



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
імені ВОЛОДИМИРА ДАЛЯ

**С.О.Кондратов, В.Г.Іванов**

## **ЧИСЕЛЬНІ МЕТОДИ**

**Частина 1. Чисельні методи алгебри. Методи наближень**  
(для студентів спеціальностей  
121 «Інженерія програмного забезпечення»,  
126 «Інформаційні технології та системи»)

**Навчальний посібник**

**(Електронне видання)**

**ЗАТВЕРДЖЕНО**  
на засіданні кафедри інформаційних  
технологій та програмування  
Протокол № 11 від 17.06.2025

Київ 2025

УДК 519.6

Навчальний посібник з дисципліни «Чисельні методи. Частина 1. Чисельні методи алгебри. Методи наближень» для здобувачів першого (бакалаврського) рівня освіти за спеціальностей 121 «Інженерія програмного забезпечення», 126 «Інформаційні системи та технології»(Електронне видання)/ Уклад.: Кондратов С.О. Іванов В.Г. - Київ: Вид-во СНУ ім. В. Даля, 2025. – 262 с.

У першій частині розглянуто базові чисельні методи для розв'язання задач алгебри і теорії наближень. Посібник містить матеріали 7 лекцій, разом із завданням для самостійної роботи і контрольними запитаннями з кожної теми. Матеріали лекцій ілюстровані прикладами розв'язання задач з використанням відкритих комп'ютерних систем: OOo Calc, Scilab.

Розглянуто і схвалено на засіданні кафедри ІТП

Протокол № ..11.. від “...17...”,,,червня....2025 р.

Зав.кафедри ІТП.....Захожай О.І.

## ЗМІСТ

ЛЕКЦІЯ 1. ОСОБЛИВОСТІ ЧИСЕЛЬНИХ МЕТОДІВ І ЕЛЕМЕНТИ ТЕОРІЇ ПОГРІШНОСТЕЙ.....	5
Мета лекції.....	5
План лекції.....	5
1.1 Особливості й етапи чисельного рішення задач на ЕОМ.....	6
1.1.1 Етапи розв'язання задач.....	6
1.1.2 Принципи укладання алгоритмів обчислювальних задач.....	7
1.1.3 Особливості чисельного розв'язання математичних задач.....	10
1.2 Елементи теорії погрішностей.....	12
1.2.1 Класифікація погрішностей.....	12
1.2.2 Абсолютна й відносна погрішності.....	13
1.2.3 Запис наближених чисел. Правила округлення.....	14
1.2.4. Погрішності функцій.....	16
1.2.5 Погрішності при складних обчисленнях.....	19
1.2.6 Погрішності обчислень на комп'ютерах.....	20
1.2.7 Комп'ютерне моделювання погрішностей.....	22
Контрольні запитання.....	27
Завдання для самостійної роботи.....	28
Завдання 1.1.....	28
Завдання 1.2.....	29
ЛЕКЦІЯ 2 СИСТЕМИ ЛІНІЙНИХ АЛГЕБРАЇЧНИХ РІВНЯНЬ: РОЗВ'ЯЗАННЯ МЕТОДОМ ВИКЛЮЧЕННЯ, СТІЙКІСТЬ РОЗВ'ЯЗКІВ.....	32
Мета лекції:.....	32
План лекції:.....	32
2.1 Системи лінійних рівнянь алгебраїчних рівнянь і задачі, що приводять до них.....	33
2.2 Розв'язання системи лінійних алгебраїчних рівнянь. Метод Гаусса.....	34
2.3 Обчислення визначника й оберненої матриці методом Жордана-Гаусса.....	41
2.4 Норми і сингулярні числа матриць.....	42
2.5 Оцінка погрішності розв'язання СЛАР.....	45
2.6 Комп'ютерне дослідження і розв'язання СЛАР.....	47
Контрольні запитання.....	52
Завдання для самостійної роботи.....	54
ЛЕКЦІЯ 3. СЛАР. ІТЕРАЦІЙНІ І СПЕЦІАЛЬНІ МЕТОДИ РОЗВ'ЯЗАННЯ.....	61
Мета лекції.....	61
План лекції.....	61
3.1 Розв'язання розріджених систем. Метод прогону.....	62
3.2 Розкладання матриць на співмножники. Розкладання Холецького.....	66
3.3 Розкладання методом квадратного кореня.....	71
3.4 Теоретичні основи ітераційних методів розв'язання СЛАР.....	76
3.5 Алгоритми ітераційних методів.....	80
3.5.1 Метод Якобі.....	80
3.5.2 Метод Гаусса-Зейделя.....	83
Контрольні запитання.....	85
Завдання для самостійної роботи.....	86
Завдання 3.1.....	86
Завдання 3.2.....	89
Завдання 3.3.....	89
ЛЕКЦІЯ 4 РОЗВ'ЯЗАННЯ ПЕРЕВИЗНАЧЕНИХ Й НЕДОВИЗНАЧЕНИХ СИСТЕМ. РОЗВ'ЯЗАННЯ ПРОБЛЕМИ ВЛАСТИВИХ ЗНАЧЕНЬ.....	96
Мета лекції.....	96
План лекції.....	96
4.1 Розв'язання перевизначених систем. Метод найменших квадратів.....	97
4.2 Кістякове розкладання прямокутних матриць.....	100
4.3 Псевдообернена матриця Мура-Пенроуза.....	103
4.4 Розв'язання систем недовизначених алгебраїчних рівнянь.....	106
4.5 Поняття про коректно й некоректно поставлені задачі.....	108
4.6 Обчислення властивих значень і властивих векторів матриць.....	119
Контрольні запитання.....	125
Завдання для самостійної роботи.....	127
Завдання 4.1.....	127

ЛЕКЦІЯ 5. МЕТОДИ РОЗВ'ЯЗАННЯ НЕЛІНІЙНИХ РІВНЯНЬ І СИСТЕМ НЕЛІНІЙНИХ РІВНЯНЬ .....	140
Мета лекції .....	140
План лекції .....	140
5.1 Загальні поняття. Відокремлення коренів .....	141
5.2 Метод половинного ділення .....	145
5.3 Метод хорд .....	149
5.4 Метод дотичних (Ньютона-Рафсона). Програма «Підбір параметру» .....	152
5.5 Метод ітерацій .....	159
5.6 Методи Чебишова та Ейткена побудови ітерацій вищих порядків .....	163
5.7 Системи нелінійних рівнянь. Відокремлення коренів .....	169
5.8 Метод простих ітерацій для розв'язання систем нелінійних рівнянь .....	172
5.9 Метод Ньютона-Рафсона для систем рівнянь .....	177
5.10 Розв'язання систем методом найменших квадратів .....	182
5.11 Використання методу растрової візуалізації для вивчення області розв'язку систем нелінійних рівнянь .....	186
Контрольні питання .....	189
Завдання для самостійної роботи .....	191
Завдання 5.1 .....	191
Завдання 5.2 .....	193
ЛЕКЦІЯ 6 НАБЛИЖЕННЯ ФУНКЦІЙ .....	196
Мета лекції .....	196
План лекції .....	196
6.1 Постановка задачі .....	197
6.2 Задачі теорії наближень і функціональні простори .....	200
6.3 Поліноміальна інтерполяція .....	203
6.4 Інтерполяційний поліном Лагранжа .....	203
6.5 Розрахунки інтерполяційного полінома .....	207
6.6 Скінченні різниці різних порядків .....	209
6.7 Розділені різниці .....	214
6.8 Інтерполяційний поліном Ньютона .....	215
6.9 Обмеження методу інтерполяції .....	217
Контрольні питання .....	217
Завдання для самостійної роботи .....	220
ЛЕКЦІЯ 7. АПРОКСИМАЦІЯ МЕТОДОМ НАЙМЕНШИХ КВАДРАТІВ. ЕЛЕМЕНТИ СПЛАЙН-АПРОКСИМАЦІЇ .....	222
Мета лекції: .....	222
План лекції .....	222
7.1 Апроксимація функцій методом найменших квадратів .....	223
7.2 Точкова апроксимація функцій методом найменших квадратів .....	227
7.3 Кусково-поліноміальна апроксимація .....	237
7.4 Поняття сплайну .....	240
7.5 Способи аналітичного зображення сплайнів .....	241
7.6 Інтерполяція кубічними сплайнами .....	243
7.7 Обчислення параметрів кубічного сплайну .....	244
Контрольні питання .....	253
Завдання для самостійної роботи .....	254
Додаток А. Робота з електронним підручником у середовищі Foxit Reader .....	256
А1 Загальні відомості про програму .....	256
А2 Основні прийоми роботи з науковим текстом .....	256
А3 Розмітка тексту у Foxit Reader .....	256
А3.1 Створення і використання закладок .....	257
А3.2 Створення і використання заміток .....	259
А3.3 Використання олівця і гумки .....	260

# ЛЕКЦІЯ 1. ОСОБЛИВОСТІ ЧИСЕЛЬНИХ МЕТОДІВ І ЕЛЕМЕНТИ ТЕОРІЇ ПОГРІШНОСТЕЙ

## Мета лекції

1. Ознайомлення слухачів з особливостями чисельних методів, їх використанням на ЕОМ, етапами розв'язування задач на ЕОМ, основним типам алгоритмів чисельних методів.

2. Ознайомлення слухачів з типами погрішностей, що виникають при проведенні обчислень, методами розрахунків з наближеними числами, методами оцінки погрішностей складних функцій

## План лекції

[1.1 Особливості й етапи чисельного рішення задач на ЕОМ.](#)

[1.1.1 Етапи розв'язання задач](#)

[1.1.2 Принципи укладання алгоритмів обчислювальних задач](#)

[1.1.3 Особливості чисельного розв'язання математичних задач](#)

[1.2. Елементи теорії погрішностей](#)

[1.2.1 Класифікація погрішностей](#)

[1.2.2 Абсолютна й відносна погрішності](#)

[1.2.3 Запис наближених чисел. Правила округлення](#)

[1.2.4. Погрішності функцій](#)

[1.2.5 Погрішності при складних обчисленнях](#)

[1.2.6 Погрішності обчислень на комп'ютерах](#)

[1.2.7 Комп'ютерне моделювання погрішностей](#)

[Контрольні запитання](#)

[Завдання для самостійної роботи](#)

Матеріал лекції**1.1 Особливості й етапи чисельного рішення задач на ЕОМ****1.1.1 Етапи розв'язання задач**

Розв'язання різних задач на ЕОМ будь-якого типу містить наступні етапи:

- ✓ постановку задачі та отримання її математичного опису;
- ✓ вибір чисельного методу розв'язання;
- ✓ складання алгоритму обчислень;
- ✓ розробка проекту (програми), його редагування й налагодження.
- ✓ проведення розрахунків;
- ✓ аналіз та інтерпретація результатів обчислень.

*Постановка задач* виникає в конкретній предметній області. Вона є неформальною процедурою. На цьому етапі створюється модель об'єкту у вигляді системи рівнянь балансу і динаміки.

Як правило, сучасні моделі рідко допускають аналітичне рішення, тому наступний крок - *вибір чисельного методу рішення* задачі. На цьому етапі:

- аналізуються відомі чисельні методи з позиції перспектив їх застосування,
- аналізуються можливі ускладнення й обмеження, достоїнства й недоліки методів,
- вибираються варіанти численного розв'язання.

Цій курс стосується опису базових чисельних методів, які найчастіше використовуються для рішення типових розрахункових задач, що виникають в сучасній інженерній практиці у різноманітних галузях.

Наступні етапи стосуються безпосередньо проектування й створення програмного забезпечення. Відбувається:

- розробка блокового алгоритму розв'язання задачі (див. [1.1.2](#)),
- вибір програмних засоби реалізації цього алгоритму.

Сучасний арсенал програмних засобів обчислювальної математики є величезним, наявні універсальні пакети, які дозволяють розв'язувати практично всі стандартні обчислювальні задачі, а також організовувати представлення результатів у формі інтегрованих проектів, що містять:

- розрахункову частину,
- інформаційно-текстову частину,
- графічну частину,
- систему навігації по проекту.

Таким чином разом із програмою розрахунків створюється звітний документ. Тому важливо визначити, чи можна для рішення задачі обійтися наявними стандартними засобами або варто становити й налагоджувати програму розрахунків на однієї з мов програмування. За рахунок ефективного використання стандартних пакетів можна заощадити час і значні засоби. Якщо розв'язується деяка унікальна

задача, то доцільно використати ці пакети разом з вбудованою мовою програмування, що підвищує ефективність рішення задачі й заощаджує час.

В результаті використання зазначеної технології створюється інтегрований електронний документ, у якому представлені:

- результати розрахунків,
- їхня інтерпретація й аналіз.

Етап інтерпретації і аналізу є неформальним і відноситься до предметної області задачі, що розв'язується.

### 1.1.2 Принципи укладання алгоритмів обчислювальних задач

*Алгоритм* – це точно визначене правило дій, для якого задана вказівка, як і в якій послідовності це правило застосовувати до вихідних даних завдання, щоб одержати розв'язок.

Алгоритми можуть представлятися у вигляді:

- ✓ формульно-словесного опису;
- ✓ блок-схеми.

*Формульно - словесний опис* - це інструкція із пронумерованих або промаркірованих рядків, у яких утримується опис порядку дій при обчисленнях

#### Приклад 1.1

Укласти формульно-словесний алгоритм наближеного розрахунку квадратного кореню з числа  $a$  (алгоритм Герона). Основна формула:

$$\sqrt{a} \approx x_{n+1} = \frac{1}{2} \left( x_n + \frac{a}{x_n} \right). \quad (1.1)$$

Розв'язання:

Алгоритм:

а) уводимо вхідні дані:

- 1) число  $a$ , з якого витягаємо квадратний корінь,
- 2) початкове наближення  $x_0$ ,
- 3) точність розрахунків  $\varepsilon$ ;

б) розраховуємо наступне наближення  $x_1$  по формулі (1.1);

в) перевіряємо виконання нерівності (1.2) для модуля різниці між сусідніми двома наближеннями:

$$|x_1 - x_0| < \varepsilon; \quad (1.2)$$

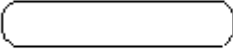

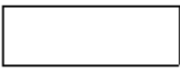
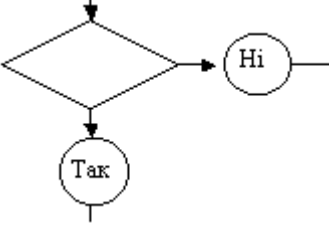
1) якщо нерівність (1.2) виконується - *розв'язок знайдено*, значення  $x_1$  - шуканий корінь, розрахований з необхідною точністю;

2) якщо нерівність *не виконується*:

- 1а) Підставити замість  $x_0$  отримане значення  $x_1$ ,
- 1б) Перейти на крок б)

Блок-схема алгоритму являє собою графічний опис формульно-словесного алгоритму, у вигляді послідовності дій, представлених символічно й сполучених стрілками. Для укладання використовують типові позначення, що наведені у таблиці 1.1. На рис. 1.2 наведено блок-схему алгоритму Герона з прикладу 1.1.

Таблиця 1.1 – Типові позначення елементів блок-схем алгоритмів

Позначення	Дія, що виконується
	Границі алгоритму – умовне позначення початку і кінця
	Зображення операцій вводу значень змінних і виводу на друк (екран монітора)
	Зображення дій зі змінними: присвоєння їм значень, математичних операцій.
	Перевірка виконання умови, записаної усередині ромба. При її виконанні подальший рух здійснюється по гілці "Так". Якщо умова не виконується - рух по гілці "Ні".



### ГЕРОН ОЛЕКСАНДРІЙСЬКИЙ (10-75)

- давньогрецький математик і механік з Олександрії. Уперше застосував роботу пари в механізмах. Він уперше досліджував п'ять типів найпростіших машин: важіль, комір, клин, гвинт і блок. Заклав основи автоматички. Винайшов ще ряд приладів, автоматів і військових машин. Вивів формули визначення площі різних геометричних фігур, розробив наближені методи

## витягу корінь

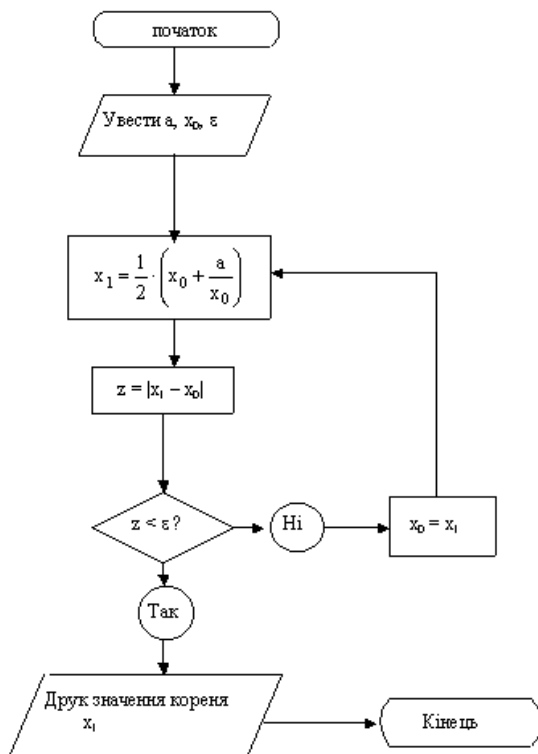


Рисунок 1.1 – Блок-схема алгоритму Герона

Крім зазначених, при укладанні блок-схем застосовуються інші символи службового характеру:

- ✓ коментарі,
- ✓ покажчики місць налагодження,
- ✓ покажчики проміжного виведення на різні носії й т.п.

Блок-схема алгоритму дає наочне уявлення ходу обчислень. При складанні більших проектів або програм використовують зображення укрупнених блоків, що містять розв'язання певної частини завдання (це називається *підзадача* або *підпрограма*). Потім кожний модуль деталізують окремо.

Блок-схема або формульно-словесний опис - *основа подальшого складання програми (програмного продукту)*. По суті програмування - це запис блок-схеми або формульно-словесного алгоритму за допомогою конкретної мови програмування, з використанням її правил і засобів.

Сучасний підхід до розв'язку задач, заснований на використанні комп'ютерних обчислювальних систем, вимагає не тільки зусиль для програмування. Пред'являються підвищені вимоги до проектування алгоритмів на рівні укрупнених блок-схем, що становлять основу сучасних обчислювальних проектів.

### 1.1.3 Особливості чисельного розв'язання математичних задач

На прикладі чисельного розв'язання задачі (1.1) ([приклад 1.1](#)) розберемо характерні риси більшості чисельних методів.

Більшість обчислювальних задач розв'язують шляхом *циклічного виконання однотипних операцій*. Це видно з рис.1.1 - наявність повернення після перевірки нерівності на попередній крок, але з новим значенням  $x_0$ . Така конструкція називається *циклом*

У обчислювальних задачах цикли бувають двох типів:

- ✓ коли циклічні процедури виконуються заздалегідь *задану кількість разів*. Це спостерігається при виконанні матричних операцій, рішенні систем лінійних рівнянь і т.д. Такі цикли називаються *циклами з переліченням*
- ✓ коли циклічні процедури виконуються заздалегідь невідому кількість разів, до досягнення деякої умови. У задачі (1.1) обчислення виконуються до досягнення умови (1.2). При досягненні цієї умови обчислення виробляються вже по іншому каналі - здійснюється вихід із циклу. Такі цикли називаються *циклами з переадресацією*.

Розрахункові процеси, в яких реалізуються цикли з переліченням – *крокові процеси*. Для них можливо точно розрахувати, яку кількість операцій необхідно виконати для розв'язання задачі в залежності від її вимірності.

До появи циклів з переадресацією приводить один з основних підходів до обчислень, що називається *методом послідовних наближень*. Його сутність полягає в тому, що спочатку задається деяке початкове наближення, по якому потім виробляється обчислення наступного наближення, і т.д., до досягнення заданої точності. У задачі (1.1) таким є значення  $x_0$ , що задає в початку обчислень.

Кожний окремих цикл методів послідовних наближень зветься *"ітерація"*, а методи в цілому називають *ітераційними*. Іноді ітераціями називають також кроки в методах із циклами з перерахуванням.

Умова досягнення розв'язку (1.2) виражається в тому, що різниця між поточним і наступним наближеннями по модулі буде зменшуватися від ітерації до ітерації. Це можна проілюструвати результатами розрахунку числа  $\sqrt{5}$  за алгоритмом Герона (таблиця 1.2)

Як впливає з таблиці 1.2, на кожній ітерації значення величини  $|x_0 - x_1|$  швидко зменшується, і після 4-й ітерації різниця між отриманим значенням відрізняється від значення кореня менш чим на  $10^{-6}$

Якщо ітераційний процес прагне до досягнення кінцевого граничного рішення, то говорять, що процес *збігається*.

Таблиця 1.2 – Послідовні наближення для розрахунку числа  $\sqrt{5}$  за алгоритмом Герона. Початкове наближення:  $x_0 = 1$

Номер ітерації	$x_0$	$x_1$	$ x_0 - x_1 $
0	1	3	2
1	3	2,333333	0,666667
2	2,333333	2,238095	0,095238
3	2,238095	2,236069	0,002026
4	2,236069	2,236068	$9,18 \cdot 10^{-7}$

Для зупинки збіжного ітераційного процесу необхідно задавати *точність одержуваного рішення*.

Якщо збіжність спостерігається при всіх значеннях  $x_0$ , що належить деякої області  $A$  ( $x \in A$ ), то ця область називається *областю збіжності*. Доведено, що алгоритм Герона збігається при будь-яких дійсних значеннях  $x_0$ , крім  $x_0 = 0$ .

Якщо в ході ітераційного процесу:

- ✓ спостерігається постійне збільшення різниці між поточним і наступним наближенням,
- ✓ величина  $|x_0 - x_1|$  у межі прагне до нескінченності,

то говорять, що процес *розбігається*. Оскільки в будь-якій мові програмування передбачено максимально можливе позитивне число, яким оперує комп'ютер, те, по його досягненні в ітераційному процесі відбувається примусова зупинка програми ("переповнення"), з видачею на екран відповідної вказівки про помилку.

Область значень  $x_0$ , у якій ітераційний процес не приводить до рішення, а спостерігається розбіжність, називається *областю розбіжності*, або *областю нестійкості*.

Може відбутися така ситуація, що обчислювальний процес сходиться дуже повільно, і різниця між поточним і наступним наближенням не перевищує заданої точності, хоча процес може бути далекий від завершення. У такому випадку можна одержати невірну відповідь.

Таким чином, обчислювальні процеси ітераційного типу *можуть істотно залежати від значення початкового наближення*, і задача полягає в тім, щоб:

- ✓ використати початкові наближення з області стійких рішень,
- ✓ досліджувати алгоритми обчислень на стійкість.

## 1.2 Елементи теорії погрешностей

### 1.2.1 Класифікація погрешностей

При розв'язанні практичних задач на будь-яких обчислювальних пристроях розрахунки, як правило, виконуються над наближеними вихідними даними. У таких випадках необхідно:

уміти грамотно організувати обчислення,

- ✓ знати, які бувають похибки,
- ✓ вміти правильно записувати наближені вихідні дані,
- ✓ вміти оцінювати похибку результату по відомих похибках компонентів,
- ✓ визначати оптимальну кількість знаків у вихідних даних, щоб результат обчислень мав наперед задану точність,
- ✓ обирати найбільш раціональний порядок обчислень, а також алгоритм обчислень, стійкий до помилок округлень.

Не маючи достатніх навичок практичних обчислень, можна при розв'язанні задач отримати результат, що не має нічого спільного з дійсним розв'язком.

Основними джерелами погрешності наближеного розв'язку прикладних задач є:

а) *погрешність математичної моделі*. Вона ставиться до класу *неусувних* погрешностей і має такі особливості:

- 1) пов'язана з фізичними допущеннями,
- 2) не контролюється в процесі чисельного рішення завдання,
- 3) може зменшитися тільки за рахунок більше точного опису фізичної задачі;

б) *погрешність вихідних даних* Ставиться до класу *неусувних* погрешностей і має такі особливості:

- 1) пов'язана з технікою одержання експериментальних даних,
- 2) значення параметрів, що входять у математичний опис завдання, вимірюються експериментально з деякою погрешністю,
- 3) може зменшитися тільки за рахунок удосконалювання техніки експерименту;

в) *погрешність наближеного методу*. При чисельному розв'язанні задач *точний оператор*, у якому кількість чисел або операцій перевищує припустимі границі, *заміняється на наближений*, який потребує кінцевої кількості операцій. Наприклад,:

- 1) інтеграл заміняють сумою,
- 2) функцію заміняють багаточленом,
- 3) будують нескінченний ітераційний процес і обривають його після кінцевого числа ітерацій.

Цю погрешність досліджують для кожного конкретного чисельного методу.

г) *Обчислювальна погрешність*, що виникає в результаті змушеного округлення чисел, наприклад, кінцевого числа розрядів ЕОМ.

Стійкість задач. Якщо рішення деякої задачі неперервне залежить від вхідних даних, тобто *малій зміні вхідних даних відповідає мала зміна рішення*, то задача називається *стійкою* за вхідним даними.

У стійкому обчислювальному алгоритмі похибки округлення *не накопичуються*.

### 1.2.2 Абсолютна й відносна погрішності

Точність наближеного числа характеризується поняттями *абсолютної* й *відносної* погрішності.

*Абсолютною погрішністю* наближеного числа  $a$  називається величина:

$$\Delta = |A - a|, \quad (1.3)$$

де  $A$  - точне значення деякої величини,

$a$  - одне з його наближених значень.

Абсолютна погрішність  $\Delta$  представляє тільки теоретичний інтерес, оскільки точне значення  $A$  не відомо. Тому на практиці частіше користуються *граничною абсолютною погрішністю*  $\Delta_a$  наближеного числа  $a$ , рівної по можливості *найменшому числу*, для якого виконується нерівність:

$$\Delta = |A - a| \leq \Delta_a. \quad (1.4)$$

Значення  $a$  та  $\Delta_a$  дозволяють указати *інтервал*, що містить *точне значення*  $A$ :

$$a - \Delta_a < A < a + \Delta_a. \quad (1.5)$$

Частіше використовується більше компактний запис:

$$A = a \pm \Delta_a. \quad (1.6)$$

Основною характеристикою точності наближеного числа є його *відносна погрішність*:

$$\delta = \frac{\Delta}{|A|}. \quad (1.7)$$

Оскільки число  $A$  невідомо, як правило, вважають:

$$\delta = \frac{\Delta}{|a|}. \quad (1.8)$$

Аналогічно з нерівності  $\delta < \delta_a$  визначають *граничну відносну погрішність* числа  $a$ , вважаючи:

$$\delta_a = \frac{\Delta_a}{|a|}. \quad (1.9)$$

Величина  $\delta$  характеризує якість наближення. Це безрозмірна величина, як правило її виражають у відсотках.

### 1.2.3 Запис наближених чисел. Правила округлення

Записувати наближене число у вигляді (1.6) у процесі розрахунків незручно. Тому в обчислювальній практиці прибігають до різних прийомів, що дозволяють тільки по записі самого наближеного числа а судити про його погрішності.

Нехай наближене число а представлене у вигляді кінцевого десяткового дробу:

$$a = \alpha_m \cdot 10^m + \alpha_{m-1} \cdot 10^{m-1} + \dots + \alpha_{m-n+1} \cdot 10^{m-n+1}, (\alpha_m \neq 0). \quad (1.10)$$

де  $\alpha_i$  - цифри числа а ( $\alpha_i = 0, \dots, 9$ ).

*Значущими цифрами* числа називають всі цифри в записі числа, починаючи з першої ненульовий ліворуч..

#### Приклад 1.2

Підкреслити значущі цифри у чисел  $a = 0,0503$ ,  $b = 0,00630500$

Розв'язання:

$a = 0,0503$  (підкреслюємо, починаючи з першої ненульової цифри ліворуч)

$b = 0,00630500$  (нульові цифри праворуч входять до складу значущих)

Значущу цифру називають *вірною*, якщо абсолютна погрішність числа *не перевершує одиниці розряду, що відповідає цій цифрі*. У протилежному випадку цифра вважається *сумнівною*.

#### Приклад 1.3

Оцінити абсолютну погрішність в числах, що наведені у прикладі 1.2

Розв'язання

В прикладній математиці та фізиці поняття «оцінити» означає розрахувати з *точністю до порядку величини*. У даному випадку доцільно одержати *верхню границю* абсолютної погрішності.

У запису числа  $a = 0,0503$  остання цифра (3) відповідає розряду  $10^{-4}$ . Тому, щоб остання цифра була вірною, необхідно, щоб абсолютна погрішність була меншою за  $10^{-4}$ , тобто:  $\Delta_a < 1 \cdot 10^{-4}$ . Це є шукана верхня границя погрішності.

У запису числа  $b = 0,00630500$  остання цифра (0) відповідає розряду  $10^{-8}$ . Тому шукана верхня границя погрішності:  $\Delta b < 1 \cdot 10^{-8}$ .

Якщо вихідне число має кілька сумнівних цифр, його варто попередньо округлити. Доводиться також округляти точні числа, що містять занадто багато або нескінченну кількість цифр.

Щоб округлити число до  $n$  значущих цифр:

- ✓ відкидають всі цифри його, що розташовані *праворуч від  $n$ -ї значущої цифри*,
- ✓ якщо це необхідно для збереження розрядів, *заміняють відкинуті цифри нулями*.

При округленні керуються наступними правилами:

*Правило 1:* остання збережена цифра *не змінюється*, якщо перша цифра, яку відкидають або заміняють, *є меншою за 5*,

*Правило 2:* остання збережена цифра *збільшується на одиницю*, якщо перша цифра, що відкинута або замінена, *є більшою 5 або дорівнює 5*;

*Правило 3:* при відкиданні або заміні тільки однієї цифри 5 остання цифра, що зберігається:

- ✓ *не змінюється*, якщо вона *парна*,
- ✓ *збільшується на одиницю*, якщо вона *непарна*.

*Правило 4:* якщо наближені числа не є остаточними для розрахунків що виконуються й з ними передбачається робити ще обчислення, варто зберігати *одну або дві сумнівні цифри*.

*Правило 5* (акад. О. Н. Крилова): наближене число треба писати так, щоб у ньому:

- а) всі значущі цифри, *крім останньої*, були вірними,
- б) лише остання цифра була *б сумнівною*,
- г) при цьому величина абсолютної погрішності не повинна перевищувати *половини одиниці останнього залишеного розряду*.



КРИЛОВ ОЛЕКСІЙ  
МИКОЛАЙОВИЧ

(1863-1945) – видатний російський математик, суднобудивельник, один з засновників сучасного математичного моделювання. Зробив значний внесок у розвиток чисельних методів алгебри, диференціальних рівнянь. Створив теорію розрахунків суден..

Величина відносної погрішності наближеного числа пов'язана з кількістю  $n$  вірних знаків цього числа і його першою значущою цифрою  $\alpha_m$  співвідношенням:

$$\delta_a \leq \frac{1}{2\alpha_m \cdot 10^{n-1}}. \quad (1.11)$$

Ця формула дозволяє грубо оцінити кількість  $n$  вірних знаків наближеного числа, якщо відомі:

- а) його відносна погрішність  $\delta_a$ ,
- б) його перша значуща цифра  $\alpha_m$ .

#### Приклад 1.4

В результаті розрахунків одержано число  $a=626735$ , оцінка абсолютної погрішності складає 680. Записати число  $a$  з врахуванням правила О.М.Крилова

#### Розв'язання

У відповідності до правила О.М.Крилова абсолютна погрішність не повинна перевищувати половини одиниці останнього залишеного розряду числа. Можна оцінити залишок числа:  $\Delta z \geq 2 \cdot \Delta a = 2 \cdot 680 = 1360$ . Тобто у числа  $a$  будуть сумнівними 4 останні цифри. Після округлення повинні залишитись 3 перших цифри, з яких перші дві ліворуч будуть вірними, а остання – сумнівною. Підкреслимо вірні цифри і закреслимо сумнівні цифри, починаючи з другої ліворуч: ~~626735~~ округлимо першу сумнівну цифру (ще не підкреслена і не закреслена) з врахуванням правил округлення, одержимо:  
 $a=627000$ .

Наведена форма запису має певний недолік: для стороннього спостерігача виникає враження, що з останніх трьох нулів лише цифра у позиції 6 є сумнівною. Тому краще за все результат записати у формі з фіксованою комою і порядком (так званий науковий формат):  $a=6,27 \cdot 10^5$ . У такому запису немає «зайвих» нулів.

### 1.2.4. Погрішності функцій

Нехай задано деяку функцію  $y = f(x_1, \dots, x_n)$  від  $n$  аргументів  $x_1, \dots, x_n$ , причому значення кожного з її аргументів  $x_i$  визначені з деякими абсолютними погрішностями  $\Delta x_i$ ,  $i = 1, 2, \dots, n$ . Потрібно знайти погрішність даної функції. Припустимо, що функція є неперервне диференціюємою в області  $D$ .

Позначимо величину абсолютної погрішності функції через  $\Delta y$ :

$$\Delta y = |f(x_1 + \Delta x_1, \dots, x_n + \Delta x_n) - f(x_1, \dots, x_n)|. \quad (1.12)$$

Припускаємо, що величини  $\Delta x_i$  досить малі. Абсолютну погрішність функції будемо вважати такою, що приблизно дорівнює її диференціалу

$$\Delta y \approx |df(x_1, \dots, x_n)| = \left| \sum_{i=1}^n \frac{\partial f}{\partial x_i} \cdot \Delta x_i \right| \leq \sum_{i=1}^n \left| \frac{\partial f}{\partial x_i} \right| \cdot \Delta x_i. \quad (1.13)$$

З (1.13) визначимо верхню межу для  $\Delta y$ :

$$\Delta_y = \sum_{i=1}^n \left| \frac{\partial f}{\partial x_i} \right| \cdot \Delta_{x_i}, \quad (1.14)$$

где  $\Delta_{x_i}$  - граничні абсолютні погрішності аргументів  $x_i$  ( $i=1, \dots, n$ );

$\Delta_y$  - гранична абсолютна погрішність функції.

Оцінка для відносної погрішності функції виходить шляхом ділення обох частин нерівності (1.13) на  $|y|$ :

$$\delta \leq \sum_{i=1}^n \frac{1}{|y|} \cdot \left| \frac{\partial f}{\partial x_i} \right| \cdot \Delta_{x_i} = \sum_{i=1}^n \left| \frac{\partial}{\partial x_i} \ln y \right| \cdot \Delta_{x_i}. \quad (1.15)$$

Граничну відносну погрішність функції  $y$  можна прийняти такою, що дорівнює:

$$\delta_y = \sum_{i=1}^n \left| \frac{\partial}{\partial x_i} \ln y \right| \cdot \Delta_{x_i}. \quad (1.16)$$

*Погрішність алгебраїчної суми:*

а) нехай  $y = x_1 + x_2 + \dots + x_n$ . За формулами (1.14, 1.15) отримаємо:

$$\Delta_y = \Delta_{x_1} + \dots + \Delta_{x_n}. \quad (1.17)$$

$$\delta_y \leq \max(\delta_{x_1}, \dots, \delta_{x_n}). \quad (1.18)$$

Звідси:

- ✓ абсолютна погрішність суми дорівнює сумі абсолютних погрішностей доданків,
- ✓ відносна погрішність не перевершує максимальний відносної погрішності доданків.

*Погрішність різниці двох чисел:*

$$\Delta y = \Delta_{x_1} + \Delta_{x_2}. \quad (1.19)$$

Якщо числа  $x_1$  і  $x_2$  близькі, то гранична відносна погрішність *може стати дуже великою*, хоча відносні погрішності кожного з доданків є малими.

### Приклад 1.5

Задано числа:  $x_1 = 1,0021$ ;  $x_2 = 1,0020$ , гранична абсолютна погрішність кожного числа: 0,00005. Розрахувати відносну погрішність кожного з чисел та їх різниці.

Розв'язання

Гранична відносна погрішність різниці дорівнює сумі їх граничних погрішностей, тобто:

$$\Delta_{\text{різн}} = 0,00005 + 0,00005 = 0,0001$$

Величина різниці складе:

$$x_2 - x_1 = 1,0021 - 1,0020 = 0,0001$$

Звідси гранична відносна погрішність різниці: 0,0001. Різниця чисел становить 0,0001.

Граничні відносні погрішності:

- вхідних чисел:  $\frac{0,00005}{1,002} \cdot 100 = 0,05\%$

- різниці:  $\frac{0,0001}{0,0001} \cdot 100\% = 100\%$

Тобто, при вирахуванні різниці двох близьких чисел відбулася різка втрата точності.

*Погрішність добутку й частки:*

нехай:

$$y = x_1 \cdot x_2 \cdot \dots \cdot x_n = \prod_{i=1}^n x_i, \quad \{x_1, \dots, x_n\} > 0. \quad (1.20)$$

Після логарифмування:

$$\ln y = \ln x_1 + \dots + \ln x_n. \quad (1.21)$$

Звідси, згідно (1.16)):

$$\delta y = \delta_{x_1} + \dots + \delta_{x_n}, \quad (1.22)$$

тобто гранична відносна погрішність *добутку* дорівнює *сумі* граничних відносних добутоків співмножників;

Аналогічно для частки:

$$y = \frac{x_1}{x_2}; \quad \delta y = \delta_{x_1} + \delta_{x_2}. \quad (1.23)$$

*Погрішність ступеня й кореня.*

Нехай  $y = x^n$ . У цьому випадку:

$$\ln y = n \cdot \ln x, \quad \delta_y = n \cdot \delta_x. \quad (1.24)$$

Якщо  $y = \sqrt[n]{x}$ :

$$\ln y = \frac{1}{n} \ln x; \quad \delta_y = \frac{1}{n} \delta_x. \quad (1.25)$$

Тобто:

- ✓ гранична відносна погрішність  $n$ -й ступеня числа в  $n$  раз *більше* граничної відносної погрішності самого числа,
- ✓ гранична відносна погрішність кореня  $n$ -й ступеня в  $n$  раз менше граничної відносної погрішності самого числа

### 1.2.5 Погрішності при складних обчисленнях

У практичних розрахунках з більшим числом операцій погрішність кожної операції як правило не визначається. Замість цього використовують правила округлення результатів дій над наближеними числами залежно від числа вірних цифр або десяткових знаків у вхідних даних.

Правила виконання складних обчислень:

а) при додаванні й вирахуванні наближених чисел зберігаємо *стільки* десяткових знаків, скільки їх було в числі з даних з *найменшим* числом десяткових знаків;

б) при множенні й розподілі зберігаємо в результаті стільки значущих цифр, скільки їх має наближене число з даних з *найменшим* числом значущих цифр;

в) при зведенні у квадрат або куб і при витягу кореня (квадратного або кубічного) у результаті залишаємо стільки значущих цифр, *скільки їх має дане число*;

г) у всіх проміжних результатах необхідно брати на *одну цифру більше*, ніж треба. В остаточному результаті ця запасна цифра відкидається;

д) якщо деякі дані мають більше десяткових знаків або значущих цифр, ніж інші, їх попередньо варто *округлити*, залишаючи лише одну *зайву цифру*.

е) Якщо вхідні дані можна брати з довільною точністю, то для одержання результату з  $k$  вірними цифрами вихідні дані беремо з таким числом вірних цифр, що дає відповідно до правил а)-в)  $k + 1$  цифру в результаті.

#### Приклад 1.6

Визначити припустиму погрішність аргументів по заданій граничній погрішності функції. Ця задача називається зворотною задачею теорії погрішності

Розв'язання

Ця задача розв'язується однозначно тільки для функції однієї змінної  $v = f(x)$ . Якщо одночасно:

1)  $f(x)$  є диференційованою,

2)  $f(x) \neq 0$ ,

те з (1.14) при  $n = 1$  отримаємо:

$$\Delta_x = \frac{1}{|f'(x)|} \cdot \Delta_y. \quad (1.26)$$

У випадку функції декількох змінних варто вводити деякі додаткові обмеження. Можна застосувати, наприклад, *принцип рівних впливів*, тобто вважати, що у формулі (1.14) всі доданки рівні. Тоді граничні абсолютні погрішності аргументів можуть із обчислені в такий спосіб:

$$\Delta_{x_i} = \frac{\Delta_y}{n \cdot \left| \frac{\partial f}{\partial x_i} \right|}, \quad i = 1, \dots, n. \quad (1.27)$$

Якщо припустити, що граничні абсолютні погрішності всіх аргументів є рівними:  $\Delta_{x_1} = \Delta_{x_2} = \dots = \Delta_{x_n}$ , те:

$$\Delta_{x_i} = \frac{\Delta_y}{\sum_{i=1}^n \left| \frac{\partial f}{\partial x_i} \right|}. \quad (1.28)$$

### 1.2.6 Погрішності обчислень на комп'ютерах

Однієї з основних особливостей обчислень на комп'ютерах є *скінчене число розрядів* при обчисленнях. Тому, якщо число розрядів у вихідних даних або проміжних результатах перевищує розрядність, те останні розряди такого числа відкидаються. Якщо кількість обчислень велика, то буде:

- ✓ накопичуватися погрішність округлень;
- ✓ знижуватися точність отриманого результату.

Втрата точності може відбуватися:

- а) при додаванні, вирахуванні, множенні *чисел різних порядків*,
- б) при розподілі чисел, якщо порядок діленого *значно менше* порядку дільника, і обчислення проводяться в режимі з фіксованою комою;
- в) у деяких випадках результати проміжних обчислень можуть попадати:
  - ✓ в область машинного нуля - з'являється мале, але відмінне від 0 число, порядку  $10^{-15}$ ;
  - ✓ в область нескінченності - відбувається переповнення розрядної сітки.

Цього частково можна уникнути, якщо представляти числа в стандартній формі із плаваючою комою. Крім того, доцільно застосовувати нормування вихідних даних і коефіцієнтів обчислювальних формул, помноживши їх на постійний множник, що зрівнює порядки.

Значення елементарних функцій обчислюються по вбудованих програмах теж з деякою погрішністю. У математичних пакетах використовуються розкладання функцій у:

- нескінченні степеневі ряди,
- нескінченні ланцюгові дроби,

- користуватися нескінченними ітераційними процесами.

Їх доводиться обривати на деякому кінцевому кроці, що також впливає на остаточний результат.

При обчисленнях над наближеними числами *погрешність результату залежить від порядку виконання обчислень.*

### Приклад 1.7

Встановити, як впливає порядок виконання на величину виразу:  $\frac{1 \cdot 6}{6}$ , якщо розрахунки виконувати з двома значущими цифрами.

#### Розв'язання

а) обчислимо вираження  $(1 \times 6) : 6$  з точністю до двох знаків після коми:  $1,00 \times 6,00 = 6,00$ ;  $6,00 : 6,00 = 1,00$ .

б) Змінимо порядок обчислень.  $(1:6) \times 6$ :

$$1) 1,00 : 6,00 = 0,17;$$

$$2) 0,17 \times 6,00 = 1,02.$$

Результати обчислень відрізняються (на 2 %), незважаючи на те, що з погляду "чистої" математики повинен вийти той самий результат.

В деяких задачах алгоритм обчислень може бути нестійким до погрешностей округлень. Наприклад, якщо в системі:

$$\begin{cases} 10,1x + 9,9y = 20,0 \\ 9,9x + 10,1y = 20,0 \end{cases}$$

що має розв'язок  $(1; 1)$ , змінити на 0,5 % праві частини:

$$\begin{cases} 10,1x + 9,9y = 20,1 \\ 9,9x + 10,1y = 19,9 \end{cases}$$

те отримаємо розв'язок  $(1,5; 0,5)$ , що відрізняється від попереднього на 50 %.

Щоб уникнути нагромадження похибок, необхідно застосовувати наступні рекомендації з вибору найкращої послідовності обчислень:

а) при додаванні великого числа що складають варто розбити їх на групи чисел, близьких один по одному, підсумувати доданки кожної групи, а потім складати отримані суми послідовно від найменшої до найбільшого;

б) при множенні декількох чисел рекомендується множити найбільший множник на найменший (з огляду на результат попереднього множення). Це запобігає влученню в процесі обчислень в область машинного нуля або нескінченності;

в) Уникати вирахування й розподілу на різницю близьких один по одному чисел, шляхом перетворення формули обчислень. Наприклад, проводити перетворення такого роду:

$$(a+b)^2 - a^2 = 2ab + b^2, \quad \frac{1}{\sqrt{a^2+b^2}-a} = \frac{\sqrt{a^2+b^2}+a}{b^2}, \quad a \gg b; \quad (1.29)$$

г) вибрати краще той алгоритм, що вимагає меншої кількості обчислень;  
 д) вибрати по можливості чисельний метод, стійкий до погрішностей округлень.

### 1.2.7 Комп'ютерне моделювання погрішностей

У багатьох практичних обчислювальних задачах як правило важко використовувати безпосередньо аналітичні методи оцінки похибок обчислень. У цих випадках для оцінки похибок складних обчислювальних схем можна використовувати комп'ютерне моделювання зокрема *метод збурень*

Сутність методу збурень полягає в наступному:

а) до кожного значення вхідних аргументів  $x_i$  ( $i = 1, 2, \dots, n$ ) додають невелике збурення  $\delta x_i$ ;

б) для збурених вхідних аргументів  $x_i + \delta x_i$  розраховують збурені значення вихідної функції  $f(x_1 + \delta x_1, x_2 + \delta x_2, \dots, x_n + \delta x_n)$ ;

в) вивчають залежність величини збурення функції від збурень аргументів.

Для вивчення залежності збурень функції від збурень аргументів використовують *статистичне моделювання*. Сутність полягає у наступному:

а) на комп'ютері генерують вибірку об'ємом  $m$  значень аргументів, як випадкових величин з відомим законом розподілу (як правило, рівномірним),

б) розраховують для кожного набору аргументів вихідну функцію, одержують вибірку значень функції,

в) для одержаної вибірки розраховуємо середнє значення  $\bar{f}$  і середнє квадратичне відхилення  $\sigma$ :

$$\bar{f} = \frac{\sum_{i=1}^n f_i}{n}. \quad (1.30)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (f_i - \bar{f})^2}{n-1}} \quad (1.31)$$

Величина  $\bar{f}$  характеризує середнє (незбурене) значення функції. Величина  $\sigma$  є мірою абсолютної похибки, а відношення  $\sigma/|\bar{f}|$  характеризує відносну погрішність. Вивчення залежності  $\sigma/|\bar{f}|$  від характеристик збурення можна встановити, чи є обчислювальна схема сталою при певному значенні вхідних аргументів і оцінити абсолютну й відносну погрішності в залежності від погрішності аргументів<sup>1)</sup>.

<sup>1)</sup> Розглянутий метод слід розглядати, як один з варіантів методу випадкових випробувань (Монте-Карло), одного з найважливіших методів сучасної прикладної математики. У подальшому ми декілька разів будемо торкатися в нашому курсі використання цього методу. Детально основи методу вивчаються в курсі «Випадкові процеси»

Для обчислювальних процесів можна вважати, що розподіл погрішностей вхідних величин підпорядковується рівномірному закону, і усі величини погрішностей зосереджені в інтервали  $(x_i - \delta x_i; x_i + \delta x_i)$ . Якщо задавати різні випадкові величини, що розподілені за рівномірним законом в цьому інтервалі, те одержимо шукану вибірку значень аргументів, для якої мірою погрішності  $\epsilon$ , наприклад, величина довжини інтервалу  $2\delta x_i$ . Оскільки в усіх мовах програмування і обчислювальних пакетах є функція генерації випадкових чисел  $\alpha$ , що рівномірно розподілені в інтервалі  $(0;1)$ , шукану випадкову величину  $\beta$  з рівномірним розподілом в  $(x_i - \delta x_i; x_i + \delta x_i)$ :

$$\beta = (x_i - \delta x_i) + 2 \cdot \delta x_i \cdot \alpha. \quad (1.32)$$

### Приклад 1.8

За допомогою комп'ютерного моделювання визначити залежність похибки виразу (1.33) в точках  $(3.5;2)$  і  $(2,01; 2)$  в залежності від погрішності вхідних даних.

$$f(x, y) = \frac{\sqrt{|x^3 - y^3|}}{1 - \exp(x - y)}. \quad (1.33)$$

### Розв'язання

За міру погрішності вхідних даних оберемо довжину напівінтервалу  $\delta$ , у середині якого буде знаходитися точка, що задана (рис.1.2). Текст скрипту Scilab для проведення розрахунків наведено в лістингу 1.8.

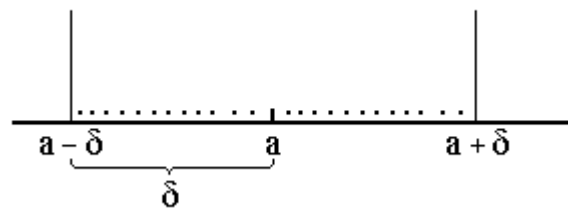


Рисунок 1.2 – Вибір міри погрішності, як напівінтервалу. Точками схематично вказані результати випадкової генерації погрішності

### Лістинг 1.8

```
function f=fi(x,y) // цільова функція
    f=sqrt(abs(x.^3-y.^3))/(1-exp(x-y))
endfunction
a=3.5; // значення x-координати точки
b=2; // значення y-координати точки
```

```

NW=10000; // кількість випробувань
delta=0.1 // напівінтервал погрішності
for i = 1:NW
    x=a-delta + 2*delta*rand();
    y=b-delta + 2*delta*rand();
    f(i)=fi(x,y); //масив значень функції
end
fs=mean(f) // середнє значення
sig=st_deviation(f) // середнє квадратичне відхилення
r=sig/abs(fs) // результат, відносна погрішність функції

```

Будемо змінювати інтервал  $\delta$  від  $1 \cdot 10^{-6}$  до 0.1.

При розв'язання задачі необхідно визначити необхідну кількість випадкових випробувань для досягнення потрібної точності результатів. На практиці достатня точність результатів – це 2 вірні значущі цифри. Вважаючи, що використовуються випадкові величини, їх слід характеризувати статистичними характеристиками. Для цього слід розрахувати декілька (не менше 5) значень цільової величини (міри точності  $r = \sigma / |\bar{f}|$ ), знайти їх середнє ( $r_s$ ), середнє квадратичне відхилення ( $\sigma_r$ ). Результати розрахунків наведені у табл. 1.2. Як впливає з цих даних, при збільшення кількості випробувань з 100 до 10000 відбувається суттєва зміна середнє квадратичного відхилення. Але подальше зростання кількості випробувань до 50000 не викликає суттєвих змін середніх, і середнє квадратичне відхилення також залишається в межах 3-го знаку. Це вказує, що кількість випробувань 10000-50000 приводить до одержання результатів з достатньою точністю.

У таблиці 1.3 наведені результати вивчення впливу величини  $\delta$  на  $r_s$ . Виходячи з них, можна зробити наступні висновки:

а) погрішність є практично постійною в межах кожної серії (про це вказує невелике значення середнє квадратичного відхилення);

б) погрішність функції має той же самий порядок, що погрішність аргументів, незалежно від величини останніх (відношення  $r_s/\delta$  є практично постійним і складає 0,78). Це вказує, що задана точка відноситься до області *стійких значень* функції.

Розглянемо, якою буде погрішність розрахунків в області нестійкості, в точці (2,01;2). Дослідження проводимо таким же чином, як наведено у табл. 1.2. Узагальнені результати моделювання наведені у табл. 1.4.

Як впливає з даних табл. 1.4, в області нестійкості спостерігається значне зростання відносної погрішності функції, причому малі зміни погрішності аргументу викликають значні погрішності функції. Відношення  $r_s/\delta$  становиться великим: від 326 до 41 і далі, зі зменшенням  $\delta$ , практично не змінюється.

Таблиця 1.2 – Вплив кількості випробувань на величину середнього значення відносної погрішності та її середнє квадратичного відхилення при  $\delta=0.1$ ,  $NW = 50000$

N	Міра погрешності ( $r_s$ )		
	NW=1000	NW=10000	NW=50000
1	0,0765	0,0776	0,0780
2	0,0779	0,0781	0,0781
3	0,0806	0,0782	0,0781
4	0,0764	0,0783	0,0782
5	0,0778	0,0784	0,0782
$r_s$	0,0778	0,0781	0,0781
$\sigma_r$	0,0016	0,0003	0,00008

Таблиця 1.3 – Вплив величини  $\delta$  на  $r_s$ 

N	$\delta=0,01$	$\delta=0,005$	$\delta=0,002$	$\delta=0,001$	$\delta=0,0005$	$\delta=0,0001$
	$r \cdot 10^3$	$r \cdot 10^3$	$r \cdot 10^3$	$r \cdot 10^4$	$r \cdot 10^4$	$r \cdot 10^5$
1	7,80	3,88	1,55	7,82	3,90	7,78
2	7,82	3,90	1,56	7,78	3,87	7,76
3	7,77	3,90	1,56	7,80	3,89	7,81
4	7,77	3,89	1,56	7,80	3,89	7,79
5	7,82	3,89	1,56	7,81	3,89	7,77
$r_s$	7,80	3,89	1,56	7,80	3,89	7,78
$\sigma_r$	0,03	0,01	0,004	0,01	0,01	0,02

Таблиця 1.4 – Моделювання погрешності в області нестійкості

$\delta$	0,01	0,005	0,001	0,0005	0,0001
$r_s$	3,26	0,375	0,0416	0,0206	0,00412
$r_s/\delta$	326	75	41,6	41,2	41,2

Користуючись алгоритмом, розглянутим вище, можна проводити кількісні дослідження впливу похибки аргументів на похибку функції. Для цього:

а) проводять серію комп'ютерних експериментів з різними величинами  $\Delta$ , для кожного із значень одержують величину  $\sigma$ ,

б) вивчають залежність  $\sigma$  від  $\Delta$ . Як правило, ця залежність є наближене ступеневою:

$$\sigma = a \cdot \Delta^b. \quad (1.34)$$

Для зручності і наочності цю залежність можна зробити лінійною, якщо перейти до логарифмічних координат:

$$\ln \sigma = \ln a + b \cdot \ln \Delta. \quad (1.35)$$

Якщо встановлено виконання залежності (1.34), рівняння (1.34) можна використовувати для розв'язання:

- прямої задачі* – оцінки погрішності функції для заданого значення  $\Delta$ ,
- зворотної задачі* – оцінка  $\Delta$  для заданої погрішності функції.

### Приклад 1.9.

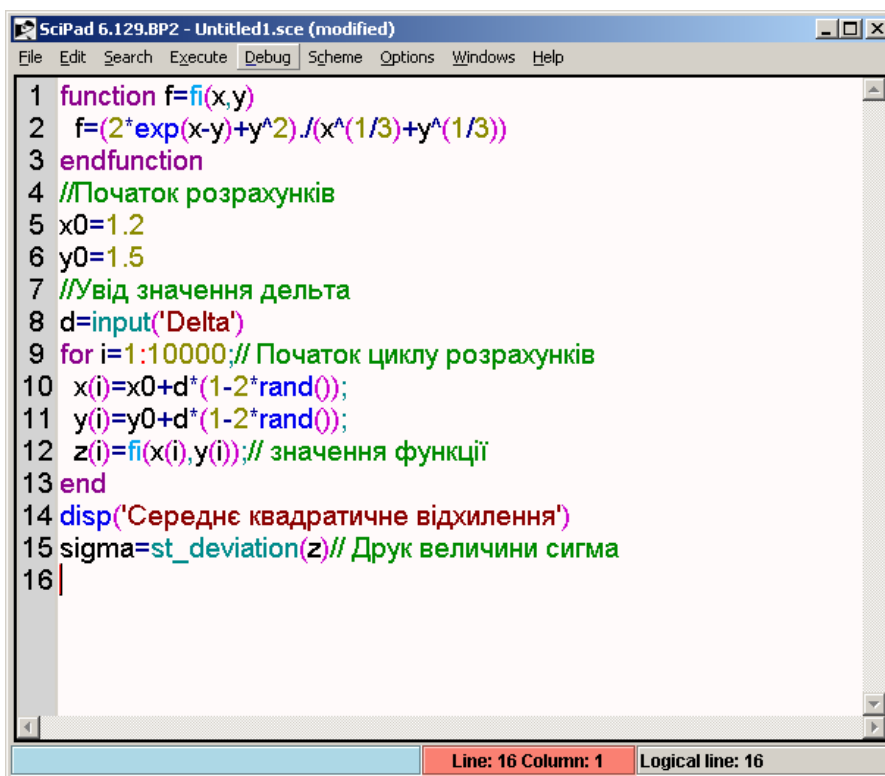
Визначити, якою повинна бути погрішність аргументів для забезпечення рівня точності порядку  $10^{-4}$  для розрахунку значення функції:

$$f(x, y) = \frac{2 \cdot e^{x-y} + y^2}{\sqrt[3]{x} + \sqrt[3]{y}}; \quad (1.36)$$

в точці (1.2; 1.5).

Розв'язання

а) укладаємо сценарій (скрипт) розрахунків Scilab (рис. 1.3)



```

1 function f=fi(x,y)
2 f=(2*exp(x-y)+y^2)/(x^(1/3)+y^(1/3))
3 endfunction
4 //Початок розрахунків
5 x0=1.2
6 y0=1.5
7 //Увід значення дельта
8 d=input('Delta')
9 for i=1:10000; // Початок циклу розрахунків
10 x(i)=x0+d*(1-2*rand());
11 y(i)=y0+d*(1-2*rand());
12 z(i)=fi(x(i),y(i)); // значення функції
13 end
14 disp('Середнє квадратичне відхилення')
15 sigma=st_deviation(z) // Друк величини сигма
16

```

Рисунок 1.3 – Скріншот сценарію розрахунків

б) виконуємо серію розрахунків при  $\Delta = 0,1; 0,05; 0,02; 0,01; 0,005; 0,002; 0,001$ , розраховуємо десятинні логарифми від значень  $\Delta$  і  $\sigma$  (рис. 1.3);

в) будуємо графік залежності  $\lg \sigma$  від  $\lg \Delta$  (рис. 1.4). Цю залежність з високим ступенем точності можна вважати прямолінійною. Находимо параметри лінійної залежності, користуючись можливостями OOO Calc;

г) виходячи з рівняння, наведеного на рис. 1.4, розраховуємо, якою повинна бути величина  $\Delta$ , щоб величина  $\sigma$  склала  $10^{-6}$ . Формули комірок:

$$C17: =(-6+0,4173)/0,9994;$$

$$D17: =10^{C17}.$$

Як впливає з даних рис. 1.3, для досягнення точності розрахунків функції (1.35)  $\sim 10^{-6}$  точність значень аргументів також повинна бути порядку  $10^{-6}$ .

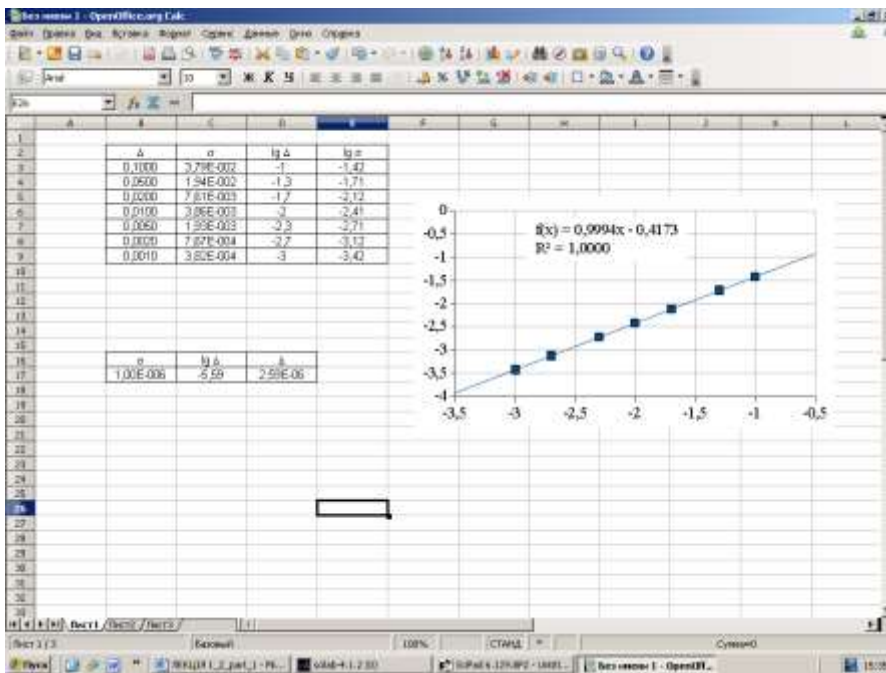


Рисунок 1.4 – Виявлення характеру залежності  $\lg \sigma$  від  $\lg \Delta$  у прикладі 1.9.

Наведені приклади показують, що комп'ютерне моделювання може бути корисною альтернативою до аналітичних досліджень, які є неможливими у випадку складних задач, які найчастіше зустрічаються на практиці.

### Контрольні запитання

- 1 Перелічіть етапи рішення завдань на ЕОМ
- 2 Що таке алгоритм?
- 3 Які існують методи подання алгоритмів?
- 4 У чим особливості формульно-словесного опису алгоритму?
- 5 У чим особливість графічного опису алгоритму?
- 6 Які основні значки використовуються при описі алгоритмів?
- 7 Яка конструкція називається циклом?
- 8 Які типи циклів використовуються при рішенні обчислювальних задач?
- 9 Який загальний алгоритм ітераційних методів?
- 10 У чим зміст поняття "збіжність процесу"?

- 11 Які явища можуть спостерігатися при виконанні ітераційних процесів на комп'ютері?
- 12 Перелічите джерела обчислювальних погрешностей. Які з них є переборними, які - немає?
- 13 У чим сутність поняття "погрешність наближеного методу"?
- 14 У чим сутність поняття "стійкий обчислювальний процес"?
- 15 У чим зміст поняття "абсолютна погрешність"?
- 16 У чим зміст поняття "відносна погрешність"?
- 17 Який з типів погрешностей є розмірною, а який - безрозмірною величиною?
- 18 Які цифри числа називаються значущими?
- 19 Яку значущу цифру називають вірною?
- 20 Сутність поняття "округлення наближеного числа"?
- 21 Правила округлення наближених чисел.
- 22 Правила Кривої для подання наближених чисел.
- 23 Як можна оцінити абсолютну погрешність функції, знаючи погрешність аргументу?
- 24 Як можна оцінити граничну відносну погрешність функції?
- 25 Як можна оцінити погрешність алгебраїчної суми?
- 26 У чим особливості погрешностей при вирахуванні близьких величин?
- 27 Як оцінити погрешність добутку й частки?
- 29 Як оцінити погрешність ступеня й кореня?
- 29 Скільки значущих цифр залишають при додаванні й вирахуванні?
- 30 Скільки значущих цифр залишають у результаті множення й розподілу?
- 31 Скільки значущих цифр залишають при зведенні в ступінь і витягу кореня?
- 32 Розрахувати погрешність функції по погрешності аргументу (завдання)
- 33 Як можна визначити погрешність аргументів, якщо задано погрешність функції однієї змінної?
- 34 Для рішення якого завдання використовується метод рівних впливів?
- 35 У результаті чого відбувається втрата точності при обчисленнях на комп'ютерах?
- 36 Які прийоми дозволяють уникнути нагромадження похибок при комп'ютерних обчисленнях?
- 37 У чому полягає сутність комп'ютерного моделювання погрешностей?

## **Завдання для самостійної роботи**

### **Завдання 1.1**

Стовпчик а). Округляючи число в графі А до трьох значущих цифр, визначити абсолютну ( $\Delta$ ) і відносну ( $\delta$ ) погрешність

Стовпчик б). Визначити абсолютну погрешність наближеного числа а, виходячи з його відносної погрешності  $\delta$

Стовпчик в) Визначити кількість вірних знаків у числі х, якщо відома його відносна погрешність  $\delta$

Варіант	А	Б		В	
		а	$\delta, \%$	х	$\delta, \%$
1	1,9583	96099,0000	0,15	18472,0000	0,55
2	7,2101	969,0200	0,76	85476,0000	0,5
3	2,6367	80,0720	0,43	21380,0000	0,21
4	5,5954	714,7800	0,55	53973,0000	0,63
5	7,4857	86,8190	0,77	80358,0000	0,67
6	3,1506	6,9973	0,55	90463,0000	0,25
7	2,7441	686,7900	0,93	66528,0000	0,07
8	2,3599	345,6700	0,55	37252,0000	0,92
9	5,8151	23956,0000	0,56	58172,0000	0,24
10	8,1583	749,8400	0,32	52899,0000	0,67
11	7,0868	5,0714	0,75	20391,0000	0,1
12	1,9025	3,4069	0,26	13565,0000	0,9
13	9,8221	55328,0000	0,33	37036,0000	0,58
14	7,6001	88,3940	0,18	8709,0000	0,34
15	3,7790	313,1700	0,75	94375,0000	0,96
16	4,7229	5144,3000	0,38	68215,0000	0,17
17	7,1353	8,9181	0,5	88421,0000	0,59
18	2,2852	38665,0000	0,19	17855,0000	0,18
19	3,8974	215,5400	0,72	77734,0000	0,1
20	3,8824	3860,0000	0,06	9948,0000	0,75
21	8,9883	3,8119	0,13	53637,0000	0,23
22	8,3746	61,8200	0,3	26058,0000	0,64
23	8,1976	546,1100	0,72	49670,0000	0,39
24	5,8997	1,7236	0,5	98696,0000	0,23
25	5,1807	7028,1000	0,53	62185,0000	0,92
26	3,9069	27536,0000	0,82	48776,0000	0,24
27	2,6996	1824,0000	0,93	3952,0000	0,37
28	2,9001	617,9100	0,15	85812,0000	0,62
29	0,2875	2300,4000	0,5	62631,0000	0,62
30	6,5698	623,3500	0,35	67681,0000	0,61

### Завдання 1.2

Шляхом комп'ютерного моделювання одержати оцінку погрішності функції від 3-х змінних в точці М, вважаючи погрішності аргументів однаковими. Встановити

характер залежності погрішності функції від погрішності аргументів і розрахувати, якою повинна бути погрішність аргументу, щоб погрішність функції була не більше  $10^{-6}$

Варіант	Функція	Координати точки
1	$\ln(x+y) + \sqrt{x+z^2}$	(2;2;1)
2	$\frac{x^{2/3} + y^{2/3} + z^{2/3}}{x \cdot y \cdot z}$	(1;2;1)
3	$\frac{x \cdot y \cdot z}{\sqrt{x+y+z}}$	(1;2;3)
4	$x \cdot \exp(-x-y-z)$	(1;0;1)
5	$x \cdot \exp(-x^2 - y^2) + y \cdot \exp(-y^2 - z^2)$	(1;1;1)
6	$\frac{\text{ch}(x+y+x)}{\text{sh}(x^2+y^2+z^2)}$	(0;0;1)
7	$\sin(x+y-z) \cdot \cos(2 \cdot x-z)$	(0;0; $\pi/4$ )
8	$\cos(2 \cdot \pi \cdot x \cdot y) + \sin(2 \cdot \pi \cdot y \cdot z)$	(0,1;0,1;0,2)
9	$\frac{\exp(x+y)}{\exp(x+z)}$	(1;1;1)
10	$\frac{\exp(x^2+y^2+z^2)}{\exp(x+y+z)}$	(0;1;0)
11	$\frac{\ln(x^2+y^2+z^2)}{\ln(x+y+z)}$	(2;2;1)
12	$\frac{\ln(x+y)}{\ln(x+z)}$	(2;1;1)
13	$\frac{\sqrt[3]{x^2+y^2+z^2}}{\ln(x+y+z)}$	(2;2;1)
14	$\frac{\cos(2 \cdot \pi \cdot x \cdot y) + \sin(2 \cdot \pi \cdot y \cdot z)}{x+y+z}$	(0;1;1)
15	$\frac{x \cdot \exp(-x^2 - y^2) + y \cdot \exp(-y^2 - z^2)}{z \cdot \exp(x+y+z)}$	(1;2;1)
16	$\frac{\text{ch}(x+y+x)}{\sqrt[3]{x^2+y^2+z^2}}$	(1;2;1)
17	$\frac{\ln(x+y) + \sqrt{x+z^2}}{x+y+z}$	(1;1;2)

## Продовження завдання 1.2

Варіант	Функція	Координати точки
18	$\frac{x+y+z}{\sqrt{x \cdot y + \sqrt{x \cdot z + \sqrt{y \cdot z}}}}$	(2;1;3)
19	$\frac{\sqrt{x \cdot y + \sqrt{x \cdot z + \sqrt{y \cdot z}}}}{x+y+z}$	(2;2;3)
20	$\ln(x + \sqrt[3]{y+z+x \cdot z})$	(1;2;3)
21	$\operatorname{ch} \left[ \frac{1+x \cdot y}{\sqrt{1+x \cdot y \cdot z}} \right]$	(2;0;0)
22	$\operatorname{sh} \left[ \frac{1+x \cdot y}{\sqrt{1+x \cdot y \cdot z}} \right]$	(0;0;2)

## ЛЕКЦІЯ 2 СИСТЕМИ ЛІНІЙНИХ АЛГЕБРАЇЧНИХ РІВНЯНЬ: РОЗВ'ЯЗАННЯ МЕТОДОМ ВИКЛЮЧЕННЯ, СТІЙКІСТЬ РОЗВ'ЯЗКІВ

### Мета лекції:

1. Ознайомлення слухачів із задачами, що приводять до систем лінійних алгебраїчних рівнянь, класифікацією та особливостями їх розв'язання на практиці, проблемами, що виникають.

2. Ознайомити з методом Гауса для розв'язання систем та його модифікаціями, алгоритмом Жордана-Гаусса і його використанням для розв'язання систем, розрахунків визначників і обернених матриць.

3. Ознайомитися з теоретичними основами методів оцінки погрішності та стійкості рішень систем алгебраїчних рівнянь, методів практичних комп'ютерних розрахунків сингулярних чисел і чисел обумовленості матриць коефіцієнтів.

### План лекції:

[2.1 Системи лінійних рівнянь алгебраїчних рівнянь і задачі, що приводять до них](#)

[2.2 Розв'язання системи лінійних алгебраїчних рівнянь. Метод Гаусса](#)

[2.3 Обчислення визначника й оберненої матриці методом Жордана-Гаусса](#)

[2.4 Норми і сингулярні числа матриць](#)

[2.5 Оцінка погрішності розв'язку СЛАР](#)

[2.6 Комп'ютерне дослідження і розв'язання СЛАР](#)

[Контрольні запитання](#)

[Завдання для самостійної роботи](#)

Матеріал лекції

## 2.1 Системи лінійних рівнянь алгебраїчних рівнянь і задачі, що приводять до них

Багато задач інженерної практики й обчислювальної математики зводяться до рішення систем лінійних алгебраїчних рівнянь (СЛАР). Це - задачі з області електротехніки, механіки, статистики, економіки та ін. Більшість цих задач пов'язано з виконанням умов балансу: матеріального або енергетичного, при наявності лінійних зв'язків між змінними.

### Приклад 2.1

На перегонку надійшло 1000 кг суміші, що містить 50% речовини А, 40 % речовини В і 10 % речовини С. Після перегонки отримано 400 кг дистилляту й 600 кг кубової рідини. За даними аналізів сумарний зміст А і В у дистилляті становить 90 %, а сумарний зміст У і С у кубовій рідині становить 45 %. Потрібно визначити склад дистилляту й кубової рідини по всіх компонентах. Втратами компонентів зневажити.

### Розв'язання

Укладемо орієнтований граф - схему матеріальних потоків на вході й виходах (рис. 2.1). Позначаємо через  $x_1, \dots, x_6$  вміст компонентів, кг, у складі дистилляту й кубової рідини у відповідності до рис. 2.1

Розраховуємо кількості кожного з компонентів у живильній рідині (рис. 2.1).

Записуємо рівняння для загальної маси дистилляту (2.1) і кубової рідини (2.2):

$$x_1 + x_2 + x_3 = 400, \quad (2.1)$$

$$x_4 + x_5 + x_6 = 600. \quad (2.2)$$

Записуємо рівняння (2.3) і (2.4) - баланс мас компонентів А і В у дистилляті й кубі, сума дорівнює масі в живильній рідині:

$$x_1 + x_4 = 500, \quad (2.3)$$

$$x_2 + x_5 = 400. \quad (2.4)$$

Записуємо рівняння (2.5) і (2.6), що виражають співвідношення компонентів у дистилляті й кубовій рідині:

$$x_1 + x_2 = 400 \cdot 90\% = 360, \quad (2.5)$$

$$x_5 + x_6 = 600 \cdot 45\% = 270. \quad (2.6)$$

Одержали систему лінійних алгебраїчних рівнянь (2.1)-(2.6), які містять 6 невідомих, що є необхідною умовою існування єдиного розв'язку.

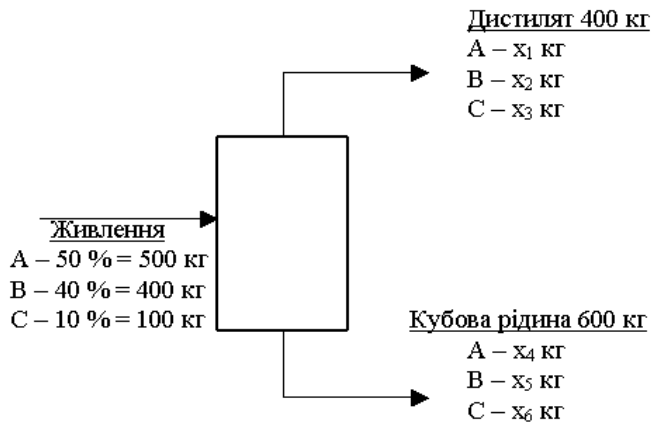


Рисунок 2.1- Схема (граф) матеріальних потоків (розклад 2.1)

## 2.2 Розв'язання системи лінійних алгебраїчних рівнянь. Метод Гаусса

Нагадаємо з курсу алгебри:

а) Матрична форма запису лінійних алгебраїчних рівнянь:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \quad (2.7)$$

где  $\mathbf{A}$  – матриця коефіцієнтів системи,

$\mathbf{x}$  – вектор-стовпчик невідомих,

$\mathbf{b}$  – вектор-стовпчик вільних членів.

Зміст скорочених позначень матриць та векторів:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}; \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}; \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} \quad (2.8)$$

в) розв'язком СЛАР називається вектор  $\mathbf{x}$ , який, після підстановки у вираз (2.7), перетворює його на тотожність;

г) СЛАР має єдиний розв'язок тоді й тільки тоді, коли:

1) кількість невідомих дорівнює кількості рівнянь системи (система, що задовольняє таким умовам, називається замкненою);

2) матриця  $\mathbf{A}$  системи не є виродженою, тобто головний визначник відрізняється від нуля ( $\det(\mathbf{A}) \neq 0$ ). Це є наслідком того, що рівняння системи є лінійно незалежними.

В більшості прикладних задач умова г2) виконується лише *наближено*. Це пов'язано з:

- а) погрішностями обчислень, особливо для систем великої вимірності;
- б) наближеним характером даних і рівнянь, пов'язаних з погрішностями даних і моделі.

Якщо система (2.7) має єдиний розв'язок, те він може бути розрахований методами матричної алгебри, наприклад за формулами Крамера. Однак застосування формул Крамера для рішення системи (2.7) при  $n > 4$  *недоцільно*, оскільки це пов'язано з виконанням занадто великої кількості операцій і значним накопиченням помилок округлення. Тому на практиці системи лінійних алгебраїчних рівнянь розв'язують за допомогою спеціальних чисельних методів.

Методи чисельного розв'язку систем лінійних алгебраїчних рівнянь загального виду можна розділити на дві групи:

- а) кінцево крокові методи, які у свою чергу, можна поділити на:

- 1) гаусівськи (виключення);
- 2) методи розкладання матриць;
- 3) методи для задач з розрідженими матрицями

- б) методи послідовних наближень (ітераційні методи).

В цій лекції буде розглянутий найбільш відомий із крокових методів рішення системи (2.7) - *метод Гаусса* (метод виключення), що лежить в основі більшості алгоритмів чисельних методів лінійної алгебри.

Ідея методу Гаусса:

- а) задана система приводиться до еквівалентній їй системі з *верхньою трикутною матрицею (прямий хід)*;

- б) з перетвореної в такий спосіб системи невідомі розраховують *послідовними підстановками*, починаючи з останнього рівняння перетвореної системи (*зворотний хід*):

$$\begin{array}{c}
 \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix} \\
 \xrightarrow{\text{Прямий хід}} \\
 \begin{pmatrix} a'_{11} & a'_{12} & \dots & a'_{1n} \\ 0 & a'_{22} & \dots & a'_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a'_{nn} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} b'_1 \\ b'_2 \\ \dots \\ b'_n \end{pmatrix} \\
 \downarrow \text{Зворотний хід} \\
 \begin{pmatrix} a''_{11} & 0 & \dots & 0 \\ 0 & a''_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a''_{nn} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} b''_1 \\ b''_2 \\ \dots \\ b''_n \end{pmatrix}
 \end{array}$$

У результаті матриця коефіцієнтів *стає діагональною*. Ділячи вільні члени на діагональні елементи, знаходимо корені рівняння.

При розрахунках вручну або на комп'ютері будують *розширену матрицю* розміром  $n \times n+1$ , шляхом приєднання до матриці коефіцієнтів стовпця вільних членів:

$$\left( \begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & a_{1n+1} \\ a_{21} & a_{22} & \dots & a_{2n} & a_{2n+1} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} & a_{nn+1} \end{array} \right) \quad (2.9)$$

$(a_{i,n+1} = b_i, \quad i = 1, 2, \dots, n)$

В процесі розрахунків відбувається перетворення елементів цієї матриці.

Будемо вважати, що елемент  $a_{11} \neq 0$ . У протилежному випадку завжди можна поміняти рівняння системи місцями, щоб це було так. Для виключення елементів першого стовпця перевіряємо умову:  $a_{i1} \neq 0$ :

- якщо умова не виконується – переходимо до наступного  $i$ ,
- якщо умова виконується:

- ✓ усі елементи  $i$ -го рядку ( $i = 2, 3, \dots, n$ ) множимо на  $-\frac{a_{i1}}{a_{11}}$ , одержуємо  $i$ -те рівняння
- ✓

$$-a_{i1} \cdot x_1 - a_{i2} \cdot \frac{a_{11}}{a_{11}} \cdot x_2 - a_{i3} \cdot \frac{a_{11}}{a_{11}} \cdot x_3 - \dots - a_{in} \cdot \frac{a_{11}}{a_{11}} \cdot x_n = -a_{in+1} \cdot \frac{a_{11}}{a_{11}}, \quad (2.10)$$

$(i = 2, 3, \dots, n)$

- ✓ додаємо до кожного з одержаних рівнянь *перше рівняння*:

$$\begin{array}{cccccc} a_{11} \cdot x_1 & + & a_{12} \cdot x_2 & & + & a_{1n} \cdot x_n & & a_{1n+1} \\ & & & & \dots + & & & \\ + & & & & & & & \\ -a_{i1} \cdot x_1 & & -a_{i2} \cdot \frac{a_{11}}{a_{11}} \cdot x_2 & & + & -a_{in} \frac{a_{11}}{a_{11}} x_n & & -a_{in+1} \frac{a_{11}}{a_{11}} \\ \hline 0 & \left( a_{12} - a_{i2} \frac{a_{11}}{a_{11}} \right) x_1 & & \left( a_{in} - a_{in} \frac{a_{11}}{a_{11}} \right) x_n & & \left( a_{1n+1} - a_{in+1} \frac{a_{11}}{a_{11}} \right) & & \\ \hline 0 & a'_{i2} & & a'_{in} & & a'_{in+1} & & \end{array}$$

$(i = 2, \dots, n)$

Наприкінці одержуємо систему рівнянь, що містить при  $x_1$  нульові коефіцієнти, тобто, розширену матрицю:

$$\left| \begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & a_{1,n+1} \\ 0 & a'_{22} & \dots & a'_{2n} & a'_{2,n+1} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & a'_{n2} & & a'_{nn} & a'_{n,n+1} \end{array} \right|$$

Далі розглядаємо систему з  $n-1$  рівняння з невідомими  $x_2, \dots, x_n$  і виконуємо аналогічні операції. Одержимо перетворену матрицю з нульовими коефіцієнтами при невідомих  $x_2$ . Подібні перетворення проводять  $n-1$  раз, поки система не прийме форму з верхньою трикутною матрицею:

$$\left| \begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1,n-1} & a_{1n} & a_{1,n+1} \\ 0 & a_{22} & & a'_{2,n-1} & a'_{2n} & a'_{2,n+1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a'_{n-} & a'_{n-} & a'_{n-} \\ & & & 1,n-1 & 1,n & 1,n+1 \\ 0 & 0 & \dots & 0 & a'_{n,n} & a'_{n,n+1} \end{array} \right| \quad (2.10)$$

На цьому закінчується прямий хід

На зворотному ході послідовно розв'язуються, починаючи з  $n$ -го, рівняння системи (2.10)

$$\left\{ \begin{array}{l} a'_{n,n} \cdot x_n = a'_{n,n+1}; \quad x_n = \frac{a'_{n,n+1}}{a'_{n,n}} \\ a'_{n-1,n-1} \cdot x_{n-1} + a'_{n-1,n} \cdot x_n = a'_{n-1,n+1} \quad x_{n-1} = \frac{a'_{n-1,n+1} - a'_{n-1,n} \cdot x_n}{a_{n-1,n-1}} \\ \dots \\ \sum_{i=1}^n a_{1,i} \cdot x_i = a_{1,n+1}; \quad x_1 = \frac{a_{1,n+1} - \sum_{i=2}^n a_{1,i} \cdot x_i}{a_{1,1}} \end{array} \right. \quad (2.11)$$

Якщо на черговому етапі виключаються елементи  $i$ -того стовпця, бажано, щоб діагональний елемент  $a_{i-1,i-1}$  був найбільшим у рядку  $i-1$ , що віднімається. Цього можна добитися шляхом перестановки рівнянь системи, з яких відбувається виключення. (крім, бути може, випадку, коли залишаються 2 рівняння). Діагональна перевага приводить до підвищення стійкості обчислювального процесу.

У процесі розв'язання систем методом Гауса можливі 2 випадки:

а) усі елементи, що стоять на головній діагоналі трикутної розширеної матриці після прямого ходу є відмінними від 0 або *машинного нуля* ( $10^{-12} - 10^{-15}$ ). В цьому випадку, згідно (2.11), одержуємо єдиний розв'язок системи.

б) у процесі розв'язання один або декілька елементів головної діагоналі приймають значення нуля або машинного нуля. В цьому випадку система *не має розв'язку*. Таким чином, можливість існування розв'язку з'ясовується безпосередньо у процесі розрахунків і не вимагає розрахунків визначнику.

Найбільш часто на практиці реалізують варіант метода Гауса - *схему повного виключення (метод Жордана-Гауса)*. Вона полягає в одночасному виключенні якого-небудь змінного із всіх рівнянь системи, *крім одного*.



### ГАУСС КАРЛ ФРІДРІХ (1777-1855)

Німецький математик, навчався і працював у брауншвейзькому і Гетингенському університетах. Вважається одним з велетенських мате-матиків світу. Зробив величезний внесок у більшість розділів математики зокрема в чисельні методи. Зробив значний внесок у фізику, зокрема в теорію магнетизму. На його честь названо одиницю виміру напруженості магнітного поля.

**Жордан (Йордан) Вільгельм**  
(1842 - 1899) – німецький геодезист. Запропонував варіант схеми розв'язання системи лінійних рівнянь Гауса (схема Йордана-Гауса)

Алгоритм:

а) на першому кроці цього методу виберемо *провідний елемент*  $a_{11}$  (перестановкою рівнянь системи можна домогтися того, що  $a_{11}$  буде найбільшим по модулю коефіцієнтом при  $x_1$ );

б) розділимо перше рівняння системи на  $a_{11}$ , у всіх інших рівняннях виключимо  $x_1$  й зведемо розширену матрицю системи до виду:

$$\left\| \begin{array}{ccccc} 1 & a_{12}^{(1)} & \dots & a_{1n}^{(1)} & a_{1n+1}^{(1)} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} & a_{2n+1}^{(1)} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} & a_{nn+1}^{(1)} \end{array} \right\| \quad (2.10)$$

де  $a_{ij}^{(1)} = \frac{a_{ij}}{a_{11}}$ ;  $a_{ij}^1 = a_{ij} - a_{ij}^{(1)} \cdot a_{i1}$ ,  $i = 2, 3, \dots, n$ ;  $j = 2, 3, \dots, n+1$ ;

в) на другому кроці виберемо провідний елемент (перестановкою рівнянь системи можна домогтися того, що  $a_{22}^{(1)}$  буде найбільшим за модулем коефіцієнтом при  $x_2$ );

г) Розділимо друге рівняння на  $a_{22}^{(1)}$ , виключимо  $x_2$  із всіх рівнянь, крім другого:

$$\left\| \begin{array}{ccccc} 1 & 0 & \dots & a_{1n}^{(2)} & a_{1n+1}^{(2)} \\ 0 & 1 & \dots & a_{2n}^{(2)} & a_{2n+1}^{(2)} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{nn}^{(2)} & a_{nn+1}^{(2)} \end{array} \right\| \quad (2.11)$$

де  $a_{2j}^{(2)} = \frac{a_{2j}^{(1)}}{a_{22}^{(1)}}$ ,  $a_{ij}^{(2)} = a_{ij}^{(1)} - a_{2j}^{(1)} \cdot a_{i2}^{(1)}$ ,  $i = 1, 3, \dots, n$ ;  $j = 3, \dots, n+1$ ;

д) після  $n$  кроків отримаємо розширену діагонального виду:

$$\left\| \begin{array}{cccc|c} 1 & 0 & \dots & 0 & a_{1n+1}^{(n)} \\ 0 & 1 & \dots & 0 & a_{2n+1}^{(n)} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & a_{nn+1}^{(n)} \end{array} \right\| \quad \begin{array}{l} (2.1 \\ 2) \end{array}$$

е) розраховуємо числові значення невідомих:

$$x_i = a_{i,n+1}^{(n)}, \quad i = 1, 2, \dots, n \quad (2.13)$$

Як у випадку методу Гаусса, алгоритм Жордана-Гаусса становиться стійким в разі, коли провідний елемент є *максимальним за модулем*.

Процес Жордана-Гаусса можна формалізувати й зробити більш наочним таким чином:

а) в системі обирають провідний елемент  $a_{pk}$ , максимальний за модулем. Рядок  $p$  називається *опорним рядком*, стовпець  $k$  – *опорним стовпцем*;

б) для усіх  $i = 1, 2, \dots, n$ ;  $j = 2, \dots, n$  розрахунок елемента  $(a_{ij})_{\text{нов.}}$  перетвореної розширеної матриці проводять за таким алгоритмом:

1) якщо  $i = p$ , (перетворення опорного рядку), те:

$$(a_{ij})_{\text{нов.}} = \frac{(a_{ij})_{\text{стар.}}}{(a_{pk})_{\text{стар.}}} \quad (j = 1, 2, \dots, n+1). \quad (2.14)$$

При цьому  $(a_{pk})_{\text{нов.}} = 1$ ;  
 2) якщо  $i \neq p, j \neq k$ , те:

$$(a_{ij})_{\text{нов.}} = (a_{ij})_{\text{стар.}} - \frac{(a_{ik})_{\text{стар.}} \cdot (a_{pj})_{\text{стар.}}}{(a_{pk})_{\text{стар.}}}; \quad (2.15)$$

3) якщо  $i \neq p, j = k$  те  $a_{ik} = 0$

Формулу (2.14) наочно ілюструвати за правилом прямокутника (рис. 2.1): дії, що відбуваються, відповідають руху в напрямку стрілок (провідний елемент виділено штриховкою).

Наведений алгоритм широко використовується не тільки при розв'язанні СЛАР, а й для розв'язання задач лінійного програмування.

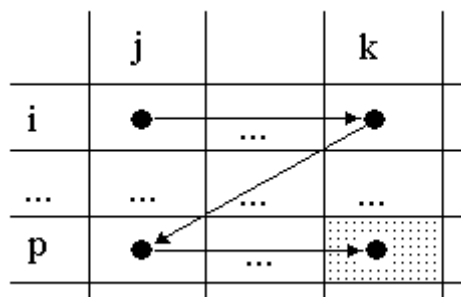


Рисунок 2.2 – Правило прямокутника

Відмітимо наступне. Формально, якщо проводити символні обчислення, вважаючи, що всі діагональні елементи на кожному кроці є провідними, можна отримати формули, що дозволяють розраховувати значення невідомих розв'язків через відомі коефіцієнти. Але, починаючи з системи рівнянь 3-ого ступеня, ці формули стають занадто громіздкими для їх практичного використання, тому усі розрахунки проводять лише чисельно. Це є особливістю будь яких крокових методів розв'язання обчислювальних задач.

Трудоємність методів обчислень, починаючи з „ручних” розрахунків, прийнято оцінювати кількістю арифметичних операцій в залежності від кількості невідомих. Оскільки методи Гаусса є кроковими, для них можна підрахувати точну кількість таких операцій на кожному кроці і для всього алгоритму в цілому. Для методу Гаусса:

а) для прямого ходу необхідно виконати таку кількість операцій<sup>1)</sup>:

1) множення і ділення:

<sup>1)</sup> Пропонуємо самостійно довести вірність сум, що наведені

$$\begin{aligned} N_{\text{мн/д}} &= n \cdot (n+1) + (n-1) \cdot n + \dots + 2 \cdot 3 + 1 \cdot 2 = \\ &= (1^2 + 2^2 + \dots + n^2) + (1 + 2 + \dots + n) = \frac{n \cdot (n+1) \cdot (n+2)}{3}; \end{aligned} \quad (2.16)$$

2) віднімання:  $N_{\text{від}} = N_{\text{мн/д}}$

б) для зворотного ходу: необхідно операцій:

1) множення та ділення:

$$N_{\text{мн/д}} = 1 + 2 + \dots + (n-1) = \frac{n \cdot (n-1)}{2}; \quad (2.17)$$

2) віднімання:  $N_{\text{від}} = N_{\text{мн/д}}$

Загальна кількість операцій:

$$N = \frac{2n(n+1)(n+2)}{3} + n(n-1) \quad (2.18)$$

З (2.18) випливає, що трудомісткість метода Гауса наближене пропорційна кубу вимірності задачі:

$$N \sim O(n^3). \quad (2.19)$$

Усі модифікації методу Гауса для розв'язання СРАР загального виду також мають такий порядок трудомісткості. Тобто, зі зростанням вимірності задачі трудомісткість швидко зростає: зі збільшенням розмірності в 10 разів трудомісткість, як міра часу виконання, збільшується приблизно в 1000 разів.

### 2.3 Обчислення визначника й оберненої матриці методом Жордана-Гауса.

У методі Жордана-Гауса над елементами матриці  $A$  роблять елементарні перетворення, які не змінюють визначника матриці, крім операції розподілу на провідний елемент.

В результаті матриця перетвориться до діагональної виду (2.12), її визначник дорівнює 1.

Якщо в процесі обчислень рядки матриці не переставлялися, то знак визначника не змінюється.

Якщо у процесі обчислень відбувалося  $k$  перестановок рядків, визначник треба помножити на  $(-1)^k$ .

Виходячи з цього, головний визначник матриці коефіцієнтів складає:

$$\Delta \equiv (-1)^k \det(\mathbf{A}) = a_{11} \cdot a_{22}^{(1)} \cdot \dots \cdot a_{nn}^{(n-1)}. \quad (2.20)$$

Одночасно з розв'язанням СЛАР можна розраховувати елементи оберненої матриці  $A^{-1}$ . Для цього будується приєднана матриця (2.21) розміром  $n \times 2n$ . Її отримують шляхом приєднання зліва до матриці коефіцієнтів одиничної матриці:

$$\left\| \begin{array}{cccc|cccc} a_{11} & a_{12} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} & 0 & 0 & \dots & 1 \end{array} \right\| \quad (2.21)$$

Після виконання  $n$  кроків методу Жордана-Гаусса ліва половина приєднаної матриці перетворюється в одиничну матрицю, а права - у обернену матрицю  $A^{-1}$ :

$$\left\| \begin{array}{cccc|cccc} 1 & 0 & \dots & 0 & d_{11} & d_{12} & \dots & d_{1n} \\ 0 & 1 & \dots & 0 & d_{21} & d_{22} & \dots & d_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & d_{n1} & d_{n2} & \dots & d_{nn} \end{array} \right\| \quad (2.22)$$

$$\left\| \begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array} \right\| \times \left\| \begin{array}{cccc} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \dots & \dots & \dots & \dots \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{array} \right\| = \left\| \begin{array}{cccc} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{array} \right\| \quad (2.23)$$

На практиці у більшості комп'ютерних пакетів и систем визначник матриць і обернену матрицю розраховують методом Жордана-Гаусса. Тобто, на відміну від «чистої» алгебри, де розв'язанню СЛАР попереджає розрахунок її головного визначника і зворотної матриці, при чисельних розрахунках усе навпаки: розв'язати систему простіше, ніж розрахувати визначник.

## 2.4 Норми і сингулярні числа матриць

У цьому розділі будуть теоретичні основи питань оцінки точності розв'язання систем лінійних алгебраїчних рівнянь. Ці питання пов'язані з поняттями:

- а) *норми* вектору і матриці,
- б) *обумовленості* матриці.

Розглянемо лінійний простір  $R^n$   $n$ - мірних векторів  $x=(x_1, \dots, x_n)$

Нормою  $\|x\|$  вектора  $x$  називають *дійсне* число, що задовольняє наступним умовам:

- а)  $\|x\| \geq 0$ , причому,  $\|x\| = 0$  *тоді й тільки тоді*, коли  $x = 0$ , (всі компоненти вектору  $x$  дорівнюють нулю).
- б)  $\|c \cdot x\| = |c| \cdot \|x\|$ , де  $c$  – число.
- в)  $\|x + y\| \leq \|x\| + \|y\|$  (нерівність трикутника).

Цим вимогам задовольняє функція (2.24) при будь-якому  $p \geq 1$ :

$$\|x\|_p = \left( \sum_{k=1}^n |x_k|^p \right)^{\frac{1}{p}}. \quad (2.24)$$

Норма такого типу називається *нормою Гельдера з показником  $p$* .

Серед гельдерових норм найпоширенішими є наступні три, які називаються *канонічними нормами*:

а)  $m$ -норма:

$$\|\mathbf{x}\|_m = \max_{1 \leq k \leq n} |x_k|, \quad (2.25)$$

б)  $\ell$  - норма:

$$\|\mathbf{x}\|_\ell = \sum_{k=1}^n |x_k|, \quad (2.26)$$

в)  $k$  – норма (синонім – *евклідова норма*):

$$\|\mathbf{x}\|_k = \sqrt{\sum_{k=1}^n |x_k|^2}. \quad (2.27)$$

Всі норми векторів рівноцінні в наступному: якщо послідовність векторів  $\{\mathbf{x}^k = (x_1^k, \dots, x_n^k)\}$  збігається за деякою нормою до вектора  $\mathbf{x}^0 = (x_1^0, \dots, x_n^0)$ :

$$\lim_{k \rightarrow \infty} \|\mathbf{x}^k - \mathbf{x}^0\| = 0, \quad (2.28)$$

то вона збігаються до вектора  $\mathbf{x}^0$  за будь-якою іншою нормою.

В скінченновимірному нормованому просторі зі *збіжності за нормою витікає збіжність по координатно і навпаки*. При цьому передбачається, що послідовність векторів  $\{x_k\}$  збігається по координатно до вектору  $\mathbf{x}^0$ , якщо для усіх  $i = 1, \dots, n$  виконується:

$$\lim_{k \rightarrow \infty} x_i^k = x_i^0. \quad (2.29)$$



ГЕЛЬДЕР ЛЮДВІГ ОТТО (1859 -1937) – німецький математик, професор Лейпцігського університету. Основні праці – в галузі алгебри, матричного аналізу, основи математики

## Приклад 2.2

Розрахувати канонічні норми вектора  $\mathbf{x} = (-1; 2; 4; -5; 6)$ .

### Розв'язання

а)  $m$  – норма:  $\|\mathbf{x}\|_m = \max(|-1|; 2; 4; |-5|; 6) = 6$

б)  $\ell$ -норма:  $\|\mathbf{x}\|_\ell = (|-1|+2+4+|-5|+6) = 18$

в)  $k$ -норма:  $\|\mathbf{x}\|_k = \sqrt{(-1)^2 + 2^2 + 4^2 + (-5)^2 + 6^2} = \sqrt{82} \approx 9,055$

Нормою матриці  $\mathbf{A}$  називається *дійсне* число  $\|\mathbf{A}\|$ , що задовольняє наступним аксіомам:

а)  $\|\mathbf{A}\| \geq 0$ ,  $\|\mathbf{A}\| = 0$  тоді й тільки тоді, якщо  $\mathbf{A} = \mathbf{0}$  ( $\mathbf{0}$  - нульова матриця);

б)  $\|\alpha \cdot \mathbf{A}\| = |\alpha| \cdot \|\mathbf{A}\|$ , де  $\alpha$  — число;

в)  $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$ .

Норма матриці  $\|\mathbf{A}\|$  називається погодженою з нормою вектора  $\|\mathbf{x}\|$ , якщо для будь-яких  $\mathbf{A}$  і  $\mathbf{x}$ :

$$\|\mathbf{A} \cdot \mathbf{x}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{x}\|. \quad (2.30)$$

С однієї й тією же нормою вектора можуть бути погоджені *різні норми матриць*.

Нехай  $\|\mathbf{A}\|$  - норма матриці, що погоджена із заданою нормою вектора  $\|\mathbf{x}\|$ . Норма  $\|\mathbf{A}\|$  називається підлеглою до нормі вектора  $\|\mathbf{x}\|$ , якщо знайдеться такий вектор  $\mathbf{x} \neq \mathbf{0}$ , що:

$$\|\mathbf{A} \cdot \mathbf{x}\| = \|\mathbf{A}\| \cdot \|\mathbf{x}\|. \quad (2.31)$$

тобто:

$$\|\mathbf{A}\| = \sup_{\|\mathbf{x}\|=1} \|\mathbf{A} \cdot \mathbf{x}\|. \quad (2.32)$$

Тобто, серед всіх норм, погоджених із заданою векторною нормою, *підлегла норма є мінімальною*. Для будь-якої норми вектора є *хоча б одна* підлегла норма матриці.

Будь-яка норма матриці задовольняє нерівностям:

а)  $\|\mathbf{E}\| \geq 1$ , де  $\mathbf{E}$  – одинична матриця,

б)  $\|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\| \geq 1$ , де  $\mathbf{A}$  – невинроджена квадратна матриця <sup>1)</sup>

Сингулярні числа матриць. У несиметричній квадратній матриці властиві значення можуть бути комплексними. Комплексні числа, на відміну від дійсних, не можна порівнювати один з одним за величиною. Але кожен квадратну матрицю можна перетворити в симетричну, якщо помножити ліворуч на транспоновану матрицю, тобто,  $\mathbf{A}^T \cdot \mathbf{A}$  - симетрична матриця. Властиві значення будь якої

<sup>1)</sup> Матриця є винродженою, якщо її визначник дорівнює нулю.

симетричної матриці завжди є дійсними, а для матриці спеціального виду  $A^T \cdot A$  всі властиві значення є позитивними.

Квадратний корінь із  $i$ -го властивого значення матриці  $A^T \cdot A$  називається сингулярним числом матриці  $A$ :

$$\sigma_i = \sqrt{\mu_i}, \quad (i = 1, \dots, n). \quad (2.33)$$

Якщо  $A$  - симетрична матриця, то її сингулярні числа збігаються з модулями властивих значень.

Для матриць можна увести норми, пов'язані з властивими значеннями. Серед них особливе місце займає спектральна норма:

$$\|A\|_{sp} = \max_i \sigma_i. \quad (2.34)$$

Показано, що для симетричної матриці  $A$  серед всіх можливих норм  $\|A\|$ , погоджених з деякою нормою вектора, спектральна норма буде мінімальною за величиною

## 2.5 Оцінка погрішності розв'язання СЛАР

У практичних розрахунках елементи матриці  $A$  и вектора  $\mathbf{b}$  у системі лінійних рівнянь (2.7) є наближеними числами, що відомі з деякими погрішностями. У цьому підрозділі ми встановимо залежність між погрішностями розв'язків СЛАР й властивостями матриці коефіцієнтів  $A$ . Для цього використаємо *метод збурень*: вивчимо, як невеликі збурення вектору вільних членів вплинуть на вектор розв'язку.

Нехай:

а) вектор  $\mathbf{x} + \Delta\mathbf{x}$  – розв'язок збуреної системи:

$$A \cdot (\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b}, \quad (2.35)$$

де  $\Delta\mathbf{b}$  – збурення вектору  $\mathbf{b}$ ;

$\Delta\mathbf{x}$  – збурення точного розв'язку  $\mathbf{x}$ .

Розкриємо дужки в (2.35) і врахуємо, що  $A \cdot \mathbf{x} = \mathbf{b}$ :

$$A \cdot \mathbf{x} + A \cdot \Delta\mathbf{x} = \mathbf{b} + \Delta\mathbf{b}; \quad A \cdot \Delta\mathbf{x} = \Delta\mathbf{b}. \quad (2.36)$$

Розв'яжемо систему (2.36):

$$\Delta\mathbf{x} = A^{-1} \cdot \Delta\mathbf{b}. \quad (2.37)$$

Враховуючи, що для добутку норм справедливо:

$$\|\mathbf{b}\| \leq \|A\| \cdot \|\mathbf{x}\|, \quad (2.38)$$

причому  $\mathbf{b} = \mathbf{A} \cdot \mathbf{x}$ , можна записати:

$$\|\Delta \mathbf{x}\| \cdot \|\mathbf{b}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\| \cdot \|\mathbf{x}\| \cdot \|\Delta \mathbf{b}\|. \quad (2.39)$$

Поділимо обидві частини (2.39) на  $\|\mathbf{b}\| \|\mathbf{x}\|$ , одержимо після скорочення:

$$\delta_x = \frac{\|\Delta \mathbf{x}\|}{\|\mathbf{x}\|} \leq \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\| \cdot \frac{\|\Delta \mathbf{b}\|}{\|\mathbf{b}\|} = \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\| \cdot \delta_b \quad (2.40)$$

З (2.40) виходить, що відносна погрішність рішення  $\delta_x$  є пропорційною відношній погрішності вільних членів;

Коефіцієнт пропорційності:

$$\text{cond}(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\|, \quad (2.41)$$

називається числом обумовленості матриці  $\mathbf{A}$ . Він є *максимально можливим коефіцієнтом підсилення* відношньої погрішності рішення при збурюванні правої частини

Якщо обрано спектральну норму матриці  $\|\mathbf{A}\|_{sp}$ :

$$\|\mathbf{A}\|_{sp} = \max_i \sigma_i, \quad \|\mathbf{A}^{-1}\| = \max_i \frac{1}{\sigma_i} = \frac{1}{\min_i \sigma_i} \quad (2.42)$$

Звідси число обумовленості для спектральної норми має вид:

$$\text{cond}(\mathbf{A}) = \frac{\max_i \sigma_i}{\min_i \sigma_i}. \quad (2.42)$$

З (2.42) випливає, що *погрішність розв'язку СЛАР визначається розкидом сингулярних чисел*:

а) якщо максимальне і мінімальне сингулярні числа розрізняються в межах 1-2 порядків – задача є *добре обумовленою*. Її розв'язок є *стійким*: невеликі збурення вектора вільних членів визивають невеликі збурення вектору розв'язків;

б) е якщо максимальне і мінімальне сингулярні числа розрізняються більше, ніж на 2 порядки – матриця  $\mathbf{B}$  є *погано обумовленою*. Розв'язок СЛАР є *нестійким*: невеликі погрішності вектору вільних членів визивають значні (а інколи – дуже великі) погрішності розв'язків.

З алгебри відомо, що величина визначника квадратної матриці дорівнює добутку його властивих значень. Останні у свою чергу визначають також величини сингулярних чисел. Визначник матриці коефіцієнтів СЛАР дорівнює точно нулю тоді й тільки тоді, коли вектори – рядки матриці є лінійно залежними. Але можливий випадок, коли за рахунок невеликих обурень рядки (рівняння) є «майже

лінійно залежними». У цьому випадку визначник відрізняється від нуля, але є дуже малим за абсолютною величиною. Це приводить до того, що розв'язки СЛАР становиться нестійким настільки, що перестають мати практичний сенс. Тому розв'язку СЛАР у багатьох випадках повинен передувати аналіз матриці коефіцієнтів для виявлення числа обумовленості і «квазілінійної» залежності рівнянь.

В разі виявлення такої залежності для дослідника є 2 можливості:

а) з позиції предметної галузі – проаналізувати вхідну предметну задачу, спробувати спростити її або переформулювати її, щоб зменшити вимірність системи рівнянь;

б) по-новому визначити поняття «розв'язок СЛАР». Цей підхід буде розглянуто у лекції 4

## 2.6 Комп'ютерне дослідження і розв'язання СЛАР

Особливості реальних СЛАР, що зустрічаються на практиці:

а) значна розмірність (від 5 до 100 і вище);

б) нецілочисельні (дробові) коефіцієнти при невідомих

Вважаючи матричну структуру СЛАР, їх зручно записувати і розв'язувати в середовищі табличних процесорів, зокрема Open Office Calc (надалі для скорочення Calc), користуючись арсеналом його матричних функцій:

MDETERM() – для розрахунків визначника,

MINVERSE() – для обернення матриць,

MMULTI() – для матричного множення.

Усі ці функції знаходяться у меню «Масив» Майстра функцій.

Розглянемо етапи комп'ютерного розв'язання СЛАР на прикладі системи рівнянь (2.1)-(2.6) (рис. 2.3).

*Перший етап* – увід розширеної матриці коефіцієнтів і вільних членів. У випадку систем великої вимірності доцільно помічати номери стовпчиків і рядків системи. Увід може бути виконано:

а) в «ручному» режимі (безпосередній увід чисел у комірки),

б) програмно, шляхом створення системи у проекті.

*Другий етап* – розрахунок визначника за допомогою функції MDETERM(). Аргументом є масив коефіцієнтів системи, результат – число. У прикладі на рис. 2.3 формула комірки E10: =MDETERM(C3:H8). Цей етап є обов'язковим: якщо визначник за абсолютною величиною є дуже малим, це вказує на те, що у системі відсутній розв'язок (система є лінійно залежною).

*Третій етап*: побудова оберненої матриці за допомогою функції MINVERSE(). Аргументом є масив коефіцієнтів. Функція повертає масив того ж самого розміру, що й початковий масив. Формула масиву C13:P18 (рис. 2.3): =MINVERSE(C3:H8). Оскільки результат – матриця, для одержання результату необхідно:

✓ спочатку окреслити (виділити) масив, де будуть розташовані елементи оберненої матриці,

- ✓ увести формулу масиву за допомогою Майстра функцій або вручну,
- ✓ після введення формули вручну необхідно натиснути комбінацію клавіш <Ctrl> + <Shift> + <Enter>.

Четвертий етап – одержання вектору-стовпця розв'язку шляхом матричного множення ( $X = A^{-1} \times B$ ) за допомогою функції MMULT(). Функція повертає вектор-стовпець, аргументи – масиви  $A^{-1}$  і  $B$ . Формула масиву K12:K17 (рис. 3.2): =MMULT(C13:H18;I3:I8). Третій і четвертий етапи можна сумістити, якщо скористуватися вкладенням функцій, з формулою масиву K12:K17: =MMULT(MINVERSE(C3:H8);I3:I8)

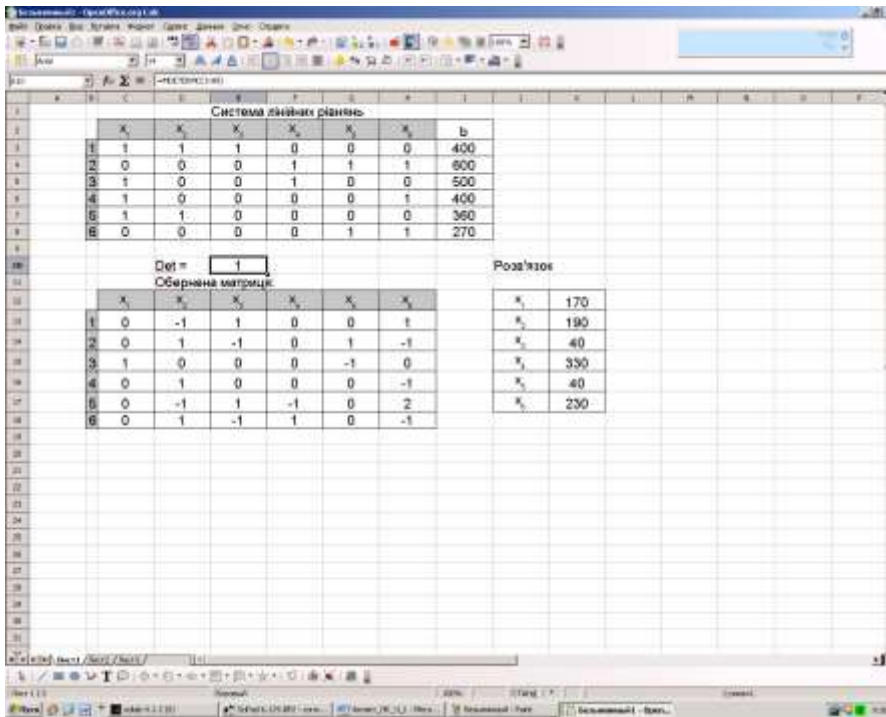


Рисунок 2.3 – Розв'язання СЛАР в Calc

При розрахунках в Calc слід вважати, що за мовчки у комірках відображаються числа лише з двома цифрами після коми. Тому треба завжди встановлювати потрібний формат вхідних і вихідних величин.

Аналогічні розрахунки проводять в середовищі MS Excel, де використовується така ж сама послідовність операцій, з врахуванням, що позначення функцій перекладені на російську мову. Функції-аналоги Excel:

МОПРЕД() = MDETERM();

МОБР() = MINVERSE();

МУМНОЖ() = MMULT.

Розв'язання СЛАР можна проводити в середовищі інших пакетів прикладної математики зокрема в Scilab. Потужний арсенал його матричних функцій дозволяє не тільки розв'язувати системи але й досліджувати стійкість розв'язків за допомогою функції cond(), що розраховує число обусовленості.

### Приклад 2.3

Розв'язати систему рівнянь:

$$\begin{array}{l} \text{У процесі} \\ \text{розрахувати} \end{array} \begin{vmatrix} 1.23 & 3.34 & -1.45 & -4.05 \\ 5.54 & -1.25 & -2.03 & 3.11 \\ - & - & 5.13 & - \\ 0.224 & 0.157 & 0.876 & \\ 0.011 & 0.783 & 0.326 & 7.15 \end{vmatrix} \times \begin{vmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{vmatrix} = \begin{vmatrix} -1.12 \\ 2.34 \\ 0.789 \\ 3.03 \end{vmatrix} \begin{array}{l} \text{розв'язання} \\ \text{число} \end{array}$$

обумовленості і обернену матрицю

Розв'язання

Скрипт Scilab для розв'язання задачі наведено у лістингу 2.3. Результати розрахунків наведені у протоколі 2.3.

#### Лістинг 2.3

```
/Уведення матриці коефіцієнтів
a=[1.23, 3.34, -1.45, -4.05; ...
5.54, -1.25, -2.03, 3.11, ; ...
-0.224, -0.157, 5.13, -0.876; ...
0.011, 0.783, 0.326, 7.15]
b=[-1.12; 2.34; 0.789; 3.03]
disp('Число обумовленості')
c=cond(a)
disp('Обернена матриця')
a_ob=inv(a)
disp('Розв`язок')
x=a_ob*b
```

#### Протокол 2.3

a =

```
1.23  3.34 -1.45 -4.05
5.54 -1.25 -2.03  3.11
-0.224 -0.157  5.13 -0.876
0.011  0.783  0.326  7.15
```

b =

```
-1.12
2.34
0.789
3.03
```

Число обумовленості

$c =$

2.8623267

Обернена матриця

$a_{ob} =$

0.0720974 0.1681097 0.0882657 - 0.0214694  
0.2427445 - 0.0527153 0.0372669 0.1649938  
0.0059724 0.0066172 0.1976673 0.0247224  
- 0.0269663 0.0052125 - 0.0132294 0.1206974

Розв'язок

$x =$

0.3172171  
0.1341072  
0.2396636  
0.3976747

Коментарі до прикладу:

а) в даному прикладі уведення вхідних даних відбувається *безпосереднє в програму*. Це не є оптимальним варіантом у загальному випадку, особливо в разі задач високої розмірності. В останньому випадку слід користуватися таким алгоритмом дій:

1) сформуувати матрицю вхідних даних в середовищі електронних таблиць, де структуру можна побачити наочно,

2) за допомогою макросу записати матрицю даних у зовнішній файл послідовного доступу,

3) зчитати дані з файлу у середовищі Scilab і розв'язати задачу.

При цьому слід мати на увазі, що файли послідовного доступу, що створюються за мовчки в Calc або MS Excel, використовують, як десятинний знак, *кому*. Для Scilab же десятинним знаком потрібна бути *крапка*. Тому у процесі створення файлу послідовного доступу слід забезпечити програмно використання крапки замість коми. Інший спосіб – це заміна коми на крапку в середовищі Блокноту або інших текстових редакторів;

б) для розв'язання системи використовуємо матричні функції Scilab:

$\text{cond}(a)$  – розрахунок числа обумовленості матриці в за спектральною нормою,

$\text{inv}(a)$  – розрахунок оберненої матриці, виходячи з матриці  $a$ .

Аргументами цих функції є квадратні матриці.

## Приклад 2.4

Класифікація чисел обумовленості

Розглянемо систему рівнянь:

$$\begin{vmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3-0,1^k & 5+0,1^k & 7-0,1^k \end{vmatrix} \times \begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix} = \begin{vmatrix} 1 \\ 2 \\ 3+\alpha \end{vmatrix} \quad (2.43)$$

де  $\alpha = 0$  для незбуреного випадку і  $\alpha = 0,01$  для збуреного випадку.

Вивчити вплив величини  $k$  на число обумовленості матриці коефіцієнтів, обурений розв'язок системи і різницю між обуреним і необуреним розв'язками. У даному випадку розв'язок незбуреної системи не залежить від  $k$ . На підставі результатів запропонувати класифікацію чисел обумовленості за величиною.

Розв'язання

В лістингу 2.4 наведено скрипт для виконання розрахунків. Результати розрахунків при  $k = 0 - 3$  наведені у таблиці 2.1.

Лістинг 2.4

```
a=[1, 2, 3; 2, 3, 4; 1, 1, 1];
k=input('k = ')
// уведення елементів останнього рядку
a(3,1)=3-0.1^k;
a(3,2)=5+.1^k;
a(3,3)=7-0.1^k;
b1=[1;2;3]; // незбурений випадок
b2=[1;2;3.01]; // збурений випадок
c=cond(a)
x1=inv(a)*b1//незбурений розв'язок
x2=inv(a)*b2//збурений розв'язок
x=abs(x1-x2); // вектор різниць
z=0;
for i=1:3
    z=z+x(i)^2;
end
z=sqrt(z) // евклідова норма
```

З результатів розрахунків випливає, що із зростанням  $k$  відбувається постійне зростання:

- ✓ числа обумовленості матриці коефіцієнтів,
- ✓ різниці між збуреним і незбуреним розв'язками.

При цьому:

при  $\text{cond} \sim 10 - 10^2$  збурення розв'язку є невеликими за величиною, вони є порівняні зі збуренням правої частини. Цей випадок можна класифікувати, як *добре обумовлену задачу*;

при  $\text{cond} \sim 10^3$  спостерігається значне збурення розв'язку у порівнянні зі збуренням правої частини, але зберігаються вірні знаки розв'язків. Цей випадок можна класифікувати, як *задовільно обумовлена задача*;

при  $\text{cond} > 10^3$  спостерігається не тільки дуже велике збурення розв'язків але й зміна їх знаків. Це є випадок *погано обумовленої системи*.

Таблиця 2.1 – Результати розрахунків задачі (приклад 3.2) для  $k = 0 - 3$

k	cond	$x_1$	$x_2$	z
0	29.466063	0.75	0.7475	0.0061237
		0.5	0.505	
		-0.25	-0.2525	
1	118.69759	0.75	0.725	0.0612372
		0.5	0.55	
		-0.25	-0.275	
2	1189.4533	0.75	0.5	0.6123724
		0.5	1	
		-0.25	-0.5	
3	11898.65	0.75	-1.75	6.1237244
		0.5	5.5	
		-0.25	-2.75	

### Контрольні запитання

- 1 Матричний запис системи лінійних алгебраїчних рівнянь. Сенс позначень.
- 2 Надати визначення розв'язку СЛАР
- 3 У яких випадках система лінійних алгебраїчних рівнянь має єдине рішення?
- 4 Яка матриця називається виродженою?
- 5 Зміст поняття „лінійна залежність векторів”.
- 6 Які класи методів розв'язання СЛАР існують?
- 7 У чому полягає основна ідея методу Гаусса?
- 8 У чому сутність прямого і зворотного ходів у методі Гаусса?
- 9 У чому полягає алгоритм методу Жордана-Гаусса?
- 10 Матрицю якого виду отримують в результаті перетворень Жордана-Гаусса?
- 11 Перша система має вимірність в 3 рази більше, ніж друга. Як наближено відноситься час розв'язання першої системи на комп'ютері у порівнянні з другою?
- 12 Як можна метод Жордана-Гаусса використати для розрахунків визначника?

- 
- 13 Як можна метод Жордана-Гаусса використати для розрахунків елементів оберненої матриці?
  - 14 Яким умовам задовольняє норма вектора?
  - 15 Як називається і записується норма загального виду?
  - 16 Які існують канонічні норми? Який вид вони мають?
  - 17 Як пов'язана збіжність за однією нормою зі збіжністю за іншою будь якою нормою?
  - 18 Як пов'язані між собою збіжність за нормою і покоординатна збіжність?
  - 19 Дайте визначення норми матриці.
  - 20 Яка норма матриці називається погодженою з нормою вектора?
  - 21 Яка норма називається підлеглою?
  - 22 Яка головна властивість підлеглої норми?
  - 21 Якими властивостями володіють властиві значення матриці виду  $A^T A$ ?
  - 22 Що таке сингулярне число матриці?
  - 23 Як пов'язані сингулярні числа симетричної матриці з її властивими значеннями?
  24. Що таке спектральна норма? Яка її головна властивість?
  - 25 Як ставиться задача оцінки погрішності розв'язку СЛАР?
  - 26 Який метод використовують для оцінки погрішності розв'язку СЛАР? У чому полягає його сутність?
  - 27 Як пов'язана погрішність розв'язку СЛАР з погрішністю вільних членів?
  - 28 Яку роль в визначення величини погрішності розв'язку СЛАР грають сингулярні числа матриці коефіцієнтів?
  - 29 Які СЛАР називають погано обумовленими? Яке це має значення з точки зору погрішності розв'язку?
  - 30 Зміст понять «стійкий і нестійкий розв'язок»
  - 31 Що є причиною нестійкого розв'язку СЛАР?
  - 32 Якими є критерії добре і погано обумовлених задач для СЛАР? Як їх можна обґрунтувати?
  - 33 Як можна розв'язати СЛАР в середовищі електронних таблиць і Scilab?
  - 34 Як можна уводити дані для розв'язання СЛАР в Scilab? Особливості введення через зовнішній файл

### Завдання для самостійної роботи

- а) укласти й відладити програми розв'язання СЛАР методом Гаусса, й Жордана-Гаусса, передбачити у неї розрахунок кількості операцій множення і ділення;
- б) розв'язати за допомогою програм систему рівнянь з таблиці 2.2;
- в) створити проекти розв'язання в середовищі Calc і Scilab, перевірити результати;
- г) визначити число обумовленості матриці коефіцієнтів, визначити, до якого ступеню обумовленості вона відноситься.

Таблиця 2.2 – Системи рівнянь для розв'язання.

№	Матриця коефіцієнтів						$b_i$
1	-	-	-	-	-	-	-
	2,923	7,574	9,179	5,385	4,266	2,300	2,356
	1	9	5	2	7	5	0
	-	-	-	-	-	-	-
	1,978	9,499	4,807	2,926	0,166	7,264	2,607
	0	9	9	4	1	2	7
	-	-	-	-	-	-	-
	4,376	2,239	4,480	1,960	6,842	1,150	3,475
	0	7	0	6	5	3	4
	-	-	-	-	-	-	-
	2,319	3,740	5,096	0,458	9,110	6,046	5,221
	3	7	8	7	8	3	6
	-	-	-	-	-	-	-
	3,518	4,025	6,293	6,218	0,314	7,316	8,342
8	0	9	0	2	3	0	
-	-	-	-	-	-	-	
2,342	9,419	8,946	0,627	1,675	5,522	5,196	
3	6	6	4	3	3	5	
-	-	-	-	-	-	-	
6,812	1,084	7,519	8,780	6,870	3,445	4,663	
4	8	6	2	7	6	7	
-	-	-	-	-	-	-	
5,667	5,152	7,864	9,732	6,817	8,285	4,247	
8	3	8	5	3	4	5	

Продовження табл. 2.2

№	Матриця коефіцієнтів						$b_i$
2	-	-	-	-	-	-	-
	9,855	8,675	9,740	4,352	6,031	0,002	3,880
	9	0	8	0	0	2	7
	-	-	-	-	-	-	-
	0,496	3,788	6,716	6,379	6,040	1,802	8,177
	9	1	1	5	6	1	5
	-	-	-	-	-	-	-
	3,609	7,151	5,216	9,009	6,685	9,235	5,037
3	8	4	1	5	9	8	
-	-	-	-	-	-	-	
7,381	1,780	0,854	4,979	1,230	3,157	3,045	
7	8	9	0	1	9	8	
-	-	-	-	-	-	-	
2,202	6,231	1,949	6,964	6,149	9,529	8,150	
5	1	5	0	8	7	1	
-	-	-	-	-	-	-	
3,759	5,878	6,190	4,106	1,889	5,442	4,929	
4	5	5	4	7	1	2	
3	-	-	-	-	-	-	-
	4,476	2,440	4,909	0,723	0,677	5,538	0,446
	5	6	7	3	1	8	1
	-	-	-	-	-	-	-
	2,465	8,240	0,488	2,381	5,853	2,644	6,736
	7	2	0	0	6	8	8
	-	-	-	-	-	-	-
	0,065	2,630	5,488	8,627	6,185	3,038	2,821
0	0	5	8	1	5	8	
-	-	-	-	-	-	-	
9,310	7,430	8,349	3,239	3,181	5,642	7,821	
6	7	8	5	4	6	0	
-	-	-	-	-	-	-	
6,900	0,413	8,737	8,494	8,551	9,136	3,127	
6	5	4	1	3	0	5	
-	-	-	-	-	-	-	
3,937	9,294	0,977	3,138	2,552	1,865	4,302	
7	5	4	1	2	5	0	

Продовження табл. 2.2

№	Матриця коефіцієнтів						$b_i$
4	-	-	-	-	-	-	-
	6,711	9,426	5,729	8,287	4,207	9,208	9,450
	0	6	4	7	3	5	8
	1,706	1,098	3,091	1,427	0,266	6,137	5,696
	2	1	7	4	6	7	5
	2,778	2,021	6,070	2,247	9,493	3,253	6,837
	9	1	7	3	6	7	7
	9,834	2,460	9,969	8,929	8,862	1,784	7,978
7	1	1	0	9	8	2	
3,204	3,209	5,741	6,541	8,413	4,704	8,574	
0	3	2	1	0	3	7	
5,066	3,694	0,268	1,222	0,570	5,550	1,620	
5	6	8	6	4	9	4	
5	9,864	2,823	0,809	8,942	1,292	7,858	7,319
	1	3	6	4	6	3	7
	0,696	2,013	8,245	3,384	0,135	0,480	7,362
	2	3	0	9	9	8	6
	6,139	0,506	6,197	0,893	6,962	1,120	3,379
	6	4	1	1	5	7	2
	7,140	8,236	4,647	2,866	1,564	8,009	7,298
	3	9	3	8	7	8	0
4,262	6,683	8,233	5,611	2,864	4,004	3,427	
0	1	3	2	8	8	7	
6,248	7,964	8,560	7,375	2,389	1,742	4,691	
6	1	1	2	3	7	5	

Продовження табл. 2.2

№	Матриця коефіцієнтів						$b_i$
6	-	-	-	-	-	-	-
	4,220	5,950	2,417	8,584	5,861	9,374	9,549
	4	4	3	3	8	3	2
	-	-	-	-	-	-	-
	2,834	6,489	8,258	8,056	0,272	5,241	8,293
	7	6	2	0	8	2	9
	-	-	-	-	-	-	-
	4,442	3,409	3,528	3,922	4,595	1,651	6,896
8	2	0	1	8	2	3	
7	-	-	-	-	-	-	-
	5,958	7,077	8,149	9,573	2,782	7,303	1,597
	6	8	4	5	6	2	8
	-	-	-	-	-	-	-
	0,393	8,007	4,006	5,690	0,166	3,487	9,449
	6	5	7	3	4	1	2
	-	-	-	-	-	-	-
	5,272	8,241	2,082	8,127	6,898	6,496	0,614
2	0	8	3	3	5	1	
7	-	-	-	-	-	-	-
	1,289	3,593	6,672	9,785	2,349	1,731	5,910
	2	2	5	3	7	1	4
	-	-	-	-	-	-	-
	9,574	9,227	1,592	4,642	0,693	7,695	1,334
	3	9	5	4	8	7	7
	-	-	-	-	-	-	-
	8,383	7,304	3,556	6,541	4,326	1,240	3,672
6	0	3	1	0	5	7	
7	-	-	-	-	-	-	-
	3,966	4,022	4,457	5,902	6,577	3,895	9,065
	6	5	7	7	3	9	1
	-	-	-	-	-	-	-
	3,887	0,881	6,144	7,493	8,944	6,466	6,119
	3	7	7	1	3	2	9
	-	-	-	-	-	-	-
	4,887	5,064	0,587	1,211	0,078	2,453	4,375
2	2	3	2	7	5	7	

Продовження табл. 2.2

№	Матриця коефіцієнтів						$b_i$
8	-	-	-	-	-	-	-
	2,401	8,519	1,682	2,490	6,317	3,645	2,061
	5	2	2	4	3	9	4
	-	-	-	-	-	-	-
	5,989	0,490	3,774	8,024	4,244	6,801	2,291
	6	8	1	3	0	5	8
	-	-	-	-	-	-	-
	8,178	9,613	8,409	9,701	7,340	6,198	5,788
	4	2	9	2	2	5	1
	-	-	-	-	-	-	-
9,579	5,845	9,205	3,285	3,878	7,188	7,994	
5	4	7	1	6	7	3	
-	-	-	-	-	-	-	
6,167	0,024	3,177	0,941	0,848	8,163	4,687	
6	7	5	9	7	5	4	
-	-	-	-	-	-	-	
6,139	2,001	9,578	9,568	5,783	8,954	8,011	
0	0	6	6	8	7	3	
9	-	-	-	-	-	-	-
	8,362	0,579	0,952	2,850	9,252	7,985	4,229
	7	3	4	3	8	5	8
	-	-	-	-	-	-	-
	8,335	4,547	5,283	8,825	1,601	0,042	7,978
	9	1	3	5	6	2	7
	-	-	-	-	-	-	-
	7,175	4,308	2,917	1,856	6,544	8,943	1,489
	0	1	1	8	9	5	9
	-	-	-	-	-	-	-
9,565	9,320	9,076	1,954	8,261	5,508	2,927	
9	8	8	0	1	2	4	
-	-	-	-	-	-	-	
2,632	4,117	5,145	1,875	6,114	9,092	9,754	
9	3	1	2	3	7	3	
-	-	-	-	-	-	-	
7,794	3,783	7,317	7,646	6,606	0,157	2,224	
3	2	4	5	4	0	0	

Продовження табл. 2.2

№	Матриця коефіцієнтів						$b_i$
10	-						-
	3,388	9,643	2,701	4,725	6,544	3,807	1,533
	6	1	2	1	4	0	8
	-						-
	6,310	0,793	1,668	6,288	8,881	1,797	2,974
	4	1	3	9	2	6	5
	-						-
3,546	4,542	2,859	1,212	4,233	7,247	8,453	
0	2	6	6	7	7	6	
-							-
0,173	0,741	3,854	5,298	1,690	1,750	4,106	
1	1	0	3	8	4	7	
-							-
6,713	9,482	0,128	8,188	6,370	8,180	9,620	
6	3	6	0	6	4	6	
-							-
6,376	8,991	9,450	4,105	7,058	5,854	5,042	
0	2	3	1	9	5	6	
11	-						-
	7,511	6,555	6,982	8,766	9,471	9,935	2,210
	5	7	2	9	3	6	8
	-						-
	7,803	8,427	9,651	5,161	2,346	4,646	8,220
	8	2	3	2	0	9	5
	-						-
6,428	8,508	0,554	9,448	9,508	1,351	2,285	
6	7	1	1	9	5	3	
-						-	
3,035	8,786	6,487	5,959	3,243	4,580	0,354	
9	8	9	6	3	3	2	
-						-	
4,260	5,301	1,079	9,299	7,690	5,995	8,091	
3	8	4	5	1	0	4	
-						-	
2,458	9,428	1,884	6,477	3,022	6,808	9,798	
2	6	4	8	9	6	4	

Продовження табл. 2.2

№	Матриця коефіцієнтів						$b_i$
12	-	-	-	-	-	-	-
	5,975	1,316	6,557	2,186	7,403	0,129	1,441
	2	5	0	6	9	2	1
	-	-	-	-	-	-	-
	6,318	4,730	3,540	8,458	2,229	2,022	1,149
	6	8	5	7	3	0	8
	-	-	-	-	-	-	-
	8,098	2,554	2,271	8,580	8,003	5,785	6,949
2	0	2	0	4	5	9	
-	-	-	-	-	-	-	
3,029	9,008	1,915	9,341	3,988	8,810	1,732	
3	5	6	4	6	8	1	
-	-	-	-	-	-	-	
5,630	7,286	1,930	4,955	0,482	3,406	8,544	
3	9	3	8	5	4	8	
-	-	-	-	-	-	-	
2,621	8,104	0,181	4,087	0,382	2,687	6,354	
8	8	9	3	8	0	6	
13	-	-	-	-	-	-	-
	8,840	5,940	9,421	4,616	6,541	2,582	0,076
	0	1	3	1	4	1	2
	-	-	-	-	-	-	-
	2,384	5,149	1,847	0,672	7,329	4,355	5,131
	7	9	9	9	2	5	4
	-	-	-	-	-	-	-
	9,483	2,510	4,478	9,180	2,661	6,640	5,398
9	2	8	5	2	1	2	
-	-	-	-	-	-	-	
7,788	1,510	2,414	1,372	2,358	7,411	6,638	
9	0	0	0	5	4	1	
-	-	-	-	-	-	-	
8,969	2,020	7,711	5,802	0,258	8,353	1,114	
8	6	2	1	8	8	2	
-	-	-	-	-	-	-	
9,369	0,656	6,128	8,489	9,487	7,862	1,085	
6	3	8	6	4	1	3	

## ЛЕКЦІЯ 3. СЛАР. ІТЕРАЦІЙНІ І СПЕЦІАЛЬНІ МЕТОДИ РОЗВ'ЯЗАННЯ

### Мета лекції

1 Ознайомитися з методом прогонки для розв'язання рівнянь з розрідженою матрицею.

2 Ознайомлення із методами трикутного розкладання матриць та їх використанням для розв'язання СЛАР

3 Ознайомлення слухачів з ітераційними методами розв'язання систем лінійних алгебраїчних рівнянь, їх алгоритмами, умовами збіжності.

### План лекції

[3.1 Розв'язання розріджених систем. Метод прогону](#)

[3.2 Розкладання матриць на співмножники. Розкладання Холецького](#)

[3.3 Розкладання методом квадратного кореня](#)

[3.4 Теоретичні основи ітераційних методів розв'язання СЛАР](#)

[3.5 Алгоритми ітераційних методів](#)

[3.5.1 Метод Якобі](#)

[3.5.2 Метод Гаусса-Зейделя](#)

[Контрольні запитання](#)

[Завдання для самостійної роботи](#)

### 3.1 Розв'язання розріджених систем. Метод прогону

В цій лекції буде продовжено огляд методів розв'язання систем лінійних алгебраїчних рівнянь.

У багатьох технічних задачах виникає необхідність розв'язувати системи лінійних рівнянь  $Ax = b$  високої розмірності, але з розрідженою матрицею коефіцієнтів  $A$ , коли більша частина елементів  $a_{ij}=0$ . З погляду практики найбільш важливими системами такого виду є:

а) системи із дводіагональною матрицею:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & 0 & \dots & 0 \\ 0 & a_{2,2} & a_{2,3} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & a_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix} \quad (3.1)$$

загальний вид  $i$ -го рівняння:

$$a_{i,i} \cdot x_i + a_{i,i+1} \cdot x_{i+1} = b_i. \quad (3.2)$$

б) системи с тридіагональною матрицею:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & 0 & \dots & 0 & 0 & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & 0 & a_{n,n-1} & a_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix} \quad (3.3)$$

Системи із тридіагональною матрицею містять наступні елементи, що відмінні від нуля:

- 1) ті, що розташовані на головній діагоналі,
- 2) ті, що розташовані на лінії, яка прилягає до головної діагоналі зверху,
- 3) ті, що розташовані на лінії, яка прилягає до головної діагоналі знизу.

Всі інші елементи дорівнюють нулю

Загальна формула  $i$ -го рівняння системи має вигляд:

$$a_{i,i-1} \cdot x_{i-1} + a_{i,i} \cdot x_i + a_{i,i+1} \cdot x_{i+1} = b_i \quad (3.4)$$

$$(a_{1,0} = 0; x_0 = 0; a_{n,n+1} = 0; x_{n+1} = 0)$$

в) рідше зустрічаються системи з п'ятидіагональною матрицею. Вони побудовані аналогічно системам з тридіагональною матрицею. При цьому відмінні від нуля елементи на діагоналях, що прилягають до головної діагоналі зверху й знизу.

Формально всі описані системи можна розв'язувати методом Жордана-Гаусса, однак це *недоцільно*, оскільки:

- ✓ при виборі найбільшого елемента й перестановках порушується структура матриці коефіцієнтів;
- ✓ необхідне виконання великої кількості операцій множення й ділення ( $\sim n^3$ ), характерних для методу Гаусса.

Для розв'язання таких систем розроблений *метод прогону*, своєрідну модифікацію методу Гаусса, що:

- ✓ враховує специфіку матриці коефіцієнтів (її структуру й наявність великої кількості нульових елементів);
- ✓ дозволяє отримати результат з виконанням меншого числа операцій.

Нижче метод прогону буде розглянутий для тридіагональної матриці.

Специфіка структури тридіагональної матриці *дозволяє організувати особливим образом розміщення системи рівнянь* у пам'яті ЕОМ. Немає необхідності зберігати  $n^2$  елементів матриці коефіцієнтів, більша частина з яких нулі. Досить зберігати чотири вектори розміру  $n$ , що представляють собою :

- a** - вектор піддіагональних елементів,
- b** - вектор діагональних елементів,
- c** - вектор наддіагональних елементів
- d** - вектор вільних членів системи рівнянь

При цьому  $a_1 = 0$  і  $c_n = 0$ . У цих позначеннях  $i$ -те рівняння тридіагональної системи має вид:

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i. \quad (3.5)$$

Систему (3.1) із дводіагональною матрицею можна розглядати, як окремий випадок системи з тридіагональною матрицею коли одна з "побічних" діагоналей (у цьому випадку  $a$ ) дорівнює 0. Загальний вид рівняння для системи (3.5):

$$b_i x_i + c_i x_{i+1} = d_i; \quad i = 1, 2, \dots, n; \quad c_n = 0; \quad x_{n+1} = 0. \quad (3.6)$$

У методі прогону, як гауссівському, можна виділити прямий хід і зворотний хід.

*Прямий хід* методу прогону для системи (3.4) приводить до виключення елементів  $a(i)$ . Тому що  $a_1 = 0$ , перше рівняння системи має вид:

$$b_1 x_1 + c_1 x_2 = d_2. \quad (3.7)$$

Алгоритм:

а) виразимо  $x_1$  через  $x_2$ :

$$x_1 = -\frac{c_1}{b_1} \cdot x_2 + \frac{d_1}{b_1}; \quad (3.8)$$

б) підставимо у друге рівняння системи, отримаємо лінійне рівняння, що зв'язує  $x_2$  та  $x_3$  і т.д.;

в) нехай вже одержали співвідношення:

$$x_i = k_i x_{i+1} + \varphi_i \quad (i = 1, 2, \dots, n). \quad (3.9)$$

г) зменшимо в (3.9) індекс на 1 і підставимо значення  $x_{i-1}$  в  $i$ -е рівняння системи (3.4):

$$a_i k_{i-1} x_i + a_i \varphi_{i-1} + d_i x_i + c_i x_{i+1} = d_i; \quad (3.10)$$

д) виразимо  $x_i$  через  $x_{i+1}$ :

$$x_i = \frac{-c_i}{b_i + a_i k_{i-1}} x_{i+1} + \frac{-a_i \varphi_{i-1} + d_i}{b_i + a_i k_{i-1}}. \quad (3.11)$$

е) порівнюючи члени з однаковими степенями  $x$  в (3.9) і (3.11), одержуємо рекурентні формули для обчислення  $k_i$  і  $\varphi_i$ :

$$k_i = \frac{-c_i}{b_i + a_i k_{i-1}}; \quad \varphi_i = \frac{-a_i \varphi_{i-1} + d_i}{b_i + a_i k_{i-1}}. \quad (3.12)$$

За допомогою формул (3.12) можна послідовно розрахувати пари значень  $k_i$ ,  $\varphi_i$  ( $i = 1, 2, \dots, n$ ):

а) для  $i = 1$   $k_0 = 0$  і  $\varphi_0 = 0$ . Звідси:

$$k_1 = \frac{-c_1}{b_1}; \quad \varphi_1 = \frac{d_1}{b_1} \quad (3.13)$$

б) далі з  $k_1$ ,  $\varphi_1$  розраховуємо  $k_2$ ,  $\varphi_2$ , і т.д.

Після розрахунків усіх значень  $k_i$ ,  $\varphi_i$  ( $i = 1, \dots, n$ ), переходимо до *зворотного ходу*:

а) розраховуємо  $x_n$  за рівнянням (3.9) при  $x_{n+1} = 0$ :

$$x_n = \varphi_n. \quad (3.14)$$

б) Далі, користуючись рівнянням (3.9), послідовно розраховуємо значення  $x_{n-1}$ ,  $x_{n-2}, \dots, x_1$ .

Аналогічно призводить розв'язання систем рівнянь із дводіагональною матрицею. Тут немає необхідності в прямому ході, і корені можна знайти послідовно шляхом зворотного ходу.

Доведено, що якщо в системах із тридіагональною матрицею:

✓ виконується умова діагональної переваги:

$$|b_i| \geq |a_i| + |c_i|, \quad (3.15)$$

✓ й при цьому, виконується хоча б одна строга нерівність, те це забезпечує *стійкість методу прогону щодо похибок округлення*.

- ✓ Для програмної реалізації методу прогону необхідно визначити масиви: невідомих,
- ✓ коефіцієнтів,
- ✓ вільних членів,
- ✓ допоміжних елементів  $k$  і  $\varphi$ .

Далі у відповідність із алгоритмом розраховують елементи масивів  $k$  і  $\varphi$ , а по них - значення  $x_i$ . Для виконання цих дій необхідно виконати таку кількість операцій множення й ділення:

а) на прямий хід:

- на обчислення  $k_1$ : 1;
- на обчислення  $k_2, \dots, k_n$ :  $(n-1) \cdot 2$ ;
- на обчислення  $\varphi_1$ : 1;
- на обчислення  $\varphi_2, \dots, \varphi_n$ :  $(n-1) \cdot 3$ ;

б) на зворотний хід:  $(n-1)$ .

Разом в сумі:  $2 + 6 \cdot (n-1)$  операцій множення и ділення. У порівнянні, для метода Гаусса кількість операцій множення и ділення складає:

$$N_{on} = \frac{n(n+1)(n+2)}{3} \quad (3.16)$$

Отже трудомісткість методу прогону зростає лінійно з ростом розмірності, у той час як методу Гаусса - пропорційно кубу вимірності.

### Приклад 3.1

Розв'язати систему з тридіагональною матрицею:

$$\begin{vmatrix} 3 & 4 & 0 & 0 & 0 & 0 \\ 1 & 6 & 2 & 0 & 0 & 0 \\ 0 & 3 & 5 & 1 & 0 & 0 \\ 0 & 0 & 2 & 3 & 2 & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 3 & 2 \end{vmatrix} \times \begin{vmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{vmatrix} = \begin{vmatrix} 5 \\ 1 \\ 4 \\ 2 \\ 3 \\ 1 \end{vmatrix} \quad (3.17)$$

Розв'язання

Згідно з позначеннями формули (3.9):

$$\mathbf{a} = (0; 1; 3; 2; 1; 3)^T;$$

$$\mathbf{b} = (3; 6; 5; 3; 4; 2)^T;$$

$$\mathbf{c} = (4; 2; 1; 2; 1; 0)^T;$$

$$\mathbf{d} = (5; 1; 4; 2; 3; 1)^T.$$

Розрахунки виконуємо в середовищі Calc. Скріншот з проектом зображено на рис. 3.1.

Хід виконання:

- а) вводимо на робочий аркуш систему у вигляді приєднаної матриці;

- б) формуємо вектори **a, b, c, d** з врахуванням, що  $a_1=c_6=0$ ;  
 в) розраховуємо  $k_1$  за формулою (3.13). Формула комірки B16:  

$$=-B14/B13$$
;  
 г) розраховуємо  $\phi_1$  за формулою (3.13). Формула комірки B17:  

$$=B15/B13$$
;  
 д) розраховуємо  $k_2$  за формулою (3.12). Формула комірки C16:  

$$=-C14/(C13+C12*B16)$$
  
 е) розраховуємо  $\phi_2$  за формулою (3.12). Формула комірки C17:  

$$=(-C12*B17+C15)/(C13+C12*B16)$$
;

Розширена матриця системи:							
	3	4	0	0	0	0	5
	1	6	2	0	0	0	1
	0	3	5	1	0	0	4
	0	0	2	3	2	0	2
	0	0	0	1	4	1	3
	0	0	0	0	3	2	1

Розв'язання							Перевірка
i	1	2	3	4	5	6	
a <sub>i</sub>	0	1	3	2	1	3	5,000000
b <sub>i</sub>	3	6	5	3	4	2	1,000000
c <sub>i</sub>	4	2	1	2	1	0	4,000000
d <sub>i</sub>	5	1	4	2	3	1	2,000000
k <sub>1</sub>	-1,333333	-0,428571	-0,269231	-0,812500	-0,313725	0,000000	3,000000
φ <sub>1</sub>	1,666667	-0,142857	1,192308	-0,156250	0,990196	-1,861111	1,000000
x <sub>i</sub>	2,759259	-0,819444	1,578704	-1,435185	1,574074	-1,861111	

Рисунок 3.1 – Проект розв'язання СЛАР до прикладу 3.1

- ж) виділіємо комірки C16:C17 і протянемо вправо до стовпця G включно;  
 з) розраховуємо  $x_6$  за формулою (3.14). Формула комірки G18:  $=G17$   
 и) розраховуємо  $x_5$  за формулою (3.9). Формула комірки F18:  

$$=F16*G18+F17$$
;  
 к) копіюємо комірку F18 і вставляємо в комірки D18 – B18;  
 л) проводимо перевірку, розрахуємо скалярні добутки векторів-рядків коефіцієнтів і вектору-рядку x в комірках I12: I17. Формула комірки I12:  

$$=SUMPRODUCT(B4:G4;$B$18:$G$18)$$

Як впливає з рис. 3.1, підстановка одержаного вектору розв'язку в рівняння перетворює його на тотожність, розв'язок виконано вірно.

### 3.2 Розкладання матриць на співмножники. Розкладання Холецького

Далі будуть розглянуті методи розв'язання систем алгебраїчних рівнянь, що засновані на розкладанні матриць на співмножники. Ці методи можна розглядати, як альтернативу гауссівським.

Важлива прикладна задача деяких дисциплін (теорії оптимізації, математичної статистики, прикладної алгебри, фізики й ін.) - розкладання квадратних матриць на добуток двох *трикутних матриць*:

$$A = B \cdot C, \quad (3.17)$$

де  $A = [a_{ij}]$  ( $i = 1, 2, \dots, n, j = 1, 2, \dots, n$ ) – квадратна матриця,

$$B = \begin{bmatrix} b_{11} & 0 & \dots & 0 \\ b_{21} & b_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix}; \quad C = \begin{bmatrix} 1 & c_{12} & \dots & c_{1n} \\ 0 & 1 & \dots & c_{2n} \\ 0 & 0 & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}. \quad (3.18)$$

Для знаходження невідомих коефіцієнтів  $b_{ij}$  і  $c_{ij}$  зробимо множення матриць  $B$  і  $C$  за правилами матричного множення:

а) помножимо перший рядок на перший стовпець, одержимо:

$$a_{11} = (b_{11}, 0, \dots, 0) \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = b_{11}. \text{ або } b_{11} = a_{11}; \quad (3.19)$$

б) аналогічно, множачи по черзі 2, ..., n - й рядки матриці  $A$  на перший стовпець матриці  $C$ , одержуємо:

$$b_{i1} = a_{i1}, \quad i = 1, 2, \dots, n; \quad (3.20)$$

в) множачи перший рядок матриці  $B$  по черзі на 1, 2, ..., n - й стовпці матриці  $C$ , одержуємо після перетворень:

$$c_{1j} = \frac{a_{1j}}{b_{11}}; \quad (3.21)$$

г) послідовно множачи стовпці й рядки й виражаючи  $b$  і  $c$ , одержуємо:

$$b_{ij} = a_{ij} - \sum_{k=1}^{j-1} b_{ik} \cdot c_{kj}, \quad (i \geq j > 1); \quad (3.22)$$

$$c_{ij} = \frac{1}{b_{ii}} \left( a_{ij} - \sum_{k=1}^{i-1} b_{ik} \cdot c_{kj} \right), \quad (1 < i < j). \quad (3.23)$$

Таким чином, для одержання елементів матриць розкладання, потрібно:

а) розрахувати елементи першого стовпця матриці В і першого рядка матриці С за формулами (3.20) і (3.21);

б) розрахувати по них елементи другого стовпця матриці В і другого рядка матриці С і т.д. до кінця.

Обчислення істотно спрощуються, якщо матриця А є симетричною У цьому випадку:

$$c_{ij} = \frac{b_{ji}}{b_{ii}}. \quad (3.24)$$

Розкладання (3.17), (3.18) називається розкладанням Холецького. Іноді матрицю В позначають буквою L (англ. Low - нижня), а матрицю С буквою U (англ. Upper - верхня) і називають розкладання (3.17), (3.18) *LU - розкладанням*.

З наведених формул виходить, що, що умовою існування розкладання Холецького є відмінність від нуля діагональних елементів матриці В:  $b_{ii} \neq 0$  ( $i = 1, 2, \dots, n$ ). Більш строго: розкладання Холецького існує *тоді й тільки тоді*, коли *визначник матриці є відмінним від нуля*. Якщо він дорівнює нулю - це приводить до того, що коефіцієнти  $b_{ii}$  дорівнюють нулю. Внаслідок цього відбувається переповнення й зупинка роботи програми при обчисленні коефіцієнтів С. Щоб уникнути цього, необхідно в програму ввести перевірку поточної величини  $b_{ii}$ . Якщо вона виявиться малою - необхідно робити примусову зупинку програми й повідомлення про неможливість розкладання.

### Приклад 3.2

Провести LU-розкладання матриці:

$$\begin{bmatrix} 1 & 2 & 6 & 5 \\ 5 & 3 & 7 & 8 \\ 9 & 11 & 8 & 12 \\ 13 & 14 & 15 & 17 \end{bmatrix}$$

### Розв'язання

Укладаємо в Scilab програму розрахунків. Текст скрипту наведено у лістингу 3.2.

### Лістинг 3.2

```
n=4// вимірність матриці
A=[1,2,6,5;5,3,7,8;9,11,8,12;13,14,15,17]
//Побудова вхідних матриць В і С
V=zeros(n,n); //нульова матриця
C=eye(n,n); //одинична матриця
for i=1:n
    if i ==1 //Перший стовпчик В і перший рядок С
        for j=1:n
            V(j,1)=A(j,1);
```

```

        C(1,j)=A(1,j)/A(1,1);
    end
else
    for j = 1: i
        B(i,j)=A(i,j) ;
        for k=1:j-1
            B(i,j)=B(i,j)-B(i,k)*C(k,j);
        end //k
    end //j
    for j=i+1:n
        C(i,j)=A(i,j);
        for k=1:i-1
            C(i,j)=C(i,j)-B(i,k)*C(k,j);
        end //k
        if abs(B(i,i))<1e-12//Перевірка існування розкладання
            disp('Розкладання не існує')
            stop
        end //if
        C(i,j)=C(i,j)/B(i,i);
    end//j
end//if
end//i
B
C
B*C

```

У програмі реалізовано алгоритм, що наведений у формулах (3.20 – 3.24). При цьому, на кожному кроці виконується перевірка умови існування розкладання:  $\text{abs}(B(i, i)) > 10^{-12}$  («машинний нуль»). Якщо розкладання не існує – відбувається друк повідомлення і зупинка програми за рахунок переходу до невизначеного оператора `stop` і друкування повідомлення про помилку.

Результати роботи програми наведені у протоколі 3.2

### Протокол 3.2

n =

4.

A =

1.	2.	6.	5.
5.	3.	7.	8.
9.	11.	8.	12.
13.	14.	15.	17.

B =

1.	0.	0.	0.
----	----	----	----

```

5. -7. 0. 0.
9. -7. -23. 0.
13. -12. -23.571429 -2.4596273
C =

```

```

1. 2. 6. 5.
0. 1. 3.2857143 2.4285714
0. 0. 1. 0.6956522
0. 0. 0. 1.
ans =

```

```

1. 2. 6. 5.
5. 3. 7. 8.
9. 11. 8. 12.
13. 14. 15. 17.

```

Як впливає з протоколу, внаслідок множення одержаних матриць В і С одержуємо початкову матрицю А. Це свідчить, що розкладання виконано вірно.

Відмітимо, що у Scilab існує функція,  $\text{lu}(A)$ , що виконує LU- розкладання матриці А, але за іншим методом, коли за основу розкладання використана верхня трикутна матриця. Внаслідок цього результат суттєво відрізняється від одержаних у протоколі 3.2. У протоколі 3.2А наведено результат використання функції  $\text{lu}(A)$  для матриці А з протоколу 3.2. Матриці L та U мають зовсім інший вигляд у порівнянні з матрицями В і С (протокол 3.2), але їх добуток також дає вхідну матрицю А.

### Протокол 3.2 А

```

-->[L,U] = lu(A)
U =

```

```

13. 14. 15. 17.
0. -2.3846154 1.2307692 1.4615385
0. 0. 5.3225806 4.2580645
0. 0. 0. 2.4
L =

```

```

0.0769231 -0.3870968 1. 0.
0.3846154 1. 0. 0.
0.6923077 -0.5483871 -0.3212121 1.
1. 0. 0. 0.

```

```

-->L*U
ans =

```

```

1. 2. 6. 5.

```

5. 3. 7. 8.  
9. 11. 8. 12.  
13. 14. 15. 17.

Розкладання Холецкого може бути використано для розв'язання систем лінійних рівнянь, як альтернатива до методу Гаусса, за таким алгоритмом:

а) приведемо розкладання по Холецкому матриці коефіцієнтів рівняння  $Ax = d$ ;

б) з виду матриць  $B$  і  $C$  виходить, що шуканий вектор  $x$  може бути розрахований з ланки рівнянь:

$$By = d; \quad Cx = y; \quad (3.25)$$

в) оскільки матриці  $B$  і  $C$  - *трикутні*, система (3.57) легко розв'язується:

$$y_1 = \frac{a_{1,n+1}}{b_{11}}, \quad (3.26)$$

$$y_i = \frac{1}{b_{ii}} \cdot \left( a_{i,n+1} - \sum_{k=1}^{i-1} b_{ik} y_k \right), \quad (i > 1); \quad (3.27)$$

$$x_n = y_n; \quad (3.28)$$

$$x_i = y_i - \sum_{k=i+1}^n c_{ik} x_k, \quad (i < n) \quad (3.29)$$



ХОЛЕЦЬКИЙ АНДРЕ-ЛУІ  
(1875-1918)

Французький математик, військовий топограф і геодезист. Запропонував метод розв'язання систем лінійних рівнянь шляхом розкладання, завдяки чому розв'язав проблему корегування геодезичної сітки.

### 3.3 Розкладання методом квадратного кореня

Для симетричних матриці, у яких  $a_{ij} = a_{ji}$ , можливо побудувати розкладання виду:

$$A = B^T \cdot B, \quad (3.30)$$

де:

$$B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ 0 & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & b_{nn} \end{bmatrix}; \quad B^T = \begin{bmatrix} b_{11} & 0 & \dots & 0 \\ b_{12} & b_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ b_{1n} & b_{2n} & \dots & b_{nn} \end{bmatrix}. \quad (3.31)$$

Виконаємо множення матриць  $B^T$  і  $B$  и дорівняємо результати до елементів матриці  $A$ , одержимо наступні рівняння:

$$b_{1i}b_{1j} + b_{2i}b_{2j} + \dots + b_{ii}b_{ij} = a_{ij} \quad (i < j), \quad (3.32)$$

$$b_{1i}^2 + b_{2i}^2 + \dots + b_{ii}^2 = a_{ii} \quad (3.33)$$

Звідси послідовно розраховуємо:

$$b_{11} = \sqrt{a_{11}}; \quad (3.34)$$

$$b_{1j} = \frac{a_{1j}}{b_{11}} \quad (j > 1) \quad (3.35)$$

$$b_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} b_{ki}^2} \quad (1 < i \leq n), \quad (3.36)$$

$$b_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} b_{ki} \cdot b_{kj}}{b_{jj}} \quad (j < i), \quad (3.37)$$

Розкладання (3.30) за допомогою (3.34) - (3.37) називають розкладанням *методом квадратного кореня*.

Умова існування розкладання – це *відмінність визначнику матриці від 0*. Подібно до розкладання Холецького, при розкладанні методом квадратного кореня поява  $b_{ii}=0$  викликає нульову величину визначнику і неможливість розкладання.

Особливість методу квадратного кореня — *можливість появи мнимих елементів* матриці  $B$ , що впливає з формули (3.36). Це створює певні складності при програмуванні. Оскільки більшість мов програмування не передбачають використання мнимих і комплексних чисел, тому доводиться окремо програмувати операції з ними. Одне з виключень — система Scilab, де передбачені операції з такими числами.

Доведено, що розкладання квадратного кореня завжди має дійсні елементи, якщо матриця  $A$  є не тільки симетрично, але й позитивно визначеною,

### Приклад 3.3

Розкласти методом квадратного кореня матрицю:

$$\begin{bmatrix} 2 & 3 & 4 & 5 \\ 3 & 1 & 4 & 6 \\ 4 & 6 & 2 & 3 \\ 5 & 2 & 3 & 2 \end{bmatrix}$$

### Розв'язання

Укладаємо в Scilab програму розрахунків. Текст скрипту наведено у лістингу 3.3.

#### Лістинг 3.3

```
n=4
a=[2,3,4,5;3,1,6,4;4,6,2,3;5,2,3,2]
b=zeros(n,n)
for i = 1:n
    for j = 1:i
        if (i==j)&(i==1)
            b(1,1)=sqrt(a(1,1));
        end //if (i==j and i==1)
        b(i,j)=a(i,j);
        if j>1
            for k=1:j-1
                b(i,j)=b(i,j)-b(i,k).*b(j,k) ;
            end //k
        end //if j>1
        if i>j
            b(i,j)=b(i,j)./b(j,j);
        end // if i>j
        if i==j
            b(i,i)=sqrt(b(i,i));
            if abs(b(i,i)) <1e-12
                disp('Роз`вязку немає')
                stop
            end
        end // if i==j
    end //j
end //i
b
```

Подібно до лістингу 3.2, у програмі відбувається контроль за величиною діагонального елемента. Результати розрахунків наведені у протоколі. Там же наведено результат перевірки. Звернемо увагу, що елементи матриці В у даному випадку є мнимими. У цьому випадку для перевірки не можна користуватися функцією транспонування В', бо у випадку мнимих і комплексних чисел відбувається зміна знаку (тобто коли одержано ермітово-супряжену матрицю). Тому

у протоколи використана генерація чисто транспонованої матриці  $D$ , з якою відбувається множення при перевірці.

### Протокол 3.3

```
--> n =
```

```
4.
```

```
a =
```

```
2. 3. 4. 5.
```

```
3. 1. 6. 4.
```

```
4. 6. 2. 3.
```

```
5. 2. 3. 2.
```

```
b =
```

```
1.4142136 0 0 0
```

```
2.1213203 1.8708287i 0 0
```

```
2.8284271 - 4.748D-16i 2.4494897i 0
```

```
3.5355339 2.9398737i 2.857738i 2.5118766
```

```
for i=1:n;for j=1:n;d(i,j)=b(j,i);end;end
```

```
-->b*d
```

```
ans =
```

```
2. 3. 4. 5.
```

```
3. 1. 6. 2.
```

```
4. 6. 2. 3.
```

```
5. 2. 3. 2.
```

Одне з найважливіших практичних застосувань методу квадратного кореня - розв'язання систем лінійних алгебраїчних рівнянь із симетричною позитивно визначеною матрицею<sup>1)</sup>. Такі задачі виникають при використанні методу найменших квадратів у статистиці, при розв'язання систем перевизначених рівнянь та інших прикладних задач.

Нехай необхідно розв'язати систему рівнянь  $Ax = D$ , де  $D$  - симетрична позитивно визначена матриця. При виконанні співвідношення (3.92) ця система еквівалентна двом рівнянням:

$$B \cdot y = d, B \cdot x = y. \quad (3.38)$$

<sup>1)</sup> Симетрична матриця  $A$  називається позитивно визначеною, якщо квадратична форма, що побудована з елементів матриці:

$$R(x) = \sum_{i=1}^n a_{ii} \cdot x_i^2 + 2 \cdot \sum_{i=1}^{n-1} \sum_{j=i+1}^n a_{ij} \cdot x_i \cdot x_j > 0$$

при будь яких значеннях  $x_i \neq 0$

Після розкриття (3.38), одержимо:

$$\begin{cases} b_{11}y_1 = d_1 \\ b_{12}y_1 + b_{22}y_2 = d_2 \\ \dots\dots\dots \\ b_{1n}y_1 + b_{2n}y_2 + \dots + b_{nn}y_n = d_n \end{cases} \quad (3.39)$$

$$\begin{cases} b_{11}x_1 + b_{12}x_2 + \dots + b_{1n}x_n = y_1 \\ b_{22}x_2 + b_{23}x_3 + \dots + b_{2n}x_n = y_2 \\ \dots\dots\dots \\ b_{nn}x_n = y_n \end{cases} \quad (3.40)$$

Звідси послідовно розраховуємо:

$$\begin{cases} y_1 = \frac{d_1}{b_{11}} \\ y_i = \frac{d_i - \sum_{k=1}^{i-1} b_{ki} \cdot y_k}{b_{ii}}, \quad (i > 1) \end{cases} \quad (3.741)$$

$$\begin{cases} x_n = \frac{y_n}{b_{nn}}, \\ x_i = \frac{y_i - \sum_{k=i+1}^n b_{ik}x_k}{b_{ii}} \quad (i < n). \end{cases} \quad (3.42)$$

Метод квадратного кореня - найбільш ефективний метод розв'язання систем нормальних рівнянь у методі найменших квадратів (див. [лекцію 4](#)), він використовує найменше із всіх методів кількість операцій множення й ділення. Хоча в цьому методі використовується витяг кореня, це не викликає істотного збільшення часу виконання.

Якщо матриця коефіцієнтів не є позитивно обумовленою, те, як показано у прикладі 3.3, елементи матриці розкладання можуть бути мнимими. При подальших обчисленнях відбувається ділення на мнине число і утворення дійсних коренів. Використання мнимих чисел не є перешкодою для розв'язання вручну або за допомогою SciLab, але при програмуванні на мовах загального призначення виникають певні проблеми. Тому для розв'язання таких систем краще користуватися іншими методами.

Систему лінійних рівнянь з несиметричною матрицею можна перетворити в систему із симетричною позитивно обумовленою матрицею, якщо помножити ліворуч обидві частини системи  $Ax = D$  на транспоновану матрицю коефіцієнтів:

$$Ax = D \Rightarrow [A^T A]x = [A^T D] \quad (3.43)$$

Для розв'язання перетвореної системи можна використати метод квадратного кореня.

### 3.4 Теоретичні основи ітераційних методів розв'язання СЛАР

Ітераційні методи розв'язання системи:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \quad (3.44)$$

базуються на виборі деякого початкового наближення  $\mathbf{x}^{(0)}$  з наступним його уточненням за рекурентними формулами до одержання наближення  $\mathbf{x}^{(k)}$  із заданою точністю.

Достоїнства:

а) процес ітерацій *легко реалізується* на ЕОМ;

б) процес є *самовиправляємим*, тобто окремі помилки в обчисленнях деякого наближення не впливають на остаточний результат розв'язку.

*Збіжність* ітераційного процесу можна довести, виходячи із загального принципу стислих відображень. Для цього введемо ряд понять,

Множина  $X$  називається *метричним простором*, якщо кожній парі елементів - векторів  $\mathbf{x}, \mathbf{y} \in X$  поставлено у відповідність ненегативне дійсне число  $\rho(\mathbf{x}, \mathbf{y})$  що називається *відстань* і задовольняє наступним аксіомам:

а)  $\rho(\mathbf{x}, \mathbf{y}) \geq 0$ , причому  $\rho(\mathbf{x}, \mathbf{y}) = 0$  тоді й тільки тоді, якщо  $\mathbf{x} = \mathbf{y}$ ;

б)  $\rho(\mathbf{x}, \mathbf{y}) = \rho(\mathbf{y}, \mathbf{x})$  (аксіома симетрії);

в)  $\rho(\mathbf{x}, \mathbf{y}) \leq \rho(\mathbf{x}, \mathbf{z}) + \rho(\mathbf{z}, \mathbf{y})$  для будь-яких елементів  $\mathbf{x}, \mathbf{y}, \mathbf{z} \in X$  (нерівність трикутника).

Елементи метричного простору називаються *точками*.

Елемент  $\mathbf{x}^{(0)}$  метричного простору  $X$  називається *границею послідовності*  $\{\mathbf{x}^{(k)}\}$  точок  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}, \dots$ , що належать  $X$ , якщо послідовність відстаней  $\rho(\mathbf{x}^{(0)}, \mathbf{x}^{(k)})$  збігається до нуля при  $k \rightarrow \infty$ , тобто:

$$\lim_{k \rightarrow \infty} \rho(\mathbf{x}^{(0)}, \mathbf{x}^{(k)}) = 0. \quad (3.45)$$

Послідовність  $\{\mathbf{x}^{(k)}\}$  на множині  $X$ , залежно від вибору *метрики*  $\rho(\mathbf{x}, \mathbf{y})$ , може бути:

- ✓ збіжною;
- ✓ розбіжною.

Послідовність  $\{\mathbf{x}^{(k)}\}$  називається *фундаментальною*, якщо для будь-якого числа  $\varepsilon > 0$  знайдеться такий номер  $N(\varepsilon)$ , що  $\rho(\mathbf{x}^{(k)}, \mathbf{x}^{(m)}) < \varepsilon$  при  $k, m > N(\varepsilon)$ .

Якщо в метричному просторі  $X$  кожна фундаментальна послідовність збігається до деякої межі, що є елементом того ж самого простору, то простір  $X$  називається *повним*.

Нехай  $X$  та  $Y$  - дві довільних множини. Якщо кожному елементу  $x \in X$  поставлений у відповідність *один і тільки один* елемент  $y \in Y$ , то говорять, що на  $X$  задане *відображення (оператор)  $A$*  множини  $X$  в  $Y$ :

$$y = Ax. \quad (3.46)$$

Відображення  $A$  метричного простору  $X$  в себе називається *стискаючим*, або *стиском*, якщо існує таке число  $0 < \alpha < 1$ , що для будь-яких двох точок  $x, y \in X$  виконується нерівність:

$$\rho(Ax, Ay) \leq \alpha \rho(x, y). \quad (3.47)$$

Точка  $x$  називається *нерухомою точкою відображення  $A$* , якщо  $Ax = x$ .

Для рівнянь виду  $Ax = x$  справедлива *теорема про нерухому точку* (або *принцип стислих відображень*), яка була сформульована і доведена видатним українським математиком Стефаном Банахом:

Теорема:

а) будь яке стискаюче відображення, що визначено в повному метричному просторі  $X$ , має одну й тільки одну нерухому точку  $x^*$ ;

б) послідовність  $\{x^{(k)}\}$ , що обумовлена рівністю

$$x^{(k+1)} = Ax^{(k)}, \quad k = 0, 1, 2, \dots, \quad (3.48)$$

сходиться до точки  $x^*$  при будь-якому виборі початкового наближення  $x^{(0)} \in X$ ;

в) має місце оцінка:

$$\rho(x^*, x^{(k)}) \leq \frac{\alpha^k}{1-\alpha} \cdot \rho(x^{(0)}, x^{(1)}). \quad (3.49)$$



### БАНАХ СТЕФАН

(1892 - 1945)

Видатний український математик. Член Польської АН та член-кореспондент АН УРСР. Один з творців сучасного функціонального аналізу та Львівської математичної школи. Вивчені ним лінійні простори, що одержали назву "простір Банаха"; мають велике значення для сучасної математики. У 1944 р. очолив факультет ЛДУ.



Доведення:

а) нехай  $\mathbf{x}^{(0)}$  - будь-яка точка в  $X$ .

б) Оскільки  $A \in$  відображення простору  $X$  у себе, те всі послідовні наближення  $\mathbf{x}^{(k)} \in X$  ( $k = 0, 1, 2, \dots$ ).

в) З умови стиску відображення (3.4) випливає виконання системи нерівностей:

$$\begin{aligned} \rho(\mathbf{x}^{(2)}, \mathbf{x}^{(1)}) &= \rho(A\mathbf{x}^{(1)}, A\mathbf{x}^{(0)}) \leq \alpha \cdot \rho(\mathbf{x}^{(1)}, \mathbf{x}^{(0)}); \\ \rho(\mathbf{x}^{(3)}, \mathbf{x}^{(2)}) &= \rho(A\mathbf{x}^{(2)}, A\mathbf{x}^{(1)}) \leq \alpha \cdot \rho(\mathbf{x}^{(2)}, \mathbf{x}^{(1)}) \leq \alpha^2 \cdot \rho(\mathbf{x}^{(1)}, \mathbf{x}^{(0)}); \\ &\dots\dots\dots \\ \rho(\mathbf{x}^{(k+1)}, \mathbf{x}^{(k)}) &\leq \alpha^k \cdot \rho(\mathbf{x}^{(1)}, \mathbf{x}^{(0)}); \end{aligned} \quad (3.50)$$

г) покажемо, що послідовність  $\{\mathbf{x}^{(k)}\}$  є фундаментальною:

1) нехай  $m > k$

2) з нерівності трикутника (див. 3.1.4 в) випливає:

$$\begin{aligned} \rho(\mathbf{x}^{(m)}, \mathbf{x}^{(k)}) &\leq \rho(\mathbf{x}^{(k+1)}, \mathbf{x}^{(k)}) + \rho(\mathbf{x}^{(k+2)}, \mathbf{x}^{(k+1)}) + \dots + \\ &\rho(\mathbf{x}^{(m)}, \mathbf{x}^{(m+1)}) \leq \\ &\leq (\alpha^k + \alpha^{k+1} + \dots + \alpha^{m-1}) \cdot \rho(\mathbf{x}^{(1)}, \mathbf{x}^{(0)}) = \frac{\alpha^k(1-\alpha^{m-k})}{1-\alpha} \quad (3.51) \\ &\rho(\mathbf{x}^{(1)}, \mathbf{x}^{(0)}) \leq \\ &\leq \frac{\alpha^k}{1-\alpha} \rho(\mathbf{x}^{(1)}, \mathbf{x}^{(0)}); \end{aligned}$$

3) тому що  $0 < \alpha < 1$ , то для будь-якого  $\varepsilon > 0$  існує номер  $N(\varepsilon)$ , такий, що при  $k > N(\varepsilon)$  и  $m > N(\varepsilon)$  буде виконуватися нерівність:

$$\rho(\mathbf{x}^{(m)}, \mathbf{x}^{(k)}) < \varepsilon, \quad (3.52)$$

4) За визначенням послідовність  $\{\mathbf{x}^{(k)}\}$  є фундаментальною;

д) з повноти простору  $X$  випливає існування границі:

$$\mathbf{x}^* = \lim_{k \rightarrow \infty} \mathbf{x}^{(k)}, \mathbf{x}^* \in X; \quad (3.53)$$

е) доведемо, що  $\mathbf{x}^*$  - є нерухомою точкою відображення  $A$ :

1) з нерівності:

$$\rho(A\mathbf{x}^{(k)}, A\mathbf{x}^*) \leq \alpha \rho(\mathbf{x}^{(k)}, \mathbf{x}^*) < \varepsilon, \quad (3.54)$$

впливає безперервність стискаючого відображення;

2) внаслідок цього:

$$A\mathbf{x}^* = A \lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \lim_{k \rightarrow \infty} A\mathbf{x}^{(k)} = \lim_{k \rightarrow \infty} \mathbf{x}^{(k+1)} = \mathbf{x}^*. \quad (3.55)$$

ж) доведемо одиничність нерухомої точки:

1) припустимо, що існує дві нерухомі точки  $\mathbf{x}^* \in X$  і  $\mathbf{y}^* \in X$ ;

2 тоді за визначенням повинні виконуватися умови:

$$Ax^* = x^*; Ay^* = y^*; \quad (3.56)$$

$$\rho(x^*, y^*) = \rho(Ax^*, Ay^*) \leq \alpha \rho(x^*, y^*); \quad (3.57)$$

3) оскільки  $0 < \alpha < 1$ , то  $\rho(x^*, y^*) = 0$ , то  $x^* = y^*$ ;

з) після переходу в нерівності (3.8) перейти до межі при  $m \rightarrow \infty$  одержимо:

$$\rho(x^*, x^{(k)}) \leq \frac{\alpha}{1-\alpha} \rho(x^{(k)}, x^{(k-1)}), \quad (3.58)$$

що й було потрібно довести.

### 3.5 Алгоритми ітераційних методів

На практиці використовують 2 ітераційних методи розв'язання СЛАР:

- ✓ метод простих ітерацій (Якобі);
- ✓ метод Гаусса-Зейделя

#### 3.5.1 Метод Якобі

Перетворимо вихідну систему алгебраїчних рівнянь  $A \cdot x = b$  до виду:

$$x = Bx + C. \quad (3.59)$$

Припустимо, що обрано початкове наближення до точного рішення:

$$x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}). \quad (3.60)$$

Як правило полагають  $x^{(0)} = C$

Послідовні наближення в методі простих ітерацій, починаючи з першого, розраховують за формулою:

$$x^{(k)} = B \cdot x^{(k-1)} + C \quad (k = 1, 2, \dots). \quad (3.61)$$

Ітераційний процес називається *таким, що збігається* до розв'язку  $x$  системи лінійних рівнянь  $A x = b$ , якщо при *будь-якому* виборі початкового наближення  $x^{(0)}$  виконується умова:

$$\lim_{k \rightarrow \infty} \|x^{(k)} - x\| = 0. \quad (3.62)$$

Найчастіше в розрахунках використовують евклідову норму.

## ЯКОБІ КАРЛ ГУСТАВ ЯКОБ (1804–1851)



Німецький математик . З 1826 по 1844 - професор математики Кенігсберзького університету, потім професор Берлінського університету. Основні роботи в галузі математичного аналізу, диференціальних рівнянь, механіки, геодезії, чисельних методів. Ім'я вченого носить відомий функціональний визначник – яacobіан.

Умови збіжності методу простої ітерації:

а) Віднімаючи (3.59) з (3.61), одержимо:

$$\mathbf{x}^{(k)} - \mathbf{x} = \mathbf{B} \cdot (\mathbf{x}^{(k-1)} - \mathbf{x}). \quad (3.63)$$

б) з (3.63):

$$\mathbf{x}^{(k)} - \mathbf{x} = \mathbf{B} \cdot (\mathbf{x}^{(k-1)} - \mathbf{x}) = \mathbf{B}^2 \cdot (\mathbf{x}^{(k-2)} - \mathbf{x}) = \dots = \mathbf{B}^{(k)} (\mathbf{x}^{(0)} - \mathbf{x}); \quad (3.64)$$

в) з (3.62) виходить, що вектор  $\mathbf{x}^{(k)} \rightarrow \mathbf{x}$  при  $k \rightarrow \infty$  тоді й тільки тоді, якщо ступінь матриці  $\mathbf{B}^k$  прагне до нульової матриці при  $k \rightarrow \infty$ . Це можливо тоді й тільки тоді, якщо всі властиві значення матриці  $\mathbf{B}$  за модулем менше одиниці;

г) звідси - необхідна й достатня умова збіжності методу простої ітерації: вона має місце тоді й тільки тоді, коли всі властиві значення матриці  $\mathbf{B}$  за модулем менше 1.

У практичних розрахунках цим критерієм користуються рідко, тому що задача знаходження власних значень є більше складною, чим розв'язання системи. Найчастіше для цього використовують одну з норм матриці. Справедливо достатня умова збіжності: *метод ітерації сходиться до рішення, коли будь-яка погоджена норма матриці  $\mathbf{B}$  менше 1.*

Найчастіше за достатню умова наявності збіжності використовують *діагональну перевагу* коефіцієнтів у вихідній матриці  $\mathbf{A}$ : якщо в будь-якому  $i$ -му рядку матриці коефіцієнтів  $\mathbf{A}$  діагональний елемент є *найбільшим за модулем*:

$$|a_{ii}| > |a_{ij}| \quad (i = 1, \dots, n, j = 1, \dots, n, i \neq j), \quad (3.65)$$

те метод простих ітерацій збігається при *будь-якому* виборі початкового наближення.

На практиці ітераційний процес проводять доти, поки евклидова норма різниці (3.66) не стане менше заданої точності  $\varepsilon$ :

$$L = \sqrt{\sum_{i=1}^n (x_i^{(k)} - x_i^{(k-1)})^2} < \varepsilon. \quad (3.66)$$

### Приклад 3.4

Розв'язати методом ітерацій систему рівнянь:

$$\begin{cases} 4x_1 + 3x_2 - x_3 + x_4 = 5 \\ 2x_1 - 6x_2 - 3x_3 + 2x_4 = 3 \\ -x_1 - x_2 + 5x_3 + 4x_4 = -2 \\ x_2 + x_3 - 3x_4 = 6 \end{cases} \quad (3.67)$$

Розв'язання:

а) у системі переважають діагональні елементи, тому метод ітерації буде збігатися;

б) приведемо систему до виду (3.59) у такий спосіб:

- 1) поділимо кожне рівняння на діагональний елемент
- 2) відокремимо діагональні члени:

$$\begin{cases} x_1 = 0 \cdot x_1 - \frac{3}{4}x_2 + \frac{1}{4}x_3 - \frac{1}{4}x_4 + \frac{5}{4}; \\ x_2 = \frac{1}{3}x_1 + 0 \cdot x_2 - \frac{1}{2}x_3 + \frac{1}{3}x_4 - \frac{1}{2}; \\ x_3 = \frac{1}{5}x_1 + \frac{1}{5}x_2 + 0 \cdot x_3 - \frac{4}{5}x_4 - \frac{2}{5}; \\ x_4 = 0 \cdot x_1 + \frac{1}{3}x_2 + \frac{1}{3}x_3 + 0 \cdot x_4 - 2 \end{cases} \quad (3.68)$$

в) використаємо в якості початкового наближення вектор:

$$\mathbf{x}^{(0)} = \left( \frac{5}{4}; -\frac{1}{2}; -\frac{2}{5}; -2 \right)^T;$$

Подальший хід рішення представлений у проекті Calc (рис. 3.2):

а) у комірки E4:I7 вводимо матрицю з правих частин системи (3.68);

б) у комірки B10:E10 вводимо початкове (нульове) наближення вектору розв'язку;

в) у комірках B11:E11 створюємо вектор першого наближення у матричній формі. Формула масиву:

=TRANSPOSE(MMULT(\$E\$4:\$H\$7;TRANSPOSE(B10:E10)))+TRANSPOSE(\$I\$4:\$I\$7);

г) у комірці F11 розраховуємо суму квадратів відхилень між нульовим і першим наближеннями. Формула комірки: =SUMXMY2(B10:E10;B11:E11);

д) у комірці G11 знаходимо евклідову норму різниці між компонентами вектору наближень по формулі (3.32), як корінь квадратний із вмісту комірки F11;

е) виділяємо комірки B11: G11 і протягуємо униз на 45 рядків

Як виходить з даних рис. 3.2, через 45 ітерацій отримане рішення, що відрізняється від попереднього не більш, ніж на  $10^{-5}$  (за евклідовою нормою).

Розв'язок системи наведено у табл. 3.1.

Таблиця 3.1 – Наближений розв'язок системи (3.67) (округлений до 4-х значущих цифр):

$x_1$	$x_2$	$x_3$	$x_4$
2,8436	-0,9778	1,6723	-1,7685

### 3.5.2 Метод Гаусса-Зейделя

Метод Гаусса-Зейделя є подібним методу простих ітерацій, з тією різницею, що на кожному кроці отримане значення наближення відразу використовується.

Розв'язання СЛАР методом ітерацій

	$x_1$	$x_2$	$x_3$	$x_4$	B
0	0	-3/4	1/4	-1/4	1,25
1	1/3	0	-1/2	1/3	-0,5
2	2/7	1/5	0	-4/5	-0,4
3	0	1/3	1/3	0	-2

k	$x_1$	$x_2$	$x_3$	$x_4$	L2	L
0	1,25	-0,5	-0,4	-2		
1	2,025000	-0,550000	1,475000	-2,300000	4,21E+000	2,05E+000
2	2,606250	-1,329167	1,937500	-1,891667	1,53E+000	1,24E+000
3	3,154167	-1,163889	1,469375	-1,797222	5,58E-001	7,47E-001
4	2,939586	-0,782373	1,751250	-1,898171	2,81E-001	5,30E-001
5	2,748136	-1,028483	1,843632	-1,677041	1,54E-001	3,93E-001
6	2,901813	-1,084601	1,580874	-1,728187	1,07E-001	3,28E-001
7	2,870867	-0,889195	1,640113	-1,834842	4,94E-002	2,72E-001
8	2,786585	-0,974715	1,751075	-1,748684	3,41E-002	1,85E-001
9	2,858229	-1,030240	1,640488	-1,741213	2,04E-002	1,43E-001
10	2,868106	-0,848572	1,643791	-1,788584	8,89E-003	8,94E-002
11	2,821523	-0,864722	1,707985	-1,768260	7,36E-003	8,68E-002
12	2,842803	-1,002905	1,688121	-1,752248	3,76E-003	8,12E-002
13	2,857270	-0,870608	1,853666	-1,778261	2,13E-003	4,62E-002
14	2,836023	-0,862329	1,685666	-1,772204	1,50E-003	3,88E-002
15	2,843556	-0,977783	1,672302	-1,768512	2,00E-003	4,47E-002
16	2,843541	-0,977803	1,672320	-1,768494	1,78E-003	3,58E-002
17	2,843566	-0,877811	1,672286	-1,768494	8,26E-010	2,87E-002
18	2,843556	-0,977794	1,672300	-1,768505	3,87E-010	1,97E-002
19	2,843547	-0,977800	1,672312	-1,768498	2,82E-010	1,68E-002
20	2,843552	-0,977806	1,672303	-1,768496	1,54E-010	1,24E-002
21	2,843554	-0,977799	1,672301	-1,768501	8,09E-011	8,99E-002

Рисунок 3.2 – Розв'язок системи лінійних рівнянь методом простих ітерацій

Алгоритм (для прикладу 3.4)

а) задають нульове наближення;

б) розраховують  $x_1^{(1)}$  з використанням  $x_2^{(0)}, \dots, x_4^{(0)}$ :

$$x_1^{(1)} = 0 \cdot x_1^{(0)} - \frac{3}{4}x_2^{(0)} + \frac{1}{4}x_3^{(0)} - \frac{1}{4}x_4^{(0)} + \frac{5}{4};$$

в) розраховують  $x_2^{(1)}$  з використанням значення  $x_1^{(1)}$ , що отримано на кроці б):

$$x_2^{(1)} = \frac{1}{3}x_1^{(1)} + 0 \cdot x_2^{(0)} - \frac{1}{2}x_3^{(0)} + \frac{1}{3}x_4^{(0)} - \frac{1}{2};$$

г) розраховують  $x_3^{(1)}$  з використанням величин  $x_1^{(1)}$  і  $x_2^{(1)}$ , що отримані на кроках б) і в):

$$x_3^{(1)} = \frac{1}{5}x_1^{(1)} + \frac{1}{5}x_2^{(1)} + 0 \cdot x_3^{(0)} - \frac{4}{5}x_4^{(0)} - \frac{2}{5};$$

д) розраховують  $x_4^{(1)}$  з використанням величин  $x_1^{(1)}, \dots, x_3^{(1)}$ , що отримані на кроках б)- г):

$$x_4^{(1)} = 0 \cdot x_1^{(1)} + \frac{1}{3}x_2^{(1)} + \frac{1}{3}x_3^{(1)} + 0 \cdot x_4^{(0)} - 2$$

е) аналогічні розрахунки проводять на наступних ітераціях, аж до одержання розв'язку за критерієм величини евклідової норми.

Умови збіжності методу Гаусса-Зейделя - такі ж, як в методі простих ітерацій, однак, як правило, він сходиться швидше методу Якобі.

Ітераційні методи не має сенсу застосовувати при розв'язку систем рівнянь із невеликим числом змінних: у цьому випадку метод Жордана-Гаусса є менш трудомісткий і швидше приводить до цілі. Однак, для задач з великою кількістю невідомих (100 і більше) ітераційні методи становляться більше ефективними, оскільки число операцій множення й ділення (як найбільш трудомістких) у методі Жордана-Гаусса зростає пропорційно кубу числа невідомих, а в методі ітерацій або Гаусса-Зейделя - пропорційно квадрату числа невідомих.

Ще одна позитивна риса ітераційних методів - *нечутливість до помилок на окремих стадіях обчислень*. Поява останніх (наприклад, при ручних розрахунках) викликає тільки збільшення кількості ітерацій, але не впливає на кінцевий результат.

Вимога діагональної переваги для збіжності обох методів не є перешкодою в загальному випадку, тому що, у принципі, будь-яку систему можна, шляхом відповідних лінійних перетворень, привести до виду з перевагою діагональних елементів.

Відзначимо: якщо матриця коефіцієнтів є погано обумовленою, це призведе до значних погрішностей рішення ітераційними методами (як правило, вже на стадії приведення до виду з діагональною перевагою). Цей дефект є "невеликовним", і треба розглядати змістовну частину завдання, що приводить до нестійкої системи, для того, щоб зрозуміти причини й постаратися від них позбутися.

Відмітимо, що відомі ще деякі ітераційні методи розв'язування СЛАР, наприклад метод найскоршого спуску або метод релаксації Саусвелла. Їх на практиці використовують порівняно рідко

### Контрольні запитання

- 1 На чому базуються ітераційні методи розв'язання СЛАР?
- 2 У чому полягають достоїнства ітераційних методів у порівнянні з кроковими?
- 3 Дайте визначення поняттю «метричний простір»
- 4 Дайте визначення поняттю «границя послідовності точок у метричному просторі»
- 5 Зміст поняття «метрика»
- 6 Зміст поняття «відображення»
- 7 Зміст поняття «стискаюче відображення»
- 8 Зміст теореми про нерухому точку
- 9 Зміст поняття «фундаментальна послідовність»
- 10 Сутність алгоритму Якобі розв'язання СЛАР
- 12 Сутність алгоритму Гауса-Зейделя розв'язання СЛАР
- 13 Умова збіжності ітераційних процесів розв'язання СЛАР
- 14 Які є необхідні і достатні умови збіжності методу Якобі?
- 15 Які достатні умови збіжності методу Якобі використовують на практиці?
- 16 Критерій кінці розрахунків в ітераційних методах розв'язання СЛАР.
- 17 Чим відрізняються ітерації методу Гауса-Зейделя від ітерацій методу Якобі?
- 18 Як на практиці обирають початкове наближення для ітераційних методів?
- 19 Чи можуть ітераційні методи допомогти одержати розв'язок погано обумовленої СЛАР з малою погрішністю?
- 20 Перелічіть матричні функції Excel
- 21 Алгоритм розв'язання СЛАР в Excel
- 22 Для чого використовується метод прогону?
- 23 Алгоритм методу прогону
- 24 У чому полягають переваги методу прогону у порівнянні з методом Гауса?
- 25 Умови забезпечення стійкості розв'язку СЛАР методом прогону
- 26 Постановка задачі побудови розкладання Холецького
- 27 Алгоритм розкладання Холецького
- 28 Умови існування розкладання Холецького
- 29 Використання розкладання Холецького для розв'язання СЛАР
- 30 Постановка задачі розкладання методом квадратного кореня
- 31 Для яких матриць можливе розкладання квадратного кореня?
- 32 Для яких матриць у методі квадратного кореня виникає необхідність операцій з комплексними числами?
- 33 Умови існування розкладання методом квадратного кореня
- 34 використання методу квадратного кореня для розв'язання СЛАР
- 35 Як можна використати метод квадратного кореня для розв'язання СЛАР з несиметричною матрицею?

36 Для розв'язання яких задач метод квадратного кореня є особливо ефективним?

### Завдання для самостійної роботи

#### Завдання 3.1

Укласти програму розв'язання розрідженої системи рівнянь методом прогонки. За допомогою неї розв'язати систему рівнянь з табл. 3.2

Таблиця 3.2 – Діагональні (a,b,c,) і вільні члени розріджених систем завдання 3.1

Ва р		1	2	3	4	5	6	7	8
1	a	0	3,25	0,06	4,57	1,86	0,93	1,91	8,61
	b	8,73	7	3	8	6	8	3	1
	c	2,25	3,85	6,05	3,93	0,71	1,54	4,87	3,16
	d	5,03	0,97	8,92	4,66	7,84	9,65	7,13	0
2	a	0	2,36	6,67	5,42	9,27	9,07	5,11	1,76
	b	2,22	8,34	6	4	9	4	3	5,31
	c	0,22	4,08	0,89	5,26	6,05	5,51	5,97	1,87
	d	9,63	2,24	0,66	1,33	9,4	0,62	7,63	0
3	a	0	9,85	5,41	4,23	3,29	6,63	2,35	4,59
	b	5,9	2	6	6	19,4	7,78	4	9,16
	c	4,33	9,22	3,62	6,35	9,43	0	7,02	2,24
	d	2,99	4,35	0,42	9,24	9,72	2,09	6,01	0
4		0	8,39	8,02	6,5	9,28	3,62	2,3	1,1
		10,4	23,8			21,7	20,4	17,2	
		2	7	9,77	22,4	1	8	4	9,53
		3,75	6,3	0,12	7,07	9,03	8,76	6,91	5,51
5		7,89	9	4,82	7,56	2,34	7,83	6,77	0
	a	0	9,41	1,7	5,43	8,37	5,12	2,13	9,46
	b	11,4	20,6	10,8		20,4			21,0
	c	2	7	2	12,6	6	7,56	5,1	7
d	8,64	8,38	8,81	2,42	9,23	0,78	1,67	8,98	
	4,97	3,84	5,84	1,1	8,52	8,92	7,65	0	

Продовження табл. 3.2

Ва р		1	2	3	4	5	6	7	8
6	a	0	6,9	6,92	2,85	9,07	0,63	9,9	2,82
	b	7,59	14,5 4	18,1 1	12,6 3	22,1 1	8,01	19,3 2	14,3 5
	c	3,77	5,42	1,56	6,23	8,24	4,63	7,85	6,46
	d	1,28	0,65	1,24	6,51	1,25	6,63	4,7	0
7	a	0	1,18	6,16	7,9	0,03	2,64	0,86	6,71
	b	19,0 8	2,47	19,0 3	17,3 4	19,5 5	14,1 9	4,93	11,6 4
	c	9,85	1,04	5,37	3,15	9,52	5,1	0,03	4,42
	d	8,18	8,68	3,53	9,39	1,56	4,03	3,74	0
8	a	0	4,28	8,75	2,13	6,06	3,49	0,53	8,25
	b	12,2 3	20,1	18,7 5	11,0 7	17,7 1	11,6 6	15,0 4	24,8 5
	c	9,78	6,5	1,04	0,14	3,15	0,76	9,63	8,25
	d	9,92	1,26	1,11	2,6	1,36	9,07	0,15	0
9	a	0	9,29	3,37	5,37	3,98	8,87	3,19	9,96
	b	16,4 3	20,8 7	14,3 1	19,5 6	14,4 7	24,3 6	12,2 4	18,0 5
	c	8,68	2,86	5,47	7,28	4,82	7	2,42	5,36
	d	1,8	9,88	1,11	2,09	0,63	2,59	5,87	0
10	a	0	9,01	1,73	0,44	3,33	1,88	5,29	8,52
	b	15,9 5	14	15,9 6	6,95	19,5 4	15,1 6	15,2 1	23,6 3
	c	8,03	3,16	5,6	6,14	8,49	6,14	3,63	7,7
	d	3,86	2,82	4,66	9,66	0,97	2,64	3,97	0
11	a	0	7,96	2,92	3,87	5,08	3,54	0,93	4,38
	b	14,0 8	14,4 7	8,65	20,6 9	23,8 3	13,6 2	6,58	6,7
	c	8,91	3,68	3,16	8,06	9,11	8,64	4,86	0,66
	d	8,74	4,45	8,39	0,18	0,57	5,24	8,81	0
12	a	0	6,73	1,64	7,46	7,35	7,01	9,05	2,46
	b	9,89	11,7 4	13,8 3	20,2 7	17,5 4	12,0 3	18,4	5,37
	c	5	0,26	2,77	5,03	10	4,77	1,58	0,49
	d	0,9	3,43	5,43	8,04	5,75	7,16	7,73	0

Продовження табл. 3.2

Ва р		1	2	3	4	5	6	7	8
13	a	0	7,42	4,24	5,85	8,68	7,74	2,51	8,29
	b	13,1 7	18,9 5		10,8 3	15,3 9	23,6 4	15,2 4	17,9 9
	c	6,52	4,6	0,25	2,43	5,91	8,3	5,16	1,16
	d	9,31	3,5	8,45	3	8,61	1,73	8,53	0
14	a	0	9,65	4,42	2,9	3,72	8,73	2,68	7,18
	b	15,3 5	29,2 4	17,2 3		20,0 4	22,2 6	13,3 5	10,0 4
	c	6,87	9,87	3,11	3,27	7,49	7,94	8,99	0,31
	d	6,92	8,04	1,27	9,28	1,77	9,15	7,01	0
15	a	0	0,57	5,25	9,28	3,73	0,79	8,77	1,79
	b	12,8 6		14,6 7		11,4 6		10,4 4	12,1 6
	c	7,77	2,38	0,24	0,71	6,98	2,02	0,45	2,55
	d	5,68	6,13	3,53	7,93	8,81	6,56	8,32	0
16	a	0	1,41	1,36	1,67	3,12	8,4	1,01	6,77
	b		10,7 1		18,3 7	11,1 3	25,0 1	15,4 8	24,9 4
	c	1,01	8,53	1,99	9,65	1,05	8,34	4,49	9,41
	d	6,6	6,19	7,88	5,82	4,95	5,06	9,28	0
17	a	0	6,37	2,06	9,03	6,28	5,49	8,64	1,42
	b		14,6 7	13,6 6	16,9 2		25,0 9	13,5 1	14,3 7
	c	3,75	6,07	9,14	1,02	1,68	9,61	2,56	4,16
	d	7,64	9,77	7,24	1,46	7,53	4,74	7,98	0
18	a	0	1,21	7,6	4,7	1,05	5,2	1,12	2,39
	b		18,5 1		19,8 1		13,5 3		
	c	8,11	9,79	2,25	8,32	3,76	2,15	3,7	1,62
	d	3,75	0,93	7,36	0,48	3,16	7,7	10	0
19	a	0	4,95	2,87	0,53	9,99	7,73	2,84	2,39
	b	11,4 4				22,1 7	25,4 4	11,5 9	
	c	2,15	2,07	2,66	6,45	8,92	8,57	1,08	0,86
	d	7,07	0,48	6,81	7,93	1,12	6,61	3,18	0

Продовження табл. 3.2

Ва р		1	2	3	4	5	6	7	8
20	a	0	9,12	9,62	4,86	8,55	1,89	1,7	5,92
	b	9,63	11,1 1	19,5 5	16,5	19,2	3	5,13	20,7 6
	c	4,51	1,52	5,84	8,48	7,58	6,46	2,53	8,51
	d	5,38	4,77	3,06	2,3	1,24	7,42	7,06	0
21	a	0	6,06	0,97	5,09	5,42	0,06	2,72	4,07
	b	12,1 7	14,2 4	11,5 5	14,1 2	21,2	10,7 9	14,0 9	8,16
	c	4,1	0,29	6,82	5,78	8,62	9,94	3,39	3,82
	d	9,38	5,71	8,68	7,94	6,43	0,32	0,72	0
22	a	0	0,47	4,71	5,36	4,85	3,21	2,73	6,62
	b	6,46	15,3 2	11,5 8	11,3	15,9 6	16,9 5	10,8	21,7 3
	c	5,68	8,15	6,3	1,73	1,18	8,42	5,04	8,98
	d	8,21	0,87	2,1	2,16	4,84	6,83	6,01	0

### Завдання 3.2

Укласти програму розв'язання СЛАР методом Холецкого. Розрахувати свій варіант системи (завдання 2). У програмі підрахувати кількість операцій множення та ділення, що необхідно виконати при розв'язання. Порівняти з аналогічними даними для метода Гаусса.

### Завдання 3.3

Укласти програму розв'язання СЛАР методами простих ітерацій і Гаусса-Зейделя. Розв'язати систему рівнянь (табл. 3.3) обома методами, починаючи з того ж самого початкового наближення. Виконати такі дослідження:

- для серії значень  $\varepsilon = 10^{-2} - 10^{-7}$ , розрахувати кількість кроків  $n$ , що необхідні для досягнення точності  $|x_n - x_{n+1}| < \varepsilon$ ;
- побудувати графік залежності  $\varepsilon$  від  $n$  в логарифмічних координатах;
- визначити, який з методів (Гаусса-Зейделя, простих ітерацій) вимагає меншої кількості ітерацій.

Таблиця 3.3 – Системи рівнянь для виконання завдання 3.3

Вар	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	b
1	6,171	0,668	1,590	4,209	3,961	5,495	9,720
	5,214	9,109	4,181	9,093	1,719	7,252	5,810
	2,529	9,239	9,814	8,875	8,991	2,720	0,497
	5,342	8,930	1,849	9,030	4,037	2,454	2,956
	4,724	4,413	6,596	6,492	9,825	9,722	8,219
	5,463	3,902	9,600	5,121	0,677	10,399	9,988
	9,601	9,087	4,085	2,360	3,981	8,971	9,597
2	8,610	5,967	5,481	7,835	2,776	2,294	3,618
	1,293	10,571	2,047	8,925	6,676	4,519	1,045
	8,002	4,955	8,811	0,086	1,608	2,478	4,235
	6,962	3,944	9,976	10,873	8,121	8,809	7,292
	9,371	4,451	4,761	5,140	9,597	2,373	4,249
	-	-	-	-	-	-	-
	-	-	-	-	-	-	-

Продовження табл. 3.3

Вар	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	b
3	-	-	-	-	-	-	-
	9,433	6,525	9,038	4,296	0,660	1,220	5,665
	-	-	-	-	-	-	-
	0,640	8,343	5,046	2,021	8,062	4,920	8,229
	-	-	-	-	-	-	-
	2,691	4,261	10,19 0	- 1,377	- 1,960	- 4,494	- 9,707
	6,052	3,922	1,648	6,190	4,457	2,868	1,331
-	-	-	-	-	-	-	
8,899	7,569	2,919	3,043	9,417	6,292	8,446	
-	-	-	-	-	10,42	-	
1,345	9,185	0,841	0,112	9,459	5	2,420	
4	-	-	-	-	-	-	-
	9,034	8,184	3,145	1,178	3,860	8,710	5,122
	-	-	-	-	-	-	-
	4,010	7,662	6,891	5,525	3,477	5,690	8,995
	-	-	-	-	-	-	-
	0,360	5,141	9,673	3,496	9,455	6,085	3,497
	8,103	7,516	1,667	9,530	9,083	5,946	3,926
-	-	-	-	-	-	-	
1,968	9,674	6,645	6,715	9,910	1,879	7,877	
-	-	-	-	-	-	-	
4,478	2,861	6,982	0,041	6,244	8,811	2,544	
5	-	-	-	-	-	-	-
	9,070	2,511	2,210	7,212	1,719	8,631	0,344
	-	-	-	-	-	-	-
	3,394	8,129	4,819	4,809	6,426	3,061	9,950
	-	-	-	-	-	-	-
	4,847	6,818	7,063	4,059	1,903	6,231	4,836
	-	-	-	-	-	-	-
1,247	8,440	1,787	8,819	4,211	3,754	5,975	
-	-	-	-	-	-	-	
6,965	1,860	9,125	5,137	9,522	7,713	9,685	
-	-	-	-	-	-	-	
2,660	1,977	8,051	1,497	5,097	8,578	8,501	

Продовження табл. 3.3

Вар	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	b
6	-	-	-	-	-	-	-
	6,803	4,592	2,391	1,669	6,406	2,699	1,442
	-	-	-	-	-	-	-
	8,018	9,667	9,661	7,611	8,734	5,807	9,915
	9,351	0,523	9,496	9,083	3,615	8,709	9,890
	2,887	1,267	2,764	-	-	-	-
7	-	-	-	-	10,59	-	-
	4,188	0,327	6,669	3,415	9	9,647	3,846
	-	-	-	-	-	-	-
	4,903	0,102	8,587	3,000	2,933	9,258	7,063
	9,458	2,420	9,372	3,100	0,647	4,370	0,360
	-	-	-	-	-	-	-
8	2,162	9,684	9,550	6,885	8,611	5,403	3,936
	5,358	1,221	8,024	5,357	1,953	7,847	0,438
	-	-	-	-	-	-	-
	9,400	0,064	0,571	9,825	7,120	5,001	7,276
	-	-	-	-	10,35	-	-
	3,541	9,616	8,226	9,162	8	1,409	1,675
8	-	-	-	-	-	-	-
	8,625	1,719	3,458	3,819	4,561	8,667	2,405
	9,028	3,573	8,048	8,203	6,751	2,833	3,837
	-	-	-	-	-	-	-
	1,178	7,317	3,637	7,049	6,320	6,260	5,470
	6,759	4,974	10,08	8,947	6,910	9,364	6,726
	9	-	-	-	-	-	-
0,784	8,226	7,235	9,631	8,695	7,480	7,972	
-	-	-	-	-	-	-	
5,415	5,406	1,226	2,361	6,158	4,219	4,820	
-	-	-	-	-	-	-	
1,754	0,994	5,596	2,964	8,198	8,634	9,667	

Продовження табл. 3.3

Вар	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	b
9	8,157	0,716	4,886	4,408	7,574	1,924	9,027
	7,823	8,587	7,969	0,794	8,306	1,886	9,587
	-	7,901	0,041	9,866	7,285	7,658	9,032
	2,904	-	3,178	9,282	9,485	7,231	2,087
	6,148	-	-	9,342	8,993	9,997	3,395
	4,139	-	-	4,108	5,561	6,684	9,329
10	-	7,414	-	2,486	1,365	5,469	4,374
	6,475	-	9,391	4,564	1,380	8,001	9,312
	8,994	4,707	9,825	0,324	9,334	1,736	8,487
	9,598	-	1,575	6,716	9,636	8,040	5,165
	6,617	8,754	9,676	7,076	9,906	8,183	3,319
	4,914	5,488	6,699	1,986	7,970	8,566	3,792
11	9,907	-	0,538	6,234	6,399	0,399	3,833
	7,318	9,098	6,812	8,809	4,487	1,296	6,451
	7,225	4,938	8,016	1,712	3,283	3,760	1,891
	9,030	-	9,563	7,541	9,937	8,249	9,598
	6,708	8,188	7,100	2,558	9,171	7,427	3,222
	1,134	2,908	3,395	1,792	3,852	4,797	1,021

Продовження табл. 3.3

Вар	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	b
12	-	-	-	-	-	-	-
	8,018	1,944	7,222	5,517	6,998	7,793	8,624
	-	-	-	-	-	-	-
	1,763	8,116	5,061	1,699	2,905	7,821	6,930
	-	-	-	-	-	-	-
	8,795	4,050	9,756	3,828	7,576	2,352	0,830
13	-	-	-	-	-	-	-
	7,663	9,088	7,146	9,354	6,061	4,421	6,673
	-	-	-	-	-	-	-
	5,075	4,355	2,852	4,271	5,710	0,462	4,063
	-	-	-	-	-	-	-
	8,874	1,773	6,923	6,821	0,414	8,986	8,722
14	-	-	-	-	-	-	-
	6,883	1,575	4,528	4,342	0,200	5,970	6,419
	-	-	-	-	-	-	-
	1,132	9,821	7,797	9,692	6,629	7,645	7,639
	-	-	-	-	-	-	-
	2,960	0,160	10,70 6	-	-	-	-
15	-	-	-	-	-	-	-
	0,811	6,756	3,111	10,29 1	-	-	-
	-	-	-	-	-	-	-
	0,481	8,366	8,263	0,641	9,818	9,325	8,988
	-	-	-	-	-	-	-
	1,355	2,137	4,475	0,019	3,810	7,249	4,363
16	-	-	-	-	-	-	-
	10,30 8	-	-	-	-	-	-
	-	-	-	-	-	-	-
	5,592	9,259	8,977	5,771	3,229	6,410	4,451
	-	-	-	-	-	-	-
	6,793	7,051	7,686	1,786	1,481	6,392	6,583
17	-	-	-	-	-	-	-
	5,820	8,079	7,560	8,563	7,884	8,212	0,339
	-	-	-	-	-	-	-
	8,959	9,345	5,699	4,647	9,404	4,030	4,802
	-	-	-	-	-	-	-
	3,567	9,719	2,362	5,700	8,193	10,52 5	- 0,746

Продовження табл. 3.3

Вар	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	b
15	-	-	-	-	-	-	-
	9,167	7,326	8,735	2,178	3,219	2,760	0,008
	-	-	-	-	-	-	-
	8,539	9,527	1,863	7,564	3,499	2,183	9,260
	2,392	8,654	9,590	1,091	3,759	4,085	5,579
	0,564	-	-	-	-	-	-
16	8,253	8,084	0,382	3,188	8,637	7,160	3,514
	-	-	-	-	-	-	-
	5,292	5,226	0,122	0,164	3,764	6,252	2,251
	7,700	7,243	0,325	4,216	3,122	4,639	7,868
	7,393	7,703	7,265	1,825	4,219	3,990	7,881
	-	-	-	-	-	-	-
17	5,600	9,019	9,800	0,664	0,935	2,938	4,112
	2,080	-	-	-	-	-	-
	2,302	7,874	4,874	8,495	5,368	0,719	6,022
	-	1,854	7,188	6,532	7,711	2,773	0,023
	6,256	7,361	3,385	1,043	0,374	7,553	5,527
	9,967	3,624	0,009	1,207	7,577	0,170	5,229
17	6,196	8,255	6,827	6,305	0,621	8,122	0,470
	7,325	6,630	7,442	7,204	3,756	1,275	6,555
	-	-	-	-	-	-	-
	4,001	7,384	3,336	8,174	0,547	6,750	5,771
	-	-	-	-	-	-	-
	0,276	3,570	1,338	2,247	6,754	6,671	5,363
17	-	-	-	-	-	-	-
	3,487	7,555	2,199	8,793	3,388	10,07 1	5,462

## ЛЕКЦІЯ 4 РОЗВ'ЯЗАННЯ ПЕРЕВИЗНАЧЕНИХ Й НЕДОВИЗНАЧЕНИХ СИСТЕМ. РОЗВ'ЯЗАННЯ ПРОБЛЕМИ ВЛАСТИВИХ ЗНАЧЕНЬ

### Мета лекції

1 Ознайомитися зі змістом понять «розв'язок недовизначених й перевизначених систем».

2 Ознайомитись зі змістом понять «коректність по Адамару» і «коректність по Тихонову», сучасними підходами розв'язку некоректних задач на основі методу регуляризації по Тихонову.

3 Ознайомитись з методами псевдо зворотної матриці, методом кісткового розкладання, методом найменших квадратів та їх використанням для розв'язання систем з недовизначеною та перевизначеною матрицями.

4 На прикладі методу обертань Якобі ознайомитися з ітераційними методами знаходження властивих значень і векторів для симетричних матриць.

### План лекції

[4.1 Розв'язання перевизначених систем. Метод найменших квадратів](#)

[4.2 Кістякове розкладання прямокутних матриць](#)

[4.3 Псевдообернена матриця Мура-Пенроуза](#)

[4.4 Розв'язання систем недовизначених алгебраїчних рівнянь](#)

[4.5 Поняття про коректно й некоректно поставлені задачі](#)

[4.6 Обчислення властивих значень і властивих векторів матриць](#)

[Контрольні запитання](#)

[Завдання для самостійної роботи](#)



з точним розв'язком. Отже, псевдорозв'язок узагальнює поняття точного розв'язку.

Задача (4.3) є мінімізаційною. Він розв'язується шляхом диференціювання функціонала  $R$  по  $x_j$  ( $j = 1, 2, \dots, n$ ) і прирівнювання похідних до нуля:

$$\frac{\partial R}{\partial x_k} = 2 \sum_{i=1}^m a_{ik} \cdot \left[ \sum_{j=1}^n a_{ij} x_j - d_i \right] = 0 \quad (k = 1, 2, \dots, n). \quad (4.4)$$

Після скорочення на 2, виконання підсумовування й поділу членів (4.4) одержуємо систему з  $n$  алгебраїчних рівнянь із  $n$  невідомими. Шляхом зіставлення членів можна показати, що ця система має вигляд:

$$A^T \cdot A \mathbf{x} = A^T \cdot \mathbf{d}, \quad (4.5)$$

де  $A$  – прямокутна матриця ( $m \times n$ ) коефіцієнтів системи (4.1):

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad (4.6)$$

Система рівнянь (4.5) називається *нормальною системою*. Властивість матриці системи (4.6): матриця  $A^T A$  є квадратною симетричною матрицею розміру  $n \times n$ . Тому:

- ✓ система (4.5) є замкненою,
- ✓ якщо її визначник відмінний від нуля, вона має *єдиний розв'язок*.

Для одержання коренів системи (4.5) можна використати всі методи розв'язання СЛАР, що описані у лекціях [2](#) і [3](#). Процедура одержання матриці  $A^T A$  є аналогічною описаної в [3.3](#) процедурі одержання системи рівнянь із симетричною матрицею для методу квадратного кореня.

### Приклад 4.1

Знайти псевдорозв'язок системи рівнянь:

$$\begin{cases} 2x_1 + 3x_2 + x_3 = 4 \\ -x_1 + x_2 - x_3 = 3 \\ x_1 + 2x_2 - x_3 = 7 \\ 3x_1 - 4x_2 - x_3 = 1 \\ 5x_1 - x_2 - x_3 = 3 \end{cases} \quad (4.7)$$

Розв'язання в OOo Calc:

а) уводимо рівняння у матричному виді (рис. 4.1)

б) у комірках J6:L8 розраховуємо матричний добуток  $A^T \times A$ . Формула масиву J6:L8: =MMULT(TRANSPOSE(C5:E9);C5:E9);

в) у комірках J13:J15 розраховуємо матричний добуток  $A^T \times B$ . Формула масиву J13:J15: =MMULT(TRANSPOSE(C5:E9);F5:F9)

г) у комірках M13:M15 розраховуємо вектор псевдорозв'язку. Формула масиву M13:M15: =MMULT(MINVERSE(J6:L8);J13:J15);

д) у комірках F20:F24 розраховуємо вільні члени. Формула комірки F20: =MMULT(B20:D20;\$M\$13:\$M\$15), далі – протягується на увесь масив;

е) у комірках G20:G24 розраховуємо різницю між заданими і розрахованими компонентами вектору вільних членів. Формула комірки G20: =(E20-F20);

ж) у комірках H20:P24 розраховуємо квадрати різниць зі стовпчику G, формула комірки H20: =G20^2

з) у комірці H25 розраховуємо суму квадратів нев'язок. Формула комірки: =SUM(H20:H24).

Як виходить з результатів, зображених на скріншоті проекту (рис. 4.1), оптимальний розв'язок приводить до достатньо великого значення суми квадратів відхилень між заданими і розрахованими значеннями елементів вектору вільних членів B-Bc. Це свідчить, що система є погано "спряженою" між собою: розв'язки окремих частин системи істотно відрізняються один від одного.

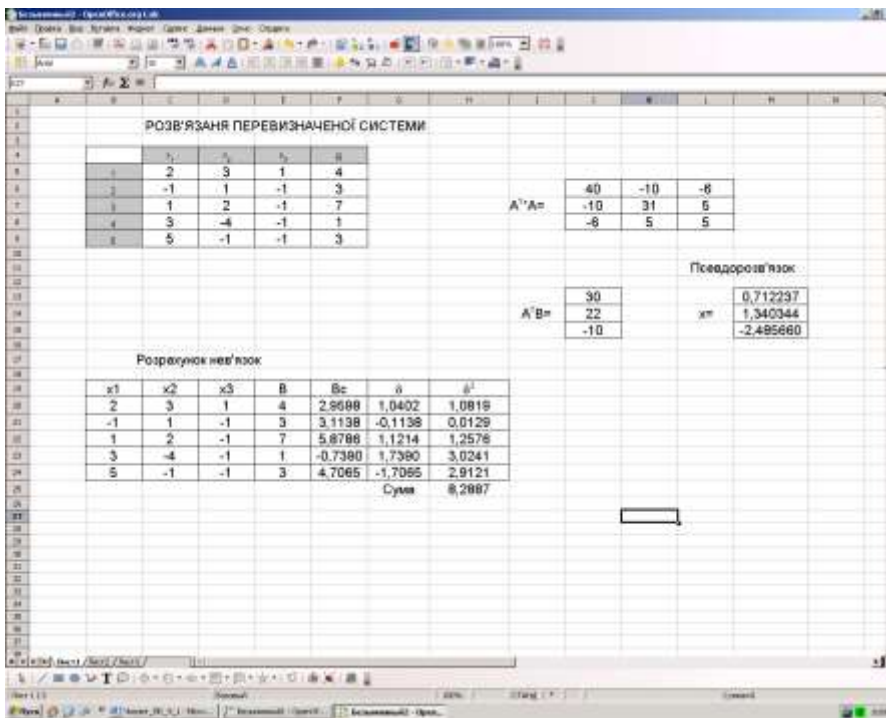


Рисунок 4.1 – Проект “Псевдорозв’язок системи рівнянь”

### Розв’язання в Scilab:

В Scilab є спеціальні засоби для розв’язання задачі МНК, але їх краще використовувати у задачах апроксимації (див. [лекцію 7](#)). Для розв’язання перевизначеної системи рівнянь доцільно використовувати матричні функції

Текст скрипту наведено у лістингу 4.1. Результати розрахунків наведені у протоколі 4.1

#### Лістинг 4.1

```
a=[2,3,1;-1,1,-1;1,2,-1;3,-4,-1;5,-1,-1]; //коєфіцієнти
b=[4;3;7;1;3]; //Вільні члени
A1=a'*a; // Розрахунок (AT)*A
B1=a'*b; // Розрахунок (AT)*B
disp('Розв`язок')
x=inv(A1)*B1//розв`язок
bc=a*x;
d=(b-bc)^2; //квадрати нев`язок
disp('x1      x2      x3      b      dc      d^2')
[a,b,bc,d]//Друк результатів
disp('Сума квадратів')
sum(d)
```

#### Протокол 4.1

Розв`язок

x =

0.7122371  
1.3403442  
- 2.4856597

x1 x2 x3 b dc d^2

ans =

2.	3.	1.	4.	2.959847	1.0819182
- 1.	1.	- 1.	3.	3.1137667	0.0129429
1.	2.	- 1.	7.	5.8785851	1.2575714
3.	- 4.	- 1.	1.	- 0.7390057	3.024141
5.	- 1.	- 1.	3.	4.706501	2.9121455

Сума квадратів

ans =

8.2887189

Результати розрахунків в обох середовищах є однаковими.

## 4.2 Кістякове розкладання прямокутних матриць

Нехай задана довільна прямокутна матриця розміру  $m \times n$ , що має ранг  $r$ . Нагадаємо, що:

- ✓ *рангом матриці* називається максимальне число її лінійно незалежних рядків або стовпців,
- ✓ ранг визначається, як *максимальний розмір* відмінного від нуля визначника підматриці, побудованої на досліджуваній матриці.

Матрицю  $A$  можна представити у вигляді добутку матриць  $B$  і  $C$ , що мають відповідно розміри  $m \times r$  і  $r \times n$ :

$$A = BC = \begin{pmatrix} b_{11} & \dots & b_{1r} \\ b_{21} & \dots & b_{2r} \\ \dots & \dots & \dots \\ b_{m1} & \dots & b_{mr} \end{pmatrix} \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ \dots & \dots & \dots & \dots \\ c_{r1} & c_{r2} & \dots & c_{rn} \end{pmatrix} \quad (r = \text{rank}(A)) \quad (4.8)$$

Ранги співмножників  $B$  і  $C$  обов'язково дорівнюють рангу вихідної матриці  $A$

$$\text{Rank}(B) = \text{rank}(C) = r. \quad (4.9)$$

Для того щоб одержати розкладання (4.8), досить в якості стовпців матриці  $B$  використати:

- ✓ будь-які  $r$  лінійно незалежних стовпців матриці  $A$ ,
- ✓ або будь-які  $r$  лінійно незалежних стовпців, через які *лінійно виражаються* стовпці матриці  $A$ .

Тоді:

- ✓ довільний  $j$ -й стовець матриці  $A$  буде *лінійною комбінацією стовпців* матриці  $B$  з коефіцієнтами  $c_{1j}, c_{2j}, \dots, c_{rj}$ ;
- ✓ ці коефіцієнти й утворять  $j$ -й стовець матриці  $C$  ( $j = 1, \dots, n$ ).

Оскільки матриці  $B$  і  $C$  мають *максимально можливий ранг*  $r$ , те квадратні матриці  $B^T B$  і  $C C^T$  є *невиродженими*:

$$|B^T B| \neq 0, \quad |C C^T| \neq 0 \quad (4.10)$$

Розкладання (4.8) називається *кістяковим розкладанням* матриці  $A$ . Це розкладання *неоднозначно*, але для практичних цілей досить одержати будь-яке кістякове розкладання (див. [4.3](#))

## Приклад 4.2

Побудувати кістякове розкладання матриці:

$$A = \begin{vmatrix} 2 & 3 & 5 & 6 & 9 & -1 \\ 4 & -1 & -5 & 1 & 0 & 3 \\ -8 & 9 & 25 & 9 & 18 & -11 \\ 6 & 5 & 2 & 7 & 1 & 1 \end{vmatrix} \quad (4.11)$$

Розв'язання:

а) в середовищі Calc за допомогою матричної функції MDETERM() розраховуємо визначники й визначаємо ранг матриці:

1) величини всіх визначників 4-го порядку, побудованих з мінорів матриці A, дорівнюють нулю;

2) величини всіх визначників третього порядку, побудованих з мінорів рядків 1,2,3 матриці A, дорівнюють нулю;

3) визначники третього порядку, побудовані з мінорів рядків 2,3,4 матриці A, відмінні від нуля

4) Висновки:

- ранг матриці A дорівнює 3,

- рядка 2,3,4 лінійно незалежні, їх можна використати, як основу кістякового розкладання;

б) за матрицю B вибираємо підматрицю, що побудована на стовпцях 1,2,3 матриці A;

в) виділяємо з її підматрицю  $\tilde{B}$ , що побудована на лінійно незалежних рядках 2,3,4 (рис. 4.2);

г) для побудови матриці C використаємо лінійно незалежні рядки 2,3,4 матриці A. Укладаємо систему лінійних алгебраїчних рівнянь для розрахунків елементів матриці C у матричному виді:

$$\tilde{B} \cdot C = A; \quad C = (\tilde{B})^{-1} \cdot A. \quad (4.12)$$

г) Систему (4.11) записуємо у виді приєднаної матриці:

$$\left| \begin{array}{ccc|ccc} 4 & -1 & -5 & 4 & -1 & -5 & 1 & 0 & 3 \\ -8 & 9 & 25 & -8 & 9 & 25 & 9 & 18 & -11 \\ 6 & 5 & 2 & 6 & 5 & 2 & 7 & 1 & 1 \end{array} \right|$$

д) знаходимо матрицю C за допомогою операцій обернення й матричного множення. На рис. 4.2 формула масиву C16:H18:

$$=MMULT(MINVERSE(C12:E14);F12:K14)$$

е) виконуємо перевірку за рівнянням  $A = BC$ . Формула масиву D25:I28:

$$=MMULT(C7:E10;C16:H18)$$

Вхідна й розрахункова матриці А збігаються (з точністю до машинного нуля).

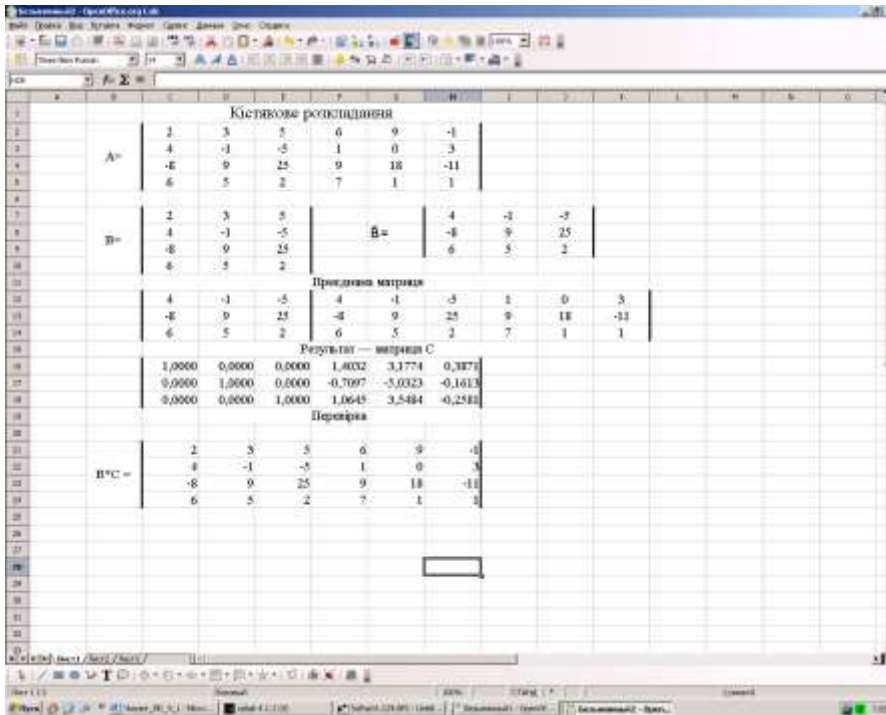


Рисунок 4.2 – Виконання кістякового розкладання

### 4.3 Псевдообернена матриця Мура-Пенроуза<sup>1)</sup>

В курсах алгебри доводиться: якщо А - квадратна й невироджена матриця, то для неї існує обернена матриця  $A^{-1}$ .

Якщо А :

- ✓ прямокутна матриця розміром  $m \times n$  ( $m \neq n$ );
  - ✓ або квадратна, але вироджена матриця, визначник якої дорівнює нулю,
- то матриця А не має оберненої й символ  $A^{-1}$  не має сенсу.

Для довільної прямокутної матриці А існує "псевдообернена" матриця  $A^+$ , що має деякі властивості оберненої матриці. У випадку, коли А - квадратна невироджена матриця, псевдообернена матриця  $A^+$  збігається зі оберненою  $A^{-1}$ .

Нехай А- довільна прямокутна матриця розміру  $m \times n$ . Доведено, що існує єдина матриця  $A^+$  розміру  $n \times m$ , що задовольняє співвідношенням:

$$A A^+ A = A; \quad A^+ A A^+ = A^+; \quad A^+ A = (A^+ A)^T; \quad A A^+ = (A A^+)^T. \quad (4.13)$$

Ця матриця називається псевдооберненою. Якщо матриця А - квадратна й невироджена ( $m = n$ ,  $\text{Det}(A) \neq 0$ ), то  $A^+ = A^{-1}$ .

<sup>1)</sup> Пенроуз Роджер (нар. 1931 р.) – американський математик. Праці в області алгебри, теорії груп, математичному апарату квантової механіки. В 1965 р, запропонував псевдообернені матриці

Для побудови псевдооберненої матриці використовується *кістякове розкладання*. Доведено, що:

$$A^+ = C^+ \cdot B^+ = C^T \cdot (C \cdot C^T)^{-1} \cdot (B^T \cdot B)^{-1} \cdot B^T. \quad (4.14)$$

### Приклад 4.3

Розрахувати для (4.11) псевдообернену матрицю.

Розв'язання:

а) Розв'язання у середовищі Calc

На рис. 4.3 наведений скріншот із проектом розрахунку псевдооберненої матриці (4.11) за результатами розрахунку матриць B і C у [прикладі 4.2](#). При його створенні формулу (4.14), занадто громіздку для прямого використання, розбили на співмножники й послідовно обчислили:

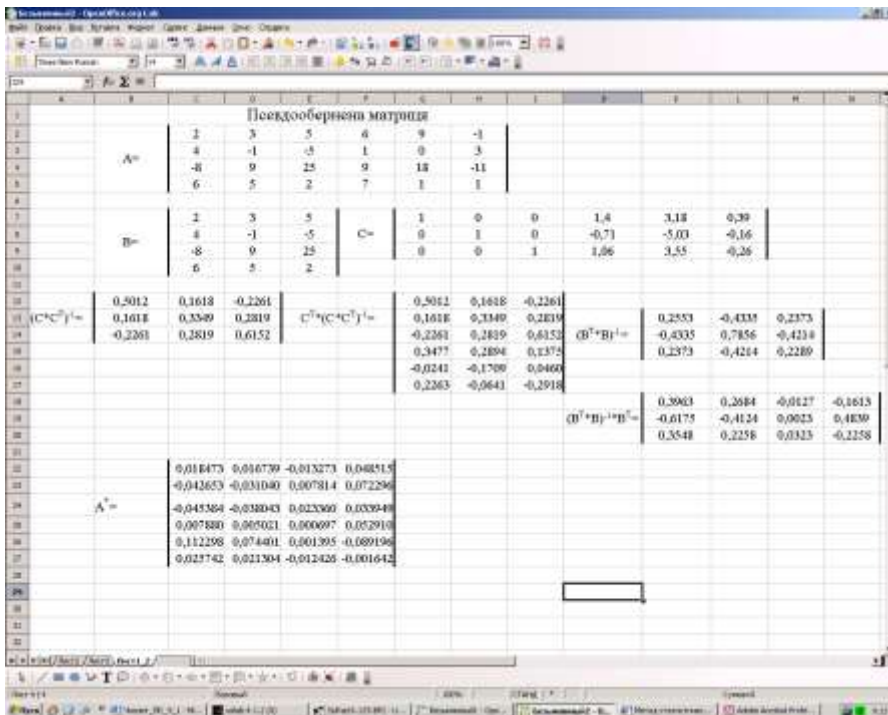
- ✓ матрицю  $(C \cdot C^T)^{-1}$  розміром  $3 \times 3$  у масиві B12:D14; формула масиву:  

$$=MINVERSE(MMULT(G7:L9;TRANSPOSE(G7:L9)));$$
- ✓ матрицю  $C^T (C \cdot C^T)^{-1}$  розміром  $6 \times 3$  у масиві G12:I17; формула масиву:  

$$=MMULT(TRANSPOSE(G7:L9);B12:D14);$$
- ✓ матрицю  $(B^T \cdot B)^{-1}$  розміром  $3 \times 3$  у масиві K13:M15; формула масиву:  

$$=MINVERSE(MMULT(TRANSPOSE(C7:E10);C7:E10));$$
- ✓ матрицю  $(B^T \cdot B)^{-1} B^T$  розміром  $4 \times 3$  у масиві K18:N20; формула масиву:  

$$=MMULT(K13:M15;TRANSPOSE(C7:E10))$$



### Рисунок 4.3 – Розрахунок псевдооберненої матриці

Далі відбувається остаточна "збірка" - виконується розрахунок псевдооберненої матриці  $A^+ = [C^T \cdot (C \cdot C^T)^{-1}] \cdot [(B^T \cdot B)^{-1} \cdot B^T]$  розміром  $6 \times 4$  у масиві C22:F27; формула масиву:

$$=MMULT(G12:I17;K18:N20);$$

#### б) Розв'язання у середовищі Scilab

В Scilab є спеціальна функція `pinv(A)`, що дозволяє автоматично розраховувати псевдообернену матрицю. У протоколі 4.3 наведено результат розрахунів

#### Протокол 4.3

```
-> A=[2,3,5,6,9,-1;...  
4,-1,-5,1,0,3;...  
-8,9,25,9,18,-11;...  
6,5,2,7,1,1]
```

```
-> A =
```

```
2. 3. 5. 6. 9. - 1.  
4. - 1. - 5. 1. 0. 3.  
- 8. 9. 25. 9. 18. - 11.  
6. 5. 2. 7. 1. 1.
```

```
-->pinv(A)
```

```
ans =
```

```
0.0184728 0.0167394 - 0.0132727 0.0485148  
- 0.0426533 - 0.0310402 0.0078139 0.0722963  
- 0.0453840 - 0.0380426 0.0233597 0.0339494  
0.0078796 0.0050208 0.0006969 0.0529098  
0.1122983 0.0744006 0.0013949 - 0.0891963  
0.0257423 0.0213037 - 0.0124264 - 0.0016419
```

В [4.2](#) було відзначено, що кістякове розкладання не є неоднозначним: за матрицю  $B$  можна вибирати різні стовпці матриці  $A$  або їхні лінійні комбінації. При цьому будуть виходити й різні матриці  $C$ . Примітною властивістю псевдооберненої матриці є її *незалежність* від матриць  $B$  и  $C$ , використаних у кістяковому розкладанні: незалежно від вибору цих матриць виходить та сама матриця  $A^+$ .



Розв'язання проводимо у середовищі Scilab, де є функція розрахунків псевдооберненої матриці.

Матриця коефіцієнтів цієї системи збігається з (4.11), тобто , рядки матриці лінійно залежні. Прямою перевіркою (знаходженням коефіцієнтів лінійної комбінації) можна переконатися, що дана система є несумісною. У лістингу 4.4. наведено скрипт для проведення розрахунків. У протоколі

#### Лістинг 4.4

```
A=[2,3,5,6,9,-1;...
4,-1,-5,1,0,3;...
-8,9,25,9,18,-11;...
6,5,2,7,1,1];
B=[8.3;5.1;2.5;1.2];
x=pinv(A)*B;//розрахунок розв'язку
disp('Результати')
disp('Система')
[A,B]//Розширена матриця коефіцієнтів
disp('Розв`язок')
x
disp('Розрахунок нев`язок')
for i=1:4
    bc(i)=A(i,:)*x;
end
d=(B-bc)^2;
[B,bc,d]
disp('Сума квадратів нев`язок')
sum(d)
```

#### Протокол 4.4

#### Результати

#### Система

ans =

```
2.  3.  5.  6.  9.  -1.  8.3
4. -1. -5.  1.  0.  3.  5.1
-8. 9. 25. 9. 18. -11. 2.5
6.  5.  2.  7.  1.  1.  1.2
```

#### Розв`язок

x =

```
0.2637316
-0.4060373
-0.4715662
```

0.1562406  
1.2079709  
0.2892740

Розрахунок нев'язок  
ans =

8.3 8.4714286 0.0293878  
5.1 4.8428571 0.0661224  
2.5 2.4142857 0.0073469  
1.2 1.2 3.994D-30

Сума квадратів нев'язок'  
ans =

0.1028571

#### 4.5 Поняття про коректно й некоректно поставлені задачі

Видатний французький математик Ж. Адамар в 1902 р. увів поняття коректності й некоректності.



АДАМАР ЖАК  
(1865 - 1936),

французький математик, іноземний член-кореспондент Російської АН (1922) і іноземний почесний член-кореспондент АН СРСР (1929). Праці в галузі диференціальних рівнянь, теорії функцій, теорії чисел, механіці.

Розглянемо операторное рівняння:

$$Ax = f, \quad x \in X, \quad f \in F, \quad (4.20)$$

де  $x$  – шуканий розв'язок,  
 $f$  - задана права частина,  
 $X$  і  $F$  - деякі гільбертові простори;

$A$  - заданий неперервний оператор (лінійний і нелінійний, інтегральний, диференціальний або алгебраїчний і т.д.).

Визначення. Задача розв'язання рівняння (4.20) називається *коректною* або *коректно поставленою* (well-posed), якщо:

- а) рішення існує,
- б) рішення є *єдиним*,
- в) рішення є *стійким*.

Якщо хоча б одне із цих умов не виконується, то задача називається *некоректною* або *некоректно поставленою* (ill-posed).

З погляду класичної лінійної алгебри розв'язки перевизначених або недовизначених систем є *некоректними*, тому що порушуються перші 2 умови Адамара. Разом з тим, багато практичних задач у фізиці, електротехніці, хімії та інших науках виявляються некоректними. Із цього погляду введення прийомів, описаних вище:

- ✓ відмова від поняття розв'язку, як вектору, що перетворює вирази в тотожності,
- ✓ введення поняття псевдорозв'язку й використання методу найменших квадратів, як основи для його одержання,
- ✓ введення поняття псевдооберненої матриці й найкращого рішення недовизначених систем,

дозволило перебороти обмеження перших двох умов коректності.

Істотною перешкодою в постановці коректних задач є *нестійкість*. Причини нестійкості для СЛАР розглянуті в розділі 2.5, вони пов'язані з більшими розходженнями у власних значеннях або сингулярних числах матриці коефіцієнтів.

Сучасний підхід до поняття коректності був уведений видатним радянським математиком академіком А.Н.Тихоновим.



ТИХОНОВ АНДРІЙ  
МИКОЛАЙОВИЧ  
(1906-1993)

видатний радянський російський математик, академік, двічі Герой Соціалістичної праці, лауреат Ленінської та державних премій, один з засновників та директор (1978-1989) Інституту прикладної математики імені Келдиша (Москва). Зробив значний внесок в теорію диференціальних рівнянь, обчислювальну математику, математичну геофізику, розробив метод регуляризації для розв'язання некоректно поставлених задач

Деякі визначення, що ставляться до теорії коректності Тихонова:

а) множина  $M$  метричного підпростору називається *компактною*, якщо з будь-якої послідовності елементів цієї множини можна виділити підпослідовність, що збігається до деякого елемента з  $M$ . При цьому будь яка множина, що є водночас:

- ✓ скінченновимірною,
- ✓ обмеженою,
- ✓ замкненою,

є *компактною*;

б) оператор  $K$ , що діє із простору  $\Phi$  у простір  $F$ :

$$K: \Phi \rightarrow F, \quad (4.21)$$

називається *цілком безперервним*, якщо будь-яку обмежену множину простору  $\Phi$  він переводить у множину, яка є компактною в  $F$ .

Такими можуть бути будь-які оператори. Так, одиничний оператор  $E$ , визначений, як  $E\Phi = \Phi$ , є цілком неперервним тільки, коли  $\Phi$  є скінченновимірним простором. У протилежному випадку  $E$  є *обмеженим оператором, але не цілком неперервним*.

в) Теорема. Якщо  $K$  - цілком неперервний оператор, а  $\Gamma$  - обмежений оператор, то оператори  $K\Gamma$ ,  $\Gamma K$  є *цілком неперервними*.

г) Наслідок. У нескінченновимірному просторі цілком безперервний оператор  $K$  не може мати обмеженого оберненого оператора  $K^{-1}$

Дійсно, у протилежному випадку одиничний оператор  $E = K^{-1} \cdot K$  повинен бути цілком неперервним у нескінченновимірному просторі, але вище вказувалося несправедливість такого припущення.

Цей наслідок є "головним винуватцем" нестійкості багатьох задач, у тому числі, розв'язання систем лінійних рівнянь.

д) Теорема. Нехай  $\Phi$  і  $F$  - метричні простори, причому  $M \subseteq \Phi$  - компактна множина з  $\Phi$ . Якщо оператор  $K$  є:

- ✓ неперервним;
- ✓ однозначно відображає множину  $M$  на множину  $N = KM \in F$ ,

то обернений оператор  $K^{-1}$  є *неперервним на множині  $N$* .

Задача розв'язання операторного рівняння (4.20) називається *коректно поставленою по Тихонову*, якщо виконані наступні умови:

а) априорі відомо, що розв'язок рівняння (4.21):

- існує;
- й належить деякій заданій множині  $M \subseteq \Phi$ , тобто,  $f \in N = KM$ ;

б) рішення є *єдиним* на множині  $M$ ;

в) існує *неперервна залежність розв'язку  $\varphi$  від правої частини  $f$* , коли варіації  $f$  не виводять розв'язок за межі множини  $M$ , тобто оператор  $K^{-1}$  є неперервним на множині  $N$ .

Множина  $M$ , на образі  $N = KM$  якої оператор існує та є неперервним, називається *множиною коректності*.

В умові в) неперервна залежність оберненого оператора гарантується тільки множині  $N$ , тобто стійкість задачі (4.20) досягається за рахунок звуження класу можливих розв'язків до множини  $M$  (або, що те ж саме, звуженням правих частин  $f$  до множини  $N$ ).

Тому задачу (4.20), коректну по Тихонову, називають також *умовно-коректним задачею*, а стійкість по Тихонову (тобто умова 4.5.6 в)) - *умовною стійкістю*.

З порівняння умов теореми 4.5.5 д) і визначення коректності по Тихонову видно: якщо  $K$  - неперервний оператор, то множиною коректності для задачі (4.20) є компактна множина  $M$  простору  $\Phi$ .

Таким чином, якщо звужити клас шуканих рішень до компакту  $M$ , те:

- а) обернений оператор стане неперервним;
- б) малим погрішностям правої частини (4.20) будуть відповідати малі погрішності рішення.

Висновок: вибором розв'язку операторного рівняння (4.20) тільки з компактної множини  $M$  *досягається його стійкість*. Це є фундаментальним положенням, на якому ґрунтуються стійкі методи рішення некоректно поставлених задач.

Для розв'язання некоректних задач використовують *метод регуляризації Тихонова*.

Суть методу. Розглянемо операторне рівняння (4.20), у якому замість точних  $A$  і  $f$  відомі їхні наближення  $\tilde{A}$  й  $\tilde{f}$ , такі, що:

$$\|f - \tilde{f}\| < \delta, \quad (4.22)$$

$$\|A - \tilde{A}\| < \varepsilon, \quad (4.23)$$

де  $\delta > 0$  і  $\varepsilon > 0$  — погрішності правої частини й оператора (точніше, їхні верхні оцінки, тому в (4.22) і (4.23) використані знаки  $<$ , а не знаки  $=$ ).

Таким чином, замість системи (4.20) розв'язується система:

$$\tilde{A}\tilde{x} = \tilde{f} \quad (4.24)$$

У методі регуляризації Тихонова ставляться дві умови:

- ✓ умова мінімізації нев'язки (4.22), як у методі найменших квадратів Гаусса,
- ✓ умова мінімізації норми рішення типу (4.23), як у методі псевдооберненої матриці Мура-Пенроуза.

Це - задача умовної мінімізації й вона розв'язується методом невизначених множників Лагранжа:

$$\|Ay - f\|^2 + \alpha \|y\|^2 \rightarrow \min, \quad (4.25)$$

где  $\alpha$  - регулюючий множник, що грає роль множника Лагранжа

З (4.25), після диференціювання по  $y$  і прирівнювання похідної нулю одержуємо *рівняння Тихонова*:

$$(\alpha E + A^T A)y = A^T f, \quad (4.26)$$

где  $E$  – одинична матриця.

Розв'язок у випадку системи лінійних рівнянь:

$$y = (\alpha E + A^T A)^{-1} A^T f. \quad (4.27)$$

Аналіз метода:

а) якщо  $\alpha=0$ , те метод регуляризації Тихонова переходить в метод найменших квадратів с *вкрай нестійким* розв'язком, але з мінімальною нев'язкою  $\|Ay-f\|^2$ ;

б) зі збільшенням  $\alpha$  розв'язок становиться більш гладким та стійким. При цьому зменшується норма розв'язку  $\|y\|^2$ , але збільшується нев'язка;

в) при деякому помірному значенні  $\alpha$  розв'язок у буде мати водночас й помірну гладкість, й помірну нев'язку. Деякі способи вибору  $\alpha$  наведені [нижче](#);

г) якщо  $\delta, \varepsilon \rightarrow 0$ , те  $\alpha \rightarrow 0$  і:

$$y = \lim_{\alpha \rightarrow 0} (\alpha E + A^T A)^{-1} \cdot f \equiv A^+ \cdot f, \quad (4.28)$$

Тобто розв'язок у переходить у нормальний псевдорозв'язок.

Таким чином метод регуляризації Тихонова є *узагальненням метода найменших квадратів Гаусса и метода псевдооберненого оператора Мура-Пенроуза*.

Метод регуляризації Тихонова є *стійким*, тобто виконується 3-й пункт коректності по Адамару. Ця стійкість обумовлена наступними обставинами:

а) оператор  $A^T A$  в (4.27) є *позитивно визначеним*, тому всі його властиві значення є дійсними і ненегативними:  $\lambda(A^T A) \geq 0$ , причому  $\lambda(A^T A)_{\min} = 0$ ;

б) наявність доданку  $\alpha E$  в (4.27) збільшує всі величини  $\lambda(A^T A)$  на  $\alpha$ , тому  $\alpha E + \lambda(A^T A)_{\min} = \alpha$ ;

в) внаслідок б):

1) оператор  $\alpha E + A^T A$  становиться *оборотним*,

2) норма оберненого оператора  $\|\alpha E + A^T A\|^{-1} = 1/\alpha \neq \infty$  и задача становиться стійною.

Способи вибору параметра регуляризації  $\alpha$ :

а) спосіб норми нев'язкі:  $\alpha$  вибирається з умови:

$$\|\alpha E + A^T A\| = \delta. \quad (4.29)$$

Якщо  $\|f\| > \delta$ , то розв'язок рівняння (4.29) відносно  $\alpha$ :

- існує;

- є єдиним;

б) спосіб підбору: знаходять розв'язки в для ряду "розумних" значень  $\alpha$  і остаточний вибір робиться на основі додаткової інформації про розв'язки, в

основному, візуально. Спосіб досить нагадує вибір контрасту телезображення. Якщо під зображенням мати на увазі величину  $\alpha$  то:

- зменшення  $\alpha$  відповідає підвищенню нестійкості рішення  $u$ , тобто збільшенню контрасту зображення,

- навпаки, збільшення  $\alpha$  відповідає зменшенню контрасту.

Незважаючи на простоту, спосіб підбору може бути досить ефективним у випадку, коли наявна певна інформація про розв'язок (ступінь гладкості, число екстремумів і т.д.), а також виконана обробка попередніх "близьких" прикладів, що дозволяє виділити область можливих значень;

в) асимптотичний спосіб, заснований на наступній залежності: при  $\varepsilon = 0$  и  $\delta \rightarrow 0$ :

$$\alpha = C\delta^2, \quad (4.30)$$

де  $C > 0$  — деяка константа.

Цій спосіб можна використовувати при малих  $\delta$ .

### Приклад 4.5

Провести дослідження стійкості і побудувати розв'язок системи перевизначених рівнянь. Коефіцієнти рівнянь і вільні члени наведені у табл. 4.1

Таблиця 4.1 – коефіцієнти і вільні члени системи рівнянь

NN	Коефіцієнти при невідомих				$b_i$
	$x_1$	$x_2$	$x_3$	$x_4$	
1	0,9	5,26	2,68	1,51	11,51
2	1,59	0,61	4,13	1,75	3,29
3	2,66	3,22	7,75	3,23	10,03
4	5,66	0,29	2,93	4,85	2,4
5	7,1	1,65	4,85	6,33	6,04
6	4,38	4,3	2,27	4,16	9,91
7	9,67	0,7	4,24	8,23	4,08
8	0,28	8,1	4,61	1,5	18,04
9	5,13	3,13	0,15	4,43	6,58
10	5,14	3,61	4,78	4,95	9,66
11	5,42	6,63	7,01	5,7	16,64

Продовження табл. 4.1

NN	Коефіцієнти при невідомих				b <sub>i</sub>
	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	
12	0,26	8,53	1,47	1,21	17,37
13	4,52	7,38	7,29	5,08	18,15
14	1,38	4,41	3,17	1,86	10,21
15	7,78	1,25	6,04	6,95	5,86
16	3,92	5,24	6,43	4,3	13,43
17	4,45	3,33	2,23	4,12	7,9

Розв'язання

Укладаємо в середовищі Scilab програму розрахунків (лістинг 4.5 А), що містить такі частини:

- а) введення коефіцієнтів системи і вільних членів у вигляді матриці а;
- б) розподіл уведеної матриці на :
  - ✓ матрицю коефіцієнтів А;
  - ✓ вектор-стовпець вільних членів В;
- в) розрахунок  $A^T \cdot A$  і  $A^T \cdot B$ ;
- г) розрахунок числа обумовленості матриці  $A^T \cdot A$ ;
- д) розрахунок розв'язку:  $xx=(A^T \cdot A)^{-1} \cdot (A^T \cdot B)$ .

Лістинг 4.5А

```

a=[0.95.26    2.681.5111.51;
1.59    0.61    4.131.753.29;
2.66    3.22    7.753.2310.03;
5.66    0.29    2.934.852.4;
7.11.65    4.856.336.04;
4.38    4.3    2.274.169.91;
9.67    0.7    4.248.234.08;
0.28    8.1    4.611.5    18.04;
5.13    3.13    0.154.436.58;
5.14    3.61    4.784.959.66;
5.42    6.63    7.015.7    16.64;
0.26    8.53    1.471.2117.37;
4.52    7.38    7.295.0818.15;
1.38    4.41    3.171.8610.21;
7.78    1.25    6.046.955.86;
3.92    5.24    6.434.3    13.43;
4.45    3.33    2.234.127.9];
for i=1:17
    for j=1:4
        A(i,j)=a(i,j);
    
```

```
end
B(i)=a(i,5);
end
//[A,B]
A1=A'*A
disp('Число обумовленості')
cond(A1)
AA=inv(A1);
xx=AA*(A'*B)
```

Результати розрахунків наведені у протоколі 4.5А. З них випливає, що число обумовленості матриці  $A^T \cdot A$  (відношення її сингулярних чисел, див. [2.5](#)). є дуже великим (16588031). Це вказує на нестійкість розв'язку системи

#### Протокол 4.5А

```
A1 =
 405.8436   212.7889   316.0075   377.5191
 212.7889   379.373    293.683    237.5405
 316.0075   293.683    380.9561    320.2764
 377.5191   237.5405    320.2764    357.7698
Число обумовленості
ans =
 16588031.
xx =
 -5.569097
  1.2500145
 -0.2514404
  7.0670086
```

Для встановлення величини нестійкості використаємо метод математичного моделювання:

а) проводимо 1000 випадкових випробувань за таким алгоритмом:

- ✓ додаємо до кожного із значень вільних членів  $B_i$  ( $i=1,2,\dots,n$ ) невеличкі збурення у вигляді випадкових чисел, що рівномірно розподілені від  $-0,01$  до  $+0,01$ ,
- ✓ розраховуємо збурений розв'язок і запам'ятуємо його;

б) за результатами випробувань знаходимо мінімальне і максимальне значення кожного з розв'язків, по їх величині визначаємо, чи є відхилення слабкими чи сильними.

Скрипт для розрахунків наведено у лістингу 4.5Б. Результати розрахунків наведені у таблиці 4.2

### Лістинг 4.5Б

```
//випадкові випробування
Z=zeros(4,1000);
for i=1:1000
    B1= B-0.01*(1-2*rand());
    xxx=AA*(A'*B1);
    Z(:,i)=xxx;
end
for i=1:4
    [i,min(Z(i,:)), max(Z(i,:))]
end
```

Таблиця 4.2 Розв'язки системи рівнянь (табл.. 4.1) при внесенні збурень

i	Значення $x_i$		
	Не збурене	Мінімальне	Максимальне
1	- 5.569	- 5.803	-5.335
2	1.250	1.222	1.278
3	- 0.251	- 0.281	-0.222
4	7.067	6.773	7.361

Як впливає з табл. 4.2, відгук невідомих  $x_1$  й  $x_4$  є майже у 50 разів вище за збурення вільного члену, тобто, система показує нестійкість за цими змінними.

Далі проводимо побудову стійкого розв'язку за Тихоновим. Для вибору параметра регуляризації скористуємося способом норми нев'язки (див. рівняння (4.29)). Користуючись тим, що:

- ✓ стійкість розв'язку характеризується *числом обумовленості*,
- ✓ розв'язок можна вважати *відносно сталим*, коли число обумовленості є меншим за  $10^3$  (див. [приклад 2.4](#)),

умову (4.29) можна записати у виді:

$$\text{cond}(\alpha E + A^T A) < 10^3. \quad (4.31)$$

Додатковою умовою для вибору параметру регуляризації може бути деяка апіорна або апостеріорна інформація про систему, що може бути відома з фізичного сенсу реальної задачі, що приводить до заданої системи рівнянь. Частіше за все ця інформація стосується величини допустимого середнє квадратичного відхилення величин вільних членів  $B_i$  й визначається експериментально для реальних фізичних або технічних задач. Припустимо, що у випадку, що

розглядається, середнє квадратичне відхилення не перевищує 0,05. Тоді умовою для вибору  $\alpha$  може бути таке міркування: залишкове середнє квадратичне відхилення, що характеризує різницю між розрахованими і заданими величинами  $B_i$ , не повинно перевищувати середнє квадратичну похибку визначення цих коефіцієнтів у 2 рази:

$$\sigma = \sqrt{\frac{\sum_{i=1}^m \left( B_i - \sum_{j=1}^n a_{ij} \cdot x_j \right)^2}{m-n}} < 2 \cdot \sigma_{B_i} = 0,1, \quad (4.32)$$

де  $m=17$ ,  $n = 4$  – відповідно кількість рівнянь і невідомих,  
 $\sigma_{B_i} = 0,05$  – стандартне відхилення вільних членів.

Це міркування запозичене з математичної статистики і регресійного аналізу, зокрема перевірки адекватності моделі. Адекватність перевіряють за допомогою відношення дисперсій (критерій Фішера), і це відношення при невеликій кількості точок і змінних не перевищує 4. Тобто квадратний корінь з цього відношення ( відношення середнє квадратичних відхилень( не перевищує 2.

Текст скрипта для розрахунків наведено в лістингу 4.5В. Він передбачає наступний алгоритм:

а) в системі  $(A^T \cdot A) X = A^T \cdot B$  ділимо кожне рівняння на діагональний елемент  $a_{zi}$ . Одержимо СЛАР, що:

- ✓ має той же самий розв'язок, що й первинна система,
- ✓ усі діагональні елементи цей системи дорівнюють 1

б) для одержання збуреної системи до діагонального елементу додаємо невелике значення  $\alpha$ ,

в) проводимо серію розрахунків з різними значеннями  $\alpha$ , що надає можливість дослідити її вплив на число обумовленості, стійкість розв'язків і величину  $\sigma$ .

#### Лістинг 4.5В

```
a=[0.95.26 2.681.51 11.51;
1.59 0.61 4.131.75 3.29;
2.66 3.22 7.75 3.23 10.03;
5.66 0.29 2.93 4.85 2.4;
7.11.65 4.85 6.33 6.04;
4.38 4.3 2.27 4.16 9.91;
9.67 0.7 4.24 8.23 4.08;
0.28 8.1 4.61 1.5 18.04;
5.13 3.13 0.15 4.43 6.58;
5.14 3.61 4.78 4.95 9.66;
5.42 6.63 7.01 5.7 16.64;
0.26 8.53 1.47 1.21 17.37;
4.52 7.38 7.29 5.08 18.15;
1.38 4.41 3.17 1.86 10.21;
7.78 1.25 6.04 6.95 5.86;
3.92 5.24 6.43 4.3 13.43;
```

```
4.45 3.33 2.234.127.9];
for i=1:17
    for j=1:4
        A(i,j)=a(i,j);
    end
    B(i)=a(i,5);
end
disp('Незбурена система')
A1=A'*A;
B1=A'*B;
for i=1:4
    lambda=A1(i,i);
    for j=1:4
        A1(i,j)=A1(i,j)/lambda;
    end
    B1(i)=B1(i)/lambda;
end
[A1,B1]
disp('Число обумовленості')
cond(A1)
disp('Розв'язок')
X=inv(A1)*B1;
X'
s=sum((A*X-B)^2);
disp('Залишкове середнє квадратичне відхилення')
sig=sqrt(s/13)
alpha=1
while alpha > 0
    disp('Збурення')
    alpha=input('alpha')
    A2=A1;
    for i=1:4
        A2(i,i)=A1(i,i)+alpha;
    end
    disp('Число обумовленості')
    cond(A2)
    disp('Розв'язок')
    X1=inv(A2)*B1;
    X1'
    s1=sum((A*X1-B)^2);
    disp('Залишкове середнє квадратичне відхилення')
    sig1=sqrt(s1/13)
end
```

Результати досліджень наведені у табл. 4.3. З цих даних випливає наступне:

Таблиця 4.3 – Вплив величини  $\alpha$  на розв'язок системи (табл. 4.1) та її характеристики

$\alpha$	cond	$\sigma$	$x_1$	$x_2$	$x_3$	$x_4$
0	$1,64 \cdot 10^7$	0,0493	- 5,5691	1,2500	- 0,2514	7,0670
0,001	3336	0,0555	- 0,0699	1,9333	0,4406	0,1909
0,002	1669	0,0604	- 0,0691	1,9294	0,4429	0,1903
0,003	1112	0,0677	- 0,0686	1,9257	0,4451	0,1902
0,004	835	0,0767	- 0,0682	1,9218	0,4473	0,1901
0,005	668	0,0868	- 0,0679	1,9181	0,4494	0,1902
0,006	557	0,0977	- 0,0675	1,9144	0,4515	0,1902
0,007	478	0,1090	- 0,0671	1,9106	0,4535	0,1903

а) із зростанням  $\alpha$  від 0 до 0,001 відбувається різке (на 4 порядки) зменшення числа обумовленості і значні зміни розв'язків, особливо  $x_1, x_4$ , що зменшуються майже на 2 порядки, а також  $x_3$ , що змінює знак. Із подальшим зростанням  $\alpha$  спостерігається повільне зменшення числа обумовленості, збільшення  $\sigma$  і зміни компонентів вектору розв'язків. Критичне значення  $\alpha$ , більше якого величини  $\sigma$  становляться занадто великими, спостерігається між 0,006 і 0,007. Тому можна вважати, що стійкий розв'язок знаходиться при  $\alpha=0,006$ . При цьому вектор розв'язку складає:

$$\mathbf{X} = (-0,0675; 1,9144; 0,4515; 0,1902),$$

він задовольняє наведеним критеріям щодо величини  $\alpha$ , одержаним з фізичного змісту задачі та даних спостережень.

#### 4.6 Обчислення властивих значень і властивих векторів матриць

Якщо перша за значимістю задача прикладної алгебри – це розв'язок систем лінійних алгебраїчних рівнянь, те друга - це знаходження властивих значень і властивих векторів квадратних матриць, тобто, знаходження всіх значень  $\lambda$  і векторів  $\mathbf{C}$ , які відповідають їм, що є розв'язками рівняння:

$$AC = \lambda C, \quad (4.33)$$

де  $A$  – квадратна матриця розмірами  $n \times n$ .

Числа  $\lambda$  називають властивими значеннями матриці  $A$ , вектора  $C$  – властивими векторами матриці  $A$ .

Суть алгебраїчного рішення полягає в тім, що вираження (4.33) приводиться до виду:

$$C(A - \lambda E) = 0, \quad (4.34)$$

де  $E$  – одинична матриця.

Єдиний розв'язок однорідної системи (4.34) лінійних алгебраїчних рівнянь існує тоді й тільки тоді, коли визначник матриці коефіцієнтів дорівнює нулю, тобто

$$\text{Det}(\lambda) = |A - \lambda E| = 0. \quad (4.35)$$

Рівняння (4.33) і визначник, який розкривають, називаються *характеристичним рівнянням і характеристичним визначником*.

Знаходження власних векторів полягає в розкритті визначника (4.35) і одержання на його основі алгебраїчного рівняння  $n$ -й ступеня відносно  $\lambda$  (характеристичне рівняння). Розв'язуючи отримане рівняння (як правило, чисельно), одержують значення  $\lambda$ .

У відповідності до основної теореми алгебри, алгебраїчне рівняння  $n$ -й ступеня має рівно  $n$  коренів, дійсних або комплексно-спряжених. Тому квадратна матриця загального виду має рівно  $n$  власних значень, у тому числі дійсних, комплексно-спряжених і кратних.

Властиві вектори знаходять шляхом розв'язання системи (4.33) при отриманих значеннях  $\lambda$ . Властиві вектори визначаються з точністю до співмножника, тому одному з компонентів надають яке-небудь числове значення, і по ньому знаходять інші компоненти.

На практиці найчастіше виникає задача знаходження властивих значень і векторів *симетричних матриць*. Такі задачі виникають у математиці (теорія квадратичних форм), механіці, фізиці, хімії, астрономії. Уперше такі задачі виникли в астрономії при вивченні вікових рухів планет. Тому визначник (4.35) для симетричних матриць іноді називають *віковим (секулярним) визначником*.

Корисна властивість симетричних матриць полягає в тім, що їхні *власні значення завжди є дійсні числа*.

Для знаходження властивих значень і властивих векторів в 19-початку 20 століття була розроблена безліч методів, спрямованих на розкриття характеристичного визначника або виділення окремих властивих значень, наприклад, найбільших за модулем. Ці методи були орієнтовані на "ручні" розрахунки й виявилися неефективними при комп'ютерній реалізації. Єдине



де

$$k = \sqrt{1 - \frac{(2a_{ik})^2}{(a_{ii} - a_{kk})^2 + (2a_{ik})^2}}; \quad (4.39)$$

$$\varepsilon = \begin{cases} \text{sign}(a_{ik}), & a_{ii} = a_{kk}, \\ \text{sign}\left(\frac{a_{ii} - a_{kk}}{a_{ik}}\right), & a_{ii} \neq a_{kk}. \end{cases} \quad (4.40)$$

У цьому випадку отримаємо матрицю  $B = T_m^{-1}AT_m$  зі зміненими  $i$ -м та  $k$ -м стовпцями та рядками:

$$\left\{ \begin{array}{l} b_{ii} = c^2 a_{ii} + s^2 a_{kk} + 2csa_{ik} \\ b_{kk} = s^2 a_{ii} + c^2 a_{kk} - 2csa_{ik} \\ b_{ik} = b_{ki} = 0 \\ b_{ij} = b_{ji} = ca_{ji} + sa_{jk} \\ b_{kj} = b_{jk} = ca_{ji} - sa_{jk} \\ b_{ji} = a_{ji} \text{ в інших випадках} \end{array} \right. \quad j=1,2,\dots,n, j \neq i, j \neq k \quad (4.41)$$

У процесі перетворень виконується співвідношення:

$$b_{ii}^2 + b_{kk}^2 = a_{ii}^2 + a_{kk}^2 + 2a_{ik}, \quad (4.42)$$

- а) сума діагональних елементів збільшується;
  - б) на таку ж величину зменшується сума позадіагональних елементів.
- Звідси виходить збіжність до діагональної матриці.

На практиці обчислювальний процес закінчують, коли найбільший за модулем позадіагональний елемент не перевищує заданого малого числа  $\varepsilon$ .

Якщо одночасно з перетвореннями  $T^{-1}AT$  проводити подібне перетворення  $T^{-1}ET$  з одиничною матрицею, те після діагоналізації одержимо матрицю, стовпцями якої є ортонормовані властиві вектори матриці  $A$ , тобто ,

тобто:

- а) скалярні добутки векторів-стовпців з різними індексами є нульовими:

$$(C_i, C_j) = \sum_{p=1}^n C_{pi} \cdot C_{pj} = 0. \quad (4.43)$$

- б) довжина (модуль) кожного вектору дорівнює 1:

$$\ell_i = \sqrt{C_{1i}^2 + C_{2i}^2 + \dots + C_{ni}^2} = 1. \quad (4.44)$$

Метод Якобі є чисельно стійким, швидкість збіжності пропорційна  $n^2$ . Він може застосовуватися у випадках кратних власних значень. Особливості програмування:

а) у багатьох програмах використовують властивість симетрії матриці й записують її не у вигляді матриці  $n \times n$ , а у вигляді вектора розміром  $n(n+1)/2$ , як компоненти якого записують діагональні й наддіагональні елементи по черзі 1,2,...,n-й рядків. Це дозволяє заощаджувати пам'ять, що істотно для матриць великого розміру при використанні комп'ютерів попередніх генерацій;

б) для забезпечення стійкості необхідно *примусово робити нульовим елемент що виключається*.

Розглянутий метод називається *методом Якобі з вибором головного елемента*. Його недолік: на кожній ітерації доводиться переглядати всі наддіагональні елементи в кількості  $n(n-1)/2$  штук. У випадку матриць великої вимірності це стає фактором, що лімітує, розрахунки. У цьому випадку ефективніше застосувати варіант методу, що називається *методом Якобі з перешкодами*. У цьому методі перед початком обчислювального процесу створюється послідовність перешкод, наприклад,  $1, 0.1, 0.01, \dots, 0.1 \times k$ . У процесі перегляду знаходили перший позадіагональний елемент, більший ніж поточна перешкода, і виключали його. Якщо всі елементи виявлялися менше поточної перешкоди - переходили до наступної перешкоди. Процес повторювали доти, поки не була пройдена остання перешкода.

У порівнянні з методом вибору головного елемента, метод з перешкодами вимагає більшого числа ітерацій. Однак їхнє збільшення компенсується меншою глибиною перегляду наддіагональних елементів, тому загальний час виконання розрахунків скорочується.

Розрахунок властивих значень і властивих векторів квадратних матриць входить до складу вільності пакетів прикладної математики. Зокрема у Scilab є функція `spes(A)`, що дозволяє розраховувати властиві значення (дійсні або комплексні) для симетричних і несиметричних матриць. Якщо використовувати конструкцію: `[A,C]=spes(X)`, те разом з властивими значеннями, видаються у формі діагональної матриці  $A$ , де за зменшенням розташовані властиві значення, розраховують матрицю ортонормованих властивих векторів  $C$ .

### Приклад 4.6

Розрахувати властиві значення і властиві вектори матриці:

$$\begin{vmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{vmatrix}$$

Розв'язання

Скрипт для розрахунків наведено у лістингу 4.6. Результати розрахунків наведено у протоколі 4.6

Лістинг 4.6

```
x=[1,2,3,4,5;6,7,8,9,10;11,12,13,14,15;...
16,17,18,19,20; 21,22,23,24,25]
[A,C]=spec(x)
```

Протокол 4.6

--> x =

```
1. 2. 3. 4. 5.
6. 7. 8. 9. 10.
11. 12. 13. 14. 15.
16. 17. 18. 19. 20.
21. 22. 23. 24. 25.
```

C =

```
- 3.6420807 0 0 0 0
0 68.642081 0 0 0
0 0 - 7.589D-16 + 2.455D-15i 0 0
0 0 0 - 7.589D-16 - 2.455D-15i 0
0 0 0 0 8.826D-16
```

A =

```
0.6749528 0.1079750 - 0.1327617 - 0.2053395i - 0.1327617 + 0.2053395i -
0.2214371
0.3603897 0.2527750 - 0.1572340 + 0.3408620i - 0.1572340 - 0.3408620i -
0.1644165
0.0458266 0.3975750 0.1626635 - 0.1006141i 0.1626635 + 0.1006141i
0.8772275
- 0.2687366 0.5423751 0.6774220 0.6774220 -
0.3754570
- 0.5832997 0.6871751 - 0.5500898 - 0.0349084i - 0.5500898 + 0.0349084i -
0.1159169
```

## Контрольні запитання

- 1 Яка СЛАР називається перевизначеною?
- 2 Постановка задачі знаходження розв'язку перевизначеної СЛАР.
- 3 Надати визначення поняття псевдорозв'язку перевизначеної СЛАР.
- 4 Сутність методу найменших квадратів для розв'язання перевизначеної СЛАР.
- 5 Алгоритм розв'язання перевизначеної СЛАР методом найменших квадратів
- 6 Чому псевдорозв'язок узагальнює поняття точного розв'язку СЛАР?
- 7 Зміст поняття «нормальна система» стосовно розв'язання перевизначеної СЛАР. Властивості нормальної системи.
- 8 Сутність кістякового розкладання прямокутних матриць.
- 9 Вимоги до рангів матриць при кістяковому розкладанні.
- 10 Чи є кістякове розкладання однозначним?
- 11 Алгоритм побудови кістякового розкладання.
- 12 Зміст поняття «псевдообернена матриця». Чому це поняття узагальнює поняття «обернена матриця»?
- 13 Яка процедура використовується для побудови псевдо оберненої матриці?
- 14 Властивості псевдо оберненої матриці Мура-Пенроуза
- 15 Алгоритм побудови псевдо оберненої матриці Мура-Пенроуза
- 16 Зміст поняття «недовизначена система»
- 17 Яким чином можна зробити однозначною процедуру розв'язання недовизначеної несумісної СЛАР?
- 18 Зміст поняття «нев'язка»
- 20 Сутність розв'язання недовизначеної несумісної СЛАР
- 21 Зміст поняття коректності за Адамаром
- 22 Яким чином в сучасній математиці знімаються умови Адамару стосовно існування єдиного розв'язку задач? Що є найбільшою перешкодою для постановки коректних задач?
- 23 В чому полягає сутність постановки коректних задач за Тихоновим?
- 24 Чому задачі коректності по Тихонову називають умовно-коректними?
- 25 Яка процедура використовується для розв'язання некоректних задач по Тихонову? В чому її сутність?
- 26 Які умови ставляться в методі регуляризації по Тихонову?
- 27 Як залежить розв'язок операторного рівняння від величини регулюючого множника в методі Тихонова?
- 28 Чому метод регуляризації Тихонова є узагальненням метода найменших квадратів Гаусса и метода псевдооберненого оператора Мура-Пенроуза?
- 29 Чим обумовлена стійкість методу регуляризації Тихонова
- 30 За якими методами можна підбирати регулюючий множник?
- 31 Постановка задачі знаходження властивих значень і властивих векторів матриць?
- 32 Зміст понять «характеристичне рівняння»ю «характеристичний визначник»

33 Скільки властивих значень має квадратна матриця розміром  $n \times n$ ? Якими вони можуть бути?

34 Чи є однозначним знаходження властивих секторів матриць?

35 Які особливості мають симетричні матриці з позиції знаходження властивих значень?

36 Сутність методу обертань Якобі знаходження властивих значень і векторів.

37 Який принцип вибору головного елемента в класичному варіанті методу Якобі? В чому його недолік?

38 До чого зводиться алгоритм обертань на кожному кроці методу Якобі?

39 Яку систему властивих векторів можна отримати внаслідок реалізації методу обертань Якобі?

40 Яка матриця використовується за початкове наближення для розрахунків властивих векторів за методом Якобі?

41 Властивості та особливості методу Якобі з точки зору програмування

42 Сутність алгоритму методу Якобі з перешкодами.

43 У чому особливості і достоїнства методу Якобі з перешкодами?

## Завдання для самостійної роботи

### Завдання 4.1.

Розв'язати перевизначену систему рівнянь (табл. 4.4) методом найменших квадратів. Провести дослідження на стійкість, у разі необхідності використати регуляризацию для знаходження стійкого розв'язку, вважаючи, що середнє квадратичне відхилення вільних членів не перевищує 0,2.

Таблиця 4.4 – Варіанти СЛАР для завдання 4.1

Вар .	№	x1	x2	x3	x4	x5	x6	b
1	1	3,65	4,42	11,0 2	- 3,56	0,38	4,72	18,6 2
	2	- 2,36	4,43	4,96	3,22	- 3,86	- 4,31	4,22
	3	- 0,88	0,50	0,29	3,96	0,05	- 5,19	- 7,23
	4	2,64	2,01	6,24	- 0,66	- 3,93	0,81	10,5 7
	5	- 2,43	- 4,01	- 8,31	- 1,80	3,91	2,46	- 7,86
	6	- 2,48	- 0,13	- 2,22	- 2,27	- 4,28	2,90	7,40
	7	1,11	- 0,84	0,27	1,37	- 1,21	- 1,85	- 2,24
	8	- 4,58	- 4,73	- 11,5 3	0,33	- 4,96	- 0,51	- 7,44
	9	2,30	0,58	3,68	2,94	- 0,54	- 3,86	- 2,57
	10	3,67	3,72	9,96	- 3,31	2,07	4,33	15,0 9

Продовження табл. 4.4

Вар	№	x1	x2	x3	x4	x5	x6	b
2	1	2,23	- 0,13	2,44	-0,99	3,51	1,35	0,78
	2	- 0,80	3,36	4,88	2,61	0,95	- 3,42	0,47
	3	- 3,20	1,47	- 0,42	-1,37	- 0,56	1,73	5,18
	4	0,79	0,06	1,28	-4,13	4,88	5,51	5,87
	5	4,86	1,54	7,72	-0,75	- 0,38	0,94	8,06
	6	1,55	0,84	3,31	-4,78	1,32	6,34	12,3 7
	7	1,06	2,34	5,11	3,18	3,81	- 4,15	- 4,12
	8	4,33	4,98	12,5 9	1,25	4,73	- 1,52	6,18
	9	3,70	- 2,11	0,77	0,09	2,78	- 0,03	- 4,00
	10	- 0,95	0,49	0,25	3,85	0,01	- 5,02	- 7,00
3	1	3,31	1,34	5,81	1,75	- 1,07	- 2,30	2,23
	2	4,43	- 1,59	2,39	-1,52	- 3,53	1,91	6,13
	3	- 4,95	- 0,49	- 5,33	0,86	0,47	- 1,12	- 5,25
	4	- 2,04	3,58	4,03	1,74	- 4,90	- 2,41	6,82
	5	1,06	0,23	1,83	0,80	- 4,73	- 1,17	3,76
	6	0,87	4,08	7,66	1,43	1,34	- 1,84	4,84
	7	- 2,12	1,28	0,28	-1,19	- 0,32	1,55	4,85
	8	- 1,96	- 3,16	- 6,51	-0,55	- 0,86	0,68	- 4,77
	9	4,60	3,36	10,2 9	-3,99	3,19	5,33	15,4 9
	10	4,52	2,77	9,25	-4,02	1,11	5,33	15,9 7

Продовження табл. 4.4

Вар	№	x1	x2	x3	x4	x5	x6	b
4	1	2,29	1,74	5,46	-0,13	1,58	0,15	4,14
	2	3,99	- 2,73	0,11	3,28	4,07	-4,23	-12,74
	3	4,13	4,84	12,1 2	-1,59	-2,87	2,00	17,87
	4	3,88	2,57	8,30	0,77	2,65	-0,99	3,77
	5	2,91	4,18	9,93	4,82	-0,12	-6,30	0,38
	6	0,88	- 2,84	- 3,19	1,50	-0,45	-2,02	-7,36
	7	- 1,43	0,65	0,05	4,93	-3,51	-6,52	-6,38
	8	- 3,47	- 1,41	- 5,21	-3,74	-3,93	4,79	7,19
	9	4,27	- 0,58	3,84	-3,64	1,65	4,83	8,27
	10	- 2,15	- 0,98	- 3,23	1,29	-3,19	-1,73	-2,30
5	1	2,28	1,52	5,10	-1,54	-3,56	1,99	11,02
	2	0,76	- 1,20	- 0,71	-4,24	-1,28	5,51	8,75
	3	0,60	- 1,40	- 1,15	-4,07	-0,84	5,35	7,64
	4	- 1,43	- 2,36	- 4,69	4,16	1,93	-5,46	-15,00
	5	- 4,34	- 1,46	- 6,17	-3,14	0,23	4,10	2,03
	6	4,41	- 1,34	2,78	-2,53	-3,28	3,22	8,38
	7	0,05	- 3,13	- 4,47	-3,29	2,74	4,37	-0,62
	8	- 4,11	- 4,28	- 10,3 9	0,69	-1,88	-0,94	-9,70
	9	3,93	3,38	9,60	-0,55	1,49	0,75	9,22
	10	0,83	- 4,53	- 5,79	4,10	-4,07	-5,41	-12,84

Продовження табл. 4.4

Вар	№	x1	x2	x3	x4	x5	x6	b
6	1	2,90	-2,78	-1,08	1,80	- 2,46	- 2,37	-4,94
	2	0,38	-2,19	-2,60	1,15	- 0,30	- 1,54	-5,54
	3	1,74	3,09	6,97	- 1,37	1,84	1,81	8,91
	4	- 0,27	-2,64	-3,99	2,45	0,35	- 3,23	-10,08
	5	4,03	0,41	5,09	- 2,50	- 3,48	3,23	11,75
	6	1,62	0,13	2,27	4,03	2,16	- 5,28	-8,43
	7	- 0,63	-0,13	-0,45	- 0,90	1,93	1,22	0,46
	8	3,40	2,53	7,78	1,94	0,27	- 2,54	2,98
	9	4,72	2,49	9,08	2,00	- 0,91	- 2,64	4,65
	10	2,52	4,05	9,31	- 1,35	- 0,11	1,77	12,80
7	1	- 0,46	-1,31	-2,06	1,98	- 0,53	- 2,60	-5,89
	2	- 1,46	-2,83	-5,42	- 1,73	2,83	2,32	-4,20
	3	- 0,46	-3,08	-4,88	4,45	1,52	- 5,86	-16,25
	4	3,61	-1,02	2,48	1,13	4,15	- 1,41	-5,12
	5	- 3,81	3,30	1,76	- 2,21	- 0,19	2,89	9,76
	6	0,30	4,61	7,95	3,29	- 3,26	- 4,37	5,50
	7	- 2,46	0,83	-0,77	- 1,74	- 4,50	2,24	8,41
	8	2,04	-1,49	0,14	- 4,68	- 1,64	6,11	10,22
	9	- 3,98	3,73	2,28	- 3,68	4,51	4,86	9,84

Продовження табл. 4.4

Вар	№	x1	x2	x3	x4	x5	x6	b
8	1	3,61	0,02	4,08	- 0,69	- 2,35	0,88	6,15
	2	-3,56	- 4,17	- 9,64	2,26	- 0,87	- 2,97	- 13,3 3
	3	3,53	- 4,87	- 3,70	1,47	- 3,11	- 1,95	- 7,43
	4	1,04	0,84	2,77	1,88	2,63	- 2,43	- 3,33
	5	-2,74	3,61	3,35	0,10	- 0,82	- 0,12	6,85
	6	-2,38	- 4,93	- 9,63	- 2,05	- 0,59	2,66	- 5,34
	7	-3,87	3,22	1,62	4,94	- 3,81	- 6,58	- 2,44
	8	-0,94	- 4,67	- 7,78	1,64	1,23	- 2,16	- 13,4 3
	9	2,80	0,66	4,28	- 4,74	1,80	6,24	12,0 6
	10	-3,73	- 3,53	- 8,79	- 3,62	1,39	4,78	- 1,59
9	1	-1,30	- 2,12	- 4,18	- 1,90	- 2,73	2,42	2,11
	2	3,92	1,91	7,34	1,61	4,13	- 2,03	- 0,36
	3	-1,07	- 4,40	- 7,52	- 3,62	- 4,83	4,67	3,18
	4	2,48	2,98	7,54	- 1,68	1,04	2,26	10,4 7
	5	1,77	4,46	9,15	- 4,53	1,92	5,94	18,1 1
	6	4,97	4,27	12,0 9	- 4,88	- 2,41	6,33	23,8 3
	7	3,49	- 3,55	- 1,67	1,77	- 0,87	- 2,34	- 7,39
	8	-4,04	1,91	- 0,62	3,53	3,76	- 4,57	- 8,26

Продовження табл. 4.4

Вар	№	x1	x2	x3	x4	x5	x6	b
10	1	0,49	-0,70	- 0,22	0,42	- 4,80	- 0,62	2,60
	2	-0,68	0,57	0,59	1,43	- 2,42	- 1,87	0,41
	3	-3,35	-1,32	- 5,03	- 0,79	3,49	1,06	- 4,88
	4	3,84	1,16	6,07	1,67	1,20	- 2,20	0,42
	5	-3,77	4,51	3,69	3,52	1,67	- 4,62	- 1,55
	6	-2,49	-2,93	- 6,69	- 3,66	- 2,22	4,72	3,05
	7	2,13	1,30	4,61	- 1,88	- 3,51	2,42	11,3 3
	8	4,28	-3,30	- 0,41	3,64	0,34	- 4,80	- 11,4 1
	9	0,28	3,31	5,85	- 1,95	0,73	2,56	10,7 5
	10	-1,62	4,62	6,05	- 2,79	3,18	3,74	12,0 9
11	1	0,55	2,38	4,67	- 4,66	- 4,22	6,08	18,9 7
	2	-4,78	1,76	- 1,60	2,53	0,15	- 3,30	- 3,85
	3	3,16	3,76	9,44	- 2,31	- 4,68	2,99	18,4 7
	4	-3,88	-2,15	- 6,85	0,59	4,95	- 0,70	- 10,8 6
	5	0,63	-1,71	- 1,62	0,28	0,56	- 0,34	- 3,30
	6	0,25	-2,43	- 3,08	- 3,66	- 0,91	4,77	4,60
	7	1,46	-4,77	- 5,54	- 4,57	2,03	6,02	0,34
	8	3,99	-2,47	0,54	- 1,88	0,08	2,43	1,79

Продовження табл. 4.4

Вар	№	x1	x2	x3	x4	x5	x6	b
12	1	3,90	- 1,95	1,24	- 0,97	1,86	1,31	-0,45
	2	3,13	0,15	3,75	- 1,11	-2,01	1,38	6,52
	3	- 0,78	2,77	4,00	- 3,83	-4,26	4,97	17,34
	4	1,29	3,07	6,53	1,24	-2,20	- 1,66	6,51
	5	- 4,90	0,39	- 3,86	2,50	1,44	- 3,27	-7,66
	6	4,78	1,56	7,70	0,56	-2,57	- 0,82	7,25
	7	- 0,42	- 3,50	- 5,51	- 1,18	-0,99	1,58	-3,07
	8	- 4,20	2,87	0,68	- 0,78	-1,19	1,05	6,72
	9	- 3,87	- 2,17	- 6,87	- 3,72	-4,94	4,81	6,37
	10	3,99	- 0,53	3,56	- 0,51	4,79	0,81	-1,03
13	1	4,81	- 1,18	3,38	0,95	-2,14	- 1,27	0,71
	2	2,60	4,91	10,7 4	4,16	0,13	- 5,43	2,70
	3	- 2,39	2,30	1,61	2,81	-1,67	- 3,75	-0,73
	4	3,66	- 2,86	- 0,42	- 2,29	1,04	2,98	1,04
	5	3,87	- 0,42	3,64	- 0,23	-0,56	0,33	3,03
	6	- 3,87	- 2,30	- 7,09	0,50	-4,49	- 0,72	-3,20
	7	- 0,04	4,10	6,84	- 2,93	-0,83	3,83	15,59
	8	- 2,17	- 0,02	- 1,72	- 4,13	4,72	5,53	4,56
	9	3,08	2,66	7,66	- 1,01	-2,92	1,30	12,08
	10	3,37				0,05	1,19	-2,77

14	1	0,99	4,52	8,46	- 2,19	4,15	2,97	11,26
	2	3,21	- 1,51	1,31	1,79	-4,68	- 2,39	-0,39
	3	2,07	- 2,83	- 1,94	- 3,76	-4,11	4,82	7,63
	4	- 0,85	- 4,21	- 7,01	- 0,86	-3,78	1,03	-3,03
	5	- 0,97	3,28	4,63	- 1,20	2,13	1,63	7,35
	6	4,02	0,56	5,29	4,73	-4,30	- 6,32	-2,68
	7	1,47	4,08	8,33	- 4,82	-1,37	6,35	20,85
	8	2,00	1,94	5,52	2,98	-3,94	- 3,97	2,57
	9	1,44	4,53	8,96	- 2,89	-4,49	3,72	20,09
	10	- 2,96	- 3,14	- 7,46	4,80	2,36	- 6,23	-19,02

Продовження табл. 4.4

Вар	№	x1	x2	x3	x4	x5	x6	b
15	1	3,52	1,08	5,68	4,39	- 2,44	- 5,85	- 2,60
	2	- 4,67	2,75	0,07	- 2,92	4,10	3,89	6,35
	3	4,26	1,82	7,50	2,86	- 0,92	- 3,80	1,11
	4	- 3,99	1,77	- 0,80	3,79	- 4,57	- 5,07	- 2,29
	5	- 0,55	- 1,00	- 1,72	1,38	- 4,23	- 1,89	- 1,10
	6	0,81	3,02	5,96	3,02	- 3,04	- 4,06	2,89
	7	4,11	- 4,11	- 1,85	- 2,87	- 3,70	3,70	4,03
	8	- 3,29	- 0,16	- 3,09	- 2,90	4,92	3,84	0,76
	9	- 0,02	- 2,03	- 2,73	1,61	- 1,42	- 2,10	- 5,43
	10	- 3,54	2,72	1,17	- 1,73	- 4,65	2,17	11,4 8

Завдання 4.2

В середовищі Ooo Calc укласти проект виконання кістякового розкладання, користуючись матричними функціями. На основі кістякового розкладання укласти проект розрахунків псевдо оберненої матриці Мура-Пенроуза і розв'язання системи недовизначених рівнянь з табл. 4.5

Таблиця 4.5 – Варіанти СЛАР для завдання 4.2

Ва р.	№	x1	x2	x3	x4	x5	x6	b
1	1	2,13 8	4,18 8	- 4,55 3	2,30 7	1,53 6	- 4,37 8	3,21 6
	2	1,08 0	- 1,99 9	3,99 8	3,27 7	1,18 1	- 4,35 1	- 2,21 2
	3	4,70 5	4,35 0	- 1,93 7	4,77 6	0,26 2	- 4,04 0	- 2,46 8
	4	- 2,79 9	3,19 4	- 0,79 3	2,62 5	3,60 8	0,72 4	- 0,10 3
	5	3,80 9	5,35 5	- 1,50 8	6,71 4	2,15 1	- 4,94 9	4,71 3
2	1	3,21 1	- 2,03 2	- 1,05 1	- 4,40 4	- 2,76 6	- 4,59 2	- 4,19 8
	2	- 2,07 3	2,09 2	- 1,45 0	3,80 5	- 1,22 0	1,34 8	- 3,80 6
	3	2,73 8	4,21 7	2,92 7	- 0,83 9	- 0,74 1	- 3,75 2	- 1,92 6
	4	0,46 8	- 0,39 1	4,80 5	3,41 6	- 3,65 3	- 4,73 3	4,67 7
	5	2,55 2	4,36 9	4,49 0	1,30 8	- 2,71 6	- 5,59 9	- 2,20 7

Продовження табл.. 4.5

Ва р.	№	x1	x2	x3	x4	x5	x6	b
3	1	- 3,96 7	- 1,72 6	- 0,44 3	- 2,34 3	- 1,52 3	- 1,84 7	- 3,34 9
	2	0,32 0	1,43 3	1,04 1	- 1,14 5	4,49 9	4,39 8	- 0,33 5
	3	0,28 5	- 1,79 7	0,51 0	- 4,75 9	2,02 7	- 4,59 9	3,25 1
	4	- 2,62 4	- 2,32 7	- 0,87 3	0,98 0	- 2,20 6	- 0,71 9	4,94 3
	5	- 0,96 5	- 2,49 0	0,36 1	- 4,65 7	2,06 7	- 3,84 4	- 1,47 3
4	1	4,13 8	- 3,67 0	3,02 0	2,56 5	- 2,89 7	1,78 8	4,28 5
	2	- 4,76 9	- 2,52 0	0,33 0	- 2,79 8	- 1,64 3	- 4,37 9	4,45 3
	3	3,60 1	3,41 4	1,86 7	2,61 4	0,79 3	- 3,32 3	- 1,90 7
	4	- 4,71 1	1,75 2	0,27 4	- 4,93 4	- 0,11 7	4,56 5	0,06 9
	5	0,60 4	3,27 7	2,19 0	- 0,06 3	0,17 2	- 2,29 7	- 1,14 5

Продовження табл.. 4.5

Ва р.	№	x1	x2	x3	x4	x5	x6	b
5	1	- 0,65 7	0,36 5	- 1,67 3	3,43 1	- 2,36 0	- 1,53 8	4,75 1
	2	1,31 0	3,52 4	1,68 1	- 1,98 3	- 2,15 7	- 2,58 6	4,04 7
	3	0,24 7	1,01 7	0,41 2	3,04 7	3,88 1	- 0,42 8	2,51 6
	4	- 3,72 3	- 2,84 0	- 1,40 6	- 0,24 4	- 0,62 3	- 2,87 7	3,56 4
	5	- 1,04 8	0,67 4	0,12 8	2,58 8	2,89 0	- 2,34 7	- 0,34 2
6	1	2,34 1	1,83 3	- 2,17 6	3,07 8	3,86 2	- 0,35 1	- 0,92 9
	2	- 1,38 0	3,07 4	1,37 5	- 2,56 5	- 2,96 5	3,19 0	- 0,67 9
	3	- 0,38 5	- 3,52 2	- 4,06 5	- 0,76 0	- 4,77 3	- 1,34 0	- 3,18 5
	4	2,96 1	- 0,82 9	- 4,15 3	2,85 4	1,23 3	2,95 3	1,22 0
	5	0,66 4	- 2,95 8	- 5,52 4	0,00 4	- 4,67 1	0,69 0	1,92 1

Продовження табл. 4.5

	№	x1	x2	x3	x4	x5	x6	b
7	1	2,94 7	4,93 8	1,39 5	4,94 8	3,07 1	- 3,09 8	3,53 0
	2	0,98 2	0,96 7	- 1,86 7	- 2,07 9	0,80 0	- 0,91 6	- 1,50 2
	3	- 4,96 7	4,79 8	- 3,35 8	- 3,60 3	2,02 7	- 4,74 6	- 3,20 3
	4	1,79 3	3,01 1	3,92 3	- 1,58 4	- 0,58 8	- 1,99 1	- 3,56 4
	5	- 3,71 5	6,47 9	- 2,01 4	- 4,44 0	2,09 9	- 5,88 5	- 1,50 3

# ЛЕКЦІЯ 5. МЕТОДИ РОЗВ'ЯЗАННЯ НЕЛІНІЙНИХ РІВНЯНЬ І СИСТЕМ НЕЛІНІЙНИХ РІВНЯНЬ

## Мета лекції

- 1 Ознайомлення слухачів з постановкою задач розв'язання нелінійних рівнянь і систем нелінійних рівнянь.
- 2 Ознайомлення з методами відокремлення коренів рівнянь, їх локалізації для рівнянь і систем рівнянь;
- 3 Ознайомлення з базовими алгоритмами чисельного розв'язання нелінійних рівнянь та їх систем, аналіз збіжності та умов використання
- 4 Ознайомлення з методами прискорення збіжності, особливостями програмної реалізації методів розв'язання нелінійних рівнянь та їх систем в базових програмних пакетах

## План лекції

- [5.1 Загальні поняття. Відокремлення коренів](#)
  - [5.2 Метод половинного ділення](#)
  - [5.3 Метод хорд](#)
  - [5.4 Метод дотичних \(Ньютона-Рафсона\). Програма «Подбор параметра»](#)
  - [5.5 Метод ітерацій](#)
  - [5.6 Методи Чебишева та Ейткена побудови ітерацій вищих порядків](#)
  - [5.7 Системи нелінійних рівнянь. Відокремлення коренів](#)
  - [5.8 Метод ітерацій для розв'язання систем нелінійних рівнянь](#)
  - [5.9 Метод Ньютона-Рафсона для систем рівнянь](#)
  - [5.10 Розв'язання систем методом найменших квадратів](#)
  - [5.11 Використання методу растрової візуалізації для вивчення області розв'язку систем нелінійних рівнянь](#)
- [Контрольні запитання](#)  
[Завдання для самостійної роботи](#)

## Матеріал лекції

**5.1 Загальні поняття. Відокремлення коренів**

Дано рівняння:

$$f(x)=0, \quad (5.1)$$

де  $f(x)$  – алгебраїчна або трансцендентна функція одного невідомого  $x$ .

Якщо  $f(x^*) = 0$ , то  $x^*$  називається коренем рівняння (5.1) або нулем функції  $f(x)$ .

Будемо припускати, що рівняння (5.1) має лише *ізолювані корені*. Це означає, що навколо кожного кореня існує *окіл, що не містить інших коренів* цього рівняння (рис. 5.1)

Наближене обчислення ізолюваних дійсних корінь рівняння (5.1) складається із двох етапів:

а) *відокремлення коренів* - виділення відрізка, що належить області існування функції  $f(x)$ , на якому розташований *один і тільки один корінь*.

б) *уточнення коренів* - обчислення їх з необхідною точністю, тобто :

$$|f(x)| < \varepsilon, \quad (5.2)$$

де  $\varepsilon$  - мале позитивне число, що *характеризує точність* досягнення розв'язку.

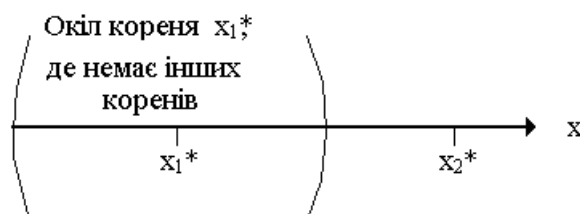


Рисунок 5.1 – Ізолювані корені рівняння (5.1)

Процес відокремлення коренів заснований на теоремі Больцано-Коши: якщо *неперервна* функція  $f(x)$  приймає на кінцях відрізка  $[a;b]$  значення *різних знаків*, тобто

$$f(a) \cdot f(b) < 0, \quad (5.3)$$

те:

- ✓ усередині цього відрізка міститься *принаймні один корінь*;

- ✓ цей корінь буде єдиним, якщо усередині цього відрізка існує похідна  $f'(x)$ , що зберігає усередині відрізка *постійний знак*.

Можливі випадки наведені на рис. 5.2

Відокремлення коренів рівняння найкраще проводити графічним методом, визначаючи методом проб і помилок можливі границі коренів. Це легко зробити в середовищі Excel або Oo Calc, використовуючи Майстра діаграм

Іноді простіше замінити рівняння (5.1) еквівалентним йому рівнянням:  $g(x) = \varphi(x)$ , а потім знайти приблизно абсциси точок перетинання графіків  $y = g(x)$  і  $y = \varphi(x)$ .

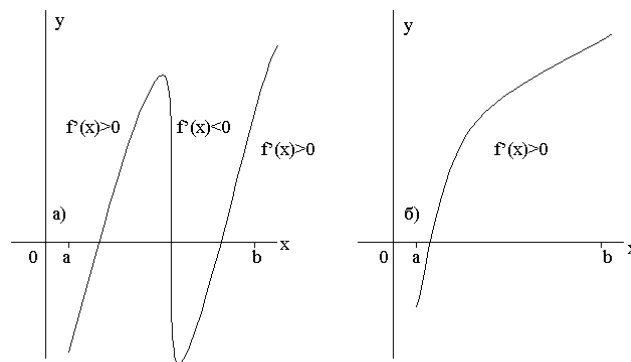


Рисунок 5.2 - Ілюстрація до теореми Больцано-Коши:

- а) випадок, коли на відрізку знак похідній не зберігається (кілька коренів).  
б) випадок постійної за знаком похідної на відрізку (один корінь)

### Приклад 5.1

Відокремити корені рівняння:

$$2x^2 + 3xe^{2x} - 15 = 0 \quad (5.4)$$

Розв'язок:

а) перший спосіб:

1) будемо графік функції, визначаємо, у яких інтервалах відбувається зміна знака функції й перетинання осі абсцис.

2) з рис. 5.3 виходить, що в діапазоні  $(-5,5)$  рівняння (5.4) має 2 корені, відповідно в інтервалах  $(-3;2)$  і  $(0;1)$ .

б) другий спосіб

1) Перепишемо рівняння (4.3) в еквівалентному виді:

$$3xe^{2x} = 15 - 2x^2 \quad (5.5)$$

2) будемо графіки залежностей:  $y = 3xe^{2x}$  і  $n = 15 - 2x^2$  (4.5) і знаходимо точки перетинання кривих (рис. 5.4). З рис. 5.4 виходить, що криві перетинаються у двох точках, одна з яких - в інтервалі  $(-3;-2)$ , а друга - в інтервалі  $(0;1)$ .



### БОЛЬЦАНО БЕРНАРД (1781-1848)

видатний чеський математик, філософ, теолог, логік.). У працях з логічних основ математичного аналізу першим підійшов до арифметичної теорії дійсного числа. При житті Больцано надрукував анонімно тільки п'ять невеликих математичних творів і ряд філософських праць. Основна частина великої рукописної спадщини Больцано біла опублікована майже через 100 років після його смерті. Їм були висунуті базисні теореми й поняття математики, до яких світова наука підійшла істотно пізніше.:

Теорема Больцано-Коші була доведена Больцано в 1817 р., незалежно від Коші, який довів неї в 1821 р.

### КОШІ ОГЮСТЕН ЛУЇ (1789-1857)



великий французький математик, зробив значний внесок у розвиток математичного аналізу, теорії диференціальних рівнянь, теорій функцій комплексних змінних, теорії пружності і механіки твердого тіла Першим дав визначення поняттю неперервності функції. Мав також праці в галузі теорії чисел, алгебри, геометрії, оптиці.

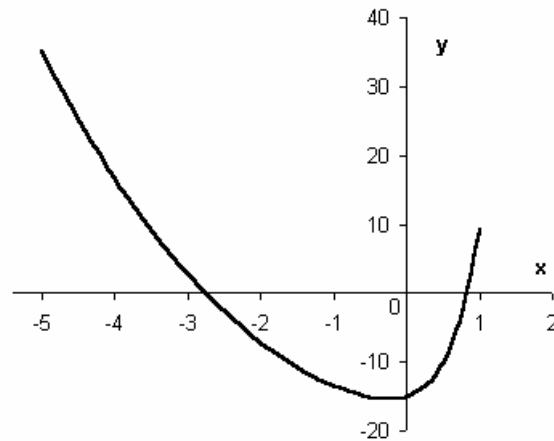


Рисунок 5.3 – Відокремлення кореня за першим способом

При відокремленні коренів першим способом важко довести, що в інших областях, які не були розглянуті, дійсних коренів немає. При використанні другого способу легше довести, оскільки складові функції є більш простими, чим вхідна:

а) при  $x < -3$ :

- ✓ функція  $y = 3e^{2x}$  буде асимптотично прагнути до нуля,
- ✓ функція  $y = 15 - 2x^2$  буде необмежено зменшуватися

Отже, після точки перетинання в інтервалі  $(-3; -2)$  при необмеженому зменшенні  $x$  ці функції мають різні тенденції зміни й не можуть більше перетинатися;

б) при  $x > 1$ :

- ✓ гілка  $y = 3e^{2x}$  буде необмежено зростати;
- ✓ гілка  $y = 15 - 2x^2$  буде необмежено убувати.

Отже, після точки перетинання в інтервалі  $(0; 1)$  при необмеженому збільшенні  $x$  ці функції мають різні тенденції зміни й також не можуть більше перетинатися

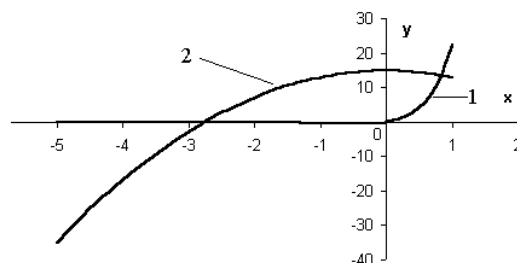


Рисунок 5.4 – Відокремлення коренів за другим способом:

- 1 – гілка  $y = 15 - 2x^2$ ;
- 2 – гілка  $y = 3e^{2x}$

Відзначимо, що умова наявності різних знаків на кінцях інтервалу *не є необхідною* для наявності на цьому інтервалі кореня. Наприклад, функція, що зображена на рис. 5.5, має на кінцях відрізка  $[a,b]$  однакові знаки ("плюс"), але, усередині цього відрізка є корінь у точці  $x^*$ . Така ситуація можлива, для функцій виду:

$$f(x) = (x-c)^{2k} \varphi(x), \quad a < c < b, \quad (5.6)$$

де  $\varphi(x)$  - функція від  $x$ , що не має нулів на відрізку  $[a,b]$  і зберігає постійний знак,  $2k$  - кратність кореня ( $k$ -ціле,  $k > 0$ ).

Приклад - функція

$$f(x) = (x-1)^2 (x^2+1), \quad (5.7)$$

що має корінь  $x=1$ . Він водночас є точкою мінімуму цієї функції.

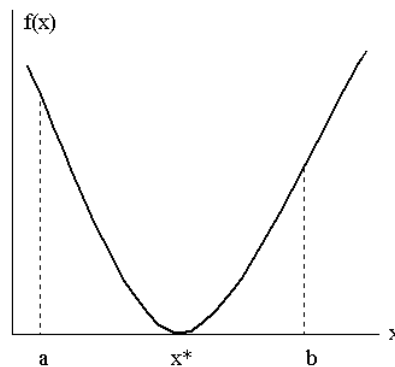


Рисунок 5.5 – Графік функції  $(x-c)^{2k} \varphi(x)$  ( $c=x^*$ )

## 5.2 Метод половинного ділення

Метод половинного ділення (синоніми: "метод дихотомії", "метод бісекції") є найбільш древнім (він був відомий ще давньогрецьким математикам) і найбільш простим і надійним алгоритмом знаходження коріння рівняння (5.1). Він відноситься до *крокових методів*.

Необхідними умовами застосовності методу дихотомії є виконання умов теореми Больцано-Коши (див. 5.1.4), тобто ,

- ✓ наявність *неперервності* функції на відрізку  $[a,b]$ ,
- ✓ *різні знаки* функції на кінцях інтервалу:  $f(a) f(b) < 0$

Постановка задачі: при виконанні наведених умов знайти відрізок довжиною не більше заданого  $\varepsilon > 0$ , усередині якого буде перебувати корінь.

Алгоритм методу (будемо вважати, що  $f(a) < 0$ ;  $f(b) > 0$ , у протилежному випадку треба поміняти позначення):

а) розраховуємо *середню точку* відрізка:

$$c = \frac{a+b}{2}. \quad (5.8)$$

б) розраховуємо значення  $f(c)$ ;

в) перевіряємо виконання умови:

$$|b - a| < \varepsilon; \quad (5.9)$$

1) Якщо (5.9) виконується - завдання вирішене, корінь  $x=c$  (кінець)

2) Якщо (5.9) не виконується - перехід на крок г)

г) Визначаємо знак  $f(c)$ :

1) Якщо  $f(c) < 0$ :

1а) приймаємо:  $a = c$ ,

1б) переходимо на крок а)

2) Якщо  $f(c) > 0$ :

2а) приймаємо  $b = c$ ,

2б) Переходимо на крок а)

Хід розв'язання представлений на рис. 5.6. Індокси ставляться до номерів кроків.

Оцінка кількості кроків методу. На кожному кроці методу дихотомії відбувається скорочення в 2 рази довжини інтервалу локалізації кореня:

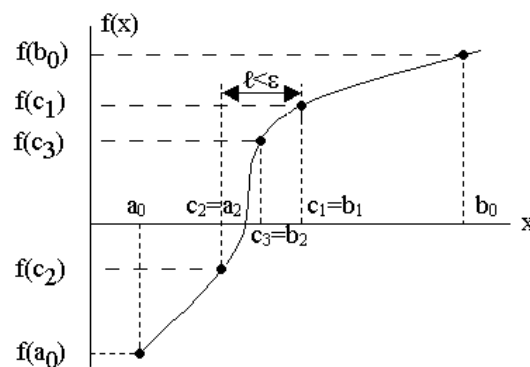


Рисунок 5.6 – Хід розв'язання в методі половинного ділення

а) нехай  $l_0 = |b_0 - a_0|$  довжина первісного інтервалу локалізації кореня;

б) тоді  $\frac{l_0}{\varepsilon}$  - необхідне *кратне зменшення* довжини інтервалу до досягнення необхідної точності;

в) на кожному кроці методу дихотомії довжина інтервалу локалізації *скорочується в 2 рази*. Після  $n$  кроків кратне зменшення довжини складе  $2^n$ ;

г) звідси, одержуємо нерівність для оцінки  $n$ :

$$2^n \geq \frac{\ell}{\varepsilon}. \quad (5.10)$$

Оскільки обидві частини нерівності (5.10) є позитивними, візьмемо від обох частин десятковий логарифм. З врахуванням, що  $\lg 2 = 0,3010 \approx 0,3$ , одержуємо оцінку кількості кроків:

$$n \geq \frac{\lg\left(\frac{\ell}{\varepsilon}\right)}{0,3}. \quad (5.11)$$

При виконанні розрахунків на кожному кроці, крім 1-го, виконується *одне* обчислення значення функції  $f(x)$ . На першому кроці виробляється *3 обчислення* функції (у точках  $a, b, c$ ). Таким чином, кількість обчислень функції (це - сама трудомістка операція) у ході всього процесу складе  $n+2$

### Приклад 5.2

Первісна довжина інтервалу локалізації:  $\ell=1$ . Необхідно знайти розв'язок в інтервалі з довжиною  $\varepsilon < 10^{-6}$ , тобто скоротити інтервал локалізації в мільйон разів. Розрахуємо число кроків методу дихотомії по (5.11):

$$n \geq \frac{\lg(10^6)}{0,3} = \frac{6}{0,3} = 20. \quad (5.12)$$

Обмеження методу дихотомії:

- ✓ метод *не можна* використати для розв'язання рівнянь, для яких функція  $f(x)$  має *однакові знаки* на кінцях інтервалу при наявності кореня усередині інтервалу (див. рис. 5.5);
- ✓ метод *не можна* застосовувати, якщо на інтервалі *коріння не відділені* (тобто на відрізку наявні кілька коренів). У цьому випадку хід рішення *непередбачений*.

*Достоїнство* методу дихотомії - гарантована збіжність до рішення із заданою точністю, якщо дотримані всі умови застосовності методу.

Недолік - порівняно велика кількість обчислень функції, що може бути істотно при масовому рішенні великої кількості рівнянь.

### Приклад 5.3

Розв'язати рівняння:  $2x^2 + 3xe^{2x} - 15 = 0$  в інтервалі  $[0;1]$  методом дихотомії в середовищі Ooo Calc.

Розв'язання

Скриншот проекту представлений на рис. 5.7. Порядок розв'язання:

а) готується форма таблиці (рядок 3).

б) у перший рядок:

- заносять початкові значення  $a$  й  $b$ .

- розраховують :

➤ в комірці H4 - довжину інтервалу, формула комірки:  
 $=ABS(F4-B4);$

➤ в комірці D4 - середина інтервалу, формула комірки:  
 $=ABS(F4-B4);$

➤ в комірках C4,E4,G4 - значення  $f(a)$ ,  $f(b)$ ,  $f(c)$ , відповідно. Завдяки однотипності розрахунків досить створити формулу в одній з комірок і потім скопіювати її в інші. Формула комірки C4:  
 $=2*B4^2+3*B4*EXP(2*B4)-15;$

в) у другому рядку:

➤ створюють формулу комірки B5 (вибір значення для лівого краю нового інтервалу):  
 $=IF(E4<0;D4;B4);$

➤ створюють формулу комірки F5 (вибір значення для правого краю нового інтервалу):  
 $=IF(E4>0;D4;F4);$

➤ виділяють у першому рядку комірки C4:E4 і простягають на другий рядок;

➤ виділяють у першому рядку комірки G4:H4 і простягають на другий рядок;

г) виділяють другий рядок таблиці й простягають її до кінця таблиці. За рішення приймають значення  $x = C$  в останньому рядку.

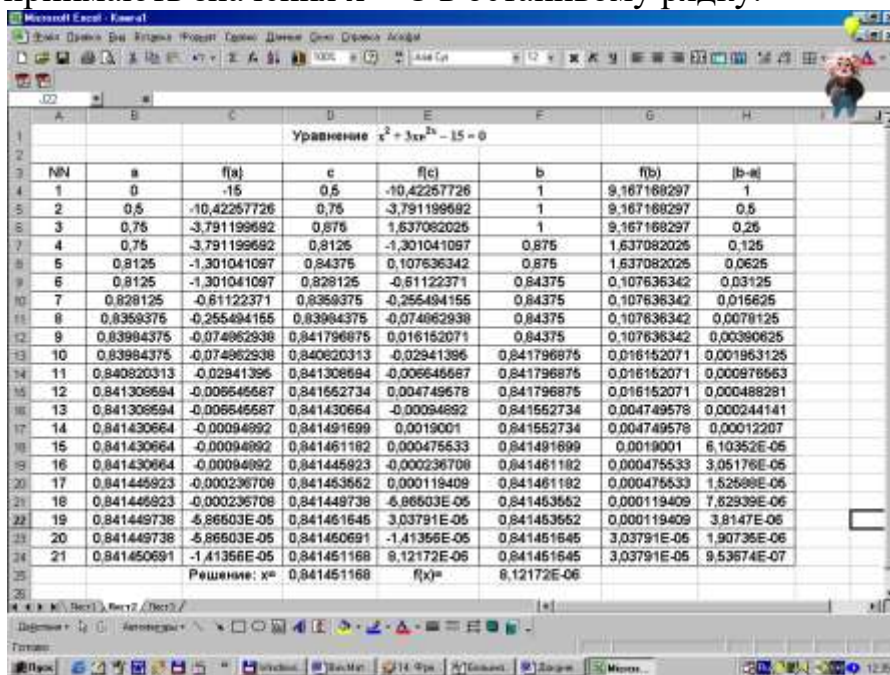


Рисунок 5.7 – Проект «Метод дихотомії»

Як виходить з рис. 5.7, дійсно, на 21 кроці довжина відрізка локалізації стала менш  $10^{-6}$ , як це впливає з теоретичної оцінки. При цьому, до досить малої величини зменшилося й значення функції  $f(x)$ . Це свідчить про правильність розв'язання задачі.

### 5.3 Метод хорд

Метод хорд є природним продовженням методу дихотомії. Цей метод за назвою "метод фальшивого становища" (*regula falsi*) відомий з кінця 16 століття.

Ідея методу:

а) як в методі дихотомії, необхідно, щоб функція була *неперервною* на відрізку  $[a,b]$  і мала на кінцях цього відрізка *різні знаки*.

б) для знаходження приблизного положення кореня:

- замінимо криву хордою АВ (рис. 5.8);
- будемо вважати, що відрізок, який відтинає хордою по осі абсцис, *приблизно дорівнює кореню рівняння  $f(x) = 0$* .
- рівняння хорди АВ (рис. 5.8) має вид:

$$y = a_0 + a_1 \cdot x, \quad (5.13)$$

де невідомі коефіцієнти можуть бути знайдені шляхом розв'язання системи рівнянь виду (5.13) у точках  $x_0$  та  $x_n$ :

$$\begin{cases} a_0 + a_1 \cdot x_0 = f(x_0); \\ a_0 + a_1 \cdot x_n = f(x_n); \end{cases} \quad (5.14)$$

4) розв'язок системи (5.14) має вид:

$$\begin{cases} a_0 = \frac{f(x_0) \cdot x_n - f(x_n) \cdot x_0}{x_n - x_0}; \\ a_1 = \frac{f(x_n) - f(x_0)}{x_n - x_0}; \end{cases} \quad (5.15)$$

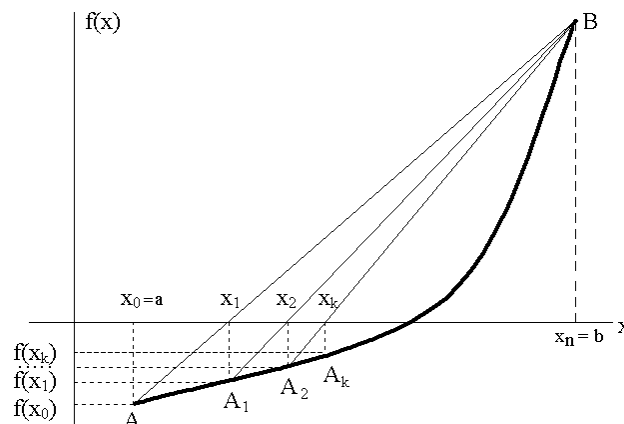


Рисунок 5.8 – Метод хорд

- підставимо у (5.13) коефіцієнти з (5.15) і значення  $y=0$ , знайдемо значення  $x$ , що відповідає точці перетину хорди з віссю  $Ox$ :

$$y = \frac{f(x) \cdot x_n - f(x_n) \cdot x_0}{x_n - x_0} + \frac{f(x_n) - f(x_0)}{x_n - x_0} \cdot x = 0; \quad (5.16)$$

$$x_1 = x = x_0 - \frac{f(x_0) \cdot (x_n - x_0)}{f(x_n) - f(x_0)} = x_0 - \frac{f(x_0) \cdot (b - x_0)}{f(b) - f(x_0)}; \quad (5.17)$$

в) як виходить з рис. 5.8, значення  $x_1$  на перетині хорди з віссю абсцис значно ближче до кореня, ніж  $x_0$ . Використаємо його як нове наближення. Для цього за лівий кінець хорди візьмемо точку  $A_1 = (x_1, f(x_1))$  і проведемо ті ж обчислення. Одержимо:

$$x_2 = x = x_1 - \frac{f(x_1) \cdot (x_n - x_1)}{f(x_n) - f(x_1)} = x_1 - \frac{f(x_1) \cdot (b - x_1)}{f(b) - f(x_1)}; \quad (5.18)$$

д) використовуючи друге наближення, одержуємо третє, і т.д. У підсумку одержуємо нескінченний ітераційний процес послідовного уточнення кореню:

$$x_{k+1} = x_k - \frac{f(x_k) \cdot (x_n - x_k)}{f(x_n) - f(x_k)} = x_k - \frac{f(x_k) \cdot (b - x_k)}{f(b) - f(x_k)}. \quad (5.19)$$

Ітераційний процес проводять доти, доки почне виконуватися умова:

$$|x_{k+1} - x_k| < \varepsilon, \quad (5.20)$$

де  $\varepsilon$  - мале позитивне число, що характеризує точність визначення кореня

Установлено, якщо:

- ✓  $x^*$  - корінь рівняння  $f(x)=0$ ;
- ✓  $f'(x^*) \neq 0$ ,  $f''(x^*) \neq 0$ ;
- ✓  $f'(x^*)$  - неперервна функція,

то існує такий окіл точки  $x^*$ , що якщо  $a$  й  $b$ , що створюють хорду - різні точки у цьому околі, то метод хорд збігається. Швидкість збіжності - трохи вище, ніж у методі дихотомії.

У методі хорд можна фіксувати кожної з кінців, вид формули при цьому не зміниться. Наближення до кореня у методі хорд завжди відбувається з одного боку - із внутрішньої сторони опуклості (увігнутості).

Оскільки передумови для використання методу хорд - такі ж самі, як для методу дихотомії, то методом хорд також не можна розв'язувати рівняння виду, що наведений на рис.5.5. При спробі розв'язання такого рівняння метод хорд починає розбігатися.

#### Приклад 5.4

Розв'язати в ООо Calc методом хорд рівняння  $2x^2 + 3xe^{2x} - 15 = 0$  в інтервалі  $[0;1]$  (див. [приклад 5.3](#)).

## Розв'язання

Проект наведено на скріншоті (рис. 5.9).

Рівняння: $2x^2 + 3xe^{2x} - 15 = 0$				
Метод хорд				
	a	b	f(b)	
	0	1	0,1671683	
k	$x_k$	$f(x_k)$	$x_{k+1}$	$ x_k - x_{k+1} $
0	0,0000000	-15,0000000	0,6206768	0,6206768
1	0,6206768	-7,7863484	0,7948909	0,1742142
2	0,7948909	-2,0450104	0,8323012	0,0374102
3	0,8323012	-0,4218807	0,8398909	0,0073798
4	0,8398909	-0,0824280	0,8411098	0,0014287
5	0,8411098	-0,0150283	0,8413852	0,0002756
6	0,8413852	-0,0030706	0,8414383	0,0000631
7	0,8414383	-0,0005918	0,8414488	0,0000102
8	0,8414488	-0,0001140	0,8414505	0,0000020
9	0,8414505	-0,0000220	0,8414508	0,0000004
10	0,8414508	-0,0000042	0,8414510	0,0000001

Рисунок 5.9 – Реалізація методу хорд

Порядок створення проекту:

а) організуємо комірки C3, D3, E3, куди поміщаємо значення a, b і формулу для розрахунку  $f(b)$ . Формула комірки G3:

$$=2*D4^2+3*D4*EXP(2*D4)-15;$$

б) будуємо таблицю для виконання розрахунків:

- в комірку B7 поміщаємо значення  $x = 0$  для лівого (вільного) кінця хорди;
- в комірку C7 копіюємо з комірки G3 формулу для розрахунку значення функції;
- в комірці D7 створюємо формулу (5.19) для розрахунку  $x_{k+1}$ . Формула комірки D7:

$$=B7-C7*($D$4-B7)/($E$4-C7);$$

- в комірці E7 створюємо формулу для розрахунків  $|x_{k+1} - x_k|$ :

$$=ABS(B7-D7);$$

в) створюємо другий рядок таблиці:

- в комірку B8 адресуємо вміст E7. Формула комірки: =E7;
- виділяємо комірки C7 – E7 і простягаємо їх на рядок 8;

д) виділяємо комірки другого рядка таблиці (B8 – E8) і простягаємо униз на 10-12 рядків. Одержуємо результат.

Як виходить з даних розрахункової таблиці (рис. 5.9), для досягнення точності близько  $10^{-6}$  для методу хорд знадобилося 10 кроків (проти 21 у методі дихотомії). Це вказує, що метод хорд є більш ефективним у порівнянні з методом дихотомії.

#### 5.4 Метод дотичних (Ньютона-Рафсона). Програма «Підбір параметру»

Метод Ньютона-Рафсона (метод дотичних) - один з найбільш ефективних методів рішення нелінійних рівнянь. Цей метод запропоновано незалежно друг від друга І.Ньютоном і Рафсоном в 1685-1690 р. Незважаючи на солідний вік, цей метод дотепер залишається основним у чисельному рішенні рівнянь.

Основні положення методу:

а) В околі ізольованого кореня вибираємо точку  $x_0$ , розраховуємо  $f(x_0)$  і похідну  $f'(x_0)$ ;

б) в околі точки  $x_0$  приблизно представимо функцію по формулі Тейлора, обмежившись першими двома членами розкладання:

$$f(x) \approx f(x_0) + f'(x_0) \cdot (x - x_0); \quad (5.21)$$

в) прийнемо в (5.23)  $f(x) = 0$ :

$$f(x_0) + f'(x_0) \cdot (x - x_0) = 0. \quad (5.22)$$

г) знайдемо з (5.22) приблизне значення кореня:

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}; \quad (5.23)$$

д) значення, що отримали, можна використати як нове наближення  $x_1$  до розв'язку рівняння  $f(x) = 0$ . Виходячи з нього будуємо послідовно друге, третє й наступні наближення, тобто, одержуємо *ітераційний процес* (5.24)

$$\begin{aligned} x_2 &= x_1 - \frac{f(x_1)}{f'(x_1)}; \\ &\dots\dots\dots \end{aligned} \quad (5.24)$$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Геометрична інтерпретація: на кривій  $y = f(x)$  при  $k=0$  через точку  $A = (x_0, f(x_0))$  проводимо дотичну до перетинання з віссю абсцис. (рис. 5.10) Одержуємо на осі абсцис точку  $x_1$ , що буде новим наближенням, і т.д.

Метод дотичних відрізняється *високою швидкістю збіжності*, (він має квадратичну збіжність, тобто швидкість збіжності приблизно пропорційна квадрату різниці між черговим наближенням й розв'язком). Тому якщо відомо добре початкове наближення кореня – метод буде збігатися всього за декілька ітерацій. Однак, на відміну від методу хорд, тут потрібно на кожному кроці обчислювати не тільки значення функції, але її похідній тощо.

На відміну від методу дихотомії й хорд, у методі дотичних для розв'язання необхідна тільки *одна початкова точка*, і це дозволяє успішно вирішувати цим методом рівняння, функції яких мають графіків, подібні зображеним на [рис. 5.5](#).

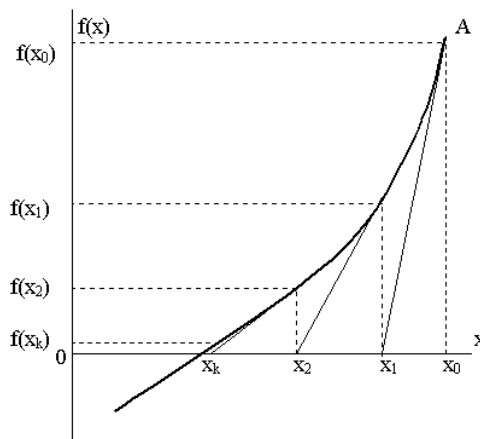


Рисунок 5.10 – Геометрична інтерпретація методу Ньютона-Рафсона

### НЬЮТОН ІСААК (1643 – 1727)



великий англійський математик і фізик, один з творців фізики. Створив диференціальне та інтегральне обчислення, закон всемірного тяжіння, винайшов дзеркальний телескоп, масу наукових відкриттів у математиці, механіці, астрономії. Заслуги в науці був призначений королевою Англії до лицарського чину. На честь Ньютона названо одиницю вимірювання сили.

### РАФСОН ДЖОЗЕФ

англійський математик, член Королівської спілки, послідовник І.Ньютона. В 1690 р., незалежно від І.Ньютона, запропонував метод розв'язання рівнянь

### Приклад 5.5

Розв'язати в області  $\mathbb{C}$  рівняння  $2x^2 + 3xe^{2x} - 15 = 0$  в інтервалі  $[0;1]$  методом Ньютона-Рафсона.

Розв'язання

Скріншот проекту наведено на рис. 5.11. Порядок виконання:

а) розраховуємо вираження для похідної:

$$\frac{df}{dx} = 4 \cdot x + e^{2x} \cdot (3 + 6 \cdot x); \quad (5.25)$$

б) створюємо "шапку" таблиці розрахунків (рядок 4);

k	$x_k$	$f(x_k)$	$f'(x_k)$	D	$x_{k+1}$
0	0.5000000	-10.4225773	-18.3098910	-0.5692383	1.0692383
1	1.0692383	14.5089113	84.1809917	0.1725526	0.9966857
2	0.8968957	2.7852643	53.9767073	0.0516012	0.8452845
3	0.8452845	0.1798421	47.1504941	0.0038142	0.8414703
4	0.8414703	0.0008987	46.8794821	0.0000193	0.8414510
5	0.8414510	0.0000000	46.8770930	0.0000000	0.8414510

Рисунок 5.11 – Проект «Розв'язання рівняння методом дотичних

в) в комірку B5 поміщаємо початкове наближення. Тому що корінь перебуває на відрізку [0;1] - вибираємо за початкове наближення середину цього інтервалу – точку  $x = 0,5$

г) в комірці C5 створюємо формулу для розрахунку функції

д) в комірці D5 створюємо формулу для розрахунку похідної. Формула:  $=4*B5+(3+6*B5)*EXP(2*B5)$ ;

е) в комірці E5 створюємо формулу для розрахунку поправки. Формула:  $=C5/D5$ ;

ж) в комірці F5 створюємо формулу для розрахунку  $x_{k+1}$ :  $=B5-E5$ ;

з) у другому рядку таблиці зв'язуємо комірки B6 і F5. Формула комірки B6:  $=F5$ ;

и) у першому рядку таблиці виділяємо комірки C5 - F5 і простягаємо на другий рядок;

к) виділяємо другий рядок і простягаємо на 8-10 рядків униз. Одержуємо розв'язок.

Як виходить з даних рис. 5.11, метод сходиться дуже швидко, усього за 6 ітерацій досягнута точність  $\sim 10^{-7}$ . При цьому, однак, необхідно обчислювати в кожній точці не тільки функцію, але й похідну.

Відмітимо, що кількість ітерацій залежить від вибору початкового наближення. У розглянутому випадку якщо обрати  $x_0=0$ , то для досягнення розв'язку з точністю до  $10^{-6}$  необхідно 12 ітерацій.

У практичних розрахунках часто аналітичне обчислення похідної буває з ряду причин скрутним (наприклад, коли функція задана алгоритмічно). У цьому випадку прибігають до чисельного розрахунку похідних. Найбільше часто використовують наступну формулу для розрахунку похідної:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} \quad (5.26)$$

де  $h$  - приріст, який, як правило, обирають в межах  $10^{-3} - 10^{-5}$  при  $x \sim 1$ .

При використанні формули (5.27) на кожному кроці доводиться розраховувати 3 значення функції.

Метод дотичних, подібно до методу хорд, також полягає в *однобічному русі* до рішення, але, на відміну від методу хорд, рух іде із *зовнішньої сторони* опуклості (увігнутості).

Ще більшою ефективністю, що істотно при "ручних" розрахунках, має комбінований метод, у якому послідовно проводять спочатку один крок методу хорд, потім, з отриманої точки - 1 крок методу дотичних, після чого цикл повторюють. Це приводить до ще більш швидкої збіжності.

Оскільки розв'язання рівнянь є масовою операцією в інженерно-економічних та наукових розрахунках, у середовищі програм прикладної математики реалізована можливість її автоматичної реалізації за рахунок внутрішніх засобів: функцій або надбудов

В ООо Calc таким внутрішнім засобом є програма "Підбор параметра", що перебуває в меню кнопки "Сервіс". Ця програма реалізує метод Ньютона-Рафсона із чисельним розрахунком похідних.

Для розв'язання рівняння  $f(x)=0$  необхідно створити таблицю, у якій задано:

- ✓ початкове наближення;
- ✓ розрахунок значень  $f(x)$

Після створення таблиці викликається програма «Підбор параметра». Її вікно зображене на скріншоті (рис. 5.12).

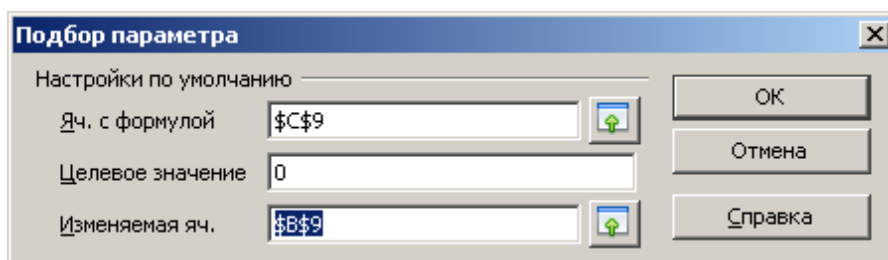


Рисунок 5.12 – Вікно програми «Підбор параметра»

Порядок виконання подальших розрахунків:

- а) у віконце "Яч. С формулой" помістити адресу цільової комірки, де міститься значення  $f(x)$ ;
- б) у віконце "Целевое значение" помістити число 0;
- в) у віконце "Изменяемая ячейка" помістити адреса комірки , у якій перебуває значення  $x$ ;
- г) натиснути клавішу "ОК";
- г) Нажать клавишу «ОК». При цьому з'являється діагностичне вікно (рис. 5.13), воно вказує, що рішення знайдене (або не знайдене) і поточне значення функції.

Далі:

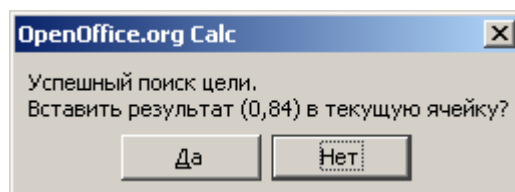


Рисунок 5.13 – Вікно діагностики програми «Подбор параметра»

- ✓ при натисканні "Да" у цільовій комірці й комірці аргументу з'являються рішення (якщо рішення не знайдене - в комірці, де обчислюється функція, з'являється відповідна діагностика)
- ✓ при натисканні клавіші "Нет" у таблиці розрахунків відновлюються вхідні значення величин, що були задані перед розрахунком.

Відсутність рішення може бути обумовлена не тільки його відсутністю, як такого, але, також, вибором невдалого початкового значення. Тому завжди варто попередньо відокремити корінь, а потім - вибирати наближення усередині інтервалу локалізації, що гарантує одержання рішення.

Особливості програми "Подбор параметров":

- ✓ після знаходження розв'язку розрахункова формула у комірці розрахунків *автоматично знищується*. Тому, змінивши вихідні дані, необхідно знову викликати програму й робити розрахунки;
- ✓ програма може бути використана, як об'єктне-орієнтована команда, при програмуванні.

В Scilab для розв'язання нелінійних рівнянь використовується функція:

$$\text{fsolve}(x_0, f),$$

де  $x_0$  – початкове наближення,

$a$  – права частина рівняння, що визначена, наприклад, як підпрограма-функція.

Широкі можливості програмування дозволяють використовувати для розв'язання в Scilab рівнянь вельми складної структури, що містять у тілі рівняння такі чисельні операції, як сумування, інтегрування, розв'язання інших рівнянь і т.п.

### Приклад 5.6

Користуючись Scilab, вивчити залежність кореня рівняння (5.27) від цілочисельного параметру  $n$  у діапазоні від 1 до 10

$$f(x) = \sum_{i=2}^n \frac{i \cdot x^{i-1} - 1}{x^i + 1}. \quad (5.27)$$

### Розв'язання

Спочатку визначимо, у якій області знаходиться розв'язок рівняння (5.27), з тим, щоб обрати початкове наближення. Для цього укладаємо скрипт (лістинг 5.6А), що дозволяє побудувати графіки залежності. З графіка (рис. 5.14) випливає, що для  $n = 1..10$  корень знаходиться в інтервалі  $(0,5;1)$

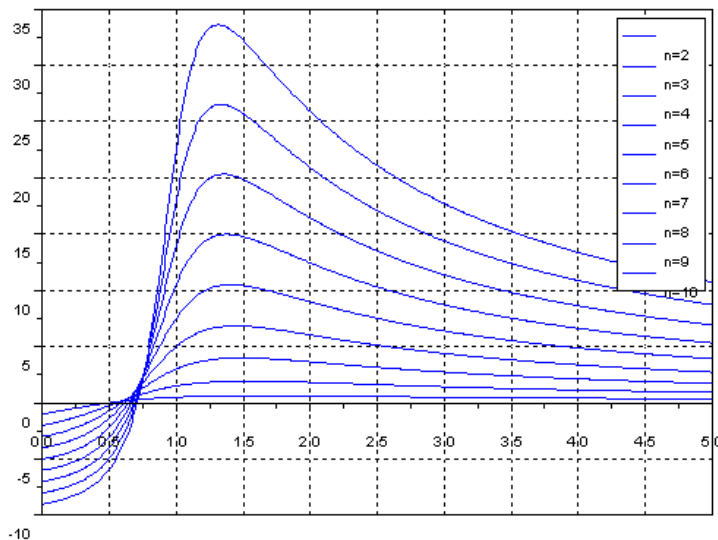


Рисунок 5.14 – Локалізація кореня рівняння (5.27).

### Лістинг 5.5А

```
function y= fi(n,x) // Підпрограма-функція
y=0
for i=1:n
```

```

    y=y+(i*x.^(i-1)-1)./(x.^i+1)
end
endfunction
for n=2:10
x=0:0.01:5;
plot(x,fi(n,x))//побудова графіка
end
xgrid()// сітка
legend('n=2','n=3','n=4','n=5','n=6','n=7',...
'n=8','n=9','n=10') // легенда

```

Для розв'язання рівнянь укладаємо скрипт (лістинг 5.5Б). Для спрощення введено глобальну змінну n, щоб не створювати проблем при роботі функції fsolve. У скрипті передбачено друку таблиці залежності кореня від n. Результати наведені у табл. 5.1

#### Лістинг 5.5Б

```

global n
function y= fi(x)
    y=0
    for i=1:n
        y=y+(i*x.^(i-1)-1)./(x.^i+1)
    end
endfunction
for i = 1:9
    n=i+1;
    z(i)=n;
    xx(i)=fsolve(0.5,fi);
end
[z,xx]// друк z = n і коренів

```

Таблиця 5.1 – Залежність коренів рівняння (5.27) від величини n

n	xx	n	xx	n	xx
2	0,5	5	0,6186558	8	0,6810682
3	0,5505801	6	0,6431430	9	0,6961726
4	0,588634	7	0,6636173	10	0,7094114

Як впливає з табл. 5.1, із зростанням n спостерігається монотонне зростання величини кореня.

Неважко довести, що інших коренів у рівняння (5.27) немає. З рис. 5.14, ліва частина рівняння із зростанням x спочатку зростає, а потім починає зменшуватися і асимптотично прагне до 0 при  $n \rightarrow \infty$ .

## 5.5 Метод ітерацій

Метод ітерацій (послідовних наближень) є загальним методом розв'язання нелінійних рівнянь виду  $f(x) = 0$ . Він багато в чому аналогічний методу ітерацій при розв'язанні систем лінійних рівнянь.

Перший етап - приведення рівняння загального виду до виду, *придатного для методу ітерації*:

$$x = \varphi(x) \quad (5.28)$$

Це можна зробити, якщо покласти:

$$\varphi(x) = x + g(x) \cdot f(x), \quad (5.29)$$

де  $g(x)$  – довільна неперервна знакопостійна функція

Виберемо деяке нульове наближення  $x_0 \in [a, b]$  кореня рівняння (5.29) і підставимо його в праву частину (5.28). Одержимо:

$$x_1 = \varphi(x_0). \quad (5.30)$$

Подальші наближення розрахуємо за формулами:

$$x_{n+1} = \varphi(x_n) \quad (n=1, 2, \dots) \quad (5.31)$$

Якщо послідовність  $\{x_n\}$  *збігається*, тобто існує границя:

$$\lim_{n \rightarrow \infty} x_n = x^*, \quad (5.32)$$

те переходячи до границі в (5.30), одержимо для неперервної функції:

$$\lim_{n \rightarrow \infty} x_{n+1} = \varphi(\lim_{n \rightarrow \infty} x_n) \Rightarrow x^* = \varphi(x^*). \quad (5.33)$$

З (5.33) виходить, що границя  $x^*$  є *коренем* рівняння (5.30).

Розв'язок  $x^*$  є *нерухомою точкою* відображення  $\varphi(x)$ . Тому умови збіжності ітераційного процесу (5.30) можна одержати із принципу стислих відображень (див. [3.4](#)).

Нехай функція  $\varphi(x)$  є *диференціюємою* на  $[a; b]$ , причому всі її значення *належать* відріжку  $[a; b]$ ;

Уведемо *відстань*:

$$\rho(x, y) = |x - y|. \quad (5.34)$$

У цьому разі множина точок відріжку  $[a; b]$  буде *повним метричним простором*;

в) користуючись формулою Лагранжа, отримаємо:

$$\rho(\varphi(x), \varphi(y)) = |\varphi(x) - \varphi(y)| = |\varphi'(c)| \cdot |x - y| \leq q \cdot \rho(x, y), \quad (5.35)$$

де  $c$  – точка, що лежить між  $x$  і  $y$ ;  $x$  і  $y \in [a, b]$ .

З (5.35) виходить: якщо виконується умова:

$$|\varphi'(x)| \leq q < 1 \quad \forall x \in [a, b], \quad (5.36)$$

те:

- ✓ відображення  $\varphi(x)$  буде стискальним;
- ✓ процес ітерації (5.30) буде збігатися до єдиного кореня рівняння (5.28) на відрізку  $[a, b]$ , незалежно від вибору початкового наближення, розташованого на цьому відрізку.

На основі методу стискальних відображень можна одержати оцінку погрішності методу простої ітерації:

$$|x^* - x| \leq \frac{q}{q-1} \cdot |x_n - x_{n-1}| \quad (n = 1, 2, 3, \dots). \quad (5.37)$$

З (5.37) виходить:

- ✓ чим менше  $q$ , тим швидше збіжність;
- ✓ при  $0 < |\varphi'(x)| \leq q < 1$  розрахунки можна припинити, якщо для двох послідовних наближень забезпечується виконання нерівності:

$$|x_n - x_{n-1}| \leq \frac{1-q}{q} \cdot \varepsilon, \quad (5.38)$$

де  $\varepsilon$  - точність проведення розв'язання

Окремі випадки формула (5.38):

а) при  $0 < \varphi'(x) \leq 1/2$  або  $-1 < \varphi'(x) \leq 0$ :

$$|x^* - x_n| < \varepsilon, \quad \text{якщо } |x_n - x_{n-1}| < \varepsilon; \quad (5.39)$$

б) при  $1/2 < \varphi'(x) < 1$  виконання умови  $|x_n - x_{n-1}| < \varepsilon$  не обов'язково спричиняє виконання умови збіжності  $|x^* - x_n| < \varepsilon$ . У цьому випадку критерій зупинки є таким: ітерації варто припинити, якщо для трьох послідовних наближень виконується нерівність:

$$\left| \frac{q}{1-q} (x_n - x_{n-1}) \right| = \frac{(x_n - x_{n-1})^2}{|2x_{n-1} - x_n - x_{n-2}|} < \varepsilon. \quad (5.40)$$

Нерівність (5.40) слід використовувати як критерій закінчення розрахунків, у тих випадках, коли важко або неможливо оцінити значення  $|\varphi'(x)|$ .

Збіжність визначається правильним вибором ітеруючої функції  $\varphi(x)$ . Останню треба вибирати так, щоб виконувалася умова (5.37).

### Приклад 5.7

Підбрати ітеруючу функцію для розв'язання рівняння  $x^2 = a$ .

Розв'язання

а) Перепишемо рівняння  $x^2 = a$  переписати у виді:  $x = a/x$  і обираємо за ітеруючу функцію  $\varphi(x) = a/x$ , одержуємо ітераційний процес:

$$x_{n+1} = \frac{a}{x_n} \quad (5.41)$$

Перевіримо збіжність цього процесу. Оберемо  $a=2$ , за початкове наближення оберемо  $x_0=1$ . Послідовність перших 5 ітерацій:

i	0	1	2	3	4	5
$x_i$	1,000	2,000	1,000	2,000	1,000	2,000

Як випливає з наведених даних, ітераційний процес (5.41) не є збіжним, результати коливаються між  $x_0$  і  $2/x_0$ .

б) розділяємо обидві частини рівняння  $x = a/x$  на 2 і перепишемо у виді:

$$\frac{1}{2}x = x - \frac{1}{2}x = \frac{a}{2x}; \quad x = \frac{1}{2}x + \frac{a}{2x} = \frac{1}{2} \cdot \left( x + \frac{a}{x} \right). \quad (5.42)$$

На основі формули (5.42) організуємо ітераційний процес:

$$x_{n+1} = \frac{1}{2} \cdot \left( x_n + \frac{a}{x_n} \right) \quad (5.43)$$

Формула (5.42) - це формула Герона (див. [приклад 1.1](#)), що збігається дуже швидко.

Висновок: можливість збіжності залежить від вибору ітеруючої функції.

На практиці ітеруючу функцію часто вибирають у виді:

$$\varphi(x) = x - \alpha \cdot f(x). \quad (5.44)$$

Виконання умови (5.37) збіжності ітераційного процесу забезпечується вибором постійного числа  $\alpha$  (ітеруючого множника), що задовольняє умові:

$$|\varphi'(x)| = |1 - \alpha f'(x)| < 1 \text{ або } 0 < \alpha < 2, \quad (5.45)$$

звідки:

$$\alpha f'(x) > 0 \text{ і } |\alpha| \leq \frac{2}{m}, \quad (5.46)$$

де  $m$  – нижня границя модуля похідної  $f'(x)$  на  $[a, b]$ :

$$|f'(x)| \geq m. \quad (5.47)$$

Геометрична інтерпретація методу ітерацій. Для рівняння виду (5.28) будемо 2 лінії:

- ✓  $y = x$
- ✓  $y = \varphi(x)$ .

Точка їх перетинання дає шукане рішення. Можливі випадки розташування прямої і кривої наведени на рис. 5.14. Рух іде по стрілочках "від кривої до прямої".

Як виходить з рис. 5.14, при  $|\varphi'(x)| > 1$  метод ітерацій починає розбігатися навіть у тих випадках, коли початкове наближення обране близько до рішення.

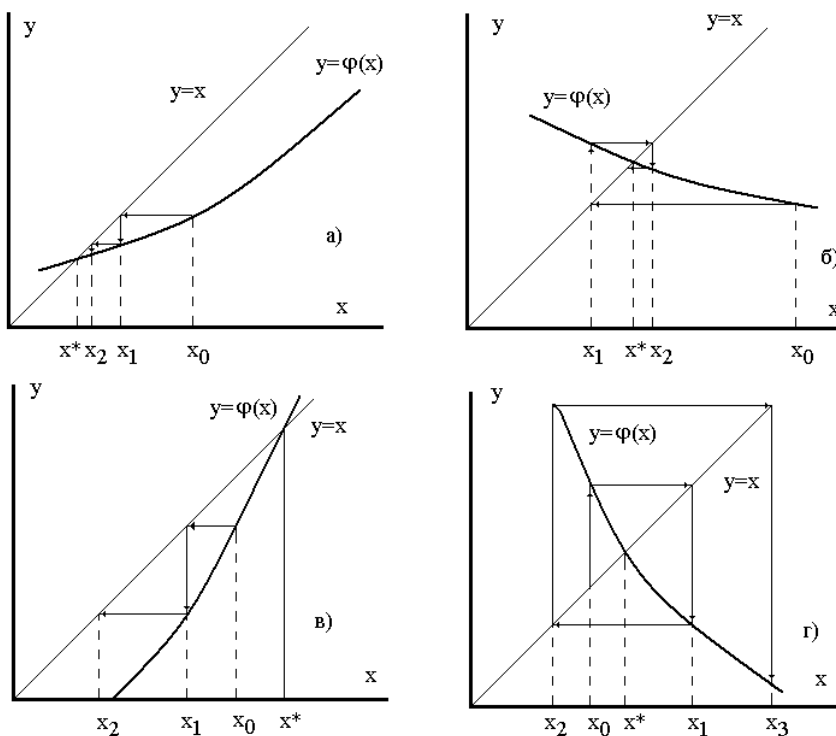


Рисунок 5.14 – Геометрична інтерпретація методу ітерацій:

а)  $0 < \varphi'(x) < 1$ ; б)  $-1 < \varphi'(x) < 0$ ; в)  $\varphi'(x) > 1$ ; г)  $\varphi'(x) < -1$

Метод ітерацій для розв'язання нелінійних рівнянь, також, як ітераційний метод розв'язання систем лінійних алгебраїчних рівнянь, був запропонований німецьким математиком К.Г.Якобі.

## 5.6 Методи Чебишова та Ейткена побудови ітерацій вищих порядків

У 1838 р. П.Л.Чебишов запропонував метод відшукування дійсних корінь рівняння  $f(x)=0$ , частковими випадками якого з'явилися багато методів що були розроблені до нього. В основі методу Чебишова лежить розкладання функції, оберненої до функції  $f(x)$ , по формулі Тейлора.

Нехай рівняння  $f(x) = 0$  на відрізку  $[a,b]$  має корінь  $x = \alpha$ .

Нехай функція  $f(x)$  має наступні властивості:

- ✓ вона є *неперервною* на відрізку  $[a,b]$  разом зі своїми похідними досить високого порядку;
- ✓ на відрізку  $[a,b]$  її похідна не є *нульовою*:  $f'(x) \neq 0$

При цих припущеннях функція  $y = f(x)$  має *обернену функцію*  $x=F(y)$ , яка визначена на відрізку  $[c,d]$ , що є *областю значень*  $f(x)$  при  $x \in [a,b]$ .

Функція  $F(y)$  має *стільки ж* неперервних похідних, *скільки їх має*  $f(x)$ .

За визначенням:

$$x \equiv F(f(x)) \quad (x \in [a,b]); \quad y \equiv f\{F(y)\} \quad (y \in [c,d]), \quad (5.48)$$

те:

$$\alpha = F(0). \quad (5.49)$$



### ЧЕБИШОВ ПАФНУТІЙ ЛЬВОВИЧ (1821-1894)

видатний російський математик і механік. Зробив великий внесок в теорію чисел, теорії наближення функцій, теорію ймовірностей. Був професором Санкт-Петербурзького університету, членом російської Академії наук, багатьох академій Європи. Створив відому наукову школу математиків. Його ім'ям названо декілька об'єктів математики, а також – один з кратерів на Місяці

Запишемо формулу Тейлора для розкладання  $F(0)$  в околі точки  $y \in [c,d]$ :

$$\begin{aligned} F(0) &= \alpha = F(y) + \sum_{k=1}^r \frac{F^{(k)}(y)}{k!} (0-y)^k + R_{r+1} = \\ &= F(y) + \sum_{k=1}^r (-1)^k \frac{F^{(k)}(y)}{k!} y^k + R_{r+1}, \end{aligned} \quad (5.50)$$

где  $R_{r+1}$  – остаточный член разложения:

$$R_{r+1} = (-1)^{r+1} \frac{F^{(r+1)}(\eta)}{(r+1)!} \cdot y^{r+1}, \quad (5.51)$$

де  $\eta \in [0; y]$ .

Підставимо в (5.50), (5.51) вираження  $y = f(x)$ :

$$\alpha = x + \sum_{k=1}^r (-1)^k \frac{F^{(k)}[f(x)]}{k!} [f(x)]^k + (-1)^{r+1} \frac{F^{(r+1)}(\eta)}{(r+1)!} \cdot [f(x)]^{r+1}. \quad (5.52)$$

Для спрощення запису введемо позначення:

$$a_k(x) \equiv F^{(k)}[f(x)]; \quad \varphi_r(x) \equiv x + \sum_{k=1}^r (-1)^k \frac{a_k(x)}{k!} [f(x)]^k. \quad (5.53)$$

Рівняння:

$$x = \varphi_r(x) \quad (5.54)$$

має корінь  $x = \alpha$ , оскільки:

$$\varphi_r(\alpha) = \alpha - \sum_{k=1}^r (-1)^k \frac{a_k(\alpha)}{k!} [f(\alpha)]^k = \alpha, \quad (5.55)$$

де за визначенням  $f(\alpha) = 0$ .

Покладемо:

$$x_{n+1} = \varphi_r(x_n), \quad (n = 0, 1, 2, \dots; x_0 \in [a, b]) \quad (5.56)$$

отримаємо ітераційний метод  $(r+1)$ -го порядку, оскільки для нього всі похідні порядку до  $r$  включно дорівнюють 0:

$$\varphi_r^{(\ell)}(\alpha) = 0; \quad (\ell = 1, 2, \dots, r). \quad (5.57)$$

Якщо точка  $x_0$  лежить достатньо близько до  $\alpha$ , то послідовність збігається, оскільки існує такий окіл точки  $\alpha$ , у який:

$$|\varphi'(x)| \leq q < 1, \quad (5.58)$$

й для збіжності  $\{x_n\}$  треба вимагати, щоб точка  $x_0$  належала до цього окілу.

Функцію  $\varphi_r(x)$  можна виразити в явному виді, якщо провести послідовні диференціювання тотожності (5.48):

$$\begin{aligned}
 F'[f(x)] \cdot f'(x) &= 1; \\
 F''[f(x)] \cdot [f'(x)]^2 + F'[f(x)] \cdot f''(x) &= 0 \\
 F'''[f(x)] \cdot [f'(x)]^3 + 3F''[f(x)] \cdot f'(x) \cdot f''(x) + F'[f(x)] \cdot f'''(x) &= 0
 \end{aligned}
 \tag{5.59}$$

.....

або

$$\begin{aligned}
 a_1(x) \cdot f'(x) &= 1; \\
 a_2(x) \cdot [f'(x)]^2 + a_1(x) \cdot f''(x) &= 0; \\
 a_3(x) \cdot [f'(x)]^3 + 3a_2(x) \cdot f'(x) \cdot f''(x) + a_1(x) \cdot f'''(x) &= 0
 \end{aligned}
 \tag{5.60}$$

З (5.60) можна послідовно знайти  $a_1(x)$ ,  $a_2(x)$ , ... й розрахувати  $\varphi_r(x)$

Часткові випадки:

✓ при  $r = 1$  одержуємо *метод Ньютона*:

$$\varphi_1(x) = x - \frac{f(x)}{f'(x)}; \quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.
 \tag{5.61}$$

✓ при  $r=2$  одержуємо *метод другого порядку*:

$$\begin{cases}
 \varphi_2(x) = x - \frac{f(x)}{f'(x)} - \frac{f''(x)f^2(x)}{2f'^3(x)} \\
 x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f''(x_n)f^2(x_n)}{2f'^3(x_n)}
 \end{cases}
 \tag{5.62}$$

Ітераційні методи високих порядків збігаються швидше, ніж методи більш низьких порядків, але вимагають аналітичного обчислення похідних високого порядку. Для одержання оцінки погрішності й швидкості збіжності:

а) підставимо в рівність (5.52)  $x = x_n$ . З врахуванням (5.51) одержимо:

$$|\alpha - x_{n+1}| = (-1)^{r+1} \frac{F^{(r+1)}[f(\xi)]}{(r+1)!} \cdot [f(x_n)]^{r+1},
 \tag{5.63}$$

де  $\xi \in [\alpha; x_n]$ ;

б) покладемо:

$$L = \max_{x \in [a, b]} |f'(x)|,
 \tag{5.64}$$

$$M_{r+1} = \max_{x \in [a, b]} |F^{(r+1)}[f(x)]|,
 \tag{5.65}$$

в) з врахуванням (5.66), що випливає з теореми Лагранжа, одержимо оцінку (5.67):

$$|f(x_n)| = |f(x_n) - f(\alpha)| = |f'(\eta)| \cdot |x_n - \alpha| \leq L |x_n - \alpha|. \quad (5.66)$$

$$|\alpha - x_{n+1}| \leq \frac{M_{r+1} L^{r+1}}{(r+1)!} \cdot |x_n - \alpha|^{r+1} = q_r |x_n - \alpha|^{r+1}, \quad (5.67)$$

Где  $q = \frac{M_{r+1} L^{r+1}}{(r+1)!}$ ;

г) з (5.67) виходить:

$$|\alpha - x_m| \leq q \sum_{k=0}^{m-1} (r+1)^k |x_0 - \alpha|^{(r+1)^m} = (q |x_0 - \alpha|)^{\frac{(r+1)^m - 1}{r}} |x_0 - \alpha|^{\frac{(r+1)^m (r-1) + 1}{r}}; \quad (5.68)$$

д) таким чином, якщо одночасно:

$$1) |x_0 - \alpha| < 1;$$

$$2) q \cdot |x_0 - \alpha| = \omega < 1,$$

то отримаємо оцінку:

$$|\alpha - x_m| \leq \omega^{\frac{(r+1)^m - 1}{r}}. \quad (5.69)$$

З (5.69) випливає дуже швидко збіжність методу. Наприклад для методу Ньютона ( $r=1$ ) при  $\omega < 0,1$  отримаємо:

$$|\alpha - x_1| \leq 10^{-1}; \quad |\alpha - x_2| \leq 10^{-3}; \quad |\alpha - x_3| \leq 10^{-7}; \quad |\alpha - x_4| \leq 10^{-15} \quad (5.70)$$

тобто кількість вірних десяткових знаків швидко наростає.

О. Ейткен запропонував одержувати ітерації більше високого порядку з декількох послідовних ітерацій того самого порядку.

Нехай є ітерації:

$$x_n^{(1)} = \varphi_1(x_{n-1}^{(1)}), \quad x_n^{(2)} = \varphi_2(x_{n-1}^{(2)}) \quad (n = 1, 2, \dots), \quad (5.71)$$

порядку  $r$ , що збігаються до  $x = \alpha$ .

За допомогою функцій  $\varphi_1(x)$  и  $\varphi_2(x)$  побудуємо функцію  $\Phi(x)$ :

$$\Phi(x) = \frac{x\varphi_1[\varphi_2(x)] - \varphi_1(x) \cdot \varphi_2(x)}{x - \varphi_1(x) - \varphi_2(x) + \varphi_1[\varphi_2(x)]}. \quad (5.72)$$

У цьому випадку доведено, що ітерація:

$$x_n = \Phi(x_{n-1}) \quad (n = 1, 2, \dots) \quad (5.73)$$

має порядок вище за  $r$ , якщо виконується умова:

$$[\varphi'_1(\alpha) - 1] \cdot [\varphi'_2(\alpha) - 1] \neq 0. \quad (5.74)$$

При побудові ітерації (5.73) функцію  $\Phi(x)$  можна знаходити у такий спосіб, що не вимагає необхідності її явного вираження:

а) виходячи з  $x_0$ , послідовно знаходимо:



ЕЙТКЕН ОЛЕКСАНДР КРЕЙГ  
(1895-1967)

англійський математик, професор математики Единбурзького університету. Наукові праці в галузі чисельних методів. Відрізнявся феноменальною пам'яттю і видатними здібностями до усних розрахунків, міг по пам'яті відтворити число  $\pi$  до 500 знаку у будь-якому порядку

$$x_1 = \varphi(x_0), \quad x_2 = \varphi(x_1); \quad (5.75)$$

б) визначаємо  $x_3$  за допомогою співвідношення:

$$x_3 = \frac{x_0 x_2 - x_1^2}{x_0 - 2x_1 + x_2}. \quad (5.76)$$

в) перевіряємо умову збіжності ітераційного процесу. Якщо воно не виконується - приймаємо  $x_0 = x_3$  і повторюємо кроки а) і б).

Описаний процес Ейткена виявляється дуже ефективним для прискорення збіжності ітераційних процесів при розв'язанні нелінійних рівнянь

### Приклад 5.8

Розв'язати рівняння

$$f(x) = x^2 - e^x + 2 = 0$$

методом ітерацій з використанням процесу Ейткена, з початкової точки  $x_0 = 0$ , використовуючи ітерууючий множник  $-0,1$  і методом П.Л.Чебишева

Розв'язання:

а) приводимо рівняння до виду, придатного до ітерації, у відповідності до (5.44):

$$x = x - 0,1 \cdot (x^2 - e^x + 2); \quad (5.76)$$

б) спочатку проводимо розрахунки методом ітерацій без використання методу Ейткена. У результаті встановлення вірного 5-го знака маємо лише після 100-ї ітерації маємо ( $x_{100} = 1,319046$ ,  $\varphi(x_{100}) = 1,319049$ ;

в) Проводимо процес Ейткена:

- перший крок (перші 3 ітерації):

i	$x_i$	$\varphi(x_i)$
0	0	0,1
1	0,1	0,190483
2	0,190483	0,273128
3	1,050741	

- другий крок (нові 3 ітерації):

i	$x_i$	$\varphi(x_i)$
0	1,050741	1,07517
1	1,07517	1,09772
2	1,09772	1,118486
3	1,368379	

- третій крок (нові 3 ітерації):

i	$x_i$	$\varphi(x_i)$
0	1,368379	1,362727
1	1,362727	1,357746
2	1,357746	1,353352
3	1,320758	

- четвертий крок (нові 3 ітерації):

i	$x_i$	$\varphi(x_i)$
0	1,320758	1,320573
1	1,320573	1,320407
2	1,320407	1,32026
3	1,319076	

п'ятий крок (нові 3 ітерації):

i	$x_i$	$\varphi(x_i)$
0	1,319076	1,319075
1	1,319075	1,319075
2	1,319075	1,319075

Як виходить з наведених даних, застосування методу Ейткена привело до збіжності після 15 ітерацій з точністю до 7 знака.

Для порівняння проведемо рішення за допомогою методу Ньютона (другого порядку) і Чебишова (3-го порядку).  $f(x)=x^2 - e^x + 2$ ,  $f'(x) = 2x - e^x$ ,  $f''(x) = 2 - e^x$ :

а) метод Ньютона:  $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$

n	$x_n$	$f(x_n)$	$f'(x_n)$	$f(x_n)/f'(x_n)$	$x_{n+1}$
0	0	1	-1	-1	1
1	1	0,2817182	0,7182818	0,3922112	1,3922112
2	1,3922112	0,0854855	1,2393151	0,068978	1,3232332
3	1,3232332	0,0045981	1,1090778	0,0041459	1,3190873
4	1,3190873	-1,504E-05	1,1018318	1,365E-05	1,3190737
5	1,3190737	-1,622E-10	-1,101808	1,472E-10	1,3190737

б) метод Чебишова третього порядку:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f''(x_n)f^2(x_n)}{2f'^3(x_n)}$$

n	$x_n$	$f(x_n)$	$f'(x_n)$	$f''(x_n)$	$x_{n+1}$
0	0	1	-1	1	1,5
1	1,5	0,2316891	1,4816891	2,4816891	1,3231552
2	1,3231552	0,0045116	1,1089409	1,7552513	1,3190737
3	1,3190737	-5,118E-08	1,1018081	1,7399555	1,3190737

З наведених даних виходить: чим вище порядок методу, тим швидше збіжність.

### 5.7 Системи нелінійних рівнянь. Відокремлення коренів

Загальний вид системи з n нелінійних рівнянь із n невідомими:

$$\begin{cases} f_1(x_1, \dots, x_n) = 0, \\ f_2(x_1, \dots, x_n) = 0, \\ \dots \\ f_n(x_1, \dots, x_n) = 0 \end{cases} \quad (5.77)$$

де  $f_i$  - деякі алгебраїчні або трансцендентні функції.

Векторна форма запису: позначимо:  $\mathbf{x} = (x_1, \dots, x_n)$ ,  $\mathbf{f} = (f_1, \dots, f_n)$ . Звідси, запис системи у векторній формі:

$$\mathbf{f}(\mathbf{x}) = 0. \quad (5.78)$$

Розв'язання системи (5.77) - значно більше складна задача, чим розв'язання одного рівняння. Всі методи розв'язання систем - ітераційні, можливість одержання розв'язку як правило залежить від того, наскільки *вдало обране початкове наближення* - вектор  $\mathbf{x}^0 = (x_1^0, \dots, x_n^0)$ .

Графічне відокремлення коренів, що використовується при розв'язанні рівнянь із один невідомим, можна використати для системи із двох рівнянь:

$$\begin{cases} f_1(x_1, x_2) = 0 \\ f_2(x_1, x_2) = 0 \end{cases} \quad (5.79)$$

Для цього необхідно:

- а) задати серію значень  $x_2 = C_i$ ,
- б) знайти в кожній точці розв'язок рівнянь:

$$\begin{cases} f_1(x_1, C_i) = 0; \\ f_2(x_1, C_i) = 0; \end{cases} \quad (5.80)$$

тобто:

- 1) розрахувати значення  $x_1$ , що відповідають цим значенням  $C_i$ ;
- 2) одержати таблиці значень  $x_1$  як *функції від*  $x_2$  для  $f_1$  і  $f_2$ ;
- в) побудувати на одному графіку обидві криві (для  $f_1$  і  $f_2$ ) і знайти *точки їхнього перетинання*, які й будуть *розв'язками* в даній області.

При цьому необхідне враховувати, що розв'язок системи (5.80) є можливим не при усіх значеннях  $C_i$ .

Якщо у рівняннях можливе відокремити змінні, це значно полегшує знаходження коренів

### Приклад 5.9

Визначити наявність коренів при  $x_1 \in [-4; 4]$ ,  $x_2 \in [-4; 4]$  для системи рівнянь:

$$\begin{cases} e^{x_1} - e^{x_2} - 1 = 0 \\ x_1^3 + x_2^3 - 1 = 0 \end{cases} \quad (5.81)$$

Розв'язання

Перш за все, розв'яжемо кожне з рівнянь системи відносно  $x_1$ :

$$\begin{cases} x_2 = \log(e^{x_1} - 1) \\ x_2 = \sqrt[3]{1 - x_1^3}; \end{cases} \quad (5.82)$$

Область допустимих значень для  $x_2$  з першого рівняння системи (5.82):

$$e^{x_1} - 1 > 0; \quad x_1 > 0. \quad (5.83)$$

Тому замість заданої області  $[-4;4]$  використаємо її частину  $(0;4]$ .

Для зрахунків и побудови графіку використаємо систему Scilab. Укладемо скрипт для виконання розрахунків і побудови графіку (лістинг 5.9)

### Лістинг 5.9

```
for i=1:100
    x1(i)=0.04*i;
    f=(-1+exp(x1(i)));
    x2(i)=log(f);
end
plot(x1,x2,'b')
for i=1:100
    x1(i)=0.04*i;
    z=(1-x1(i)^3);
    x2(i)=abs(z)^(1/3)*sign(z);
end
plot(x1,x2,'r')
xgrid()
```

В результаті виконання скрипту одержуємо графік (рис. 5.15) з якого виходить, що корінь рівняння знаходиться в околі точки  $(0,8;0,4)$ .

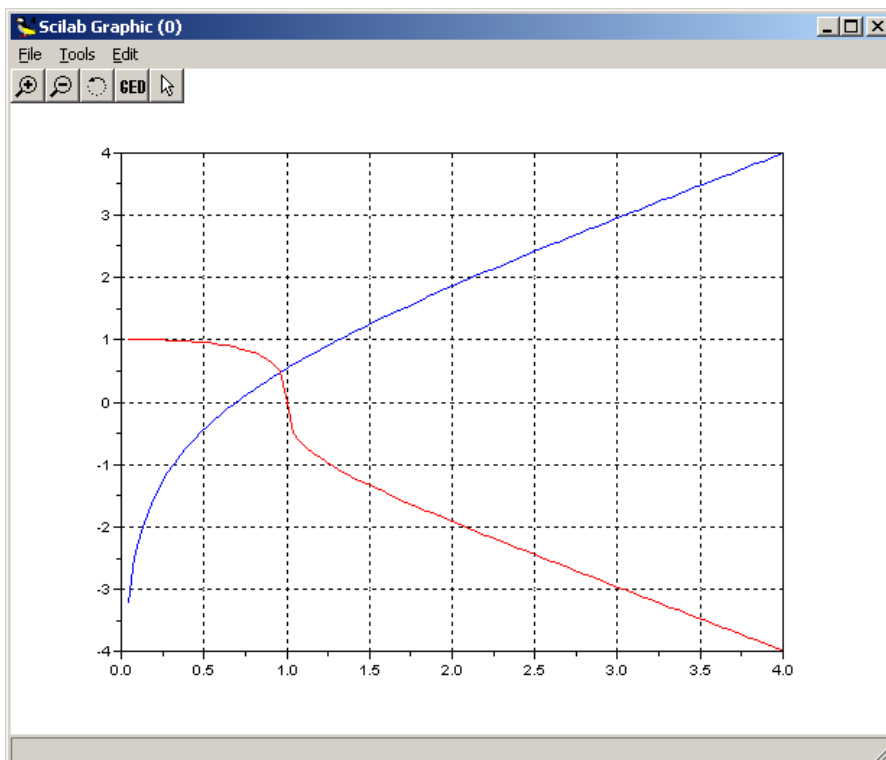


Рис. 5.15 – Скріншот графіків рівнянь (5.82). Сіня лінія – перше рівняння, червона лінія – друге рівняння системи (5.82)

Відмітимо, що в інших обчислювальних середовищах існують засоби для побудови графіків неявних функцій. Наприклад в Maple для розв'язання задачі достатньо записати команду для побудови графіків двох функцій, заданих неявно:

```
> with(plots);
> implicitplot({exp(x1)-exp(x2)-1=0,x1^3+x2^3-1=0}, x1=-4..4,x2=-4..4, axes=boxed,
color=black);
```

### 5.8 Метод простих ітерацій для розв'язання систем нелінійних рівнянь

Замінімо нелінійну систему загального виду (5.78) на еквівалентну систему спеціального виду:

$$\mathbf{x} = \Phi(\mathbf{x}) \quad (5.84)$$

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

$\mathbf{x} = \{x_1, x_2, \dots, x_n\}$  – вектор, компонентами якого є невідомі, що визначаються (розв'язки системи).

Припустимо, що:

- ✓ система (5.82) в обмеженій замкнутій області  $D$   $n$ - вимірного простору має єдиний розв'язок  $\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$ ;
- ✓ компоненти  $x_i^0$  вектору  $\mathbf{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)$  є числами, що близькі до відповідних компонентів  $x_i^*$  вектору  $\mathbf{x}^*$ .

Будемо знаходити наступні наближення до точного розв'язку за допомогою методу послідовних наближень (простої ітерації) за формулами:

$$\mathbf{x}^{k+1} = \Phi(\mathbf{x}^k), \quad (5.85)$$

або у координатній формі:

$$x_i^{k+1} = \varphi_i(x_1^k, x_2^k, \dots, x_n^k), \quad (i=1, 2, \dots, n; k = 1, 2, \dots). \quad (5.86)$$

У цьому розділі, через використання векторних величин,

- *нижній індекс* буде вказувати на номер змінної або рівняння,

- *верхній індекс* буде вказувати на номер ітерації.

Нехай  $\rho(\mathbf{x}^1, \mathbf{x}^2) = \|\mathbf{x}^1 - \mathbf{x}^2\|$  - відстань між елементами  $\mathbf{x}^1$  і  $\mathbf{x}^2$  у просторі  $X$ . За норму вектору можна вибрати будь-яку канонічну норму:

- ✓ евклідову,
- ✓ суму модулів;
- ✓ найбільший модуль.

У відповідність до *принципу стискаючих відображень* система рівнянь (5.84) має *єдиний розв'язок*  $\mathbf{x}^* \in D$ , який може бути знайдено *методом ітерацій* при *будь-якому* виборі початкового наближення  $\mathbf{x}^0 \in D$ , якщо *одночасно*:

- ✓ всі послідовні наближення  $\mathbf{x}^k \in D$  ( $k = 1, 2, \dots$ );
- ✓ відображення  $\varphi(\mathbf{x}) \in D$  *стискаючим* у  $D$ .

Припустимо, що:

а) у деякій опуклій замкнутій області  $D$  функції  $\varphi_i(\mathbf{x})$  мають неперервні часткові похідні  $\frac{\partial \varphi_i}{\partial x_j}$ ;

б) в області  $D$  система (5.82) має *єдиний розв'язок*  $\mathbf{x}^*$ ;

в) нехай для *будь-якого* початкового наближення  $\mathbf{x}^0 \in D$  всі наступні наближення  $\mathbf{x}^k \in D$ .

В околі розв'язку  $\mathbf{x}^*$  у відповідності до узагальненої формули Лагранжа:

$$\mathbf{x}_i^{k+1} - \mathbf{x}_i^* = \varphi_i(\mathbf{x}^k) - \varphi_i(\mathbf{x}^*) = \sum_{j=1}^n \frac{\partial \varphi_i(\mathbf{p}_i^k)}{\partial x_j} \cdot (\mathbf{x}_j^k - \mathbf{x}_j^*) \quad (5.87)$$

$(i = 1, 2, \dots, n)$

где  $\mathbf{p}_i^k$  -  $i$ -та координата деякої точки  $P$ , що знаходиться між  $\mathbf{x}^k$  та  $\mathbf{x}^*$

З цього можна побудувати оцінку для відстані:

$$\begin{aligned} \rho(\mathbf{x}^{k+1}, \mathbf{x}^*) &= \rho(\varphi(\mathbf{x}^k), \varphi(\mathbf{x}^*)) = \|\varphi(\mathbf{x}^k) - \varphi(\mathbf{x}^*)\| = \\ &= \|J(\mathbf{P}^k) \cdot (\mathbf{x}^k - \mathbf{x}^*)\| \leq \|J(\mathbf{P}^k)\| \cdot \rho(\mathbf{x}^k, \mathbf{x}^*), \end{aligned} \quad (5.88)$$

де  $J(\mathbf{x})$  – матриця Якобі (якобіан) системи (5.84):

$$J(\mathbf{x}) = \begin{pmatrix} \frac{\partial \varphi_1(\mathbf{x})}{\partial x_1} & \frac{\partial \varphi_1(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial \varphi_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial \varphi_2(\mathbf{x})}{\partial x_1} & \frac{\partial \varphi_2(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial \varphi_2(\mathbf{x})}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial \varphi_n(\mathbf{x})}{\partial x_1} & \frac{\partial \varphi_n(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial \varphi_n(\mathbf{x})}{\partial x_n} \end{pmatrix} \quad (5.89)$$

У відповідності до *принципу стискаючих відображень* метод послідовних наближень (5.83) збігається до розв'язку  $\mathbf{x}^*$  системи (5.84), якщо *будь-яка* погоджена норма якобіану  $\|J(\mathbf{P}^k)\|$  буде *менше 1*

На практиці найчастіше використовують матрицю  $M$  с елементами:

$$M_{ij} = \max_D \left| \frac{\partial \varphi_i}{\partial x_j} \right|, \quad (5.90)$$

норма якої *мажоруює*<sup>1)</sup> норму якобіана

<sup>1)</sup> Нагадаємо, що функція  $u(x)$  називається *мажоруючою* по відношенню до функції  $y(x)$  при  $x \in D$ , якщо для *будь-якого*  $x \in D$  завжди  $u(x) \geq y(x)$

Посилимо нерівність (5.86) за рахунок введення  $M$ :

$$\rho(\varphi(\mathbf{x}^k), \varphi(\mathbf{x}^*)) \leq \|J(P^k)\| \cdot \rho(\mathbf{x}^k, \mathbf{x}^*) \leq \|M\| \cdot \rho(\mathbf{x}^k, \mathbf{x}^*) \quad (5.91)$$

Тоді відображення (5.82) буде стискаючим в  $D$ , якщо для будь-якої погодженої норми матриці  $M$  виконується умова:

$$\|M\| < 1. \quad (5.92)$$

З огляду на найбільше часто застосовувані норми, достатні умови збіжності методу ітерацій означають дотримання однієї з наступних умов:

$$\begin{cases} \max_{1 \leq i \leq n} \sum_{j=1}^m M_{ij} = q_i < 1; \\ \max_{1 \leq j \leq n} \sum_{i=1}^m M_{ij} = q_j < 1; \\ \sqrt{\sum_{i=1}^n \sum_{j=1}^n M_{ij}^2} = q < 1. \end{cases} \quad (5.93)$$

Закінчення обчислювального процесу фіксується так саме, як при розв'язанні методом ітерацій нелінійного рівняння (див. 5.5):

- ✓ якщо  $\|M\| \leq 1/2$ , то для визначення розв'язку системи (5.82) з точністю  $\varepsilon$  ітераційний процес (5.83) продовжують до виконання умови:

$$\|\mathbf{x}^k - \mathbf{x}^{k-1}\| < \varepsilon; \quad (5.94)$$

- ✓ при  $1/2 < \|M\| < 1$  виконання умови  $\|\mathbf{x}_n - \mathbf{x}_{n-1}\| < \varepsilon$  не обов'язково спричиняє виконання умови збіжності  $\|\mathbf{x}^* - \mathbf{x}_n\| < \varepsilon$ . У цьому випадку ітерації варто припинити, якщо для трьох послідовних наближень виконується нерівність (5.95), що є аналогічним до (5.40):

$$\frac{(x_i^n - x_i^{n-1})^2}{|2x_i^{n-1} - x_i^n - x_i^{n-2}|} < \varepsilon \quad (i = 1, 2, \dots, n) \quad (5.95)$$

У випадку єдиного нелінійного рівняння  $f(x)=0$  його порівняно легко можна привести до виду, придатному для методу ітерації шляхом введення ітеруючого множника. Для систем з декількох рівнянь приведення системи загального виду до виду (5.84), із забезпеченням виконання достатніх умов (5.93) є непростою задачею. Наприклад, для системи із двох рівнянь:

$$\begin{cases} f_1(x, y) = 0; \\ f_2(x, y) = 0; \end{cases} \quad (5.96)$$

перетворення до виду, зручному для методу ітерацій:

$$\begin{cases} x = \varphi_1(x, y); \\ y = \varphi_2(x, y) \end{cases} \quad (5.97)$$

може бути здійснене шляхом комбінування функцій  $f_1(x, y)$  і  $f_2(x, y)$ :

$$\begin{cases} \varphi_1(x, y) = x + \alpha f_1(x, y) + \beta f_2(x, y); \\ \varphi_2(x, y) = y + \gamma f_1(x, y) + \delta f_2(x, y); \end{cases} \quad (5.98)$$

Коефіцієнти  $\alpha, \beta, \gamma, \delta$  можна визначити як наближене рішення системи рівнянь:

$$\begin{cases} 1 + \alpha \frac{\partial f_1(x_0, y_0)}{\partial x} + \beta \frac{\partial f_2(x_0, y_0)}{\partial x} = 0; \\ \alpha \frac{\partial f_1(x_0, y_0)}{\partial y} + \beta \frac{\partial f_2(x_0, y_0)}{\partial y} = 0; \\ \gamma \frac{\partial f_1(x_0, y_0)}{\partial x} + \delta \frac{\partial f_2(x_0, y_0)}{\partial x} = 0; \\ 1 + \gamma \frac{\partial f_1(x_0, y_0)}{\partial y} + \delta \frac{\partial f_2(x_0, y_0)}{\partial y} = 0. \end{cases} \quad (5.99)$$

Якщо частинні похідні не сильно змінюються в околі точки  $(x^0, y^0)$ , то при такому виборі ітеруючих функцій умови (5.93) виконуються.

### Приклад 5.10

Розв'язати методом ітерацій систему:

$$\begin{cases} x_1 = \frac{1}{2} \cdot \ln(x_1^2 + x_2^2) + 2,1; \\ x_2 = \arctg\left(\frac{x_2}{x_1}\right) + 0,1; \end{cases} \quad (5.100)$$

використовуючи за початкове наближення точку  $M_0(3;0)$ . Обчислення проводити до збігу двох послідовних наближень із точністю не менш  $10^{-6}$ .

Розв'язання:

а) будуємо таблицю (табл 5.1), куди послідовно поміщаємо:

- ✓ номер поточної ітерації  $k$ ,
- ✓ поточні значення  $x_1$  і  $x_2$ ;
- ✓ розраховані на кожній ітерації значення:

$$\begin{aligned} \varphi_1(x_1^{(k)}, x_2^{(k)}) &= \\ 0,5 \cdot \ln\{[x_1^{(k)}]^2 + [x_2^{(k)}]^2\} + 2,1; \\ \varphi_2(x_1^{(k)}, x_2^{(k)}) &= \arctg(x_2^{(k)}/x_1^{(k)}) + \\ 0,1; \end{aligned} \quad (5.101)$$

✓ евклідову норму  $\rho_k$ :

$$\rho_k = \sqrt{(x_1^k - x_1^{k-1})^2 + (x_2^k - x_2^{k-1})^2}; \quad (5.102)$$

б) результати використовуємо на наступній ітерації, як нові початкові наближення;

в) процес закінчуємо, коли евклідова норма стає менше  $10^{-6}$ . Як виходить з таблиці 5.1, для цього необхідно 11 кроків;

Таблиця 5.1 – Розв'язання систем рівнянь (5.98) методом ітерацій

k	$x_1^{(k)}$	$x_2^{(k)}$	$\varphi_1(x_1^{(k)}, x_2^{(k)})$	$\varphi_2(x_1^{(k)}, x_2^{(k)})$	$\rho_k$
0	3	0	3,19861 2	0,1	
1	3,1986 12	0,1	3,26320 6	0,131253	0,07175 7
2	3,2632 06	0,1312 53	3,28351 8	0,140201	0,02219 6
3	3,2835 18	0,1402 01	3,28982 6	0,142672	0,00677 5
4	3,2898 26	0,1426 72	3,29177 4	0,143341	0,00205 9
5	3,2917 74	0,1433 41	3,29237 4	0,143518	0,00062 5
6	3,2923 74	0,1435 18	3,29255 8	0,143563	0,00019
7	3,2925 58	0,1435 63	3,29261 4	0,143575	5,76E- 05
8	3,2926 14	0,1435 75	3,29263 2	0,143577	1,75E- 05
9	3,2926 32	0,1435 77	3,29263 7	0,143578	5,3E-06
10	3,2926 37	0,1435 78	3,29263 9	0,143578	1,61E- 06
11	3,2926 39	0,1435 78	3,29263 9	0,143578	4,88E- 07

г) перевіримо, чи можна використати для прискорення збіжності процес Ейткена, застосовуючи його покомпонентно. Для цього після 3-х ітерацій розраховали по формулі Ейткена (5.103) ітерації більш високого порядку.

$$\begin{cases} x_1^{(3)} = \frac{x_1^{(0)}x_1^{(2)} - [x_1^{(1)}]^2}{x_1^{(0)} - 2x_1^{(1)} + x_1^{(2)}} \\ x_2^{(3)} = \frac{x_2^{(0)}x_2^{(2)} - [x_2^{(1)}]^2}{x_2^{(0)} - 2x_2^{(1)} + x_2^{(2)}} \end{cases} \quad (5.103)$$

Результат:

k	$x_1^{(k)}$	$x_2^{(k)}$	$\varphi_1(x_1^{(k)}, x_2^{(k)})$	$\varphi_2(x_1^{(k)}, x_2^{(k)})$	$\rho_k$
3	3,29433 8	0,14546 2	3,293179	0,14412 6	-

Цей результат значно краще (ближче до розв'язку), чим 3-я ітерація в таблиці 5.1. Отже метод Ейткена допомагає прискорити збіжність також й систем рівнянь.

### 5.9 Метод Ньютона-Рафсона для систем рівнянь

Цей метод є поширенням методу дотичних на багатовимірний випадок. Застосовність і умови збіжності методу були запропоновані радянським математиком Л.В.Канторовичем. У матричному виді ітераційний процес відбувається відповідно до формули:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \mathbf{J}^{-1}(\mathbf{x}^k) \cdot \mathbf{f}(\mathbf{x}^k) \quad (k = 1, 2, \dots) \quad (5.104)$$

де  $\mathbf{J}$  – матриця Якобі (якобіан), яка є багатовимірним аналогом похідної:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (5.105)$$

Збіжність методу встановлюється по евклідовій нормі - відстані між  $x_k$  і  $x_{k+1}$ . Ітераційний процес (5.102) збігається до розв'язку  $x^*$  з точністю  $\varepsilon$ , якщо:

$$\|\mathbf{x}^{k+1} - \mathbf{x}^k\| = \sqrt{\sum_{i=1}^n (x_i^{k+1} - x_i^k)^2} < \varepsilon. \quad (5.106)$$

При цьому приблизно вважається:  $x^{k+1} = x^*$

Умови збіжності методу Ньютона-Рафсона: якщо:

а) другі часткові похідні  $f(x)$  є неперервними поблизу розв'язку  $x^*$ ;

б)  $\det \mathbf{J}(x) \neq 0$  в околі кореня;

в) початкове наближення  $x_0$  розташовано поблизу розв'язку,

те метод Ньютона-Рафсона:

✓ буде збігатися,

- ✓ швидкість збіжності буде *квадратичною* (зменшення похибки на кожному кроці пропорційно  $1/n^2$ )

Метод Ньютона-Рафсона - один з основних методів розв'язання систем нелінійних рівнянь. У середовищі Excel або OooCalc без додаткового програмування, реконструювання алгоритму є скрутним (необхідно велика кількість "ручних" операцій копіювання матриць).

Недолік методу: *чутливість* до вибору початкового наближення: при невдалому виборі (вдалі від кореня) метод може збігатися занадто повільно або навіть почати розбігатися.

Особлива незручність методу Ньютона пов'язане з обчисленням якобіана: для системи з  $n$  рівнянь прийде обчислювати  $n^2$  частинних похідних і вносити їх у вигляді підпрограми в програму. Щоб уникнути цього можна використати чисельний розрахунок похідних, як це робилося в одновимірному аналогу (див. 5.4). Для цього фіксують значення всіх змінних, крім тієї, по якій обчислюють похідну, і далі розрахунок ведуть за формулою:

$$\frac{\partial f_i}{\partial x_j} \approx \left[ \frac{f_i(x_j + h) - f_i(x_j - h)}{2h} \right]_{x_1, x_2, \dots, x_{j-1}, x_{j+1}, \dots, x_n} \quad (5.107)$$

Ще одне "проблемне місце" методу Ньютона - необхідність обертання матриці Якобі. Цієї операції можна уникнути, якщо записати систему (5.104) у вигляді (5.108):

$$\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} = \mathbf{v}^{(k)} = -\mathbf{J}^{-1}(\mathbf{x}^{(k)}) \cdot \mathbf{f}(\mathbf{x}^{(k)}). \quad (5.108)$$

де  $\mathbf{v}$  – вектор напрямку від точки (k) до точки (k+1).

Помножимо обидві частини рівняння (5.104) на  $\mathbf{J}(\mathbf{x}^{(k)})$ . Отримаємо систему лінійних рівнянь:

$$\mathbf{J}(\mathbf{x}^{(k)}) \cdot \mathbf{v}^{(k)} = -\mathbf{f}(\mathbf{x}^{(k)}). \quad (5.109)$$



КАНТОРОВИЧ ЛЕОНІД ВІТАЛІЙОВИЧ  
(1912-1986)

Видатний радянський математик, академік АН СРСР, член багатьох академій наук світу один з засновників математичної економіки, один з творців методу лінійного програмування. Виконав багато праць в галузі функціонального аналізу, зокрема в галузі нелінійних просторів і систем нелінійних рівнянь. Лауреат Нобелівської премії з економіки (1976), Ленінської та Державних премій СРСР.

Розв'язання системи (5.109) відносно  $\mathbf{v}^{(k)}$  – це процедура що у порівнянні з алгоритмом обертання матриці, є:

- ✓ більше швидко у виконанні;
- ✓ менш трудомісткою у програмуванні.

Після знаходження вектора напрямку нове наближення розраховується за формулою:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{v}^{(k)}. \quad (5.110)$$

Модифікований метод Ньютона-Рафсона. Для зменшення трудомісткості використають модифікований метод Ньютона-Рафсона, у якому якобіан та його обернену матрицю розраховують *один раз* на початку процесу (у початковій точці  $\mathbf{x}^0$ ) й *далі не змінюють*:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \mathbf{J}^{-1}(\mathbf{x}^0) \cdot \mathbf{f}(\mathbf{x}^k) \quad (k = 1, 2, \dots). \quad (5.111)$$

Порівнюючи (5.111) з методом ітерацій, (5.85) неважко бачити, що обидві формули збігаються з точністю до позначень, тобто вони є ідентичними. Таким чином, виконуючи розрахунки за формулою (5.111), одержуємо прийом, що дозволяє зводити систему загального виду до виду, придатного для ітерації. При цьому, матриця  $\mathbf{J}^{-1}(\mathbf{x}^0)$  відіграє роль матриці ітеруєчих множників кожного рівняння.

### Приклад 5.11

Розв'язати систему рівнянь (5.100), починаючи з точки  $\mathbf{x}(3;0)$ .

Розв'язання:

а) перепишемо систему у стандартній формі:

$$\begin{cases} f_1 = x_1 - \frac{1}{2} \cdot \ln(x_1^2 + x_2^2) - 2,1 = 0; \\ f_2 = x_2 - \arctg\left(\frac{x_2}{x_1}\right) - 0,1 = 0; \end{cases} \quad (5.112)$$

б) розрахуємо аналітично елементи якобіану:

$$\begin{aligned} J_{11} &= \frac{\partial f_1}{\partial x_1} = 1 - \frac{x_1}{x_1^2 + x_2^2}; & J_{12} &= \frac{\partial f_1}{\partial x_2} = -\frac{x_2}{x_1^2 + x_2^2} \\ J_{21} &= \frac{\partial f_2}{\partial x_1} = -\frac{x_2}{x_1^2 + x_2^2}; & J_{22} &= \frac{\partial f_2}{\partial x_2} = 1 - \frac{x_1}{x_1^2 + x_2^2}; \end{aligned} \quad (5.113)$$

в) на робочому аркуші Ooo Calc складемо формули для розрахунку першої ітерації (рис. 5.16):

- у комірці B2, B3 поміщаємо компоненти вектору початкового наближення;

- у комірках C2, C3 проводимо розрахунки правих частин функцій  $f_1, f_2$ .  
Формули комірок:

$$C2: =B3-LN(B3^2+B4^2)/2-2,1;$$

$$C3: =B4-ATAN(B4/B3)-0,1;$$

- в комірках D2:E3 проводимо розрахунки елементів якобіану. Формули комірок:

$$D3 \text{ и } E4: =1-B3/(B3^2+B4^2);$$

$$D4 \text{ и } E3: =-B4/(B3^2+B4^2);$$

- у масиві F3:G4 проводимо обчислення оберненої матриці Якоби. Формула масиву:  $\{=MINVERSE(D3:E4)\}$ ;

- у масиві H3:H4 розраховуємо поправки до рішення шляхом матричного множення, формула масиву:  $\{=MMULT(F3:G4;C3:C4)\}$ ;

- у масиві I3:I4 розраховуємо нове наближення розв'язку. Формула масиву:  $\{=B3:B4-H3:H4\}$ ;

г) далі копіюємо таблицю, замість нульового наближення підставляємо перше й т.д.;

д) за міру збіжності використали мінімум максимального значення  $f_i$ . Як виходить з рис. 4.16, процес збігся за 4 ітерації. При цьому модуль максимального значення функції – порядку  $10^{-7}$ .

Порівнюючи ці результати з методом простих ітерацій, може виникнути відчуття, що метод Ньютона, оскільки зійшовся всього за 4 ітерації, зажадав і менших обчислень. В цьому випадку це не так. На кожному етапі методу ітерацій виробляється 2 обчислення функцій. Але кожному етапі методу Ньютона виробляється обчислення 2-х функцій і 4-х похідних. Якщо прийняти трудомісткість обчислення функції й похідній однаковим, то в цьому випадку метод Ньютона виявляється більше трудомістким. У реальних задачах високої розмірності в разі можливості організації процесу за методом простих ітерацій збіжність як правило виявляється досить поганою, і метод Ньютона стає набагато більше ефективним.

		$x_1$	$x_2$	$f$	Якобіан		$J^{-1}$	D	$x_{i+1}$
Крок 1									
1	1	3.00000	-0.19061	0.88687	0.00000	1.50000	0.00000	-0.29792	3.29792
1	2	0.00000	-0.10000	0.00000	0.69667	0.00000	1.50000	-0.16000	0.16000
Крок 2									
1	1	3.29792	0.00359	0.69740	-0.01376	1.43445	0.02831	0.00528	3.29263
1	2	0.16000	0.00465	-0.01376	0.69740	0.02831	1.43445	0.00663	0.14337
Крок 3									
1	1	3.29263	-4.43E-07	0.69687	-0.01320	1.43561	0.02719	-4.50E-06	3.29264
1	2	0.14337	-1.42E-04	-0.01320	0.69687	0.02719	1.43561	-2.04E-04	0.14358
Крок 4									
1	1	3.29264	-1.90E-09	0.69687	-0.01322	1.43561	0.02723	5.13E-10	3.29264
1	2	0.14358	1.18E-07	-0.01322	0.69687	0.02723	1.43561	1.71E-07	0.14358

Рисунок 5.16 – Розв'язання системи рівнянь методом Ньютона-Рафсона.

Метод Ньютона-Рафсона для розв'язання систем реалізований у середовищі Scilab. Для цього використовується функція:

$$\text{fsolve}([x_1^0, \dots, x_n^0], \text{func}),$$

де  $[x_1^0, \dots, x_n^0]$  – вектор початкових наближень,

func – ім'я вектор-функції лівих частин системи, визначеної у підпрограмі.

### Приклад 5.12

Розв'язати в середовищі Scilab систему (5.112)

Розв'язання:

Скрипт для проведення розрахунків наведено у лістингу 5.12.

Лістинг 5.12

```
function [y]=fi(x)
  y(1)=x(1)-0.5*log(x(1).^2+x(2).^2)-2.1
  y(2)=x(2)-atan(x(2)./x(1))-0.1
endfunction
z=fsolve([3;0],fi)
```

Результат розрахунків:

Z =

3.2926393  
0.1435782

Порівнюючи з результатами, що наведені на рис. 15.6, видно, що вони збігаються.

### 5.10 Розв'язання систем методом найменших квадратів

Можна показати: якщо система рівнянь (5.77) має розв'язок, то задача знаходження цього рішення *еквівалентні мінімізації суми квадратів функцій* – лівих частин рівнянь системи:

$$R = f_1^2 + f_2^2 + \dots + f_n^2 \rightarrow \min, \quad (5.114)$$

тобто

$$\mathbf{x}^* = \arg \min_x (R) \quad (5.115)$$

Вираз (5.115) означає, що  $\mathbf{x}^*$  є *мінімумом функції*  $R$  (аргументом, що неї мінімізує).

Мінімальне значення  $R$  - це  $R = 0$ , воно досягається тоді й тільки тоді, коли кожне зі складових дорівнює 0, а це - еквівалентно розв'язанню системи (5.77).

У принципі, алгоритми розв'язання задач багатовимірної мінімізації значно складніше, ніж для задач розв'язання систем рівнянь. Якщо діяти *ab ovo* (з початку, буквально "від яйця"), тобто самостійно розробляти алгоритм і програму, то розробити програму рішення систем рівнянь значно легше, ніж сучасну ефективну програму оптимізації. Однак, ситуація міняється, якщо в розпорядженні користувача є ефективна програма оптимізації. У цьому випадку він може використати неї для багатьох цілей, у тому числі, розв'язання систем рівнянь.

У середовищі Excel у надбудовах наявна ефективна сучасна програма оптимізації "Поиск решения". Вона дозволяє вирішувати широке коло задач, у тому числі – розв'язання нелінійних рівнянь і систем рівнянь. Ця програма після установки перебуває в меню кнопки "Сервіс". На жаль, у відкритому аналозі Excel, пакеті OOo Calc, надбудова «Поиск решения» дозволяє розв'язувати тільки задачі лінійного програмування.

Перед викликом програми необхідно виконати наступні дії:

а) сформувати таблиці для розрахунків, виділивши цільову комірку - суму квадратів значень функцій;

б) задати деяке початкове наближення й визначити для нього значення цільової функції.

### Приклад 5.13

Розв'язати систему рівнянь (5.81) методом найменших квадратів в середовищі Excel.

Розв'язання:

Приклад організації обчислень наведено на скріншоте (рис. 5.17). У цьому проекті:

- в комірці B7 і C7 заносимо початкові наближення;
- у комірці F7 організуємо обчислення першої функції системи. Формула комірки:  $=\text{EXP}(B7)-\text{EXP}(C7)-1$ ;
- у комірці F8 організуємо обчислення другої функції. Формула комірки:  $=B7^3+C7^3-1$ ;
- в комірках G7,G8 розраховуємо квадрати кожної з функцій;
- в комірці G9 розраховуємо суму квадратів функцій.

Після виклику програми "Поиск решения" з'являється вікно програми (рис. 5.17), куди заносять наступну інформацію:

- у віконце "Установить целевую ячейку" поміщаємо адресу комірки G9 (кляцнувши по осередку курсором);
- прапорець "равной" установимо біля мітки "значению 0";
- у віконце "Изменяя ячейки" помістимо адреси комірок аргументів B7,C7 (шляхом виділення діапазону комірок курсором);

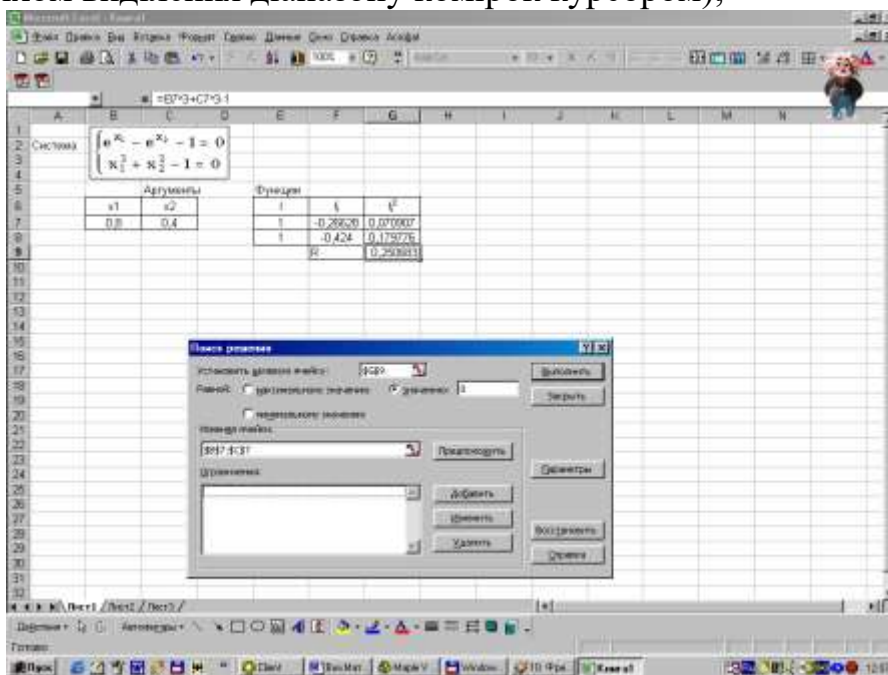


Рисунок 5.17 - Розв'язання системи рівнянь методом найменших квадратів

г) натискаємо кнопку "Виконати". У випадку розв'язання задачі з'являється вікно (рис. 5.18), воно свідчить, що задачу виконано успішно. У цьому випадку треба натиснути кнопку "ОК" і одержати на робочому аркуші розв'язок. Результат розв'язання задачі наведений у таблиці 5.2;

д) якщо розв'язку *не знайдено* - також з'являється інформаційне вікно, що констатує це. У цьому вікні можна залишити отримане найкраще значення, а можна повернутися до первісних значень

Великою позитивною рисою надбудови «Поиск решения» є те, що її, як об'єкт Excel, можна використовувати у складі програм-макросів на мові VBA. Для вставки можна використовувати запис за допомогою вбудованого макрорекордери VBA. Типовий вигляд конструкції, що генерується, має вид:

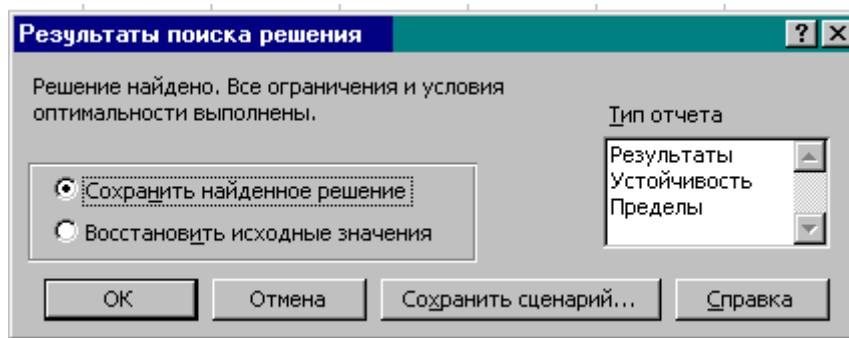


Рисунок 5.18 – Вікно завершення роботи програми «Поиск решения»

Таблица 5.2 – Результаты розв'язання системи рівнянь (рис. 5.17)

Аргументы		Функции		
$x_1$	$x_2$	i	$f_i$	$f_i^2$
0,961778	0,480222	1	-9E-05	8,09E-09
		2	0,000407	1,65E-07
			R	1,73E-07

```
SolverOk SetCell:="$D$105", MaxMinVal:=2, ValueOf:="0",
ByChange:="$B$96:$G$96"
SolverOptions MaxTime:=1000, Iterations:=10000,
Precision:=0.0000001,
AssumeLinear:=False, StepThru:=False, Estimates:=2,
Derivatives:=2,
SearchOption:=1, IntTolerance:=0.00001,
Scaling:=False, Convergence:=0.000001,
AssumeNonNeg:=False
SolverSolve UserFinish:=True
```

В програмній конструкції вказуються:

- ✓ адреси комірок, в яких знаходяться дані для розрахунків,
- ✓ характеристики точності, що встановлюються за допомогою опції «Параметры» надбудови.

Їх можна встановити штучно або за допомогою макро рекордера при запису макросу.

Останній необов'язковий оператор конструкції - SolverSolve UserFinish:=True, він дозволяє позбавитися в автоматичному режимі появи після кожного розрахунку віконця з констатацією закінчення розрахунків.

Для активізації надбудови в середовищі VBA необхідно:

- ✓ на панелі інструментів натиснути «Tools» – References;
- ✓ у віконці поставити прапорець біля надбудови Solver.

Метод найменших квадратів можна реалізувати у середовищі Scilab. Для цього слід використовувати функцію optim(), що має конструкцію:

$$[f, \text{хopt}] = \text{optim}(\text{cst}, \text{x0}),$$

Де f – значення цільової функції (суми квадратів лівих частин) у точці мінімуму хopt,

cst – функція, що повертає цільову функцію та її градієнт,

x0 – початкове наближення мінімуму.

Структура функції cst:

```
function [f,g,ind]=cst(x,ind)
f=gg(x);
g=numdiff(gg,x);
endfunction
```

У цьому блоці:

f – цільова функція, яка визначена, як зовнішня функція gg,

g – градієнт функції, що розраховується чисельно за допомогою функції numdiff(gg,x) (див. частину 2, лекція 1)

Значення параметру ind є внутрішнім параметром, що забезпечує зв'язок між optim й cst, при використанні функції optim параметр ind слід визначати у функції cst.

### Приклад 5.14

Розв'язати у Scilab систему рівнянь (5.81) методом найменших квадратів.

Розв'язання:

Скрипт для проведення розрахунків наведено у лістингу 5.14

### Лістинг 5.14

```
//Початкове наближення x0
x0=[0;0]
//Цільова функція - сума квадратів лівих частин системи
function y=gg(x)
//Увага! x - масив з двох невідомих!.
f1=exp(x(1))-exp(x(2))-1
f2=x(1).^3+x(2).^3-1
y=f1^2+f2^2
```

```

endfunction
//Формування функції cst, що повертає суму квадратів та
//її градієнт
function [f,g,ind]=cst(x,ind)
f=gg(x);
g=numdiff(gg,x);
endfunction
//виклик функції optim, друк результатів
[f,xopt]=optim(cst,x0)

```

У результаті одержуємо:

```

xopt =
    0.9616834
    0.4800138
f =
    1.374D-15

```

Одержаний розв'язок збігається з таким, що розглянуто у прикладах [5.11](#) і [5.12](#) з точністю до 5-го знаку.

На закінчення, відмітимо наступне. Користуючись методом найменших квадратів, можна модифікувати метод Ньютона-Рафсона, додавши йому властивості оптимізаційного процесу. Для цього у формулі (5.111) можна замість фіксованого кроку, що дорівнює одиниці, використати крок змінної довжини, що мінімізує суму квадратів значень функцій:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{h} \cdot \mathbf{v}^{(k)}; \quad (5.116)$$

$$\mathbf{h} = \arg \min_{\mathbf{h}} \left( \sum_{i=1}^n [f_i(\mathbf{x}^{(k)} + \mathbf{h} \cdot \mathbf{v}^{(k)})]^2 \right). \quad (5.117)$$

Такий процес виявляється більше ефективним, ніж класичний варіант методу Ньютона-Рафсона, особливо у випадку задач високої розмірності.

## 5.11 Використання методу растрової візуалізації для вивчення області розв'язку систем нелінійних рівнянь

Методологію методу найменших квадратів можна використовувати для графічної візуалізації області розв'язків нелінійних рівнянь для задач будь якої вимірності. Для цього можна використовувати комп'ютерний метод растрової візуалізації, розроблений автором цього посібника

Сутність методу полягає в наступному:

а) нехай у прямокутній області  $D$  задана *неперервна і обмежена* функція  $n$  змінних  $y=f(\mathbf{x})$ , яка має в області  $D$  екстремум, наприклад мінімум у точці  $\mathbf{x}^{(1)}$ ;

б) розглянемо проекцію функції на площину  $x_k O y$  ( $k = 1, 2, \dots, n$ ). Оскільки функція  $f(x)$  є обмеженою, то *обмеженою* буде також її ортогональна проекція на цю площину;

в) внаслідок наявності мінімуму на проекції нижня границя також буде мати екстремум, що є проекцією екстремуму функції на цю площину;

г) якщо функція в області  $D$  має декілька екстремумів, тобто є *мультимодальною*, то на проекції завжди буде видно по крайній мірі глобальний екстремум (мінімум);

д) критерієм наявності екстремуму є наявність однотипного екстремуму на кожній з проекцій, причому, він завжди відповідає точці мінімуму функції.

Уявлення про характер границь функції можна отримати, якщо:

а) в прямокутній області  $D$  побудувати систему точок  $x_1, x_2, \dots, x_p$ , що рівномірно розташовані всередині цієї області;

б) в кожній точці розрахувати значення функції,

в) побудувати систему графіків проекцій по кожній зі змінних (у вигляді точкового графіку);

г) при наявності великої кількості точок одержуємо «растрове» зображення, на якому чітко видно особливості проекції, на основі яких можна «дешифрувати» особливості функції  $f(x)$ .

Метод растрової візуалізації, сумісно з виразом (5.110), можна використовувати для вивчення наявності коренів системи рівнянь в області  $D$ . Для цього треба побудувати, а потім вивчити растрове зображення проекції по кожній зі змінних функціонала – сумі квадратів лівих частин кожного з рівнянь системи.

### Приклад 5.15.

Визначити кількість коренів системи рівнянь:

$$\begin{cases} f_1(x, y) = x^2 + y^2 - 1 = 0; \\ f_2(x, y) = e^{-x^2} + e^{-y^2} - 1,225 = 0; \end{cases} \quad (5.118)$$

в прямокутній області:  $x \in (-1; 1)$ ,  $y \in (-1; 1)$ .

Розв'язання

а) генеруємо в області  $D$  10000 значень  $x$  і  $y$ , що рівномірно розташовані в цій області. Це можна зробити за допомогою функції  $RAND()$  в  $Ооо\ Calc$ . Зокрема для генерації чисел, що рівномірно розташовані в інтервалі  $(-1; 1)$  формула комірки для розрахунків координат:

$$=-1 * RAND() + 2;$$

б) розраховуємо в кожній точці значення  $f_1(x, y)$ ,  $f_2(x, y)$ , цільової функції  $R = \sqrt{[f_1(x, y)]^2 + [f_2(x, y)]^2}$  (добування кореня приводить до більш зручного графіка, на якому краще видно особливості);

в) будуємо графіки залежності  $R$  від  $x$  і  $R$  від  $y$ . Оскільки задача є симетричною, обидва графіка будуть однаковими, тому приводимо лише один графік (рис. 5.19).

З графіка (рис. 5.19) випливає, що при  $x$  (і, відповідно,  $y$ ) в інтервалах  $(-1;-0,5)$  і  $(0,5;1)$  спостерігається мінімум  $R$ , який дорівнює нулю. Для вивчення деталей збільшено зображення у цих областях (в Excel – протягнемо зображення униз). Як випливає зі збільшеного зображення (рис. 5.20), наявні 4 точки, де може бути  $R=0$ :  $x = \pm 0,6; \pm 0,8$ . Аналогічний набір дає графік залежності  $R$  від  $y$ .

Перевірка шляхом підстановки показує, що з можливих комбінацій набори  $x$  і  $y$ , такі, що приводять до значень  $R \approx 0$ , є наступні:

$(-0,6;-0,8); (-0,6;0,8); (0,6;-0,8); (0,6;0,8)$ ,

всього 4 кореня. За допомогою надбудови «Поиск решения» або скрипту Scilab [прикладу 5.14](#) знайшли більш точні розв'язки:

$(-0,599844;-0,800117); (-0,599844; 0,800117); (0,599844;-0,800117);$   
 $(0,599844;0,800117)$ .

В розглянутому прикладі, що містить 2 змінні, безумовно, можна користуватися «класичним» методом (див. [5.7](#)), який дає такі ж самі результати.

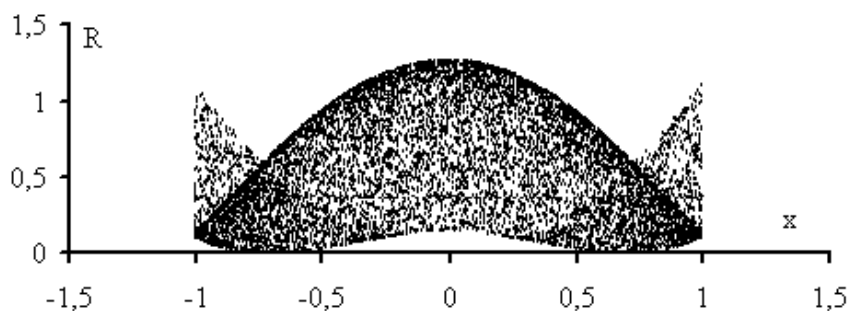


Рисунок 5.19- Залежність  $R$  від  $x$  для прикладу 5.11.5

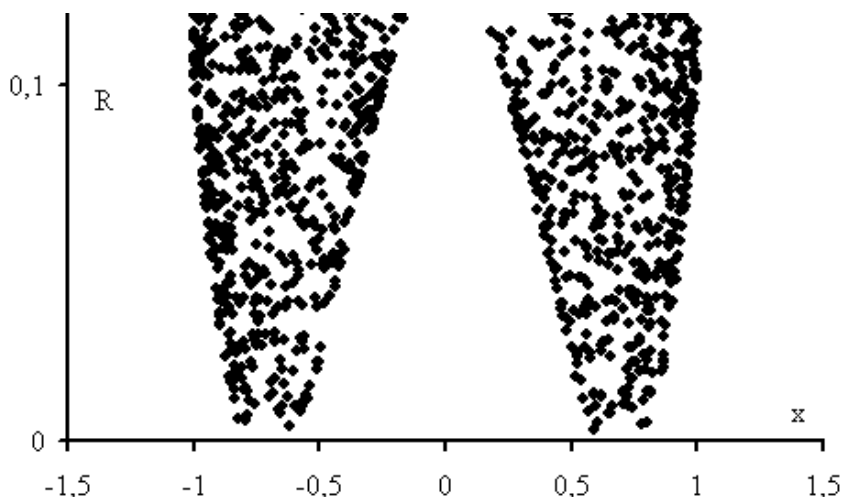


Рисунок 5.20 – Збільшене зображення областей розв'язків рис. 5.19

Оскільки метод растрової візуалізації базується на побудові ортогональних проекції по кожній змінній, він може бути використаний для будь якої кількості

змінних рівнянь системи. Тому він дозволяє знайти кількість коренів і місця їх локалізації в заданій області для задач з кількістю змінних вищою за 2.

### Контрольні питання

- 1 Записати загальну форму для нелінійного рівняння.
- 2 Яка величина називається коренем рівняння?
- 3 Зміст поняття „ізолюваний корень”
- 4 У чому зміст операції відокремлення коренів рівняння?
- 5 Зміст теореми Больцано-Коши. Для чого вона може бути використана на практиці?
- 6 Як можна компактно записати, що функція  $f(x)$  приймає на кінцях відрізка  $[a, b]$  значення різних знаків?
- 7 Яким чином можна проводити відокремлення коренів рівняння?
- 8 Яким чином можна проводити відокремлення коренів системи з двох рівнянь графічним методом?
- 9 Алгоритм методу половинного ділення
- 10 До якого типу методів відноситься метод половинного ділення?
- 11 Яким є необхідні умови використання методу половинного ділення?
- 12 Оцінка кількості кроків методу половинного ділення.
- 13 Які є обмеження методу половинного ділення?
- 14 У чому достоїнства і недоліки методу половинного ділення?
- 15 У чому полягає сутність методу хорд?
- 16 Вивести формулу методу хорд
- 17 Умови закінченні ітераційного процесу методу хорд?
- 18 Умови збіжності методу хорд
- 19 У чому полягає геометрична особливість наближення до кореня методом хорд?
- 20 Які типи рівнянь не можна розв'язувати методом хорд?
- 21 У чому полягає сутність методу дотичних для розв'язання рівнянь з одним невідомим?
- 22 Вивести формулу методу дотичних
- 23 У чому полягає геометрична інтерпретація методу дотичних?
- 24 Зміст поняття „квадратична збіжність”
- 25 У якому методі розв'язання нелінійних рівнянь необхідно розраховувати похідну?
- 26 Які типи рівнянь можна розв'язувати методом дотичних на відміну від методу хорд?
- 27 Яким чином можна застосовувати метод Ньютон в разі неможливості аналітичного розрахунку похідної?
- 28 У чому подібність і різниця в русі до розв'язку у методі дотичних у порівнянні з методом хорд?

- 29 Яка надбудова Excel використовується для розв'язання нелінійних рівнянь? Як нею користуватися?
- 30 У чому полягає сутність методу ітерацій для розв'язання нелінійного рівняння?
- 31 Алгоритм методу ітерацій для розв'язання нелінійного рівняння
- 32 Звідки можна одержати умови збіжності методу ітерації для нелінійного рівняння? У чому вони полягають?
- 33 Який вид мають рівняння, до розв'язання яких можна використовувати метод ітерацій? Яким чином до цього можна привести рівняння загального виду?
- 34 У чому полягають особливості критеріїв закінчення розрахунків для методу ітерацій розв'язання нелінійних рівнянь?
- 35 Яким чином обирають на практиці ітеруючу функцію для розв'язання систем рівнянь методом ітерацій?
- 36 Яким умовам повинен задовольняти ітераційний множник?
- 37 У чому полягає геометрична інтерпретація руху наближень у методі ітерацій? У якому випадку спостерігається розбіжність?
- 38 Навіщо треба створювати ітераційні методи високих порядків, які в них є позитивні риси?
- 39 Який метод називається методом  $r$ -го порядку збіжності?
- 40 У чому сутність методу Ейткена? Для розв'язання яких задач його можна застосовувати?
- 41 Скільки послідовних наближень треба знати, щоб можна було використати метод Ейткена?
- 42 Як в загальному випадку записується система нелінійних рівнянь? Як її можна записати у векторній формі?
- 43 Як записується система нелінійних рівнянь, що є придатною до методу ітерацій?
- 44 Як записати якобіан для правої частини системи нелінійних рівнянь для методу простих ітерацій?
- 45 Яку матрицю використовують на практиці для вивчення збіжності методу ітерацій для розв'язання системи нелінійних рівнянь?
- 46 У чому полягають особливості критерії закінчення розрахунків при розв'язанні систем нелінійних рівнянь методом ітерацій?
- 47 За допомогою якого прийому можна скоротити кількість послідовних наближень при розв'язанні системи нелінійних рівнянь методом ітерацій?
- 48 Як записується у матричному виді алгоритм розв'язання систем нелінійних рівнянь методом Ньютона-Рафсона?
- 49 Що є багатовимірним аналогом похідної в методі Ньютона-Рафсона для систем нелінійних рівнянь?
- 50 Умови збіжності методу Ньютона-Рафсона для систем нелінійних рівнянь.
- 51 Які вимоги існують до початкового наближення у методі Ньютона-Рафсона?)?
- 52 У чому полягають «проблемні місця» методу Ньютона-Рафсона для систем нелінійних рівнянь? Яким чином їх можна вирішити?

53 Як можна позбавитися аналітичного розрахунку якобіану в методя Ньютона-Рафсона?

54 Як можна уникнути процедури обертання матриці Якобі в методі Ньютона-Рафсона для систем нелінійних рівнянь?

55 Як можна модифікувати метод Ньютона-Рафсона, перетворів його на оптимізаційну задачу?

56 Як за допомогою метода Ньютона-Рафсона привести систему нелінійних рівнянь до виду, придатного до методу ітерацій?

57 У чому полягає сутність розв'язання системи нелінійних рівнянь методом найменших квадратів?

58 За допомогою якої надбудови Excel можна розв'язувати систему нелінійних рівнянь методом найменших квадратів?

59 Алгоритм розв'язання систем нелінійних рівнянь в Scilab за методом найменших квадратів.

60 У чому полягає сутність методу растрової візуалізації?

61 Яким чином метод растрової візуалізації можна використовувати для знаходження кількості коренів та їх локалізації у заданій прямокутній області?

## **Завдання для самостійної роботи**

### **Завдання 5.1**

Для нелінійних рівнянь (варіанти завдань наведені у табл. 5.3) знайти кількість коренів у заданій області, розрахувати корені с точністю до  $10^{-5}$  методами:

- ✓ дихотомії,
- ✓ хорд,
- ✓ дотичних,
- ✓ простих ітерацій з використанням методу Ейткена

Перевірити розрахунки за допомогою надбудови «Підбір параметру» Ooo Calc і в Scilab

Таблиця 5.3 – Варіанти рівнянь для завдання 5.1

Вар.	Рівняння	Область
1	$x^2 + x \cdot \cos(2 \cdot \pi \cdot x) - 1 = 0$	[0;2]
2	$x^2 + x \cdot \sin(2 \cdot \pi \cdot x) - 1 = 0$	[-2;2]
3	$x^3 + x^2 \cdot \cos(2 \cdot \pi \cdot x) - x \cdot \sin(2 \cdot \pi \cdot x) + 3 = 0$	[-2;1]
4	$\frac{x^3 - x^2 \cdot \cos(2 \cdot \pi \cdot x)}{1 + e^{0,5 \cdot x}} + 1 = 0$	[-2;1]
5	$x^3 \cdot \cos(3 \cdot \pi \cdot x) \cdot e^x - x^2 \cdot \cos(2 \cdot \pi \cdot x) \cdot e^{0,5 \cdot x} + 5 = 0$	[-5;2]
6	$\sin(2 \cdot \pi \cdot x) \cdot \ln(x+1) + 2 \cdot \ln(x^2+2) \cdot \cos(2 \cdot \pi \cdot x) - 1 = 0$	[-0,5;1]
7	$\frac{x^4 - 1}{x^2 + 3} + 2 \cdot x \cdot \cos(2 \cdot \pi \cdot x) = 0$	[-2;-0,5]
8	$x^2 \cdot \cos(2 \cdot x) + x^{\frac{2}{3}} - 2 = 0$	[0;5]
9	$x^{\frac{2}{3}} \cdot \cos\left(2 \cdot x^{\frac{3}{2}}\right) + x^{\frac{3}{2}} \cdot \sin\left(2 \cdot x^{\frac{2}{3}}\right)$	[0;12]
10	$2 \cdot e^x \cdot \cos(2 \cdot \pi \cdot x) + x + 1$	[-3;0]
11	$x \cdot \ln(x^2 + 1) \cdot \cos(x) - 4 \cdot x + 1 = 0$	[0;15]
12	$x^2 \cdot \ln(x^2 + 1) \cdot \cos(x) - 4 \cdot x + 1 = 0$	[0;15]
13	$x \cdot \ln(x^2 + 1) \cdot \cos(x) - x^{\frac{3}{2}} + 2 \cdot x + 1 = 0$	[0;10]
14	$\sqrt{x} \cdot \ln(x^2 + 1) \cdot \cos(x) - x^{\frac{3}{2}} + 2 \cdot x + 1 = 0$	[0;10]
15	$e^{0,1 \cdot x} \cdot \cos(x) + e^{-0,2 \cdot x} \cdot \sin(x) = 0$	[0;10]
16	$e^{-0,1 \cdot x} \cdot \cos(x) + e^{0,2 \cdot x} \cdot \sin(x) = 0$	[-3;8]
17	$e^{-0,1 \cdot x} \cdot \cos(x^2) + e^{0,2 \cdot x} \cdot \sin(x) = 0$	[0;8]
18	$e^{-0,1 \cdot x} \cdot \cos[\arctg(x)] + x \cdot \cos(x) = 0$	[0;10]
19	$2 \cdot \cos[\arctg(x)] - 3 \cdot \sin(x) + 2 = 0$	[5;12]
20	$2 \cdot \cos[\arctg(x)] + 3 \cdot \sin(x) = 0$	[-3;8]
21	$\sin(x) + \sin(2 \cdot x) + \cos(3 \cdot x) - 1 = 0$	[-1;4]
22	$\sin(x) + \sin(2 \cdot x) + \sin(3 \cdot x) + 1 = 0$	[2;5]
23	$2 \cdot \sin(x) + 3 \cdot \sin(2 \cdot x) + 4 \cdot \sin(3 \cdot x) + 1 = 0$	[0;4]
24	$2 \cdot \sin(x) - 3 \cdot \sin(2 \cdot x) - 4 \cdot \sin(3 \cdot x) + 1 = 0$	[-3;1]
25	$2 \cdot \cos(x) - 3 \cdot \cos(2 \cdot x) - 4 \cdot \cos(3 \cdot x) + 1 = 0$	[1;5]
26	$3 \cdot \cos(x) - 2 \cdot \cos(2 \cdot x) - \cos(3 \cdot x) + 1 = 0$	[0;10]
27	$2 \cdot x^2 \cdot \cos(2 \cdot x) + 3 \cdot x \cdot \sin(x) - 5 = 0$	[-3;5]
28	$2 \cdot \sqrt{x} \cdot \cos(2 \cdot x) + 3 \cdot x \cdot \sin(x) - 5 = 0$	[0;8]
29	$2 \cdot x^2 \cdot \sin\left(\frac{2 \cdot x^2 - 1}{x + 1}\right) + 3 \cdot x \cdot \sin(x) + 1 = 0$	[0;7]
30	$2 \cdot \sqrt{x} \cdot \sin\left(\frac{2 \cdot x^{1,5} - 1}{x + 1}\right) + 3 \cdot x \cdot \sin(x) + 1 = 0$	[0;7]

## Завдання 5.2

Для систем нелінійних рівнянь (табл. 5.4)

а) знайти кількість коренів у заданої області,

б) скласти програму розрахунків методом Ньютона-Рафсона (середовища, що рекомендуються: StarBasic, SciLab) і знайти значення коренів з точністю до  $10^{-5}$ ,

в) Перевірити результати за методом найменших квадратів та за допомогою функції `fsolve()`.

*Рекомендація.* При побудові графічного образу методом растрової візуалізації доцільно після проведення розрахунків впорядкувати таблицю результатів по величині цільової функції і відсікти найбільші її значення, як неінформативні з точки зору знаходження коренів.

Таблиця 5.4 – Варіанти завдання 5.2

Вар.	Рівняння	Область
1	$\begin{cases} x_1^2 + x_2^2 - 3 \cdot x_3 + 4 = 0 \\ x_3 \cdot e^{(2 \cdot x_1 - x_2)} - 3 = 0 \\ x_1 + x_2 + x_3 - 6 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;2) \\ x_2 \in (1;3) \\ x_3 \in (1;4) \end{cases}$
2	$\begin{cases} x_1^2 + x_2^2 - 3 \cdot x_3 - 4 = 0 \\ x_3 \cdot e^{(2 \cdot x_1 - x_2)} - 3 = 0 \\ x_1 + x_2 + x_3 - 6 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;2) \\ x_2 \in (0;3) \\ x_3 \in (0;4) \end{cases}$
3	$\begin{cases} x_1^2 + x_2^2 - 3 \cdot x_3 - 4 = 0 \\ x_3 \cdot e^{(2 \cdot x_1 - x_2)} - 3 = 0 \\ x_1 \cdot x_2 \cdot x_3 - 6 = 0 \end{cases}$	$\begin{cases} x_1 \in (-2;2) \\ x_2 \in (-3;3) \\ x_3 \in (0;4) \end{cases}$
4	$\begin{cases} x_1^2 + x_2^2 - 3 \cdot x_3 - 4 = 0 \\ x_3 \cdot e^{(2 \cdot x_1 + x_2)} - 3 = 0 \\ x_1 \cdot x_2 \cdot x_3 - 6 = 0 \end{cases}$	$\begin{cases} x_1 \in (-2;2) \\ x_2 \in (-3;3) \\ x_3 \in (0;4) \end{cases}$
5	$\begin{cases} x_1 \cdot \sin(x_2) + x_2 \cdot \sin(x_1) - 2,5922 = 0 \\ x_1^2 + x_2^2 + x_3^2 - 14 = 0 \\ x_1 \cdot x_2 \cdot x_3 - 6 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;3) \\ x_2 \in (0;3) \\ x_3 \in (0;4) \end{cases}$
6	$\begin{cases} \sin(x_1 \cdot x_2) + \sin(x_2 \cdot x_3) - 0,6299 = 0 \\ x_1 + x_2^2 - x_3 - 1 = 0 \\ x_1 \cdot x_2 \cdot x_3 - 6 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;3) \\ x_2 \in (0;3) \\ x_3 \in (0;4) \end{cases}$
7	$\begin{cases} x_1^4 + x_2^4 - x_3 - 14 = 0 \\ x_1^2 + x_2^2 + x_3^2 - 14 = 0 \\ x_1 + x_2 + x_3 - 6 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;3) \\ x_2 \in (0;3) \\ x_3 \in (0;4) \end{cases}$

## Продовження табл. 5.4

Вар.	Рівняння	Область
8	$\begin{cases} x_1 \cdot e^{-x_2} + x_2 \cdot e^{-x_3} - 1,3423 = 0 \\ x_1^2 + x_2^2 - 4 \cdot x_3^2 - 4 = 0 \\ x_1 \cdot x_2 \cdot x_3 - 1 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;3) \\ x_2 \in (0;3) \\ x_3 \in (0;2) \end{cases}$
9	$\begin{cases} x_1 \cdot x_2 \cdot e^{-x_2} - 0,7358 = 0 \\ x_1^2 + x_2^2 + x_3^2 - 6 = 0 \\ x_1 \cdot x_2 - e^{x_3} + 0,7183 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;3) \\ x_2 \in (0;3) \\ x_3 \in (0;3) \end{cases}$
10	$\begin{cases} x_1 \cdot x_2 \cdot e^{-x_2} - 0,7358 = 0 \\ x_1^2 + x_2^2 + x_3^2 - 6 = 0 \\ x_1^3 + x_2^3 + x_3^3 - 10 \end{cases}$	$\begin{cases} x_1 \in (0;3) \\ x_2 \in (0;3) \\ x_3 \in (0;3) \end{cases}$
11	$\begin{cases} x_1 \cdot x_2 \cdot e^{-x_2} - 0,7358 = 0 \\ x_1^2 + x_2^2 + x_3^2 - 6 = 0 \\ x_1^3 + x_2^3 + 2 \cdot x_3^3 - 11 \end{cases}$	$\begin{cases} x_1 \in (0;3) \\ x_2 \in (0;3) \\ x_3 \in (0;3) \end{cases}$
12	$\begin{cases} x_1 \cdot x_2 \cdot e^{-x_2} + x_3 - 1,7358 = 0 \\ x_1^2 + x_2^2 - x_3^2 - 4 = 0 \\ x_1^3 + x_2^3 + 2 \cdot x_3^3 - 11 \end{cases}$	$\begin{cases} x_1 \in (0;3) \\ x_2 \in (0;3) \\ x_3 \in (0;3) \end{cases}$
13	$\begin{cases} x_1 \cdot x_2 \cdot e^{-x_2} + x_3 - 1,7358 = 0 \\ \sqrt[3]{x_1^2 + x_2^2} - \sqrt{x^2} - 4 = 0 \\ x_1^3 + x_2^3 - 2 \cdot x_3^3 - 7 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;3) \\ x_2 \in (0;3) \\ x_3 \in (0;3) \end{cases}$
14	$\begin{cases} x_1 \cdot x_2 + x_2 \cdot x_3 - 10 = 0 \\ x_1^2 - 4 \cdot x_2^2 + x_3^2 - 1 = 0 \\ x_1 - x_2 + x_3 - 3 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;5) \\ x_2 \in (0;5) \\ x_3 \in (0;5) \end{cases}$
15	$\begin{cases} x_1 \cdot x_2 + x_2 \cdot x_3 - 10 = 0 \\ x_1^2 - 4 \cdot x_2^2 + x_3^2 - 1 = 0 \\ \text{ch}(x_1) + \text{ch}(x_2) + \text{ch}(x_3) - 32,614 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;5) \\ x_2 \in (0;5) \\ x_3 \in (0;5) \end{cases}$
16	$\begin{cases} x_1^2 \cdot x_2 + x_2 \cdot x_3^2 - 20 = 0 \\ x_1^2 - x_2^2 + x_3^2 - 1 = 0 \\ \text{ch}(x_1) + \text{ch}(x_2) + \text{ch}(x_3) - 32,614 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;5) \\ x_2 \in (0;5) \\ x_3 \in (0;5) \end{cases}$
17	$\begin{cases} x_1^2 \cdot x_2 + x_2 \cdot x_3^2 - 20 = 0 \\ x_1^2 - x_2^2 + x_3^2 - 6 = 0 \\ \text{arcsh}(x_1) + \text{arcsh}(x_2) + \text{arcsh}(x_3) - 4,143 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;4) \\ x_2 \in (0;4) \\ x_3 \in (0;4) \end{cases}$
18	$\begin{cases} x_1^2 \cdot e^{x_2} + x_2 \cdot x_3^2 - 25,389 = 0 \\ x_1^2 - x_2^2 + x_3^2 - 6 = 0 \\ \text{arcsh}(x_1) + \text{arcsh}(x_2) + \text{arcsh}(x_3) - 4,143 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;4) \\ x_2 \in (0;4) \\ x_3 \in (0;4) \end{cases}$

## Продовження табл. 5.4

Вар.	Рівняння	Область
19	$\begin{cases} (x_1 + 2) \cdot (x_2 + 2) \cdot e^{-x_3} - 1,624 = 0 \\ x_1^2 + x_2^2 + x_3^2 - 9 = 0 \\ \operatorname{ch}^2(x_1) + \operatorname{ch}^2(x_2) - \operatorname{ch}(x_3) - 12,773 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;3) \\ x_2 \in (0;3) \\ x_3 \in (0;3) \end{cases}$
20	$\begin{cases} (x_1 - 1) \cdot (x_2 - 1) \cdot \cos(x_3 - 1) - 1,081 = 0 \\ (x_1 - x_2)^2 + 3 \cdot x_3 - 13 = 0 \\ \sin^2(x_1) + \sin^2(x_2) + \cos(2 \cdot x_3) - 0,1930 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;4) \\ x_2 \in (0;4) \\ x_3 \in (0;4) \end{cases}$
21	$\begin{cases} \ln[(x_1 + 1) \cdot (x_2 + 1)] \cdot \sin(2 \cdot x_3) - 2,091 = 0 \\ (x_1 - x_2)^2 + \operatorname{ch}(x_3) - 2,543 = 0 \\ \operatorname{arctg}(x_1 \cdot x_2 \cdot x_3) - 1,406 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;4) \\ x_2 \in (0;4) \\ x_3 \in (0;4) \end{cases}$
22	$\begin{cases} \cos(x_1 \cdot x_2) + \cos(x_2 \cdot x_3) - 0,544 = 0 \\ \cos(x_1 + x_3) - x_2 + 2,654 = 0 \\ \ln(x_1 - x_2 + x_3) + x_2 - 3,792 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;4) \\ x_2 \in (0;4) \\ x_3 \in (0;4) \end{cases}$
23	$\begin{cases} \sqrt{x_1 \cdot x_2 + x_2 \cdot x_3} - \ln(x_2) - 2,135 = 0 \\ x_1 + x_2 + x_3 - e^{x_2} - 1,389 = 0 \\ (x_1 - x_3)^2 + x_2 - 3 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;5) \\ x_2 \in (0;5) \\ x_3 \in (0;5) \end{cases}$
24	$\begin{cases} \sqrt[4]{x_1 \cdot x_2 + x_2 \cdot x_3} + \sqrt{x_2} - 2,510 = 0 \\ \frac{\ln(x_1 \cdot x_3)}{x_2} - 1,898 = 0 \\ (x_1 + x_3)^4 - x_2^3 - 730,16 = 0 \end{cases}$	$\begin{cases} x_1 \in (0;5) \\ x_2 \in (0;5) \\ x_3 \in (0;5) \end{cases}$
25	$\begin{cases} \sqrt{x_2 \cdot x_3} + \sqrt{x_1 \cdot x_3} - \operatorname{ch}(x_3) - 1,4890 = 0 \\ e^{(x_2 - x_1)} + x_3^3 - 2,221 = 0 \\ \sqrt{x_1 + x_2 + x_3} - \sqrt{x_1 + x_2 - x_3} - 0,4690 = 0 \end{cases}$	

## ЛЕКЦІЯ 6 НАБЛИЖЕННЯ ФУНКЦІЙ

### Мета лекції

1 Ознайомити слухачів з постановкою задач апроксимації даних, типами задач апроксимації, зв'язком з функціональними просторами.

2 Ознайомити слухачів з побудовою полінома Лагранжа, його властивостями, галузями застосування, одержання коефіцієнтів полінома за допомогою ЕОМ;

3 Ознайомити слухачів зі скінченно-різницевим підходом до побудови інтерполяційного поліному, з методом Ньютона побудови коефіцієнтів способами апроксимації .

### План лекції

[6.1 Постановка задачі](#)

[6.2. Задачі теорії наближень і функціональні простори](#)

[6.3 Поліноміальна інтерполяція](#)

[6.4 Інтерполяційний поліном Лагранжа](#)

[6.5 Розрахунки інтерполяційного полінома](#)

[6.6 Скінченні різниці різних порядків](#)

[6.7 Розділені різниці](#)

[6.8 Інтерполяційний поліном Ньютона](#)

[6.9 Обмеження методу інтерполяції](#)

[Контрольні питання](#)

[Завдання для самостійної роботи](#)

## Матеріали лекції

### 6.1 Постановка задачі

Задачі *наближення (апроксимації)* полягають у заміні функціональної залежності, заданої на множині  $X \subseteq \mathbb{R}$  у вигляді:

- ✓ таблиці,
- ✓ графіка,
- ✓ формули,
- ✓ функції, заданої в неявному виді,

що є більше простою і зручною для розрахунків наближених значень вхідної функції..

Якщо  $X$  складається з *дискретної* множини точок  $x_0, x_1, \dots, x_n$ , то наближення називається *точковим*.

Якщо множина  $X$  є *неперервною* (відрізок  $[a;b]$ ), то наближення називається *інтегральним*.

Нехай функція  $f(x)$  апроксимується на множині  $X$  функцією  $\varphi(x)$ , яка одночасно є:

- ✓ більше зручної для обчислень;
- ✓ близькою в деякому змісті до  $f(x)$ .

*Якість апроксимації*, тобто близькість двох функцій, можна оцінювати по-різному залежно від фізичного змісту задачі, що розв'язується. Відповідно до цього в теорії наближень розглядаються різні функціональні простори з уведеної в них метрикою, що дозволяє оцінити  $\rho(f, \varphi)$  - відстань між цими функціями.

Простір  $L$  називається *нормованим*, якщо кожному елементу  $f \in L$  поставлено у відповідність дійсне число  $\|f\|$ , яке називається *нормою* і задовольняє наступним аксіомам:

- а)  $\|f\| \geq 0$ ,  $\|f\| = 0$  тоді й тільки тоді, коли  $f = 0$ ;
- б)  $\|\alpha \cdot f\| = |\alpha| \cdot \|f\|$ , де  $\alpha \in \mathbb{R}$ ;
- в)  $\|f + \varphi\| \leq \|f\| + \|\varphi\|$

Нагадаємо зміст поняття "*повний простір*":

а) послідовність  $\{x^{(k)}\}$ , що побудована з елементів множини  $X$  називається *фундаментальною*, якщо для будь-якого малого  $\varepsilon > 0$  знайдеться такий номер  $N(\varepsilon)$ , що метрика  $\rho(x^{(k)}, x^{(m)}) < \varepsilon$  при  $k, m > N(\varepsilon)$  ( $x^{(k)}, x^{(m)} < \varepsilon$  при  $k, m > N(\varepsilon)$ );

б) простір  $X$  називається *повним*, якщо кожна фундаментальна послідовність цього простору *збігається* до деякої межі, яка є *елементом простору  $X$* .

*Повний*<sup>1)</sup> лінійний нормований простір називається *Банаховим* або *B-простором*.

Нормований простір стає *метричним*, якщо покласти метрику  $\rho(f, \varphi) = \|f - \varphi\|$  ( $f, \varphi \in L$ ).

Дійсний лінійний простір  $L$  називається *евклідовим*, якщо кожній парі його елементів  $f$  і  $\varphi$  поставлено у відповідність дійсне число  $(f, \varphi)$ , яке називається *скалярним добутком* і задовольняє наступним аксіомами:

- а)  $(f, f) \geq 0$ ,  $(f, f) = 0$  тоді й тільки тоді, коли  $f = 0$ ;

$$\text{б) } (f, \varphi) = (\varphi, f);$$

$$\text{в) } (\alpha f, \varphi) = \alpha(f, \varphi); \alpha \in \mathbf{R};$$

$$\text{г) } (f + \varphi, g) = (f, g) + (\varphi, g).$$

Повний лінійний евклідовий простір  $L$  називається *гільбертовим*. Гільбертів простір одночасно є банаховим.

Норму функції  $f(x) \in L$  можна визначити через скалярний добуток:

$$\|f\| = \sqrt{(f, f)}. \quad (6.1)$$

Скінченна система елементів  $\varphi_0, \varphi_1, \dots, \varphi_n$ , простору  $L$  називається *лінійно незалежною*, якщо їх лінійна комбінація є нульовим елементом простору  $L$ , тобто рівність:

$$c_0\varphi_0 + c_1\varphi_1 + \dots + c_n\varphi_n = 0 \quad (6.2)$$

виконується лише за умовою:

$$c_0 = c_1 = \dots = c_n = 0. \quad (6.3)$$

У протилежному випадку ця система називається *лінійно залежною*

Нескінченна система елементів  $\varphi_1, \varphi_2, \dots, \varphi_n \dots$  називається *лінійно незалежною*, якщо будь-яка її підсистема, що складається з кінцевого числа елементів, є *лінійно незалежною*.

Необхідна й достатня умова лінійної незалежності скінченної системи елементів  $\varphi_0, \varphi_1, \dots, \varphi_n$  - це *відмінність від нуля* визначника, побудованого зі скалярні добутоків (визначника Грама):

$$\begin{vmatrix} (\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) & \dots & (\varphi_0, \varphi_n) \\ (\varphi_1, \varphi_0) & (\varphi_1, \varphi_1) & \dots & (\varphi_1, \varphi_n) \\ \dots & \dots & \dots & \dots \\ (\varphi_n, \varphi_0) & (\varphi_n, \varphi_1) & \dots & (\varphi_n, \varphi_n) \end{vmatrix} \quad (6.4)$$

Якщо в просторі  $L$ :

а) існує система  $n$  лінійно незалежних елементів,

б) будь-які  $n + 1$  елементи цього простору є лінійно залежними,

то число  $n$  називають *вимірністю простору  $L$*  і позначають:  $n = \dim L$ .

Якщо в  $L$  існує злічене або незліченне нескінченне число лінійно незалежних елементів, то простір  $L$  називають *нескінченновимірним* і вважають  $\dim L = \infty$ .

Система лінійно незалежних елементів  $\varphi_1, \varphi_2, \dots, \varphi_n$   $n$ -вимірного простору  $L$  називається *базисом* цього простору, якщо для кожного елемента  $f$  простору  $L$  найдуться дійсні числа  $c_1, c_2, \dots, c_n$  такі, що буде справедливою рівність:

$$f = \sum_{i=1}^n c_i \varphi_i, \quad (6.5)$$



### ГРАМ ЙОРГЕН ПЕДЕРСЕН (1850-1916)

Відомий датський математик, професор університету Копенгагена. Працював в галузі «чистої» та прикладної математики. Найбільш відоме його досягнення – розробка процедури ортогоналізації матриць (процес Грама-Шміда), а також – функціональний визначник, що носить його ім'я

Послідовність  $\{\varphi_n(x)\}$  функцій нескінченновимірного банахова простору  $L$  називається *базисом* в  $L$ , якщо кожна функція  $f(x) \in L$  розкладається в ряд:

$$f(x) = \sum_{i=1}^{\infty} c_i \varphi_i(x), \quad (6.6)$$

який збігається за нормою простору  $L$ .

Ортогональність функцій в гільбертовому просторі: функції  $\varphi_1$  і  $\varphi_2$  називаються *ортогональними*, якщо їх скалярний добуток дорівнює нулю, тобто.  $(\varphi_1, \varphi_2) = 0$ .

Якщо  $\|\varphi\| = 1$ , то функцію  $\varphi(x)$  називають *нормованою*.

Скінченна або нескінченна система функцій  $\varphi_0, \varphi_1, \dots$  називається *ортонормованою*, якщо її елементи попарно ортогональні і нормовані:

$$(\varphi_i, \varphi_k) = \begin{cases} 0, & i \neq k, \\ 1, & i = k. \end{cases} \quad (6.7)$$

Ортонормована система функцій у гільбертовому просторі  $L$  називається *повною*, якщо її не можна доповнити елементами з  $L$  так, щоб нова система була ортонормованою.

Зміст: в  $L$  не існує ненульового елемента, ортогонального до всіх елементів даної системи.

Повна ортонормована система функцій утворює базис у просторі  $L$ .

Якщо розглядати лінійний простір дійсних функцій, заданих на множині  $X$ , то система його лінійно незалежних елементів називається *системою лінійно незалежних на  $X$  функцій*.

Нехай  $f(x)$  є елементом лінійного нормованого простору  $L$ . Виконаємо наступне:

- а) виберемо в  $L$  систему  $n + 1$  лінійно незалежних функцій  $\varphi_0(x), \varphi_1(x), \dots, \varphi_n(x)$ ;  
 б) утворимо лінійну комбінацію з постійними коефіцієнтами:

$$\varphi(x) = \sum_{i=0}^n c_i \varphi_i(x). \quad (6.8)$$

Функція (6.8) називається *узагальненим многочленом порядку  $n$* .

Часткові випадки узагальненого многочлена:

- ✓ при  $\varphi_i(x) = x^i$  ( $i = 0, 1, 2, \dots, n$ ) одержуємо *алгебраїчний многочлен ступеня  $n$* :

$$\varphi(x) = c_0 + c_1 x + \dots + c_n x^n = \sum_{i=0}^n c_i x^i. \quad (6.9)$$

- ✓ при  $\varphi_0(x) = 1, \varphi_1(x) = \cos(x), \varphi_2(x) = \sin(x), \dots, \varphi_{2n-1}(x) = \cos(x), \varphi_{2n}(x) = \sin(x)$  одержуємо *тригонометричний многочлен порядку  $n$* :

$$\varphi(x) = \frac{a_0}{2} + a_2 \cos(x) + b_1 \sin(x) + \dots + a_n \cos(nx) + b_n \sin(nx). \quad (6.10)$$

## 6.2 Задачі теорії наближень і функціональні простори

Будемо апроксимувати функцію  $f(x)$  узагальненим поліномом  $n$ -го порядку  $\varphi(x)$  виду (6.8). Параметри  $c_i$  ( $i = 1, 2, \dots, n$ ) цього поліному визначаються з умови мінімуму відстані  $(f, \varphi) = \|f - \varphi\|$  у метриці простору  $L$ . Є 3 найважливіших типи наближень, широко використовуваних на практиці:

- а) найкраще рівномірне наближення;  
 б) середнє квадратичне наближення на неперервній множині,  
 в) середнє квадратичне наближення на дискретній множині точок.

Найкраще рівномірне наближення:

- а) нехай функція  $f(x)$  належить простору  $C[a, b]$  дійсних функцій, неперервних на відрізку  $[a, b]$ . Уведемо поняття *рівномірної (чебышовської) норми*, як максимального значення модуля функції на цій множині:

$$\|f\|_{C[a; b]} = \max_{x \in [a; b]} |f(x)|; \quad (6.11)$$

- б) у відповідності до метрики (6.10) за міру близькості функцій  $f(x)$  і  $\varphi(x)$  приймають їх максимальне відхилення в усіх окремо взятих точках відрізка  $[a, b]$ :

$$\rho(f, \varphi) \equiv \|f - \varphi\|_{C[a;b]} = \max_{x \in [a;b]} |f(x) - \varphi(x)|. \quad (6.12)$$

Тоді:

- ✓ функція  $\varphi(x)$ , що забезпечує мінімум відстані  $\rho$ , називається найкращим рівномірним наближенням функції  $f(x)$  на відрізку  $[a;b]$ ;
- ✓ простір  $C[a,b]$  є повним нормованим (банаховим) простором.

Середньоквадратичне наближення на неперервній множині.

Розглянемо *гільбертів* простір  $L_2^p[a,b]$  дійсних функцій, квадрат яких інтегрується з вагою  $p(x)$  на відрізку  $[a;b]$ , тобто для  $f(x) \in L_2^p[a,b]$ , буде скінченним інтеграл у змісті Лебега:

$$\int_a^b p(x) f^2(x) dx. \quad (6.13)$$

Вважаємо, що на відрізку  $[a;b]$  вагова функція<sup>1)</sup> має такі властивості:

- ✓  $p(x) > 0$ ;
- ✓  $\int_a^b p(x) dx > 0$ ;

Скалярний добуток функцій у просторі  $L_2^p[a,b]$  визначається, як інтеграл:

$$(f, \varphi) = \int_a^b p(x) f(x) \varphi(x) dx; \quad (6.14)$$

Середня квадратична норма уводиться таким чином:

$$\|f(x)\|_{L_2^p[a;b]} = (f, f)^{1/2} = \left( \int_a^b p(x) f^2(x) dx \right)^{1/2}. \quad (6.15)$$

Відстань між функціями:

$$\rho(f, \varphi) \equiv \|f - \varphi\|_{L_2^p[a;b]} = \left( \int_a^b p(x) [f(x) - \varphi(x)]^2 dx \right)^{1/2} \quad (6.16)$$

називається *зваженим середнім квадратичним відхиленням* або *погрешністю наближення* в просторі  $L_2^p[a,b]$ ;

Збіжність функціональної послідовності в змісті метрики простору називається *збіжністю в середньому*.

Функція:

---

<sup>1)</sup> Вагову функцію доцільно вводити в тих випадках, коли  $f(x)$  апроксимується в різних точках з різним ступенем точності. Якщо  $p(x) = 1$  для всіх  $x \in [a;b]$ , то використають позначення  $L_2[a;b]$ .

$$\varphi(x) = \sum_{i=0}^n c_i \varphi_i(x), \quad (6.17)$$

що забезпечує мінімум квадрата відстані (6.16) називається *найкращим інтегральним зваженим середнім квадратичним наближенням* функції  $f(x)$

Середнє квадратичне наближення на дискретній множині точок:

а) простір функцій, заданих своїми значеннями на дискретній множині  $X$  точок  $x_0, x_1, \dots, x_n$ , що належать відрізку  $[a;b]$ , буде *гильбертовим* простором розмірності  $n + 1$ , якщо в ньому визначено скалярний добуток

$$(\varphi_k, \varphi_j) = \sum_{i=0}^m p_i \cdot \varphi_k(x_i) \cdot \varphi_j(x_i), \quad (p_i > 0); \quad (6.18)$$

б) в цьому просторі вводяться:

1) норма:

$$\|f\|_X = \left( \sum_{i=0}^m p_i \cdot [f(x_i)]^2 \right)^{1/2}, \quad (6.19)$$

2) зважене середнє квадратичне відхилення:

$$\rho(f, \varphi) = \|f - \varphi\|_X = \left( \sum_{i=0}^m p_i [f(x_i) - \varphi(x_i)]^2 \right)^{1/2}. \quad (6.20)$$

За допомогою позитивних ваг  $p_i$  можна враховувати неоднакову точність виміру значень функції;

в) функція:

$$\varphi(x) = \sum_{i=0}^n c_i \varphi_i(x), \quad (6.21)$$

що забезпечує мінімум квадрату відстані (6.20), називається *найкращим точковим зваженим середнім квадратичним наближенням* функції  $f(x)$ .

Базисні функції  $\varphi_i(x)$  ( $i = 1, \dots, n$ ) передбачаються *лінійно незалежними*:

- ✓ в інтегральному наближенні - на відрізку  $[a,b]$ ;
- ✓ у точковому наближенні - на множині точок  $[x_i]_0^m$ , причому, кількість точок - не менше розмірності простору ( $m \geq n$ )

Якщо:

а) значення функції  $f(x)$  задані в  $n + 1$  точках  $x_0, x_1, \dots, x_n$  відрізку  $[a;b]$ , названих вузловими точками;

б) існує функція  $\varphi(x)$ , яка:

- 1) досить проста для обчислень;
- 2) значення функції  $\varphi(x)$  збігаються з  $f(x)$  у вузлових точках  $x_0, \dots, x_n$ ;

3) в інших точках відрізка  $[a;b]$  функція  $\varphi(x)$  *приблизно* представляє  $f(x)$ , то таке наближення називається *інтерполяцією*.

### 6.3 Поліноміальна інтерполяція

Постановка задачі. Задані:

а) відрізок  $[x_0; x_n]$ ;

б) на цьому відрізку - точки  $x_0, x_1, \dots, x_n$  (вузли інтерполяції);

в) значення функції в цих точках:  $y_0 = f(x_0), y_1 = f(x_1), \dots, y_n = f(x_n)$ .

Необхідно побудувати функцію  $\varphi(x)$ , яка у вузлах інтерполяції збігається зі значеннями функції  $f(x)$ .

Таблиця значень функції  $f(x)$  може бути інтерпольована нескінченною множиною різних функцій, тому потрібно мати деякий *критерій* вибору.

Багато функцій, що інтерпольують, будуються як лінійні комбінації елементарних функцій. Приклади:

а) лінійні комбінації одночленів  $[x^k]_0^n$  приводять до алгебраїчних *поліномів*;

б) лінійні комбінації тригонометричних функцій  $\{\cos(kx); \sin(kx)\}_0^n$  приводять до *тригонометричних поліномів*.

У численних додатках апарата інтерполяції в основному доводиться розв'язувати дві задачі:

а) знайти *аналітичне вираження* для функції, заданим таблицею, графіком або у вигляді складної для додатків формули;

б) обчислити значення функції  $f(x)$  у точках, що не збігаються з вузлами інтерполяції, за допомогою порівняно нескладного алгоритму.

Інтерполяційні поліноми знаходять широке застосування в:

- чисельному диференціюванні;

- чисельному інтегруванні;

- чисельному розв'язанні диференціальних рівнянь.

Особливість інтерполяції: точки, у яких розраховується наближене значення, *обов'язково повинні перебувати усередині відрізка  $[x_0, x_n]$ , утвореного крайніми вузловими точками*.

Іноді потрібно обчислити наближене значення функції *за межами* відрізка  $[x_0, x_n]$ . Така задача називається *задачею екстраполяції*.

### 6.4 Интерполяционный полином Лагранжа

Виберемо за функцію  $\varphi(x)$ , що наближає, поліном  $P_n(x)$  ступеня  $n$ , значення якого збігаються зі значеннями функції  $f(x)$  у вузлах інтерполяції:

$$P_n(x_0)=y_0; P_n(x_1)=y_1; \dots; P_n(x_n)=y_n. \quad (6.22)$$

З точки хору геометрії (6.22) – це задача про побудову параболі  $n$ -го порядку виду  $y = P_n(x)$ , що *перетинається* із графіком функції  $y = f(x)$  в  $(n+1)$ -й наперед заданій точці (рис. 6.1).

Покажемо, що така задача має *єдине рішення*:



$$L_i(x) = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases} \quad (6.26)$$

Визначимо  $L_i(x)$  як *поліном ступеня  $n$* , що:

а) дорівнює нулю в точках  $x_0, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ ;

б) дорівнює одиниці в точці  $x_i$ .

Цій умові відповідає поліном:

$$L_i(x) = A(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n). \quad (6.27)$$

Для знаходження коефіцієнту  $A$  підставимо в (6.26)  $x=x_i$ :

$$L_i(x_i) = 1 = A(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n). \quad (6.28)$$

Виразимо з (6.28) значення  $A$ , одержимо остаточно:

$$L_i(x) = \frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}. \quad (6.29)$$

Поліном  $y_i \cdot L_i(x)$  має такі властивості:

а) приймає значення  $y_i$  в точці  $x_i$

б) дорівнює нулю в усіх точках  $x_j$  ( $j \neq i$ )

Позначимо:

$$\omega_n(x) = (x-x_0)(x-x_1)\dots(x-x_n). \quad (6.30)$$

Продиференціюємо (6.30) в точці  $x = x_i$ :

$$\omega'_n(x_i) = (x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n) \quad (6.31)$$

Після підстановки (6.30) і (6.31) в (6.29), одержимо:

$$L_i(x) = \frac{\omega_n(x)}{(x-x_i) \cdot \omega'_n(x_i)}. \quad (6.32)$$

Лангражеві коефіцієнти  $L_i(x)$  називають також *множниками впливу* відповідних вузлів інтерполяції.

Інтерполяційний поліном ступеня  $n$  виду:

$$P_n(x) = \sum_{i=0}^n y_i L_i(x) = \sum_{i=0}^n y_i \frac{\omega_n(x)}{(x-x_i) \omega'_n(x_i)}. \quad (6.33)$$

носить назву *полінома Лагранжа*

Кількість арифметичних операцій, що необхідні для його обчислення, пропорційно  $n^2$ .

Якщо вузли інтерполяції є *рівновіддаленими* друг від друга, тобто:

$$x_{i+1} - x_i = h = \text{const} \quad (i = 0, 1, 2, \dots, n), \quad (6.34)$$

то, якщо

а) увести змінну:  $t = (x - x_0)/h$ ,

б) виразити через неї  $x$ ,

можна записати лагранжеві коефіцієнти у виді:

$$L_i(t) = (-1)^{n-i} \cdot C_n^i \cdot \frac{t(t-1) \cdot (t-2) \dots (t-n)}{(t-i) \cdot n!}, \quad (6.35)$$

де  $C_n^i = \frac{n!}{i!(n-i)!}$  - біноміальні коефіцієнти.

Виходячи з (6.35), одержимо формулу для інтерполяційного полінома Лагранжа з *рівновіддаленими* вузлами:

$$P_n(t) = \sum_{i=0}^n y_i L_i(t) = \frac{t(t-1)(t-2) \dots (t-n)}{n!} \sum_{i=0}^n (-1)^{n-i} \frac{C_n^i}{t-i} y_i, \quad (6.36)$$

Особливості формули (6.36): коефіцієнти  $L_i(t)$  *не залежать*:

- ✓ від вибору кроку  $h$ ;
- ✓ від виду функції  $f(x)$ .

Погрішність інтерполяційного полінома Лагранжа:

Нехай функція  $f(x)$  має  $n+1$  *неперервну похідну* для всіх  $x \in [x_0; x_n]$ . Показано, що залишковий член інтерполяційного полінома Лагранжа має вид:

$$R_n(x) = f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \cdot \omega_n(x), \quad (6.37)$$

де  $\xi$  - внутрішня точка мінімального відрізка, що містить вузли інтерполяції  $x_0, \dots, x_n$  і точку  $x$ ;

Погрішність залежить від:

- ✓ вибору вузлів  $x_i$ ,
- ✓ вибору точки  $x$ ,
- ✓ властивостей функції  $f(x)$ ;



### ЛАГРАНЖ ЖОЗЕФ ЛУІ (1736-1813)

Видатний французький математик і механік, член Паризької Академії наук, іноземний член Петербурзької Академії наук. Головні праці – в галузі математичного аналізу, варіаційного числення і аналітичної механіки. Запропонований ним інтерполяційний поліном знаходить широке застосування у багатьох чисельних методах

Принципи впливу місце розташування точки  $x$ :

- ✓ погрішність інтерполяції буде *меншою*, якщо точка  $x$  розташована ближче до *середини відрізка*  $[x_0; x_n]$ ;
- ✓ погрішність інтерполяції буде *більшою*, якщо точка  $x$  розташована *ближче* до будь-якого *кінця відрізка*  $[x_0; x_n]$ ;
- ✓ при *екстраполяції* погрішність завжди є *більшою*, у порівнянні з інтерполяцією, а у багатьох випадках вона непередбачена.

### 6.5 Розрахунки інтерполяційного полінома

Вибір методу чисельної реалізації інтерполяції залежить від постановки конкретної задачі. Найвні такі типові випадки:

а) якщо необхідно вирішити *разову задачу* - побудувати для набору даних інтерполяційний поліном, то немає необхідності прибігати до програмування: досить скористатися пакетами прикладних програм, у які "зашиті" або розрахунок інтерполяційного полінома, або розв'язання системи лінійних рівнянь.

б) якщо процедура інтерполяції входить *до складу програмного комплексу*, що розробляється для розв'язання деяких масових задач - у цьому випадку доводиться укладати комп'ютерну програму (у вигляді підпрограми або підпрограми-функції), що є частиною цього комплексу.

6.5.2 На практиці виникають дві задачі, що пов'язані з інтерполяцією:

- а) розрахунок *коефіцієнтів* інтерполяційного поліному,
  - б) розрахунок *значень поліному* у точках, що відрізняються від вузлових.
- Для розрахунку значень полінома:

$$P_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (6.38)$$

в комп'ютерних програмах використовують *метод Горнера*<sup>1)</sup>. Для цього:

- а) поліном (6.35) представляють у формі:

$$P(x) = (\dots(((a_nx + a_{n-1})x + a_{n-2})x + a_{n-3})x + \dots + a_0). \quad (6.39)$$

<sup>1)</sup> Горнер Уїльям Джордж (1786-1837) – англійський математик. Основні праці – в галузі теорії алгебраїчних рівнянь, розробив схему Горнера ділення многочлену на двочлен.

б) з (6.39) для значення  $x = \xi$  послідовно розраховують числа:

$$\begin{cases} b_n = a_n; \\ b_{n-1} = a_{n-1} + b_n \xi; \\ b_{n-2} = a_{n-2} + b_{n-1} \xi; \\ \dots\dots\dots \\ b_0 = a_0 + b_1 \xi \end{cases} \quad (6.40)$$

При цьому  $b_0 = P_n(\xi)$ .

### Приклад 6.1

При  $x = 1,2$  розрахувати значення полінома

$$P_4(x) = 3 - x + 2,5x^2 - 3,2x^3 + 0,1x^4 \quad (6.41)$$

Розв'язання

Згідно (6.40), послідовно знаходимо:

$$b_4 = 0,1;$$

$$b_3 = -3,2 + 0,1 \cdot 1,2 = -3,08;$$

$$b_2 = 2,5 - 3,08 \cdot 1,2 = -1,196$$

$$b_1 = -1 - 1,196 \cdot 1,2 = -2,4352$$

$$b_0 = 3 - 2,4352 \cdot 1,2 = 0,07776.$$

$$\text{Звідси } P_4(x=1,2) = b_0 = 0,07776$$

### Приклад 6.2

Побудувати інтерполяційний поліном для функції:

$$y = 1 - e^{-x} \quad (6.42)$$

Розрахувати значення  $y$  в точках  $0,1,2,3,4,5$ . Побудувати графік функції та інтерполяційного полінома в точках  $x_0 \div x_3$  і визначити його екстраполяційні властивості

Розв'язання

а) виконаємо інтерполяцію поліномом 3-й ступеня в точках  $x_0=0, x_1=1, x_2=2, x_3=3$ . Для знаходження коефіцієнтів полінома Лагранжа в середовищі *Ооо Calc* використовуємо не безпосередні розрахунки коефіцієнтів, а розв'язання системи рівнянь (6.24), що попереджають виводу коефіцієнтів поліному Лагранжа. Для цього:

1) будуємо таблицю (комірки E4:I7) куди поміщаємо коефіцієнти системи рівнянь (6.24) і значення  $y$  (рис. 6.2)

2) у масиві (L3:L6) проводиться розв'язання системи. Формула масиву:  $=MMULT(MINVERSE(E4:H7);I4:I7)$

б) будуємо таблицю значень функції й інтерполяційного полінома в діапазоні значень  $x$  від 0 до 5 (тобто виходимо за межі відрізка інтерполяції)

в) по таблиці будуємо графік для вихідної функції й полінома (рис. 6.2)

З графіка на рис. 6.2 виходить:

а) всередині відрізка інтерполяції значення полінома й вихідної функції близькі, тобто, поліном Лагранжа добре апроксимує функцію не тільки у вузлових точках;

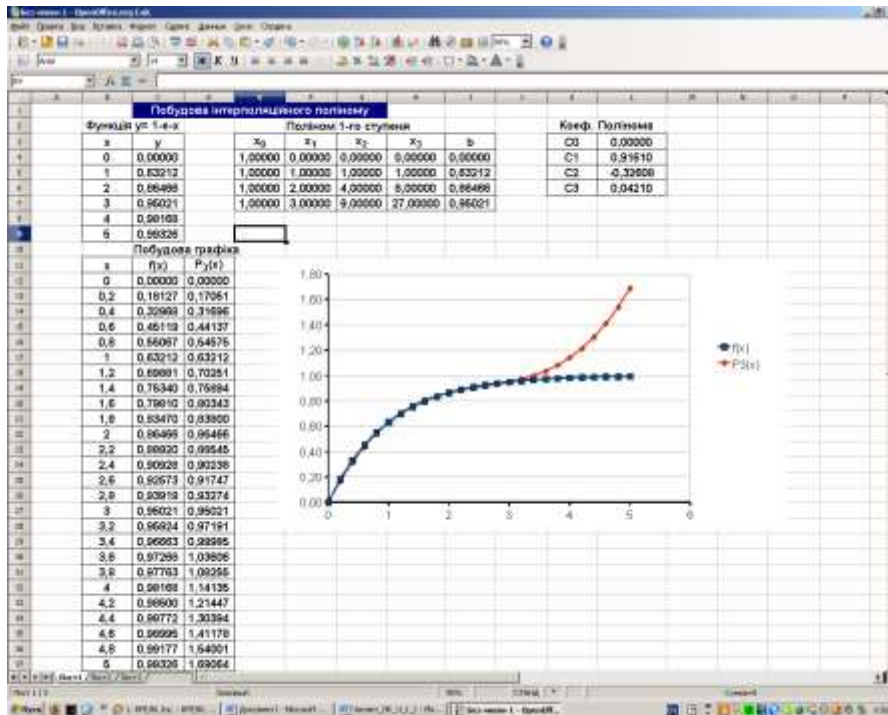


Рисунок 6.2 – Інтерполяція функції (6.42) поліномом 3-го ступеня. Графік функції (6.42) (1), графік інтерполяційного полінома (2)

б) у вузлових точках значення функції й полінома збігаються в межах погрешності обчислень, як це й виходить з теорії;

в) за межами відрізка інтерполяції спостерігаються значні розбіжності в значеннях полінома й функції.

Розбіжності підсилюються в міру *видалення* від граничної вузлової точки. Отже, інтерполяційний поліном має *погані екстраполяційні властивості*.

## 6.6 Скінченні різниці різних порядків

Нехай  $y = f(x)$  - задана функція. Позначимо через  $x = h$  фіксовану величину збільшення аргументу (*крок*).

Виразення:

Вираз:

$$\Delta y = \Delta f(x) = f(x+\Delta x) - f(x) \quad (6.43)$$

називається *першою скінченною різницею* функції  $y$ .

Аналогічно послідовно визначаються скінченні різниці вищих порядків:

$$\Delta^n y = \Delta(\Delta^{n-1} y) \quad (n = 2, 3, \dots) \quad (6.44)$$

З (6.44) формула для обчислення скінченної різниці другого порядку має вид:

$$\begin{aligned}\Delta^2 y &= \Delta[f(x+\Delta x) - f(x)] = [f(x+2\Delta x) - f(x+\Delta x)] - \\ &\quad [f(x+\Delta x) - f(x)] = \\ &= f(x+2\Delta x) - 2f(x+\Delta x) + f(x)\end{aligned}\quad (6.45)$$

### Приклад 6.3

Побудувати скінченні різниці для функції:  $y = x^3$ , вважаючи крок  $\Delta x = 1$ .

Розв'язання

а) різниця першого порядку:

$$\Delta f(x) = (x+1)^3 - x^3 = 3x^2 + 3x + 1;$$

б) різниця другого порядку:

$$\Delta^2 f(x) = \Delta f(x+1) - \Delta f(x) = [3(x+1)^2 + 3(x+1) + 1] - [3x^2 + 3x + 1] = 6x + 6;$$

в) різниця третього порядку:

$$\Delta^3 f(x) = \Delta^2 f(x+1) - \Delta^2 f(x) = [6(x+1) + 6] - [6x + 6] = 6;$$

г) різниці четвертого й більше високого порядків:

$$\Delta^k f(x) = 0 \quad (k > 3)$$

Є вірним твердження: якщо функція - *поліном ступеня n* виду (6.24), то скінченні різниці n-го порядку *постійні*:

$$\Delta^n P^n(x) = n! a_n h^n = \text{const}, \quad (6.46)$$

де  $h = \Delta x$

Символ  $\Delta$  можна розглядати, як *оператор*, що ставить у відповідність функції  $y = f(x)$  функцію  $\Delta y = f(x+\Delta x) - f(x)$  (величина  $\Delta x$  є постійною).

Властивості оператора  $\Delta$ :

а)  $\Delta(u+v) = \Delta u + \Delta v$ ;

б)  $\Delta(C \cdot u) = C \cdot \Delta u$  ( $C$  – константа);

в)  $\Delta^m(\Delta^n u) = \Delta^{m+n} u$  ( $m, n$  – цілі невід'ємні числа. За визначенням:  $\Delta^0 u = u$ ).

Вираження функції через скінченні різниці різних порядків:

а) з формули (6.43):.

$$f(x+\Delta x) = f(x) + \Delta f(x); \quad (6.47)$$

б) розглянувши  $\Delta$ , як символічний множник, одержимо

$$f(x+\Delta x) = (1+\Delta) \cdot f(x); \quad (6.48)$$

в) послідовно застосуємо співвідношення (6.48)  $n$  раз:

$$f(x+n \cdot \Delta x) = (1 + \Delta)^n f(x); \quad (6.49)$$

г) скористаємося розкладанням правої частини (6.49), як бінома Ньютона:

$$f(x + n\Delta x) = \sum_{m=0}^n C_n^m \cdot \Delta^m f(x), \quad (6.50)$$

де  $C_n^m$  - біноміальні коефіцієнти (кількість сполучень):

$$C_n^m = \frac{n!}{m!(n-m)!} = \frac{\prod_{k=n-(m-1)}^n k}{m!} = \frac{n(n-1)\dots[n-(m-1)]}{m!}. \quad (6.51)$$

За допомогою формули (6.50) можна виразити через скінченні різниці значення функції, що послідовно зростають з кроком  $\Delta x$ .

Вираз скінченної різниці  $n$ -го порядку через значення функції в  $n$  точках:

а) скористаємося тотожністю:

$$\Delta = (1+\Delta) - 1; \quad (6.52)$$

б) запишемо різницю  $n$ -го порядку й застосуємо до неї біном Ньютона:

$$\Delta^n f(x) = [(1+\Delta)^n f(x) - C_n^1 (1+\Delta)^{n-1} f(x) + C_n^2 (1+\Delta)^{n-2} - \dots + (-1)^n f(x)]; \quad (6.53)$$

в) в силу (6.49):

$$\Delta^n f(x) = f(x + n\Delta x) - C_n^1 f[x + (n-1)\Delta x] + C_n^2 f[x + (n-2)\Delta x] - \dots + (-1)^n f(x) \quad (6.54)$$

Формула (6.54) дає вираз скінченної різниці  $n$ -го порядку функції  $f(x)$  через послідовні значення цієї функції.

Нехай функція  $f(x)$  має на відрізку  $(x; x+\Delta x)$  неперервну похідну  $f^{(n)}(x)$ . Тоді буде вірною формула:

$$\Delta^k f(x) = (\Delta x)^k f^{(k)}(x + \theta \cdot k \cdot \Delta x) \quad (0 < \theta < 1). \quad (6.55)$$

Ця формула має значення в дослідженнях, пов'язаних із практичним використанням скінченних різниць в обчислювальних процедурах.

Часто доводиться розглядати функції  $y = f(x)$ , які задані табличними значеннями  $y_i = f(x_i)$  для системи *рівновіддалених* точок  $x_i$  ( $i = 0, 1, 2, \dots$ ), де:

$$\Delta x_i = x_{i+1} - x_i = h = \text{const}. \quad (6.56)$$

У цьому випадку кінцеві різниці послідовності  $y_i$  визначаються співвідношеннями:

$$\begin{cases} \Delta y_i = y_{i+1} - y_i; \\ \Delta^2 y_i = \Delta(\Delta y_i) = \Delta y_{i+1} - \Delta y_i; \\ \dots\dots\dots \\ \Delta^n y_i = \Delta(\Delta^{n-1} y_i) = \Delta^{n-1} y_{i+1} - \Delta^{n-1} y_i. \end{cases} \quad (6.57)$$

Методами, описаними вище, можна одержати формулу для скінченних приростів високого порядку функцій, заданих у виді таблиць, яка аналогічна (6.51):

$$\Delta^n y_i = y_{n+i} - C_n^1 y_{n+i-1} + C_n^2 y_{n+i-2} - \dots + (-1)^n y_i. \quad (6.58)$$

З формули (6.58) виходить, що для обчислення  $n$ -ї різниці  $\Delta^n y_i$  треба знати  $n+1$  член  $y_i, y_{i+1}, \dots, y_{i+n}$  даної послідовності.

Приклади:

$$\Delta^2 y_i = y_{i+2} - 2y_{i+1} + y_i;$$

$$\Delta^3 y_i = y_{i+3} - 3y_{i+2} + 3y_{i+1} - y_i$$

Скінчені різниці різних порядків зручно розташовувати у формі таблиць двох видів:

- ✓ горизонтальної таблиці різниць (табл. 6.1);
- ✓ діагональної таблиці різниць (табл. 6.2)

Таблиця 6.1 – Горизонтальна таблиця різниць

x	y	$\Delta y$	$\Delta^2 y$	$\Delta^3 y$	...
$x_0$	$y_0$	$\Delta y_0$	$\Delta^2 y_0$	$\Delta^3 y_0$	
$x_1$	$y_1$	$\Delta y_1$	$\Delta^2 y_1$	$\Delta^3 y_1$	
$x_2$	$y_2$	$\Delta y_2$	$\Delta^2 y_2$	$\Delta^3 y_2$	
.....	...	...	...	...	...

Таблиця 6.2 – Діагональна таблиця різниць

x	y	$\Delta y$	$\Delta^2 y$	$\Delta^3 y$
$x_0$	$y_0$			
$x_1$	$y_1$	$\Delta y_0$		
$x_2$	$y_2$	$\Delta y_1$	$\Delta^2 y_0$	
$x_3$	$y_3$	$\Delta y_2$	$\Delta^2 y_1$	$\Delta^3 y_0$

#### Приклад 6.4

Скласти горизонтальну таблицю різниць функції:

$$y = 2x^3 - 2x^2 + 3x - 1, \quad (6.59)$$

виходячи з початкового значення  $x_0 = 0$  з кроком  $h=1$  до  $x=8$

Розв'язання

а) вважаємо  $x_0 = 0, x_1 = 1, x_2=2, \dots, x_8=8$ , розраховуємо значення  $y$  і заносимо в таблицю (табл. 6.3);

б) функція - поліном 3-й ступеня. Тому треті різниці є постійними. Значення різниць ідуть "драбинкою".

Таблиця 6.3 - Складання горизонтальної таблиці різниць для функції (6.59)

x	y	$\Delta y$	$\Delta^2 y$	$\Delta^3 y$
0	-1	3	8	12
1	2	11	20	12
2	13	31	32	12
3	44	63	44	12
4	107	107	56	12
5	214	163	68	12
6	377	231	80	
7	608	311		
8	919			

### 6.7 Розділені різниці

Нехай функція  $f(x)$  задана у виді таблиці в не обов'язково рівновіддалених точках  $x_0, x_1, \dots, x_n$  ( $x_{i+1} - x_i = h_i \neq \text{const}$ ). Значення функції в цих точках, відповідно:  $y_0 = f(x_0), y_1 = f(x_1), \dots, y_n = f(x_n)$ .

Розділені різниці першого порядку – це величини *скінчених різниць* функцій, які *віднесені* на кожному інтервалі до *одиночної довжини* кроку  $h_i = x_{i+1} - x_i$  ( $i = 0, 1, \dots, n-1$ ):

$$\left\{ \begin{array}{l} f(x_0; x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}; \\ f(x_1; x_2) = \frac{f(x_2) - f(x_1)}{x_2 - x_1}; \\ \dots \dots \dots \\ f(x_{n-1}; x_n) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}. \end{array} \right. \quad (6.60)$$

Розділені різниці першого порядку мають зміст середніх швидкостей зміни функції на кожному інтервалі.

Розділені різниці другого порядку визначаються, як розділені різниці від перших різниць:

$$\left\{ \begin{array}{l} f(x_0; x_1; x_2) = \frac{f(x_1; x_2) - f(x_0; x_1)}{x_2 - x_0}; \\ f(x_1; x_2; x_3) = \frac{f(x_2; x_3) - f(x_1; x_2)}{x_3 - x_1}; \\ \dots \dots \dots \\ f(x_{n-2}; x_{n-1}; x_n) = \frac{f(x_n; x_{n-1}) - f(x_{n-1}; x_{n-2})}{x_n - x_{n-2}}. \end{array} \right. \quad (6.61)$$

У загальному випадку розділені різниці  $k$ -го порядку визначаються через розділені різниці  $(k-1)$ -го порядку по рекурентній формулі:

$$f(x_i; x_{i+1}; \dots; x_{i+k}) = \frac{f(x_{i+1}; x_{i+2}; \dots; x_{i+k}) - f(x_i; x_{i+1}; \dots; x_{i+k-1})}{x_{i+k} - x_i}. \quad (6.62)$$

Властивості розділених різниць:

- ✓ розділені різниці  $k$ -го порядку є *симетричними функціями* своїх аргументів і виражаються через *вузлові значення* функції:

$$f(x_i, x_{i+1}, \dots, x_{i+k}) = \sum_{j=i}^{i+k} \frac{f(x_j)}{\omega'_k(x_j)}, \quad (6.63)$$

де  $\omega'_n(x_i) = (x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)$  (див. (6.31));

- ✓ якщо функція  $f(x)$  на відрізку  $[x_0; x_n]$  є  $n$  разів неперервне диференціюємою, то виконується співвідношення:

$$f(x_0; x_1; \dots; x_n) = \frac{1}{n!} f^{(n)}(\xi), \quad (6.64)$$

де  $x_0 < \xi < x_n$ ;

- ✓ розділені різниці  $n$ -го порядку від алгебраїчного многочлена  $n$ -го ступеня є *постійними*, а різниці більше високих порядків *дорівнюють нулю*;
- ✓ кінцеві різниці для рівновіддалених проміжків можна розглядати, як окремий випадок розділених різниць.

## 6.8 Інтерполяційний поліном Ньютона

Для багатьох практичних задач інтерполяції доцільніше виразити інтерполяційний поліном через розділені різниці, чим використати форму Лагранжа. Інтерполяційний поліном може бути записаний через розділені різниці у вигляді:

$$P_n(x) = f(x_0) + (x-x_0)f(x_0; x_1) + (x-x_0)(x-x_1)f(x_0; x_1; x_2) + \dots + (x-x_0)(x-x_1) \dots (x-x_{n-1})f(x_0; x_1; \dots; x_n) \quad (6.65)$$

Поліном, записаний у виді (6.65), називається *інтерполяційним поліномом Ньютона для нерівновіддалених вузлів*.

Поліноми  $n$ -го ступеня, що визначаються формулами (6.65) і (6.36) є тотожно рівними тому що приймають однакові значення в  $n+1$ -й вузловій точці. Тобто, вираз (6.65) суть інша форма запису полінома Лагранжа.

Форма запису (6.65) зручна тим, що при додаванні до вузлів  $x_0, x_1, \dots, x_n$  нового вузла  $x_{n+1}$ :

- а) всі обчислені раніше доданки залишаються без змін,

б) в (6.65) додається тільки новий доданок.

Якщо в (6.62) покласти  $n=1$ , то одержимо формулу для лінійної інтерполяції, яку часто використовують на практиці:

$$P_1(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0} \cdot (x - x_0). \quad (6.66)$$

Найчастіше формула (6.66) використовується для розрахунку значень між вузлами функцій, що задані таблицями. При цьому допускають, що при невеликій зміні функції у вузлових точках закон зміни функції між вузлами можна вважати приблизно лінійним.

### Приклад 6.5

Щільність 10%-ного розчину спирту становить 0,9819 г/см<sup>3</sup>, а 20 %-ного - 0,9686. Розрахувати щільність 13 %-ного розчину спирту.

Розв'язання:

Підставляємо значення в формулу (6.66):  $x_0 = 10$ ,  $x_1 = 20$ ,  $x=13$ ,  $y_0 = 0,9818$ ;  $y_1 = 0,9686$ :

$$y = 0,9818 + \frac{(0,9686 - 0,9818)}{(20 - 10)} \cdot (13 - 10) = 0,9774$$

Для рівновіддалених вузлів вираження для полінома Ньютона спрощується. У цьому випадку:  $x_i = x_0 + i h$  ( $i = 0, 1, 2, \dots, n$ ), що приводить до виконання для розділених різниць співвідношення:

$$f(x_0; x_1; \dots; x_n) = \frac{\Delta^n y_0}{n! h^n}. \quad (6.67)$$

Для інтерполяції функції в точці  $x$ , близької до  $x_0$ :

а) уведемо нову змінну:

$$t = \frac{x - x_0}{h}. \quad (6.68)$$

Тоді вузлам інтерполяції відповідають значення:

$$t_0 = 0; t_1 = 1; \dots; t_n = n; \quad (6.69)$$

б) множники в (6.65) перетворюються в такий спосіб:

$$\frac{x - x_0}{h} = t; \quad \frac{x - x_1}{h} = t - 1, \dots, \frac{x - x_{n-1}}{h} = t - n + 1. \quad (6.70)$$

в) Підставивши (6.70) в (6.65), одержимо інтерполяційну формулу Ньютона для інтерполяції вперед:

$$P_n(t) = y_0 + \frac{t}{1!} \cdot \Delta y_0 + \frac{t(t-1)}{2!} \cdot \Delta^2 y_0 + \dots + \frac{t(t-1)\dots(t-n+1)}{n!} \Delta^n y_0. \quad (6.71)$$

Для інтерполяції функції, у точці  $x$ , близької до кінцевого вузла  $x_n$ , зручніше ввести змінну:

$$t = \frac{x - x_n}{h}. \quad (6.72)$$

У цьому випадку одержимо формулу інтерполяційного полінома Ньютона для інтерполяції назад:

$$P_n(t) = y_n + \frac{t}{1!} \cdot \Delta y_{n-1} + \frac{t(t+1)}{2!} \cdot \Delta^2 y_{n-2} + \dots + \frac{t(t+1)\dots(t+n-1)}{n!} \Delta^n y_0. \quad (6.70)$$

У цій формулі використовуються скінчені різниці, що розташовані на нижній діагоналі різницевої таблиці.

## 6.9 Обмеження методу інтерполяції.

Інтерполяція поліномами широко використовується не тільки, як самостійна задача, але також як основа для розробки методів чисельного диференціювання й інтегрування, рішення диференціальних рівнянь і т.д.

Методу інтерполяції притаманні наступні обмеження:

- ✓ метод забезпечує аналітичне подання або обчислення функції  $f(x)$ , що задана у виді таблиці, лише на *невеликому* інтервалі з *невеликим* числом вузлів інтерполяції;
- ✓ метод *непридатний* для аналітичного подання табличних функцій з більшою кількістю вузлів через виникаючу нестійкість і зростання погрішності. Навіть незначні помилки в значеннях функції приводять до того, що різниці високих порядків носять випадковий характер;
- ✓ метод *непридатний* для обробки *результатів спостережень*, що містять випадкові погрішності.

На практиці інтерполяція *обмежується поліномами 3-4 ступеня*, поліноми більше високого ступеня використовуються рідко.

Для розв'язання задач апроксимації на великій кількості вузлів, а також, для задач, де обробляються результати спостережень, використовують інші методи, засновані на *середнє квадратичній апроксимації*, а також, на апроксимації *сплайнами*.

## Контрольні питання

- 1 Як називаються задачі, що полягають у заміні функціональної залежності, яка задана у виді таблиць, графіків, складних формул на більш прості функції?
- 2 Дайте визначення поняття «точкове наближення»
- 3 Яке наближення називається інтегральним?
- 4 Як можна оцінити якість апроксимації?
- 5 Зміст поняття нормованого простору?
- 6 Зміст поняття «банахів простір»
- 7 За якою ознакою лінійний простір можна віднесити до евклідового простору?
- 8 За якою ознакою лінійний евклідовий простір відносять до гільбертового простору?
- 9 Зміст поняття лінійної незалежності скінченної системи функцій
- 10 Як відбувається перехід від лінійно незалежної скінченної системи до лінійно незалежної нескінченної системи функцій?
- 11 У чому полягає необхідна і достатня умова лінійної незалежності скінченної системи елементів?
- 12 У лінійному просторі будь-які 3 елементи є лінійно незалежними. Чи можна стверджувати, що вимірність цього простору дорівнює 3?
- 13 Зміст поняття базису функціонального простору
- 14 Як відбувається перехід від базису скінченно вимірного простору до базису нескінченновимірного банахова простору?
- 15 Зміст поняття «ортонормовані функції»
- 16 Яка система функцій є повною у гільбертовому просторі?
- 17 У чому полягає зміст повноти функцій к гільбертовому просторі?
- 18 Написати формулу узагальненого многочлену порядку  $n$ .
- 19 Як з узагальненого многочлену можна одержати алгебраїчний та тригонометричний многочлени?
- 20 Які типи наближень використовуються на практиці?
- 21 Що приймають за міру близькості функцій при рівномірному наближенні?
- 22 Як визначається відстань між функціями у інтегральному середньому квадратичному наближенні?
- 23 Як визначається відстань між функціями у дискретному середньому квадратичному наближенні?
- 24 Зміст зважених величин у середньому квадратичних наближеннях. У якому випадку їх використовують?
- 25 Зміст поняття інтерполяція
- 26 У чому полягає постановка задачі поліноміальної інтерполяції?
- 27 Зміст поняття «вузол інтерполяції»
- 28 Які типи задач розв'язують методом інтерполяції?
- 29 У чому полягають особливості задач інтерполяції?
- 31 Зміст поняття «екстраполяція»
- 32 Як ставиться задача побудови інтерполяційного поліному Лагранжа?
- 33 Скільки інтерполяційних поліномів можна побудувати на заданій сукупності точок?

34 Функцію, що задана у виді таблиці, інтерполюють неповним поліномом  $y = a_1x + a_3x^3$ . Записати для цього випадку систему рівнянь для знаходження коефіцієнтів і визначник Вандермонта.

35 Як можна на практиці будувати поліном Лагранжа?

36 У якому випадку Лагранжеві коефіцієнти не залежать від функції  $f(x)$ ?

37 Де погрішність інтерполяції буде меншою: на кінцях або у середині інтервалу?

38 Як співвідносяться між собою погрішності інтерполяції та екстраполяції?

39 Які є типові задачі чисельної реалізації інтерполяції?

40 У чому сутність алгоритму розрахунків значень поліному методом Горнера?

41 Зміст поняття «скінченна різниця першого порядку»

42 Як можна обчислити скінченну різницю другого порядку?

43 Побудова на практиці скінчених різниць від многочленів

44 Властивості скінчених різниць від многочленів

45 Властивості різницевого оператора

46 Як виразити функцію через скінчені різниці?

47 Як виразити скінченні різниці через послідовні значення функції?

49 Як побудувати таблиці різниць для рівновіддалених значень функції?

50 Зміст поняття «розділена різниця першого порядку».

51 Як побудувати розділені різниці другого порядку?

52 Якими властивостями володіють розділені різниці?

53 Зміст поняття «інтерполяційний поліном Ньютона»

54 У чому зручність запису інтерполяційного поліному через різниці?

55 Записати формулу для лінійної інтерполяції

56 Як за допомогою лінійної інтерполяції знайти значення табличної функції у проміжку між вузловими точками?

57 У чому полягають особливості побудови інтерполяційного полінома Ньютона для рівновіддалених точок?

58 У чому особливості інтерполяції уперед і назад?

59 Які обмеження притаманні методу інтерполяції?

### Завдання для самостійної роботи

У табл. 6.4 наведені функції та інтервали, на яких вони визначені. Необхідно в середовищі Ooo Calc:

а) розділити інтервал на 10 рівних частин, розрахувати значення функцій у кожній точці;

б) користуючись двома кінцевими точками і 1-3 точками в середині інтервалу, побудувати інтерполяційний поліном 2-4 порядків; користуючись методом Лагранжа и методом Ньютона з інтерполяцією уперед і назад;

в) розрахувати значення інтерполяційного полінома у кожній точці, знайти модуль максимального відхилення між значеннями функції та поліному,

г) зробити кусково-лінійною інтерполяцію для сусідніх точок з парними номерами: (0;2), (2;4), ... (8;10). Розрахувати значення поліному у непарних точках, знайти модуль максимального відхилення між значеннями функції та поліному;

д) на основі розрахунків визначити, який із способів інтерполяції є більш точним.

Таблиця 6.4 – Варіанти завдань

№	Функція	Інтервал
1	$y = 2 \cdot \ln(e^x + 2)$	[0;3]
2	$y = \frac{\sin(2 \cdot x + 1)}{1 + \cos(3 \cdot x)}$	[-0,1;0,9]
3	$y = e^x \cdot \cos(x)$	[0;1]
4	$y = e^{-x} \cdot \cos(x)$	[0;1]
5	$y = e^x \cdot \sin(x)$	[0;1]
6	$y = e^{-x} \cdot \sin(x)$	[0;1]
7	$y = \operatorname{arctg}(x \cdot e^x)$	[0;1]
8	$y = \frac{1}{\operatorname{arctg}(e^x + 1)}$	[0;2]
9	$y = \operatorname{ch}(x) \cdot \cos(x)$	[0;2]

## Продовження табл. 6.1

№	Функція	Інтервал
10	$y = \operatorname{ch}(x) \cdot \sin(x)$	[0;2]
11	$y = \operatorname{sh}(x) \cdot \cos(x)$	[0;2]
12	$y = \operatorname{sh}(x) \cdot \sin(x)$	[0;2]
13	$y = \operatorname{ash}(x) \cdot \sin(x)$	[0;2]
14	$y = \operatorname{ach}(x+1) \cdot \sin(x)$	[0;2]
15	$y = \arcsin(x) \cdot e^{-x}$	[-1;1]
16	$y = \arccos(x) \cdot e^{-x}$	[-1;1]
17	$y = \operatorname{ch}(x) \cdot \operatorname{sh}(x)$	[-1;1]
18	$y = \operatorname{ch}(x) + \operatorname{sh}(x)$	[-1;1]
19	$y = \operatorname{ch}(x) - \operatorname{th}(x)$	[-1;1]
20	$y = \operatorname{ch}(x) \cdot \operatorname{th}^2(x)$	[-1;1]
21	$y = \operatorname{ch}(x) + \operatorname{th}^2(x)$	[-1;1]
22	$y = \operatorname{ch}^2(x) + \operatorname{th}^2(x+1)$	[-1;1]
23	$y = 2 \cdot e^{-\frac{1}{x}}$	[1;11]
24	$y = 2 \cdot e^{-\frac{1}{x}} \cdot \cos(0,1 \cdot x)$	[1;11]
25	$y = 2 \cdot e^{-\frac{1}{x}} \cdot \sin(0,2 \cdot x)$	[1;11]
26	$y = \frac{2 \cdot e^{-\frac{1}{x}}}{\sin(0,2 \cdot x)}$	[1;11]
27	$y = \frac{\operatorname{ch}(x)}{1 + \sin(x)}$	[0;2]
28	$y = \frac{\operatorname{ch}(x) \cdot \cos(x)}{1 + \sin(x)}$	[0;2]
29	$y = \frac{\operatorname{ch}(x) \cdot \sin(x)}{1 + \cos^2(x)}$	[0;2]
30	$y = \frac{\operatorname{ch}(x) \cdot \cos(x)}{1 + \cos^2(x)}$	[0;2]

## ЛЕКЦІЯ 7. АПРОКСИМАЦІЯ МЕТОДОМ НАЙМЕНШИХ КВАДРАТІВ. ЕЛЕМЕНТИ СПЛАЙН-АПРОКСИМАЦІЇ.

### Мета лекції:

1 Ознайомити слухачів з особливостями апроксимації функцій методом найменших квадратів.

2 Надати уявлення про практичні аспекти апроксимації методом найменших квадратів за допомогою комп'ютерних програм

3 Ознайомити слухачів з деякими аспектами кусково-поліноміальної апроксимації.

4 Ознайомити слухачів з основами сплайн-інтерполяції зокрема за допомогою кубічних сплайнів, надати уявлення про практичні аспекти обчислення сплайнів за допомогою комп'ютерів.

### План лекції

[7.1 Апроксимація функцій методом найменших квадратів](#)

[7.2 Точкова апроксимація функцій методом найменших квадратів](#)

[7.3 Кусково-поліноміальна апроксимація](#)

[7.4 Поняття сплайну](#)

[7.5 Способи аналітичного зображення сплайнів](#)

[7.6 Інтерполяція кубічними сплайнами](#)

[7.7 Обчислення параметрів кубічного сплайну](#)

[Контрольні питання](#)

[Завдання для самостійної роботи](#)

## 7.1 Апроксимація функцій методом найменших квадратів

Якщо значення функції  $f(x)$  отримані в результаті експерименту з випадковими погрішностями (шумами), для апроксимації розумніше застосовувати середнє квадратичні наближення, які:

- ✓ згладжують окремі локальні помилки спостережень;
- ✓ краще представляють реальну функціональну залежність.

Ілюстрація розходження між інтерполяцією й середнє квадратичною апроксимацією наведена на рис. 7.1. Як виходить з рис. 7.1:

- ✓ лінія інтерполяції проходить через всі точки, створюючи досить вигадливу залежність, якої не може відповідати фізичному змісту задачі,
- ✓ лінія середнє квадратичної апроксимації проходить усереднено й по своїй конфігурації є більше "природної" і простий.

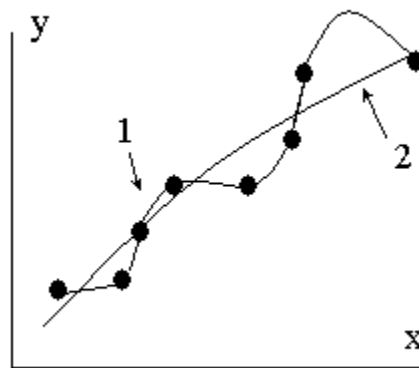


Рисунок 7.1 – Методи наближення експериментальної залежності шляхом інтерполяції (1) і середнє квадратичної апроксимації (2)

Для побудови найкращого середньоквадратичного наближення використовується метод найменших квадратів (МНК) наступним чином:

а) нехай функція  $f(x)$  задана значеннями, які можуть містити випадкові помилки, на дискретній або неперервній множині  $X \subseteq [a, b]$ ;

б) оберемо систему лінійно незалежних на  $X$  функцій  $\varphi_0(x), \varphi_1(x), \dots, \varphi_n(x)$  і розглянемо їх усілякі лінійні комбінації:

$$\varphi(x) = \sum_{i=1}^n c_i \varphi_i(x); \quad (7.1)$$

в) будемо апроксимувати функцію  $f(x)$  узагальненим поліномом (6.74). За принципом найменших квадратів (див. 4.1), коефіцієнти  $c_0, c_1, \dots, c_n$  узагальненого полінома (6.74) визначаються з умови мінімуму функціонала:



Переваги ортогонального базису:

- ✓ простота обчислень;
- ✓ при збільшенні розмірності базису *всі попередні коефіцієнти не змінюються* (у будь-якій неортогональній системі всі обчислення коефіцієнтів треба робити заново);
- ✓ *стійкість результатів*, можливість використання для рішення завдань із більшою кількістю параметрів.

Властивості ортогонального базису:

а) сума квадратів відхилень:

$$\delta_n^2 = \|f - \varphi\|^2 = (f, f) - \sum_{i=0}^n c_i^2 (\varphi_i, \varphi_i) = \|f\|^2 - \sum_{i=0}^n c_i^2 \|\varphi_i\|^2. \quad (7.7)$$

Оскільки  $\delta_n^2 \geq 0$ , те з (7.7) випливає нерівність:

### ФУР'Є ЖАН БАТИСТ ЖОЗЕФ (1768-1830)



Видатний французький математик, член Паризької Академії наук, один із засновників математичної фізики. На честь Фур'є названо декілька методів, які він запропонував: розкладання функцій у тригонометричні ряди, рівняння теплопровідності, метод розв'язання диференціальних рівнянь у частинних похідних шляхом поділу змінних. Іноземний член Петербурзької Академії наук, учасник походу Наполеона до Єгипту.

$$\sum_{i=0}^n c_i^2 \|\varphi_i\|^2 \leq \|f\|^2; \quad (7.8)$$

б) якщо базисні функції  $\{\varphi_i(x)\}_0^n$  є ортонормованими, то для коефіцієнтів Фур'є виконується *нерівність Бесселя*:

$$\sum_{i=0}^n c_i^2 \leq \|f\|^2. \quad (7.9)$$

В силу рівності (7.9) при збільшенні порядку  $n$  узагальненого полінома  $\varphi(x) = \sum_{i=0}^n c_i \varphi_i(x)$  погрішність середнє квадратичного наближення  $\delta_n^2$  зменшується.

Якщо:

$$\lim_{n \rightarrow \infty} \left\| f - \sum_{i=0}^n c_i \varphi_i \right\|_{L_2^D[a,b]} = 0, \quad (7.10)$$

те говорять, що послідовність узагальнених поліномів по ортогональній системі збігається в середньому до функції  $f(x)$ .



### БЕССЕЛЬ ФРІДРІХ ВІЛЬГЕЛЬМ (1784-1846)

Відомий німецький математик, астроном і геодезист, член Берлінської Академії наук, почесний іноземний член Петербурзької Академії наук. Зробив значний внесок у математичну фізику, чисельні методи, теорію спеціальних функцій. Ряд об'єктів математики, зокрема нерівність, циліндричні функції, інтерполяційна формула носять його ім'я

Збіжність у середньому пов'язана з поняттям *повноти* системи базисних функцій. Якщо система функцій  $\varphi_0(x), \varphi_1(x), \varphi_2(x) \dots$  є повною в лінійному нормованому просторі  $L$ , то для будь-якої функції  $f(x) \in L$  і будь-якого  $\varepsilon > 0$  існує узагальнений поліном  $n$ -го порядку:

$$\varphi(x) = \sum_{i=0}^n c_i \varphi_i(x) \in H_{n+1} \subset L, \quad (7.11)$$

такий, що  $\|f - \varphi\| < \varepsilon$ . В цьому випадку ряд  $\varphi(x) = \sum_{i=0}^{\infty} c_i \varphi_i(x)$  збігається до функції  $f(x)$  в метриці  $L$ , тобто для  $n$ -й часткової суми  $S_n(x) = \varphi(x)$  виконується співвідношення:

$$\lim_{n \rightarrow \infty} \|f - S_n\| = 0. \quad (7.12)$$

Узагальнений ряд Фур'є  $\sum_{i=0}^{\infty} c_i \varphi_i(x)$  для довільної функції  $f(x) \in L_2^p[a, b]$  по *повній* ортонормированій системі функцій  $\varphi_0(x), \varphi_1(x), \varphi_2(x) \dots$  збігається у середньому до  $f(x)$ .

Умова повноти ортонормированої системи функцій еквівалентно тому, що для будь-якої функції  $f(x) \in L_2^p[a, b]$  виконується умова:

$$\sum_{i=0}^{\infty} c_i^2 = \|f\|^2, \quad (7.13)$$

де  $c_i$  - коефіцієнти Фур'є функції  $f(x)$  по даній системі функцій.

Вираз (7.13) називається *рівністю Парсеваля*<sup>1)</sup>. Він виходить із нерівності Бесселя (7.9) при  $n \rightarrow \infty$ .

Рівність (7.13) називають також *рівнянням замкнутості*. У просторі  $L_2^p[a, b]$  поняття замкнутості й повноти еквівалентні.

Якщо  $X$  - дискретна множина точок  $x_0, x_1, \dots, x_m$ , то будь-яка система  $(m + 1)$ -й функцій, лінійно незалежних на множині точок  $X$ , буде повною на цій множині.

## 7.2 Точкова апроксимація функцій методом найменших квадратів

Нехай у результаті спостережень на дискретній множині точок  $x_0, x_1, \dots, x_m$  отримана таблиця значень функції  $f(x)$ :  $y_0=f(x_0); y_1=f(x_1), \dots, y_m=f(x_m)$ . Побудуємо на цій множині середнє квадратичне наближення у вигляді узагальненого полінома:

$$\varphi(x) = \sum_{i=0}^n c_i \varphi_i(x). \quad (7.14)$$

Допущення: значення аргументу визначені значно точніше, ніж значення функції.:

Принцип найменших квадратів: потрібно, щоб зважена сума квадратів відхилень значень узагальненого полінома  $\varphi(x)$  від заданих значень функції  $f(x)$  у всіх вузлах  $x_i (i = 0, 1, \dots, n)$  була мінімальною:

$$\begin{aligned} \Phi(c_0, c_1, \dots, c_n) &= \sum_{i=0}^m p_i (\varphi(x_i) - f(x_i))^2 = \\ &= \sum_{i=0}^m p_i \left( \sum_{j=0}^n c_j \varphi_j(x_i) - f(x_i) \right)^2 \rightarrow \min \end{aligned} \quad (7.15)$$

Параметри полінома  $\varphi(x)$  визначають з умови мінімуму  $\Phi$ :

$$\begin{aligned} \frac{\partial \Phi(c_0, c_1, \dots, c_n)}{\partial c_k} &= 2 \sum_{i=0}^m p_i \left( \sum_{j=0}^n c_j \varphi_j(x_i) - f(x_i) \right) \cdot \varphi_k(x_i) = 0. \\ (k &= 1, 2, \dots, n) \end{aligned} \quad (7.16)$$

Значення цих параметрів є розв'язками системи алгебраїчних рівнянь (7.17), які можна отримати внаслідок перетворень рівнянь (7.16):

$$\begin{aligned} \sum_{i=0}^m p_i \left[ \sum_{j=0}^n c_j \varphi_j(x_i) \varphi_k(x_i) \right] &= \sum_{i=0}^m p_i f(x_i) \varphi_k(x_i). \\ (k &= 1, 2, \dots, n) \end{aligned} \quad (7.17)$$

<sup>1)</sup> Парсеваль Марк Антуан (1755-1836) – французький математик. Основні праці в галузі диференціальних рівнянь і функцій дійсної змінної

Якщо в методі найменших квадратів  $n=m$ , то точкова середнє квадратична апроксимація алгебраїчним многочленом збігається з інтерполяцією поліномом Лагранжа. На практиці кількість шуканих параметрів вибирають значно меншою за кількістю числа спостережень:  $n \ll m$ . Тоді випадкові помилки окремих вимірів погашають один одного й розв'язання стає більше достовірним.

Вибір базисних функцій  $\varphi_i(x)$  ( $i = 0, 1, \dots, n$ ) частіше за все диктується фізичним змістом задачі, для якої отримані експериментальні дані. Іноді ті самі вихідні дані майже з однаковою точністю вдається згладити кривими, що є зовсім різними за функціональною природою.

Найбільш часто на практиці вважають  $\varphi_i(x) = x^i$ , тобто, як апроксимуючу функцію використовують многочлен:

$$\varphi(x) = c_0 + c_1x + \dots + c_nx^n. \quad (7.18)$$

Ступінь многочлена вибирають значно менше, ніж кількість точок.

У випадку поліноміальної апроксимації система рівнянь (7.17) має вигляд:

$$\sum_{i=0}^m p_i \left[ \sum_{j=0}^n c_j x_i^j \cdot x_i^k \right] = \sum_{i=0}^m p_i f(x_i) \cdot x_i^k. \quad (7.19)$$

$$(k = 1, 2, \dots, n)$$

Ваги  $p_i$  призначені, щоб урахувати різну точність (погрішність) у різних точках:

а) якщо точність по всьому масиві даних є однаковою - дані називаються *рівноточними*, і ваги всіх спостережень уважають однаковими. У результаті відбувається їхнє скорочення в рівняннях (7.17). Це еквівалентно тому, що в *рівноточних спостереженнях всі ваги рівні 1*;

б) у випадку *нерівноточних спостережень* ваги будуються на основі середнє квадратичних відхилень  $\sigma_i$  в кожній точці  $u_i$ :

➤ обчислюється узагальнене середнє квадратичне відхилення:

$$\sigma_S = \sqrt{\frac{\sigma_0 + \sigma_1 + \dots + \sigma_n}{n+1}}; \quad (7.20)$$

➤ вагу  $i$ -го спостереження розраховують по формулі:

$$p_i = \frac{\sigma_S}{\sigma_i} \quad (i = 0, 1, 2, \dots, n) \quad (7.21)$$

Відмітимо, що задача, що є аналогічною (7.17) – це задача регресійного аналізу у математичній статистиці й аналізі даних. Задачі регресійного аналізу полягають у наступному: по заданій сукупності результатів спостережень, що містять статистичні похибки, необхідно знайти коефіцієнти заданого рівняння зв'язку змінних і дослідити їх статистичні властивості. Перша частина задачі розв'язується

методом найменших квадратів, аналогічно розглянутому вище. При цьому виникає питання оцінки, наскільки добре рівняння описує дані.

Нехай є набір даних у вигляді таблиці, що містить:

- ✓  $n$  стовпців вхідних змінних  $x_1, x_2, \dots, x_n$  (вони називаються факторами);
- ✓ стовпець вихідної змінної  $y$ .

Нехай відоме, що зв'язок між змінними (функціональний або функціонально-стохастичний) задається моделлю:

$$\tilde{y} = f(x_1, x_2, \dots, x_n, a_0, \dots, a_q), \quad (7.22)$$

де  $a_0, \dots, a_q$  – параметри моделі, що підлягають визначенню.

Для визначення параметрів використовується МНК, тобто розбудовується функціонал (7.23):

$$\begin{aligned} \Phi(a_0, a_1, \dots, a_q) &= \sum_{i=1}^m p_i (y_i - \tilde{y}_i)^2 = \\ &= \sum_{i=1}^m p_i [y_i - f(x_{1,i}, \dots, x_{n,i}, a_0, \dots, a_q)]^2 \rightarrow \min \end{aligned} \quad (7.23)$$

Порівнюючи, можна прийти до висновку, що функціонали (7.15) і (7.23) збігаються з точністю до позначень, тому:

- ✓ для розв'язання задач (7.15) і (7.23) можна використовувати ті ж самі методи,
- ✓ при інтерпретації результатів можна використовувати ті ж самі характеристики.

У регресійному аналізі прийнято важливою характеристикою вибірки значень вихідної величини є загальна сума квадратів відхилень:

$$SS_y^2 = \sum_{i=1}^m (y_i - \bar{y})^2, \quad (7.24)$$

де  $\bar{y} = \frac{\sum_{i=1}^m y_i}{m}$  - середнє арифметичне значення  $y$ .

Доведено, що для виявленої регресійної залежності (7.22) сума квадратів (7.24) може бути представлена у виді суми двох складових:

$$SS_y^2 = \sum_{i=1}^m (y_i - \bar{y})^2 = \sum_{i=1}^m (y_i - \tilde{y}_i)^2 + \sum_{i=1}^m (\tilde{y}_i - \bar{y})^2. \quad (7.25)$$

Перше складове:

$$SS_{\text{зан.}}^2 = \sum_{i=1}^m (y_i - \tilde{y}_i)^2 \quad (7.26)$$

називається *залишковою сумою квадратів*. Воно характеризує відхилення величин  $y$ , що задані й розраховані за рівнянням моделі (7.22) (при величині параметрів  $a_0, \dots, a_q$ , що забезпечують мінімум функціоналу (7.23)), тобто, *є мірою якості моделі*. Чим менше ця величина, тим ближче рівняння моделі до даних.

Друге складове:

$$SS_{\text{пер.}}^2 = \sum_{i=1}^m (\tilde{y}_i - \bar{y})^2 \quad (7.27)$$

називається *сумою квадратів відхилень відносно регресії*, воно характеризує розсіяння точок, що лежать на лінії регресії, відносно середнього.

За мірою якості моделей в регресійному аналізі використовують: залишкову дисперсію:

$$S_{\text{зал.}}^2 = \frac{SS_{\text{зал.}}^2}{m-q} = \frac{\sum_{i=1}^m (y_i - \tilde{y}_i)^2}{m-q}, \quad (7.28)$$

де  $m-q$  носить назву *кількості ступенів волі*. Ця величина характеризує кількість незалежних точок, що можуть бути використані для перевірки вірності моделі ( $q$  точок використовуються для одержання параметрів моделі);

залишкове середнє квадратичне відхилення:

$$S_{\text{зал.}} = \sqrt{S_{\text{зал.}}^2} = \sqrt{\frac{\sum_{i=1}^m (y_i - \tilde{y}_i)^2}{m-q}}. \quad (7.29)$$

Коефіцієнт детермінації:

$$R^2 = \frac{SS_{\text{пер.}}^2}{SS_y^2} = 1 - \frac{SS_{\text{зал.}}^2}{SS_y^2}. \quad (7.30)$$

Коефіцієнт детермінації характеризує частку відхилень від середнього значення  $\bar{y}$ , яка може бути пояснена наявністю рівняння регресії (7.15). Він змінюється від 0 до 1. Чим ближче  $R^2$  до 1, тим ближче лінія регресії до даних.

Ті ж самі характеристики можна на практиці використовувати для оцінки якості середнє квадратичної апроксимації по МНК, у випадку детермінованих задач, тобто, заміни однієї функції іншою, що є більш простою. На практиці частіше за все невідомо, за допомогою якого рівняння слід апроксимувати дані або функцію, що задано. Тому доводиться проводити дослідження декількох варіантів рівнянь апроксимації.

Для розрахунків точкової апроксимації методом найменших квадратів можна використати стандартні пакети й програми регресійного статистичного аналізу. У середовищі OOo Calc для цього можна використати стандартну функцію LINEST()

### Приклад 7.1

На відрізку  $[0;1,2]$  з кроком 0,1 побудувати точкову середнє квадратичну апроксимацію функції:

$$y = \frac{\text{sh}(x) \cdot \sin(x^2 + 1)}{\text{ch}(x^2 + 1)} \quad (7.31)$$

Використати систему базисних функцій:

$$\varphi_0 = 1; \varphi_1 = x; \varphi_2 = x^2, \varphi_3 = x^3, \varphi_4 = x^4.$$

Перевірити, як впливає на залишкову дисперсію і коефіцієнт детермінації. Визначити достатню кількість функцій для апроксимації, щоб забезпечити середнє квадратичне відхилення не більше 0,02

Розв'язання

Проект ООо Calc наведено на рис. 7.2. Як впливає з графіка на рис. 7.2, залежність між  $y$  і  $x$  є суттєво нелінійною, тому для апроксимації доцільно використовувати системи функцій, починаючи з квадратичної. Система функцій буде:

$$1) y = a_0\varphi_0(x) + a_1\varphi_1(x) + a_2\varphi_2(x) = a_0 + a_1x + a_2x^2;$$

$$2) y = a_0\varphi_0(x) + a_1\varphi_1(x) + a_2\varphi_2(x) + a_3\varphi_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3;$$

$$3) y = a_0\varphi_0(x) + a_1\varphi_1(x) + a_2\varphi_2(x) + a_3\varphi_3(x) + a_4\varphi_4(x) = \\ = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4;$$

У середовищі Ооо Calc:

- ✓ формуємо таблицю даних (рис. 7.2, стовпці В,С);
- ✓ формуємо стовпці значень базисних функцій (рядки D, E, F, G).

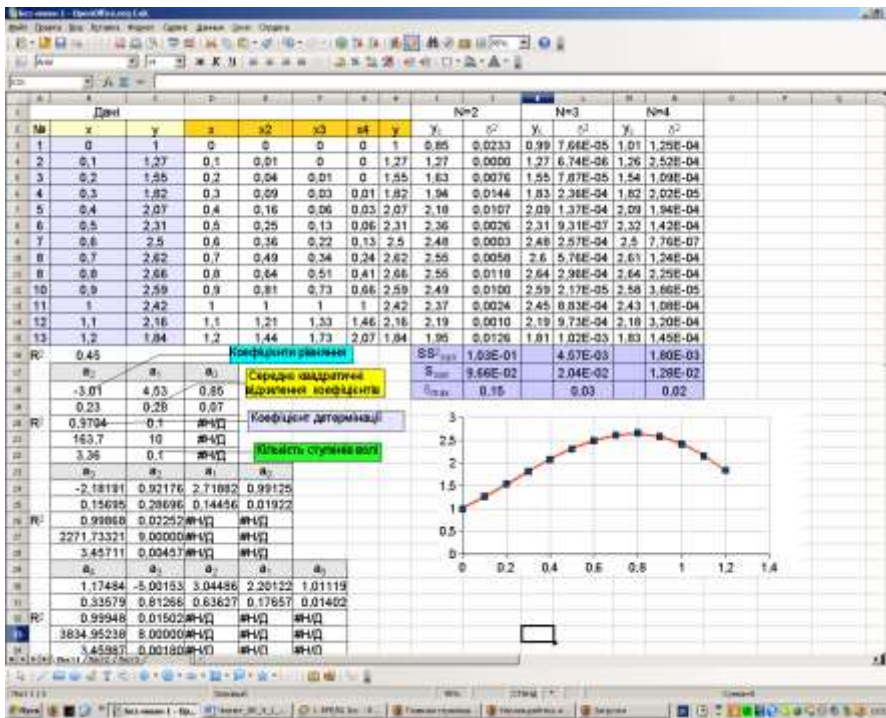


Рисунок 7.2 – Проект «Апроксимація» до прикладу 7.1

Формули комірок (по рядку 3):

- D3: =B3;
- E3: =D3\*\$D3
- F3: =E3\*\$D3
- G3: =F3\*\$D3
- H3: = C3

Формуємо масив комірок B18:C22 для розрахунків функції LINEST() і викликаємо Майстра функцій для виконання розрахунків. Функція LINEST() знаходиться серед функцій масиву. Робоче вікно цей функції наведено на рис. 7.3. Аргументи функції LINEST() наступні:

- ✓ значення Y – вводяться координати масиву вихідної змінної;
- ✓ значення X – вводяться координати масиву вхідних змінних (у даному випадку – x і x<sup>2</sup>);

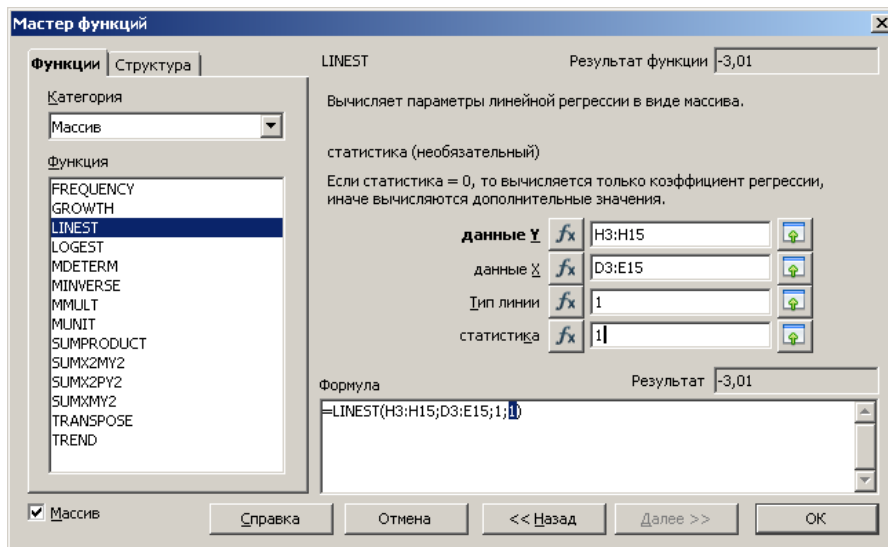


Рисунок 7.3 – Робоче вікно функції LINEST()

- ✓ Тип лінії. Уводиться:
  - 0 - якщо лінія повинна проходити через начало координат ;
  - будь-яке число, що відмінне від 0 (у нашому випадку 1), якщо лінія не проходить через начало координат;
- ✓ Статистика. Уводиться:
  - 0 - якщо на екран виводяться тільки коефіцієнти,
  - будь-яке число, що відмінне від 0 (у нашому випадку 1), якщо на екран виводиться статистика коефіцієнтів (у тому числі коефіцієнт детермінації)

В результаті одержуємо таблицю вихідних значень функції LINEST() (рис. 7.2), що містить:

- ✓ у першому рядку – значення коефіцієнтів, що розташовані з права наліво за зростанням індексу;
- ✓ у другому рядку розташовані середнє квадратичні відхилення коефіцієнтів,
- ✓ у першій клітинці третього рядку міститься коефіцієнт детермінації,
- ✓ у другій клітинці четвертого рядку міститься кількість ступенів волі,
- ✓ в інших клітинках містяться сумі, що необхідні для статистичного аналізу, але не є суттєвими з точки зору апроксимації.

Далі для набору базисних функцій 1, x, x<sup>2</sup> розраховуємо в комірках I3:I15 значення апроксимуючої функції – поліному другого порядку. Формула комірки I3: = $\$D\$18+\$C\$18*D3+\$B\$18*E3$ ;

В комірках J3:J15 розраховуємо  $\delta_i^2=(y_i - y_{ic})^2$  – квадрати нев'язок між даними і моделлю. Формула комірки J3: =(H3-I3)^2

В комірці J16 розраховуємо суму квадраів відхилень, формула комірки: =SUM(J3:J15).

В комірці J17 розраховуємо залишкове середнє квадратичне відхилення. Формула комірки: =(J16/11)^0,5.

В комірці J18 розраховуємо модуль максимального відхилення. Формула комірки: =(MAX(J3:J15)^0,5).

Аналогічні розрахунки проводимо для N=3 (апроксимація кубічним поліномом) і N = 4 (апроксимація поліномом 4-го ступеня). Формули масивів і комірок наведені у табл. 7.1

Таблиця 7.1 Формули комірок і масивів проекту (рис. 7.2)

Комірка (масив)	формула	Що розраховує
B24:E28	=LINEST(H3:H15;D3:F15;1;1)	Функцію LINEST() для N = 3
K3:K15	=\$E\$24+\$D\$24*D3+\$C\$24*E3+\$B\$24*F3 (комірка K3)	Функцію що апроксимує для N=3
B30:F34	=LINEST(H3:H15;D3:G15;1;1)	Функцію LINEST() для N = 4
L3:L15	=\$F\$30+\$E\$30*D3+\$D\$30*E3+\$C\$30*F3+\$B\$30*G3 (комірка L3)	Функцію що апроксимує для N=4

Як впливає з результатів розрахунків, заданим критеріям точності задовольняє апроксимація поліномами 3 і 4 ступеня. Різниця між останніми є невеликою. Як що розглядати закономірності зміни коефіцієнту детермінації від ступеня поліному, те можна прийти до висновку, що найбільш сильна зміна відбувається при зростанні ступені поліному від 1 до 2 (від 0,45 до 0,97). Відмітимо, що для простої лінійної залежності коефіцієнт детермінації може бути розрахований, як квадрат коефіцієнту кореляції, за допомогою функції CORREL().

Шукане рівняння апроксимації буде:

$$y = 1,01119 + 2,20122 \cdot x + 3,04486 \cdot x^2 - 5,00153 \cdot x^3 + 1,17484 \cdot x^4. \quad (7.32)$$

Як правило задача апроксимації по МНК не є однозначною: з різними системами базисних функцій можна одержати однакові якості апроксимації за критеріями, що наведені вище.

### Приклад 7.2

Одержати апроксимацію даних (приклад 7.1) за допомогою базисних функцій:

$$\varphi_0 = 1; \varphi_1 = x; \varphi_2 = e^x; \varphi_3 = e^{2x}$$

### Розв'язання

Скріншот проекту наведено на рис. 7.4. Порядок виконання розрахунків:

а) будуємо таблицю значень базисних функцій, включаючи  $\varphi_0 = 1$ ;

б) будуємо таблицю результатів для функції LINEST(). Формула масиву B20:E24: =LINEST(H3:H15;D3:G15;0;1). Відмітимо, що ми використовуємо опцію «тип лінії» 0, оскільки використовуємо постійне значення  $\varphi_0 = 1$  за базисну функцію;

в) у масиві I3:I15 розраховуємо значення  $u_c$ , що лежать на лінії апроксимуючої функції. Формула комірок I3:

$$= \$E\$20 + \$D\$20 * E3 + \$C\$20 * F3 + \$B\$20 * G3;$$

д) за графічний образ точності новодимо залежність між  $u_c$  від  $u$ . Як впливає з рис. 7.4, усі точки добре розташовуються вздовж прямоку, з коефіцієнтом детермінації 0,996, що є дуже близьким до 1.

Виходячі з результатів розрахунків, функція, що апроксимує, має вид:

$$y = 1,19031 + 3,89112 \cdot x + 0,20237 \cdot e^x - 0,42903 \cdot e^{2 \cdot x} \quad (7.33)$$

Якість апроксимації (критерій – коефіцієнт детермінації) є майже ідентичною з апроксимацією поліномом (приклад 7.1)

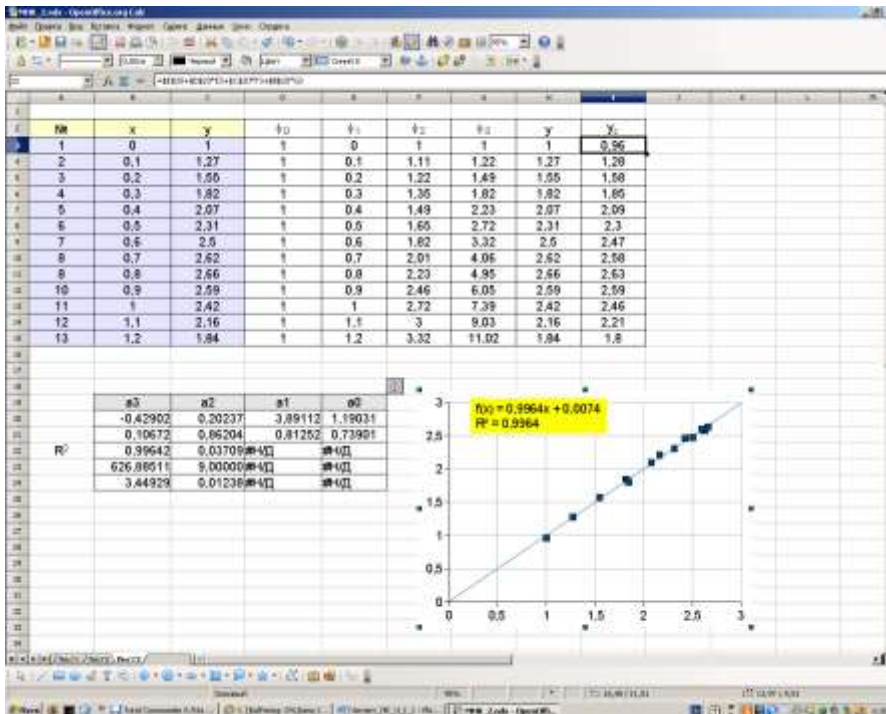


Рисунок 7.4 – Проект розрахунків до прикладу 7.2

Розрахунки за методом найменших квадратів можна проводити також в середовищі Scilab за допомогою функції `datafit()`. Структура функції:

$$[c, err] = \text{datafit}(G, z, c0),$$

де  $c$  - вектор коефіцієнтів, що визначаються;

$err$  – похибка (сума квадратів відхилень);

$G$  – цільова функція, що визначає нев'язки,

$z$  – сполучений масив вхідних і вихідної змінної

$c0$  – початкове наближення вектору коефіцієнтів, що визначаються.

Перед використанням цієї функції необхідно:

- ✓ визначити сполучений масив змінних,
- ✓ визначити підпрограму-функцію  $G$ .

У сполученому масиві відгук у рекомендується організувати, як *останній за номером вектор-рядок*.

Підпрограма-функція  $G$  розраховує нев'язки, що виражені через сполучені змінні  $z$  і вектори коефіцієнтів  $c$ .

### Приклад 7.3

Розв'язати [приклад 7.2](#) в середовищі Scilab

Розв'язання

У лістингу 7.3 наведено текст скрипту для розв'язання задачі.

### Лістинг 7.3

```
//Функція, що розраховує різниці між заданими
//та апроксимованими значеннями у
//Перед використанням необхідно визначити
//z=[x;y] - матрицю вхідних даних і
//c - вектор початкових значень коефіцієнтів,
//його розмірність повинна збігатися з
//кількістю шуканих коефіцієнтів
function [zr]=G(c, z)
zr=z(2)-c(1)-c(2)*z(1)-c(3)*exp(z(1))-c(4)*exp(2*z(1))
endfunction
//Исходные данные
x=[0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,1.1,1.2];
y=[1,1.27,1.55,1.82,2.07,2.31,2.5,2.62,...
2.66,2.59,2.42,2.16,1.84];
//Формування матриці вхідних даних
z=[x;y];
//Вектор початкових наближень
c0=[0;0;0;0];
//Решение задачи
[c,err]=datafit(G, z, c0)
for i=1:13// Друк вихідних результатів
    yc(i)=c(1)+c(2)*x(i)+c(3)*exp(x(i))+c(4)*exp(2*x(i));
```

```
end  
[x;y;yc']'
```

Результати роботи програми наведені у протоколі 7.3

Протокол 7.3

```
err =  
0.0122163  
с =  
1.1746699  
3.8781517  
0.2213860  
- 0.4321777  
ans =  
0. 1. 0.9638781  
0.1 1.27 1.2792913  
0.2 1.55 1.5759682  
0.3 1.82 1.849476  
0.4 2.07 2.0943704  
0.5 2.31 2.3039686  
0.6 2.5 2.4700718  
0.7 2.62 2.5826255  
0.8 2.66 2.6293044  
0.9 2.59 2.5950051  
1. 2.42 2.4612255  
1.1 2.16 2.2053071  
1.2 1.84 1.7995078
```

Порівнюючи з результатами у прикладі 7.2, можна переконатися, що вони збігаються з результатами прикладу 7.3.

Для розв'язання задач апроксимації МНК в середовищі Scilab також можна користуватися функцією `optim()`, що описана у розділі [5.10](#).

### 7.3 Кусково-поліноміальна апроксимація

Проблема інтерполяції на великому масиві: при великій кількості точок  $n$  виникає *осциляція* многочлена  $P_n(x)$  між вузлами, оскільки він має точки максимуму й мінімуму в речовинних нулях похідної.

Можливий шлях усунення - зменшення ступеня полінома за рахунок використання при інтерполяції тільки частини точок (кускова апроксимація). Тоді для всієї множини точок виходить складова крива, що є *неперервною* на всьому інтервалі, але у вузлах "склейки" окремих кривих її *похідні будуть розривними*.

Недоліки:

а) погрішність такого наближення може виявитися значною;

б) воно неприйнятно в тих практично важливих задачах, коли гладкість апроксимуючої функції є суттєвою.

Побудова полінома невисокого ступеня для великої кількості точок методом найменших квадратів часто *не забезпечує* задовільної відповідності кривої заданим точкам, а підвищення ступеня полінома приводить до істотного ускладнення обчислень.

### Приклад.7.4

Побудувати інтерполяційний поліном і кусково-поліноміальну апроксимацію поліномами 3-й ступеня для функції, заданої таблицею (у припущенні її максимальної гладкості). Значення аргументу й функції задані в таблиці 7.2

Таблиця 7.2 - Значення x и y для прикладу 7. 4

x	y	x	y	x	y
0	0,6	4	0,55	8	0,78
1	0,55	5	0,67	9	0,72
2	0,35	6	0,75	10	0,6
3	0,13	7	0,8		

- ✓ за допомогою OOo Calc методами, описаними в [розділі 6.5](#), будемо інтерполяційний поліном Лагранжа 10-й ступеня. Його рівняння:

$$y = 0,6 - 10,2226x + 26,85239x^2 - 27,84688x^3 + 15,23x^4 - 4,946x^5 + 1,006x^6 - 0,13x^7 + 0,01x^8 + 9 \cdot 10^{-6}x^{10}; \quad (7.34)$$

✓ Графіки шуканої функції й полінома Лагранжа наведені на рис.

7.5. Як виходить з рис. 7.5, інтерполяція поліномом Лагранжа слід визнати незадовільною: хоча є збіг значень функції й полінома у вузлових точках, між ними поліном незадовільно передає характер зміни функції;

- ✓ кусково-поліноміальна інтерполяція:
  - Розділимо область зміни на наступні інтервали, що наведені в таблиці 7.2 . Відзначимо, що кінці 1 і 2 інтервалів одночасно є, початком відповідно 2 і 3 інтервалів. Це приводить до того, що значення функції на границях інтервалів *однакові*;
  - усередині кожного інтервалу будемо інтерполяційний поліном. Для першого й другого інтервалу порядок інтерполяційного полінома дорівнює 3, для 3-го - чотирьом. Рівняння таких локальних поліномів наведені в таблиці 7.4.

На рис. 7.6 наведені графіки функції й кусково-поліноміальної інтерполяції

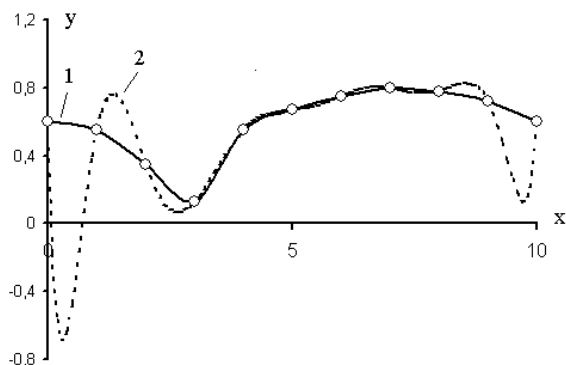


Рисунок 7.5 - Графіки заданої функції (1) і полінома Лагранжа (2)

Таблиця 7.3 - Поділ даних табл. 7.1 на 3 інтервали

1-й інтервал		2-й інтервал		3-й інтервал	
x	y	x	y	x	y
0	0,6	3	0,13	6	0,75
1	0,55	4	0,55	7	0,8
2	0,35	5	0,67	8	0,78
3	0,13	6	0,75	9	0,72
				10	0,6

Таблиця 7.4 - Інтерполяційні поліноми на ділянках 1-3

Номер	Границі	Рівняння полінома
1	[0;3]	$y = 0,6 + 0,068333x - 0,14x^2 + 0,021667x^3$
2	[3;6]	$y = -5,53 + 3,506667x - 0,67x^2 + 0,043333x^3$
3	[6;10]	$y = -9 + 4,6725x - 0,83792x^2 + 0,0675x^3 - 0,00208x_4$

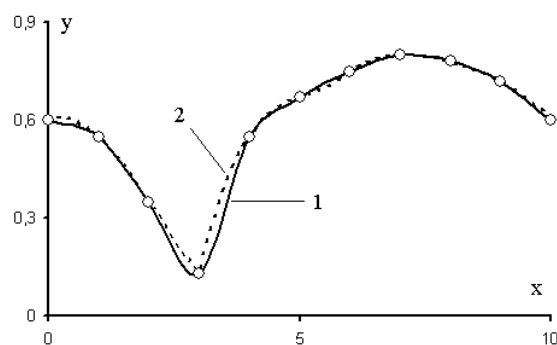


Рисунок 7.6 - Графік заданої функції (1) і кусково-поліноміальної інтерполяції (2)

Інтерпретація. Поліном на рис. 7.6 значно краще, ніж інтерполяційний поліном Лагранжа. Разом з тим, якщо розрахувати похідні в граничних точках, то вони виявляються *різними*, тобто, похідна в граничних крапках перетерплює *розрив*. Для розглянутого приклада в точці  $x=3$  похідна з боку 1-го інтервалу дорівнює  $-0,77167$ , а з боку 2-го інтервалу:  $-0,51333$ . Для багатьох практичних задач потрібно, щоб похідні у всіх точках не перетерплювали розриву.

## 7.4 Поняття сплайну

Подальший розвиток кускової інтерполяції - апроксимація *сплайнами* (сплайн-функціями), яка була розроблена в 70-х роках 20 століття<sup>1)</sup>.

У загальному випадку сплайн являє собою функцію, яку "склеєно" з узагальнених поліномів таким чином, що в точках "склейки" їхні значення й значення похідних до деякого порядку збігаються. Найбільш простими є кусочно-поліноміальні сплайни, для яких за базисні функції беруться алгебраїчні поліноми.

Сплайни в цей час широко застосовуються в різних комп'ютерних технологіях: в автоматизації проектування на ЕОМ, у конструюванні кривих і поверхонь, в обробці й поданні геометричної інформації, а також у прикладній математиці - у теорії апроксимації, при чисельному інтегруванні й диференціюванні, чисельному рішенні диференціальних і інтегральних рівнянь.

Апарат сплайнов володіє рядом переваг у порівнянні із класичною інтерполяцією:

- ✓ *простота реалізацією на ЕОМ*. При обчисленні параметрів інтерполяційного сплайна, на відміну від інтерполяційних поліномів, потрібне рішення системи алгебраїчних рівнянь із трьохдіагональною матрицею;
- ✓ *забезпечення високої точності* апроксимації одночасно функції та її 1 і 2-й похідних;
- ✓ у ряді важливих випадків сплайни дозволяють одержати наближення з *мінімально можливою погрешністю* на даному класі функцій у порівнянні з іншими методами;
- ✓ сплайни мають *стійкість щодо локальних збурювань*: незначна зміна значень функції в одному або декількох вузлах інтерполяції мало позначається на його значеннях на деякій видаленні від цих точок;
- ✓ сплайни можна застосовувати для наближення функцій, у яких *ступінь гладкості неоднакова* на різних ділянках відрізка апроксимації.
- ✓ сплайни порівняно просто *узагальнюються* на багатовимірні задачі.

Назва сплайн (англ. Spline - гнучка креслярська лінійка) походить із технічного креслення. Для проведення гладких кривих через систему точок у креслярській

<sup>1)</sup>Вперше термін «сплайн» для кускових гладких многочленів використав у 1946 р. американський математик румунського походження Ісаак Якоб Шенберг (1903-1990).

практиці використовують гнучкі лінійки. Їх згинають так, щоб вони стосувалися крапок і після цього проводять лінію.

Нехай на відрізку  $[a;b]$  задана сітка вузлів  $\Delta$ :

$$a = x_0 < x_1 < x_2 < \dots < x_n = b. \quad (7.35)$$

Позначимо:

- через  $C_k[a; b]$  – множину функцій, що неперервні на  $[a;b]$  і мають неперервні похідні до  $k$ -го порядку:

- через  $P_n$  – множину многочленів ступеня не вище  $n$ .

Визначення. Функція називається *поліноміальним сплайном* ступеня  $n$  дефекту  $k$  ( $1 < k < n$ ) з вузлами на сітці  $\Delta$ , якщо:

- ✓  $s_n^k(x) \in P_n(x) \forall x \in [x_i; x_{i+1}]$  ( $i = 0, 1, 2, \dots, n-1$ );
- ✓  $s_n^k(x) \in C^{n-k}[a; b]$ .

Дефект  $k$  визначає ступінь гладкості сплайну у вузлах і означає, що:

- функція  $s_n^k(x)$  та її похідні до  $(n - k)$ -го порядку є *неперервними* на  $[a; b]$ ;
- похідна порядку  $(n - k + 1)$  може бути *розривною* в точках  $x_i$  ( $i = 1, 2, \dots, n - 1$ ).

Якщо  $k = 1$ , то для сплайнів дефекту 1 вживають позначення  $S_n(x)$ .

Будемо розглядати сплайни як апарат наближення функції  $f(x)$ , що задана своїми значеннями на дискретній множині точок. Для сплайнів *непарного ступеня* вузли інтерполяції вибирають, як правило, у вузлах сплайна.

Нехай на відрізку  $[a; b]$  у вузлах сітки  $\Delta$  задані значення функції  $f(x_i) = y_i$  ( $i = 0, 1, \dots, n$ ).

Сплайн  $s_{2n+1}^k(x)$  називається *інтерполяційним*, якщо він задовольняє умовам інтерполяції:

$$s_{2n+1}^k(x_i) = f(x_i) \quad (i = 0, 1, 2, \dots, n). \quad (7.36)$$

Для однозначного визначення параметрів сплайна необхідно задавати :

- а) умови інтерполяції (7.36);
- б) додаткові крайові умови (див. [7.6](#))

Для сплайнів *парного ступеня* вузли інтерполяції вибирають *відмінними від вузлів сплайну*, тому що інакше може виявитися, що при деяких умовах сплайн не існує або ж процес обчислення параметрів сплайну буде нестійким.

## 7.5 Способи аналітичного зображення сплайнів

У цьому розділі будуть розглянуті різні способи аналітичного зображення сплайнів залежно від вибору базисних функцій.

Кусково-поліноміальне зображення. Сплайн на кожному відрізку  $[x_i; x_{i+1}]$  ( $i = 0, 1, \dots, n-1$ ) можна представити у вигляді:

а) полінома відносно *початкової* точки відрізка:

$$S_n^k(x) = \sum_{j=0}^n a_{ij}(x - x_i)^j, \quad x \in [x_i; x_{i+1}], \quad (7.37)$$

б) полінома відносно *кінцевої* точки відрізка:

$$S_n^k(x) = \sum_{j=0}^n b_{ij}(x - x_{i+1})^j, \quad x \in [x_i; x_{i+1}], \quad (7.38)$$

Зображення сплайну через усічену ступеневу функцію: усічена ступенева функція для змінної  $x$  і  $N \in \mathbb{N}$  визначається як:

$$x_+^N = \begin{cases} x^N; & x \geq 0; \\ 0; & x < 0; \end{cases} \quad (7.39)$$

Вона є сплайном *ступеня  $n$  дефекту  $1$*  с вузлом в точці  $x=0$ ;

Аналогічно усічена ступенева функція  $(x - x_i)_+^N$ , що зв'язана з вузлами сітки  $\Delta$  (рис. 7.7), є неперервною в точці  $x=x_i$  разом із своїми похідними порядку до  $N-1$ . Отже вона є сплайном ступеня  $n$  дефекту  $1$  з єдиним вузлом  $x_i$ ;

Функції  $x_j$  при  $j = 0, 1, \dots, n$  і  $(x - x_i)_+^m$  при  $m = N - k + 1, \dots, N$ ;  $i = 1, 2, \dots, n-1$  є лінійно незалежними на відрізку  $[a; b]$ , тому їх можна прийняти за базис. Звідси одержуємо зображення сплайна на відрізку  $[a; b]$  у вигляді усічених ступеневих функцій.

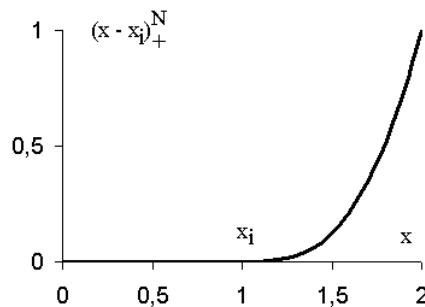


Рисунок 7.7 – Усічена ступенева функція  $(x - x_i)_+^m$  при  $N=3$ ,  $x_i = 1$

$$S(x) \equiv S_n^k(x) = \sum_{r=0}^N c_r (x - x_0)^r + \sum_{i=1}^{n-1} \sum_{j=N-k+1}^N a_{ij} (x - x_i)_+^j; \quad x \in [a; b], \quad (7.40)$$

де  $c_r = \frac{S^{(r)}(x_0)}{r!}$ ;

$a_{ij} = \frac{[S^{(j)}(x_{i+0}) - S^{(j)}(x_i - 0)]}{j!}$  - величини стрибків похідних, розривних у вузлах;

$r = 0, 1, 2, \dots, N$ ;  $i = 1, 2, \dots, n-1$ ;  $j = N-k+1, \dots, N$

Доведено, що таке зображення сплайну є *єдиним*.

Крім розглянутих подань, існують інші зображення сплайнів (через так звані В-сплайни, фундаментальні сплайни).

## 7.6 Інтерполяція кубічними сплайнами

На практиці найбільш широке застосування одержали інтерполяційні *кубічні сплайни*. Це обумовлено тим, що:

- ✓ поліном третього ступеня - це найпростіша крива, що має *точку перегину*, яка забезпечує його гарні інтерполяційні можливості;
- ✓ кубічні сплайни мають на всьому відрізку апроксимації *неперервні похідні до другого порядку*. Така гладкість звичайно виявляється достатньою для більшості завдань;
- ✓ невисокий ступінь полінома в кубічному сплайні *спрощує обчислення* й зменшує обчислювальну погрішність.

Нехай функція  $f(x)$  задана на відрізку  $[a;b]$  у вузлах сітки  $\Delta$ :  $a = x_0 < x_1 < x_2 < \dots < x_n = b$  значеннями  $f(x_i) = y_i$  ( $i = 0, 1, \dots, n$ ). Будемо розглядати сплайни третього ступеня дефекту 1.

Функція  $S_3(x)$  називається *кубічним сплайном*, що інтерполює функцію  $f(x)$  у вузлах сітки  $\Delta$ , якщо:

- ✓ на кожному з відрізків  $[x_i; x_{i+1}]$  ( $i = 0, 1, \dots, n-1$ ) вона є многочленом не вище третього ступеня:

$$S_3(x) \in P_3; \quad (7.41)$$

- ✓  $S(x)$  є неперервною на  $[a;b]$  разом зі своїми похідними до *другого порядку* включно, тобто:

$$S_3(x) \in C^2[a,b]; \quad (7.42)$$

- ✓ у вузлах сітки функція  $S_3(x)$  задовольняє умовам інтерполяції:

$$S_3(x_i) = f(x_i) \quad (i = 0, 1, 2, \dots, n). \quad (7.43)$$

Скористаємося кусочно-поліноміальним зображенням (7.40) кубічного сплайна на відрізку  $(x_i; x_{i+1}]$ :

$$S_3(x) = a_{i0} + a_{i1}(x-x_i) + a_{i2}(x-x_i)^2 + a_{i3}(x-x_i)^3, \quad x \in [x_i; x_{i+1}]. \quad (7.44)$$

Завдання побудови сплайна полягають у знаходженні всіх коефіцієнтів (параметрів)  $a_{i0}, a_{i1}, a_{i2}, a_{i3}$ . При цьому:

- ✓ на кожному частковому відрізку  $[x_i; x_{i+1}]$  кубічний сплайн визначається *чотирма параметрами*  $(a_{i0}, a_{i1}, a_{i2}, a_{i3})$ ;
- ✓ на всьому відрізку  $[a;b]$  число параметрів дорівнює  $4 \cdot n$ .

Для визначення параметрів сплайна потрібно скористатися:

- $(n+1)$ -м умовою інтерполяції (7.44),
- умовами безперервності сплайна у внутрішніх вузлах:

$$S_3(x_{i-0}) = S_3(x_{i+0}) \quad (i = 1, 2, \dots, n-1) \quad (7.45)$$

- умовами неперервності першої й другої похідних сплайна у внутрішніх вузлах:

$$S_3'(x_{i-0}) = S_3'(x_{i+0}) \quad (i = 1, 2, \dots, n-1) \quad (7.46)$$

$$S_3''(x_{i-0}) = S_3''(x_{i+0}) \quad (i = 1, 2, \dots, n-1) \quad (7.47)$$

Загальна кількість зв'язків (умов):

$$(n+1) + 3(n-1) = 4n-2.$$

Для замкненості задачі бракує двох рівнянь зв'язку.

Відсутні для однозначного визначення сплайна дві умови задаються *додатково* у вигляді *обмеження* на значення сплайна або його похідних *на кінцях відрізка*  $[a, b]$ , залежно від фізичного змісту конкретного завдання.

Найбільше часто застосовують наступні *крайові умови*:

а)  $S_3'(a) = f'(a), S_3'(b) = f'(b)$ ;

б)  $S_3''(a) = f''(a), S_3''(b) = f''(b)$ ;

в)  $S_3^{(r)}(a) = S_3^{(r)}(b), r = 1, 2$  (умова періодичності);

г)  $S_3'''(x_{i-0}) = S_3'''(x_{i+0}) \quad (i = 1, 2, \dots, n-1)$  – додаткові умови неперервності, де  $S_3'''(x_{i-0})$  і  $S_3'''(x_{i+0})$  - відповідно, праві й ліва одnobічні похідні третього порядку в точці  $x_i$ .

Крайові умови (в) застосовують для інтерполяції *періодичних функцій* періоду  $T = b-a$ , у тому числі - замкнутих кривих.

Кубічний сплайн, що задовольняє крайовим умовам б), причому  $f'(a) = f'(b) = 0$  (умова нульової кривизни), називається *природним*.

## 7.7 Обчислення параметрів кубічного сплайну

Уведемо наступні позначення:

$$h_i = x_{i+1} - x_i \quad (i = 0, 1, \dots, n-1); \quad (7.48)$$

$$M_i = S_3''(x_i), \quad (7.49)$$

$$m_i = S_3'(x_i) \quad (i = 0, 1, 2, \dots, n). \quad (7.50)$$

Величина  $M_i$  називається *моментом сплайна* в точці  $x_i$ .

Величина  $m_i$  називається *нахилом сплайна* в точці  $x_i$ .

Оскільки  $S_3(x)$  - кубічна функція, то на кожному частковому відрізку друга похідна  $S_3''(x)$  буде лінійною функцією від  $x$ . Тому для  $x \in [x_i, x_{i+1}]$  по формулі лінійної інтерполяції (6.66) запишемо:

$$\begin{aligned} S_3''(x) &= S_3''(x_i) + \frac{S_3''(x_{i+1}) - S_3''(x_i)}{x_{i+1} - x_i} \cdot (x - x_i) = \\ &= M_i + \frac{M_{i+1} - M_i}{h_i} \cdot (x - x_i) \end{aligned} \quad (7.51)$$

Проінтегруємо двічі (7.51):

$$S_3'(x) = S_3'(x_i) + M_i(x - x_i) + \frac{M_{i+1} - M_i}{2h_i}(x - x_i)^2; \quad (7.52)$$

$$S_3(x) = y_i + S_3'(x_i)(x - x_i) + \frac{M_i}{2}(x - x_i)^2 + \frac{M_{i+1} - M_i}{6h_i}(x - x_i)^3. \quad (7.53)$$

Підставимо у (7.53)  $x = x_{i+1}$ :

$$y_{i+1} = y_i + S_3'(x_i)h_i + \frac{M_i}{2}h_i^2 + \frac{M_{i+1} - M_i}{6h_i} \cdot h_i^3, \quad (7.54)$$

звідси:

$$S_3'(x_i) = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}(2M_i + M_{i+1}). \quad (7.55)$$

Підставим вираз (7.55) у формули (7.52), (7.53):

$$S_3(x) = y_i + \left[ \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}(2M_i + M_{i+1}) \right] (x - x_i) + \frac{M_i}{2}(x - x_i)^2 + \frac{M_{i+1} - M_i}{6h_i}(x - x_i)^3, \quad (7.56)$$

$$\begin{aligned} S_3'(x) &= \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}(2M_i + M_{i+1}) + M_i(x - x_i) + \\ &+ \frac{M_{i+1} - M_i}{2h_i}(x - x_i)^2, \end{aligned} \quad (7.57)$$

$$x \in [x_i; x_{i+1}] \quad (i=0, 1, 2, \dots, n-1)$$

Формули (7.56) і (7.57) виражають кубічний сплайн і його першу похідну на відрізку  $[x_i; x_{i+1}]$  через значення функції й моменти у вузлових точках.

З (7.56), (7.57) можна виразити параметри сплайна на кожному інтервалі в кусочно-поліноміальному поданні:

$$\begin{cases} a_{i0} = y_i; \\ a_{i1} = \frac{y_{i+1} - y_i}{h_i}; \\ a_{i2} = \frac{M_i}{2}; \\ a_{i3} = \frac{M_{i+1} - M_i}{6h_i}. \end{cases} \quad (7.58)$$

Для визначення моментів  $M_i, M_{i+1}$  використаємо властивість неперервності першій похідній. Для цього дорівнюємо вираження для першій похідній праворуч і ліворуч від кожної граничної крапки. Після перетворень одержимо  $n-1$  рівняння виду:

$$\alpha_i M_{i-1} + 2M_i + \beta M_{i+1} = d_i, \quad (7.59)$$

де:

$$\alpha_i = \frac{h_{i+1}}{h_i + h_{i-1}}; \quad (7.60)$$

$$\beta_i = \frac{h_{i-1}}{h_i + h_{i-1}} = 1 - \alpha_i; \quad (7.61)$$

$$d_i = \frac{6}{h_i + h_{i-1}} \left[ \frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}} \right]. \quad (7.62)$$

$$(i=1, 2, \dots, n-1).$$

Для визначення моментів  $M_i$  у всіх вузлах необхідно до системи (7.59) додати ще два рівняння, що впливають із крайових умов. Можуть бути задані наступні крайові умови:

а) У крайових точках збігаються перші похідні сплайна й функції:

$$S_3'(a) = f'(a), S_3'(b) = f'(b) \quad (7.63)$$

Додаткові рівняння мають вигляд:

$$2M_0 + M_1 = d_0, \quad (7.64)$$

$$M_{n-1} + 2M_n = d_n, \quad (7.65)$$

де:

$$d_0 = \frac{6}{h_0} \left( \frac{y_1 - y_0}{h_0} - f'(a) \right); \quad (7.66)$$

$$d_n = \frac{6}{h_{n-1}} \left( f'(b) - \frac{y_n - y_{n-1}}{h_{n-1}} \right). \quad (7.67)$$

Ці крайові умови можна використати, коли відбувається сплайн-апроксимація заданої аналітичної функції. *Спосіб непридатний для апроксимації функцій, що задані таблицями значень.*

б) У крайніх точках збігаються другі похідні функції й сплайна:

$$S_3''(a) = f''(a), S_3''(b) = f''(b). \quad (7.68)$$

Оскільки друга похідна – це шуканий момент, це означає, що:

$$M_0 = f''(a), M_n = f''(b). \quad (7.69)$$

Особливості:

- ✓ умова також використається для апроксимації функцій, заданих аналітично;
- ✓ якщо є підстави вважати, що  $f'(a) = f'(b) = 0$  (тобто, що по краях є точка перегину), те цей спосіб можна використати для апроксимації табличних функцій;
- ✓ у цьому способі для знаходження моментів досить розв'язати  $n-2$  рівняння (7.59).

г) Сплайн - *періодичний* з періодом  $(b-a)$ . Значення сплайна і його першої й другої похідних у крайових точках *попарно збігаються*:

$$в) S_3^{(r)}(a) = S_3^{(r)}(b), r = 1, 2. \quad (7.70)$$

Крайові умови:

$$2M_1 + \beta_1 M_2 + \alpha_1 M_n = d_1; \quad (7.71)$$

$$\beta_n M_1 + \alpha_n M_{n-1} + 2M_n = d_n; \quad (7.72)$$

$$M_0 = M_n. \quad (7.73)$$

Цей спосіб використається:

- для апроксимації *періодичних* функцій
- для апроксимації функцій, що утворюють *замкнені криві*.

д) Уводяться *додаткові умови* неперервності - *рівність похідних 3-го порядку* праворуч і ліворуч від всіх внутрішніх точок:

$$S_3'''(x_i-0) = S_3'''(x_i+0) \quad (i = 1, 2, \dots, n-1). \quad (7.74)$$

Крайові умови::

$$(1 + \beta_1) \cdot M_1 + (\beta_1 - \alpha_1) \cdot M_2 = \beta_1 \cdot \alpha_1; \quad (7.75)$$

$$(\alpha_{n-1} - \beta_{n-1}) M_{n-2} + (1 + \alpha_{n-1}) M_{n-1} = \alpha_{n-1} d_{n-1}; \quad (7.76)$$

$$M_0 = \frac{M_1 - \alpha_1 M_2}{\beta_1}; \quad (7.77)$$

$$M_n = \frac{M_{n-1} - \beta_{n-1} \cdot M_{N-2}}{\alpha_{n-1}}. \quad (7.78)$$

У результаті виходить система з  $(n-1)$  лінійних алгебраїчних рівнянь із трьохдіагональною матрицею, з перевагою головної діагоналі. Внаслідок цього матриця системи буде невивродженою, що свідчить про єдине рішення системи. Два моменти, що залишилися розраховують по рівняннях (7.77) і (7.78).

Цей метод може бути використаний для побудови сплайн-апроксимації табличних функцій.

Для розв'язання системи (7.59), (7.75), (7.76) використовують метод прогону, який у цьому випадку є особливо ефективним. Оскільки в матриці коефіцієнтів є діагональна перевага, процес прогону є *стійким щодо помилок округлення*.

При кусочно-поліноміальному зображенні сплайна потрібно містити у пам'яті комп'ютеру:

- вузли сітки  $x_i$ ;
- значення функції  $y_i$ ;
- моменти  $M_i$  ( $i = 0, 1, \dots, n$ ).

Достоїнства сплайнов:

- ✓ інтерполяційні сплайни в ряді завдань забезпечують *мінімально можливу погрішність* наближення на даному класі функцій серед всіх многочленів фіксованого ступеня;
- ✓ серед всіх функцій:
  - що інтерполують  $f(x)$  у заданих точках  $i$
  - що мають інтегрувальну із квадратом другу похідну на  $[a, b]$ , кубічний сплайн є *єдиною функцією*, що володіє властивістю мінімальної кривизни, тобто, є *самою гладкою* з функцій, що інтерполують  $f(x)$ ;

в) властивість локальності: зміна значення функції у вузлі  $x_i$  мало впливає на значення сплайна і його похідних у точках, досить видалених від цього вузла.

### Приклад 7.5

Побудувати сплайн і графік сплайн-апроксимації для табличної функції:

$x_i$	0	1	3	3,5	5	6
$y_i$	1	2	6	6,5	6	3,5

Розв'язання

а) розраховуємо моменти сплайна. За крайові використаємо умови (7.77). Робочий аркуш проекту представлений на скріншоті (рис. 7.8);

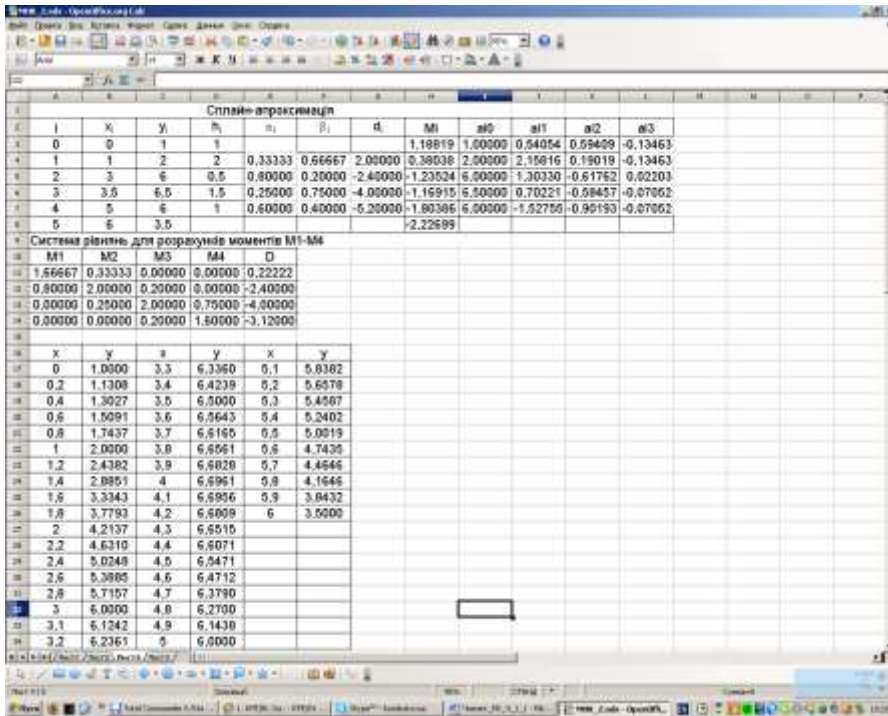


Рисунок 7.8 – Сплайн-апроксимація

- б) вводимо вихідні дані - значення  $x_i, y_i$ ;
- в) у комірках D3-D7 розраховуємо значення  $h_i$ ; формула комірки D3: =B4-B3. Далі - копіювання на весь діапазон;
- г) в комірках E4 - E7 розраховуємо значення параметрів  $\alpha$  по формулі (7.60). Формула комірки E4: =D3/(D3+D4). Далі - копіювання на весь діапазон;
- д) в комірках F4 - F7 розраховуємо значення параметрів  $\beta$  по формулі (7.61). Формула комірки F4: =1-E4. Далі - копіювання на весь діапазон;
- е) в комірках G4:G7 розраховуємо значення параметрів  $d$  по формулі (7.62) Формула комірки G4: =6/(D4+D3)\*((C5-C4)/D4-(C4-C3)/D3). Далі - копіювання на весь діапазон;
- ж) в комірках A11:D14 створюємо систему рівнянь для розрахунку моментів  $M_1 - M_4$  по формулах (7.59), (7.75), (7.76). Формули ненульових комірок наведені в табл. 7.5
- з) у масиві комірок H4:H7 будуємо розв'язок системи рівнянь для  $M_1 - M_4$ . Формула масиву: =МУМНОЖ(МОБР(A11:D14);E11:E14);

Таблиця 7.5 – Формули ненульових комірок для системи рівнянь сплайну (рис. 7.8)

Комірка	Рівняння ненульового елемента	Формула комірки
A11	$a_{11} = 1 + \beta_1 \cdot$	=1+F4

B11	$a_{12} = \beta_1 - \alpha_1$	=F4-E4
E11	$b_1 = \beta_1 \cdot \alpha_1$	=E4*F4
A12	$a_{21} = \alpha_2$	=E5
B12	$a_{22} = 2$	=2
C12	$a_{23} = \beta_2$	=F5
E12	$b_2 = d_2$	=G5
B13	$a_{32} = \alpha_3$	=E6
C13	$a_{33} = 2$	=2
D13	$a_{34} = \beta_3$	=F6
E13	$b_3 = d_3$	=G6
C14	$a_{43} = \alpha_4 - \beta_4$	=E7-F7
D14	$a_{44} = 1 + \alpha_4$	=1+E7
E14	$b_4 = \beta_4 \cdot d_4$	=E7*G7

і) в комірці Н3 розраховуємо значення моменту  $M_0$  по формулі (7.66). Формула комірки: =(H4-E4\*N5)/F4;

к) в комірці Н8 розраховуємо момент  $M5$  по формулі (7.67). Формула комірки: =(H7-F7\*N6)/E7;

л) в комірці І3:І7 розраховуємо коефіцієнти  $a_0$  для кожного з діапазонів. Розрахунок виробляється по формулі (7.58). Формула комірки І3: = С3. Далі копіюємо на весь діапазон;

м) в комірках І3:І7 розраховуємо коефіцієнти  $a_1$  для кожного з діапазонів. Розрахунок виробляється по формулі (7.58). Формула комірки І3: =(С4-С3)/D3-D3/6\*(2\*N3+H4). Далі копіюємо на весь діапазон;

н) в комірках К3:К7 розраховуємо коефіцієнти  $a_2$  для кожного з діапазонів. Розрахунок виробляється по формулі (7.48). Формула комірки К3: =Н3/2. Далі копіюємо на весь діапазон;

о) в комірках L3:L7 розраховуємо коефіцієнти  $a_3$  для кожного з діапазонів. Розрахунок виробляється по формулі (7.48). Формула комірки L3: =(H4-N3)/6/D3. Далі копіюємо на весь діапазон;

п) далі розраховуємо значення сплайна в кожному з діапазонів. У діапазоні значень  $x$  від 0 до 1 розрахунок- по формулі комірки В17 :

$$=I\$3+J\$3*(A17-\$A\$17)+K\$3*(A17-\$A\$17)^2+L\$3*(A17-\$A\$17)^3$$

Далі копіюємо на діапазон до В22. Аналогічно робимо розрахунок для інших діапазонів;

р) по отриманій таблиці будемо графік сплайна (рис. 7.9).

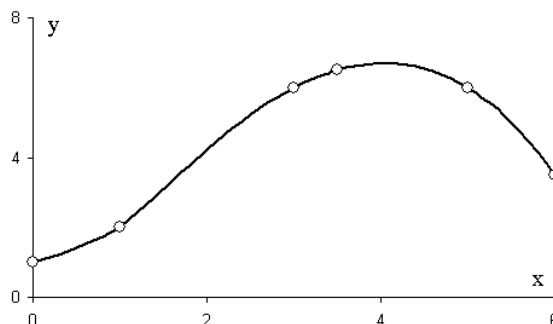


Рисунок 7.0– Графік сплайн-апроксимації

Метод сплайнов має значне число модифікацій, у тому числі - метод згладжених сплайнів, симбіоз методу сплайнів і найменших квадратів, що дозволяє будувати гладкі апроксимації складних функцій за даними з погрішностями.

Значно зручніше будувати сплайни у системі Scilab, де наявні спеціалізовані функції:

В Scilab наявні можливості:

- ✓ інтерполяції лінійним сплайном (кусково-лінійна апроксимація, див. [7.3](#))
- ✓ інтерполяції кубічними сплайнами

В Scilab для побудови кусково-лінійної інтерполяції (лінійного сплайну) використовується функція:

$$u = \text{interpln}(z, x);$$

де  $z$  – матриця вхідних даних, що одержується сполученням векторів  $x$  і  $y$ ,  
 $x$  – вектор абсцис,

$n$  – вектор ординат (значень функції і лінійного сплайну у точках  $x$ )

Для побудови кубічного сплайну використовують двоетапну процедуру:

- ✓ на першому кроці розраховуються *моменти сплайну* за допомогою функції:

$$d = \text{splin}(x, y),$$

де  $x$  – строго зростаючий вектор, що містить, як мінімум, дві компоненти,

$y$  – вектор того ж формату, що й  $x$

- ✓ на другому кроці розраховують значення інтерполяційного сплайну у точці:

$$u = \text{interp}(t, x, y, d),$$

де  $t$  - вектор абсцис, що задається,

$x, y$  – координати точок сплайну.

### Приклад 7.6

За даними, що наведені нижче, побудувати лінійний і кубічний сплайни та їх графіки.

x	0	1	2	3	4	5	6	7	8	9	10
y	1	1,5	2,7	4,5	5,9	6	5	5,3	4	2	1,1

обудувати таблицю значень обох сплайнів в діапазоні від 4 до 6 з кроком 0.2.

Розв'язання

Скрипт для проведення розрахунків наведено у лістинг 7.6

Лістинг 7.6

```
x=[0,1,2,3,4,5,6,7,8,9,10];
y=[1,1.5,2.7,4.5,5.9,6,5,5.3,4,2,1.1];
z=[x;y];
g=interpLn(z,x);
plot(x,g,'ks-');
coef=splin(x,y);
t=0:0.1:10;
u1=interp(t,x,y,coef);
plot(t,u1,'b')
h=0.2;
for i=1:11
    tt(i)=4+(i-1)*h;
    z1(i)=interpLn(z,tt(i));
    z2(i)=interp(tt(i),x,y,coef);
end
disp('Значення сплайну')
disp('x          лін          куб')
[tt,z1,z2]
```

Результати роботи програми наведені у протоколі 7.6 і на графіку (рис. 7.10).

Протокол 7.6

Значення сплайну

x лін куб

ans =

```
4.    5.9    5.9
4.2   5.92   6.0505411
4.4   5.94   6.1425909
4.6   5.96   6.1693701
4.8   5.98   6.1240996
```

5.	6.	6.
5.2	5.8	5.7986138
5.4	5.6	5.5547701
5.6	5.4	5.3116194
5.8	5.2	5.1123125
6.	5.	5.

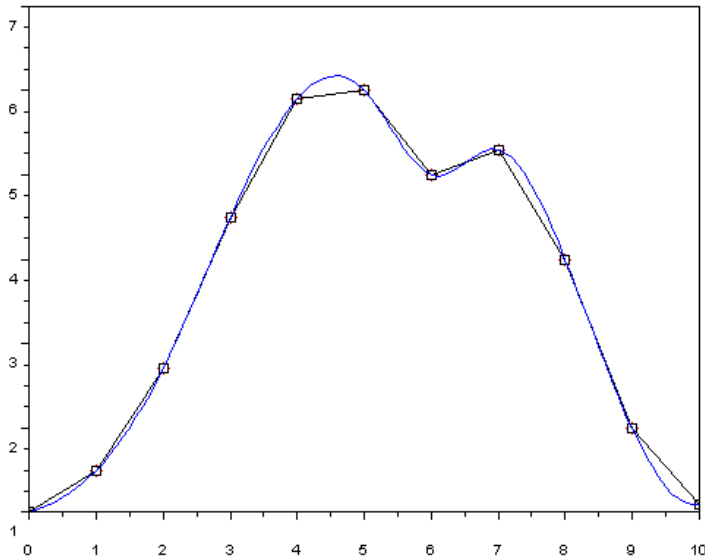


Рисунок 7.10 – Графіки лінійного і кубічного сплайнів (до прикладу 7.6)

### Контрольні питання

- 1 У чому полягають позитивні риси апроксимації результатів експериментів або спостережень методом найменших квадратів?
- 2 Як графічно розрізняються характер функцій, що апроксимують, для поліноміальної інтерполяції і МНК?
- 3 Алгоритм побудови найкращого середнє квадратичного наближення.
- 4 Зміст поняття «нормальна система» для апроксимації МНК за допомогою узагальнених поліномів
- 5 У якому випадку параметри апроксимації МНК визначаються найбільш просто?
- 6 Зміст поняття «узагальнені коефіцієнти Фур'є»
- 7 У чому є переваги ортогонального базис для розв'язання задач МНК?
- 8 Які властивості має ортогональний базис МНК
- 9 Як пов'язані між собою збіжність в середньому і повнота системи базисних функцій для задач МНК?
- 10 Які допущення використовуються при точковій апроксимації методом найменших квадратів?
- 11 У чому полягає принцип найменших квадратів?

- 12 Як співвідносяться між собою вимірність задачі МНК і кількість точок для апроксимації?
- 13 Які дані називаються рівноточними?
- 14 Яку вагу використовують для системи рівноточних спостережень?
- 15 Як можна розрахувати вагу для нерівноточних спостережень?
- 16 Як можна оцінити якість апроксимації МНК?
- 17 Які проблеми виникають при апроксимації поліномами високого ступеня на великому масиві даних?
- 18 Який є можливий шлях усунення використання поліномів високого ступеня?
- 19 Принципи побудови кусково-поліноміальної апроксимації МНК.
- 20 Що являє собою сплайн?
- 21 Переваги апарату сплайнів у порівнянні з інтерполяцією по Лагранжу
- 22 Математичне визначення поняття сплайна
- 23 Що визначає дефект сплайна?
- 24 Для сплайнів якого ступеня вузли інтерполяції і вузли сплайна збігаються?
- 25 Записати умови інтерполяції для сплайну.
- 26 Сутність кусково-поліноміального зображення сплайна
- 27 Сутність зображення сплайну через усічену ступеневу функцію.
- 28 Чим обумовлено популярність використання на практиці кубічних сплайнів?
- 29 Яка функція називається кубічним сплайном?
- 30 Скільки параметрів треба задати для визначення сплайна з 5 вузлами?
- 31 Які умови використовуються для визначення параметрів сплайна?
- 32 Яким чином забезпечується замкненість задачі сплайн-апроксимації?
- 33 Які види крайових умов використовуються для визначення сплайнів?
- 34 Які крайові умови використовують для сплайн апроксимації функцій, що задані таблицею?
- 35 Які величини називаються моментом і нахилом сплайну?
- 36 У чому полягає суть визначення параметрів сплайну на практиці?
- 37 Якого виду системи рівнянь одержують для визначення параметрів сплайну? У чому їх позитивні риси?
- 38 Яку інформацію треба зберігати у пам'яті комп'ютера для кусково-поліноміального зображення сплайна і розрахунків його значень?
- 39 У чому зміст властивості локальності сплайну?
- 40 Як розрахувати сплайн в середовищі Scilab?

### **Завдання для самостійної роботи**

Для [завдання 6](#) побудувати:

- а) середнє квадратичну апроксимацію поліномами 1 – 4 порядку,
- б) середнє квадратичну апроксимацію тригонометричними поліномами 1-4 порядку. Побудувати графіки, виявити, які з апроксимації мають середнє квадратичне відхилення не більше 0,02,
- в) побудувати сплайн-апроксимацію, користуючись:

- 1) уся сукупність точок,
- 2) парні точки, з подальшим розрахунком значень сплайну в непарних точках. Знайти суму квадратів різниць сплайну і заданої функції у непарних точках. Порівняти з величиною суми вквadratів відхилень для поліноміальної та тригонометричної апроксимації.

## Додаток А. Робота з електронним підручником у середовищі Foxit Reader

### A1 Загальні відомості про програму

Програма Foxit Reader є програмним продуктом, що вільно розповсюджується компанією Foxit Corporation (США). Вона може бути безкоштовно скачана з сайту: <http://www.foxitsoftware.com/pdf/reader/>. Програма призначена для читання файлів з розширенням .pdf. У порівнянні з програми серії Acrobat reader фірми Adobe, що також безкоштовно розповсюджуються через Інтернет, програма Foxit Reader має такі переваги:

- ✓ вона «важить» значен менше, ніж Acrobat Reader (всього 5-6 Мб) і працює значно скоріше,
- ✓ сервісні можливості програми значно вище, ніж у Acrobat reader і наближаються до можливостей, як рідера, ліцензійного пакету Acrobat Professional,
- ✓ програма не вимагає установки в реєстрі і може працювати при запуску не тільки з жорсткого диску, але й з флеш-карти,
- ✓ програма має інтуїтивно ясний інтерфейс, що є подібним для Adobe Acrobat і дозволяє бістро засвоїти роботу. Наявні коментарі, що спливають, а також мануал на англійській мові.

Робота в середовищі Foxit Reader практично не відрізняється від роботи в середовищі Acrobat Reader та Acrobat Profesional. Тому головну увагу нижче буде надано можливостям програми для культурної роботи з тестом електронної книги.

### A2 Основні прийоми роботи з науковим текстом

Культурна робота з класичним паперовим науковим текстом – це робота з олівцем у руці. Читач, у процесі вивчення тексту, може робити олівцем різноманітні помітки, що виражають його емоції, почуття під час читання. Як правило, на полях книги ставляться такі типові позначки:

- ✓ *Знак запитання*, що вказує на неясність у тексті, над якою слід поміркувати,
- ✓ *Знак оклику*, що вказує на особливо цікаве або важливе місце у тексті,
- ✓ *Знак замітки*, частіше за все NB (nota bene –лат «звернути увагу»), або інші.

На полях книги, або на паперових вкладках прийнято писати коментарі до того чи іншого фрагменту тексту. При цьому фрагмент, до якого відноситься коментар, підкреслюють (порядково або збоку). Це все полегшує читачеві розуміння матеріалу, відокремити найбільш суттєву інформацію, породжує думки, що можуть бути використані у подальшій роботі.

Ті ж самі прийоми можуть бути використані при роботі з електронними книгами, зокрема з даним посібником. Але можливості електронного видання часто є значно більш широкими у порівнянні з паперовим.

### A3 Розмітка тексту у Foxit Reader

У форматі .pdf текст може бути одержаний за двома методами:

- ✓ шляхом компіляції текстового файлу, створеного у текстовому редакторі. у цьому випадку, разом з графічним образом, міститься й попередній текстовий шар,
- ✓ шляхом влучення сканованого зображення тексту.

У обох випадках для розмітки тексту можна використовувати такі *універсальні засоби*:

- створення закладок (bookmarks),
- створення заміток (notes),
- створення коментарів, що спливають,
- виділення тексту графічними засобами,
- виділення тексту і створення позначок електронними олівцем та гумкою

У *першому випадку* наявність текстового шару дає додаткові можливості роботи з текстом:

- ✓ пошук по ключовим словам,
- ✓ виділення кольором тексту шляхом:
  - підкреслення,
  - закреслення,
  - забарвлення рядку або окремого слова,

Розглянемо кожний зі засобів.

### **А3.1 Створення і використання закладок**

Закладки грають роль гіперпосилань на сторінки тексту. Вони можуть бути створені:

- ✓ автоматично при компіляції текстового файлу в разі наявності структурування назв розділів і підрозділів (цей прийом був використаний у цьому підручнику);
- ✓ штучно з використанням засобу «закладки», з використанням кнопки:



Кнопка знаходиться на панелі на лівому краю робочого екрану.

Після натиснення на цю кнопку відкривається вікно закладок у лівій частині екрану. (рис. А.1). Якщо закладки не встановлені – це вікно є пустим. Якщо наявні закладки – у віконці можна побачити їх перелік (рис. А1)

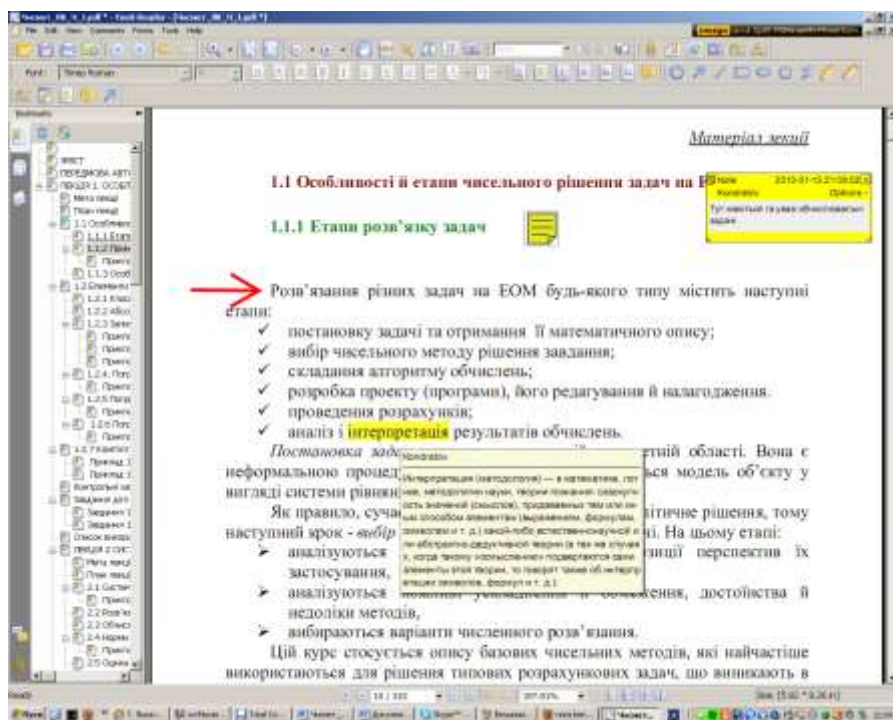


Рисунок А1 – Вікно закладок і розмітка .pdf-файла у Foxit reader

При наведенні на закладку курсору спливає повна назва закладки.

Для *переходу* на сторінку, що пов'язана з закладкою, треба:

- навести на закладку курсор;
- клацнути лівою кнопкою миши.

Для встановлення *нової закладки* необхідно:

- ✓ відилити закладку, після якої ви бажаєте встановити нову закладку. Якщо екран пустий – вона встановлюється автоматично;
- ✓ виставити на екрані сторінку, на яку треба переходити,
- ✓ навести на кнопку з зображенням закладки у верхній частині вікна закладок (при цьому сплине надпис «Click to add a new bookmark» («Клацнута для додання нової закладки»);
- ✓ клацнути лівою кнопкою миши
- ✓ у виділеному рядку, що з'явився, увести надпис закладки (наприклад назву підрозділу, на яку треба виходити),
- ✓ натиснути кнопку <Enter>

Для *редагування* надпису:

- ✓ Навести курсор на закладку,
- ✓ Клацнути праву кнопку миши. При цьому з'являється меню редагування (рис. А2)
- ✓ Вибрати необхідну кнопку і виконати редагування.

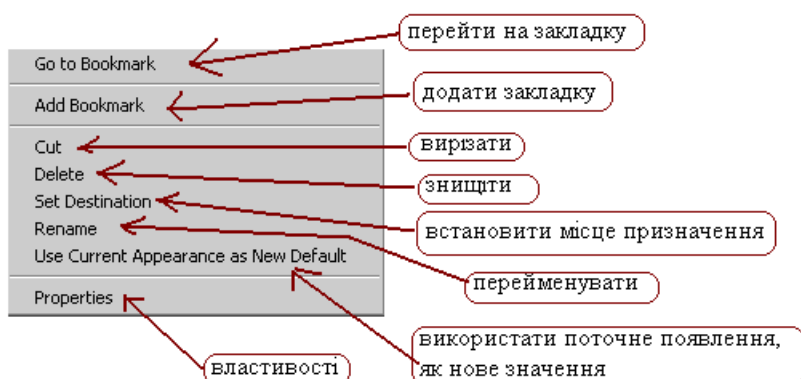


Рисунок А2 – Меню редагування закладки

Зокрема у меню кнопки «Properties» можна змінити колер та стиль надпису.

### А3.2 Створення і використання заміток

Для створення заміток треба:

- ✓ навести курсор на вільне місце на аркуші біля тексту, стосовно якого Ви бажаєте написати коментар або замітку,
- ✓ відкрити панель інструментів: View – Toolbars – Favorite Tools
- ✓ натиснути на кнопку з позначкою:



при цьому на екрані з'явиться ця позначка і відкриється віконце для коментаря. В ньому можна надрукувати коментар до тексту,

Створений коментар можна:

- ✓ залишити на аркуші, як це показано на рис. А.1
- ✓ закрити звичайним чином. У цьому випадку коментар *спливає при наведення курсора* на позначку замітки.

Коментарі до тексту та замітки можна створити також при використанні інших позначок для виділення тексту:

- ✓ стрілки, що знаходиться на панелі Favorite Tools
- ✓ інструментів, що знаходяться на панелі Drawing Markup Tools:
  - прямокутного виділення,
  - стрілки,
  - прямої лінії,
  - ламаної лінії,
  - олівця

В усіх випадках для відкриття коментарю треба:

- ✓ провести лінію або відмітити текст прямокутником,
- ✓ навести на лінію курсор і натиснути праву кнопку миши,
- ✓ у меню, що відкриється, натиснути кнопку Open Popur Note, При цьому відкриється віконце коментарю.

За допомогою означеного меню можна міняти властивості об'єктів, нариснув кнопку Properties. Зокрема можна поміняти колер лінії і забарвити область тексту, як це зроблено на рис. А1. Подібно описаному вище створюються для кожного об'єкту спливаючі коментарі: для цього достатньо вимкнути коментар.

### А3.3 Використання олівця і гумки

Найбільш універсальним інструментом для виділення тексту є електронний олівець. Він знаходиться на панелі інструментів Drawing Markup Tools і має позначку:



За допомогою олівця можна підкреслювати текст, помічати його різними позначками (рис. А3). При цьому у кожному випадку можна створити коментарі.

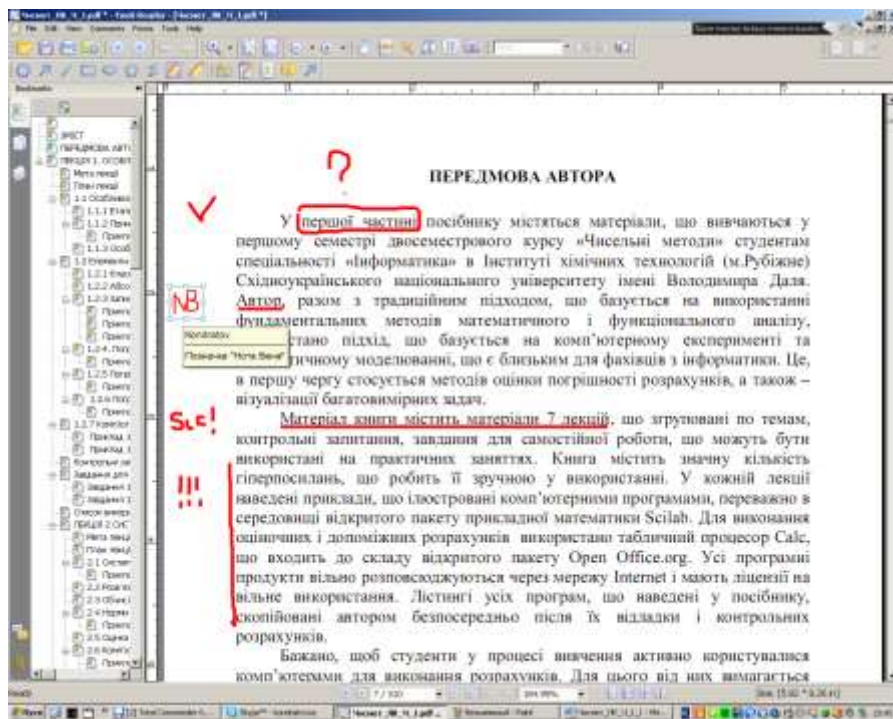


Рисунок А.3 – Використання електронного олівця для розмітки

Для видалення позначок можна використовувати електронну гумку, що знаходиться та той же самої панелі, праворуч від олівця. При використанні цього інструменту видаляється тільки позначка, що зроблена олівцем, на текст або рисунки основного документу гумка *не* діє.

Прийоми, що наведені, на думку автору, дозволяють більш ефективно вивчати матеріал і більш глибоко розуміти його. Це є особливо важним для математичних

дисциплін. Недаром виданий американський математик Р.Хемминг закінчив свою книгу «Чисельні методи» таким висловлюванням:  
«У математиці головне – не число, а розуміння»!

## ЕЛЕКТРОННЕ НАВЧАЛЬНЕ ВИДАННЯ

*Кондратов Сергій Олексійович. Іванов Вітаний Геннадійович*

Чисельні методи. Частина 1. Чисельні методи алгебри. Методи наближень:  
Навчальний посібник для студентів спеціальності «Інформатика». – Рубіжне:  
Інститут хімічних технологій СНУ ім. В.Даля, 2025 р. - 312 с.

Навчальне видання

**Навчальний посібник**

**з дисципліни**

**«Чисельні методи. Частина 1. Чисельні методи алгебри. Методи наближень»**

для здобувачів першого (бакалаврського) рівня освіти

за спеціальністю

(для студентів спеціальностей

121 «Інженерія програмного забезпечення»,

126 «Інформаційні технології та системи»)

(Електронне видання

Укладачі:

*Кондратов Сергій Олексійович.*

*Іванов Віталій Геннадійович*

Оригінал-макет В. Г. Іванов

Підписано до друку \_\_. \_\_. 202\_\_.

Формат 60x84 1/16. Папір типогр. Гарнітура Times.

Друк офсетний. Умов. друк. арк. \_\_. Обл.-вид. арк. \_\_.

Тираж \_\_ екз. Вид. № \_\_. Замов. № \_\_. Ціна договірна.

Видавництво Східноукраїнського національного університету  
імені Володимира Даля

Свідоцтво про реєстрацію: серія ДК № 1620 від 18.12.03 р.

Адреса університета: вул. Іоанна Павла II, 17

м. Київ, 01042, Україна

e-mail: [vidavnictvosnu.ua@gmail.com](mailto:vidavnictvosnu.ua@gmail.com)