

Міністерство науки і освіти України
Тернопільський державний економічний університет

На правах рукопису

КОВАЛЬЧУК Ольга Ярославівна

УДК 519.6:513.3

**АЛГОРТМИ ДЛЯ СИСТЕМ З ТЕПЛЦЕВИМИ λ -МАТРИЦЯМИ
ТА ЇХ ЗАСТОСУВАННЯ**

01.05.02 – математичне моделювання та обчислювальні методи

Дисертація
на здобуття наукового ступеня
кандидата фізико-математичних наук

Науковий керівник
Недашковський Микола Олександрович,
доктор фізико-математичних наук,
професор

Тернопіль – 2005

Зміст

Вступ	4
Розділ 1 Теоретичні основи тепліцевих матриць	11
1.1. Огляд літератури.....	11
1.2. Класи задач розв’язування СЛАР з тепліцевими (ганкелевими) λ -матрицями.....	17
1.3. Деякі властивості тепліцевих матриць.....	23
Розділ 2 Алгоритми розв’язування систем лінійних алгебраїчних рівнянь з тепліцевими λ-матрицями для послідовних моделей обчислень, оцінки їх арактеристк	34
2.1. Алгоритми знаходження матриці, оберненої до тепліцевої λ -матриці з алгебраїчними поліноміальними елементами.....	34
2.2. Алгоритми обчислення розв’язків СЛАР з тепліцевими λ -матрицями з алгебраїчними поліноміальними елементами.....	46
2.3. Алгоритм розв’язування СЛАР з тепліцевими λ -матрицями з тригонометричними поліноміальними елементами.....	74
2.4. Розв’язування СЛАР з ганкелевими λ -матрицями з алгебраїчними поліноміальними елементами.....	81
2.5. Послідовні і паралельні моделі обчислень.....	87
2.6 Оцінки характеристик алгоритмів розв’язування СЛАР з тепліцевими λ -матрицями.....	89
Розділ 3 Алгоритми розв’язування систем лінійних алгебраїчних рівнянь з блочно–тепліцевими λ-матрицями для послідовних і паралельних моделей обчислень	91
3.1. Алгоритми розв’язування СЛАР з блочно-тепліцевими λ -матрицями.....	91
3.2. Вкористання додаткової пам’яті і паралельні алгоритми для розв’язування СЛАР з тепліцевими λ -матрицями.....	103

3.2.1. Використання додаткової пам'яті при розв'язуванні СЛАР з блочно-тепліцевими λ -матрицями.....	103
3.2.2. Паралельні моделі для реалізації алгоритмів розв'язування СЛАР з тепліцевими λ -матрицями.....	109
3.3. Алгоритми розв'язування СЛАР з тепліцевими λ -матрицями гіллястими ланцюговими дробами	117
Розділ 4 Застосування тепліцевих λ-матриць в математичному моделюванні.....	124
4.1. Розв'язування інтегральних рівнянь в аеродинамічних дослідженнях з допомогою тепліцевих матриць.....	124
4.1.1. Визначення частот коливань.....	132
4.1.2. Змінні коливання консольного крила сталого перерізу.....	139
4.1.3. Нестационарна задача.....	134
4.2. Про збіжність обчислених власних значень.....	135
4.3. Гідродинамічні задачі.....	139
4.4 Дослідження стійкості розв'язків математичної моделі імунного захисту.....	142
4.5. Дослідження щільності кісткової тканини.....	150
ВИСНОВКИ.....	156
Список використаних джерел.....	158
Додаток	173

ВСТУП

Актуальність теми. При дослідженнях математичних моделей з використанням методів лінійної алгебри традиційним і завжди актуальним є етап знаходження розв'язків систем лінійних алгебраїчних рівнянь (СЛАР). Ще зовсім недавно ЕОМ використовувалися в основному для розв'язування різних класів таких задач з початковими даними чисельного характеру.

Останнім часом при проведенні досліджень за допомогою математичного моделювання у фізиці, електротехніці та інших областях науки і техніки виник також стійкий інтерес до використання комп'ютерів для символічних перетворень та одержання аналітичних розв'язків, хоча й у "чистій математиці" завжди гостро відчувалась необхідність у використанні ЕОМ для виконання найбільш трудомісткої чорнової роботи. Однією з перших удалих спроб реалізації такого підходу було створення мови "Аналітик" для машин серії "МИР" в Інституті кібернетики НАН України ще в 60-х роках ХХ ст.

Сьогодні існують й успішно розвиваються декілька напрямів і концепцій для виконання символічних перетворень. З комп'ютерних систем універсального характеру найбільш відомими є Reduce, muMATH, MATHEMATICA, MAPLE, MatLab, MathCad і DERIVE. Їх можна використовувати для розв'язування різних задач комп'ютерної алгебри, у тому числі і для знаходження розв'язків систем лінійних алгебраїчних рівнянь. Але цей розділ у названих комп'ютерних системах ще не настільки добре розвинутий, як відповідні методи для числових систем.

Так, в останні два десятиліття поряд з розвитком традиційних досліджень методів і розробкою алгоритмів розв'язування СЛАР з матрицями, елементами яких є числа, значний розвиток одержали дослідження для символічних перетворень і знаходження аналітичних розв'язків СЛАР з λ -матрицями, елементами яких є алгебраїчні та тригонометричні поліноми. Подібні задачі зустрічаються у математичному

моделюванні та дослідженні функціонування електронних схем, систем масового обслуговування з очікуванням, перехідних процесах у схемотехніці та інших прикладних задачах.

Такими дослідженнями займались Д.К.Фадєєв, В.М.Фадєєва, С.А.Абрамов, В.М.Кублановська, М.О.Недашковський, Г.І.Малашонок, J.Lipson, J.Smit, D.Mazykelli, P.T.Moenk, J.Carter, S.Cabay, B.Domzy, E.H.Bareiss, M.T.MacClellan та інші автори.

Серед вказаних досліджень не одержали належного розвитку СЛАР з теплицевими λ -матрицями, тобто матрицями, в яких на кожній діагоналі, розташованій паралельно до головної, стоять однакові елементи. Такі матриці повністю визначаються елементами першого стовпця та першого рядка.

СЛАР з теплицевими матрицями, елементами яких є дійсні числа, зустрічаються в багатьох прикладних задачах і математичних проблемах. Так, наприклад, в обчислювальних задачах прикладної електродинаміки, оптики, акустики, обробки зображень, автоматичного регулювання, побудови авторегресійних фільтрів та в інших прикладних проблемах відповідні задачі часто зводяться до розв'язування СЛАР з теплицевими матрицями. Континуальними аналогами СЛАР, в яких матриці коефіцієнтів теплицеві, є інтегральні рівняння, ядра яких залежать від різниці аргументів.

З теплицевими тісно пов'язаний ще один клас матриць – ганкелеві. Простою перестановкою стовпців (стрічок) в оберненому порядку теплицеву (ганкелеву) матрицю можна перетворити в дві різні ганкелеві (теплицеві) матриці. Тому для розв'язування СЛАР як з ганкелевими, так і з теплицевими матрицями можна використовувати одні і ті ж обчислювальні схеми.

Інтерес до теорії теплицевих і ганкелевих матриць, елементами яких є дійсні числа, і розв'язування СЛАР з цими матрицями завдяки їх широкому застосуванню не послаблюється ще з початку двадцятого століття. Найбільш вагомий внесок у розвиток даної теорії зробили W.Trench, R.Blahut, В.В.Воеводін, Є.Є.Тиртишніков, І.С.Юхвідов та інші.

Оскільки СЛАР з тепліцевими λ -матрицями є недостатньо вивченими, актуальною задачею є розробка ефективних алгоритмів їх розв'язування на основі використання специфічних властивостей цих матриць.

Поряд з цим задача обертання матриці – відшукування для заданої неособливої матриці A ($\det A \neq 0$) оберненої матриці A^{-1} , яку потрібно при цьому розв'язувати, є однією з центральних і важких задач теорії матриць. Важливість її розв'язування хоча б для окремих класів матриць як в теоретичному плані, так і в прикладних задачах (розв'язування систем лінійних алгебраїчних рівнянь) не викликає сумнівів, тим більше, що проблема в багатьох її аспектах потребує поглибленого дослідження.

Завдяки специфічним властивостям тепліцевих (ганкелевих) матриць можлива побудова економічних обчислювальних схем відшукування для них обернених матриць. Такі алгоритми розроблені для тепліцевих матриць з дійсними та комплексними елементами. Доцільним є пошук аналогічних алгоритмів для тепліцевих λ -матриць.

Ці дослідження є важливими для розв'язування як вказаних вище прикладних задач, так і оптимізаційних задач, в яких цільова функція залежить від розв'язків СЛАР з тепліцевими λ -матрицями, тобто є функцією параметра λ . Такими є, наприклад, задачі знаходження розв'язків СЛАР із заданими властивостями: мінімальний за нормою розв'язок, мінімально віддалений від заданого розв'язку тощо.

Дана робота присвячена побудові ефективних алгоритмів розв'язування тепліцевих та ганкелевих СЛАР з λ -матрицями і виконана впродовж 2000–2005 років на кафедрі автоматизованих систем і програмування Тернопільського державного економічного університету (м.Тернопіль).

Зв'язок роботи з науковими програмами, планами, темами. Базовою для даних дисертаційних досліджень стала комплексна науково-дослідна робота Центральної науково-дослідної лабораторії Тернопільського

державного медичного університету імені І.Я.Горбачевського на тему “Структурно-функціональне обґрунтування магнітолазерного впливу для профілактики і корекції уражень товстої кишки” (номер державної реєстрації 0101U001312), яка виконувалася на замовлення та за фінансової підтримки МОЗ України.

Мета і завдання дослідження. Метою даної дисертаційної роботи є побудова та обґрунтування ефективних алгоритмів розв’язування СЛАР з тепліцевими та ганкелевими λ -матрицями. Для досягнення мети були поставлені такі задачі:

- побудова ефективних алгоритмів розв’язування СЛАР з тепліцевими λ -матрицями, елементами яких є алгебраїчні поліноми;
- розробка швидких алгоритмів розв’язування СЛАР з блочно-тепліцевими λ -матрицями;
- розробка ефективних алгоритмів розв’язування тепліцевих та ганкелевих СЛАР з тригонометричними λ -матрицями;
- побудова моделей для реалізації алгоритмів розв’язування СЛАР з блочно–тепліцевими λ -матрицями з поліноміальними елементами на багатопроцесорних обчислювальних системах;
- аналіз похибок заокруглення, які виникають при комп’ютерній реалізації побудованих алгоритмів.

Методи дослідження. Знаходження розв’язків систем лінійних алгебраїчних рівнянь з тепліцевими (ганкелевими) λ -матрицями базується на теорії тепліцевих матриць, швидкому перетворенні Фур’є, теорії ланцюгових дробів. Для відшукування розв’язків таких СЛАР розроблено поліноміальні модифікації відповідних алгоритмів лінійної алгебри для числових матриць з дійсними та комплексними елементами: Тренча, Левінсона, Тиртишнікова.

Наукова новизна одержаних результатів, що виносяться до захисту:

- на основі економічних схем знаходження оберненої матриці розроблено алгоритми, які узагальнюють числові методи лінійної алгебри для тепліцевих (ганкелевих) матриць з дійсними та комплексними

елементами, на випадок СЛАР з тепліцевими (ганкелевими) λ -матрицями. Зроблено аналіз оцінок (зокрема і похибок заокруглення) створених алгоритмів;

- розроблено схему розв’язування СЛАР з блочно-тепліцевими (ганкелевими) матрицями. Для побудованих алгоритмів досліджено їх характеристики;

- побудовано моделі для реалізації алгоритмів розв’язування СЛАР з блочно–тепліцевими λ -матрицями з поліноміальними елементами на багатопроцесорних обчислювальних системах;

- проведено аналіз оцінок (пам’яті комп’ютера, числа арифметичних операцій, комп’ютерного часу, похибок розв’язку СЛАР з тепліцевими λ -матрицями) для побудованих алгоритмів та проведено їх аналіз;

- для математичних моделей імунного захисту організму та щільності кісткової тканини досліджено стійкість розв’язків. Комп’ютерні алгоритми, розроблені на основі цих моделей, впроваджено в навчальний процес у Тернопільському державному медичному університеті імені І.Я.Горбачевського.

Обґрунтованість і достовірність. Достовірність підтверджується строгим доведенням основних результатів, яке ґрунтується на використанні загальноприйнятих математичних перетворень та апарату лінійної алгебри, теорії тепліцевих і ганкелевих матриць, теорії ланцюгових дробів тощо.

Теоретичне обґрунтування розроблених алгоритмів підтверджується результатами розв’язування числових експериментів.

Наукове значення роботи. Поряд із важливістю досліджень стосовно розв’язків тепліцевих (ганкелевих) СЛАР з λ -матрицями отримані результати можуть бути застосовані при розв’язуванні задач оптимізації, наприклад, обчисленні найменшого за нормою або найменш віддаленого від задані точки розв’язку СЛАР з тепліцевою λ -матрицею, а також деяких задач лінійного параметричного програмування.

Практичне значення отриманих результатів.

Для математичних моделей імунного захисту організму та щільності кісткової тканини досліджено стійкість розв'язків, і на їх основі розроблено алгоритми, які впроваджено в навчальний процес у Тернопільському державному медичному університеті імені І.Я.Горбачевського.

Особистий внесок. Основні результати, що включені в дисертацію, отримані автором особисто. У праці [42], яка опублікована спільно з науковим керівником, М.О.Недашковському належить ідея використання гіллястих ланцюгових дробів для побудови алгоритмів розв'язування СЛАР з теплицевими λ -матрицями, результати щодо розв'язування СЛАР з блочно-трихдіагональними матрицями та матричних поліноміальних рівнянь. У працях [37] та [48–61], написаних у співавторстві, автору дисертаційної роботи належать усі теоретичні та практичні результати, крім постановок задач та ідей теоретичних досліджень.

Апробація результатів дисертації. Результати роботи доповідались на Міжнародній науковій конференції “Розробка та застосування математичних методів у науково-технічних дослідженнях” (8–10 жовтня 1998 р., м.Львів), VIII та IX Міжнародних наукових конференціях імені академіка М.Кравчука (11–14 травня 2000 р. та 16–19 травня 2002 р., м.Київ), II Міжнародній науково–практичній конференції “Інформаційні технології в охороні здоров'я та практичній медицині” (19–21 червня 2002 р., м.Київ), XLV підсумковій (міжрегіональній) науково-практичній конференції (7 червня 2002 р., м.Тернопіль), IV Міжнародній науково-практичній конференції студентів, аспірантів та молодих вчених (1–3 липня 2002 р., м.Київ), VI Кримській Міжнародній Математичній школі “Метод функцій Ляпунова та його застосування” (МФЛ–2002, Крим, Алушта, 8–15 вересня 2002 р.), Міжнародній науковій конференції “Шості Боголюбівські читання” (26–30 серпня 2003 р., м.Чернівці), Міжнародній конференції “Прогнозування та прийняття рішень в умовах невизначеності” (8–12 вересня 2003 р., Київ–Алушта), Міжнародному науково-практичному семінарі “Прогнозування та

прийняття рішень в умовах невизначеності” (25–30 травня 2004 р., м.Тернопіль).

Публікації. За матеріалами проведених досліджень опубліковано 19 наукових робіт, у тому числі 5 статей – у виданнях з переліку, затвердженому ВАК України.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ТЕПЛІЦЕВИХ МАТРИЦЬ

1.1. Огляд літератури

Відомо, що системи лінійних алгебраїчних рівнянь (СЛАР) з тепліцевими матрицями, елементами яких є дійсні числа, мають місце в багатьох прикладних задачах і математичних проблемах. Так, наприклад, в обчислювальних прикладних задачах обробки зображень, автоматичного регулювання, при дослідженні функціонування електронних схем, систем масового обслуговування з очікуванням та в інших прикладних проблемах відповідні задачі часто зводяться до розв'язування СЛАР з тепліцевими матрицями [6, 27, 93].

З тепліцевими тісно пов'язаний ще один клас матриць – ганкелеві [6, 35]. Простою перестановкою стовпців (стрічок) в оберненому порядку тепліцеву (ганкелеву) матрицю можна перетворити в дві різні ганкелеві (тепліцеві) матриці. Тому для розв'язування СЛАР як з тепліцевими, так і з ганкелевими матрицями можна використовувати одні і ті ж обчислювальні схеми.

Теорія ганкелевих та тепліцевих матриць відноситься до тих розділів математики, про які не можна сказати, що їм не приділялось багато уваги в математичній літературі. Навпаки, цим теоріям присвячено багато журнальних статей та цілі монографії, але інтерес до них не послаблюється ще з кінця XIX століття. Це пояснюється, в першу чергу, широким колом застосувань названих теорій – в алгебрі, теорії функцій, гармонічному аналізі, проблемі моментів, функціональному аналізі, теорії ймовірностей та інших прикладних питаннях.

Окрім вище згаданих областей безпосереднього застосування, є ще один розділ математики, в якому тепліцеві та ганкелеві матриці відіграють роль своєрідних моделей. Континуальним аналогом системи лінійних алгебраїчних рівнянь, в яких матриці коефіцієнтів тепліцеві (тобто елементи

цих матриць залежать лише від різниці номерів стрічки та стовпця), є інтегральні рівняння з ядрами, що залежать від різниці аргументів, які охоплюють, зокрема, такий важливий для теоретичної фізики клас, як рівняння Вінера-Хопфа. Нерідко факти, виявлені на алгебраїчному рівні для вище згаданих лінійних систем, дають змогу отримати по аналогії нові результати для інтегральних рівнянь. Така ж ситуація має місце по відношенню до ганкелевих матриць (тобто таких, елементи яких залежать лише від суми індексів) і ядер, що залежать від суми аргументів.

Ще до 70-х років минулого століття в чисто алгебраїчному плані теплицевим та ганкелевим матрицям і формам не було присвячено жодної монографії російською чи українською мовами. Якщо про ганкелеві матриці та форми можна було знайти деякі відомості з “Теорії матриць” Ф.Р.Гантмахера [23], то про теплицеві матриці замовчували практично всі відомі на той час російські та перевідні курси лінійної алгебри та теорії матриць. Лише кількох стрічок присвятив теплицевим матрицям Р.Белман [9]. Відома на той час монографія У.Гренадера та Г.Сеґе “Теплицевы матрицы и их приложения” [27] повністю присвячена аналітичним проблемам.

Сам термін “теплицева форма” був введений в літературу після публікації робіт К.Каратедорі, О.Тепліца, Е.Фішера, Г.Герглотца, Ф.Ріса (1907-1915 рр.). Мова йшла в основному про форми, коефіцієнти яких пов’язані з деяким степеневим рядом, рядом Лорана або рядом Фур’є, а зовсім не про форми загального вигляду та їх чисто алгебраїчні властивості.

Вже в 70-х роках минулого століття в журнальній літературі була опублікована велика кількість результатів, які стосувались алгебри ганкелевих та теплицевих матриць і відповідних форм; ці результати склались в доволі чітку теорію. Її витoki знаходяться ще в мемуарах Г.Фробеніуса [116, 117] (1894, 1912 рр.).

Останніми роками велика увага приділяється розв’язуванню СЛАР з символічними елементами [7, 31, 72,]. Серед них значне місце займають СЛАР з поліноміальними елементами, які виникають в некласичних задачах

для диференційних рівнянь [78], в динамічному програмуванні [104], в алгоритмах оптимізації електронних схем [30]. СЛАР з поліноміальними елементами мають вигляд

$$A(\lambda)Z(\lambda) = B(\lambda), \quad (1.1)$$

де $A(\lambda)$ – теплицева λ -матриця розмірності $n \times n$, елементами якої є алгебраїчні або тригонометричні поліноми змінної (параметра) λ , тобто

$$a_{ij}(\lambda) = \sum_{k=0}^l a_{ij}^{(k)} \lambda^{l-k}, \quad i = \overline{1, n}; j = \overline{1, n+1}.$$

Права частина рівняння (1.1) $B(\lambda)$ – вектор-функція $B(\lambda) = \{B_0(\lambda), B_1(\lambda), \dots, B_{n-1}(\lambda)\}^T$, елементами якої є алгебраїчні або тригонометричні поліноми степеня l від λ .

Найпростішим методом розв’язування системи (1.1) є визначення її невідомих $Z_j(\lambda)$ у вигляді відношення двох поліномів порядку ln за формулою Крамера, для чого необхідно виконати $O(l^n n!)$ арифметичних дій. Та Е.Н.Вареїсс запропонував алгоритм [107] розв’язування систем рівнянь з поліноміальними елементами, на реалізацію якого потрібно $O(l^2 n^5)$ арифметичних операцій. Така швидкість обчислень досягається в результаті виділення спільного множника для мінорів різного порядку.

Г.І.Малашонок в [48] узагальнив алгоритм, який запропонував Е.Н.Вареїсс, і назвав його “методом гомоморфних образів”. На його реалізацію витрачається $\frac{1}{6}(2n^3 + 3n^2 - 5n)$ додавань та $\frac{1}{2}(n^3 + 2n^2 - n) - 1$ множень поліномів l -го степеня.

В роботах [120, 121] М.Т.Масцеллан розглядає прямий модулярний алгоритм зі складністю $O(n^4 l + n^3 l^2)$. Модулярні алгоритми для розв’язування поліноміальних систем застосовували також Е.Н.Вареїсс, J.D.Lipson [119].

Задачі, близькі до (1.1), вивчались С.А.Абрамовим [2] та В.М.Кублановською [43-44]. Спектральна задача знаходження λ і $U(\lambda)$, таких, що $D(\lambda)U(\lambda)=0$, яка розглядалась В.М.Кублановською для пучка $D(\lambda) = \lambda^n A_0 + \lambda^{n-1} A_1 + \dots + \lambda A_{n-1} + A_n$, утвореного матрицями $A_0, A_1, \dots, A_{n-1}, A_n$ розмірності $m \times n$, є частковим випадком СЛАР з поліноміальними елементами (1.1).

Ряд праць [69-77] присвячено М.О.Недашковським методам розв'язування СЛАР з λ -матрицями (1.1). Ним побудовані обчислювальні схеми – аналоги неунітарних алгоритмів лінійної алгебри на випадок алгебри поліномів [70], схеми виконання обчислень в багатомодульній системі лишків [77], одержано спосіб зведення до систем з числовими коефіцієнтами спеціального вигляду [72]. М.О.Недашковським описані також методи розв'язування розріджених систем з поліноміальними елементами [73].

В.М.Фаддєвою та Д.К.Фаддєвим в [96] описаний перший із наближених методів розв'язування систем з поліноміальними елементами. Це наближений алгоритм інтерполяційного типу. За ним розв'язки системи (1.1) шукаються у вигляді поліномів степеня \ln . R.T.Moenck і J.H.Carter [22] запропонували шукати розв'язок системи (1.1) у вигляді часткової суми ряду, на що необхідно $O(l^2 n^3)$ операцій. Крім згаданих авторів побудовою наближених методів для даного класу задач займалися T.Sasaki, H.Murao [124] та інші.

Як виявилось, більшість із методів розв'язування системи (1.1) з поліноміальними елементами від однієї змінної λ непридатні для розв'язування систем лінійних алгебраїчних рівнянь з m -мірними λ -матрицями, які виникають при використанні методів матричної лінеаризації для поліноміальних рівнянь, в апроксимаціях Паде [7]. Такі системи мають вигляд

$$A(\lambda_1, \lambda_2, \dots, \lambda_m)X(\lambda_1, \lambda_2, \dots, \lambda_m) = B(\lambda_1, \lambda_2, \dots, \lambda_m), \quad (1.2)$$

де $A(\lambda_1, \lambda_2, \dots, \lambda_m)$ – регулярна матриця розмірності $n \times n$, елементами якої є многочлени від $\lambda_1, \lambda_2, \dots, \lambda_m$ степеня l . Права частина рівняння

$$B(\lambda_1, \dots, \lambda_m) = (a_{1,n+1}(\lambda_1, \dots, \lambda_m), a_{2,n+1}(\lambda_1, \dots, \lambda_m), \dots, a_{n,n+1}(\lambda_1, \dots, \lambda_m))^T -$$

вектор многочленів степеня l від $\lambda_1, \lambda_2, \dots, \lambda_m$. Отже,

$$a_{ij}(\lambda_1, \lambda_2, \dots, \lambda_m) = \sum_{k_1+k_2+\dots+k_m=0}^l a_{ij(k_1 k_2 \dots k_m)} \lambda_1^{k_1} \lambda_2^{k_2} \dots \lambda_m^{k_m}, \quad i = \overline{1, n}; \quad j = \overline{1, n+1}.$$

Більшість із методів розв'язування системи (1.1) з поліноміальними елементами від однієї змінної λ непридатні для багатовимірного випадку (1.2). В першу чергу це пов'язано з тим, що відповідний допоміжний апарат не розрахований на випадок функцій від кількох змінних. Дослідження нових підходів, що дозволяють отримати дієві методи розв'язування СЛАР з m -мірними λ -матрицями займалась І.Босікова [13, 14].

Вивченням системи (1.1), а також пошуком методів їх розв'язування займалися D.Mazukelli, E.H.Vareiss [107]. В основу одного із запропонованих ними способів розв'язування покладался апарат багатовимірних апроксимацій Паде, який дозволяє подати наближений розв'язок системи (1.2) у вигляді частинної суми степеневого ряду [107].

В багатьох застосуваннях, наприклад, в неklasичних задачах для диференційних рівнянь [69] виникає необхідність розгляду систем лінійних алгебраїчних рівнянь з тригонометричними елементами. Це системи вигляду

$$A(\lambda)X(\lambda) = B(\lambda), \quad (1.3)$$

де $A(\lambda)$ – регулярна матриця розмірності $n \times n$, елементами якої є многочлени від комплексної змінної λ степеня l . $B(\lambda)$ – це вектор $(a_{1,n+1}(\lambda), a_{2,n+1}(\lambda), \dots, a_{n,n+1}(\lambda))^T$ многочленів від комплексної змінної λ степеня l ,

тобто
$$a_{ij}(\lambda) = \sum_{k=-l}^l a_{ij}^{(k)} e^{ik\lambda}, \quad (i = \overline{1, n}; j = \overline{1, n+1}), \quad \text{де}$$

$$2a_{ij}^{(k)} = \begin{cases} c_{ij}^{(k)} - id_{ij}^{(k)}, & k \geq 0; \\ c_{ij}^{(k)} + id_{ij}^{(k)}, & k < 0. \end{cases}$$

Актуальність розвитку методів розв'язування системи (1.3) пов'язана з тим, що тригонометричні многочлени є ефективним засобом наближення періодичних функцій і використовуються в різних практичних задачах.

СЛАР з λ -матрицями можуть бути застосовані для розв'язування оптимізаційних задач. На розробку ефективних методів розв'язування задач оптимізації спрямовані зусилля потужних наукових шкіл, створених І.В.Сергієнком, Н.З.Шором і Ю.Г.Стояном.

Предметом розгляду даної роботи є постановка класів задач розв'язування СЛАР з теплицевими λ -матрицями, елементами яких є алгебраїчні та тригонометричні поліноми; дослідження властивостей теплицевих λ -матриць, які використовуються при розробці алгоритмів розв'язування СЛАР з теплицевими λ -матрицями; побудова послідовних та паралельних моделей обчислень, на основі яких будуть розроблені ефективні алгоритми розв'язування СЛАР з теплицевими λ -матрицями. Математичний апарат для побудови алгоритмів розв'язування таких систем достатньо не розроблений, тому доцільними є дослідження в цій області.

1.2. Класи задач розв'язування СЛАР з теплицевими (ганкелевими) λ -матрицями

Розглядається клас задач розв'язування систем лінійних алгебраїчних рівнянь (СЛАР) вигляду (1.1), де

$$A(\lambda) = T_{n-1}(\lambda) = \begin{vmatrix} P_0(\lambda) & P_{-1}(\lambda) & P_{-2}(\lambda) & \dots & P_{-n+1}(\lambda) \\ P_1(\lambda) & P_0(\lambda) & P_{-1}(\lambda) & \dots & P_{-n+2}(\lambda) \\ P_2(\lambda) & P_1(\lambda) & P_0(\lambda) & \dots & P_{-n+3}(\lambda) \\ \dots & \dots & \dots & \dots & \dots \\ P_{n-1}(\lambda) & P_{n-2}(\lambda) & P_{n-3}(\lambda) & \dots & P_0(\lambda) \end{vmatrix} - \quad (1.4)$$

тепліцева λ -матриця розмірності $n \times n$, елементами якої є алгебраїчні або тригонометричні поліноми змінної (параметра) λ . Права частина рівняння (1.4) $B(\lambda)$ – це вектор-функція $B(\lambda) = \{B_0(\lambda), B_1(\lambda), \dots, B_{n-1}(\lambda)\}^T$, елементами якої є відповідно алгебраїчні або тригонометричні поліноми.

ОЗНАЧЕННЯ 1.1. *Матриця називається тепліцевою, якщо на кожній її діагоналі, розташованій паралельно до головної, стоять однакові елементи.*

Така матриця повністю визначається елементами першого стовпця та першого рядка.

Елементи матриці $A(\lambda)$ (як і вектора $B(\lambda)$) – це відповідно алгебраїчні поліноми степеня (порядка) l з дійсними коефіцієнтами $a_{j,k}(\lambda)$, $k = \overline{0, l}$, а

саме $P_j(\lambda) = \sum_{k=0}^{\ell} a_{j,k} \lambda^{\ell-k}$, $j = 0, \pm 1, \pm 2, \dots, \pm(n-1)$ або тригонометричні

поліноми з дійсними коефіцієнтами

$$P_j(\lambda) = T_j(\lambda) = \frac{a_{j0}}{2} + \sum_{s=1}^l a_{js} \cos s\lambda + \bar{a}_{js} \sin s\lambda, \quad j = 0, \pm 1, \pm 2, \dots, \pm(n-1).$$

Тригонометричний поліном в комплексній формі – це сума

$$T(\lambda) = \sum_{s=-l}^l C_s e^{is\lambda}, \quad \text{де } C_s = \begin{cases} a_s - ib_s, & s \geq 0; \\ a_{-s} + ib_{-s}, & s < 0. \end{cases}$$

Покладається, що значення для λ та коефіцієнти многочленів беруться з поля дійсних чисел R , так що коли елементи матриці $A(\lambda)$ обчислюються для деякого часткового значення λ , наприклад $\lambda = \lambda_0 \in F^{m \times n}$, то $A(\lambda_0) \in R^n \times R^n$.

ОЗНАЧЕННЯ 1.2. *Матриця називається регулярною (невиродженою), якщо $\det(A(\lambda_0)) \neq 0$.*

В подальших дослідженнях будемо припускати, що матриця $A(\lambda)$ не вироджена.

З теплицевими тісно пов'язаний ще один клас матриць – ганкелеві, в яких побічна (друга) діагональ і всі паралельні до неї діагоналі складаються з рівних (своїх для кожної діагоналі) елементів

$$A(\lambda) = H_{n-1}(\lambda) = \begin{vmatrix} h_0(\lambda) & h_1(\lambda) & h_2(\lambda) & \dots & h_{n-1}(\lambda) \\ h_1(\lambda) & h_2(\lambda) & h_3(\lambda) & \dots & h_n(\lambda) \\ h_2(\lambda) & h_3(\lambda) & h_4(\lambda) & \dots & h_{n+1}(\lambda) \\ \dots & \dots & \dots & \dots & \dots \\ h_{n-1}(\lambda) & h_n(\lambda) & h_{n+1}(\lambda) & \dots & h_{2n-2}(\lambda) \end{vmatrix} \quad (1.5)$$

Теплицеві та ганкелеві матриці пов'язані між собою наступним перетворенням: будь-яка ганкелева (теплицева) матриця простою перестановкою стовпців або стрічок в оберненому порядку, тобто останній стовпець (стрічка) – на місце першого, передостанній стовпець (стрічка) – на місце другого і т.д. може бути перетворена в дві різні теплицеві (ганкелеві) матриці. Тому для розв'язування систем з ганкелевими і теплицевими матрицями можна використовувати одні і ті ж обчислювальні схеми.

ТВЕРДЖЕННЯ 1.1. *Будь-яка теплицева матриця простою перестановкою стовпців (стрічок) може бути перетворена в ганкелеву і навпаки.*

Достатньо, наприклад, в матриці (1.4) переставити стовпці в оберненому порядку і отримаємо ганкелеву матрицю

$$H_{n-1}^I(\lambda) = \begin{vmatrix} P_{-n+1}(\lambda) & P_{-n+2}(\lambda) & \dots & P_{-1}(\lambda) & P_0(\lambda) \\ P_{-n+2}(\lambda) & P_{-n+3}(\lambda) & \dots & P_0(\lambda) & P_1(\lambda) \\ \dots & \dots & \dots & \dots & \dots \\ P_{-1}(\lambda) & P_0(\lambda) & \dots & P_{n-3}(\lambda) & P_{n-2}(\lambda) \\ P_0(\lambda) & P_1(\lambda) & \dots & P_{n-2}(\lambda) & P_{n-1}(\lambda) \end{vmatrix}. \quad (1.6)$$

Аналогічні маніпуляції зі стрічками матриці (1.4) приводять до ганкелевої, але вже іншої матриці

$$H_{n-1}^{-}(\lambda) = \begin{vmatrix} P_{n-1}(\lambda) & P_{n-2}(\lambda) & \dots & P_1(\lambda) & P_0(\lambda) \\ P_{n-2}(\lambda) & P_{n-3}(\lambda) & \dots & P_0(\lambda) & P_{-1}(\lambda) \\ \dots & \dots & \dots & \dots & \dots \\ P_1(\lambda) & P_0(\lambda) & \dots & P_{-n+3}(\lambda) & P_{-n+2}(\lambda) \\ P_0(\lambda) & P_{-1}(\lambda) & \dots & P_{-n+2}(\lambda) & P_{-n+1}(\lambda) \end{vmatrix} \quad (1.7)$$

Якщо ввести позначення $H_{n-1}^I(\lambda) \equiv \|h_{i+j}^I(\lambda)\|_{i,j=0}^{n-1}$, $H_{n-1}^{-}(\lambda) \equiv \|h_{i+j}^{-}(\lambda)\|_{i,j=0}^{n-1}$, то можемо записати рівність

$$h_j^I(\lambda) = P_{j-(n-1)}(\lambda), \quad h_j^{-}(\lambda) = P_{-j+(n-1)}(\lambda) \quad (j = 0, 1, 2, \dots, 2n-2). \quad (1.8)$$

Аналогічно можна із ганкелевої матриці (1.5) з допомогою формули

$$P_p^I(\lambda) = h_{p+n-1}(\lambda); \quad P_p^{-}(\lambda) = h_{-p+n-1}(\lambda) \quad (p = 0, \pm 1, \dots, \pm(n-1)) \quad (1.9)$$

отримати дві тепліцеві матриці

$$T_{n-1}^I(\lambda) = \|P_{p-q}^I(\lambda)\|_{p,q=0}^{n-1}(\lambda) = \begin{vmatrix} P_{n-1}(\lambda) & P_{n-2}(\lambda) & \dots & P_1(\lambda) & P_0(\lambda) \\ P_n(\lambda) & P_{n-1}(\lambda) & \dots & P_2(\lambda) & P_1(\lambda) \\ \dots & \dots & \dots & \dots & \dots \\ P_{2n-3}(\lambda) & P_{2n-4}(\lambda) & \dots & P_{n-1}(\lambda) & P_{n-2}(\lambda) \\ P_{2n-2}(\lambda) & P_{2n-3}(\lambda) & \dots & P_n(\lambda) & P_{n-1}(\lambda) \end{vmatrix},$$

$$T_{n-1}^{-}(\lambda) = \|P_{p-q}^{-}(\lambda)\|_{p,q=0}^{n-1}(\lambda) = \begin{vmatrix} P_{n-1}(\lambda) & P_n(\lambda) & \dots & P_{2n-3}(\lambda) & P_{2n-2}(\lambda) \\ P_{n-2}(\lambda) & P_{n-1}(\lambda) & \dots & P_{2n-4}(\lambda) & P_{2n-3}(\lambda) \\ \dots & \dots & \dots & \dots & \dots \\ P_1(\lambda) & P_2(\lambda) & \dots & P_{n-1}(\lambda) & P_n(\lambda) \\ P_0(\lambda) & P_1(\lambda) & \dots & P_{n-2}(\lambda) & P_{n-1}(\lambda) \end{vmatrix},$$

пов'язані між собою звичайним транспонуванням $(T_{n-1}^{-}(\lambda))^T = T_{n-1}^I(\lambda)$,

$$(T_{n-1}^I(\lambda))^T = T_{n-1}^{-}(\lambda).$$

Перетворення (1.8) і (1.9) можна записати за допомогою множення заданих матриць на фіксовану матрицю J_n порядку n

$$J_n = \begin{vmatrix} 0 & & & 1 \\ & & 1 & \\ & & \dots & \\ & 1 & & \\ 1 & & & 0 \end{vmatrix},$$

що безпосередньо можна перевірити співвідношеннями $H_{n-1}^+(\lambda) = T_{n-1}(\lambda)J_n$, $H_{n-1}^-(\lambda) = J_n T_{n-1}(\lambda)$, $T_{n-1}^+(\lambda) = H_{n-1}(\lambda)J_n$, $T_{n-1}^-(\lambda) = J_n H_{n-1}(\lambda)$. Цими формулами зручно користуватись ще й тому, що матриця J_n є ермітовою (дійсною та симетричною) та унітарною одночасно, тобто $J_n^* = J_n^T = J_n$, $J_n^{-1} = J_n^* (= J_n)$, а тому $J_n^2 = E$.

Подальші дослідження будуть стосуватись тепліцевих матриць всіх вище названих типів. Задача полягає у знаходженні розв'язку СЛАР (1.1) як вектор-функції параметра λ , тобто у вигляді $Z(\lambda) = \{Z_1(\lambda), Z_2(\lambda), \dots, Z_n(\lambda)\}^T$.

Оскільки в наступних дослідженнях робиться припущення, що матриця $A(\lambda)$ не вироджена в області визначення λ та коефіцієнтів поліномів, то обернена матриця $A^{-1}(\lambda)$ існує і розв'язок СЛАР (1.1) обчислюється у вигляді $Z(\lambda) = A^{-1}(\lambda)B(\lambda)$.

В дисертаційній роботі будуть розглянуті такі класи задач:

I. Розв'язування системи лінійних алгебраїчних рівнянь вигляду $A(\lambda)Z(\lambda) = B(\lambda)$, де елементи невиродженої тепліцевої λ -матриці порядку n та вектора $B(\lambda)$ – алгебраїчні поліноми степеня l .

II. Розв'язування системи лінійних алгебраїчних рівнянь $A(\lambda)Z(\lambda) = B(\lambda)$ з симетричною тепліцевою λ -матрицею, елементи якої є алгебраїчними поліномами, тобто

$$A(\lambda) = \begin{vmatrix} P_0(\lambda) & P_{-1}(\lambda) & P_{-2}(\lambda) & \dots & P_{-n+1}(\lambda) \\ P_1(\lambda) & P_0(\lambda) & P_{-1}(\lambda) & \dots & P_{-n+2}(\lambda) \\ P_2(\lambda) & P_1(\lambda) & P_0(\lambda) & \dots & P_{-n+3}(\lambda) \\ \dots & \dots & \dots & \dots & \dots \\ P_{n-1}(\lambda) & P_{n-2}(\lambda) & P_{n-3}(\lambda) & \dots & P_0(\lambda) \end{vmatrix}.$$

III. Розв'язування системи лінійних алгебраїчних рівнянь $A(\lambda)Z(\lambda) = B(\lambda)$ з невідродженою теплицевою λ -матрицею $A(\lambda)$ порядку n , елементами якої є тригонометричні поліноми.

IV. Розв'язування системи лінійних алгебраїчних рівнянь $H(\lambda)Z(\lambda) = B(\lambda)$ з невідродженою ганкелевою λ -матрицею. Елементи ганкелевої матриці та вектора $B(\lambda)$ – алгебраїчні поліноми степеня l .

V. Розв'язування системи лінійних алгебраїчних рівнянь $A(\lambda)Z(\lambda) = B(\lambda)$ з блочно–теплицевою λ -матрицею

$$A(\lambda) = \begin{bmatrix} P_0(\lambda) & P_{-1}(\lambda) & P_{-2}(\lambda) & \dots & P_{-n+1}(\lambda) \\ P_1(\lambda) & P_0(\lambda) & P_{-1}(\lambda) & \dots & P_{-n+2}(\lambda) \\ P_2(\lambda) & P_1(\lambda) & P_0(\lambda) & \dots & P_{-n+3}(\lambda) \\ \dots & \dots & \dots & \dots & \dots \\ P_{n-1}(\lambda) & P_{n-2}(\lambda) & P_{n-3}(\lambda) & \dots & P_0(\lambda) \end{bmatrix},$$

яка складається з квадратних блоків $P_{ij}(\lambda) = P_{i-j}(\lambda)$, $1 \leq i \leq j \leq n$, де кожен блок – це $P \times P$ -матриця, елементами якої є поліноми порядку l .

Матрицю можна розбити на блоки за правилом $A(\lambda) = \begin{bmatrix} A_{11}(\lambda) & A_{12}(\lambda) \\ A_{21}(\lambda) & A_{22}(\lambda) \end{bmatrix}$,

де $A_{11}(\lambda), \dots, A_{22}(\lambda)$ – менші матриці, розміри яких очевидним чином доповнюють один одного до розмірів початкової матриці $A(\lambda)$. А саме, сума кількості стрічок матриці $A_{11}(\lambda)$ (або $A_{12}(\lambda)$) та кількості стрічок матриці $A_{21}(\lambda)$ (або $A_{22}(\lambda)$) дорівнює кількості стрічок матриці $A(\lambda)$. Аналогічне твердження виконується для кількості стовпців.

Матриці можна перемножити поблочно, тобто якщо

$$A(\lambda) = \begin{bmatrix} A_{11}(\lambda) & A_{12}(\lambda) \\ A_{21}(\lambda) & A_{22}(\lambda) \end{bmatrix}, \quad B(\lambda) = \begin{bmatrix} B_{11}(\lambda) & B_{12}(\lambda) \\ B_{21}(\lambda) & B_{22}(\lambda) \end{bmatrix} \mathbf{i}$$

$$C(\lambda) = A(\lambda)B(\lambda), \text{ то}$$

$$C(\lambda) = \begin{bmatrix} A_{11}(\lambda)B_{11}(\lambda) + A_{12}(\lambda)B_{21}(\lambda) & A_{11}(\lambda)B_{12}(\lambda) + A_{12}(\lambda)B_{22}(\lambda) \\ A_{21}(\lambda)B_{11}(\lambda) + A_{22}(\lambda)B_{21}(\lambda) & A_{21}(\lambda)B_{12}(\lambda) + A_{22}(\lambda)B_{22}(\lambda) \end{bmatrix}$$

за умови, що розміри всіх блоків вибрані коректно в тому смислі, що всі матричні добутки і суми визначені, тобто кількість стовпців матриці $A_{11}(\lambda)$ дорівнює кількості стрічок матриці $B_{11}(\lambda)$, кількість стовпців матриці A_{12} дорівнює кількості стрічок матриці $B_{21}(\lambda)$ і т.д. Крім того, рівними є як кількість стрічок, так і кількість стовпців матричних добутків $A_{11}(\lambda)$, $B_{11}(\lambda)$, ... $A_{12}(\lambda)$, $B_{21}(\lambda)$ і т.д.

1.3. Деякі властивості теплицевих матриць

При розробці обчислювальних алгоритмів для класів задач розв'язування СЛАР з теплицевими λ -матрицями особливе значення має використання їх специфіки.

Надалі будуть використані не широко відомі результати з теорії теплицевих матриць, тому для зручності вони зібрані в цьому параграфі [16, 20, 21, 93].

Серед теплицевих матриць виділяються так звані циркулянтні матриці або циркулянти.

ОЗНАЧЕННЯ 1.3. *Циркулянтними матрицями або циркулянтами називають матриці, для елементів яких рівність $a_{i_1 j_2} = a_{i_2 j_1}$ має місце за виконання умови $i_1 - j_1 = i_2 - j_2 \pmod{n}$.*

Циркулянтна матриця порядку n має вигляд

$$\begin{vmatrix} a_0 & a_{-1} & a_{-2} & \dots & a_{-n+1} \\ a_{-n+1} & a_0 & a_{-1} & \dots & a_{-n+2} \\ a_{-n+2} & a_{-n+1} & a_0 & \dots & a_{-n+3} \\ \dots & \dots & \dots & \dots & \dots \\ a_{-1} & a_{-2} & a_{-3} & \dots & a_0 \end{vmatrix}. \quad (1.10)$$

Кожна стрічка циркулянтної матриці, крім першої, отримується з попередньої за допомогою зсуву вправо на одну позицію і перенесенням останнього елемента на перше місце. Таке ж твердження справедливе і для стовпців циркулянтної матриці.

Прикладом циркулянтної матриці може бути $n \times n$ -матриця перестановок

$$P = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{vmatrix} \quad (1.11)$$

та будь-які її n степенів.

ТВЕРДЖЕННЯ 1.2. *Суми двох циркулянтів і циркулянт, помножених на число, є циркулянтами.*

Тому матричний многочлен

$$A = a_0 I + a_1 P + \dots + a_{n-1} P_{n-1} \quad (1.12)$$

представляє циркулянтну матрицю при будь-якому наборі елементів a_0, a_1, \dots, a_{n-1} , які утворюють її перший стовпець. В той же час будь-яка циркулянтна матриця з першим стовпцем, що складається з елементів a_0, a_1, \dots, a_{n-1} , може бути представлена за допомогою матричного многочлена вигляду (1.12). Коефіцієнти розкладу визначаються однозначно.

ЛЕМА 1.1. *Матриця $A(\lambda)$ порядку n є циркулянтном тоді і лише тоді, коли має місце рівність*

$$A(\lambda)P = PA(\lambda), \quad (1.13)$$

де P - матриця перестановок вигляду (1.11).

Доведення. Необхідність рівності (1.13) випливає з розкладу (1.12) і можливості перестановки будь-яких степенів P . Щоб встановити достатність, розглянемо довільну матрицю $A(\lambda)$ з елементами $a_{ij}(\lambda)$, що задовольняють співвідношенню (1.13). Врахуємо, що $P^T P = I$ і запишемо (1.13) у вигляді

$$A(\lambda) = PA(\lambda)P^T. \quad (1.14)$$

В позиції i, j лівої частини цієї рівності стоїть елемент $a_{ij}(\lambda)$, в правій частині – елемент $a_{i-1, j-1}(\lambda)$ при $i, j > 1$, $a_{i-1, n}(\lambda)$ при $j = 1$ і $a_{n, j-1}(\lambda)$ при $i = 1$. Отже, $a_{ij}(\lambda) = a_{i-1, j-1}(\lambda)$ при $i, j > 1$, $a_{i, 1}(\lambda) = a_{i-1, n}(\lambda)$ при $j > 1$, $a_{1, j}(\lambda) = a_{n, j-1}(\lambda)$ при $j > 1$, які описують циркулянтну матрицю.

Довільна тепліцева матриця визначається через елементи першої стрічки та першого стовпця, але залежність оберненої матриці безпосередньо від цих елементів досить складна. Тому для зручності використовуються інші величини в якості параметрів, що задають обернену матрицю. В загальному випадку матрицю, обернену до тепліцевої, можна записати за розв'язками двох систем лінійних алгебраїчних рівнянь, в яких матрицею коефіцієнтів є дана тепліцева матриця, а праві частини вибираються певним чином і залежать від початкової тепліцевої матриці.

Для отримання оберненої матриці використаємо алгоритм, за яким початковій тепліцевій матриці та оберненій до неї (як і будь-якій квадратній матриці) ставиться у відповідність деяка квадратна матриця такого ж порядку. Для матриці, що відповідає початковій тепліцевій матриці, знайдемо скелетний розклад і від нього перейдемо до скелетного розкладу матриці, що відповідає оберненій. За цим розкладом побудуємо представлення оберненої матриці.

Для цього візьмемо матрицю перестановок вигляду (1.11)

$$Q_n = \left| \begin{array}{ccc|cc} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \end{array} \right|$$

і довільній матриці $A(\lambda)$ порядку n з елементами $a_{ij}(\lambda)$, де $j \leq n$ поставимо у відповідність матрицю

$$\begin{aligned} & A(\lambda)Q_n - Q_n A(\lambda) = \\ & = \left| \begin{array}{cc|cc} a_{12}(\lambda) - a_{n1}(\lambda) & a_{13}(\lambda) - a_{n2}(\lambda) & a_{1n}(\lambda) - a_{n,n-1}(\lambda) & a_{11}(\lambda) - a_{nn}(\lambda) \\ a_{22}(\lambda) - a_{11}(\lambda) & a_{23}(\lambda) - a_{12}(\lambda) & a_{2n}(\lambda) - a_{1,n-1}(\lambda) & a_{21}(\lambda) - a_{1n}(\lambda) \end{array} \right| \end{aligned} \quad (1.15)$$

За лемою 1.1 теплицевій матриці вигляду

$$A(\lambda) = \left| \begin{array}{cc|c} a_0(\lambda) & a_{-1}(\lambda) & a_{-n+1}(\lambda) \\ a_1(\lambda) & a_0(\lambda) & a_{-n+2}(\lambda) \\ \hline a_{n-1}(\lambda) & a_{n-2}(\lambda) & a_0(\lambda) \end{array} \right| \quad (1.16)$$

буде поставлена у відповідність матриця

$$\begin{aligned} & A(\lambda)Q_n - QA_n(\lambda) = \\ & = \left| \begin{array}{cc|cc} a_{-1}(\lambda) - a_{n-1}(\lambda) & a_{-2}(\lambda) - a_{n-2}(\lambda) & a_{-n+1}(\lambda) - a_1(\lambda) & 0 \\ 0 & 0 & 0 & -(a_{-n+1}(\lambda) - a_1(\lambda)) \\ \hline 0 & 0 & 0 & -(a_{-1}(\lambda) - a_{n-1}(\lambda)) \end{array} \right| \end{aligned} \quad (1.17)$$

Ранг цієї матриці дорівнює 0 або 2. Визначимо її скелетний розклад.

Для цього введемо два вектор-стовпці і матрицю перестановок J_n порядку n такого вигляду

$$g = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad e = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad J_n = \begin{bmatrix} 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \end{bmatrix}. \quad (1.18)$$

Тоді отримаємо $A(\lambda)Q_n - Q_n A(\lambda) = g(J_n e)^T - e(J_n g)^T$.

Якщо $A(\lambda)$ – невироджена матриця, то, помноживши обидві частини рівності зліва і справа на $A^{-1}(\lambda)$ і змінивши знаки на протилежні, отримаємо

$$A^{-1}(\lambda)Q_n - Q_n A^{-1}(\lambda) = A^{-1}(\lambda)e(J_n g)^T A^{-1}(\lambda) - A^{-1}(\lambda)g(J_n e)^T A^{-1}(\lambda). \quad (1.19)$$

Для спрощення даного виразу використаємо одну з властивостей тепліцевої матриці.

ЛЕМА 1.2 Для будь-якої тепліцевої матриці $A(\lambda)$ порядку n виконується рівність $A^T(\lambda) = J_n A(\lambda) J_n$.

Доведення. Для будь-яких i, j , де $1 \leq i, j \leq n$ в позиції i, j матриці $J_n A(\lambda) J_n$ стоїть елемент, що займає в матриці $A(\lambda)$ позицію $n+1-i, n+1-j$. Цей елемент має вигляд $a_{(n+1-i)-(n+1-j)}(\lambda) = a_{j-i}(\lambda)$; очевидно, саме він стоїть в позиції i, j матриці $A^T(\lambda)$.

Наслідок з леми 1.2. Для будь-якої тепліцевої матриці $A(\lambda)$ і будь-яких вектор-стовпців x, y порядку n рівність $A(\lambda)x = y$ виконується тоді і тільки тоді, коли має місце рівність $A^T(\lambda)J_n x = J_n y$.

ОЗНАЧЕННЯ 1.4. Квадратна λ -матриця порядку n , яка переходить при множенні її з обох сторін на J_n в транспоновану матрицю, називається персиметричною.

Лема 1.1 стверджує, що будь-яка тепліцева матриця є персиметричною. У випадку дійсних або комплексних матриць має місце рівність

$$(A^{-1}(\lambda))^T = (A^T(\lambda))^{-1}, \quad (1.20)$$

тому невироджена матриця персиметрична тоді і тільки тоді, коли обернена до неї є персиметричною. Отже, (1.19) можна записати у вигляді

$$A^{-1}(\lambda)Q_n - Q_n A^{-1}(\lambda) = A^{-1}(\lambda)e(J_n A^{-1}(\lambda)g)^T - A^{-1}(\lambda)g(J_n A^{-1}(\lambda)e)^T. \quad (1.21)$$

Тепер відтворимо $A^{-1}(\lambda)$ за координатами векторів, що входять в скелетний розклад (1.21).

ТЕОРЕМА 1.1. *Дійсна або комплексна теплицева матриця $A(\lambda)$ порядку n є невиродженою тоді і тільки тоді, коли існують розв'язки $x = [x_0 x_1 \dots x_{n-1}]^T$ і $v = [v_0 v_1 \dots v_{n-1}]^T$ двох систем*

$$A(\lambda)x = e, \quad A(\lambda)v = g, \quad (1.22)$$

де вектор-стовпці e, g визначені в (1.18). Для оберненої матриці можна записати

$$\begin{aligned} A^{-1}(\lambda) &= \left| \begin{array}{ccc|ccc} x_0 & 0 & 0 & 1 & v_{n-1} & v_2 & v_1 \\ x_1 & x_0 & 0 & 0 & 0 & 1 & v_{n-1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{n-1} & x_{n-2} & x_0 & 0 & 0 & 0 & 1 \end{array} \right| - \\ &= \left| \begin{array}{ccc|ccc} v_0 - 1 & 0 & 0 & 0 & x_{n-1} & x_1 \\ v_1 & v_0 - 1 & 0 & 0 & 0 & x_{n-1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ v_{n-1} & v_{n-2} & v_0 - 1 & 0 & 0 & 0 \end{array} \right| \end{aligned} \quad (1.23)$$

Доведення. Спочатку покажемо, що з існування розв'язків системи (1.23) випливає невиродженість матриці $A(\lambda)$. Нехай

$$\gamma A(\lambda) = 0 \quad (1.24)$$

для деякої вектор-стрічки $\gamma = [\gamma_0 \gamma_1 \dots \gamma_{n-1}]$. Помноживши цю рівність справа на вектор-стовпець x , знайдемо

$$\gamma_0 = \gamma e = 0. \quad (1.25)$$

Отже, в будь-якій рівній нулю лінійній комбінації стрічок матриці $A(\lambda)$ коефіцієнт першої стрічки дорівнює нулю.

Розглянемо вектор-стрічки $\gamma Q_n^0, \dots, \gamma Q_n^{n-1}$ і врахуємо, що перша координата i -ї стрічки дорівнює γ_i . Покажемо, що кожна з цих стрічок містить коефіцієнти рівної нулю лінійної комбінації стрічок матриці $A(\lambda)$. Для стрічки $\gamma = \gamma Q_n^0$, це справедливо за припущенням. Нехай для деякого $i, 0 \leq i \leq n-2$ має місце рівність $\gamma Q_n^i A(\lambda) = 0$. І оскільки $g = A(\lambda)v$ і перша координата вектора γQ_n^i дорівнює нулю, отримаємо

$$\begin{aligned} \gamma Q_n^{i+1} A(\lambda) &= \gamma Q_n^i (A(\lambda)Q_n - g(J_n e)^T + e(J_n g)^T) = \\ &= \gamma Q_n^i A(\lambda)(Q_n - v(J_n e)^T) + \gamma Q_n^i e(J_n g)^T = 0. \end{aligned} \quad (1.26)$$

Отже, всі координати вектор-стрічки γ дорівнюють нулю і матриця $A(\lambda)$ невинроджена.

У співвідношенні (1.21) елементи матриці $A^{-1}(\lambda)$ позначимо $a_{i,j}^{-1}(\lambda)$. При умові, що $1 < i$ та $i < n$ в позиції i, j матриці лівої частини стоїть елемент вигляду $a_{i,j+1}^{-1}(\lambda) - a_{i-1,j}^{-1}(\lambda)$ і в силу (1.21) він визначається виразом $x_{i-1}v_{n-j} - v_{n-1}x_{n-j}$. Замінімо j на $j-1$ для всіх i, j , де $1 \leq i, j \leq n$ і знайдемо $a_{i,j}^{-1}(\lambda) = a_{i-1,j-1}^{-1}(\lambda) + x_{i-1}v_{n+1-j} - v_{i-1}x_{n+1-j}$.

Для тих самих значень i, j елементи матриці правої частини формули (1.23) пов'язані такими ж співвідношеннями. Для доведення (1.23) залишилось перевірити, що перший стовпець і перша стрічка матриці правої частини дійсно є першим стовпцем і першою стрічкою матриці $A^{-1}(\lambda)$. Стосовно першого стовпця це очевидно, оскільки він співпадає з x . Для

першої стрічки при $1 < j$ згідно з (1.21) отримаємо

$$a_{1,j}^{-1}(\lambda) = a_{n,j-1}^{-1}(\lambda) + x_0 v_{n+1-j} - v_0 x_{n+1-j}.$$

Оскільки матриця $A^{-1}(\lambda)$ персиметрична, то $a_{n,j-1}^{-1}(\lambda) = a_{n+2-j,1}^{-1}(\lambda) = x_{n+1-j}$.

$$\text{Знаходимо } a_{1,j}^{-1}(\lambda) = x_0 v_{n+1-j} - (v_0 - 1)x_{n+1-j}.$$

Теорема доведена.

Наслідок 1 з теореми 1.1. Для симетричної матриці $A(\lambda)$ при $1 < j \leq n$ виконуються співвідношення $x_{j-1} = x_0 v_{n+1-j} - (v_0 - 1)x_{n+1-j}$.

Наслідок 2 з теореми 1.1. Якщо для тепліцевої матриці $A(\lambda)$ порядку n існують розв'язки $y = [y_0 y_1 \dots y_{n-1}]^T$, $u = [u_0 u_1 \dots u_{n-1}]^T$ двох систем

$$A(\lambda)y = \hat{e}, \quad A(\lambda)u = \hat{g} \quad (1.27)$$

з правими частинами $\hat{e} = [0 \dots 0 1]^T$, $\hat{g} = [a_{-n+1}(\lambda) - a_1(\lambda) \dots a_{-1}(\lambda) - a_{n-1}(\lambda) 0]^T$, то матриця $A(\lambda)$ невироджена і обернена до неї має вигляд

$$A^{-1}(\lambda) = \begin{array}{c} \left| \begin{array}{ccc|ccc} y_{n-1} & y_{n-2} & y_0 & 1 & 0 & 0 & 0 \\ 0 & y_{n-1} & y_1 & u_0 & 1 & 0 & 0 \\ \hline 0 & 0 & y_{n-1} & u_{n-1} & u_{n-3} & u_0 & 1 \end{array} \right| - \\ \left| \begin{array}{ccc|ccc} u_{n-1} - 1 & u_{n-2} & u_0 & 0 & 0 & 0 & 0 \\ 0 & u_{n-1} - 1 & u_1 & y_0 & 0 & 0 & 0 \\ \hline 0 & 0 & u_{n-1} - 1 & y_{n-2} & y_{n-3} & y_0 & 0 \end{array} \right| \end{array} \quad (1.28)$$

В силу наслідку з леми 1.2. вектор-стовпці $J_n y$ і $J_n u$, записані для матриці $A^T(\lambda)$, будуть розв'язками систем вигляду (1.25).

З точки зору існуючих обчислювальних алгоритмів найефективнішим є представлення оберненої матриці через елементи її першого та останнього стовпців.

Нехай $A(\lambda)$ – невироджена комплексна або дійсна тепліцева λ -матриця з елементами $a_{i,j}^{-1}(\lambda) = a_{i-j}^{-1}(\lambda)$ і $x = [x_0 x_1 \dots x_{n-1}]^T$, $y = [y_0 y_1 \dots y_{n-1}]^T$ – відповідно перший і останній стовпці матриці, тобто $a_{i,1}^{-1}(\lambda) = x_{i-1}, \dots, a_{i,i}^{-1}(\lambda) = y_{i-1}$ для будь-яких i, j . Матриця $A^{-1}(\lambda)$ персиметрична, тому $a_{1,1}^{-1}(\lambda) = a_{n,n}^{-1}(\lambda) = x_0 = y_{n-1}$. Щоб отримати формулу для $A^{-1}(\lambda)$, припустимо додатково, що $x_0 \neq 0$.

Використаємо теорему 1.1 і врахуємо (1.19) та персиметричність матриці $A^{-1}(\lambda)$. Запишемо $a_{1,j}^{-1}(\lambda) = a_{j,n}^{-1}(\lambda) = x_0 v_{n+1-j} - (v_0 - 1)x_{n+1-j} = y_{n-j}$ при $2 \leq j \leq n$, звідки, покладаючи $l = n + 1 - j$ і враховуючи, що $x_0 \neq 0$, знаходимо $v_l = x_0^{-1} y_{l-1} + x_0 (v_0 - 1)x_l$ при $1 \leq l \leq n - 1$.

В (1.29) величини v_l замінимо даними виразами. Всі члени, що містять v_l , взаємно знищуються. Враховуючи, що $x_0 = y_{n-1}$, отримаємо

$$A^{-1}(\lambda) = = x_0^{-1} \left\{ \begin{array}{ccc|ccc|ccc|ccc} x_0 & 0 & 0 & y_{n-1} & y_{n-2} & y_0 & 0 & 0 & 0 & 0 & x_{n-1} & x_1 \\ x_1 & x_0 & 0 & 0 & y_{n-1} & y_1 & y_0 & 0 & 0 & 0 & 0 & x_{n-1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{n-1} & x_{n-2} & x_0 & 0 & 0 & y_{n-1} & y_{n-2} & y_{n-3} & 0 & 0 & 0 & 0 \end{array} \right\}. \quad (1.29)$$

Це представлення отримане як наслідок з теореми 1.1. Насправді невиродженість матриці $A(\lambda)$ можна вивести лише з існування розв'язків x , y двох систем

$$A(\lambda)x = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad A(\lambda)y = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (1.30)$$

при умові, що $x_0 \neq 0$.

Якщо системи (1.30) мають розв'язок, то незалежно від виродженості тепліцевої матриці $A(\lambda)$ має місце рівність $x_0 = y_{n-1}$.

ТЕОРЕМА 1.2. Нехай $A(\lambda)$ – дійсна або комплексна тепліцева λ -матриця порядку n та існують розв'язки $x = [x_0 x_1 \dots x_{n-1}]^T$, $y = [y_0 y_1 \dots y_{n-1}]^T$ двох систем (1.30), причому $x_0 \neq 0$. Тоді матриця $A(\lambda)$ невинроджена і обернена до неї може бути представлена у вигляді (1.29).

Наслідок 1 з теореми 1.2. В умовах теореми 1.2 справедливе представлення

$$A^{-1}(\lambda) = x_0^{-1} \left\{ \begin{array}{ccc|ccc} y_{n-1} & y_{n-2} & y_0 & x_0 & 0 & 0 \\ 0 & y_{n-1} & y_1 & x_1 & x_0 & 0 \\ 0 & 0 & y_{n-1} & x_{n-1} & x_{n-2} & x_0 \end{array} \right\} - \left. \begin{array}{ccc|ccc} 0 & x_{n-1} & x_1 & 0 & 0 & 0 \\ 0 & 0 & x_{n-1} & y_0 & 0 & 0 \\ 0 & 0 & 0 & y_{n-2} & y_{n-3} & 0 \end{array} \right\}.$$

Наслідок 2 з теореми 1.2. В умовах теореми 1.2 в матриці $A(\lambda)$ ведуча підматриця невинроджена і обернена до неї має вигляд

$$\hat{A}(\lambda) = \begin{array}{ccc|c} a_0 & a_{-1} & a_{-n+2} & \\ a_1 & a_0 & a_{-n+3} & \\ \hline a_{n-2} & a_{n-3} & a_0 & \end{array}.$$

Дана дисертаційна робота присвячена розробці ефективних алгоритмів розв'язування СЛАР з тепліцевими (ганкелевими) λ -матрицями з алгебраїчними і тригонометричними поліноміальними елементами та оцінці їх характеристик; побудові економічних алгоритмів знаходження оберненої матриці $A^{-1}(\lambda)$ для довільної і симетричної тепліцевої λ -матриці $A(\lambda)$ з поліноміальними елементами.

РОЗДІЛ 2

АЛГОРИТМИ РОЗВ'ЯЗУВАННЯ СИСТЕМ ЛІНІЙНИХ АЛГЕБРАЇЧНИХ РІВНЯНЬ З ТЕПЛІЦЕВИМИ λ -МАТРИЦЯМИ ДЛЯ ПОСЛІДОВНИХ МОДЕЛЕЙ ОБЧИСЛЕНЬ, ОЦІНКИ ЇХ ХАРАКТЕРИСТИК

2.1. Алгоритми знаходження матриці, оберненої до тепліцевої матриці з алгебраїчними поліноміальними елементами

Розглянемо клас задач розв'язування систем лінійних алгебраїчних рівнянь з тепліцевими λ -матрицями вигляду (1.1), де елементи λ -матриці

$$A(\lambda) \text{ та вектора } B(\lambda) \text{ є алгебраїчними поліномами степеня } l \quad P(\lambda) = \sum_{k=0}^{\ell} a_k \lambda^{\ell-k}$$

Значення для λ та коефіцієнти алгебраїчних многочленів беруться з деякого поля $F^{m \times n}$ так, що елементи матриці $A(\lambda)$ обчислюються для деякого часткового значення λ , наприклад $\lambda = \lambda_0 \in F^{m \times n}$.

Для розв'язування таких СЛАР з тепліцевими λ -матрицями виникає потреба знаходження матриці $A^{-1}(\lambda)$, оберненої до $A(\lambda)$.

Знаходження матриці, оберненої до довільної тепліцевої λ -матриці.

Використовуючи наведені в розділі 1 результати, побудуємо аналог швидкого алгоритму [39] знаходження оберненої до дійсної матриці на випадок тепліцевої λ -матриці. Ідея полягає в обчисленні першого та останнього стовпців оберненої матриці без відшукування інших її елементів. Позначимо через вектори $\alpha^{(k)}(\lambda)$ та $\beta^{(k)}(\lambda)$, $k = 1, \dots, n-1$ відповідно перший та останній стовпці оберненої матриці для кожної з відсічених систем $A_k(\lambda)$, $k = 1, \dots, n-1$. Алгоритм базується на наступній лемі.

ЛЕМА 2.1. *Нехай $k \geq 2$ і тепліцеві матриці $A_k(\lambda)$ і $A_{k-1}(\lambda)$ невироджені. Тоді невиродженість матриці $A_{k-2}(\lambda)$ необхідна і достатня*

для того, щоб кожен із стовпців $\alpha^{(k)}(\lambda)$, $\beta^{(k)}(\lambda)$ виражався у вигляді

$$\text{лінійної комбінації стовпців } \begin{bmatrix} \alpha^{(k-1)}(\lambda) \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \beta^{(k-1)}(\lambda) \end{bmatrix}.$$

Наслідок з лема 2.1. В умовах лема 2.1 стовпці

$$\begin{bmatrix} \alpha^{(k-1)}(\lambda) \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \beta^{(k-1)}(\lambda) \end{bmatrix} \text{ лінійно незалежні і коефіцієнти лінійних комбінацій,}$$

які представляють $\alpha^{(k)}(\lambda)$, $\beta^{(k)}(\lambda)$, виражаються однозначно.

Головним наслідком з лема 2.1 є можливість рекурентного обчислення вектор-стовпців $\alpha^{(k)}(\lambda)$, $\beta^{(k)}(\lambda)$.

Алгоритм (\bar{A}_1) для відшукування оберненої до теплицевої λ -матриці.

Запишемо формули для знаходження першого та останнього стовпців оберненої матриці [21]

$$\text{для } k=0 \quad \alpha_0^{(0)}(\lambda) = \beta_0^{(0)}(\lambda) = \frac{1}{P_0(\lambda)},$$

для $k=1, \dots, n-1$

$$F_k(\lambda) = P_k(\lambda)\alpha_0^{(k-1)}(\lambda) + P_{k-1}(\lambda)\alpha_1^{(k-1)}(\lambda) + \dots + P_1(\lambda)\alpha_{k-1}^{(k-1)}(\lambda),$$

$$R_k(\lambda) = P_{-1}(\lambda)\beta_0^{(k-1)}(\lambda) + P_{-2}(\lambda)\beta_1^{(k-1)}(\lambda) + \dots + P_{-k}(\lambda)\beta_{k-1}^{(k-1)}(\lambda),$$

$$U_k(\lambda) = \frac{1}{1 - F_k(\lambda)R_k(\lambda)}, \quad S_k(\lambda) = -U_k(\lambda)F_k(\lambda),$$

$$V_k(\lambda) = -U_k(\lambda)R_k(\lambda), \tag{2.1}$$

$$\begin{bmatrix} \alpha_0^{(k)}(\lambda) \\ \alpha_1^{(k)}(\lambda) \\ \dots \\ \alpha_{k-1}^{(k)}(\lambda) \\ \alpha_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \alpha_0^{(k-1)}(\lambda) \\ \alpha_1^{(k-1)}(\lambda) \\ \dots \\ \alpha_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} U_k(\lambda) + \begin{bmatrix} 0 \\ \beta_0^{(k-1)}(\lambda) \\ \dots \\ \beta_{k-2}^{(k-1)}(\lambda) \\ \beta_{k-1}^{(k-1)}(\lambda) \end{bmatrix} S_k(\lambda),$$

$$\begin{bmatrix} \beta_0^{(k)}(\lambda) \\ \beta_1^{(k)}(\lambda) \\ \beta_{k-1}^{(k)}(\lambda) \\ \beta_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \alpha_0^{(k-1)}(\lambda) \\ \alpha_1^{(k-1)}(\lambda) \\ \alpha_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} V_k(\lambda) + \begin{bmatrix} 0 \\ \beta_0^{(k-1)}(\lambda) \\ \beta_{k-2}^{(k-1)}(\lambda) \\ \beta_{k-1}^{(k-1)}(\lambda) \end{bmatrix} U_k(\lambda).$$

Теоретично в (2.1) є лише одне місце алгоритму, яке може привести до похибки (ділення на нуль). Це операція ділення, яка визначає величину $U_k(\lambda)$. Але за лемою 2.1 помилка (теоретично) неможлива, якщо в матриці $A(\lambda)$ будуть відмінними від нуля всі ведучі мінори. При цьому гарантується, що $1 - F_k(\lambda)R_k(\lambda) \neq 0$ для будь-якого k .

Після того, як знайдено перший та останній стовпці оберненої матриці, можна повністю відтворити $A^{-1}(\lambda)$ за такою схемою [21]

$$A^{-1}(\lambda) = \alpha_0^{-1}(\lambda) \left\{ \begin{array}{l} \left| \begin{array}{cccccc} \alpha_0(\lambda) & 0 & 0 & \dots & 0 \\ \alpha_1(\lambda) & \alpha_0(\lambda) & 0 & \dots & 0 \\ \alpha_{n-1}(\lambda) & \alpha_{n-2}(\lambda) & \alpha_{n-3}(\lambda) & \dots & \alpha_0(\lambda) \end{array} \right| \times \\ \times \left| \begin{array}{ccccc} \beta_{n-1}(\lambda) & \beta_{n-2}(\lambda) & \beta_{n-3}(\lambda) & \dots & \beta_0(\lambda) \\ 0 & \beta_{n-1}(\lambda) & \beta_{n-2}(\lambda) & \dots & \beta_1(\lambda) \\ 0 & 0 & 0 & \dots & \beta_{n-1}(\lambda) \end{array} \right| - \\ - \left| \begin{array}{ccccc} 0 & 0 & \dots & 0 & 0 \\ \beta_0(\lambda) & 0 & \dots & 0 & 0 \\ \beta_{n-2}(\lambda) & \beta_{n-3}(\lambda) & \dots & \beta_0(\lambda) & 0 \end{array} \right| \times \\ \times \left| \begin{array}{ccccc} 0 & \alpha_{n-1}(\lambda) & \dots & \alpha_2(\lambda) & \alpha_1(\lambda) \\ 0 & 0 & \dots & 0 & \alpha_{n-1}(\lambda) \\ 0 & 0 & \dots & 0 & 0 \end{array} \right| \end{array} \right\} \quad (2.2)$$

Для ілюстрації розглянемо застосування цього алгоритму до тепліцевої матриці третього порядку з поліноміальними елементами.

Приклад 2.1. Задана матриця

$$A(\lambda) = \begin{vmatrix} 1-\lambda & 2-2\lambda & 1-\lambda \\ -1+\lambda & 1-\lambda & 2-2\lambda \\ 2-\lambda & -1+\lambda & 1-\lambda \end{vmatrix}. \quad (2.3)$$

Знайти обернену до неї.

Розв'язування. Застосуємо алгоритм (2.1) до матриці (2.3).

$$\text{для } k=0 \quad \alpha_0^{(0)}(\lambda) = \beta_0^{(0)}(\lambda) = \frac{1}{1-\lambda},$$

$$\text{для } k=1 \quad F_1(\lambda) = P_1(\lambda)\alpha_0^{(0)}(\lambda) = \frac{-1+\lambda}{1-\lambda} = -1,$$

$$R_1(\lambda) = P_{-1}(\lambda)\beta_0^{(0)}(\lambda) = \frac{2-2\lambda}{1-\lambda} = 2,$$

$$U_1(\lambda) = \frac{1}{1-F_1(\lambda)R_1(\lambda)} = \frac{1}{3}, \quad S_1(\lambda) = -U_1(\lambda)F_1(\lambda) = \frac{1}{3},$$

$$V_1(\lambda) = -U_1(\lambda)R_1(\lambda) = -\frac{2}{3}, \quad \begin{bmatrix} \alpha_0^{(1)}(\lambda) \\ \alpha_1^{(1)}(\lambda) \end{bmatrix} = \begin{bmatrix} \alpha_0^{(0)}(\lambda) \\ 0 \end{bmatrix} U_1(\lambda) +$$

$$\begin{bmatrix} 0 \\ \beta_0^{(0)}(\lambda) \end{bmatrix} S_1(\lambda) = \begin{bmatrix} \frac{1}{1-\lambda} \\ 0 \end{bmatrix} \frac{1}{3} + \begin{bmatrix} 0 \\ \frac{1}{1-\lambda} \end{bmatrix} \frac{1}{3} = \begin{bmatrix} \frac{1}{3(1-\lambda)} \\ \frac{1}{3(1-\lambda)} \end{bmatrix},$$

$$\begin{bmatrix} \beta_0^{(1)}(\lambda) \\ \beta_1^{(1)}(\lambda) \end{bmatrix} = \begin{bmatrix} \alpha_0^{(0)}(\lambda) \\ 0 \end{bmatrix} V_1(\lambda) + \begin{bmatrix} 0 \\ \beta_0^{(0)}(\lambda) \end{bmatrix} U_1(\lambda) =$$

$$= \begin{bmatrix} \frac{1}{1-\lambda} \\ 0 \end{bmatrix} \left(-\frac{2}{3}\right) + \begin{bmatrix} 0 \\ \frac{1}{1-\lambda} \end{bmatrix} \frac{1}{3} = \begin{bmatrix} -\frac{2}{3(1-\lambda)} \\ \frac{1}{3(1-\lambda)} \end{bmatrix};$$

$$\text{для } k=2 \quad F_2(\lambda) = P_2(\lambda)\alpha_0^{(1)}(\lambda) + P_1(\lambda)\alpha_1^{(1)}(\lambda) = \frac{2-\lambda}{3(1-\lambda)} + \frac{-1+\lambda}{3(1-\lambda)} = \frac{1}{3(1-\lambda)},$$

$$R_2(\lambda) = P_{-1}(\lambda)\beta_0^{(1)}(\lambda) + P_{-2}(\lambda)\beta_1^{(1)}(\lambda) = -\frac{2(2-2\lambda)}{3(1-\lambda)} + \frac{1-\lambda}{3(1-\lambda)} = -1,$$

$$U_2(\lambda) = \frac{1}{1-F_2(\lambda)R_2(\lambda)} = \frac{3(1-\lambda)}{4-3\lambda},$$

$$S_2(\lambda) = -U_2(\lambda)F_2(\lambda) = -\frac{1}{4-3\lambda},$$

$$V_2(\lambda) = -U_2(\lambda)R_2(\lambda) = \frac{3(1-\lambda)}{4-3\lambda},$$

$$\begin{aligned} \begin{bmatrix} \alpha_0^{(2)}(\lambda) \\ \alpha_1^{(2)}(\lambda) \\ \alpha_2^{(2)}(\lambda) \end{bmatrix} &= \begin{bmatrix} \alpha_0^{(1)}(\lambda) \\ \alpha_1^{(1)}(\lambda) \\ 0 \end{bmatrix} U_2(\lambda) + \begin{bmatrix} 0 \\ \beta_0^{(1)}(\lambda) \\ \beta_1^{(1)}(\lambda) \end{bmatrix} S_2(\lambda) = \\ &= \begin{bmatrix} \frac{1}{3(1-\lambda)} \\ \frac{1}{3(1-\lambda)} \\ 0 \end{bmatrix} \frac{3(1-\lambda)}{4-3\lambda} + \begin{bmatrix} 0 \\ \frac{1}{3(1-\lambda)} \\ \frac{1}{3(1-\lambda)} \end{bmatrix} \left(-\frac{1}{4-3\lambda} \right) = \begin{bmatrix} \frac{1}{4-3\lambda} \\ \frac{1}{5-3\lambda} \\ -\frac{1}{3(1-\lambda)(4-3\lambda)} \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} \begin{bmatrix} \beta_0^{(2)}(\lambda) \\ \beta_1^{(2)}(\lambda) \\ \beta_2^{(2)}(\lambda) \end{bmatrix} &= \begin{bmatrix} \alpha_0^{(1)}(\lambda) \\ \alpha_1^{(1)}(\lambda) \\ 0 \end{bmatrix} V_2(\lambda) + \begin{bmatrix} 0 \\ \beta_0^{(1)}(\lambda) \\ \beta_1^{(1)}(\lambda) \end{bmatrix} U_2(\lambda) = \\ &= \begin{bmatrix} \frac{1}{3(1-\lambda)} \\ \frac{1}{3(1-\lambda)} \\ 0 \end{bmatrix} \frac{3(1-\lambda)}{4-3\lambda} + \begin{bmatrix} 0 \\ \frac{1}{3(1-\lambda)} \\ \frac{1}{3(1-\lambda)} \end{bmatrix} \frac{3(1-\lambda)}{4-3\lambda} = \begin{bmatrix} \frac{1}{4-3\lambda} \\ \frac{1}{4-3\lambda} \\ \frac{1}{4-3\lambda} \end{bmatrix}. \end{aligned}$$

Для матриці (2.3) запишемо обернену за формулою (2.2)

$$\begin{aligned}
A^{-1}(\lambda) &= (4-3\lambda) \left\{ \begin{array}{ccc} \frac{1}{4-3\lambda} & 0 & 0 \\ \frac{4-3\lambda}{5-3\lambda} & \frac{1}{4-3\lambda} & 0 \\ \frac{1}{3(1-\lambda)(4-3\lambda)} & \frac{1}{3(1-\lambda)(4-3\lambda)} & \frac{1}{4-3\lambda} \end{array} \right\} \times \\
& \left| \begin{array}{ccc} \frac{1}{4-3\lambda} & -\frac{1}{4-3\lambda} & \frac{1}{4-3\lambda} \\ 0 & \frac{1}{4-3\lambda} & -\frac{1}{4-3\lambda} \\ 0 & 0 & \frac{1}{4-3\lambda} \end{array} \right| - \left| \begin{array}{ccc} 0 & 0 & 0 \\ \frac{1}{4-3\lambda} & 0 & 0 \\ \frac{1}{4-3\lambda} & \frac{1}{4-3\lambda} & 0 \end{array} \right| \times \\
& \left. \begin{array}{ccc} 0 & -\frac{1}{3(1-\lambda)(4-3\lambda)} & \frac{5-3\lambda}{3(1-\lambda)(4-3\lambda)} \\ \times 0 & 0 & -\frac{1}{3(1-\lambda)(4-3\lambda)} \\ 0 & 0 & 0 \end{array} \right\} = \\
& = \left| \begin{array}{ccc} \frac{1}{4-3\lambda} & -\frac{1}{4-3\lambda} & \frac{1}{4-3\lambda} \\ \frac{4-3\lambda}{5-3\lambda} & \frac{1}{3(1-\lambda)(4-3\lambda)} & -\frac{1}{3(1-\lambda)(4-3\lambda)} \\ \frac{1}{3(1-\lambda)(4-3\lambda)} & \frac{1}{3(1-\lambda)(4-3\lambda)} & \frac{1}{4-3\lambda} \end{array} \right|.
\end{aligned}$$

Безпосередньою перевіркою можна переконатись, що ця матриця є оберненою до матриці (2.3).

Алгоритм з нормуванням (\bar{A}_2) для відшукування оберненої до тепліцевої λ -матриці.

Існує можливість і більш економічної реалізації наведеного алгоритму [21]. За допомогою нормування стовпців, що обчислюються, завжди можна досягти того, щоб один з коефіцієнтів у лінійній комбінації дорівнював 1. При цьому доведеться обчислювати не стовпці $\alpha^{(k)}(\lambda)$ та $\beta^{(k)}(\lambda)$

безпосередньо, а деякі інші, колінеарні до них, разом з відповідними нормувальними множниками. Тоді на кожному кроці буде виконуватись менше число множень, а кількість операцій додавання–віднімання не зміниться [39].

Здійснювати нормування можна по-різному [21]. При $k = 0$ початкові нормувальні множники взагалі можуть бути будь-якими, відмінними від 0. Деяка довільність може відповідати вибору одного з коефіцієнтів лінійної комбінації, який стане одиницею.

Нехай на k -ому кроці обчислюються вектор-стовпці

$$\tilde{\alpha}^{(k)}(\lambda) = \begin{bmatrix} \tilde{\alpha}_0^{(k)}(\lambda) \\ \tilde{\alpha}_1^{(k)}(\lambda) \\ \dots \\ \tilde{\alpha}_{k-1}^{(k)}(\lambda) \\ \tilde{\alpha}_k^{(k)}(\lambda) \end{bmatrix}, \quad \tilde{\beta}^{(k)}(\lambda) = \begin{bmatrix} \tilde{\beta}_0^{(k)}(\lambda) \\ \tilde{\beta}_1^{(k)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-1}^{(k)}(\lambda) \\ \tilde{\beta}_k^{(k)}(\lambda) \end{bmatrix}$$

і нормувальні множники $G_k(\lambda)$, $Q_k(\lambda)$, що їм відповідають і забезпечують виконання співвідношень

$$\alpha^{(k)}(\lambda) = \tilde{\alpha}^{(k)}(\lambda)G_k(\lambda), \quad \beta^{(k)}(\lambda) = \tilde{\beta}^{(k)}(\lambda)Q_k(\lambda). \quad (2.4)$$

Алгоритм з нормуванням (\bar{A}_2) для відшукування оберненої до тепліцевої матриці з поліноміальними елементами можна описати так:

$$\text{для } k = 0 \quad G_0(\lambda) = Q_0(\lambda) = \frac{1}{P_0(\lambda)},$$

$$\tilde{\alpha}_0^{(0)}(\lambda) = \frac{1}{P_0(\lambda)G_0(\lambda)}, \quad \tilde{\beta}_0^{(0)}(\lambda) = \frac{1}{P_0(\lambda)Q_0(\lambda)};$$

для $k = 1, \dots, n-1$

$$\tilde{F}_k(\lambda) = P_k(\lambda)\tilde{\alpha}_0^{(k-1)}(\lambda) + P_{k-1}(\lambda)\tilde{\alpha}_1^{(k-1)}(\lambda) + \dots + P_1(\lambda)\tilde{\alpha}_{k-1}^{(k-1)}(\lambda),$$

$$\tilde{R}_k(\lambda) = P_{-1}(\lambda)\tilde{\beta}_0^{(k-1)}(\lambda) + P_{-2}(\lambda)\tilde{\beta}_1^{(k-1)}(\lambda) + \dots + P_{-k}(\lambda)\tilde{\beta}_{k-1}^{(k-1)}(\lambda),$$

$$\tilde{S}_k(\lambda) = -\tilde{F}_k(\lambda)G_{k-1}(\lambda), \quad \tilde{V}_k(\lambda) = -\tilde{R}_k(\lambda)Q_{k-1}(\lambda),$$

$$\tilde{U}_k(\lambda) = \frac{1}{1 - \tilde{S}_k(\lambda)\tilde{V}_k(\lambda)}, \quad G_k(\lambda) = G_{k-1}(\lambda)\tilde{U}_k(\lambda), \quad Q_k(\lambda) = Q_{k-1}(\lambda)\tilde{U}_k(\lambda),$$

$$\begin{bmatrix} \tilde{\alpha}_0^{(k)}(\lambda) \\ \tilde{\alpha}_1^{(k)}(\lambda) \\ \dots \\ \tilde{\alpha}_{k-1}^{(k)}(\lambda) \\ \tilde{\alpha}_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(k-1)}(\lambda) \\ \tilde{\alpha}_1^{(k-1)}(\lambda) \\ \dots \\ \tilde{\alpha}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix} \tilde{S}_k(\lambda),$$

(2.5)

$$\begin{bmatrix} \tilde{\beta}_0^{(k)}(\lambda) \\ \tilde{\beta}_1^{(k)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-1}^{(k)}(\lambda) \\ \tilde{\beta}_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(k-1)}(\lambda) \\ \tilde{\alpha}_1^{(k-1)}(\lambda) \\ \dots \\ \tilde{\alpha}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} \tilde{V}_k(\lambda) + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix}.$$

Слід відзначити, що в процесі виконання (2.5) перші координати векторів $\tilde{\alpha}^{(k)}(\lambda)$ і останні координати векторів $\tilde{\beta}^{(k)}(\lambda)$ не змінюються, тобто $\tilde{\alpha}_0^{(0)}(\lambda) = \tilde{\alpha}_0^{(k)}(\lambda)$, $\tilde{\beta}_0^{(0)}(\lambda) = \tilde{\beta}_k^{(k)}(\lambda)$ для всіх k . Крім того, вибір однакових початкових нормувальних множників $G_0(\lambda) = Q_0(\lambda)$ призводить до їх співпадання на всіх кроках алгоритму. Звичайно покладають

$$G_0(\lambda) = Q_0(\lambda) = 1 \quad \text{або} \quad G_0(\lambda) = Q_0(\lambda) = \frac{1}{P_0(\lambda)}.$$

Оцінки характеристик алгоритмів \bar{A}_1 і \bar{A}_2 .

Обчислювальна складність алгоритму \bar{A}_1 знаходження оберненої до тепліцевої λ -матриці. Підрахуємо кількість операцій, необхідних для реалізації формул (2.2) алгоритму \bar{A}_1 на комп'ютері. Для обчислення кожної з величин $F_k(\lambda)$, $R_k(\lambda)$ на k -ому кроці потрібно виконати по kl^2 операцій множення та $(k-1)l^2$ операцій додавання-віднімання. Разом це складе $2kl^2$ операцій множення та $2(k-1)l^2$ операцій додавання-віднімання. Для

визначення величин $V_k(\lambda)$, $U_k(\lambda)$ та $S_k(\lambda)$ необхідно здійснити $3l^2$ операцій множення, l^2 операцій віднімання та l^2 операцій ділення. Щоб обчислити значення всіх $\alpha_i^{(k)}(\lambda)$, $\beta_i^{(k)}(\lambda)$ ($i=0,1,\dots,k-1$), треба виконати на кожному кроці $2(k+1)l^2$ операцій додавання-віднімання і $2kl^2$ операцій множення.

Отже, на k -ому кроці алгоритму (2.1) виконується порядку $6kl^2$ операцій множення і $4kl^2$ операцій додавання-віднімання. Тому загальні затрати складають з точністю до головного члена $3n^2l^2$ операцій множення і $2n^2l^2$ операцій додавання-віднімання. Операції ділення в головний член не входять.

Обчислювальна складність алгоритму з нормуванням \bar{A}_2 знаходження оберненої до тепліцевої λ -матриці. Як бачимо, в алгоритмі (2.5) при обчисленні величин $\alpha_i^{(k)}(\lambda)$ та $\beta_i^{(k)}(\lambda)$, $i=0,1,\dots,k-1$ на кожному кроці виконується на l^2 операцій множення менше, ніж в алгоритмі без нормування. В результаті їх буде $4kl^2$ замість $4kl^2$, як в алгоритмі \bar{A}_1 (2.1).

Оцінка використаної пам'яті. Щоб реалізувати (2.1) на комп'ютері, необхідно забезпечити запам'ятовування, крім елементів початкової матриці $P_i(\lambda)$, $-n+1 \leq i \leq n-1$, двох векторів довжиною не більше n , координатами якого є поліноми степеня не вище l^2 .

Алгоритм з нормуванням (\hat{A}_2) знаходження оберненої до симетричної тепліцевої λ -матриці. Нехай тепліцева матриця $A(\lambda)$ – симетрична. В цьому випадку всі ведучі підматриці $A_k(\lambda)$, $0 \leq k \leq n-1$ і обернені до них також симетричні і одночасно є персиметричними. Отже, для всіх k мають місце співвідношення

$$\alpha_i^{(k)}(\lambda) = \beta_{k-i}^{(k)}(\lambda), \quad 0 \leq i \leq k, \quad (2.6)$$

в силу симетричності матриці $P_i(\lambda) - P_{-i}(\lambda)$, $0 \leq i \leq n-1$, тому згідно (2.1) знаходимо

$$F_k(\lambda) = R_k(\lambda), \quad S_k(\lambda) = V_k(\lambda), \quad 0 \leq k \leq n-1.$$

Крім того, вважаючи, що в (2.5) $G_0(\lambda) = Q_0(\lambda)$, і враховуючи (2.4) і (2.1), отримаємо

$$\tilde{F}_k(\lambda) = \tilde{R}_k(\lambda), \quad \tilde{S}_k(\lambda) = \tilde{V}_k(\lambda), \quad \tilde{\alpha}_i(\lambda) = \tilde{\beta}_{n-i}(\lambda) \quad (2.7)$$

для всіх k, i , де $0 \leq i \leq k$, $0 \leq k \leq n-1$. Враховуючи (2.1) та ідею алгоритму \bar{A}_2 , (2.5) можна використати для обчислення оберненої до симетричної тепліцевої λ -матриці (алгоритм \hat{A}_2)

$$\text{для } k=0 \quad G_0(\lambda) = \frac{1}{P_0(\lambda)}, \quad \tilde{\alpha}_0^{(0)}(\lambda) = \frac{1}{P_0(\lambda)G_0(\lambda)},$$

для $k=1, \dots, n-1$

$$\begin{aligned} \tilde{F}_k(\lambda) &= P_k(\lambda)\tilde{\alpha}_0^{(k-1)}(\lambda) + P_{k-1}(\lambda)\tilde{\alpha}_1^{(k-1)}(\lambda) + \dots + P_1(\lambda)\tilde{\alpha}_{k-1}^{(k-1)}(\lambda), \\ \tilde{S}_k(\lambda) &= -\tilde{F}_k(\lambda)G_{k-1}(\lambda), \quad G_k(\lambda) = \frac{G_{k-1}(\lambda)}{1 - \tilde{S}_k^2(\lambda)}, \end{aligned} \quad (2.8)$$

$$\begin{bmatrix} \tilde{\alpha}_0^{(k)}(\lambda) \\ \tilde{\alpha}_1^{(k)}(\lambda) \\ \tilde{\alpha}_{k-1}^{(k)}(\lambda) \\ \tilde{\alpha}_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(k-1)}(\lambda) \\ \tilde{\alpha}_1^{(k-1)}(\lambda) \\ \tilde{\alpha}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix} \tilde{S}_k(\lambda).$$

Оцінки характеристик алгоритму з нормуванням (\hat{A}_2) знаходження оберненої до симетричної тепліцевої λ -матриці.

Обчислювальна складність. У випадку симетричної матриці кількість операцій скорочується вдвічі і для реалізації (2.1) треба виконати $2kl^2$ операцій множення і $2kl^2$ операцій додавання-віднімання на k -ому кроці. Тому загальні затрати складають з точністю до головного члена лише n^2l^2 операцій множення і n^2l^2 операцій додавання-віднімання.

Оцінка необхідної оперативної пам'яті. У випадку симетричної матриці пам'ять комп'ютера також скорочується вдвічі за рахунок того, що матриця $A(\lambda)$ задається лише своїм першим стовпцем, (а не стовпцем і стрічкою, як звичайна тепліцева матриця), і в процесі обчислень необхідно зберігати лише один вектор довжиною не більше n з поліноміальними координатами степеня не вище l^2 .

Обчислення розв'язків СЛАР із симетричною поліноміальною тепліцевою λ -матрицею (алгоритм \hat{A}_{22}).

$$\text{для } k=0 \quad G_0(\lambda) = \frac{1}{P_0(\lambda)} \quad \tilde{\alpha}_0^{(0)}(\lambda) = \frac{1}{P_0(\lambda)G_0(\lambda)},$$

для $k=1, \dots, n-1$

$$\tilde{F}_k(\lambda) = P_k(\lambda)\tilde{\alpha}_0^{(k-1)}(\lambda) + P_{k-1}(\lambda)\tilde{\alpha}_1^{(k-1)}(\lambda) + \dots + P_1(\lambda)\tilde{\alpha}_{k-1}^{(k-1)}(\lambda),$$

$$\tilde{S}_k(\lambda) = -\tilde{F}_k(\lambda)G_{k-1}(\lambda), \quad G_k(\lambda) = \frac{G_{k-1}(\lambda)}{1 - \tilde{S}_k^2(\lambda)},$$

$$\begin{bmatrix} \tilde{\alpha}_0^{(k)}(\lambda) \\ \tilde{\alpha}_1^{(k)}(\lambda) \\ \tilde{\alpha}_{k-1}^{(k)}(\lambda) \\ \tilde{\alpha}_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(k-1)}(\lambda) \\ \tilde{\alpha}_1^{(k-1)}(\lambda) \\ \tilde{\alpha}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix} \tilde{S}_k(\lambda),$$

Оскільки всі $A_k(\lambda)$ — симетричні і тепліцеві, то знаючи вектор $\tilde{\alpha}^{(k-1)}(\lambda)$, можна відтворити вектор $\tilde{\beta}^{(k-1)}(\lambda)$ за формулами

$$\alpha_0^{(k-1)}(\lambda) = \beta_{k-1}^{(k-1)}(\lambda), \alpha_1^{(k-1)}(\lambda) = \beta_{k-2}^{(k-1)}(\lambda), \dots, \alpha_{k-2}^{(k-1)}(\lambda) = \beta_1^{(k-1)}(\lambda), \alpha_{k-1}^{(k-1)}(\lambda) = \beta_0^{(k-1)}(\lambda).$$

$$D_k(\lambda) = B_k(\lambda) - P_k(\lambda)Z_0^{(k-1)}(\lambda) - P_{k-1}(\lambda)Z_1^{(k-1)}(\lambda) - \dots - P_1(\lambda)Z_{k-1}^{(k-1)}(\lambda),$$

$$W_k(\lambda) = D_k(\lambda)G_k(\lambda),$$

$$\begin{bmatrix} Z_0^{(k)}(\lambda) \\ Z_1^{(k)}(\lambda) \\ \dots \\ Z_{k-1}^{(k)}(\lambda) \\ Z_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} Z_0^{(k-1)}(\lambda) \\ Z_1^{(k-1)}(\lambda) \\ \dots \\ Z_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} \tilde{\beta}_0^{(k)}(\lambda) \\ \tilde{\beta}_1^{(k)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-1}^{(k)}(\lambda) \\ \tilde{\beta}_k^{(k)}(\lambda) \end{bmatrix} W_k(\lambda). \quad (2.9)$$

Оцінки характеристик обчислення розв'язків СЛАР із симетричною поліноміальною тепліцевою λ -матрицею (алгоритм \hat{A}_{22}).

У випадку симетричної матриці кількість операцій скорочується вдвоє і для реалізації (2.11) треба виконати $2kl^2$ операцій множення і $2kl^2$ операцій додавання-віднімання на k -ому кроці. Тому загальні затрати складають з точністю до головного члена лише n^2l^2 операцій множення і n^2l^2 операцій додавання-віднімання.

Оцінка пам'яті. Пам'ять комп'ютера (кількість комірок ОП) для алгоритму \hat{A}_{22} також скорочується вдвічі за рахунок того, що матриця $A(\lambda)$ задається лише своїм першим стовпцем, (а не стовпцем і стрічкою, як звичайна тепліцева матриця), і в процесі обчислень необхідно зберігати лише один вектор довжиною не більше n з поліноміальними координатами степеня не вище l^2 .

2.2. Алгоритми обчислення розв'язків СЛАР з тепліцевими матрицями з алгебраїчними поліноміальними елементами.

Розв'язується система лінійних алгебраїчних рівнянь вигляду (1.1).

Нехай матриця $A(\lambda)$ – деяка поліноміальна тепліцева λ -матриця. Припустимо, що всі ведучі мінори матриці $A(\lambda)$ відмінні від нуля. В цьому випадку для знаходження оберненої матриці можна застосувати один з алгоритмів, наведених вище і знайти представлення $A^{-1}(\lambda)$ у вигляді суми двох парних добутків тепліцевих трикутних матриць [36], а потім помножити її на вектор-стовпець $B(\lambda)$. По суті розв'язування системи (1.1) можна

розділити на два самостійних етапи: обернення матриці і обробка довільної правої частини [37].

Розглянемо відсічені системи

$$A_k(\lambda)Z^{(k)}(\lambda) = B^{(k)}(\lambda), \quad (2.10)$$

де

$$A_k(\lambda) = \begin{bmatrix} P_0(\lambda) & P_{-1}(\lambda) & P_{-2}(\lambda) & \dots & P_{-k}(\lambda) \\ P_1(\lambda) & P_0(\lambda) & P_{-1}(\lambda) & \dots & P_{-k+1}(\lambda) \\ P_2(\lambda) & P_1(\lambda) & P_0(\lambda) & \dots & P_{-k+2}(\lambda) \\ \dots & \dots & \dots & \dots & \dots \\ P_k(\lambda) & P_{k-1}(\lambda) & P_{k-2}(\lambda) & \dots & P_0(\lambda) \end{bmatrix},$$

$$Z^{(k)}(\lambda) = \begin{bmatrix} Z_0^{(k)}(\lambda) \\ Z_1^{(k)}(\lambda) \\ \dots \\ Z_{k-1}^{(k)}(\lambda) \\ Z_k^{(k)}(\lambda) \end{bmatrix}, \quad B^{(k)}(\lambda) = \begin{bmatrix} B_0(\lambda) \\ B_1(\lambda) \\ \dots \\ B_{k-1}(\lambda) \\ B_k(\lambda) \end{bmatrix}.$$

Вектор $B_{n-1}(\lambda)$ співпадає з $B(\lambda)$, отже, $Z_{n-1}(\lambda)$ є шуканим розв'язком

$Z(\lambda)$. $Z_0(\lambda)$ визначається як $Z_0^{(0)}(\lambda) = \frac{B_0(\lambda)}{P_0(\lambda)}$. Якщо вже відомий вектор при

$1 \leq k \leq n-1$, то $Z_k(\lambda)$ будемо шукати у вигляді

$$Z^{(k)}(\lambda) = \begin{bmatrix} Z_0^{(k-1)}(\lambda) \\ Z_1^{(k-1)}(\lambda) \\ \dots \\ Z_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + t^{(k)}(\lambda). \quad (2.11)$$

Помножимо обидві частини рівності зліва на матрицю $B_{n-1}(\lambda)$. Враховуючи (2.10), отримаємо

$$A_k(\lambda)t^{(k)}(\lambda) = \begin{bmatrix} 0 \\ \dots \\ 0 \\ D_k \end{bmatrix},$$

де $D_k(\lambda) = B_k(\lambda) - P_k(\lambda)Z_0^{(k-1)}(\lambda) - P_{k-1}(\lambda)Z_1^{(k-1)}(\lambda) - \dots - P_1(\lambda)Z_{k-1}^{(k-1)}(\lambda)$ (2.12)

Отже, вектор-стовпець $t^{(k)}(\lambda)$ пропорційний до останнього стовпця матриці $A_k^{-1}(\lambda)$. Якщо позначити його $\beta^{(k)}(\lambda)$, отримаємо

$$Z^{(k)}(\lambda) = \begin{bmatrix} Z_0^{(k-1)}(\lambda) \\ Z_1^{(k-1)}(\lambda) \\ \dots \\ Z_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + D_k(\lambda) \beta^{(k)}(\lambda). \quad (2.13)$$

Формули (2.12), (2.13) описують деякий рекурентний процес обчислення векторів $Z^{(k)}(\lambda)$ – алгоритм (\bar{A}_{11}) обчислення розв'язків системи лінійних алгебраїчних рівнянь з тепліцевою λ -матрицею.

Для того, щоб його використати, залишається знайти спосіб рекурентного обчислення векторів $\beta^{(k)}(\lambda)$. Для цього можна використати алгоритми (2.1) або (2.5).

Опишемо алгоритм з нормуванням для розв'язування системи $A(\lambda)Z(\lambda) = B(\lambda)$ в загальному вигляді (алгоритм \bar{A}_{22}), використовуючи для знаходження оберненої матриці алгоритм з нормуванням:

$$\text{для } k=0 \quad G_0(\lambda) = Q_0(\lambda) = \frac{1}{P_0(\lambda)},$$

$$\tilde{\alpha}_0^{(0)}(\lambda) = \frac{1}{P_0(\lambda)G_0(\lambda)}, \quad \tilde{\beta}_0^{(0)}(\lambda) = \frac{1}{P_0(\lambda)Q_0(\lambda)}, \quad Z_0^{(0)}(\lambda) = \frac{B_0(\lambda)}{P_0(\lambda)}$$

для $k=1, \dots, n-1$

$$\tilde{F}_k(\lambda) = P_k(\lambda)\tilde{\alpha}_0^{(k-1)}(\lambda) + P_{k-1}(\lambda)\tilde{\alpha}_1^{(k-1)}(\lambda) + \dots + P_1(\lambda)\tilde{\alpha}_{k-1}^{(k-1)}(\lambda),$$

$$\tilde{R}_k(\lambda) = P_{-1}(\lambda)\tilde{\beta}_0^{(k-1)}(\lambda) + P_{-2}(\lambda)\tilde{\beta}_1^{(k-1)}(\lambda) + \dots + P_{-k}(\lambda)\tilde{\beta}_{k-1}^{(k-1)}(\lambda),$$

$$\tilde{S}_k(\lambda) = -\tilde{F}_k(\lambda)G_{k-1}(\lambda), \quad \tilde{V}_k(\lambda) = -\tilde{R}_k(\lambda)Q_{k-1}(\lambda), \quad \tilde{U}_k(\lambda) = \frac{1}{1 - \tilde{S}_k(\lambda)\tilde{V}_k(\lambda)},$$

$$G_k(\lambda) = G_{k-1}(\lambda)\tilde{U}_k(\lambda), \quad Q_k(\lambda) = Q_{k-1}(\lambda)\tilde{U}_k(\lambda),$$

$$\begin{bmatrix} \tilde{\alpha}_0^{(k)}(\lambda) \\ \tilde{\alpha}_1^{(k)}(\lambda) \\ \dots \\ \tilde{\alpha}_{k-1}^{(k)}(\lambda) \\ \tilde{\alpha}_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(k-1)}(\lambda) \\ \tilde{\alpha}_1^{(k-1)}(\lambda) \\ \dots \\ \tilde{\alpha}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix} \tilde{S}_k(\lambda), \quad (2.14)$$

$$\begin{bmatrix} \tilde{\beta}_0^{(k)}(\lambda) \\ \tilde{\beta}_1^{(k)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-1}^{(k)}(\lambda) \\ \tilde{\beta}_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(k-1)}(\lambda) \\ \tilde{\alpha}_1^{(k-1)}(\lambda) \\ \dots \\ \tilde{\alpha}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} \tilde{V}_k(\lambda) + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix}.$$

$$D_k(\lambda) = B_k(\lambda) - P_k(\lambda)Z_0^{(k-1)}(\lambda) - P_{k-1}(\lambda)Z_1^{(k-1)}(\lambda) - \dots - P_1(\lambda)Z_{k-1}^{(k-1)}(\lambda),$$

$$W_k(\lambda) = D_k(\lambda)Q_k(\lambda),$$

$$\begin{bmatrix} Z_0^{(k)}(\lambda) \\ Z_1^{(k)}(\lambda) \\ \dots \\ Z_{k-1}^{(k)}(\lambda) \\ Z_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} Z_0^{(k-1)}(\lambda) \\ Z_1^{(k-1)}(\lambda) \\ \dots \\ Z_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix} W_k(\lambda).$$

Процес (2.14) реалізується за припущення, що всі ведучі мінори матриці відмінні від нуля і дозволяє при $k = n - 1$ знайти розв'язок системи (1.1).

Рекурентні співвідношення (2.14) служать для обчислення вектора

$$\begin{bmatrix} Z_0^{(k)}(\lambda) \\ Z_1^{(k)}(\lambda) \\ \dots \\ Z_{k-1}^{(k)}(\lambda) \\ Z_k^{(k)}(\lambda) \end{bmatrix},$$

якщо вже визначені невідомі системи рівнянь

$$A_{k-1}(\lambda)Z_{k-1}(\lambda) = B_{k-1}(\lambda) \quad (2.15)$$

При реалізації алгоритму на комп'ютері, якщо проводити обчислення за алгоритмом (2.14), виконуючи лише операції множення та додавання многочленів, може виникнути так зване явище “розбухання” проміжних даних і порядок поліномів в обчисленнях буде зростати зі швидкістю 2^{2^k} , де

k – порядок розв’язуваної відсіченої системи. Щоб уникнути цього, проведемо додаткові дослідження.

Будемо шукати $\tilde{\alpha}^{(k)}(\lambda)$ та $\tilde{\beta}^{(k)}(\lambda)$ у вигляді відношення двох поліномів з невідомими коефіцієнтами

$$\frac{\tilde{\alpha}^{(k)}(\lambda)}{C_k(\lambda)} = \begin{bmatrix} \frac{\tilde{\alpha}_0^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{\tilde{\alpha}_1^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \dots \\ \frac{\tilde{\alpha}_{k-1}^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\tilde{\beta}_0^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \dots \\ \frac{\tilde{\beta}_{k-2}^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{\tilde{\beta}_{k-1}^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \end{bmatrix} \tilde{S}_k(\lambda), \quad (2.16)$$

$$\frac{\tilde{\beta}^{(k)}(\lambda)}{C_k(\lambda)} = \begin{bmatrix} \frac{\tilde{\alpha}_0^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{\tilde{\alpha}_1^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \dots \\ \frac{\tilde{\alpha}_{k-1}^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ 0 \end{bmatrix} \tilde{V}_k(\lambda) + \begin{bmatrix} 0 \\ \frac{\tilde{\beta}_0^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \dots \\ \frac{\tilde{\beta}_{k-2}^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{\tilde{\beta}_{k-1}^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \end{bmatrix}.$$

ТЕОРЕМА 2.1. Якщо в процесі реалізації алгоритму (2.14) всі головні мінори тепліцевої матриці не дорівнюють нулю, то $\forall k=0,1,\dots,n-1$ та $i=0,1,\dots,k$ поліноми $C_{k-1}(\lambda)$ є дільниками многочленів $\tilde{\alpha}_i^{(k-1)}(\lambda) + \tilde{\beta}_i^{(k-1)}(\lambda)S_k(\lambda)$ та $\tilde{\alpha}_i^{(k-1)}(\lambda)V_k(\lambda) + \tilde{\beta}_i^{(k-1)}(\lambda)$.

Доведення. Якщо для відсічених систем (2.15) визначити невідомі за правилом Крамера та теоремою Лапласа, то $\tilde{\alpha}^{(k)}(\lambda)$, $\tilde{\beta}^{(k)}(\lambda)$ та $C_k(\lambda)$ будуть поліномами порядку $(k+1)l$. По індукції можна припустити, що $\tilde{\alpha}^{(k-1)}(\lambda)$, $\tilde{\beta}^{(k-1)}(\lambda)$ та $C_{k-1}(\lambda)$ мають порядок kl . Тоді в силу того, що $\tilde{\alpha}^{(k)}(\lambda)$ та $\tilde{\beta}^{(k)}(\lambda)$ мають порядок $(k+1)l$, многочлени $\tilde{\alpha}_i^{(k-1)}(\lambda) + \tilde{\beta}_i^{(k-1)}(\lambda)S_k(\lambda)$ та $\tilde{\alpha}_i^{(k-1)}(\lambda)V_k(\lambda) + \tilde{\beta}_i^{(k-1)}(\lambda)$ націло діляться на $C_{k-1}(\lambda)$, що й треба було довести.

З врахуванням цього факту вноситься суттєва корекція в схему обчислень за алгоритмом (2.14). На кожному кроці відразу після обчислення $\tilde{\alpha}_i^{(k)}(\lambda)$ та $\tilde{\beta}_i^{(k)}(\lambda)$ ($i = 0, \dots, k$) отримані поліноми діляться на $C_{k-1}(\lambda)$.

В загальному випадку $\deg[Z_k(\lambda)] = (k+1)l$.

Алгоритм дозволяє обчислити всі $Z_k(\lambda)$ чисельниками та знаменниками яких є поліноми степеня ln .

Для алгоритму без нормування для обчислення розв'язків СЛАР з теплицевими λ -матрицями справедливе аналогічне твердження.

ТЕОРЕМА 2.2. *Якщо в процесі реалізації обчислювального процесу за формулами (2.1), (2.12), (2.13) всі головні мінори теплицевої матриці не дорівнюють нулю, то $\forall k = 0, 1, \dots, n-1$ поліноми $C_{k-1}(\lambda)$ є дільниками многочленів*

$$\tilde{\alpha}_i^{(k-1)}(\lambda)U_k(\lambda) + \tilde{\beta}_i^{(k-1)}(\lambda)S_k(\lambda) \text{ та } \tilde{\alpha}_i^{(k-1)}(\lambda)S_k(\lambda) + \tilde{\beta}_i^{(k-1)}(\lambda)U_k(\lambda).$$

Доведення аналогічне доведенню теореми 2.2 з дослівним повторенням всіх його складових.

На кожному кроці відразу після обчислення $\tilde{\alpha}_i^{(k)}(\lambda)$ та $\tilde{\beta}_i^{(k)}(\lambda)$ $i = 0, 1, \dots, k-1$ отримані поліноми діляться на $C_{k-1}(\lambda)$.

В загальному випадку $\deg[Z_k(\lambda)] = (k+1)l$.

Проілюструємо дію алгоритму (2.14) на прикладі.

Приклад 2.2. Розв'язування системи $A(\lambda)Z(\lambda) = B(\lambda)$ з матрицею (2.3) та правою частиною $B^T(\lambda) = [-\lambda + 1 \quad 2\lambda - 2 \quad -2\lambda + 4]$.

$$\text{для } k = 0 \quad Z_0^{(0)}(\lambda) = P_0^{-1}(\lambda)B_0(\lambda) = \frac{1}{1-\lambda}(-\lambda + 1) = 1,$$

$$G_0(\lambda) = Q_0(\lambda) = P_0(\lambda) = 1 - \lambda,$$

$$\tilde{\alpha}_0^{(0)}(\lambda) = P_0^{-1}(\lambda)G_0^{-1}(\lambda) = \frac{1}{(1-\lambda)^2}, \quad \tilde{\beta}_0^{(0)}(\lambda) = P_0^{-1}(\lambda)Q_0^{-1}(\lambda) = \frac{1}{(1-\lambda)^2};$$

для $k=1$: $\tilde{F}_1(\lambda) = P_1(\lambda)\tilde{\alpha}_0^{(0)}(\lambda) = -\frac{1}{1-\lambda}$, $\tilde{R}_1(\lambda) = P_{-1}(\lambda)\tilde{\beta}_0^{(0)}(\lambda) = \frac{2}{1-\lambda}$,

$$S_1(\lambda) = -Q_0(\lambda)\tilde{F}_1(\lambda) = 1, \quad V_1(\lambda) = -G_0(\lambda)\tilde{R}_1(\lambda) = -2,$$

$$G_1(\lambda) = (1 - V_1(\lambda)S_1(\lambda))^{-1}G_0(\lambda) = \frac{1-\lambda}{3},$$

$$Q_1(\lambda) = (1 - S_1(\lambda)V_1(\lambda))^{-1}Q_0(\lambda) = \frac{1-\lambda}{3},$$

$$\begin{bmatrix} \tilde{\alpha}_0^{(1)}(\lambda) \\ \tilde{\alpha}_1^{(1)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(0)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(0)}(\lambda) \end{bmatrix} S_1(\lambda) = \begin{bmatrix} \frac{1}{(1-\lambda)^2} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{(1-\lambda)^2} \end{bmatrix} = \begin{bmatrix} \frac{1}{(1-\lambda)^2} \\ \frac{1}{(1-\lambda)^2} \end{bmatrix},$$

$$\begin{bmatrix} \tilde{\beta}_0^{(1)}(\lambda) \\ \tilde{\beta}_1^{(1)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(0)}(\lambda) \\ 0 \end{bmatrix} V_1(\lambda) + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(0)}(\lambda) \end{bmatrix} =$$

$$\begin{bmatrix} \frac{1}{(1-\lambda)^2} \\ 0 \end{bmatrix} (-2) + \begin{bmatrix} 0 \\ \frac{1}{(1-\lambda)^2} \end{bmatrix} = \begin{bmatrix} -\frac{2}{(1-\lambda)^2} \\ \frac{1}{(1-\lambda)^2} \end{bmatrix},$$

$$D_1(\lambda) = B_1(\lambda) - P_1(\lambda)Z_0^{(0)}(\lambda) = \lambda - 1,$$

$$W_1(\lambda) = Q_1(\lambda)D_1(\lambda) = -\frac{(1-\lambda)^2}{3},$$

$$\begin{bmatrix} z_0^{(1)}(\lambda) \\ z_1^{(1)}(\lambda) \end{bmatrix} = \begin{bmatrix} z_0^{(0)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} \tilde{\beta}_0^{(1)}(\lambda) \\ \tilde{\beta}_1^{(1)}(\lambda) \end{bmatrix} W_1(\lambda) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -\frac{2}{(1-\lambda)^2} \\ \frac{1}{(1-\lambda)^2} \end{bmatrix} \left(-\frac{(1-\lambda)^2}{3} \right) = \begin{bmatrix} \frac{5}{3} \\ -\frac{1}{3} \end{bmatrix}.$$

для $k=2$:

$$\tilde{F}_2(\lambda) = P_2(\lambda)\tilde{\alpha}_0^{(1)}(\lambda) + P_1(\lambda)\tilde{\alpha}_1^{(1)}(\lambda) = \frac{2-\lambda}{(1-\lambda)^2} + \frac{-1+\lambda}{(1-\lambda)^2} = \frac{1}{(1-\lambda)^2},$$

$$\tilde{R}_2(\lambda) = P_{-1}(\lambda)\tilde{\beta}_0^{(1)}(\lambda) + P_{-2}(\lambda)\tilde{\beta}_1^{(1)}(\lambda) = -\frac{2(2-2\lambda)}{(1-\lambda)^2} + \frac{1-\lambda}{(1-\lambda)^2} = -\frac{3}{1-\lambda},$$

$$S_2(\lambda) = -Q_1(\lambda)\tilde{F}_2(\lambda) = -\frac{1}{3(1-\lambda)}, \quad V_2(\lambda) = -G_1(\lambda)\tilde{R}_2(\lambda) = 1,$$

$$G_2(\lambda) = (1 - V_2(\lambda)S_2(\lambda))^{-1}G_1(\lambda) = \frac{(1-\lambda)^2}{4-3\lambda},$$

$$Q_2(\lambda) = (1 - S_2(\lambda)V_2(\lambda))^{-1}Q_1(\lambda) = \frac{(1-\lambda)^2}{4-3\lambda},$$

$$\begin{bmatrix} \tilde{\alpha}_0^{(2)}(\lambda) \\ \tilde{\alpha}_1^{(2)}(\lambda) \\ \tilde{\alpha}_2^{(2)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(1)}(\lambda) \\ \tilde{\alpha}_1^{(1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(1)}(\lambda) \\ \tilde{\beta}_1^{(1)}(\lambda) \end{bmatrix} S_2(\lambda) =$$

$$= \begin{bmatrix} \frac{1}{(1-\lambda)^2} \\ \frac{1}{(1-\lambda)^2} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2}{(1-\lambda)^2} \\ \frac{1}{(1-\lambda)^2} \end{bmatrix} \left(-\frac{1}{3(1-\lambda)} \right) = \begin{bmatrix} \frac{1}{(1-\lambda)^2} \\ \frac{(1-\lambda)^2 - 3\lambda + 5}{3(1-\lambda)^3} \\ -\frac{1}{3(1-\lambda)^3} \end{bmatrix},$$

$$\begin{bmatrix} \tilde{\beta}_0^{(2)}(\lambda) \\ \tilde{\beta}_1^{(2)}(\lambda) \\ \tilde{\beta}_2^{(2)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(1)}(\lambda) \\ \tilde{\alpha}_1^{(1)}(\lambda) \\ 0 \end{bmatrix} V_2(\lambda) + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(1)}(\lambda) \\ \tilde{\beta}_1^{(1)}(\lambda) \end{bmatrix} = \begin{bmatrix} \frac{1}{(1-\lambda)^2} \\ \frac{1}{(1-\lambda)^2} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2}{(1-\lambda)^2} \\ \frac{1}{(1-\lambda)^2} \end{bmatrix} = \begin{bmatrix} \frac{1}{(1-\lambda)^2} \\ \frac{1}{(1-\lambda)^2} \\ \frac{1}{(1-\lambda)^2} \end{bmatrix},$$

$$D_2(\lambda) = B_2(\lambda) - P_2(\lambda)Z_0^{(1)}(\lambda) - P_1(\lambda)Z_1^{(1)}(\lambda) = \frac{1}{3},$$

$$W_2(\lambda) = Q_2(\lambda)D_2(\lambda) = \frac{(1-\lambda)^2}{3(4-3\lambda)},$$

$$\begin{bmatrix} Z_0^{(2)}(\lambda) \\ Z_1^{(2)}(\lambda) \\ Z_2^{(2)}(\lambda) \end{bmatrix} = \begin{bmatrix} Z_0^{(1)}(\lambda) \\ Z_1^{(1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} \tilde{\beta}_0^{(2)}(\lambda) \\ \tilde{\beta}_1^{(2)}(\lambda) \\ \tilde{\beta}_2^{(2)}(\lambda) \end{bmatrix} W_2(\lambda) = \begin{bmatrix} \frac{5}{3} \\ -\frac{1}{3} \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{1}{(1-\lambda)^2} \\ \frac{1}{(1-\lambda)^2} \\ \frac{1}{(1-\lambda)^2} \end{bmatrix} \frac{(1-\lambda)^2}{3(4-3\lambda)} = \begin{bmatrix} \frac{-15\lambda + 21}{3(4-3\lambda)} \\ \frac{3\lambda - 5}{3(4-3\lambda)} \\ \frac{1}{3(4-3\lambda)} \end{bmatrix}.$$

Неважно перекопатись, що отриманий вектор є розв'язком системи $A(\lambda)Z(\lambda) = B(\lambda)$ з матрицею (2.3) та правою частиною $B^T(\lambda) = [-\lambda + 1 \quad 2\lambda - 2 \quad -2\lambda + 4]$.

Реалізація алгоритму (2.13) вимагає виконання n^2l^2 операцій множення та n^2l^2 операцій додавання-віднімання. Але загальні затрати, пов'язані з розв'язанням системи $A(\lambda)Z(\lambda) = B(\lambda)$, будуть включати і ту обчислювальну роботу, яка проводиться в алгоритмі (2.1) або (2.5), тобто $4n^2l^2$ операцій множення та $3n^2l^2$ операцій додавання-віднімання з використанням алгоритму (2.1) і $3n^2l^2$ операцій множення та $3n^2l^2$ операцій додавання-віднімання, якщо $\alpha^{(k)}(\lambda)$, $\beta^{(k)}(\lambda)$ обчислюються за алгоритмом (2.5).

Обчислювальна складність алгоритму з нормуванням (\bar{A}_{22}) для обчислення розв'язків СЛАР з теплицевими λ -матрицями.

Для реалізації алгоритму треба виконати (з врахуванням тих затрат, що проводяться в алгоритмі (2.5)) порядку $3n^2l^2$ операцій множення і $3n^2l^2$ операцій додавання-віднімання. Якщо ж замість алгоритму (2.5) використати алгоритм (2.1), то кількість операцій множення зросте до $4n^2l^2$. Вектор-розв'язок може бути отриманий на місці вектора-правої частини.

Оцінки точності обчислення розв'язків.

Будь-яка обчислювальна задача має *вхідні дані* та обчислений *результат*. При розв'язуванні системи лінійних алгебраїчних рівнянь (1.1) з невідродженою теплицевою λ -матрицею $A(\lambda)$ порядку n , елементами якої є алгебраїчні поліноми степеня l , вхідні дані – це $(3n - 1)l$ коефіцієнтів системи (враховуючи праві частини); шуканим розв'язком є nl -мірний вектор $Z(\lambda)$.

Лише в деяких випадках вхідні дані точно відповідають тій задачі, яку потрібно розв'язати. Початкові дані обчислювальної задачі, як правило, отримані в результаті певних вимірювань або попередніх обчислень і містять їх помилки, які неможливо усунути в процесі обчислень.

При записі вхідних даних в пам'ять ЕОМ можливі *похибки представлення*, пов'язані з тим, що для опису числа відводиться фіксована кількість розрядів. Крім того, майже завжди виникають помилки при проведенні обчислень вже в межах самої задачі (внаслідок неточного виконання арифметичних операцій). Помилки цього типу (так звані

обчислювальні помилки) в багатьох випадках в сукупності рівносильні точному розв'язку такої ж задачі, але зі зміненими вхідними даними.

Навіть за умови, що попередні вимірювання та обчислення проводились з високою точністю і для розв'язування задачі вибрано стійкий метод обчислень, помилки вхідних даних, хоча й малі, але все ж будуть.

Під *обумовленістю обчислювальної задачі* розуміють вплив малих змін вхідної інформації на її розв'язок. Якщо малим змінам вхідних даних відповідають малі зміни розв'язку, то задача *добре обумовлена*. Якщо ж можливі значні зміни розв'язку, то *задача погано обумовлена*.

Охарактеризуємо кількісно степінь обумовленості обчислювальної задачі.

Нехай $A(\lambda)$ – вектор, складений з точних вхідних даних задачі, $Z(\lambda)$ – її розв'язок. І нехай вектор $\Delta A(\lambda)$ характеризує помилки, внесені у початкові дані, а вектор $\Delta Z(\lambda)$ – зміну розв'язку, викликану цими помилками. Щоб оцінити “величину” вектора $\Delta Z(\lambda)$, поставимо у відповідність кожному вектору число, яке назвемо його довжиною

Довжину вектора з координатами $Z_1(\lambda), Z_2(\lambda), \dots, Z_n(\lambda)$ обчислимо за формулою $\|Z(\lambda)\|_2 = (Z_1^2(\lambda) + Z_2^2(\lambda) + \dots + Z_n^2(\lambda))^{1/2}$.

Для визначення міри величини вектора в обчислювальній математиці часто використовують так звані *векторні норми*. Застосуємо дві з них, які

використовуються найчастіше – $\|Z(\lambda)\|_\infty = \max_{1 \leq i \leq n} |Z_i(\lambda)|$ та $\|Z(\lambda)\|_1 = |Z(\lambda)_1| + |Z(\lambda)_2| + \dots + |Z(\lambda)_n|$.

Нехай помилки, що вносяться у вхідні дані, є малими, тобто для заданого малого додатнього числа ε і вибраної векторної норми виконується нерівність $\|\Delta A(\lambda)\| < \varepsilon$.

Нехай розв'язується система лінійних алгебраїчних рівнянь $A(\lambda)Z(\lambda) = B(\lambda)$ з невинродженою теплицевою λ -матрицею. Нехай $A^{-1}(\lambda)$ – обернена матриця для $A(\lambda)$. Ров'язок системи представимо у вигляді $Z(\lambda) = A^{-1}(\lambda)B(\lambda)$.

Продиференціюємо останню формулу, вважаючи $B(\lambda)$, $A(\lambda)$ і $A^{-1}(\lambda)$ змінними $dZ(\lambda) = dA^{-1}(\lambda)B(\lambda) + A(\lambda)dB(\lambda)$, звідки за [17]

$$dZ(\lambda) = -A^{-1}(\lambda)(dA(\lambda)Z(\lambda) - dB(\lambda)). \quad (2.17)$$

Отже, при “великій” матриці $A^{-1}(\lambda)$ помилки в розв'язку суттєво зростають порівняно з помилками $dA(\lambda)$ і $dB(\lambda)$ вхідних даних. Надамо цьому твердженню кількісну оцінку.

Зафіксуємо довільну векторну норму $\|Z(\lambda)\|$ і для будь-якої матриці

$A(\lambda)$ отримаємо *матричну норму* $\|A(\lambda)\| = \max_{x \neq 0} \frac{\|A(\lambda)Z(\lambda)\|}{\|Z(\lambda)\|}$, тобто поставимо

у відповідність кожній матриці невід'ємне число – її норму. Для векторної

норми $\|Z(\lambda)\|_\infty$ отримаємо $\|A(\lambda)\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |P_{ij}(\lambda)|$.

При виборі $\|Z(\lambda)\|_1$ маємо матричну норму $\|A(\lambda)\|_1 = \max_{1 \leq i \leq n} \sum_{j=1}^n |P_{ij}(\lambda)|$.

Обидві норми задовольняють нерівність $\|A(\lambda)\| \geq \max_{i,j} |P_{ij}(\lambda)|$, тобто матрична норма дійсно є мірою величини елементів матриці.

З останньої нерівності впливає така властивість матричної норми $\|A(\lambda)Z(\lambda)\| \leq \|A(\lambda)\| \|Z(\lambda)\|$.

З врахуванням цієї нерівності формулу (2.17) запишемо у вигляді:

$\|dZ(\lambda)\| \leq \|A^{-1}(\lambda)\| (\|dA(\lambda)\| \|Z(\lambda)\| + \|dB(\lambda)\|)$, тобто норма оберненої матриці

відіграє роль коефіцієнта підсилення помилки при переході від $A(\lambda)$, $B(\lambda)$ до розв'язку $Z(\lambda)$.

Перетворимо останню нерівність до вигляду співвідношень між відносними похибками вхідних даних $\varepsilon_{A(\lambda)} = \frac{\|dA(\lambda)\|}{\|A(\lambda)\|}$, $\varepsilon_{B(\lambda)} = \frac{\|dB(\lambda)\|}{\|B(\lambda)\|}$ і

відносною похибкою розв'язку $\varepsilon_{Z(\lambda)} = \frac{\|dZ(\lambda)\|}{\|Z(\lambda)\|}$, для якої отримаємо

$$\varepsilon_{Z(\lambda)} \leq \|A(\lambda)\| \|A^{-1}(\lambda)\| (\varepsilon_{A(\lambda)} + \varepsilon_{B(\lambda)}).$$

Отже, дійсним коефіцієнтом підсилення похибки (відносної) від введення даних до розв'язку є добуток $\|A(\lambda)\| \|A^{-1}(\lambda)\|$, а не сама лише норма оберненої матриці. Цей добуток називається *числом обумовленості* матриці $A(\lambda)$ (або системи $A(\lambda)Z(\lambda) = B(\lambda)$).

Якщо число обумовленості матриці близьке до одиниці, то можна вважати, що помилки вхідних даних не суттєво впливають на збільшення помилок розв'язку. Системи з такими властивостями називаються *добре обумовленими*. При великих значеннях числа обумовленості можливе різке збільшення похибки в розв'язку порівняно з порядком похибок в $A(\lambda)$, $B(\lambda)$. Системи $A(\lambda)Z(\lambda) = B(\lambda)$ з великим числом обумовленості називають *погано обумовленими*.

Оцінімо число обумовленості системи $A(\lambda)Z(\lambda) = B(\lambda)$ з теплицевою λ -матрицею

$$A(\lambda) = \begin{vmatrix} 1 - \lambda & 2 - 2\lambda & 1 - \lambda \\ -1 + \lambda & 1 - \lambda & 2 - 2\lambda \\ 2 - \lambda & -1 + \lambda & 1 - \lambda \end{vmatrix}$$

та правою частиною $B^T(\lambda) = [-\lambda + 1 \quad 2\lambda - 2 \quad -2\lambda + 4]$.

Для матриці $A(\lambda)$ норму запишемо $\|A(\lambda)\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |P_{ij}(\lambda)| = |-4\lambda + 4|$.

Визначник $A(\lambda)$ дорівнює $-4\lambda^3 - 30\lambda^2 - 40\lambda + 10$, а алгебраїчне доповнення $A_{23}(\lambda) = \lambda^2 - 4\lambda + 3$. Тому в оберненій матриці в позиції (2,3) знаходиться

елемент, по модулю не менший за $\frac{\lambda^2 - 4\lambda + 3}{-4\lambda^3 + 30\lambda^2 - 40\lambda + 10}$.

Отже, число обумовленості $\|A(\lambda)\| \|A^{-1}(\lambda)\|$ не менше від $\frac{-4\lambda^3 + 20\lambda^2 + 4\lambda + 12}{-4\lambda^3 + 30\lambda^2 - 40\lambda + 10}$ або $\|A(\lambda)\| \|A^{-1}(\lambda)\|$ є близьким до 1, тобто матриця $A(\lambda)$ добре обумовлена.

Якщо система $A(\lambda)Z(\lambda) = B(\lambda)$ погано обумовлена, необхідно до початку процесу її розв'язування врівноважити матрицю $A(\lambda)$, домноживши стрічки або стовпці на відповідні масштабні множники для того, щоб добитись приблизної рівності норм стрічок і стовпців. Множники вибираються у вигляді цілих степенів основи машинної арифметики, що використовується для даного обчислювального процесу. В цьому випадку врівноважування не вносить похибок в коефіцієнти системи. Праві частини множаться на ті ж множники, що й одноіменні стрічки; масштабування стовпців (тобто одноіменних невідомих) враховується шляхом обертання обчисленого розв'язку до початкових масштабів.

Стійкість числового алгоритму.

Погано обумовленою може бути не сама задача, а лише алгоритм, вибраний для її розв'язування. Якщо обчислений розв'язок значно відрізняється від точного внаслідок виконання числового алгоритму, то такий алгоритм називають *нестійким*.

При реалізації алгоритму на ЕОМ виникають похибки заокруглення даних, сумарний ефект яких необхідно враховувати при розв'язуванні задачі [19], [21].

Нехай обчислювальна задача з початковими даними $A(\lambda)$ розв'язується з допомогою деякого точного алгоритму f . Результат R розв'язування задачі запишемо у вигляді $R = f(A(\lambda))$.

При реалізації алгоритму f на ЕОМ всі його операції будуть замінені машинними псевдоопераціями, а сам алгоритм – деяким машинним алгоритмом $R = f_i(A(\lambda))$, результат виконання якого запишемо $Z_i(\lambda) = f_i(A(\lambda))$.

Різницю $\Delta = Z(\lambda)_i - Z(\lambda)$ називають *похибкою обчислення* на ЕОМ. Такий метод врахування сумарної похибки заокруглення називається *прямим аналізом похибок*. Для багатьох числових методів похибки проміжних обчислень в сукупності рівносильні випадку, коли б ті ж методи (в нашому випадку алгоритм f) точно розв'язували б кожен свою задачу, попередньо змінивши вхідні дані (наприклад, на $A_i(\lambda)$): $Z_i(\lambda) = f(A_i(\lambda))$.

Різницю $R = Z_i(\lambda) - Z(\lambda)$ називають *еквівалентним збуренням*, яке також характеризує похибку розв'язування задачі. Останню рівність запишемо у вигляді $Z_i(\lambda) = f(A(\lambda) + R)$. $Z_i(\lambda)$ можна розглядати як розв'язок тієї ж задачі із збуреними на R вхідними даними. Для отримання кількісної оцінки впливу похибок заокруглення використовують так званий *зворотній аналіз похибок* [120].

Для аналізу похибок заокруглення, що виникають на ЕОМ, використаємо методику Дж.Х.Уілкінсона [120].

Представлення чисел в ЕОМ.

При реалізації алгоритмів на ЕОМ зазвичай використовують два режими обчислень – з фіксованою та плаваючою комою. Останній є основним для ЕОМ. В цьому режимі кожне ненульове число z представляється парою чисел a та b : $z = p^b a$, де p – основа машинної

арифметики, ціле b – порядок числа, a – мантиса, $\frac{1}{p} \leq |a| \leq 1$.

Для представлення мантиси виділяється t розрядів (кількість знаків після коми для a). Найчастіше в ЕОМ використовується двійкова система числення, тому число z має t -бітову довжину.

Запис $s = fl(z * y)$ означає, що стандартне число s з плаваючою комою одержане з стандартних чисел z та y в результаті виконання на ЕОМ деякої арифметичної операції (додавання, віднімання, множення або ділення) в режимі з плаваючою комою.

Для аналізу похибок заокруглення методів, описаних в попередніх розділах, застосуємо метод зворотнього аналізу, оскільки використання прямого методу для оцінки похибок обчислювальних методів є досить трудомістким і часто призводить до огрублення загальних оцінок схеми обчислень. Запропонована Гівенсом ідея зворотнього аналізу похибок дає змогу значно спростити сам процес аналізу і отримати більш достовірні верхні оцінки якості знайдених розв'язків.

Деякі відомості з теорії похибок.

Нехай $A(\lambda)$ – прямокутна матриця, яка перетворюється в процесі реалізації деякого чисельного методу. Будемо припускати, що математичний алгоритм полягає у побудові послідовності $A(\lambda) = A_0(\lambda), A_1(\lambda), A_2(\lambda), \dots, A_N(\lambda)$, де $A_i(\lambda) = L_i(\lambda)A_{i-1}(\lambda)$ $i = 1, 2, \dots, N$ і матриці $L_i(\lambda)$ – невироджені. Якщо $L(\lambda) = L_n(\lambda)L_{n-1}(\lambda) \dots L_1(\lambda)$, то

$$A_N(\lambda) = L(\lambda)A(\lambda). \quad (2.18)$$

Отже, матриця $A_N(\lambda)$ одержується в результаті точного множення матриці $L(\lambda)$ на матрицю $A(\lambda)$.

Реальний обчислювальний процес в загальному випадку приводить до побудови такої послідовності

$$A_i(\lambda) = \hat{L}_i(\lambda)A_{i-1}(\lambda) + \mu_{i-1} \quad i = 1, 2, \dots, N. \quad (2.19)$$

$\hat{L}_i(\lambda)$ – матриці, реально одержані в процесі обчислень, μ_i – матриця похибок від множення $\hat{A}_{i-1}(\lambda)$ на $\hat{L}_i(\lambda)$. Далі одержимо

$$\begin{aligned} A_i(\lambda) &= \hat{L}_N(\lambda)A_{N-1}(\lambda) + \mu_{N-1} = \\ &= \hat{L}_N(\lambda)\hat{L}_{N-1}(\lambda)\dots\hat{L}_1(\lambda)(A(\lambda) + \hat{L}_1^{-1}(\lambda)\mu_0 + \hat{L}_1^{-1}(\lambda)\hat{L}_2^{-1}(\lambda)\mu_1 + \hat{L}_1^{-1}(\lambda)\hat{L}_2^{-1}(\lambda)\dots\hat{L}_M^{-1}(\lambda)\mu_{N-1}) \end{aligned}$$

Позначимо

$$\left. \begin{aligned} L(\lambda) &= \hat{L}_N(\lambda)\hat{L}_{N-1}(\lambda)\dots\hat{L}_1(\lambda) \\ E_N &= \hat{L}_1^{-1}(\lambda)\mu_0 + \hat{L}_1^{-1}(\lambda)\hat{L}_2^{-1}(\lambda)\mu_1 + \hat{L}_1^{-1}(\lambda)\hat{L}_2^{-1}(\lambda)\dots\hat{L}_M^{-1}(\lambda)\mu_{N-1} \end{aligned} \right\} \quad (2.20)$$

і отримаємо

$$A_N(\lambda) = L(\lambda)(A(\lambda) + E_N). \quad (2.21)$$

Аналізуючи (2.18) та (2.21) робимо висновок, що реально знайдена в процесі обчислень матриця $A_N(\lambda)$ може бути представлена як така, що одержана в результаті точного множення збуреної матриці $A(\lambda) + E_N$ на матрицю $L(\lambda)$.

Наведена схема [21] дозволяє враховувати вплив похибок заокруглення в багатьох обчислювальних алгоритмах, які побудовані на елементарних перетвореннях матриць. Але в багатьох випадках доводиться проводити оцінку похибок іншими способами. Сформулюємо один результат більш загального характеру.

ТВЕРДЖЕННЯ 2.1. *Нехай деякий масив вхідних даних $A\{P_1(\lambda), P_2(\lambda), \dots, P_n(\lambda)\}$ обробляється за алгоритмом ϕ , який складається з етапів $\phi_1, \phi_2, \dots, \phi_k$ так, що*

$$\hat{A}_i(\lambda) = \phi_i(A(\lambda)). \quad (2.22)$$

Тоді реалізації ϕ_i алгоритму ϕ на ЕОМ відповідають еквівалентні збурення γ масиву вхідних даних $A(\lambda)$ такі, що

$$\gamma = \prod_{i=1}^k (1 + \gamma^{(i)}) - 1 \quad (2.23)$$

Доведення. Використовуючи метод зворотнього аналізу заокруглення похибок, машинній реалізації відношення (2.22) поставимо у відповідність точну реалізацію (2.22) але зі збуреними вхідними даними

$$A^{(i)}(\lambda) = \varphi_i(A^{(i-1)}(\lambda)[1 + \gamma^{(i)}]) \quad (2.24)$$

З врахуванням (2.24) для реалізації першого кроку на ЕОМ запишемо: $A^{(1)}(\lambda) = \varphi_1(A^{(0)}(\lambda)[1 + \gamma^{(1)}])$. Для другого кроку $A^{(2)}(\lambda) = \varphi_2(A^{(1)}(\lambda)[1 + \gamma^{(2)}])$. Продовжуючи цей процес на k -ому кроці одержимо

$$A^{(k)}(\lambda) = \varphi_k(A^{(k-1)}(\lambda)[1 + \gamma^{(k)}]) = \varphi_k([1 + \gamma^{(k)}]\varphi_{k-1}(A^{(k-2)}(\lambda)[1 + \gamma^{(k-1)}])).$$

У випадку, коли при реалізації довільного обчислювального алгоритму φ_i виконується хоча б одна арифметична операція, для відповідного еквівалентного збурення ε мають місце співвідношення

$$(1 + \delta)\phi(Z(\lambda)) = \phi_i(Z(\lambda)[1 + \varepsilon]), \quad (2.25)$$

де $|\delta| > 0$ – відносна похибка алгоритму. Виходячи з (2.25), запишемо

$$\begin{aligned} A^{(2)}(\lambda) &= \varphi_2(A^{(1)}(\lambda)[1 + \gamma^{(2)}]) = \varphi_2([1 + \gamma^{(2)}]\varphi_1(A(\lambda)[1 + \gamma^{(1)}])) = \\ &= \varphi_2(\varphi_1(A(\lambda)[1 + \eta^{(2)}][1 + \gamma^{(1)}])), \\ A^{(3)}(\lambda) &= \varphi_3(A^{(2)}(\lambda)[1 + \gamma^{(3)}]) = \varphi_3([1 + \gamma^{(3)}]\varphi_2(\varphi_1(A(\lambda)[1 + \eta^{(2)}][1 + \gamma^{(1)}]))) = \end{aligned}$$

$$\begin{aligned}
&= \varphi_3 \varphi_2 ([1 + \gamma^{(3)}]) (\varphi_1 (A(\lambda) [1 + \eta^{(2)}] [1 + \gamma^{(1)}])) = \\
&= \varphi_3 (\varphi_2 (\varphi_1 (A(\lambda) [1 + \eta^{(3)}] [1 + \eta^{(2)}] [1 + \gamma^{(1)}])))
\end{aligned}$$

.

$$A^{(k)}(\lambda) = \varphi_k (\varphi_{k-1} (\dots \varphi_1 (A(\lambda) [1 + \eta^{(k)}] [1 + \eta^{(k-1)}] \dots [1 + \eta^{(2)}] [1 + \gamma^{(1)}])) \dots),$$

де $|\eta^{(i)}| > 0$ – відносні похибки окремих етапів алгоритмів і $\|\eta^{(i)}\| \leq \|\gamma^{(i)}\|$,

оскільки відносна похибка алгоритму в загальному випадку не може бути менша від похибок вхідних даних. Отже, якщо

$$\varphi(A(\lambda)) = \varphi_k [\varphi_{k-1} [\dots \varphi_1 [A(\lambda) (1 + \eta^{(k)}) (1 + \eta^{(k-1)}) \dots (1 + \eta^{(2)}) (1 + \gamma^{(1)})]]],$$

$$\text{то } \varphi_i (A(\lambda) (1 + \gamma)) = \varphi_k (\varphi_{k-1} (\dots \varphi_1 (\prod_{i=2}^k [1 + \eta^{(i)}] [1 + \gamma^{(1)}])).$$

$$\text{Звідки отримаємо } \gamma = \prod_{i=2}^k [1 + \eta^{(i)}] [1 + \gamma^{(1)}] - 1.$$

З урахуванням того, що $\|\eta^{(i)}\| \leq \|\gamma^{(i)}\|$ на основі останньої рівності маємо

$$\begin{aligned}
\|\gamma\|_l &\leq \prod_{i=2}^k (1 + \|\eta^{(i)}\|_l) (1 + \|\gamma^{(1)}\|_l) - 1 \leq \\
&\leq \prod_{i=2}^k (1 + \|\gamma^{(i)}\|_l) (1 + \|\gamma^{(1)}\|_l) - 1 = \prod_{i=1}^k (1 + \|\gamma^{(i)}\|_l) - 1.
\end{aligned}$$

Що й треба було довести.

Наслідок з твердження 2.1. Якщо при реалізації деякого обчислювального алгоритму ϕ_i на ЕОМ виконується хоча б одна арифметична операція на кожному з k етапів, так що

$$\left. \begin{aligned}
(1 + \delta)\phi(x) &= \phi_i(x[1 + \varepsilon]) \\
\|\varepsilon\| &\leq \|\delta\|
\end{aligned} \right\}, \quad (2.26)$$

то для сумарної похибки заокруглення на ЕОМ δ і відповідного

еквівалентного збурення ε мають місце співвідношення $\|\alpha\| \leq \prod_{i=1}^k (1 + \|\gamma^{(i)}\|) - 1$.

Проведемо аналіз чисельної стійкості раніше запропонованих алгоритмів засобами зворотнього аналізу похибок. Для цього використаємо деякі результати, одержані Дж.Х.Уілкінсоном [120], В.В.Воєводіним [21] та іншими дослідниками стійкості обчислювальних методів. Сформулюємо їх без доведення.

При аналізі похибок заокруглення часто доводиться проводити оцінки вигляду

$$(1 - \xi^{-t})^r \leq 1 + \varepsilon \leq (1 + \xi^{-t})^r \quad (2.27)$$

Для спрощення користування цими формулами у всіх практичних застосуваннях припускається, що величина r буде обмежена зверху умовою $r\varepsilon^{-t} < 0.1$. Це обмеження для кожного прийняттого значення t спричинено в першу чергу обмеженістю оперативної пам'яті ЕОМ [120]. Із врахуванням (2.27) можна записати

$$\left. \begin{aligned} (1 + \xi^{-t})^r &< 1 + 1.06 \cdot \xi^{-t} \\ (1 - \xi^{-t})^r &> 1 - 1.06 \cdot \xi^{-t} \end{aligned} \right\} \quad (2.28)$$

Аналіз похибок заокруглення для алгоритмів розв'язування СЛАР з теплицевими λ -матрицями.

Розглянемо випадок, коли для накопичення скалярних добутків застосовується арифметика з плаваючою комою. Використаємо оцінки похибок для арифметичних операцій при обчисленні невідомих.

Визначення частки та остачі від ділення алгебраїчних поліномів $B_0(\lambda)$ на $P_0(\lambda)$ степеня не вище l для отримання $Z_0^{(0)}(\lambda) = \frac{B_0(\lambda)}{P_0(\lambda)}$ може бути реалізовано за алгоритмом Евкліда. Тоді для обчислення елементів $Z_k^{(k)}(\lambda)$ на ЕОМ в системі з плаваючою комою зі звичайною точністю матимемо

$$\begin{aligned}
f_l(Z_k^{(k)}) &= f_l \left[b_k^{(k-1)} - \frac{p_s}{p_0} b_{k-1}^{(k-1)} \right] \equiv (1 + \varepsilon'_k) \left\{ f_l[b_k^{(k-1)}] - \frac{p_s}{p_0} (1 + \beta_k)(1 + \alpha_k) f_l[b_{k-1}^{(k-1)}] \right\} \equiv \\
&\equiv (1 + \varepsilon'_k) f_l[b_k^{(k)}] - \frac{p_s}{p_0} f_l[b_{k-1}^{(k-1)}] (1 + \varepsilon''_k), \\
&\begin{cases} \|\varepsilon'_i\| \leq \xi, & \text{при } b_k^{(k-1)} \neq 0; \\ \varepsilon'_i = -1, & \text{при } b_k^{(k-1)} = 0; \\ \|\varepsilon''_i\| \leq 3 \times 2.01 \xi, & \text{при } b_{k-1}^{(k-1)} \neq 0; \\ \varepsilon''_i = -1, & \text{при } b_{k-1}^{(k-1)} = 0. \end{cases}
\end{aligned}$$

А для режиму з накопиченням маємо

$$f_l(Z_k^{(k)}) = f_l \left[b_k^{(k-1)} - \frac{p_s}{p_0} b_{k-1}^{(k-1)} \right] \equiv (1 + \varepsilon''_{k-1}) f_l[b_k^{(k-2)}] - \frac{p_{s-1}}{p_0} f_l[b_{k-1}^{(k-2)}] (1 + \varepsilon''_{k-1})$$

де

$$\begin{cases} \|\varepsilon'_{k-1}\| \leq \xi, & \text{при } b_{k-1}^{(k-2)} \neq 0; \\ \varepsilon'_{k-1} = -1, & \text{при } b_{k-1}^{(k-2)} = 0; \\ \|\varepsilon''_{k-1}\| \leq 3 \times 2.01 \xi, & \text{при } b_{k-1}^{(k-2)} \neq 0; \\ \varepsilon''_{k-1} = -1, & \text{при } b_{k-1}^{(k-2)} = 0. \end{cases}$$

$$\begin{aligned}
\text{І далі } f_l(b_k^{(k)}) &= (1 + \varepsilon'_k) f_l[b_k^{(k-1)}] - \frac{p_s}{p_0} f_l[b_{k-1}^{(k-1)}] (1 + \varepsilon''_k) = \\
&\equiv (1 + \varepsilon'_k) \left\{ (1 + \varepsilon'_{k-1}) f_l[b_k^{(k-2)}] - \frac{p_{s-1}}{p_0} f_l[b_{k-1}^{(k-2)}] (1 + \varepsilon''_{k-1}) \right\} - \\
&- \frac{p_s}{p_0} (1 + \varepsilon''_k) \left\{ (1 + \mu'_{k-1}) f_l[b_{k-1}^{(k-2)}] - \frac{p_{s-1}}{p_0} f_l[b_{k-2}^{(k-2)}] (1 + \mu''_{k-1}) \right\} = \\
&= (1 + \gamma'_k) f_l[b_k^{(k-2)}] - \left[\frac{p_{s-1}}{p_0} + \frac{p_s}{p_1} \right] (1 + \gamma''_k) f_l[b_{k-1}^{(k-2)}] + (1 + \gamma'''_k) \frac{p_s p_{s-1}}{p_2} f_l[b_{k-2}^{(k-2)}],
\end{aligned}$$

$$\text{де } |\gamma'_k| \leq 2 \cdot 2.01 \xi^{-t+1}; |\gamma''_k| \leq 4 \cdot 2.01 \xi^{-t+1}; |\gamma'''_k| \leq 4 \cdot 2.01 \xi^{-t+1}.$$

Продовжуючи процес далі, можна одержати функціональну залежність

$$fl[b_s^{(k)}] = \phi[b_0(1 + \varepsilon_0), b_1(1 + \varepsilon_1), \dots, b_n(1 + \varepsilon_n)],$$

де
$$|\varepsilon_k| \leq C_k 2.01 \cdot 3 \cdot \xi^{-t+1}.$$

Згідно досліджень Дж.Х.Уілкінсона [94] константа C_k задовольняє співвідношенню $|C_k| \leq \xi^n$.

При обчисленні добутку двох поліномів степеня не вище l в режимі з плаваючою комою зі звичайною точністю для еквівалентних збурень отримаємо

$$fl[W_k(\lambda)] = fl[D_k(\lambda)Q_k(\lambda)] = \sum_{i=0}^{2l} \lambda^i \sum_{j=0}^l (1 + \varepsilon_{ij}) d_j q_{i-j} = fl\left[\sum_{i=0}^{2l} \lambda^i w_i (1 + \gamma_i)\right],$$

де, згідно з результатами Дж.Х.Уілкінсона [94] і В.В.Воєводіна [21]

$$\begin{cases} |\gamma_i| \leq 2.01(2l+1)\xi^{-t+1}, & \text{при } w_i \neq 0, \\ \gamma_i = -1, & \text{при } w_i = 0, \\ |\varepsilon_{ij}| \leq \xi^{-t+1}, & \text{при } d_j \beta_{i-j} \neq 0, \\ \varepsilon_{ij} = -1, & \text{при } d_j \beta_{i-j} = 0. \end{cases}.$$

Для суми двох поліномів при обчисленні в режимі з плаваючою комою зі звичайною точністю маємо

$$fl[Z_i^k(\lambda)] = fl[Z_i^{k-1}(\lambda) + \hat{\beta}_i^{k-1}(\lambda)] = \sum_{m=0}^M (1 + \varepsilon_m) (z_m^k + \check{\beta}_m^k) \lambda^{M-m}$$

де
$$\begin{cases} |\varepsilon_m| \leq \xi^{-t+1}, & \text{при } z_m^k + \check{\beta}_m^k \neq 0; \\ \varepsilon_m = -1 & , \text{ при } z_m^k + \check{\beta}_m^k = 0. \end{cases}$$

Для еквівалентних збурень, що виникають при обчисленні скалярних добутків (2.15), мають місце оцінки $\|\varepsilon_m\| \leq C_m n \max_{-r \leq k \leq r} \{ \|P_k(\lambda)\| \} \xi^{-t+1}$, де $|C_m| \leq \xi^{-t+1}$ і для режиму з накопиченням $\|\varepsilon_m\| \leq C_m 2,01 \max_{-r \leq k \leq r} \{ \|P_k(\lambda)\| \} \xi^{-t+1}$.

ТЕОРЕМА 2.3. *Якщо матриця $A(\lambda)$ системи (1.1) з поліноміальними елементами $P_k(\lambda) = \sum_{i=0}^{\ell} a_i \lambda^{\ell-i}$ не вироджена, то при реалізації алгоритму розв'язування СЛАР з теплицевими λ -матрицями на ЕОМ в режимі плаваючої коми зі звичайною точністю для еквівалентних збурень має місце нерівність*

$$\|\xi_{ij}\| \leq 2,1nl \max_{-l \leq k \leq l} \{ \|P_k(\lambda)\| \} \xi^{-t+1} \quad i, j = \overline{0, n-1}. \quad (2.29)$$

Доведення. Нехай вже обчислено всі $W_i(\lambda)$ та $Z_i(\lambda)$ для $i = \overline{0, k-1}$. Тоді для невідомих $W_k(\lambda)$, $Z_k(\lambda)$ при обчисленнях з плаваючою комою можна записати

$$\begin{aligned} \mathcal{H}[D_k(\lambda)] &\equiv \hat{B}_k(\lambda)(1 + \hat{\varepsilon}_k) - \hat{P}_k(\lambda) \hat{Z}_0^{(k-1)}(\lambda)(1 + \hat{\varepsilon}_{k,0}) - \\ &- \hat{P}_{k-1}(\lambda) \hat{Z}_1^{(k-1)}(\lambda)(1 + \hat{\varepsilon}_{k-1,1}) - \dots - \hat{P}_1(\lambda) \hat{Z}_{k-1}^{(k-1)}(\lambda)(1 + \hat{\varepsilon}_{1,k-1}), \\ \mathcal{H}[Z_i^k(\lambda)] &= \mathcal{H}[\hat{Z}_i^{k-1}(\lambda) + \hat{\beta}_i^{k-1}(\lambda) \hat{D}_k(\lambda)] \equiv \left. \begin{aligned} &\hat{Z}_i^{k-1}(\lambda)(1 + \hat{\varepsilon}_{i,k-1}) + \hat{\beta}_i^{k-1}(\lambda) \hat{D}_k(\lambda)(1 + \hat{\varepsilon}_{i,k-1}) \end{aligned} \right\}. \end{aligned}$$

Для еквівалентних збурень, що виникають при обчисленні скалярних добутків (2.15), мають місце оцінки

$$\|\hat{\varepsilon}_{ik}\| \leq 2,01 \max_{-l \leq k \leq l} \{ \|A_k(\lambda)\| \} (2nl) \xi^{-t+1}.$$

З урахуванням цього факту, а також нерівностей Дж.Х.Уілкінсона [94]

$$\begin{cases} (1 + \xi^{-t})^r < 1 + 1,06 \cdot \xi^{-t}, \\ (1 - \xi^{-t})^r > 1 - 1,06 \cdot \xi^{-t} \end{cases}$$

$\forall nl > 0, \forall t > 0$, таких, що $nl\xi^{-t} < 0.1$ має місце нерівність

$$f[Z_i^k(\lambda)] \equiv \{\hat{Z}_i^{k-1}(\lambda) + \hat{\beta}_i^{k-1}(\lambda)\hat{W}_k(\lambda)\}(1 + \hat{\varepsilon}_{i,k}),$$

$$\text{де } \|\hat{\xi}_{ij}\| \leq \left(1 + \|\hat{\varepsilon}_{i,k}\|\right) \left(1 + \|\hat{\varepsilon}_k\|\right) - 1 \leq 4.02 \max_{-l \leq k \leq l} \{\|A_k\|\} (2nl)\xi^{-t+1}.$$

Далі отримаємо

$$\|\hat{\xi}_{ij}\| \leq 4.02 \max_{-l \leq k \leq l} \{\|A_k(\lambda)\|\} (2nl)\xi^{-t+1} = 8.04 \max_{-l \leq k \leq l} \{\|A_k(\lambda)\|\} nl\xi^{-t+1}.$$

Теорему доведено.

ТЕОРЕМА 2.4. Якщо матриця $A(\lambda)$ системи (1.1) з поліноміальними елементами $P_k(\lambda) = \sum_{i=0}^{\ell} a_i \lambda^{\ell-i}$ невироджена, то при реалізації алгоритму розв'язування СЛАР з теплицевими λ -матрицями на ЕОМ в режимі плаваючої коми обчислення з подвійною точністю еквівалентні збурення задовольняють нерівність

$$\|\xi_{ij}\| \leq 2,1 \max_{-l \leq k \leq l} \{\|P_k(\lambda)\|\} \xi^{-t+1} \quad (i, j = \overline{0, n-1}). \quad (2.30)$$

Доведення. Нехай для накопичення скалярних добутків використовується режим обчислення ЕОМ з подвійною точністю $fi(\bullet)$. Тоді для k -го кроку обчислень маємо

$$f[D_k(\lambda)] \equiv \{\hat{B}_k(\lambda) - \hat{P}_k(\lambda)\hat{Z}_0^{(k-1)}(\lambda) - \hat{P}_{k-1}(\lambda)\hat{Z}_1^{(k-1)}(\lambda) - \dots - \hat{P}_1(\lambda)\hat{Z}_{k-1}^{(k-1)}(\lambda)\}(1 + \hat{\varepsilon}_i)$$

$$f[Z_i^k(\lambda)] = f[\hat{Z}_i^{k-1}(\lambda) + \hat{\beta}_i^{k-1}(\lambda)\hat{D}_k(\lambda)] \equiv \{\hat{Z}_i^{k-1}(\lambda) + \hat{\beta}_i^{k-1}(\lambda)\hat{D}_k(\lambda)\}(1 + \hat{\gamma}_{i,k-1}).$$

Для еквівалентних збурень, що виникають при такому способі обчислення скалярних добутків (2.15), мають місце оцінки

$$\left\| \hat{\varepsilon}_i \right\| \leq \max_{-l \leq s \leq l} \{ \|A_k(\lambda)\| \} \xi^{-t+1}, \quad \left\| \hat{\gamma}_j \right\| \leq 2 \times 2.01 \max_{-l \leq k \leq l} \{ \|A_k(\lambda)\| \} \xi^{-t+1}$$

Таким чином, реалізації кожного кроку алгоритму на ЕОМ відповідають невеликі відносні еквівалентні збурення порядку ξ^{-t+1} . Але алгоритм в загальному випадку не забезпечує обмеження росту коефіцієнтів поліномів $Z_i^{(k)}(\lambda)$. В зв'язку з цим можливий експоненціальний ріст абсолютних еквівалентних збурень γ_j (гіпотеза Уілкінсона [94] для $l = 0$).

Реально обчисленим значенням $Z_i^{(k)}(\lambda)$ відповідають еквівалентні збурення елементів вхідної матриці, що не перевищують за абсолютною величиною $2 \cdot 1.06 \cdot \xi^{-t+1}$.

Оцінки такого порядку, як (2.29), (2.30), мають місце і для алгоритму з нормуванням розв'язування систем лінійних алгебраїчних рівнянь з теплицевими λ -матрицями.

Побудова вище описаних алгоритмів можлива лише при припущенні, що всі ведучі мінори вхідної матриці відмінні від нуля. Тому внаслідок цього обмеження не забезпечується розв'язування системи (1.1) у випадку будь-якої невинродженої матриці $A(\lambda)$. Щоб отримати такий алгоритм, треба звільнитись від умов, накладених на ведучі мінори.

Будемо будувати послідовність розв'язків відсічених систем (2.15), але лише для тих значень k , які відповідають невинродженим підматрицям $A_k(\lambda)$. Маючи вектор $Z^{(k)}(\lambda)$ – розв'язок системи (1.1) для невинродженої підматриці $A_{k-1}(\lambda)$, спробуємо здійснити “стрибок через ненульові мінори” і в результаті отримати вектор $Z^{(k^1)}(\lambda)$ – розв'язок системи (1.1) для невинродженої матриці

$A_{k^1}(\lambda)$, який володіє такою властивістю: для всіх цілих m , які знаходяться між $k-1$ і k^1 , матриці $A_m(\lambda)$ вироджені.

Нехай $k-1 < k^1$, $A_{k-1}(\lambda)$ – невироджена матриця і $A_k(\lambda)$ – вироджена. Існує деяка нетривіальна лінійна комбінація стовпців матриці $A_k(\lambda)$, рівна ненульовому вектору. Дефект матриці $A_k(\lambda)$ дорівнює 1, тому коефіцієнти лінійної комбінації визначаються однозначно з точністю до спільного множника. Коефіцієнт при j -ому стовпці позначимо $\omega_{j-1}(\lambda)$ і утворимо вектор $\omega(\lambda) = [\omega_0(\lambda) \ \omega_1(\lambda) \ \dots \ \omega_k(\lambda)]^T$.

Координата $\omega_k(\lambda)$ не може дорівнювати нулю, бо інакше буде нульовою нетривіальна лінійна комбінація стовпців матриці $A_{k-1}(\lambda)$, що суперечить її невиродженості. Будемо вважати, що $\omega_k(\lambda) = 1$. За цієї умови нормування вектор $\omega(\lambda)$ визначається однозначно.

Складемо лінійну комбінацію стовпців матриці $A(\lambda)$, причому j -й стовпець візьмемо з нульовим коефіцієнтом, якщо $k+1 < j \leq n$, і з коефіцієнтом, рівним $\omega_{j-1}(\lambda)$, якщо $1 \leq j \leq k+1$. В результаті отримаємо вектор-стовпець довжиною n з координатами $\delta_0(\lambda), \delta_1(\lambda), \dots, \delta_{n-1}(\lambda)$, де наперед задано $\delta_m(\lambda) = 0$ при $0 \leq m \leq k$. Але й інші координати можуть бути рівними нулю. Нехай ціле число k_+ таке, що $\delta_{k_+}(\lambda) \neq 0$ і $\delta_m(\lambda) = 0$ для всіх $0 \leq m \leq k_+$. Тобто k_+ – індекс першої ненульової координати. Для будь-якого m , де $k \leq m \leq k_+$, матриця $A_m(\lambda)$ – вироджена, оскільки нетривіальна лінійна комбінація її стовпців дорівнює нулю.

Складемо ще одну лінійну комбінацію стовпців матриці $A(\lambda)$, але тепер J -й стовпець візьмемо з нульовим коефіцієнтом, якщо $1 \leq j \leq n - (k+1)$, і з коефіцієнтом, рівним $\omega_{j^1-1}(\lambda)$, якщо $n - (k+1) < j \leq n$, $j^1 = j - n + k + 1$. В результаті отримаємо вектор-стовпець довжиною n , і його координати, починаючи з першої і закінчуючи останньою, позначимо

$\gamma_{n-1}(\lambda), \gamma_{n-2}(\lambda), \dots, \gamma_0(\lambda)$. Нехай ціле число k , таке, що $\gamma_{k_-}(\lambda) \neq 0$ і $\gamma_m(\lambda) = 0$ для $0 \leq m \leq k_-$, тобто k_- – індекс останньої ненульової координати даного вектор-стовпця, $k \leq k_-$. В той же час, оскільки $A(\lambda)$ є тепліцевою для будь-якого m , де $mk \leq m < k_-$, матриця $A_m(\lambda)$ вироджена.

Отже, для m , що задовольняє нерівності $k \leq m < \max(k_+, k_-)$, матриця $A_m(\lambda)$ – вироджена.

ЛЕМА 2.2. Припустимо, що для тепліцевої матриці $A(\lambda)$ ведуча підматриця $A_k(\lambda)$ вироджена, лінійна комбінація її стовпців з коефіцієнтами $\omega_0(\lambda), \omega_1(\lambda), \dots, \omega_k(\lambda)$ нетривіальна і дорівнює нулю. Розглянемо два вектор-стовпці

$$\begin{bmatrix} \delta_0(\lambda) \\ \delta_1(\lambda) \\ \dots \\ \delta_{n-1}(\lambda) \end{bmatrix} = A(\lambda) \begin{bmatrix} \omega_0(\lambda) \\ \omega_1(\lambda) \\ \dots \\ \omega_k(\lambda) \\ 0 \\ \dots \\ 0 \end{bmatrix}, \quad \begin{bmatrix} \gamma_0(\lambda) \\ \gamma_1(\lambda) \\ \dots \\ \gamma_{n-1}(\lambda) \end{bmatrix} = A(\lambda) \begin{bmatrix} 0 \\ 0 \\ \dots \\ \omega_0(\lambda) \\ \omega_1(\lambda) \\ \dots \\ \omega_k(\lambda) \end{bmatrix}$$

і припустимо, що існують цілі числа k_+ і k_- , такі, що

$$\delta_{k_+}(\lambda) \neq 0, \gamma_{k_-}(\lambda) \neq 0, \delta_i(\lambda) = 0, \gamma_j(\lambda) = 0 \quad \text{при } 0 \leq i \leq k_+, 0 \leq j \leq k_-.$$

Покладемо $k^1 = k_+ + k_- - k - 1$. Тоді для будь-якого m , що задовольняє нерівності $k \leq m < k^1$, матриця $A_m(\lambda)$ – невироджена. Крім того, якщо матриця $A_{k-1}(\lambda)$ – невироджена або ж $k = 0$, то матриця $A_{k^1}(\lambda)$ – невироджена, за умови, що $k^1 \leq n - 1$.

Наслідок з лемми 2.2. Для симетричної тепліцевої матриці в умовах попередньої лемми $k_+ = k_-$.

В обчислювальному плані ця лема доводить можливість “стрибка через нульові мінори”, про який говорилось вище.

Для реалізації алгоритму обчислення оберненої матриці зі “стрибками” за умови, що початкова матриця не має двох сусідніх мінорів, одночасно рівних нулю, і кількість мінорів не залежить від порядку матриці, необхідно виконати (з точністю до головного члена) таку ж кількість операцій, як і в алгоритмі (2.13). Якщо ж кількість нульових ведучих мінорів залежить від порядку вхідної матриці, то в алгоритмі зі “стрибками” буде виконано не більше $\frac{13}{6}l^2n^2$ множень. Кількість операцій додавання-віднімання така сама.

Якщо нульові і ненульові мінори чергуються, то достатньо виконання $2l^2n^2$ операцій множення і такого ж числа операцій додавання-віднімання.

“Сtribок” через m нульових мінорів пов’язаний з виконанням не більше, ніж $3kl^2 + 7kml^2 + \frac{m^2l^2}{2}$ операцій множення і такої ж кількості операцій додавання-віднімання. У випадку невірності ведучих підматриць при реалізації $m+1$ кроку в процесі (2.13) необхідно було б виконати $4kl^2 + 4kml^2 + 2m^2l^2$ операцій множення і такого ж числа операцій додавання-віднімання.

Отже, при достатньо великій кількості розташованих підряд нульових мінорів алгоритм зі “стрибком” потребує обчислювальних затрат в чотири рази менших, ніж звичайний алгоритм.

В Додатку описані програма, яка реалізує алгоритм з нормуванням (\bar{A}_{22}) обчислення розв’язку системи лінійних алгебраїчних рівнянь з теплицевою λ -матрицею і результати розв’язування тестових задач.

№ тестової задачі	$\lambda = -0.01$	$\lambda = 0$	$\lambda = 0.01$
	$z(\lambda)$	$z(\lambda)$	$z(\lambda)$
1	$\begin{bmatrix} 1,74 \\ -0,42 \\ 0,08 \end{bmatrix}$	$\begin{bmatrix} 1,75 \\ -0,42 \\ 0,08 \end{bmatrix}$	$\begin{bmatrix} 1,75 \\ -0,4 \\ 0,09 \end{bmatrix}$

5	$\begin{bmatrix} 0,99 \\ 0 \\ 0 \\ 0,99 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1,1 \\ 0 \\ 0 \\ 1,1 \end{bmatrix}$
6	$\begin{bmatrix} 1,1 \\ -1,1 \\ -1,1 \\ 1,1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 0,9 \\ 1,01 \\ 0,9 \\ 1,01 \end{bmatrix}$
7	$\begin{bmatrix} 0,98 \\ 0,98 \\ 0,98 \\ 0,98 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1,02 \\ 1,02 \\ 1,02 \\ 1,02 \end{bmatrix}$

Таблиця 1. Результати тестування щодо точності обчислення розв'язків СЛАР з тепліцевою λ -матрицею при різних значеннях параметра λ

2.3. Алгоритм розв'язування СЛАР з тепліцевими матрицями з тригонометричними поліноміальними елементами

Розв'язується СЛАР вигляду (1.1) з тепліцевою λ -матрицею, де елементи $A(\lambda)$ та вектора $B(\lambda)$ є тригонометричними поліномами степеня l

$$P(\lambda) = T(\lambda) = \frac{a_0}{2} + \sum_{s=1}^l a_s \cos s\lambda + b_s \sin s\lambda$$

Нехай значення для λ та коефіцієнти тригонометричних многочленів беруться з деякого поля $F^{m \times n}$ так, що елементи матриці $A(\lambda)$ обчислюються для деякого часткового значення λ , наприклад $\lambda = \lambda_0 \in F^{m \times n}$.

Для розв'язування СЛАР з тригонометричною тепліцевою матрицею застосуємо ідею методу Є.Тиртишнікова для тепліцевих матриць з числовими елементами [21], узагальнивши його на випадок тригонометричної λ -матриці. Розроблений алгоритм базується на схемі

знаходження оберненої матриці за допомогою обчислення її першого та останнього стовпців без відшукування її інших елементів.

Розглянемо відсічені системи

$$A_k(\lambda)Z^{(k)}(\lambda) = B^{(k)}(\lambda),$$

$$\text{де } A_k(\lambda) = \begin{vmatrix} P_0(\lambda) & P_{-1}(\lambda) & P_{-2}(\lambda) & \dots & P_{-k}(\lambda) \\ P_1(\lambda) & P_0(\lambda) & P_{-1}(\lambda) & \dots & P_{-k+1}(\lambda) \\ P_2(\lambda) & P_1(\lambda) & P_0(\lambda) & \dots & P_{-k+2}(\lambda) \\ \dots & \dots & \dots & \dots & \dots \\ P_k(\lambda) & P_{k-1}(\lambda) & P_{k-2}(\lambda) & \dots & P_0(\lambda) \end{vmatrix}, \quad Z^{(k)}(\lambda) = \begin{bmatrix} Z_0^{(k)}(\lambda) \\ Z_1^{(k)}(\lambda) \\ \dots \\ Z_{k-1}^{(k)}(\lambda) \\ Z_k^{(k)}(\lambda) \end{bmatrix},$$

$$B^{(k)}(\lambda) = \begin{bmatrix} B_0(\lambda) \\ B_1(\lambda) \\ \dots \\ B_{k-1}(\lambda) \\ B_k(\lambda) \end{bmatrix}.$$

Будемо вважати, що в матриці $A(\lambda)$ всі ведучі мінори відмінні від 0 . Це рівносильно невиродженості ведучих підматриць

Позначимо через вектори $\alpha^{(k)}(\lambda)$ та $\beta^{(k)}(\lambda)$, $k=1, \dots, n-1$ відповідно перший та останній стовпці оберненої матриці для кожної з відсічених систем $A_k(\lambda)Z^{(k)}(\lambda) = B^{(k)}(\lambda)$, $k=1, \dots, n-1$.

Вектор $B^{(n-1)}(\lambda)$ співпадає з $B(\lambda)$, отже, $Z^{(n-1)}(\lambda)$ є шуканим розв'язком

$Z(\lambda)$. $Z^{(0)}(\lambda)$ визначається як $Z_0^{(0)}(\lambda) = \frac{B_0(\lambda)}{P_0(\lambda)}$. Якщо вже відомий вектор

$Z^{(k-1)}(\lambda)$ при $1 \leq k \leq n-1$, то $Z^{(k)}(\lambda)$ будемо шукати у вигляді

$$\begin{bmatrix} Z_0^{(k)}(\lambda) \\ Z_1^{(k)}(\lambda) \\ \dots \\ Z_{k-1}^{(k)}(\lambda) \\ Z_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} Z_0^{(k-1)}(\lambda) \\ Z_1^{(k-1)}(\lambda) \\ \dots \\ Z_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix} W_k(\lambda), \quad (2.31)$$

$$W_k(\lambda) = D_k(\lambda)Q_k(\lambda) \quad Wk(\lambda) = Dk(\lambda)Qk(\lambda), \quad D_k(\lambda) = B_k(\lambda) - \sum_{l=k}^{k-1} P_l(\lambda)Z_{l-k}^{(k-1)}(\lambda),$$

$$\tilde{F}_k(\lambda) = \sum_{l=k}^{k-1} P_l(\lambda)\tilde{\alpha}_{l-k}^{(k-1)}(\lambda), \quad \tilde{R}_k(\lambda) = \sum_{l=1}^{k-1} P_{-l}(\lambda)\tilde{\beta}_{l-1}^{(k-1)}(\lambda),$$

$$\tilde{S}_k(\lambda) = -\tilde{F}_k(\lambda)G_{k-1}(\lambda), \quad \tilde{V}_k(\lambda) = -\tilde{R}_k(\lambda)Q_{k-1}(\lambda), \quad \tilde{U}_k(\lambda) = \frac{1}{1 - \tilde{S}_k(\lambda)\tilde{V}_k(\lambda)},$$

$$G_k(\lambda) = G_{k-1}(\lambda)\tilde{U}_k(\lambda), \quad Q_k(\lambda) = Q_{k-1}(\lambda)\tilde{U}_k(\lambda),$$

$$\begin{bmatrix} \tilde{\alpha}_0^{(k)}(\lambda) \\ \tilde{\alpha}_1^{(k)}(\lambda) \\ \dots \\ \tilde{\alpha}_{k-1}^{(k)}(\lambda) \\ \tilde{\alpha}_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(k-1)}(\lambda) \\ \tilde{\alpha}_1^{(k-1)}(\lambda) \\ \dots \\ \tilde{\alpha}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix} \tilde{S}_k(\lambda),$$

$$\begin{bmatrix} \tilde{\beta}_0^{(k)}(\lambda) \\ \tilde{\beta}_1^{(k)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-1}^{(k)}(\lambda) \\ \tilde{\beta}_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(k-1)}(\lambda) \\ \tilde{\alpha}_1^{(k-1)}(\lambda) \\ \dots \\ \tilde{\alpha}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} \tilde{V}_k(\lambda) + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix},$$

На кожному кроці мають місце такі рівності

$$\alpha_0^{(k)}(\lambda) = \beta_k^{(k)}(\lambda) \quad \text{та} \quad \alpha_k^{(k)}(\lambda) = \beta_0^{(k)}(\lambda) \quad (k=1,2,\dots,n),$$

що дещо зменшує обчислювальні затрати, необхідні для реалізації алгоритму.

Нормувальні множники виберемо таким чином $G_0(\lambda) = Q_0(\lambda) = \frac{1}{P_0(\lambda)}$.

Їх рівність забезпечить співпадання величин $G_i(\lambda)$, $Q_i(\lambda)$ на кожному кроці [21], що значно зменшить обчислювальні затрати.

$$\text{Для } k=0 \text{ маємо } \tilde{\alpha}_0^{(0)}(\lambda) = \frac{1}{P_0(\lambda)G_0(\lambda)}, \quad \tilde{\beta}_0^{(0)}(\lambda) = \frac{1}{P_0(\lambda)Q_0(\lambda)}.$$

При реалізації алгоритму на комп'ютері, якщо проводити обчислення за алгоритмом (2.31), виконуючи лише операції множення та додавання многочленів, може виникнути так зване явище “розбухання” проміжних

даних і порядок поліномів в обчисленнях буде наростати з швидкістю 2^{2^k} , де k – порядок розв’язуваної відсіченої системи.

Тому будемо шукати розв’язок у вигляді співвідношення двох поліномів з невідомими коефіцієнтами

$$\frac{Z^{(k)}(\lambda)}{C_k(\lambda)} = \begin{bmatrix} \frac{z_0^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{z_1^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \dots \\ \frac{z_{k-1}^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\tilde{\beta}_0^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \dots \\ \frac{\tilde{\beta}_{k-2}^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{\tilde{\beta}_{k-1}^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{\tilde{\beta}_{k-1}^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \end{bmatrix} W_k(\lambda).$$

ТЕОРЕМА 2.5. *Якщо в процесі реалізації алгоритму (2.31) всі головні мінори тепліцевої матриці не дорівнюють нулю, то $\forall k = 0, 1, \dots, n-1$ та $i = 0, 1, \dots, k$ поліноми $C_{k-1}(\lambda)$ є дільниками многочлена $Z^{(k-1)}(\lambda) + \tilde{\beta}_i^{(k-1)}(\lambda)W_k(\lambda)$.*

Доведення аналогічне до доведення теореми 2.4.

З врахуванням цього факту вноситься суттєва корекція в схему обчислень за алгоритмом (2.31). На кожному кроці відразу після обчислення $Z_i^{(k-1)}(\lambda)$, $k = 0, 1, \dots, n-1$, $i = 0, \dots, k$ отримані поліноми діляться на $C_{k-1}(\lambda)$.

Для наглядності наведемо приклад реалізації описаного алгоритму.

Приклад 2.3. Нехай задано матрицю

$$A(\lambda) = \begin{vmatrix} P_0(\lambda) & P_{-1}(\lambda) \\ P_1(\lambda) & P_0(\lambda) \end{vmatrix} = \begin{vmatrix} e^{-2i\lambda} + 1 + e^{i\lambda} & e^{-i\lambda} + 2 + 2e^{i\lambda} \\ 2e^{-i\lambda} + e^{i\lambda} & e^{-2i\lambda} + 1 + e^{i\lambda} \end{vmatrix}$$

$$\text{і праву частину } B(\lambda) = \begin{vmatrix} e^{-i\lambda} + 2 + e^{i\lambda} \\ e^{-2i\lambda} + e^{-i\lambda} + e^{i\lambda} \end{vmatrix}$$

На початку підрахунки всіх проміжних величин будемо проводити формально, шукаючи їх у вигляді співвідношень двох тригонометричних поліномів з невідомими коефіцієнтами і виконуючи лише скорочення в

чисельнику та знаменнику на спільні множники. Виберемо нормувальні
МНОЖНИКИ

$$\tilde{\alpha}_0^{(0)}(\lambda) = \tilde{\beta}_0^{(0)}(\lambda) = \frac{1}{(P_0(\lambda))^2}.$$

На першому кроці отримаємо

$$\tilde{F}_1(\lambda) = P_1(\lambda)\tilde{\alpha}_0^{(0)}(\lambda) = \frac{P_1(\lambda)}{(P_0(\lambda))^2}, \quad \tilde{R}_1(\lambda) = P_{-1}(\lambda)\tilde{\beta}_0^{(0)}(\lambda) = \frac{P_{-1}(\lambda)}{(P_0(\lambda))^2},$$

$$S_1(\lambda) = -Q_0(\lambda)\tilde{F}_1(\lambda) = -\frac{P_0(\lambda)P_1(\lambda)}{(P_0(\lambda))^2} = -\frac{P_1(\lambda)}{P_0(\lambda)},$$

$$V_1(\lambda) = -G_0(\lambda)\tilde{R}_1(\lambda) = -\frac{P_0(\lambda)P_{-1}(\lambda)}{(P_0(\lambda))^2} = -\frac{P_{-1}(\lambda)}{P_0(\lambda)},$$

$$\begin{aligned} G_1(\lambda) = Q_1(\lambda) &= (1 - V_1(\lambda)S_1(\lambda))^{-1}G_0(\lambda) = \left(1 - \frac{P_1(\lambda)P_{-1}(\lambda)}{(P_0(\lambda))^2}\right)^{-1}P_0(\lambda) = \\ &= \left(\frac{(P_0(\lambda))^2 - P_1(\lambda)P_{-1}(\lambda)}{(P_0(\lambda))^2}\right)^{-1}P_0(\lambda) = \frac{(P_0(\lambda))^3}{(P_0(\lambda))^2 - P_1(\lambda)P_{-1}(\lambda)}, \end{aligned}$$

$$\begin{aligned} \begin{bmatrix} \tilde{\alpha}_0^{(1)}(\lambda) \\ \tilde{\alpha}_1^{(1)}(\lambda) \end{bmatrix} &= \begin{bmatrix} \tilde{\alpha}_0^{(0)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(0)}(\lambda) \end{bmatrix} S_1(\lambda) = \begin{bmatrix} \frac{1}{(P_0(\lambda))^2} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{(P_0(\lambda))^2} \end{bmatrix} \left(-\frac{P_1(\lambda)}{P_0(\lambda)}\right) = \\ &= \begin{bmatrix} \frac{1}{(P_0(\lambda))^2} \\ -\frac{P_1(\lambda)}{(P_0(\lambda))^3} \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} \begin{bmatrix} \tilde{\beta}_0^{(1)}(\lambda) \\ \tilde{\beta}_1^{(1)}(\lambda) \end{bmatrix} &= \begin{bmatrix} \tilde{\alpha}_0^{(0)}(\lambda) \\ 0 \end{bmatrix} V_1(\lambda) + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(0)}(\lambda) \end{bmatrix} = \begin{bmatrix} \frac{1}{(P_0(\lambda))^2} \\ 0 \end{bmatrix} \left(-\frac{P_{-1}(\lambda)}{P_0(\lambda)}\right) + \begin{bmatrix} 0 \\ \frac{1}{(P_0(\lambda))^2} \end{bmatrix} = \\ &= \begin{bmatrix} -\frac{P_{-1}(\lambda)}{(P_0(\lambda))^3} \\ \frac{1}{(P_0(\lambda))^2} \end{bmatrix}, \end{aligned}$$

$$D_1(\lambda) = B_1(\lambda) - P_1(\lambda)Z_0^{(0)}(\lambda) = B_1(\lambda) - \frac{P_1(\lambda)B_0(\lambda)}{P_0(\lambda)},$$

$$\begin{aligned}
W_1(\lambda) &= Q_1(\lambda)D_1(\lambda) = \left(B_1(\lambda) - \frac{P_1(\lambda)B_0(\lambda)}{P_0(\lambda)} \right) \frac{(P_0(\lambda))^3}{(P_0(\lambda))^2 - P_1(\lambda)P_{-1}(\lambda)} = \\
&= \frac{(B_1(\lambda)P_0(\lambda) - P_1(\lambda)B_0(\lambda))(P_0(\lambda))^2}{(P_0(\lambda))^2 - P_1(\lambda)P_{-1}(\lambda)}, \\
\begin{bmatrix} z_0^{(1)}(\lambda) \\ z_1^{(1)}(\lambda) \end{bmatrix} &= \begin{bmatrix} z_0^{(0)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} \tilde{\beta}_0^{(1)}(\lambda) \\ \tilde{\beta}_1^{(1)}(\lambda) \end{bmatrix} W_1(\lambda) = \begin{bmatrix} \frac{B_0(\lambda)}{P_0(\lambda)} \\ 0 \end{bmatrix} + \begin{bmatrix} -\frac{P_{-1}(\lambda)}{(P_0(\lambda))^3} \\ \frac{1}{(P_0(\lambda))^2} \end{bmatrix} x \\
&= x \frac{(B_1(\lambda)P_0(\lambda) - P_1(\lambda)B_0(\lambda))(P_0(\lambda))^2}{(P_0(\lambda))^2 - P_1(\lambda)P_{-1}(\lambda)} = \\
&= \begin{bmatrix} \frac{B_0(\lambda)}{P_0(\lambda)} - \frac{P_{-1}(\lambda)(B_1(\lambda)P_0(\lambda) - P_1(\lambda)B_0(\lambda))}{P_0(\lambda)((P_0(\lambda))^2 - P_1(\lambda)P_{-1}(\lambda))} \\ \frac{B_1(\lambda)P_0(\lambda) - P_1(\lambda)B_0(\lambda)}{(P_0(\lambda))^2 - P_1(\lambda)P_{-1}(\lambda)} \end{bmatrix} = \\
&= \begin{bmatrix} \frac{B_0(\lambda)((P_0(\lambda))^2 - P_1(\lambda)P_{-1}(\lambda)) - P_{-1}(\lambda)(B_1(\lambda)P_0(\lambda) - P_1(\lambda)B_0(\lambda))}{P_0(\lambda)((P_0(\lambda))^2 - P_1(\lambda)P_{-1}(\lambda))} \\ \frac{B_1(\lambda)P_0(\lambda) - P_1(\lambda)B_0(\lambda)}{(P_0(\lambda))^2 - P_1(\lambda)P_{-1}(\lambda)} \end{bmatrix}.
\end{aligned}$$

Тепер підставимо конкретні значення у наведені вище формули

$$\begin{aligned}
(P_0(\lambda))^2 &= (e^{-2i\lambda} + 1 + e^{i\lambda})(e^{-2i\lambda} + 1 + e^{i\lambda}) = e^{-4i\lambda} + 2e^{-2i\lambda} + 2e^{-i\lambda} + 1 + 2e^{i\lambda} + e^{2i\lambda}, \\
P_1(\lambda)P_{-1}(\lambda) &= (2e^{-i\lambda} + e^{i\lambda})(e^{-i\lambda} + 2 + 2e^{i\lambda}) = 2e^{-2i\lambda} + 4e^{-2i\lambda} + 4e^{-i\lambda} + 5 + 2e^{i\lambda} + 2e^{2i\lambda}, \\
(P_0(\lambda))^2 - P_1(\lambda)P_{-1}(\lambda) &= e^{-4i\lambda} - 2e^{-2i\lambda} - 4 - e^{2i\lambda}, \\
B_0(\lambda)((P_0(\lambda))^2 - P_1(\lambda)P_{-1}(\lambda)) &= e^{-5i\lambda} + 2e^{-4i\lambda} + e^{-3i\lambda} - 2e^{-2i\lambda} - 8e^{-i\lambda} - 5e^{i\lambda} - 2e^{2i\lambda} - e^{3i\lambda}, \\
B_1(\lambda)P_0(\lambda) &= (e^{-2i\lambda} + e^{-i\lambda} + e^{i\lambda})(e^{-2i\lambda} + 1 + e^{i\lambda}) = \\
&= e^{-4i\lambda} + 2e^{-3i\lambda} + 2e^{-2i\lambda} + 3e^{-i\lambda} + 1 + e^{i\lambda} + 2e^{2i\lambda}, \\
P_1(\lambda)B_0(\lambda) &= (2e^{-i\lambda} + e^{i\lambda})(e^{-i\lambda} + 2 + e^{i\lambda}) = 2e^{-2i\lambda} + 4e^{-i\lambda} + 3 + 2e^{i\lambda} + e^{2i\lambda}, \\
B_1(\lambda)P_0(\lambda) - P_1(\lambda)B_0(\lambda) &= e^{-4i\lambda} + e^{-3i\lambda} - e^{-2i\lambda} - e^{-i\lambda} - 2 - e^{i\lambda}, \\
P_{-1}(\lambda)(B_1(\lambda)P_0(\lambda) - P_1(\lambda)B_0(\lambda)) &= e^{-5i\lambda} + 3e^{-4i\lambda} + 3e^{-3i\lambda} - e^{-2i\lambda} - 6e^{-i\lambda} - 7 - 6e^{i\lambda} - 2e^{2i\lambda},
\end{aligned}$$

$$P_0(\lambda)((P_0(\lambda))^2 - P_1(\lambda)P_{-1}(\lambda)) = (e^{-2i\lambda} + 1 + e^{i\lambda})(e^{-4i\lambda} + 2e^{-2i\lambda} - 4 - e^{2i\lambda}) = e^{-6i\lambda} + e^{-4i\lambda} - e^{-3i\lambda} - 4e^{-2i\lambda} - 2e^{-i\lambda} - 7 - 4e^{i\lambda} - 2e^{i2\lambda} - e^{3i\lambda}.$$

В результаті отримаємо розв'язок

$$\begin{bmatrix} Z_0^{(1)}(\lambda) \\ Z_1^{(1)}(\lambda) \end{bmatrix} = \begin{bmatrix} \frac{-e^{-4i\lambda} - 2e^{-3i\lambda} - e^{-2i\lambda} - 2e^{-i\lambda} - 3 + e^{i\lambda} - e^{3i\lambda}}{e^{-6i\lambda} - e^{-4i\lambda} - e^{-3i\lambda} - 4e^{-2i\lambda} - 2e^{-i\lambda} - 7 - e^{i\lambda} - e^{2i\lambda} - e^{3i\lambda}} \\ \frac{e^{-4i\lambda} + e^{-3i\lambda} - e^{-2i\lambda} - e^{-i\lambda} - 2 - e^{i\lambda}}{e^{-4i\lambda} - 2e^{-i\lambda} - 4e^{2i\lambda}} \end{bmatrix}.$$

Оцінки характеристик алгоритму.

Для реалізації даного алгоритму на комп'ютері достатньо виконати порядку $3n^2(2l+1)$ операцій множення та $3n^2(l^2+2l+1)$ операцій додавання-віднімання. Вектор-розв'язок може бути отриманий на місці вектора – правої частини.

2.4. Розв'язування СЛАР із ганкелевими матрицями з алгебраїчними поліноміальними елементами

Розглянемо СЛАР з ганкелевою λ -матрицею

$$H(\lambda)Z(\lambda) = B(\lambda), \quad (2.32)$$

$$H(\lambda) = \begin{vmatrix} h_0(\lambda) & h_1(\lambda) & h_2(\lambda) & \dots & h_{n-1}(\lambda) \\ h_1(\lambda) & h_2(\lambda) & h_3(\lambda) & \dots & h_n(\lambda) \\ h_2(\lambda) & h_3(\lambda) & h_4(\lambda) & \dots & h_{n+1}(\lambda) \\ \dots & \dots & \dots & \dots & \dots \\ h_{n-1}(\lambda) & h_n(\lambda) & h_{n+1}(\lambda) & \dots & h_{2n-2}(\lambda) \end{vmatrix},$$

де елементи матриці $H(\lambda)$ та вектора $B(\lambda)$ є алгебраїчними поліномами степеня l .

Ганкелева матриця $H(\lambda)$ задається за допомогою лише елементів її першого рядка та останнього стовпця.

Покладається, що значення для λ та коефіцієнти многочленів беруться із поля дійсних чисел \mathbb{R} , так що коли елементи матриці $H(\lambda)$ обчислюються для деякого часткового значення λ , наприклад $\lambda = \lambda_0 \in \mathbb{R}$, то $H(\lambda_0) \in \mathbb{R}^n \times \mathbb{R}^n$.

Припустимо, що всі ведучі мінори λ -матриці $H(\lambda)$ відмінні від нуля. В цьому випадку для знаходження оберненої до ганкелевої λ -матриці $H(\lambda)$ можна за допомогою перестановки стовпців або рядків звести її до тепліцевої λ -матриці і використати описаний вище алгоритм обчислення $H^{-1}(\lambda)$ для тепліцевих λ -матриць. Але в цьому випадку доведеться накладати додаткові умови, вимагаючи, щоб всі мінори утвореної тепліцевої λ -матриці були відмінними від нуля. Тому побудуємо алгоритм, який відповідає деякому компактному способу задання матриці $H^{-1}(\lambda)$, який використовує відмінність від нуля всіх ведучих мінорів матриці $H(\lambda)$.

Спочатку розглянемо процес, який полягає в послідовному виключенні елементів і в результаті зводить матрицю $H(\lambda)$ до трикутного вигляду. Покладемо $H_1(\lambda) = H(\lambda)$, $F_1(\lambda) = I$, де I – одинична матриця. При $2 \leq k \leq n$ знайдемо матриці $F_k(\lambda)$ і $H_k(\lambda)$, які задовольняють співвідношенню

$$F_k(\lambda)H(\lambda) = H_k(\lambda), \quad (2.33)$$

$$\text{де } F_k(\lambda) = \left| \begin{array}{cc|cc|cc|c} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ f_{21}^{(k)}(\lambda) & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline f_{k1}^{(k)}(\lambda) & f_{k2}^{(k)}(\lambda) & f_{kk}^{(k)}(\lambda) & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right|,$$

$$H_k(\lambda) = \begin{array}{c|ccc|ccc} h_{11}^{(k)}(\lambda) & h_{12}^{(k)}(\lambda) & h_{1k}^{(k)}(\lambda) & h_{1k+1}^{(k)}(\lambda) & h_{1k+2}^{(k)}(\lambda) & h_{1n}^{(k)}(\lambda) \\ 0 & h_{22}^{(k)}(\lambda) & h_{2k}^{(k)}(\lambda) & h_{2k+1}^{(k)}(\lambda) & h_{2k+2}^{(k)}(\lambda) & h_{2n}^{(k)}(\lambda) \\ 0 & 0 & h_{kk}^{(k)}(\lambda) & h_{kk+1}^{(k)}(\lambda) & h_{kk+2}^{(k)}(\lambda) & h_{kn}^{(k)}(\lambda) \\ \hline h_k(\lambda) & h_{k+1}(\lambda) & h_{2k-1}(\lambda) & h_{2k}(\lambda) & h_{2k-1}(\lambda) & h_{n+k-2}(\lambda) \\ h_{k+1}(\lambda) & h_{k+2}(\lambda) & h_{2k}(\lambda) & h_{2k+1}(\lambda) & h_{2k+2}(\lambda) & h_{n+k-1}(\lambda) \\ h_{n-1}(\lambda) & h_n(\lambda) & h_{n+k-2}(\lambda) & h_{n+k-1}(\lambda) & h_{n+k}(\lambda) & h_{2n-2}(\lambda)0 \end{array}. \quad (2.34)$$

Припустимо, що (2.34) реалізовано при всіх $t < k$, де $2 \leq k \leq n$ і спробуємо отримати $F_k(\lambda)$ і $H_k(\lambda)$. Розглянемо матрицю

$$\hat{F}_k(\lambda) = \begin{array}{c|cc|cc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ f_{21}^{(k-1)}(\lambda) & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline f_{k-1,1}^{(k-1)}(\lambda) & f_{k-1,2}^{(k-1)}(\lambda) & f_{k-1,k-2}^{(k-1)}(\lambda) & 1 & 0 & 0 & 0 & 0 \\ 0 & f_{k-1,1}^{(k-1)}(\lambda) & f_{k-1,k-3}^{(k-1)}(\lambda) & f_{k-1,k-2}^{(k-1)}(\lambda) & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array},$$

в якій перші $k-1$ стрічок нічим не відрізняються від тих же стрічок матриці $F_{k-1}(\lambda)$, а k -а стрічка отримується з $k-1$ -ої стрічки матриці $F_{k-1}(\lambda)$ шляхом зсуву вправо на одну позицію і присвоєння нуля першому елементу цієї стрічки. В матриці $F_k(\lambda)H(\lambda)$ перші $k-1$ стрічок співпадають з аналогічними стрічками матриці $H_{k-1}(\lambda)$, а k -а стрічка буде мати нулі у всіх позиціях k, j , де $j \leq k-3$, і без врахування останнього елемента отримується з останньої стрічки шляхом зсуву на одну позицію вліво, оскільки матриця $H(\lambda)$ – ганкелева.

Невиродженість ведучих підматриць в $H(\lambda)$ породжує відмінність від нуля елементів. Зокрема, відмінні від нуля елементи $h_{k-2,k-2}^{(k-1)}(\lambda)$ і $h_{k-1,k-1}^{(k-1)}(\lambda)$. Отже, за допомогою $k-2$ -ої та $k-1$ -ої стрічок матриці $H_{k-1}(\lambda)$, які співпадають з аналогічними стрічками матриці $\hat{F}_k(\lambda)H(\lambda)$, можна виключити елементи матриці $\hat{F}_k(\lambda)H(\lambda)$, які стоять відповідно в позиціях

$k, k-2$ і $k, k-1$. Ці дії можна записати за допомогою таких матричних співвідношень

$$F_k(\lambda) = \overline{F}_k(\lambda) \hat{F}_k(\lambda), \quad (2.35)$$

де

$$\overline{F}_k(\lambda) = \left(\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & \alpha_k(\lambda) & \beta_k(\lambda) & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right),$$

$$\alpha_k(\lambda) = -\frac{h_{k-1,k-1}^{(k-1)}(\lambda)}{h_{k-2,k-2}^{(k-1)}(\lambda)}, \quad \beta_k(\lambda) = -\frac{h_{k-1,k}^{(k-1)}(\lambda) + \alpha_k h_{k-2,k-1}^{(k-1)}(\lambda)}{h_{k-1,k-1}^{(k-1)}(\lambda)}.$$

Обчислення цих матриць за формулами (2.34), (2.35) потребує, в силу їх специфіки, виконання лише $O(n^2 l^2)$ арифметичних операцій. Економія здійснюється за рахунок описаних вище зсувів.

При даних обмеженнях на матрицю $H(\lambda)$ обернену матрицю будемо задавати в такому вигляді [21]

$$H^{-1}(\lambda) = \left. \begin{array}{l} \left[\begin{array}{c} p_1(\lambda) \\ p_2(\lambda) \\ \vdots \\ p_n(\lambda) \end{array} \right] p_n^{-1}(\lambda) [p_1(\lambda) p_2(\lambda) \dots p_1(\lambda)] + \\ + \gamma \left\{ \begin{array}{cc|cc} \hat{p}_1(\lambda) & 0 & 0 & 0 \\ \hat{p}_2(\lambda) & \hat{p}_1(\lambda) & 0 & 0 \\ \hline \hat{p}_{n-1}(\lambda) & \hat{p}_{n-2}(\lambda) & \hat{p}_1(\lambda) & 0 \\ 0 & \hat{p}_{n-1}(\lambda) & \hat{p}_2(\lambda) & \hat{p}_1(\lambda) \end{array} \right\}, \end{array} \right. \quad (2.36)$$

де

$$\gamma(\lambda) = \frac{h_{n-1}(\lambda) \hat{p}_1(\lambda) + h_n(\lambda) \hat{p}_2(\lambda) + \dots + h_{2n-3}(\lambda) \hat{p}_{n-1}(\lambda)}{p_n(\lambda)},$$

а величини $p(\lambda)$ і $\hat{p}(\lambda)$ визначаються як

$$p(\lambda) = \frac{1}{h_{n,n}^{(n)}(\lambda)} [f_{n,1}^{(n)}, \dots, f_{n,n-1}^{(n)}, 1]^T, \quad \hat{p}(\lambda) = \frac{1}{h_{n-1,n-1}^{(n-1)}(\lambda)} [f_{n-1,1}^{(n-1)}, \dots, f_{n-1,n-2}^{(n-1)}, 1]^T.$$

Отже, для ганкелевої матриці можна знайти обернену, обчислюючи лише коефіцієнти $h_{kk}^{(k)}(\lambda)$ і стрічки $f^{(k)}(\lambda) = [f_{k1}^{(k)}(\lambda) \dots f_{k,k-1}^{(k)}(\lambda) \ 1]$, і при цьому для побудови оберненої матриці потрібні лише стрічки $f^{(n)}(\lambda)$ та $f^{(n-1)}(\lambda)$ і немає необхідності збереження стрічок $f^{(k)}(\lambda)$ для всіх k .

Для зручності введемо позначення: $r_k(\lambda) = h_{kk}^{(k)}(\lambda)$, $s_k(\lambda) = h_{k-1,k}^{(k-1)}(\lambda)$. Тоді шуканий алгоритм можна записати так

для $k = 1, 2$

$$f_{21}^{(2)}(\lambda) = -\frac{h_1(\lambda)}{h_0(\lambda)}, r_1(\lambda) = h_0(\lambda), r_2(\lambda) = h_2(\lambda) + f_{21}^{(2)}(\lambda)h_1(\lambda), s_2(\lambda) = h_1(\lambda);$$

для $k = 3, \dots, n$

$$s_k(\lambda) = h_{k-1}(\lambda)f_{k-1,1}^{(k-1)}(\lambda) + h_k(\lambda)f_{k-1,2}^{(k-1)}(\lambda) + \dots + h_{2k-4}(\lambda)f_{k-1,k-2}^{(k-1)}(\lambda) + h_{2k-3}(\lambda),$$

$$\alpha_k(\lambda) = -\frac{r_{k-1}(\lambda)}{r_{k-2}(\lambda)}, \beta_k(\lambda) = -\frac{s_{k-1}(\lambda) + \alpha_k(\lambda)s_{k-1}(\lambda)}{r_{k-1}(\lambda)},$$

$$\begin{bmatrix} f_{k1}^{(k)}(\lambda) \\ f_{k2}^{(k)}(\lambda) \\ \dots \\ f_{k,k-2}^{(k)}(\lambda) \\ f_{k,k-1}^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} 0 \\ f_{k-1,1}^{(k-1)}(\lambda) \\ \dots \\ f_{k-1,k-3}^{(k-1)}(\lambda) \\ f_{k-1,k-2}^{(k-1)}(\lambda) \end{bmatrix} + \beta_k(\lambda) \begin{bmatrix} f_{k-1,1}^{(k-1)}(\lambda) \\ f_{k-1,2}^{(k-1)}(\lambda) \\ \dots \\ f_{k-1,k-2}^{(k-1)}(\lambda) \\ 1 \end{bmatrix} + \alpha_k(\lambda) \begin{bmatrix} f_{k-2,1}^{(k-2)}(\lambda) \\ f_{k-2,k-3}^{(k-2)}(\lambda) \\ \dots \\ 1 \\ 0 \end{bmatrix} \quad (2.37)$$

$$r_k(\lambda) = h_{k-1}(\lambda)f_{k,1}^{(k)}(\lambda) + h_k(\lambda)f_{k,2}^{(k)}(\lambda) + \dots + h_{2k-3}(\lambda)f_{k,k-1}^{(k)}(\lambda) + h_{2k-2}(\lambda)$$

При реалізації алгоритму на комп'ютері, якщо проводити обчислення за формулами (2.37), виконуючи лише операції множення та додавання многочленів, може виникнути так зване “розбухання” проміжних даних і порядок поліномів в обчисленнях буде наростати з швидкістю 2^{2^k} , де k – порядок розв’язуваної відсіченої системи. Щоб уникнути цього, проведемо додаткові дослідження.

Будемо шукати $f^{(k)}(\lambda)$ у вигляді відношення двох поліномів з невідомими коефіцієнтами

$$\frac{f^{(k)}(\lambda)}{C_k(\lambda)} = \begin{bmatrix} 0 \\ \frac{f^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{f^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{f^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{f^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{f^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \end{bmatrix} + \beta_k(\lambda) \begin{bmatrix} \frac{f^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{f^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{f^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{f^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{f^{(k-1)}(\lambda)}{C_{k-1}(\lambda)} \\ 1 \end{bmatrix} + \alpha_k(\lambda) \begin{bmatrix} \frac{f^{(k-2)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{f^{(k-2)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{f^{(k-2)}(\lambda)}{C_{k-1}(\lambda)} \\ \frac{f^{(k-2)}(\lambda)}{C_{k-1}(\lambda)} \\ 1 \\ 0 \end{bmatrix}$$

ТЕОРЕМА 2.6. Якщо в процесі реалізації алгоритму (2.37) всі головні мінори ганкелевої матриці не дорівнюють нулю, то $\forall k = 0, 1, \dots, n-1$ та $i = 0, 1, \dots, k$ поліноми $C_{k-1}(\lambda)$ є дільниками многочленів $f^{(k-1)}(\lambda) + \beta_k(\lambda)f^{(k-1)}(\lambda) + \alpha_k(\lambda)f^{(k-2)}(\lambda)$.

Доведення аналогічне доведенню теореми 2.5.

З врахуванням цього факту вноситься суттєва корекція в схему обчислень за алгоритмом (2.37). На кожному кроці відразу після обчислення $f_i^{(k)}(\lambda)$ ($i = 0, 1, \dots, k$) отримані поліноми діляться на $C_{k-1}(\lambda)$.

Даний рекурентний процес проводиться з припущення, що всі ведучі мінори ганкелевої матриці відмінні від нуля. Реалізація формул (2.37) потребує виконання $2n^2l^2$ операцій множення і стільки ж операцій додавання-віднімання. При цьому, окрім інформації про ганкелеву матрицю $H(\lambda)$, треба запам'ятати на комп'ютері два вектора довжини не більше $2n^2l^2$.

Обернена матриця, отримана за формулою (2.36), є ганкелевою [35]. За допомогою перестановки її стовпців або рядків зведемо її до відповідної тепліцевої матриці $A^{-1}(\lambda)$. Для виконання таких перестановок використаємо алгоритм з заміщенням [21], який потребує $2N^2(l+1)\log_2 N$ обчислень значень відповідних підстановок з використанням додаткової пам'яті $n(l+1)$ комірок.

Шуканий роз'язок $Z(\lambda) = A^{-1}(\lambda)B(\lambda)$ можна отримати, використавши швидке перетворення Фур'є, за $6n \log_2 nl^2$ операцій комплексного множення і $l^2 n \log_2 nl^2$ операцій комплексного додавання–віднімання.

2.5. Послідовні і паралельні моделі обчислень.

При розробці алгоритмів для класів задач розв'язування СЛАР з теплицевим λ -матрицями має використовуватись та чи інша модель обчислень, тобто традиційні однопроцесорні чи багатопроцесорні комп'ютери. Це залежить від класу задач (зокрема, розмірності матриці $A(\lambda)$) та вимог до точності та часу розв'язку задачі на комп'ютері.

Послідовні моделі обчислень. Розглянемо загальну модель сучасного однопроцесорного комп'ютера. Звичайно комп'ютер містить процесор, який забезпечує виконання арифметичних операцій та пам'ять: надоперативну пам'ять (НОП) або місцеву пам'ять, оперативну або основну (ОП), пам'ять з прямим доступом на магнітних дисках і т.п.

Пам'ять сучасних комп'ютерів має ієрархічну структуру запам'ятовуючих пристроїв, які мають різну швидкодію та об'єм. Чим вище рівень, тим вища швидкодія відповідної пам'яті, але й менший її об'єм. До верхнього рівня належать запам'ятовуючі пристрої, з якими процесор взаємодіє безпосередньо в процесі виконання програми. Це основна (оперативна) пам'ять (ОП). Вона має порівняно невеликий об'єм, але цей недолік компенсується практично необмеженим об'ємом зовнішньої пам'яті (ЗП). Однак, зовнішні запам'ятовуючі пристрої порівняно повільні. Тому обчислювальний процес повинен відбуватись з якнайменшим числом звертань до ЗП і максимально можливим використанням ОП.

Кожен рівень ієрархії може містити кілька модулів відповідних пристроїв накопичення, що забезпечує збереження необхідної для даного рівня кількості інформації. Ієрархічна структура пам'яті дозволяє економічно

ефективно поєднувати зберігання великих об'ємів інформації зі швидким доступом до неї в процесі обробки.

У ряді випадків швидкодія ОП є недостатньою, і в склад машини доводиться вводити швидкісну пам'ять (буферну або кеш-пам'ять) на кілька сотень або тисяч машинних слів з циклом звертання кілька десятків наносекунд.

Структура більшості сучасних комп'ютерів містить об'єднану кеш-пам'ять для фрагментів команд і груп даних. Ця пам'ять суттєво більш швидкодіюча, ніж ОП, але має порівняно невеликий об'єм.

При звертанні процесора до ОП для зчитування в кеш передається блок інформації, що містить потрібне слово, при цьому відбувається попередня перевірка на наявність даної інформації в кеш. При виявленні співпадання запис дублюється в самій кеш без звертання до ОП. Це дає можливість суттєво зменшити середній час, що витрачається на вибірку даних.

Паралельні моделі обчислень. В наш час широко розповсюджені багатопроцесорні комп'ютери. Розробляються і створюються нові багатопроцесорні комп'ютери. Їх опис та характеристики дано в [15, 20 67, 79, 88, 102].

Метою побудови паралельних алгоритмів є зменшення часу розв'язування задачі на багатопроцесорному комп'ютері.

При врахуванні обмежень можливостей і ресурсів з певним режимом роботи багатопроцесорного комп'ютера алгоритми розв'язування СЛАР з теплицевими матрицями залежать від його структури та архітектури [90].

2.6. Оцінки характеристик алгоритмів розв'язування СЛАР з теплицевими λ -матрицями

Нехай для класу задач P розв'язування СЛАР з теплицевою λ -матрицею застосовується (або розробляється) алгоритм \hat{A} для комп'ютера (моделі обчислень) S .

В практиці чисельного розв'язування задач за допомогою комп'ютера розглядають наступні характеристики задач, алгоритмів і моделей обчислень [17-19, 81, 97]:

M – необхідна пам'ять для розв'язку задачі $\bar{p} \in P$ на комп'ютері C ;

T – час, необхідний для розв'язку задачі;

E – повна похибка розв'язку задачі $\bar{p} \in P$ на комп'ютері C за допомогою алгоритму \hat{A} .

На основі аналізу оцінок характеристик робиться висновок про гарантовану якість розв'язку задачі та порівняльний аналіз різних алгоритмів розв'язування задачі по точності і швидкодії [81].

Оцінка пам'яті комп'ютера. Оцінка пам'яті комп'ютера (моделі обчислень C), відповідно ОП, кеш-пам'яті та ЗП, необхідної для розв'язку СЛАР з тепліцевою λ -матрицею розмірності $n \times n$, елементами якої є алгебраїчні поліноми розмірності l , обчислюється за формулою $M \leq M_1 + M_2 + MP + M_z$, де M_1 – масив розмірності $(3 \times n - 1) \times l$ для коефіцієнтів поліномів, які використовуються в процесі розв'язку задачі, M_2 – масив для проміжних результатів, що визначається конкретним алгоритмом, MP – складова оцінка пам'яті, яка є постійною для обчислювального алгоритму конкретного типу комп'ютера і його програмного забезпечення, M_z – масив розмірності $n \times l$ для коефіцієнтів поліномів, які є розв'язками СЛАР із заданою тепліцевою λ -матрицею.

Оцінка часу розв'язування задачі.

Час T , необхідний для розв'язування СЛАР з тепліцевою λ -матрицею за допомогою відповідного алгоритму \hat{A} на комп'ютері (моделі обчислень C), витрачається на такі види робіт: T_1 – ввід та вивід даних; T_2 – обчислення розв'язку $Z(\lambda) = \{Z_1(\lambda), Z_2(\lambda), \dots, Z_n(\lambda)\}$; T_3 – обмін із зовнішніми накопичувачами. Обмін даними здійснюється між ЗП, ОП і кеш-пам'яттю.

Оскільки деякі з перерахованих робіт можуть виконуватись одночасно, то $T \leq T_1 + T_2 + T_3$.

Оцінка комп'ютерного часу T_2 побудови розв'язку СЛАР з тепліцевою λ -матрицею з допомогою відповідного послідовного алгоритму \hat{A} здійснюється за такою схемою $T_2 \leq \theta_+(n,l) \cdot t_+ + \theta_-(n,l) \cdot t_-$, де $\theta_+(n,l)$, $\theta_-(n,l)$ – відповідно оцінки обчислювальної складності (необхідної кількості арифметичних операцій: додавання–віднімання та множення) алгоритму, t_+ , t_- – оцінки часу обчислення арифметичних операцій відповідно додавання–віднімання та множення.

Висновки до розділу 2

В другому розділі описано розроблені ефективні алгоритми, які узагальнюють числові методи лінійної алгебри для тепліцевих (ганкелевих) матриць з дійсними та комплексними елементами, що ґрунтуються на економічних схемах знаходження оберненої матриці, на випадок розв'язування СЛАР з тепліцевими (ганкелевими) λ -матрицями. Зроблено аналіз оцінок характеристик створених алгоритмів.

Отримані результати опубліковані у працях [37, 39, 40, 41].

РОЗДІЛ 3

ПОСЛІДОВНІ ТА ПАРАЛЕЛЬНІ МОДЕЛІ РОЗВ'ЯЗУВАННЯ СИСТЕМ ЛІНІЙНИХ АЛГЕБРАЇЧНИХ РІВНЯНЬ З БЛОЧНО- ТЕПЛІЦЕВИМИ λ -МАТРИЦЯМИ

У даному розділі розглядається задача розв'язування СЛАР з блочно-тепліцевими λ -матрицями, елементами яких є алгебраїчні поліноми (підрозд. 3.1.); моделі для реалізації алгоритмів розв'язування СЛАР з тепліцевими λ -матрицями на багатопроцесорних обчислювальних системах (підрозд. 3.2.); використання гіллястих ланцюгових дробів для розв'язування СЛАР з тепліцевими λ -матрицями (підрозд. 3.3.).

3.1. Алгоритми розв'язування СЛАР з блочно-тепліцевими λ -матрицями.

Розглядається задача розв'язування СЛАР

$$A(\lambda)Z(\lambda) = B(\lambda), \quad (3.1)$$

де

$$A(\lambda) = \begin{vmatrix} P_0(\lambda) & P_{-1}(\lambda) & P_{-2}(\lambda) & \dots & P_{-n+1}(\lambda) \\ P_1(\lambda) & P_0(\lambda) & P_{-1}(\lambda) & \dots & P_{-n+2}(\lambda) \\ P_2(\lambda) & P_1(\lambda) & P_0(\lambda) & \dots & P_{-n+3}(\lambda) \\ \dots & \dots & \dots & \dots & \dots \\ P_{n-1}(\lambda) & P_{n-2}(\lambda) & P_{n-3}(\lambda) & \dots & P_0(\lambda) \end{vmatrix} - \quad (3.2)$$

блочно-тепліцева матриця, яка складається з квадратних блоків $P_{ij}(\lambda) = P_{i-j}(\lambda)$, $1 \leq i \leq n-1$, де кожен блок – це $P \times P$ -матриця, елементами якої є алгебраїчні поліноми порядку l . Значення для λ та коефіцієнти многочленів беруться з деякого поля $F^{m \times n}$ так, що елементи матриці $A(\lambda)$ обчислюються для деякого часткового значення λ , наприклад $\lambda = \lambda_0 \in F^{m \times n}$.

Не завжди специфіку задачі вдається врахувати за допомогою вибору спеціального числового методу. Але в деяких випадках можна досягти значної економії, особливо для задач з матрицями великих розмірів. Щоб

використати специфіку матриць, часто і доводиться розглядати їх як блочні. Це властиво більшості прикладних задач, тому виникає потреба в розробці економічних алгоритмів для блочних матриць.

Алгоритми знаходження матриці, оберненої до блочно-тепліцевої λ -матриці.

Нехай $A(\lambda)$ – тепліцева λ -матриця розмірності $n \times n$. Розіб'ємо її на клітки $A_{ij}(\lambda)$ порядку p , $n = pr$. Матриця $A(\lambda)$ називається *клітково-тепліцевою*, якщо $A_{ij}(\lambda) = A_{st}(\lambda)$ при $i - j = s - t$. Така матриця володіє чітко

вираженою специфікою і може бути задана масивом чисел величиною $\frac{2n^2l^2}{r}$,

а не n^2l^2 , як в загальному випадку (l – степінь поліномів елементів матриці).

Якщо систему з клітково-тепліцевою матрицею розв'язувати за допомогою звичайних методів, то вже після перших перетворень специфіка матриці порушується. Тому для ефективного використання пам'яті комп'ютера слід використовувати спеціальні числові методи розв'язування систем з клітково-тепліцевими матрицями [19, 21, 93]. Застосуємо ідею методу Левінсона для числових систем [10] до розв'язування систем вигляду $A(\lambda)Z(\lambda) = B(\lambda)$.

Нехай $P_k(\lambda)$, $0 \leq k \leq n - 1$ – ведуча підматриця, яка складається з блоків $P_{i-j}(\lambda)$, де $1 \leq i, j \leq k + 1$, причому $P_{n-1}(\lambda) = A(\lambda)$. Ідея алгоритму полягає в обчисленні першого та останнього стовпців оберненої матриці без обчислення елементів інших стовпців. Будемо вважати, що всі ведучі мінори матриці (3.2) без обмеження загальності відмінні від нуля. Тоді для будь-якого k матриця $P_{k-1}(\lambda)$ виражається через її перший та останній блочні стовпці або першу та останню блочні стрічки [21], які позначимо відповідно $\alpha_i^{(k)}(\lambda)$, $\beta_i^{(k)}(\lambda)$, $\varphi_i^{(k)}(\lambda)$, $\psi_i^{(k)}(\lambda)$ ($0 \leq i \leq k$).

Для такої матриці алгоритм обчислення послідовності її першого та останнього блочних стовпців запишемо у вигляді

для $k = 0$ $\alpha_0^{(0)}(\lambda) = \beta_0^{(0)}(\lambda) = (P_0(\lambda))^{-1}$,

для $k = 1, \dots, n-1$

$$\begin{aligned} F_k(\lambda) &= P_k(\lambda)\alpha_0^{(k-1)}(\lambda) + P_{k-1}(\lambda)\alpha_1^{(k-1)}(\lambda) + \dots + P_1(\lambda)\alpha_{k-1}^{(k-1)}(\lambda), \\ R_k(\lambda) &= P_{-1}(\lambda)\beta_0^{(k-1)}(\lambda) + P_{-2}(\lambda)\beta_1^{(k-1)}(\lambda) + \dots + P_{-k}(\lambda)\beta_{k-1}^{(k-1)}(\lambda), \\ V_k(\lambda) &= (I - R_k(\lambda)F_k(\lambda))^{-1}, \quad S_k(\lambda) = -F_k(\lambda)V_k(\lambda), \\ U_k(\lambda) &= (I - F_k(\lambda)R_k(\lambda))^{-1}, \quad K_k(\lambda) = -R_k(\lambda)U_k(\lambda), \end{aligned}$$

де I – одинична матриця порядку P ,

$$\begin{aligned} \begin{bmatrix} \alpha_0^{(k)}(\lambda) \\ \alpha_1^{(k)}(\lambda) \\ \alpha_{k-1}^{(k)}(\lambda) \\ \alpha_k^{(k)}(\lambda) \end{bmatrix} &= \begin{bmatrix} \alpha_0^{(k-1)}(\lambda) \\ \alpha_1^{(k-1)}(\lambda) \\ \alpha_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} V_k(\lambda) + \begin{bmatrix} 0 \\ \beta_0^{(k-1)}(\lambda) \\ \beta_{k-2}^{(k-1)}(\lambda) \\ \beta_{k-1}^{(k-1)}(\lambda) \end{bmatrix} S_k(\lambda), \\ \begin{bmatrix} \beta_0^{(k)}(\lambda) \\ \beta_1^{(k)}(\lambda) \\ \beta_{k-1}^{(k)}(\lambda) \\ \beta_k^{(k)}(\lambda) \end{bmatrix} &= \begin{bmatrix} \alpha_0^{(k-1)}(\lambda) \\ \alpha_1^{(k-1)}(\lambda) \\ \alpha_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} U_k(\lambda) + \begin{bmatrix} 0 \\ \beta_0^{(k-1)}(\lambda) \\ \beta_{k-2}^{(k-1)}(\lambda) \\ \beta_{k-1}^{(k-1)}(\lambda) \end{bmatrix} K_k(\lambda). \end{aligned} \quad (3.3)$$

Аналогічний алгоритм можна записати для обчислення послідовності перших та останніх блочних рядків матриці (3.2)

для $k = 0$ $\varphi_0^{(0)}(\lambda) = \psi_0^{(0)}(\lambda) = (P_0(\lambda))^{-1}$,

для $k = 1, \dots, n-1$

$$\begin{aligned} \hat{F}_k(\lambda) &= \varphi_0^{(k-1)}(\lambda)P_{-k}(\lambda) + \varphi_1^{(k-1)}(\lambda)P_{-k+1}(\lambda) + \dots + \varphi_{k-1}^{(k-1)}(\lambda)P_{-1}(\lambda), \\ \hat{R}_k(\lambda) &= \psi_0^{(k-1)}(\lambda)P_1(\lambda) + \psi_1^{(k-1)}(\lambda)P_2(\lambda) + \dots + \psi_{k-1}^{(k-1)}(\lambda)P_k(\lambda), \\ \hat{V}_k(\lambda) &= (I - \hat{F}_k(\lambda)\hat{R}_k(\lambda))^{-1}, \quad \hat{S}_k(\lambda) = -\hat{V}_k(\lambda)\hat{F}_k(\lambda), \\ \hat{U}_k(\lambda) &= (I - \hat{R}_k(\lambda)\hat{F}_k(\lambda))^{-1}, \quad \hat{K}_k(\lambda) = -\hat{U}_k(\lambda)\hat{R}_k(\lambda) \\ \begin{bmatrix} \varphi_0^{(k)}(\lambda) \\ \varphi_1^{(k)}(\lambda) \\ \varphi_{k-1}^{(k)}(\lambda) \\ \varphi_k^{(k)}(\lambda) \end{bmatrix} &= \tilde{V}_k(\lambda) \begin{bmatrix} \varphi_0^{(k-1)}(\lambda) \\ \varphi_1^{(k-1)}(\lambda) \\ \varphi_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \tilde{S}_k(\lambda) \begin{bmatrix} 0 \\ \psi_0^{(k-1)}(\lambda) \\ \psi_{k-2}^{(k-1)}(\lambda) \\ \psi_{k-1}^{(k-1)}(\lambda) \end{bmatrix}, \end{aligned} \quad (3.4)$$

$$\begin{bmatrix} \psi_0^{(k)}(\lambda) \\ \psi_1^{(k)}(\lambda) \\ \dots \\ \psi_{k-1}^{(k)}(\lambda) \\ \psi_k^{(k)}(\lambda) \end{bmatrix} = \tilde{K}_k(\lambda) \begin{bmatrix} \varphi_0^{(k-1)}(\lambda) \\ \varphi_1^{(k-1)}(\lambda) \\ \dots \\ \varphi_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \tilde{U}_k(\lambda) \begin{bmatrix} 0 \\ \psi_0^{(k-1)}(\lambda) \\ \dots \\ \psi_{k-1}^{(k-1)}(\lambda) \\ \psi_{k-1}^{(k-1)}(\lambda) \end{bmatrix}.$$

Оцінки складності алгоритмів (3.3) і (3.4). Ці алгоритми володіють однаковими характеристиками [36]. Вони реалізуються за умови невідродженості ведучих підматриць $P_k(\lambda)$, для всіх $0 \leq k \leq n-1$.

Кожен з них потребує виконання $3p^3l^2n^2$ операцій множення і $(3p^3l^2 - p2l^2)n^2$ операцій додавання-віднімання.

В процесі обчислень для кожного з цих алгоритмів необхідна пам'ять для розміщення $2p^2 \ln$ чисел і пам'ять для зберігання $2p^2 \ln$ чисел, що задають перший блочний стовпець і першу блочну стрічку, які визначають початкову матрицю $A(\lambda)$.

Є певна відповідність, за якою множенню і додаванню блоків у блочному алгоритмі відповідає множення і додавання чисел у скалярному. Але на відміну від скалярних алгоритмів у блочних при великих p головний член операцій додавання буде залежати лише від кількості множень блоків. Так в скалярному варіанті алгоритму (3.3) кількість операцій додавання-віднімання $2l^2n^2$, а в блочному його головний член залежності від p і n дорівнює $3p^2l^2n^2$, що відповідає $3l^2n^2$ операціям множення, які вимагаються в скалярному алгоритмі.

Блочний алгоритм з нормуванням. З точки зору економічності особливого значення набуває розробка блочних алгоритмів з нормуванням, які вимагають меншої кількості операцій множення, ніж скалярні [36]. Застосуємо такий алгоритм для знаходження оберненої матриці і в нашому випадку.

Будемо вважати, що для будь-якого $0 \leq k \leq n-1$ в результаті виконання k -ого кроку не обчислюються блоки $\alpha_i^{(k)}(\lambda)$ і $\beta_i^{(k)}(\lambda)$, або блоки $\varphi_i^{(k)}(\lambda)$,

$\psi_i^{(k)}(\lambda)$ ($0 \leq i \leq k$) безпосередньо, а замість цього знаходяться блоки $\tilde{\alpha}_i^{(k)}(\lambda)$, $\tilde{\beta}_i^{(k)}(\lambda)$ або $\tilde{\varphi}_i^{(k)}(\lambda)$, $\tilde{\psi}_i^{(k)}(\lambda)$ ($0 \leq i \leq k$) і невироджені блоки $G_k(\lambda)$ і $Q_k(\lambda)$ або $\hat{G}_k(\lambda)$ і $\hat{Q}_k(\lambda)$, які відіграють роль нормувальних множників, що забезпечують справедливість співвідношень

$$\alpha_i^{(k)}(\lambda) = \tilde{\alpha}_i^{(k)}(\lambda)G_k(\lambda), \quad \beta_i^{(k)}(\lambda) = \tilde{\beta}_i^{(k)}(\lambda)Q_k(\lambda), \quad 0 \leq i \leq k \quad (3.5)$$

або при обчисленні блочних стрічок

$$\varphi_{ii}^{(k)}(\lambda) = \hat{G}_k(\lambda)\tilde{\varphi}_i^{(k)}(\lambda), \quad \psi_i^{(k)}(\lambda) = \hat{Q}_k(\lambda)\tilde{\psi}_i^{(k)}(\lambda), \quad 0 \leq i \leq k \quad (3.6)$$

Тоді алгоритм з нормуванням для обчислення першого та останнього блочних стовпців оберненої матриці можна сформулювати так

для $k = 0$ $G_0(\lambda) = Q_0(\lambda) = P_0(\lambda)$,

$$\tilde{\alpha}_0^{(0)}(\lambda) = P_0^{-1}(\lambda)G_0^{-1}(\lambda), \quad \tilde{\beta}_0^{(0)}(\lambda) = P_0^{-1}(\lambda)Q_0^{-1}(\lambda),$$

для $k = 1, \dots, n-1$

$$\tilde{F}_k(\lambda) = P_k(\lambda)\tilde{\alpha}_0^{(k-1)}(\lambda) + P_{k-1}(\lambda)\tilde{\alpha}_1^{(k-1)}(\lambda) + \dots + P_1(\lambda)\tilde{\alpha}_{k-1}^{(k-1)}(\lambda),$$

$$\tilde{R}_k(\lambda) = P_{-1}(\lambda)\tilde{\beta}_0^{(k-1)}(\lambda) + P_{-2}(\lambda)\tilde{\beta}_1^{(k-1)}(\lambda) + \dots + P_{-k}(\lambda)\tilde{\beta}_{k-1}^{(k-1)}(\lambda),$$

$$\tilde{S}_k(\lambda) = -Q_{k-1}(\lambda)\tilde{F}_k(\lambda), \quad \tilde{V}_k(\lambda) = -G_{k-1}(\lambda)\tilde{R}_k(\lambda),$$

$$G_k(\lambda) = (I - \tilde{V}_k(\lambda)\tilde{S}_k(\lambda))^{-1}G_{k-1}(\lambda), \quad Q_k(\lambda) = (I - \tilde{S}_k(\lambda)\tilde{V}_k(\lambda))^{-1}Q_{k-1}(\lambda),$$

$$\begin{bmatrix} \tilde{\alpha}_0^{(k)}(\lambda) \\ \tilde{\alpha}_1^{(k)}(\lambda) \\ \tilde{\alpha}_{k-1}^{(k)}(\lambda) \\ \tilde{\alpha}_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(k-1)}(\lambda) \\ \tilde{\alpha}_1^{(k-1)}(\lambda) \\ \tilde{\alpha}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix} \tilde{S}_k(\lambda), \quad (3.7)$$

$$\begin{bmatrix} \tilde{\beta}_0^{(k)}(\lambda) \\ \tilde{\beta}_1^{(k)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k)}(\lambda) \\ \tilde{\beta}_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(k-1)}(\lambda) \\ \tilde{\alpha}_1^{(k-1)}(\lambda) \\ \tilde{\alpha}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} \tilde{V}_k(\lambda) + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix}.$$

Аналогічний алгоритм можна записати для обчислення послідовності перших та останніх блочних стрічок матриці (3.2)

$$\text{для } k=0 \quad \hat{G}_0(\lambda) = \hat{Q}_0(\lambda) = P_0(\lambda),$$

$$\tilde{\varphi}_0^{(0)}(\lambda) = \hat{G}_0^{-1}(\lambda)P_0^{-1}(\lambda), \quad \tilde{\psi}_0^{(0)}(\lambda) = \hat{Q}_0^{-1}(\lambda)P_0^{-1}(\lambda),$$

для $k=1, \dots, n-1$

$$\tilde{F}_k(\lambda) = \tilde{\varphi}_0^{(k-1)}(\lambda)P_{-k}(\lambda) + \tilde{\varphi}_1^{(k-1)}(\lambda)P_{-k+1}(\lambda) + \dots + \tilde{\varphi}_{k-1}P_{-1}(\lambda)^{(k-1)}(\lambda),$$

$$\tilde{R}_k(\lambda) = \tilde{\psi}_0^{(k-1)}(\lambda)P_1(\lambda) + \tilde{\psi}_1^{(k-1)}(\lambda)P_2(\lambda) + \dots + \tilde{\psi}_{k-1}^{(k-1)}(\lambda)P_k(\lambda),$$

$$\tilde{S}_k(\lambda) = -\tilde{F}_k(\lambda)\hat{Q}_{k-1}(\lambda), \quad \tilde{V}_k(\lambda) = -\tilde{R}_k(\lambda)\hat{G}_{k-1}(\lambda),$$

$$\hat{G}_k(\lambda) = \hat{G}_{k-1}(\lambda)\left(I - \tilde{S}_k(\lambda)\tilde{V}_k(\lambda)\right)^{-1}, \quad \hat{Q}_k(\lambda) = \hat{Q}_{k-1}(\lambda)\left(I - \tilde{V}_k(\lambda)\tilde{S}_k(\lambda)\right)^{-1},$$

$$\begin{bmatrix} \tilde{\varphi}_0^{(k)}(\lambda) \\ \tilde{\varphi}_1^{(k)}(\lambda) \\ \tilde{\varphi}_{k-1}^{(k)}(\lambda) \\ \tilde{\varphi}_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\varphi}_0^{(k-1)}(\lambda) \\ \tilde{\varphi}_1^{(k-1)}(\lambda) \\ \tilde{\varphi}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \tilde{S}_k(\lambda) \begin{bmatrix} 0 \\ \tilde{\psi}_0^{(k-1)}(\lambda) \\ \tilde{\psi}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\psi}_{k-1}^{(k-1)}(\lambda) \end{bmatrix}, \quad (3.8)$$

$$\begin{bmatrix} \tilde{\psi}_0^{(k)}(\lambda) \\ \tilde{\psi}_1^{(k)}(\lambda) \\ \tilde{\psi}_{k-1}^{(k)}(\lambda) \\ \tilde{\psi}_k^{(k)}(\lambda) \end{bmatrix} = \tilde{V}_k(\lambda) \begin{bmatrix} \tilde{\varphi}_0^{(k-1)}(\lambda) \\ \tilde{\varphi}_1^{(k-1)}(\lambda) \\ \tilde{\varphi}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\psi}_0^{(k-1)}(\lambda) \\ \tilde{\psi}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\psi}_{k-1}^{(k-1)}(\lambda) \end{bmatrix}.$$

Оцінки складності алгоритмів (3.7) і (3.8). Кожен з цих алгоритмів потребує виконання $3p^3l^2n^2$ операцій множення і стільки ж операцій додавання-віднімання (останнє число складається з $2p^2(p-1)l^2n^2$ операцій додавання, пов'язаних з множенням блоків і $2p^2l^2n^2$ операцій додавання, що відповідають додаванню блоків). Алгоритми (3.7), (3.8) є більш економічними, ніж алгоритми (3.3), (3.4).

Блочний алгоритм з використанням специфіки блочно-тепліцевих матриць. Серед блочно-тепліцевих матриць зустрічаються такі, блоки яких

мають певну специфіку, яка дає можливість більш економічної реалізації таких алгоритмів для матриць спеціальних підкласів.

В якості прикладу розглянемо блочно-теплицеву λ -матрицю порядку n , що складається з $P \times P$ блоків $P_{ij}(\lambda) = P_{i-j}(\lambda)$, які пов'язані між собою такими співвідношеннями

$$P_{ij}(\lambda) = \mu P_j(\lambda) \eta, P_i(\lambda) = \mu, 0 \leq i \leq n-1 \quad (3.9)$$

де μ, η – квадратні блоки порядку P , такі, що

$$\mu^2 = \eta^2 = I. \quad (3.10)$$

Зокрема, матриці μ і η можуть бути симетричними матрицями перестановок.

У випадку, коли блоки матриці $A(\lambda)$ задовольняють умовам (3.9), (3.10), алгоритм (3.7) запишемо у вигляді

$$\text{для } k=0 \quad Q_0(\lambda) = P_0(\lambda), \quad \tilde{\beta}_0^{(0)}(\lambda) = P_0^{-1}(\lambda) Q_0^{-1}(\lambda),$$

для $k=1, \dots, n-1$

$$\tilde{R}_k(\lambda) = P_{-1}(\lambda) \tilde{\beta}_0^{(k-1)}(\lambda) + P_{-2}(\lambda) \tilde{\beta}_1^{(k-1)}(\lambda) + \dots + P_{-k}(\lambda) \tilde{\beta}_{k-1}^{(k-1)}(\lambda),$$

$$\tilde{U}_k(\lambda) = -Q_{k-1}(\lambda) \mu \tilde{R}_k(\lambda), \quad Q_k(\lambda) = (I - \tilde{U}_k^2(\lambda))^{-1} Q_{k-1}(\lambda),$$

$$\begin{bmatrix} \tilde{\beta}_0^{(k)}(\lambda) \\ \tilde{\beta}_1^{(k)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k)}(\lambda) \\ \tilde{\beta}_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \eta \tilde{\beta}_0^{(k-1)}(\lambda) \\ \eta \tilde{\beta}_1^{(k-1)}(\lambda) \\ \eta \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} \tilde{U}_k(\lambda) + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix}. \quad (3.11)$$

Аналогічно можна записати модифікований алгоритм (3.8) для матриці, блоки якої задовольняють умовам (3.9), (3.10)

$$\text{для } k=0 \quad \hat{Q}_0(\lambda) = P_0(\lambda), \quad \tilde{\psi}_0^{(0)}(\lambda) = \hat{Q}_0^{-1}(\lambda) P_0^{-1}(\lambda),$$

для $k = 1, \dots, n - 1$

$$\begin{aligned} \tilde{R}_k(\lambda) &= \tilde{\psi}_0^{(k-1)}(\lambda)P_1(\lambda) + \tilde{\psi}_1^{(k-1)}(\lambda)P_2(\lambda) + \dots + \tilde{\psi}_{k-1}^{(k-1)}(\lambda)P_k(\lambda), \\ \tilde{U}_k(\lambda) &= -\tilde{R}_k(\lambda)\eta\hat{Q}_{k-1}(\lambda), \quad \hat{Q}_k(\lambda) = \hat{Q}_{k-1}(\lambda)\left(I - \tilde{U}_k^2(\lambda)\right)^{-1}, \end{aligned} \quad (3.12)$$

$$\begin{bmatrix} \tilde{\psi}_0^{(k)}(\lambda) \\ \tilde{\psi}_1^{(k)}(\lambda) \\ \dots \\ \tilde{\psi}_{k-1}^{(k)}(\lambda) \\ \tilde{\psi}_k^{(k)}(\lambda) \end{bmatrix} = \tilde{U}_k(\lambda) \begin{bmatrix} \tilde{\psi}_0^{(k-1)}(\lambda)\mu \\ \tilde{\psi}_1^{(k-1)}(\lambda)\mu \\ \dots \\ \tilde{\psi}_{k-1}^{(k-1)}(\lambda)\mu \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\psi}_0^{(k-1)}(\lambda) \\ \dots \\ \tilde{\psi}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\psi}_{k-1}^{(k-1)}(\lambda) \end{bmatrix}.$$

Слід відзначити, що в процесах (3.11) і (3.12) обчислюються блоки з поліноміальними елементами степеня не вище ln . В загальному випадку

$$\deg[\beta_i^{(k)}(\lambda)] = \deg[\psi_i^{(k)}(\lambda)] = (k + 1)l.$$

Оцінки необхідної пам'яті та складності алгоритмів (3.9) і (3.10).

Алгоритми (3.9), (3.10) вимагають наявності однакової пам'яті комп'ютера, де необхідно зберігати p^3nl^2 чисел для кожного алгоритму та інформацію про початкову матрицю, що потребує p^3nl слів машинної пам'яті.

Для реалізації кожного з алгоритмів необхідно виконати $p^3n^2l^2$ операцій множення і стільки ж операцій додавання-віднімання.

Алгоритми обчислення розв'язків СЛАР з блочно-тепліцевими λ -матрицями.

Нехай розв'язується система (3.1). Порівняно з двоетапним процесом в ряді випадків може давати перевагу блочний варіант методу Тиртишнікова для систем з дійсними елементами [21]. Застосуємо цей алгоритм для розв'язування систем з тепліцевими λ -матрицями. Його обчислювальна схема

$$\text{для } k = 0 \quad Z_0^{(0)}(\lambda) = P_0^{-1}(\lambda)B_0(\lambda),$$

для $k = 1, \dots, n - 1$

$$D_k(\lambda) = B_k(\lambda) - P_k(\lambda)Z_0^{(k-1)}(\lambda) - P_{k-1}(\lambda)Z_1^{(k-1)}(\lambda) - \dots - P_1(\lambda)Z_{k-1}^{(k-1)}(\lambda),$$

$$W_k(\lambda) = Q_k(\lambda)D_k(\lambda),$$

$$\begin{bmatrix} Z_0^{(k)}(\lambda) \\ Z_1^{(k)}(\lambda) \\ \dots \\ Z_{k-1}^{(k)}(\lambda) \\ Z_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} Z_0^{(k-1)}(\lambda) \\ Z_1^{(k-1)}(\lambda) \\ \dots \\ Z_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} \tilde{\beta}_0^{(k)}(\lambda) \\ \tilde{\beta}_1^{(k)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-1}^{(k)}(\lambda) \\ \tilde{\beta}_k^{(k)}(\lambda) \end{bmatrix} W_k(\lambda). \quad (3.13)$$

Оцінки складності алгоритму (3.13). Реалізація алгоритму (3.13) вимагає виконання $p^2 l^2 n^2$ операцій множення і стільки ж операцій додавання-віднімання. Але загальні затрати, пов'язані з розв'язуванням системи $A(\lambda)Z(\lambda) = B(\lambda)$, будуть включати додатково і ту обчислювальну роботу, яка проводиться в алгоритмах (3.7) або (3.3), тобто виконання $(3p^3 + p^2)l^2 n^2$ операцій множення і $3p^3 l^2 n^2$ операцій додавання-віднімання з використанням алгоритму (3.3) та $(2p^3 + p^2)l^2 n^2$ операцій множення і $(2p^3 - p^2)l^2 n^2$ операцій додавання-віднімання, якщо проводити обчислення першого та останнього блочних стовпців оберненої матриці за алгоритмом (3.7).

Блочний алгоритм з нормуванням. Запишемо повністю обчислювальну схему блочного варіанту методу Тиртишнікова для систем дійсними та комплексними елементами [21], який дає переваги при розв'язуванні систем вигляду $A(\lambda)Z(\lambda) = B(\lambda)$ з клітково-теплицевою матрицею.

При $k = 0$ стовпці $\alpha_i^{(k)}(\lambda)$ і $\beta_i^{(k)}(\lambda)$, $0 \leq i \leq n$ співпадають і містять лише одну клітку $P_0^{-1}(\lambda)$, а $Z_0^{(0)}(\lambda) = P_0^{-1}(\lambda)B_0(\lambda)$.

Для відшукування першого та останнього блочних стовпців оберненої матриці застосуємо алгоритм з нормуванням.

$$G_0(\lambda) = Q_0(\lambda) = P_0^{-1}(\lambda), \quad \tilde{\alpha}_0^{(0)}(\lambda) = P_0^{-1}(\lambda)G_0^{-1}(\lambda), \quad \tilde{\beta}_0^{(0)}(\lambda) = P_0^{-1}(\lambda)Q_0^{-1}(\lambda).$$

Припустимо, що відомі $\tilde{\alpha}_i^{(k-1)}(\lambda)$ і $\tilde{\beta}_i^{(k-1)}(\lambda)$ $0 \leq i \leq n$. Будемо шукати $\tilde{\alpha}_i^{(k)}(\lambda)$ і $\tilde{\beta}_i^{(k)}(\lambda)$ у вигляді

$$\begin{bmatrix} \tilde{\alpha}_0^{(k)}(\lambda) \\ \tilde{\alpha}_1^{(k)}(\lambda) \\ \dots \\ \tilde{\alpha}_{k-1}^{(k)}(\lambda) \\ \tilde{\alpha}_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(k-1)}(\lambda) \\ \tilde{\alpha}_1^{(k-1)}(\lambda) \\ \dots \\ \tilde{\alpha}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \tilde{\beta}_1^{(k-1)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix} S_k(\lambda),$$

$$\begin{bmatrix} \tilde{\beta}_0^{(k)}(\lambda) \\ \tilde{\beta}_1^{(k)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-1}^{(k)}(\lambda) \\ \tilde{\beta}_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(k-1)}(\lambda) \\ \tilde{\alpha}_1^{(k-1)}(\lambda) \\ \dots \\ \tilde{\alpha}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} V_k(\lambda) + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \tilde{\beta}_1^{(k-1)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix},$$

де $S_k(\lambda)$, $V_k(\lambda)$ – деякі матриці порядку P . Оскільки $\alpha_i^{(k)}(\lambda)$ і $\beta_i^{(k)}(\lambda)$ $0 \leq i \leq n-1$ є клітковими стовпцями матриці $P_{k-1}(\lambda)$, то

$$P_k(\lambda)\alpha_i^{(k)}(\lambda) = \begin{bmatrix} I \\ 0 \\ \dots \\ 0 \end{bmatrix}, \quad P_k(\lambda)\beta_i^{(k)}(\lambda) = \begin{bmatrix} 0 \\ 0 \\ \dots \\ I \end{bmatrix}.$$

Тут I – одинична матриця порядку P . Позначимо

$$\begin{aligned} \tilde{F}_k(\lambda) &= P_k(\lambda)\tilde{\alpha}_0^{(k-1)}(\lambda) + P_{k-1}(\lambda)\tilde{\alpha}_1^{(k-1)}(\lambda) + \dots + P_1(\lambda)\tilde{\alpha}_{k-1}^{(k-1)}(\lambda), \\ \tilde{R}_k(\lambda) &= P_{-1}(\lambda)\tilde{\beta}_0^{(k-1)}(\lambda) + P_{-2}(\lambda)\tilde{\beta}_1^{(k-1)}(\lambda) + \dots + P_{-k}(\lambda)\tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{aligned}$$

і з останніх співвідношень отримаємо

$$\begin{aligned} S_k(\lambda) &= -Q_{k-1}(\lambda)\tilde{F}_k(\lambda), & V_k(\lambda) &= -G_{k-1}(\lambda)\tilde{R}_k(\lambda), \\ G_k(\lambda) &= (I - V_k(\lambda)S_k(\lambda))^{-1}G_{k-1}(\lambda), & Q_k(\lambda) &= (I - S_k(\lambda)V_k(\lambda))^{-1}Q_{k-1}(\lambda). \end{aligned}$$

Отже, для матриць $A_0^{-1}(\lambda), \dots, A_{n-1}^{-1}(\lambda)$ можна послідовно обчислити перший та останній кліткові стовпці, не обчислюючи елементи її інших стовпців.

Розглянемо відсічені системи $A_k(\lambda)Z^{(k)}(\lambda) = B^{(k)}(\lambda)$, $k = 1, \dots, n$. Нехай

$$\begin{bmatrix} Z_0^{(k)}(\lambda) \\ Z_1^{(k)}(\lambda) \\ \dots \\ Z_{k-1}^{(k)}(\lambda) \\ Z_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} Z_0^{(k-1)}(\lambda) \\ Z_1^{(k-1)}(\lambda) \\ \dots \\ Z_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} t_0^{(k)}(\lambda) \\ t_1^{(k)}(\lambda) \\ \dots \\ t_{k-1}^{(k)}(\lambda) \\ t_k^{(k)}(\lambda) \end{bmatrix}.$$

Підставимо цей вираз в рівняння $A_k(\lambda)Z^{(k)}(\lambda) = B^{(k)}(\lambda)$ і враховуючи, що вектор $Z^{(k-1)}(\lambda)$ задовольняє рівнянню $A_{k-1}(\lambda)Z^{(k-1)}(\lambda) = B^{(k-1)}(\lambda)$, робимо висновок, що вектор поправки $t^{(k)}(\lambda)$ з елементами $t_0^{(k)}(\lambda), \dots, t_k^{(k)}(\lambda)$, є розв'язком системи $A_k(\lambda)Z^{(k)}(\lambda) = d_k(\lambda)$, де

$$d_k(\lambda) = \begin{bmatrix} 0 \\ \dots \\ 0 \\ D_k \end{bmatrix},$$

$$D_k(\lambda) = B_k(\lambda) - P_k(\lambda)Z_0^{(k-1)}(\lambda) - P_{k-1}(\lambda)Z_1^{(k-1)}(\lambda) - \dots - P_1(\lambda)Z_{k-1}^{(k-1)}(\lambda).$$

Позначимо $W_k(\lambda) = Q_k(\lambda)D_k(\lambda)$ і остаточно отримаємо

$$\begin{bmatrix} Z_0^{(k)}(\lambda) \\ Z_1^{(k)}(\lambda) \\ \dots \\ Z_{k-1}^{(k)}(\lambda) \\ Z_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} Z_0^{(k-1)}(\lambda) \\ Z_1^{(k-1)}(\lambda) \\ \dots \\ Z_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} \tilde{\beta}_0^{(k)}(\lambda) \\ \tilde{\beta}_1^{(k)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-1}^{(k)}(\lambda) \\ \tilde{\beta}_k^{(k)}(\lambda) \end{bmatrix} W_k(\lambda).$$

Тут $D_k(\lambda)$ і $W_k(\lambda)$ – вектор-стовпці довжиною p ; $Q_k(\lambda)$ і $\tilde{\beta}_i(\lambda)$, $0 \leq i \leq p$ – $p \times p$ -блоки такі, що блоки $\tilde{\beta}_i(\lambda)Q_k(\lambda)$ формують останній блочний стовпець матриці $P_{k-1}(\lambda)$. Вектор $t_i^{(k)}(\lambda)$ є лінійною комбінацією останніх стовпців матриці $A_{k-1}(\lambda)$, координати вектора $d_k(\lambda)$ є

коефіцієнтами даної лінійної комбінації. Отже, для знаходження розв'язку системи $A(\lambda)Z(\lambda) = B(\lambda)$ достатньо рекурентно обчислити останній блочний стовпець матриці $P_{k-1}(\lambda)$.

Процес реалізує обчислення векторних стовпців, елементами яких є поліноми степеня ln . В загальному випадку $\deg[Z_i^{(k)}(\lambda)] = (k+1)l$.

Оцінки складності алгоритмів. Розглянутий метод розв'язування систем з клітково-тепліцевими матрицями досить ефективний порівняно з іншими прямими методами. Складність методу для СЛАР з тепліцевими λ -матрицями при реалізації на комп'ютері складає порядку $O(l^2 n^4)$ в загальному випадку, $O(4p^3 l^2 n^3)$ для кліткового аналогу і $O(4p^3 l^2 n^2)$ для блочного алгоритму з нормуванням.

Блочні алгоритми є економнішими порівняно зі скалярними, оскільки в них головний член операцій додавання залежить лише від кількості множень блоків.

3.2. Використання додаткової пам'яті та паралельні алгоритми для розв'язування СЛАР з тепліцевими λ -матрицями

3.2.1. Використання додаткової пам'яті при розв'язуванні СЛАР з блочно-тепліцевими λ -матрицями

При розв'язуванні на традиційному однопроцесорному комп'ютері СЛАР з матрицями великої розмірності виникає необхідність запам'ятовувати набагато більше інформації, ніж можна розмістити в частині оперативної пам'яті (ОП), призначеній для користувача. В цьому випадку виникає потреба використання зовнішньої пам'яті [21, 75].

Розглянемо задачу розв'язування СЛАР з блочно-тепліцевими λ -матрицями з поліноміальними елементами швидкими алгоритмами (які приведені в підрозд. 3.1), коли обмін даними здійснюється між ЗП, ОП і кеш-пам'яттю [38].

В роботі [75] проаналізовано обмін даними між двома рівнями пам'яті – ОП та зовнішньою пам'яттю. Використання кеш-пам'яті для прискорення процесу розв'язування великих СЛАР в літературі детально ще не розглядалось. Тому доцільно дослідити всі переваги саме цього процесу.

Розіб'ємо дану СЛАР з тепліцевою λ -матрицею на m квадратних блоків

$$\begin{pmatrix} P_1(\lambda) & P_2(\lambda) & P_3(\lambda) & \dots & P_m(\lambda) \\ P_0(\lambda) & P_1(\lambda) & P_2(\lambda) & \dots & P_{m+1}(\lambda) \\ P_{-1}(\lambda) & P_0(\lambda) & P_1(\lambda) & \dots & P_{m+2}(\lambda) \\ \dots & \dots & \dots & \dots & \dots \\ P_m(\lambda) & P_{m-1}(\lambda) & P_{m-2}(\lambda) & \dots & P_1(\lambda) \end{pmatrix} \begin{pmatrix} Z_1^{(k)}(\lambda) \\ Z_2^{(k)}(\lambda) \\ \dots \\ Z_{m-1}^{(k)}(\lambda) \\ Z_m^{(k)}(\lambda) \end{pmatrix} = \begin{pmatrix} B_1(\lambda) \\ B_2(\lambda) \\ \dots \\ B_{m-1}(\lambda) \\ B_m(\lambda) \end{pmatrix}. \quad (3.14)$$

Система (3.1) може бути розв'язана за допомогою сукупності рекурентних співвідношень (2.8), з використанням алгоритму з нормуванням для обчислення першого та останнього стовпців оберненої матриці

$$\tilde{F}_k(\lambda) = \sum_{l=k}^{k-1} P_l(\lambda) \tilde{\alpha}_{l-k}^{(k-1)}(\lambda), \quad \tilde{R}_k(\lambda) = \sum_{l=1}^{k-1} P_{-l}(\lambda) \tilde{\beta}_{l-1}^{(k-1)}(\lambda),$$

$$\tilde{S}_k(\lambda) = -\tilde{F}_k(\lambda) G_{k-1}(\lambda), \quad \tilde{V}_k(\lambda) = -\tilde{R}_k(\lambda) Q_{k-1}(\lambda), \quad \tilde{U}_k(\lambda) = \frac{1}{1 - \tilde{S}_k(\lambda) \tilde{V}_k(\lambda)},$$

$$G_k(\lambda) = G_{k-1}(\lambda) \tilde{U}_k(\lambda), \quad Q_k(\lambda) = Q_{k-1}(\lambda) \tilde{U}_k(\lambda),$$

$$\begin{bmatrix} \tilde{\alpha}_0^{(k)}(\lambda) \\ \tilde{\alpha}_1^{(k)}(\lambda) \\ \dots \\ \tilde{\alpha}_{k-1}^{(k)}(\lambda) \\ \tilde{\alpha}_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(k-1)}(\lambda) \\ \tilde{\alpha}_1^{(k-1)}(\lambda) \\ \dots \\ \tilde{\alpha}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix} \tilde{S}_k(\lambda),$$

$$\begin{bmatrix} \tilde{\beta}_0^{(k)}(\lambda) \\ \tilde{\beta}_1^{(k)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-1}^{(k)}(\lambda) \\ \tilde{\beta}_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^{(k-1)}(\lambda) \\ \tilde{\alpha}_1^{(k-1)}(\lambda) \\ \dots \\ \tilde{\alpha}_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} \tilde{V}_k(\lambda) + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix}.$$

$$D_k(\lambda) = B_k(\lambda) - P_k(\lambda) Z_0^{(k-1)}(\lambda) - P_{k-1}(\lambda) Z_1^{(k-1)}(\lambda) - \dots - P_1(\lambda) Z_{k-1}^{(k-1)}(\lambda),$$

$$W_k(\lambda) = D_k(\lambda) Q_k(\lambda),$$

$$\begin{bmatrix} Z_0^{(k)}(\lambda) \\ Z_1^{(k)}(\lambda) \\ \dots \\ Z_{k-1}^{(k)}(\lambda) \\ Z_k^{(k)}(\lambda) \end{bmatrix} = \begin{bmatrix} Z_0^{(k-1)}(\lambda) \\ Z_1^{(k-1)}(\lambda) \\ \dots \\ Z_{k-1}^{(k-1)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\beta}_0^{(k-1)}(\lambda) \\ \dots \\ \tilde{\beta}_{k-2}^{(k-1)}(\lambda) \\ \tilde{\beta}_{k-1}^{(k-1)}(\lambda) \end{bmatrix} W_k(\lambda).$$

Як було встановлено раніше, кількість операцій, необхідних для розв'язання даної системи за алгоритмом (2.5) з точністю до головного члена складає $Q \cong 6l^2n^2$. В загальному випадку для реалізації даного методу треба виконати $6l^2m^2$ обмінів.

Припустимо, що для організації обмінів відведена частина ОП, достатня для зберігання L слів, не враховуючи пам'яті для збереження програми.

Розділимо дані порівну між ЗП і кеш. Позначимо через τ і ν відповідно середній час очікування та вибірки одного машинного слова із ЗП. Тоді загальний час T_1 , який треба затратити на обмін L словами між ОП і ЗП, можна обчислити як $T_1 = \tau + L\nu$.

Середній час, що витрачається на вибірку одного машинного слова даних з кеш, можна обчислити за формулою $t_{\text{сер.}} = t_{\text{кеш}} + \frac{t_{\text{ОПбл.}}}{k_{\text{сер.}}}$, де $t_{\text{кеш}}$ – час зчитування одного слова з кеш, $t_{\text{ОПбл.}}$ – час зчитування блоку з ОП, $k_{\text{сер.}}$ – середня кількість звертань до кеш між двома послідовними звертаннями до ОП.

Для обміну L словами між ОП та кеш треба затратити час $T_2 = Lt_{\text{сер.}}$. Для реалізації даного методу на обмін даними необхідно витратити час

$$T_{\text{заг.}} = \left(6 \left[\left(\frac{m}{2} \right)^2 + p^2 \right] \tau + 6 \left[\left(\frac{m}{2} \right)^2 + p^2 \right] L\nu + 6 \left[\left(m - \frac{m}{2} \right)^2 + p^2 \right] Lt_{\text{сер.}} \right) l^2$$

Однією з найважливіших характеристик алгоритму є його висота. *Висотою алгоритму* називається кількість тактів, за які може бути отриманий результат на багатопроцесорній(ому) ЕОМ (комп'ютері).

Висота алгоритму характеризує кількість рівнів або ярусів обчислювальної схеми. Задача побудови паралельного алгоритму полягає в перетворенні початкового алгоритму таким чином, щоб зменшити його висоту. Цим самим забезпечується можливість виконання обчислень на багатопроцесорному комп'ютері (ЕОМ) за якомога менший час, що в деяких випадках має вирішальне значення.

Прискоренням S_p називають відношення часу реалізації деякого алгоритму на одному процесорі T_1 паралельної обчислювальної системи до

часу T_p його реалізації на p процесорах $S_p = \frac{T_1}{T_p}$. *Економічністю* E_p

називають відношення $E_p = \frac{S_p}{p}$

Оскільки час обміну даними між ОП і кеш суттєво менший за час обміну інформацією між ОП та ЗП (різниця порядку 10^{-9}), то основний інтерес для нас становить саме мінімізація часу

$$T = \left(6 \left[\left(\frac{m}{2} \right)^2 + p^2 \right] \tau + 6 \left[\left(\frac{m}{2} \right)^2 + p^2 \right] Lv \right) l^2.$$

Функція $T(p)$ досягає мінімуму при $p = \sqrt{\frac{n}{2}}$. Покладемо $L = 5p^2$ і отримаємо $T = 6(n\tau + 2,5n^2v)l^2 = (6n\tau + 15n^2v)l^2$.

Коефіцієнт втрат можна обчислити за формулою

$$E = \frac{6n^2 + 6n\tau + 2,5n^2v}{6n^2} \rightarrow 1.$$

Якщо ж для відшукування першого та останнього стовпців оберненої матриці використовувати алгоритм (2.1), то $Q \cong 7l^2n^2$. Для реалізації даного методу треба виконати $7l^2m^2$ обмінів. Далі підрахуємо

$$T_{\text{заг.}} = \left(7 \left[\left(\frac{m}{2} \right)^2 + p^2 \right] \tau + 7 \left[\left(\frac{m}{2} \right)^2 + p^2 \right] Lv \right) l^2,$$

$$T = \left(7 \left[\left(\frac{m}{2} \right)^2 + p^2 \right] \tau + 7 \left[\left(\frac{m}{2} \right)^2 + p^2 \right] Lv \right) l^2,$$

$$T = 7(n\tau + 2,5n^2v)l^2 = (7n\tau + 17,5n^2v)l^2.$$

Коефіцієнт втрат можна обчислити за формулою

$$E = \frac{7n^2 + 7n\tau + 17,5n^2v}{7n^2} \rightarrow 1.$$

Розглянемо випадок, коли за тих самих умов в ЗП та кеш зберігається не однакова кількість даних. Розмістимо їх s -ту частину в ЗП, а решту – в кеш. Отримаємо такі результати

$$T_{\text{заг.}} = 6 \left[\left(\frac{m}{s} \right)^2 + p^2 \right] \tau l^2 + 6 \left[\left(\frac{m}{s} \right)^2 + p^2 \right] Lv l^2 + 6 \left[\left(m - \frac{m}{s} \right)^2 + p^2 \right] l^2 Lt_{\text{сер.}}$$

для алгоритму з нормуванням

$$T_{\text{заг.}} = 7 \left[\left(\frac{m}{s} \right)^2 + p^2 \right] \tau l^2 + 7 \left[\left(\frac{m}{s} \right)^2 + p^2 \right] Lv l^2 + 7 \left[\left(m - \frac{m}{s} \right)^2 + p^2 \right] l^2 Lt_{\text{сер.}}$$

для алгоритму без нормування.

Оскільки час обміну даними між ОП і кеш суттєво менший за час обміну інформацією між ОП та ЗП (різниця порядку 10^{-9}), то основний інтерес для нас становить саме мінімізація часу відповідно

$$T = \left(6 \left[\left(\frac{m}{s} \right)^2 + p^2 \right] \tau + 6 \left[\left(\frac{m}{s} \right)^2 + p^2 \right] Lv \right) l^2$$

або

$$T = \left(7 \left[\left(\frac{m}{s} \right)^2 + p^2 \right] \tau + 7 \left[\left(\frac{m}{s} \right)^2 + p^2 \right] Lv \right) l^2.$$

Функція $T(p)$ досягає мінімуму при $p = \sqrt{\frac{n}{s}}$. Покладемо $L=5p^2$ і

отримаємо $T = 6 \left(\frac{2}{s} n \tau + \frac{10}{s^2} n^2 v \right) l^2$ для алгоритму (2.5) та

$T = 7 \left(\frac{2}{s} n \tau + \frac{10}{s^2} n^2 v \right) l^2$ для алгоритму без нормування.

Коефіцієнт втрат для кожного з методів можна обчислити за формулою

$$E = \frac{6n^2 + \frac{12}{s} n \tau + \frac{60}{s^2} n^2 v}{6n^2} \rightarrow 1 \quad \text{та} \quad E = \frac{7n^3 + \frac{14}{s} n \tau + \frac{70}{s^{5/2}} n^2 v}{7n^2} \rightarrow 1 \rightarrow 1.$$

Продуктивність і обчислювальні можливості ЕОМ (комп'ютера) в значній мірі визначаються складом і характеристиками ЗП. В складі ЕОМ використовуються одночасно кілька типів пам'яті, які відрізняються принципами дії, характеристиками і призначенням.

Розглянемо випадок, коли при реалізації аналогу методу Тиртишнікова для систем з λ -матрицями на комп'ютері використовуються k деяких ЗП; дані між ними розподілені порівну. Тоді для розв'язування задачі за

алгоритмом (2.5) необхідно витратити час $T = 6\left(\frac{m^2}{k^2} + p^2\right)(\tau + Lv)l^2$ (час обміну інформацією між ОП і кожним з даних ЗП вважаються несуттєво відмінними).

Мінімум функції $T(p)$ досягається при $p = \sqrt{\frac{n}{k}}$. При $L = 5p^2$

отримаємо $T = \left(\frac{12n}{k}\tau + \frac{60n^2}{k^2}\nu\right)l^2$.

Обчислимо коефіцієнт втрат $E = \frac{6n^2 + \frac{12n}{k}\tau + \frac{60n^2}{k^2}\nu}{6n^2} \rightarrow 1$

Для алгоритму без нормування отримаємо $T = \left(\frac{m^2}{k^2} + p^2\right)(\tau + Lv)l^2k$,

$T = \left(\frac{14n}{k}\tau + \frac{70n^2}{k^2}\nu\right)l^2$, $E = \frac{7n^2 + \frac{14n}{k}\tau + \frac{70n^2}{k^2}\nu}{7n^2} \rightarrow 1$.

3.2.2. Паралельні моделі для реалізації алгоритмів розв'язування СЛАР з теплицевими λ -матрицями

В наш час широко розповсюджені багатопроцесорні комп'ютери (ЕОМ). Їх опис та характеристики дано в [22, 67, 88]. Розробляються і створюються нові багатопроцесорні комп'ютери. Для їх ефективного використання необхідна наявність алгоритмів з паралельною організацією обчислень. Це потребує розробки нових алгоритмів або вдосконалення вже існуючих для їх ефективної реалізації на багатопроцесорних комп'ютерах.

При побудові паралельних обчислень важливою ознакою є властивий їй режим (найчастіше це SIMD або MIMD). При врахуванні обмежень можливостей і ресурсів з певним режимом роботи багатопроцесорного

комп'ютера алгоритми розв'язування задач залежать від його структури та архітектури [90].

Є багато публікацій, присвячених паралельним обчисленням в задачах лінійної алгебри [15, 67, 79, 88, 90, 102]. Детальний аналіз розпаралелення ітераційних методів проведено в [79]. В [75] проведено розпаралелення прямих кліткових методів розв'язування СЛАР.

Дослідимо паралельність швидких алгоритмів розв'язування СЛАР з теплицевими λ -матрицями з поліноміальними елементами для двох моделей - концепції необмеженого паралелізму [79] і для багатопроцесорного комп'ютера типу MIMD [75].

Вибір MIMD-архітектури зумовлений тим, що обчислювальні системи цього типу є найбільш поширеними серед паралельних комп'ютерів. Але їх використання передбачає певні обмеження при складанні алгоритму.

Концепція необмеженого паралелізму. Нехай для реалізації алгоритму є p -процесорна обчислювальна система. В основі концепції необмеженого паралелізму лежить припущення про використання такої ідеалізованої моделі паралельної обчислювальної системи, яка має необхідну кількість ідентичних процесорів, а також як завгодно велику оперативну пам'ять, одночасно доступну для всіх процесорів. Кожний процесор за одиницю часу може виконати достатньо точно довільну одномісну або двомісну операцію. Час виконання допоміжних операцій, а також час взаємодії з пам'яттю і час, затрачений на управління процесом, вважаються як завгодно малими. Ніякі конфлікти при звертанні процесорів до спільної пам'яті не виникають. Кожний процесор зчитує свої операнди з пам'яті і після виконання операції записує туди результат. Після закінчення обчислювального процесу всі результати залишаються в пам'яті.

Запропонований алгоритм з нормуванням визначення розв'язків СЛАР з теплицевими λ -матрицями, елементами яких є поліноми степеня l , може бути ефективно реалізований на багатопроцесорному комп'ютері. Його реалізація буде здійснюватись по ярусах, і на кожному ярусі паралельних

обчислень усіма паралельними процесорами виконуватимуться однакові арифметичні операції – додавання, віднімання, ділення та множення.

Розглядаємо k -й крок реалізації алгоритму. Обчислення всіх $\alpha_i^{(k)}(\lambda)$, $\beta_i^{(k)}(\lambda)$, та $Z_i^{(k)}(\lambda)$, $i = 1, 2, \dots, k - 1$ можна визначити незалежно, при повному завантаженні всіх P процесорів, і в такий спосіб висота алгоритму зменшиться в P разів $T_p = \frac{6n^2l^2}{P}$.

Час реалізації даного алгоритму на одному процесорі складає $T_1 = 6n^2l^2$. Отже, на P -процесорній обчислювальній системі буде досягнуто прискорення

$$S_p = \frac{T_1}{T_p} = \frac{6n^2l^2}{\frac{6n^2l^2}{P}} = P \text{ при ефективності } E_p = \frac{S_p}{P} = 1.$$

Аналогічні результати отримаємо і для алгоритму без нормування.

Наступні дослідження проведемо для машин класу MIMD [75] за умов обмеженості обчислювальних ресурсів і технічних можливостей.

Нехай є деякий універсальний серійний комп'ютер, розміщений на периферії. Він здійснює обслуговування паралельної обчислювальної системи і управління зовнішніми пристроями. В системі існує комутатор зв'язку, до каналів якого підключені управляючі пристрої (УП), арифметичні процесори (АП) і спеціальні процесори (СП).

Обчислювальний процес за загальною програмою організується за допомогою периферійного універсального комп'ютера. Будемо припускати, що АП мають певний, відносно великий об'єм оперативної пам'яті і здатні виконувати чисельну та логічну обробку інформації в автономному і керованому режимах. Оператори програми, управляючі інструкції та інформація, яка обробляється, поступають в АП або по мірі необхідності, або у випадку звільнення модулів у обчислювальному процесі.

Комутаційні процесори (КП) здійснюють з'єднання модулів для обміну даними. Передача інформації проводиться в напівдуплексному режимі. Можливі попарні з'єднання довільної кількості АП.

Організація обчислювального процесу по загальній програмі, узгодження роботи АП і управління комутаційними і периферійними процесорами виконується з допомогою управляючих процесорів.

Управляючий комп'ютер керує всіма периферійними пристроями (дисководами, моніторами, принтерами і т.д.) і виконує роль своєрідного буфера при обміні інформацією між АП і зовнішньою пам'яттю. На відміну від АП, управляюча машина не може проводити обмін даними одночасно з m АП (m – кількість каналів зв'язку між АП і периферійною ЕОМ. Є ще кілька каналів для зв'язку з периферійними пристроями, які можуть використовуватись одночасно).

Така багатопроцесорна ОС дозволяє виконувати одночасно обмін даними між ЗП і кількома периферійними процесорами (до m включно), а також проводити незалежну обробку даних в решті АП.

Далі будемо користуватись такими позначеннями: M – об'єм оперативної пам'яті АП, N – об'єм оперативної пам'яті управляючої машини (в словах). Нехай t_c – час синхронізації (настройки) системи для обміну інформацією, t_1 – час пошуку певного набору даних у зовнішній пам'яті, t_2 – час передачі одного слова з зовнішньої пам'яті в управляючу машину, t_3 – середній час комутаційних затримок при встановленні зв'язку між двома модулями, t_{AA} – час передачі одного слова між двома модулями, t – середній час виконання однієї арифметичної операції і нехай $\tau = \frac{t_{AA}}{t}$.

Обмін інформацією ПОС може проводитись одним з описаних нижче способів [75].

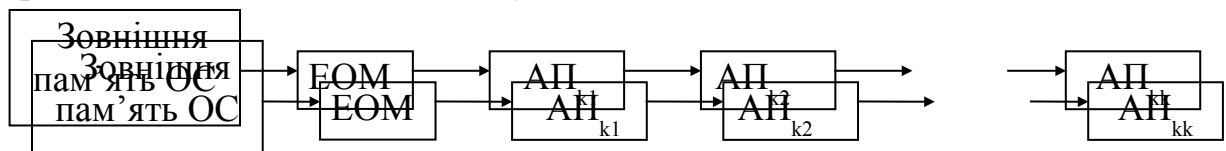
Ланцюговий принцип передачі даних полягає в тому, що відбувається передача даних у всі процесори групи, що проводять обробку інформації.

Обмін даними виконується за таким правилом: АП, що отримав набір даних при обміні з деякими іншими АП, виконує передачу даних іншим АП, не починаючи обробку власної інформації, аж до закінчення процесу інформаційного обміну. При кожному наступному обміні кількість АП, які передають інформацію, збільшується вдвічі.

Принцип привілейованої передачі даних – організація обмінів даними між АП, при якому допоміжний процесор після передачі даних в деякий модуль відразу починає обробку власної інформації. Аналогічним чином діють всі модулі, що входять в дану групу АП.

Централізований принцип передачі даних – це процес передачі інформації з управляючої машини одночасно в r АП ($r \leq m$).

Магістральна обробка даних – процедура перетворення наборів даних D_i ($j=1,2,\dots,k$), що знаходяться в АП, за допомогою наборів даних d_j ($j=1,2,\dots,m$), які викликаються або з універсальної ЕОМ, або із зовнішньої пам'яті. При виконанні даної процедури набір даних d_j передається після d_{j-1} за наступною схемою:



Розглянемо тепер особливості застосування описаних процедур передачі даних. Нехай необхідно передати набір даних з допоміжного АП у всі АП групи, кількість яких дорівнює p . В такому випадку при ланцюговій передачі даних необхідно виконати $\lceil \log_2 p \rceil$ переривань обміну. Можна також спочатку передати вказаний набір в універсальний комп'ютер, а потім скористатися централізованим принципом передачі інформації. Якщо $\lceil \log_2 p \rceil < 1 + \left\lceil \frac{p-1}{m} \right\rceil$, то процедура централізованої передачі даних вимагатиме меншої кількості переривань обміну, а отже, і меншого часу. Тут $\lceil m \rceil$ – найменше ціле число, більше за m .

Нехай пам'ять кожного АП зберігає q стрічок і p стовпців вихідної матриці, і при цьому q та p такі, що $2pq = 2n$. Крім того, в пам'яті АП знаходиться k проміжних значень $\alpha_i^{(k)}(\lambda)$, k проміжних значень $\beta_i^{(k)}(\lambda)$ та стільки ж проміжних значень $Z_i^{(k)}(\lambda)$. Якщо в системі використовується централізований принцип передачі даних, то час τ пересилки одного машинного слова з довільного АП через ПМ у всі решту АП задовольняє нерівності $\tau \leq \frac{2p}{n}(t_c + t_{AA})$.

При ланцюговому принципі передача слова з одного АП у всі решта АП займає час $\tau \leq (t_c + t_{AA}) \log_2 p$.

Розглядаємо тепер k -й крок реалізації алгоритму. Обчислення всіх $\tilde{F}_k(\lambda)$, $\tilde{R}_k(\lambda)$, $\tilde{U}_k(\lambda)$, $\tilde{V}_k(\lambda)$ і $\tilde{S}_k(\lambda)$, очевидно, можуть бути виконані без обміну інформацією за час $2(2k-1)tl^2$. Обчислення всіх $\tilde{\alpha}_i^{(k)}(\lambda)$, $\tilde{\beta}_i^{(k)}(\lambda)$ $i = 0, 1, \dots, k-1$, будемо здійснювати за формулами (2.5), для чого затратимо

час $\frac{2(2k-1)l^2t}{p}$, а $Z_i^{(k)}(\lambda)$, $i = 0, 1, \dots, k-1$ знайдемо за формулами (2.13) за час

$\frac{2l^2t}{p}$. Знайдені значення $\tilde{\alpha}_i^{(k)}(\lambda)$, $\tilde{\beta}_i^{(k)}(\lambda)$ та $Z_i^{(k)}(\lambda)$ з відповідного процесора, на якому їх було обчислено на попередньому кроці, передаються за час τ в решту АП, і весь цикл повторюється. По такій схемі всі невідомі $\tilde{\alpha}_i^{(k)}(\lambda)$, $\tilde{\beta}_i^{(k)}(\lambda)$ та $Z_i^{(k)}(\lambda)$, $i = 0, 1, \dots, k-1$ будуть обчислені за час

$3kl^2\tau + \frac{4ktl^2}{p}$. Тому з точністю головного члена можна записати

$$T_p = \frac{6n^2tl^2}{p} + n^2l^2\tau.$$

Отже, за рахунок розпаралелювання досягається прискорення

$$\frac{T_1}{T_p} = \frac{6n^2tl^2}{\frac{6n^2tl^2}{p} + n^2l^2\tau} \text{ при ефективності } E_p = \frac{S_p}{p} = \frac{\frac{6n^2tl^2}{p} + n^2l^2\tau}{p}.$$

Для існуючих обчислювальних систем $\tau \cong 10^{-9}t$, тому існують такі значення n , для яких прискорення $S_p \rightarrow p$, а ефективність $E_p \rightarrow 1$. Отже, для даного алгоритму на комп'ютері типу MIMD вдається добитись практично тих же оцінок, що й для концепції необмеженого паралелізму, причому цей результат має місце як для централізованої, так і для ланцюгової схем обміну між процесорами. Аналогічні результати отримаємо і для алгоритму без нормування.

Проблема розв'язання великих систем лінійних алгебраїчних рівнянь не є тривіальною. В кожному конкретному випадку необхідний попередній аналіз не лише властивостей структури даної системи, а й дослідження фізичних властивостей задачі, яка до неї приводить. Без цього неможливо розв'язати питання вибору відповідного методу розв'язування і довжини машинного слова, що гарантує достовірність отриманих результатів.

Лише після виконання такого аналізу можна переходити до побудови обчислювальних схем з розпаралеленням процесу обробки інформації. На цьому етапі має зміст хоча б приблизно оцінити ефективність розпаралелювання. Лише пристосовуючись до специфіки архітектурної побудови масово-паралельної обчислювальної системи можлива побудова ефективних обчислювальних схем.

При розв'язуванні систем з використанням зовнішньої пам'яті великого значення набуває питання розбиття даних.

Вимоги до швидкодії та об'ємів оперативної пам'яті обчислювальної системи постійно зростають. Темпи їх росту випереджають темпи розвитку обчислювальної техніки. Тому вже зараз є необхідність в створенні

паралельних обчислювальних алгоритмів з використанням зовнішньої пам'яті. Обчислювальні методи алгебри є одним з базових інструментів при моделюванні на комп'ютері, і це важлива частина програмного забезпечення для комп'ютерів всіх поколінь. Тому актуальною є проблема побудови і дослідження стійких, більш економічних алгоритмів розв'язування систем лінійних алгебраїчних рівнянь, ніж ті, що існують на сьогоднішній день.

3.3. Алгоритми розв'язування СЛАР з теплицевими λ -матрицями гіллястими ланцюговими дробами

Розглянемо задачу розв'язування СЛАР з теплицевими λ -матрицями

$$A(\lambda)Z(\lambda) = B(\lambda), \quad (3.15)$$

де елементи матриці $A(\lambda)$ та вектора $B(\lambda)$ – алгебраїчні поліноми степеня l .

Далі мова йтиме про побудову алгоритмів розв'язування СЛАР (3.15) за допомогою гіллястих ланцюгових дробів [11, 69, 70, 82]. Цей підхід дає можливість одержати нові алгоритми розв'язування СЛАР з теплицевими λ -матрицями [42].

Визначення. Попередні результати. При записі розв'язків математичних задач як у класичній математиці, так і в нових її розділах, результат представляється у вигляді деякої послідовності (скінченної або нескінченної) математичних символів. У якості зручного засобу для запису розв'язків часто використовуються ряди, нескінченні ряди та ланцюгові дроби.

Нетривіальним узагальненням ланцюгових дробів є гіллясті ланцюгові дроби (ГЛД), які є композицією

$$\left. \begin{aligned} S_0(w) &= s_0(w), \\ S_m(w) &= S_{m-1}(s_{k_1 k_2 \dots k_m}(w)) \quad (m = 1, 2, \dots) \end{aligned} \right\}$$

дробово-лінійних перетворень (ДЛП) вигляду

$$\left. \begin{aligned} s_0(w) &= w, \\ s_{k_1 k_2 \dots k_m} &= \sum_{k_m=1}^N \frac{a_{k_1 k_2 \dots k_m}}{b_{k_1 k_2 \dots k_m} + w} \end{aligned} \right\}$$

ОЗНАЧЕННЯ 3.1. Нескінченим гіллястим ланцюговим дробом називається композиція $S(w)$ нескінченної кількості дробово-лінійних перетворень.

Далі ГЛД буде записуватися у вигляді

$$D = \sum_{k_1=1}^N \frac{a_{k_1}}{b_{k_1}} + \sum_{k_2=1}^N \frac{a_{k_1 k_2}}{b_{k_1 k_2}} + \sum_{k_3=1}^N \frac{a_{k_1 k_2 k_3}}{b_{k_1 k_2 k_3}} + \dots + \sum_{k_i=1}^N \frac{a_{k_1 k_2 k_3 \dots k_i}}{b_{k_1 k_2 k_3 \dots k_i}} + \dots$$

Кінцевий дріб

$$D_m = \sum_{k_1=1}^N \frac{a_{k_1}}{b_{k_1}} + \sum_{k_2=1}^N \frac{a_{k_1 k_2}}{b_{k_1 k_2}} + \sum_{k_3=1}^N \frac{a_{k_1 k_2 k_3}}{b_{k_1 k_2 k_3}} + \dots + \sum_{k_m=1}^N \frac{a_{k_1 k_2 k_3 \dots k_m}}{b_{k_1 k_2 k_3 \dots k_m}}$$

називається m -м підхідним дробом нескінченного ГЛД.

Далі також будуть використані позначення

$$D_m^{k_1 k_2 k_3 \dots k_i} = b_{k_1 k_2 k_3 \dots k_i} + \sum_{k_{s+1}=1}^N \frac{a_{k_1 k_2 k_3 \dots k_{s+1}}}{b_{k_1 k_2 k_3 \dots k_{s+1}}} + \dots + \sum_{k_m=1}^N \frac{a_{k_1 k_2 k_3 \dots k_m}}{b_{k_1 k_2 k_3 \dots k_m}}.$$

Як відомо, ряди, нескінченні ряди та ланцюгові дроби можна розглядати як окремий випадок композицій ДЛП. Проте, ця корисна особливість компенсується недоліками – композиції ДЛП дуже незручні для запису і тим більше для вивчення через надмірну громіздкість.

Розглянемо представлення композиції дробово-лінійних композицій достатньо загального вигляду гіллястим ланцюговим дробом. З цією метою введемо в розгляд таку сукупність ДЛП

$$\left. \begin{aligned} D_m^0 &= b_0 + \sum_{k_1=1}^N \frac{a_{k_1} + D_m^{k_1}}{b_{k_1} + D_m^{k_1}}; \\ D_m^{k_1} &= b_{k_1} + \sum_{k_2=1}^N \frac{a_{k_1 k_2} + D_m^{k_1 k_2}}{b_{k_1 k_2} + D_m^{k_1 k_2}}; \\ D_m^{k_1 k_2 \dots k_m} &= b_{k_1 k_2 \dots k_m} + \sum_{k_2=1}^N \frac{a_{k_1 k_2 \dots k_m} + D_m^{k_1 k_2 \dots k_i}}{b_{k_1 k_2 \dots k_m} + D_m^{k_1 k_2 \dots k_i}}. \end{aligned} \right\} \cdot \quad (3.16)$$

ТЕОРЕМА 3.1. *Композиція дробово-лінійних композицій (3.16) може бути представлена гіллястим ланцюговим дробом канонічного вигляду і при цьому m -й підхідний ланцюговий дріб буде відповідати композиції m дробово-лінійних перетворень (ДЛП).*

Доведення. Дійсно, на підставі (3.16) можна записати

$$D_m^0 = b_0 + \sum_{k_1=1}^N \frac{b_{k_1} + D_m^{k_1} + a_{k_1} - b_{k_1}}{b_{k_1} + D_m^{k_1}} = b_0 + N + \sum_{k_1=1}^N \frac{a_{k_1} - b_{k_1}}{b_{k_1} + D_m^{k_1}}.$$

Аналогічно

$$\begin{aligned} D_m^{k_1 k_2 \dots k_i} &= b_{k_1 k_2 \dots k_i} + \sum_{k_2=1}^N \frac{b_{k_1 k_2 \dots k_{i+1}} + D_m^{k_1 k_2 \dots k_{i+1}} + a_{k_1 k_2 \dots k_{i+1}} - b_{k_1 k_2 \dots k_{i+1}}}{b_{k_1 k_2 \dots k_{i+1}} + D_m^{k_1 k_2 \dots k_{i+1}}} = \\ &= b_{k_1 k_2 \dots k_i} + N + \sum_{k_{i+1}=1}^N \frac{a_{k_1 k_2 \dots k_{i+1}} - b_{k_1 k_2 \dots k_{i+1}}}{b_{k_1 k_2 \dots k_{i+1}} + D_m^{k_1 k_2 \dots k_{i+1}}} \end{aligned}$$

З урахуванням введених позначень композиція ДЛП може бути подана гіллястим дробом

$$\begin{aligned} D_m^0 &= b_0 + \\ &+ \sum_{k_1=1}^N \frac{a_{k_1} - b_{k_1}}{b_{k_1} + N + \sum_{k_2=1}^N \frac{a_{k_1 k_2} - b_{k_1 k_2}}{b_{k_1 k_2} + N + \dots}} \\ &\quad + \sum_{k_i=1}^N \frac{a_{k_1 k_2 \dots k_i} - b_{k_1 k_2 \dots k_i}}{b_{k_1 k_2 \dots k_i} + N + \dots} \end{aligned}$$

що і було потрібно довести.

Таким чином, багато алгоритмів із достатньо складною структурою можуть бути подані ланцюговими дробами.

Системи лінійних алгебраїчних рівнянь із щільним заповненням матриці.

Розглянемо систему лінійних алгебраїчних рівнянь вигляду

$$Ax = b \quad (3.17)$$

із символічними елементами. Без обмеження загальності, вважається, що $n = 2^s$, де s — ціле число (якщо це не так, то систему можна доповнити новими невідомими $x_{n+1} = 1, \dots, x_{n+k} = 1$).

Розіб'ємо матрицю A на блоки, записавши (3.17) у вигляді

$$\begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}.$$

Тоді для визначення невідомої X_2 на основі формули Шура [20] можна одержати таку систему рівнянь

$$(A_{2,2} - A_{2,1} A_{1,1}^{-1} A_{1,2}) X_2 = B_2 - A_{2,1} A_{1,1}^{-1} B_1$$

З використанням тотожності $PQ^{-1} = [(Q^{-1})^T P^T]^T$ для $A_{2,1} \cdot A_{1,1}^{-1}$ можна записати $A_{2,1} \cdot A_{1,1}^{-1} = [(A_{1,1}^{-1})^T \cdot A_{2,1}^T]^T$.

Або більш детально

$$A_{2,1} \cdot A_{1,1}^{-1} = \left[\begin{pmatrix} a_{1,1} & a_{2,1} & \dots & a_{p-1,1} & a_{p,1} \\ a_{1,2} & a_{2,2} & \dots & a_{p-1,2} & a_{p,2} \\ \dots & \dots & \dots & \dots & \dots \\ a_{1,p} & a_{2,p} & \dots & a_{p-1,p} & a_{p,p} \end{pmatrix}^{-1} \begin{pmatrix} a_{p+1,1} & a_{p+2,1} & \dots & a_{n-1,1} & a_{n,1} \\ a_{p+1,2} & a_{p+2,2} & \dots & a_{n-1,2} & a_{n,2} \\ \dots & \dots & \dots & \dots & \dots \\ a_{p+1,p} & a_{p+2,p} & \dots & a_{n-1,p} & a_{n,p} \end{pmatrix} \right]^T,$$

де $p = \frac{n}{2} = 2^{s-1}$.

Неважко перевірити, що добуток $(A_{1,1}^{-1})^T (a_{p+1,1} \ a_{p+1,2} \ \dots \ a_{p+1,p})^T$ є не що інше, як розв'язок такої системи лінійних алгебраїчних рівнянь

$$\begin{pmatrix} a_{1,1} & a_{2,1} & \cdots & a_{p-1,1} & a_{p,1} \\ a_{1,2} & a_{2,2} & \cdots & a_{p-1,2} & a_{p,2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{1,p} & a_{2,p} & \cdots & a_{p-1,p} & a_{p,p} \end{pmatrix} \begin{pmatrix} z_{p+i,1} \\ z_{p+i,2} \\ \cdots \\ z_{p+i,p} \end{pmatrix} = \begin{pmatrix} a_{p+i,1} \\ a_{p+i,2} \\ \cdots \\ a_{p+i,p} \end{pmatrix}, \quad i = \overline{1, p}$$

Введемо тепер такі позначення

$$X^{(1)} = \begin{pmatrix} X_{p+1,1} & X_{p+2,1} & \cdots & X_{n,1} \\ X_{p+1,2} & X_{p+2,2} & \cdots & X_{n,2} \\ \cdots & \cdots & \cdots & \cdots \\ X_{p+1,p} & X_{p+2,p} & \cdots & X_{n,p} \end{pmatrix} \text{ або } X^{(1)} = (A_{1,1}^{-1})^T A_{2,1}^T \cdot$$

Тоді для визначення X_2 можна записати таку систему

$$(A_{2,2} - [X^{(1)}]^T A_{1,2}) X_2 = B_2 - [X^{(1)}]^T B_1.$$

Звідки

$$X_2 = \frac{B_2 - [X^{(1)}]^T B_1}{|A_{2,2} - [X^{(1)}]^T A_{1,2}|} = \frac{(A_{2,2} - [X^{(1)}]^T A_{1,2}) A_{1,2}^{-1} B_1 - A_{2,2} A_{1,2}^{-1} B_1 + B_2}{|A_{2,2} - [X^{(1)}]^T A_{1,2}|},$$

$$X_2 = A_{1,2}^{-1} B_1 + \frac{B_2 - A_{2,2} A_{1,2}^{-1} B_1}{|A_{2,2} - [X^{(1)}]^T A_{1,2}|}$$

де $\frac{\cdot}{\cdot}$ – позначення, введене Прингсгеймом [42] для звичайних ланцюгових дробів. Тут і далі воно буде означати множення на обернену матрицю зліва.

Невідомі X_1 можна визначити з матричного рівняння

$$A_{1,1} X_1 = B_1 - A_{1,2} X_2 \text{ за формулою } X_1 = A_{1,1}^{-1} (B_1 - A_{1,2} X_2).$$

Позначимо через $A \left\| \begin{matrix} i_1 & i_2 & \dots & i_k \\ j_1 & j_2 & \dots & j_l \end{matrix} \right\|$ блок матриці A , утворений на перетині рядків i_1, i_2, \dots, i_k і стовпчиків j_1, j_2, \dots, j_l , і розглянемо такі системи лінійних алгебраїчних рівнянь

$$\begin{pmatrix} A_{1,1}^{(k)} & A_{1,2}^{(k)} \\ A_{2,1}^{(k)} & A_{2,2}^{(k)} \end{pmatrix} \begin{pmatrix} X_1^{(k)} \\ X_2^{(k)} \end{pmatrix} = \begin{pmatrix} A_{1,3}^{(k)} \\ A_{2,3}^{(k)} \end{pmatrix}, \quad (k=1,2,\dots,s-1),$$

де

$$\left. \begin{aligned} A_{1,1}^{(k)} &= \left\| \begin{matrix} 1 & 2 & \dots & 2^{s-k-1} \\ 1 & 2 & \dots & 2^{s-k-1} \end{matrix} \right\|, & A_{1,2}^{(k)} &= \left\| \begin{matrix} 1 & 2 & \dots & 2^{s-k-1} \\ 2^{s-k-1} + 1 & 2^{s-k} + 2 & \dots & 2^{s-k} \end{matrix} \right\|, \\ A_{2,1}^{(k)} &= \left\| \begin{matrix} 2^{s-k-1} + 1 & 2^{s-k-1} + 2 & \dots & 2^{s-k} \\ 1 & 2 & \dots & 2^{s-k-1} \end{matrix} \right\|, & A_{2,2}^{(k)} &= \left\| \begin{matrix} 2^{s-k-1} + 1 & 2^{s-k-1} + 2 & \dots & 2^{s-k} \\ 2^{s-k-1} + 1 & 2^{s-k-1} + 2 & \dots & 2^{s-k} \end{matrix} \right\|. \end{aligned} \right\}.$$

$(A_{1,3}^{(k)} \quad A_{2,3}^{(k)})^T$ — стовпчик матриці $A = \left\| \begin{matrix} 1 & 2 & \dots & 2^{s-k} \\ 2^{s-k} + 1 & 2^{s-k} + 2 & \dots & 2^{s-k+1} \end{matrix} \right\|$, $k=2,4,6,\dots$.

Для $k=1,3,5,\dots$ беруться такі ж клітки, але з транспонованої матриці A^T . Тому розв'язок може бути записаний в такий спосіб

$$X_2 = A_{1,2}^{-1} B_1 + \frac{B_2 - A_{2,2} A_{1,2}^{-1} B_1}{\left[\begin{array}{c} [A_{1,1}^{(1)}]^{-1} (A_{1,3}^{(1)} - A_{1,2}^{(1)} X_2^{(1)}) \\ |A_{2,2} - \left([A_{1,2}^{(1)}]^{-1} A_{1,3}^{(1)} + \frac{A_{2,3}^{(1)} - A_{2,2}^{(1)} [A_{1,2}^{(1)}]^{-1} A_{1,3}^{(1)}}{|A_{2,2}^{(1)} - \dots - A_{2,1}^{(s-1)} [A_{1,1}^{(s-1)}]^{-1} A_{1,2}^{(s-1)}| \right) \end{array} \right]^T} A_{1,2}},$$

$$X_1 = A_{1,1}^{-1} (B_1 - A_{1,2} X_2).$$

Розглянутий алгоритм придатний для розв'язку як СЛАР із символічними елементами, так і для числових систем.

Системи лінійних алгебраїчних рівнянь із теплицевими λ -матрицями.

Вище викладене можна застосувати до розв'язування СЛАР (3.1), коли матриця A з символічними елементами є теплицевою λ -матрицею. Для

одержання обернених λ -матриць $A_{1,1}^{-1}$, $A_{1,2}^{-1}$, $[A_{1,2}^{(1)}]^{-1}$ і т.д. можуть використовуватися запропоновані в підрозділі 2.1 алгоритми для випадку, коли елементами тепліцевої λ -матриці є алгебраїчними або тригонометричними поліномами.

Висновки до розділу 3

В третьому розділі розроблено та обгрунтовано ефективні алгоритми розв'язування СЛАР з блочно-тепліцевими λ -матрицями, запропоновано моделі для реалізації алгоритмів розв'язування СЛАР з блочно-тепліцевими λ -матрицями на багатопроцесорних обчислювальних системах, розроблено алгоритми розв'язування СЛАР з використанням гіллястих ланцюгових дробів.

Отримані результати опубліковані у працях [36, 38, 42].

РОЗДІЛ 4.

ЗАСТОСУВАННЯ ТЕПЛІЦЕВИХ λ -МАТРИЦЬ В МАТЕМАТИЧНОМУ МОДЕЛЮВАННІ

До розв'язування систем лінійних алгебраїчних рівнянь (СЛАР) з тепліцевими матрицями часто зводяться відповідні обчислювальні прикладні задачі оптики, акустики, обробки зображень, автоматичного регулювання, тощо [6, 10, 27, 35, 93].

Необхідність розв'язувати СЛАР з тепліцевими λ -матрицями, елементами яких є алгебраїчні або тригонометричні поліноми, виникає при дослідженні на стійкість розв'язків СЛАР з тепліцевими матрицями, при створенні і дослідженні математичних моделей різних прикладних задач економіки, медицини, електродинаміки, гідроаеродинаміки, техніки, зокрема при дослідженні функціонування електронних схем, систем масового обслуговування з очікуванням, перехідних процесів у схемотехніці та інших прикладних задачах [6, 10, 16, 27, 30, 93].

При створенні та дослідженні математичних моделей деяких процесів і явищ в медицині в окремих випадках виникає також необхідність розв'язувати системи лінійних алгебраїчних рівнянь з тепліцевими λ -матрицями [49-61].

Мають місце такі випадки:

1) в рівнянні (1.1) елементи тепліцевої матриці $A(\lambda)$ – дійсні числа, а елементами вектора $B(\lambda)$ – відповідно алгебраїчні поліноми степеня l (зокрема лінійні функції, $l = 1$) або тригонометричні поліноми;

2) елементами тепліцевої матриці $A(\lambda)$ і вектора $B(\lambda)$ – відповідно алгебраїчні або тригонометричні поліноми.

Такі задачі виникають при якісному аналізі розв'язків СЛАР з тепліцевими матрицями, при дослідженні стійкості, ідентифікації параметрів систем, які описуються диференціальними рівняннями тощо. Ці питання

зводяться до задач оптимізації (мінімізації) деякої цільової функції [8, 78] однієї змінної λ на множині її допустимих значень F , а саме

$$\inf_{\lambda \in F} \psi(Z(\lambda)), \quad (4.1)$$

де $Z(\lambda) = \{Z_1(\lambda), Z_2(\lambda), \dots, Z_n(\lambda)\}$ – розв’язок СЛАР (1.1) як функція параметра λ .

Прикладами таких задач можуть бути наступні:

- знайти $\lambda^* \in F$, а відповідно і розв’язок СЛАР (1.1)

$Z(\lambda^*) = \{Z_1(\lambda^*), Z_2(\lambda^*), \dots, Z_n(\lambda^*)\}$, при якому забезпечується мінімум функції

$$\psi(\lambda) = \sqrt{\sum_{i=1}^n Z_i^2(\lambda)}$$

- або функції

$$\psi(\lambda) = \sqrt{\sum_{i=1}^n (Z_i(\lambda) - z_i^0)^2},$$

де $z^0 = \{z_1^0, z_2^0, \dots, z_n^0\}$ – задана точка.

4.1. Розв’язування інтегральних рівнянь в аеродинамічних дослідженнях з допомогою тепліцевих матриць.

Розв’язування аеродинамічних задач часто зводиться до проблеми знаходження частотного спектру інтегрального рівняння Фредгольма другого роду з параметром. При розв’язуванні інтегральних рівнянь виникають, як правило, щільні матриці, ядра яких часто мають специфічний вигляд. Крім того, в аеродинамічних задачах побудова матриць досить істотно залежить як від вигляду області інтегрування, так і від редукції до алгебраїчної задачі.

Якщо розглядати власні коливання консольно закріпленого крила з розподіленим вздовж нього навантаженням $q(x)$, то статичний прогин крила в точці x можна записати [4]

$$f(x) = \int_0^l G(x,s)q(s)ds, \quad (4.2)$$

де $G(x,s)$ – функція Гріна (функція впливу), яка визначає прогин в точці з абсцисою x під дією одиничної сили, прикладеної в точці з абсцисою s .

При коливаннях на крило буде діяти сила інерції $-m(s)\frac{\partial^2 f(s,t)}{\partial t^2}$ від поточного навантаження $q(x)$, тут $m(s) = \frac{q(s)}{g}$. Тому рівняння (4.2) можна записати у вигляді

$$f(x,t) = -\int_0^l G(x,s)m(s)\frac{\partial^2 f(s,t)}{\partial t^2}ds \quad (4.3)$$

Розділяючи в (4.3) змінні за методом Фур'є, приходимо до інтегрального рівняння Фредгольма другого роду з параметром λ , невідомою функцією $f(x)$ і ядром $K(x,s) = G(x,s)m(x,s)$

$$f(x) = \lambda \int_0^l K(x,s)f(s)ds \quad (4.4)$$

Ядро $K(x,s)$ при $m(x) = const$ буде симетричним, оскільки є симетричною функція Гріна $G(x,s)$. Як відомо, при деякому значенні λ існує розв'язок інтегрального рівняння (4.4), відмінний від нуля, тобто $f(x) \neq 0$; отже, λ є характеристичним числом або власним значенням ядра інтегрального рівняння (4.4). Кожний розв'язок $f(x)$ буде фундаментальною або власною функцією, яка відповідає власному значенню λ .

Розглядаючи рівняння (4.4) в ряді точок $i = x_i \in (0,l)$, $i = \overline{1,N}$, його можна звести до системи алгебраїчних рівнянь

$$\sum_{k=1}^N \sum_{j=1}^N K(\lambda, x_k, x_j) f_j(x_j) = f_k(x_k), \quad (4.5)$$

де $K(\lambda, x_k, x_j)$ – матриця з елементами

$$\left. \begin{aligned} K(\lambda, x_1, x_1) &= \lambda G(x_1, x_1) m(x_1) \\ K(\lambda, x_1, x_2) &= \lambda G(x_1, x_2) m(x_2) \\ K(\lambda, x_1, x_n) &= \lambda G(x_1, x_n) m(x_n) \\ K(\lambda, x_n, x_1) &= \lambda G(x_n, x_1) m(x_1) \\ K(\lambda, x_n, x_2) &= \lambda G(x_n, x_2) m(x_2) \\ K(\lambda, x_n, x_n) &= \lambda G(x_n, x_n) m(x_n) \end{aligned} \right\} \quad (4.6)$$

Якщо розглядати крило сталого перерізу при $m(x) = const$, ядро $K(x, s)$ інтегрального рівняння (4.4) буде симетричним, а матриця коефіцієнтів у системі алгебраїчних рівнянь (4.5) – тепліцевою λ -матрицею.

Для знаходження власних значень λ інтегрального рівняння (4.4) використаємо властивості рядів Фредгольма. На основі теорії Фредгольма функцію $D(x)$, що визначає величини власних значень λ , можна розкласти в ряд за степенями λ [3]

$$D(\lambda) = 1 + \sum_{n=1}^{\infty} (-1)^n \frac{a_n(\lambda)}{n!} \quad (4.7)$$

Ряд (4.7) одержано шляхом переходу до границь у визначнику, який складається із коефіцієнтів при невідомих функціях систем лінійних алгебраїчних рівнянь (4.5), що апроксимують інтегральне рівняння (4.4).

Коефіцієнти $a_n(\lambda)$ визначаються за допомогою наступних детермінантних формул

$$a_n(\lambda) = \int_0^l \int_0^l \dots \int_0^l \begin{vmatrix} K(\lambda, x_1, x_1) & K(\lambda, x_1, x_2) & \dots & K(\lambda, x_1, x_n) \\ \dots & \dots & \dots & \dots \\ K(\lambda, x_n, x_1) & K(\lambda, x_n, x_2) & \dots & K(\lambda, x_n, x_n) \end{vmatrix} dx_1 \dots dx_n, \quad (4.8)$$

де елементи визначника в (4.8) обчислюються за формулами (4.6).

З врахуванням (4.8) робимо висновок, що ряд (4.7) є степеневим рядом, що абсолютно збігається при всіх значення параметра λ , а самі власні значення λ шукаються як корені характеристичного рівняння $D(\lambda) = 0$.

Розглянемо крило сталого перерізу у випадку, коли розподілена маса крила $m(x) = \text{const}$. Тоді ядро інтегрального рівняння (4.2) буде симетричним, а матриці у формулах (4.4) будуть теплицевими λ -матрицями.

4.1.1. Визначення частот коливань.

Після обчислення коефіцієнтів $a_n(\lambda)$ можна визначити частоти коливань. Значення частот знаходяться як корені рівняння $D(\lambda) = 0$, яке є рівнянням частот коливань з врахуванням того факту, що власне значення $\lambda = p^2$, де p – частота коливань. Прирівняємо до нуля n членів ряду (4.3) і побудуємо алгебраїчне рівняння n -го порядку для визначення n перших частот коливань. Складемо, наприклад, рівняння для відшукування першої частоти в першому наближенні.

З рівняння (4.3) маємо

$$1 - a_1 \lambda = 0, \quad (4.9)$$

звідки

$$p_1 = \sqrt{\lambda} = \sqrt{\frac{1}{a_1}} \quad (4.10)$$

Формула (4.10) дає нижню оцінку частоти.

При відшуванні першої частоти в другому наближенні та другої частоти в першому наближенні рівняння частот (4.3) будуть другого порядку відносно λ

$$1 - a_1 \lambda + \frac{a_2}{2} \lambda^2 = 0. \quad (4.11)$$

З рівняння (4.11) маємо $\lambda_{1,2} = \frac{a_1 \pm \sqrt{a_1^2 - 2a_2}}{a_2}$ і відповідно

$$p_1 = \sqrt{\frac{a_1 - \sqrt{a_1^2 - 2a_2}}{a_2}}, \quad (4.12)$$

$$p_2 = \sqrt{\frac{a_1 + \sqrt{a_1^2 - 2a_2}}{a_2}}, \quad (4.13)$$

де p_1 – значення першої частоти в другому наближенні, а p_2 – значення другої частоти в першому наближенні.

При відшуванні першої частоти в третьому наближенні, другої частоти в другому наближенні та третьої частоти в першому наближенні рівняння

$$\text{частоти набуде вигляду } 1 - a_1 \lambda + \frac{a_2}{2} \lambda^2 - \frac{a_3}{6} \lambda^3 = 0.$$

Аналогічно для визначення кожної з необхідних n частот, $n = 1, 2, \dots$ будемо будувати окреме рівняння.

Зауважимо, що коефіцієнти алгебраїчного рівняння (4.3) визначаються через тепліцеві матриці.

4.1.2. Змінні коливання консольного крила сталого перерізу.

У цьому випадку при $m(x) = \text{const}$ функцію впливу (функцію Гріна) запишемо у вигляді

$$G(\bar{x}, \bar{s}) = \frac{l^3}{EI} \int_0^{\bar{s}} (\bar{s} - \bar{u})(\bar{x} - \bar{u}) d\bar{u} \quad \text{для } \bar{x} \geq \bar{s}, \quad (4.14)$$

де $\bar{x} = \frac{x}{l}$ – відносна координата перерізу від точки його закріплення, EI – змінна жорсткість крила, l – довжина крила. Проінтегруємо вираз (4.14) і

отримаємо $G(\bar{x}, \bar{s}) = \frac{l^3}{6EI} \bar{s}^{-2} (3\bar{x} - \bar{s})$ для $\bar{x} \geq \bar{s}$.

З врахуванням властивості симетрії функції Гріна

$$G(\bar{x}, \bar{s}) = \frac{l^3}{6EI} \bar{x}^{-2} (3\bar{s} - \bar{x}) \text{ для } \bar{x} \leq \bar{s}.$$

Після цього знаходимо коефіцієнти a_1 і a_2 .

Розглядаючи крило сталого перерізу на основі (4.4) з врахуванням (4.5) запишемо перший детермінантний коефіцієнт

$$a_1 = l \int_0^1 K(\bar{x}, \bar{x}) d\bar{x} = ml \int_0^1 G(\bar{x}, \bar{x}) d\bar{x} = \frac{ml^4}{3EI} \int_0^1 \bar{x}^{-3} d\bar{x}.$$

Після інтегрування остаточно отримаємо

$$a_1 = \frac{1}{12} \frac{ml^4}{EI}. \quad (4.15)$$

Використовуючи (4.4) і враховуючи (4.5), маємо

$$\begin{aligned} a_2 &= l^2 \int_0^1 \int_0^1 \begin{vmatrix} K(\bar{x}, \bar{x}) & K(\bar{x}, \bar{s}) \\ K(\bar{s}, \bar{x}) & K(\bar{s}, \bar{s}) \end{vmatrix} d\bar{x} d\bar{s} = \\ &= m^2 l^2 \int_0^1 \int_0^1 G(\bar{x}, \bar{x}) G(\bar{s}, \bar{s}) d\bar{x} d\bar{s} - m^2 l^2 \int_0^1 \int_0^1 G(\bar{x}, \bar{s}) G(\bar{x}, \bar{s}) d\bar{x} d\bar{s} = \\ &= m^2 l^2 \left[\int_0^1 G(\bar{x}, \bar{x}) d\bar{x} \right]^2 - m^2 l^2 \int_0^1 \int_0^1 [G(\bar{x}, \bar{s})]^2 d\bar{x} d\bar{s} = \\ &= \frac{m^2 l^8}{9(EI)^2} \left[\int_0^1 \bar{x}^{-2} d\bar{x} \right]^2 - \frac{m^2 l^8}{36(EI)^2} \int_0^1 \left\{ \int_0^{\bar{x}} \bar{s}^{-4} (3\bar{x} - \bar{s})^2 d\bar{s} + \int_{\bar{s}}^1 \bar{x}^{-2} (3\bar{s} - \bar{x})^2 d\bar{s} \right\} d\bar{x}. \end{aligned}$$

Після інтегрування отримаємо

$$a_2 = \frac{m^2 l^8}{2520(EI)^2}. \quad (4.16)$$

Для визначення частоти змінних коливань крила в першому наближенні підставимо (4.15) в (4.10) і отримаємо

$$p_1 = \frac{3,46}{l^2} \sqrt{\frac{EI}{m}}. \quad (4.17)$$

Для визначення першої частоти в другому наближенні та другої частоти в першому наближенні підставимо (4.15) і (4.16) у вирази (4.12) та (4.13)

$$p_1 = \frac{3,516}{l^2} \sqrt{\frac{EI}{m}} \quad p_2 = \frac{20,19}{l^2} \sqrt{\frac{EI}{m}}. \quad (4.18)$$

4.1.3. Нестационарна задача.

Розглянемо інтегральне рівняння (4.3). Нехай в точках $x = x_i$, $i = \overline{1, n}$, $n < N$ проходить зменшення з часом t величини маси крила, наприклад, внаслідок витрати палива під час польоту (паливні баки знаходяться на крилі), тобто нехай $m_i(x_i, t) = m_i(x_i) - \Delta m(x_i) = m_i(x_i) - Ct = Qm_i(x_i)$, де $m_i(x_i) = m_i$ – значення маси в точці $x = x_i$ в момент часу $t = 0$, $\Delta m(x_i)$ – величина, на яку зменшується маса крила в точці x_i , Q – відомий коефіцієнт пропорційності.

Тоді рівняння коливань (4.3) можна записати таким чином

$$f(x,t) = -\int_0^l G(x,s)m(s)\frac{\partial^2 f(s,t)}{\partial t^2} ds - \sum_{i=1}^n G(x,s_i)Qm_i\frac{\partial^2 f(s_i,t)}{\partial t^2} ds. \quad (4.19)$$

Розділяючи змінні в (4.19), приходимо до інтегрального рівняння коливань крила з урахуванням зміни з часом t маси крила в окремих його точках

$$f(x) = \lambda \left[\int_0^l G(x,s)m(s)f(s)ds + \sum_{i=1}^n G(x,s_i)QG_i(x,s_i)m_i f(s_i) \right]. \quad (4.20)$$

Розглядаючи крило сталого перерізу $m(x) = const$, рівняння (4.20) буде інтегральним рівнянням Фредгольма другого роду з симетричним ядром, якщо інтеграл розглядати в понятті Стілььєса. Запишемо його у вигляді

$$(S) \quad \int_0^l K(\lambda, x, s)f(s)dm(s), \quad (4.21)$$

де позначка (S) означає, що інтеграл розглядається в понятті Стілььєса.

Розглянемо рівняння (4.21) в ряді точок $x_k \in (0, l)$, $k = \overline{1, N}$, зведемо його до такої системи лінійних алгебраїчних рівнянь

$$\sum_{k=1}^n \sum_{j=1}^N K(\lambda, x_k, x_j) f_j(x_j) = f_k(x_k). \quad (4.22)$$

В системі алгебраїчних рівнянь (4.22) елементи матриці K запишуться

$$K(\lambda, x_k, x_j) = \lambda \left[G(x_k, x_j) + Q \sum_{i=1}^n G(x_k, x_j) \right] m_j(x_j). \quad (4.23)$$

Таким чином, розв'язок нестационарної задачі (4.19) зведено до системи алгебраїчних рівнянь (4.22), де матриця $\|K(\lambda, x_k, x_j)\|$ є тепліцевою λ -матрицею.

4.2. Про збіжність обчислених власних значень.

Власні значення λ інтегрального рівняння (4.4) знаходяться як корені характеристичного рівняння (4.7). Виникає питання про збіжність величин знайдених наближених власних значень для точних величин власних значень. Вважається, що похибка в самому процесі обчислень відсутня і виникає лише внаслідок застосування інтегрального рівняння (4.2), яке, в принципі, описує коливальний рух крила наближено.

Покажемо, що коли порядок матриць $\|K(\lambda, x_i, x_j)\|$, $i, j = \overline{1, n}$ в (4.5) буде необмежено зростати, тобто кількість точок x_n розбиття інтервалу інтегрування $(0, l)$ буде необмежено збільшуватись ($n \rightarrow \infty$), то наближені величини власних значень $\lambda(n)$ будуть прямувати до точних величин власних значень λ .

Власні форми коливань крила як одновимірного об'єкта представляються інтегральним рівнянням

$$f(x) = \lambda \int_0^l G(x, s) m(s) f(s) ds. \quad (4.24)$$

Підставимо в це рівняння вираз

$$f^{(n)}(x) = f(x) \sqrt{m(x)}, \quad (4.25)$$

приходимо до інтегрального рівняння з симетричним ядром вигляду

$$f^{(n)}(x) = \lambda^{(n)} \int_0^l K^{(n)}(x, s) f^{(n)}(s) ds, \quad (4.26)$$

де $K^{(n)}(x, s) = G(x, s) \sqrt{m(x)m(s)}$.

Індекс n у виразах $f^{(n)}(x)$, $\lambda^{(n)}$, $K^{(n)}(x, s)$ показує порядок матриць $\|K^{(n)}(\lambda, x_i, x_j)\|$, тобто кількість точок $x_i \in (0, l)$, в яких беруться значення елементів ядра інтегрального рівняння (4.26).

Інтегральне рівняння точних власних форм коливань цього ж крила представимо у вигляді

$$f_T(x) = \lambda \int_0^l G_T(x, s) m(s) f(s) ds. \quad (4.27)$$

Підставимо $f_T(x) = f_T(x) \sqrt{m(x)}$, рівняння (4.27) зведемо до інтегрального рівняння з симетричним ядром

$$f(x) = \lambda \int_0^l K(x, s) f(s) ds \quad (4.28)$$

Рівняння (4.26) і (4.27) є однорідними рівняннями Фредгольма другого роду з симетричними ядрами, для яких справедлива теорема Гільберта-Шмідта [12, 24].

Припускається, що збільшуючи число взятих на крилі точок x_i , можна отримати таку нерівність

$$|K(x, s) - K^{(n)}(x, s)| \leq \varepsilon, \quad (4.29)$$

де $\varepsilon > 0$ – достатньо мале значення.

Нехай $\lambda_i^{(n)}$ – i -е власне значення рівняння (4.26). Покажемо, що $\lambda_i^{(n)} \rightarrow \lambda_i$ при $n \rightarrow \infty$.

Запишемо рівняння (4.27) і (4.28) в операторному вигляді

$$f^{(n)}(x) - \lambda^{(n)} B f^{(n)}(x) = 0, \quad (4.30)$$

де $B f^{(n)}(x) = \int_0^l K^{(n)}(x, s) f^{(n)}(s) ds$, а також

$$f(x) - \lambda A f(x) = 0, \quad (4.31)$$

де $A f(x) = \int_0^l K(x, s) f(s) ds$.

Очевидно, що нерівність (4.29) еквівалентна двом нерівностям

$$|K(x, s) - K^{(n)}(x, s)| \leq \varepsilon, \quad |K^{(n)}(x, s) - K(x, s)| \leq \varepsilon.$$

Ці нерівності мають рівноцінні значення при доведенні збіжності власних значень.

Представимо першу з цих нерівностей у вигляді

$$K(x, s) - K^{(n)}(x, s) = \zeta(x, s), \quad (4.32)$$

де $\zeta(x, s) \leq \varepsilon$.

Оператори A і B , що входять в рівняння (4.30) і (4.31) як оператори Фредгольма, є достатньо неперервні і самоспряжені, тобто симетричні.

Для того, щоб довести збіжність власних значень, використаємо теорему Вейля [68].

Нехай B і $A = B + C$ – достатньо неперервні самоспряжені оператори в просторі Гільберта. Прономеруємо додатні власні значення оператора A таким чином: $0 < \lambda_1^{(A)} \leq \lambda_2^{(A)} \leq \lambda_3^{(A)} \leq \dots$. Аналогічно прономеруємо власні значення оператора B $0 < \lambda_1^{(B)} \leq \lambda_2^{(B)} \leq \lambda_3^{(B)} \leq \dots$. Тоді згідно теореми Вейля має місце нерівність

$$\left| \frac{1}{\lambda_i^{(B)}} - \frac{1}{\lambda_i^{(A)}} \right| \leq \|C\|, \quad (4.33)$$

де $\|C\|$ – норма оператора C , яка є достатньо малою. Із співвідношення (4.33) маємо

$$|\lambda_i^{(A)} - \lambda_i^{(B)}| \leq |\lambda_i^{(A)}| |\lambda_i^{(B)}| \|C\|. \quad (4.34)$$

Використавши нерівність $|\lambda_i^{(A)}| \leq |\lambda_i^{(A)} - \lambda_i^{(B)}| + |\lambda_i^{(B)}|$, з (4.34) отримаємо

$$|\lambda_i^{(A)} - \lambda_i^{(B)}| \leq \left\{ |\lambda_i^{(B)}| + |\lambda_i^{(A)} - \lambda_i^{(B)}| \right\} \|C\| |\lambda_i^{(B)}|, \text{ звідки маємо таку оцінку для власних}$$

значень

$$|\lambda_i^{(A)} - \lambda_i^{(B)}| \leq \frac{|\lambda_i^{(B)}|^2 \|C\|}{1 - |\lambda_i^{(B)}| \|C\|}. \quad (4.35)$$

Ця нерівність вірна при умові, що $|\lambda_i^{(B)}| \|C\| < 1$.

Підставимо (4.32) в рівняння (4.31) і одержимо

$$f(x) - \lambda A f(x) = f(x) - \lambda \int_0^l [K^{(n)}(x, s) + \zeta(x, s)] f(s) ds,$$

звідки випливає рівність таких операторів: $A = N + C$, де

$$Nf(x) = \int_0^l K^{(n)}(x, s) f(s) ds \text{ і } Cf(x) = \int_0^l \zeta(x, s) f(s) ds.$$

Отже, оператор A задовольняє теоремі Вейля.

Використовуючи нерівність (4.35), отримаємо

$$|\lambda_i^{(A)} - \lambda_i^{(B)}| \leq \frac{\|C\| |\lambda_i^{(N)}|^2}{1 - |\lambda_i^{(N)}| \|C\|}, \quad (4.36)$$

де $\lambda_i^{(N)}$ – власні значення оператора N , а $\lambda_i^{(A)}$ – точні власні значення.

Оскільки оператори N і B мають одне і те ж ядро, то власні значення цих операторів будуть співпадати [12, 64].

В результаті отримаємо

$$|\lambda_i^{(A)} - \lambda_i^{(B)}| \leq \frac{\|C\| |\lambda_j^{(B)}|^2}{1 - |\lambda_j^{(B)}| \|C\|}.$$

Розглядаючи цю нерівність при $n \rightarrow \infty$, приходимо до висновку про збіжність знайдених наближених власних значень до величин точних власних значень, тобто $\lambda_i^{(B)} \rightarrow \lambda_i^{(A)}$ при $n \rightarrow \infty$, оскільки $\|C\| \rightarrow 0$.

4.3. Гідродинамічні задачі

При розв'язуванні гідродинамічних задач в рівняннях гідродинаміки і крайових умовах також зустрічаються матриці типу теплицевих. Замкнута система рівнянь руху в'язкості ньютонівської рідини, що не стискається,

складається з рівняння нерозривності $\frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} + \frac{\partial V_z}{\partial z} = 0$ та трьох рівнянь

Нав'є-Стокса [47]

$$\frac{\partial V_x}{\partial t} = V_x \frac{\partial V_x}{\partial X} + V_y \frac{\partial V_x}{\partial Y} + V_z \frac{\partial V_x}{\partial Z} = -\frac{1}{\rho} \frac{\partial p}{\partial X} + \nu \left(\frac{\partial^2 V_x}{\partial X^2} + \frac{\partial^2 V_x}{\partial Y^2} + \frac{\partial^2 V_x}{\partial Z^2} \right) + g_x \quad (4.37)$$

$$\frac{\partial V_y}{\partial t} = V_x \frac{\partial V_y}{\partial X} + V_y \frac{\partial V_y}{\partial Y} + V_z \frac{\partial V_y}{\partial Z} = \frac{1}{\rho} \frac{\partial p}{\partial Y} + \nu \left(\frac{\partial^2 V_y}{\partial X^2} + \frac{\partial^2 V_y}{\partial Y^2} + \frac{\partial^2 V_y}{\partial Z^2} \right) + g_y \quad (4.38)$$

$$\frac{\partial V_z}{\partial t} = V_x \frac{\partial V_z}{\partial X} + V_y \frac{\partial V_z}{\partial Y} + V_z \frac{\partial V_z}{\partial Z} = \frac{1}{\rho} \frac{\partial p}{\partial Z} + \nu \left(\frac{\partial^2 V_z}{\partial X^2} + \frac{\partial^2 V_z}{\partial Y^2} + \frac{\partial^2 V_z}{\partial Z^2} \right) + g_z.$$

Рівняння (4.37), (4.38) записані в прямокутній декартовій системі координат; X, Y, Z – відповідні координати розглядуваної точки фізичного простору, t – час, g_x, g_y, g_z – компоненти вектора густини масової сили (наприклад, сили тяжіння), ν – кінематична в'язкість речовини. Шуканими величинами є три компоненти швидкості V_x, V_y, V_z і тиск p .

Розглянемо випадки, характерні для градієнтних потоків з неоднорідною структурою потоку. Довільне стаціонарне поле швидкостей $\bar{V}(\bar{R})$ в середовищі, що не стискається, в околі точки $\bar{R} = 0$, яку прийmemo за початок відліку, наближено може бути представлене у вигляді двох членів розкладу в ряд Тейлора [46]

$$V_k(\bar{R}) = V_k(\bar{0}) + G_{km} X_m, \quad G_{km} \equiv \left(\frac{\partial V_k}{\partial X_m} \right)_{R=0}, \quad G_{11} + G_{22} + G_{33} = 0, \quad (4.39)$$

де V_k і G_{km} – компоненти швидкості рідини і тензора зсуву в декартовій системі координат X_1, X_2, X_3 .

Частковий випадок при $G_{km} = 0$ відповідає випадку однорідного поступального потоку. При $V_k(\bar{0}) = 0$ формула (4.39) описує поле швидкостей в довільному лінійному зсувному потоці. Будь-який тензор $\|G_{km}\|$ може бути представлений у вигляді суми симетричного і антисиметричного тензорів

$$\|G_{km}\| = \|E_{km}\| + \|\Omega_{km}\| \quad E_{km} = E_{mk} = \frac{1}{2}(G_{km} + G_{mk}) \quad \Omega_{km} = -\Omega_{mk} = \frac{1}{2}(G_{km} - G_{mk}).$$

Серед лінійних зсувних потоків, що найчастіше розглядаються, можна виділити такі типи:

1. Простий зсув, або потік Куетта.

Розглядаються такі рівняння гідродинаміки

$$V_x = GY, \quad V_y = 0, \quad V_z = 0,$$

де матриці тензора зсуву записують у вигляді

$$\|G_{km}\| = \begin{vmatrix} 0 & G & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}, \quad \|E_{km}\| = \begin{vmatrix} 0 & \frac{1}{2}G & 0 \\ \frac{1}{2}G & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}, \quad \|\Omega_{km}\| = \begin{vmatrix} 0 & \frac{1}{2}G & 0 \\ -\frac{1}{2}G & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}.$$

Величина G в даному випадку є градієнтом швидкості потоку або швидкістю деформації. Потік Куетта може бути реалізований між двома паралельними площинами, що рухаються.

2. Другий випадок – плоский безвихровий рух. Розглядаються такі гідродинамічні рівняння

$$V_x = \frac{1}{2}GY, \quad V_y = \frac{1}{2}GX, \quad V_z = 0.$$

Матриці тензора зсуву запишемо

$$\|G_{km}\| = \begin{vmatrix} 0 & \frac{1}{2}G & 0 \\ \frac{1}{2}G & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}, \quad \|E_{km}\| = \begin{vmatrix} 0 & \frac{1}{2}G & 0 \\ \frac{1}{2}G & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}, \quad \|\Omega_{km}\| = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}.$$

Цей потік має таку ж деформаційну складову руху, як і простий зсув, але не має поворотної складової.

Отже, при дослідженні цих потоків, а також при плоскому деформаційному зсуві, плоскому обертанні твердого тіла, осесиметричному зсуві та інших задачах виникає необхідність оперувати матрицями типу Тепліца.

4.4 Дослідження стійкості розв'язків математичної моделі імунного захисту.

Розглядається модель запального процесу інфекційної природи [60]. В моделі враховуються наступні визначальні для перебігу процесу чинники:

- I. популяція антигенів V , що розмножуються в організмі;
- II. популяція антитілотвірних клітин (плазмоклітин) C ;
- III. кількість антитіл (імуноглобулінів) F в організмі;
- IV. ступінь пошкодження органа m .

Рівняння, що визначають динаміку процесу, мають вигляд

$$\begin{aligned} \frac{dV}{dt} &= (\beta - \gamma F)V, \\ \frac{dC}{dt} &= \xi(m)\alpha V(t-\tau)F(t-\tau) - \mu_c(C - C_0), \\ \frac{dF}{dt} &= \rho C - (\mu_f + \eta \gamma V)F, \\ \frac{dm}{dt} &= \sigma V - \mu_m m \end{aligned} \tag{4.40}$$

з початковими умовами при $t \in [-\tau, 0]$: $V(t) = V_0, F(t) = F_0, C(t) = C_0, m(t) = 0$.

Тут β – коефіцієнт розмноження антигена; γ – коефіцієнт, що визначає ймовірність нейтралізації антигена антитілом; α – коефіцієнт, що зумовлює ймовірність зустрічі антиген-антитіло; μ_c – коефіцієнт, обернений до часу життя плазмоклітин; ρ – швидкість виробництва антитіл однією плазмоклітиною; μ_f – коефіцієнт, обернено пропорційний до часу розпаду антитіл; η – число антитіл, необхідних для нейтралізації одного антигена; σ – коефіцієнт, що визначає швидкість загибелі клітин за рахунок пошкоджуючої дії антигена; μ_m – коефіцієнт, що враховує швидкість відновлення пошкодженого органу; τ – фаза запізнення (час, за який здійснюється формування каскаду плазмоклітин); $\xi(m)$ – неперервна незростаюча функція ($0 \leq \xi(m) \leq 1$), що характеризує порушення нормального функціонування імунної системи через значне пошкодження органу-мішені.

Перераховані параметри додатні та є специфічними як для типу антигена, так і для органу і конкретного організму.

Представлена вище система диференціальних рівнянь має два стани рівноваги [54]. Один з них – тривіальний, інший позначимо (V^*, F^*, C^*, m^*) . Здійснивши лінеаризацію системи звичайних диференціальних рівнянь (ЗДР) в околі точки (V^*, F^*, C^*, m^*) , отримаємо лінійну систему ЗДР з постійними коефіцієнтами

$$\frac{dx_1}{dt} = \beta x_1 - \gamma F^* x_1 - \gamma V^* x_3$$

$$\frac{dx_2}{dt} = \xi(m^*) \alpha F^* x_1(t - \tau) + \xi(m^*) \alpha V^* x_3(t - \tau) - \mu_c x_2 + \alpha V^* F^* \frac{d\xi(m^*)}{dm} x_4$$

$$\frac{dx_3}{dt} = \rho x_2 - \mu_f x_3 - \eta \gamma V^* x_3 - \eta \gamma F^* x_1$$

$$\frac{dx_4}{dt} = \sigma x_1 - \mu_m x_4$$

Характеристичний поліном отриманої лінійної системи ЗДР є квазіполіномом (експоненціальним поліномом) четвертого степеня

$$\lambda^4 + a_1 \lambda^3 + a_2 \lambda^2 + a_3 \lambda + a_4 + b_1 \lambda^2 e^{-\lambda \tau} + b_2 \lambda e^{-\lambda \tau} + b_3 e^{-\lambda \tau} = 0, \quad (4.41)$$

де

$$\begin{aligned} a_1 &= \gamma F^* + \eta V^* + \mu_f - \beta + \mu_m + \mu_c, \\ a_2 &= -\beta \mu_f + \mu_c \mu_m + \mu_c \mu_f - \beta \mu_m - \beta \mu_c + \gamma F^* \mu_m + \eta \gamma V^* \mu_m + \mu_c \eta \gamma V^* + \\ &+ \mu_f \mu_m - \beta \eta \gamma V^* + \gamma F^* \mu_f + \gamma F^* \mu_c, \\ a_3 &= \mu_c \mu_f \mu_m - \beta \mu_c \mu_f - \beta \mu_c \eta \gamma V^* - \beta \mu_f \mu_m - \beta \eta \gamma V^* \mu_m - \beta \mu_c \mu_m + \\ &+ \mu_c \eta \gamma V^* \mu_m + \gamma F^* \mu_c \mu_m + \gamma F^* \mu_c \mu_f + \gamma F^* \mu_f \mu_m, \\ a_4 &= \sigma \rho \gamma (V^*)^2 \frac{d\xi(m^*)}{dm} \alpha F^* - \beta \mu_c \mu_f \mu_m - \beta \mu_c \eta \gamma V^* \mu_m + \gamma F^* \mu_c \mu_f \mu_m, \\ b_1 &= -\rho \xi(m^*) \alpha V^*, \\ b_2 &= \beta \rho \xi(m^*) \alpha V^* - \rho \mu_m \xi(m^*) \alpha V^*, \\ b_3 &= \beta \rho \mu_m \xi(m^*) \alpha V^*. \end{aligned}$$

Дослідження стійкості розв'язків моделі імунного захисту.

При вивченні розміщення коренів рівняння на основі експоненціального полінома (4.41) буде використано наступний результат, доведений в роботі [123] з використанням теореми Руше [115].

ЛЕМА 4.1. Для експоненціального полінома

$$\begin{aligned} P(\lambda, e^{-\lambda \tau_1}, \dots, e^{-\lambda \tau_m}) &= \lambda^n + p_1^{(0)} \lambda^{n-1} + \dots + p_{n-1}^{(0)} \lambda + p_n^{(0)} \\ &+ [p_1^{(1)} \lambda^{n-1} + \dots + p_{n-1}^{(1)} \lambda + p_n^{(1)}] e^{-\lambda \tau_1} \\ &+ \dots + [p_1^{(m-1)} \lambda^{n-1} + \dots + p_{n-1}^{(m-1)} \lambda + p_n^{(m-1)}] e^{-\lambda \tau_m}, \end{aligned}$$

де $\tau_i \geq 0, i=1,2,\dots,m$ і $p_j^{(i)}, i=0,1,\dots,m-1, j=1,2,\dots,n$ є константами, при зміні $(\tau_1, \tau_2, \dots, \tau_m)$ сума порядків нулів $P(\lambda, e^{-\lambda \tau_1}, \dots, e^{-\lambda \tau_m})$ у відкритій правій напівплощині може змінюватися лише коли нуль з'являється на уявній осі, або перетинає уявну вісь.

Зрозуміло, що $i\omega (\omega > 0)$ буде коренем рівняння (4.41) тоді і тільки тоді, КОЛИ

$$\begin{aligned} \omega^4 - ia_1 \omega^3 - a_2 \omega^2 + ia_3 \omega + a_4 - b_1 \omega^2 (\cos \omega \tau - i \sin \omega \tau) + \\ + ib_2 \omega (\cos \omega \tau - i \sin \omega \tau) + b_3 (\cos \omega \tau - i \sin \omega \tau) = 0. \end{aligned}$$

Розділяючи дійсну і уявну частини, маємо

$$\begin{aligned} w^4 - a_2 w^2 + a_4 &= b_1 w^2 \cos w\tau - b_2 w \sin w\tau - b_3 \cos w\tau, \\ a_1 w^3 + a_3 w &= -b_1 w^2 \sin w\tau - b_2 w \cos w\tau + b_3 \sin w\tau. \end{aligned} \quad (4.42)$$

Додавши квадрати обох рівнянь (4.42), отримаємо

$$\begin{aligned} w^8 + (a_1^2 - 2a_2)w^6 + (a_2^2 + 2a_4 + 2a_1a_3)w^4 + (a_3^2 - 2a_2a_4)w^2 + a_4^2 &= \\ = b_1^2 w^4 + b_2^2 w^2 + b_3^2 - 2b_1b_3 w^2, \text{ тобто} & \\ w^8 + (a_1^2 - 2a_2)w^6 + (a_2^2 + 2a_4 + 2a_1a_3 - b_1^2)w^4 + & \\ + (a_3^2 - 2a_2a_4 - b_2^2 + 2b_1b_3)w^2 + (a_4^2 - b_3^2) &= 0. \end{aligned} \quad (4.43)$$

Покладемо $z = w^2$ і введемо позначення

$$\begin{aligned} p &= a_1^2 - 2a_2, \quad q = a_2^2 + 2a_4 + 2a_1a_3 - b_1^2, \\ r &= a_3^2 - 2a_2a_4 - b_2^2 + 2b_1b_3, \quad s = a_4^2 - b_3^2. \end{aligned}$$

Тоді рівняння (4.43) набуде вигляду

$$z^4 + pz^3 + qz^2 + rz + s = 0. \quad (4.44)$$

ТВЕРДЖЕННЯ 4.1. *Якщо $s < 0$, то рівняння (4.44) має принаймні один додатний розв'язок.*

Доведення. Позначимо

$$h(z) = z^4 + pz^3 + qz^2 + rz + s. \quad (4.45)$$

Зрозуміло, що $h(0) = s < 0$, а $\lim_{z \rightarrow \infty} h(z) = \infty$. Звідси випливає, що існує $z_0 \in (0, \infty)$, при якому $h(z_0) = 0$, що й потрібно було показати.

ТВЕРДЖЕННЯ 4.2 *Якщо $s \geq 0$ і рівняння (4.44) має додатні дійсні корені, то*

$$\Delta = \frac{\xi^2}{4} + \frac{\eta^3}{27} \geq 0, \quad (4.46)$$

$$\text{де } \xi = \frac{p^3}{24} - \frac{pq}{8} + \frac{r}{4}, \quad \eta = -\frac{3p^2}{16} + \frac{q}{2}.$$

Доведення. Із (4.45) маємо $\frac{dh(z)}{dz} = 4z^3 + 3pz^2 + 2qz + r$.

Покладемо

$$4z^3 + 3pz^2 + 2qz + r = 0. \quad (4.47)$$

Тоді три корені рівняння (4.47) (із врахуванням кратності) можуть бути знайдені за формулою Кардано [54]

$$z_{1,2,3} = \sqrt[3]{-\frac{\xi}{2} + \sqrt{\frac{\xi^2}{4} + \frac{\eta^3}{27}}} + \sqrt[3]{-\frac{\xi}{2} - \sqrt{\frac{\xi^2}{4} + \frac{\eta^3}{27}}}, \quad (4.48)$$

причому, вибираючи послідовно по одному з трьох значень кубічного кореня

$$\alpha = \sqrt[3]{-\frac{\xi}{2} + \sqrt{\frac{\xi^2}{4} + \frac{\eta^3}{27}}}, \quad \text{потрібно з трьох можливих значень кореня}$$

$$\beta = \sqrt[3]{-\frac{\xi}{2} - \sqrt{\frac{\xi^2}{4} + \frac{\eta^3}{27}}} \quad \text{вибрати те, для якого } \alpha\beta = -\frac{\eta}{3}.$$

Якщо $\Delta \leq 0$, то (4.47) не має дійсних коренів. Отже, функція $h(z)$ є зростаючою. З умови $h(0) = s \geq 0$ випливає, що рівняння (4.44) не має додатних дійсних коренів. Отримали суперечність що й доводить справедливість твердження.

У випадку, коли $\Delta \geq 0$, серед коренів z_1, z_2, z_3 згідно формул (4.48) існує принаймі один, який є локальним мінімумом $h(z)$.

$$z^* = \arg \min_{i=1,3} h(z_i)$$

Позначимо

ТВЕРДЖЕННЯ 4.3. *Якщо $s \geq 0$, рівняння (4.44) має додатні корені тоді і тільки тоді, коли $z^* > 0$ і $h(z^*) \leq 0$.*

Доведення. Достатність твердження є очевидною. При доведенні необхідності скористаємося доведенням від супротивного. Припустимо, що рівняння (4.44) має додатні корені, але при цьому або $z^* \leq 0$ або $z^* > 0$ і $h(z^*) \geq 0$.

Якщо $z^* \leq 0$, то оскільки $h(z)$ є зростаючою при $z \geq z^*$ і $h(0) = s \geq 0$, звідси випливає, що $h(z)$ не має додатних дійсних коренів. Якщо

$z^* \triangleright 0$ і $h(z^*) \geq 0$, то $h(z)$ не має додатних дійсних коренів в силу означення z^* .

Отже, в загальному випадку маємо

ЛЕМА 4.2. *Якщо $s < 0$, то рівняння (4.44) має принаймні один додатний корінь.*

Якщо $s \geq 0$, то рівняння (4.44) має додатні корені тоді і тільки тоді, коли $z^ > 0$ і $h(z^*) \leq 0$. Якщо ж $s \geq 0$ і $\Delta < 0$, то рівняння (4.44) не має додатних коренів.*

Припустимо, що рівняння (4.44) має додатні корені. Не обмежуючи загальності, припустимо, що воно має чотири додатних корені, які позначимо відповідно z_1, z_2, z_3, z_4 . Тоді рівняння (4.43) має чотири додатних корені

$$w_1 = \sqrt{z_1}, w_2 = \sqrt{z_2}, w_3 = \sqrt{z_3}, w_4 = \sqrt{z_4}.$$

Позначимо

$$\tau_k^{(j)} = \frac{1}{w_k} \left[\arcsin \frac{a_1 w_k^3 + a_3 w_k}{\sqrt{b_2^2 w_k^2 + (b_3 - b_1 w_k^2)^2}} - \varphi + 2(j-1)\pi \right],$$

$$k = 1, 2, 3, 4, j = 0, 1, \dots$$

Тут φ є розв'язком

$$\sin \varphi = \frac{-b_2 w_k}{\sqrt{b_2^2 w_k^2 + (b_3 - b_1 w_k^2)^2}}, \quad \cos \varphi = \frac{b_3 - b_1 w_k^2}{\sqrt{b_2^2 w_k^2 + (b_3 - b_1 w_k^2)^2}}.$$

Тоді, як випливає з другого рівняння (1.43), $\pm i w_k$ є парами чисто уявних коренів рівняння (4.41) при $\tau = \tau_k^{(j)}$, $k = 1, 2, 3$, $j = 0, 1, \dots$. Видно, що

$$\lim_{j \rightarrow \infty} \tau_k^{(j)} = \infty, \quad k = 1, 2, 3$$

Отже, означимо $\tau_0 = \tau_{k_0}^{(j_0)} = \min_{1 \leq k \leq 3, j \geq 1} \{\tau_k^{(j)}\}$, $w_0 = w_{k_0}$.

ТЕОРЕМА 4.1. *Припустимо, що всі головні мінори Гурвіціана*

$$\begin{vmatrix} a_1 & 1 & 0 & 0 \\ a_3 + b_2 & a_2 + b_1 & a_1 & 1 \\ 0 & a_4 + b_3 & a_3 + b_2 & a_2 + b_1 \\ 0 & 0 & 0 & a_4 + b_3 \end{vmatrix} \quad (4.49)$$

додатні.

Якщо $s \geq 0$ і $\Delta < 0$, тоді всі корені рівняння (4.41) мають від'ємні дійсні частини при всіх $\tau \geq 0$. Якщо $s < 0$ або $s \geq 0$, $z^* > 0$ і $h(z^*) \leq 0$, тоді всі корені рівняння (4.41) мають від'ємні дійсні частини при $\tau \in [0, \tau_0)$.

Доведення. При $\tau = 0$ рівняння (4.41) набуває вигляду

$$\lambda^4 + a_1 \lambda^3 + (a_2 + b_1) \lambda^2 + (a_3 + b_2) \lambda + (a_4 + b_3) = 0. \quad (4.50)$$

За критерієм Гурвіца всі корені рівняння (4.50) мають від'ємні дійсні частини тоді і тільки тоді, коли всі головні мінори гурвіціана (4.49) додатні.

Якщо $s \geq 0$ і $\Delta < 0$, то за лемою 4.2 рівняння (4.41) не має коренів з нульовою дійсною частиною для всіх $\tau \geq 0$. Коли $\tau < 0$ або $s \geq 0$, $z^* > 0$ і $h(z^*) \leq 0$, то за лемою 4.2 маємо, що коли $\tau \neq \tau_k^{(j)}$, $k = 1, 2, 3$, $j \geq 1$, то рівняння (4.41) не має коренів з нульовою дійсною частиною і τ_0 є мінімальним значенням τ , таким, що рівняння (4.41) має чисто уявні корені. Згідно леми 1 коренів з додатньою дійсною частиною рівняння (4.41) при цьому не має. Теорема доведена.

Доведений вище результат можна переформулювати в термінах коефіцієнтів моделі імунного захисту, таким чином отримавши достатню умову стійкості.

ТЕОРЕМА 4.2. Припустимо, що коефіцієнти моделі імунного захисту (4.40) задовольняють умови теореми 4.1.

Тоді якщо $s \geq 0$ і $\Delta < 0$, то стан рівноваги (V^*, F^*, C^*, m^*) системи ЗДР (4.40) є абсолютно стійким (асимптотично стійким для всіх $\tau \geq 0$). Якщо ж

$s < 0$ або $s \geq 0$, $z^* > 0$ і $h(z^*) \leq 0$, тоді стан рівноваги (V^*, F^*, C^*, m^*) системи ЗДР (4.40) є асимптотично стійким при $\tau \in [0, \tau_0)$.

4.5. Дослідження щільності кісткової тканини.

Дослідженню функції структури тканини та встановленню механізмів адаптації м'яких тканин з тканиною кістки в останні роки приділяється велика увага вітчизняних та зарубіжних вчених [130-133, 136-140]. Такий інтерес пояснюється двома причинами: з механічної точки зору кістка – історично найбільш вивчена тканина і її поведінка є простішою порівняно з поведінкою м'яких тканин, а також більше відомо про механіку кістки відносно її структури [145-149]. Кістка також є хорошою ілюстрацією ієрархічного принципу функції структури, який властивий всім біологічним тканинам.

Кістка людини та інших ссавців класифікується двома типами [130, 133]: коркова кістка, ще відома як компактна кістка та трабекулярна кістка (Trabecular), відома як решітчаста або губчаста. Ці два типи виділені на основі пористості та мікроструктури кістки. Коркова кістка набагато щільніша, її пористість складає від 5 % до 10 %. Вона є первинною, розташована між довгими кістками і формує зовнішню раковину навколо решітчастої кістки в кінці суглобу та хребта [149].

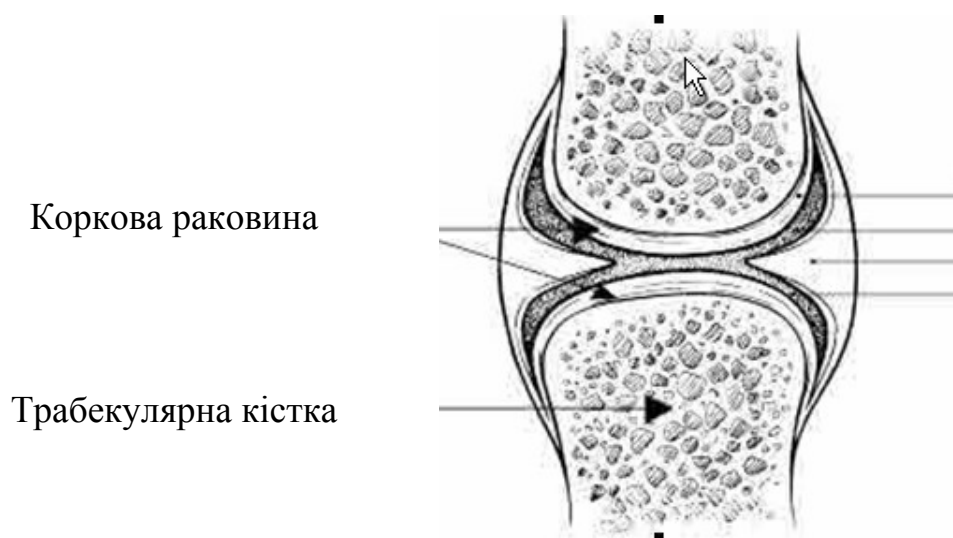


Рис 4.1. Коркова раковина навколо колінного суглобу кістки

Основа структури першого рівня коркової кістки – остеон (osteons). Трабекулярна кістка є більш пористою (від 50 % до 90 %).

Як і всі біологічні тканини, коркова кістка має ієрархічну багаторівневу структуру [135].

Рівень 0. Твердий матеріал (Solid Material).

Рівень 1. Вторинний остеон (Secondary Osteons).

Первинний остеон (Primary Osteons).

Сіткоподібний (Plexiform).

Проміжна кістка (Interstitial Bone).

Рівень 2. Пластинчастий (Lamellae).

Лакунарний (Lacunae).

Цементуючий (Cement Lines).

Рівень 3. Колаген (Collagen).

Мінерал (Mineral).

З'єднання (Composite).

Compact bone
(Osteon, Haversian system)

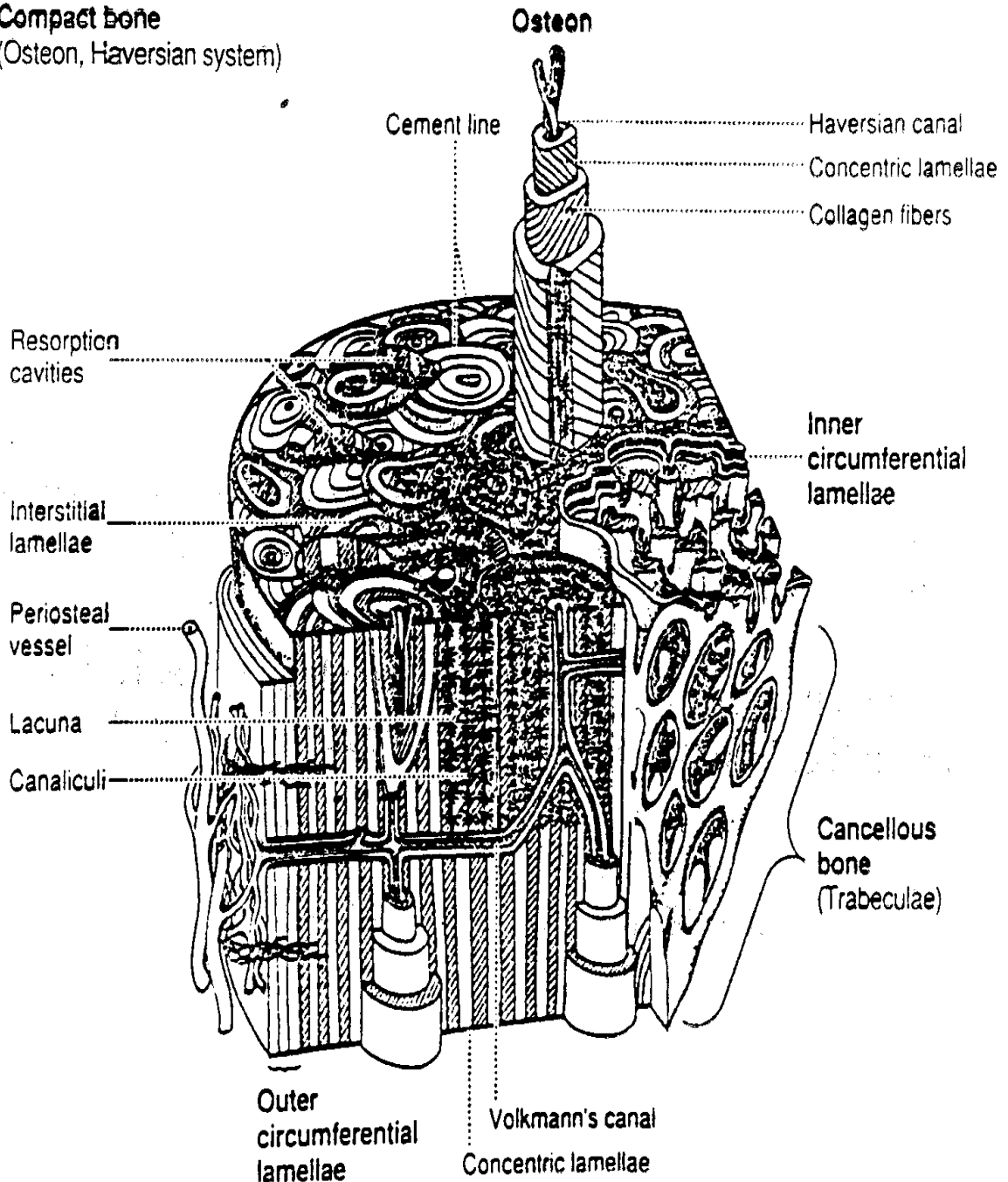


Рис. 4.2. Ієрархічна структура коркової кістки

Як відомо [154], різні аспекти структури коркової та трабекулярної кісток впливають на їх функцію, виражену механічними властивостями жорсткості та сили. Важливо відмітити, що можна визначити матеріальні властивості як жорсткості так і сили для кожного рівня структури [155]. Крім того, матеріальні властивості на більш макроскопічному рівні є функцією від

жорсткості, сили, просторового розташування або архітектури матеріалів на більш мікроскопічному рівні. Жорсткість коркової кістки (жорсткість на 0-у рівні) є функцією розташування остеона (архітектура на 1-у рівні), жорсткості окремого остеона (матеріальні властивості на 1-у рівні), степені мінералізації остеона (просторове розташування та властивості матеріалів на 3-у рівні і т.д. Одна з цілей біомеханіки полягає в тому, щоб визначити, як зміни певного структурного рівня, можливо через захворювання або свідоме втручання (з метою змінити структуру кістки), впливають на здатність навантаження кістки.

Навантаження на цілу кістку (визначене, наприклад, аналізом оптимізації [153]) спричиняє загальну напруженість та напруженість на кожному рівні структури. Величина загальної напруженості та напруженості, спричиненої вантажем – функція ієрархічних матеріальних властивостей. Далі загальна напруженість та напруженість на певному рівні сприймаються клітиною, щоб визначити, чи є їх величини в допустимих межах. В протилежному випадку клітина може діяти на матрикс кістки, щоб змінити її структуру, а отже, і її механічні властивості. Клітини, що діють разом, як припускають, регулюють структуру кістки у відповідь на їх механічне подразнення. Це в свою чергу визначає напруженість та силу тканини. Захворювання, які впливають на здатність клітин відчувати механічну напруженість або змінювати кістковий матрикс, можуть призвести до зміни механічних властивостей кістки.

Оскільки, коркова та трабекулярна кістки мають ієрархічну структуру, можна визначити їх механічну поведінку, включаючи загальну напруженість та напруженість і матеріальні властивості на кожному рівні структури. Наприклад, властивості коркової кістки часто визначають механічною дією розтяжного шаблону на корковий діафіз [153].

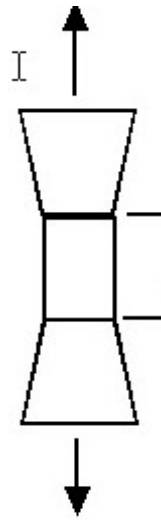


Рис. 4.3. Дія механічного шаблону на коркову кістку

Наступне рівняння пов'язує середню напруженість на 0-у рівні та напруженість на першому рівні [153]

$$\sigma_{ij}^{eff} = \sigma_{ij}^0 = \frac{1}{V^{test}} \int_{V^{test}} \sigma_{ij}^{osteon} dV^{test} = \frac{1}{V^1} \int_{V^1} \sigma_{ij}^1 dV^1,$$

де σ_{ij}^{eff} – ефективна напруженість, σ^0 – напруженість на 0-у рівні, V^{test} – об'єм, V^1 – об'єм на першому рівні, σ^{osteon} – напруженість на рівні остеон, σ^1 – напруженість на 1-у рівні. Запишемо ці ж співвідношення для середньої напруженості

$$\varepsilon_{ij}^{eff} = \varepsilon_{ij}^0 = \frac{1}{V^{test}} \int_{V^{test}} \varepsilon_{ij}^{osteon} dV^{test} = \frac{1}{V^1} \int_{V^1} \varepsilon_{ij}^1 dV^1,$$

де ε – напруженість.

Навіть за умови, коли можливо записати напруженість для певного макроскопічного рівня, цього не можна зробити для мікроскопічного рівня, оскільки на цьому рівні властивості змінюються за різними фазами матеріалу.

Будемо вважати, що є n фаз мікроструктури, макроскопічного рівня, тоді можна записати рівняння

$$C_{ijkl}^{eff} = C_{ijkl}^0 = \frac{1}{V^{test}} \int_{V^1} C_{ijpq}^1 M_{pqkl}^1 dV^1 + \frac{1}{V^{test}} \int_{V^2} C_{ijpq}^2 M_{pqkl}^2 dV^2 + \\ + \dots + \frac{1}{V^{test}} \int_{V^N} C_{ijpq}^N M_{pqkl}^N dV^N = \sum_{n=1}^N \frac{1}{V^{test}} \int_{V^n} C_{ijpq}^n M_{pqkl}^n dV^n$$

або в матричній формі

$$[C^{eff}] = [C^0] = \sum_{n=1}^N \frac{1}{V^{test}} \int_{V^n} [C^n] [M^n] dV^n$$

$\begin{matrix} 6 \times 6 & 6 \times 6 \end{matrix}$

де M – матриця локалізації напруженості. Форма M може змінюватись в залежності від прийнятої моделі (для обчислювальних або аналітичних моделей). Ефективні властивості на макроскопічному рівні залежать від мікроскопічних властивостей $[C]$ та просторового розташування або архітектури мікроскопічних властивостей $[M]$. Останнє рівняння виражає співвідношення між будь-якими рівнями, визначеними в [корковій](#) і трабекулярній структурах кістки.

Основні особливості структури трабекулярної кістки на 1-у рівні - висока пористість, заплутана архітектура та складна структура пластини трабекули. Ці особливості поряд з мінералізацією можуть бути головними факторами, що впливають на ефективність напруженості трабекулярної кістки. Однак, існує істотна відмінність між структурою 1-го рівня коркової кістки та структурою 1-го рівня трабекулярної кістки.

Будемо визначати кількість трабекулярної структури, використовуючи методи стереології. Ці методи використовують точки та лінії, проведені між структурними межами фаз. Спочатку це було зроблено в 2-вимірному зображенні з використанням мікроскопа, але зараз є можливість з допомогою комп'ютерних алгоритмів отримувати якісне 3-вимірне зображення.

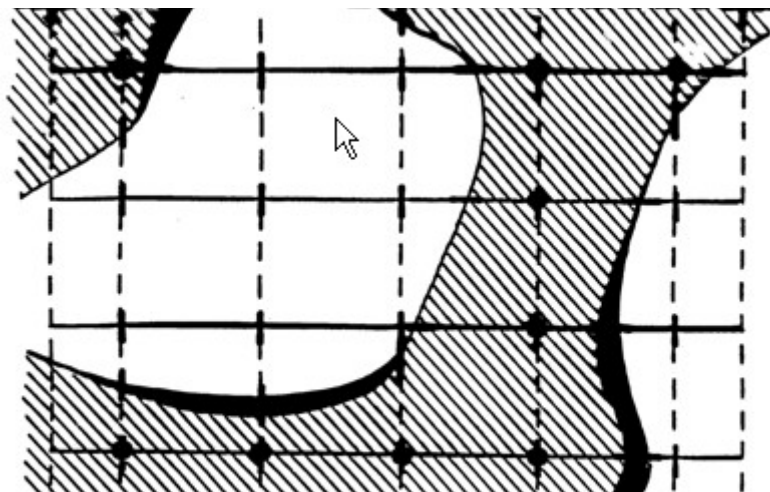


Рис. 4.4. Трабекула

Береться до уваги міра фракції (інверсія пористості об'єму) кістки – відношення перетинів сітки в межах кістки до загальної кількості перетинів сітки та кількість перетинів між кісткою та найближчим місцем сутності (на рис. 4.4. білий колір). Ця величина – фактично функція орієнтації, оскільки сітки обертаються від 0 до 360 градусів і перетинаються, як показано на малюнку, утворюючи еліпс. Ці виміри дають можливість оцінити відстані між перетинами кісткового мозку і вирахувати ширину трабекули. В тривимірному варіанті результатом обертання сіток є еліпсоїд. Загальне рівняння для еліпсоїда (без врахування термінів, які відображають орієнтацію еліпсоїда) є таким

$$A_{11}x_1^2 + 2A_{12}x_1x_2 + 2A_{13}x_1x_3 + A_{22}x_2^2 + 2A_{23}x_2x_3 + A_{33}x_3^2 = 1,$$

де A_{ij} – коефіцієнти еліпсоїда (середні міри довжин точок перетину в основній системі координат).

Останню рівність можна переписати у вигляді [135]

$$\{x_1 \ x_2 \ x_3\} \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{13} & A_{23} & A_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = 1 \Rightarrow x_i A_{ij} x_j = 1.$$

Окрім статистичних експериментальних підходів встановлення співвідношення функції та структури кістки, є також аналітичні обчислювальні моделі. Найбільш відомими та простими з них є моделі Реусса та Войта, які припускають, що напруженість у всіх рівнях мікроструктури рівна. Це веде до обчислювання з приросту ефективної жорсткості частки міри жорсткості кожної стадії. В термінах загальних відношень між жорсткістю та структурою, місцева матриця – просто матриця ідентичності. Якщо ми виконуємо інтеграцію об'єму кожної мікроструктури, то отримуємо усереднену матрицю, яка виражає частку об'єму кістки на діагоналі

$$\begin{aligned}
 [C^{eff}] = [C^0] &= \sum_{n=1}^N \frac{1}{V^{test}} \int_{V^n} [C^n] [M^n] dV^n = \\
 &= \sum_{n=1}^N [C^n]_{6 \times 6} \frac{1}{V^{test}} \int_{V^n} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} dV^n = \sum_{n=1}^N [C^n]_{6 \times 6} \begin{bmatrix} v_f^n & 0 & 0 & 0 & 0 & 0 \\ 0 & v_f^n & 0 & 0 & 0 & 0 \\ 0 & 0 & v_f^n & 0 & 0 & 0 \\ 0 & 0 & 0 & v_f^n & 0 & 0 \\ 0 & 0 & 0 & 0 & v_f^n & 0 \\ 0 & 0 & 0 & 0 & 0 & v_f^n \end{bmatrix}
 \end{aligned}$$

де тональна частота кількості – частка об'єму n -ої стадії. При припущенні, що напруженість кожної стадії рівна, отримуємо модель Реусса. Вона протилежна до моделі Войта, в якій має вагу об'єм кожної стадії

$$[S^{eff}] = \sum_{n=1}^N [S^n]_{6 \times 6} \begin{bmatrix} v_f^n & 0 & 0 & 0 & 0 & 0 \\ 0 & v_f^n & 0 & 0 & 0 & 0 \\ 0 & 0 & v_f^n & 0 & 0 & 0 \\ 0 & 0 & 0 & v_f^n & 0 & 0 \\ 0 & 0 & 0 & 0 & v_f^n & 0 \\ 0 & 0 & 0 & 0 & 0 & v_f^n \end{bmatrix},$$

де S^{eff} – матриця ефективної інверсії жорсткості, S^n – матриця інверсії жорсткості на n -й стадії, $[S] = [C]^{-1}$.

Через припущення про розподілення напруженості на мікроскопічному рівні обчислювальні та аналітичні моделі забезпечують обмеження на реальну жорсткість.

Висновки до розділу 4.

В багатьох розділах математики, фізики та механіки використовуються системи лінійних алгебраїчних рівнянь з теплицевими матрицями. Так проблема побудови функції Ляпунова по суті є проблемою знаходження

розв'язків лінійних диференціальних рівнянь першого порядку, які в свою чергу зводяться до систем лінійних алгебраїчних рівнянь. Таким чином, ця задача тісно пов'язана з проблемою знаходження спектру лінійного оператора.

Ідеологічною основою розв'язку цієї проблеми є спектральні властивості відповідних скінченних матриць. Використовуючи методи спектрального аналізу, можна побудувати функцію Ляпунова, яка задовольняє необхідним та достатнім умовам асимптотичної стійкості нульового розв'язку. В [32] проблема Ляпунова зведена до проблеми знаходження спектру однорідного інтегрального рівняння Фредгольма другого роду i , зокрема, до проблеми спектра матриці Тепліца. Зокрема показано, що для динамічних одновимірних систем будується інтегральний оператор, який задається якраз матрицею Тепліца.

Отже, тепліцеві матриці мають місце при дослідженні різноманітних теоретичних і прикладних проблем.

ВИСНОВКИ

Дисертаційна робота присвячена побудові алгоритмів розв'язування СЛАР з тепліцевими λ -матрицями, елементами яких є алгебраїчні і тригонометричні поліноми. Розроблені алгоритми можуть бути застосовані в математичному моделюванні при розгляді прикладних задач економіки, медицини, механіки, коли виникає необхідність розв'язувати і досліджувати розв'язки СЛАР з тепліцевими λ -матрицями.

Математичні моделі, одержані теоретичні результати та обчислювальні алгоритми, запропоновані в роботі, можуть бути використані при розробці програмних засобів та для подальших досліджень, пов'язаних з розв'язуванням тепліцевих систем лінійних алгебраїчних рівнянь з λ -матрицями, що виникають в багатьох задачах моделювання, зокрема в медичних.

В роботі одержано такі основні результати:

- розроблено та обгрунтовано ефективні алгоритми, які узагальнюють числові методи лінійної алгебри для тепліцевих (ганкелевих) матриць з дійсними та комплексними елементами, що базуються на економічних схемах знаходження оберненої матриці, на випадок розв'язування СЛАР з тепліцевими (ганкелевими) λ -матрицями. Проведено аналіз оцінок характеристик створених алгоритмів;

- запропоновано схему розв'язування блочно-тепліцевих (блочно-ганкелевих) СЛАР з тригонометричними елементами порядку l ;

- створено моделі для реалізації алгоритмів розв'язування СЛАР з блочно-тепліцевими λ -матрицями з поліноміальними елементами на багатопроцесорних обчислювальних системах;

- розроблено та обгрунтовано алгоритми розв'язування тепліцевих та ганкелевих СЛАР з тригонометричними λ -матрицями;

– здійснено аналіз похибок заокруглення при реалізації запропонованих алгоритмів на ЕОМ;

– запропоновано алгоритм розв’язування систем лінійних алгебраїчних рівнянь з теплицевими λ -матрицями за допомогою гіллястих ланцюгових дробів;

– для математичних моделей імунного захисту організму та щільності кісткової тканини досліджено стійкість розв’язків, і на їх основі розроблено алгоритми, які впроваджено в навчальний процес у Тернопільському державному медичному університеті імені І.Я.Горбачевського.

