$$

— показник збіжності методу Річардсона.

Параметри $\alpha_0, \dots, \alpha_{p-1}$ методу Річардсона вибирають так, щоб показник збіжності був найбільшим, тобто з умови

$$\min_{\alpha_0, \dots, \alpha_{p-1}} \max_i |\lambda_i(G_p)|. \quad (3.20)$$

Зауважимо, що

$$\lambda(G_p) = (1 - \alpha_{p-1}\lambda(A))(1 - \alpha_{p-2}\lambda(A)) \cdots (1 - \alpha_0\lambda(A)) \equiv P_p(\lambda(A)).$$

Тому задачу (3.20) можна записати у такому вигляді

$$\min_{\alpha_0, \dots, \alpha_{p-1}} \max_i |P_p(\lambda_i(A))|. \quad (3.21)$$

Щоб розв'язати цю задачу, необхідно знати всі власні значення матриці A . Оскільки матриця A симетрична та додатно визначена, то

$$0 < m \leq \lambda_i(A) \leq M, \quad i = \overline{1, n}.$$

Припускаючи, що константи m і M відомі, задачу (3.21) замінюють більш загальною задачею:

$$\min_{\alpha_0, \dots, \alpha_{p-1}} \max_{m \leq \lambda \leq M} |P_p(\lambda)|. \quad (3.22)$$

Заміною змінних

$$x = \frac{2}{M-m}\lambda - \frac{M+m}{M-m} \quad \left(\lambda = \frac{M-m}{2}x + \frac{M+m}{2} \right)$$

проміжок $[m, M]$ відображається у проміжок $[-1, 1]$. Оскільки на проміжку $[-1, 1]$ найменше ухиляється від нуля багаточлен $2^{1-p}T_p(x)$ ($T_p(x)$ — багаточлен Чебишева), то розв'язком задачі (3.22) буде багаточлен:

$$(1 - \alpha_{p-1}\lambda)(1 - \alpha_{p-2}\lambda) \cdots (1 - \alpha_0\lambda) = \frac{T_p(\frac{2}{M-m}\lambda - \frac{M+m}{M-m})}{T_p(-\frac{M+m}{M-m})}. \quad (3.23)$$

Корені багаточлена в лівій частині рівності (3.23):

$$\lambda_i = \frac{1}{\alpha_i}, \quad i = \overline{0, p-1}.$$

Корені багаточлена у правій частині рівності (3.23):

$$\lambda_i = \frac{M-m}{2} \cos \frac{(2i+1)\pi}{2p} + \frac{M+m}{2}, \quad i = \overline{0, p-1}.$$

З умови рівності цих коренів отримуємо,

$$\alpha_i = \left(\frac{M-m}{2} \cos \frac{(2i+1)\pi}{2p} + \frac{M+m}{2} \right)^{-1}, \quad i = \overline{0, p-1}. \quad (3.24)$$

Цей набір параметрів ітераційного процесу називається чебишевським. За такого вибору параметрів

$$\rho(G_p) = \max_{m \leq \lambda \leq M} |(1 - \alpha_{p-1}\lambda)(1 - \alpha_{p-2}\lambda) \cdots (1 - \alpha_0\lambda)| = \frac{1}{|T_p(-\frac{M+m}{M-m})|}. \quad (3.25)$$

Оскільки

$$\frac{M+m}{M-m} = 1 + \frac{2m}{M-m} > 1,$$

то

$$\left| T_p \left(-\frac{M+m}{M-m} \right) \right| = T_p \left(\frac{M+m}{M-m} \right) > 1.$$

Тому умова (3.18) збіжності ітераційного процесу виконується.

Обчислимо тепер показник збіжності методу Річардсона (3.19). Для обчислення

$$T_p \left(\frac{M+m}{M-m} \right)$$

скористаємося наступним поданням багаточлена Чебишева:

$$T_p(x) = \frac{(x - \sqrt{x^2 - 1})^p}{2} + \frac{(x + \sqrt{x^2 - 1})^p}{2}.$$

Якщо покласти

$$x = \frac{M+m}{M-m} = \frac{1 + \frac{1}{\text{cond}_c A}}{1 - \frac{1}{\text{cond}_c A}} = \frac{1 + \xi}{1 - \xi},$$

де $\xi = \frac{1}{\text{cond}_c A}$, то

$$x + \sqrt{x^2 - 1} = \frac{1 + \xi}{1 - \xi} + \sqrt{\frac{(1 + \xi)^2}{(1 - \xi)^2} - 1} = \frac{1 + \xi}{1 - \xi} + \frac{2\sqrt{\xi}}{1 - \xi} = \frac{1 + \sqrt{\xi}}{1 - \sqrt{\xi}},$$

$$x - \sqrt{x^2 - 1} = \frac{1 + \xi}{1 - \xi} - \frac{2\sqrt{\xi}}{1 - \xi} = \frac{1 - \sqrt{\xi}}{1 + \sqrt{\xi}}.$$

Поклавши

$$\eta = \frac{1 - \sqrt{\xi}}{1 + \sqrt{\xi}}$$

маємо

$$T_p \left(\frac{M+m}{M-m} \right) = \frac{\eta^p}{2} + \frac{1}{2\eta^p} = \frac{\eta^{2p} + 1}{2\eta^p}. \quad (3.26)$$

Значить,

$$\rho(G_p) = \frac{1}{T_p(\frac{M+m}{M-m})} = \frac{2\eta^p}{\eta^{2p} + 1}.$$

Тоді коефіцієнт збіжності методу Річардсона

$$\nu = -\frac{\ln \rho(G_p)}{p} = -\frac{1}{p} \ln \frac{2\eta^p}{\eta^{2p} + 1} = -\frac{1}{p} \ln \eta^p + \frac{1}{p} \ln \frac{\eta^{2p} + 1}{2} = -\ln \eta + \frac{1}{p} \ln \frac{\eta^{2p} + 1}{2}.$$

Тому за великих p можна покласти

$$\nu \approx -\ln \eta = -\ln \frac{1 - \sqrt{\xi}}{1 + \sqrt{\xi}} = 2\sqrt{\xi} + O(\xi).$$

Значить, коефіцієнт збіжності методу Річардсона

$$\nu \approx 2\sqrt{\xi} = \frac{2}{\sqrt{\text{cond}_c A}}. \quad (3.27)$$

Якщо $p = 1$, то метод Річардсона визначає метод простої ітерації з оптимальним параметром:

$$x_{k+1} = x_k - \alpha_0(Ax_k - b), \quad k = 0, 1, 2, \dots, \quad (3.28)$$

де з (3.24)

$$\alpha_0 = \frac{2}{m + M} \quad \text{— оптимальний параметр.}$$

Визначимо показник збіжності простої ітерації з оптимальним параметром. З (3.25) маємо,

$$\rho(G_1) = \frac{1}{|T_1(-\frac{M+m}{M-m})|} = \frac{M-m}{M+m} = \frac{1 - \frac{m}{M}}{1 + \frac{m}{M}} = \frac{1 - \frac{1}{\text{cond}_c(A)}}{1 + \frac{1}{\text{cond}_c(A)}},$$

тоді

$$\begin{aligned} \nu &= -\ln \rho(G) = -\ln \frac{1 - \frac{1}{\text{cond}_c(A)}}{1 + \frac{1}{\text{cond}_c(A)}} = \\ &= \ln \left(1 + \frac{1}{\text{cond}_c(A)} \right) - \ln \left(1 - \frac{1}{\text{cond}_c(A)} \right) = \\ &= \frac{2}{\text{cond}_c(A)} + O \left(\frac{1}{\text{cond}_c^2(A)} \right). \end{aligned}$$

Таким чином, проста ітерація з оптимальним вибором параметром має показник збіжності

$$\nu \approx \frac{2}{\text{cond}_c(A)}. \quad (3.29)$$

Порівнюючи показники збіжності (3.29) і (3.27), робимо висновок, що

для системи з додатно визначеною симетричною матрицею метод Річардсона збігається швидше простої ітерації з оптимальним параметром.

3.4. Завдання до лабораторної роботи

Студенти повинні знати:

- метод простої ітерації розв'язання СЛАР;
- показник збіжності ітераційного процесу;
- проста ітерація з оптимальним параметром та її показник збіжності;
- метод Річардсона та його показник збіжності.

Завдання

Функція $A = \text{gallery}('wathen', nx, ny)$ обчислює розріджену матрицю A розміру $n \times n$, де $n = 3 * nx * ny + 2 * nx + 2 * ny + 1$, яка з'являється при апроксимації методом скінченних елементів деякої крайової задачі у прямокутнику (параметри задачі при кожному виклику функції `gallery` змінюються випадково) і є симетричною та додатно визначеною.

Методом Річардсона, використовуючи різні значення параметра p , знайти наближений розв'язок системи

$$Ax = b, \quad (3.30)$$

де $A = \text{gallery}('wathen', nx, ny)$, $b = Ax_*$, $x_* = \text{rand}(n, 1)$. Для закінчення ітераційного процесу використати оцінку

$$\|x_k - x_*\| \leq \varepsilon \|x_0 - x_*\|,$$

де $\varepsilon = e^{-10}$.

Обчислити теоретичні показники збіжності простої ітерації з оптимальним параметром ν_{iter} та методу Річардсона ν_{rich} . Знаючи число ітерацій k , яке затрачено на знаходження наближеного розв'язку системи (3.30), обчислити фактичний показник збіжності ітераційного процесу ν .

Провести обчислювальні експерименти для заповнення наступної та-

блиці:

Результати обчислювального експерименту

				$p = 1$		$p = 5$		$p = 10$		$p = 20$	
	$cond_c(A)$	ν_{iter}	ν_{rich}	k	ν	k	ν	k	ν	k	ν
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											

Варіанти завдань

Вар.	n_x	n_y
1	3	3
2	4	4
3	5	5
4	6	6
5	3	4
6	3	5
7	3	6
8	4	5
9	5	6
10	5	6

До звіту включити:

1. текст програми мовою пакета MATLAB;
2. результати обчислювального експерименту;
3. результати обчислювального експерименту;
4. графіки залежності показників збіжності (теоретичних та фактичних) від числа обумовленості матриці
5. висновки.

4. ІТЕРАЦІЙНІ МЕТОДИ ПІДПРОСТОРУ КРИЛОВА

Для матриці $A \in R^{n \times n}$ та довільного вектора $v \in R^n$ m -ий підпростір Крилова визначається так:

$$\mathcal{K}_m(A, v) = \text{span} \{v, Av, \dots, A^{m-1}v\}. \quad (4.1)$$

Добре відомо, що підпростори Крилова утворюють вкладену послідовність підпросторів, найбільша розмірність яких

$$d \equiv \dim \mathcal{K}_n(A, v) \leq n,$$

тобто,

$$\mathcal{K}_1(A, v) \subset \dots \subset \mathcal{K}_d(A, v) = \dots = \mathcal{K}_n(A, v).$$

Зокрема, для будь-якого $m \leq d$ розмірність підпростору Крилова $\mathcal{K}_m(A, v)$ дорівнює m .

Алгоритм Арнольдї будує ортонормований базис підпростору Крилова $\mathcal{K}_m(A, v)$. Фактично Алгоритм Арнольдї — це процес Грама-Шмідта побудови ортонормованого базису $\{v_1, v_2, \dots, v_m\}$ для векторів $\{v, Av, \dots, A^{m-1}v\}$.

Алгоритм 4.1 Алгоритм Арнольдї

1. $v_1 = v / \|v\|_2$
2. *For* $j = 1, 2, \dots, m$ *Do*
3. *For* $i = 1, 2, \dots, j$ *Do*
4. $h_{ij} = (Av_j, v_i)$
5. *End Do*
6. $w_j = Av_j - \sum_{i=1}^j h_{ij}v_i$
7. $h_{j+1,j} = \|w_j\|_2$
8. *If* $h_{j+1,j} = 0$ *Then Stop*
9. $v_{j+1} = w_j / h_{j+1,j}$
10. *End Do*

Теорема 4.1 *Мають місце такі рівності:*

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T = \quad (4.2)$$

$$= V_{m+1} \bar{H}_m, \quad (4.3)$$

$$V_m^T AV_m = H_m. \quad (4.4)$$

Тут $V_m = [v_1, v_2, \dots, v_m]$, H_m — верхня Хесенбергова матриця розміру $m \times m$ з елементами h_{ij} , матрицю $\bar{H}_m$ отримуємо з матриці H_m додаванням рядка $[0, \dots, 0, h_{m+1,m}]$.

Доведення. З рядків з номерами 6 та 9 алгоритму Арнольдї слідує рівності

$$Av_j = \sum_{i=1}^{j+1} h_{ij} v_i \quad j = \overline{1, m}. \quad (4.5)$$

Записуючи рівності (4.5) матричному вигляді, отримуємо (4.2) та (4.3). Рівність (4.4) випливає з (4.2), оскільки вектор v_{m+1} ортогональний стовпцям матриці V_m . $\square$

4.1. Метод Арнольдї

Метод Арнольдї є проєкційним методом розв'язання системи лінійних рівнянь алгебри $Ax = b$. Інша його назва — FOM (Full Orthogonalization Method) [5, 9, 10].

Метод будується наступним чином. Нехай x_0 — довільне початкове наближення, $r_0 = b - Ax_0$ — нев'язка початкового наближення, $\mathcal{K}_m(A, r_0)$ — підпростір Крилова і $\dim \mathcal{K}_m(A, r_0) = m$. Знайдемо такий вектор

$$x_m \in x_0 + \mathcal{K}_m(A, r_0), \quad (4.6)$$

що

$$r_m \equiv b - Ax_m \perp \mathcal{L} = \mathcal{K}_m(A, r_0). \quad (4.7)$$

Використовуючи алгоритм Арнольді, побудуємо ортонормований базис

$$V_m = [v_1, v_2, \dots, v_m]$$

підпростору Крилова $\mathcal{K}_m(A, r_0)$, де $v_1 = r_0/\beta$, $\beta = \|r_0\|_2$. Тоді

$$V_m^T r_0 = V_m^T \beta v_1 = \beta e_1.$$

Вектор x_m шукатимемо у вигляді

$$x_m = x_0 + V_m y_m. \quad (4.8)$$

Використовуючи умову (4.7) для визначення вектора y_m , отримуємо ланцюжок рівностей

$$\begin{aligned} V_m^T r_m &= V_m^T (b - Ax_m) = V_m^T (b - A(x_0 + V_m y_m)) = V_m^T r_0 - V_m^T A V_m y_m = \\ &= V_m^T \beta v_1 - H_m y_m = \beta e_1 - H_m y_m = 0. \end{aligned}$$

З останньої рівності в цьому ланцюжку отримуємо рівняння для визначення вектора y_m

$$H_m y_m = \beta e_1. \quad (4.9)$$

Звідси

$$y_m = H_m^{-1} \beta e_1.$$

Знаючи y_m , за формулою (4.8) знаходимо наближення x_m до розв'язку системи $Ax = b$, побудоване за m кроків методу Арнольді.

Має місце така теорема.

Теорема 4.2 *Якщо x_m — наближене розв'язок системи $Ax = b$, побудований за m кроків методу Арнольді, то*

$$r_m \equiv b - Ax_m = -h_{m+1,m} e_m^T y_m v_{m+1}, \quad (4.10)$$

тому

$$\|b - Ax_m\|_2 = h_{m+1,m} |e_m^T y_m|. \quad (4.11)$$

Доведення. З (4.8), (4.2), (4.4) та (4.9)

$$\begin{aligned}
 b - Ax_m &= b - Ax_0 - AV_my_m = r_0 - AV_my_m = \\
 &= \beta v_1 - V_m H_m y_m - h_{m+1,m} e_m^T y_m v_{m+1} = \\
 &= \beta v_1 - V_m \beta e_1 - h_{m+1,m} e_m^T y_m v_{m+1} = -h_{m+1,m} e_m^T y_m v_{m+1}.
 \end{aligned}$$

Для доказу (4.11) залишилося помітити, що вектор v_{m+1} одиничної довжини. $\square$

З (4.11) випливає, що для обчислення норми невязки r_m немає необхідності обчислювати x_m і саму невязку $r_m = b - Ax_m$.

Ітерації методу Арнольді повторюють, поки не виконується нерівність

$$||r_m||_2 = h_{m+1,m} |e_m^T y_m| \leq \varepsilon ||r_0||_2$$

для заданого ε . Після завершення ітераційного процесу обчислюємо $x_m = V_m y_m$.

Наведемо алгоритм методу Арнольді.

Алгоритм 4.2 Метод Арнольді (FOM)

1. $r_0 = b - Ax_0; \beta = ||r_0||_2; v_1 = r_0/\beta$

2. $j = 0$

3. *Repeat*

4. $j = j + 1$

5. *For* $i = 1, 2, \dots, j$ *Do*

6. $h_{ij} = (Av_j, v_i)$

7. *End Do*

8. $w_j = Av_j - \sum_{i=1}^j h_{ij} v_i$

9. $h_{j+1,j} = ||w_j||_2$

10. *If* $h_{j+1,j} = 0$ *Then Stop*

11. $v_{j+1} = w_j/h_{j+1,j}$

12. $y_j = H_j^{-1} \beta e_1$

13. *Until* $h_{j+1,j} |e_j^T y_j| < \varepsilon ||r_0||_2$

$$14. \quad x_j = x_0 + V_j y_j$$

4.2. Метод узагальненої мінімізації нев'язки

В англomовній літературі цей метод називається GMRES (Generalized Minimum Residual Method)[6, 8–13].

Нехай: x_0 — початкове наближення до розв'язку системи $Ax = b$; $r_0 = b - Ax_0$ — нев'язка початкового наближення x_0 ; стовпці матриці V_m , побудованої алгоритмом Арнольді, що утворюють ортонормований базис підпростору Крилова $\mathcal{K}_m(A, r_0)$. Тоді будь-який вектор

$$x \in x_0 + \mathcal{K}_m(A, r_0)$$

можна представити у вигляді

$$x = x_0 + V_m y.$$

Визначимо функціонал

$$J(y) = \|b - Ax\|_2 = \|b - A(x_0 + V_m y)\|_2.$$

З (4.3) випливає, що

$$\begin{aligned} b - Ax &= b - A(x_0 + V_m y) = r_0 - AV_m y = \beta v_1 - V_{m+1} \bar{H}_m y = \\ &= V_{m+1}(\beta e_1 - \bar{H}_m y). \end{aligned}$$

Оскільки стовпці матриці V_{m+1} ортонормовані, то матриця V_{m+1} зберігає 2-норму вектора. Тому

$$J(y) = \|b - Ax\|_2 = \|V_{m+1}(\beta e_1 - \bar{H}_m y)\|_2 = \|\beta e_1 - \bar{H}_m y\|_2. \quad (4.12)$$

Розглянемо лінійну задачу найменших квадратів

$$\min_{y \in \mathcal{K}_m(A, r_0)} \|\beta e_1 - \bar{H}_m y\|_2. \quad (4.13)$$

Нехай $y_m = \overline{H}_m^+ \beta e_1$ — розв’язок задачі (4.13), де $\overline{H}_m^+$ — псевдозворотня матриця. Тоді,

$$x_m = x_0 + V_m y_m,$$

є наближеним розв’язком системи $Ax = b$, що побудований за m кроків алгоритму GMRES.

З (4.12),(4.13) випливає, що метод GMRES мінімізує нев’язку на підпросторі Крилова $\mathcal{K}_m(A, r_0)$. Більше того, з (4.12) отримуємо

$$\|r_m\|_2 = \|b - Ax_m\|_2 = \|\beta e_1 - \overline{H}_m y_m\|_2.$$

Наведемо алгоритм методу GMRES.

Алгоритм 4.3 *GMRES*

1. $r_0 = b - Ax_0; \beta = \|r_0\|_2; v_1 = r_0/\beta$
2. $j = 0$
3. *Repeat*
4. $j = j + 1$
5. *For* $i = 1, 2, \dots, j$ *Do*
6. $h_{ij} = (Av_j, v_i)$
7. *End Do*
8. $w_j = Av_j - \sum_{i=1}^j h_{ij} v_i$
9. $h_{j+1,j} = \|w_j\|_2$
10. *If* $h_{j+1,j} = 0$ *Then Stop*
11. $v_{j+1} = w_j/h_{j+1,j}$
12. $y_j = \overline{H}_j^+ \beta e_1$
13. *Until* $\|\beta e_1 - \overline{H}_j y_j\|_2 < \varepsilon \|r_0\|_2$
14. $x_j = x_0 + V_j y_j$

4.3. Метод Ланцоша

Спочатку розглянемо алгоритм Ланцоша побудови ортонормованого базису підпростору Крилова $\mathcal{K}_m(A, v)$ для симетричної матриці A . Алгоритм Ланцоша — це алгоритм Арнольдї для симетричної матриці A .

Отже, нехай $A = A^T \in R^{n \times n}$. Нагадаємо, що алгоритм Арнольдї будує матрицю

$$V_m = [v_1, v_2, \dots, v_m],$$

стовпці якої утворюють ортонормований базис $\mathcal{K}_m(A, v)$. При цьому

$$V_m^T A V_m = H_m, \quad (4.14)$$

де H_m — верхня Хесенбергова матриця. Оскільки матриця A симетрична, то матриця H_m теж симетрична, отже, тридіагональна. Позначимо

$$\begin{aligned} \alpha_i &= h_{ii}, & i &= \overline{1, m}, \\ \beta_i &= h_{i-1, i}, & i &= \overline{2, m}, \end{aligned}$$

де h_{ij} — елементи матриці H_m . Визначимо матрицю

$$T_m = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \beta_m & \alpha_m \end{bmatrix}$$

Тоді (4.14) набуде вигляду

$$V_m^T A V_m = T_m. \quad (4.15)$$

Рівність (4.2) у нових позначеннях виглядає так

$$A V_m = V_m T_m + \beta_{m+1} v_{m+1} e_m^T. \quad (4.16)$$

Записуючи матричну рівність (4.16) по-векторно, отримаємо наступний алгоритм Ланцош.

Алгоритм 4.4 Алгоритм Ланцоша

1. $v_1 = v / \|v\|_2; \beta_1 = 0; v_0 = 0$
2. *For* $j = 1, 2, \dots, m$ *Do*
3. $w_j = A v_j - \beta_j v_{j-1}$
4. $\alpha_j = w_j^T v_j$
5. $w_j = w_j - \alpha_j v_j$
6. $\beta_{j+1} = \|w_j\|_2$

7. *If $\beta_{j+1} = 0$ Then Stop*
8. $v_{j+1} = w_j / \beta_{j+1}$
9. *End Do*

Справді, із (4.16) маємо

$$Av_j = \beta_j v_{j-1} + \alpha_j v_j + \beta_{j+1} v_{j+1}, \quad j = \overline{1, m}, \quad (4.17)$$

де $v_0 = 0$, $\beta_1 = 1$. Припустимо, що

$$v_1, v_2, \dots, v_j, \quad \alpha_1, \alpha_2, \dots, \alpha_{j-1}, \quad \beta_2, \dots, \beta_j$$

вже визначено і на j -му кроці. Треба визначити α_j , β_{j+1} і v_{j+1} . Спочатку позначимо

$$w_j = \alpha_j v_j + \beta_{j+1} v_{j+1}. \quad (4.18)$$

Тоді з (4.17) можна визначити

$$w_j = Av_j - \beta_j v_{j-1}.$$

Враховуючи (4.18),

$$\alpha_j = w_j^T v_j.$$

Покладемо тепер

$$w_j = w_j - \alpha_j v_j.$$

Знову враховуючи (4.18), знаходимо

$$\beta_{j+1} = \|w_j\|_2, \quad v_{j+1} = w_j / \beta_{j+1}.$$

На цьому j -й крок алгоритму закінчується.

Розглянемо тепер метод Ланцоша розв'язання систем лінійних алгебраїчних рівнянь. Фактично метод Ланцоша — це метод Арнольдї (FOM) для симетричної матриці. Як і метод Арнольдї, метод Ланцоша за m кроків

знаходити наближених розв'язок

$$x_m = x_0 + V_m y_m,$$

де

$$y_m = H_m^{-1}(\beta e_1) = T_m^{-1}(\beta e_1).$$

При цьому формули (4.10),(4.11) набувають вигляду

$$r_m \equiv b - Ax_m = -\beta_{m+1} e_m^T y_m v_{m+1}, \quad (4.19)$$

тому

$$\|b - Ax_m\|_2 = \beta_{m+1} |e_m^T y_m|. \quad (4.20)$$

Запишемо тепер алгоритм методу Ланцоша.

Алгоритм 4.5 Метод Ланцоша розв'язання СЛАР

1. $r_0 = b - Ax_0; \beta = \|r_0\|_2; v_1 = r_0/\beta$
2. $j = 0; \beta_1 = 0; v_0 = 0$
3. *Repeat*
4. $j = j + 1$
5. $w_j = Av_j - \beta_j v_{j-1}$
6. $\alpha_j = w_j^T v_j$
7. $w_j = w_j - \alpha_j v_j$
8. $\beta_{j+1} = \|w_j\|_2$
9. *If $\beta_{j+1} = 0$ Then Stop*
10. $v_{j+1} = w_j/\beta_{j+1}$
11. $t_{jj} = \alpha_j; t_{j+1,j} = \beta_{j+1}$
12. $y_j = T_j^{-1}(\beta e_1)$
13. $t_{j,j+1} = \beta_{j+1}$
14. *Until $\beta_{j+1} |e_j^T y_j| < \varepsilon \|r_0\|_2$*
15. $x_j = x_0 + V_j y_j$

4.4. Метод спряжених градієнтів

Якщо матриця системи

$$Ax = b \quad (4.21)$$

симетрична та додатно визначена, то наближений розв'язок системи (4.21) доцільно шукати методом спряжених градієнтів (CG — Conjugate Gradient)[6, 8–13], який опишемо, удосконалюючи метод Ланцоша. Оскільки матриця A системи (4.21) симетрична та додатно визначена, то тридіагональна матриця T_m у (4.15) та (4.16) також симетрична та додатно визначена. Справді, з (4.15) $\forall x \in R^m$, $x \neq 0$, отримуємо

$$x^T T_m x = x^T V_m^T A V_m x = (V_m x)^T A V_m x > 0.$$

Отже, для матриці T_m існує LU -розкладання

$$T_m = L_m U_m.$$

Тоді

$$x_m = x_0 + V_m T_m^{-1}(\beta e_1) = x_0 + V_m U_m^{-1} L_m^{-1}(\beta e_1).$$

Позначимо

$$P_m = V_m U_m^{-1}, \quad (4.22)$$

$$z_m = L_m^{-1} \beta e_1.$$

Тоді,

$$x_m = x_0 + P_m z_m. \quad (4.23)$$

Оскільки, U_m дводіагональна матриця, то з (4.22) випливає, що останній стовець матриці V_m можна виразити через передостанній та останній стовпці матриці P_m :

$$p_{m-1} u_{m-1,m} + p_m u_{mm} = v_m,$$

або

$$p_m = \frac{1}{u_{mm}}(v_m - u_{m-1,m} p_{m-1}). \quad (4.24)$$

Лема 4.1 Нехай x_m — наближений розв'язок системи $Ax = b$ з симетри-

чною додатно визначеною матрицею $A \in R^{n \times n}$, який побудовано за m кроків методу Ланцоша. Тоді

$$r_j^T r_i = 0, \quad i \neq j, \quad (i, j = \overline{1, m}), \quad (4.25)$$

вектори p_i A -спряжені, тобто,

$$p_i^T A p_j = 0, \quad i \neq j, \quad (i, j = \overline{1, m}). \quad (4.26)$$

Доведення. (4.25) відразу випливає з (4.19), оскільки вектори v_i ($i = \overline{1, m+1}$) ортонормовані. Враховуючи (4.22), отримуємо

$$\begin{aligned} P_m^T A P_m &= (V_m U_m^{-1})^T A V_m U_m^{-1} = U_m^{-T} V_m^T A V_m U_m^{-1} = \\ &= U_m^{-T} T_m U_m^{-1} = U_m^{-T} L_m U_m U_m^{-1} = U_m^{-T} L_m. \end{aligned}$$

Матриця $U_m^{-T} L_m$ нижня трикутна, як добуток нижніх трикутних матриць, з іншого боку, вона симетрична, отже, вона діагональна. $\square$

Наближений розв'язок x_m системи лінійних алгебраїчних рівнянь (4.21) шукатимемо у вигляді (4.23). Приймемо позначення

$$z_m = [\alpha_1, \alpha_2, \dots, \alpha_m]^T.$$

Компоненти α_i вектора z_m підлягають визначенню. З (4.23) слідує, що

$$x_j = x_{j-1} + \alpha_j p_j \quad j = 1, 2, \dots \quad (4.27)$$

Звідси

$$r_j = r_{j-1} - \alpha_j A p_j \quad j = 1, 2, \dots \quad (4.28)$$

За попередньою лемою вектори r_j ортогональні один одному. Тому

$$r_j^T r_{j-1} = (r_{j-1} - \alpha_j A p_j)^T r_{j-1} = r_{j-1}^T r_{j-1} - \alpha_j (A p_j)^T r_{j-1} = 0.$$

З останньої рівності

$$\alpha_j = \frac{r_{j-1}^T r_{j-1}}{p_j^T A r_{j-1}}. \quad (4.29)$$

З (4.19) і (4.24) випливає, що вектор p_m є лінійною комбінацією векторів p_{m-1} і r_{m-1} :

$$p_m = \frac{1}{u_{mm}} \left(\frac{r_{m-1}}{-\beta_m e_{m-1}^T y_{m-1}} - u_{m-1,m} p_{m-1} \right).$$

Звідси отримуємо рекурентну формулу

$$\tilde{p}_{j+1} = r_j + \beta_j \tilde{p}_j, \quad j = 1, 2, \dots,$$

де $\tilde{p}_j$ — це вектор p_j , помножений на деяке число. Зауважимо, що вектори $\tilde{p}_j$ теж A -спряжені. Надалі замість векторів p_j будемо використовувати вектори $\tilde{p}_j$, зберігши попередні позначення. Таким чином, маємо рекурентну формулу

$$p_{j+1} = r_j + \beta_j p_j, \quad j = 1, 2, \dots. \quad (4.30)$$

Враховуючи A -спряженість векторів p_j та симетричність матриці A , отримуємо

$$(A p_j)^T r_j = (A p_j)^T (p_{j+1} - \beta_j p_j) = -\beta_j p_j^T A p_j.$$

Тепер можна визначити

$$\beta_j = -\frac{p_j^T A r_j}{p_j^T A p_j}. \quad (4.31)$$

З формул (4.27)–(4.31) отримуємо одну з реалізацій методу спряжених градієнтів (CG).

Алгоритм 4.6 Метод спряжених градієнтів (CG)

1. $r_0 = b - A x_0; p_1 = r_0$
2. $j = 0$
3. *Repeat*
4. $j = j + 1$
5. $\alpha_j = \frac{r_{j-1}^T r_{j-1}}{p_j^T A r_{j-1}}$
6. $x_j = x_{j-1} + \alpha_j p_j$

7. $r_j = r_{j-1} - \alpha_j A p_j$
8. $\beta_j = -\frac{p_j^T A r_j}{p_j^T A p_j}$
9. $p_{j+1} = r_j + \beta_j p_j$
10. *Until* $\|r_j\|_2 < \varepsilon \|r_0\|_2$

4.5. Передобумовлення

Відомо, що швидкість збіжності ітераційного методу залежить від числа обумовленості матриці. Щоб збільшити швидкість збіжності ітераційних методів, використовують передобумовлення. Передобумовлення полягає в наступному. Система

$$Ax = b$$

замінюється системою

$$M^{-1}Ax = M^{-1}b.$$

Матриця M , яку називають передобумовлювачем, повинна бути близькою до матриці A і мати наступні властивості:

1. матриця $M^{-1}A$ має бути добре обумовлена;
2. система $My = f$ має легко розв'язуватись.

Продуманий вибір, який залежить від конкретної задачі, матриці M може зробити число обумовленості матриці $M^{-1}A$ набагато меншим числа обумовленості матриці A , і тим самим прискорити збіжність ітераційного методу. Часто без передобумовлювача ітераційний метод взагалі не збігається.

Якщо матриця A симетрична, то бажано, щоб матриця передобумовленої системи теж була симетрична. Тому для системи $Ax = b$ передумовлену систему шукають у вигляді

$$M_1^{-1}AM_2^{-1}y = M_1^{-1}b,$$

де $y = M_2x$, а передобумовлювачі M_1 і M_2 вибирають так, щоб матриця $M_1^{-1}AM_2^{-1}$ була симетричною. Наприклад, якщо для симетричної додатно визначеної розрідженої матриці A побудовано неповне розкладання Холе-

СЬКОГО

$$A = LL^T + R,$$

то можна покласти $M_1 = L$ і $M_2 = L^T$.

4.6. Завдання до лабораторної роботи

Студенти повинні знати:

- алгоритм Ланцоша побудови ОНБ підпростору Крилова симетричної матриці;
- метод Ланцоша розв'язання СЛАР;
- метод спряжених градієнтів (CG);
- попереднє обумовлення СЛАР;
- Неповне розкладання Холеського.

Завдання

Функція $A = \text{gallery}('wathen', nx, ny)$ обчислює розріджену матрицю A розміру $n \times n$, де $n = 3 * nx * ny + 2 * nx + 2 * ny + 1$, яка з'являється при апроксимації методом скінченних елементів деякої крайової задачі у прямокутнику (параметри задачі при кожному виклику функції `gallery` змінюються випадково) і є симетричною та додатно визначеною. Параметри nx , ny визначають точність апроксимації крайової задачі.

Розглянемо систему

$$Ax = b \tag{4.32}$$

та попередньо обумовлену систему

$$L^{-1}AL^{-T}y = L^{-1}b, \tag{4.33}$$

де b — довільний вектор, L — множник неполного розкладання Холеського матриці

$$A = LL^T + R.$$

Неповне розкладання Холеського у пакеті MATLAB можна побудувати функцією

$$[R] = cholinc(A, '0'),$$

де $R = L^T$.

Скласти програму мовою пакета MATLAB, яка визначає скільки ітерацій методу спряжених градієнтів треба провести, щоб початкова нев'язка (для $x_0 = 0$) системи (4.32) (системи (4.33)) зменшилася в 10^6 разів.

Провести обчислювальний експеримент із матрицями A різної розмірності. Результати експерименту занести до таблиці

Результати обчислювального експеримента

nx, ny	$n = size(A, 1)$	$k (Ax = b)$	$k (L^{-1}AL^{-T}y = L^{-1}b)$
(nx, ny)			
$2(nx, ny)$			
$4(nx, ny)$			
$6(nx, ny)$			
$8(nx, ny)$			
$10(nx, ny)$			

Варіанти завдань

Вар.	nx	ny
1	3	3
2	4	4
3	5	5
4	6	6
5	3	4
6	3	5
7	3	6
8	4	5
9	5	6
10	5	6

До звіту включити:

1. текст програми мовою пакета MATLAB;
2. результати обчислювального експерименту;
3. висновки.

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 ПОДАТОК

Навчальне видання

МЕТОДИ ОБЧИСЛЕНЬ

Частина I

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

Укладачі

Вербіцький Віктор Васильович

Платонова Євгенія Вікторівна

В авторській редакції

Підп. до друку 31.10.2023. Формат 60x84/16.

Папір офсетний. Цифровий друк.

Ум.-друк. арк. 3,02. Наклад 30 прим. Зам. № 2312-11.

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