

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

І. А. Дичка, М. В. Онай, Т. М. Заболотня

МЕТОДИ ПОБУДОВИ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ МАТРИЧНИХ ОБЧИСЛЕНЬ: КОМП'ЮТЕРНИЙ ПРАКТИКУМ

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для студентів,
які навчаються за спеціальністю 121 «Інженерія програмного
забезпечення», освітньою програмою «Інженерія програмного забезпечення
мультимедійних та інформаційно-пошукових систем»*

Київ
КПІ ім. Ігоря Сікорського
2023

Рецензент *Романкевич Віталій Олексійович, д-р техн. наук, проф.*

Відповідальний редактор *Онай Микола Володимирович, канд. техн. наук, доц.*

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського
(протокол №08 від 02.06.2023 р.)
за поданням Вченої ради факультету прикладної математики
(протокол №10 від 29.05.2023 р.)*

Електронне мережне навчальне видання

*Дичка Іван Андрійович, д-р техн. наук, проф.
Онай Микола Володимирович, канд. техн. наук, доц.
Заболотня Тетяна Миколаївна, канд. техн. наук, доц.*

МЕТОДИ ПОБУДОВИ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ МАТРИЧНИХ ОБЧИСЛЕНЬ: КОМП'ЮТЕРНИЙ ПРАКТИКУМ

Методи побудови програмних засобів для матричних обчислень: комп'ютерний практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 121 «Інженерія програмного забезпечення», освітньої програми «Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем» / І. А. Дичка, М. В. Онай, Т. М. Заболотня; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 2.07 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2023. – 146 с.

Навчальний посібник розроблено для ознайомлення студентів з теоретичними відомостями та практичними прийомами розв'язання математичних задач за допомогою комп'ютерних засобів, а також вимогами до виконання лабораторних робіт, зокрема правилами їх оформлення. Навчальне видання призначене для студентів, які навчаються за спеціальністю 121 Інженерія програмного забезпечення, освітньою програмою «Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем» факультету прикладної математики КПІ ім. Ігоря Сікорського.

© І. А. Дичка, М. В. Онай, Т. М. Заболотня 2023

© КПІ ім. Ігоря Сікорського, 2023

ЗМІСТ

Вступ.....	4
Загальні вимоги до оформлення звіту з лабораторних робіт.....	5
Лабораторна робота №1. Часткова алгебраїчна проблема власних значень.....	7
Лабораторна робота №2. Повна алгебраїчна проблема власних значень.....	38
Лабораторна робота №3. Перший етап сингулярного розкладення прямокутної матриці	62
Лабораторна робота №4. Другий етап сингулярного розкладення прямокутної матриці та його застосування.....	81
Лабораторна робота №5. Системи нелінійних рівнянь.....	94
Лабораторна робота №6. Апроксимація функцій	115

ВСТУП

Прості математичні задачі малої розмірності, що вивчаються в курсі математичного аналізу та лінійної алгебри, допускають можливість отримання аналітичних розв'язків. В той же час при моделюванні систем та процесів виникають складні математичні задачі великої розмірності, які можуть не мати аналітичних розв'язків та вимагати застосування чисельних методів виконання не тривіальних матричних операцій, що вивчаються в рамках цієї дисципліни. До того ж складні математичні задачі розв'язувати вручну майже неможливо, тому виникає необхідність у застосуванні комп'ютерних засобів у поєднанні з чисельними методами для розв'язання таких задач. Тому актуальним є вивчення методів побудови програмних засобів для матричних обчислень.

У цьому навчальному посібнику розглядаються основні принципи та практичні прийоми побудови методів та алгоритмів розв'язання математичних задач на комп'ютері. Навчальний посібник складається з 6 розділів, кожен з яких присвячений виконанню певної лабораторної роботи з дисципліни «Методи побудови програмних засобів для матричних обчислень», яка входить до фахового каталогу вибіркових дисциплін навчального плану підготовки бакалаврів за спеціальністю 121 «Інженерія програмного забезпечення».

В кожному розділі надаються короткі теоретичні відомості з певної теми, завдання на лабораторну роботу з цієї теми, вказівки щодо виконання завдання, а також наводяться вимоги до оформлення звіту з виконаної лабораторної роботи та контрольні питання для самоперевірки.

Лабораторні роботи з дисципліни «Методи побудови програмних засобів для матричних обчислень» розраховані на 36 академічних годин аудиторних занять.

Загальні вимоги до оформлення звіту з лабораторних робіт

Лабораторна робота має бути подана в електронному та друкованому вигляді.

Електронна версія зберігається в банку даних кафедри програмного забезпечення комп'ютерних систем ФПМ КПІ ім. Ігоря Сікорського. Файл з копією лабораторної роботи здається на кафедру разом з друкованим примірником безпосередньо під час захисту. Формат файлу *.docx або *.doc, або *.rtf, або *.pdf.

Звіт необхідно друкувати на одному боці аркуша білого паперу формату А4 (210x297 мм).

Основний текст звіту має бути набраний з дотриманням таких вимог:

- шрифт Times New Roman 14 пт;
- відступ першого рядка 12.5 мм ;
- міжрядковий інтервал 1.5;
- вирівнювання по ширині;
- поля: верхнє та нижнє – 20 мм; ліве – 30 мм; праве – 15 мм;
- від краю до верхнього/нижнього колонтитула 12.5 мм.

Текст в таблицях має бути набраний з дотриманням таких вимог (при необхідності дозволяється таблиці розмішувати в альбомному форматі):

- шрифт Times New Roman 12 пт;
- міжрядковий інтервал 1.0;
- інтервал перед 6 пт;
- інтервал після 6 пт.

Всі рисунки повинні мати під рисунковий напис. Підрисунковий напис вирівнюється по центру і починається зі скорочення “Рис.”, потім

ставиться пробіл та порядковий номер рисунку. Після номера рисунка ставиться крапка, пробіл та пишеться назва рисунка.

На всі рисунки розміщені у звіті має бути посилання в тексті звіту. Посилання на рисунок у тексті виконується за його номером, розташованим після скорочення “рис.”.

Нумерацію сторінок виконують арабськими цифрами. Першою сторінкою звіту з лабораторної роботи є оформлений за зразком титульний аркуш, який включають до загальної нумерації, але номер сторінки на ньому не проставляють. На всіх наступних сторінках обов’язково проставляють у правому нижньому куті номер сторінки без крапки в кінці використовуючи шрифт Times New Roman 10 пт.

На кожній сторінці, окрім титульної, в правому верхньому куті має бути надруковано прізвище, ініціали студента та номер групи.

ЛАБОРАТОРНА РОБОТА №1. ЧАСТКОВА АЛГЕБРАЇЧНА ПРОБЛЕМА ВЛАСНИХ ЗНАЧЕНЬ

Мета роботи: набути вміння та досвід використання, побудови методів розв'язання часткової проблеми власних значень, використання систем комп'ютерної математики. Окрім цього мета даної лабораторної роботи передбачає закріплення, поглиблення й узагальнення теоретичних знань щодо методів розв'язання часткової проблеми власних значень.

Теоретичні відомості

Розрізняють *повну* (алгебраїчну, або матричну) *проблему власних значень*, що передбачає знаходження всіх власних пар $\{\lambda, \mathbf{x}\}$ матриці \mathbf{A} , та *часткові проблеми власних значень*, коли необхідно знайти одне або декілька власних чисел λ та, можливо, власних векторів \mathbf{x} , що їм відповідають.

До методів розв'язання часткової проблеми власних значень відносяться:

- степеневий метод;
- метод скалярних добутків;
- метод зворотних ітерацій.

Степеневий метод

Нехай дійсна $n \times n$ -матриця \mathbf{A} має рівно n лінійно незалежних власних векторів (базис):

$$\mathbf{x}_1 = \begin{pmatrix} x_{11} \\ x_{12} \\ \dots \\ x_{1n} \end{pmatrix}, \quad \mathbf{x}_2 = \begin{pmatrix} x_{21} \\ x_{22} \\ \dots \\ x_{2n} \end{pmatrix}, \quad \dots, \quad \mathbf{x}_n = \begin{pmatrix} x_{n1} \\ x_{n2} \\ \dots \\ x_{nn} \end{pmatrix}. \quad (1.1)$$

Припустимо, що нумерація цих векторів відповідає впорядкуванню за незростанням власних чисел, що їм відповідають, де перша з нерівностей строга (такий випадок має місце коли матриця \mathbf{A} є дійсною та всі її компоненти додатні):

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|. \quad (1.2)$$

Поставимо задачу: наближено знайти найбільше за модулем власне число λ_1 та власний вектор \mathbf{x}_1 , що йому відповідає.

Візьмемо довільний ненульовий вектор $\mathbf{y}^{(0)}$ та запишемо його розкладення по базису з власних векторів $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$:

$$\mathbf{y}^{(0)} = \sum_{i=1}^n c_i \mathbf{x}_i. \quad (1.3)$$

Будемо вважати, що $c_1 \neq 0$.

Помножимо розкладення вектора $\mathbf{y}^{(0)}$ за базисом $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ зліва на матрицю \mathbf{A} :

$$\mathbf{y}^{(1)} = \mathbf{A}\mathbf{y}^{(0)} = \sum_{i=1}^n c_i \mathbf{A}\mathbf{x}_i. \quad (1.4)$$

Очевидно, що k -те перетворення вектора $\mathbf{y}^{(0)}$ за допомогою матриці \mathbf{A} дає вектор

$$\mathbf{y}^{(k)} = \mathbf{A}\mathbf{y}^{(k-1)} = \mathbf{A}^k \mathbf{y}^{(0)} = \sum_{i=1}^n c_i \lambda_i^k \mathbf{x}_i \quad (1.5)$$

Розглянемо відношення компонент вектора $\mathbf{y}^{(k)}$ до компонент вектора $\mathbf{y}^{(k-1)}$:

$$\frac{y_j^{(k)}}{y_j^{(k-1)}} = \frac{\sum_{i=1}^n c_i \lambda_i^k x_{ij}}{\sum_{i=1}^n c_i \lambda_i^{k-1} x_{ij}} = \lambda_1 \cdot \frac{1 + \sum_{i=2}^n \frac{c_i}{c_1} \cdot \frac{x_{ij}}{x_{1j}} \left(\frac{\lambda_i}{\lambda_1}\right)^k}{1 + \sum_{i=2}^n \frac{c_i}{c_1} \cdot \frac{x_{ij}}{x_{1j}} \left(\frac{\lambda_i}{\lambda_1}\right)^{k-1}} \xrightarrow{k \rightarrow \infty} \lambda_1. \quad (1.6)$$

На основі цих міркувань можна побудувати наступні алгоритми.

Алгоритм №1 реалізації степеневого методу:

1. Задати n -мірний вектор $\mathbf{y}^{(0)}$ та числа $\delta > 0$, $\varepsilon > 0$ і $\lambda^{(0)}$, а також задати значення p (на ітераціях кратних p будемо проводити нормування).

2. Обчислити вектор $\mathbf{z}^{(0)} = \frac{\mathbf{y}^{(0)}}{\|\mathbf{y}^{(0)}\|}$ та прийняти $k = 1$.

3. Обчислити вектор $\mathbf{y}^{(k)} = \mathbf{A}\mathbf{z}^{(k-1)}$.

4. Обчислити відношення $\lambda_i^{(k)} = \frac{y_i^{(k)}}{z_i^{(k-1)}}$ при $i \in S_{k-1}$.

5. Якщо $k \bmod p = 0$, то виконати нормування $\mathbf{z}^{(k)} = \frac{\mathbf{y}^{(k)}}{\|\mathbf{y}^{(k)}\|}$,

інакше $\mathbf{z}^{(k)} = \mathbf{y}^{(k)}$.

6. Якщо критерій зупинки ітераційного процесу не виконується, то збільшити значення k на 1 та перейти до п.3 алгоритму, інакше

завершити ітераційний процес, за власне значення λ_1 прийняти усереднене значення

$$\lambda_1 = \frac{\sum_{i \in S_{k-1}} \lambda_i^{(k)}}{|S_{k-1}|},$$

а за нормований власний вектор \mathbf{x}_1 – вектор $\mathbf{z}^{(k)}$.

Можна використовувати один з двох критеріїв зупинки ітераційного процесу: $|\lambda_i^{(k)} - \lambda_i^{(k-1)}| \leq \varepsilon$ при всіх $i \in S_{k-1}$ або $\|\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\| \leq \varepsilon$.

Алгоритм №1 реалізації степеневого методу з використанням Δ^2 -процесу Ейткена:

1. Задати n -мірний вектор $\mathbf{y}^{(0)}$ та числа $\delta > 0$, $\varepsilon > 0$ і $\lambda^{(0)}$, а також задати значення p (на ітераціях кратних p будемо проводити нормування) та q (на ітераціях кратних q будемо проводити покоординатне уточнення вектора $\mathbf{z}^{(k)}$).
2. Обчислити вектор $\mathbf{z}^{(0)} = \frac{\mathbf{y}^{(0)}}{\|\mathbf{y}^{(0)}\|}$ та прийняти $k = 1$.
3. Обчислити вектор $\mathbf{y}^{(k)} = \mathbf{A}\mathbf{z}^{(k-1)}$.
4. Обчислити відношення $\lambda_i^{(k)} = \frac{y_i^{(k)}}{z_i^{(k-1)}}$ при $i \in S_{k-1}$.
5. Якщо $k \bmod p = 0$, то виконати нормування $\mathbf{z}^{(k)} = \frac{\mathbf{y}^{(k)}}{\|\mathbf{y}^{(k)}\|}$,
інакше $\mathbf{z}^{(k)} = \mathbf{y}^{(k)}$.
6. Якщо $k \bmod q = 0$, то провести покоординатне уточнення

$$\tilde{z}_i^{(k)} = z_i^{(k-2)} - \frac{\left(z_i^{(k-1)} - z_i^{(k-2)}\right)^2}{z_i^{(k)} - 2 \cdot z_i^{(k-1)} + z_i^{(k-2)}}, \quad i = \overline{1..n}.$$

7. Якщо критерій зупинки ітераційного процесу не виконується, то збільшити значення k на 1 та перейти до п.3 алгоритму, інакше завершити ітераційний процес, обчислити

$$\tilde{\lambda}_i^{(k)} = \lambda_i^{(k-2)} - \frac{\left(\lambda_i^{(k-1)} - \lambda_i^{(k-2)}\right)^2}{\lambda_i^{(k)} - 2 \cdot \lambda_i^{(k-1)} + \lambda_i^{(k-2)}}, \quad i \in S_{k-1}.$$

та за власне значення λ_1 прийняти усереднене значення

$$\lambda_1 = \frac{\sum_{i \in S_{k-1}} \tilde{\lambda}_i^{(k)}}{|S_{k-1}|},$$

а за нормований власний вектор \mathbf{x}_1 – вектор $\mathbf{z}^{(k)}$.

Можна використовувати один з двох критеріїв зупинки ітераційного процесу: $\left|\lambda_i^{(k)} - \lambda_i^{(k-1)}\right| \leq \varepsilon$ при всіх $i \in S_{k-1}$ або $\|\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\| \leq \varepsilon$.

Алгоритм №2 реалізації степеневого методу:

1. Задати n -мірний вектор $\mathbf{y}^{(0)}$ та числа $\delta > 0$, $\varepsilon > 0$ і $\lambda^{(0)}$, r – число, яке визначає початковий показник степеня матриці \mathbf{A} , а також задати значення p (на ітераціях кратних p будемо проводити нормування).

2. Обчислити вектор $\mathbf{z}^{(0)} = \frac{\mathbf{y}^{(0)}}{\|\mathbf{y}^{(0)}\|}$, $\mathbf{B} = \mathbf{A}^{2r}$ та прийняти $k = 1$.

3. Обчислити вектор $\mathbf{y}^{(k)} = \mathbf{B} \mathbf{z}^{(k-1)}$.

4. Обчислити відношення $\lambda_i^{(k)} = \frac{y_i^{(k)}}{z_i^{(k-1)}}$ при $i \in S_{k-1}$.

5. Якщо $k \bmod p = 0$, то виконати нормування $\mathbf{z}^{(k)} = \frac{\mathbf{y}^{(k)}}{\|\mathbf{y}^{(k)}\|}$,

інакше $\mathbf{z}^{(k)} = \mathbf{y}^{(k)}$.

6. Якщо критерій зупинки ітераційного процесу не виконується, то збільшити значення k на 1 та перейти до п.3 алгоритму, інакше завершити ітераційний процес та перейти до п.7.

7. Обчислити $\mathbf{y}^{(k+1)} = \mathbf{A}\mathbf{z}^{(k)}$ та відношення $\lambda_i^{(k+1)} = \frac{y_i^{(k+1)}}{z_i^{(k)}}$ при $i \in S_k$ і

за власне значення λ_1 прийняти усереднене значення

$$\lambda_1 = \frac{\sum_{i \in S_k} \lambda_i^{(k+1)}}{|S_k|},$$

а за нормований власний вектор \mathbf{x}_1 – вектор $\frac{\mathbf{y}^{(k+1)}}{\|\mathbf{y}^{(k+1)}\|}$.

Можна використовувати один з двох критеріїв зупинки ітераційного процесу: $|\lambda_i^{(k)} - \lambda_i^{(k-1)}| \leq \varepsilon$ при всіх $i \in S_{k-1}$ або $\|\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\| \leq \varepsilon$.

Для останнього алгоритму також можна виконувати прискорення за допомогою Δ^2 -процесу Ейткена, аналогічно як в алгоритмі №1.

Алгоритм реалізації степеневого методу для знаходження другої власної пари (використовується коли знайдена перша власна пара $\{\lambda_1; \mathbf{x}_1\}$):

1. Задати точність $\varepsilon \geq \varepsilon_1$ (ε_1 – точність з якою була знайдена степеневим методом перша власна пара), $\delta > 0$, $\lambda_i^{(0)}$, $i = \overline{1..n}$ та n -мірний вектор $\mathbf{y}^{(0)}$ такий, що $\|\mathbf{y}^{(0)} - \mathbf{x}_1\| \geq 100 \cdot \varepsilon$ (в іншому випадку

може спостерігатись збіжність ітераційного процесу до першої власної пари). Прийняти $k = 1$.

2. Обчислити вектор $\mathbf{y}^{(k)} = \mathbf{A}\mathbf{y}^{(k-1)}$.
3. Обчислити відношення $\lambda_i^{(k)} = \frac{y_i^{(k)} - \lambda_1 y_i^{(k-1)}}{y_i^{(k-1)} - \lambda_1 y_i^{(k-2)}}$ при $i \in S_{k-1}$, де S_{k-1} – множина індексів для яких $y_i^{(k-1)} - \lambda_1 y_i^{(k-2)} > \delta$.
4. Якщо критерій зупинки ітераційного процесу не виконується, то збільшити значення k на 1 та перейти до п.2, інакше до п.5.

Критерієм зупинки ітераційного процесу є виконання нерівності $\left| \lambda_i^{(k)} - \lambda_i^{(k-1)} \right| \leq \varepsilon$ для всіх $i \in S_{k-1}$ при цьому якщо знаки компонент $\lambda_i^{(k)}$, $i \in S_{k-1}$ на певній ітерації є однаковими, то необхідно за допомогою прийому Гарвіка контролювати покомпонентну збіжність. Якщо нерівність Гарвіка сигналізує, що порушується покомпонентна збіжність, то незалежно від виконання критерію зупинки ітераційного процесу виконати наступне: $\mathbf{y}^{(k)} = \mathbf{y}^{(k-1)}$, $\lambda_i^{(k)} = \lambda_i^{(k-1)}$ для всіх $i \in S_{k-1}$, зменшити значення k на 1 та перейти до п.5.

5. Завершити ітераційний процес, за власне значення λ_2 прийняти усереднене значення

$$\lambda_2 = \frac{\sum_{i \in S_{k-1}} \lambda_i^{(k)}}{|S_{k-1}|},$$

а за нормований власний вектор $\mathbf{x}_2 = \frac{\mathbf{y}^{(k)} - \lambda_1 \mathbf{y}^{(k-1)}}{\left\| \mathbf{y}^{(k)} - \lambda_1 \mathbf{y}^{(k-1)} \right\|}$.

Алгоритм реалізації методу слідів:

1. Задати $\varepsilon > 0$, обчислити $\lambda^{(0)} = \frac{Tr \mathbf{A}^2}{Tr \mathbf{A}}$ та $\mathbf{B} = \mathbf{A}^m$, де $m = 2^p$.

Прийняти $k = 1$.

2. Обчислити $\lambda^{(k)} = \frac{Tr(\mathbf{B} \cdot \mathbf{A})}{Tr \mathbf{B}}$.

3. Якщо $|\lambda^{(k)} - \lambda^{(k-1)}| > \varepsilon$, то обчислити $\mathbf{B} = \mathbf{B} \cdot \mathbf{B}$, збільшити значення k на 1 та перейти до п.2 алгоритму, інакше завершити ітераційний процес.

Метод скалярних добутків

Розглянемо симетричні додатновизначені матриці та евклідовий простір \mathbb{R}^n .

Нехай вектор $\mathbf{y}^{(k)}$ будується за формулою (1.5) визначеною в степеневому методі. Знайдемо скалярні добутки $(\mathbf{y}^{(k)}; \mathbf{y}^{(k)})$ та $(\mathbf{y}^{(k)}; \mathbf{y}^{(k-1)})$, враховуючи ортонормованість власних векторів, тобто умову $(\mathbf{x}_i, \mathbf{x}_j) = \delta_{ij}$ при $i, j \in \{1, 2, \dots, n\}$:

$$\begin{aligned} (\mathbf{y}^{(k)}; \mathbf{y}^{(k)}) &= \sum_{i=1}^n c_i^2 \lambda_i^{2k} = \\ &= c_1^2 \lambda_1^{2k} \left(1 + \sum_{i=2}^n \left(\frac{c_i}{\lambda_1} \right)^2 \left(\frac{\lambda_i}{\lambda_1} \right)^{2k} \right); \end{aligned} \quad (1.7)$$

$$\begin{aligned} (\mathbf{y}^{(k)}; \mathbf{y}^{(k-1)}) &= \sum_{i=1}^n c_i^2 \lambda_i^{2k-1} = \\ &= c_1^2 \lambda_1^{2k-1} \left(1 + \sum_{i=2}^n \left(\frac{c_i}{\lambda_1} \right)^2 \left(\frac{\lambda_i}{\lambda_1} \right)^{2k-1} \right). \end{aligned} \quad (1.8)$$

Легко бачити, що відношення цих чисел

$$\frac{\left(\mathbf{y}^{(k)}; \mathbf{y}^{(k)}\right)}{\left(\mathbf{y}^{(k)}; \mathbf{y}^{(k-1)}\right)} = \lambda_1 \cdot \frac{1 + \sum_{i=2}^n \left(\frac{c_i}{\lambda_1}\right)^2 \left(\frac{\lambda_i}{\lambda_1}\right)^{2k}}{1 + \sum_{i=2}^n \left(\frac{c_i}{\lambda_1}\right)^2 \left(\frac{\lambda_i}{\lambda_1}\right)^{2k-1}} \xrightarrow{k \rightarrow \infty} \lambda_1 \quad (1.9)$$

Модифікація степеневого методу, що базується на такому підході називається *методом скалярних добутків*.

Реалізувати його можливо у вигляді наступних алгоритмів.

Алгоритм №1 реалізації методу скалярних добутків:

1. Задати n -мірний вектор $\mathbf{y}^{(0)} \neq \mathbf{0}$ та числа $\varepsilon > 0$ і $\lambda^{(0)}$, прийняти $k = 1$.

2. Обчислити $s^{(0)} = \left(\mathbf{y}^{(0)}; \mathbf{y}^{(0)}\right)$, $p^{(0)} = \left\| \mathbf{y}^{(0)} \right\|_2 = \sqrt{s^{(0)}}$ та $\mathbf{z}^{(0)} = \frac{\mathbf{y}^{(0)}}{p^{(0)}}$.

3. Знайти $\mathbf{y}^{(k)} = \mathbf{A}\mathbf{z}^{(k-1)}$.

4. Обчислити скаляри $s^{(k)} = \left(\mathbf{y}^{(k)}; \mathbf{y}^{(k)}\right)$, $t^{(k)} = \left(\mathbf{y}^{(k)}; \mathbf{z}^{(k-1)}\right)$ і

$$p^{(k)} = \left\| \mathbf{y}^{(k)} \right\|_2 = \sqrt{s^{(k)}}.$$

5. Знайти наближення до найбільшого за абсолютною величиною власного числа $\lambda^{(k)} = \frac{s^{(k)}}{t^{(k)}}$.

6. Обчислити наближення до нормованого власного вектора $\mathbf{z}^{(k)} = \frac{\mathbf{y}^{(k)}}{p^{(k)}}$

7. Якщо $\left| \lambda^{(k)} - \lambda^{(k-1)} \right| \leq \varepsilon$ та/або $\left\| \mathbf{z}^{(k)} - \mathbf{z}^{(k-1)} \right\| \leq \varepsilon$, то завершити роботу алгоритму, за власне значення λ_1 прийняти $\lambda^{(k)}$, а за

нормований власний вектор \mathbf{x}_1 – вектор $\mathbf{z}^{(k)}$, інакше збільшити значення k на одиницю та перейти до третього кроку алгоритму.

Алгоритм №2 реалізації методу скалярних добутків:

1. Задати n -мірний вектор $\mathbf{y}^{(0)} \neq \mathbf{0}$ та числа $\varepsilon > 0$ і $\lambda^{(0)}$, прийняти $k = 1$.

2. Пронормувати заданий вектор: $\mathbf{z}^{(0)} = \frac{\mathbf{y}^{(0)}}{\|\mathbf{y}^{(0)}\|_2}$.

3. Знайти $\mathbf{y}^{(k)} = \mathbf{A}\mathbf{z}^{(k-1)}$.

4. Обчислити наближення до найбільшого за абсолютною величиною

$$\text{власного числа } \lambda^{(k)} = \frac{(\mathbf{y}^{(k)}; \mathbf{z}^{(k-1)})}{(\mathbf{z}^{(k-1)}; \mathbf{z}^{(k-1)})}.$$

5. Обчислити наближення до нормованого власного

$$\text{вектора } \mathbf{z}^{(k)} = \frac{\mathbf{y}^{(k)}}{\|\mathbf{y}^{(k)}\|_2}.$$

6. Якщо $|\lambda^{(k)} - \lambda^{(k-1)}| \leq \varepsilon$ та/або $\|\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\| \leq \varepsilon$, то завершити роботу алгоритму, за власне значення λ_1 прийняти $\lambda^{(k)}$, а за нормований власний вектор \mathbf{x}_1 – вектор $\mathbf{z}^{(k)}$, інакше збільшити значення k на одиницю та перейти до третього кроку алгоритму.

Наявність ортонормованого базиса з власних векторів матриці \mathbf{A} дозволяє для послідовного обчислення власних пар $\{\lambda_i; \mathbf{x}_i\}$ при $i \geq 2$ застосовувати спеціальні формули та на їх основі побудувати наступні алгоритми.

Алгоритм №1 реалізації методу скалярних добутків для пошуку j -ої власної пари (вважається, що старші $j-1$ власних пар є відомими):

1. Задати n -мірний вектор $\mathbf{y}^{(0)} \neq \mathbf{0}$ та числа $\varepsilon > 0$ і $\lambda^{(0)}$, а також задати значення q (на ітераціях кратних q будемо проводити ортогоналізацію вектора $\mathbf{z}^{(k)}$) та прийняти $k = 1$.
2. Провести ортогоналізацію цього вектора: $\mathbf{y}^{(0)} = \mathbf{y}^{(0)} - \sum_{i=1}^{j-1} (\mathbf{y}^{(0)}; \mathbf{x}_i) \mathbf{x}_i$.
3. Обчислити скаляри $s^{(0)} = (\mathbf{y}^{(0)}; \mathbf{y}^{(0)})$ та $p^{(0)} = \|\mathbf{y}^{(0)}\|_2 = \sqrt{s^{(0)}}$ та вектор $\mathbf{z}^{(0)} = \frac{\mathbf{y}^{(0)}}{p^{(0)}}$.
4. Обчислити вектор $\mathbf{y}^{(k)} = \mathbf{A}\mathbf{z}^{(k-1)}$ та скаляри $s^{(k)} = (\mathbf{y}^{(k)}; \mathbf{y}^{(k)})$, $t^{(k)} = (\mathbf{y}^{(k)}; \mathbf{z}^{(k-1)})$ і $p^{(k)} = \|\mathbf{y}^{(k)}\|_2 = \sqrt{s^{(k)}}$.
5. Обчислити наближення до шуканого власного числа $\lambda^{(k)} = \frac{s^{(k)}}{t^{(k)}}$.
6. Обчислити наближення до нормованого власного вектора $\mathbf{z}^{(k)} = \frac{\mathbf{y}^{(k)}}{p^{(k)}}$
8. Якщо $k \bmod q = 0$, то провести ортогоналізацію $\mathbf{z}^{(k)} = \mathbf{z}^{(k)} - \sum_{i=1}^{j-1} (\mathbf{z}^{(k)}; \mathbf{x}_i) \mathbf{x}_i$.
9. Якщо $|\lambda^{(k)} - \lambda^{(k-1)}| \leq \varepsilon$ та/або $\|\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\| \leq \varepsilon$, то завершити роботу алгоритму, за власне значення λ_1 прийняти $\lambda^{(k)}$, а за

нормований власний вектор \mathbf{x}_1 – вектор $\mathbf{z}^{(k)}$, інакше збільшити значення k на одиницю та перейти до четвертого кроку алгоритму.

Алгоритм №2 реалізації методу скалярних добутків для пошуку j -ої власної пари (вважається, що старші $j-1$ власних пар є відомими):

1. Задати n -мірний вектор $\mathbf{y}^{(0)} \neq \mathbf{0}$ та числа $\varepsilon > 0$ і $\lambda^{(0)}$, а також задати значення q (на ітераціях кратних q будемо проводити ортогоналізацію вектора $\mathbf{z}^{(k)}$) та прийняти $k = 1$.

2. Провести ортогоналізацію цього вектора: $\mathbf{y}^{(0)} = \mathbf{y}^{(0)} - \sum_{i=1}^{j-1} (\mathbf{y}^{(0)}; \mathbf{x}_i) \mathbf{x}_i$.

3. Пронормувати заданий вектор: $\mathbf{z}^{(0)} = \frac{\mathbf{y}^{(0)}}{\|\mathbf{y}^{(0)}\|_2}$.

4. Знайти $\mathbf{y}^{(k)} = \mathbf{A}\mathbf{z}^{(k-1)}$.

5. Обчислити наближення до шуканого власного числа

$$\lambda^{(k)} = \frac{(\mathbf{y}^{(k)}; \mathbf{z}^{(k-1)})}{(\mathbf{z}^{(k-1)}; \mathbf{z}^{(k-1)})}.$$

6. Обчислити наближення до нормованого власного

вектора $\mathbf{z}^{(k)} = \frac{\mathbf{y}^{(k)}}{\|\mathbf{y}^{(k)}\|_2}$.

7. Якщо $k \bmod q = 0$, то провести ортогоналізацію

$$\mathbf{z}^{(k)} = \mathbf{z}^{(k)} - \sum_{i=1}^{j-1} (\mathbf{z}^{(k)}; \mathbf{x}_i) \mathbf{x}_i.$$

8. Якщо $|\lambda^{(k)} - \lambda^{(k-1)}| \leq \varepsilon$ та/або $\|\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\| \leq \varepsilon$, то завершити роботу алгоритму, за власне значення λ_1 прийняти $\lambda^{(k)}$, а за

нормований власний вектор \mathbf{x}_1 – вектор $\mathbf{z}^{(k)}$, інакше збільшити значення k на одиницю та перейти до четвертого кроку алгоритму.

Метод зворотних ітерацій та його модифікації

Нами був розглянутий випадок коли вже знайдено найбільше за модулем власне значення і необхідно знайти найменше, однак бувають випадки коли необхідно знайти тільки молодшу власну пару матриці \mathbf{A} , тоді можливо обійтись і без обчислення λ_1 , застосовуючи степеневий метод до матриці \mathbf{A}^{-1} .

Легко показати, що найменшим за модулем власним значенням матриці \mathbf{A} є величина зворотна до найбільшого за модулем власного числа матриці \mathbf{A}^{-1} , яке може бути отримане застосуванням степеневого методу до матриці \mathbf{A}^{-1} .

Замість прямих ітерацій (1.5), що вимагають попереднього знаходження матриці \mathbf{A}^{-1} , можна побудувати ту саму послідовність векторів $\mathbf{y}^{(k)}$, $\mathbf{y}^{(k)}$ розв'язуючи при $k = 1, 2, 3, \dots$ лінійні системи

$$\mathbf{A}\mathbf{y}^{(k)} = \mathbf{y}^{(k-1)}. \quad (1.10)$$

Оскільки всі системи (1.10) мають одну й ту саму матрицю коефіцієнтів, то найбільш трудомістка частина метода Гаусса для їх розв'язання – LU -факторизація матриці \mathbf{A} – може бути виконана лише один раз.

Застосування зворотних ітерацій до знаходження молодшої пари матриці \mathbf{A} не вимагає написання спеціального алгоритму, достатньо лише

змінити відповідні кроки в алгоритмах, що реалізують степеневий метод та метод скалярних добутків на розв'язання СЛАР $\mathbf{A}\mathbf{y}^{(k)} = \mathbf{z}^{(k-1)}$.

В практичних задачах, інколи виникає необхідність за відомим наближеним власним значенням матриці знайти власний вектор, що йому відповідає.

Для знаходження власних векторів є зручним модифікований певним чином метод зворотних ітерацій.

Нехай для власного числа λ_j матриці простої структури \mathbf{A} відомо його наближення μ , таке, що

$$|\lambda_j - \mu| < |\lambda_i - \mu| \quad \forall i \neq j, \quad (1.11)$$

тобто число μ до власного числа λ_j , ближче ніж до будь-якого іншого власного числа матриці \mathbf{A} .

Починаючи з вектора $\mathbf{z}^{(0)}$, такого, що $\|\mathbf{z}^{(0)}\| = 1$, утворимо послідовність нормованих векторів $\mathbf{z}^{(k)}$ за формулами

$$(\mathbf{A} - \mu\mathbf{I})\mathbf{y}^{(k)} = \mathbf{z}^{(k-1)}; \quad (1.12)$$

$$\mathbf{z}^{(k)} = \frac{\mathbf{y}^{(k)}}{\|\mathbf{y}^{(k)}\|}, \quad k = 1, 2, \dots \quad (1.13)$$

Алгоритм реалізації методу зворотних ітерацій з фіксованими (сталими) зсувами для пошуку j -ої власної пари складається з таких кроків:

1. Задати вектор $\mathbf{y}^{(0)} \neq \mathbf{0}$ та числа $\varepsilon > 0$, $\delta > 0$ і μ (наближення до шуканого власного числа), прийняти $\lambda^{(0)} = \mu$ та $k = 1$.
2. Пронормувати заданий початковий вектор: $\mathbf{z}^{(0)} = \frac{\mathbf{y}^{(0)}}{\|\mathbf{y}^{(0)}\|}$.

3. Розв'язати СЛАР: $(\mathbf{A} - \mu \mathbf{I})\mathbf{y}^{(k)} = \mathbf{z}^{(k-1)}$.

4. Знайти наближення до j -го власного числа $\lambda^{(k)} = \mu + \frac{1}{|S_k|} \cdot \sum_{i \in S_k} \frac{z_i^{(k-1)}}{y_i^{(k)}}$

5. Обчислити наближення до шуканого власного

вектора $\mathbf{z}^{(k)} = \frac{\mathbf{y}^{(k)}}{\|\mathbf{y}^{(k)}\|}$.

6. Якщо $|\lambda^{(k)} - \lambda^{(k-1)}| \leq \varepsilon$ та/або $\|\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\| \leq \varepsilon$, то завершити роботу алгоритму, за власне значення λ_j прийняти $\lambda^{(k)}$, а за нормований власний вектор \mathbf{x}_j – вектор $\mathbf{z}^{(k)}$, інакше збільшити значення k на одиницю та перейти до третього кроку алгоритму.

Як показує практика метод зворотних ітерацій зі зсувом має більшу швидкість збіжності ніж звичайний степеневий метод, хоча вона також є лінійною. Але ще більшої швидкості збіжності (квадратичної) можна досягти шляхом введення змінних зсувів, що визначаються деякою послідовністю чисел $\mu_0, \mu_1, \mu_2, \dots$, що збігається до початкового власного значення. В якості таких чисел можна використовувати наближення $\lambda_j^{(k)}$ до власного значення λ_j .

Таким чином, зворотні ітерації зі змінними зсувами можна визначити наступною сукупністю рівностей:

$$(\mathbf{A} - \lambda_j^{(k-1)} \mathbf{I})\mathbf{y}^{(k)} = \mathbf{z}^{(k-1)}; \quad (1.14)$$

$$\mathbf{z}^{(k)} = \frac{\mathbf{y}^{(k)}}{\|\mathbf{y}^{(k)}\|}, \quad (1.15)$$

$$\lambda_j^{(k)} = \lambda_j^{(k-1)} + \left\langle \frac{z_i^{(k-1)}}{y_i^{(k)}} \right\rangle, \quad (1.16)$$

де $k = 1, 2, \dots$, а число $\lambda_j^{(0)}$ та вектор $\mathbf{z}^{(0)}$ ($\|\mathbf{z}^{(0)}\| = 1$) задаються на початку обчислень.

Алгоритм реалізації методу зворотних ітерацій зі змінними зсувами для пошуку j -ої власної пари:

1. Задати n -мірний вектор $\mathbf{y}^{(0)} \neq \mathbf{0}$ та числа $\varepsilon > 0$, $\delta > 0$ і $\lambda^{(0)}$ (наближення до шуканого власного числа), та прийняти $k = 1$.
2. Пронормувати заданий початковий вектор: $\mathbf{z}^{(0)} = \frac{\mathbf{y}^{(0)}}{\|\mathbf{y}^{(0)}\|}$.
3. Розв'язати СЛАР: $(\mathbf{A} - \lambda^{(k-1)} \mathbf{I})\mathbf{y}^{(k)} = \mathbf{z}^{(k-1)}$.
4. Знайти наближення до шуканого власного числа:

$$\lambda^{(k)} = \lambda^{(k-1)} + \frac{1}{|S_k|} \cdot \sum_{i \in S_k} \frac{z_i^{(k-1)}}{y_i^{(k)}}.$$

5. Обчислити наближення до шуканого власного вектора $\mathbf{z}^{(k)} = \frac{\mathbf{y}^{(k)}}{\|\mathbf{y}^{(k)}\|}$.
6. Якщо $|\lambda^{(k)} - \lambda^{(k-1)}| \leq \varepsilon$ та/або $\|\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\| \leq \varepsilon$, то завершити роботу алгоритму, за власне значення λ_j прийняти $\lambda^{(k)}$, а за нормований власний вектор \mathbf{x}_j – вектор $\mathbf{z}^{(k)}$, інакше збільшити значення k на одиницю та перейти до третього кроку алгоритму.

Потрібно розуміти різницю в реалізації зворотних ітерацій зі сталими та змінними зсувами: впершому випадку при кожному k розв'язуються

лінійні системи з однаковою матрицею коефіцієнтів, в другому випадку на різних кроках необхідно розв'язувати цілком різні системи.

Метод зворотних ітерацій зі змінним зсувом також можливо модифікувати. Якщо в якості параметра зсуву брати відношення Релея, то отримаємо метод зворотних ітерацій з відношеннями Релея або *RQI*-метод.

Алгоритм реалізації методу зворотних ітерацій з відношеннями Релея:

1. Задати n -мірний вектор $\mathbf{y}^{(0)} \neq \mathbf{0}$, число $\varepsilon > 0$ та прийняти $k = 1$.

2. Пронормувати заданий початковий вектор: $\mathbf{z}^{(0)} = \frac{\mathbf{y}^{(0)}}{\|\mathbf{y}^{(0)}\|}$.

3. Обчислити $\rho_{k-1} = \frac{(\mathbf{A}\mathbf{z}^{(k-1)}; \mathbf{z}^{(k-1)})}{(\mathbf{z}^{(k-1)}; \mathbf{z}^{(k-1)})}$.

4. Знайти $\mathbf{y}^{(k)}$ з рівняння $(\mathbf{A} - \rho_{k-1}\mathbf{I})\mathbf{y}^{(k)} = \mathbf{z}^{(k-1)}$.

5. Пронормувати $\mathbf{y}^{(k)}$, тобто $\mathbf{z}^{(k)} = \frac{\mathbf{y}^{(k)}}{\|\mathbf{y}^{(k)}\|}$.

6. Перевірити ρ_{k-1} та $\mathbf{z}^{(k)}$ на збіжність, наприклад, використовуючи прийом Гарвіка.

7. Якщо $|\rho_k - \rho_{k-1}| < \varepsilon$ та/або $\|\mathbf{y}^{(k)}\| > C$ ($C > 0$ – велика

константа), то вважається, що пара $\{\rho_{k-1}; \mathbf{z}^{(k)}\}$ є власною

парою матриці \mathbf{A} або обчислюється величина ρ_k і за власну

пару матриці \mathbf{A} береться $\{\rho_k; \mathbf{z}^{(k)}\}$. Інакше збільшити

значення k на одиницю та перейти до п. 3.

Завдання на лабораторну роботу

Розробити програму на мові програмування C# у середовищі розробки Visual Studio 2013 (або вище), яка буде працювати у віконному режимі та дозволяти наступне (точність $\varepsilon \leq 10^{-3}$):

1. Знаходити два найбільших і два найменших за абсолютною величиною власних числа матриці заданої за варіантом (*табл. 1.2*) та власних векторів, що їм відповідають степеневим методом, використовуючи алгоритм заданий за варіантом (*табл. 1.1*).
2. Виконувати завдання з п.1, але для прискорення ітераційного процесу знаходження власних чисел та векторів використовувати Δ^2 -процес Ейткена. Порівняти необхідну кількість ітерацій для знаходження власних чисел та векторів звичайним степеневим методом та із залученням Δ^2 -процесу Ейткена.
3. Знаходити найменше за абсолютною величиною власне число матриці заданої за варіантом (для алгоритмів №1 та №2 взяти матрицю з *табл. 2*, для алгоритмів №3 та №4 – з *табл. 1.3*) та власного вектора, що йому відповідає методом зворотних ітерацій, використовуючи алгоритм заданий за варіантом (*табл. 1.1*). Вибір методу розв'язання СЛАР, що з'явиться в процесі обчислень обґрунтувати.
4. Знаходити всі власні числа матриці заданої за варіантом (*табл. 1.3*) та власні вектори, що їм відповідають методом скалярних добутків, використовуючи алгоритм заданий за варіантом (*табл. 1.1*).

Для матриць заданих за варіантом, знайти всі власні числа та власні вектори, що їм відповідають у MatLab 7.0 (або вище), MathCAD 15.0 (або вище) та Mathematica 7.0 (або вище) / Maple 14.0 (або вище).

Додаткове завдання

Значення p визначити як $p = k \bmod 3$, де k – номер варіанту.

p	0	1	2
Пункти додатковго завдання за варіантом	1, 2, 3, 4	5, 6, 7	8, 9, 10, 11

Розробити програму на мові програмування C# у середовищі розробки Visual Studio 2013 (або вище), яка буде працювати у віконному режимі та дозволяти наступне (точність $\varepsilon \leq 10^{-3}$):

1. Згенерувати 5 довільних чисел в діапазоні $(\lambda_{\min} - 0.1 \cdot \lambda_{\min}; \lambda_{\max} + 0.1 \cdot \lambda_{\max})$ для матриці з табл. 1.2 і табл. 1.3.
2. Використовуючи згенеровані числа з попереднього пункту, як початкові наближення до власних чисел відповідної матриці знайти власні вектори, що відповідають цим числам, при цьому уточнюючи ці власні числа методом зворотних ітерацій з фіксованими (сталими) зсувами, використовуючи алгоритм заданий за варіантом (перша цифра в табл. 1 номер алгоритму для матриці №1, друга – для матриці №2).

3. Використовуючи згенеровані числа з п.1 додаткового завдання, як початкові наближення до власних чисел відповідної матриці знайти власні вектори, що відповідають цим числам, при цьому уточнюючи ці власні числа методом зворотних ітерацій зі змінними зсувами, використовуючи алгоритм заданий за варіантом (перша цифра в *табл. 1.1* номер алгоритму для матриці №1, друга – для матриці №2).
4. Виконувати порівняльний аналіз методів реалізованих в п.2 та п.3 додаткового завдання.
5. Знаходити власне число матриці, заданої за варіантом (*табл. 1.3*) та власний вектор, що йому відповідає методом зворотних ітерацій з відношеннями Релея.
5. Знаходити найбільше та найменше за абсолютною величиною власне число матриці заданої за варіантом (*табл. 1.2*) методом слідів.

Вимоги до оформлення звіту

Звіт має включати:

1. Постановку задачі за варіантом.
2. Математичне підґрунття для виконання лабораторної роботи: Цей розділ має містити тільки використані математичні формули при виконанні лабораторної роботи.
3. Значення власних чисел та власних векторів, матриць заданих за варіантом, отриманих кожним з методів:

Матриця (вказати номер матриці)	MatLab		MathCAD		Mathematica / Maple	
	Власні числа	Власні вектори	Власні числа	Власні вектори	Власні числа	Власні вектори

4. Значення власних чисел та власних векторів, матриць заданих за варіантом, отриманих кожним з методів:

Матриця № ____

Метод та алгоритм, що його реалізує	Кількість ітерацій	C#	
		Власні числа	Власні вектори

5. Висновки.

Варіанти завдань

Номер варіанту визначається за наступною таблицею:

№ за списком викладача	Варіант №	№ за списком викладача	Варіант №
1	9	13	20
2	1	14	25
3	8	15	2
4	22	16	3
5	4	17	15
6	24	18	7
7	18	19	14
8	5	20	21
9	13	21	6
10	12	22	23
11	17	23	16
12	19	24	11

Таблиця 1.1. Перелік алгоритмів

Номер варіанту	Номер завдання			Номер додаткового завдання	
	1	3	4	2	3
1.	1	4	2	1, 4	2, 3
2.	2	2	1	1, 3	1, 4
3.	2	3	2	2, 4	1, 3
4.	1	1	2	2, 3	2, 4
5.	1	2	1	1, 4	2, 3
6.	1	3	1	2, 4	1, 4
7.	2	3	2	1, 3	1, 3
8.	2	1	2	2, 3	2, 4
9.	1	2	2	1, 4	1, 4
10.	2	4	1	2, 4	2, 3
11.	2	3	1	1, 4	1, 4
12.	1	4	2	2, 4	1, 3
13.	1	3	2	2, 3	2, 4
14.	2	1	1	1, 4	1, 3
15.	2	4	1	2, 3	2, 3
16.	2	3	1	1, 3	1, 4
17.	1	2	1	2, 4	1, 3
18.	1	3	2	2, 3	2, 4
19.	1	4	2	1, 3	1, 3
20.	2	2	1	2, 3	1, 4
21.	1	3	2	1, 4	2, 3
22.	2	4	1	2, 4	1, 3
23.	1	4	2	2, 3	2, 4
24.	1	3	2	1, 4	2, 3

Таблиця 1.2. Перелік матриць

№	Матриця				
1	-289	-29	679	-416	534
	-166	-947	627	327	-705
	710	-107	-622	186	-752
	397	-133	347	-569	-227
	-595	-33	-404	453	-436
2	543	123	-439	-511	-140
	-682	523	983	-36	-54
	-722	616	782	-399	328
	210	-659	987	977	-275
	822	-344	763	-292	502
3	318	10	-55	122	-37
	228	975	315	124	389
	274	489	-106	-462	476
	-431	278	609	554	-117
	-619	971	194	-599	663
4	863	-758	-380	-208	-294
	-17	756	-345	553	771
	-856	129	872	661	654
	854	818	401	878	-108
	-847	78	579	599	834
5	979	578	106	880	135
	391	995	216	-146	51
	-354	557	852	-529	-83
	165	875	228	492	478
	217	45	160	530	145
6	685	-140	-617	-632	303
	-557	720	192	403	-872
	-423	388	578	-43	866
	-666	394	-167	196	-71
	-534	479	191	-112	794

Продовження табл. 1.2

№	Матриця				
7	-5	166	-606	456	-906
	243	966	428	-176	90
	-336	-201	548	300	-211
	-511	694	773	922	-922
	239	421	944	-583	982
8	-951	-2	459	-236	-788
	-47	-540	271	171	-693
	712	351	-970	-631	-982
	806	279	-790	-932	-459
	-419	-251	379	-367	-873
9	-985	503	15	364	366
	611	52	-32	-30	760
	450	136	-964	-930	-420
	-242	-923	201	-448	-674
	-811	481	-163	211	-609
10	-888	92	138	671	148
	-61	-493	-974	739	98
	-231	-208	-833	804	621
	7	42	359	-477	29
	-866	58	976	-247	-592
11	700	81	178	-3	-876
	-268	892	-124	-233	263
	817	-561	753	-478	541
	-531	174	-319	868	564
	-901	302	-349	-405	839
12	888	66	514	-89	-68
	-657	561	-853	546	13
	576	-303	990	-311	15
	-200	755	816	769	288
	-698	725	865	708	566

Продовження табл. 1.2

№	Матриця				
13	-540	-512	11	273	458
	-132	-728	161	950	-98
	301	852	-982	-827	-129
	-383	91	-878	-523	177
	467	869	-829	-635	-242
14	-691	-585	593	212	733
	-331	-308	342	-82	138
	56	-79	-778	27	267
	-134	400	-139	-758	-418
	704	-435	428	642	-743
15	834	-210	-527	-335	877
	80	732	259	420	-33
	22	572	688	37	49
	-78	811	267	426	-229
	-74	-587	424	308	117
16	-753	585	918	-479	-991
	-5	-942	-265	-291	29
	729	293	-752	-988	-194
	5	-492	-88	-583	-388
	-303	229	516	-36	-859
17	554	877	-869	-337	-247
	155	732	-508	-206	-128
	-139	-960	409	-622	322
	-331	-996	847	978	360
	-69	-43	-93	60	428
18	-973	-378	-195	683	-487
	905	-779	-69	676	362
	707	223	-868	724	135
	715	630	128	-917	408
	308	596	-572	-204	-54

Продовження табл. 1.2

№	Матриця				
19	-792	-108	826	981	431
	275	-996	-578	344	-166
	283	174	-943	610	95
	306	976	106	-705	196
	6	-932	-294	-692	-831
20	-751	-193	-2	-437	337
	405	-773	404	-834	114
	-916	-490	-488	294	915
	-76	-15	349	-913	233
	826	462	759	-953	-984
21	-456	460	-699	-650	85
	-9	-976	-621	556	-468
	-147	-274	-276	388	308
	197	-295	-791	-279	-951
	-2	-20	159	-678	-976
22	946	-172	615	385	-907
	516	121	433	297	-638
	-147	998	986	-240	882
	682	969	471	613	-397
	-499	119	-588	-103	463
23	-954	252	561	-334	-306
	-333	-650	-649	191	-35
	887	-390	-309	206	491
	201	582	815	-275	82
	-544	-900	-857	-924	-911
24	597	-109	85	-381	53
	-34	303	830	-339	-619
	-85	-59	937	254	-407
	812	443	774	578	61
	-628	-946	-855	452	555

Таблиця 1.3. Перелік матриць

№	Матриця				
1	966	665	-281	156	-155
	665	783	-237	51	-211
	-281	-237	882	-582	-429
	156	51	-582	717	100
	-155	-211	-429	100	907
2	960	-633	592	103	-198
	-633	794	-486	-203	422
	592	-486	694	23	-411
	103	-203	23	169	42
	-198	422	-411	42	702
3	996	-168	-334	-436	297
	-168	790	35	-401	-53
	-334	35	993	337	219
	-436	-401	337	723	-6
	297	-53	219	-6	793
4	932	82	-578	-519	593
	82	834	-350	-100	374
	-578	-350	834	352	-570
	-519	-100	352	653	-253
	593	374	-570	-253	898
5	740	453	619	-156	84
	453	883	762	135	67
	619	762	931	-135	53
	-156	135	-135	882	-47
	84	67	53	-47	749
6	653	-527	258	-529	-158
	-527	740	6	334	568
	258	6	995	-62	344
	-529	334	-62	773	-86
	-158	568	344	-86	827

Продовження табл. 1.3

№	Матриця				
7	861	514	-31	-87	3
	514	667	-50	455	302
	-31	-50	385	73	-379
	-87	455	73	893	281
	3	302	-379	281	644
8	919	580	-358	45	171
	580	821	-652	398	439
	-358	-652	775	-171	-352
	45	398	-171	483	314
	171	439	-352	314	862
9	888	518	406	-33	-25
	518	622	563	264	37
	406	563	990	270	-241
	-33	264	270	506	251
	-25	37	-241	251	707
10	864	307	269	-369	275
	307	537	195	-280	192
	269	195	412	-289	-4
	-369	-280	-289	375	-174
	275	192	-4	-174	377
11	897	-319	520	-119	200
	-319	839	-364	-377	-463
	520	-364	661	-109	0
	-119	-377	-109	443	446
	200	-463	0	446	613
12	942	727	-266	629	51
	727	957	-436	687	-128
	-266	-436	686	-108	-111
	629	687	-108	966	-245
	51	-128	-111	-245	239

Продовження табл. 1.3

№	Матриця				
13	816	542	83	-156	-254
	542	517	-87	-51	-152
	83	-87	961	-476	222
	-156	-51	-476	464	-207
	-254	-152	222	-207	516
14	894	207	-248	-269	-281
	207	646	-42	-42	464
	-248	-42	970	225	-15
	-269	-42	225	174	-5
	-281	464	-15	-5	917
15	613	-114	54	-39	-56
	-114	638	493	-25	472
	54	493	854	422	448
	-39	-25	422	941	252
	-56	472	448	252	895
16	885	288	-288	-329	-23
	288	966	89	-428	-152
	-288	89	639	-38	94
	-329	-428	-38	514	141
	-23	-152	94	141	259
17	644	166	-127	-246	59
	166	989	163	-104	173
	-127	163	559	-326	59
	-246	-104	-326	416	-167
	59	173	59	-167	492
18	931	585	351	-75	390
	585	867	-14	129	306
	351	-14	801	-308	-32
	-75	129	-308	356	-175
	390	306	-32	-175	839

Продовження табл. 1.3

№	Матриця				
19	845	-599	83	-512	-118
	-599	868	34	233	-42
	83	34	831	-358	-249
	-512	233	-358	839	386
	-118	-42	-249	386	385
20	818	-287	-162	184	-383
	-287	718	-161	10	3
	-162	-161	667	-588	-239
	184	10	-588	828	-4
	-383	3	-239	-4	843
21	925	-437	313	544	64
	-437	657	-187	-291	354
	313	-187	320	231	-62
	544	-291	231	996	-497
	64	354	-62	-497	948
22	771	-469	-41	751	307
	-469	895	185	-326	112
	-41	185	870	-365	188
	751	-326	-365	975	291
	307	112	188	291	609
23	521	-126	-327	-39	528
	-126	866	629	-79	-312
	-327	629	911	323	-394
	-39	-79	323	548	82
	528	-312	-394	82	944
24	482	207	-481	-45	-217
	207	855	-625	182	-357
	-481	-625	938	-275	196
	-45	182	-275	923	-99
	-217	-357	196	-99	397

Питання для самоперевірки

1. В чому полягає алгебраїчна проблема власних значень? Навести прикладні задач які зводяться до вирішення алгебраїчної проблеми власних значень.
2. В чому полягає часткова проблема власних значень? Навести приклади задач в яких виникає потреба вирішення часткової проблеми власних значень.
3. В чому полягає повна проблема власних значень? Навести приклади задач в яких виникає потреба вирішення повної проблеми власних значень.
4. Основні властивості власних чисел та векторів.
5. Основні властивості власних чисел та векторів. Матричне перетворення подібності.
6. Який клас матриць називається матрицями простої структури?
7. Що таке відношення Релея?
8. Для якого класу матриць застосовується степеневий метод?
9. Припустимо, що дійсна матриця має не кратне власне число абсолютне значення якого строго домінує над іншими у спектрі. Чи може бути це власне число комплексним? Відповідь обґрунтувати.
10. Показати, що відношення $\frac{y_j^{(k)}}{y_j^{(k-1)}}$ у степеневому методі при $k \rightarrow \infty$ буде прямувати до λ_1 .
11. В чому полягає прийом Гарвіка? Коли і з якою метою його застосовують?
12. Який характер збіжності ітераційної послідовності у степеневому методі? Відповідь обґрунтувати.

13. В чому полягає основна відмінність алгоритму №2 від алгоритму №1 степеневого методу? Які переваги та недоліки алгоритму №2 порівняно з алгоритмом №1?
14. Як знайти найменше за абсолютною величиною власне число та власний вектор, що йому відповідає степеневим методом?
15. Які особливості пошуку найбільших та найменших за абсолютною величиною кратних власних чисел та власних векторів, що їм відповідають степеневим методом?
16. Довести справедливість ітераційної формули для пошуку другого власного числа степеневим методом.
17. Як розкласти будь-який вектор за базисом, що утворюють власні вектори матриці простої структури?
18. Обґрунтувати коректність методу слідів для пошуку найбільшого за абсолютною величиною власного числа. Чи можна за допомогою методу слідів знайти власний вектор, що відповідає найбільшому за абсолютною величиною власному числу?
19. Чому використання Δ^2 -процесу Ейткена для λ в степеневому методі не дає суттєвого зменшення кількості ітерацій на відміну від покоординатного уточнення за Δ^2 -процесом Ейткена для вектора \mathbf{y} ?
20. Для якого класу матриць зазвичай використовується метод скалярних добутків розв'язання часткової проблеми власних значень? Чому саме для такого класу матриць?
21. Показати, що відношення $\frac{(\mathbf{y}^{(k)}; \mathbf{y}^{(k)})}{(\mathbf{y}^{(k)}; \mathbf{y}^{(k-1)})}$, яке використовується в алгоритмі №1 метода скалярних добутків, при $k \rightarrow \infty$ прямує до λ_1 .
22. Чим відрізняються алгоритми №1 та №2 методу скалярних добутків?

23. Довести, що відношення $\frac{\left(\mathbf{y}^{(k)}; \mathbf{y}^{(k)}\right)}{\left(\mathbf{y}^{(k)}; \mathbf{y}^{(k-1)}\right)}$ та $\frac{\left(\mathbf{y}^{(k+1)}; \mathbf{y}^{(k)}\right)}{\left(\mathbf{y}^{(k)}; \mathbf{y}^{(k)}\right)}$ мають однаковий ліміт при $k \rightarrow \infty$.
24. Чому алгоритм №2 методу скалярних добутків вимагає більше алгебраїчних операцій ніж алгоритм №1?
25. На чому ґрунтуються алгоритми №1 та №2 методу скалярних добутків пошуку j -ої власної пари коли вже знайдені старші $j - 1$ власних пар?
26. В чому полягає головна ідея методу зворотних ітерацій? В чому схожість та відмінність алгоритмів №1, №2, №3 та №4 реалізації методу зворотних ітерацій та алгоритмів №1, №2 реалізації степеневого методу та методу скалярних добутків?
27. Якими методами доцільніше розв'язувати СЛАР, що виникають в процесі розв'язання часткової проблеми власних значень методом зворотних ітерацій? Відповідь обґрунтувати.
28. Яка головна мета всіх модифікацій методу зворотних ітерацій?
29. За рахунок чого в методі зворотних ітерацій зі зсувами відбувається домінування коефіцієнтів $c_j^{(k)}$ в розкладенні ітерованого вектора за базисом з власних векторів?
30. Чи вимагає метод зворотних ітерацій зі сталими зсувами для знаходження j -ої власної пари знання інших власних пар? Відповідь обґрунтувати.
31. Що буде якщо в методі зворотних ітерацій зі зсувами пошук j -ої власної пари обрати початковий вектор ортогональним вектору \mathbf{x}_j ?
32. Які апріорні дані необхідні для того, що знайти j -ий власний вектор методом зворотних ітерацій з фіксованими (сталими) зсувами?

33. Які алгоритми реалізації методу зворотних ітерацій з фіксованими (сталими) зсувами для пошуку j -ої власної пари Ви знаєте? В чому полягає їх відмінність? Для якого класу матриць доцільно застосовувати той чи інший алгоритм?
34. Чим відрізняються метод зворотних ітерацій з фіксованими (сталими) зсувами та метод зворотних ітерацій зі змінними зсувами? Яким чином рекомендується визначати значення зсувів на кожній ітерації методу зворотних ітерацій зі змінними зсувами?
35. Які апріорні дані необхідні для того, що знайти j -ий власний вектор методом зворотних ітерацій зі змінними зсувами?
36. Які алгоритми реалізації методу зворотних ітерацій зі змінними зсувами для пошуку j -ої власної пари Ви знаєте? В чому полягає їх відмінність? Для якого класу матриць доцільно застосовувати той чи інший алгоритм?
37. В чому полягає головна ідея методу зворотних ітерацій з відношеннями Релея?

Рекомендована література

1. William H. Press Numerical Recipes in C. The Art of Scientific Computing / William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery // Cambridge University Press, 2002. – 1018 p.
2. Walter Gautschi Numerical Analysis [Електронний ресурс], 2012. Режим доступу: http://www.ikiu.ac.ir/public-files/profiles/items/090ad_1410599906.pdf
3. McDonough J. M. Computational Numerical Analysis [Електронний ресурс], 2007. Режим доступу: <http://web.engr.uky.edu/~acfd/egr537-lctrs.pdf>

ЛАБОРАТОРНА РОБОТА №2. ПОВНА АЛГЕБРАЇЧНА ПРОБЛЕМА ВЛАСНИХ ЗНАЧЕНЬ

Мета роботи: набути вміння та досвід використання методів розв'язання повної проблеми власних значень, використання систем комп'ютерної математики. Окрім цього мета даної лабораторної роботи передбачає закріплення, поглиблення й узагальнення теоретичних знань щодо методів розв'язання повної проблеми власних значень.

Теоретичні відомості

Розрізняють *повну* (алгебраїчну, або матричну) *проблему власних значень*, що передбачає знаходження всіх власних пар $\{\lambda, \mathbf{x}\}$ матриці \mathbf{A} , та *часткові проблеми власних значень*, коли необхідно знайти одне або декілька власних чисел λ та, можливо, власних векторів \mathbf{x} , що їм відповідають.

До методів розв'язання часткової проблеми власних значень відносяться:

- метод Якобі;
- LU -метод;
- QR -метод.

Метод Якобі розв'язання повної проблеми власних значень

Від розгляду часткової проблеми власних значень перейдемо до повної проблеми власних значень.

Легко перевірити, що матриця \mathbf{T}_{ij} є ортогональною при будь-яких $i = 1..n$, $j = 1..n$. Тоді матриця

$$\mathbf{B} = \mathbf{T}_{ij}^T \mathbf{A} \mathbf{T}_{ij} \quad (2.1)$$

подібна до \mathbf{A} , тобто має той самий набір власних чисел, що й матриця \mathbf{A} .

Класичний ітераційний метод обертань, запропонований Якобі, передбачає побудову послідовності матриць

$$\mathbf{B}_0 = \mathbf{A}; \mathbf{B}_1; \mathbf{B}_2; \dots; \mathbf{B}_k; \dots \quad (2.2)$$

за допомогою перетворень:

$$\mathbf{B}_k = \mathbf{T}_{ij}^T \mathbf{B}_{k-1} \mathbf{T}_{ij}, \quad (2.3)$$

такої, що на k -му кроці обнулюються максимальний за модулем елемент матриці \mathbf{B}_{k-1} .

Елементи матриці \mathbf{T}_{ij} необхідно обрати так, щоб елемент матриці \mathbf{B}_k з індексом $(i; j)$ став нульовим, а саме кут α необхідно визначати з рівності:

$$\operatorname{tg} 2\alpha = \frac{2a_{ij}}{a_{ii} - a_{jj}}. \quad (2.4)$$

Для визначеності вважають, що $\alpha \in \left(-\frac{\pi}{4}; \frac{\pi}{4}\right]$.

Зрозуміло, що немає необхідності безпосередньо знаходити кут α , оскільки потрібні для виконання перетворення числа c та s можливо отримати через значення $\operatorname{tg} 2\alpha$ за формулами тригонометрії.

Формули, що визначають один крок методу обертань Якобі мають такий вигляд:

1. Виконати присвоєння: $k = 1$, $\mathbf{B}_{k-1} = \mathbf{A}$.

2. Визначити індекси i та j ключового (приреченого) елемента $b_{ij}^{(k-1)}$.

3. Обчислити $p = 2b_{ij}^{(k-1)}$, $q = b_{ii}^{(k-1)} - b_{jj}^{(k-1)}$, $d = \sqrt{p^2 + q^2}$.

4. Обчислити $r = \frac{|q|}{2d}$, $c = \sqrt{\frac{1}{2} + r}$, $s = \text{sign}(pq) \sqrt{\frac{1}{2} - r}$. Якщо

$|p| \ll |q|$, то краще взяти $s = \text{sign}(pq) \frac{|p|}{2cd}$.

5. Обчислити нові діагональні елементи:

$$b_{ii}^{(k)} = c^2 b_{ii}^{(k-1)} + s^2 b_{jj}^{(k-1)} + 2cs b_{ij}^{(k-1)};$$

$$b_{jj}^{(k)} = s^2 b_{ii}^{(k-1)} + c^2 b_{jj}^{(k-1)} - 2cs b_{ij}^{(k-1)}.$$

6. Прийняти $b_{ij}^{(k)} = b_{ji}^{(k)} = 0$ або для контролю обчислити:

$$b_{ij}^{(k)} = b_{ji}^{(k)} = (c^2 - s^2) b_{ij}^{(k-1)} - cs (b_{ii}^{(k-1)} - b_{jj}^{(k-1)}).$$

7. При $l = 1..n$ ($l \neq i, l \neq j$), обчислити:

$$b_{il}^{(k)} = b_{li}^{(k)} = c b_{li}^{(k-1)} + s b_{lj}^{(k-1)};$$

$$b_{jl}^{(k)} = b_{lj}^{(k)} = -s b_{li}^{(k-1)} + c b_{lj}^{(k-1)}.$$

Можна довести, що за власні вектори $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ матриці \mathbf{A} можуть бути наближено прийняті стовпці результуючої матриці, що отримується праворуч від матриці \mathbf{A} в ланцюжку перетворень подібності

$$\mathbf{V}_1 = \mathbf{T}_{i_0 j_0}^T \mathbf{A} \mathbf{T}_{i_0 j_0};$$

$$\mathbf{V}_2 = \mathbf{T}_{i_1 j_1}^T \mathbf{V}_1 \mathbf{T}_{i_1 j_1} = \mathbf{T}_{i_1 j_1}^T \mathbf{T}_{i_0 j_0}^T \mathbf{A} \mathbf{T}_{i_0 j_0} \mathbf{T}_{i_1 j_1};$$

.....

$$\mathbf{V}_k = \mathbf{T}_{i_{k-1} j_{k-1}}^T \dots \mathbf{T}_{i_0 j_0}^T \mathbf{A} \mathbf{T}_{i_0 j_0} \dots \mathbf{T}_{i_{k-1} j_{k-1}}.$$

Тобто в наведеному ітераційному процесі власними векторами будуть стовпці матриці $\mathbf{T}_k = \mathbf{T}_{i_0 j_0} \mathbf{T}_{i_1 j_1} \dots \mathbf{T}_{i_{k-1} j_{k-1}}$.

LU-метод розв'язання повної несиметричної проблеми власних значень

Часто алгоритми наближеного розв'язання повних проблем власних значень засновані на приведенні початкових матриць до подібних матриць не діагонального, а трикутного виду. Найбільш простий з таких алгоритмів обчислення власних чисел ґрунтується на LU -факторизації матриць.

Нехай данну $n \times n$ -матрицю \mathbf{A} можна подати у вигляді $\mathbf{A} = \mathbf{L}\mathbf{U}$. Введемо позначення $\mathbf{A}_1 = \mathbf{U}\mathbf{L}$, звідси $\mathbf{U} = \mathbf{A}_1\mathbf{L}^{-1}$.

Нехай отримана матриця \mathbf{A}_1 також може бути розкладена у добуток нижньої та верхньої трикутної матриць, позначимо їх, як \mathbf{L}_1 та \mathbf{U}_1 . Тоді $\mathbf{A}_1 = \mathbf{L}_1\mathbf{U}_1$ і ми можемо визначити матрицю \mathbf{A}_2 , як:

$$\mathbf{A}_2 = \mathbf{U}_1\mathbf{L}_1 \Leftrightarrow \mathbf{U}_1 = \mathbf{A}_2\mathbf{L}_1^{-1}. \quad (2.5)$$

Підставляючи вираз (2.5) у $\mathbf{A}_1 = \mathbf{L}_1\mathbf{U}_1$ отримуємо:

$$\mathbf{A}_1 = \mathbf{L}_1\mathbf{A}_2\mathbf{L}_1^{-1}. \quad (2.6)$$

Отже, \mathbf{A}_1 подібна \mathbf{A}_2 та вони мають однакові власні числа.

Такий процес побудови подібних матриць і складає основу LU -алгоритму. Він визначається двома формулами:

$$\mathbf{A}_k = \mathbf{L}_k\mathbf{U}_k, \quad \mathbf{A}_{k+1} = \mathbf{U}_k\mathbf{L}_k, \quad (2.7)$$

де $\mathbf{A}_0 = \mathbf{A}$, $k = 0, 1, 2, \dots$

LU -алгоритм розв'язання повної проблеми власних значень є недостатньо чисельно стійким. Ви пам'ятаєте, що покращення цього параметра можна отримати шляхом перестановок рядків та стовпців, тобто шляхом реалізації LUP -факторизації матриці, але це сильно відбивається на обчислювальній складності цього методу. На фоні можливої нестійкості несиметричної проблеми власних значень цей факт може грати досить

суттєву роль. Також підвищити стійкість та прискорити ітераційний процес можливо шляхом певних зсувів.

В якості критерія зупинки ітераційного процесу можна розглядати малість норми строго трикутної підматриці, яка має бути нульовою.

QR-метод розв'язання повної несиметричної проблеми власних значень

Одним зі стійких методів розв'язання повної проблеми власних значень є метод, що ґрунтується на ортогональних перетвореннях матриць та називається *QR*-методом.

За своєю ідеєю *QR*-метод схожий до *LU*-методу.

Починаючи з $\mathbf{A}_0 = \mathbf{A}$ будується послідовність матриць (\mathbf{A}_k) за формулами

$$\mathbf{A}_k = \mathbf{Q}_k \mathbf{R}_k, \quad \mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{Q}_k, \quad \text{де } k = 0, 1, 2, \dots \quad (2.8)$$

На основі властивості ортогональних матриць $(\mathbf{Q}_k^T = \mathbf{Q}_k^{-1})$ у відповідності до (2.8) можливо записати подання даної матриці \mathbf{A} у вигляді

$$\mathbf{A} = \left(\prod_{i=0}^k \mathbf{Q}_i \right) \mathbf{A}_{k+1} \left(\prod_{j=k}^0 \mathbf{Q}_j^T \right).$$

При певних обмеженнях, найважливішим з яких є наявність різних за абсолютною величиною власних чисел матриці \mathbf{A} , послідовність матриць (\mathbf{A}_k) збігається до матриці правої трикутної форми на діагоналі, якої будуть знаходитись власні числа у випадку дійсного спектру такої матриці. Якщо ж дійсна матриця має комплексні власні числа, то послідовність матриць (\mathbf{A}_k) збігається до матриці блочно-трикутної структури з максимальним розміром блоків на діагоналі 2×2 , що відповідають парам комплексно-спряжених власних чисел.

Будемо називати алгоритм, що визначається формулами (2.8) QR -алгоритмом №0.

Алгоритм №0 реалізації QR -методу:

1. Прийняти $k = 0$ та $\mathbf{A}_k = \mathbf{A}$.
2. Виконати QR -розкладення матриці \mathbf{A}_k :

$$\mathbf{A}_k = \mathbf{Q}_k \mathbf{R}_k .$$

3. Знайти матрицю \mathbf{A}_{k+1} :

$$\mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{Q}_k .$$

4. Якщо критерій зупинки ітераційного процесу не виконується, то збільшити значення k на 1 та перейти до п.2.

Зрозуміло, що за допомогою $(n - 2)$ -х перетворень Хаусхолдера будь-яку дійсну $n \times n$ -матрицю можна звести до форми Хесенберга. Для цього в процесі QR -факторизації, що розглядався нами раніше, роль головної діагоналі необхідно віддавати першій головній піддіагоналі.

Один з підходів до модифікації QR -методу полягає у застосуванні формул (2.8) до послідовності матриць \mathbf{A}_i у формі Хесенберга, тобто пропонується розбити QR -алгоритм №0 на два етапи: першим етапом є приведення початкової матриці до форми Хесенберга, а другим етапом застосування формул (2.8) до отриманої матриці у формі Хесенберга. Такий різновид QR -методу назовемо QR -алгоритмом №1.

Алгоритм №1 реалізації QR -методу:

Перший етап:

1. Прийняти $\mathbf{A}_0 = \mathbf{A}$.
2. Приведення матриці до форми Хесенберга за допомогою перетворень Хаусхолдера:

$$\mathbf{A}_k = \mathbf{H}_k \mathbf{A}_{k-1} \mathbf{H}_k, \quad k = 1, 2, \dots, n - 2$$

3. Виконати присвоєння:

$$\mathbf{B}_0 = \mathbf{A}_{n-2}.$$

Другий етап:

1. Прийняти $k = 0$.

2. Виконати QR -розкладення матриці \mathbf{B}_k :

$$\mathbf{B}_k = \mathbf{Q}_k \mathbf{R}_k$$

3. Знайти матрицю \mathbf{B}_{k+1} :

$$\mathbf{B}_{k+1} = \mathbf{R}_k \mathbf{Q}_k$$

4. Якщо критерій зупинки ітераційного процесу не виконується, то збільшити значення k на 1 та перейти до п.2.

Але на практиці такий алгоритм використовується дуже рідко. Зазвичай на другому етапі QR -алгоритму використовують не формули, а ортогональне перетворення Гівенса.

Отже, ще одним різновидом QR -алгоритму є двоетапний алгоритм, де на першому етапі виконується приведення початкової матриці до форми Хесенберга, а на другому етапі отримана матриця перетвореннями Гівенса приводиться до правої трикутної форми. Назвемо такий алгоритм, що реалізує QR -метод, як QR -алгоритм №2.

Алгоритм №2.1 реалізації QR -методу:

Перший етап:

1. Прийняти $\mathbf{A}_0 = \mathbf{A}$.

2. Приведення матриці до форми Хесенберга за допомогою перетворень Хаусхолдера:

$$\mathbf{A}_k = \mathbf{H}_k \mathbf{A}_{k-1} \mathbf{H}_k, \quad k = 1, 2, \dots, n-2$$

3. Виконати присвоєння:

$$\mathbf{B}_0 = \mathbf{A}_{n-2}.$$

Другий етап:

1. Прийняти $k = 0$.
2. За допомогою перетворень Гівенса привести матрицю \mathbf{B}_k до правої трикутної форми:

$$\mathbf{R}_k = \prod_{i=n-1}^1 \mathbf{G}_i \mathbf{B}_k,$$

3. Знайти ортогональну матрицю \mathbf{Q}_k :

$$\mathbf{Q}_k = \prod_{i=1}^{n-1} \mathbf{G}_i^T,$$

4. Обчислити:

$$\mathbf{B}_{k+1} = \mathbf{R}_k \mathbf{Q}_k$$

5. Якщо критерій зупинки ітераційного процесу не виконується, то збільшити значення k на 1 та перейти до п.2.

Алгоритм №2.2 реалізації QR -методу:

Перший етап:

1. Прийняти $\mathbf{A}_0 = \mathbf{A}$.
2. Приведення матриці до форми Хесенберга за допомогою перетворень Хаусхолдера:

$$\mathbf{A}_k = \mathbf{H}_k \mathbf{A}_{k-1} \mathbf{H}_k, \quad k = 1, 2, \dots, n-2$$

3. Виконати присвоєння:

$$\mathbf{B}_0 = \mathbf{A}_{n-2}.$$

Другий етап:

1. Прийняти $k = 0$.
2. За допомогою перетворень Гівенса побудувати матрицю ортогонально-подібну до \mathbf{B}_k :

$$\mathbf{B}_{k+1} = \left(\prod_{i=n-1}^1 \mathbf{G}_i \right) \mathbf{B}_k \left(\prod_{i=1}^{n-1} \mathbf{G}_i^T \right),$$

3. Якщо критерій зупинки ітераційного процесу не виконується, то збільшити значення k на 1 та перейти до п.2.

Відмітимо, що весь QR -метод можна побудувати тільки на основі перетворень Гівенса, тобто не приводячи початкову матрицю \mathbf{A} до форми Хесенберга (або до тридіагонального вигляду, якщо \mathbf{A} симетрична) іншими перетвореннями. Такий метод називають QR -алгоритм №3 або методом Гівенса.

Алгоритм №3.1 реалізації QR -методу:

Перший етап:

1. Прийняти $\mathbf{A}_0 = \mathbf{A}$.
2. Привести матрицю \mathbf{A}_0 до форми Хесенберга за допомогою перетворень Гівенса:

$$\mathbf{B}_0 = \left(\prod_{j=2}^{n-1} \prod_{i=j+1}^n \mathbf{G}_{ij} \right) \mathbf{A}_0 \left(\prod_{j=2}^{n-1} \prod_{i=j+1}^n \mathbf{G}_{ij}^T \right),$$

Другий етап:

1. Прийняти $k = 0$.
2. За допомогою перетворень Гівенса привести матрицю \mathbf{B}_k до правої трикутної форми:

$$\mathbf{R}_k = \prod_{i=n-1}^1 \mathbf{G}_i \mathbf{B}_k,$$

3. Знайти ортогональну матрицю \mathbf{Q}_k :

$$\mathbf{Q}_k = \prod_{i=1}^{n-1} \mathbf{G}_i^T,$$

4. Обчислити:

$$\mathbf{B}_{k+1} = \mathbf{R}_k \mathbf{Q}_k$$

5. Якщо критерій зупинки ітераційного процесу не виконується, то збільшити значення k на 1 та перейти до п.2.

Алгоритм №3.2 реалізації QR -методу:

Перший етап:

1. Прийняти $\mathbf{A}_0 = \mathbf{A}$.
2. Привести матрицю \mathbf{A}_0 до форми Хесенберга за допомогою перетворень Гівенса:

$$\mathbf{V}_0 = \left(\prod_{j=2}^n \prod_{i=j+1}^n \mathbf{G}_{ij} \right) \mathbf{A}_0 \left(\prod_{j=2}^{n-1} \prod_{i=j+1}^n \mathbf{G}_{ij}^T \right),$$

Другий етап:

1. Прийняти $k = 0$.
2. За допомогою перетворень Гівенса побудувати матрицю ортогонально-подібну до \mathbf{V}_k :

$$\mathbf{V}_{k+1} = \left(\prod_{i=n-1}^1 \mathbf{G}_i \right) \mathbf{V}_k \left(\prod_{i=1}^{n-1} \mathbf{G}_i^T \right),$$

3. Якщо критерій зупинки ітераційного процесу не виконується, то збільшити значення k на 1 та перейти до п.2.

Якщо порівняти приведення несиметричних матриць до форми Хесенберга методом Гівенса та методом Хаусхолдера, то метод Гівенса вимагає більше арифметичних операцій, тому зазвичай для таких цілей віддають перевагу методу Хаусхолдера.

Можна показати, що в тому випадку коли матриця \mathbf{A} припускає строге впорядкування абсолютних величин власних чисел

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|, \quad (2.9)$$

то послідовність матриць \mathbf{A}_k в процесі збігається до матриці виду

$$\mathbf{U} = \begin{pmatrix} \lambda_1 & * & * & \dots & * \\ 0 & \lambda_1 & * & \dots & * \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \lambda_n \end{pmatrix} \quad (2.10)$$

зі швидкістю анулювання піддіагональних елементів

$$\text{якщо } \mathbf{A}_0 \text{ має форму Хесенберга, то } a_{i+1,i}^{(k)} = O\left(\left(\frac{|\lambda_{i+1}|}{|\lambda_i|}\right)^k\right) \quad (2.11)$$

$$\text{якщо } \mathbf{A}_0 \text{ не має форми Хесенберга, то } a_{i,j}^{(k)} = O\left(\left(\frac{|\lambda_i|}{|\lambda_j|}\right)^k\right) \text{ при } i > j. \quad (2.12)$$

Звідси зрозуміло, що навіть попереднє приведення початкової матриці \mathbf{A} до форми Хесенберга гарантує лише лінійну збіжність QR -алгоритму з сильним уповільненням швидкості анулювання піддіагональних елементів матриці \mathbf{A}_k в тій її частині, де на діагоналі граничної матриці \mathbf{U} розміщуються близькі за модулем власні числа.

Існують різноманітні способи пришвидшення ітераційного процесу. Найбільш вживані серед них ґрунтуються на вдалому покроковому перетворенні спектру.

Один з підходів до покрокового зміщення спектру полягає в тому, щоб на кожному кроці виконувати зміщення на величину, що дорівнює найменшому за абсолютною величиною діагональному елементу матриці. На основі цих міркувань побудуємо ще один алгоритм реалізації QR -методу.

Алгоритм №4 реалізації QR -методу:

1. Прийняти $k = 0$, $\tilde{k} = inf$, $\mu_k = 0$.

2. Провести зсув спектру: $\mathbf{A}_k = \mathbf{A} - \mu_k \mathbf{I}$.

3. Виконати QR -розкладення матриці \mathbf{A}_k :

$$\mathbf{A}_k = \mathbf{Q}_k \mathbf{R}_k.$$

4. Знайти матрицю \mathbf{A}_{k+1} :

$$\mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{Q}_k + \mu_k \mathbf{I}.$$

5. Збільшити значення k на 1

6. Виконати присвоєння:

$$\mu_k = \min(\text{diag}(\mathbf{A}_k))$$

7. Якщо критерій зупинки ітераційного процесу не виконується, то перейти до п.2.

Нехай на деякому кроці \tilde{k} алгоритму є підстави вважати справедливою наближену рівність

$$\lambda_n \approx a_{nn}^{(\tilde{k})}. \quad (2.13)$$

Зазвичай, на практиці, цей факт перевіряють за наступним критерієм

$$\left| a_{nn}^{(k)} - a_{nn}^{(k-1)} \right| \leq \frac{1}{3} \left| a_{nn}^{(k-1)} \right|. \quad (2.14)$$

Якщо виконується нерівність (2.14), то виконується і наближена рівність (2.13).

Звідси можна зробити висновок: якщо зсунути спектр матриці \mathbf{A} на величину $a_{nn}^{(\tilde{k})}$ та застосувати QR -алгоритм до матриці $\tilde{\mathbf{A}}_0 = \mathbf{A}_0 - a_{nn}^{(\tilde{k})} \mathbf{I}$, то

замість характеристики швидкості збіжності $O\left(\left(\frac{|\lambda_n|}{|\lambda_{n-1}|}\right)^k\right)$ будемо мати

характеристику $O\left(\left(\frac{|\lambda_n - a_{nn}^{(\tilde{k})}|}{|\lambda_{n-1} - a_{nn}^{(\tilde{k})}|}\right)^k\right)$, яка може виявитись значно кращою

враховуючи малість чисельника останнього виразу.

Такий зсув називається зсувом за Релеєм. Застосовуючи такий зсув при кожному k , починаючи зі значення $\tilde{k} \geq 1$, що задовольняє нерівність (2.14), отримують квадратично збіжний процес обнуління $(n-1)$ -го елемента в n -му рядку.

Щоб виписати власне число λ_n , тепер потрібно врахувати всі зміщення спектру, тобто до останнього елемента діагоналі після обнуління елемента в позиції $(n-1; n)$ потрібно додати величини всіх накопичених зсувів. Можна робити іншим чином, на кожному кроці алгоритму робити зворотні зсуви, тобто:

$$\mathbf{A}_k - \mu_k \mathbf{I} = \mathbf{Q}_k \mathbf{R}_k; \quad (2.15)$$

$$\mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{Q}_k + \mu_k \mathbf{I}, \quad (2.16)$$

де

$$\mu_k = \begin{cases} 0, & \text{якщо } k < \tilde{k}; \\ a_{nn}^{(k)}, & \text{якщо } k \geq \tilde{k}. \end{cases} \quad (2.17)$$

При цьому не варто забувати, що таким чином може бути знайдене найменше за модулем дійсне власне число, після чого виконання алгоритму продовжується зі зменшеною матрицею.

Назвемо QR -алгоритм зі зсувами QR -алгоритмом №5.

Алгоритм №5 реалізації QR -методу:

1. Прийняти $k = 0$, $\tilde{k} = inf$, $\mu_k = 0$.
2. Провести зсув спектру: $\mathbf{A}_k = \mathbf{A} - \mu_k \mathbf{I}$.
3. Виконати QR -розкладення матриці \mathbf{A}_k :

$$\mathbf{A}_k = \mathbf{Q}_k \mathbf{R}_k.$$

4. Знайти матрицю \mathbf{A}_{k+1} :

$$\mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{Q}_k + \mu_k \mathbf{I}.$$

5. Збільшити значення k на 1
6. Якщо $\tilde{k} == \text{inf}$ та виконується нерівність

$$\left| a_{nn}^{(k)} - a_{nn}^{(k-1)} \right| \leq \frac{1}{3} \left| a_{nn}^{(k-1)} \right|,$$

то $\tilde{k} = k$.

7. Виконати присвоєння:

$$\mu_k = \begin{cases} 0, & \text{якщо } k < \tilde{k}; \\ a_{nn}^{(k)}, & \text{якщо } k \geq \tilde{k}. \end{cases}$$

8. Якщо критерій зупинки ітераційного процесу не виконується, то перейти до п.2.

Відмітимо, що наявність кратних дійсних власних значень не повинна відбиватися на роботі алгоритму – всі вони можуть бути їм послідовно знайдені.

Сформулюємо без доведення ще дві важливі теореми.

Теорема про єдиність \mathbf{Q} . Нехай \mathbf{Q}_1 та \mathbf{Q}_2 – дійсні ортогональні матриці, що трансформують деяку матрицю \mathbf{A} до нерозкладних матриць Хесенберга:

$$\mathbf{B}_{\mathbf{Q}_1} = \mathbf{Q}_1^T \mathbf{A} \mathbf{Q}_1; \quad (2.18)$$

$$\mathbf{B}_{\mathbf{Q}_2} = \mathbf{Q}_2^T \mathbf{A} \mathbf{Q}_2. \quad (2.19)$$

Якщо перші стовпці матриць \mathbf{Q}_1 та \mathbf{Q}_2 співпадають, то знайдеться діагональна ортогональна матриця \mathbf{D} (тобто діагональна матриця з ± 1 на головній діагоналі) така, що

$$\mathbf{Q}_2 = \mathbf{Q}_1 \mathbf{D}; \quad (2.20)$$

$$\mathbf{B}_{\mathbf{Q}_2} = \mathbf{D} \mathbf{B}_{\mathbf{Q}_1} \mathbf{D}. \quad (2.21)$$

Іншими словами, матриці Q_1 та Q_2 можуть відрізнятися лише знаками стовпців, а хесенбергові матриці B_{Q_1} та B_{Q_2} – тільки знаками піддіагональних елементів.

Теорема про неявне Q або неявна Q -теорема. Нехай $Q^T A Q = B$ – нерозкладна верхня матриця Хесенберга. Тоді перший стовпець матриці Q однозначно (з точністю до знаків) визначає її другий, третій, ... n -й стовпець.

Верхня хесенбергова матриця називається нерозкладною, якщо всі елементи на її першій піддіагоналі відмінні від нуля. Якщо на деякому кроці алгоритму в матриці Хесенберга з'явиться нульовий піддіагональний елемент, то звідси буде слідувати її розкладеність, тобто алгоритм далі можна буде продовжувати з матрицями меншого розміру.

Варто відмітити, що існують матриці, для яких досить складно шукати всі власні числа та власні вектори.

Прикладом симетричної матриці, для якої складно знайти всі власні числа та власні вектори, що їм відповідають є матриця Ресера.

Ця матриця слугувала каменем спотикання для багатьох алгоритмів обчислення власних значень. Тільки QR -алгоритм Френсіса, вдосконалений Уїлкінсоном дозволяє розібратися з даною матрицею. Матриця Ресера – це матриця восьмого порядку з цілочисельними елементами, яка має наступний спектр:

- пара кратних власних значень;
- 3 близьких власних значення;
- нульове власне значення;
- мале ненульове власне значення.

Завдання на лабораторну роботу

Розробити програму на мові програмування C# у середовищі розробки Visual Studio 2013 (або вище), яка буде працювати у віконному режимі та дозволяти наступне (точність $\varepsilon \leq 10^{-3}$):

1. Знаходити всі власні числа матриці заданої за варіантом (табл. 1.3) та власних векторів, що їм відповідають алгоритмом заданим за варіантом (табл. 1.1), що реалізує метод Якобі.
2. Знаходити всі власні числа матриці заданої за варіантом (табл. 1.2) та власні вектори, що їм відповідають LU -методом.

0	Одинична діагональ у матриці L
1	Одинична діагональ у матриці U

3. Знаходити всі власні числа матриці заданої за варіантом (табл. 1.2) та власні вектори, що їм відповідають заданим за варіантом алгоритмом (табл. 1.1), що реалізує QR -метод.

Виконати порівняння (за часовими показниками, кількістю елементарних операцій, кількістю ітерацій і т.д.) методів розв'язання проблеми власних значень.

Для матриць заданих за варіантом, знайти всі власні числа та власні вектори, що їм відповідають у MatLab 7.0 (або вище), MathCAD 15.0 (або вище) та Mathematica 7.0 (або вище) / Maple 14.0 (або вище).

Додаткове завдання

Значення p визначити як $p = k \bmod 3$, де k – номер варіанту.

p	0	1	2
Пункти додатковго завдання за варіантом	1, 2, 3, 4	5, 6, 7	8, 9, 10, 11

Розробити програму на мові програмування C# у середовищі розробки Visual Studio 2013 (або вище), яка буде працювати у віконному режимі та дозволяти наступне (точність $\varepsilon \leq 10^{-3}$):

1. Розв'язувати повну проблему власних значень алгоритмом №4, №5 та №6, що реалізує QR -метод та порівнювати їх швидкодію.
2. Розв'язувати повну проблему власних значень алгоритмом заданим за варіантом (табл. 1.1), що реалізує LQ -метод.
3. Протестувати реалізовані методи розв'язання повної проблеми власних значень на матриці Ресера.
4. Згенерувати більше 10 квадратних несиметричних та таких, що не містять нульових елементів, матриць різної розмірності (розмірність має бути не менше 5) з різними дійсними власними числами. Для згенерованих матриць знайти два найбільших і два найменших за абсолютною величиною власних числа та власних векторів, що їм відповідають методами, які задані в п.1 та п.2 першої лабораторної роботи, порівняти їх за часовими

показниками та кількістю виконуваних елементарних операцій. Побудувати графік залежності кількості виконуваних елементарних операцій від розмірності матриці та графік залежності часу роботи методів від розмірності матриці.

5. Знаходити найбільше та найменше за абсолютною величиною власне число матриці заданої за варіантом (*табл. 1.2*) методом слідів.

Вимоги до оформлення звіту

Звіт має включати:

1. Постановку задачі за варіантом.
2. Математичне підґрунття для виконання другої лабораторної роботи: цей розділ має містити тільки використані математичні формули при виконанні лабораторної роботи.
3. Значення власних чисел та власних векторів, матриць заданих за варіантом, отриманих кожним з методів:

Матриця (вказати номер матриці)	MatLab		MathCAD		Mathematica / Maple	
	Власні числа	Власні вектори	Власні числа	Власні вектори	Власні числа	Власні вектори

4. Значення власних чисел та власних векторів, матриць заданих за варіантом, отриманих кожним з методів:

Матриця № ____

Метод та алгоритм, що його реалізує	Кількість ітерацій	C#	
		Власні числа	Власні вектори

5. Висновки.

Варіанти завдань

Номер варіанту визначається за наступною таблицею:

№ за списком викладача	Варіант №	№ за списком викладача	Варіант №
1	9	13	20
2	1	14	25
3	8	15	2
4	22	16	3
5	4	17	15
6	24	18	7
7	18	19	14
8	5	20	21
9	13	21	6
10	12	22	23
11	17	23	16
12	19	24	11

Таблиця 1.1. Перелік алгоритмів

Номер варіанту	Номер завдання			Номер додаткового завдання
	1	2	3	2
1.	3	1	2	3
2.	2	0	3	4
3.	0	1	0	2
4.	4	1	1	0
5.	3	1	3	1
6.	2	1	2	1
7.	0	0	0	0
8.	5	0	1	2
9.	3	0	1	3
10.	2	1	0	4
11.	4	0	2	4
12.	1	0	3	3
13.	3	1	3	2
14.	5	1	2	1
15.	2	0	1	0
16.	0	1	0	1
17.	3	0	1	0
18.	4	1	0	2
19.	1	0	2	3
20.	3	1	3	4
21.	4	0	3	3
22.	3	1	2	4
23.	2	1	1	2
24.	5	0	3	4

Питання для самоперевірки

1. В чому полягає алгебраїчна проблема власних значень? Навести прикладні задач які зводяться до вирішення алгебраїчної проблеми власних значень.
2. В чому полягає повна проблема власних значень? Навести приклади задач в яких виникає потреба вирішення повної проблеми власних значень.
3. Основні властивості власних чисел та векторів.
4. Основні властивості власних чисел та векторів. Матричне перетворення подібності.
5. Яку структуру має матриця обертань?
6. Сукупність формул, що визначають метод Якобі.
7. Чи є скінченним метод Якобі? Відповідь обґрунтувати.
8. Які існують стратегії вибору ключового елемента в методі Якобі?
9. Чим відрізняються поняття ключовий та приречений елемент?
10. Яким чином визначити кут обертання матриці в методі Якобі (формули та їх виведення)?
11. Чому метод Якобі не є ітераційним за суттю?
12. Довести той факт, що послідовність матриць, яка отримується при перетвореннях подібності в методі Якобі прямує до діагональної матриці з власними числами на діагоналі.
13. Як визначити всі власні вектори в методі Якобі?
14. Метод Якобі для Хесенбергової матриці.
15. LU -метод розв'язання повної несиметричної проблеми власних значень. Переваги та недоліки.
16. Переваги та недоліки LU -методу.

17. QR -метод розв'язання повної несиметричної проблеми власних значень.
18. Довести інваріантність матриці Хесенберга по відношенню до ортогональних перетворень.
19. Перетворення Гівенса.
20. Приведення довільної матриці до форми Хесенберга за допомогою перетворень Гівенса.
21. Різновиди алгоритмів, що реалізують QR -метод.
22. Швидкість збіжності QR -методу.
23. Процедура вичерпування у QR -методі.
24. Зсуви Уїлкінсона у QR -методі.
25. Теорема про неявне Q та про єдиність.
26. За рахунок чого в двокроковому QR -алгоритмі Френсіса не виникає потреби у виконанні операцій над комплексними числами?
27. Яким чином в QR -алгоритмі з неявними зсувами виконується обчислення матриці $(k + 2)$ -ої ітерації за відомою матрицею з k -ої ітерації?
28. Як в QR -алгоритмі з неявними зсувами знайти всі власні вектори?
29. Критерії зупинки ітераційного процесу в методах розв'язання повної проблеми власних значень.
30. Приклад матриці для якої досить складно розв'язати повну симетричну проблему власних значень.

Рекомендована література

1. William H. Press Numerical Recipes in C. The Art of Scientific Computing / William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery // Cambridge University Press, 2002. – 1018 p.

2. Walter Gautschi Numerical Analysis [Электронный ресурс], 2012.
Режим доступа: http://www.ikiu.ac.ir/public-files/profiles/items/090ad_1410599906.pdf
3. McDonough J. M. Computational Numerical Analysis [Электронный ресурс], 2007. Режим доступа: <http://web.engr.uky.edu/~acfd/egr537-lctrs.pdf>

ЛАБОРАТОРНА РОБОТА №3. ПЕРШИЙ ЕТАП СИНГУЛЯРНОГО РОЗКЛАДЕННЯ ПРЯМОКУТНОЇ МАТРИЦІ

Мета роботи: набути вміння та досвід використання методів виконання сингулярного розкладення прямокутних матриць, використання систем комп'ютерної математики. Окрім цього мета даної лабораторної роботи передбачає закріплення, поглиблення й узагальнення теоретичних знань щодо методів виконання сингулярного розкладення прямокутних матриць.

Теоретичні відомості

Для будь-якої дійсної $m \times n$ -матриці \mathbf{A} , матриці $\mathbf{A}^T \mathbf{A}$ та $\mathbf{A} \mathbf{A}^T$ є квадратними $n \times n$ - та $m \times m$ -матрицями відповідно. Спектри матриць $\mathbf{A}^T \mathbf{A}$ та $\mathbf{A} \mathbf{A}^T$ співпадають в їх загальній частині (за мінімальним з розмірів m та n). Такі матриці мають дійсний та невід'ємний спектр.

Сингулярними числами дійсної $m \times n$ -матриці \mathbf{A} називають арифметичні квадратні корені з власних чисел $\lambda_1, \lambda_2, \dots, \lambda_k$ матриць $\mathbf{A}^T \mathbf{A}$ та $\mathbf{A} \mathbf{A}^T$, де $k = \min \{ m, n \}$.

Зазвичай сингулярні числа позначаються буквою σ та нумеруються в порядку спадання:

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k (\geq 0).$$

Кількість ненульових власних чисел матриці \mathbf{A} співпадає з її рангом, а також з рангом матриць $\mathbf{A}^T \mathbf{A}$ та $\mathbf{A} \mathbf{A}^T$. Інколи сингулярними числами називають тільки ненульові числа σ_i , тобто під k мається на увазі загальний ранг вказаних матриць.

У квадратних матриць добуток всіх сингулярних чисел дорівнює модулю визначника. Звідси – природність критерія невідродженості квадратної матриці, що полягає у відмінності від нуля всіх її сингулярних чисел.

Для обчислення сингулярних чисел прямокутної матриці, а також для отримання деякої іншої додаткової інформації про неї використовують сингулярне розкладення.

Подання дійсної $m \times n$ -матриці \mathbf{A} у вигляді

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}, \quad (3.1)$$

де $\mathbf{\Sigma}$ – діагональна $m \times n$ -матриця з діагоналлю з незростаючих сингулярних чисел $\sigma_1, \sigma_2, \dots, \sigma_k$, а \mathbf{U} та \mathbf{V} – ортогональні відповідно $m \times m$ - та $n \times n$ -матриці, називається сингулярним розкладенням матриці \mathbf{A} .

Існує ще одне визначення сингулярного числа σ через сукупність рівностей

$$\mathbf{A}\mathbf{x} = \sigma\mathbf{y}, \quad \mathbf{A}^T\mathbf{y} = \sigma\mathbf{x}, \quad (3.2)$$

де n -мірний вектор \mathbf{x} та m -мірний вектор \mathbf{y} – *правий та лівий сингулярні вектори* відповідно. Ці вектори утворюють ортогональні базиси, тобто існують такі ортогональні $n \times n$ -матриця \mathbf{X} та $m \times m$ -матриця \mathbf{Y} з сингулярних вектор-стовпців, що векторно-матричні рівності (3.2) можна переписати у матричному вигляді:

$$\mathbf{A}\mathbf{X} = \mathbf{Y}\mathbf{\Sigma}, \quad \mathbf{A}^T\mathbf{Y} = \mathbf{X}\mathbf{\Sigma}. \quad (3.3)$$

Оскільки матриця \mathbf{X} є ортогональною, то рівність (3.3) можна переписати у вигляді

$$\mathbf{A} = \mathbf{Y}\mathbf{\Sigma}\mathbf{X}^T. \quad (3.4)$$

Рівність (3.4) співпадає з (3.1), якщо ототожнити \mathbf{U} з \mathbf{Y} , а \mathbf{V} з \mathbf{X}^T .

Доведено, що будь-яка дійсна матриця має дійсне сингулярне розкладення.

В тому випадку коли $m > n$ матриця Σ буде мати вигляд:

$$\begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sigma_n \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (3.5)$$

В тому випадку коли $m < n$ матриця Σ буде мати вигляд:

$$\begin{pmatrix} \sigma_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sigma_m & 0 & \dots & 0 \end{pmatrix}. \quad (3.6)$$

Оскільки матриці \mathbf{U} та \mathbf{V} ортогональні, то цілком природно отримати їх в результаті виконання послідовностей простих ортогональних перетворень.

Повертаючись до процесу побудови сингулярного розкладення зазначимо, що зазвичай такий процес розбивають на два етапи:

1. Приведення початкової $m \times n$ -матриці \mathbf{A} до дводіагональної форми (етап дводіагоналізації), що реалізується за допомогою перетворень Хаусхолдера.
2. Дводіагональна матриця приводиться до діагонального вигляду (етап діагоналізації), що досягається застосуванням до неї QR -алгоритма на основі перетворень Гівенса.

Етап дводіагоналізації

Розглянемо, як буде виглядати результуюча дводіагональна матриця для трьох випадків співвідношення m та n :

1. Якщо $m=n$, то результуюча дводіагональна матриця приймає вигляд:

$$\begin{pmatrix} \beta_1^{(1)} & \beta_2^{(1)} & 0 & 0 & \dots & 0 & 0 \\ 0 & \beta_1^{(2)} & \beta_2^{(2)} & 0 & \dots & 0 & 0 \\ 0 & 0 & \beta_1^{(3)} & \beta_2^{(3)} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & \beta_1^{(t-1)} & a_{t-1,t}^{(t-1)} \\ 0 & 0 & 0 & 0 & \dots & 0 & a_{tt}^{(t-1)} \end{pmatrix}, \quad (3.7)$$

Для цього необхідна $(t-1)$ -а матриця Хаусхолдера типу $\mathbf{H}_1^{(i)}$ та $(t-2)$ -і матриці Хаусхолдера типу $\mathbf{H}_2^{(j)}$, тоді

$$\mathbf{H}_1^{(t-1)} \dots \mathbf{H}_1^{(2)} \mathbf{H}_1^{(1)} \mathbf{A} \mathbf{H}_2^{(1)} \mathbf{H}_2^{(2)} \dots \mathbf{H}_2^{(t-2)} = \mathbf{B}. \quad (3.8)$$

Якщо ввести позначення $\mathbf{H}_1 = \mathbf{H}_1^{(1)} \mathbf{H}_1^{(2)} \dots \mathbf{H}_1^{(t-1)}$ та $\mathbf{H}_2 = \mathbf{H}_2^{(1)} \mathbf{H}_2^{(2)} \dots \mathbf{H}_2^{(t-2)}$, то рівність (3.8) можна записати таким чином:

$$\mathbf{H}_1 \mathbf{B} = \mathbf{A} \mathbf{H}_2 \quad \text{або} \quad \mathbf{A} = \mathbf{H}_1 \mathbf{B} \mathbf{H}_2^T. \quad (3.9)$$

2. Якщо $m > n$ (кількість рядків перевищує кількість стовпців), то, порівняно з першим випадком, необхідно виконати ще одне ліве перетворення $\mathbf{H}_1^{(t)}$, щоб створити нулі під елементом $a_{tt}^{(t-1)}$, в результаті чого матриця (3.7) доповнюється знизу нульовими

рядками в кількості $m - n$, а елемент $a_{tt}^{(t-1)}$ замінюється на $\beta_1^{(n)}$, тобто

$$\mathbf{H}_1^{(t)} \dots \mathbf{H}_1^{(2)} \mathbf{H}_1^{(1)} \mathbf{A} \mathbf{H}_2^{(1)} \mathbf{H}_2^{(2)} \dots \mathbf{H}_2^{(t-2)} = \begin{pmatrix} \mathbf{B} \\ \mathbf{0} \end{pmatrix}. \quad (3.10)$$

Якщо ввести позначення $\mathbf{H}_1 = \mathbf{H}_1^{(1)} \mathbf{H}_1^{(2)} \dots \mathbf{H}_1^{(t)}$ та $\mathbf{H}_2 = \mathbf{H}_2^{(1)} \mathbf{H}_2^{(2)} \dots \mathbf{H}_2^{(t-2)}$, тоді рівність (3.10) можна записати таким чином:

$$\mathbf{H}_1 \begin{pmatrix} \mathbf{B} \\ \mathbf{0} \end{pmatrix} = \mathbf{A} \mathbf{H}_2 \quad \text{або} \quad \mathbf{A} = \mathbf{H}_1 \begin{pmatrix} \mathbf{B} \\ \mathbf{0} \end{pmatrix} \mathbf{H}_2^T. \quad (3.11)$$

3. Якщо $m < n$ (кількість рядків менше ніж кількість стовпців), то порівняно з першим випадком, необхідно виконати ще одне праве перетворення $\mathbf{H}_2^{(t-1)}$, щоб створити нулі праворуч від елемента $a_{t-1t}^{(t-1)}$, та праве перетворення $\mathbf{H}_2^{(t)}$, яке створює нулі праворуч від елемента $a_{tt}^{(t-1)}$ (якщо $m > n + 1$) в результаті чого матриця (3.7) приймає вигляд:

$$\begin{pmatrix} \beta_1^{(1)} & \beta_2^{(1)} & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & \beta_1^{(2)} & \beta_2^{(2)} & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \beta_1^{(3)} & \beta_2^{(3)} & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & \beta_1^{(t-1)} & \beta_2^{(t-1)} & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & a_{tt}^{(t-1)} & \beta_2^{(t)} & 0 & \dots & 0 \end{pmatrix}. \quad (3.12)$$

Далі необхідно звести матрицю (3.12) до вигляду $(\mathbf{B} | \mathbf{0})$, тобто анулювати елемент $\beta_2^{(t)}$. Це можна зробити діючи на

матрицю (3.12) праворуч перетвореннями Гівенса витісняючи ненульовий елемент вгору по $(t+1)$ -му стовпцю.

Після цього будемо мати матрицю:

$$\mathbf{H}_1^{(t-1)} \dots \mathbf{H}_1^{(2)} \mathbf{H}_1^{(1)} \mathbf{A} \mathbf{H}_2^{(1)} \mathbf{H}_2^{(2)} \dots \mathbf{H}_2^{(t)} = (\mathbf{V} | \mathbf{0}). \quad (3.13)$$

Ввівши позначення $\mathbf{H}_1 = \mathbf{H}_1^{(1)} \mathbf{H}_1^{(2)} \dots \mathbf{H}_1^{(t-1)}$ та $\mathbf{H}_2 = \mathbf{H}_2^{(1)} \mathbf{H}_2^{(2)} \dots \mathbf{H}_2^{(t)}$, рівність (3.13) можна записати таким чином:

$$\mathbf{H}_1 (\mathbf{V} | \mathbf{0}) = \mathbf{A} \mathbf{H}_2 \quad \text{або} \quad \mathbf{A} = \mathbf{H}_1 (\mathbf{V} | \mathbf{0}) \mathbf{H}_2^T. \quad (3.14)$$

До цього моменту ми розглядали випадок приведення початкової матриці \mathbf{A} до верхньої дводіагональної форми, зрозуміло, що також має право на існування підхід заснований на приведенні до нижньої дводіагональної форми.

Введемо позначення: \mathbf{V} – будь-яка квадратна нижня дводіагональна матриця; $t = \min \{ m; n \}$.

В цьому випадку матриця Хаусходера $\mathbf{H}_1^{(1)}$ будується за тим самим принципом, що й при QB -розкладенні (приведення до верхньої форми Хесенберга), а матриця $\mathbf{H}_2^{(1)}$ будується так само, як при LQ -розкладенні (за основу беруться формули LQ -факторизації), де в даному випадку \mathbf{V} – верхня матриця Хесенберга:

1. Якщо $m = n$, то результуюча дводіагональна матриця приймає вигляд:

$$\begin{pmatrix} \beta_2^{(1)} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \beta_1^{(1)} & \beta_2^{(2)} & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \beta_1^{(2)} & \beta_2^{(3)} & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \beta_1^{(3)} & \beta_2^{(4)} & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & \beta_1^{(t-2)} & \beta_2^{(t-1)} & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & a_{t-1}^{(t-1)} & a_{tt}^{(t-1)} \end{pmatrix}. \quad (3.15)$$

Для цього необхідна $(t-2)$ -і матриці Хаусхолдера типу $\mathbf{H}_1^{(i)}$

та $(t-1)$ -а матриця Хаусхолдера типу $\mathbf{H}_2^{(j)}$, тоді

$$\mathbf{V} = \mathbf{H}_1^{(t-2)} \dots \mathbf{H}_1^{(2)} \mathbf{H}_1^{(1)} \mathbf{A} \mathbf{H}_2^{(1)} \mathbf{H}_2^{(2)} \dots \mathbf{H}_2^{(t-1)}. \quad (3.16)$$

Якщо ввести позначення $\mathbf{H}_1 = \mathbf{H}_1^{(1)} \mathbf{H}_1^{(2)} \dots \mathbf{H}_1^{(t-2)}$

та $\mathbf{H}_2 = \mathbf{H}_2^{(1)} \mathbf{H}_2^{(2)} \dots \mathbf{H}_2^{(t-1)}$, то рівність (3.16) можна записати таким

чином:

$$\mathbf{H}_1 \mathbf{V} = \mathbf{A} \mathbf{H}_2 \quad \text{або} \quad \mathbf{A} = \mathbf{H}_1 \mathbf{V} \mathbf{H}_2^T. \quad (3.17)$$

2. Якщо $m > n$ (кількість рядків перевищує кількість стовпців), то порівняно з першим випадком, необхідно виконати ще одне ліве перетворення $\mathbf{H}_1^{(t-1)}$, щоб створити нулі під елементом $a_{t-1,t-1}^{(t-1)}$, та ліве перетворення $\mathbf{H}_1^{(t)}$, яке створює нулі під елементом $a_{t+1,t}^{(t-1)}$ (якщо $m > n + 1$) в результаті чого матриця (3.15) приймає вигляд:

$$\begin{pmatrix} \beta_2^{(1)} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \beta_1^{(1)} & \beta_2^{(2)} & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \beta_1^{(2)} & \beta_2^{(3)} & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \beta_1^{(3)} & \beta_2^{(4)} & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & \beta_1^{(t-2)} & \beta_2^{(t-1)} & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & \beta_1^{(t-1)} & a_{tt}^{(t-1)} \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & \beta_1^{(t)} \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}$$

Далі необхідно звести отриману матрицю до вигляду $\begin{pmatrix} \mathbf{B} \\ \mathbf{0} \end{pmatrix}$, тобто

анулювати елемент $\beta_1^{(t)}$. Це можна зробити діючи на дану матрицю послідовністю перетворень Гівенса.

Після цього будемо мати матрицю:

$$\mathbf{H}_1^{(t)} \dots \mathbf{H}_1^{(2)} \mathbf{H}_1^{(1)} \mathbf{A} \mathbf{H}_2^{(1)} \mathbf{H}_2^{(2)} \dots \mathbf{H}_2^{(t-1)} = \begin{pmatrix} \mathbf{B} \\ \mathbf{0} \end{pmatrix}, \quad (3.18)$$

Ввівши позначення $\mathbf{H}_1 = \mathbf{H}_1^{(1)} \mathbf{H}_1^{(2)} \dots \mathbf{H}_1^{(t)}$ та $\mathbf{H}_2 = \mathbf{H}_2^{(1)} \mathbf{H}_2^{(2)} \dots \mathbf{H}_2^{(t-1)}$, рівність (3.18) можна записати таким чином:

$$\mathbf{H}_1 \begin{pmatrix} \mathbf{B} \\ \mathbf{0} \end{pmatrix} = \mathbf{A} \mathbf{H}_2 \quad \text{або} \quad \mathbf{A} = \mathbf{H}_1 \begin{pmatrix} \mathbf{B} \\ \mathbf{0} \end{pmatrix} \mathbf{H}_2^T. \quad (3.19)$$

3. Якщо $m < n$ (кількість менше ніж кількість стовпців), то порівняно з першим випадком, необхідно виконати ще одне праве перетворення $\mathbf{H}_2^{(t)}$, щоб створити нулі праворуч від елементом $a_{tt}^{(t-1)}$, в результаті чого матриця (3.15) доповнюється праворуч

нульовими стовпцями в кількості $n - m$, а елемент $a_{tt}^{(t-1)}$ замінюється на $\beta_2^{(t)}$, тобто

$$\mathbf{H}_1^{(t-2)} \dots \mathbf{H}_1^{(2)} \mathbf{H}_1^{(1)} \mathbf{A} \mathbf{H}_2^{(1)} \mathbf{H}_2^{(2)} \dots \mathbf{H}_2^{(t)} = (\mathbf{B} | \mathbf{0}). \quad (3.20)$$

Ввівши позначення $\mathbf{H}_1 = \mathbf{H}_1^{(1)} \mathbf{H}_1^{(2)} \dots \mathbf{H}_1^{(t-2)}$ та $\mathbf{H}_2 = \mathbf{H}_2^{(1)} \mathbf{H}_2^{(2)} \dots \mathbf{H}_2^{(t)}$, рівність (3.20) можна записати таким чином:

$$\mathbf{H}_1 (\mathbf{B} | \mathbf{0}) = \mathbf{A} \mathbf{H}_2 \quad \text{або} \quad \mathbf{A} = \mathbf{H}_1 (\mathbf{B} | \mathbf{0}) \mathbf{H}_2^T. \quad (3.21)$$

Таким чином можемо зробити висновок, що при приведенні матриці до верхньої дводіагональної форми у випадку $m < n$ виникає необхідність після виконання перетворень Хаусхолдера застосовувати перетворення Гівенса, а при приведенні до нижньої дводіагональної форми така сама необхідність виникає при $m > n$, тому інколи перед початком *SVD*-розкладення виконується аналіз кількості рядків та стовпців матриці: якщо $m \geq n$, то таку матрицю приводять до верхньої дводіагональної форми, в іншому випадку до нижньої дводіагональної форми.

Отже, робимо висновок, що після описаного приведення до дводіагональної форми можна відкинути нульові рядки або стовпці та продовжувати сингулярне розкладення з дводіагональною квадратною матрицею \mathbf{B} .

Завдання на лабораторну роботу

Розробити програму на мові програмування C# у середовищі розробки Visual Studio 2013 (або вище), яка буде працювати у віконному режимі та дозволяти наступне (точність $\varepsilon \leq 10^{-3}$):

Виконати перший етап сингулярного розкладення матриць заданих за варіантом (*табл. 3.1*). Спосіб приведення початкової матриці до дводіагонального вигляду визначити за останнім стовпчиком *табл. 3.1*, де

0	Приведення до верхнього дводіагонального вигляду перетвореннями Хаусхолдера
1	Приведення до верхнього дводіагонального вигляду перетвореннями Гівенса
2	Приведення до нижнього дводіагонального вигляду перетвореннями Хаусхолдера
3	Приведення до нижнього дводіагонального вигляду перетвореннями Гівенса

Вимоги до оформлення звіту

Звіт має включати:

1. Постановку задачі за варіантом.
2. Математичне підґрунття для виконання лабораторної роботи: цей розділ має містити тільки використані математичні формули при виконанні цієї лабораторної роботи.
3. Результати виконання першого етапу сингулярного розкладення.
4. Висновки.

Варіанти завдань

Номер варіанту визначається за наступною таблицею:

№ за списком викладача	Варіант №	№ за списком викладача	Варіант №
1	9	13	20
2	1	14	25
3	8	15	2
4	22	16	3
5	4	17	15
6	24	18	7
7	18	19	14
8	5	20	21
9	13	21	6
10	12	22	23
11	17	23	16
12	19	24	11

Таблиця 3.1. Перелік матриць

№	Матриця A_1	Матриця A_2	Приведення до дводіагонального вигляду
1	-42 127 -72 134 -124 99 -2 -86 -3 90 -86 101 7 -19 47 88 -111 -31 82 -141 47 78 -6 73 17 -143 128 149 -17 66 86 100 31 -135 -117	23 -131 -105 -4 101 37 -113 -136 128 -105 -11 47 69 -150 -23 10 -45 -111 146 -1 -104 -10 -40 -49 116 144 105 10 -144 -41 85 53 -75 -93 3	2
2	-16 103 140 -61 63 -77 -127 80 18 142 91 -79 22 -130 150 98 96 125 -130 147 106 -28 -1 -100 -105 -10 -10 -101 135 138 142 136 -52 94 9	145 -97 88 54 124 -117 -30 26 40 -123 20 55 14 120 103 139 115 -6 134 57 -58 -9 10 -149 -54 -121 -106 -132 14 -6 3 31 3 84 -84	0
3	-85 -40 -72 -96 54 -147 -148 4 -113 -134 43 31 -16 24 -140 5 -6 -48 -52 -64 -77 -58 102 -70 -127 -92 74 145 15 121 -123 102 38 -96 104	124 -87 -86 50 138 129 -104 136 -111 -127 88 64 9 -28 -46 6 130 -50 -100 38 -56 -63 122 31 58 -17 54 58 116 -29 -37 -89 40 127 118	0
4	96 -11 -7 77 -73 29 -40 -130 103 -139 11 54 -27 81 143 -51 20 141 144 68 -27 46 84 -117 -106 88 -3 69 -31 -106 -47 -31 80 -2 62	101 30 -22 -44 31 24 128 -74 132 -105 82 42 134 -43 10 -118 -76 115 -112 -22 -72 -20 120 -16 70 -1 -141 86 -103 15 10 -28 -57 129 3	0
5	-121 45 40 74 -35 104 79 100 -141 116 123 23 -69 -21 -74 -147 40 -30 -139 123 7 -67 16 143 119 45 102 -17 7 -31 -35 -22 -123 123 38	54 85 -110 10 -135 -46 109 144 -93 -15 128 93 -101 132 -113 -21 22 120 -50 -142 54 76 -146 88 13 -82 137 126 98 -53 -24 121 97 54 -73	2

Продовження табл. 3.1

№	Матриця A_1	Матриця A_2	Приведення до дводіагонального вигляду
6	-9 -60 -61 5 -148 -124 -27 -11 7 14 99 -79 128 -120 3 56 -92 -86 150 -76 -70 62 -150 -42 -137 141 -96 122 38 103 -95 7 54 -32 -136	50 -18 -31 -75 130 29 -127 132 -111 76 -16 -123 -78 -44 11 -18 7 42 137 -129 -111 -31 14 -3 63 -101 -60 -103 51 -32 -124 148 142 94 -132	1
7	-127 98 -93 147 -128 88 -60 44 -113 17 -41 -35 -149 71 -68 26 45 -65 -103 -111 -95 96 42 -20 60 -127 80 28 100 -4 -104 -38 -53 -42 -96	-19 -36 95 129 84 149 -14 51 75 -37 120 61 -3 -111 14 -103 142 106 -2 1 78 33 -133 32 -73 141 81 20 109 -48 -49 -65 -7 -34 45	3
8	68 -70 38 88 42 -48 -97 -142 1 -123 -68 -21 -110 -67 -128 -99 -7 59 -114 -96 49 86 5 116 -141 11 -111 13 142 68 99 -135 93 133 -107	86 127 96 86 21 50 -47 1 -84 72 -23 -46 55 -52 -72 -125 78 134 17 88 129 70 -128 139 -150 -60 -46 77 -101 81 -10 145 -103 -75 -64	1
9	-130 16 -89 143 140 -85 -81 49 -114 -130 111 -148 125 6 -19 -26 80 -148 97 -54 49 -144 74 41 -110 85 -32 90 137 -110 -76 -74 123 135 92	23 33 -35 92 41 12 105 130 66 -42 87 -23 132 134 -118 -59 -64 -65 122 48 -37 70 -12 95 -130 -25 -32 -130 142 -136 -15 -134 -104 -73 -96	2
10	-55 -28 38 112 -131 -12 -52 103 -87 3 -79 74 3 101 36 -142 74 -101 108 70 48 -98 65 7 -81 -103 -115 123 -7 -144 91 -98 -85 117 -109	-14 -28 -44 -113 126 -12 -133 94 -18 -48 -14 48 92 -107 129 54 119 -128 57 98 -99 52 -11 14 49 106 -93 38 -38 136 75 61 -10 -143 -142	3

Продовження табл. 3.1

№	Матриця A_1	Матриця A_2	Приведення до дводіагонального вигляду
11	-99 69 -112 112 -146 -53 82 -110 -40 26 91 121 -112 -79 139 -60 -109 131 -94 105 83 89 -68 14 -148 16 -93 133 -74 40 16 -142 42 -58 -42	-133 -61 -82 75 98 -72 84 -36 30 -71 -26 -59 27 35 67 -105 3 -134 97 -6 -107 -122 -19 -86 -33 20 -91 65 50 -147 -46 -8 -134 -79 -30	3
12	27 122 39 94 113 -16 40 -132 -55 112 128 122 52 -57 -98 -122 39 -7 -47 105 -37 -146 -59 50 138 14 -55 5 109 81 -117 -117 62 79 113	-100 -71 -93 80 16 -132 -134 6 125 83 -55 -29 -86 -17 38 35 110 -75 75 13 11 64 -122 -50 -90 -4 -27 -110 -58 38 -110 -130 -35 121 12	0
13	-145 39 -12 127 86 -63 107 -41 3 31 143 120 -66 125 -10 80 -46 -128 -123 -61 -77 -4 -17 148 -110 55 54 -101 -121 -62 -109 61 -30 -56 -100	35 109 -50 -31 -122 -8 101 51 -67 -26 88 -54 -108 -53 -139 10 -26 28 116 136 -40 -149 7 146 -57 47 115 89 -108 20 -133 121 56 -19 -121	3
14	50 -138 -61 -103 -71 -26 25 -128 3 -56 -1 19 -62 31 -95 135 -44 -80 -102 -16 136 114 -46 41 -52 70 37 105 104 -66 -35 37 -102 85 130	-90 -95 17 28 -35 -40 134 103 -126 42 33 -30 -45 -45 -23 -11 -140 82 -53 39 -92 13 -141 63 -134 17 49 126 8 -20 -100 107 -62 148 -64	2
15	23 -51 118 -80 -119 120 -21 46 -90 69 -85 66 -138 -36 45 141 125 1 29 -8 -20 117 119 -70 130 86 -110 -34 37 -121 8 -114 -63 92 30	36 136 -77 -89 94 -137 -80 -98 75 -64 -20 122 -107 -68 -63 77 139 -108 12 -80 136 -145 13 -81 -37 96 -76 -46 61 -66 11 88 63 -99 -61	0

Продовження табл. 3.1

№	Матриця A_1	Матриця A_2	Приведення до дводіагонального вигляду
16	35 122 12 22 -142 -43 -66 111 -148 84 -41 34 17 66 20 -130 49 -144 -15 -128 111 -90 -5 48 -75 -13 138 93 76 -110 -127 50 71 92 19	8 -114 -72 -103 -108 -12 123 -73 28 -2 -39 -127 5 31 -27 -42 107 109 113 -69 -41 135 66 68 56 -24 -81 30 126 7 -91 40 -4 120 51	3
17	18 -50 117 40 -46 -83 36 108 -62 66 -119 148 29 37 -142 -147 45 47 -136 -130 -133 12 125 149 129 -53 -81 -20 -88 -124 84 72 -63 32 -50	-25 -53 63 -11 120 110 -121 -66 -121 43 -89 -14 35 21 30 -99 -98 111 -133 -142 -52 -140 -39 -132 29 -119 -53 -15 -131 -139 -28 -144 150 -11 23	2
18	-78 93 45 -23 -60 124 -15 35 -68 -30 98 -75 -9 -17 5 -17 137 23 38 -132 145 -108 124 10 -81 24 4 -37 -34 -115 -80 142 -82 112 -121	-123 -71 138 -56 -35 -122 -69 144 98 3 -133 7 -49 -86 46 -51 19 -137 118 -107 40 -12 133 149 94 -28 -77 90 109 -77 82 -26 31 -36 -88	1
19	-85 -135 -139 -128 105 108 84 68 140 1 109 -22 112 33 -148 -56 -66 -64 -35 126 -52 -50 47 -141 -27 102 -40 -81 108 70 -2 89 37 31 -105	52 -112 147 20 1 118 48 36 -42 -145 76 -135 52 6 -148 -93 -139 6 -134 56 -51 72 65 92 26 -50 59 131 148 -97 108 -93 39 90 -76	2
20	123 -67 -56 7 37 -45 -70 149 -68 68 131 -82 146 66 101 129 -150 140 84 -145 26 118 50 -126 -90 -120 -14 68 -84 -9 -40 24 -50 -89 -37	-58 19 19 26 -120 -101 -62 140 99 -92 141 -106 -33 42 130 138 -84 25 52 90 116 -37 -128 62 -120 42 -30 -87 35 117 29 -100 -38 77 -6	1

Продовження табл. 3.1

№	Матриця A_1	Матриця A_2	Приведення до дводіагонального вигляду
21	$\begin{pmatrix} -27 & 72 & 3 & -64 & 27 \\ 42 & -74 & 82 & 38 & -117 \\ 107 & 66 & 22 & -11 & -90 \\ 79 & 58 & 137 & -112 & -101 \\ 143 & 82 & -99 & 15 & -139 \\ 85 & 74 & 123 & 141 & -68 \\ 131 & -55 & 76 & -17 & -81 \end{pmatrix}$	$\begin{pmatrix} -30 & 26 & 139 & -122 & -17 & 126 & -34 \\ 128 & 84 & -56 & -52 & 80 & 147 & 31 \\ -1 & 47 & 89 & -59 & 55 & 130 & 18 \\ 33 & 9 & -65 & -145 & 65 & -12 & 104 \\ -149 & 98 & -146 & -102 & -12 & 122 & -65 \end{pmatrix}$	1
22	$\begin{pmatrix} 123 & -95 & 71 & -68 & 97 \\ 24 & -91 & -59 & 150 & 29 \\ 83 & 109 & -138 & 101 & 85 \\ 49 & -113 & 101 & 88 & 147 \\ -9 & 44 & -39 & 47 & 87 \\ -84 & -19 & 49 & 13 & 106 \\ 31 & 34 & 119 & -34 & -4 \end{pmatrix}$	$\begin{pmatrix} 125 & 4 & -38 & 130 & 90 & 78 & 68 \\ 97 & 136 & -54 & -40 & 28 & 48 & -47 \\ 15 & 103 & 85 & -47 & -121 & 30 & -141 \\ -146 & 131 & -87 & 58 & 18 & -82 & 103 \\ -122 & 142 & 138 & 65 & -138 & 64 & 81 \end{pmatrix}$	3
23	$\begin{pmatrix} 23 & 45 & 19 & 28 & 103 \\ -64 & 145 & 40 & 57 & -1 \\ -70 & -92 & -1 & 44 & 79 \\ 117 & 99 & -133 & 37 & -96 \\ -40 & -2 & 5 & 137 & 58 \\ -105 & 27 & -134 & -97 & 125 \\ 134 & -137 & 103 & -8 & -102 \end{pmatrix}$	$\begin{pmatrix} 122 & -145 & 75 & 138 & -118 & -68 & -120 \\ -9 & 4 & -50 & 35 & -102 & 99 & 51 \\ 124 & -51 & 40 & -105 & -61 & 88 & -46 \\ -52 & -49 & -25 & 102 & -23 & -68 & 51 \\ 58 & 136 & 23 & 68 & 9 & 35 & -62 \end{pmatrix}$	0
24	$\begin{pmatrix} 96 & 106 & -26 & 128 & -129 \\ 128 & -21 & 128 & 1 & 62 \\ 4 & 128 & -134 & 91 & 130 \\ -92 & -38 & -35 & -45 & -118 \\ 31 & 60 & -78 & -12 & 46 \\ -91 & 27 & -82 & 136 & -107 \\ -27 & -120 & -43 & -110 & 58 \end{pmatrix}$	$\begin{pmatrix} -83 & -132 & 1 & 45 & -113 & 54 & 74 \\ 24 & 133 & 9 & 127 & -103 & -148 & 134 \\ 31 & -123 & -36 & -51 & -26 & -112 & 126 \\ -74 & -50 & 81 & 94 & 56 & 79 & -83 \\ 46 & 45 & -65 & 87 & 47 & 103 & 3 \end{pmatrix}$	1

Таблиця 3.2. Перелік векторів

№	Вектор b_1^T	Вектор b_2^T
1	-85 94 -109 115 127 -147 -37	-35 -57 -149 95 42
2	-128 -57 119 101 -150 42 91	-126 136 -146 -116 -147
3	-31 -100 -21 -25 69 -28 136	-3 92 -52 15 -33
4	-36 -128 -27 -107 90 130 -149	19 56 -123 112 133
5	20 119 -86 -149 115 -80 -77	116 126 -60 -128 80
6	-55 85 142 26 84 69 45	61 -124 35 -98 46
7	-120 -90 -110 -53 136 10 -76	90 -84 108 122 -63
8	41 87 20 -37 97 -59 -54	32 80 104 121 29
9	7 134 147 -27 -39 -82 -16	23 -95 -63 -12 -46
10	81 141 -34 149 -52 -109 -35	-8 54 -116 -79 -63
11	-116 12 -25 5 116 -106 -20	-11 62 -30 -146 -128
12	-130 44 -53 42 114 -38 80	108 -91 -104 -132 48
13	-55 -117 100 142 -85 62 -139	136 -150 -62 -136 -17
14	-30 -36 28 -130 -89 67 23	15 126 -123 -73 -22
15	-80 -141 24 103 17 102 -89	-29 -60 77 -42 -113
16	12 -130 147 -75 -56 -60 -138	119 -124 12 -22 35
17	8 -76 13 85 7 130 -106	-128 -133 -60 7 19
18	117 -140 102 2 -116 -3 30	119 74 11 141 20
19	95 123 84 77 -29 61 20	3 72 -74 104 11
20	-48 -131 79 -29 52 15 -135	-116 -111 -131 -127 38
21	-43 -110 150 4 -34 -75 -41	49 31 47 -57 -51
22	113 -50 -89 -2 -141 99 -71	3 62 88 -51 125
23	-22 -138 -83 50 112 -47 -142	54 -32 -33 -88 68
24	38 -137 90 146 -19 -47 -136	87 7 35 18 70

Питання для самоперевірки

1. Визначення сингулярних чисел.
2. Чи можуть бути сингулярні числа від'ємними? Чому?
3. Які вектори називаються лівими сингулярними?
4. Які вектори називаються правими сингулярними?
5. Яке розкладення матриці називається сингулярним?
6. Для яких матриць застосовується сингулярне розкладення?
7. Яка матриця називається нормальною?
8. З яких етапів складається процес сингулярного розкладення матриці?
9. Які існують способи реалізації першого етапу сингулярного розкладення матриці?
10. На яких формулах ґрунтується реалізація першого етапу сингулярного розкладення матриці?
11. На якому алгоритмі ґрунтується сингулярне розкладення дводіагональної матриці? Чому саме на цьому алгоритмі? Що дозволяє врахувати цей алгоритм?

Рекомендована література

1. Singular Value Decomposition [Електронний ресурс] Режим доступу: <https://www.cs.cmu.edu/~venkatg/teaching/CStheory-infoage/book-chapter-4.pdf>
2. Courtney Remani Numerical Methods for Solving Systems of Nonlinear Equations [Електронний ресурс], 2013. Режим доступу: <https://www.lakeheadu.ca/sites/default/files/uploads/77/docs/RemaniFinal.pdf>
3. Андруник В.А. Чисельні методи в комп'ютерних науках: навчальний посібник / Андруник В.А., Висоцька В.А., Пасічник В.В., Чирун Л.Б., Чирун Л.В. Ч // Том 2 за ред. В.В. Пасічника – Львів: Видавництво «Новий Світ», 2020. – 536 с

ЛАБОРАТОРНА РОБОТА №4. ДРУГИЙ ЕТАП СИНГУЛЯРНОГО РОЗКЛАДЕННЯ ПРЯМОКУТНОЇ МАТРИЦІ ТА ЙОГО ЗАСТОСУВАННЯ

Мета роботи: набути вміння та досвід використання, побудови методів виконання сингулярного розкладення прямокутних матриць та найпростіших його застосувань, використання систем комп'ютерної математики. Окрім цього мета даної лабораторної роботи передбачає закріплення, поглиблення й узагальнення теоретичних знань щодо методів виконання сингулярного розкладення прямокутних матриць.

Теоретичні відомості

Для будь-якої дійсної $m \times n$ -матриці \mathbf{A} , матриці $\mathbf{A}^T \mathbf{A}$ та $\mathbf{A} \mathbf{A}^T$ є квадратними $n \times n$ - та $m \times m$ -матрицями відповідно. Спектри матриць $\mathbf{A}^T \mathbf{A}$ та $\mathbf{A} \mathbf{A}^T$ співпадають в їх загальній частині (за мінімальним з розмірів m та n). Такі матриці мають дійсний та невід'ємний спектр.

Сингулярними числами дійсної $m \times n$ -матриці \mathbf{A} називають арифметичні квадратні корені з власних чисел $\lambda_1, \lambda_2, \dots, \lambda_k$ матриць $\mathbf{A}^T \mathbf{A}$ та $\mathbf{A} \mathbf{A}^T$, де $k = \min \{ m, n \}$.

Зазвичай сингулярні числа позначаються буквою σ та нумеруються в порядку спадання:

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k (\geq 0).$$

Кількість ненульових власних чисел матриці \mathbf{A} співпадає з її рангом, а також з рангом матриць $\mathbf{A}^T \mathbf{A}$ та $\mathbf{A} \mathbf{A}^T$. Інколи сингулярними числами називають тільки ненульові числа σ_i , тобто під k мається на увазі загальний ранг вказаних матриць.

У квадратних матриць добуток всіх сингулярних чисел дорівнює модулю визначника. Звідси – природність критерія невинродженості квадратної матриці, що полягає у відмінності від нуля всіх її сингулярних чисел.

Для обчислення сингулярних чисел прямокутної матриці, а також для отримання деякої іншої додаткової інформації про неї використовують сингулярне розкладення.

Подання дійсної $m \times n$ -матриці \mathbf{A} у вигляді

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}, \quad (4.1)$$

де $\mathbf{\Sigma}$ – діагональна $m \times n$ -матриця з діагоналлю з незростаючих сингулярних чисел $\sigma_1, \sigma_2, \dots, \sigma_k$, а \mathbf{U} та \mathbf{V} – ортогональні відповідно $m \times m$ - та $n \times n$ -матриці, називається сингулярним розкладенням матриці \mathbf{A} .

Припустимо, що у дійсної $m \times n$ -матриці \mathbf{A} відоме сингулярне розкладення:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}. \quad (4.2)$$

Виходячи з матриці $\mathbf{\Sigma}$ можна підрахувати кількість ненульових сингулярних чисел, а це в свою чергу є рангом r матриці \mathbf{A} .

Якщо $m = n$, то знаючи сингулярні числа квадратної матриці можна обчислити абсолютну величину її визначника:

$$|\det \mathbf{A}| = \prod_{i=1}^n \sigma_i. \quad (4.3)$$

Зрозуміло, що при $r < n$ (кількість ненульових сингулярних чисел менша за n) маємо $\det \mathbf{A} = 0$.

Знання набору сингулярних чисел прямокутної матриці \mathbf{A} також дозволяє обчислити її число *спектральне число обумовленості* $\text{cond} \mathbf{A}$ наступним чином:

$$\text{cond } \mathbf{A} = \frac{\sigma_1}{\sigma_r}, \quad (4.4)$$

де σ_1 – найбільше, а σ_r – найменше ненульове сингулярне число.

Знаючи сингулярне розкладення матриці можна знайти розв'язок однорідної СЛАР.

Нехай дана однорідна СЛАР:

$$\mathbf{Ax} = \mathbf{0}, \quad (4.5)$$

з $m \times n$ -матрицею \mathbf{A} рангу r .

Якщо відоме сингулярне розкладення матриці \mathbf{A} , то можемо записати:

$$\mathbf{Ax} = \mathbf{0} \Leftrightarrow \mathbf{U}\Sigma\mathbf{Vx} = \mathbf{0} \Leftrightarrow \Sigma\mathbf{Vx} = \mathbf{0}. \quad (4.6)$$

Введемо позначення $\mathbf{y} = \mathbf{Vx}$, тоді

Помилка! Джерело посилання не знайдено. можна записати таким чином:

$$\Sigma\mathbf{y} = \mathbf{0}. \quad (4.7)$$

Очевидно, що розв'язком системи

Помилка! Джерело посилання не знайдено. буде вектор

$\mathbf{y} = (0; \dots; 0; c_1; \dots; c_{n-r})^T$, де r – ранг матриці \mathbf{A} , а $c_1; \dots; c_{n-r}$ – параметри, що можуть приймати будь-які дійсні значення.

Тоді шуканий розв'язок \mathbf{x} даної системи

Помилка! Джерело посилання не знайдено. буде мати вигляд:

$$\mathbf{x} = \mathbf{V}^T \mathbf{y} = 0 \cdot \begin{pmatrix} v_{11} \\ v_{12} \\ \dots \\ v_{1n} \end{pmatrix} + \dots + 0 \cdot \begin{pmatrix} v_{r1} \\ v_{r2} \\ \dots \\ v_{rn} \end{pmatrix} + c_1 \cdot \begin{pmatrix} v_{r+1;1} \\ v_{r+1;2} \\ \dots \\ v_{r+1;n} \end{pmatrix} + \dots + c_{n-r} \cdot \begin{pmatrix} v_{n1} \\ v_{n2} \\ \dots \\ v_{nn} \end{pmatrix}. \quad (4.8)$$

Таким чином отримано загальний розв'язок однорідної системи **Помилка! Джерело посилання не знайдено.** у вигляді лінійної

комбінації рядків правої ортогональної матриці \mathbf{V} в сингулярному розкладенні.

За допомогою отриманого сингулярного розкладення можна розв'язувати довільну СЛАР:

$$\mathbf{Ax} = \mathbf{b}. \quad (4.9)$$

Тоді

$$\mathbf{Ax} = \mathbf{b} \Leftrightarrow \mathbf{U}\Sigma\mathbf{Vx} = \mathbf{b} \Leftrightarrow \Sigma\mathbf{y} = \mathbf{z}, \quad (4.10)$$

де $\mathbf{y} = \mathbf{Vx}$, $\mathbf{z} = \mathbf{U}^T\mathbf{b} = (z_1; z_2; \dots; z_m)^T$.

Якщо довільна СЛАР є сумісною, то можна стверджувати, що:

- якщо $m > n$ та $r = n$, то розв'язок СЛАР буде єдиним; якщо $m > n$ та $r < n$ (матриця неповного рангу), то розв'язок СЛАР буде не єдиним; хоча варто відмітити, що зазвичай така СЛАР є несумісною;
- якщо $m < n$, то незалежно від рангу розв'язок СЛАР буде не єдиним; зокрема якщо $r = m$, то розв'язок СЛАР буде при будь-якій правій частині.

При розв'язанні перевизначених СЛАР вводять поняття псевдорозв'язку.

Вектор \mathbf{x} на якому досягається мінімум функції $\Phi(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|_2^2 = (\mathbf{Ax} - \mathbf{b}; \mathbf{Ax} - \mathbf{b})$ називається *псевдорозв'язком* системи $\mathbf{Ax} = \mathbf{b}$.

Для знаходження псевдорозв'язку необхідно розв'язати нормальну систему:

$$\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}, \quad (4.11)$$

матриця коефіцієнтів у якій є завжди квадратною.

Нормальним псевдорозв'язком називається вектор \mathbf{x} , який мінімізує функцію $\Phi(\mathbf{x})$ та має найменшу норму серед векторів, що мінімізують $\Phi(\mathbf{x})$.

Повертаючись до процесу побудови сингулярного розкладення зазначимо, що зазвичай такий процес розбивають на два етапи:

3. Приведення початкової $m \times n$ -матриці \mathbf{A} до дводіагональної форми (етап дводіагоналізації), що реалізується за допомогою перетворень Хаусхолдера.
4. Дводіагональна матриця приводиться до діагонального вигляду (етап діагоналізації), що досягається застосуванням до неї QR -алгоритма на основі перетворень Гівенса.

Розкладення дводіагональної матриці

Нехай після першого етапу була отримана квадратна дводіагональна $t \times t$ -матриця \mathbf{B} , тоді $\mathbf{M} = \mathbf{B}^T \mathbf{B}$ є Хесенберговою тридіагональною матрицею та має бути ортогонально приведена до діагонального виду.

Нехай \mathbf{S} і \mathbf{T} – ортогональні матриці, вимоги до яких сформуємо далі. Побудуємо матрицю \mathbf{B}_1 наступним чином:

$$\mathbf{B}_1 = \mathbf{S} \mathbf{B} \mathbf{T}^T. \quad (4.12)$$

Дводіагональна матриця \mathbf{B} має вигляд:

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & 0 & \cdots & 0 & 0 \\ 0 & b_{22} & b_{23} & \cdots & 0 & 0 \\ 0 & 0 & b_{33} & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & b_{n-1\ n-1} & b_{n-1\ n} \\ 0 & 0 & 0 & \cdots & 0 & b_{n\ n} \end{pmatrix}. \quad (4.13)$$

Матрицю \mathbf{T}_2 визначимо, як матрицю Гівенса \mathbf{G}_1 з кутом повороту φ_2 , а матрицю $\mathbf{B}\mathbf{T}_2^T$ позначимо, як \mathbf{P} , тоді матриця \mathbf{P} буде відрізнятися від матриці \mathbf{B} лише елементами підматриці:

$$\begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix} = \begin{pmatrix} b_{11} \cos \varphi_2 + b_{12} \sin \varphi_2 & b_{12} \cos \varphi_2 - b_{11} \sin \varphi_2 \\ b_{22} \sin \varphi_2 & b_{22} \cos \varphi_2 \end{pmatrix} \quad (4.14)$$

Далі матрицю \mathbf{S}_2 визначаємо, як матрицю Гівенса \mathbf{G}_1 з кутом повороту θ_2 , а матрицю $\mathbf{S}_2\mathbf{P}$ позначимо, як \mathbf{L} , тоді матриця \mathbf{L} буде відрізнятися від матриці \mathbf{P} лише елементами підматриці:

$$\begin{pmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \end{pmatrix} = \begin{pmatrix} p_{11} \cos \theta_2 + p_{21} \sin \theta_2 & p_{12} \cos \theta_2 + p_{22} \sin \theta_2 & b_{23} \sin \theta_2 \\ p_{21} \cos \theta_2 - p_{11} \sin \theta_2 & p_{22} \cos \theta_2 - p_{12} \sin \theta_2 & b_{23} \cos \theta_2 \end{pmatrix}. \quad (4.15)$$

Тепер розглянемо виходячи з яких принципів необхідно обирати кут повороту φ_2 та θ_2 .

Почнемо з кута θ_2 . Нагадаємо, що матриця \mathbf{S}_2 має підтримувати дводіагональну структуру матриці, що отримується шляхом таких перетворень, в даному випадку верхню дводіагональну структуру, тому кут θ_2 необхідно обирати таким чином, щоб він анулював піддіагональний елемент l_{21} , але в результаті такого перетворення залишається ненульовим елемент l_{13} , тому кут φ_3 в матриці \mathbf{T}_3 необхідно обирати так, щоб цей ненульовий елемент після множення справа на матрицю \mathbf{T}_3^T став нульовим.

Після такого перетворення ми отримаємо ненульовий піддіагональний елемент на позиції $(3; 2)$. Цей ненульовий елемент анулюється шляхом застосування лівого перетворення Гівенса \mathbf{S}_3 і т.д. Такий процес, коли ненульовий елемент переганяється з нижньої піддіагонали у верхню наддіагональ та назад з послідовним зміщенням його на одну позицію уздовж головної діагонали до правого нижнього кута, називається *процесом переслідування*.

Описаний процес переслідування завершується після $(n-1)$ -го кроку лівого перетворення Гівенса, умовно кажучи, витісняючи ненульовий елемент за межі матриці вправо. Отримана при цьому дводіагональна матриця

$$\mathbf{B}_1 = \mathbf{S}_n \dots \mathbf{S}_3 \mathbf{S}_2 \mathbf{B} \mathbf{T}_2^T \mathbf{T}_3^T \dots \mathbf{T}_n^T \quad (4.16)$$

приймається за результат першої ітерації алгоритма, тобто далі виконуються такі ж самі дії з матрицею \mathbf{B}_1 .

На перший погляд процедура отримання матриці \mathbf{B}_1 досить повно описана, але звертаю Вашу увагу, що ми визначили, яким чином обирати кут $\varphi_3, \varphi_4, \dots, \varphi_n$, але й досі не визначили як обирати кут φ_2 .

При виборі кута φ_2 потрібно згадати, що має забезпечуватись збіжність послідовності тридіагональних матриць $\mathbf{M} = \mathbf{B}^T \mathbf{B}$ до діагонального виду в процесі ітерацій QR -алгоритму, але до цього моменту це ніяким чином не було враховано, тому це необхідно врахувати при виборі кута φ_2 .

Тепер залишається виконати правильну збірку всіх матриць лівих та правих ортогональних перетворень для фіксованих матриць \mathbf{U} та \mathbf{V} в розкладенні:

$$\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}. \quad (4.17)$$

Згадаємо, що на першому етапі (дводіагоналізації) за допомогою перетворень Хаусхолдера та в необхідних випадках перетворень Гівенса були отримані подання:

$$\mathbf{A} = \mathbf{H}_1 \mathbf{B} \mathbf{H}_2^T, \quad \text{якщо } m = n; \quad (4.18)$$

$$\mathbf{A} = \mathbf{H}_1 \begin{pmatrix} \mathbf{B} \\ \mathbf{0} \end{pmatrix} \mathbf{H}_2^T, \quad \text{якщо } m > n; \quad (4.19)$$

$$\mathbf{A} = \mathbf{H}_1 (\mathbf{B} | \mathbf{0}) \mathbf{H}_2^T, \quad \text{якщо } m < n. \quad (4.20)$$

Позначимо

$$\mathbf{S} = \mathbf{S}_n^{(l)} \dots \mathbf{S}_3^{(l)} \mathbf{S}_2^{(l)} \quad \text{та} \quad \mathbf{T} = \mathbf{T}_n^{(l)} \dots \mathbf{T}_3^{(l)} \mathbf{T}_2^{(l)}. \quad (4.21)$$

Матриці \mathbf{S} та \mathbf{T} є результатами послідовних застосувань відповідно лівих та правих перетворень обертань в розглянутому процесі переслідування, тоді можемо записати наступне:

$$\tilde{\Sigma} = \mathbf{S} \mathbf{B} \mathbf{T}^T \Leftrightarrow \mathbf{V} = \mathbf{S}^T \tilde{\Sigma} \mathbf{T}, \quad (4.22)$$

де $\tilde{\Sigma} = \text{diag}(\pm\sigma_1; \pm\sigma_2; \dots; \pm\sigma_n)$.

Об'єднуючи **Помилка! Джерело посилання не знайдено.** та **Помилка! Джерело посилання не знайдено.** при $m = n$, маємо:

$$\mathbf{A} = \mathbf{H}_1 \mathbf{S}^T \tilde{\Sigma} \mathbf{T} \mathbf{H}_2^T, \quad (4.23)$$

тобто отримали розкладення типу

Помилка! Джерело посилання не знайдено., де $\mathbf{U} = \mathbf{H}_1 \mathbf{S}^T$ та $\mathbf{V} = \mathbf{T} \mathbf{H}_2^T$.

Відмітимо, що для кратних сингулярних чисел відповідні стовпці матриць \mathbf{U} та \mathbf{V} визначені неоднозначно.

Завдання на лабораторну роботу

Розробити програму на мові програмування C# у середовищі розробки Visual Studio 2013 (або вище), яка буде працювати у віконному режимі та дозволяти наступне (точність $\varepsilon \leq 10^{-3}$):

1. Знаходити всі сингулярні числа матриць заданих за варіантом (*табл. 3.1*) і правих та лівих сингулярних векторів, що їм відповідають.
2. За отриманим в п.1 сингулярним розкладенням матриць, знаходити:
 - ранг матриць;
 - псевдообернені матриці;
 - число обумовленості (двома способами);
 - нормальний псевдорозв'язок СЛАР (вектор вільних членів взяти з *табл. 2.2*);
 - загальний розв'язок СЛАР.

Для матриць заданих за варіантом, знайти всі сингулярні числа і ліві та праві сингулярні вектори, що їм відповідають, а також виконати завдання з п.2 у будь-якому математичному пакеті.

Вимоги до оформлення звіту

Звіт має включати:

1. Постановку задачі за варіантом.
2. Математичне підґрунття для виконання лабораторної роботи: цей розділ має містити тільки використані математичні формули при виконанні даної лабораторної роботи.
3. Значення сингулярних чисел і лівих та правих сингулярних векторів, матриць заданих за варіантом:

Матриця	Кількість ітерацій (етап 2)	С#			Математичний пакет		
		Сингулярн і числа	Ліві сингулярн і вектори	Праві сингулярн і вектори	Сингулярн і числа	Ліві сингулярн і вектори	Праві сингулярн і вектори

4. Результати з п.1 та п.2 завдання.

Матриця	С#				Математичний пакет			
	Ранг матриці	Псевдообернен а матриця	Число обумовленості (перший спосіб)	Число обумовленості (другий спосіб)	Ранг матриці	Псевдообернен а матриця	Число обумовленості (перший спосіб)	Число обумовленості (другий спосіб)

5. Результати з п.1 та п.2 завдання.

СЛАР	С#		Математичний пакет	
	Нормальний псевдорозв'язок	Загальний розв'язок	Нормальний псевдорозв'язок	Загальний розв'язок

6. Висновки.

Варіанти завдань

Номер варіанту визначається за таблицею з лабораторної роботи №3.

Питання для самоперевірки

1. В чому полягає неявний зсув на другому етапі сингулярного розкладення? Навіщо потрібно його виконувати? Чому цей зсув називають неявним?
2. В чому полягає процедура вичерпування?
3. Якщо k -им діагональним елементом матриці \mathbf{B} , у процесі перетворень, є досить мала величина, то що це означає і які додаткові перетворення необхідно виконати над матрицею \mathbf{B} ? Навіщо виконувати ці додаткові перетворення над матрицею \mathbf{B} ?
4. Чи можливо обчислити визначник квадратної матриці, якщо відомі її сингулярні числа? Якщо так, то яким чином?
5. Яка СЛАР називається однорідною?
6. Застосування сингулярного розкладення для розв'язання однорідної СЛАР.
7. Застосування сингулярного розкладення для розв'язання довільної неоднорідної СЛАР. Описати різні випадки співвідношення між кількістю рядків, стовпців матриці та її рангом.
8. Що таке псевдорозв'язок СЛАР?
9. Що таке нормальний псевдорозв'язок СЛАР?
10. Як отримати нормальний псевдорозв'язок СЛАР, маючи сингулярне розкладення матриці коефіцієнтів?
11. Яка матриця називається псевдо оберненою? Як її побудувати маючи сингулярне розкладення?
12. Дати визначення матриці Мура-Пенроуза.
13. Дати визначення загального розв'язку неоднорідної СЛАР.
14. Як отримати загальний розв'язок неоднорідної СЛАР, якщо відоме сингулярне розкладення матриці коефіцієнтів?

15. Два способи знаходження числа обумовленості матриці: використовуючи знайдені сингулярні числа та знайдену псевдообернену матрицю.
16. Яка СЛАР називається нормальною?
17. За яким критерієм можна визначити чи є СЛАР нормальною?
18. У нормальній СЛАР матриця коефіцієнтів завжди квадратна?
19. Довести, що для знаходження псевдорозв'язку СЛАР достатньо розв'язати нормальну СЛАР, яка відповідає заданій СЛАР.

Рекомендована література

1. Singular Value Decomposition [Електронний ресурс] Режим доступу: <https://www.cs.cmu.edu/~venkatg/teaching/CStheory-infoage/book-chapter-4.pdf>
2. Courtney Remani Numerical Methods for Solving Systems of Nonlinear Equations [Електронний ресурс], 2013. Режим доступу: <https://www.lakeheadu.ca/sites/default/files/uploads/77/docs/RemaniFinal.pdf>
3. Андруник В.А. Чисельні методи в комп'ютерних науках: навчальний посібник / Андруник В.А., Висоцька В.А., Пасічник В.В., Чирун Л.Б., Чирун Л.В. Ч // Том 2 за ред. В.В. Пасічника – Львів: Видавництво «Новий Світ», 2020. – 536 с

ЛАБОРАТОРНА РОБОТА №5. СИСТЕМИ НЕЛІНІЙНИХ РІВНЯНЬ

Мета роботи: опанувати методи наближеного розв'язання систем нелінійних рівнянь.

Теоретичні відомості

В задачах проектування та дослідження поведінки реальних об'єктів, процесів та систем математичні моделі повинні відображати реальні фізичні нелінійні процеси. При цьому ці процеси залежать, зазвичай, від багатьох змінних, тобто вони описуються системами нелінійних рівнянь (СНР) великої розмірності.

Введемо позначення:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}, \quad F(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \dots \\ f_n(\mathbf{x}) \end{pmatrix}, \quad \mathbf{0} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \end{pmatrix}. \quad (5.1)$$

Тоді систему нелінійних рівнянь можна записати таким чином:

$$F(\mathbf{x}) = \mathbf{0} \quad (5.2)$$

відносно векторної функції F та векторного аргументу \mathbf{x} .

Геометричний зміст СНР полягає в знаходженні точки перетину n -мірних гіперповерхонь в n -мірному гіперпросторі.

У системи нелінійних рівнянь визначити кількість векторів-розв'язків завчасно неможливо. Система може взагалі не мати розв'язків, мати один розв'язок або безкінечну кількість розв'язків. У зв'язку з цим, методи чисельного розв'язання СНР вимагають, щоб початкове наближення було задане достатньо близько до шуканого розв'язку.

Таким чином розв'язання СНР, так само як і скалярних рівнянь, складається з двох етапів:

1. Локалізація коренів.
2. Уточнення коренів.

Оскільки СНР виникають внаслідок розгляду фізичних та хімічних процесів, то інколи виходячи зі змісту поставленої задачі можна накласти певні обмеження на розв'язок СНР та відповідно на початкове наближення, яке вибирається при уточненні коренів.

Метод простої ітерації (метод послідовних наближень)

Нехай СНР (5.2) перетворена до вигляду:

$$\mathbf{x} = \Phi(\mathbf{x}), \quad (5.3)$$

де $\Phi(\mathbf{x}) = (\varphi_1(\mathbf{x}); \varphi_2(\mathbf{x}); \dots; \varphi_n(\mathbf{x}))^T$.

Задача (5.3) є задачею про нерухому точку нелінійного відображення $\Phi: \mathbb{R}_n \rightarrow \mathbb{R}_n$. Для цієї задачі можна записати рекурентну рівність:

$$\mathbf{x}^{(k+1)} = \Phi(\mathbf{x}^{(k)}), \quad k = 0, 1, 2, \dots, \quad (5.4)$$

яка визначає метод простої ітерації.

Теорема 3.1. *Нехай функція $\Phi(\mathbf{x})$ та замкнена множина $M \subseteq D(\Phi) \subseteq \mathbb{R}_n$ ($D(\Phi)$ це область визначення функції $\Phi(\mathbf{x})$) такі, що:*

1. $\Phi(\mathbf{x}) \in M \quad \forall \mathbf{x} \in M$;
2. $\exists q : \|\mathbf{J}(\Phi(\mathbf{x}))\| \leq q < 1$ та $\|\Phi(\mathbf{x}) - \Phi(\tilde{\mathbf{x}})\| \leq q \|\mathbf{x} - \tilde{\mathbf{x}}\| \quad \forall \mathbf{x}, \tilde{\mathbf{x}} \in M$.

Тоді $\Phi(\mathbf{x})$ має в M єдину нерухому точку \mathbf{x}^* ; послідовність $(\mathbf{x}^{(k)})$, що визначається МПІ (5.4), при будь-якому $\mathbf{x}^{(0)} \in M$ збігається до \mathbf{x}^* та справедливі оцінки:

$$\|\mathbf{x}^* - \mathbf{x}^{(k)}\| \leq \frac{q}{1-q} \|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\| \leq \frac{q^k}{1-q} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\| \quad \forall k \in \mathbb{N}. \quad (5.5)$$

Одним із серйозних недоліків методу простої ітерації є складність вибору функції $\Phi(\mathbf{x})$, яка задовольняла достатню умову збіжності (п.2 теореми 3.1). Тому дуже часто для розв'язання СНР застосовують методи, які є частковим випадком методу простої ітерації.

Якщо (\mathbf{A}_k) – деяка послідовність неособливих дійсних $n \times n$ -матриць, то послідовність задач

$$\mathbf{x} = \mathbf{x} - \mathbf{A}_k F(\mathbf{x}), \quad k = 0, 1, 2, \dots \quad (5.6)$$

має ті самі розв'язки, що й початкове рівняння $F(\mathbf{x}) = \mathbf{0}$ та для знаходження їх наближеного розв'язку можна записати ітераційний процес

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{A}_k F(\mathbf{x}^{(k)}), \quad k = 0, 1, 2, \dots, \quad (5.7)$$

що має вид методу простої ітерації (5.4) при $\Phi(\mathbf{x}) = \Phi_k(\mathbf{x}) = \mathbf{x} - \mathbf{A}_k F(\mathbf{x})$.

Метод Ньютона та його модифікації

Припускаючи, що функція $F(\mathbf{x})$ достатню кількість разів неперервна диференційовна, розкладемо її в околі точки $\mathbf{x}^{(k)}$ в ряд Тейлора, залишивши тільки лінійні доданки та отримане розкладення підставимо у рівняння $F(\mathbf{x}) = \mathbf{0}$:

$$F(\mathbf{x}^{(k)}) + F'(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)}) = \mathbf{0}. \quad (5.8)$$

Замінивши точку \mathbf{x} на $\mathbf{x}^{(k+1)}$ в (5.8) отримаємо:

$$F'(\mathbf{x}^{(k)})\left(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right) = -F(\mathbf{x}^{(k)}). \quad (5.9)$$

Таким чином ми отримали *метод Ньютона розв'язання СНР в неявному вигляді*. Виразивши $\mathbf{x}^{(k+1)}$ через інші елементи рівності (5.9), отримаємо формулу, що визначає *явний метод Ньютона розв'язання СНР*:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \left[F'(\mathbf{x}^{(k)})\right]^{-1} F(\mathbf{x}^{(k)}). \quad (5.10)$$

Існує ряд теорем, що встановлюють швидкість збіжності методу Ньютона в загальному випадку та при тих чи інших припущеннях. Узагальнюючи результати цих теорем відмітимо, що при достатній гладкості $F(\mathbf{x})$ та достатньо гарному початковому наближенні $\mathbf{x}^{(0)}$ збіжність послідовності $(\mathbf{x}^{(k)})$, що породжується методом Ньютона, до розв'язку СНР буде квадратичною так само, як і в одномірному випадку.

Якщо матрицю Якобі $F'(\mathbf{x})$ обчислити та знайти обернену до неї лише один раз – в початковій точці $\mathbf{x}^{(0)}$, то отримаємо *спрощений метод Ньютона*:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \left[F'(\mathbf{x}^{(0)})\right]^{-1} F(\mathbf{x}^{(k)}) \quad (5.11)$$

Явний метод Ньютона (5.10) можна модифікувати таким чином (цей метод дістав назву *двоступеневий метод Ньютона*):

$$\begin{cases} \mathbf{z}^{(k)} = \mathbf{x}^{(k)} - \mathbf{A}_k F(\mathbf{x}^{(k)}), \\ \mathbf{x}^{(k+1)} = \mathbf{z}^{(k)} - \mathbf{A}_k F(\mathbf{z}^{(k)}), \end{cases} \quad (5.12)$$

де $\mathbf{A}_k = \left[F'(\mathbf{x}^{(k)})\right]^{-1}$.

Пошук оберненої матриці Якобі на кожному кроці явного методу Ньютона (5.10) можна виконувати не точно, а наближено. Для цього можна, наприклад, застосувати ітераційний процес Шульца, обмежившись лише одним кроком процесу другого порядку, в якому за початкову матрицю приймається матриця, отримана в результаті попереднього $(k-1)$ -го кроку. Така модифікація методу Ньютона має назву *апроксимаційний аналог №1 методу Ньютона*.

Застосування тієї ж послідовності апроксимацій обернених матриць до найпростішого двоступеневого методу Ньютона (5.12) визначає його *апроксимаційний аналог*:

$$\begin{cases} \mathbf{z}^{(k)} = \mathbf{x}^{(k)} - \mathbf{A}_k F(\mathbf{x}^{(k)}), \\ \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{A}_k F(\mathbf{z}^{(k)}), \\ \mathbf{\Psi}_k = \mathbf{I} - F'(\mathbf{x}^{(k+1)}) \mathbf{A}_k, \\ \mathbf{A}_{k+1} = \mathbf{A}_k + \mathbf{A}_k \mathbf{\Psi}_k. \end{cases} \quad (5.13)$$

Відмітимо, що збільшити швидкість збіжності апроксимаційних аналогів методу Ньютона можна шляхом підвищення порядку апроксимації обернених матриць, наприклад, за рахунок додавання ще одного доданку в формулі для підрахунку \mathbf{A}_{k+1} .

Такі методи будемо називати *апроксимаційними аналогами №2 методу Ньютона*.

Явний метод Ньютона можна модифікувати замінивши часткові похідні в матриці Якобі різницевиими відношеннями, тобто в якості матриці \mathbf{A}_k використовувати матрицю \mathbf{V} елементи якої обчислюються таким чином:

$$v_{ij} = \frac{f_i(x_1, \dots, x_j + h_j, \dots, x_n) - f_i(x_1, \dots, x_j, \dots, x_n)}{h_j}, \quad i = 1..n, \quad j = 1..n. \quad (5.14)$$

Шляхом такої дискретизації отримуємо різницевий (або дискретний) метод Ньютона:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{V}_k F(\mathbf{x}^{(k)}). \quad (5.15)$$

При вдалому виборі послідовності малих векторів $\mathbf{h}^{(k)}$ (сталій або такої, що збігається до нуля) метод (5.15) має надлінійну швидкість збіжності.

Знаходження послідовності $\mathbf{h}^{(k)}$ можна зв'язати з деякої векторною послідовністю, що збігається до нуля, наприклад з послідовністю нев'язок $F(\mathbf{x}^{(k)})$ або поправок $\mathbf{p}^{(k)}$. Реалізуючи другий варіант маємо $h_j^{(k)} = x_j^{(k-1)} - x_j^{(k)}$, де $j=1..n$, а $k=1, 2, 3, \dots$. В такому випадку отримуємо узагальнення скалярного методу січних:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{B}_k^{-1} F(\mathbf{x}^{(k)}), \quad (5.16)$$

де елементи матриці \mathbf{B}_k знаходяться за формулою

$$b_{ij}^{(k)} = \frac{f_i \left(x_1^{(k)}, \dots, \overbrace{x_j^{(k-1)} + h_j^{(k)}}^{x_j^{(k-1)}}, \dots, x_n^{(k)} \right) - f_i \left(x_1^{(k)}, \dots, x_j^{(k)}, \dots, x_n^{(k)} \right)}{x_j^{(k-1)} - x_j^{(k)}}, \quad k=1, 2, 3, \dots \quad (5.17)$$

Так само як і для випадку $n=1$ збіжність методу січних має порядок $\frac{1+\sqrt{5}}{2}$.

До методу січних так само як і до методу Ньютона можна застосувати покрокову апроксимацію обернених матриць, на основі методу Шульца та отримати відповідні апроксимаційні аналоги методу січних.

Варто відмітити, що для функції n змінних відомо декілька різновидів метода січних. Запропонований метод січних (5.16) є представником

сімейства методу січних, що базується на апроксимації матриць Якобі. Різні способи апроксимації матриць Якобі призводять до різних варіантів методу січних.

Метод січних Бройдена

Нагадаємо формулу ітераційного процесу для методу січних при $n = 1$:

$$x_{k+1} = x_k - b_k^{-1} \cdot f(x_k), \quad (5.18)$$

де $b_k = \frac{f(x_{k-1}) - f(x_k)}{x_{k-1} - x_k}$.

Вираз для обчислення коефіцієнту b_k переписаний таким чином

$$b_k(x_{k-1} - x_k) = f(x_{k-1}) - f(x_k) \quad (5.19)$$

називається *співвідношенням січних* в \mathbb{R}_1 . Воно легко узагальнюється на n -мірний випадок:

$$\mathbf{B}_k(\mathbf{x}^{(k-1)} - \mathbf{x}^{(k)}) = F(\mathbf{x}^{(k-1)}) - F(\mathbf{x}^{(k)}), \quad (5.20)$$

де \mathbf{B}_k – деяка матриця лінійного перетворення в \mathbb{R}_n .

Введемо позначення $\mathbf{s}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}$ та $\mathbf{y}^{(k)} = F(\mathbf{x}^{(k)}) - F(\mathbf{x}^{(k-1)})$. Тоді співвідношення січних в \mathbb{R}_n прийме вигляд:

$$\mathbf{B}_k \mathbf{s}^{(k)} = \mathbf{y}^{(k)}. \quad (5.21)$$

По аналогії з одномірним випадком ітераційна формула методу січних приймає наступний вигляд:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{B}_k^{-1} \cdot F(\mathbf{x}^{(k)}). \quad (5.22)$$

Для того, щоб ця ітераційна формула узагальнювала метод січних (5.18), неособливу $n \times n$ -матриця \mathbf{B}_k потрібно підбирати так, щоб

вона задовольняла співвідношення січних (5.21), тобто матриця \mathbf{B}_k має перетворювати заданий вектор $\mathbf{s}^{(k)}$ в інший заданий вектор $\mathbf{y}^{(k)}$. Не складно зрозуміти, що при $n > 1$ ця вимога однозначно не визначає матрицю \mathbf{B}_k . Звідси можна зробити висновок, що можуть бути різні узагальнення одномірного методу січних.

Перепишемо рівність (5.22) замінивши $\mathbf{x}^{(k+1)}$ на \mathbf{x} :

$$F(\mathbf{x}^{(k)}) + \mathbf{B}_k(\mathbf{x} - \mathbf{x}^{(k)}) = \mathbf{0}. \quad (5.23)$$

Позначимо результат виразу в лівій частині рівності (5.23), як вектор \mathbf{q}_k , тобто

$$\mathbf{q}_k = F(\mathbf{x}^{(k)}) + \mathbf{B}_k(\mathbf{x} - \mathbf{x}^{(k)}). \quad (5.24)$$

При виборі матриці \mathbf{B}_k Чарльз Джордж Бройден виходив з того, що зміни у виразі \mathbf{q}_k порівняно з \mathbf{q}_{k-1} мають бути мінімальними.

На основі цих міркувань можна побудувати алгоритм №1, що реалізує метод Бройдена:

1. Якщо $k = 0$, то задати початковий вектор $\mathbf{x}^{(0)}$ та матрицю \mathbf{B}_0 , інакше обчислити чергову матрицю \mathbf{B}_{k+1} за формулою:

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{(\mathbf{y}^{(k+1)} - \mathbf{B}_k \mathbf{s}^{(k+1)}) \cdot (\mathbf{s}^{(k+1)})^T}{(\mathbf{s}^{(k+1)})^T \cdot \mathbf{s}^{(k+1)}}. \quad (5.25)$$

2. Розв'язати СЛАР:

$$\mathbf{B}_k \mathbf{s}^{(k+1)} = -F(\mathbf{x}^{(k)}) \quad (5.26)$$

відносно вектора $\mathbf{s}^{(k+1)}$ (див. (5.22)).

3. Знайти вектори $\mathbf{x}^{(k+1)}$ та $\mathbf{y}^{(k+1)}$:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k+1)}; \quad (5.27)$$

$$\mathbf{y}^{(k+1)} = F(\mathbf{x}^{(k+1)}) - F(\mathbf{x}^{(k)}). \quad (5.28)$$

4. Виконати перевірку критерію останови: наприклад, перевірити на малість величини $\|\mathbf{s}^{(k+1)}\|$ та/або $\|\mathbf{y}^{(k+1)}\|$. Якщо задана точність не досягнута, то збільшити значення k на одиницю та перейти до п.1.

В якості матриці \mathbf{B}_0 для запуску ітераційного процесу Бroyдена беруть матрицю Якобі $F'(\mathbf{x}^{(0)})$ або яку-небудь її апроксимацію.

Можна довести, що при достатній близькості $\mathbf{x}^{(0)}$ до \mathbf{x}^* та \mathbf{B}_0 до $F'(\mathbf{x}^{(0)})$ буде спостерігатися *лінійна збіжність* послідовності $(\mathbf{x}^{(k)})$ до \mathbf{x}^* , а при деяких додаткових умовах послідовність наближень за методом Бroyдена буде мати *надлінійну збіжність*.

Спростивши формулу перерахунку (5.25) отримаємо наступну формулу:

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{F(\mathbf{x}^{(k+1)}) \cdot (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})^T}{\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|_2^2}. \quad (5.29)$$

Відмітимо, що при використанні формули перерахунку (5.29) немає необхідності обчислювати вектор $\mathbf{y}^{(k+1)}$. Такий алгоритм реалізації методу Бroyдена назвемо алгоритмом №2.

Метод Брауна

На відміну від покрокової лінеаризації векторної функції $F(\mathbf{x})$, що приводить до методу Ньютона (5.10), Кеннетом М. Брауном у 1966 році запропоновано проводити на кожному ітераційному кроці почергову лінеаризацію компонент вектор-функції $F(\mathbf{x})$ та послідовно розв'язувати отримані таким чином рівняння.

Нехай необхідно знайти розв'язок системи:

$$\begin{cases} f(x, y) = 0; \\ g(x, y) = 0. \end{cases} \quad (5.30)$$

При реалізації метода Брауна розв'язання системи необхідно виконувати обчислення при $k = 0, 1, 2, \dots$ за допомогою сукупності формул:

$$\begin{aligned} \tilde{x}_k &= x_k - \frac{f(x_k, y_k)}{f'_x(x_k, y_k)}, \\ q_k &= \frac{g(\tilde{x}_k, y_k) \cdot f'_x(x_k, y_k)}{f'_x(x_k, y_k) g'_y(\tilde{x}_k, y_k) - f'_y(x_k, y_k) g'_x(\tilde{x}_k, y_k)}, \\ p_k &= \frac{f(x_k, y_k) - q_k f'_y(x_k, y_k)}{f'_x(x_k, y_k)}, \\ x_{k+1} &= x_k - p_k, \\ y_{k+1} &= y_k - q_k. \end{aligned} \quad (5.31)$$

Обчислення за методом Брауна необхідно закінчувати, коли буде виконуватись нерівність $\max\{|p_{k-1}|, |q_{k-1}|\} < \varepsilon$. В процесі обчислень також необхідно контролювати малість знаменника в обчислювальних формулах та збіжність ітераційної послідовності.

Метод Брауна має квадратичну швидкість збіжності, але якщо часткові похідні замінити різницеvими відношеннями, то можна сподіватися на більшу ефективність цього методу порівняно з методом Ньютона.

Завдання на лабораторну роботу

Розробити програму мовою програмування C# у середовищі розробки Visual Studio 2013 (або вище), яка буде працювати у віконному режимі та дозволяти уточнювати (точність $\varepsilon \leq 10^{-7}$) корені системи нелінійних рівнянь, заданої за варіантом (табл. 5.1):

1. Методом, який задано в третьому стовпчику *табл. 5.1*, де

0	Явний метод Ньютона (обчислення оберненої матриці одним з прямих методів)
1	Неявний метод Ньютона (обґрунтувати вибір способу розв'язання СЛАР)
2	Спрощений метод Ньютона
3	Двоступеневий метод Ньютона (обчислення оберненої матриці прямими методами)
4	Апроксимаційний аналог №1 метода Ньютона
5	Апроксимаційний аналог №2 метода Ньютона
6	Апроксимаційний аналог №1 двоступеневого метода Ньютона
7	Апроксимаційний аналог №2 двоступеневого метода Ньютона

та проводити замір часу роботи даного методу.

2. Дискретним аналогом метода з п.1 та проводити замір часу роботи даного методу.

3. Методом, який задано в четвертому стовпчику *табл. 5.1*, де

0	Явний метод січних (обчислення оберненої матриці прямими методами)
1	Неявний метод січних (обґрунтувати вибір способу розв'язання СЛАР)
2	Апроксимаційний аналог №1 метода січних
3	Апроксимаційний аналог №2 метода січних

та проводити замір часу роботи даного методу.

4. Зейделевою модифікацією методів з п.1, п.2 та п.3 та проводити заміри часу роботи даних методів.

Розв'язати задану за варіантом систему нелінійних рівнянь (табл. 5.1) у математичному пакеті *MatLab* з точністю $\varepsilon \leq 10^{-7}$.

Вказівки щодо виконання завдання

При розв'язанні у *MatLab* використати функцію *fsolve*, попередньо встановивши опції виводу результатів таким чином:

```
options=optimset('Display','iter');
```

Додаткове завдання

До розробленої програми додати можливість розв'язувати систему нелінійних рівнянь:

1. Алгоритмом, що реалізує метод січних Бройдена, який заданий за варіантом (номер варіанту студента по модулю 2), де

0	Алгоритм №1
1	Алгоритм №2

та проводити замір часу роботи даного алгоритму.

2. Методом Брауна та проводити замір часу роботи даного методу.

Вимоги до оформлення звіту

Звіт має включати:

1. Постановку задачі за варіантом.
2. Математичне підґрунття для виконання даної лабораторної роботи (перелік формул, що були використані при розробленні програми на мові програмування C#).
3. Основні етапи процесу локалізації коренів системи нелінійних рівнянь, що розв'язується.
4. Значення коренів отримані методами заданими за варіантом, звести до таблиці (при необхідності дозволяється таблицю розміщувати в альбомному форматі):

Метод з п.1 (вказати назву методу)	Зейделева модифікація методу з п.1	Метод з п.2 (вказати назву методу)	Зейделева модифікація методу з п.2 (вказати назву методу)	Метод з п.3 (вказати назву методу)	Зейделева модифікація методу з п.3 (вказати назву методу)	<i>MatLab</i>

5. Кількість ітерацій необхідну для уточнення коренів методами заданими за варіантом до таблиці (при необхідності дозволяється таблицю розміщувати в альбомному форматі):

Метод з п.1 (вказати назву методу)	Зейделева модифікація методу з п.1	Метод з п.2 (вказати назву методу)	Зейделева модифікація методу з п.2 (вказати назву методу)	Метод з п.3 (вказати назву методу)	Зейделева модифікація методу з п.3 (вказати назву методу)	<i>MatLab</i>

6. Час роботи методів уточнення коренів заданих за варіантом звести до таблиці (при необхідності дозволяється таблицю розміщувати в альбомному форматі):

Метод з п.1 (вказати назву методу)	Зейделева модифікація методу з п.1	Метод з п.2 (вказати назву методу)	Зейделева модифікація методу з п.2 (вказати назву методу)	Метод з п.3 (вказати назву методу)	Зейделева модифікація методу з п.3 (вказати назву методу)	<i>MatLab</i>

7. Висновки.

Варианти завдань

Номер варіанту визначається за наступною таблицею:

№ за списком викладача	Варіант №	№ за списком викладача	Варіант №
1	16	13	8
2	1	14	20
3	17	15	2
4	22	16	19
5	6	17	18
6	23	18	7
7	5	19	14
8	13	20	4
9	12	21	15
10	11	22	3
11	24	23	10
12	21	24	9

Таблиця 5.1. Перелік систем нелінійних рівнянь

№	Система нелінійних рівнянь	Метод з п.1	Метод з п.3
1	$\begin{cases} x^2 + y^3 = 9; \\ x^5 = \log_2 9 - y . \end{cases}$	3	1
2	$\begin{cases} x^5 + y^3 + \sin y = x; \\ (x^3 - y)^2 = \log_{10} 9 - y . \end{cases}$	4	0
3	$\begin{cases} y^3 - \sin x = \log_2 x; \\ (x - y^3)^3 = \ln 9 - y . \end{cases}$	1	3
4	$\begin{cases} \sin(y^3) - x + 6 = \log_2 y; \\ \sin(x - y^2) = \ln 9 - x . \end{cases}$	5	2
5	$\begin{cases} \sin x - y = \log y ; \\ e^{\sin(x-y^2)} = 9 - x . \end{cases}$	3	2
6	$\begin{cases} \sin(e^x) - \cos y = \log_{10} y^x + 7; \\ e^{\sin(x-y)} - x + 15 = 9 - x - y . \end{cases}$	7	3
7	$\begin{cases} \log_2 y = \log_{10} y^x + x; \\ e^{\cos(0.1x-3y)} - y + x = 10 - y . \end{cases}$	0	1
8	$\begin{cases} x^7 + \pi \log_2 y = \sqrt{ y } + x; \\ x^9 - e^{3y} - \sqrt{y+x} = 1 - x . \end{cases}$	6	0
9	$\begin{cases} x^5 + \log_2(\pi x) = \sqrt{ y +x}; \\ e^{3xy} - \sqrt{y+x} = 1 - x . \end{cases}$	2	2

Продовження табл. 5.1

№	Система нелінійних рівнянь	Метод з п.1	Метод з п.3
10	$\begin{cases} x^9 + y \cdot \log_{10} x = x \cdot y ; \\ e^{3xy} - x\sqrt{ x \cdot y } = y . \end{cases}$	7	3
11	$\begin{cases} x^3 + y \cdot \log_2 x = x \cdot y ; \\ \log_2(3xy) - x\sqrt{x} = y - x; \\ x > 0. \end{cases}$	4	2
12	$\begin{cases} x + y = x^{\sin x} + 2\pi; \\ \operatorname{ch}(3y - \pi) - 3x\sqrt{y} = 8y - x + 18; \\ x \in (0; 30). \end{cases}$	6	0
13	$\begin{cases} e^y - x = 2^{x+7} - 5\pi \cdot x - 7; \\ \operatorname{sh}(3y) - x\sqrt{ y } = 7y - x + 5. \end{cases}$	1	3
14	$\begin{cases} x = \cos^3 y + x^2 \log_2(\operatorname{sh} x + \operatorname{ch} x) - x^2 + 15; \\ x + y = \cos^3 x + (x \cdot \log_2(\operatorname{sh} y))^3 - x^3 - 5. \end{cases}$	2	1
15	$\begin{cases} \operatorname{sh}(y + 5) = \cos^3 x + x^3 \operatorname{sh} x - x^5 - 35; \\ \cos x = y^2 + x^3 - x^6 + 25. \end{cases}$	5	3
16	$\begin{cases} \operatorname{sh}(x + 5) = \cos^2(x + y) + x^2 \cdot \operatorname{sh} y - x^2 + 15; \\ \cos y = x^2 + x^3 - y^6 + 205\pi. \end{cases}$	3	1
17	$\begin{cases} \operatorname{sh}(y + 5) = \sin^3(x + y) + x^3 \operatorname{ch} y - x^3 + 15; \\ \sin y = y^2 + x^3 - y^5 + y^8 - 155\pi. \end{cases}$	0	2
18	$\begin{cases} \operatorname{sh}(x + y + 5) = \sin^3(x - 5) + x^2 \operatorname{sh} y + 15; \\ \sin y = x^3 - y^5 + y^8 - 155\pi \cdot x - 56\pi^4. \end{cases}$	7	0

Продовження табл. 5.1

№	Система нелінійних рівнянь	Метод з п.1	Метод з п.3
19	$\begin{cases} 5 \cdot \cos(3y + 5) = \pi \sin^3(x - 5) + 5y^3 \operatorname{ch}(x + y); \\ \operatorname{sh} x = \sqrt{y^3} - \log_2(x^5 - y). \end{cases}$	6	1
20	$\begin{cases} \operatorname{ch}(3y + 5) = \pi \sin^2(x - 5) + y^2 \operatorname{ch}(y); \\ \operatorname{sh} x = 15 \left(\sqrt{ y + 7 } \right)^3 - \log_{10} x^3 + y . \end{cases}$	5	0
21	$\begin{cases} \sin y = \sin^2 x + x^4 - x^2 - \cos^2 x - 13x - 10; \\ \operatorname{ch} y = \sqrt{ \sin x } + x^3 - \cos^2 y - x + 10. \end{cases}$	1	3
22	$\begin{cases} \sin(y + x) = \sin^2 y + x^4 - y^2 - \cos^2 x - 13\pi x - 10; \\ \cos y = \sqrt{ \operatorname{sh} x } + x^3 - \cos^3 y - x^2 - 55\pi. \end{cases}$	4	2
23	$\begin{cases} y + x = \sin y + y^3 - \cos^2 y - 13\pi y + 10; \\ \operatorname{ch} y = \operatorname{sh}(y + x) + y^3 - \cos^3 y - y^2 - 35\pi. \end{cases}$	2	0
24	$\begin{cases} \cos x = \sin(y - x) + y^2 - \cos^3 x - 3y + x - 7; \\ \operatorname{sh} y = \cos(y + x) + y^5 - \operatorname{ch} y + 15\pi y^2. \end{cases}$	0	3

Питання для самоперевірки

1. Запис системи нелінійних рівнянь у векторно-матричному вигляді.
2. Які методи розв'язання нелінійного рівняння не можна узагальнити на випадок системи нелінійних рівнянь? Чому?
3. Геометричний зміст системи нелінійних рівнянь.
4. Чи існують прямі методи розв'язання систем нелінійних рівнянь?
5. Яку кількість розв'язків може мати система нелінійних рівнянь?
6. З яких етапів складається процес розв'язання системи нелінійних рівнянь?
7. Які існують підходи для визначення початкового наближення до розв'язку системи нелінійних рівнянь?
8. Метод простої ітерації та задача про нерухому точку в контексті розв'язання системи нелінійних рівнянь.
9. Теорема про збіжність ітераційної послідовності, що генерується МПІ при розв'язанні системи нелінійних рівнянь.
10. Яка множина називається замкненою?
11. Часткові випадки достатніх умов збіжності послідовності, що генерується МПІ при розв'язанні системи нелінійних рівнянь.
12. Підходи до вибору матричного параметру в методі простої ітерації розв'язання системи нелінійних рівнянь.
13. Недоліки методу простої ітерації розв'язання системи нелінійних рівнянь.
14. Виведення ітераційної формули явного методу Ньютона для розв'язання системи нелінійних рівнянь.
15. Вивід ітераційної формули неявного методу Ньютона для розв'язання системи нелінійних рівнянь.

16. Чи виникає необхідність в розв'язанні СЛАР при використанні неявного методу Ньютона для розв'язання системи нелінійних рівнянь. Чому виникає така необхідність? Відповідь обґрунтувати.
17. Яким чином при розв'язанні системи нелінійних рівнянь ітераційними методами використовується матриця Якобі?
18. Як побудувати матрицю Якобі?
19. Спощений метод Ньютона для розв'язання системи нелінійних рівнянь та ідеї, що спонукали для його отримання.
20. Порівняння явного, неявного та спрощеного методу Ньютона для розв'язання системи нелінійних рівнянь.
21. Ітераційні формули та ідея двоступеневого методу Ньютона розв'язання системи нелінійних рівнянь.
22. Апроксимаційні аналоги методу Ньютона для розв'язання системи нелінійних рівнянь. Ідея та ітераційні формули.
23. Якщо не вдається обрати гарне початкове наближення, то яким чином виконувати уточнення коренів в такому випадку?
24. В чому полягає ідея дискретних аналогів різноманітних модифікацій методу Ньютона?
25. Порівняння дискретних модифікацій методу Ньютона з їх неперервними аналогами. Яким чином необхідно обирати крок дискретизації?
26. Метод січних розв'язання систем нелінійних рівнянь та його зв'язок з дискретними аналогами різноманітних модифікацій методу Ньютона.
27. Які методи розв'язання систем нелінійних рівнянь є однокроковими та двокроковими? Чому?
28. Співвідношення січних в загальному випадку.
29. Довести, що метод січних для розв'язання одного нелінійного рівняння має декілька узагальнень на багатовимірний випадок.

30. Виведення методу січних Бройдена розв'язання систем нелінійних рівнянь.
31. Алгоритми, що реалізують метод січних Бройдена розв'язання систем нелінійних рівнянь.
32. Виведення другого алгоритму на основі першого алгоритму, що реалізує метод січних Бройдена розв'язання систем нелінійних рівнянь.
33. Основна ідея метода Брауна розв'язання систем нелінійних рівнянь.
34. Виведення ітераційних формул метода Брауна для розв'язання систем двох нелінійних рівнянь. Який вигляд приймуть ці формули для системи трьох нелінійних рівнянь?
35. Ідея та ітераційні формули Зейделевої модифікації методів розв'язання систем нелінійних рівнянь.
36. Критерій зупинки ітераційних процесів в методах розв'язання систем нелінійних рівнянь.
37. Порядок збіжності методів розв'язання систем нелінійних рівнянь.
38. Які загальні недоліки можна виділити у методів розв'язання систем нелінійних рівнянь?
39. Показати зв'язок задачі розв'язання системи нелінійних рівнянь із задачею оптимізації.

Рекомендована література

1. William H. Press Numerical Recipes in C. The Art of Scientific Computing / William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery // Cambridge University Press, 2002. – 1018 p.
2. Walter Gautschi Numerical Analysis [Електронний ресурс], 2012. Режим доступу: http://www.ikiu.ac.ir/public-files/profiles/items/090ad_1410599906.pdf

3. McDonough J. M. Computational Numerical Analysis [Электронный ресурс], 2007. Режим доступа: <http://web.engr.uky.edu/~acfd/egr537-lctrs.pdf>

ЛАБОРАТОРНА РОБОТА №6. АПРОКСИМАЦІЯ ФУНКЦІЙ

Мета роботи: опанувати основні методи апроксимації нелінійних функцій.

Теоретичні відомості

В основі багатьох методів математичного аналізу лежить заміна однієї функції $f(x)$ іншою функцією $\varphi(x)$, що є близькою до $f(x)$, але більш простою, що дозволяє досить просто виконувати над нею певні операції. Таку заміну функції будемо називати *апроксимацією* або просто наближенням функції $f(x)$ функцією $\varphi(x)$.

Якщо функція задана таблицею, то під *інтерполяцією* розуміють знаходження проміжних значень таблично-заданої функції строго в середині таблиці, а під *екстраполяцією* – знаходження її наближених значень за межами проміжку заданих точок, тобто вузлових точок.

Нехай в точках x_0, x_1, \dots, x_n таких, що $a \leq x_0 < \dots < x_n \leq b$, відомі значення функції $y = f(x)$, тобто на відрізку $[a, b]$ задана таблична (сіткова) функція (табл. 6.1).

Таблиця 6.1. Таблично-задана функція $f(x)$

x	x_0	x_1	...	x_n
y	y_0	y_1	...	y_n

Задача поліноміальної інтерполяції для таблично-заданої функції $f(x)$ формулюється наступним чином: знайти многочлен $P_n(x)$ такий, що виконується сукупність умов інтерполяції

$$P_n(x_i) = y_i \quad \forall i \in \mathbb{N}_0. \quad (6.1)$$

Інтерполяційний многочлен Лагранжа

Будемо будувати многочлен n -го степеня $L_n(x)$ у вигляді лінійної комбінації $\sum_{i=0}^n c_i l_i(x)$ многочленів $l_i(x)$ степеня n ($i \in \mathbb{N}_0$). Для того, щоб такий многочлен був інтерполяційним для функції $f(x)$, достатньо зафіксувати в якості коефіцієнтів c_i цієї лінійної комбінації задані таблицею 2.1.1 значення $y_i = f(x_i)$, а від базисних многочленів $l_i(x)$ вимагати виконання умови

$$l_i(x_j) = \delta_{ij} = \begin{cases} 0, & \text{якщо } j \neq i \\ 1, & \text{якщо } j = i \end{cases} \quad \forall j, i \in \mathbb{N}_0 \quad (6.2)$$

На основі цих міркувань отримуємо інтерполяційний многочлен Лагранжа:

$$L_n(x) = \sum_{i=0}^n \frac{(x-x_0)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)} y_i. \quad (6.3)$$

Інтерполяційна схема Ейткена

Безпосереднє використання многочлена Лагранжа у формі (6.3) є не дуже зручним через його громіздкість, що є причиною великих обчислювальних витрат. Якщо функція задана таблично, то, зазвичай, завчасно невідомо якої степені многочлен потрібно використовувати для інтерполявання даної функції з заданою точністю, а поступове збільшення

точності за рахунок повторних обчислень значень $L_n(x)$ кожного разу збільшуючи показник степеня n при прямому застосуванні формули (6.3) є неприйнятним, через погану перебудовуваність $L_{n-1}(x)$ в $L_n(x)$.

Можливо побудувати рекурентне задання послідовності інтерполяційних многочленів Лагранжа, яке називають *інтерполяційною схемою Ейткена*:

$$f(x) \approx P_{i,i+k}(x) = \frac{1}{x_{i+k} - x_i} \left| \begin{array}{cc} x - x_i & P_{i,i+k-1}(x) \\ x - x_{i+k} & P_{i+1,i+k}(x) \end{array} \right|, \quad (6.4)$$

де $k = 1, 2, \dots$; $P_{i,i}(x) = y_i$.

Організація обчислень за формулою (6.4) повинна бути такою, що коли завчасно не відома степінь інтерполяційного многочлена, який слід використати для обчислення $y(\tilde{x})$, та дана таблиця значень функції достатньо велика, то має виконуватись поступове підвищення степеня k інтерполуючих її многочленів за рахунок підключення нових, все більш віддалених від \tilde{x} вузлів.

Скінченно-різницеві інтерполяційні формули

Скінченні різниці є аналогом похідних. Звідси є справедливими багато їх властивостей, які співпадають з властивостями похідних.

На основі цього можна зробити висновок: якщо скінченні різниці n -го порядку деякої функції $y = f(x)$ стали в будь-якій точці x при різних фіксованих кроках h , то ця функція $f(x)$ є многочлен степеня n .

Для функції $y = f(x)$, що задана таблицею своїх значень y_0, y_1, \dots, y_n у вузлах x_0, x_1, \dots, x_n , де $x_i = x_0 + ih$, $i = 0, 1, \dots, n$, $h > 0$ – деяка стала величина, що називається кроком сітки (таблиці), скінченні різниці різних порядків зручно розміщувати в одній спільній таблиці (див. табл. 4.2) з

вузлами та значеннями функції (останні можна інтерпретувати, як скінченні різниці нульового порядку). Цю таблицю називають таблицею скінченних різниць.

Таблиця 6.2. Діагональна таблиця скінченних різниць

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$...
x_0	y_0					
		Δy_0				
x_1	y_1		$\Delta^2 y_0$			
		Δy_1		$\Delta^3 y_0$		
x_2	y_2		$\Delta^2 y_1$		$\Delta^4 y_0$	
		Δy_2		$\Delta^3 y_1$...
x_3	y_3		$\Delta^2 y_2$		$\Delta^4 y_1$	
		Δy_3		$\Delta^3 y_2$		
x_4	y_4		$\Delta^2 y_3$		\vdots	
		Δy_4		\vdots		
x_5	y_5		\vdots			
		\vdots				
\vdots	\vdots					

Якщо в інтерполяційному многочлені Лагранжа (6.3) всі доданки однотипні та відіграють однакову роль в утворенні результату, то хотілося б мати таке подання інтерполяційного многочлена, в якому доданки розміщувались в порядку спадання їх значущості. Така структура інтерполяційного многочлена дозволила б більш просто перебудувати його степінь, додаючи або відкидаючи віддалені від початку його запису члени.

На основі цих міркувань отримується *перша інтерполяційна формула Ньютона*:

$$f(x) \approx P_n(x_0 + qh) = y_0 + q\Delta y_0 + \frac{q(q-1)}{2!} \Delta^2 y_0 + \dots + \frac{q(q-1)\dots(q-n+1)}{n!} \Delta^n y_0 \quad (6.5)$$

Дану формулу доцільніше за все використовувати при значеннях $|q| < 1$, а саме, для *інтерполювання вперед* (при $x \in (x_0, x_1)$), тобто при $q \in (0, 1)$) та *екстраполювання назад* (при $x < x_0$, тобто при $q < 0$).

Можна легко дійти висновку, що за базовий вузол для формули (6.5) є сенс приймати найближчий до заданої фіксованої точки x , якщо за ним є достатня кількість вузлів для побудови необхідних для (6.5) скінченних різниць. Тобто ця формула є непринятною для інтерполювання в кінці таблиці.

Для інтерполювання вкінці відрізка використовується *друга інтерполяційна формула Ньютона*:

$$f(x) \approx P_n(x_n + qh) = y_n + q\Delta y_{n-1} + \frac{q(q-1)}{2!} \Delta^2 y_{n-2} + \dots + \frac{q(q-1)\dots(q-n+1)}{n!} \Delta^n y_0. \quad (6.6)$$

Другу інтерполяційну формулу Ньютона доцільно використовувати при значеннях $|q| < 1$, тобто в околицях вузла x_n для інтерполювання назад (при $q \in (-1, 0)$) та екстраполювання вперед (при $q > 0$).

Таким чином, інтерполяційні формули Ньютона перш за все призначені для інтерполяції на початку або в кінці таблиці. Але здебільшого потрібно виконувати інтерполяцію всередині таблиці, коли відповідна таблиця є досить великою.

Будемо вважати, що вузол x_0 знаходиться в середині таблиці, та нумерація інших вузлів виконується відносно x_0 , з використанням як додатних, так і від'ємних індексів, тобто $x_i = x_0 + ih$, де $i = 0, \pm 1, \pm 2, \dots$. Тоді центральна частина таблиці скінченних різниць буде проіндексована так як показано в таблиці 6.3.

Всі скінченні різниці підкреслені в табл. 6.3, тобто скінченні різниці, що знаходяться в одному рядку з x_0 та y_0 та на пів рядка вище або нижче, називаються *центральними різницями*.

Таблиця 6.3. Центральні скінченні різниці

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$	$\Delta^6 y_i$...
...	...							
		...						
x_{-3}	y_{-3}		...					
		Δy_{-3}		...				
x_{-2}	y_{-2}		$\Delta^2 y_{-3}$...			
		Δy_{-2}		$\Delta^3 y_{-3}$...		
x_{-1}	y_{-1}		$\Delta^2 y_{-2}$		$\Delta^4 y_{-3}$...	
		Δy_{-1}		$\Delta^3 y_{-2}$		$\Delta^5 y_{-3}$		
x_0	y_0	Δy_{-1}	$\Delta^2 y_{-1}$	$\Delta^3 y_{-2}$	$\Delta^4 y_{-2}$	$\Delta^5 y_{-3}$	$\Delta^6 y_{-3}$...
		Δy_0		$\Delta^3 y_{-1}$		$\Delta^5 y_{-2}$		
x_1	y_1		$\Delta^2 y_0$		$\Delta^4 y_{-1}$...	
		Δy_1		$\Delta^3 y_0$...		
x_2	y_2		$\Delta^2 y_1$...			
		Δy_2		...				
x_3	y_3		...					
		...						
...	...							

Однією з інтерполяційних формул, що оперує центральними скінченними різницями є *перша інтерполяційна формула Гаусса* (для інтерполювання вперед):

$$f(x) \approx \bar{P}(x_0 + qh) = y_0 + q\Delta y_0 + \frac{q(q-1)}{2!} \Delta^2 y_{-1} + \frac{(q+1)q(q-1)}{3!} \Delta^3 y_{-1} + \frac{(q+1)q(q-1)(q-2)}{4!} \Delta^4 y_{-2} + \dots \quad (6.7)$$

Аналогічно, можна побудувати *другу інтерполяційну формулу Гаусса* (для інтерполювання назад):

$$f(x) \approx P(x_0 + qh) = y_0 + q\Delta y_{-1} + \frac{(q+1)q}{2!} \Delta^2 y_{-1} + \frac{(q+1)q(q-1)}{3!} \Delta^3 y_{-2} + \frac{(q+2)(q+1)q(q-1)}{4!} \Delta^4 y_{-2} + \dots \quad (6.8)$$

Ми бачимо, що в інтерполяційних формулах Гаусса використовуються або тільки нижні центральні різниці або тільки верхні. Природним було б для більш кращої інтерполяції знайти формулу в якій би враховувались як нижні так і верхні центральні скінченні різниці. Таку формулу можна отримати, як напівсуму першої та другої інтерполяційної формули Гаусса:

$$f(x) \approx P_s(x_0 + qh) = y_0 + q \frac{\Delta y_{-1} + \Delta y_0}{2} + \frac{q^2}{2!} \Delta^2 y_{-1} + \frac{q(q^2-1)}{3!} \cdot \frac{\Delta^3 y_{-2} + \Delta^3 y_{-1}}{2} + \frac{q^2(q^2-1)}{4!} \Delta^4 y_{-2} + \dots + \frac{q(q^2-1^2)(q^2-2^2)\dots(q^2-(n+1)^2)}{(2n-1)!} \cdot \frac{\Delta^{2n-1} y_{-n} + \Delta^{2n-1} y_{-(n-1)}}{2} + \frac{q^2(q^2-1^2)(q^2-2^2)\dots(q^2-(n+1)^2)}{(2n)!} \Delta^{2n} y_{-n}. \quad (6.9)$$

Інтерполяційна формула (6.9) називається *інтерполяційною формулою Стірлінга*.

Формула Стірлінга використовується для інтерполювання в середині таблиці при значеннях $|q|$ близьких до нуля, тобто коли значення x є близьким до одного з середніх вузлів, а саме коли $|q| < 0.25$.

Ще однією формулою, яка враховує як нижні центральні різниці так і верхні є *інтерполяційною формулою Бесселя*:

$$\begin{aligned}
 f(x) \approx P_B(x_0 + qh) = & \frac{y_0 + y_1}{2} + \left(q - \frac{1}{2}\right) \Delta y_0 + \frac{q(q-1)}{2!} \cdot \frac{\Delta^2 y_{-1} + \Delta^2 y_0}{2} + \\
 & + \frac{\left(q - \frac{1}{2}\right) q(q-1)}{3!} \cdot \Delta^3 y_{-1} + \frac{q(q^2 - 1^2)(q-2)}{4!} \cdot \frac{\Delta^2 y_{-1} + \Delta^2 y_0}{2} + \dots + \\
 & + \frac{q(q^2 - 1^2)(q^2 - 2^2) \dots (q^2 - n^2)(q+n-1)}{(2n)!} \cdot \frac{\Delta^{2n} y_{-n} + \Delta^{2n} y_{-(n-1)}}{2} + \\
 & + \frac{\left(q - \frac{1}{2}\right) q(q^2 - 1^2)(q^2 - 2^2) \dots (q^2 - n^2)(q+n-1)}{(2n+1)!} \cdot \Delta^{2n+1} y_{-n}.
 \end{aligned} \tag{6.10}$$

Дана формула також використовується для інтерполювання в середині таблиці при значеннях x близьких до середини між двома вузлами, тобто при значеннях q близьких до 0.5, а саме коли $q \in [0.25; 0.75]$.

Інтерполяційні формули Ньютона для нерівновіддалених вузлів

Для побудови інтерполяційних формул, що мають ті ж переваги перед інтерполяційною формулою Лагранжа, що й скінченнорізницеві формули та застосовуються до нерівновіддалених вузлів замість скінченних різниць використовують розділені різниці або їх ще називають різницевиими відношеннями.

Якщо визначені розділені різниці k -го порядку $f(x_i; x_{i+1}; \dots; x_{i+k})$, то $(k+1)$ -і різниці відношення визначаються через них таким чином:

$$f(x_{i-1}; x_i; \dots; x_{i+k}) = \frac{f(x_i; x_{i+1}; \dots; x_{i+k}) - f(x_{i-1}; x_i; \dots; x_{i+k-1})}{x_{i+k} - x_{i-1}}. \quad (6.11)$$

По аналогії зі скінченними різницями, аналізуючи таблицю розділених різниць (табл. 4.4) за порядком розділених різниць, які майже співпадають можна робити висновки про необхідну степінь многочлена для інтерполювання даної функції.

Таблиця 6.4. Таблиця розділених різниць

x_i	$f(x_i)$	$f(x_i; x_{i+1})$	$f(x_i; x_{i+1}; x_{i+2})$	$f(x_i; x_{i+1}; x_{i+2}; x_{i+3})$	$f(x_i; x_{i+1}; x_{i+2}; x_{i+3}; x_{i+4})$...
x_0	$f(x_0)$					
		$f(x_0; x_1)$				
x_1	$f(x_1)$		$f(x_0; x_1; x_2)$			
		$f(x_1; x_2)$		$f(x_0; x_1; x_2; x_3)$		
x_2	$f(x_2)$		$f(x_1; x_2; x_3)$		$f(x_0; x_1; x_2; x_3; x_4)$	
		$f(x_2; x_3)$		$f(x_1; x_2; x_3; x_4)$...
x_3	$f(x_3)$		$f(x_2; x_3; x_4)$		$f(x_1; x_2; x_3; x_4; x_5)$	
		$f(x_3; x_4)$		$f(x_2; x_3; x_4; x_5)$...
x_4	$f(x_4)$		$f(x_3; x_4; x_5)$		$f(x_2; x_3; x_4; x_5; x_6)$	
		$f(x_4; x_5)$		$f(x_3; x_4; x_5; x_6)$		⋮
x_5	$f(x_5)$		$f(x_4; x_5; x_6)$		⋮	
		$f(x_5; x_6)$		⋮		
x_6	$f(x_6)$		⋮			
		⋮				
⋮	⋮					

На основі міркувань аналогічних до побудови скінченно-різницевих інтерполяційних формул отримуємо першу *інтерполяційну формулу Ньютона для нерівновіддалених вузлів*:

$$\begin{aligned} f(x) \approx P_n(x) = & f(x_0) + f(x_0; x_1)(x - x_0) + \\ & + f(x_0; x_1; x_2)(x - x_0)(x - x_1) + \dots + \\ & + f(x_0; x_1; \dots; x_n)(x - x_0) \dots (x - x_{n-1}). \end{aligned} \quad (6.12)$$

Так само можна отримати і другу інтерполяційну формулу Ньютона для нерівновіддалених вузлів.

Завдання на лабораторну роботу

Розробити програму мовою програмування C# у середовищі розробки Visual Studio 2013 (або вище), яка буде працювати у віконному режимі та дозволяти виконувати наступне:

1. Будувати дві інтерполяційні формули, що задані за варіантом (*табл. 6.5, табл. 6.8*) для довільного набору вузлових точок (передбачити можливість введення вузлових точок з клавіатури).
2. Для кожної заданої за варіантом функції (*табл. 6.6, табл. 6.7*) будувати задану за варіантом інтерполяційну формулу (*табл. 6.5, табл. 6.8*) та обчислювати її значення в контрольних точках з точністю $\varepsilon \leq 10^{-3}$.

Виконати інтерполяцію заданих за варіантом функцій у будь-якому математичному пакті та знайти їх значення в контрольних точках двома способами:

- шляхом пошуку коефіцієнтів інтерполяційного многочлена використовуючи матрицю Вандермонда;
- за допомогою однієї зі спеціальних функцій.

Вимоги до оформлення звіту

Звіт має включати:

1. Постановку задачі за варіантом.
2. Математичне підґрунття для виконання даної лабораторної роботи (перелік формул, що були використані при розробленні програми на мові програмування C#).
3. Обрану сітку для аналітично-заданої функції та її значення у вузлових точках.
4. Значення функцій, заданих за варіантом, у контрольних точках:

Функція № _____

x_i	Точне значення функції y_i (для аналітично заданої за варіантом функції)	C#	Математичний пакет (вказати назву)	
		Інтерполяційна формула задана за варіантом (вказати назву)	З використанням матриці Вандермонда	З використанням спеціальної функції (вказати назву)

5. Висновки.

Варіанти завдань

Номер варіанту визначається за наступною таблицею:

№ за списком викладача	Варіант №	№ за списком викладача	Варіант №
1	7	14	11
2	3	15	21
3	15	16	2
4	14	17	13
5	16	18	5
6	24	19	12
7	4	20	25
8	23	21	20
9	17	22	6
10	9	23	8
11	18	24	19
12	22	25	1
13	10		

Таблиця 6.5. Варіанти завдань

Варіант №	Функція №	Інтерполяційна формула
1	10	7
	36	2
2	27	1
	15	4
3	19	3
	42	7
4	13	8
	35	5
5	18	1
	50	2
6	20	0
	33	4
7	16	1
	34	6
8	43	3
	21	1
9	45	3
	17	7
10	2	0
	28	2
11	29	4
	11	1
12	23	8
	26	6
13	3	0
	32	2
14	12	0
	37	6
15	22	0
	49	5
16	41	3
	8	7
17	44	6
	14	0
18	1	8
	30	5
19	38	3
	25	7
20	4	4
	47	8
21	6	1
	48	2
22	7	0
	46	5
23	39	8
	24	6
24	40	5
	5	7
25	9	8
	31	4

Таблиця 6.6. Перелік таблично заданих функцій

№	x	Y	Контрольні точки
1	-11.017	-5.694	
	-10.866	-5.454	
	-10.132	-1.263	
	-9.622	3.438	
	-9.197	7.394	
	-8.960	9.334	
	-8.318	12.758	
	-7.169	10.869	-10.239
	-5.753	2.393	-7.800
	-4.801	0.499	-7.789
	-4.035	2.107	-6.943
	-3.221	4.937	-5.080
	1.287	11.221	-1.637
	1.694	16.222	1.101
	2.191	26.691	1.246
	2.573	40.438	2.649
	2.740	48.810	3.606
	2.994	65.540	6.579
	3.183	81.932	7.766
	4.158	267.787	8.668
	5.322	1090.737	
	7.853	20212.857	
	10.580	416416.307	
	10.623	436054.740	
11.134	762499.282		
11.558	1209350.605		
12.412	3050130.945		
2	-20.061	127.098	
	-18.460	96.514	
	-18.119	95.938	
	-17.455	98.498	
	-15.386	93.513	
	-13.808	59.084	
	-13.282	48.788	
	-11.270	39.285	-18.935
	-10.951	40.464	-18.803
	-10.280	41.970	-18.643
	-6.300	6.931	-13.345
	-4.606	7.561	-12.055
	-3.902	7.902	-9.187
	-3.656	7.638	-8.333
	-1.280	0.178	-5.474
	-0.228	-0.205	-4.726
	-0.068	-0.066	-2.364
	1.428	0.883	3.217
	11.248	44.985	17.228
	12.692	66.286	19.033
	13.094	68.463	
	15.104	63.612	
	18.180	124.429	
	18.424	129.921	
19.127	140.341		
19.942	141.739		
21.238	134.857		

Продовження табл. 6.6

№	x	y	Контрольні точки
3	-16.983	-988.970	
	-15.499	-758.238	
	-14.721	-642.322	
	-13.610	-494.333	
	-12.345	-365.457	
	-10.205	-221.545	-15.545
	-9.862	-201.571	-15.391
	-9.787	-197.252	-12.491
	-8.853	-144.525	-12.485
	-2.363	-3.782	-10.506
	-0.696	0.572	-9.502
	0.102	-0.094	-2.526
	1.439	0.820	-1.879
	1.763	1.908	-0.648
	5.428	27.402	0.799
	5.563	29.381	2.686
	6.633	53.025	3.531
	7.057	66.848	6.775
	8.696	139.386	8.474
	9.000	154.701	10.146
	9.446	177.564	
	9.963	204.559	
	10.319	223.691	
	12.886	417.382	
12.906	419.451		
4	-8.801	8.737	
	-7.511	-1.450	
	-7.398	-2.152	
	-7.327	-2.556	
	-6.768	-4.542	
	-6.577	-4.701	
	-5.438	-1.166	
	-4.368	3.889	
	-3.638	4.967	-6.264
	-3.033	4.015	-1.765
	-3.011	3.958	-1.055
	-2.814	3.377	-0.642
	-1.916	0.356	1.022
	-1.421	-0.849	1.841
	-1.205	-1.179	2.327
	0.638	0.659	2.721
	0.910	1.253	3.337
	0.967	1.339	3.343
	1.166	1.532	4.181
	3.010	-1.750	4.322
	3.437	-2.345	5.111
	3.747	-2.329	7.309
	5.362	4.126	8.414
	5.702	5.956	
5.983	7.208		
6.252	8.052		
6.633	8.466		
7.223	7.047		
7.389	6.207		
8.522	-2.348		
8.950	-5.309		

Продовження табл. 6.6

№	x	Y	Контрольні точки
5	-8.331	-108.240	
	-8.244	-104.418	
	-6.489	-53.314	
	-5.929	-43.739	
	-5.664	-39.630	
	-4.047	-16.441	
	-3.294	-7.651	
	-2.763	-3.198	-5.168
	-2.482	-1.536	-4.682
	-1.042	0.673	-3.607
	-0.575	0.275	-3.483
	-0.501	0.215	-3.007
	-0.106	0.011	-1.056
	-0.061	0.004	-0.690
	0.341	0.122	-0.155
	0.357	0.134	-0.063
	2.811	5.354	0.941
	2.868	5.492	1.119
	3.214	6.407	2.989
	4.833	17.786	3.000
5.247	24.374	5.847	
5.462	28.585	8.322	
5.468	28.717		
6.243	48.427		
6.964	71.930		
8.498	129.550		
8.686	136.898		
9.006	149.734		
9.076	152.628		
6	-13.648	-853.845	
	-13.009	-745.547	
	-12.595	-678.540	
	-12.292	-630.918	
	-12.101	-601.405	
	-11.504	-513.132	
	-10.837	-422.536	
	-9.512	-277.368	
	-9.395	-267.042	
	-8.964	-232.072	-11.454
	-7.840	-160.742	-10.239
	-7.454	-140.959	-9.529
	-5.591	-62.544	-8.832
	-5.550	-61.094	-7.970
	-5.298	-52.514	-5.455
	-4.840	-38.421	-0.990
	-3.680	-13.453	0.153
	-2.657	-3.901	0.825
	-2.297	-2.515	1.060
	-2.058	-1.943	3.396
0.957	0.843	6.310	
1.220	1.024	7.239	
1.433	1.178		
2.633	3.783		
2.934	5.549		
3.025	6.226		
3.200	7.726		
4.691	34.297		
4.898	40.064		
5.180	48.661		
7.920	165.059		

Продовження табл. 6.6

№	x	y	Контрольні точки
7	-14.942	4.629	
	-13.031	4.875	
	-11.994	5.865	
	-11.972	5.883	
	-11.682	6.097	
	-11.484	6.207	
	-9.345	5.251	
	-9.110	5.022	
	-7.849	4.341	
	-6.583	5.063	-12.146
	-6.537	5.108	-11.910
	-5.079	6.333	-11.122
	-0.940	5.138	-10.717
	-0.246	5.962	-6.701
	1.297	8.196	-6.418
	2.235	9.166	-3.593
	2.356	9.276	-2.016
	4.671	14.655	1.553
	4.724	14.934	4.123
	4.786	15.267	4.474
4.941	16.177	5.004	
5.076	17.041	6.625	
5.531	20.525		
5.658	21.667		
6.477	31.015		
6.895	37.316		
7.557	50.017		
8.139	64.796		
8.679	82.681		
8.977	94.730		
8.998	95.640		
8	-12.520	770274.489	
	-12.346	708319.300	
	-11.449	450392.411	
	-11.264	408603.647	
	-10.168	221096.314	
	-8.740	89152.926	
	-6.088	10199.865	-11.245
	-3.850	666.821	-9.976
	-3.188	225.266	-9.544
	-3.025	168.338	-4.614
	-2.451	58.077	-4.426
	-1.878	23.146	0.498
	1.359	17.559	1.197
	2.160	36.446	1.577
	3.335	290.353	3.161
	3.732	554.870	5.417
	3.885	702.109	6.277
	4.549	1785.518	8.526
	5.103	3545.063	9.395
	6.497	15054.053	9.502
7.074	25077.165	10.563	
7.210	28101.220	10.788	
7.283	29853.353	11.624	
7.488	35270.261		
8.782	91780.084		
10.170	221292.740		
12.441	741698.313		
14.075	1554839.076		
14.208	1645386.017		

Продовження табл. 6.6

№	x	Y	Контрольні точки
9	-14.630	-95715.604	
	-14.356	-87100.338	
	-12.839	-49825.856	
	-12.446	-42651.381	
	-12.125	-37434.580	
	-11.006	-23078.380	
	-10.093	-14961.202	
	-9.457	-10797.809	-13.349
	-7.890	-4353.541	-8.241
	-6.879	-2189.230	-7.584
	-6.100	-1200.561	-7.284
	-5.224	-553.074	-6.101
	-2.472	-4.319	-5.380
	-2.043	4.054	-5.170
	-1.895	5.627	-4.676
	-1.894	5.636	-4.145
	-1.295	8.047	0.456
	-0.655	7.702	2.025
	-0.592	7.641	2.349
	1.392	9.438	2.499
	3.540	85.411	2.955
	4.614	301.508	5.032
	5.536	746.454	9.719
	5.993	1109.628	9.784
6.722	1971.170		
7.680	3832.642		
9.424	10624.034		
10.522	18420.574		
10.653	19594.451		
11.108	24153.372		
12.475	43161.597		
10	-3.707	20.690	
	-3.558	21.166	
	-2.468	24.278	
	-1.519	25.090	-2.404
	0.184	24.538	-0.972
	1.317	25.047	-0.184
	2.672	23.839	-0.121
	3.434	21.572	1.958
	3.600	21.030	2.285
	4.047	19.721	3.123
	4.380	19.054	3.234
	4.490	18.921	3.676
	4.932	18.925	4.378
	5.050	19.087	4.599
	5.583	20.688	5.844
	5.854	21.993	5.909
	6.380	25.130	6.362
	6.937	28.534	6.730
	8.564	29.365	9.868
	9.840	18.639	12.086
	12.310	21.367	
	12.331	21.618	
	12.684	26.002	

Продовження табл. 6.6

№	x	y	Контрольні точки	
11	-13.008	41.133		
	-12.623	40.662		
	-11.241	51.561		
	-10.772	56.684		
	-9.683	62.938		
	-9.258	62.283		
	-8.425	57.035	-12.169	
	-8.189	55.072	-9.528	
	-7.828	52.116	-9.314	
	-6.545	46.740	-8.325	
	-5.916	48.157	-7.992	
	-5.057	52.555	-6.598	
	-4.721	54.279	-6.257	
	-4.092	56.513	-6.164	
	-4.067	56.565	-5.814	
	-3.794	56.943	-5.119	
	-3.511	56.956	-3.895	
	-2.029	53.322	-3.009	
	-1.526	52.254	-1.872	
	-1.480	52.192	0.115	
	-1.381	52.079	0.863	
	-1.173	51.945		
	0.207	53.730		
	0.214	53.743		
	0.248	53.807		
	0.389	54.060		
	2.649	51.459		
	12	-11.898	23.258	
		-11.600	24.472	
		-11.442	24.979	
		-11.328	25.280	
		-11.025	25.813	
		-10.144	25.150	
		-9.935	24.579	
		-9.432	22.775	
		-9.406	22.670	
-8.482		18.960		
-7.893		17.209		
-7.336		16.343	-9.967	
-7.112		16.211	-9.604	
-7.016		16.187	-9.016	
-4.724		16.339	-8.692	
-4.075		14.873	-6.692	
-3.577		13.265	-6.662	
-3.437		12.777	-4.445	
-3.365		12.524	-3.816	
-1.997		9.019	-3.142	
-1.695		8.912	-0.818	
-1.644		8.926	-0.000	
-0.617		10.981	0.471	
-0.501		11.387	3.267	
-0.326		12.036	3.312	
0.146		13.904	5.503	
0.276		14.412		
0.431		15.002		
0.470		15.145		
1.534		17.657		
2.353		17.114		
3.232		14.581		
3.430		13.890		
4.842		10.137		
5.605		9.796		
6.022		10.000		
6.181	10.108			

Продовження табл. 6.6

№	x	y	Контрольні точки
13	-8.809	7.932	
	-8.769	8.380	
	-8.196	11.151	
	-7.413	7.019	
	-7.096	8.005	
	-6.885	9.779	
	-5.367	7.788	
	-4.861	8.883	
	-4.658	10.469	
	-4.249	12.471	
	-3.423	6.439	-7.442
	-2.784	6.117	-6.452
	-2.496	8.239	-5.213
	-2.244	9.550	-3.075
	-1.869	8.754	-2.222
	-1.523	5.628	-2.027
	-1.365	4.188	-1.625
	-0.978	2.699	-1.215
	-0.890	2.907	-1.039
	-0.539	5.280	-0.449
	0.110	8.165	1.597
	0.190	7.858	3.650
	0.570	5.011	4.282
	1.822	8.412	
	2.204	9.637	
	2.363	9.075	
	2.776	6.169	
3.529	7.388		
4.209	12.465		
4.384	12.180		
4.548	11.283		
4.755	9.694		
4.992	8.059		
14	2.211	3.755	
	2.228	3.745	
	2.306	3.701	
	3.686	4.977	
	4.054	6.513	
	4.200	7.291	
	4.489	9.069	
	5.039	13.085	3.322
	5.637	17.571	4.376
	5.700	18.000	5.630
	6.394	21.538	9.011
	7.350	21.533	11.485
	11.072	30.782	12.175
	11.565	37.868	12.428
	11.787	40.788	13.922
	12.144	44.721	15.205
	12.452	47.097	15.260
	13.182	48.077	19.718
	14.168	40.447	21.245
	14.389	38.106	22.015
	15.763	31.050	
	15.831	31.333	
	16.034	32.618	
	17.760	62.628	
	21.303	52.033	
	22.334	51.658	
	23.865	86.156	

Продовження табл. 6.6

№	x	y	Контрольні точки
15	2.607	3.576	
	3.351	4.095	
	4.432	8.695	
	5.015	12.903	
	6.159	20.628	
	6.888	22.279	
	6.965	22.247	
	7.014	22.207	3.532
	7.577	20.738	3.998
	7.611	20.601	5.437
	8.152	18.031	6.296
	8.486	16.453	7.480
	8.673	15.720	7.700
	8.990	14.911	8.484
	9.062	14.823	8.660
	9.235	14.781	8.663
	9.781	16.458	9.933
	10.303	20.777	10.160
	10.563	23.796	10.477
	10.583	24.048	11.601
	10.935	28.812	
	11.535	37.460	
	11.941	42.617	
	12.007	43.345	
	12.814	48.427	
	13.884	43.319	
	14.726	34.753	
16	2.109	34.498	
	2.510	47.674	
	3.105	84.673	
	3.472	126.054	
	3.801	183.089	
	4.310	332.235	
	4.530	430.991	
	5.762	1844.222	2.736
	5.855	2056.552	4.483
	6.448	4088.466	5.351
	6.755	5815.929	5.913
	6.771	5920.709	7.953
	6.987	7581.716	8.438
	7.261	10361.183	8.901
	7.443	12733.556	9.145
	8.006	24030.026	9.175
	9.166	87712.581	10.065
	9.260	97366.477	11.016
	10.676	462254.684	11.391
	10.748	500019.705	13.073
	11.164	787729.455	
	12.091	2154222.335	
	12.849	4886724.521	
	13.115	6510612.811	
	13.683	11990136.653	
	13.874	14706106.102	
	14.379	25265827.982	

Продовження табл. 6.6

№	x	y	Контрольні точки
17	-4.028	27.796	
	-2.958	28.075	
	-0.713	24.432	
	-0.620	24.483	
	0.078	25.484	
	0.107	25.546	
	1.863	36.780	-2.494
	1.973	38.739	-1.821
	2.935	77.701	-1.781
	3.665	165.218	-1.597
	3.673	166.834	-0.946
	5.166	933.128	1.995
	5.566	1484.031	4.605
	6.074	2670.490	5.096
	8.794	58041.093	6.858
	9.259	97274.709	7.931
	9.432	117787.816	8.197
	10.309	309287.140	9.280
	10.488	376536.216	10.026
	10.915	599972.987	
	11.650	1335726.588	
	12.243	2540680.150	
	13.082	6282048.006	
	13.911	15309358.576	
14.291	23002391.250		
18	-5.377	11.063	
	-5.199	10.696	
	-4.164	11.702	
	-3.768	13.023	
	-2.618	16.440	
	-2.612	16.448	
	-1.089	15.920	-4.075
	-0.049	15.277	-2.166
	0.102	15.444	-1.542
	0.657	16.992	-1.226
	1.785	27.709	-0.456
	2.538	48.872	-0.324
	3.488	128.241	3.533
	5.213	968.618	4.663
	5.402	1210.135	5.057
	6.170	2965.699	5.535
	6.440	4049.704	6.981
	6.519	4435.108	8.516
	6.889	6782.583	9.417
	7.901	21353.516	10.079
	8.144	28056.714	10.204
	10.822	542555.964	11.590
	11.331	945062.349	11.609
	12.757	4427884.826	
12.963	5523941.931		
13.314	8061056.431		
13.676	11895793.517		
13.801	13595838.004		
14.072	18183546.817		

Продовження табл. 6.6

№	x	y	Контрольні точки
19	-14.989	28.599	
	-13.791	12.640	
	-13.021	5.627	
	-12.542	4.786	
	-12.002	7.187	
	-11.753	9.254	
	-11.595	10.779	
	-10.826	19.148	
	-10.603	21.381	
	-7.992	18.429	-11.915
	-6.047	11.448	-10.211
	-4.110	19.671	-10.159
	-4.106	19.681	-8.875
	-3.683	20.507	-5.828
	-3.391	20.645	-5.243
	-3.251	20.595	-4.305
	-2.668	19.768	-4.123
	-2.179	18.682	-3.361
	-2.063	18.428	-3.303
	-1.524	17.471	-2.755
	-0.070	18.188	-1.191
	0.327	19.021	-0.052
	1.472	21.827	3.226
	2.379	26.401	3.661
	2.939	33.339	
	3.163	37.801	
	3.716	55.291	
	3.964	67.272	
	4.040	71.614	
	4.338	92.317	
	4.454	102.173	
	5.451	254.080	
	6.756	882.586	
6.777	900.889		
6.779	902.393		

Продовження табл. 6.6

№	x	y	Контрольні точки
20	-9.319	-206.546	
	-9.166	-215.645	
	-8.594	-236.654	
	-8.536	-237.463	
	-8.288	-238.011	
	-7.784	-225.579	
	-7.146	-190.626	
	-6.384	-140.269	
	-6.165	-127.277	
	-5.506	-98.611	
	-5.205	-88.650	-7.665
	-2.021	-45.542	-7.274
	-1.865	-41.825	-6.581
	-1.011	-20.935	-6.494
	-0.607	-11.589	-6.288
	0.762	18.470	-3.923
	2.043	49.288	-1.416
	2.611	61.210	-1.393
	3.861	74.742	-1.166
	5.094	86.898	-0.551
	5.490	97.992	2.536
	6.766	168.031	3.765
	7.501	215.333	6.893
	9.674	185.939	10.139
	9.922	167.380	15.964
	11.210	119.081	
	12.740	302.542	
	13.061	362.367	
	13.191	386.040	
	13.232	393.205	
15.142	448.805		
15.229	434.680		
15.357	411.636		
15.594	363.198		
16.579	146.413		
21	-4.793	-140.383	
	-4.031	-110.378	
	-3.768	-99.286	
	-3.383	-83.750	
	-2.904	-66.756	
	-2.817	-64.001	
	-2.442	-53.224	
	0.760	16.335	
	0.879	18.861	
	1.752	36.956	-2.836
	1.762	37.174	-2.829
	2.930	67.260	-0.818
	4.705	137.312	-0.812
	5.522	158.385	1.898
	5.675	159.940	1.919
	5.991	160.521	1.945
	6.133	159.687	2.267
	7.069	144.009	5.810
	7.157	142.427	7.281
	8.358	152.025	8.749
	8.851	183.542	9.044
	9.145	210.834	10.354
	9.594	261.688	
	9.825	290.253	
	9.926	302.876	
	10.545	373.699	
	10.604	379.278	
	10.628	381.480	
	11.516	418.441	
	12.810	302.551	
13.103	262.426		

Продовження табл. 6.6

№	x	y	Контрольні точки
22	-5.808	-7.245	
	-4.072	0.062	
	-3.781	1.180	
	-3.645	1.604	
	-3.488	2.010	
	-3.407	2.182	
	-2.141	2.152	
	-1.001	1.089	
	0.327	3.108	
	0.596	3.783	
	1.518	5.523	-3.091
	1.734	5.692	-0.543
	2.330	5.674	-0.422
	2.410	5.631	-0.021
	5.672	16.411	0.770
	5.970	18.630	0.929
	6.746	22.821	1.889
	6.840	23.089	3.678
	6.845	23.102	3.764
	7.456	23.286	10.625
	8.503	18.406	10.683
	8.603	17.782	12.539
	9.988	13.565	14.853
	10.657	17.463	14.953
	12.313	38.231	20.292
	12.515	40.286	20.890
	13.542	43.403	23.016
	13.976	40.716	
	14.545	34.622	
	15.232	26.445	
17.259	32.462		
17.607	38.931		
18.022	47.243		
18.102	48.817		
18.531	56.535		
22.880	32.598		
23.862	51.997		

Продовження табл. 6.6

№	x	y	Контрольні точки
23	-2.987	-16.593	
	-2.966	-16.449	
	-2.259	-12.305	
	-2.132	-11.656	
	-1.808	-10.039	
	-1.614	-9.053	
	-1.597	-8.970	
	-1.575	-8.854	
	-1.251	-7.089	
	-1.108	-6.250	
	-0.793	-4.243	-2.477
	-0.788	-4.209	-1.195
	-0.564	-2.641	-1.014
	-0.427	-1.626	-0.905
	-0.399	-1.410	-0.463
	-0.319	-0.800	-0.449
	-0.024	1.556	0.165
	0.604	6.821	0.533
	0.760	8.109	3.376
	0.902	9.257	3.616
	0.939	9.553	4.374
	1.036	10.320	5.355
	1.249	11.929	5.529
	1.294	12.261	
	1.827	15.796	
	1.951	16.524	
	2.238	18.097	
	2.989	21.863	
	4.064	28.948	
	4.851	37.111	
	5.334	43.209	
	6.011	51.996	
	6.394	56.453	
24	-4.834	176.068	
	-4.577	155.109	
	-4.032	116.562	
	-3.433	88.770	
	-2.905	66.289	
	-2.487	47.133	
	-1.631	20.321	
	-0.635	5.292	
	-0.250	2.297	
	-0.114	1.955	-3.404
	0.903	7.431	-2.953
	1.676	22.427	-2.947
	2.336	44.347	-1.685
	2.464	48.556	-0.805
	2.656	54.695	-0.416
	3.528	90.726	0.592
	3.575	93.467	1.865
	4.049	124.942	3.516
	4.232	137.476	4.508
	4.793	171.571	4.880
	4.902	177.749	6.405
	5.629	230.282	6.797
	5.974	264.946	8.109
	6.414	310.191	8.845
	6.429	311.548	9.650
	6.673	333.599	10.838
	6.805	344.616	
	7.230	380.098	
	7.710	431.980	
	7.946	463.637	
	7.961	465.688	
	9.232	622.091	
	10.410	804.699	
11.086	904.851		
11.515	963.700		

Продовження табл. 6.6

№	x	y	Контрольні точки
25	-0.764	4.178	
	-0.508	2.766	
	-0.504	2.747	
	-0.089	1.683	
	0.108	1.669	
	0.270	1.841	
	0.616	2.884	
	0.790	3.776	
	1.203	6.912	
	1.262	7.477	
	1.322	8.086	
	1.411	9.035	0.055
	2.279	21.939	0.565
	2.476	25.762	0.997
	2.931	35.917	1.092
	3.075	39.512	1.198
	3.333	46.417	1.384
	3.366	47.329	1.965
	3.612	54.545	2.567
	3.889	63.279	2.575
	4.368	80.024	3.119
	4.481	84.300	3.255
	4.777	96.027	3.296
	4.940	102.841	3.307
	5.074	108.634	5.044
	5.240	116.040	5.077
	5.333	120.327	6.271
	5.393	123.123	6.477
	6.156	161.645	
	6.621	187.680	
	7.030	212.150	
	7.414	236.493	
	7.572	246.913	
7.841	265.083		
7.842	265.197		
8.173	288.452		
8.636	322.566		
8.810	335.846		
8.995	350.278		

Таблиця 6.7. Перелік аналітично заданих функцій

№	Функція	Контрольні точки
26	$f(x) = \sqrt{x^3} + \sin x$	2.556 3.594 4.725 4.866 5.199 7.192 7.382 8.735 9.406 9.449 9.714 9.995 11.514 12.213 13.027 13.038 14.015
27	$f(x) = \sqrt{x^3} + \cos 2x$	2.050 2.310 2.464 6.726 6.812 7.665 8.912 11.286 11.877 13.702 13.935 14.263 15.093 15.482 15.674 16.044 17.798
28	$f(x) = \ln x^3 + \cos x + x$	8.420 8.677 10.250 11.213 11.288 11.942 12.288 12.463 12.834 12.913 14.207 16.404 16.639 16.942 18.683 18.795 19.071

Продовження табл. 6.7

№	Функція	Контрольні точки
29	$f(x) = \log_{10} x^4 + \cos 3x + x^2$	9.711 10.031 10.048 10.873 12.080 12.120 12.158 13.418 14.973 19.321 20.525 20.890 22.416 23.730 24.195 26.906 29.050
30	$f(x) = e^{x^2} + x^3 \cos 3x + x$	1.514 1.544 1.571 1.589 1.716 1.733 1.818 1.821 1.885 1.913 1.921 1.937 1.940 1.978 2.040 2.186 2.380
31	$f(x) = e^x + x^2 \cos x + x^2$	-2.338 -0.541 0.143 0.162 0.281 0.501 0.583 1.286 1.623 1.789 1.993 2.081 2.517 3.087 3.162 3.305 3.794

Продовження табл. 6.7

№	Функція	Контрольні точки
32	$f(x) = e^x + x^2 \ln x + x$	4.977 4.979 5.869 6.306 8.452 8.821 9.257 9.832 10.082 10.240 11.620 11.635 11.778 11.808 12.495 13.904 14.234
33	$f(x) = e^x + x \log_{10} x + x$	5.269 5.561 6.462 6.787 9.220 9.421 9.747 9.783 9.808 12.425 12.604 13.512 14.224
34	$f(x) = \cos x + x \log_{10} x^3 + x^2$	4.522 5.469 6.963 7.038 7.408 7.955 8.409 8.414 9.316 9.769 13.064 13.101 14.454

Продовження табл. 6.7

№	Функція	Контрольні точки
35	$f(x) = \sin x + x \log_2 x + x^2$	1.999 2.604 3.862 4.174 4.383 6.263 6.566 8.023 8.453 9.109 9.291 9.405 9.631 10.685 11.276 11.510 13.560
36	$f(x) = x^3 \sin x + \log_{10} x + x$	3.258 3.480 4.534 4.956 5.363 7.112 8.472 8.709 9.049 10.119 10.323 11.326 11.872
37	$f(x) = x^3 \cos x + e^x + x$	1.730 3.072 4.381 5.055 5.931 8.305 8.473 9.222 9.865 11.840 12.756 13.736 13.992
38	$f(x) = x^2 \sin x + xe^x + x$	-2.100 -1.911 -1.672 -1.464 -1.235 -0.988 -0.890 -0.629 -0.448 -0.425 -0.125 -0.067 0.299

Продовження табл. 6.7

№	Функція	Контрольні точки
39	$f(x) = x^2 \cos x + xe^x + \sqrt{x^3}$	1.772 1.905 2.673 2.746 3.110 3.249 3.271 3.451 4.401 4.477 4.714 5.140 5.557 5.562 5.760
40	$f(x) = x \sin x + e^x + x\sqrt{x^4}$	-1.532 -0.692 -0.587 -0.530 0.034 0.200 0.303 0.988 1.070 1.499 1.587 1.647 1.745 1.989 2.148
41	$f(x) = x \sin x + xe^x + x\sqrt{x^3}$	3.563 3.782 4.744 5.445 5.592 5.826 5.907 5.925 6.353 6.499 6.657 6.683 6.917 6.933 7.670
42	$f(x) = x \sin x + x^2 e^x + \log_2 x^3$	3.821 3.828 3.833 4.012 4.080 4.516 4.837 4.841 4.903 5.221 5.494 6.978 7.205

Продовження табл. 6.7

№	Функція	Контрольні точки
43	$f(x) = x \sin x + x^2 e^x + shx$	4.988 5.468 5.891 6.176 6.332 6.631 6.686 6.879 7.185 7.505 7.740 7.870 8.046
44	$f(x) = \sin(x^2) + e^x \ln x + ch x$	0.280 1.257 1.409 2.048 2.050 3.439 4.507 4.913 4.986 5.517 5.815 6.330 7.762 7.789 7.885
45	$f(x) = \sin x + \cos x \cdot \ln x + ch x$	-1.910 -1.907 -0.757 -0.527 -0.009 1.191 1.469 2.239 3.058 3.133 3.399 4.314 4.350
46	$f(x) = \log_2 x + \sin x \cdot \ln x + ch(\log_{10} x) + x$	-4.166 -3.216 -2.818 -1.643 -1.436 -1.262 -1.118 -0.934 -0.221 0.075 0.445 0.595 0.615 2.619 2.749 3.266 3.900 4.619 5.232

Продовження табл. 6.7

№	Функція	Контрольні точки
47	$f(x) = x \ln x + e^x \cdot \log_{10}(x) + \operatorname{sh} \frac{x}{2} + x$	-3.091 -3.003 -2.637 -2.453 -2.445 -1.936 -1.819 0.091 0.098 0.120 0.575 0.867 0.922 1.085 1.769 1.956 2.219
48	$f(x) = \sqrt{ x } + \log_{10}\left(\frac{x}{5}\right) + e^x \cdot \sin\left(\frac{x}{2}\right) + x$	-13.386 -11.300 -11.224 -10.779 -10.370 -9.970 -8.197 -8.137 -7.520 -5.441 -3.688 -3.647 -3.546 -3.115 -0.579 0.920 0.966 2.077 2.745
49	$f(x) = \sqrt[3]{x} + e^x \ln \frac{x}{3} + \cos \sqrt{ x } - x \cdot \operatorname{sh}(2x)$	-15.588 -11.288 -9.757 -7.703 -7.402 -6.339 -3.509 -3.049 -1.033 1.733 2.492 3.790 6.053 9.435 9.548 10.451 11.066

Продовження табл. 6.7

№	Функція	Контрольні точки
50	$f(x) = \sqrt[5]{x} + e^x \cdot \log_{10} x^2 + \cos x + x \cdot \operatorname{ch}(5\sqrt{ x })$	-13.533 -11.521 -8.718 -7.046 -6.166 -5.797 -4.147 -3.699 -2.883 -1.451 0.115 3.455 3.926 5.877 6.887 7.485 9.133 9.295 10.354

Таблиця 6.8. Перелік інтерполяційних формул

№	Метод
0	Інтерполяційний многочлен Лагранжа
1	Схема Ейткена
2	Перша інтерполяційна формула Ньютона
3	Друга інтерполяційна формула Ньютона
4	Перша інтерполяційна формула Гаусса
5	Друга інтерполяційна формула Гаусса
6	Інтерполяційна формула Стірлінга та інтерполяційна формула Бесселя
7	Перша інтерполяційна формула Ньютона для нерівновіддалених вузлів
8	Друга інтерполяційна формула Ньютона для нерівновіддалених вузлів

Питання для самоперевірки

1. Дайте визначення термінів апроксимація, інтерполяція, екстраполяція.
2. Як будується визначник Вандермонда та яким чином його можна використати при інтерполяції?
3. В чому полягає ідея побудови інтерполяційного многочлена Лагранжа?
4. Еквівалентом якої інтерполяційної формули є схема Ейткена? Відповідь обґрунтувати.
5. Яка перевага схеми Ейткена порівняно з інтерполяційною формулою Лагранжа?
6. Чи можливо за таблично заданою функцією побудувати кілька різних інтерполяційних многочленів Лагранжа? Відповідь обґрунтувати.
7. Пояснити зміст термінів білінійна, біквадратична та бікубічна інтерполяція?
8. Дайте визначення скінченної різниці довільного порядку.
9. Які основні властивості скінченних різниць?
10. Який взаємозв'язок наявний між скінченними різницями та похідними?
11. Яка особливість скінченних різниць n -го порядку многочлена n -го степеня? Чому вони дорівнюють? Чому дорівнюють скінченні різниці порядку більше n для многочлена n -го степеня?
12. Як обрати оптимальну степінь інтерполяційного многочлена побудувавши таблицю скінченних різниць?
13. Які інтерполяційні формули слід обрати для знаходження значення таблично заданої функції в контрольній точці, яка знаходиться на початку, в середині або в кінці таблиці?

14. Яка перевага скінченно-різницевих інтерполяційних формул над інтерполяційною формулою Лагранжа?
15. Які інтерполяційні формули використовуються тільки для рівновіддалених вузлів? Відповідь обґрунтувати.
16. Дайте визначення центральних скінченних різниць.
17. На основі яких інтерполяційних формул та яким чином отримуються формули Стірлінга та Бесселя?
18. Чи є формули Стірлінга та Бесселя інтерполяційними в класичному розумінні цього терміну?
19. При яких значеннях q використовуються скінченно-різницеві інтерполяційні формули Ньютона, Гаусса, Бесселя та Стірлінга?
20. Які саме скінченні різниці використовуються в першій інтерполяційній формулі Гаусса?
21. Які саме скінченні різниці використовуються в другій інтерполяційній формулі Гаусса?
22. Яку кількість вузлових точок доцільно брати в формулах Стірлінга та Бесселя? Відповідь обґрунтувати.
23. Що спільного між скінченно-різницевими інтерполяційними многочленами та інтерполяційним многочленом Лагранжа?
24. Дайте визначення розділеної різниці довільного порядку.
25. Яка особливість розділених різниць n -го порядку многочлена n -го степеня?
26. Побудувати другу інтерполяційну формулу Ньютона для нерівновіддалених вузлів.
27. Які інтерполяційні формули використовуються для нерівновіддалених вузлів? Відповідь обґрунтувати.

Рекомендована література

1. William H. Press Numerical Recipes in C. The Art of Scientific Computing / William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery // Cambridge University Press, 2002. – 1018 p.
2. Walter Gautschi Numerical Analysis [Електронний ресурс], 2012. Режим доступу: http://www.ikiu.ac.ir/public-files/profiles/items/090ad_1410599906.pdf
3. McDonough J. M. Computational Numerical Analysis [Електронний ресурс], 2007. Режим доступу: <http://web.engr.uky.edu/~acfd/egr537-lctrs.pdf>