МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

На правах рукопису

Семчишин Ліда Михайлівна

УДК 518.25

АЛГОРИТМИ КОМП'ЮТЕРНОЇ АЛГЕБРИ ДЛЯ РОЗВ'ЯЗАННЯ МАТРИЧНИХ РІВНЯНЬ

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 – 2011

ЗМІСТ

ВСТУП

Однією з найважливіших складових частин базового програмного забезпечення сучасних комп'ютерних систем є обчислювальні методи алгебри, тобто методи, які ґрунтуються на побудові скінченної послідовності дій над скінченною множиною чисел. Використовуючи такі методи, розв'язок математичної задачі отримують у вигляді числового результату. В останні роки зросла також потреба широкого використання алгоритмів комп'ютерної алгебри для розрахунку і оптимізації економіко-математичних моделей, в задачах лінійного і параметричного програмування. Подібні задачі зустрічаються також в задачах хімічної кінетики, при розв'язанні задач синтезу великих електронних схем, в задачах будівельної механіки, в динамічному програмуванні.

Вагомий внесок у розвиток методів комп'ютерної алгебри та їх застосування в математичному моделюванні зробили В.М. Фаддєєв, Д.К. Фаддєєва [99; 100], В.В. Воеводин [13-20], В.М. Кублановська [45], М.О. Недашковський [60-71], І.М. Молчанов, О.М. Хіміч [102; 103], О.Ф. Сопронюк [87], І.І. Босікова [8-10], О.Я. Ковальчук [41], Г.І. Малашонок [54], Е.Е. Тиртишніков [95], Дж.Х. Уілкінсон [97], M.T. Macclelian [130], E.H. Bareiss [112], J.D. Lipson [128], J. Smit [139], R.T. Moenk [133], S. Cabay [114] та інші.

Проте в практичному застосуванні виникають проблеми як числового, так і символьного розв'язування квадратних матричних рівнянь, алгебричних матричних рівнянь довільного порядку. Досліджуючи ті чи інші процеси або явища, використовуючи математичні методи й обчислювальні системи, спочатку будують математичну модель досліджуваного об'єкта, тобто описують об'єкт за допомогою математичних співвідношень (системи лінійних чи нелінійних рівнянь, диференціальних рівнянь тощо). Як результат одержують математичну задачу, яку розв'язати або дуже важко, або неможливо. Тоді побудовану математичну модель дискретизують, тобто перетворюють до такого вигляду, щоб розв'язок можна було знайти (звичайно, з певною похибкою) у вигляді числового результату за допомогою послідовності арифметичних і логічних операцій або в аналітичному вигляді. Таке перетворення виконують, застосовуючи обчислювальні методи. Закономірно, що на цьому етапі виникає низка проблем, пов'язаних зі збіжністю алгоритмів, їхньою обчислювальною стійкістю, оцінкою похибки розв'язку і т. ін. Реалізація дискретної моделі, зазвичай, відбувається за допомогою ЕОМ. Числовий результат розв'язування дискретної задачі аналізують з метою перевірки адекватності побудованої математичної моделі реальній дійсності. Якщо з'ясується, що математична модель недостатньо відображає реальну дійсність, її уточнюють, і весь процес дослідження повторюється.

Першою моделлю, що описує взаємозв'язок економіки і математики, стала міжгалузева балансова модель Леонтьєва [2; 4; 21; 24; 26; 30; 34-38; 40; 42; 46; 49; 51; 92; 93; 106].

Питання вивчення міжгалузевих моделей в економіко-математичних дослідженнях актуальне, відповідає на важливі методологічні та змістовні питання економічної науки, допомагає оцінити можливості та перспективи використання математичного моделювання в економіці.

Використання математичного моделювання в економіці дозволяє поглибити кількісний економічний аналіз, розширити сферу економічної інформації, інтенсифікувати економічні розрахунки.

Відзначимо, що проведення досліджень за допомогою математичного моделювання в економіці зумовило стійкий інтерес до використання комп'ютерів для символьних перетворень та одержання аналітичних розв'язків. Це дає можливість дослідження не тільки кількісних, а й якісних характеристик об'єкта, що моделюється.

Застосування методів економіко-математичного моделювання дає змогу аналізувати якісно і кількісно складні економічні процеси. Нові методи моделювання, засновані на строгих математичних розв'язаннях економічних завдань із застосуванням виявлених законів економіки виробництва, у поєднанні із сучасною обчислювальною технікою, сприяють створенню високоефективних систем для аналізу стану та науково обґрунтованого прогнозування розвитку економіки підприємств, галузей і країни загалом, дають можливість усвідомлено управляти економічними процесами виробництва.

Побудова та дослідження математичних моделей прикладних задач з використанням методів лінійної алгебри традиційно пов'язані із задачею розв'язування систем лінійних алгебричних рівнянь (СЛАР) [8-10; 14; 25; 32; 35; 47; 111; 117; 124; 127; 136; 140; 145].

Практичне застосування моделей Леонтьєва висуває до методів розв'язування матричних рівнянь ряд вимог, а саме: робота з матрицями великої розмірності, погана зумовленість і виродженість матриці нормативних коефіцієнтів, цілочислові змінні, що значно звужує можливість застосування відомих числових методів, нечітка інформація про значення нормативних коефіцієнтів, розрідженість матриці нормативних коефіцієнтів тощо.

Недостатня вивченість СЛАР з $\lambda-$матрицями від багатьох змінних, які можна застосовувати в узагальнених моделях Леонтьєва, виокремлює задачу пошуку нових ефективних алгоритмів комп'ютерної алгебри як особливо актуальну.

У значній кількості прикладних задач виникає необхідність використання динамічних міжгалузевих моделей в економіці. Тому питання міжгалузевих моделей розглядаються у багатьох публікаціях вітчизняних і зарубіжних учених. Застосування моделей Леонтьєва набуло широкого практичного значення в різних сферах науки, особливо в економічних дослідженнях і для розв'язування еколого-економічних математичних задач. Використанню міжгалузевих моделей присвячено праці О.Ф. Волошина, Н.Б. Чорней [21], В.С. Григорківа [26], І.М. Ляшенка [52], М. Інтрилігатора [33], С.А. Жукова [34], А.Ф. Кабака [37], В.І. Кудіна [46], М.О. Недашковського [64].

Наше дослідження присвячене побудові алгоритмів розв'язування широкого кола систем алгебричних рівнянь у моделях Леонтьєва і застосуванню СЛАР з $\lambda-$ матрицями.

Актуальність теми. Для об’єктивного аналізу ефективності розв'язування систем лінійних алгебричних рівнянь у моделях Леонтьєва необхідні розробка та дослідження математичних моделей взаємодії економіки та математики з використанням, зокрема, систем лінійних алгебричних рівнянь з $\lambda-$матрицями.

Зважаючи на недостатню ефективність існуючих методів розв'язування СЛАР для міжгалузевої моделі Леонтьєва, актуальним є застосування методу послідовного аналізу варіантів, який добре проявив себе при розв'язуванні багатьох прикладних задач.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконана на кафедрі економічної кібернетики та інформатики Тернопільського національного економічного університету і є складовою частиною досліджень наукової держбюджетної теми "Аналітична теорія гіллястих ланцюгових дробів і її застосування в комп'ютерній алгебрі, диференціальних рівняннях і в моделюванні електронного бізнесу" (номер держреєстрації №0110U001442).

Мета і завдання дослідження. Мета дисертаційної роботи – побудова методів розв'язування окремих видів СЛАР з $\lambda-$матрицями в моделях Леонтьєва. Для досягнення мети були поставлені такі завдання:

1) розробити стійкі методи розв'язування розріджених СЛАР із блочними елементами. Знайти кількість записів для символьного розв'язання та кількість арифметичних операцій для числової реалізації алгоритму множення матриць. Охарактеризувати складність алгоритму з погляду комп'ютерної алгебри. Провести порівняння отриманого алгоритму та існуючих методів;
2) створити алгоритми розв'язування СЛАР з багатовимірними $\lambda-$матрицями для динамічної моделі Леонтьєва. Розвинути скінченно-різницевий підхід до розв'язування систем лінійних алгебричних рівнянь із двовимірними $\lambda-$матрицями. Дослідити зведення системи лінійних алгебричних рівнянь з $m-$вимірними $\lambda-$матрицями до системи з числовими елементами й обчислити кількість арифметичних операцій;
3) розробити новий підхід до розв'язування кліткових алгоритмів для систем лінійних алгебричних рівнянь з блочними елементами в моделях

Леонтьєва. Описати алгоритм для тридіагональної блочної системи лінійних алгебричних рівнянь. Розглянути блочний варіант другого алгоритму відсічних систем, а також провести розрахунок кількості операцій, потрібних для його реалізації;

4) визначити умови розв'язності прямокутних і розріджених систем лінійних рівнянь з $\lambda-$матрицями в моделях Леонтьєва. Розробити новий підхід до розв'язання СЛАР Леонтьєва із прямокутними матрицями. Створити оптимізаційну модель з матрицями міжгалузевого балансу та запропонувати ефективний обчислювальний метод реалізації цієї моделі;
5) запропонувати новий підхід до розв'язування погано зумовлених СЛАР у моделі Леонтьєва. Виконати аналіз складності числової реалізації алгоритму розв'язування СЛАР на ЕОМ. Дослідити обчислювальну стійкість запропонованого алгоритму розв'язування СЛАР у моделі Леонтьєва. Охарактеризувати складність алгоритму та показати його ефективність з погляду комп'ютерної алгебри;
6) провести тестування якості запропонованих обчислювальних алгоритмів і відповідних оцінок їх основних характеристик.

Об'єкт дослідження – алгоритми комп'ютерної алгебри для розв'язування матричних рівнянь у моделях Леонтьєва.

Предмет дослідження – алгоритми розв'язування СЛАР з $\lambda-$матрицями в узагальнених моделях Леонтьєва.

Методи дослідження. У роботі застосовуються методи математичного моделювання для дослідження моделей міжгалузевого балансу; методи розв'язування СЛАР з $\lambda-$матрицями від двох змінних на основі скінченно-різницевого підходу; методи обчислювальної математики для побудови узагальнених та динамічних моделей Леонтьєва; елементи теорії прикладного програмування для побудови пакета програм; методи організації та проведення обчислювальних експериментів числового дослідження систем лінійних алгебричних рівнянь.

Наукова новизна одержаних результатів.

У процесі розв'язання поставлених задач уперше отримані такі наукові результати:

- розвинено та обґрунтовано матричну модель міжгалузевого балансу, проведено аналіз її статистичних характеристик;
- для підвищення швидкодії програмних засобів розв'язування СЛАР з $\lambda-$матрицями в моделях Леонтьєва удосконалено деякі алгоритми, що забезпечують розв'язування систем даного класу;
- для математичного моделювання динамічного варіанта моделі Леонтьєва розроблено методи комп'ютерної алгебри, які розширюють можливості сучасних дослідницьких комп'ютерних технологій;
- розроблено схему зведення СЛАР динамічного варіанта моделі Леонтьєва з $\lambda-$матрицями від багатьох змінних до систем з числовими коефіцієнтами спеціального вигляду;
- методами зворотного аналізу встановлено обчислювальну стійкість запропонованих алгоритмів до похибок заокруглення, які виникають при реалізації на комп'ютерних обчислювальних системах;
- розроблено комплекс програм для розширення можливостей системи MatLab і підвищення її функціональності з проблем розв'язування задач, які описуються системами лінійних алгебричних рівнянь з $\lambda-$матрицями.

Практичне значення отриманих результатів. Розроблено комплекс взаємозв'язаних алгоритмів і програм для комп'ютерного дослідження динамічних об'єктів, що описуються системами лінійних алгебричних рівнянь з $\lambda-$матрицями у моделях Леонтьєва. Ці алгоритми суттєво розширюють можливості вказаних комп'ютерних засобів проведення наукових та інженерних розрахунків у динамічних моделях.

Особистий внесок здобувача. Усі результати дисертаційної роботи, винесені на захист, отримані автором самостійно. У роботі [64], яка опублікована спільно з науковим керівником, співавтору належить ідея ефективного методу реалізації динамічної міжгалузевої моделі.

Апробація результатів дисертації. Основні результати дисертаційних досліджень доповідалися на конференціях: XII та XIII міжнародні наукові конференції імені академіка М. Кравчука (м. Київ, 15-17 травня 2008 р. та 13-15 травня 2010 р.); II міжнародна наукова конференція "Сучасні проблеми механіки та математики" (м. Львів, 25-29 травня 2008 р.); I міжнародна науково-методична конференція "Математичні методи, моделі та інформаційні технології в економіці" (м. Чернівці, 1-4 квітня 2009 р.); Український математичний конгрес – 2009 (м. Київ, 27-29 серпня 2009 р.).; неодноразово доповідалися на семінарах кафедри економічної кібернетики та інформатики Тернопільського національного економічного університету, на семінарі факультету прикладної математики Чернівецького національного університету ім. Ю. Федьковича і відділу числових методів та комп’ютерного моделювання Інституту кібернетики ім. В.М. Глушкова НАН України.

Публікації. На тему дисертації опубліковано 12 праць, у тому числі 7 статей, з них 5 статей у наукових журналах і збірниках наукових праць, які входять до переліку ВАК України з даної спеціальності, 5 тез доповідей, опублікованих у матеріалах наукових міжнародних конференцій.

Структура та обсяг дисертації.

Дисертаційна робота складається зі вступу, п'яти розділів, висновків, списку використаних джерел та кількох додатків. Список використаних джерел містить 146 найменувань і розташований на 14 сторінках. Загальний обсяг дисертації складає 163 сторінки.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД РЕЗУЛЬТАТІВ ІЗ РОЗВ'ЯЗАННЯ ТА ЗАСТОСУВАННЯ МАТРИЧНИХ РІВНЯНЬ

1.1. Комп'ютерна алгебра

Комп'ютерна алгебра, або *символьні і алгебраїчні обчислення,* надає можливість комп'ютерній системі маніпулювати математичними виразами, заданими у вигляді символів, а не чисел, подібно до того як це робиться в алгебрі за допомогою олівця на папері.

До недавнього часу комп'ютерні системи застосовували переважно для розв'язання задач із вхідними даними числового характеру. Комп'ютер виконував деяку сукупність арифметичних або логічних операцій, результатом яких знову ж таки був набір чисел (за винятком задач обробки текстів, що не належать до задач обчислювальної математики). Проте в останні десятиліття в електротехніці, теорії гравітації (для дослідження можливих варіантів загальної теорії відносності), у неврології, де система рівнянь моделює розповсюдження сигналу нервом, виник значний інтерес до використання ком'ютерів і для символьних перетворень та аналітичного розв'язання задач. Власне кажучи, і в "чистій математиці" завжди відчувалась необхідність використання обчислювальних систем для виконання рутинної роботи. Маючи справу переважно з точними числами (цілими і раціональними числами нескінченної точності) та алгебраїчними виразами в символьному вигляді, системи комп'ютерної алгебри дозволяють звільнити вчених від монотонної важкої роботи, пов'язаної з числовими похибками (зрізання і заокруглення). Отже, допомагають їм глибше зрозуміти різні явища, що вивчаються, не занурюючись у цифрові дані.

Однією з перших успішних спроб реалізації комп'ютерних символьних перетворень було створення і використання мови “Аналітик” для комп'ютерів серії "МИР" в Інституті кібернетики Академії наук України ще в 60-х рр. минулого століття. Інтерес до таких досліджень значною мірою зріс у зв'язку зі створенням комп'ютерів V та VI поколінь, дослідженнями з розробки квантових комп'ютерів.

Зупинимося на формулюванні ряду загальних понять та тверджень, корисних для проведення символьних перетворень на обчислювальних системах. У числовому аналізі для характеристики ефективності алгоритмів вживаються такі терміни: оцінка кількості виконуваних арифметичних операцій, обсяг пам'яті, необхідної для реалізації, числова стійкість, точність представлення початкових даних і т. д. З цього звичного "арсеналу" для оцінки якості методів комп'ютерної алгебри надалі нам буде потрібний лише об'єм пам'яті та час реалізації алгоритму. Також першочерговими виявляються нові важливі характеристики – форма комп'ютерного представлення результату, можливість участі людини в процесі перетворень у діалоговому режимі і т. д. Символьні перетворення ставлять високі вимоги до зовнішніх пристроїв комп'ютерної системи, мов програмування, що використовуються. Тому потрібні інші підходи й у трактуванні відомих класичних методів та створення принципово нових алгоритмів.

Насамперед зупинимось на декількох поняттях, розгляд яких необхідний для подальшого викладу.

Означення 1.1. *Терм* – це мовний вираз, призначений для позначення об’єктів.

Наприклад, вирази 1, 0 + 1, $\lim\limits_{x \to 0} \dfrac{\sin x}{x}$ є різними термами, які позначають один і той самий об’єкт алгебри багатосортних термів найбільш точно, відповідають назві «символьна алгебра», яку іноді застосовують замість терміна «комп'ютерна алгебра».

Означення 1.2. *Комп'ютерним алгоритмом* $f(\mathrm{a}_1, \mathrm{a}_2, \ldots, \mathrm{a}_\mathrm{n})$ називатимемо точне правило, за яким із множини $\{a_i\}$ символьних вхідних даних за допомогою знаків арифметичних операцій та дужок може бути одержаний розв'язок поставленої задачі.

Наведемо один допоміжний результат, який дозволяє одержувати оцінки необхідної кількості записів для багатьох символьних алгоритмів [62].

ТЕОРЕМА 1.1. *Нехай деяка обчислювальна задача із вхідними даними* $\{a_i\}$ *розв'язується на ЕОМ за алгоритмом* $y(\mathrm{a}_1, \mathrm{a}_2, \ldots, \mathrm{a}_\mathrm{n})$ *і складається з* k *кроків* y_j $(j = 1, 2, \ldots k)$. *Якщо на кожному кроці реалізації алгоритму* $f(\mathrm{A})$ *має місце хоча б один запис виду* $y_{\mathrm{ji}_1}(\mathrm{A}) * y_{\mathrm{ji}_2}(\mathrm{A})$, *який використовує результат попереднього кроку, то загальна складність* Q_y *задачі буде не меншою* 2^k, *але не більшою* H^k *записів, де* H *- найбільша ширина алгоритму на* k *кроках.*

При записуванні розв'язків математичних задач як в класичній математиці, так і в нових її розділах, результат подається як деяка послідовність (скінченна або нескінченна) математичних символів. Як зручний спосіб для запису розв'язку дуже часто використовується сума, ряд, нескінченний добуток та ланцюговий дріб.

Рядом, як відомо, називають нескінченну суму вигляду:

$$\sum_{i=1}^{\infty} a_i = a_1 + a_2 + \ldots + a_i + \ldots \qquad (1.1)$$

У багатьох теоретичних задачах і застосуваннях доводиться мати справу зі скінченними сумами, тобто виразами вигляду:

$$\sum_{i=1}^{N} a_i = a_1 + a_2 + \ldots + a_i + \ldots + a_N \ . \qquad (1.2)$$

Нескінченними добутками називають вирази:

$$\prod_{i=1}^{\infty} a_i = a_1 \cdot a_2 \cdot a_3 \cdot \ldots \cdot a_i \cdot \ldots \qquad (1.3)$$

Часто також доводиться мати справу із скінченними добутками:

$$\prod_{i=1}^{\infty} a_i = a_1 \cdot a_2 \cdot a_3 \cdot \ldots \cdot a_i \cdot \ldots \qquad (1.4)$$

Ланцюговим або неперервним дробом [7], [63], [91] називають вирази вигляду:

$$b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{\ddots}{+ \cfrac{a_i}{b_i + \ddots}}}} \tag{1.5}$$

Скінченними ланцюговими дробами називатимемо вирази вигляду:

$$b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \cfrac{\ddots}{+ \cfrac{a_N}{b_N}}}}}. \tag{1.6}$$

Для компактності запису дуже часто використовується форма запису ланцюгових дробів, запропонована Прінгсгеймом [7]:

$$b_0 + \sum_{i=1}^{N} \frac{a_i|}{|b_i} = b_0 + \frac{a_1|}{|b_1} + \frac{a_2|}{|b_2} + \frac{a_3|}{|b_3} + \ldots + \frac{a_N|}{|b_N}. \tag{1.7}$$

Існує зв'язок між рядами, нескінченними добутками і ланцюговими дробами. Для його встановлення потрібно розглянути послідовність $\{t_n\}$ - дробово-лінійних перетворень (ДЛП):

$$t_n\{w\} = \frac{a_n + c_n w}{b_n + d_n w}, \quad a_n d_n - b_n c_n \neq 0, \tag{1.8}$$

які іноді називають ще перетвореннями Мебіуса [7]. Говорять, що послідовність $\{t_n\}$ *породжує* послідовність $\{T_n\}$, якщо

$$T_0(w) = t_0(w), \; T_n(w) = T_{n-1}(t_n(w)), \quad n = 1, 2, 3, \ldots \tag{1.9}$$

Виявляється, що композиція відповідних дробово-лінійних перетворень дозволяє задати, зокрема, і ряди, і нескінченні добутки, і неперервні дроби.

Справді, нескінченний ряд можна розглядати як композицію дробово-лінійних перетворень $\{T_n\}$, для яких $c_n = 1, b_n = d_n = 0$

$$\left.\begin{array}{l} p_n(w) = a_n + w, \\ p_o(w) = p_o(w),\ \ p_n(w) = p_{n-1}(p_n(w)),\ \ \mathrm{n} = 1,2,3,\ldots.8 \end{array}\right\}. \qquad (1.10)$$

Нескінченний добуток може бути визначений як композиція дробово-лінійних перетворень $\{R_n\}$, в якій $a_n = b_n = d_n = 0$:

$$\left.\begin{array}{l} r_n(w) = c_n w, \\ R_o(w) = r_o(w), \\ R_n(w) = R_0(r_n(w)), \quad n = 1,2,3,\ldots \end{array}\right\}. \qquad (1.11)$$

Аналогічно й неперервний дріб визначається як композиція ДЛП вигляду:

$$\left.\begin{array}{l} S_0(w) = s_0(w), \quad S_n(w) = S_{n-1}(s_n(w)), \quad n = 1,2,3.\ldots. \\ s_0(w) = b_0 + w\ ,\ \ s_n(w) = \dfrac{a_n}{b_n + w} \end{array}\right\}. \qquad (1.12)$$

Нетривіальним узагальненням ланцюгових дробів є гіллясті ланцюгові дроби (ГЛД) [90].

Нескінченним гіллястим ланцюговим дробом називатимемо вираз D виду

$$D = \sum_{k_1=1}^{N} \frac{a_{k_1}|}{|b_{k_1}} + \sum_{k_2=1}^{N} \frac{a_{k_1k_2}|}{|b_{k_1k_2}} + \sum_{k_3=1}^{N} \frac{a_{k_1k_2k_3}|}{|b_{k_1k_2k_3}} + \ldots + \sum_{k_i=1}^{N} \frac{a_{k_1k_2k_3\ldots k_i}|}{|b_{k_1k_2k_3\ldots k_i}} + \ldots \qquad (1.13)$$

Скінченний дріб

$$D_m = \sum_{k_1=1}^{N} \frac{a_{k_1}|}{|b_{k_1}} + \sum_{k_2=1}^{N} \frac{a_{k_1k_2}|}{|b_{k_1k_2}} + \sum_{k_3=1}^{N} \frac{a_{k_1k_2k_3}|}{|b_{k_1k_2k_3}} + \ldots + \sum_{k_{mi}=1}^{N} \frac{a_{k_1k_2k_3\ldots k_{mi}}|}{|b_{k_1k_2k_3\ldots k_m}}$$

називають m - им підхідним дробом нескінченного ГЛД (1.13).

Надалі будуть потрібні позначення

$$D^m_{k_1k_2k_3\ldots k_i} = b_{k_1k_2k_3\ldots k_i} + \sum_{k_{s+1}=1}^{N} \frac{a_{k_1k_2k_3\ldots k_{s+1}}|}{|b_{k_1k_2k_3\ldots k_{s+1}}} + \ldots + \sum_{k_m=1}^{N} \frac{a_{k_1k_2k_3\ldots k_m}|}{|b_{k_1k_2k_3\ldots k_m}}. \qquad (1.14)$$

Як було відзначено раніше, і ряди, і нескінченні добутки, і ланцюгові дроби можна представляти як частковий випадок композицій ДЛП загального виду. Однак ця хороша властивість з надлишком компенсується іншими недоліками – композиції ДЛП дуже незручні для запису та для вивчення через надмірну громіздкість.

Проте виявляється, що існує зв'язок між дробово-лінійними композиціями достатньо загального виду і скінченними гіллястими ланцюговими дробами [62], якщо ввести до розгляду таку сукупність ДЛП:

$$\left.\begin{aligned} D_m^0 &= b_0 + \sum_{k_1=1}^{N} \frac{a_{k_1} + D_m^{k_1}}{b_{k_1} + D_m^{k_1}}; \\ D_m^{k_1} &= b_{k_1} + \sum_{k_2=1}^{N} \frac{a_{k_1 k_2} + D_m^{k_1 k_2}}{b_{k_1 k_2} + D_m^{k_1 k_2}}; \\ D_m^{k_1 k_2 \ldots k_m} &= b_{k_1 k_2 \ldots k_m} + \sum_{k_2=1}^{N} \frac{a_{k_1 k_2 \ldots k_m} + D_m^{k_1 k_2 \ldots k_i}}{b_{k_1 k_2 \ldots k_m} + D_m^{k_1 k_2 \ldots k_i}}. \end{aligned}\right\} \tag{1.15}$$

ТЕОРЕМА 1.2. *Композиція дробово-лінійних композицій (1.15) може бути представлена у вигляді гіллястого ланцюгового дробу канонічного виду і при цьому m - й підхідний ланцюговий дріб буде відповідати композиції m дробово-лінійних перетворень.*

Даний факт свідчить про те, що більшість алгоритмів з досить складною організацією можуть бути представлені ланцюговими дробами. Отже, достатньо широкий спектр задач комп'ютерної алгебри може бути розв'язаний за допомогою апарату ГЛД.

1.2. Системи лінійних алгебричних рівнянь спеціального вигляду

1.2.1. Алгоритми для розв'язання лінійних алгебричних систем із символьними елементами

Сьогодні існує й успішно розвивається декілька напрямків і концепцій щодо виконання символьних перетворень. Із комп'ютерних систем універсального характеру широкого розповсюдження набули REDUCE, muMATH, SCRATCHPAD, MATHEMATICA, MAPLE, MatLab, DERIVE, MatCad. З більшим чи меншим успіхом їх можна застосувати для різних задач комп'ютерної алгебри, у тому числі й розв'язання систем лінійних алгебричних рівнянь. Однак цей розділ ще не настільки високо розвинутий, як методи для числових систем.

Зупинимося на специфіці побудови ефективних алгоритмів для розв'язання символьних лінійних систем алгебричних рівнянь. Стосовно визначення обернених матриць та інших пов'язаних з цим задач, зокрема щодо розв'язання систем лінійних рівнянь і визначення детермінантів, безпосередньо алгоритми числового аналізу застосовувати не вдається, оскільки труднощі, що виникають у комп'ютерній алгебрі та в числовому аналізі суттєво відрізняються. Передусім, у комп'ютерній алгебрі немає проблеми числової стійкості, тому будь-який ненульовий елемент є хорошим провідним елементом для алгоритму виключення. У цьому аспекті комп'ютерна алгебра простіша від числового аналізу. З іншого боку, з'являється серйозна проблема так званого "розбухання даних" як у проміжних перетвореннях, так і в остаточному результаті.

Інша непроста проблема, особливо при обчисленні детермінантів матриць, – проблема ділення. За правилом Крамера, визначник матриці – це сума (можливо, зі знаками мінус) добутків елементів матриці. Отже, коли ці елементи належать до деякого кільця, то й визначник буде належати до цього ж кільця. Але методи, пов'язані з виключенням невідомих, вимагають ділення. Ці ділення можуть просто не бути виконаними. Наприклад, не можна ділити 5 на 2 в кільці лишків за модулем 10, але при цьому визначник матриці $\begin{pmatrix} 5 & 2 \\ 2 & 5 \end{pmatrix}$ коректно визначений і рівний 1. Навіть якщо ділення можливе, можуть знадобитися обчислення з дробами, які є надзвичайно трудоємкими в зв'язку з необхідністю обчислювати найбільший спільний дільник (що часто нетривіально).

Проаналізуємо тепер особливості розв'язання на ЕОМ систем алгебричних рівнянь з символьними елементами.

Нехай задана система такого вигляду:

$$\sum_{j=1}^{n} a_{i,j} x_j = a_{i,n+1} \quad (i = 1,2,3,...,n), \tag{1.16}$$

елементи $a_{i,j}$ якої є символами.

Для одержання запису розв'язків x_j $(j = 1,2,...,n)$ системи (1.16) розглянемо два підходи.

Подання невідомих у вигляді ієрархічної системи функціонально-зв’язаних виразів. Прикладом такого способу запису [3] розв'язків може бути така система:

$$x_1 = 3 \cdot F_1 + 5 \cdot F_2, \tag{1.17}$$

де $F_1 = F_2 + F_3 / F_4$ і $F_3 = F_4 / (F_5 + F_6)$, сукупність зв’язаних між собою співвідношень трьох рівнів.

Повний аналітичний запис розв’язків.

За такого підходу розв'язки систем (1.16) подаються у вигляді $x_j = y_j\ (a_{i1}, a_{i2},...,\ a_{in})\ (i = 1,2,...\ n)$. Права частина (1.16) за такого способу може бути досить громіздкою у зв'язку з відомими властивостями загального розв'язку.

Розглянемо тепер обидва підходи більш детально.

1.2.2. Концепція ієрархічних систем функціональних співвідношень

Запис розв'язків символьних систем виду (1.16) у вигляді ієрархічної системи функціонально-зв'язаних співвідношень не є, взагалі кажучи, повним вирішенням проблеми. Однак набір записів подібних співвідношень у багатьох випадках [3] дозволяє одержати відповіді на ті ж питання, що й повний аналітичний розв'язок. Для систем високих порядків це, очевидно єдиний прийнятний спосіб запису розв'язків на ЕОМ.

Проведемо оцінку ефективності можливого узагальнення деяких числових методів розв'язання алгебричних систем лінійних рівнянь у випадку систем з символьними елементами.

<u>*Метод виключення*</u>. Спочатку детально зупинимося на одному з найкраще відпрацьованих числових методів лінійної алгебри – алгоритмі виключення Гауса. Під методом Гауса, як і в числовому аналізі, розумітимемо алгоритм розв'язання систем, що складається з двох етапів: прямого і зворотного ходу. Прямий хід полягає в послідовному виключенні невідомих за функціонально-зв'язаними співвідношеннями:

$$\left.\begin{aligned} a_{i,j}^{(1)} &= a_{i,j} - \frac{a_{i,1}a_{1,j}}{a_{1,1}} && \left(i = \overline{2,n};\ j = \overline{2,n+1}\right); \\ a_{i,j}^{(k)} &= a_{i,j}^{(k-1)} - \frac{a_{i,k}^{(k-1)}a_{j,k}^{(k-1)}}{a_{k,k}^{(k-1)}} && \left(k = \overline{2,n-1};\ i = \overline{k+1,n};\ j = \overline{k+1,n+1}\right). \end{aligned}\right\} \tag{1.18}$$

Потім, під час зворотного ходу, послідовно визначаються всі невідомі за співвідношенням:

$$\left.\begin{aligned} x_n &= a_{n,n-1}^{(n-1)} / a_{n,n}^{(n-1)}; \\ x_i &= \left(a_{i,n+1}^{(i-1)} - a_{i,j}^{(i-1)}x_j\right) / a_{i,i}^{(i-1)}. \end{aligned}\right\} \tag{1.19}$$

Тепер оцінимо кількість символів, що записуються ЕОМ при реалізації алгоритму. Згідно (1.18) під час прямого ходу методу виключення на k - ому кроці буде виконано $(n-k)(n+1-k)\times 5$ записів на ЕОМ.

Отже, з точністю до головного члена, при виконанні прямого ходу буде виконано $5/3n^3$ записів. При зворотному ході запишеться $3/2n^2$ символів.

Отже, з точністю до головного члена, складність алгоритму – $5/3n^3$ записів. Для інших відомих методів лінійної алгебри оцінки часу запису будуть того ж самого порядку.

1.2.3. Концепція аналітичного запису розв'язків систем рівнянь

Розв'язати задачу запису розв'язків системи в аналітичному вигляді за рахунок тривіального узагальнення звичайних числових методів, як правило, не вдається.

Розглянемо, наприклад, можливості застосування методу виключення для розв'язання даної задачі. Дійсно, співвідношенями (1.18) та (1.19) можна скористатися, щоб поетапно представити невідомі $(i = 1, 2, \ldots n)$. Однак в методі Гауса для запису $a_{i,j}^{(k)}$ на k - ому поверсі використовуються 4 записи $(k - 1)$ - го поверху алгоритму $a_{i,j}^{(k-1)}, a_{i,k}^{(k-1)}, a_{j,k}^{(k-1)}, a_{k,k}^{(k-1)}$. Нескладні розрахунки показують, що тоді згідно з теоремою 1.1 лише для виконання прямого ходу методу і запису x_n потрібно буде 4^n елементарних записів. А для зворотного ходу кількість записів наростає ще скоріше і може бути оцінена як $O\left[\left(4^n\right)!\right]$. Такі обчислювальні схеми називають **NP-** складними [19].

Метод Крамера, що записує визначник матриці порядку n вигляді суми $n!$ добутків по n елементів матриці, для чисел має надзвичайно низьку ефективність: число операцій дорівнює $O\left[n(n!)\right]$ замість $O\left(n^3\right)$ в алгоритмі виключення. Але в комп'ютерній алгебрі ціна операції залежить від розміру даних, які використовуються. Тому для матриць поліномів від багатьох змінних ефективність алгоритму Крамера значно збільшується, порівняно з іншими методами, що базуються на виключенні.

Ефективні методи обчислення детермінантів обговорюються в роботі Е.Е. Тыртышникова [95]. Для обчислення визначників з символьними елементами пропонується застосовувати, власне кажучи, його означення:

$$A\begin{bmatrix} 1 & 2 & 3 & \cdots & n \\ 1 & 2 & 3 & \cdots & n \end{bmatrix} = \sum_{l=1}^{n-1} d_l \prod_{i=1}^{n} a_{il}, \text{ причому } l_i \neq l_j \text{ для всіх } i \neq j;$$

вектор $L = (l_1, l_2, \ldots l_n)$ – не що інше як перестановка елементів натурального ряду чисел від 1 до n ;

$\left\| a_{ij} \right\|$ - елементи матриці розміру $n \times n$;

$$d_l = \begin{cases} +1, & \text{якщо перестановка елементів парна,} \\ -1, & \text{якщо перестановка елементів непарна;} \end{cases}$$

k - кількість перестановок без урахування перестановок з нульовими елементами.

Для алгоритмів, що базуються на подібному підході, характерний один недолік, який звужує сферу їх застосування, – незручність та неефективність використання для розв'язання систем з розрідженими матрицями. Тому зупинимося на методах розв'язання символьних систем лінійних алгебричних рівнянь, що використовують апарат ланцюгових дробів [90]. Такий підхід придатний як для систем загального вигляду, так і для розріджених систем багатьох типів.

Спочатку опишемо загальний алгоритм розв'язання символьних систем з щільно заповненими матрицями. Для компактності подальших записів введемо ряд позначень. Під $A\begin{bmatrix} i_1 & i_2 & i_3 & \dots & i_s \\ j_1 & j_2 & j_3 & \dots & j_s \end{bmatrix}$, як і раніше, розуміється мінор, розміщений на перетині рядків $i_1, i_2, \dots, i_s$ та стовпців $j_1, j_2, \dots, j_s$ $(s = 1, 2, \dots, n)$; N - множина. $N = \{1, 2, 3, \dots, n\}$, і за аналогією через $N(k_1, k_2, \dots k_s)$ позначимо множину:

$$N(k_1, k_2, \dots k_s) = \{\, j \in N, j \ne k_1, k_2, \dots k_s \,\}.$$

З урахуванням цих позначень, на основі правила Крамера для системи (1.16) з символьними елементами, можемо записати:

$$x_i = \sum_{j_1 \in N} \cfrac{(-1)^{i+j_1} a_{j_1,n+1}}{\sum\limits_{j_2 \in N(j_1)} \cfrac{(-1)^{i+j_2} a_{j_2,i}}{\sum\limits_{j_3 \in N(i)} \cfrac{(-1)^{j_2+j_3}}{a_{j_1 j_3} + \dots \; \dots + \sum\limits_{j_{2n-1} \in N(j_1 \dots j_{2n-1})} \cfrac{(-1)^{j_{2N-2}+j_{2N-1}} a_{j_{2N-2}, j_{2N-1}}}{a_{j_{2N-3}, j_{2N-1}}}}}}.$$

Суттєвою відмінністю даного підходу [62] є, передусім його універсальність. Крім того, на відміну від інших алгоритмів, вдається одержати конструктивний аналітичний запис для розв'язків системи.

Складність даного алгоритму – це $10n^2(n\,!)$ елементарних записів. Як і алгоритм, запропонований в [95], цей метод передбачає дуже високі вимоги до пам'яті, швидкодії, зовнішніх пристроїв ЕОМ. Однак сама ідея використання ланцюгових дробів для запису розв'язків лінійних алгебричних систем дуже приваблива для розріджених систем з символьними елементами, про що і йтиме мова далі.

1.3. Розв'язування матричних рівнянь, пов'язаних з моделями Леонтьєва

Моделі Леонтьєва – надзвичайно ефективний апарат для моделювання багатьох задач економіки та екології і зводяться до специфічних матричних рівнянь. Математичний апарат для побудови алгоритмів розв'язання подібних задач ще недостатньо розроблений, тому актуальними є дослідження в цій галузі. Зауважимо, що системи лінійних алгебричних рівнянь доводиться розв'язувати для дослідження різних економіко-математичних процесів не лише у моделях Леонтьєва.

Першою моделлю, що описує взаємозв'язок економіки і математики, була міжгалузева балансова модель Леонтьєва [26; 75]:

$$X(t) = A(t)X(t) + Y(t),$$

або

$$X = \left(E - A(t)\right)^{-1} Y(t), \tag{1.20}$$

де $\left(E - A(t)\right)^{-1}$ – матриця коефіцієнтів повних потреб у випуску продукції для одержання одиниць відповідних видів кінцевої продукції. Зміст коефіцієнтів b_{ij}, що утворюють матрицю $B = \left(E - A(t)\right)^{-1}$, у тому, що вони характеризують об'єми виробництва продукції i, необхідні для одержання одиниці кінцевої продукції j - го виду.

Для наведеної моделі Леонтьєва характерна складність роботи з матрицями великої розмірності, погана обумовленість та виродженість матриці нормативних коефіцієнтів, цілочислові змінні, нечітка інформація про значення нормативних коефіцієнтів, розрідженість матриці. Це значно звужує можливість застосування відомих числових методів лінійної алгебри.

Проблему виродженості в багатьох випадках намагаються долати за допомогою методів знаходження псевдооберненої матриці [22; 48], що далеко не завжди приводить до ефективного розв'язання поставленої проблеми.

Проблемі цілочисельності змінних присвячено також багато праць, серед яких окремо слід виділити [55; 56; 59; 103; 110; 115; 125; 129; 134; 141].

Проблеми великої розмірності та розрідженості, як правило, при розв’язуванні економіко-математичних задач завжди стоять поруч. При цьому використовуються відомі факти з теорії розріджених матриць та розробляються нові [74; 96; 109; 142; 146].

Реалізації методів знаходження розв'язків балансових моделей на ЕОМ та програмному забезпеченню ЕОМ присвячені роботи [28; 29; 56; 58; 101; 116; 118; 121; 122]. Проте більшість обчислювальних методів лінійного та нелінійного програмування мають недоліки, пов’язані з похибками комп’ютерних обчислень.

Міжгалузевий баланс виробництва та розподілу продукції є основним видом матричних моделей. Прикладом таких моделей є матрична модель міжгалузевого балансу (модель Леонтьєва), яка використовується для відображення системи матеріально-речових зв'язків процесу відтворення суспільного продукту.

Аналіз моделі Леонтьєва з математичного погляду пов'язаний з теорією невід'ємних матриць [95; 105; 107; 119].

Останніми роками велика увага приділяється розв’язанню СЛАР із символьними елементами [39; 100; 103]. Серед них значне місце займають

СЛАР у моделях Леонтьєва, СЛАР, які виникають в некласичних задачах для диференціальних рівнянь [52; 108], в динамічному програмуванні [73], в алгоритмах оптимізації електронних схем [28]. Досліджувати лінійні алгебричні системи, зокрема, доводиться в моделях Леонтьєва при розгляді системи диференціальних рівнянь:

$$X(t) = B(t)\left(E - A(t)\right)^{-1}\frac{dX(t)}{dt} + C(t), \tag{1.21}$$

де $X(t) = [x_j(t)]$ – вектор-стовпець обсягів виробництва;

$\dfrac{dX(t)}{dt} = \left[\dfrac{dx_j(t)}{dt}\right]$ – вектор-стовпець абсолютних приростів виробництва;

$C(t)$ – вектор-стовпець споживання (разом з невиробничим нагромадженням);

$A(t) = \left(a_{ij}(t)\right)$ - матриця коефіцієнтів прямих матеріальних витрат (на відміну від коефіцієнтів статичного міжгалузевого балансу, коефіцієнти в динамічній моделі включають також витрати на відшкодування вибуття і капітальний ремонт основних виробничих фондів) ($i, j \in J$, $J = \{1, \ldots, n\}$);

$B(t) = \left(b(t)\right)$ - матриця коефіцієнтів капіталомісткості приростів виробництва (витрати виробничого нагромадження на одиницю приросту відповідних видів продукції) ($i, j \in J$, $J = \{1, \ldots, n\}$);

$B(t)\left(E - A(t)\right)^{-1}$ - матриця коефіцієнтів повного приросту капіталомісткості, тобто повних витрат виробничого нагромадження на одиничні прирости елементів використовуваного національного доходу.

1.4. Системи лінійних алгебричних рівнянь з λ – матрицями

Одним з важливих розділів систем комп'ютерної алгебри є процедури для операцій з поліномами. У цьому розділі зупинимось на числових методах розв'язання систем лінійних алгебричних рівнянь з поліноміальними елементами. Нехай задано матрицю $A(\lambda)$ розміру $n \times n$, елементами якої є многочлени від λ.

Припустимо, що значення для λ і коефіцієнтів многочленів з деякого поля $F^{nr\, n}$ беруть так, що елементи матриці A обчислюються для деякого часткового значення λ, наприклад $\lambda = \lambda_0 \in F^{n\times n}\cdot$ Будемо розглядати системи рівнянь вигляду:

$$A(\lambda)x(\lambda) = b(\lambda), \tag{1.22}$$

де $A(\lambda)$ – регулярна матриця порядку *n.* Права частина (1.22) $b(\lambda)-$ це вектор $\left(a_{1,n+1}(\lambda),\ \ a_{2,n+1}(\lambda),\ \ \ldots\ \ a_{m,n+1}(\lambda)\right)^T$. Матриця $A(\lambda)$ і вектор $b(\lambda)$ мають степінь l, тобто

$$a_{i,j}(\lambda) = \sum_{k=1}^{l} a_{i,j,k}\lambda^k, \tag{1.23}$$

де $i = 1,2,\ldots n$; $j = 1,2,\ldots n + 1$. Причому, повинен існувати хоча б один елемент, для якого $a_{i,j}$ №0.

Подібні задачі зустрічаються в алгоритмах оптимізації електронних схем [28; 43; 50; 88; 89], при вивченні перехідних процесів в схемотехніці [103].

У динамічному програмуванні [36] вивчається задача квадратичного програмування. Шукають

$$K^* = \inf K_0(x), x \in M \subseteq E^n$$

при обмеженнях

$$K_i(x) \le 0,\ \ i \in I;\ \ K_i(x) = 0,\ \ j \in J;\quad I \bigcap J = \varnothing,$$

де $K_\nu(x) = \left(A_\nu(x,x) + \left(l_\lambda, x\right) + c_\nu -\right.$квадратичні функції, визначені на $n-$вимірному евклідовому просторі E^n;

$A_\lambda -$ симетричні матриці розмірності $n \times n$*;*

$l_\lambda - n -$ вимірні вектори;

$c_\lambda -$ числа;

Для оцінки знизу K^*будують функцію

$$\varphi(\lambda) = -\frac{1}{4}\left(x(\lambda), l(\lambda)\right) + c(\lambda);$$

для якої $x(\lambda)$ знаходиться з лінійної системи рівнянь:

$$2A(\lambda)x(\lambda) + l(\lambda) = 0.$$

Спільне застосування структурного методу та методу малого параметра приводить до розв’язування нестаціонарних задач з внутрішньою нелінійністю

$$\rho(u)\frac{\partial u}{\partial t} = div\left[\lambda(u) grad(u)\right] + f$$

за крайових умов

$$\left[\lambda(u)\frac{\partial u}{\partial t} + a(u - u_0)\right]\Big/_{\Gamma} = 0;$$

$$u\Big/_{t=0} = u = const.$$

У даному випадку потрібно обчислити $det\ A(\lambda)$ з елементами поліномами λ [20].

У системах масового обслуговування з очікуванням з вхідним ерлангівським потоком [114] використовуються системи з $\lambda-$матрицями

$$\begin{pmatrix} s-\lambda & 0 & \ldots & \lambda\hat{H}(s) \\ \lambda & s-\lambda & & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & s-\lambda \end{pmatrix},$$

де $\hat{H}(s)$ - функція розподілу.

Обчислення результатів [105] розв'язання нелінійних систем алгебричних рівнянь методом матричної лінеаризації [107] теж приводить до систем вигляду (1.21). Необхідність розв'язання систем з $\lambda-$матрицями або обчислення визначників зустрічається також в некласичних задачах для диференціальних рівнянь [27; 108], в будівельній механіці [16].

С.А. Абрамов вивчав задачу, близьку до (1.22). Досліджувались розв'язки лінійних рекурентних рівнянь

$$\sum_{n=0}^{n} a_n(x)F(x+n) = b(x)$$

з поліноміальними елементами $a_n(x), b(x) \in K[x]$.

В.М. Кублановська для пучка

$$D(l) = l^n A_0 + l^{n-1} A_1 + \ldots + A_n,$$

утвореного матрицями $A_1, A_2, \ldots, A_n$ розміру $m \times n$, у циклі робіт [45] вивчала спектральну задачу:

$$D(\lambda)U = 0.$$

Це рівняння є частковим випадком (1.22).

Зупинимося тепер безпосередньо на розв’язуванні систем виду (1.22). Під точним розв'язком системи (1.22) (якщо він існує) прийнято розуміти відношення двох поліномів з невідомими коефіцієнтами:

$$x(\lambda) = \sum_{j=0}^{nl} \lambda^j X_{nl-j} \Big/ \sum_{j=0}^{nl} \lambda^j y_{nl-j} \,. \tag{1.24}$$

Тут $X_1, X_2, \ldots, X_{nl}$ - вектори розмірності n, а $y_1, y_2, \ldots, y_{nl}$ - скаляри.

За деяких додаткових умов відношення поліномів може бути представлене нескінченним степеневим рядом:

$$x(\lambda) = \sum_{k=0}^{\infty} \lambda^k w_k, \tag{1.25}$$

де w_k для всіх $k \geq 0$ є векторами з коефіцієнтами з F^n. Подібний розв'язок шукається в багатьох роботах, присвячених системам виду (1.22). Зокрема це стосується роботи [62].

Найпростішим способом визначення невідомих, як і у випадку числових систем, є застосування формули Крамера з наступним розкриттям визначників за мінорами. Тоді розв'язки $x(\lambda)$ одержуються у вигляді відношення поліномів порядку ln в результаті $O\left(l^n n!\right)$ арифметичних дій.

Можна також спробувати застосувати для розв'язання системи (1.22) формальний алгоритм, подібний до методу виключення для числових систем:

$$\left.\begin{aligned}
&a_{ij}^{(s)}(\lambda) = a_{ij}^{(s-1)}(\lambda)a_{kk}^{(s-1)}(\lambda) - a_{ik}^{(s-1)}(\lambda)a_{kj}^{(s-1)}(\lambda);\\
&\text{де}\quad i = k+1,\ k+2,\dots,n;\ \ j = k+1, k+2,\dots,n+1;\\
&x_n(\lambda) = \frac{a_{nn+1}^{(n)}(\lambda)}{a_{nn}^{(n)}(\lambda)};\\
&x_{n-1}(\lambda) = \frac{a_{n-1,n+1}^{(n-1)}(\lambda)a_{nn}^{(n)}(\lambda) - a_{nn+1}^{(n)}(\lambda)a_{n-1n}^{(n)}(\lambda)}{a_{nn}^{(n)}(\lambda)}.
\end{aligned}\right\}$$

Ця схема дозволяє уникати виконання операцій ділення многочленів, хоча ділення є невід'ємним атрибутом для більшості числових методів. Однак при цьому підході спостерігається явище швидкого росту степеня поліноміальних елементів матриць $A^{(i)}(\lambda)$ – так зване “розбухання” проміжних даних. Внаслідок цього число арифметичних операцій досягає $O(l^{2^n})$.

У монографії П. Ланкастера [48] описаний алгоритм зведення поліноміальних матриць до діагонального виду, який, природно, можна застосувати й для розв'язання систем рівнянь. Але складність такого методу з точністю до головного члена – $O(2^n n l^2)$.

Перший алгоритм з поліноміальною швидкістю $O(l^2 n^2)$ реалізації на ЕОМ, правдоподібно, належить Е. Барейсу (E.H. Bareiss) [112]. Дуже суттєве скорочення кількості операцій, порівняно із зазначеним підходом, в запропонованій Барейсом схемі, досягається за рахунок виділення спільного множника мінорів різного порядку із наступним його скороченням.

Незалежно від Барейса, Р.В. Дмитришин емпіричним шляхом для конкретних систем вигляду (1.22), що виникли в задачах синтезу електронних схем, знайшов узагальнення для прямого ходу методу виключення [28] з такою ж кількістю операцій.

У 1973 році М. Макклелан (M.T. Macclelian) [130] запропонував прямий модулярний алгоритм для роз'язання систем з поліноміальними елементами. Його складність $O(ln^4+l^2n^3)$ операцій над F. Теоретичне узагальнення цього підходу та обчислювальні експерименти опубліковані Макклеланом в роботі [131].

Модулярні алгоритми також ґрунтовно досліджували Дж.Д. Ліпсон (J.D.Lipson) [128], Т. Магнанті, Дж. Орлін (T. Magnanti, J. Orlin) [132].

Алгоритм Е. Барейса в 1987 р. був узагальнений Г.І.Малашонком [54] для комутативних кілець і названий методом гоморфних образів. Для реалізації алгоритму потрібно $1/6\left(2n^3 + 3n^2 - 5n\right)$ операцій додавання та $1/2\left(n^3 + 2n^2 - n\right) - 1$ множень у деякому кільці. Однак, на жаль, автор розробив свою схему лише для кільця цілих чисел або в його розширенні □ .

Досить скупий арсенал методів, присвячених проблемі розв'язання розріджених систем (1.22). Один з них використовує правило Крамера та розклад Лапласа по мінорах. Основна стратегія при цьому у випадку розріджених матриць полягає у повторному використанні проміжних результатів обчислень у наступних розрахунках.

Інший шлях – скористатись гаусівським виключенням (або його варіаціями, описаними для щільних матриць). Можна вибирати рядки для виключення (або стовпці) відповідно до числа ненульових елементів, які вони містять. Таке “раціональне виключення” написати досить легко, але в загальному випадку матриця стає все менш і менш розрідженою, і час обчислень залишається майже таким, як і для щільних матриць, – пропорційний n^3. Цей алгоритм використовували Д. Копперсміт і Д. Давенпорт (D. Coppersmith і D. Davenport [120]) для розв'язання системи 1061 порядку, і при цьому спостерігалося суттєве розбухання інформації в проміжних результатах.

Зовсім недавно низка авторів почала застосовувати в комп'ютерній алгебрі такі ітераційні методи, як метод Ланцьоша [105] і метод спряжених градієнтів [107]. Однак їхня швидкість нижча, ніж для методів виключення.

Існує також декілька способів розв'язання систем з поліноміальними елементами. Їх можна віднести до наблежених методів. Сфера застосування цих алгоритмів суттєво вужча, ніж у схемах описаних раніше, але зате вони потребують значно меншого об'єму обчислювальної роботи.

Перший з них – наближений алгоритм інтерполяційного типу, описаний В.М. Фаддєєвою та Д.К. Фаддєєвим [100]. У цьому алгоритму ідея полягає у відшуканні розв'язків (1.22) у вигляді поліномів порядку ln . Для визначення невідомих коефіцієнтів шуканих многочленів спочатку необхідно визначити значення $x(\lambda)$ в ln точках. Для цього потрібно виконати, з точністю до головного члена, $O(ln^4 + l^2n^3)$ арифметичних дій.

Результати Макклелана, Р. Моенк та Дж. Картер (R.T.Moenck and J.H. Carter) [133] у 1979 році запропонували шукати розв'язок у вигляді відрізка ряду (1.25). Після знаходження коефіцієнтів відрізка ряду за допомогою перетворення Паде [5] розв'язок записується в раціональній формі виду (1.23). Подібний підхід дає можливість скоротити необхідну кількість операцій до $O\left(l^2n^3\right)$.

Розвинуто цей підхід в роботі Ст. Кабаї та Б. Домбзі (S. Cabay, B. Dombzy) [114]. Тут проведено глибоке дослідження співвідношення між відрізком ряду (1.25) та перетворенням Паде. Наведено алгоритми програм та результати обчислювальних експериментів на ЕОМ.

Т.Сасакі і Н. Марао (T. Sasaki, H. Murao) [138] запропонували наближений метод, в якому до кільця додають декілька змінних і зберігають лише декілька членів від цих змінних.

Крім розв'язання систем лінійних рівнянь (1.22) у ряді досліджень вивчались суміжні питання Х. Занка та Д. Луі (X. Zhang D. Lui) [146]. Запропоновано ефективний алгоритм для обернення матриці Ганкеля від $Z\left(x_1,x_2,\ldots,x_n\right)$. Складність алгоритму $O\left[\left(n^{n+3}Gn+n^{n+2}G^{n+1}\right)\log_2 n\log_2^2 L\right]$, де G та L– відповідно максимальна сума порядків змінних та норма вхідного представлення.

Більшість з методів розв’язування систем $A(\lambda)Z(\lambda)=B(\lambda)$, з поліноміальними елементами від однієї змінної непридатні для багатовимірного випадку (1.22). Передусім це пов’язано з тим, що відповідний допоміжний апарат не розрахований на випадок функцій від кількох змінних.

На розробку ефективних методів розв’язання задач оптимізації спрямовані зусилля потужних наукових шкіл, створених І.В.Сергієнком [88; 89], Н.З. Шором і С.І. Стеценком [108]. Системи лінійних алгебричних рівнянь з $\lambda-$матрицями можуть бути успішно застосовані для розв’язання оптимізаційних задач.

У роботі О.Л. Горбачук, Л.І. Комарницької, Ю.П. Матуріна [25] проаналізовано системи лінійних рівнянь, розглянуто елементарні перетворення СЛАР.

Цикл праць М.О. Недашковського [60-71] присвячено методам розв’язання СЛАР з λ-матрицями. Зокрема побудовані обчислювальні схеми – аналоги неунітарних алгоритмів лінійної алгебри на випадок алгебри поліномів [71], схеми виконання обчислень в багатомодульній системі лишків [68], одержано спосіб зведення до систем з числовими коефіцієнтами спеціального виду [60]. М.О. Недашковським описані також методи розв’язування розріджених систем з поліноміальними коефіцієнтами і елементами-рядами [62].

Велика увага приділяється розв’язуванню СЛАР з символьними елементами [11; 13; 32; 102]. Серед них значне місце займають СЛАР з поліноміальними елементами, які виникають у некласичних задачах для диференціальних рівнянь [108], у динамічному програмуванні [43], в алгоритмах оптимізації електронних схем [53].

Як виявилося, більшість з методів розв'язання систем (1.22) з поліноміальними елементами від однієї змінної λ непридатні для розв'язання систем лінійних алгебричних рівнянь з m - вимірними $\lambda-$матрицями, які виникають під час застосування методів матричної лінеаризації для поліноміальних рівнянь, в апроксимаціях Паде [5]. Такі системи мають вигляд:

$$A\left(\lambda_1,\lambda_2,\dots,\lambda_m\right)X\left(\lambda_1,\lambda_2,\dots,\lambda_m\right)=B\left(\lambda_1,\lambda_2,\dots,\lambda_m\right), \qquad (1.26)$$

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

$$B(\lambda_1,...,\lambda_m) = \left(a_{1,n+1}(\lambda_1,...,\lambda_m)\right),\left(a_{2,n+1}(\lambda_1,...,\lambda_m)\right),...,\left(a_{n,n+1}(\lambda_1,...,\lambda_m)\right)^T$$

є вектор, елементами якого є многочлени степеня l від $\lambda_1,\lambda_2,...,\lambda_m$. Отже,

$$a_{ij}(\lambda_1,\lambda_2,...,\lambda_m) = \sum_{k_1+k_2+...+k_m=0}^{l} a_{ij(k_1k_2...k_m)}\lambda_1^{k_1}\lambda_2^{k_2}...\lambda_m^{k_m} \;, \left(i=\overline{1,n};\; j=\overline{1,n+1}\right).$$

Більшість з методів розв’язання систем $A(\lambda)Z(\lambda)=B(\lambda)$, з поліноміальними елементами від однієї змінної l непридатні для багатовимірного випадку (1.22). Насамперед це пов’язано з тим, що відповідний допоміжний апарат не розрахований на випадок функцій від кількох змінних. Дослідженням нових підходів, що дозволять отримати дієві методи розв’язання СЛАР з m-вимірними λ-матрицями займалась І. Босікова [8- 10].

Висновки до першого розділу

В даному розділі розглянуто теоретико-числові основи обчислювальних методів на основі теорії СЛАР з λ-матрицями, які модифікують прямі числові методи лінійної алгебри.

Проведено аналіз комп'ютерної системи для розв'язування задач із вхідними даними числового характеру. Запропоновано порівняльну характеристику розвитку декількох напрямків і концепцій для виконання символьних перетворень. Проаналізовано методи комп'ютерної алгебри для розв'язання СЛАР.

РОЗДІЛ 2. СИСТЕМИ ЛІНІЙНИХ АЛГЕБРИЧНИХ РІВНЯНЬ СПЕЦІАЛЬНОГО ВИДУ

2.1. Теоретико-числові основи матричних обчислень

Теоретико-числові результати відіграють важливу роль під час розв'язування систем лінійних алгебричних рівнянь. Матриці з числовими елементами є природним узагальненням чисел і широко використовуються в алгебрі, як приклади алгебричних структур. Матриці є одним з базових інструментів при моделюванні на ЕОМ. Елементи теорії матриць, теорії систем лінійних алгебричних рівнянь використовуються для розв'язання багатьох економічних задач. Здебільшого в розробці і використанні баз даних, де майже вся інформація зберігається і обробляється в матричній формі.

Ряд монографій присвячена теорії чисел СЛАР, серед яких праці В.Н. Фаддєєвої, Д.К. Фаддєєва [99; 100], І.М. Молчанова [58], Дж. Форсайта, М. Малькольма та К. Моулера [101], R.J. Goult, R.F. Hoskins, J.A. Milner та M.J.Pratt [123], F.John [124], S.G.Kellison [126], J.R.Westlake [143] та інших. Довідник з алгоритмів лінійної алгебри надрукували Дж. Уілкінсон і К. Райнш [98], А.Бородін і І.Мунро зробили огляд обчислювальної складності алгоритмів лінійної алгебри [113]. Питання складності також розглядаються в працях А. Ахо (A.V.Aho), Дж. Хопкрофта (J.E. Hopcroft), Дж. Ульмана (J.D. Ullman) [1], J.E.Savage [137], S. Winograd [144].

Під час числового розв'язування математичних і прикладних задач досить часто на якомусь етапі виникають похибки таких трьох типів [18, 31, 10]:

1. **Неусувна похибка.** Такого типу похибка пов'язана з наближеним характером початкової моделі (зокрема, з неможливістю урахувати всі чинники в процесі вивчення модельованого явища). Крім того, параметрами математичного опису моделі є наближені числа (наприклад, неможливо виконати абсолютно точні вимірювання). Для обчислювача таку похибку задачі вважають неусувною (безумовною). Неусувну похибку поділяють на два види:

а) похибка, яка є наслідком неточності задання числових даних, які входять у математичний опис задачі;

б) похибка, яка є наслідком невідповідності математичного опису задачі до реальності (цю частину похибки називають похибкою математичної моделі).

2. **Похибка методу.** Ця похибка пов'язана зі способом розв'язування сформульованої математичної задачі. Вона з'являється внаслідок заміни початкової математичної моделі іншою або скінченною послідовністю інших, наприклад лінійних моделей. У разі створення числових методів закладається можливість відстежування таких похибок і доведення їх до як завгодно малого значення. Тому похибка методу є усувною (умовною).

3. **Похибка заокруглень (похибка операцій**). Цей тип похибок зумовлений необхідністю виконувати арифметичні операції над числами, які зображаються в пам'яті комп'ютерів наближено залежно від кількості розрядів, що пов'язано з використовуваною обчислювальною технікою.

Усі три описані типи похибок у сумі становлять повну похибку результату розв'язування задачі. Оскільки перший тип похибок не залежить від обчислювача, то він слугує для нього лише орієнтиром точності, з якою треба розраховувати математичну модель. Нема потреби розв'язувати задачу точніше, ніж це зумовлене невизначеністю початкових даних. Отже, похибку методу підпорядковують похибці задачі. Нарешті, під час виведення оцінок похибок числових методів звичайно припускають, що всі операції над числами виконують точно. Це означає, що похибка заокруглень не повинна суттєво впливати на результати реалізації методів, тобто повинна підпорядковуватися похибці методу. Вплив похибок заокруглень потрібно враховувати як на стадії відбору й алгоритмізації числових методів, так і в разі вибору обчислювальних і програмних засобів, а також виконання окремих дій та обчислення значень функції.

2.2. Елементи теорії систем лінійних алгебричних рівнянь з l - матрицями

В останні роки поряд з традиційно великою увагою до розв'язання числових систем лінійних алгебричних рівнянь значно підвищився інтерес до систем з поліноміальними елементами.

Системи лінійних алгебричних рівнянь з поліноміальними елементами мають вигляд [61; 71]:

$$A(\lambda)Z(\lambda) = B(\lambda), \tag{2.1}$$

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

$$a_{ij}(\lambda) = \sum_{k=0}^{l} a_{ij}^{(k)} \lambda^{l-k}\ , \quad \left(i=\overline{1,n}\ ; j=\overline{1,n+1}\right). \tag{2.2}$$

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

Система (2.1) може мати єдиний розв'язок, більш ніж один розв'язок, або не мати розв'язків. Тому звідси випливають такі визначення.

Означення 2.1. Дві системи рівнянь називаються *рівносильними* (*еквівалентними*), якщо вони мають однакові множини розв'язків.

Означення 2.2. Система рівнянь називається *сумісною*, якщо вона має хоча б один розв'язок.

Означення 2.3. Система рівнянь називається *несумісною*, якщо вона не має жодного розв'язку, тобто множина розв'язків порожня множина Ø.

Систему (2.1) зручно записувати в матричному вигляді

$$A(\lambda)\vec{x}(\lambda) = \vec{B}(\lambda), \tag{2.3}$$

де

$$\vec{x}(\lambda)=\begin{pmatrix} x_1(\lambda) \\ x_2(\lambda \\ \cdots \\ x_n(\lambda \end{pmatrix}, \quad \vec{B}(\lambda)=\begin{pmatrix} b_1(\lambda) \\ b_2(\lambda \\ \cdots \\ b_n(\lambda \end{pmatrix}. \tag{2.4}$$

Форму (2.3) запису системи (2.1) називають *матричною*. Крім такої форми зручно використовувати запис у вигляді розширеної матриці:

$$\left(A(\lambda)\,|\,\vec{B}(\lambda)\right)=\left(\begin{array}{cccc|c} A_{11}(\lambda) & A_{12}(\lambda) & \cdots & A_{1n}(\lambda) & B_1(\lambda) \\ A_{21}(\lambda) & A_{22}(\lambda) & \cdots & A_{2n}(\lambda) & B_2(\lambda) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ A_{n1}(\lambda) & A_{n2}(\lambda) & \cdots & A_{nn}(\lambda) & B_n(\lambda) \end{array}\right), \tag{2.5}$$

Розглянемо деякі важливі твердження про розв’язки систем двох лінійних рівнянь з двома невідомими.

ЛЕМА 2.1. *Якщо матриця $A(\lambda)$, елементами якої є многочлени від λ, неособлива, то система (2.1) має єдиний розв’язок.*

ДОВЕДЕННЯ. Оскільки матриця $A(\lambda)$ неособлива, то для неї існує обернена матриця $A^{-1}(\lambda)$. Звідси випливає, що вектор $\vec{C}(\lambda)=A^{-1}(\lambda)\,\vec{B}(\lambda)$– розв’язок системи (1.3). Справді,

$$A(\lambda)\vec{C}(\lambda)=A(\lambda)\left(A^{-1}(\lambda)\vec{B}(\lambda)\right)=\left(A(\lambda)A^{-1}(\lambda)\right)\vec{B}(\lambda)=E\vec{B}(\lambda)=\vec{B}(\lambda).$$

Покажемо, що це єдиний розв’язок системи (2.1). Справді, нехай $\vec{d}(\lambda)$ – якийсь розв'язок системи (2.1). Тоді $A(\lambda)\vec{d}(\lambda)=\vec{B}(\lambda)$, та $A(\lambda)^{-1}(\lambda)\left(A(\lambda)\vec{d}(\lambda)\right)=A^{-1}(\lambda)\vec{B}(\lambda)$, тобто $\vec{d}(\lambda)=A^{-1}(\lambda)\,\vec{B}(\lambda)$. Це показує, що $\vec{d}(\lambda)=\vec{C}(\lambda)$. Лему доведено.

ЛЕМА 2.2. *Якщо матриця $A(\lambda)$ особлива, то система $A(\lambda)\vec{x}(\lambda)=\vec{o}(\lambda)$ має ненульові розв’язки.*

ДОВЕДЕННЯ. 1). Нехай $A(\lambda)$ – нуль-матриця, тоді $A \times \begin{pmatrix} 1 \\ 0 \\ \cdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \cdots \\ 0 \end{pmatrix}$.

Отже, $\begin{pmatrix}1\\0\\\ldots\\0\end{pmatrix}$ - ненульовий розв'язок системи $A(\lambda)\vec{x}(\lambda)=\vec{o}(\lambda)$.

2). Припустимо тепер, що $A(\lambda)$ – не нуль-матриця. Тому не всі $a_{ij}(\lambda)$ дорівнюють нулю. Зрозуміло, що

$$A\times\begin{pmatrix}a_{nn}(\lambda)\\-a_{nn-1}(\lambda)\\\cdots\\-a_{n1}(\lambda)\end{pmatrix}=\begin{pmatrix}a_{11}(\lambda)&a_{21}(\lambda)&\cdots&a_{n1}(\lambda)\\a_{21}(\lambda)&a_{22}(\lambda)&\cdots&a_{2n}(\lambda)\\\cdots&\cdots&\cdots&\cdots\\a_{n1}(\lambda)&a_{n2}(\lambda)&\cdots&a_{nn}(\lambda)\end{pmatrix}\begin{pmatrix}a_{nn}(\lambda)\\-a_{nn-1}(\lambda)\\\cdots\\-a_{n1}(\lambda)\end{pmatrix}=\begin{pmatrix}\det(A(\lambda))\\0\\\cdots\\0\end{pmatrix}=\begin{pmatrix}0\\0\\\cdots\\0\end{pmatrix}$$

…………………………………………………………………………………………

$$A\times\begin{pmatrix}-a_{1n}(\lambda)\\-a_{1n-1}(\lambda)\\\cdots\\a_{11}(\lambda)\end{pmatrix}=\begin{pmatrix}a_{11}(\lambda)&a_{21}(\lambda)&\cdots&a_{n1}(\lambda)\\a_{21}(\lambda)&a_{22}(\lambda)&\cdots&a_{2n}(\lambda)\\\cdots&\cdots&\cdots&\cdots\\a_{n1}(\lambda)&a_{n2}(\lambda)&\cdots&a_{nn}(\lambda)\end{pmatrix}\begin{pmatrix}-a_{1n}(\lambda)\\-a_{1n-1}(\lambda)\\\cdots\\a_{11}(\lambda)\end{pmatrix}=\begin{pmatrix}0\\0\\\cdots\\\det(A(\lambda))\end{pmatrix}=\begin{pmatrix}0\\0\\\cdots\\0\end{pmatrix}.$$

Оскільки не всі $a_{ij}(\lambda)$ дорівнюють нулю, то з останніх двох рівностей випливає, що система $A(\lambda)\vec{x}(\lambda)=\vec{o}(\lambda)$ має ненульові розв’язки. Лему доведено.

ТЕОРЕМА 2.1. *Система (2.1) має єдиний розв’язок тоді і тільки тоді, коли матриця цієї системи неособлива.*

ДОВЕДЕННЯ. З леми 2.1 випливає, що якщо $A(\lambda)$ неособлива, то система (2.1) має єдиний розв’язок. Розглянемо протилежний випадок. Нехай система (2.1) має єдиний розв’язок $\vec{c}$. Покажемо, що тоді система $A(\lambda)\vec{x}(\lambda)=\vec{o}(\lambda)$ має теж єдиний розв’язок (нуль-вектор). Справді, припустимо, що ненульовий вектор $\vec{p}(\lambda)=\begin{pmatrix}p_1(\lambda)\\p_2(\lambda)\\\cdots\\p_n(\lambda)\end{pmatrix}$ – розв’язок $A(\lambda)\vec{x}(\lambda)=\vec{o}(\lambda)$. Тоді $\vec{c}(\lambda)+\vec{p}(\lambda)$ – розв’язок системи (2.1). Справді,

$$A(\lambda)\left(\vec{c}(\lambda)+\vec{p}(\lambda)\right)=A(\lambda)\vec{c}(\lambda)+A(\lambda)\vec{p}(\lambda)=\vec{b}(\lambda)+\vec{o}(\lambda)=\vec{b}(\lambda).$$

Зрозуміло, що $\vec{c}(\lambda)+\vec{p}(\lambda)\neq\vec{c}(\lambda)$, бо $\vec{p}(\lambda)\neq\vec{o}(\lambda)$.

Тобто, система (2.1) має два різні розв'язки і ми прийшли до суперечності.

<u>*Наслідок*</u>. Система $A(\lambda)\vec{x}(\lambda)=\vec{o}(\lambda)$, (2.6)

де $A(\lambda)=\begin{pmatrix} a_{11}(\lambda) & a_{21}(\lambda) & \cdots & a_{n1}(\lambda) \\ a_{21}(\lambda) & a_{22}(\lambda) & \cdots & a_{2n}(\lambda) \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1}(\lambda) & a_{n2}(\lambda) & \cdots & a_{nn}(\lambda) \end{pmatrix}$, $\vec{x}(\lambda)=\begin{pmatrix} x_1(\lambda) \\ x_2(\lambda) \\ \ldots \\ x_n(\lambda) \end{pmatrix}$, $\vec{o}(\lambda)=\begin{pmatrix} 0(\lambda) \\ 0(\lambda) \\ \ldots \\ 0(\lambda) \end{pmatrix}$, має лише нульовий розв'язок тоді і тільки тоді, коли $\det\left(A(\lambda)\right)\neq 0$.

Нехай маємо систему з k лінійних рівнянь з n невідомими $x_1(\lambda), x_2(\lambda),\ldots,x_n(\lambda)$:

$$\begin{cases} a_{11}(\lambda)x_1(\lambda)+a_{12}(\lambda)x_2(\lambda)+\ldots+a_{1n}(\lambda)x_n(\lambda)=b_1; \\ a_{21}(\lambda)x_1(\lambda)+a_{22}(\lambda)x_2(\lambda)+\ldots+a_{2n}(\lambda)x_n(\lambda)=b_2; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{k1}(\lambda)x_1(\lambda)+a_{k2}(\lambda)x_2(\lambda)+\ldots+a_{kn}(\lambda)x_n(\lambda)=b_k, \end{cases} \tag{2.7}$$

де $a_{ij}\left(i=1,2,\ldots,k;\ j=1,2,\ldots,n\right)$– деякі числа. Зокрема, можливо, що $k=1$.

Означення 2.4. Розв'язком системи (2.7) називається будь-яка впорядкована сукупність чисел $c_1, c_2, \ldots, c_n$, яка задовольняє систему, і означає наступне: якщо в кожне рівняння системи замість x_i підставити $c_i\left(i=1,2,\ldots,n\right)$, то одержимо k правильних числових рівностей.

Отже, розв'язок системи можна вважати вектором з n компонентами.

Введемо тепер позначення:

$$A=\begin{pmatrix} a_{11}(\lambda) & a_{12}(\lambda) & \ldots & a_{1n}(\lambda) \\ a_{21}(\lambda) & a_{22}(\lambda) & \ldots & a_{2n}(\lambda) \\ \ldots\ldots & \ldots\ldots & \ldots & \ldots \\ a_{k1}(\lambda) & a_{k2}(\lambda) & \ldots & a_{kn}(\lambda) \end{pmatrix},\quad \vec{x}=\begin{pmatrix} x_1(\lambda) \\ x_2(\lambda) \\ \ldots \\ x_n(\lambda) \end{pmatrix},\quad \vec{b}=\begin{pmatrix} b_1(\lambda) \\ b_2(\lambda) \\ \ldots \\ b_k(\lambda) \end{pmatrix}.$$

Тоді систему (2.7) можна записати у матричному вигляді:

$$A(\lambda)\vec{x}(\lambda)=\vec{b}(\lambda). \tag{2.8}$$

Матриця $A=\left[a_{ij}(\lambda)\right]$ називається матрицею системи (2.7). Приєднавши до матриці A стовпець вільних членів $\vec{b}$, одержимо так звану розширену матрицю системи:

$$\left(A\,\middle|\,\vec{b}\right)=\left(\begin{array}{cccc|c} a_{11}(\lambda) & a_{12}(\lambda) & \dots & a_{1n}(\lambda) & b_1(\lambda) \\ a_{21}(\lambda) & a_{22}(\lambda) & \dots & a_{2n}(\lambda) & b_2(\lambda) \\ \dots & \dots & \dots & \dots & \dots \\ a_{k1}(\lambda) & a_{k2}(\lambda) & \dots & a_{kn}(\lambda) & b_k(\lambda) \end{array}\right). \tag{2.9}$$

Зрозуміло, що система лінійних рівнянь однозначно визначається своєю розширеною матрицею.

Згідно з означення рівносильних систем відомо, що несумісні системи рівносильні.

Зауважимо, що іноді систему лінійних рівнянь (2.7) зручно записувати у векторній формі:

$$\vec{a}_1(\lambda)x_1(\lambda)+\vec{a}_2(\lambda)x_2(\lambda)+\ldots+\vec{a}_n(\lambda)x_n(\lambda)=\vec{b}(\lambda), \tag{2.10}$$

де $\vec{a}_1=\begin{pmatrix} a_{11}(\lambda) \\ a_{21}(\lambda) \\ \cdots \\ a_{k1}(\lambda) \end{pmatrix}$, $\vec{a}_2=\begin{pmatrix} a_{12}(\lambda) \\ a_{22}(\lambda) \\ \cdots \\ a_{k2}(\lambda) \end{pmatrix}$, $\ldots$ $\vec{a}_n=\begin{pmatrix} a_{1n}(\lambda) \\ a_{2n}(\lambda) \\ \cdots \\ a_{kn}(\lambda) \end{pmatrix}$ – стовпці матриці A.

2.3. Елементарні перетворення систем лінійних рівнянь з $\lambda-$матрицями

Нехай маємо два лінійні алгебричні рівняння з $x_1(\lambda), x_2(\lambda),\ldots x_n(\lambda)$ невідомими:

$$\eta_1 x_1(\lambda)+\eta_2 x_2(\lambda)+\ldots+\eta_n x_n(\lambda)=p$$

і

$$\mu_1 x_1(\lambda)+\mu_2 x_2(\lambda)+\ldots+\mu_n x_n(\lambda)=q\,.$$

Означення 2.5. *Сумою* лінійних рівнянь $\eta_1 x_1(\lambda)+\eta_2 x_2(\lambda)+\ldots+\eta_n x_n(\lambda)=p$ і $\mu_1 x_1(\lambda)+\mu_2 x_2(\lambda)+\ldots+\mu_n x_n(\lambda)=q$ називається лінійне рівняння

$$(\eta_1+\mu_1)x_1+(\eta_2+\mu_2)x_2+\ldots+(\eta_n+\mu_n)x_n=p+q.$$

Означення 2.6. *Добутком* лінійних рівнянь $\eta_1 x_1(\lambda)+\ \ldots+\eta_n x_n(\lambda)=p$ на число μ називається лінійне рівняння

$$(\mu\eta_1)x_1(\lambda)+(\mu\eta_2)x_2(\lambda)+\ldots+(\mu\eta_n)x_n(\lambda)=\mu p\,.$$

Нехай тепер маємо дві системи з однаковим числом невідомих, друга система утворена з першої за допомогою елементарного перетворення. Другу систему можна одержати з першої шляхом перестановки місцями в першій системі її i-го та j-го рівнянь $(i\neq j)$. Таке перетворення записують так:

$$1)\begin{cases}\eta_{11}x_1(\lambda)+\eta_{12}x_2(\lambda)+...+\eta_{1n}x_n(\lambda)=p_1;\\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ \eta_{i1}x_1(\lambda)+\eta_{i2}x_2(\lambda)+...+\eta_{in}x_n(\lambda)=p_i;\\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ \eta_{j1}x_1(\lambda)+\eta_{j2}x_2(\lambda)+...+\eta_{jn}x_n(\lambda)=p_j;\\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\end{cases}\to\begin{cases}\eta_{11}x_1(\lambda)+\eta_{12}x_2(\lambda)+...+\eta_{1n}x_n(\lambda)=p_1;\\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ \eta_{j1}x_1(\lambda)+\eta_{j2}x_2(\lambda)+...+\eta_{jn}x_n(\lambda)=p_j;\\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ \eta_{i1}x_1(\lambda)+\eta_{i2}x_2(\lambda)+...+\eta_{in}x_n(\lambda)=p_i;\\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\end{cases}$$

Кажуть, що друга система утворена з першої за допомогою елементарного перетворення, якщо другу систему можна одержати з першої заміною i-го рівняння першої системи сумою її i-го рівняння та j-го рівняння, помноженого на деяке число λ $(i\neq j)$. Таке перетворення записують у вигляді:

$$2)\begin{cases}\eta_{11}x_1(\lambda)+\eta_{12}x_2(\lambda)+...+\eta_{1n}x_n(\lambda)=p_1;\\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ \eta_{i1}x_1(\lambda)+\eta_{i2}x_2(\lambda)+...+\eta_{in}x_n(\lambda)=p_i;\\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ \eta_{j1}x_1(\lambda)+\eta_{j2}x_2(\lambda)+...+\eta_{jn}x_n(\lambda)=p_j;\\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\end{cases}\to$$

$$\begin{cases} \eta_{11}x_1(\lambda)+\eta_{12}x_2(\lambda)+\ldots+\eta_{1n}x_n(\lambda)=p_1; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \mu\left(\eta_{i1}+\eta_{j1}\right)x_1(\lambda)+\mu\left(\eta_{i2}+\eta_{j2}\right)x_2(\lambda)+\ldots+\mu\left(\eta_{in}+\eta_{jn}\right)x_n(\lambda)=\mu\left(p_i+p_j\right); \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots . \end{cases}$$

ТЕОРЕМА 2.2. *Елементарні перетворення переводять систему лінійних рівнянь в рівносильну їй систему.*

Попередньо розглянемо лему.

ЛЕМА 2.3. *Якщо друга система лінійних рівнянь одержана з першої за допомогою елементарних перетворень, то кожен розв'язок першої системи є розв'язком другої.*

ДОВЕДЕННЯ. У випадку елементарного перетворення типу 1) твердження леми очевидне. Нехай перша система лінійних рівнянь має вигляд:

$$\begin{cases} \eta_{11}x_1(\lambda)+\eta_{12}x_2(\lambda)+\ldots+\mu_{1n}x_n(\lambda)=b_1; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \eta_{i1}x_1(\lambda)+\eta_{i2}x_2(\lambda)+\ldots+\eta_{in}x_n(\lambda)=b_i; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \eta_{j1}x_1(\lambda)+\eta_{j2}x_2(\lambda)+\ldots+\eta_{jn}x_n(\lambda)=b_j; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \eta_{k1}x_1(\lambda)+\eta_{k2}x_2(\lambda)+\ldots+\eta_{kn}x_n(\lambda)=b_k, \end{cases} \tag{2.11}$$

де $i \neq j$, а другу систему отримаємо з першої за допомогою елементарного перетворення 2), де $\lambda -$ деяке число.

Нехай також $\vec{c} = \begin{pmatrix} c_1 \\ c_2 \\ \ldots \\ c_n \end{pmatrix}$ – розв'язок першої системи, тоді є правильними такі числові рівності:

$$
\begin{array}{l}
h_{11}c_1 + h_{12}c_2 + \ldots + h_{1n}c_n = b_1; \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
h_{i1}c_1 + h_{i2}c_2 + \ldots + h_{in}c_n = b_2; \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
h_{j1}c_1 + h_{j2}c_2 + \ldots + h_{jn}c_n = b_j; \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
h_{k1}c_1 + h_{k2}c_2 + \ldots + h_{kn}c_n = b_k.
\end{array}
\tag{2.12}
$$

Помножимо почленно j-ту рівність з (2.12) на μ і одержимо правильну числову рівність

$$\mu h_{j1}c_1 + \mu h_{j2}c_2 + \ldots + \mu h_{jn}c_n = \mu b_j.$$

Додавши тепер почленно до i-тої рівності з (2.12) останню рівність, одержимо правильну числову рівність

$$h_{i1}c_1 + h_{i2}c_2 + \ldots + h_{in}c_n + \mu h_{j1}c_1 + \mu h_{j2}c_2 + \ldots + \mu h_{jn}c_n = b_i + \mu b_j,$$

тобто

$$\left(h_{i1} + \mu h_{j1}\right)c_1 + \left(h_{i2} + \mu h_{j2}\right)c_2 + \ldots + \left(h_{in} + \mu h_{jn}\right)c_n = b_i + \mu b_j. \tag{2.13}$$

Отже, правильні наступні числові рівності:

$$
\begin{array}{l}
h_{11}c_1 + h_{12}c_2 + \ldots + h_{1n}c_n = b_1; \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\left(h_{i1} + \mu h_{j1}\right)c_1 + \left(h_{i2} + \mu h_{j2}\right)c_2 + \ldots + \left(h_{in} + \mu h_{jn}\right)c_n = b_i + \mu b_j; \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
h_{j1}c_1 + h_{j2}c_2 + \ldots + h_{jn}c_n = b_j; \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
h_{k1}c_1 + h_{k2}c_2 + \ldots + h_{kn}c_n = b_k.
\end{array}
\tag{2.14}
$$

Всі рівності у (2.14) ті ж самі, що й в (2.12), лише i-та рівність замінена рівністю (2.13). Отже, $\vec{c}$ також розв'язок тієї системи, яка одержана із системи (2.11) за допомогою елементарного перетворення 2). Лему доведено.

ДОВЕДЕННЯ ТЕОРЕМИ. Нехай друга система лінійних рівнянь одержується з першої за допомогою елементарного перетворення, тоді і перша система одержується з другої за допомогою елементарного перетворення (див. зауваження після означення елементарного перетворення 2). Тоді за лемою кожен розв'язок першої системи є розв'язком другої системи, і навпаки, кожен розв'язок другої системи є розв'язком першої системи. Теорему доведено.

2.4. Східчасті системи

Означення 2.7. *Східчастою* називається система лінійних рівнянь, матриця якої є східчастою, тобто це система, яка має такий вигляд:

$$\left|\begin{matrix} 0 & \ldots & 0 & a_{1k_1} & \ldots & a_{1n} \\ 0 & \ldots & 0 & a_{2k_2} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & a_{rk_r} & \ldots & a_{rn} \\ 0 & \ldots & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & \ldots & \ldots & 0 \end{matrix}\right|$$

або

$$\begin{cases} 0\cdot x_1(\lambda)+\ldots+0\cdot x_{k_1-1}(\lambda)+a_{1k_1}x_{k_1}(\lambda)+\ldots+a_{1n}x_n(\lambda)=b_1; \\ 0\cdot x_1(\lambda)+\ldots+0\cdot x_{k_2-1}(\lambda)+a_{2k_2}x_{k_2}(\lambda)+\ldots+a_{2n}x_n(\lambda)=b_2; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ 0\cdot x_1(\lambda)+\ldots+0\cdot x_{k_r-1}(\lambda)+a_{rk_r}x_{k_r}(\lambda)+\ldots+a_{rn}x_n(\lambda)=b_r; \\ 0\cdot x_1(\lambda)+\ldots+0\cdot x_{k_r-1}(\lambda)+0\cdot x_{k_r}(\lambda)+\ldots+0\cdot x_n(\lambda)=b_{r+1}; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ 0\cdot x_1(\lambda)+\ldots+0\cdot x_m(\lambda)+\ldots+0\cdot x_n(\lambda)=b_m, \end{cases} \tag{2.15}$$

де $a_{1k_1}\neq 0,\ a_{2k_2}\neq 0,\ldots,a_{rk_r}\neq 0,\ 1\leq k_1<k_2<\ldots<k_r\leq n,$ або система вигляду

$$\begin{cases} 0\cdot x_1(\lambda)+\ldots+0\cdot x_n(\lambda)=b_1; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ 0\cdot x_1(\lambda)+\ldots+0\cdot x_n(\lambda)=b_m. \end{cases} \tag{2.16}$$

Отже, система називається *східчастою*, якщо у розширеній її матриці $\left(A\,\middle|\,\vec{b}\right)$ матриця A східчаста.

Приклади східчастих систем

1). $x_1 + x_2 + 2x_3 = 1$ – східчаста система, що складається з одного рівняння (тут $k_1 = 1,\ a_{1k_1} \neq 0,\ r = 1$).

2). $\begin{cases} 0 \cdot x_1 + 2x_2 + x_3 - 3x_4 + x_5 = 1; \\ 0 \cdot x_1 + 0 \cdot x_2 + 2x_3 + x_4 - x_5 = 2; \\ 0 \cdot x_1 + 0 \cdot x_2 + 0 \cdot x_3 + 0 \cdot x_4 + 0 \cdot x_5 = -1 \end{cases}$ – східчаста система (тут $k_1 = 2,\ k_2 = 3,\ r = 2$).

3). $\begin{cases} 0 \cdot x_1 + 0 \cdot x_2 = 0; \\ 0 \cdot x_1 + 0 \cdot x_2 = 1 \end{cases}$ – східчаста система (тут $r = 0$).

Системи виду (2.16) мають множину розв'язків, що складаються з усіх векторів $\begin{pmatrix} c_1 \\ c_2 \\ \ldots \\ c_n \end{pmatrix}$, де $c_1, c_2, ..., c_n$ - довільні числа, якщо $b_1 = b_2 = \ldots = b_m = 0;$ системи виду (2.16) є несумісними, якщо хоча б одне з чисел $b_1, b_2, \ldots, b_m$ відмінне від 0.

Розглянемо системи виду (2.15). У такій системі невідомі $x_{k_1}, \ldots, x_{k_r}$ називаються *головними невідомими*, а всі інші невідомі – *вільними невідомими*.

Розглянемо два випадки:

I. Нехай серед чисел $b_{r+1}, \ldots, b_m$ хоча б одне відмінне від 0. Тоді система (2.15) містить рівняння $0 \cdot x_1 + 0 \cdot x_2 + \ldots + 0 \cdot x_n = b_i$, де $b_i \neq 0,\ r + 1 \leq i \leq m$.

Зрозуміло, що жодна сукупність чисел $c_1, c_2, \ldots, c_n$ не задовольняє рівняння. Отже, система (2.15) розв'язків немає. Тобто (2.15) – несумісна система.

II. Нехай $b_{r+1} = \ldots = b_m = 0.$ Для зручності подальших міркувань перепозначимо невідомі $x_1, x_2, \ldots, x_n$ буквами $y_1, y_2, \ldots, y_n$ так, щоб головні невідомі $x_{k_1}, x_{k_2}, \ldots, x_{k_r}$ були позначені відповідно через

$y_1, y_2, \ldots, y_r$ $\left(x_{k_1} = y_1, x_{k_2} = y_2, \ldots, x_{k_r} = y_r\right)$, а всі інші (тобто вільні невідомі) через $y_{r+1}, y_{r+2}, \ldots, y_n$.

Відповідно до цього елементи матриці A системи (2.15) будуть перепозначені на елементи матриці C, тобто система (2.15) матиме такий вигляд:

$$\begin{cases} c_{11}y_1(\lambda) + c_{12}y_2(\lambda) + \ldots + c_{1r}y_r(\lambda) + c_{1r+1}y_{r+1}(\lambda) + \ldots + c_{1n}y_n(\lambda) = b_1; \\ \qquad c_{22}y_2(\lambda) + \ldots + c_{2r}y_r(\lambda) + c_{2r+1}y_{r+1}(\lambda) + \ldots + c_{2n}y_n(\lambda) = b_2; \\ \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \qquad c_{r-1,r-1}y_{r-1}(\lambda) + c_{r-1,r}y_r(\lambda) + \ldots + c_{r-1n}y_n(\lambda) = b_{r-1}; \\ \qquad c_{rr}y_r(\lambda) + c_{r,r+1}y_{r+1}(\lambda) + \ldots + c_{rn}y_n(\lambda) = b_r, \end{cases} \tag{2.17}$$

де $c_{11} \neq 0, \quad c_{22} \neq 0, \ldots, c_{rr} \neq 0.$

(Рівняння системи (2.15), починаючи з $(r+1)$-го, мали вигляд $0\cdot x_1 + 0\cdot x_2 + \ldots + 0\cdot x_n = 0$, і тому їх опущено, оскільки при вилученні нуль-рівнянь (рівнянь виду $0\cdot x_1(\lambda) + 0\cdot x_2(\lambda) + \ldots + 0\cdot x_n(\lambda) = 0$) система переходить у рівносильну їй систему).

Надамо тепер вільним невідомим $y_{r+1}(\lambda), y_{r+2}(\lambda), \ldots, y_n(\lambda)$ будь-які конкретні числові значення $u_{r+1}(\lambda), u_{r+2}(\lambda), \ldots, u_n(\lambda)$. Тоді з останнього рівняння визначається $y_r(\lambda)$, а саме:

$$y_r(\lambda) = \left(-\frac{c_{r,r+1}}{c_{rr}}\right)u_{r+1}(\lambda) + \ldots + \left(-\frac{c_{rn}}{c_{rr}}\right)u_n(\lambda) + \left(\frac{b_r}{c_{rr}}\right). \tag{2.18}$$

Тоді з попереднього рівняння ($(r-1)$-го рівняння) можна визначити $y_{r-1}(\lambda)$ і т. д.

Якщо вже визначені $y_r(\lambda), y_{r-1}(\lambda), \ldots, y_{i+1}(\lambda)$, то з i-го рівняння $c_{ii}y_i(\lambda) + c_{i,i+1}y_{i+1}(\lambda) + \ldots + c_{i,r-1}y_{r-1}(\lambda) + c_{ir}y_r(\lambda) + c_{i,r+1}u_{r+1}(\lambda) + \ldots + c_{in}u_n(\lambda) = b_i$

$y_i(\lambda)$ визначається однозначно

$$y_i(\lambda)=\left(-\frac{c_{i,i+1}}{c_{ii}}\right)y_{i+1}(\lambda)+\ldots+\left(-\frac{c_{i,r-1}}{c_{ii}}\right)y_{r-1}(\lambda)+\left(-\frac{c_{ir}}{c_{ii}}\right)y_r(\lambda)+\left(-\frac{c_{i,r+1}}{c_{ii}}\right)u_{r+1}(\lambda)+$$
$$+\ldots+\left(-\frac{c_{in}}{c_{ii}}\right)u_n(\lambda)+\left(\frac{b_i}{c_{ii}}\right).$$

Отже, рухаючись вгору системою, можна визначити $y_r(\lambda), y_{r-1}(\lambda),\ldots, y_1(\lambda)$, тобто визначити усі головні невідомі через числа $u_{r+1}(\lambda), u_{r+2}(\lambda),\ldots, u_n(\lambda)$.

Тоді система (2.17), а тому й система (2.15), є сумісні, причому при $r=n$ система має єдиний розв'язок, а при $r<n$ більше ніж один розв'язок. Зрозуміло, що всі розв'язки системи (2.17) одержуються вказаним методом. Тобто можна знайти всі розв'язки системи.

Приклад. Нехай маємо систему:

$$\begin{cases} x_1 + 2x_2 - x_3 + x_4 = 1; \\ \qquad\qquad\quad x_3 + x_4 = 0. \end{cases}$$

Ця система східчаста і має вигляд 1) ($k_1=1,\ k_2=3,\ r=2;\ x_1, x_3 -$ головні невідомі; $x_2, x_4 -$ вільні невідомі). Нехай $x_2=u_2,\ x_4=u_4,$ де $u_2, u_4 -$ довільні числа. Тоді з останнього рівняння системи маємо $x_3 = -u_4$, а x_1 можна визначити з першого рівняння:

$$x_1 = -2u_2 + x_3 - u_4 + 1 = -2u_2 + (-u_4) - u_4 + 1 = -2u_2 - 2u_4 + 1,$$

тобто $x_1 = -2u_2 - 2u_4 + 1.$

Отже, $\left\{\left(-2u_2 - 2u_4 + 1, u_2, -u_4, u_4\right) \middle| \, u_2 \in \mathbb{R}, u_4 \in \mathbb{R}\right\}$ – множина усіх розв'язків даної системи.

Вище було наведено метод розв'язання східчастих систем (обернений хід методу Гаусса – послідовне знаходження головних невідомих $y_r(\lambda), y_{r-1}(\lambda),\ldots, y_1(\lambda)$ через вільні невідомі). Покажемо, що кожна система лінійних рівнянь за допомогою методу відсічених систем може бути зведена до системи східчастого вигляду.

Розглянемо прямокутну матрицю [62] $A \in R^{m \times n}$ із щільним заповненням і припустимо, що $m \times n$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n-1} & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n-1} & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n-1} & a_{3n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn-1} & a_{nn} \\ a_{n+11} & a_{n+12} & a_{n+13} & \dots & a_{n+1n-1} & a_{n+1n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn-1} & a_{mn} \end{pmatrix}. \tag{2.19}$$

Розглянемо також матрицю $B \in R^{m \times n}$, що має заповнення:

$$\begin{pmatrix} b_{11} & b_{12} & b_{13} & \dots & b_{1n-1} & b_{1n} \\ 0 & b_{22} & b_{23} & \dots & b_{2n-1} & b_{2n} \\ 0 & 0 & b_{33} & \dots & b_{3n-1} & b_{3n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & b_{nn-1} & b_{nn} \\ 0 & 0 & 0 & \dots & 0 & b_{n+1n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}. \tag{2.20}$$

ТЕОРЕМА 2.3. *Прямокутна матриця* $A \in R^{m \times n}$, *всі мінори якої* $A\begin{bmatrix} 1 & 2 & 3 & \dots & i-1 & i \\ 1 & 2 & 3 & \dots & i-1 & i \end{bmatrix}$ *відмінні від нуля, може бути зведена до матриці* $B \in R^{m \times n}$ *трапецеподібної форми (2.20), елементи* b_{ij} *якої мають вигляд:*

$$b_{ij} = a_{ij} - \sum_{l=1}^{i-1} a_{lj} x_l^{(i-1)}, \tag{2.21}$$

де $x_l^{(i-1)}$ *- розв'язки систем лінійних рівнянь виду*

$$\sum_{l=1}^{i-1} a_{ls} x_l^{(i-1)} = a_{js}, \ (s = 1, 2, 3, \dots, i-1). \tag{2.22}$$

ДОВЕДЕННЯ. Якщо матриця виду (2.22) за допомогою еквівалентних перетворень зведена до матриці (2.21), то згідно з [62], їх елементи пов'язані між собою співвідношенням:

$$b_{ij} = A\begin{bmatrix} 1 & 2 & 3 & \dots & i-1 & i \\ 1 & 2 & 3 & \dots & i-1 & j \end{bmatrix} / A\begin{bmatrix} 1 & 2 & 3 & \dots & i-2 & i-1 \\ 1 & 2 & 3 & \dots & i-2 & i-1 \end{bmatrix} \qquad (2.23)$$

Розкриваючи визначник у чисельнику за мінорами, останню рівність подамо у вигляді:

$$b_{ij} = \left(a_{ij} A\begin{bmatrix} 1 & 2 & 3 & \dots & i-2 & i-1 \\ 1 & 2 & 3 & \dots & i-2 & i-1 \end{bmatrix} - \sum_{l=1}^{i-1} a_{ij} A\begin{bmatrix} 1 & 2 & \dots & l-1 & i & l+1 & \dots & i-1 \\ 1 & 2 & \dots & l-1 & i & l+1 & \dots & i-1 \end{bmatrix} \right) /$$

$$/ A\begin{bmatrix} 1 & 2 & 3 & \dots & i-2 & i-1 \\ 1 & 2 & 3 & \dots & i-2 & i-1 \end{bmatrix} = a_{ij} - \sum_{l=1}^{i-1} a_{ij} \frac{A\begin{bmatrix} 1 & 2 & \dots & l-1 & i & l+1 & \dots & i-1 \\ 1 & 2 & \dots & l-1 & i & l+1 & \dots & i-1 \end{bmatrix}}{A\begin{bmatrix} 1 & 2 & 3 & \dots & i-2 & i-1 \\ 1 & 2 & 3 & \dots & i-2 & i-1 \end{bmatrix}}.$$

Якщо розглянути тепер систему виду (2.22), то для невідомої $x_l^{(i-1)}$ за правилом Крамера можна записати:

$$x_l^{(i-1)} = A\begin{bmatrix} 1 & 2 & \dots & l-1 & i & l+1 & \dots & i-1 \\ 1 & 2 & \dots & l-1 & l & l+1 & \dots & i-1 \end{bmatrix} / A\begin{bmatrix} 1 & 2 & 3 & \dots & i-2 & i-1 \\ 1 & 2 & 3 & \dots & i-2 & i-1 \end{bmatrix},$$

звідси й випливає дане твердження. Теорема 2.3 доведена.

Теорема 2.3 дозволяє створювати ефективні алгоритми для проведення еквівалентних перетворень матриць та визначників.

Висновки до другого розділу

В даному розділі описано дослідження розв'язків систем двох лінійних рівнянь з двома невідомими, а також системи лінійних рівнянь з λ-матрицями.

Наведено елементарні перетворення системи лінійних рівнянь, проаналізовано східчасті системи з λ-матрицями.

Показано зведення системи лінійних рівнянь до східчастого вигляду за методом відсічених систем.

Результати другого розділу опубліковані у праці [81] та доповідались на конференціях [77; 79].

РОЗДІЛ 3. МАТЕМАТИЧНІ МОДЕЛІ МІЖГАЛУЗЕВОГО БАЛАНСУ

Вивчення будь-якого реального явища або процесу на перших стадіях проходить шлях спостереження, експериментів, накопичення фактів, їх осмислення, які закінчуються створенням математичної моделі. Математична модель, як правило, це певні рівняння або нерівності, які досліджуються математичними методами. Математичною мовою вказуються особливості моделі, існування можливих розв'язків задачі, їх властивості.

Поняття математичного моделювання інтерпретується (трактується) різними авторами по різному. Ми пов'яжемо математичне моделювання зі спеціалізацією – *обчислювальні методи*. Під математичним моделюванням розумітимемо методи дослідження процесів або явищ шляхом побудови їхніх математичних моделей і дослідження цих процесів. Основою для методу є адекватність між змінними складеного рівняння і досліджуваного процесу. На практиці ці процеси, як правило, не будуть абсолютно ідентичні. При цьому можна удосконалювати, досліджувану математичну модель, яка точніше описуватиме цей процес.

Модель – це деякий об’єкт, подібний до явищ, що відображає суттєві аспекти цих явищ.

Моделювання – процес побудови моделі, за допомогою якої вивчається функціонування об’єктів різної природи.

Математична модель (ММ) – система математичних формул, нерівностей або рівнянь, які досить адекватно описують явища та процеси, властиві оригіналу.

Дослідження математичної моделі дає змогу отримувати характеристики реального об'єкта чи системи. Тип математичної моделі залежить як від природи системи, так і від задач дослідження. Схема таких досліджень починається з постановки задачі і завершується проведенням ефективного обчислювального експерименту. Процес моделювання передбачає наявність трьох структурних елементів:

- ✓ об'єкта дослідження;
- ✓ суб'єкта (дослідник);
- ✓ моделі, що опосередковує відносини між суб'єктом і об'єктом.

Побудова ММ у загальному випадку складається з розглянутих далі етапів.

Етап 1. Постановка задачі та її якісний аналіз.

На цьому етапі потрібно сформулювати зміст задачі, визначити передумови й висловити припущення. Необхідно виділити найважливіші властивості об'єкта моделювання, вивчити його структуру, дослідити взаємозв'язки між його елементами, а також хоча б попередньо сформулювати гіпотези, що пояснюють поведінку й розвиток об'єкта (динаміку руху). При цьому складні об'єкти розбиваються на частини (елементи) окремого дослідження: визначаються зв'язки та логічні співвідношення між ними, їхні кількісні та якісні властивості. Зазначені дії становлять етап системного аналізу задачі, в результаті якого об'єкт подається у вигляді системи.

Етап 2. Побудова математичної моделі.

Цей етап полягає у формалізації моделі, тобто вираженні її у вигляді конкретних математичних залежностей (рівнянь, систем, нерівностей тощо). Процес побудови моделі складається з кількох стадій. Спочатку вивчають можливості її застосування в розглядуваному конкретному випадку, уточнюють перелік змінних та параметрів, форми зв’язку між ними. Для складних об’єктів доцільно будувати кілька різноаспектних моделей.

Етап 3. Математичний аналіз моделі.

На цьому етапі математичними методами досліджують загальні властивості моделей та розв’язків. Може трапитись так, що раніше виконаний системний аналіз зумовив такий набір елементів, властивостей і співвідношень, для якого немає прийнятного методу розв'язання задачі. Тоді доводиться повертатися до етапу системного аналізу. Важливим моментом є доведення існування розв'язків сформульованої задачі. Під час аналітичного аналізу з'ясовують кількість розв'язків (єдиний чи неєдиний), визначають змінні та параметри, які можуть входити до розв'язку, а також межі та тенденції їх зміни.

Проте моделі складних економічних об'єктів дуже погано піддаються аналітичному дослідженню. У таких випадках використовують числові методи дослідження. Як правило, задачі, що виникають в економічній практиці, намагаються звести до відомих моделей, для яких розроблено методи й алгоритми розв'язання.

Етап 4. Узагальнення та підготовка початкової інформації для дослідження даного класу задач.

Математичне моделювання висуває жорсткі вимоги до якості інформації. У процесі підготовки інформації використовуються методи теорії ймовірностей, математичної статистики, а також економічної статистики для агрегування, групування даних, оцінювання вірогідності даних тощо.

В процесі системного економіко-математичного моделювання результати функціонування одних моделей виступають початковою інформацією для інших.

Етап 5. Числове моделювання.

Цей етап передбачає розробку алгоритмів числового розв’язання задачі, підготовку комп'ютерних програм та безпосереднє виконання розрахунків. При цьому виникають значні труднощі, зумовлені великою розмірністю задач. Числове моделювання істотно доповнює результати аналітичного дослідження.

Етап 6. Аналіз числових результатів та їх застосування.

На цьому етапі передусім з'ясовується найважливіше питання щодо правильності й повноти результатів моделювання та можливості їх практичного використання, а також досліджуються можливі напрямки подальшого вдосконалення моделі.

Перелічені етапи математичного моделювання перебувають у тісному взаємозв'язку. Можуть існувати ще зворотні зв'язки між етапами. Так, на етапі побудови моделі може з'ясуватися, що постановка задачі суперечлива чи призводить до занадто складної математичної моделі, тоді попередню постановку доводиться коригувати.

Найчастіше потреба повернення до попереднього етапу постає на етапі підготовки початкової інформації. Якщо необхідної інформації немає або її пошук викликає великі витрати, доводиться повертатися до етапу формалізації і пристосовуватися до наявної інформації.

Отже, моделювання – циклічний процес. Після останнього етапу необхідно переходити до першого й уточнювати постановку задачі згідно зі здобутими результатами, потім – до другого й уточнювати (коригувати) математичний модуль, далі – до третього і т. д.

Будь-яке прогнозування та моделювання економічних процесів завжди припускає об'єднання теорії (математичної моделі) із практикою (експериментом і статистичними даними). Як приклад можна назвати моделі міжгалузевого балансу.

Міжгалузевий баланс – це метод обліку й аналізу макроекономічної інформації, призначений для представлення взаємозв'язку між макроекономічними показниками й об'ємно-вартісними показниками окремих галузей. Під галуззю в міжгалузевому балансі розуміють групу виробництв, що випускають однорідну чи схожу за своїми споживчими властивостями продукцію на основі однотипних технологій. Такі галузі ще називаються чистими, на відміну від виробничих галузей, сформованих за принципом підпорядкованості виробництв загальному керівному органу (міністерству, відомству). Основу інформаційного забезпечення міжгалузевих моделей становить технологічна матриця, що містить коефіцієнти прямих матеріальних витрат на виробництво одиниці продукції. Ця матриця є базою економіко-математичних міжгалузевих моделей. Різноманітні модифікації моделі міжгалузевого балансу виробництва й розподілу продукції дозволяють розширити коло показників, які охоплює модель.

3.1. Матрична модель міжгалузевого балансу

Міжгалузевий балансовий метод, який використовується для аналізу та планування виробництва й розподілу продукції на різних ієрархічних рівнях управління, покладено в основу матричного моделювання.

Основним видом матричних моделей є міжгалузевий баланс виробництва та розподілу продукції. Прикладом таких моделей є матрична модель міжгалузевого балансу (модель В. Леонтьєва), яка розробляється для відображення системи матеріально-речових зв'язків процесу відтворення суспільного продукту.

Балансові моделі будуються як числові матриці – прямокутні таблиці чисел. У зв'язку з цим балансові моделі належать до типу матричних економіко-математичних моделей. У матричних моделях балансовий метод дає чітке математичне вираження. Розглянемо схему міжгалузевого балансу, в якій виділяють чотири основних частини, кожна з яких має різний економічний зміст. Їх зазвичай називають квадрантами балансу.

Перший квадрант – це таблиця міжгалузевих потоків. Показники, що містяться на перетині рядків і стовпців, є обсягами міжгалузевих потоків продукції x_{ij}, i та j - відповідно номери галузей виробників і споживачів. Перший квадрант – за формою квадратна матриця *n*-го порядку, сума всіх елементів якої дорівнює річному фонду відтворення амортизації засобів виробництва у матеріальній сфері.

У другому квадранті подано кінцеву продукцію всіх галузей матеріального виробництва, де кінцевою продукцією вважається продукція, що виходить зі сфери виробництва в кінцеве використання (на споживання та накопичення). У таблиці цей розділ подано в узагальненому вигляді як один стовпчик величин Y_{ii}; у розгорнутій схемі балансу кінцевий продукт кожної галузі можна подати диференційовано за напрямами використання: на особисте споживання населення, суспільне споживання, на накопичення, покриття збитків, експорт тощо.

У третьому квадранті представлена чиста продукція як сума оплати праці та чистого прибутку.

Четвертий квадрант показує розподіл і використання національного доходу. Дані четвертого квадранта важливі для відображення в міжгалузевій моделі балансу доходів і витрат населення, джерел фінансування капіталовкладень, поточних витрат невиробничої сфери, для аналізу загальної структури доходів за групами споживачів.

Припустимо, що розглядається виробнича сфера з n різних галузей, кожна з яких виробляє свій однорідний продукт. У процесі виробництва свого виду продукції кожна галузь має потребу в продукції інших галузей (так зване виробниче споживання).

Нехай в деякий момент часу складено балансовий звіт за якимось об'єктом за рік на основі підсумкових даних за такою таблицею:

Галузі виробництва	Галузі-споживання					Кінцевий продукт	Валовий продукт
	1	2	3	...	n		
1	x_{11}	x_{12}	x_{13}	...	x_{1n}	Y_1	X_1
2	x_{21}	x_{22}	x_{23}	...	x_{2n}	Y_2	X_2
...	...	...	...	I	...	II	…
n	x_{n1}	x_{n2}	x_{n3}	...	x_{nn}	Y_n	Y_n
Оплата праці	v_1	v_2	v_3	III	v_n	IV	
Чистий дохід	m_1	m_2	m_3	...	m_n		
Валовий продукт	X_1	X_2	X_3	...	X_n		$\sum_{i=1}^{n} X_i = \sum_{j=1}^{n} X_j$

Таблиця 3.1.1.

Кожна галузь у матричній моделі міжгалузевого балансу представлена двічі: як галузь виробництва (індекс $i, i=\overline{1,n}$) і як галузь споживання (індекс $j, j=\overline{1,n}$), тобто i-а галузь виробництва відображена в балансі рядком, а j-а галузь споживання – стовпцем. Позначимо зв'язки між галузями через міжгалузеві потоки x_{ij}. Які показують, яка вартість засобів виробництва i-ої галузі споживається як матеріальні витрати в j-й галузі. Позначимо через x_i, x_j – валову продукцію у вартісному вираженні відповідно i-ої, j-ої галузі виробництва, Y_i – обсяг кінцевої продукції i-ої галузі, тобто частину, яка була спожита у невиробничій сфері для створення запасів, інвестицій тощо.

Рядок $x_{i1}, x_{i2}, ..., x_{in}$ характеризує розподіл річного обсягу продукції i–ої галузі у всіх галузях виробничої сфери. Тоді валова продукція i-ої галузі становитиме:

$$X_i = \sum_{j=1}^{n} x_{ij} + Y_i, \; i = \overline{1,n}. \tag{3.1}$$

Отже, валова продукція будь-якої галузі дорівнює сумі матеріальних витрат галузей, які споживають її продукцію, і кінцевої продукції даної галузі.

Рівняння (3.1) – це система n рівнянь, що називається системою рівнянь розподілу або використання продукції галузей матеріального виробництва.

У стовпцях матричної моделі міжгалузевого балансу представлена структура матеріальних витрат $\left(x_{1j}, x_{2j}, ..., x_{nj}\right)$ і чистої продукції $\left(v_j + m_j\right)$ кожної галузі.

Тобто сума матеріальних витрат будь-якої галузі споживання і її чистої продукції дорівнює валовій продукції даної галузі:

$$X_j = \sum_{i=1}^{n} x_{ij} + \nu_j + m_j, \; j = \overline{1,n}. \tag{3.2}$$

Співвідношення (3.2) – це система, яка відображає вартісний склад продукції всіх галузей матеріального виробництва.

Відомий американський економіст В. Леонтьєв, аналізуючи у 1936 р. причини економічної депресії США 1929-1932 рр., помітив важливу закономірність: протягом тривалого часу величини $a_{ij} = x_{ij} / x_i$, $i, j = \overline{1, n}$. змінюються дуже мало і можуть вважатися сталими, тобто обсяг споживання j-ою галуззю продукції i-ої галузі в процесі виробництва своєї продукції обсягом x_i є технологічною сталою. За таких умов технологію виробництва вважають лінійною. Числа a_{ij} називають *коефіцієнтами прямих матеріальних витрат.*

Означення 3.1. Коефіцієнтом прямих матеріальних витрат a_{ij} називається коефіцієнт, який показує скільки продукції i-ої галузі необхідно (якщо враховувати тільки прямі витрати) для виробництва одиниці продукції j-ої галузі.

Величини a_{ij} називають коефіцієнтами прямих матеріальних витрат та обчислюють так: $a_{ij} = \dfrac{x_{ij}}{X_j}$, $a_{ij} = const$, $i, j = 1, ...n.$

На відміну від умови лінійності економічної системи вважається, що кожна галузь виробляє лише один вид продукції (спільне виробництво різних товарів не припустиме) і в ній виробляється, купується, споживається та інвестується n видів продукції. Тому, зважаючи на ці припущення, виробниче споживання i-ого продукту всіма галузями дорівнює $\sum_{j=1}^{n} a_{ij} x_j$, а чистий випуск i-го продукту складає величину:

$$x_i - \sum_{j=1}^{n} a_{ij} x_j, \quad \left(i = \overline{1, n}\right). \tag{3.3}$$

Прирівнявши кінцеву продукцію до чистого випуску (3.2), одержимо систему рівнянь:

$$x_i - \sum_{j=1}^{n} a_{ij} x_j = y_i, \quad \left(i = \overline{1, n}\right). \tag{3.4}$$

Нехай $A=\left(a_{ij}\right)-$матриця коефіцієнтів прямих матеріальних витрат, $X-$вектор-стовпець валової продукції і Y - вектор-стовпець кінцевої продукції. Тоді система рівнянь (3.4) у матричній формі набуде вигляду

$$X = AX + Y. \tag{3.5}$$

Систему рівнянь (3.4), або у матричній формі (3.5), називають моделлю міжгалузевого балансу (моделлю Леонтьєва, моделлю "витрати-випуск").

Користуючись цією моделлю можна виконувати три види розрахунків, а саме:

- ✓ визначити обсяги Y_j кінцевої продукції кожної галузі, задавши в моделі її обсяги X_i валової продукції $Y=\left(E-A\right)X$, де $E-$одинична матриця n-го порядку;
- ✓ знайти обсяг X_i валової продукції кожної галузі, задавши обсяги Y_j кінцевої продукції всіх галузей:

$$X = \left(E - A\right)^{-1} Y, \tag{3.6}$$

де $\left(E - A\right)^{-1}$ - матриця обернена до матриці $\left(E - A\right)$;

- ✓ обчислити обсяги кінцевої продукції Y_j для деяких галузей і обсяги валової продукції X_i для решти галузей, задавши обсяги валової продукції X_i для деяких галузей, а для решти галузей – обсяги кінцевої продукції Y_j.

Якщо визначник матриці $\left|\left(E - A\right)\right|$ № 0, тобто ця матриця невироджена, то існує обернена до неї матриця. Позначивши її через $B = \left(E - A\right)^{-1}$, можна записати систему рівнянь у матричній формі: $X = BY$.

Нехай b_{ij} – елементи матриці B, тоді з цього матричного рівняння для будь-якої i-ої галузі можна отримати таке співвідношення:

$$\sum_{j=1}^{n} b_{ij} Y_j = X_i, \quad \left(i=\overline{1,n}\right). \tag{3.7}$$

Із (3.7) випливає, що обсяг валової продукції – зважена сума обсягів кінцевої продукції, причому "вагами" є коефіцієнти b_{ij}, що показують скільки всього потрібно виготовити продукції i-ої галузі, щоб у сферу кінцевого використання надійшла одиниця продукції j-ої галузі. Коефіцієнти b_{ij} називають *коефіцієнтами повних матеріальних витрат* і охоплюють як прямі, так і непрямі витрати всіх порядків. Прямі витрати показують кількість засобів виробництва, що витрачаються безпосередньо під час виготовлення певного продукту, а непрямі стосуються попередніх стадій виробництва і входять у виробництво продукту опосередковано через інші (проміжні) засоби виробництва.

Означення 3.2. Коефіцієнтом повних матеріальних витрат b_{ij} називається коефіцієнт, який показує скільки продукції i-ої галузі необхідно виробити, щоб з урахуванням прямих і непрямих її витрат одержати одиницю кінцевої продукції j-ої галузі.

Для того щоб визначити як позначиться на валовому випуску деякої галузі передбачувана зміна обсягів кінцевої продукції всіх галузей, використаємо коефіцієнти повних матеріальних витрат і запишемо:

$$\sum_{j=1}^{n} b_{ij}\Delta Y_j = \Delta X_i, \quad \left(i=\overline{1,n};\ j=\overline{1,n}\right),$$

де ΔX_i, ΔY_j – зміна (приріст) обсягу відповідної валової і кінцевої продукції.

Аналізуючи модель міжгалузевих балансів, розглянемо перш за все основні властивості матриці A коефіцієнтів прямих матеріальних витрат. За означенням коефіцієнти прямих витрат є невід'ємними. Крім цього, оскільки відтворення було б неможливим, якщо для власного відтворення в галузі витрачалося більше продукту ніж створювалося, то діагональні елементи матриці A, очевидно, менші за одиницю: $a_{ij} < 1$.

Означення 3.3. Невід'ємну матрицю A називатимемо продуктивною, якщо існує невід'ємний вектор X і 0, такий що $X > AX$.

Умова $X > AX$, означає існування додатного вектора кінцевої продукції Y для моделі міжгалузевого балансу (3.5).

Для того щоб матриця коефіцієнтів прямих матеріальних витрат A була продуктивною, необхідно і достатньо, щоб виконувалась одна з наведених далі умов:

1) існує невід'ємна матриця $(E - A)^{-1} \geq 0$;

2) матричний ряд $E + A + A^2 + A^3 + ... = \sum_{k=0}^{\infty} A^k$ збіжний, причому його сума дорівнює матриці $(E - A)^{-1}$;

3) найбільше за модулем власне значення матриці A, тобто розв'язок характеристичного рівняння $|A - \lambda E| = 0$, строго менше від одиниці;

4) усі головні мінори матриці $(E - A)$, тобто визначники матриць, утворені елементами перших рядків і перших стовпців цієї матриці, порядку від 1 до n, додатні.

Розглянемо третю умову. Нехай λ^* - найбільший за модулем корінь характеристичного рівняння, наведеного в умові 3), продуктивності матриці A. Значення його може бути оцінкою загального рівня коефіцієнтів прямих матеріальних витрат, а отже значення $\left(1 - \lambda^*\right)$ характеризує залишок після витрат, тобто продуктивність. Чим більше $\left(1 - \lambda^*\right)$, тим більші можливості досягти ще й інших цілей, крім поточного виробничого споживання. Це означає, що чим вищий загальний рівень коефіцієнтів матриці A, тим більше найбільше за модулем власне значення λ^* і тим нижчий рівень продуктивності, та навпаки: чим нижчий загальний рівень коефіцієнтів матриці A, тим менше найбільше за модулем власне значення і тим вища продуктивність.

Проаналізуємо матрицю коефіцієнтів повних матеріальних витрат, тобто матрицю $B = (E - A)^{-1}$. Коефіцієнт цієї матриці показує, скільки всього потрібно виробити продукції i-ої галузі, щоб одержати одиницю кінцевої продукції j-ої галузі. Наведемо ще одне визначення коефіцієнта повних матеріальних витрат, узявши до уваги, що крім прямих витрат існують непрямі виробничі витрати під час виготовлення тієї чи іншої продукції будь-якої галузі. Розглянемо, наприклад, формування витрат електроенергії на випуск дерев'яних столів, обмежившись технологічним ланцюжком "*дерево – дошки – бруски (заготовки деталей) – столи*". Витрати електроенергії, що супроводжуються виготовленням стола і брусків, називатимемо прямими витратами. Відповідні витрати на виготовлення брусків з дошок отримають назву непрямих витрат першого порядку, а витрати електроенергії, що необхідні для одержання дощок з дерева – непрямих витрат другого порядку і т.д. Звідси випливає ще одне означення.

Означення 3.4. Коефіцієнтом повних матеріальних витрат d_{ij} називається сума прямих і непрямих витрат продукції i-ої галузі для виробництва одиниці продукції j-ої галузі з урахуванням усіх проміжних продуктів на всіх попередніх стадіях виробництва.

Нехай (a_{ij})- коефіцієнт непрямих матеріальних витрат k-го порядку. Тоді має місце формула $d_{ij} = a_{ij} + a_{ij}^{(1)} + a_{ij}^{(2)} + \ldots + a_{ij}^{(k)} + \ldots$, або у матричному вигляді $D = A + A^{(1)} + A^{(2)} + \ldots + A^{(k)} + \ldots$. За змістом коефіцієнтів непрямих матеріальних витрат запишемо матричні співвідношення:

$$
\begin{aligned}
&A^{(1)} = AA = A^2;\\
&A^{(2)} = AA^{(1)} = AA^2 = A^3;\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&A^{(k)} = AA^{(k-1)} = AA^k = A^{k+1}.
\end{aligned}
$$

Аналогічно матричну формулу подамо у вигляді $D = A + A^2 + A^3 + \ldots = \sum_{k=1}^{\infty} A^{(k)}$. Якщо матриця A коефіцієнтів прямих матеріальних витрат є продуктивною, то з умови 2 продуктивності випливає існування матриці $B = (E - A)^{-1}$, що є сумою збіжного матричного ряду:

$$B = (E - A)^{-1} = E + A + A^2 + A^3 + \ldots = \sum_{k=0}^{\infty} A^k.$$

Порівнюючи два останніх співвідношення, встановимо такий зв'язок між двома матрицями коефіцієнтів повних матеріальних витрат: $B = E + D$ або за елементним записом:

$$b_{ij} = \begin{cases} d_{ij}, \text{ якщо } i \neq j; \\ 1 + d_{ij}, \text{ якщо } i = j. \end{cases}$$

Цей зв'язок визначає розбіжність між коефіцієнтами матриць B і D. Коефіцієнти матриці D враховують тільки витрати на виробництво продукції, а коефіцієнти матриці B, крім витрат, містять одиницю кінцевої продукції, що виходить за сферу виробництва. Отже, за допомогою міжгалузевої моделі на основі прямих і повних витрат праці на одиницю продукції можна побудувати балансові моделі.

3.2. Дослідження продуктивності моделі міжгалузевого балансу

Аналіз моделі Леонтьєва з математичного погляду пов'язаний із теорією невід'ємних матриць. Останні в свою чергу поділяють на два класи: розкладні та нерозкладні.

Означення 3.5. Невід'ємна матриця $B = (b_{ij})_{i,j=1}^{n}$ називається розкладною, якщо множину індексів $K = \{1, \ldots, n\}$ можна розбити на таких дві підмножини $K_1 = \{i_1, \ldots, i_k\}$, $K_2 = \{j_1, \ldots, j_m\}$, причому $K_1 \cap K_2 = \varnothing$, $K_1 \cup K_2 = K$, і $(k + m = n)$ $b_{i_a j_b} = 0$, $i_a \in K_1$, $j_b \in K_2$.

Всі інші невід'ємні матриці називаються нерозкладними. Зазначимо, що підмножини K_1 і K_2 називають ізольованими. Перенумеруємо індекси так, щоб вони набули вигляду $K_1 = \{1,2,\ldots,l\}$, а $K_2 = \{l+1,\ldots,k\}$. Тоді для матриці B вони означатимуть одночасну перестановку рядків і стовпців. Матрицю B за допомогою такої операції можна звести до вигляду:

$$B = \begin{pmatrix} B_{11} & 0 \\ B_{21} & B_{22} \end{pmatrix},$$

де B_{11} і B_{22} – квадратні матриці.

Перерахуємо без доведення деякі властивості нерозкладних матриць:

1. Нерозкладна матриця не може мати ні нульових рядків, ні нульових стовпців.
2. Якщо матриця B нерозкладна, а вектор $x > 0$, то $Bx > 0$.
3. Якщо B - нерозкладна матриця розмірності $(m \times m)$, то $(I_m + B)^{m-1} > 0$, тобто всі елементи матриці $(I_m + B)^{m-1}$ додатні.
4. Якщо B - нерозкладна матриця, то для будь-якої пари індексів (i,j) $(i,j \in I)$ існує натуральне число n, для якого $b_{ij}^{(k)} > 0$, де $b_{ij}^{(k)} \in B^k$.
5. Для того щоб матриця B була нерозкладною, необхідно і достатньо, щоб для будь-яких індексів i,j існувала послідовність $\{l_k\}$, $1 \le l_k \le n$, $k = \overline{1,m}$ така, що $l_1 = i$, $l_m = j$, $b_{l_k l_{k+1}} > 0$, $k = \overline{1,m-1}$.
6. Якщо B – нерозкладна матриця, а n – довільне натуральне число, то в матриці B^n не може бути ні нульових рядків, ні нульових стовпців.

Розкладність матриці прямих матеріальних витрат $B = (b_{ij})_{i,j=1}^{n}$ означає, що існує група ізольованих галузей, які можуть функціонувати незалежно від інших галузей. У випадку нерозкладності цієї матриці такої групи галузей не існує. Розглянемо деякі факти з теорії невід'ємних матриць [95].

ТЕОРЕМА 3.1. *(Перрона-Фробеніуса про спектральні властивості невід'ємної нерозкладної матриці). Нехай* $B = \left(b_{ij}\right)_{i,j=1}^{n}$ *– невід'ємна нерозкладна квадратна матриця, а* $\Lambda\left(A\right) = \left\{\lambda_1, \lambda_2, ..., \lambda_z\right\}$, $z \le n$ *– множина її власних значень. Тоді в множині* $\Lambda\left(B\right)$ *є таке власне значення* $\lambda_B > 0$, *що* $\lambda_B \ge \left|\lambda_t\right|$, $t = \overline{1, z}$, *і йому відповідає єдиний власний вектор*

$c_B > 0, \left(0 \text{ O } R^n\right)$. *Це число* λ_B *називають числом Фробеніуса, а вектор*

c_B - *вектором Фробеніуса матриці* B *і* $B_{\chi_B} = \lambda_B \chi_B$. Доведення теореми подано в [26].

Зауваження 3.1. Вектор χ_B називатимемо правим вектором Фробеніуса, оскільки існує вектор Y_B такий, що виконується умова $Y_B^T B = \lambda_B Y_B^T$ або $B^T Y_B = \lambda_B Y_B$.

Вектор Y_B називатимемо лівим вектором Фробеніуса. У випадку нерозкладності матриці B , транспонована матриця B^T теж буде нерозкладною.

ТЕОРЕМА 3.2. *Нехай* $B = \left(b_{ij}\right)_{i,j=1}^{n}$ *– невід'ємна квадратна матриця, а* $\Lambda\left(A\right)$ *– множина її власних значень. Тоді в множині* $\text{L}\left(B\right)$ *є таке власне значення* $\lambda_B \ge 0$, *що* $\lambda_B \ge \left|\lambda\right|$, $\lambda \in \Lambda\left(B\right)$, *і йому відповідає власний вектор*

c_B і $0, \left(0 \text{ O } R^n\right)$.

Доведення теореми подано в роботі [26].

ТЕОРЕМА 3.3. *Нехай* $B = \left(b_{ij}\right)_{i,j=1}^{n}$ *– невід'ємна квадратна матриця. Тоді для її фробеніусівського числа* λ_B *справджуються такі оцінки:*

$$f \le \lambda_B \le F, \; t \le \lambda_B \le T, \tag{2.8}$$

де $f = \min\limits_{i=\overline{1,n}} f_i$, $F = \max\limits_{i=\overline{1,n}} f_i$, $t = \min\limits_{j=\overline{1,n}} t_j$, $T = \max\limits_{j=\overline{1,n}} t_j$, $f_i = \sum\limits_{j=1}^{n} b_{ij}$, $t_j = \sum\limits_{i=1}^{n} b_{ij}$.

Доведення теореми подано в роботі [26].

Якщо, крім вище згаданих умов, матриця B ще й нерозкладна, то нерівності виду (2.8) є строгими, за винятком випадку $f = F$, $t = T$.

ТЕОРЕМА 3.4. *Для продуктивності моделі міжгалузевого балансу (моделі Леонтьєва) з невід'ємною нерозкладною матрицею* $B = \left(b_{ij}\right)_{i,j=1}^{n}$ *необхідно і достатньо, щоб фробеніусівське число* λ_B *даної матриці задовольняло нерівність* $\lambda_B < 1$.

ДОВЕДЕННЯ. *Достатність*. Припустимо, що $\lambda_B < 1$, доведемо, що звідси випливає продуктивність моделі. Згідно з теоремою 3.1 $B_{\chi_B} = \lambda_B \chi_B$, $\lambda_B > 0$, $0 < \lambda_B < 1$, то $B^k \chi_B = B^{k-1}\left(B\chi_B\right) = \lambda_B\left(B^{k-1}\chi_B\right) = \lambda_B^2\left(B^{k-2}\chi_B\right) = \ldots =$ $= \lambda_B^k \chi_B$, тому $\lim\limits_{k\to\infty} B^k \chi_B = \lim\limits_{k\to\infty} \lambda_B^k \chi_B = 0$. Враховуючи те, що $\chi_B > 0$ (матриця B нерозкладна) і $B^k \geq 0$, робимо висновок, що $\lim\limits_{k\to\infty} B^k = 0$.

Для довільної квадратної матриці B справедлива тотожність:

$$\left(I_m - B\right)\left(I_m + B + B^2 + \ldots + B^{k-1}\right) = I_m - B^k. \tag{3.9}$$

Оскільки границя правої частини (3.9) при $k \to \infty$ існує і дорівнює I_m, то існує також границя лівої частини. Отже $\left(I_m - B\right)\sum\limits_{k=0}^{\infty} B^k = I_m$, тобто ряд $\sum\limits_{k=0}^{\infty} B^k$ збігається, а його сума є матрицею $\left(I_m - B\right)^{-1}$, оберненою до $\left(I_m - B\right)$. Отже, існування матриці $\left(I_m - B\right)^{-1}$ доведено. Окрім цього, ця матриця невід'ємна (складається з невід'ємних елементів), бо $\left(I_m - B\right)^{-1} \sum\limits_{k=0}^{\infty} B^k \geq 0$ у зв'язку з тим, що $B^k \geq 0$. Це означає, що для довільного вектора існує невід'ємний розв'язок (3.6) системи (3.5), тобто модель продуктивна.

Необхідність. Покажемо, що з продуктивності моделі міжгалузевого балансу (моделі Леонтьєва) випливає нерівність $\lambda_B < 1$.

Легко зрозуміти, що для будь-якого вектора $y \geq 0$ існує вектор валового випуску $x \geq 0$, для якого $x - Bx = y$, $y > 0$, тобто $x \geq Bx$. Помножимо скалярно останню нерівність на вектор:

$$p_B \langle x, p_B \rangle > \langle Bx, p_B \rangle = \langle x, B^T p_B \rangle = \lambda_B \langle x, p_B \rangle.$$

Оскільки $x \geq 0$, $p_B > 0$, матриця B нерозкладна, то $\lambda_B < 1$, що й потрібно було довести. Теорему доведено.

Зауважимо, що аналогічну теорему для моделі Леонтьєва подано в [26].

ТЕОРЕМА 3.5. *Якщо система* $x = Ax + y, x \geq 0$ *має розв'язок при деякому* $y > 0$, *модель міжгалузевого балансу (модель Леонтьєва) продуктивна.*

Доведення цієї теореми легко провести за схемою доведення необхідності попередньої теореми.

ТЕОРЕМА 3.6. *Якщо матриця* $B = (b_{ij})_{i,j=1}^{n}$ *є невід'ємною та нерозкладною, а*

$$f_i = \sum_{j=1}^{n} b_{ij} \leq 1, \; i = \overline{1,n}, \tag{3.10}$$

і хоча б для одного рядка i_0 *виконується нерівність* $f_{i_0} < 1$, *то модель Леонтьєва, яка відповідає цій матриці, продуктивна.*

ДОВЕДЕННЯ. Нехай $e = (1,1,\ldots,1)^T \in R^n$, а p_B лівий вектор Фробеніуса. Тоді $p_B^T Be = \sum_{i=1}^{n} f_i (p_B)_i < \sum_{i=1}^{n} (p_B)_i$. З іншого боку,

$$p_B^T Be = \lambda_B \langle p_B, e \rangle = \lambda_B \sum_{i=1}^{n} (p_B)_i.$$

Порівнявши отримані співвідношення, маємо $\lambda_B < 1$, тобто за теоремою 3.4 міжгалузева балансова модель (модель Леонтьєва) продуктивна.

На основі моделі Леонтьєва розроблено багато інших важливих економіко-математичних моделей. Розглянемо ще модель виробництва з урахуванням споживання.

Як відомо, кінцева продукція y складається з невиробничого споживання та деякого фонду накопичення. Нехай $c \in R^n_+$, $c \neq 0$ - вектор оплати праці одного робітника в кожній з галузей. Тоді матеріальні витрати на виробництво визначаються вектором $(Bx + Dc)$, де D – загальна кількість працюючих, Bx і Dc – відповідно вектори виробничого й невиробничого споживання $(Dc < y)$. Баланс матеріальних і трудових витрат приводить до такої системи нерівностей:

$$\begin{cases} Bx + Dc \leq x, \\ \langle d,x \rangle \leq D, \\ x \geq 0. \end{cases} \tag{3.11}$$

Питання існування розв'язку системи (3.11) потребує окремого дослідження. Введемо до розгляду нові змінні. Нехай $Z = x/D$ – валовий випуск у розрахунку на одного працівника. Тоді від (3.11) можна перейти до системи

$$\begin{cases} Bx + c \leq z, \\ \langle d,z \rangle \leq 1, \\ z \geq 0. \end{cases} \tag{3.12}$$

Означення 3.7. Модель виробництва з урахуванням споживання (3.11) називається c - продуктивною, якщо система нерівностей (3.12) сумісна, тобто має розв'язок.

Якщо $B = (b_{ij})_{i,j=1}^{n}$ - невід'ємна і нерозкладна матриця, а $c \neq 0$, то із c - продуктивності випливає звичайна продуктивність матриці (чи моделі).

Справді, з (3.12) випливає, що $Bz \le z$, причому хоча б для однієї координати виконується строга нерівність. Оскільки в даному випадку вектор Фробеніуса $p_B > 0$, то $\langle Bz, p_B\rangle < \langle z, p_B\rangle$. Крім того, $\langle Bz, p_B\rangle = \langle z, B^T p_B\rangle =$ $= \lambda_B \langle z, p_B\rangle$. Тому $\lambda_B \langle z, p_B\rangle < \langle z, p_B\rangle$, тобто $\lambda_B < 1$.

ТЕОРЕМА 3.7. *Для c-продуктивності моделі (3.11) з невід'ємною нерозкладною матрицею B необхідно й достатньо, щоб виконувалась нерівність*

$$\left\langle d\left(I_n - B\right)^{-1} c\right\rangle \le 1. \tag{3.13}$$

ДОВЕДЕННЯ. *Необхідність.* Матриця $\left(I_n - B\right)^{-1}$ існує на основі (3.11), $\lambda_B < 1$ у випадку c-продуктивності моделі. З першої нерівності (3.12) отримаємо $z = \left(I_n - B\right)^{-1} c \ge 0$, а з другої (3.12) – нерівність (3.13).

Достатність. Нехай виконується (3.13). Покажемо, що модель (3.11) c- продуктивна. Очевидно, що $z = \left(I_n - B\right)^{-1} c \ge 0$ є розв'язком системи (3.12), оскільки цей вектор задовольняє систему $Bz + c = z$, а отже, є допустимим для (3.12).

Можна розглянути також інший критерій c-продуктивності. Введемо до розгляду матрицю $G = \left(g_{ij}\right)_{i,j=1}^{n}$, де $g_{ij} = c_i d_j, i,j = \overline{1,n}$.. Для всіх $x \in R^n$

$$Gx = \langle d,x\rangle c. \tag{3.14}$$

ТЕОРЕМА 3.8. *Для c-продуктивності моделі (3.11) з невід'ємною нерозкладною матрицею B необхідно і достатньо, щоб число Фробеніуса λ_{B+D} матриці $\left(B + D\right)$ не перевищувало одиниці $\lambda_{B+D} \le 1$.*

ДОВЕДЕННЯ. *Необхідність*. Нехай модель (3.11) c -продуктивна, тобто (3.12) сумісна. Тоді з (3.12) і (3.14) випливає, що $Bz + \langle d,z \rangle c$ J $Bz + c$ J z, тобто

$$(B + G)z \ \mathrm{J} \ z. \tag{3.15}$$

Помноживши (3.15) на лівий вектор Фробеніуса p_{B+G}, отримаємо

$$\lambda_{B+G} \langle p_{B+G}, z \rangle = p^T_{B+G} (B+G) z \le \langle p_{B+G}, z \rangle. \tag{3.16}$$

Оскільки B - нерозкладна, то G і 0 і матриця $(B + G)$ також буде нерозкладною, тобто для неї $p_{B+G} > 0$. Враховуючи, що z і 0 $(z$ №0$)$, можемо стверджувати, що $\langle p_{B+G}, z \rangle > 0,$ із (3.16) маємо $\lambda_{B+G} \le 1$.

Достатність. Припускаючи, що $\lambda_{B+G} \le 1,$ встановимо c -продуктивність моделі (3.11). Дійсно, для правого вектора Фробеніуса z_{B+G} матриці $(B + G)$, нормованою умовою $\langle d, z_{B+G} \rangle = 1$, маємо:

$(B+G) z_{B+G} = \lambda_{B+G} z_{B+G} \le z_{B+G}$ або, згідно з (3.11),

$B z_{B+G} + G z_{B+G} = B z_{B+G} + \langle d, z_{B+G} \rangle c = B z_{B+G} + c$ J z_{B+G}.

Отже, остання нерівність, умова нормування вектора z_{B+G} та його невід'ємність означають, що система (3.12) має розв'язок, тобто модель (3.11) продуктивна.

3.3. Статистичні характеристики моделі міжгалузевого балансу

Використовуючи модель Леонтьева, можна спостерігати за реакцією виробничої системи з технологічною матрицею B на зміну вектора кінцевої продукції y. Всі компоненти вектора $y \in B$ є в даному випадку є екзогенними змінними.

Означення 3.8. Незалежні змінні $X_1,\ldots,X_n$, що задані заздалегідь чи за межами моделі, називаються *екзогенними* змінними (регресорами). Залежна змінна Y, що визначається як розв'язок рівняння, називається *ендогенною* змінною (регресандом).

У вигляді строгих математичних тверджень сформулюємо та обґрунтуємо результати досліджень.

ТЕОРЕМА 3.9. *Якщо в моделі (3.5) матриця* $B = \left(b_{ij}\right)_{i,j=1}^{n}$ *невід'ємна, нерозкладна й продуктивна, а вектор* $y \ge 0,\ (y \ne 0)$, *то вектор валового випуску додатний:* $x(y) > 0.$

ДОВЕДЕННЯ. Оскільки $B-$продуктивна матриця, то $C = \left(I_n - B\right)^{-1} = I_n + B + B^2 + \ldots,\ C > 0$, тобто

$$x\left(y\right) = C\left(y\right) = y + By + B^2y + \ldots \tag{3.17}$$

Крім того, на основі властивостей нерозкладних матриць для будь-якої пари індексів $\left(i,j\right) = \left(i,j \in I = \{1,2,\ldots,n\}\right)$, існує таке натуральне число k, що $b_{ij}^{(k)} > 0,\ b_{ij}^{(k)} \in B^k$. Це означає, що $\left(B^k y\right)_i = \sum_{j=1}^{n} b_{ij}^{(k)} y_j$ і $b_{ij}^{(k)} > 0$. Отже, із (3.17) випливає, що для всіх $i = \overline{1,n}$

$$\left(x\left(y\right)\right)_i = y_i + \left(By\right)_i + \left(B^2y\right)_i + \ldots + \left(B^ky\right)_i + \ldots > 0.$$

Отже, встановлено достатність вектора валового випуску. Теорему доведено.

Проаналізуємо тепер, як зміниться вектор валового випуску $x\left(y\right)$, якщо попит на перший товар (тобто перша координата вектора y) збільшиться.

ТЕОРЕМА 3.10. *Нехай в моделі (3.5) матриця* $B = \left(b_{ij}\right)_{i,j=1}^{n}$ - *невід'ємна, нерозкладна і продуктивна.*

Якщо $y = \left(y_1, y_2, \ldots, y_n\right)^T$, $\tilde{y} = \left(\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_n\right)^T$, *причому* $y \geq 0$, *то* $y > \tilde{y}$, *то*

$$\max_{i=\overline{1,n}} \frac{x_j(\tilde{y})}{x_j(y)} = \frac{x_1(\tilde{y})}{x_1(y)} \tag{3.18}$$

і

$$\frac{x_1(\tilde{y})}{x_1(y)} = \frac{x_i(\tilde{y})}{x_i(y)} \tag{3.19}$$

при $i \neq 1$, *то* $y_i = 0$.

ДОВЕДЕННЯ. Оскільки $C = \left(I_n - B\right)^{-1} > 0$, то, згідно з (3.17),

$$x(\tilde{y}) = C\tilde{y} = C\left(y + \tilde{y} - y\right) = Cy + C\left(\tilde{y} - y\right) > Cy = x(y). \tag{3.20}$$

Вважаючи, що $x_j(\tilde{y}) = \gamma_j x_j(y)$, з нерівності (3.20) матимемо: $\gamma_j > 1,\ j = \overline{1,n}.$. З рівностей $x_i(y) = \sum_{j=1}^{n} b_{ij} x_j(y) + y_i$, $x_j(y) = x_j(\tilde{y}) \big/ \gamma_j$ отримаємо співвідношення:

$$x_i(\tilde{y}) = \sum_{j=1}^{n} b_{ij} \frac{\gamma_i x_j(\tilde{y})}{\gamma_j} + \gamma_i y_i. \tag{3.21}$$

Нехай $Q = \left\{ i \big/ \gamma_i = \max_{j=\overline{1,n}} \gamma_j \right\}$. Якщо $1 \in Q$, то рівність (3.18) буде встановлена. Припустивши, що $1 \notin Q$ для всіх $i \in Q$, матимемо $\tilde{y}_i = y_i$ та $\gamma_i \big/ \gamma_j > 1$ при $j \notin Q$, тобто $\gamma_i y_i > \gamma_i \tilde{y}_i > \tilde{y}_i$,

$$\sum_{j=1}^{n} b_{ij} \frac{\gamma_i}{\gamma_j} x_i(\tilde{y}) + \gamma_i y_i > \sum_{j=1}^{n} b_{ij} x_j(\tilde{y}) + \tilde{y}_i.$$

Враховуючи (3.21) отримуємо нерівність:

$$x_i(\tilde{y}) > \sum_{j=1}^{n} b_{ij} x_j(\tilde{y}) + \tilde{y}_i. \tag{3.22}$$

Нерівність (3.22) суперечить концепції моделі Леонтьєва. Усунути цю суперечність можна лише припустивши, що $b_{ij} = 0,\ i \in Q,\ j \notin Q$, але це означало б ізольованість підмножини $Q \subset I = \{1, 2, \ldots, n\}$, тобто розкладність матриці B. Отже, $1 \in Q$ і рівність (3.18) доведена.

Тепер нехай виконується рівність (3.19) при $i \neq 1$ та $y_i > 0$. З того, що $y_i = \tilde{y}_i$ при $i \neq 1$, випливає, що $\gamma_i y_i = \gamma_i \tilde{y}_i > \tilde{y}_i$ та $\gamma_i / \gamma_j \geq 1$, $j = \overline{1,n}$. Однак за цих умов ми знову приходимо до суперечливої нерівності (3.22). Отже, якщо при $i \neq 1$ виконується (3.19), то $y_i = 0$, що й треба було довести.

Зауважимо, що при $y > 0$ з теореми 3.10 випливають нерівності

$$\frac{x_1(\tilde{y})}{x_1(y)} = \frac{x_j(\tilde{y})}{xj(y)}, \; j = \overline{2,n}.$$

Означення 3.9. Еластичністю i-го валового продукту відносно попиту на j-й кінцевий продукт називається величина

$$E_{ij} = \frac{y_j}{x_i(y)} \times \frac{\partial x_i(y)}{\partial y_j} = \frac{\partial x_i(y)}{\partial y_j} : \frac{x_i(y)}{y_j} = \frac{\partial x_i(y)}{x_i(y)} : \frac{\partial y_j}{y_j}.$$

Очевидно, еластичність дорівнює відношенню граничної ефективності до середньої ефективності.

ТЕОРЕМА 3.11. *Якщо матриця $B = (b_{ij})_{i,j=1}^{n}$ – невід'ємна, нерозкладна й продуктивна, то еластичність будь-якого валового продукту відносно попиту на довільний кінцевий продукт не перевищує одиниці: $E_{ij} \leq 1$ $\left(i, j = \overline{1,n}\right)$. Якщо ж $y_i > 0$, то $E_{ij} < 1$ при $i \neq j$.*

ДОВЕДЕННЯ. Проведемо доведення лише для E_{i1}, вважаючи, що $\tilde{y}_1 > y_1 > 0$, $\tilde{y}_j = y_j$, $j = \overline{2,n}$. Як і раніше вважатимемо $x_i(\tilde{y}) = \gamma_i x_i(y)$, а $\tilde{y}_1 = \mu_1 y_1$ $(\mu_1 > 1)$. Тоді

$$\frac{x_i(\tilde{y}) - x_i(y)}{x_i(y)} \times \frac{y_1}{\tilde{y}_1 - y_1} = \frac{\gamma_i - 1}{\mu_1 - 1}. \tag{3.23}$$

З теореми 3.10 випливає, що $g_j \le g_1,\ j=\overline{1,n}$. Оскільки $g_1/g_j \ge 1,\ j=\overline{1,n}$, то на основі (3.21) при $i=1$ матимемо:

$$x_1(\tilde{y}) \ge \sum_{j=1}^{n} b_{ij}x_j(\tilde{y}) + g_1 y_1 = \sum_{j=1}^{n} b_{ij}x_j(\tilde{y}) + g_1\frac{\tilde{y}_1}{m_1}.$$

Крім того, з моделі Леонтьєва випливає, що $\tilde{y}_1 \ge g_1\tilde{y}_1/m_1$, тобто $g_1/m_1 \le 1$ або $g_j \le g_1 \le m_1,\ j=\overline{1,n}$. З (3.23) маємо:

$$\frac{x_i(\tilde{y}) - x_i(y)}{\tilde{y}_1 - y_1}\cdot\frac{y_1}{x_i(y)} = \frac{g_i - 1}{m_1 - 1} \le \frac{m_1 - 1}{m_1 - 1} = 1. \tag{3.24}$$

Перейшовши у (3.24) до границі при $\tilde{y} \to y$, отримаємо нерівність $E_{ij} \le 1$. Якщо $y_1 > 0$, то попередня нерівність буде строгою. Аналогічно встановлюються оцінки для всіх інших еластичностей. Теорему доведено.

Висновки до третього розділу

У даному розділі розглянуто етапи побудови математичних моделей і схему міжгалузевого балансу, в якій виділяють чотири основних частини – квадранти балансу.

Описано дослідження продуктивності моделі міжгалузевого балансу використовуючи теорію невід'ємних матриць.

На основі моделі Леонтьєва розроблено модель виробництва з урахуванням споживання.

Проведено порівняльну статистику моделі міжгалузевого балансу.

Отримані результати опубліковані в працях [64; 76].

РОЗДІЛ 4. УЗАГАЛЬНЕНІ МАТРИЧНІ МОДЕЛІ ЛЕОНТЬЄВА

Узагальнення моделей в наукових дослідженнях і практиці та розширення сфери їх застосування є невід’ємною рисою технічного прогресу. Ефективність побудованих узагальнень визначається продуктивністю ЕОМ та якістю обчислювальних алгоритмів і програм, що використовуються. Сучасні наукові дисципліни широко включають у себе необхідні інструментальні засоби, математичні моделі і методи, які дозволяють здійснювати більш високий рівень формалізації й абстрактного опису найважливіших істотних зв'язків техніко-економічних змінних систем і об'єктів, оцінювати форму і параметри залежностей їх змінних, отримувати нові знання про об'єкти, визначати найкращий розв'язок в тій чи іншій ситуації, формулювати висновки, адекватні вивченому об'єкту, компактно викладати основні теоретичні положення.

4.1. Алгебричні моделі задач економіки в класичній постановці

У значній кількості прикладних економічних задач виникає необхідність використання динамічних математичних моделей.

Достатньо часто в основу алгебричного моделювання покладено міжгалузевий балансовий метод, який використовується для аналізу та планування виробництва і розподілу продукції на різних ієрархічних рівнях управління.

Розглянемо динамічну міжгалузеву модель, що є класичним прикладом використання систем диференціальних рівнянь у дослідженні проблем економічного зростання. Математична модель має вигляд:

$$x(t) = a(t)x(t) + b(t)\frac{dx(t)}{dt} + c(t). \tag{4.1}$$

При дезагрегуванні (деталізації) цієї моделі до галузевого рівня ендогенні та екзогенні змінні $x(t)$, $\frac{dx(t)}{dt}$, $c(t)$ замінюються векторами стовпцями $X(t)$, $\frac{dX(t)}{dt}$, $C(t)$, а параметри a і b – квадратними матрицями A і B. У результаті одержимо систему лінійних диференціальних рівнянь першого степеня, що є узагальненою динамічною моделлю Леонтьєва:

$$X(t) = A(t)X(t) + B(t)\frac{dX(t)}{dt} + C(t), \tag{4.2}$$

де

$X(t) = [x_j(t)]$ – вектор-стовпець обсягів виробництва;

$\frac{dX(t)}{dt} = \left[\frac{dx_j(t)}{dt}\right]$ – вектор-стовпець абсолютних приростів виробництва;

$C(t)$ – вектор-стовпець споживання (разом із невиробничим нагромадженням);

$A(t) = \left(a_{ij}(t)\right)$ – матриця коефіцієнтів прямих матеріальних витрат (на відміну від коефіцієнтів статичного міжгалузевого балансу коефіцієнти в динамічній моделі включають також витрати на відшкодування вибуття і капітальний ремонт основних виробничих фондів) ($i, j \in J$, $J = \{1, \ldots, n\}$);

$B(t) = \left(b_{ij}(t)\right)$ – матриця коефіцієнтів капіталомісткості приростів виробництва (витрати виробничого нагромадження на одиницю приросту відповідних видів продукції) ($i, j \in J$, $J = \{1, \ldots, n\}$).

У такому випадку статична модель міжгалузевого балансу може бути записана:

$$X = A(t)X(t) + Y(t) \tag{4.3}$$

Означення 4.1. Матриця A (всі елементи якої невід'ємні) називається продуктивною, якщо для будь-яких векторів X і Y з невід'ємними компонентами існує розв'язок рівняння (4.3).

При цьому модель Леонтьєва називають продуктивною. Систему (4.3) можна записати у вигляді:

$$X = \left(E - A(t)\right)^{-1} Y(t), \tag{4.4}$$

де $\left(E - A(t)\right)^{-1}$ – матриця коефіцієнтів повних потреб у випуску продукції для одержання одиниць відповідних видів кінцевої продукції. Зміст коефіцієнтів b_{ij}, що утворюють матрицю $B = \left(E - A(t)\right)^{-1}$ полягає у тому, що вони характеризують об'єми виробництва продукції i, необхідні для одержання одиниці кінцевої продукції виду j.

Система (3.4) має розв'язок, якщо матриця $\left(E - A(t)\right)$ є невиродженою.

Відповідність між статичною і динамічною моделями міжгалузевого балансу для кожного t встановлюють за допомогою матричного рівняння:

$$Y(t) = B(t)\frac{dX(t)}{dt} + C(t). \tag{4.5}$$

У статичних моделях усі залежності стосуються одного моменту або часового періоду. Динамічні моделі характеризують зміни з часом економічних процесів.

Оскільки $\dfrac{dX(t)}{dt} = \left(E - A(t)\right)^{-1}\dfrac{dY(t)}{dt}$, то замість (4.2) можна досліджувати систему диференціальних рівнянь:

$$Y(t) = B(t)\left(E - A(t)\right)^{-1}\frac{dY(t)}{dt} + C(t), \tag{4.6}$$

де $B(t)\left(E - A(t)\right)^{-1}$ – матриця коефіцієнтів повного приросту капіталомісткості, тобто повних витрат виробничого нагромадження на одиничні прирости елементів використовуваного національного доходу.

Припустимо, що $A(t)$ – матриця продукції. У подальшому аналізі вважатимемо матрицю $A(t)$ нерозкладною, а матрицю $B(t)$ – невиродженою.

Означення 4.2. Матриці A називаються розкладними, якщо одночасною перестановкою рядків і стовпців їх можна звести до вигляду

$$A = \begin{bmatrix} A_{11} & * \\ 0 & A_{22} \end{bmatrix},$$

де A_{11}, A_{22} - квадратні блоки із ненульових елементів;

0 – блок із нульових елементів;

* – блок, елементи якого можуть набувати довільних значень.

Наслідок 4.1. Блок 0 не може складатися лише з одного елемента.

Означення 4.3. Матриця A називаються нерозкладною, якщо для неї не існує таких одночасних перестановок рядків та стовпців, які зводили б її до розкладної форми (навіть не зважаючи на наявність великої кількості нульових елементів).

Тоді $\left(E - A(t)\right)^{-1} > E + A(t),\ B(t)\left(E - A(t)\right)^{-1} > B(t).$

Розглянемо випадок, коли матриці $B(t)$ і $B\left(t\right)\left(E - A(t)\right)^{-1}$ включають не все виробниче нагромадження, а тільки нагромадження основних виробничих фондів.

На перший погляд, ці припущення неприпустимо штучні, оскільки дійсні матриці $A(t)$, як правило, розкладні, а матриці $B(t)$ мають нульові рядки (зокрема, за галузями, що виробляють тільки предмети споживання). Але після зведення системи (4.6) до рівнянь тільки для фондоутворюючих галузей обидва припущення стають цілком правомірними.

Очевидно, що економічне значення мають тільки розв’язки $X\left(t\right) > 0$.

В моделі без зовнішньої торгівлі може бути накладено на вектор $Y\left(t\right)$ аналогічну вимогу.

Зосередимо тепер увагу на задачі розв’язання матричного рівняння (4.3).

Ввівши для матриць в (4.3) позначення $G(t) = E - A(t)$, одержимо еквівалентне матричне рівняння:

$$G(t)X(t) = Y(t), \tag{4.7}$$

де $G(t)$- матриця розміру $n \times n$, елементи якої поліноми від t. Припустимо, що значення для t і коефіцієнтів многочленів беруться з деякого поля F так: елементи матриці $G(t)$ обчислюються для деякого часткового значення t, наприклад $t = t_0 \in F^{n \times m}$. $Y(t)$- вектор $(y_1(t), y_2(t), \ldots, y_n(t))^T$. Матриця $G(t)$ і вектор $Y(t)$ мають степінь l, тобто

$$G(t) = \sum_{i=0}^{l} G_i t^i,\ Y(t) = \sum_{i=0}^{l} Y_i t^i,\ \left(i = \overline{1,n}\right).$$

Тоді розв'язок системи природно шукати у вигляді відношення двох поліномів з невідомими коефіцієнтами:

$$X(t) = \sum_{j=0}^{nl} t^j X_{nl-j} \Big/ \sum_{j=0}^{nl} t^j z_{nl-j},$$

де $X_0, X_1, \ldots, X_{\ln}$ - вектори розмірності r, а $z_0, z_1, \ldots, z_{\ln}$ - скалярні величини. Тоді систему (4.7) можна записати так:

$$G(\lambda)X(\lambda) = \left(\lambda^l G_0 + \lambda^{l-1} G_1 + \lambda^{l-2} G_2 + \ldots + \lambda^2 G_{l-2} + \lambda^1 G_{l-1} + G_l\right)\left(\lambda^{nl} X_0 +\right.$$

$$\left.\lambda^{nl-1} X_1 + \lambda^{nl-2} X_2 + \ldots + \lambda^2 X_{nl-2} + \lambda^1 X_{nl-1} + X_{nl}\right) = \left(\lambda^l Y_0 + \lambda^{l-1} B Y_1 + \lambda^{l-2} B Y_2 +\right.$$

$$\left.+ \ldots + \lambda^2 Y_{l-2} + \lambda^1 Y_{l-1} + Y_l\right)\left(\lambda^{nl} z_0 + \lambda^{nl-1} z_1 + \lambda^{nl-2} z_2 + \ldots + \lambda^2 z_{nl-2} + \lambda^1 z_{nl-1} + z_{nl}\right).$$

Згрупувавши члени в лівій і правій частині рівності при однакових степенях λ, її можна переписати у вигляді

$$\lambda^{(n+1)l} G_0 X_0 + \lambda^{l(n+1)-1}(G_1 X_0 + G_1 X_0) + \lambda^{l(n+1)-2}(G_0 X_2 + G_1 X_1 + G_2 X_0) +$$

$$+\lambda^{l(n+1)-3}(G_0 X_3 + G_1 X_2 + G_2 X_1 + G_3 X_0) + \ \ldots\ + \lambda^{l(n+1)-l} \sum_{s=0}^{l} G_s X_{l-s} +$$

$$+\lambda^{l(n+1)-(l+1)} \sum_{s=0}^{l} G_s X_{l+1-s} + \ldots + \lambda^{l(n+1)-nl} \sum_{s=0}^{l} G_s X_{nl-s} + \ldots + \lambda(G_{l-1} X_{nl} + G_l X_{nl-1}) +$$

$$+G_l X_{nl} = \lambda^{(n+1)l} Y_0 z_0 + \lambda^{l(n+1)-1}(Y_1 z_0 + Y_1 z_0) + \lambda^{l(n+1)-2}(Y_0 z_2 + Y_1 z_1 + Y_2 z_0) +$$

$$+\lambda^{l(n+1)-3}(G_0 X_3 + G_1 X_2 + G_2 X_1 + G_3 X_0) + \ \ldots\ + \lambda^{l(n+1)-l} \sum_{s=0}^{l} G_s X_{l-s} +$$

$$+\lambda^{l(n+1)-(l+1)} \sum_{s=0}^{l} G_s X_{l+1-s} + \ldots + \lambda^{l(n+1)-nl} \sum_{s=0}^{l} G_s X_{nl-s} + \ldots + \lambda(G_{l-1} X_{nl} + G_l X_{nl-1}) +$$

$$+G_l X_{nl} = \lambda^{(n+1)l} Y_0 z_0 + \lambda^{l(n+1)-1}(Y_1 z_0 + Y_1 z_0) + \lambda^{l(n+1)-2}(Y_0 z_2 + Y_1 z_1 + Y_2 z_0) +$$
$$+\lambda^{(n+1)l} G_0 X_0 + \lambda^{l(n+1)-1}(G_1 X_0 + G_1 X_0) + \lambda^{l(n+1)-2}(G_0 X_2 + G_1 X_1 + G_2 X_0) +$$
$$+\lambda^{l(n+1)-3}(Y_0 z_3 + Y_1 z_2 + Y_2 z_1 + Y_3 z_0) + \quad . \ . \ . + \lambda^{l(n+1)-l} \sum_{s=0}^{l} Y_s z_{l-s} +$$
$$+\lambda^{l(n+1)-(l+1)} \sum_{s=0}^{l} Y_s z_{l+1-s} + \ldots + \lambda^{l(n+1)-nl} \sum_{s=0}^{l} Y_s z_{nl-s} + \ldots + \lambda(Y_{l-1} z_{nl} + Y_l z_{nl-1}) +$$
$$+\lambda(Y_{l-1} z_{nl} + Y_l z_{nl-1}) + Y_l z_{nl}.$$

Оскільки $A(t) \in F^{n \times n}$, а $Y(t) \in F^{n}$, то в загальному випадку, прирівнявши коефіцієнти при однакових степенях, одержимо числову систему $m\left(l + s + 1\right)$ рівнянь з $m\left(s + 1\right)$ невідомими x_{ij} і $\left(s + 1\right)$ невідомими z_j.

$$\begin{cases} \mathrm{G}_0 X_0 - Y_0 z_0 = 0; \\ \mathrm{G}_0 X_1 + G_1 X_0 - \left(Y_0 z_1 + Y_1 z_0\right) = 0; \\ \mathrm{G}_0 X_2 + G_1 X_1 + G_2 X_0 + \left(Y_0 z_2 + Y_1 z_1 + Y_2 z_0\right) = 0; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \sum_{s=0}^{l} G_s X_{p-s} - \sum_{s=0}^{l} Y_s z_{p-s} = 0; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \mathrm{G}_{l-1} X_{nl} + G_l X_{nl-1} - \left(Y_{l-1} z_{nl} + Y_l z_{nl-1}\right) = 0; \\ A_l X_{nl} - Y_l z_{nl} = 0. \end{cases} \tag{4.8}$$

Для розв’язання даної системи можуть бути застосовані кілька алгоритмів, але предметом розгляду буде обчислювальна схема розрізання, запропонована в [62].

Схема розрізання. Розіб’ємо матрицю одержаної системи (4.8) на блоки так:

$$\left\| \begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array} \right\|,$$

де C_{11} - квадратна матриця розміром $\left(n + 1\right)\left(l + 2\right)$, а C_{12}, C_{21}, C_{22} - прямокутні матриці відповідних розмірів.

Припустимо, що мінор C_{11} відмінний від нуля (для цього достатньо, щоб виконувалась умова $\det G_0 \neq 0$). Надамо z_{nl} якого-небудь значення, наприклад, $z_{nl} = 1$, і запишемо неоднорідну систему:

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \begin{pmatrix} X \\ Z \end{pmatrix} = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix}, \tag{4.9}$$

Її розв'язки з точністю до множника збігаються з розв'язками системи (4.8).

У такому випадку на основі теореми Шура [95] вектор Z може бути визначений із системи рівнянь

$$\left(C_{22} - C_{21}C_{11}^{-1}C_{12}\right)Z = Y_2 - C_{21}C_{11}^{-1}Y_1. \tag{4.10}$$

Після того як обчислено Z, можна відразу переходити до знаходження невідомого вектора X із системи рівнянь

$$C_{11}X = Y_1 - C_{12}Z. \tag{4.11}$$

Отже, обчислення невідомих X та Z у системі (4.9) можна звести до розв'язання двох систем (4.10) і (4.11) меншого порядку. Проте для цього необхідно виконати кілька проміжних матричних операцій. Тому зупинимося більш детально на виконанні кожної ланки алгоритму та оцінці обчислювальних затрат даної схеми на кожному кроці реалізації.

1. Обчислення $C_{11}^{-1}C_{12}$ **та** $C_{11}^{-1}Y_1$. Обчислимо спочатку добуток $C_{11}^{-1}Y_1$. Введемо позначення згідно з [12]. Для цього достатньо на основі [62] для всіх $i = \overline{0,2nl}$ обчислити розв'язки систем вигляду:

$$\begin{pmatrix} G_0 & 0 & 0 & \dots & 0 \\ G_1 & G_0 & 0 & \dots & 0 \\ G_2 & G_1 & G_0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & G_0 \end{pmatrix} \cdot \begin{pmatrix} W_0^{(i)} \\ W_1^{(i)} \\ W_2^{(i)} \\ \dots \\ W_{nl}^{(i)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \dots \\ 0 \end{pmatrix}. \tag{4.12}$$

Матриця C_{11} – блочно-трикутна, квадратна і, крім того, є клітково-теплiцевого типу. Перелічені властивості дають можливість суттєво знизити обчислювальні затрати. Справді, для обчислення $W_j^{(i)}\left(i = 2,3,\ldots,nl;\ j = 0,1,2,\ldots,nl\right)$ досить знайти розв'язок системи

$$\begin{pmatrix} G_0 & 0 & 0 & \ldots & 0 & \ldots & 0 \\ G_1 & G_0 & 0 & \ldots & 0 & \ldots & 0 \\ G_2 & G_1 & G_0 & \ldots & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ G_l & G_{l-1} & G_{l-2} & \ldots & G_0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 0 & \ldots & G_0 \end{pmatrix} \times \begin{pmatrix} W_0^{(1)} \\ W_1^{(1)} \\ W_2^{(1)} \\ \ldots \\ W_l^{(1)} \\ \ldots \\ W_{nl}^{(1)} \end{pmatrix} = \begin{pmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \\ \ldots \\ Y_l \\ 0 \end{pmatrix}. \tag{4.13}$$

Тоді за співвідношеннями:

$$\begin{cases} W_0^{(i)} = W_1^{(i)} = \ldots = W_{i-1}^{(i)} = 0\ ; \\ W_i^{(i)} = W_0^{(0)}\ ; \\ W_{i+1}^{(i)} = W_1^{(0)}\ ; \\ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ W_{nl}^{(i)} = W_{nl-1}^{(0)}. \end{cases} \tag{4.14}$$

визначається решта невідомих $W_j^{(i)}\left(i=2,3,\ldots,nl;\ j=0,1,2,\ldots,nl\right)$. Що стосується системи (4.13), її розв’язки можна знаходити за формулами [62]:

$$\begin{cases} W_i^{(1)} = G_0^{-1}\left[Y_i - \sum\limits_{j=1}^{i} G_j W_{i-j}^{(1)}\right], & \left(i = \overline{1,l}\right); \\ W_i^{(1)} = G_0^{-1}\left[-\sum\limits_{j=1}^{i} G_j W_{i-j}^{(1)}\right], & \left(i = \overline{l+1;nl}\right). \end{cases} \tag{4.15}$$

Для реалізації першого кроку, а саме для знаходження G_0^{-1}, необхідно на комп’ютері виконати $4/3n^2$ арифметичних операцій. Для обчислення всіх добутків $G_0^{-1}G_j$ треба виконати $n^3 l$ операцій додавання та множення. Множення вектора порядку n на квадратну матрицю розмірності $n \times n$ вимагає виконання n^2 операцій. Тому для обчислення кожного $W_j^{(1)}$ потрібно

$n^2 l$ арифметичних дій. Отже, для визначення усіх $W_j^{(1)}$ $\left(i = 1, 2, \ldots, nl\right)$ слід виконати $l^2 n^3$ операцій.

Результатом обчислення добутку $C_{11}^{-1} C_{12}$ є матриця вигляду:

$$\begin{pmatrix} W_0^{(1)} & 0 & 0 & \ldots & 0 & \ldots & 0 \\ W_1^{(1)} & W_0^{(1)} & 0 & \ldots & 0 & \ldots & 0 \\ W_2^{(1)} & W_1^{(1)} & W_0^{(1)} & \ldots & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ W_l^{(1)} & W_{l-1}^{(1)} & W_{l-2}^{(1)} & \ldots & W_0^{(1)} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ W_{nl}^{(1)} & W_{nl-1}^{(1)} & W_{nl-2}^{(1)} & \ldots & W_{nl-l}^{(1)} & \ldots & W_{l+1}^{(1)} \end{pmatrix}. \quad (4.16)$$

Результатом добутку $C_{11}^{-1} Y_1$ є вектор: $\left(0, \ldots, 0, W_0^{(1)}, W_1^{(1)}, \ldots, W_l^{(1)}\right)^T$.

Відзначимо, що $W_i^{(1)} \left(i = \overline{0, nl}\right)$ – це вектори розмірності n .

2. Обчислення $C_{21}\left(C_{11}^{-1} Y_1\right)$ **та** $C_{21}\left(C_{11}^{-1} C_{12}\right)$. Спочатку знайдемо кількість операцій для обчислення добутку матриць $C_{21}\left(C_{11}^{-1} Y_1\right)$.

Оскільки $\left(C_{11}^{-1} Y_1\right)$ – це вектор розмірності $n^2 l$, а матриця C_{21}– розмірності $nl \times nl$, то для обчислення добутку потрібно виконати $n^2 l^2$ множень та додавань.

Схема обчислення добутку $C_{21}\left(C_{11}^{-1} C_{12}\right)$ потребує більш детального дослідження. Розглянемо добуток двох наступних матриць:

$$\begin{bmatrix} 0 & \ldots & 0 & G_l & G_{l-1} & \ldots & G_3 & G_2 & G_1 \\ 0 & \ldots & 0 & 0 & G_l & \ldots & G_4 & G_3 & G_2 \\ 0 & \ldots & 0 & 0 & 0 & \ldots & G_5 & G_4 & G_3 \\ \ldots & \ldots & 0 & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & G_l & G_{l-1} \\ 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & G_l \end{bmatrix} \begin{bmatrix} W_0^{(1)} & 0 & 0 & \ldots & 0 & \ldots & 0 \\ W_1^{(1)} & W_0^{(1)} & 0 & \ldots & 0 & \ldots & 0 \\ W_2^{(1)} & W_1^{(1)} & W_0^{(1)} & \ldots & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ W_l^{(1)} & W_{l-1}^{(1)} & W_{l-2}^{(1)} & \ldots & W_0^{(1)} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ W_{nl}^{(1)} & W_{nl-1}^{(1)} & W_{nl-2}^{(1)} & \ldots & W_{nl-l}^{(1)} & \ldots & W_{l+1}^{(1)} \end{bmatrix}. \quad (4.17)$$

Результат скалярного добутку першого рядка першої матриці на перший стовпець другої матриці – це діагональні елементи матриці добутку (4.17). Скалярний добуток кожного рядка, починаючи з другого, першої матриці та першого стовпця другої матриці дає усі піддіагональні елементи. Всього для реалізації таких обчислень потрібно виконати $1/2n^2l^2$ операцій множення та додавання.

За аналогією для знаходження всіх наддіагональних елементів добутку (4.17) достатньо перемножити перший рядок першої матриці $\begin{pmatrix} 0 & \dots & 0 & G_l & G_{l-1} & \dots & G_3 & G_2 & G_1 \end{pmatrix}$ на кожний стовпець другої матриці, починаючи з другого. Виконання усієї процедури потребує $1/2n^3l^2$ операцій множення та додавання.

Виконавши описані операції, отримаємо матрицю G^* розміру $nl \times nl$

$$G^* = \begin{bmatrix} G^*_{11} & G^*_{12} & \dots & G^*_{1,l} \\ G^*_{21} & G^*_{22} & \dots & G^*_{2,l} \\ \dots & \dots & \dots & \dots \\ G^*_{l,1} & G^*_{l,2} & \dots & G^*_{l,l} \end{bmatrix}.$$

3. Визначення невідомих Z. Для знаходження вектора Y із системи (3.10), спочатку треба обчислити значення виразів $C_{22} - C_{21}\left(C_{11}^{-1}C_{12}\right)$ та $Y_2 - C_{21}\left(C_{11}^{-1}Y_1\right)$. На попередньому кроці реалізації алгоритму було встановлено, що добуток $C_{21}\left(C_{11}^{-1}C_{12}\right) = A^*$ – це матриця розмірності $nl \times nl$, а добуток $C_{21}\left(C_{11}^{-1}B_1\right) = Y^*$ – вектор порядку nl. Отже, для обчислення різниці $C_{22} - \left(C_{21}C_{11}^{-1}C_{12}\right)$ потрібно виконати n^2l^2 операцій додавання, а для знаходження різниці $Y_2 - \left(C_{21}C_{11}^{-1}Y_1\right)$ – nl додавань.

Для визначення вектора Z nl-го порядку з nl невідомими із системи (4.14) потрібно виконати $O\left(nl\right)^b$ арифметичних операцій. При використанні алгоритму відсічених систем $\beta = 3$, а для швидких схем типу Штрассена $\beta = \log_2 7$ і менше [29] для великих значень n.

4. Визначення невідомих X . На цьому етапі для визначення вектора X із системи (4.15) треба спочатку обчислити праву частину цієї системи Y_1 - $C_{12}Z$. Тому для множення добутку $C_{12}Z$ необхідно виконати $n^2l^2/2$ операцій множення і $n^2l^2/2$ – додавання. Розв'язок системи (3.15) обчислюватимемо за формулами [62], на що необхідно затратити з точністю до головного члена по $n^3l^2/2$ адитивних та мультиплікативних операцій.

Отже, для реалізації всіх чотирьох етапів алгоритму потрібно виконати $2l^2n^3 + O(nl)^b$ на комп'ютерній системі арифметичних операцій.

Швидка версія схеми розрізання. Описаний алгоритм розрізання для лінійних алгебричних систем з λ-матрицями допускає й прискорену, більш економічну для великих порядків n обчислювальну схему [62].

Так само, як і в основному алгоритмі розрізання, розіб'ємо матрицю G одержаної системи числових алгебричних рівнянь на блоки так:

$$\begin{pmatrix}
W & & & & & I & & & & \\
W & W & & C_{11} & & I & I & & C_{12} & \\
W & W & W & & & I & I & I & & \\
W & W & W & W & & I & I & I & I & \\
 & W & W & W & W & & I & I & I & I \\
 & & W & W & W & & & I & I & I \\
 & C_{21} & & W & W & & C_{22} & & I & I \\
 & & & & W & & & & & I
\end{pmatrix}.$$

Тут C_{11} - як і раніше квадратна матриця розміру $(l+2)m$, а C_{21}, C_{12} і C_{22} - прямокутні матриці відповідних розмірів. Так само запишемо систему:

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}\begin{pmatrix} X \\ Z \end{pmatrix} = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix}. \tag{4.18}$$

Знову за формулою Шура [94] вектор Z можна визначити з системи рівнянь:

$$\left(C_{22}-C_{21}C_{11}^{-1}C_{12}\right)Z=Y_2-C_{21}C_{11}^{-1}Y_1. \tag{4.19}$$

Після того, як визначено Z, можна відразу ж переходити до обчислення X з системи

$$C_{11}X=Y_1-C_{12}Z. \tag{4.20}$$

Тобто, обчислення невідомих X та Z у системі (4.18) можна звести до розв'язання двох систем (4.19) і (4.20) меншого порядку.

Зупинимося тепер на деталізації виконання кожної ланки прискореного алгоритму та оцінці обчислювальних затрат даної схеми на кожному кроці реалізації.

Етап 1. *(Обчислення $C_{11}^{-1}C_{12}$ та $C_{11}^{-1}Y_1$).* Розглянемо спочатку процес обчислення добутку $C_{11}^{-1}C_{12}$. Як відомо [62], для цього досить для всіх $i=1,2,\dots nl$ обчислити розв'язки систем виду

$$\begin{pmatrix} A_0 & 0 & 0 & \dots & 0 & \dots & 0 \\ A_1 & G_0 & 0 & \dots & 0 & \dots & 0 \\ A_2 & G_1 & G_0 & \dots & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ A_l & G_{l-1} & G_{l-2} & \dots & G_0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & \dots & G_0 \end{pmatrix} \begin{pmatrix} W_0^{(i)} \\ W_1^{(i)} \\ W_2^{(i)} \\ \dots \\ W_l^{(i)} \\ \dots \\ W_{nl}^{(i)} \end{pmatrix} = \begin{pmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \\ \dots \\ Y_l \\ 0 \end{pmatrix}. \tag{4.21}$$

За характером заповнення D_{11} – блочно-трикутна квадратна матриця, причому клітково-тепліцевого типу. З урахуванням цієї обставини, неважко зробити висновок, що насправді достатньо розв'язати лише систему:

$$\begin{pmatrix} A_0 & 0 & 0 & \ldots & 0 & \ldots & 0 \\ A_1 & G_0 & 0 & \ldots & 0 & \ldots & 0 \\ A_2 & G_1 & G_0 & \ldots & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ A_l & G_{l-1} & G_{l-2} & \ldots & G_0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 0 & \ldots & G_0 \end{pmatrix} \begin{pmatrix} W_0^{(1)} \\ W_1^{(1)} \\ W_2^{(1)} \\ \ldots \\ W_l^{(1)} \\ \ldots \\ W_{nl}^{(1)} \end{pmatrix} = \begin{pmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \\ \ldots \\ Y_l \\ 0 \end{pmatrix}. \qquad (4.22)$$

Після обчислення $W_j^{(1)}$ *(j = 1,2,..nl)* решту невідомих $W_j^{(i)}$ *(i=2,3,...nl; j=0,1,2,... nl)* можна визначити зі співвідношень

$$\left.\begin{array}{l} W_o^{(i)} = W_1^{(i)} = \ldots = W_{i-1}^{(i)} = 0; \\ W_i^{(i)} = W_o^{(i)}; \\ W_{i+1}^{(i)} = W_1^{(i)}; \\ \ldots\ldots\ldots\ldots\ldots \\ W_{nl}^{(i)} = W_{nl-i}^{(i)}. \end{array}\right\}. \qquad (4.23)$$

Знайдемо тепер обчислювальні затрати, потрібні для розв'язання системи (4.22). Як було вже зазначено, її матриця є клітково-тепліцевою і, крім того, блочно-трикутною та ще й стрічковою. Для її розв'язання опишемо аналог відповідного методу Є.Тиртишнікова [95], застосувавши його для конкретного випадку.

Введемо до розгляду матриці

$$U = \begin{pmatrix} G_0 & 0 & 0 & \ldots & 0 \\ G_1 & G_0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ G_l & G_{l-1} & G_{l-2} & \ldots & G_0 \end{pmatrix}; \quad V = \begin{pmatrix} 0 & G_l & G_{l-1} & \ldots & G_0 \\ 0 & 0 & G_l & \ldots & G_1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 0. \end{pmatrix}.$$

Тоді G можна записати як блочно-дводіагональну матрицю

$$G = \begin{pmatrix} U & 0 & 0 & \ldots & 0 & 0 \\ V & U & 0 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & V & U \end{pmatrix}. \qquad (4.24)$$

Відповідно до (4.24) введемо також наступні представлення для векторів-стовпців W та Y :

$$W = \begin{pmatrix} W^{(1)} \\ W^{(2)} \\ \dots \\ W^{(n)} \end{pmatrix}; \quad \mathrm{Y} = \begin{pmatrix} Y^{(1)} \\ Y^{(2)} \\ \dots \\ Y^{(n)} \end{pmatrix}.$$

Беручи до уваги (4.24), можна записати:

$$\left.\begin{array}{l} W^{(1)} = U^{-1}B^{(1)}; \\ W^{(2)} = Y^{(2)} - U^{-1}VW^{(1)}; \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ W^{(n)} = Y^{(n)} - U^{-1}VW^{(n-1)}. \end{array}\right\}. \tag{4.25}$$

За введеними позначеннями бачимо, що матриця U є нижньою трикутною, тому можна зосередити увагу на алгоритмі обчислення елементів U^{-1}. Як відомо [29], обернена матриця для лівої тепліцевої матриці залишається тепліцевою лівою трикутною, тобто має вигляд:

$$U^{-1} = \begin{pmatrix} G_0^{-1} & 0 & 0 & \dots & 0 \\ G_1 & G_0^{-1} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ G_l^{-1} & G_{l-1}^{-1} & G_{l-2}^{-1} & \dots & G_0^{-1} \end{pmatrix}. \tag{4.26}$$

Припустимо тепер, що $1+1$ є степенем двійки. Тоді процес знаходження U^{-1} легко зводиться до обертання її провідної підматриці $U_{l/2}$, яка має вдвічі менший порядок. Дійсно, згідно з (4.26) знаходимо:

$$U_{l/2}^{-1} = \begin{pmatrix} G_0^{-1} & 0 & 0 & \dots & 0 \\ G_1 & G_0^{-1} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ G_{l/2}^{-1} & G_{l/2-1}^{-1} & G_{l/2-2}^{-1} & \dots & G_0^{-1} \end{pmatrix}. \tag{4.27}$$

Матриця $U_{l/2}^{-1}$ є провідною підматрицею матриці U^{-1}. Вона має половину елементів, які визначають матрицю U^{-1}. Щоб знайти другу половину, розглянемо таке співвідношення:

$$\left.\begin{aligned}&\begin{pmatrix} G^{(-1)}_{l/2+1}\\ G^{(-1)}_{l/2+2}\\ \ldots\\ G^{(-1)}_{l}\end{pmatrix}=\begin{pmatrix} G^{-1}_{0} & 0 & 0 & \ldots & 0\\ G_{1} & G^{-1}_{0} & 0 & \ldots & 0\\ \ldots & \ldots & \ldots & \ldots & \ldots\\ G^{-1}_{l/2} & G^{-1}_{l/2-1} & G^{-1}_{l/2-2} & \ldots & G^{-1}_{0}\end{pmatrix}\times\\ &\times\begin{pmatrix} G_{l/2+1} & G_{l/2} & G_{l/2-1} & \ldots & G_{1}\\ G_{l/2+2} & G_{l/2+1} & G_{l/2} & \ldots & G_{2}\\ \ldots & \ldots & \ldots & \ldots & \ldots\\ G_{l} & G_{l-1} & G_{l-2} & \ldots & G_{l/2+1}\end{pmatrix}\begin{pmatrix} G^{(-1)}_{0}\\ G^{(-1)}_{1}\\ \ldots\\ G^{(-1)}_{l/2}\end{pmatrix}\end{aligned}\right\}. \tag{4.28}$$

Відповідно до (4.27) і (4.28) обертання тепліцевої трикутної матриці порядку l зводиться до розв'язання аналогічної задачі для матриці порядку $l\ /\ 2$ і виконання множення на блочний вектор тепліцевих матриць порядку $l\ /\ 2$. Тоді одне таке множення можна реалізувати за допомогою $3n^2$ дискретних перетворень Фур'є, тобто за допомогою $\frac{3}{2}n^2\, l\log_2 l$ операцій множення та $\frac{3}{2}n^2 l\log_2 l$ операцій додавання [62].

Отже, в запропонованому рекурсивному алгоритмі число операцій множення визначається за формулою $Q = 3n^2 l\log_2 l + \frac{l}{2}\log_2\frac{l}{2} + \frac{l^2}{2}\log_2\frac{l^2}{2} + \ldots$,

що дорівнює $2l\log_2(n\ /\ 2) + 2$. Тому $Q = 6n^2 l\log_2 l$. Число ж операцій додавання-віднімання вдвічі більше.

Відзначимо, що коли l не є степенем двійки, то можна вважати U провідною підматрицею деякої трикутної тепліцевої матриці порядку, який дорівнює степені двійки. Знайшовши обернену їй, знайдемо також U^{-1}.

Отже, можна надалі вважати, що l є степенем двійки. Тоді, відповідно до щойно розглянутого алгоритму, на знаходження U^{-1} потрібно $6n^2 l\log_2 l$ множень і $12n^2 l\log_2 l$ додавань. Згідно з (4.13), обчислення W пов'язане з реалізацією $2l + 1$ множень на “блочний вектор” тепліцевих матриць блочного порядку не вище l, серед яких лише 4 різних. У такому разі потрібно з точністю до головного члена $4n^3 \log_2 l + 12nl\log_2 l$ множень та $8n^3 \log_2 l + 24nl\log_2 l$ додавань.

З урахуванням (4.11), можна записати:

$$C_{11}^{-1}C_{12} = \begin{pmatrix} W_0^{(1)} & 0 & 0 & \dots & 0 & \dots & 0 \\ W_1^{(1)} & W_0^{(1)} & 0 & \dots & 0 & \dots & 0 \\ W_2^{(1)} & W_1^{(1)} & W_0^{(1)} & \dots & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ W_l^{(1)} & W_{l-1}^{(1)} & W_{l-2}^{(1)} & \dots & W_0^{(1)} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ W_{nl}^{(1)} & W_{nl-1}^{(1)} & W_{nl-2}^{(1)} & \dots & W_{nl-l}^{(1)} & \dots & W_{l+1}^{(1)} \end{pmatrix}. \tag{4.29}$$

Неважко переконатися, що

$$C_{11}^{-1}Z_1 = \begin{pmatrix} 0 & 0 & \dots & W_0^{(1)} & W_1^{(1)} & \dots & W_l^{(1)} \end{pmatrix}^T.$$

Причому, $W_i^{(1)}$ – це вектори розмірності *n*. Отже, для визначення $C_{11}^{-1}Y_1$ не потрібно виконувати додаткові обчислення.

Етап 2. (*Обчислення* $C_{21}(C_{11}^{-1}Y_1)$ *та* $C_{21}(C_{11}^{-1}C_{12})$). Обчислення матричного добутку $C_{21}(C_{11}^{-1}Y_1)$ можна виконати за попередньою схемою, затративши не більше n^2l^2 множень та додавань. Схему ж обчислення $C_{21}(C_{11}^{-1}C_{12})$, як показує більш ретельне дослідження, можна суттєво прискорити. Для цього добуток двох матриць

$$\begin{pmatrix} 0 & \dots & 0 & G_l & G_{l-1} & \dots & G_3 & G_2 & G_1 \\ 0 & \dots & 0 & 0 & G_l & \dots & G_4 & G_3 & G_2 \\ 0 & \dots & 0 & 0 & 0 & \dots & G_5 & G_4 & G_3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & G_l & G_{l1} \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & G_l \end{pmatrix} \begin{pmatrix} W_0^{(1)} & 0 & 0 & \dots & 0 & \dots & 0 \\ W_1^{(1)} & W_0^{(1)} & 0 & \dots & 0 & \dots & 0 \\ W_2^{(1)} & W_1^{(1)} & W_0^{(1)} & \dots & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ W_l^{(1)} & W_{l-1}^{(1)} & W_{l-2}^{(1)} & \dots & W_0^{(1)} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ W_{nl}^{(1)} & W_{nl-1}^{(1)} & W_{nl-2}^{(1)} & \dots & W_{nl-l}^{(1)} & \dots & W_{l+1}^{(1)} \end{pmatrix}$$

обчислюватимемо інакше, ніж у попередній схемі. Для множення стовпця з матричними компонентами

$$\begin{pmatrix} W_0^{(1)} & W_1^{(1)} & \dots & W_l^{(1)} & W_{l+1}^{(1)} & \dots & W_{l(n-1)-1}^{(1)} & \dots & W_{nl}^{(1)} \end{pmatrix}^T$$

на клітково-тепліцеву матрицю, записану в добутку зліва, використаємо швидке перетворення Фур'є. Для його реалізації потрібно виконати $4n^2 l \log_2 l$ операцій множення та $4n^2 l \log_2 l$ додавання. Внаслідок цього знайдемо всі піддіагональні елементи шуканої матриці.

За аналогією для знаходження всіх наддіагональних елементів добутку достатньо буде перемножити рядок

$$\begin{pmatrix} 0 & 0 & \ldots & 0 & A_l & A_{l-1} & \ldots & A_2 & A_1 \end{pmatrix} \tag{4.30}$$

на кожний стовпець другої матриці. Тоді, помноживши рядки (4.30) на другий стовпець другої матриці, знайдемо як проміжні величини елементи першої наддіагоналі. Помноживши цей рядок на третій стовпець, знайдемо заодно також всі елементи другої наддіагоналі і т. д. Отже, всього для виконання процедури із застосуванням швидкого перетворення Фур'є потрібно виконати $4n^2 l \log_2 nl$ операцій множення та $4n^2 l \log_2 nl$ додавання.

Тоді, виконавши описані операції, знайдемо матрицю A розміру $nl \times nl$ загального вигляду

$$C^* = \begin{pmatrix} C^*_{11} & C^*_{1,2} & \ldots & C^*_{1,l} \\ C^*_{2,1} & C^*_{2,2} & \ldots & C^*_{2,l} \\ \ldots & \ldots & \ldots & \ldots \\ C^*_{l,1} & C^*_{l,2} & \ldots & C^*_{l,l} \end{pmatrix},$$

яку використаємо для подальших обчислень.

Етап 3 (*визначення* Z). На цьому етапі спочатку потрібно обчислити величини $C_{22} - C_{21}(C_{11}^{-1}C_{12})$ та $Y_2 - C_{21}(C_{11}^{-1}Y_1)$. Як було показано на попередньому кроці реалізації алгоритму, $C_{21}(C_{11}^{-1}C_{12}) = A^*$ - це матриця $nl \times nl$, а $C_{21}(C_{11}^{-1}Y_1) = Y^*$ - вектор розмірності nl. Тому для обчислення $C_{22} - C_{21}(C_{11}^{-1}C_{12})$ досить виконати $n^2 l^2$ операцій додавання і для $Y_2 - C_{21}(C_{11}^{-1}Y_1)$ - nl додавань. Безпосередньо для визначення Z з системи nl-го порядку з nl невідомими потрібно витратити $O(nl)^b$ операцій. Застосовуючи алгоритм відсічених систем, одержуємо $b = 3$, для швидких схем типу Штрассена – $b = \log_2 7$, а для найновіших методів ще менше – $b = 2{,}3176$.

Етап 4 (*визначення* X). На даному кроці треба спочатку обчислити праву частину $Y_l - C_{12}Z$. Для її знаходження можна скористатися й попередньою схемою, затративши $n^2l^2 / 2$ операцій множення і таку ж кількість додавань. А от обчислення X можна провести й за схемою, описаною в пункті 2. Тоді затратимо $4n^2 l \log_2 l$ операцій множення та $8n^2 l \log_2 l$ додавання.

Отже, для повної реалізації алгоритму потрібно затратити $C_1(nl)^b + 4n^3 l \log_2 l + 4n^2 l \log_2 nl$ мультиплікативних операцій та $C_2(nl)^b + 8n^3 l \log_2 l + 8n^2 l \log_2 nl$ адитивних дій на ЕОМ.

Сталі C_1, C_2 та β залежать від обраного алгоритму розв'язання систем лінійних алгебричних рівнянь.

Тому, для великих n та l швидкий алгоритм розрізання стає значно ефективнішим порівнянно з основною схемою. Після розв’язання системи (4.7) за допомогою (4.5) встановлюється зв’язок між узагальненою статичною і динамічною моделями Леонтьєва.

Загальний розв’язок динамічної системи подамо у вигляді:

$$Y_1(t) = \sum_{l=1}^{l} d_l K_l e^{\lambda_l t}, \qquad (4.31)$$

де λ_l – корені характеристичного рівняння n - го порядку

$$\det\left(E - \lambda B(t)\left(E - A(t)\right)^{-1}\right) = 0. \qquad (4.32)$$

(λ_l – збігаються з величинами, оберненими до власних значень матриці $B(t)\left(E - A(t)\right)^{-1}$);

K_l - відповідні до λ_l, власні вектори матриці $B(t)\left(E - A(t)\right)^{-1}$, тобто нетривіальні розв’язки системи однорідних рівнянь

$$\left[E - \lambda_l B(t)\left(E - A^{-1}(t)\right)\right] K_l = 0. \qquad (4.33)$$

Деякі корені λ_l можуть виявитися комплексними. Разом з тим, будь-якому комплексному кореню відповідає спряжений до нього. Кожна пара комплексно спряжених коренів $\lambda = \alpha \pm i\beta$ подана парою доданків $Ce^{at}\cos bt + De^{at}\sin bt$, де C і D – сталі вектори розмірності n, що породжують коливання зі сталою частотою p та амплітудою e^{at}.

Величини d_l у формулі (4.31) є невизначеними сталими, що однозначно визначаються за початковою умовою $Y_1(0) = Y(0)$. Або

$$\sum_{l=1}^{n} d_l K_l = Y(0). \tag{4.34}$$

Ця система з n лінійних рівнянь відносно $d_1,\ldots,d_l$, яка має єдиний розв’язок.

У загальному випадку немає підстав розраховувати на те, що у розв'язку системи (4.33) відмінною від нуля буде єдина компонента d_l. Отже, у типовій ситуації єдина траєкторія системи (4.1), яка виходить з початкової точки $Y(0)$, виявляється комбінацією експонент, що зростають різними темпами. Останнє твердження вказує на істотну відмінність міжгалузевої моделі від її макроекономічного прототипу.

Однак, певна схожість розв'язків однорідних макроекономічної та міжгалузевої моделей зберігається. Це стає очевидним з наведених нижче міркувань.

Відповідно до зроблених припущень, матриця $B(E - A)^{-1}$ додатна. Тоді, відповідно до теореми О. Перрона [26; 51], вона має додатне власне число (корінь Фробеніуса–Перрона), що за абсолютною величиною більше, за усі інші власні числа цієї матриці, а також відповідний строго додатний власний вектор $\hat{K}$. Власні ж вектори, що відповідають відмінним від $\hat{S}$ власним значенням, з необхідністю мають компоненти різних знаків.

З теорії невід'ємних матриць відомо, що корінь Фробеніуса–Перрона знаходиться між максимальною і мінімальною сумами елементів стовпців матриці $B(E-A)^{-1}$. Позначимо через $\tilde{\beta}_j = \sum_{i \in J} b_{ij}$ суму елементів j-го стовпця цієї матриці, тобто повну капіталомісткість продукції j-ої галузі. Тоді

$$\min_j \tilde{\beta}_j \le \hat{S} \le \max_j \tilde{\beta}_j.$$

Разом з тим, параметр $\hat{\lambda} = \frac{1}{\hat{S}}$, що належить розв’язку (4.8) як показник експоненти, знаходиться в проміжку:

$$\min_j \frac{1}{\tilde{\beta}_j} \le \hat{\lambda} \le \max_j \frac{1}{\tilde{\beta}_j}. \tag{4.35}$$

Отже, показник експоненти є оберненою величиною до деякої середньої з повних галузевих капіталомісткостей. У випадку ж їхньої рівності $\tilde{\beta}_j = B_0, (j = 1,\ldots,n)$ $\hat{\lambda}$ збігається з $\hat{\lambda} = \frac{1}{B_0}$, що дає підставу називати величину $\hat{\lambda}$ технологічним темпом приросту в міжгалузевій динамічній моделі (4.2).

Траєкторія $Y_1(t)$, пов’язана з початковими умовами рівностями (4.33), є сумою експонент. Очевидно, що при $t \to \infty$ у сумі починає переважати доданок з максимальною (серед номерів l з $d_l \ne 0$) дійсною частиною λ_l. Можливі дві взаємовиключні ситуації:

1) домінуючою є експонента $e^{\hat{\lambda}t}$;
2) домінує інший доданок з темпом λ_l, відмінним від $\hat{\lambda}$.

У першому випадку темпи приросту продукції кожної галузі при $t \to \infty$ прямують до технологічного темпу зростання $\hat{\lambda}$, а гранична галузева структура національного доходу визначається пропорціями між компонентами власного вектора $\hat{K}$.

У другому випадку динаміка $Y_1(t)$ усе більше визначається власним вектором K_l, що відповідає власному значенню $\frac{1}{\lambda_l} \neq \hat{S}$ з матриці $B(E - A)^{-1}$. Такий власний вектор, як було зазначено вище, обов'язково має компоненти різних знаків. Тому при досить великих t у розв'язку $Y_1(t)$ неодмінно з'являються від'ємні компоненти і тим самим втрачається економічний зміст розв'язку. Ще раніше спадні компоненти з'являються у траєкторії

$$X_l(t) = \sum_{l=1}^{n} d_l (E - A)^{-1} K_l e^{\lambda_l t}, \tag{4.36}$$

що також суперечить припущенням моделі. Така ситуація високо ймовірна. Наприклад, матриця $B(E - A)^{-1}$ може мати додатне власне число $S_l < \hat{S}$. Тоді в сумі (4.34) з'явиться доданок виду $d_l K_l e^{\lambda_l t}$, де $\lambda_l = \frac{1}{S_l} > \hat{\lambda}$, а вектор K_l має компоненти різних знаків. Завдяки рівності $S_l K_l = B(E - A)^{-1} K_l$ вектор $(E - A)^{-1} K_l$, що входить у суму (3.36) при $e^{\lambda_l t}$, також має знакозмінні компоненти, внаслідок чого стають від'ємними деякі компоненти $\frac{dX_l}{dt}$.

Отже, розв'язок (4.31), в якому домінує доданок з темпом, відмінним від $\hat{\lambda}$, економічно неприйнятий.

Зазначеними особливостями розв'язків однорідного рівняння (4.7) є принципово відмінні риси міжгалузевої моделі (4.2), порівняно з її макроекономічним аналогом, де розв'язок втрачає допустимість тільки через надмірні вимоги до зростання споживання.

4.2. Матричне рівняння для динамічної моделі Леонтьєва

Розглянемо систему диференціальних рівнянь, яка виникає в динамічній моделі Леонтьєва

$$Y(t) = B(t)\left(E - A(t)\right)^{-1} \times \frac{dY(t)}{dt} + C(t). \tag{4.37}$$

Оскільки $\left(E - A(t)\right)^{-1} \times \frac{dY(t)}{dt} = \frac{dX(t)}{dt}$, то матимемо: $Y(t) = B(t)\frac{dX(t)}{dt} + C(t)$. Спростивши, отримаємо систему

$$\left(B(t,x) - E\right)Y(t,x) = C(t,x), \tag{4.38}$$

де $B(t,x)$ - квадратна матриця порядку n, а $C(t,x)$ - вектор $\left(C_{1,n+1}(t,x), C_{2,n+1}(t,x), \ldots, C_{n,n+1}(t,x)\right)^T$, елементи якого є поліномами степеня m вигляду

$$C_{ij} = \sum_{p=0}^{m} \sum_{s=0}^{p} d_s^{(p)} t_i^s x_j^{p-s}, \left(i = \overline{1,n+1};\ j = \overline{1,n}\right) \tag{4.39}$$

що залежить від рівновіддалених на Δt і Δx координат t і x деякої прямокутної області, $Y(t,x)$ - вектор

$$\left(y_{1,n+1}(t,x), y_{2,n+1}(t,x), \ldots, y_{n,n+1}(t,x)\right)^T.$$

Знайдемо розв'язок системи (4.37) за алгоритмом, розробленим для двовимірних $\lambda-$матриць. З цією метою скористаємося скінченно-різницевим підходом [11; 62], в якому був встановлений зв'язок [65] між елементами в рядку та між рядками матриці у випадку, коли елементами матриці a_{ij} є многочлени другого порядку. Узагальнюючи цей результат, одержимо також рекурентні формули для обчислення елементів довільного рядка матриці $B(t,x)$ системи (4.25).

За аналогією [62] введемо деякі позначення. Нехай $\Delta b_{j,i}$ – перша різниця між елементами j-го рядка матриці $B(t,x)$: $\Delta b_{j,i} = b_{j,i+1} - b_{j,i}$;

$\Delta^2 b_{j,i}$ – друга різниця. Якщо $b_{j,i+1}$ віддалений на проміжок $\Delta\vartheta$ як від $b_{j,i}$ так і від $b_{j,i+2}$, то величина даної різниці визначається таким співвідношенням: $\Delta^2 b_{j,i} = \Delta b_{j,i+1} - \Delta b_{j,i}$, а $\Delta b_{j,i+1} = b_{j,i+2} - b_{j,i+1}$. Аналогічно обчислюється третя різниця $\Delta^3 b_{j,i}$ і наступні. Різниці $m+1$ для полінома (4.39) дорівнюють нулю.

Отже, обчисливши один раз значення різниці $\Delta^m b_{j,i}$ і коригуючи попередні різниці $\Delta^k b_{j,i}\left(1 \le k \le m-1\right)$, визначимо елементи будь-якого рядка матриці $B\left(t,x\right)$ за наступними рекурентними формулами:

$$\begin{aligned}
&b_{j,i+1} = b_{j,i} + \Delta b_{j,i};\\
&\Delta b_{j,i+1} = \Delta b_{j,i} + \Delta^2 b_{j,i};\\
&\Delta^2 b_{j,i+1} = \Delta^2 b_{j,i} + \Delta^3 b_{j,i};\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&\Delta^{m-2} b_{j,i+1} = \Delta^{m-2} b_{j,i} + \Delta^{m-1} b_{j,i};\\
&\Delta^{m-1} b_{j,i+1} = \Delta^{m-1} b_{j,i} + \Delta^m b_{j,i}.
\end{aligned} \tag{4.40}$$

Враховуючи рівності (4.40), одержимо формулу для обчислення елементів рядка матриці $B\left(t,x\right)$ системи (4.38).

$$\begin{aligned}
b_{j,k} = b_{j,1} + \left(k-1\right)\Delta b_{j,1} + \frac{\left(k-1\right)\left(k-2\right)}{2!}\Delta^2 b_{j,1} +\\
+\frac{\left(k-1\right)\left(k-2\right)\left(k-3\right)}{3!}\Delta^3 b_{j,1} + \ldots + \frac{\left(k-1\right)\left(k-2\right)\ldots\left(k-m\right)}{m!}\Delta^m b_{j,1},
\end{aligned} \tag{4.41}$$

де множники при різницях $\Delta b_{j,1}, \Delta^2 b_{j,1}, \ldots, \Delta^m b_{j,1}$ є інтерполяційними коефіцієнтами Грегорі-Ньютона [62].

Варто зазначити, що немає необхідності окремо обчислювати кожний із коефіцієнтів, достатньо знайти лише множники, що стоять біля першої Δ та другої різниць Δ^2.

Інші множники одержимо автоматично з раніше обчислених на два порядки нижчих різниць. Формулу (4.28) запишемо у вигляді:

$$
\begin{aligned}
b_{j,k} = {} & b_{j,1} + (k-1)\Delta b_{j,1} + \Delta^2 b_{j,1}\sum_{i=1}^{k-2} i + \Delta^3 b_{j,1}\sum_{i=1}^{k-3} i\cdot q_{k-(i+1)}(\Delta) + \\
& + \Delta^4 b_{j,1}\sum_{i=1}^{k-4} i\cdot q_{k-(i+1)}(\Delta^2) + \Delta^5 b_{j,1}\sum_{i=1}^{k-5} i\cdot q_{k-(i+1)}(\Delta^3) + \ldots + \\
& + \Delta^l b_{j,1}\sum_{i=1}^{k-l} i\cdot q_{k-(i+1)}(\Delta^{l-2}) + \ldots + \Delta^m b_{j,1}\sum_{i=1}^{k-m} i q_{k-(i+1)}(\Delta^{m-2}),
\end{aligned}
\tag{4.42}
$$

де $q_{k-(i+1)}(\Delta^r)$, $\left(i = \overline{1, k-r}\right)$ коефіцієнти при різних Δ^r у числі $b_{j,k-(i+1)}$.

Зупинимося на застосуванні теорії різниць для розв'язання систем алгебричних рівнянь. Елементи рядка матриці $B(t,x)$ обчислюються за формулою (4.38). Введемо позначення:

$$
B = \begin{bmatrix} b_{1,1} \\ b_{2,1} \\ \ldots \\ b_{n,1} \end{bmatrix},\ \Delta B = \begin{bmatrix} \Delta b_{1,1} \\ \Delta b_{2,1} \\ \ldots \\ \Delta b_{n,1} \end{bmatrix},\ \Delta^2 B = \begin{bmatrix} \Delta^2 b_{1,1} \\ \Delta^2 b_{2,1} \\ \ldots \\ \Delta^2 b_{n,1} \end{bmatrix}, \ldots, \Delta^m B = \begin{bmatrix} \Delta^m b_{1,1} \\ \Delta^m b_{2,1} \\ \ldots \\ \Delta^m b_{n,1} \end{bmatrix},\ Y = \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{bmatrix}.
$$

Тоді систему (4.38) можна записати у вигляді:

$$
\begin{aligned}
& y_1(E - By_1) + y_2(E - (B + \Delta B)y_2) + y_3(E - (B + 2\Delta B + \\
& + \Delta^2 B)y^3 + y_4(E - (B - 3\Delta B + 3\Delta^2 B + \Delta^3 B)y^4) + \ldots + \\
& + y_m(E - y_m(\Delta^l B \sum_{i=1}^{n-l} i q_{n-(i+1)}(\Delta^{l-2}) = C + 3\Delta C + 3\Delta^2 C + \ldots + m\Delta^{m-1}C.
\end{aligned}
$$

Згрупувавши члени у лівій і правій частинах рівності при C, ΔC, $\Delta^2 C, \ldots, \Delta^m C$ і B, ΔB, $\Delta^2 B, \ldots, \Delta^m B$ та згрупувавши після цього відповідні коефіцієнти, одержимо систему:

$$
\begin{aligned}
& y_1 - By_1^2 + y_2(B + \Delta B)y_2^2 + y_3^2 - (B + 2\Delta B + \Delta^2 B)y^3 + \\
& y_4 - (B - 3\Delta B + 3\Delta^2 B + \Delta^3 B)y_4^2 + \ldots + y_m - \\
& - (\Delta^l B \sum_{i=1}^{n-l} i\cdot q_{n-(i+1)}(\Delta^{l-2})y_m^2 = C + 3\Delta C + 3\Delta^2 C + \ldots + m\Delta^{m-1}C.
\end{aligned}
$$

$$\begin{cases} y_1 + y_2 + y_3 + y_4 + \ldots + y_m + y_{m+1} = 1; \\ y_1^2 + y_2^2 + y_3^2 + y_4^2 + \ldots + y_m^2 + y_{m+1}^2 = 1^2; \\ y_2 + 2y_3 + 3y_4 + \ldots + (m-1)y_m + my_{m+1} + \ldots + (n-1)y_n = n; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_m + y_{m+1}\sum\limits_{i=1}^{2} iq_{n-1}\left(\Delta^{m-3}\right) + \ldots + y^m \sum\limits_{i=1}^{n-(m-1)} iq_{n-1}\left(\Delta^{m-3}\right) = \sum\limits_{i=1}^{n+1-(m-1)} iq_{n-1}\left(\Delta^{m-3}\right); \\ y_{m+1} + \ldots + y_m \sum\limits_{i=1}^{n-m} iq_{n-i}\left(\Delta^{m-2}\right) = \sum\limits_{i=1}^{n+1-m} iq_{n-1}\left(\Delta^{m-2}\right). \end{cases}$$

За наведеними позначеннями і з урахуванням порядку $n = m + 1$ систему (4.25) запишемо як систему числових лінійних алгебричних рівнянь:

$$\begin{cases} y_1 + y_2 + y_3 + y_4 + \ldots + y_m + y_{m+1} = 1; \\ y_2 + 2y_3 + 3y_4 + \ldots + (m-1)y_m + my_{m+1} = m + 1; \\ y_3 + 3y_4 + \ldots + y_m \sum\limits_{i=1}^{m-2} i + y_{m+1}\sum\limits_{i=1}^{m-1} i = \sum\limits_{i=1}^{m} i; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_m + y_{m+1}\sum\limits_{i=1}^{2} i \cdot q_{m+1-i}\left(\Delta^{m-3}\right) = \sum\limits_{i=1}^{3} i \cdot q_{m+1-i}\left(\Delta^{m-3}\right); \\ y_{m+1} = \sum\limits_{i=1}^{2} i \cdot q_{m+1-i}\left(\Delta^{m-2}\right), \end{cases} \tag{4.43}$$

відносно невідомих $y_i \quad (i = 1, 2, \ldots, m + 1)$. Очевидно, що дана система має не вироджену матрицю трикутного вигляду. Для її розв’язання на комп'ютері достатньо виконати по $(m + 1)^2 / 2$ операцій додавання і множення.

4.2.1. Розв'язання систем лінійних алгебричних рівнянь з m-вимірними λ-матрицями

Розглянемо систему лінійних алгебричних рівнянь, як m-вимірний аналог системи

$$\left(B\left(\lambda_1,\lambda_2,...,\lambda_m\right)-E\right)Y\left(\lambda_1,\lambda_2,...,\lambda_m\right)=C\left(\lambda_1,\lambda_2,...,\lambda_m\right), \qquad (4.44)$$

в якій $B\left(\lambda_1,\lambda_2,...,\lambda_m\right)$ – регулярна матриця розміру n ґ n з елементами многочленами степеня l. Права частина рівняння визначається як вектор

$$C\left(\lambda_1,\lambda_2,...,\lambda_m\right)=\left(c_{1,n+1}\left(\lambda_1,\lambda_2,...,\lambda_m\right),c_{2,n+1}\left(\lambda_1,\lambda_2,...,\lambda_m\right),...,c_{n,n+1}\left(\lambda_1,\lambda_2,...,\lambda_m\right)\right)^T$$

многочленів степеня l. Елементи системи (4.44) задаються формулами:

$$b_{i,j}\left(\lambda_1,\lambda_2,...,\lambda_m\right)=\sum_{k_1+k_2+...+k_m=0}^{l} b_{i,j(k_1k_2...k_m)}\lambda_1^{k_1}\lambda_2^{k_2}...\lambda_m^{k_m}\ \left(i=\overline{1,n};\ j=\overline{1,n+1}\right).$$

Випадок, коли $m = 1$, тобто C, B та Y залежать від однієї змінної λ, розглянуто в [62]. З огляду на це опишемо метод, що дозволяє звести розв'язання системи (4.31) до обчислення невідомих систем лінійних алгебричних рівнянь з числовими коефіцієнтами спеціального вигляду [11; 23].

4.2.2. Зведення системи алгебричних рівнянь з m-вимірними λ-матрицями до системи з числовими елементами

Поліноміальні матриці $B\left(\lambda_1,\lambda_2,...,\lambda_m\right),C\left(\lambda_1,\lambda_2,...,\lambda_m\right)$ подамо як матричні поліноми

$$B\left(\lambda_1,\lambda_2,...,\lambda_m\right)=\sum_{k_1+k_2+...+k_m=0}^{l}\lambda_1^{k_1}\lambda_2^{k_2}...\lambda_m^{k_m}B_{k_1k_2...k_m}$$

та

$$C\left(\lambda_1,\lambda_2,...,\lambda_m\right)=\sum_{k_1+k_2+...+k_m=0}^{l}\lambda_1^{k_1}\lambda_2^{k_2}...\lambda_m^{k_m}C_{k_1k_2...k_m}.$$

Розв'язок системи шукатимемо у вигляді відношення двох поліномів

$$Y\left(\lambda_1,\lambda_2,...,\lambda_m\right)=\frac{\sum_{k_1+k_2+...+k_m=0}^{l}\lambda_1^{k_1}\lambda_2^{k_2}...\lambda_m^{k_m}Y_{k_1k_2...k_m}}{\sum_{k_1+k_2+...+k_m=0}^{l}\lambda_1^{k_1}\lambda_2^{k_2}...\lambda_m^{k_m}Z_{k_1k_2...k_m}}, \qquad (4.45)$$

де $Y_{k_1k_2...k_m}$ - вектори розмірності n, $Z_{k_1k_2...k_m}$ - скалярні величини.

Невідомі $Y_{k_1k_2\ldots k_m}$ та $Z_{k_1k_2\ldots k_m}$ обчислимо методом невизначених коефіцієнтів. Враховуючи (4.45), систему (4.44) запишемо у вигляді:

$$\begin{gathered}\sum_{k_1+k_2+\ldots+k_m=0}^{l}\lambda_1^{k_1}\lambda_2^{k_2}\ldots\lambda_m^{k_m}Y_{k_1k_2\ldots k_m}\left(E-\sum_{k_1+k_2+\ldots+k_m=0}^{nl}\lambda_1^{k_1}\lambda_2^{k_2}\ldots\lambda_m^{k_m}B_{k_1k_2\ldots k_m}\right)=\\=\sum_{k_1+k_2+\ldots+k_m=0}^{l}\lambda_1^{k_1}\lambda_2^{k_2}\ldots\lambda_m^{k_m}Z_{k_1k_2\ldots k_m}\sum_{k_1+k_2+\ldots+k_m=0}^{nl}\lambda_1^{k_1}\lambda_2^{k_2}\ldots\lambda_m^{k_m}C_{k_1k_2\ldots k_m}\end{gathered}\tag{4.46}$$

Для встановлення загальних закономірностей розглянемо рівняння (4.46) більш детально:

$$[Y_{00..0}+\sum_{k_1=1}^{nl}\lambda_1^{k_1}Y_{k_1 0..0}+\sum_{k_2=1}^{nl}\lambda_2^{k_2}Y_{0k_2 0..0}+\sum_{k_m=1}^{nl}\lambda_m^{k_m}Y_{00..0k_m}+\sum_{k_1+k_2=2}^{nl}\lambda_1^{k_1}\lambda_2^{k_2}Y_{k_1k_2 0..0}+$$

$$+\ldots+\sum_{k_s+k_r=2}^{nl}\lambda_s^{k_s}\lambda_r^{k_r}Y_{0..0k_s 0..0k_r 0..0}+\ldots+\sum_{k_{m-1}+k_m=2}^{nl}\lambda_{m-1}^{k_{m-1}}\lambda_m^{k_m}Y_{0..0k_{m-1}k_m}+\ldots+$$

$$+\sum_{\substack{k_1+k_2+\ldots+k_t=t\\t<m}}^{nl}\lambda_1^{k_1}\lambda_2^{k_2}\ldots\lambda_t^{k_t}Y_{k_1k_2..k_t 0..0}+\sum_{\substack{k_s+k_r+\ldots+k_p=t\\t<m}}^{nl}\lambda_s^{k_s}\lambda_r^{k_r}\ldots\lambda_p^{k_p}Y_{0..0k_s..k_r..k_p 0..0}+\ldots+$$

$$+\sum_{\substack{k_{m-t}+\ldots+k_m=t\\t<m}}^{nl}\lambda_{m-t}^{k_{m-t}}\ldots\lambda_m^{k_m}Y_{0..0k_{m-t}k_m}+\ldots+\sum_{k_1+k_2+\ldots+k_m=m}^{nl}\lambda_1^{k_1}\lambda_2^{k_2}\ldots\lambda_m^{k_m}Y_{k_1k_2..k_m}]\times[E-B_{00..0}+$$

$$+\sum_{k_1=1}^{l}\lambda_1^{k_1}B_{k_1 0..0}+\sum_{k_2=1}^{l}\lambda_2^{k_2}B_{0k_2 0..0}+\ldots+\sum_{k_m=1}^{l}\lambda_m^{k_m}B_{00..0k_m}+\sum_{k_1+k_2=2}^{l}\lambda_1^{k_1}\lambda_2^{k_2}B_{k_1k_2 0..0}+$$

$$+\ldots+\sum_{k_s+k_r=2}^{l}\lambda_s^{k_s}\lambda_r^{k_r}B_{0..0k_s 0..0k_r 0..0}+\ldots+\sum_{k_{m-1}+k_m=2}^{l}\lambda_{m-1}^{k_{m-1}}\lambda_m^{k_m}B_{0..0k_{m-1}k_m}+\ldots+$$

$$+\sum_{\substack{k_1+k_2+\ldots+k_t=t\\t<m}}^{l}\lambda_1^{k_1}\lambda_2^{k_2}\ldots\lambda_t^{k_t}B_{k_1k_2..k_t 0..0}+\sum_{\substack{k_s+k_r+\ldots+k_p=t\\t<m}}^{l}\lambda_s^{k_s}\lambda_r^{k_r}\ldots\lambda_p^{k_p}B_{0..0k_s..k_r..k_p 0..0}+\ldots+$$

$$+\sum_{\substack{k_{m-t}+\ldots+k_m=t\\t<m}}^{l}\lambda_{m-t}^{k_{m-t}}\ldots\lambda_m^{k_m}B_{0..0k_{m-t}k_m}+\ldots+\sum_{k_1+k_2+\ldots+k_m=m}^{l}\lambda_1^{k_1}\lambda_2^{k_2}\ldots\lambda_m^{k_m}B_{k_1k_2..k_m}]=$$

$$=[Z_{00..0}+\sum_{k_1=1}^{nl}\lambda_1^{k_1}Z_{k_1 0..0}+\sum_{k_2=1}^{nl}\lambda_2^{k_2}Z_{0k_2 0..0}+\sum_{k_m=1}^{nl}\lambda_m^{k_m}Z_{00..0k_m}+\sum_{k_1+k_2=2}^{nl}\lambda_1^{k_1}\lambda_2^{k_2}Z_{k_1k_2 0..0}+$$

$$+\ldots+\sum_{k_s+k_r=2}^{nl}\lambda_s^{k_s}\lambda_r^{k_r}Z_{0..0k_s 0..0k_r 0..0}+\ldots+\sum_{k_{m-1}+k_m=2}^{nl}\lambda_{m-1}^{k_{m-1}}\lambda_m^{k_m}Z_{0..0k_{m-1}k_m}+\ldots+$$

$$+\sum_{\substack{k_1+k_2+\ldots+k_t=t\\t<m}}^{nl}\lambda_1^{k_1}\lambda_2^{k_2}\ldots\lambda_t^{k_t}Z_{k_1k_2..k_t 0..0}+\sum_{\substack{k_s+k_r+\ldots+k_p=t\\t<m}}^{nl}\lambda_s^{k_s}\lambda_r^{k_r}\ldots\lambda_p^{k_p}Z_{0..0k_s..k_r..k_p 0..0}+\ldots+$$

$$+\sum_{\substack{k_{m-t}+\ldots+k_m=t\\ t<m}}^{nl}\lambda_{m-t}^{k_{m-t}}\ldots\lambda_m^{k_m}Z_{0..0k_{m-t}k_m}+\ldots+\sum_{k_1+k_2+\ldots+k_m=m}^{nl}\lambda_1^{k_1}\lambda_2^{k_2}\ldots\lambda_m^{k_m}Z_{k_1k_2..k_m}]\times[C_{00..0}+$$

$$+\sum_{k_1=1}^{l}\lambda_1^{k_1}C_{k_1 0..0}+\sum_{k_2=1}^{l}\lambda_2^{k_2}C_{0k_2 0..0}+\ldots+\sum_{k_m=1}^{l}\lambda_m^{k_m}C_{00..0k_m}+\sum_{k_1+k_2=2}\lambda_1^{k_1}\lambda_2^{k_2}C_{k_1k_2 0..0}+$$

$$+\ldots+\sum_{k_s+k_r=2}^{l}\lambda_s^{k_s}\lambda_r^{k_r}C_{0..0k_s 0..0k_r 0..0}+\ldots+\sum_{k_{m-1}+k_m=2}^{l}\lambda_{m-1}^{k_{m-1}}\lambda_m^{k_m}C_{0..0k_{m-1}k_m}+\ldots+$$

$$+\sum_{\substack{k_1+k_2+\ldots+k_t=t\\ t<m}}^{l}\lambda_1^{k_1}\lambda_2^{k_2}\ldots\lambda_t^{k_t}C_{k_1k_2..k_t 0..0}+\sum_{\substack{k_s+k_r+\ldots+k_p=t\\ t<m}}^{l}\lambda_s^{k_s}\lambda_r^{k_r}\ldots\lambda_p^{k_p}C_{0..0k_s..k_r..k_p 0..0}+\ldots+$$

$$+\sum_{\substack{k_{m-t}+\ldots+k_m=t\\ t<m}}^{l}\lambda_{m-t}^{k_{m-t}}\ldots\lambda_m^{k_m}C_{0..0k_{m-t}k_m}+\ldots+\sum_{k_1+k_2+\ldots+k_m=m}^{l}\lambda_1^{k_1}\lambda_2^{k_2}\ldots\lambda_m^{k_m}C_{k_1k_2..k_m}].$$

Якщо згрупувати члени у лівій і правій частинах отриманого рівняння та прирівняти після цього коефіцієнти при однакових степенях λ, одержимо систему з числовими елементами для визначення невідомих матричних коефіцієнтів $Y_{k_1k_2\ldots k_m}$ та $Z_{k_1k_2\ldots k_m}$.

$$Y_{00..0}\left(E-B_{00..0}\right)-Z_{00..0}C_{00..0}=0;$$

$$Y_{10..0}\left(E-B_{00..0}\right)+Y_{00..0}\left(E-B_{10..0}\right)-\left[Z_{10..0}C_{00..0}+Z_{00..0}C_{10..0}\right]=0;$$

…………………………………………………

$$\sum_{k_1=0}^{l}Y_{l-k_1,0..0}\left(E-B_{k_1 0..0}\right)-\sum_{k_1=0}^{l}Z_{l-k_1,0..0}C_{k_1 0..0}=0;$$

…………………………………………………

$$\sum_{k_1=0}^{l}Y_{nl-k_1,0..0}\left(E-B_{k_1 0..0}\right)-\sum_{k_1=0}^{l}Z_{nl-k_1,0..0}C_{k_1 0..0}=0;$$

…………………………………………………

$$Y_{nl,0..0}\left(E-B_{l0..0}\right)-Z_{nl,0..0}C_{l0..0}=0;$$

$$\sum_{k_s=0}^{l}Y_{0..0,l-k_s,0..0}\left(E-B_{0..0k_s 0..0}\right)-\sum_{k_s=0}^{l}Z_{0..0,l-k_s,0..0}C_{0..0k_s 0..0}=0;$$

…………………………………………………

$$\sum_{k_1=0}^{l}\sum_{k_2=0}^{l} Y_{l-k_1,l-k_2,0..0}\left(E-B_{k_1k_2 0..0}\right)-\sum_{k_1=0}^{l}\sum_{k_2=0}^{l} Z_{l-k_1,l-k_2,0..0}C_{k_1k_2 0..0}=0;$$

$$\sum_{k_1=0}^{q}\sum_{k_2=0}^{q}\cdots\sum_{\substack{k_t=0\\ k_1+k_2+..+k_t=q,t<m}}^{q} Y_{q-k_1,q-k_2,\dots,q-k_t,0..0}\left(E-B_{k_1k_2\dots k_t 0..0}\right)-$$

....

$$-\sum_{k_1=0}^{q}\sum_{k_2=0}^{q}\cdots\sum_{\substack{k_t=0\\ k_1+k_2+..+k_t=q,t<m}}^{q} Y_{q-k_1,q-k_2,\dots,q-k_t,0..0}C_{k_1,k_2,\dots k_t 0..0}=0;$$

.... (4.47)

$$\sum_{k_1=0}^{l}\sum_{\substack{k_2=0\\ k_1+k_2+..+k_m=l}}^{l}\cdots\sum_{k_m=0}^{l} Y_{l-k_1,l-k_2,\dots,l-k_m,0..0}\left(E-B_{k_1k_2\dots k_m}\right)-$$

$$-\sum_{k_1=0}^{l}\sum_{\substack{k_2=0\\ k_1+k_2+..+k_m=l}}^{l}\cdots\sum_{k_m=0}^{l} Z_{l-k_1,l-k_2,\dots,l-k_m}C_{k_1k_2\dots k_m}=0;$$

....

$$\sum_{k_1=0}^{l}\sum_{\substack{k_2=0\\ k_1+k_2+..+k_m=nl+l}}^{l}\cdots\sum_{k_m=0}^{l} Y_{nl+l-k_1,nl+l-k_2,\dots,nl+l-k_m}\left(E-B_{k_1k_2\dots k_m}\right)-$$

$$-\sum_{k_1=0}^{l}\sum_{\substack{k_2=0\\ k_1+k_2+..+k_m=nl+l}}^{l}\cdots\sum_{k_m=0}^{l} Z_{nl+l-k_1,nl+l-k_2,\dots,nl+l-k_m}C_{k_1k_2\dots k_m}=0.$$

4.2.3. Оцінки характеристик алгоритмів для розв'язання системи з числовими елементами

Поліном степеня m з n невідомими $f\left(y_1,y_2,\dots,y_m\right)$ є сумою одночленів вигляду $y_1^{a_1}y_2^{a_2}y_3^{a_3}\dots y_m^{a_m}$, а максимальна кількість одночленів $m(n,m)$ визначається за формулою [62]:

$$m(n,m)=\frac{(n+m)!}{n!m!} \tag{4.48}$$

У лівій та правій частинах рівняння (4.46) маємо поліноми степеня $(nl + l)$. Тоді за формулою (4.48) максимальна кількість рівнянь N_{eqn} у системі (4.47) обмежена зверху

$$N_{eqn} \le \frac{(nl + l + m)!}{(nl + l)m!}, \tag{4.49}$$

максимальна кількість N_{sim} невідомих $Y_{k_1 k_2 \ldots k_m}$ та $Z_{k_1 k_2 \ldots k_m}$ також обмежена зверху

$$N_{sim} \le (n + 1)\frac{(nl + m)!}{(nl)!m!}. \tag{4.50}$$

4.2.4. Розв'язання блочної системи з числовими елементами

Для розв'язання отриманої системи (4.47) використаємо алгоритм схеми розрізання [62]. Для цього достатньо систему (4.47) записати у спеціальному вигляді. З цією метою позначимо через B_r матрицю, що складається із матриць $B_{k_1 k_2 \ldots k_m}$, сума індексів яких задовольняє умову $\sum_{i=1}^{m} k_i = r$, $\left(r = \overline{0, l}\right)$

Тоді отримаємо матриці $B_0, B_1, B_2, \ldots, B_r, \ldots, B_l$. Матриця $B_0 = (B_{00\ldots0})$ складається з одного елемента. Матриці

$$B_1 = \begin{bmatrix} B_{10\ldots0} & 0 & \ldots & 0 \\ 0 & B_{010\ldots0} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & B_{0\ldots01} \end{bmatrix}, \quad B_2 = \begin{bmatrix} B_{20\ldots0} & B_{110\ldots0} & B_{1010\ldots0} & \ldots & B_{10\ldots01} \\ 0 & B_{020\ldots0} & B_{0110\ldots0} & \ldots & B_{010\ldots01} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & B_{0\ldots02} \end{bmatrix}$$

мають розмірність $m \times m$. Матриця

$$B_3 = \begin{pmatrix} B_{30..0} & B_{210...0} & B_{2010...0} & B_{20010...0} & \dots & B_{20...010} & B_{20...001} \\ 0 & B_{120...0} & B_{1110...0} & B_{11010...0} & \dots & B_{110...010} & B_{110...001} \\ 0 & 0 & B_{1020...0} & B_{10110...0} & \dots & B_{1010...010} & B_{1010...01} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & B_{10...020} & B_{10...011} \\ 0 & 0 & 0 & 0 & \dots & 0 & B_{10...02} \\ 0 & B_{030...0} & B_{0210...0} & B_{02010...0} & \dots & B_{020...010} & B_{020...01} \\ 0 & 0 & B_{0120...0} & B_{01110...0} & \dots & B_{011...010} & B_{0110...01} \\ 0 & 0 & 0 & B_{01020...0} & \dots & B_{01010...010} & B_{01010...01} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & B_{010...020} & B_{010...011} \\ 0 & 0 & 0 & 0 & \dots & 0 & B_{010...02} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & B_{0...030} & B_{0...021} \\ 0 & 0 & 0 & 0 & \dots & 0 & B_{0...012} \\ 0 & 0 & 0 & 0 & \dots & 0 & B_{0...03} \end{pmatrix}$$

вже має m стовпців і $1+2+3+...+m = \frac{(1+m)m}{2}$ рядків. Подібно записуються матриці $B_4, B_5, B_6, \dots, B_l$. Кожна з них має по m стовпців, а кількість рядків у матриці B_l не перевищує $\frac{(l+m)!}{l!m!}$.

Можна також записати матриці $C_i \left(i = \overline{1,l}\right), X_i \left(i = \overline{1,nl}\right)$ та $Z_i \left(i = \overline{1,nl}\right)$. Кількість рядків у матрицях X_{nl} і Z_{nl} не перевищує $\frac{(nl+m)!}{(nl)!m!}$.

Використовуючи означення добутку Кронекера [95], систему (4.47) запишемо у вигляді:

$$X \otimes (E - B) - Z \otimes C = 0, \tag{4.51}$$

або

$$X - X \otimes B - Z \otimes C = 0, \tag{4.52}$$

де $X \otimes B$, $Z \otimes C$ - тензорні добутки.

Систему (4.47) розв'яжемо за алгоритмом схеми розрізання [62]. Для цього достатньо матрицю системи (4.52) розділити на блоки:

$$\begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix},$$

де M_{11} - квадратна матриця, розміри якої не перевищують $\dfrac{(nl+l+m)!}{(nl+l)!\,m!\,(n+1)}$, а M_{12}, M_{21}, M_{22} - прямокутні матриці відповідних розмірів. Включаючи останні $(n+1)\dfrac{(nl+m)!}{(nl)!\,m!} - \dfrac{(nl+l+m)!}{(nl+l)!\,m!}$ невідомі як параметри, які будуть містити у собі лише останні $2l+1$ компонент вектора N_2, одержимо неоднорідну квадратну систему:

$$\begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \begin{pmatrix} U \\ V \end{pmatrix} = \begin{pmatrix} N_1 \\ N_2 \end{pmatrix},$$

порядок якої обмежений числом $\dfrac{(nl+l+m)!}{(nl+l)!\,m!}$, а розв'язки дають параметричну сім'ю розв'язків системи (4.52). Отже, пошук невідомих U і V зводиться до розв'язання двох систем меншого порядку. Вектор V визначається із системи

$$\left(M_{22} - M_{21} \times \left(M_{11}^{-1} \times M_{12}\right)\right) \times V = N_2 - M_{21} \times \left(M_{11}^{-1} \times N_1\right), \qquad (4.53)$$

причому $M_{11} \neq 0$ і U можна знайти із системи

$$M_{11} \times U = N_1 - M_{12} \times V. \qquad (4.54)$$

Розв'язання отриманих систем (4.53) та (4.54) вимагає деяких проміжних матричних операцій [62]. Для обчислення добутку $M_{11}^{-1} \times N_1$ достатньо знайти розв'язки систем:

$$\begin{bmatrix} B_0 & 0 & 0 & \dots & 0 & \dots & 0 \\ B_1 & B_0 & 0 & \dots & 0 & \dots & 0 \\ B_2 & B_1 & B_0 & \dots & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ B_l & B_{l-1} & B_{l-2} & \dots & B_0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & \dots & B_0 \end{bmatrix} \cdot \begin{bmatrix} W_0^{(i)} \\ W_1^{(i)} \\ \dots \\ W_l^{(i)} \\ \dots \\ W^{(i)}_{\frac{(nl+l+m)!}{(nl+l)!m!(n+1)}} \end{bmatrix} = \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ \dots \\ C_l \\ \dots \\ 0 \end{bmatrix}, \qquad (4.55)$$

де $i = 0, 1, \dots, nl$.

Перш за все, потрібно розв'язати систему при $i = 0$, а решту невідомих визначимо за співвідношеннями:

$$\begin{cases} W_0^{(i)} = W_1^{(i)} = \dots W_{i-1}^{(i)} = 0; \\ W_i^{(i)} = W_0^{(0)} \\ W_{i+1}^{(i)} = W_1^{(0)} \\ \dots\dots\dots\dots\dots\dots \\ W^{(i)}_{\frac{(nl+l+m)!}{(nl+l)!m!(n+1)}} = W^{(0)}_{\frac{(nl+l+m)!}{(nl+l)!m!(n+1)}-1}. \end{cases}$$

Розв'язки системи (4.55) при $i = 0$ обчислимо за формулами:

$$\begin{aligned} & W_i^{(0)} = B_0^{-1} \cdot \left[B_i - \sum_{j=1}^{i} B_j \cdot W_{i-j}^{(0)} \right], \ \left(i = \overline{0,1} \right); \\ & W_i^{(0)} = B_0^{-1} \cdot \left[B_i - \sum_{j=1}^{i} B_j \cdot W_{i-j}^{(0)} \right], \ \left(i = \overline{l+1, \frac{(nl+l+m)!}{(nl+l)!m!(n+1)}} \right). \end{aligned} \qquad (4.56)$$

Для знаходження всіх $W_j^{(0)} \left(j = \overline{l+1, \frac{(nl+l+m)!}{(nl+l)!m!(n+1)}} \right)$ треба витратити з точністю до головного члена $n^3 m \dfrac{(l+m)!}{l!m!} \dfrac{(nl+l+m)!}{(nl+l)!m!(n+1)}$ арифметичних операцій. Обчислення добутків $M_{21} \cdot \left(M_{11}^{-1} \cdot M_{12} \right)$ та $M_{21} \cdot \left(M_{11}^{-1} \cdot N_1 \right)$ вимагає виконання $n^3 \left[\dfrac{(nl+l+m)!}{(nl+l)!m!(n+1)} \right]^2$ арифметичних дій. Для визначення невідомих V із системи

$\frac{(nl+l+m)!n}{(nl+l)!m!(n+1)}$-го порядку треба затратити $O\left(\frac{(nl+l+m)!}{(nl+l)!m!}\right)^b$ операцій. При використанні алгоритму відсічених систем $b=3$ для розв'язання системи (4.55) спочатку треба знайти добуток $M_{12} \times V$. Для цього необхідно $n^2\left[\frac{(nl+l+m)!}{(nl+l)!m!(n+1)}\right]^2$ операцій, після чого систему можна розв'язати за формулами (4.56), на що потрібно $n^3\left[\frac{(nl+l+m)!}{(nl+l)!m!(n+1)}\right]^2$ дій. Тому для повної реалізації алгоритму схеми розрізання для системи (4.55) потрібно виконати $n^3\left[\frac{(nl+l+m)!}{(nl+l)!m!(n+1)}\right]^2+O\left(\frac{(nl+l+m)!}{(nl+l)!m!}\right)^b$ арифметичних операцій на комп'ютері.

4.3. Класична диференціальна модель Леонтьєва

У попередньому розділі, аналізуючи модель Леонтьєва, зроблено певні припущення, що спрощують розв'язання системи диференціальних рівнянь. Опишемо шляхи удосконалення цієї моделі, проаналізувавши деякі твердження.

1. Перехід до системи нерівностей з умовами “необоротності” капітальних вкладень

За умови (4.2) вектор чистих капітальних вкладень пов’язаний з приростами виробництва співвідношенням

$$U(t)=B\frac{dX}{dt},$$

де $\frac{dx_j(t)}{dt} > 0$. Якщо $\frac{dx_j(t)}{dt} < 0$, то $u_{ij} = b_{ij}\frac{dx_j(t)}{dt} < 0$. Це означає повернення ресурсів у баланси продукції за повними нормами капіталомісткості.

Тому розв’язки систем (4.2) і (4.4) можна використовувати тільки тоді, коли вони задовольняють додаткову умову

$$\frac{dX}{dt} \geq 0. \tag{4.57}$$

Звичайно, що додаткова умова (4.57) не є необхідна, адже важливо не заборонити від'ємні прирости, а виключити феномен “оборотності” капітальних вкладень. Тому буде доречно безпосередньо включити в модель умову “необоротності” капіталовкладень

$$u_{ij} = b_{ij}\left[\frac{dx_j(t)}{dt}\right], \tag{4.58}$$

де $$\left[\frac{dx_j(t)}{dt}\right]_+ = \begin{cases} \frac{dx_j(t)}{dt}, \ \textit{якщо}\ \frac{dx_j(t)}{dt} \geq 0; \\ 0, \ \textit{якщо}\ \frac{dx_j(t)}{dt} < 0. \end{cases}$$

З урахуванням умови (4.58) динамічна модель Леонтьєва набуває вигляду

$$X(t) = A(t)X(t) + B(t)\left[\frac{dX}{dt}\right]_+ + C(t). \tag{4.59}$$

Розглянемо випадок, коли система рівнянь (4.59) не матиме допустимих траєкторій, що проходять через задану початкову точку $X(0) \geq 0$. Така ситуація виникає, якщо, наприклад, $C(t) = 0$, а вектор $X(0)$ визначається з умови $B^{-1}(t)\left(E - A(t)\right)X \geq 0$, без виконання якої рівність $\left[\frac{dX}{dt}\right]_+ = B^{-1}(t)\left(E - A(t)\right)X$ неможлива. У зв’язку з цим доцільно замінити систему рівнянь (4.59) системою нерівностей:

$$X(t) \geq A(t)X(t) + B(t)\left[\frac{dX}{dt}\right]_+ + C(t). \tag{4.60}$$

Завдяки цьому вдається уникнути багатьох проблем, що виникали при попередньому аналізі. Так, модель замкненої виробничої системи

$$X(t) \geq A(t)X(t) + B(t)\left[\frac{dX}{dt}\right]_{+}, \tag{4.61}$$

на відміну від рівняння (4.60), має розв'язок за будь-якого початкового значення $X(0)$ і 0, $X(t) = X(0)e^{lt}$.

Підставивши його в (4.61), одержимо нерівність

$$X(0) - A(t)X(0) \geq \lambda B(t)X(0),$$

що при всіх λ з проміжку $[0, \lambda(X(0))]$, де

$$\lambda(X(0)) = \min_{i \in J} \frac{x_i(0) - \sum_{j \in J} a_{ij}(t)x_j(0)}{\sum_{j \in J} b_{ij}(t)x_j(0)}. \tag{4.62}$$

Справджується нерівність $\lambda(X(0)) \leq \hat{\lambda}$, яка перетворюється на рівність, якщо вектор $(E - A)X(0)$ пропорційний власному вектору $\hat{X}$, який відповідає $\hat{\lambda}$.

Отже, для моделі замкненої виробничої системи (4.61) максимально можливий постійний темп приросту виробництва всіх галузей l існує за будь-якого початкового стану $X(0) \geq 0$.

Модель (4.62), що включає потік споживання $C(t) = C_0 e^{rt}$, має розв'язок $X(t) = X(0)e^{rt}$. Справді, підставивши $X(t) = X(0)e^{rt}$ і $C(t) = C_0 e^{rt}$ у (4.60), отримуємо

$$X(0) \geq A(t)X(0) + rB(t)X(0) + C_0.$$

Зауважимо, якщо, наприклад, задане значення C_0 задовольняє нерівність $(E - A)X(0) > C_0$, то максимально можливий темп приросту r_0 дорівнює:

$$r_0 = \min_{i \in J} \frac{x_i(0) - \sum_{j \in J} a_{ij}(t)x_j(0) - c_{0i}}{\sum_{j \in J} b_{ij}(t)x_j(0)}.$$

Однак окремі компоненти споживання $\tilde{C}(t)$ можуть перевищувати заданий закон зростання: $\tilde{C}(t) = (C_0 + \Delta C_0)e^{rt}$, де вектор $\Delta C_0 \geq 0$, а окремі його компоненти ΔC_{0i} дорівнюють $\Delta c_{0i} = x_{i0} - \sum_{j \in J} a_{ij}(t) x_j(0)$.

2. Врахування резервів виробничих потужностей

Для моделі Леонтьєва (4.2) припустимо, що приріст виробництва може здійснюватись тільки за рахунок виробничого нагромадження. Тоді з $u_{ij}(t) = 0$ випливає $\frac{dx_j(t)}{dt} = 0$. Насправді розширення виробництва може відбуватися й у випадку більш повного використання наявних виробничих потужностей (основних виробничих фондів). Дійсно, позначивши через $L_j(t)$ максимально можливий обсяг виробництва продукції j-ої галузі на діючих основних фондах у році t, введемо умови: $x_j(t) \leq L_j(t)$.

Тому, якщо $x_j(t) \leq L_j(t)$, то стає припустимим $u_{ij}(t) = 0$ при $\frac{dx_j(t)}{dt} > 0$. Отже, введення додаткових умов $x_j(t) \leq L_j(t)$, $j \in J$ уточнює розв'язок системи (4.2).

3. Побудова міжгалузевих моделей з інвестиційними лагами

Припустимо тепер, що в моделі Леонтьєва (4.2) миттєво перетворюється виробниче нагромадження (капітальних вкладень) у прирости виробництва. Удосконалимо модель, включаючи інвестиційні лаги.

Означення 4.4. *Лаг* — показник, що характеризує часовий інтервал між двома взаємопов'язаними явищами, одне з яких є причиною, а друге – наслідком. Наприклад, існує лаг між виділенням капіталовкладень на будівництво і введенням у дію збудованих об'єктів. Лаг застосовують у математичних моделях, в яких враховується наявність різних проміжків часу.

Означення 4.5. *Інвестиційний лаг* – розрив з часом між здійсненням інвестицій і їх окупністю.

Однак, якщо визначати інвестиційний лаг у кожній j-й галузі як зосереджений і рівний $\bar{t}_j$, то система рівнянь виробництва і розподілу продукції (національного доходу) набуває вигляду:

$$x_i(t) = \sum_{j \in J} a_{ij}(t) x_j(t) + \sum_{j \in J} b_{ij}(t) \frac{dx_j}{dt}\left(t + \bar{\tau}_j\right) + c_i(t), \; i \in J. \tag{4.63}$$

Інвестиційні лаги можуть диференціюватися не тільки за галузями – “споживачами” капітальних вкладень, але й за матеріально-речовими елементами капітальних вкладень (продукція машинобудування, будівництва і т. ін.). Позначивши через t_{ij} лаг капіталовкладень, що надходять з i-ї галузі і забезпечують приріст виробництва в j-й галузі, одержуємо таку систему рівнянь:

$$x_i(t) = \sum_{j \in J} a_{ij}(t) x_j(t) + \sum_{j \in J} b_{ij}(t) \frac{dx_j}{dt}\left(t + \bar{\tau}_{ij}\right) + c_i(t), \; i \in J. \tag{4.64}$$

У динамічну міжгалузеву модель можуть бути включені також і розподілені лаги. Тоді змінюється зміст матриці B .

Методи розв’язування систем диференціальних рівнянь з лагами (запізнюючими аргументами) розроблено досить добре, однак якісний аналіз розв’язків істотно ускладнюється.

4. Врахування динаміки коефіцієнтів матеріаломісткості і капіталомісткості виробництва

Одне з найсуттєвіших припущень аналізованої моделі (4.2) – незмінність з часом матриць матеріальних і капітальних витрат A і B . Але тоді з моделі випадає найважливіший фактор економічної динаміки – науково-технічний прогрес.

Включення в модель матриць $A(t)$ і $B(t)$, що залежать від часу, підвищує її прикладне значення. Теоретичний аналіз узагальненої моделі можливий за досить простих законів зміни $A(t)$ і $B(t)$, наприклад за рівномірної зміни всіх коефіцієнтів матриці або коефіцієнтів окремих рядків матриці.

4.4. Динамічний варіант диференціальної моделі Леонтьєва

Спробуємо проаналізувати динамічну модель Леонтьєва (4.2). Нехай максимізується сумарний фонд споживання за плановий період T $\int_0^T GC(t)dt$, де порівняння різних споживчих благ здійснюється за допомогою вектора сталих коефіцієнтів $G = (g_i) > 0$. Максимум вказаного критерію потрібно знайти за умов:

$$\begin{cases} X(t) = A(t)X(t) + B(t)\dfrac{dX(t)}{dt} + C(t); \\ \dfrac{dX(t)}{dt} \geq 0; \\ X(0) = X_0. \end{cases} \tag{4.65}$$

Умова $\frac{dX(t)}{dt} \geq 0$ гарантує “необоротність” капіталовкладень, але, разом з тим, істотно звужує область вибору. Із (4.65) виведемо співвідношення для $Y(t)$, $\frac{dY(t)}{dt}$, $U(t)$:

$$U(t) = B(t)\left(E - A(t)\right)^{-1}\frac{dY(t)}{dt};$$

$$\frac{dY(t)}{dt} = \left(E - A(t)\right)B^{-1}(t)U(t);$$

$$\frac{dX(t)}{dt} = \left(E - A(t)\right)^{-1}\frac{dY(t)}{dt} = B^{-1}(t)U(t).$$

Зауважимо, що коли $\frac{dY(t)}{dt} = B^{-1}(t)U(t) \geq 0$, то і $U(t) \geq 0$.

Сформулюємо задачу оптимального управління

$$\begin{cases} \int\limits_0^T G\left[Y(t) - U(t)\right]dt \to \max; \\ \dfrac{dY(t)}{dt} = \left(E - A(t)\right)B^{-1}(t)U(t); \\ U(t) \le Y(t); \\ B^{-1}(t)U(t) \ge 0; \\ Y(0) = Y_0. \end{cases} \tag{4.66}$$

Модель (4.66) дає змогу знаходити такі траєкторії, які дають більшу величину сумарного споживання $\int\limits_0^T GC(t)dt$, задовольняючи умови (4.65).

Як показує аналіз вказаної задачі, максимум досягається на основі зосередження всіх ресурсів національного доходу в різні моменти часу або цілком на споживанні, або на виробничому нагромадженні за миттєвої зміни режиму відтворення.

До виробничого нагромадження належать розширення та якісне вдосконалення основних фондів сфери матеріального виробництва.

Дослідження задачі (4.66) з використанням принципу максимуму Понтрягіна [29], виявило такі властивості:

Існує величина $T_0 > 0$ така, що:

1) коли $T \le T_0$, то виробництво не збільшується й увесь національний дохід споживається:

$$y_i^*(t) = c_i^*(t) = y_i(0), \quad u_i^o(t) = 0, \quad i \in J;$$

2) коли $T > T_0$, то аналогічна ситуація має місце наприкінці планового періоду на відрізку $\left[T - T_0, T_0\right]$ де весь національний дохід цілком витрачається на споживання.

Виробниче нагромадження здійснюється на відрізку $\left[0, T - T_0\right]$ однак не можна гарантувати, що $u_i^*(t) > 0$ для всіх галузей.

Особливість динамічної диференційної моделі Леонтьєва також у тому, що в кожен часовий момент продукція певної галузі не обов'язково має бути витрачена або тільки на споживання, або тільки на нагромадження. Різні галузі "поводять" себе по-різному. Для будь-якого $t \in [0, T_0]$ виділяють дві підмножини (можливо, такі, що перетинаються) галузей $J_1(t), J_2(t) \subseteq J$. Коли $i \in J_1(t)$ всю продукцію, що виходить зі сфери виробництва, спрямовують на нагромадження $u_i^*(t) = y_i^*(t)$.

Тобто, коли $i \in J_1(t)$ обсяги виробництва не зростають $\frac{dx_i^*(t)}{dt} = 0$.

Зокрема, за $t > T - T_0$ маємо $J_2(t) = J$.

Отже, поведінка оптимального розв'язку істотно залежить від тривалості планового періоду. За досить великої величини T оптимальна траєкторія обов'язково містить стрибкоподібні зміни споживання і нагромадження, принаймні для продукції деяких галузей.

Для пом'якшення (згладжування) різких змін оптимальної траєкторії задачі (4.66) можна використовувати ряд методів:

1) поділ споживання на екзогенно та ендогенно зумовлені частини й умови неперервного зростання споживання (зокрема максимізація темпу r);
2) задання нижніх і верхніх меж норми виробничого нагромадження.

Однак в динамічній диференціальній моделі Леонтьєва число різких змін траєкторії у багато разів більше. Тому здебільшого зазначених методів буде недостатньо. Для згладжування оптимальних траєкторій можуть знадобитися більш диференційовані й гнучкі регулятори. Наприклад, можна пов'язати між собою вектори $C(t)$ і $U(t)$ так, щоб уникнути зміни структури кінцевої продукції. У найпростішому випадку $c_i(t) = k_i y_i(t)$, де $0 \le k_i \le 1$ – незмінні норми споживання продукції галузей. Тоді $u_i(t) = (1 - k_i) y_i(t)$. Коефіцієнти k_i можна розглядати як змінні величини. Зокрема, прийнявши

$k_i(t) = k_i(0)p(t)$, де $p(t) > 0$, будемо контролювати не тільки структуру використання продукції кожної галузі, але й загальне співвідношення між споживанням і нагромадженням. Якщо встановлюється $p(t) > 1$, то це означає збільшення норми споживання, порівняно з початковим моментом часу, якщо ж вважати $p(t) < 1$, то збільшується частка нагромадження.

У більш загальному вигляді $C(t) = K(t)Y(t)$,

де $K(t) = k_{ij}(t)$ – змінна з часом матриця взаємозв'язку споживання і використовуваного національного доходу; елементи матриці $K(t)$ беруть з деякого фіксованого класу гладких функцій, в межах якого й здійснюють оптимізацію.

Очевидно, що приєднання такого виду умов до задачі (3.66) зменшує значення цільової функції споживання. Однак забезпечується гладкість динаміки споживання, нагромадження і виробництва. Паралельно зменшується залежність оптимальних розв'язків від тривалості планового періоду і зменшується деформація структури виробництва та розподілу продукції наприкінці періоду.

Висновки до четвертого розділу

В даному розділі розглядається використання числових алгоритмів для узагальнених моделей В. Леонтьєва.

Запропоновано динамічну міжгалузеву модель, яка є класичним прикладом використання систем диференціальних рівнянь у дослідженні проблем економічного зростання.

Розвинуто новий підхід до розв'язання систем лінійних алгебричних рівнянь з m–вимірними λ–матрицями в моделі Леонтьєва. Створено оптимізаційну модель блочної системи з числовими елементами для матриць міжгалузевого балансу та запропоновано ефективний обчислювальний метод реалізації цієї моделі. Побудовано міжгалузеву модель з інвестиційними лагами (запізнюючими аргументами).

Результати третього розділу опубліковані у роботах [80; 84] та доповідались на конференціях [78; 86].

РОЗДІЛ 5. МЕТОДИ КОМП'ЮТЕРНОЇ АЛГЕБРИ ДИНАМІЧНИХ ВАРІАНТІВ МОДЕЛІ ЛЕОНТЬЄВА

Математичне моделювання в наукових дослідженнях і практиці невід'ємна риса науково-технічного прогресу. Його ефективність визначається продуктивністю комп'ютерних систем та якістю обчислювальних алгоритмів і програм, що використовуються.

5.1. Числові методи розв'язання розріджених систем лінійних алгебричних рівнянь

Числові методи алгебри – одні з базових інструментів математичного моделювання і важлива частина програмного забезпечення для комп'ютерів усіх поколінь.

Числові методи алгебри на практиці присвячені проблемі розв'язання систем алгебричних рівнянь з елементами, що задані наближено. Існує багато задач, наприклад у цілочисловому програмуванні [3, 29], де початкові дані систем лінійних рівнянь задаються точно і розв'язки також потрібно визначити точно. Побудова ефективних методів визначення невідомих для таких систем – потрібна і досить непроста задача.

У значній кількості прикладних задач виникає необхідність розв'язання розріджених систем рівнянь з блочними елементами [62]. Розглянемо метод розв'язування розріджених систем з деякими найхарактернішими способами заповнення.

Нехай система лінійних алгебричних рівнянь вигляду:

$$\begin{pmatrix} A_{1,1} & A_{1,2} & 0 & \dots & 0 & 0 & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & \dots & 0 & 0 & 0 \\ 0 & A_{3,2} & A_{3,3} & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & A_{n-1,n-2} & A_{n-1,n-1} & A_{n-1,n} \\ 0 & 0 & 0 & \dots & 0 & A_{n,n-1} & A_{n,n} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ \dots \\ X_{n-1} \\ X_n \end{pmatrix} = \begin{pmatrix} A_{1,n+1} \\ A_{2,n+1} \\ A_{3,n+1} \\ \dots \\ A_{n-1,n+1} \\ A_{n,n+1} \end{pmatrix}, \tag{5.1}$$

елементи якої A_{ij} - це блоки розмірності $m \times m$. Позначимо через $A\begin{bmatrix} i_1 & i_2 & \dots & i_k \\ j_1 & j_2 & \dots & j_k \end{bmatrix}$ мінор, розміщений на перетині блочних рядків $i_1, i_2, \dots, i_k$ та блочних стовпців $j_1, j_2, \dots, j_k$. За узагальненим правилом Крамера [44] маємо:

$$X_1 = \begin{vmatrix} A_{1,1} & A_{2,1} & \cdots & A_{1,n+1} & 0 & 0 \\ A_{1,2} & A_{2,2} & \ddots & A_{2,n+1} & \ddots & 0 \\ \ddots & A_{3,2} & \ddots & A_{3,n+1} & \ddots & \cdots \\ 0 & \ddots & \ddots & \ddots & \ddots & A_{n-1,n} \\ 0 & 0 & \ddots & A_{n,n+1} & \cdots & A_{n,n} \end{vmatrix} \Bigg/ \begin{vmatrix} A_{1,1} & A_{1,2} & 0 & \cdots & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & \cdots & 0 \\ 0 & A_{3,2} & A_{3,3} & \ddots & 0 \\ \cdots & \cdots & \ddots & \ddots & A_{n-1,n} \\ 0 & 0 & 0 & \cdots & A_{n,n} \end{vmatrix}$$

Розкладаючи чисельник за мінорами, запишемо:

$$X_i = \left(A\begin{bmatrix} 1 & 2 & \dots & n \\ 1 & 2 & \dots & n \end{bmatrix} \right)^{-1} \left(\sum_{k=1}^{n} (-1)^{i+k} A_{k,n+1} A\begin{bmatrix} 1 & 2 & \dots & i-1 \\ 1 & 2 & \dots & i-1 \end{bmatrix} \prod_{s=i+1}^{k-1} A_{s,s+1} A\begin{bmatrix} k+1 & \dots & n \\ k+1 & \dots & n \end{bmatrix} \right) \quad (5.2)$$

Введемо позначення:

$$a_{ik} = A\begin{bmatrix} 1 & 2 & \dots & i-1 \\ 1 & 2 & \dots & i-1 \end{bmatrix} \prod_{s=i+1}^{k-1} A_{s,s+1} A\begin{bmatrix} k+1 & \dots & n \\ k+1 & \dots & n \end{bmatrix} \quad i,k = \overline{1,n}.$$

Тоді для визначення невідомої одержимо співвідношення:

$$X_i = \left(A\begin{bmatrix} 1 & 2 & \dots & n \\ 1 & 2 & \dots & n \end{bmatrix} \right)^{-1} \sum_{k=1}^{n} (-1)^{i+k} A_{k,n+1} a_{i,k}. \quad (5.3)$$

Для компактності запису надалі позначимо результат виконання операції множення на обернену матрицю зліва у вигляді $C^{-1}D = D / C$. Тоді вираз $D_1 / (C_1 + D_2 / C_2)$ означатиме $(C_1 + C_2^{-1} D_2)^{-1} \cdot D_1$.

Якщо до співвідношення (5.3) застосувати відому рівність Леонарда Ейлера [13], яка пов'язує ланцюгові дроби з рядами та скінченними сумами, то для X_i одержимо:

$$X_i = \left(A\begin{bmatrix} 1 & 2 & \dots & n \\ 1 & 2 & \dots & n \end{bmatrix} \right)^{-1} \sum_{k=1}^{n} (-1)^{1+k} A_{k,n+1} a_{i,k} = \frac{A\begin{bmatrix} 2 & 3 & \dots & n \\ 2 & 3 & \dots & n \end{bmatrix}}{A\begin{bmatrix} 1 & 2 & \dots & n \\ 1 & 2 & \dots & n \end{bmatrix}}$$

$$\times\cfrac{A_{1,n+1}}{E+\cfrac{A_{2,n+1}a_{1,2}\,/\,A_{1,n+1}a_{1,1}}{E-\dfrac{A_{2,n+1}a_{1,2}}{A_{1,n+1}a_{1,2}}+\cfrac{A_{3,n+1}a_{1,3}\,/\,A_{2,n+1}a_{1,2}}{E-\dfrac{A_{3,n+1}a_{1,3}}{A_{2,n+1}a_{1,2}}+\ddots+\cfrac{A_{n,n+1}a_{1,n}\,/\,A_{n-1,n+1}a_{1,n-1}}{E-\dfrac{A_{n,n+1}a_{1,n}}{A_{n-1,n+1}a_{1,n-1}}}}}} \qquad (5.4)$$

Тут E – одинична матриця.

Вираз $a^{-1}_{1,k+1}\cdot a_{1,k}$ $(k=1,2,\ldots)$ розкладемо в ланцюгові дроби [7; 90]:

$$\frac{a_{1,k}}{a_{1,k+1}}=\frac{A_{1,k}A\begin{bmatrix}1 & \ldots & k\\ 1 & \ldots & k\end{bmatrix}}{A_{k+1,k+1}A\begin{bmatrix}1 & \ldots & k\\ 1 & \ldots & k\end{bmatrix}-A_{k+1,k}A\begin{bmatrix}1 & \ldots & k-1\\ 1 & \ldots & k-1\end{bmatrix}}=$$

$$=\frac{A_{1,k}\begin{bmatrix}1 & \ldots & k\\ 1 & \ldots & k\end{bmatrix}}{A_{k+1,k+1}A\begin{bmatrix}1 & \ldots & k\\ 1 & \ldots & k\end{bmatrix}-A_{k+1,k}A_{k,k+1}A\begin{bmatrix}1 & \ldots & k-1\\ 1 & \ldots & k-1\end{bmatrix}}=$$

$$=\frac{E}{A_{k+1,k+1}-A_{k+1,k}A_{k,k+1}A\begin{bmatrix}1 & \ldots & k-1\\ 1 & \ldots & k-1\end{bmatrix}A^{-1}\begin{bmatrix}1 & \ldots & k\\ 1 & \ldots & k\end{bmatrix}}=$$

$$=\frac{E}{A_{k+1,k+1}-A_{k+1,k}A_{k,k+1}\left(A^{-1}\begin{bmatrix}1 & \ldots & k\\ 1 & \ldots & k\end{bmatrix}^{T}A^{T}\begin{bmatrix}1 & \ldots & k-1\\ 1 & \ldots & k-1\end{bmatrix}\right)^{T}}=$$

$$=\ldots=\cfrac{E}{A_{k+1,k+1}-\cfrac{A_{k+1,k}A_{k,k+1}}{A_{k,k}-\cfrac{A_{k-1,k}A_{k,k-1}}{A_{k-1,k-1}-\ddots-\cfrac{A_{1,2}A_{2,1}}{A_{1,1}}}}}. \qquad k=\overline{1,n}. \qquad (5.5)$$

Аналогічно:

$$\frac{A\begin{bmatrix}2 & 3 & \ldots & n\\ 2 & 3 & \ldots & n\end{bmatrix}}{A\begin{bmatrix}1 & 2 & \ldots & n\\ 1 & 2 & \ldots & n\end{bmatrix}}=\frac{E}{A_{1,1}-A_{2,1}A_{1,2}\left[A^{-1}\begin{bmatrix}2 & 3 & \ldots & n\\ 2 & 3 & \ldots & n\end{bmatrix}^{T}A\begin{bmatrix}3 & 4 & \ldots & n\\ 3 & 4 & \ldots & n\end{bmatrix}^{T}\right]}=$$

$$=\ldots=\cfrac{E}{A_{1,1}-\cfrac{A_{2,1}A_{1,2}}{A_{2,2}-\cfrac{A_{2,3}A_{3,2}}{A_{3,3}-\ddots-\cfrac{A_{n-1,n}A_{n,n-1}}{A_{n,n}}}}}.$$

За такою ж схемою знаходимо решту невідомих X_i:

$$X_i=\left(A\begin{bmatrix}1 & 2 & \ldots & n\\ 1 & 2 & \ldots & n\end{bmatrix}\right)^{-1}\left(\frac{A_{i,n+1}}{2}a_{i,i}+\sum_{k=1}^{i-1}(-1)^{i+k}A_{k,n+1}a_{i,k}+\frac{A_{i,n+1}}{2}a_{i,i}+\sum_{k=i+1}^{n}(-1)^{i+k}A_{k,n+1}a_{i,k}\right)=$$

$$=\left[(-1)^{2i-1}A_{i-1,i}a_{i,i-1}/a_{i,i}+(-1)^{2i}A_{i,i}+(-1)^{2i+1}A_{i+1,i}a_{i,i+1}/a_{i,i}\right]^{-1}\times$$

$$\times\left[\cfrac{\frac{1}{2}A_{i,n+1}}{E-\cfrac{2\left(A_{i,n+1}\alpha_{i,i}\right)^{-1}A_{i-1,n+1}\alpha_{i,i-1}}{E+\cfrac{2A_{i-1,n+1}\alpha_{i,i-1}}{A_{i,n+1}\alpha_{i,i}}-\ddots-\cfrac{\frac{A_{1,n+1}\alpha_{i,1}}{A_{2,n+1}\alpha_{i,2}}}{E+\frac{A_{1,n+1}\alpha_{i,1}}{A_{2,n+1}\alpha_{i,2}}}}}-\cfrac{\frac{1}{2}A_{i,n+1}}{E-\cfrac{2\left(A_{i,n+1}\alpha_{i,i}\right)^{-1}A_{i+1,n+1}\alpha_{i,i+1}}{E+\cfrac{2A_{i+1,n+1}\alpha_{i,i+1}}{A_{i,n+1}\alpha_{i,i}}-\ddots-\cfrac{\frac{A_{n,n+1}\alpha_{i,n}}{A_{n-1,n+1}\alpha_{i,n-1}}}{E+\frac{A_{n,n+1}\alpha_{i,n}}{A_{n-1,n+1}\alpha_{i,n-1}}}}}\right].$$

А для кожного відношення $a_{i,k}\,/\,a_{i,k+1}$, у свою чергу, можна записати

$$\frac{a_{i,k}}{a_{i,k+1}}=\frac{A\begin{bmatrix}2 & \ldots & i-1\\ 2 & \ldots & i-1\end{bmatrix}\prod\limits_{s=i+1}^{k-1}A_{s,s+1}A\begin{bmatrix}k+1 & \ldots & \ldots & n\\ k+1 & \ldots & \ldots & n\end{bmatrix}}{A\begin{bmatrix}2 & \ldots & i-1\\ 2 & \ldots & i-1\end{bmatrix}\prod\limits_{s=i+1}^{k}A_{s,s+1}A\begin{bmatrix}k+2 & \ldots & \ldots & n\\ k+2 & \ldots & \ldots & n\end{bmatrix}}=\frac{A\begin{bmatrix}k+1 & \ldots & \ldots & n\\ k+1 & \ldots & \ldots & n\end{bmatrix}}{A_{k,k+1}A\begin{bmatrix}k+1 & \ldots & \ldots & n\\ k+1 & \ldots & \ldots & n\end{bmatrix}}=$$

$$=\frac{A_{k+1,k+1}}{A_{k,k+1}}-\frac{A_{k+1,k+2}A_{k+2,k+1}}{A_{k,k+1}\left(A^{-1}\begin{bmatrix}k+3 & \ldots & n\\ k+3 & \ldots & n\end{bmatrix}A^{T}\begin{bmatrix}k+2 & \ldots & n\\ k+2 & \ldots & n\end{bmatrix}^{T}\right)^{T}}=$$

$$=\ldots=\frac{A_{k+1,k+1}}{A_{k,k+1}}-\cfrac{A_{k+1,k+2}A_{k+2,k+1}/A_{k,k+1}}{A_{k+2,k+2}-\cfrac{A_{k+3,k+2}A_{k+2,k+3}}{A_{k+3,k+3}-\cfrac{A_{k+3,k+4}A_{k+4,k+3}}{A_{k+4,k+4}-\ddots\atop \ddots-\cfrac{A_{n-1,n}A_{n,n-1}}{A_{n,n}}}}}.$$

Отже, одержуємо аналітичне розвинення невідомих X_i даної розрідженої системи лінійних алгебраїчних рівнянь у скінченні матричні ланцюгові дроби.

5.1.1. Обчислювальні характеристики алгоритму

Знайдемо необхідну кількість записів для символьного розв'язування задачі та кількість операцій для числової реалізації алгоритму множення матриць $A_{ij}\times A_{kl}$.

ТВЕРДЖЕННЯ**.** *Нехай деяка обчислювальна задача з вхідними даними {A_i} розв'язується на комп'ютері за алгоритмом у* $(A_1,A_2,\ldots,A_n)$ *і складається з k кроків* ψ_j *(* $j=\overline{1,k}$*). Якщо на кожному кроці алгоритму ψ(A) реалізується хоча б один запис виду* $\psi_{ji_1}(A)\cdot\psi_{ji_2}(A)$*, який використовує результат попереднього кроку, то загальна складність* Q_ψ *задачі буде не меншою* $2^k\cdot m^{2\,2}$*, але не більшою* H^k *записів, де H — найбільша ширина алгоритму на k кроках* [29].

Використаємо це твердження для оцінки складності алгоритму з погляду комп'ютерної алгебри [62]. Для чисел $X_i \ \left(i=\overline{1,n}\right)$ на одному поверсі реалізація алгоритму вимагає одне блочне множення, одне блочне ділення, одне блочне додавання, а для *n* поверхів *3n* операцій, тобто по *n* блочних множень, ділень та додавань.

Обчислення показують, що для визначення всіх $A_{i,k} \ / \ A_{i,k+1}$ потрібно *5k* записів, якщо $k<i$, і $i \cdot 5(n-k)$, якщо $k>i$. Тому необхідно виконати таку кількість записів:

$$5\sum_{k=1}^{i} k + 5\sum_{k=i}^{n} n-k = 5\left[\frac{(1+i)i}{2} + n(n-i+1) - \frac{(i+n)(n-i+1)}{2}\right] = 5\left[i^2 + \frac{n^2}{2} + \frac{n}{2} - ni\right].$$

Отже, загальна складність методу становить:

$$5\sum_{i=1}^{n}\left[i^2 + \frac{n^2}{2} + \frac{n}{2} - ni\right] = \frac{5}{2}(n^3+n).$$

Для порівняння оцінимо складність символьного варіанту реалізації методу прогонки, яка в числовому вигляді є однією з найефективніших обчислювальних схем для даного класу задач. Відомо [16, 95], що алгоритм прогонки реалізується за співвідношеннями:

$$X_i = a_{i+1}X_{i+1} + b_{i+1}, \quad a_{i+1} = \frac{a_{i,i+1}}{a_{i,i} - a_{i,i-1}}, \qquad b_{i+1} = \frac{a_{i,n+1} + a_{i,i-1}b_i}{a_{i,i} - a_{i,i-1}}$$

для прямого та зворотного ходу відповідно.

Кількість записів для методу прогонки, який реалізується співвідношеннями:

$$X_1 = a_2X_2 + b_2, \ldots \quad, a_2 = a_{1,2} / a_{1,1}, \ldots, b_2 = a_{1,n+1} / a_{1,1},$$

$$x_2 = a_3x_3 + b_3, \quad a_3 = \frac{a_{2,3}}{a_{2,2} - a_{2,1}}, \quad b_3 = \frac{a_{2,n+1}}{a_{2,2} - a_{2,1}},$$

$$x_{n-1} = a_nx_n + b_n, \quad a_n = \frac{a_{n-1,n}}{a_{n-1,n-1} - a_{n-1,n-2}}, \quad b_n = \frac{a_{n-1,n+1}}{a_{n-1,n-1} - a_{n-1,1}},$$

оцінюється відповідно до наведеного раніше твердження. Розрахунки свідчать, що для обчислення a_{i+1} та b_{i+1} необхідно записати по i операцій блочного додавання, блочного множення та блочного ділення.

Для обчислення кожного X_k треба записати $\sum\limits_{k=1}^{i} k = (1+i)i\big/2$ записів, а для обчислення всіх x_i $i = \overline{1,n}$ потрібно $(n^3 + n^2 + 2n)\ /\ 4$ записів.

Отже, з погляду комп'ютерної алгебри, запропонований алгоритм суттєво переважає класичний алгоритм прогонки. Він може бути реалізований як в аналітичному, так і в числовому вигляді. Для реалізації запропонованого алгоритму в числовому вигляді потрібно по *6n* блочних додавань і блочних ділень і *4n* блочних множень, оскільки в цьому випадку результати обчислень проміжних дробів можна використати багаторазово.

Відзначимо, що описаний алгоритм можна також застосовувати й у випадку систем з проріdженими матрицями такого вигляду:

$$\begin{pmatrix} A_{1,1} & 0 & 0 & A_{1,k} & & & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & \ddots & A_{2,k+1} & & 0 \\ 0 & & \ddots & 0 & \ddots & & 0 \\ A_{k,1} & \ddots & 0 & \ddots & 0 & \ddots & A_{n-k,n} \\ 0 & A_{k+1,2} & \ddots & & \ddots & 0 & \ddots \\ 0 & 0 & \ddots & \ddots & 0 & \ddots & 0 \\ 0 & 0 & 0 & A_{n,n-k} & \ddots & 0 & A_{n,n} \end{pmatrix}$$

Матриці можуть також бути і обрамленими з однієї або двох сторін. Системи з подібним заповненням розпадаються на k систем вигляду (5.1), кожна з яких матиме порядок $n\ /\ k$. Алгоритм розв'язування кожної з одержаних k систем буде аналогічно до розглянутого.

5.2. Кліткові обчислювальні методи розв'язання систем лінійних алгебричних рівнянь

З появою перших ЕОМ почали розвиватися кліткові методи розв'язання систем лінійних алгебричних рівнянь. Невеликий об'єм оперативної пам'яті (ОП) не дозволяв одночасно зберігати всі елементи даної системи. Тому їх записували на зовнішніх носіях (магнітних дисках або барабанах), а в міру необхідності виконувалися обміни між зовнішньою та оперативною пам'яттю ЕОМ. У зв'язку з цим з'явились модифікації для цілого ряду числових методів, які дозволяють зменшити число обмінів між зовнішньою пам'яттю (ЗП) та ОП, а також скоротити об'єми проміжних обчислень [103].

Сучасні ЕОМ мають значно більшу оперативну пам'ять. Проте й порядок систем, які зустрічаються в практичних застосуваннях, суттєво зріс. Так, найголовнішим джерелом витрат часу в реалізаціях числових методів є обміни між загальною оперативною пам'яттю і пам'яттю окремих процесорів, обміни між процесорами і т. д. Часто в подібних випадках допомагають блочні алгоритми.

Кліткові методи виявились привабливими й для багатопроцесорних обчислювальних систем [103], послідовно-паралельних систем [103], систем з ієрархічною організацією пам'яті і т.д. Характерно, що в цих випадках дуже часто ефективно працюють ті кліткові алгоритми, мета створення яких була зовсім іншою [103].

Найчастіше вживаною операцією для кліткових методів є така:

$$A + BC,\ A \in R^{n \times n}, \tag{5.6}$$

зі щільно заповненими матрицями *A, B, C* розмірності $n \times n$. Крім цієї операції можуть виконуватися також інші довільні операції матричної алгебри. Однак кількість операцій вигляду (5.6), як правило, становить головну частину від загальної кількості арифметичних операцій. Отже, доцільна розробка ефективних алгоритмів обчислення (5.6) для використання їх на високопродуктивних функціональних пристроях, апаратна реалізація на надвеликих інтегральних схемах, або за допомогою систолічних масивів.

Усі обчислювальні методи можна поділити на два класи: ортогональні методи і методи, засновані на неунітарних перетвореннях. Кліткові методи є узагальненням звичайних числових методів, хоча часто й неочевидним. Аналіз алгоритмів лінійної алгебри показує, що більшість відомих обчислювальних схем дозволяє одержати й кліткові аналоги. Однак, якщо операції додавання і множення допускають природне узагальнення, то операція ділення не відповідає множенню на обернену підматрицю. В [103] наведено приклад невиродженої матриці, поділеної на клітки так, що кожна клітка є виродженою. Отже, для кліткових методів неможливо в загальному випадку побудувати стратегію вибору максимального елемента. Множення кліток є некомутативним, замість трикутного розкладу отримаємо блочно-трикутний і т.д.

Розглянемо неортогональні методи. Їх, в свою чергу, поділяють на методи без вибору провідного елемента і методи, де такий вибір провідного елемента здійснюється. Розглянемо задачу, що ґрунтується на побудові кліткових алгоритмів, а також дослідженні їх обчислювальної стійкості.

Запишемо систему лінійних алгебричних рівнянь у блочному вигляді:

$$\begin{pmatrix} A_{11} & A_{12} & \dots & A_{1m} \\ A_{21} & A_{22} & \dots & A_{2m} \\ \dots & \dots & \dots & \dots \\ A_{m1} & A_{m2} & \dots & A_{mm} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ \dots \\ X_m \end{pmatrix} = \begin{pmatrix} B_1 \\ B_2 \\ \dots \\ B_m \end{pmatrix}. \tag{5.7}$$

Припустимо, що всі головні мінори порядку kp $(k = 1,2,...,m)$ не дорівнюють нулю і розглянемо відсічені системи. Під відсіченою системою порядку k розуміють систему з матрицею, утвореною першими k рядками та k стовпцями матриці A початкової системи. Права частина отримується з елементів решти стовпців або рядків матриці A:

$$\begin{pmatrix} A_{11} & A_{12} & \dots & A_{1k} \\ A_{21} & A_{22} & \dots & A_{2k} \\ \dots & \dots & \dots & \dots \\ A_{k1} & A_{k2} & \dots & A_{kk} \end{pmatrix}^{-1} \begin{pmatrix} X_{1,k} \\ X_{2,k} \\ \dots \\ X_{k,k} \end{pmatrix} = \begin{pmatrix} A_{1,k+1} \\ A_{2,k+1} \\ \dots \\ A_{k,k+1} \end{pmatrix} \tag{5.8}$$

і

$$\begin{pmatrix} A_{11} & A_{21} & \dots & A_{k1} \\ A_{12} & A_{22} & \dots & A_{k2} \\ \dots & \dots & \dots & \dots \\ A_{1k} & A_{2k} & \dots & A_{kk} \end{pmatrix}^{-1} \begin{pmatrix} Y_{1,k} \\ Y_{2,k} \\ \dots \\ Y_{k,k} \end{pmatrix} = \begin{pmatrix} A_{k+1,,1} \\ A_{k+1,2} \\ \dots \\ A_{k+1,k} \end{pmatrix}. \tag{5.9}$$

Припустимо, що системи вигляду (5.8) і (5.9) уже будь-яким способом розв'язані до порядку $k-l$ включно.

Тоді для невідомих X_{kk} і Y_{kk} на основі [103] можна записати:

$$Y_{kk} = \left\{ \begin{array}{l} \left[A_{kk} - (X_{1,k-1} X_{2,k-1} \dots X_{k-1,k-1})(A_{k,1}\ A_{k,2} \dots A_{k,k-1})^T\right]^{-1} \times \\ \left[A_{k+1,k} - (X_{1,k-1} X_{2,k-1} \dots X_{k-1,k-1})(A_{k+1,1}\ A_{k+1,2} \dots A_{k+1,k-1})^T\right] \end{array} \right\}$$

і

$$X_{kk} = \left\{ \begin{array}{l} \left[A_{kk} - (Y_{1,k-1} Y_{2,k-1} \dots Y_{k-1,k-1})(A_{1,k}\ A_{2,k} \dots A_{k-1,k})^T\right]^{-1} \times \\ \left[A_{k,k+1} - (Y_{1,k-1} Y_{2,k-1} \dots Y_{k-1,k-1})(A_{1,k+1}\ A_{2,k+1} \dots A_{k-1,k+1})^T\right] \end{array} \right\}.$$

Після обчислення X_{kk} і Y_{kk} відразу можна приступати до визначення X_{sk} та Y_{sk}, $(s = 1,2,\dots,k-1)$. Для цього, розв'язуючи (5.8), можна скористатися співвідношенням:

$$\left. \begin{array}{l} X_{s,k} = B_{s,k+1} - \sum\limits_{i=s+1}^{k} B_{s,i} X_{i,k}; \\ B_{s,i} = \left(A_{i,s} - \sum\limits_{j=1}^{s-1} Y_{j,s-1} A_{s,j}\right)^{-1} \left(A_{s,i} - \sum\limits_{j=1}^{s-1} Y_{j,s-1} A_{i,j}\right) \left(i = \overline{s+1, n+1}\right) \end{array} \right\}. \tag{5.10}$$

Ще можна використати рекурентні формули:

$$\left.\begin{aligned} Y_{s,k} &= B_{k+1,s} - \sum_{i=s+1}^{k} B_{i,s} Y_{i,k}; \\ B_{i,s} &= \left(A_{s,s} - \sum_{j=1}^{s-1} Y_{j,s-1} A_{j,s} \right)^{-1} \left(A_{i,s} - \sum_{j=1}^{s-1} X_{j,s-1} A_{j,i} \right) \left(i = \overline{s+1,n} \right) \end{aligned}\right\}. \quad (5.11)$$

Обґрунтування формул (5.10) і (5.11) базується на твердженні, що

$$B_{i,s} = \begin{vmatrix} A_{1,1} & A_{2,1} & \dots & A_{s-1,1} & A_{i,1} \\ A_{1,2} & A_{2,2} & \dots & A_{s-1,2} & A_{i,2} \\ \dots & \dots & \dots & \dots & \dots \\ A_{1,s} & A_{2,s} & \dots & A_{s-1,s} & A_{i,s} \end{vmatrix} \Big/ \begin{vmatrix} A_{1,1} & A_{2,1} & \dots & A_{s-1,1} & A_{s,1} \\ A_{1,2} & A_{2,2} & \dots & A_{s-1,2} & A_{s,2} \\ \dots & \dots & \dots & \dots & \dots \\ A_{1,s} & A_{2,s} & \dots & A_{s-1,s} & A_{s,s} \end{vmatrix}. \quad (5.12)$$

Тут $A_{i,j}, B_{i,s}$ - блоки розміру $p \times p$. У даній рівності елементами є матриці розмірності $p \times p$. Отже, на основі [69], процес розв'язування початкової системи можна реалізувати сукупністю рекурентних співвідношень

$$\left.\begin{aligned} B_{k,i} &= \left(A_{k,k} - \sum_{j=1}^{s-1} Y_{j,k-1} A_{j,k} \right)^{-1} \left(A_{k,i} - \sum_{j=1}^{s-1} Y_{j,k-1} A_{j,i} \right); \\ X_{k,k} &= B_{k,k+1} \quad \left(k = \overline{1,n};\ i = \overline{k+1,n};\ s = \overline{k-1,1} \right); \\ X_{s,k} &= B_{s,k+1} - \sum_{i=s+1}^{k} B_{s,i} X_{i,k}. \end{aligned}\right\}$$

та

$$\left.\begin{aligned} B_{i,k} &= \left(A_{k,k} - \sum_{j=1}^{s-1} X_{j,k-1} A_{j,k} \right)^{-1} \left(A_{i,k} - \sum_{j=1}^{s-1} X_{j,k-1} A_{i,j} \right); \\ Y_{k,k} &= B_{k+1,k} \quad \left(k = \overline{1,n-1},\ \ i = \overline{k+1,n};\ s = \overline{k-1,,1} \right); \\ Y_{s,k} &= B_{k+1,s} - \sum_{i=s+1}^{k} B_{i,s} Y_{i,k}. \end{aligned}\right\}$$

Процес обчислень розпочинають з лівого верхнього кута матриці системи і послідовно знаходять розв'язки відсічених кліткових систем усіх порядків.

Зупинимось тепер на обчислювальних характеристиках методу.

Особливості реалізації алгоритму на ЕОМ

Складність алгоритму. Знайдемо кількість операцій, потрібних для реалізації алгоритму на ЕОМ. Нехай $n = pm$, тобто дана система складається з m блоків p-го порядку. Обчислимо Q_k - кількість операцій, необхідних для розв'язування системи, що складається з k блоків p-го порядку. Для обчислення добутку

$$(X_{1,k-i-j}, X_{2,k-i-j}, \ldots, X_{k-i,k-i-j})(A_{k+1,1}, A_{k+1,2}, \ldots, A_{k+1,k-1})^T$$

потрібно $(k-i-1)p^3$ множень та $(k-i-1)p^3$ операцій додавання. Також для визначення $B_{k+1,s}$ використовуються по $(k-i-1)p^3$ множень і додавань. Нескладні підрахунки показують, що $Q_k = (m-k)2l^3 + 4/3l^3 + lk^2$.

Всього потрібно розв'язати m пар таких систем. Тому загальну кількість арифметичних операцій знайдемо так:

$$Q = \sum_{k=1}^{m} Q_k = \sum_{k=1}^{m} \left[(m-k)2l^3 + 4/3l^3 + lk^2\right]$$

Отже, остаточно з точністю до головного члена $Q \approx 4n^3/3$ і тоді потрібно виконати $2/3n^3$ операцій множення.

5.3. Застосування алгоритмів теорії $\lambda-$матриць для динамічних матричних моделей Леонтьєва

У багатьох застосуваннях виникає необхідність розгляду більш загального випадку – систем лінійних алгебричних рівнянь з $\lambda-$матрицями для динамічних моделей Леонтьєва. Дана область числових методів мало вивчена. Це стосується як теорії, так і самого факту існування ефективних методів, які можна реалізувати на комп'ютері.

Нехай статистична модель Леонтьєва має вигляд:

$$X = Ax + y, \tag{5.13}$$

де $x = (x_1, x_2, \ldots, x_n)^T$ та $y = (y_1, y_2, \ldots, y_n)^T$ – відповідно вектори валової й кінцевої продукції, а матриця $A = (a_{ij}), (i, j = \overline{1,n})$ – матриця коефіцієнтів прямих матеріальних витрат.

Розглянемо систему лінійних алгебричних рівнянь Леонтьєва з прямокутними $\lambda-$матрицями. Запишемо систему (5.13) у вигляді:

$$X(\lambda) = A(\lambda)X(\lambda) + y(\lambda). \qquad (5.14)$$

Звідси $(A(\lambda) - E)X(\lambda) = -y(\lambda)$, де $A(\lambda)-$матриця розміру $n \times m$, елементи якої є поліномами від λ. Припустимо, що значення для λ і коефіцієнтів многочленів беруться з деякого поля F так: елементи матриці $A(\lambda)$ обчислюються для деякого часткового значення λ, наприклад $\lambda = \lambda_0 F^{n\times m}$, $y(\lambda)-$вектор $\left(a_{1,m}(l), a_{2,m}(l), \ldots, a_{n,m}(l)\right)^T$. Матриця $A(\lambda)$ і вектор $y(\lambda)$ мають степінь l, тобто $a_{ij}(\lambda) = \sum_{k=0}^{l} a_{i,j}\lambda^{l-k}$, $\left(i = \overline{1,n},\ j = \overline{1,m+1}\right)$.

Подамо матрицю $A(\lambda)$ і вектор $y(\lambda)$ у вигляді матричних многочленів: $A(\lambda) = \lambda^l A_0 + \lambda^{l-1}A_1 + \ldots + A_l$ та $y(\lambda) = \lambda^l y_0 + \lambda^{l-1}y_1 + \ldots + y_l$.

Розв'язок системи шукаємо у вигляді частки двох поліномів з невідомими коефіцієнтами:

$$X(t) = \sum_{i=0}^{nl}\lambda^{nl-i}x_i \Big/ \sum_{i=0}^{nl}\lambda^{nl-i}z_i\,, \qquad (5.15)$$

де k - ранг матриці $A(\lambda) - E$, $X_1, \ldots, X_{\ln}$ - вектори розмірності r, а $z_0, z_1, \ldots, z_{\ln}$ - скалярні величини. Тоді систему (5.14) можна записати так:

$$\frac{\sum\limits_{i=0}^{nl}\lambda^{nl-i}x_i}{\sum\limits_{i=0}^{nl}\lambda^{nl-i}z_i} = \left(\lambda^l A_0 + \lambda^{l-1}A_1 + \ldots + \lambda A_{l-1} + A_l\right)\cdot\frac{\sum\limits_{i=0}^{nl}\lambda^{nl-i}x_i}{\sum\limits_{i=0}^{nl}\lambda^{nl-i}z_i} + \sum_{i=0}^{l}\lambda^{l-i}y_i.$$

Домноживши праву і ліву частини рівності на $\sum\limits_{i=0}^{nl}\lambda^{nl-i}z_i$, матимемо:

$$\sum_{i=0}^{nl}\lambda^{nl-i}x_i = \left(\lambda^l A_0 + \lambda^{l-1}A_1 + \ldots + \lambda A_{l-1} + A_l\right)\cdot\sum_{i=0}^{nl}\lambda^{nl-i}x_i + \sum_{i=0}^{l}\lambda^{l-i}y_i\cdot\sum_{i=0}^{nl}\lambda^{nl-i}x_i.$$

Звідки,

$$\left(\lambda^{nl}x_0+\lambda^{nl-1}x_1+\lambda^{nl-2}x_2+\ldots+\lambda^2x_{nl-2}+\lambda x_{nl-1}+x_{nl}\right)\times$$
$$\times\left(E-\lambda^lA_0-\lambda^{l-1}A_1-\ldots-\lambda A_{l-1}-A_l\right)+\left(\lambda^ly_0+\lambda^{l-1}y_1+\ldots+y_l\right)\times$$
$$\times\left(\lambda^{nl}z_0+\lambda^{nl-1}z_1+\lambda^{nl-2}z_2+\ldots+\lambda^2z_{nl-2}+\lambda z_{nl-1}+z_{nl}\right)=0,$$

відкривши дужки та згрупувавши члени у лівій і правій частинах рівності при однакових степенях λ одержимо:

$$\lambda^{nl}x_0+\lambda^{nl-1}x_1+\lambda^{nl-2}x_2+\ldots+\lambda^2x_{nl-2}+\lambda x_{nl-1}+x_{nl}=\left(\lambda^lA_0+\lambda^{l-1}A_1+\ldots+\lambda A_{l-1}+A_l\right)\times$$
$$\times\left(\lambda^{nl}x_0+\lambda^{nl-1}x_1+\lambda^{nl-2}x_2+\ldots+\lambda^2x_{nl-2}+\lambda x_{nl-1}+x_{nl}\right)+\left(\lambda^ly_0+\lambda^{l-1}y_1+\ldots+\lambda y_{l-1}+y_l\right)\times$$
$$\times\left(\lambda^{nl}z_0+\lambda^{nl-1}z_1+\lambda^{nl-2}z_2+\ldots+\lambda^2z_{nl-2}+\lambda z_{nl-1}+z_{nl}\right).$$

Звідси,

$$\Big[\lambda^{nl}x_0+\lambda^{nl-1}x_1+\lambda^{nl-2}x_2+\ldots+\lambda^2x_{nl-2}+\lambda x_{nl-1}+x_{nl}-\lambda^{l(n+1)}x_0A_0-\lambda^{l(n+1)-1}x_0A_1-\ldots-\lambda^{ln+1}x_0A_{l-1}-$$
$$-\lambda^{ln}x_0A_0-\lambda^{l(n+1)-1}\times x_1A_0-\lambda^{l(n+1)-2}x_1A_1-\ldots-\lambda^{ln}x_1A_{l-1}-\lambda^{ln-1}x_1A_l-\lambda^{l(n+1)-2}x_2A_0-\lambda^{l(n+1)-3}\times$$
$$\times x_2A_1-\ldots-\lambda^{ln-1}x_2A_{l-1}-\lambda^{ln-2}x_2A_l-\ldots-\lambda^{l+2}x_{nl-2}A_0-\lambda^{l+1}x_{nl-2}A_1-\ldots-\lambda^3x_{nl-2}A_{l-1}-\lambda^2x_{nl-2}A_l-$$
$$-\lambda^3x_{nl-1}A_0-\lambda^lx_{nl-1}A_1-\ldots-\lambda^2x_{nl-1}A_{l-1}-\lambda x_{nl-1}A_l-\lambda^lx_{nl}A_0-\lambda^{l-1}x_{nl}A_1-\ldots-\lambda x_{nl}A_{l-1}-x_{nl}A_l\Big]+$$
$$+\Big[\lambda^{l(n+1)}y_0z_0+\lambda^{l(n+1)-1}\times y_0z_1+\lambda^{l(n+1)-2}y_0z_2+\lambda^{l+2}y_0z_{nl-2}+\lambda^{l+1}y_0z_{nl-1}+\lambda^ly_0z_{nl}+\lambda^{l(n+1)-1}\times$$
$$\times y_1z_0+\lambda^{l(n+1)-2}y_1z_1+\lambda^{l(n+1)-3}y_1z_2+\ldots+\lambda^{l+1}y_1z_{nl-2}+\lambda^ly_1z_{nl-1}+\lambda^{l-1}y_1z_{nl}+\ldots+\lambda^{nl}y_lz_0+$$
$$+\lambda^{nl-1}y_lz_1+\lambda^{nl-2}y_lz_2+\ldots+\lambda^2y_lz_{nl-2}+\lambda z_{nl-1}y_l+z_{nl}y_l\Big]=0.$$

Оскільки $A(\lambda)\in F^{n\times m}$, а $y(\lambda)\in F^n$, то в загальному випадку, прирівнявши коефіцієнти при однакових степенях, одержимо числову систему $m\left(l+s+1\right)$ рівнянь з $m\left(s+1\right)$ невідомими x_{ij} і $\left(s+1\right)$ невідомими z_j:

$$
\begin{cases}
-A_0x_0 - y_0z_0 = 0; \\
-A_1x_0 - A_0x_1 - y_0z_1 - y_1z_0 = 0; \\
-A_1x_1 - A_0x_2 - y_0z_2 - y_1z_1 = 0; \\
\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\
-\sum_{s=0}^{l} A_s x_{p-s} - \sum_{s=0}^{l} y_s z_{p-s} = 0; \\
\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\
x_{nl-1} - A_l x_{nl-1} - x_{nl}A_{l-1} - z_{nl-1}y_l = 0; \\
x_{nl} - A_l x_{nl} - z_{nl}y_l = 0
\end{cases}
\tag{5.16}
$$

Даний підхід дозволяє проводити аналіз розв'язання прямокутних систем, а також одержувати деякі теоретичні оцінки.

Можна застосувати кілька обчислювальних алгоритмів для розв'язання одержаних систем числових рівнянь (5.16). Розглянемо деякі ефективні підходи.

Схема розрізання. Схематичну матрицю числової системи подамо у вигляді:

$$
A^* = \begin{pmatrix}
\times & I & & & & & & & & \\
\times & I & \times & I & & & & & & \\
\times & I & \times & I & \times & I & & & & \\
\times & I & \times & I & \times & I & \times & I & & \\
 & & \times & I & \times & I & \times & I & \times & I \\
 & & & & \times & I & \times & I & \times & I \\
 & & & & & & \times & I & \times & I \\
 & & & & & & & & \times & I
\end{pmatrix},
\tag{5.17}
$$

де через "$\times$" позначені квадратні клітки, а через "I" – прямокутні блоки.

Для визначеності вважатимемо, що система (5.15) складається з $m\left(l+s+1\right)$ рівнянь і має $m\left(l+s+1\right)+k$ невідомих $\left(k \geq 1\right)$. Нехай, $rank\,A_0 = \min\left\{m,n\right\}$.

Перестановкою стовпців систему (5.17) перетворимо так, щоб її матриця набула заповнення

$$
\begin{pmatrix}
\times & & & & & I & & & & \\
\times & \times & & & & I & I & & & \\
\times & \times & \times & & & I & I & I & & \\
\times & \times & \times & \times & & I & I & I & I & \\
 & \times & \times & \times & \times & & I & I & I & I \\
 & & \times & \times & \times & & & I & I & I \\
 & & & \times & \times & & & & I & I \\
 & & & & \times & & & & & I
\end{pmatrix}, \tag{5.18}
$$

Загальна стратегія розв'язання цієї системи залежить від її рангу. Спочатку розглянемо найбільш трудомісткий з обчислювального погляду варіант числової системи. Для цього припустимо, що всі головні мінори схематично нарисованої матриці не дорівнюють нулю. Тоді перші $m\left(l+s+1\right)$ невідомих можуть бути виражені через k останніх розв'язків системи (5.16). При зроблених припущеннях про характер матриці (5.18) підматриця, утворена блоками "$\times$", буде невиродженою.

Отже, схематично процес перетворення матриці системи можна зобразити у вигляді:

$$
\begin{pmatrix}
\times & & & & & I & & & & \\
\times & \times & & & & I & I & & & \\
\times & \times & \times & & & I & I & I & & \\
\times & \times & \times & \times & & I & I & I & I & \\
 & \times & \times & \times & \times & & I & I & I & I \\
 & & \times & \times & \times & & & I & I & I \\
 & & & \times & \times & & & & I & I \\
 & & & & \times & & & & & I
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
O & & & & & I & & & & \\
 & O & & & & I & I & & & \\
 & & O & & & I & I & I & & \\
 & & & O & & I & I & I & I & \\
 & & & & O & I & I & I & I & I \\
 & & & & & I & I & I & I & I \\
 & & & & & I & I & I & I & I \\
 & & & & & I & I & I & I & I
\end{pmatrix}.
$$

Для реалізації алгоритму на обчислювальній системі потрібно виконати з точністю до головного члена Cm^4l^3 арифметичних операцій. Причому величина сталої C залежить від вибраного алгоритму зведення матриці до діагонального вигляду.

Схема діагоналізації. Економічні алгоритми можна також одержати з урахуванням того, що матриця системи (5.16) має клітково-тепліцеве стрічкове заповнення у рядках. Незважаючи на те що клітки не є квадратними , а прямокутними, в загальному випадку цю обставину вдається використати досить вдало.

Однорідну кліткову систему з прямокутними блоками (5.16), для наочності подальших міркувань, запишемо у матричному вигляді:

$$
\begin{pmatrix}
A_0 & & & & & & & & & & & & & & -B_0 \\
A_1 & A_0 & & & & & & & & & & & & & -B_1 \\
A_2 & A_1 & A_0 & & & & & & & & & & & & -B_2 \\
\dots & \dots & \dots & \dots & \dots & & & & & & & & & & \dots \\
A_1 & A_{l-1} & A_{l-2} & \dots & A_0 & & & & & & & & & & -B_l \\
 & A_l & A_{l-1} & \dots & A_1 & A_0 & & & & & & & & & \\
 & & A_l & \dots & A_2 & A_1 & A_0 & & & & & & & & \\
 & & & & \dots & \dots & \dots & \dots & & & & & & & \\
 & & & & & \dots & \dots & \dots & \dots & & & & & & \\
 & & & & & & A_1 & A_{l-1} & \dots & A_0 & & & & & \\
 & & & & & & & A_l & A_{l-1} & \dots & A_0 & & & & \\
 & & & & & & & & & A_l & A_{l-1} & \dots & A_0 & & -B_0 \\
 & & & & & & & & & & A_l & \dots & A_1 & A_0 & -B_0 & -B_1 \\
 & & & & & & & & & & & & \dots & \dots & \dots & \dots \\
 & & & & & & & & & & & & A_1 & A_{l-1} & -B_{l-1} & -B_l \\
 & & & & & & & & & & & & & A_l & -B_l &
\end{pmatrix}
\begin{pmatrix}
X_0 \\ X_1 \\ X_2 \\ \dots \\ X_N \\ X_{N+1} \\ X_{N+2} \\ \dots \\ X_{Nl+N} \\ \dots \\ z_0 \\ z_1 \\ z_2 \\ \dots \\ \dots \\ z_{Nl+N-1} \\ z_{Nl+N}
\end{pmatrix} = 0,
$$

де A_{ij} - прямокутні клітки розмірності $n\left(n+1\right)$.

Процес перетворення матриці системи схематично подамо у вигляді:

$$
\begin{pmatrix}
ґ & & & & & n & & & & \\
ґ & ґ & & & & n & n & & & \\
ґ & ґ & ґ & & & n & n & n & & \\
ґ & ґ & ґ & ґ & & n & n & n & n & \\
 & ґ & ґ & ґ & ґ & & n & n & n & n \\
 & & ґ & ґ & ґ & & & n & n & n \\
 & & & ґ & ґ & & & & n & n \\
 & & & & ґ & & & & & n
\end{pmatrix}
\text{Ю}
\begin{pmatrix}
O & & & & & n & & & & \\
 & O & & & & n & n & & & \\
 & & O & & & n & n & n & & \\
 & & & O & & n & n & n & n & \\
 & & & & O & n & n & n & n & n \\
 & & & & & n & n & n & n & n \\
 & & & & & n & n & n & n & n \\
 & & & & & n & n & n & n & n
\end{pmatrix}
\text{Ю}
$$

$$
\text{Ю}
\begin{pmatrix}
O & & & & & & & & & n & & & & & \\
 & O & & & & & & & & n & n & & & & \\
 & & O & & & & & & & n & n & n & & & \\
 & & & O & & & & & & n & n & n & n & & \\
 & & & & O & & & & & n & n & n & n & n & \\
 & & & & & O & & & & n & n & n & n & n & n \\
 & & & & & & O & & & n & n & n & n & n & n \\
 & & & & & & & O & & n & n & n & n & n & n \\
 & & & & & & & & O & n & n & n & n & n & n
\end{pmatrix}.
$$

Опишемо схему обчислень:

Крок 1. Переставимо місцями стовпці матриці так, щоб у лівій частині залишилось $l + s + 1$ кліткових стовпців, по m стовпців у кожному. Якщо ранг хоча б однієї з матриць A_0 чи A_l дорівнює m, то в лівому блоці матриць можна позбутись піддіагональних або наддіагональних елементів відповідно.

Крок 2. Без обмеження загальності припустимо, що $rank\, A_0 = m$ і саме m стовпців, які є лінійно незалежними, залишаються зліва в кожному з кліткових стовпців. За еквівалентними перетвореннями в лівому блоці матриці системи позбудемось піддіагональних елементів. Цей блок матриці має клітково-тепліцеву структуру. З урахуванням цього достатньо провести перетворення одного кліткового стовпця $\left(A_{m_0}\ A_{m_1}\ A_{m_l}\right)^T$, а потім одержати з його елементів решту. Для виконання елементарних перетворень лівого блоку достатньо виконати $4/3m^3 + lm^2 + l^2m^2$ операцій множення і додавання.

Виявляється, що тоді достатньо просто можна врахувати й зміни всіх стовпців у правій частині матриці. Оскільки еквівалентні перетворення не змінюють тепліцевої структури заповнення правого блоку матриць [51], то права частина теж залишиться клітково-тепліцевою. У зв'язку з цим достатньо виконати всі перетворення над першим кліточним стовпцем правого блоку. А отже, ненульові елементи решти стовпців отримаємо, послідовно зсуваючи елементи попереднього стовпця на m позицій вниз. Отже, для перетворення елементів правого блоку потрібно виконати з точністю до головного члена, $\frac{4}{3}(n + 1 - m)(l + s + 1)(m^2 + lm^2)$ адитивних і таку ж кількість мультиплікативних операцій на обчислювальній системі.

Крок 3. На даному етапі виконаємо зведення матриці до трапецеподібної форми. Після виконання попереднього кроку в правому блоці позбудемося піддіагональних елементів в останніх $(l + 1)m$ рівняннях. Для цього необхідно виконати не більше ніж $4/3l^3m^3$ операцій множення і додавання.

Крок 4. На цьому етапі розв'язується система з матрицею трапецеподібної форми. Для цього потрібно не більше ніж $(l+s+1)ml^2$ арифметичних дій. Загальна кількість операцій всіх кроків розглянутого алгоритму, з точністю до головного члена, обчислюється так:

$$Q = 4/3m^3 + 2lm^2 + (l+s+1)(n+1-m)m^2(l+1) + 4/3l^3m^3.$$

На початку було зроблено припущення, що $rank\,A_0 = \min\{m,n\}$. Нехай також без обмеження загальності, найбільший головний мінор $A\begin{bmatrix} 1 & 2 & 3 & \dots & rank \\ 1 & 2 & 3 & \dots & rank \end{bmatrix}$ ненульовий. Тоді блок з клітково-тепліцевим заповненням зводимо до трикутної, а потім і до діагональної форми, затративши по $2/3n^3 + ln^2$ додавань і множень.

У силу тепліцевого заповнення відпадає необхідність у перетворенні решти кліткових стовпців. Виявляється, що тоді достатньо просто врахувати зміни стовпців правого блоку. Дійсно, оскільки елемент a_{ij} даного стовпця $(j = n^2(l+1)+2, n^2(l+1)+3, \dots)$ одержується з $\left[n^2(l+1)+1\right]$-го стовпця зсувом його елементів на $n^2\left[j - n^2(l+1) - 1\right]$ позицій вниз.

Крок 5. Після виключення в системі (5.6) векторів X_l залишиться система порядку ln відносно $y_0, y_1, \dots, y_{nl}$. Якщо позначити через r її ранг $(r \le ln)$, то розв'язання системи у загальному випадку вимагатиме $2nlr + 4/3r^3$ операцій.

Тому загальна складність розглянутого алгоритму складає $2/3(n^3l^3 + mn^3 + n^3)$ адитивних і $2/3(n^3l^3 + mn^3 + n^3)$ мультиплікативних операцій. Для реалізації методу на ЕОМ потрібно $(l+2)n^2$ комірок пам'яті.

5.4. Розв'язання вироджених і погано обумовлених систем

Система лінійних рівнянь може мати єдиний розв'язок, нескінченну множину розв'язків або не мати розв'язку взагалі (система несумісна). В останньому випадку задача не позбавлена змісту. Тоді шукають не розв'язок у звичайному розумінні, а такий набір значень невідомих, після підстановки яких у систему ліві частини "найменше" відхиляються від правих. Слово "найменше" можна інтерпретувати по-різному, і різним інтерпретаціям відповідає метод найменших квадратів, метод найменших модулів та ін.

Під час розв'язування систем алгебричних рівнянь часто трапляється, що малі похибки правих частин або заданих коефіцієнтів призводять до великих похибок у розв'язках.

Похибки можуть виникати під час вимірювання, обчислення або заокруглення елементів матриць систем або правих частин. Такі лінійні системи алгебричних рівнянь називають некоректно поставленими або погано обумовленими. Ця термінологія застосовується з часів Ж. Адамара [29], який вважав вивчення некоректно поставлених задач недоцільним, тому що їх завжди можна після уточнення математичної моделі поставити коректно (розумно). Причому, Ж. Адамар не навів способу, як некоректну задачу зробити коректною, адже некоректні задачі зустрічаються частіше, ніж коректно поставлені задачі. Ця обставина випливає з труднощів побудови адекватних реальним процесам математичних моделей.

Відшуковуючи розв'язки моделі Леонтьєва, на практиці часто наштовхуються на такі ситуації, як виродженість або погана обумовленість матриць нормативних коефіцієнтів, велика розмірність задачі, накопичення похибки під час обчислень. Крім цього, обчислювальна задача має вхідні дані та обчислений результат. Розв'язування систем лінійних алгебричних рівнянь призводить до виникнення похибок, пов'язаних з неточністю початкових даних або похибок заокруглення. Крім того, майже завжди виникають похибки обчислень вже в межах самої задачі (внаслідок неточного виконання арифметичних операцій).

Навіть за умови, що попередні вимірювання обчислення проводились з високою точністю і для розв'язання задачі вибрано стійкий метод обчислень, похибки вхідних даних, хоча й малі, але все ж будуть. Ці похибки певною мірою впливають на розв'язок системи. Тому фактично замість розв'язку системи $X(\lambda)=A(\lambda)X(\lambda)+y(\lambda)$ ми отримуємо розв'язок якоїсь іншої системи $X(\lambda)=\overline{A}(\lambda)X(\lambda)+\overline{y}(\lambda)$. З огляду на це, виникає питання, як похибки вхідної інформації впливають на якість розв'язку системи? Точної відповіді на це питання немає, тому спробуємо хоча б оцінити вплив похибок вхідної інформації на розв'язок. Запишемо систему $X(\lambda)=A(\lambda)X(\lambda)+y(\lambda)$ або $y(\lambda)=\left(E-A(\lambda)\right)X(\lambda)$ у вигляді $X(\lambda)=\dfrac{y(\lambda)}{E-A(\lambda)}$. Вважаючи $A(\lambda)$ і $y(\lambda)$ змінними, продиференціюємо цю рівність по λ. Одержимо:

$$dX(\lambda)=\frac{dy(\lambda)\cdot\left(E-A(\lambda)\right)-y(\lambda)d\left(E-A(\lambda)\right)}{\left(E-A(\lambda)\right)^2}$$

$$dX(\lambda)=\frac{dy(\lambda)}{E-A(\lambda)}+y(\lambda)\frac{d\left(E-A(\lambda)\right)}{\left(E-A(\lambda)\right)^2}=\frac{dy(\lambda)}{E-A(\lambda)}+y(\lambda)\cdot\left(E-A(\lambda)\right)^{-2}d\left(E-A(\lambda)\right)=$$

$$=\frac{dy(\lambda)}{E-A(\lambda)}+y(\lambda)\left(-\left(E-A(\lambda)\right)^{-1}\right)=\frac{dy(\lambda)}{E-A(\lambda)}-\frac{y(\lambda)}{E-A(\lambda)}.$$

Тому $dX(\lambda)=\dfrac{dy(\lambda)-y(\lambda)}{E-A(\lambda)}$, або $dX(\lambda)=\left(E-A(\lambda)\right)^{-1}dy(\lambda)-y(\lambda)$. Доведемо наступну теорему.

ТЕОРЕМА 5.4.1. *Якщо* A - *матриця порядку* n ґ n *і* $\lambda_1,\lambda_2,\ldots,\lambda_n$ – *власні значення цієї матриці, то*

$$\max_{1\le i\le n}\left|\lambda_i\right|\le\|A\|. \tag{5.19}$$

ДОВЕДЕННЯ. Нехай λ – будь-яке власне значення матриці A. Тоді існує нульовий вектор x такий, що $Ax=\lambda x$. Тому $\|Ax\|=\|\lambda x\|$ або $\|Ax\|=|\lambda|\cdot\|x\|$. Із визначення норми матриці випливає, що $\|Ax\| \le |A| \cdot \|x\|$. Звідси $|\lambda| \cdot \|x\| \le |A| \cdot \|x\|$, або $|\lambda| \le \|A\|$.

Отже, якщо $\lambda_1, \lambda_2, ..., \lambda_n$ – власні значення матриці A, то для них виконується співвідношення (5.19), що й треба було довести.

На основі вище поданої теореми 5.4.1. запишемо:

$$\|dX(\lambda)\| = \left\|(E - A(\lambda))^{-1}\right\| \cdot \|dy(\lambda) - y(\lambda)\| \le \left\|(E - A(\lambda))^{-1}\right\| \cdot \|dy(\lambda) - y(\lambda)\| \le \left\|A(\lambda)^{-1}\right\| \cdot (\|dy(\lambda)\| - \|y(\lambda)\|).$$

Нехай $\delta_A = \dfrac{\|dA(\lambda)\|}{\|A(\lambda)\|}$, $\delta_y = \dfrac{\|dy(\lambda)\|}{\|y(\lambda)\|}$, $\delta_x = \dfrac{\|dx(\lambda)\|}{\|x(\lambda)\|}$ – відносні похибки вхідних даних і розв'язку. Тоді поділивши нерівність

$\|dX(\lambda)\| \le \left\|A(\lambda)^{-1}\right\| \cdot (\|dy(\lambda)\| - \|y(\lambda)\|)$ на $\|x\|$, одержимо

$$\frac{\|dX(\lambda)\|}{\|x(\lambda)\|} \le \left\|A(\lambda)^{-1}\right\| \cdot \left(\frac{\|dy(\lambda)\|}{\|x(\lambda)\|} - \frac{\|y(\lambda)\|}{\|x(\lambda)\|}\right).$$

Оскільки $\|y(\lambda)\| \le \|(E - A(\lambda)) \cdot x(\lambda)\|$, то $\|y(\lambda)\| \le \|(A(\lambda)) \cdot x(\lambda)\|$ або $\|x(\lambda)\| \ge \dfrac{\|y(\lambda)\|}{\|A(\lambda)\|}$.

Тому $\delta_x \le \|A(\lambda)\| \cdot \left\|(A(\lambda))^{-1}\right\| \cdot (\delta_A + \delta_y)$, або d_x J $cond\,A$ Ч$(d_A + d_y)$, де $cond\,A(\lambda) = \|A(\lambda)\| \cdot \left\|(A(\lambda))^{-1}\right\|$.

Величина $cond\,A(\lambda)$ називається числом обумовленості матриці A. Це число є мірою невизначеності розв'язку системи $X(\lambda) = A(\lambda)X(\lambda) + y(\lambda)$ за неточних вхідних даних. Оскільки $\left\|A(\lambda)(A(\lambda))^{-1}\right\| = 1$ і $\left\|A(\lambda)(A(\lambda))^{-1}\right\| \le \|A(\lambda)\| \cdot \left\|(A(\lambda))^{-1}\right\|$, то $cond\,A(\lambda) \ge 1$. Якщо $cond\,A(\lambda)$ близьке до одиниці, то можна вважати, що похибки вхідних даних переносяться у розв'язок без помітного їх збільшення. Системи з такою властивістю називаються добре обумовленими. За великих значень числа обумовленості можливе різке збільшення похибки в розв'язку порівняно з рівнем похибок в

$A(\lambda)$ і $y(\lambda)$. Системи з великим числом обумовленості називають погано обумовленими.

Означення 5.4.1. Матриця $A(\lambda)$ називається *виродженою* (*особливою, сингулярною*), якщо її визначник дорівнює нулю, тобто $\det A(\lambda) = 0$.

Розглянемо властивості вироджених матриць:

1) Рядки і стовпці виродженої матриці – лінійно-залежні (не утворюють нетривіальних лінійних комбінацій, рівних нулю).
2) Ранг матриці менший за розмірність матриці.
3) У виродженої матриці немає оберненої матриці (хоча є псевдообернена матриця). Псевдообернена матриця – узагальнена оберненій матриці. Матриця псевдообернена до матриці $A(\lambda)$ позначається $A^{+}(\lambda)$. Найбільш відомим є псевдообернення Мура-Пенроуза, яке було описано Е.Х. Муром у 1920 р. і Роджером Пенроузом у 1955 р. Псевдообернена матриця застосовується для знаходження найкращого наближення, напр. методом найменших квадратів, розв'язку систем лінійних алгебричних рівнянь.
4) Якщо матриця $A(\lambda)$ розміру $n \times n$ вироджена, то система рівнянь $A(\lambda)x(\lambda) = 0$ має ненульові розв'язки. Множина цих розв'язків позначається $\ker A(\lambda)$ і є лінійним підпростором n-вимірного простору, відмінним від нуля.
5) Матриця $A(\lambda)$ є вироджена тоді і тільки тоді, якщо серед її власних значень є нулі.

Міра обумовленості матриці

Введемо число обумовленості $m(A)$ для матриці системи лінійних алгебричних рівнянь

$$m(A) = \max_{x \neq 0,\ z \neq 0} \left[\frac{\|Ax\|}{\|Az\|} \cdot \frac{\|z\|}{\|x\|} \right], \tag{5.20}$$

яку можна також записати у вигляді:

$$\mu(A)=\left[\max_{x\neq 0}\frac{\|Ax\|}{\|x\|}\Big/\min_{z\neq 0}\frac{\|Az\|}{\|z\|}\right]=\sqrt{\frac{\sigma_{\max}}{\sigma_{\min}}},\qquad(5.21)$$

де $\sigma_{\max},\sigma_{\min}$ - відповідно максимальне та мінімальне власне значення матриці $A^T A$, A^T - транспонована матриця по відношенню до матриці A.

Дослідимо величину $\mu(A)$ в оцінці похибки при збуренні початкових значень вектора Y і матриці A. Розглянемо поряд з Y вектор $Y+\Delta Y$. Нехай x, $x+\Delta x$ – відповідно розв'язок моделей Леонтьєва $X=A(t)X(t)+Y(t)$ або $X=\left(E-A(t)\right)^{-1}Y(t)$, та $X+\Delta X=\left(E-A(t)\right)^{-1}\left(Y(t)+\Delta Y(t)\right)$. Тоді $\Delta X=\left(E-A(t)\right)^{-1}\Delta Y(t)$. Згідно з означенням $\mu(A)$ із (5.20) матимемо

$$\mu(A)=\max_{z,\Delta z}\left[\frac{\left\|\left(E-A(t)\right)^{-1}z\right\|}{\left\|\left(E-A(t)\right)^{-1}\Delta z\right\|}\cdot\frac{\|\Delta z\|}{\|z\|}\right]=\left[\frac{\left\|\left(E-A(t)\right)^{-1}\cdot X\right\|}{\left(E-A(t)\right)^{-1}\Delta X}\cdot\frac{\|\Delta X\|}{\|X\|}\right]$$

Звідки

$$\frac{\|\Delta X\|}{\|X\|}\leq\mu(A)\cdot\frac{\|\Delta Y\|}{\|Y\|},\qquad(5.22)$$

тобто $\mu(A)$ найменша константа, для якої при всіх $\Delta X, X, \Delta Y, Y$ виконується нерівність (5.22)

Отже, $\mu(A)$ виконує роль множника при зростанні відносної похибки розв'язку. Це означає, що внаслідок зміни вектора в правій частині системи зміни в розв'язку збільшуються в $\mu(A)$ раз. Інакше кажучи, нерівність (5.22) означає, що $\mu(A)$ обмежує зверху відношення відносної невизначеності вектора X до відносної невизначеності вектора Y. Важливо підкреслити, що оцінка (5.22) точна. Це свідчить про те, що за відповідного підбору векторів Y ΔY можна досягти рівності. Оцінка (5.22) досяжна, значить не можна дати більш точну оцінку, ніж (5.22) для довільних векторів Y, ΔY незалежно від їхньої величини.

Аналогічний зміст числа обумовленості в загальному випадку, коли збурюється і вектор Y і матриця A:

$$(X+\Delta X)=\left(E-(A(t)+\Delta A(t))\right)^{-1}\left(Y(t)+\Delta Y(t)\right).$$

Наведемо завершальну оцінку [29]

$$\frac{\|\Delta X\|}{\|X\|}\le\frac{\dfrac{\|\Delta A(t)\|}{\|A(t)\|}+\dfrac{\|\Delta Y(t)\|}{\|Y(t)\|}}{1-\dfrac{\|\Delta A(t)\|}{\|A\|}\mu(A)}\cdot\mu(A),\tag{5.23}$$

що отримана в припущенні невиродженості матриці A та умови

$$\frac{\|\Delta A(t)\|}{\|A(t)\|}\mu(A)<1.$$

Якщо A – вироджена матриця, то отримаємо випадок $\mu(A)=\infty$, оскільки $\min\dfrac{\|A(t)\cdot z\|}{\|z\|}=0$ і оцінки (5.22) та (5.23) втрачають сенс. Якщо ж A - невироджена матриця, то $\mu(A)=\|A(t)\|\cdot\|A^{-1}(t)\|$, де $A(t)$ обернена до $A^{-1}(t)$- матриця, тобто матриця, яка задовольняє відношення $A(t)A^{-1}(t)=A^{-1}(t)A(t)=E$.

Отже, з оцінок (5.22), (5.23) випливає, що число обумовленості матриці $A(t)$ визначальне для того, наскільки розв'язок системи $X=\left(E-A(t)\right)^{-1}Y(t)$ стійкий до збурень компонент вектора правих частин $Y(t)$ та матриці коефіцієнтів $A(t)$.

5.5. Розв'язання систем алгебричних рівнянь з $\lambda-$матрицями на паралельних обчислювальних системах

Дослідження паралельності описаних алгоритмів розв'язання матричних систем з λ-матрицями проведемо для двох найбільш часто використовуваних моделей – концепції необмеженого паралелізму [24], а також для комп'ютерів з конкретною архітектурою – багатопроцесорною обчислювальною системою типу MIMD [103].

Вибір MIMD-архітектури продиктований тим, що серед паралельних обчислювальних систем обчислювальні комп'ютери цього типу є одними з найбільш розповсюджених.

Концепція необмеженого паралелізму. Припускається, що в користуванні є p-процесорна обчислювальна система наступної ідеалізованої моделі. Паралельна система має довільне потрібне число ідентичних процесорів а також яку завгодно велику оперативну пам'ять, одночасно доступну всім процесорам. Кожний процесор за одиницю часу може виконати абсолютно 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 результат у пам'ять. Після закінчення обчислювального процесу всі результати залишаються у пам'яті.

Однією з найважливіших характеристик алгоритму є його висота. *Висотою алгоритму* H_p називається число тактів, за яке може бути отриманий результат на p-процесорній обчислювальній системі. Висота алгоритму характеризує число рівнів або ярусів обчислювальної схеми. Задача побудови паралельного алгоритму полягає в перетворенні початкового алгоритму так, щоб зменшити його висоту. Цим самим забезпечується можливість виконати обчислення на багатопроцесорній машині за якомога менший час, що в деяких випадках має вирішальне значення. Крім висоти алгоритму його ще характеризують економічність та прискорення [135].

Прискоренням S_p прийнято називати відношення часу реалізації деякого алгоритму на одному процесорі T_1 паралельної обчислювальної системи до часу T_p його реалізації на p процесорах $S_p = \dfrac{T_1}{T_p}$.

Економічністю E_p називають відношення $E_p = \dfrac{S_p}{p}$.

Оцінимо тепер ефективність використання описаних раніше числових алгоритмів розв'язання матричних систем з λ-матрицями на паралельних обчислювальних системах.

Схема розрізання. Згідно з раніше проведеними розрахунками, час реалізації алгоритму на однопроцесорній системі пропорційний складності алгоритму $T_1 = Cl^3n^3$ (стала C визначається набором складових алгоритмів). Оцінюючи ефективність розпаралелювання алгоритму, розглянемо декілька випадків.

Багатопроцесорна система має p J nl *процесорів.* На першому етапі визначаються матриці $C_{11}^{-1}C_{12}$ та $C_{11}^{-1}Y_1$, а тому $H_1 = n^3l / p$. На наступному етапі знаходяться $C_{22} - C_{21}(C_{11}^{-1}Y_1)$ та $Y_2 - C_{21}(C_{11}^{-1}C_{12})$, отже $H_2 = Cn^3l^3 / p$. Третій етап потребує розв'язання щільної системи порядку nl і тому $H_3 = \dfrac{4}{3}n^3l^3$. Четвертий етап полягає у визначенні $Y_1 - C_{12}Z$ та $X = C_{11}^{-1}\left(Y_1 - C_{21}Z\right)$ і, отже, $H_4 = \dfrac{4n^3l^3}{3p}$. Отже, $T_p = \dfrac{T_1}{p}$, $S_p = p$, $E_p = 1$.

Обчислювальна система складається з p J n^2l^2 *процесорів.* Для визначення матриць $C_{11}^{-1}C_{12}$ та $C_{11}^{-1}Y_1$ потрібно $H_1 = ln^3/ p + n\log_2 n$ тактів. На наступному етапі знаходяться матриці $C_{22} - C_{21}(C_{11}^{-1}Y_1)$ та $Y_2 - C_{21}(C_{11}^{-1}C_{12})$, отже, $H_2 = Cn^3l^3 / p$. Третій етап вимагає розв'язання щільної лінійної системи порядку nl, внаслідок чого $H_3 = nl\log_2 nl$. А на четвертому етапі матриці $Y_1 - C_{12}Z$ та $X = C_{11}^{-1}\left(Y_1 - C_{21}Z\right)$ можна обчислити за $H_4 = C_1 nl\log_2\left(nl\right) + C_2 l^3 n^3 / p$ тактів.

Отже, $T_p = [C_1 nl\log_2\left(nl\right) + C_2 l^3 n^3]/ p$.

Остаточно,

$$S_p = \frac{T_p}{T_1} = \frac{C_1 nl\log_2\left(nl\right) + C_2 l^3 n^3}{Cl^3 n^3 / p} \approx p;\; E_p \approx 1.$$

Неважко зробити висновок, що в обох випадках $(p \;\mathrm{J}\; nl$ та $p \;\mathrm{J}\; n^2 l^2)$ алгоритм розрізання для матричних λ – систем дуже добре розпаралелюється.

Паралельні обчислювальні системи класу MIMD. Подальші міркування проведемо для машин класу MIMD [103] за умов обмеженості обчислювальних ресурсів і технічних можливостей. Припускається, що багатопроцесорна система має таку структуру.

Нехай є деякий універсальний серійний комп'ютер, розміщений на периферії. Він здійснює обслуговування паралельної обчислювальної системи і управління зовнішніми пристроями. У системі існує комутатор зв'язку, до каналів якого під'єднані управляючі пристрої (УП), арифметичні процесори (АП) і спеціальні процесори (СП).

Обчислювальний процес за загальною програмою організується за допомогою периферійного універсального комп'ютера. Припустимо, що АП мають певний, відносно великий об'єм оперативної пам'яті і здатні виконувати числову і логічну обробку інформації в автономному і керованому режимах. Оператори програми, управляючі інструкції та інформація, яка обробляється, надходять в АП або за необхідністю, або у випадку звільнення в 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 за загальною програмою, узгодження роботи АП і управління комутаційними і периферійними процесорами виконується за допомогою управляючих процесорів.

Керуючий комп'ютер проводить управління всіма периферійними пристроями (дисководами, моніторами, принтерами і т. д.) та виконує роль своєрідного буфера при обмінах інформацією між АП і зовнішньою пам'яттю (під ЗП надалі матимемо на увазі магнітні носії інформації або пристрої, аналогічні за швидкодією). На відміну від АП, управляюча машина може проводити обмін даними одночасно з χ АП. Тут χ – кількість каналів зв'язку між АП і периферійною ЕОМ. Крім того, є кілька каналів для зв'язку з периферійними пристроями, що можуть бути залучені одночасно.

Подібна багатопроцесорна ОС дозволяє виконати одночасно обмін даними периферійного процесора з ЗП таількома (до χ включно) АП, а також проводить незалежну обробку даних у решті АП.

Використовуватимемо такі позначення:

M – об'єм опертивної пам'яті АП (у словах); N – об'єм оперативної пам'яті управляючої машини (у словах),

t_1 – час пошуку певного набору даних у зовнішній пам'яті; t_2 – час передачі одного слова із зовнішньої пам'яті в управляючу машину;

t_3 – середній час комутаційних затримок при встановленні зв'язку між двома модулями;

t_4 – час передачі одного слова між двома модулями;

t_5 – середній час виконання однієї арифметичної операції і нехай $t = \mathrm{t}_4/\mathrm{t}_5$.

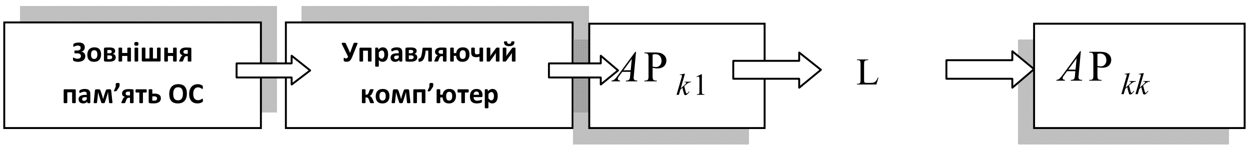
Обмін інформацією ПОС може проводитися одним з описаних нижче способів [103].

Ланцюговий принцип передачі даних – організація передачі даних з допоміжного АП у всі процесори групи, що проводять обробку інформації. Процес обміну виконується за правилом: АП, що отримав набір даних при обміні з деяким іншим АП, виконує передачу набору 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 котрому допоміжний процесор після передачі даних у деякий модуль відразу починає обробку власної інформації. Аналогічно діють всі АП, що входять у дану групу.

Централізований принцип передачі даних – процес передачі інформації з управляючої машини одночасно в r АП $(r \le \chi)$.

Магістральна обробка даних – процедура перетворення наборів даних $D_j \quad (j = 1, 2, \ldots, k)$, що знаходяться в АП, за допомогою наборів даних $d_j \quad (j = 1, 2, \ldots, c)$, 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 такою схемою



Розглянемо тепер особливості застосування описаних процедур передачі даних. Нехай необхідно передати набір даних з допоміжного АП в усі АП групи, кількість яких дорівнює p. За ланцюгової процедури передачі даних треба $\lceil \log_2 p \rceil$ переривань обміну. Можна також спочатку передати вказаний набір в універсальний комп'ютер, а потім скористатися централізованим принципом пересилання інформації.

Очевидно, що коли $\lceil \log_2 p \rceil < 1 + \left\lceil \frac{p-1}{c} \right\rceil$, то процедура централізованої передачі даних вимагатиме меншої кількості переривань обміну, а отже, й меншого часу. Тут $\lceil c \rceil$ – найменше ціле число, більше c .

Проаналізуємо тапер процес розв'язання систем рівнянь з λ – матрицями на багатопроцесорній обчислювальній системі з MIMD–архітектурою.

Рядкова схеми розв'язання. За даним алгоритмом розв'язується рядкова система порядку $n^2 l$ з шириною рядка nl. Для її розв'язання скористаємося рядковою модификацією другого алгоритму відсічених систем. Як показано раніше, на однопроцесорній ЕОМ потрібно виконати $T = nl \times n^2 l / 2 = l^3 n^5 / 2$ множень і таку ж кількість додавань. Для оцінки часу T_p виконання алгоритму на p-процесорній системі класу MIMD розглянемо декілька способів обміну даними між АП.

Припустимо, що розміри матриці $A(\lambda)$ такі, що всі початкові коефіцієнти поліномів $a_{i,j}(\lambda)$, а також коефіцієнти проміжних величин $b_{k,l}^{(j)}, x_k^{(j)}, y_k^{(j)}$ розміщені у пам'яті p арифметичних процесорів, тобто $n^2(l+1) + n^3 l^2 \le pM$.

Припустимо, що в кожному з p арифметичних процесорів розміщено по q стовпців матриці $A(\lambda)$, тобто $n = pq$. Для мінімізації часу простою АП треба розподілити дані так, щоб у j-ому АП $(j = 1,2,,...n)$ містилися $j + kp$-ті вектор-стовпці $(k = 0,1,2,...,q+1)$ матриці $A(\lambda)$. Крім того, необхідно зарезервувати в кожному АП масив розмірності $qn^2 l^2$ для зберігання проміжних результатів.

Після зроблених припущень перейдемо до розгляду обчислювальних схем, зорієнтованих на різні способи передачі даних.

Ланцюговий спосіб передачі інформації. Нехай обчислювальний процес знаходиться на k-ому кроці. На цьому етапі для обчислення поліномів $b_{i,j}$ і проміжних невідомих $x_j^{(k)}$ і $y_j^{(k)}$ необхідно обчислити скалярні добутки nl-ого порядку. Припускаємо, що всі необхідні для обчислень елементи $a_{i,j}$ початкової матриці знаходяться у відповідних АП. Визначення проміжних невідомих $x_j^{(k)}$ і $y_j^{(k)}$, а також величин $b_{i,j}$ вимагає пересилки в кожний процесор по $(k - q)$ невідомих $x_s^{(k-1)}$ і $y_s^{(k-1)}$, а також по $2kpql$ значень $b_{i,j}$. За обраного способу передачі даних на це потрібний час $]\log_2 p[\{t_3 + 2[(k - q) + knl]t_4\}$. До того ж, на виконання всіх обчислень затратимо час, рівний $2q(nl^2/2 + kl)t_5$.

Отже, на довільному k-ому етапі загальний час, затрачений обчислювальною системою на обчислення і обмін інформацією, складатиме

$$T_k =]\log_2 p\,[\{t_3 + 2[(k - q)l + kpql]t_4\} + 2q(nl^2/2 + kl]t_5).$$

Тому загальний час Т на реалізацію всіх k етапів дорівнює:

$$T =]\ \log_2 p\,[\{t_3 + 2[(n^4l^3/2 + n^4l^3/2]t_4\} + 2q(nl^2/2 + n^4l^2l/2]t_5 =$$

$$=]\ \log_2 p\,[\{t_3 + 2[(n^4l^3/2 + n^4l^3/2]t_4\} + n/p(n^2l^2 + n^5l^3)t_5)$$

і, відповідно, прискорення

$$S_p = \frac{1/2l^3n^5t_5}{]\ \log_2 p\,[\{t_3 + 2[(n^4l^3/2 + n^4l^3/2]t_4\} + n/p(n^2l^2 + n^5l^3)/pt_5} \cong p/2$$

за ефективності $E_p = S_p/p = 1/2$.

<u>Централізований спосіб передачі даних.</u> Нехай обчислювальний процес знаходиться на k-му кроці. Тоді на цьому етапі для обчислення поліномів $b_{i,j}$ і проміжних невідомих $x_{i,j}^{(k)}$ і $y_{i,j}^{(k)}$ потрібно обчислити скалярні добутки nl-го порядку. Вважаємо, що всі необхідні для обчислень елементи $a_{i,j}$ початкової матриці знаходяться у відповідних АП. Визначення проміжних невідомих $x_j^{(k)}$ і $y_j^{(k)}$, а також величин $b_{i,j}$ вимагає пересилання в кожний процесор по $(k - q)$ невідомих $x_s^{(k-1)}$ і $y_s^{(k-1)}$, а також по $2kpql$ значень $b_{i,j}$. За вибраного способу передачі даних на це потрібний буде час $p\{t_1 + t_2 + 2[(k - q) + knl]t_4\}$.

Крім того, на виконання всіх обчислень затратимо час, рівний $2q(nl^2/2 + kl)t_5$.

Отже, на довільному k-му етапі загальний час, затрачений обчислювальною системою на обчислення і обмін інформацією, складе:

$$T_k = p\{t_1 + t_2 + 2[(k - q)l + kpql]t_4\} + 2q(nl^2/2 + kl]t_5.$$

Отже, загальний час T на реалізацію всіх k етапів

$$T = p\{t_1 + t_2 + 2[(n^4l^3/2 + n^4l^3/2]t_4\} + 2q(nl^2/2 + n^4l^2/2]t =$$

$$= p\{t_1 + t_2 + 2[(n^4 l^3/2 + n^4 l^3/2]t_4\} + n/p(n^2 l^2 + n^5 l^3)t_5)$$

і, відповідно, прискорення

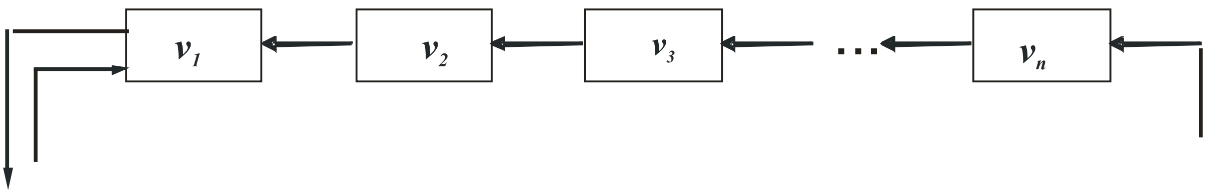
$$S_p = \frac{1/2 l^3 n^5 t_5}{p\{t_1 + t_2 + 2[(n^4 l^3/2 + n^4 l^3/2]t_4\} + n/p(n^2 l^2 + n^5 l^3)/p t_5}$$

за ефективності $E_p = S_p/p = 1/2$.

Схеми діагоналізації та розрізання. Перелічені методи, природно, відрізняються за своєю реалізацією від двох попередніх і одним від іншого. Однак в них є одна спільна риса - коефіцієнти при t_1, t_2, t_3 і t_4 у всіх схемах, принаймні в n раз менші, ніж коефіцієнти при t_5. У зв'язку з цим можна зробити висновок, що прискорення S_p і ефективність E_p для цих схем з точністю до множника рівні прискоренню і ефективності для вже розглянутих алгоритмів.

Систолічні масиви для розв’язання систем із прямокутними λ-матрицями. За схемою діагоналізації задача зводиться до розв’язання числових систем з клітково-тепліцевим заповненням. Безперечна перевага такого підходу – можливість застосування добре розроблених векторних *T*-алгоритмів, які допускають ефективну реалізацію на так званих *систолічних масивах*.

Так, для системи числових лінійних алгебричних рівнянь $Az = b$ з тепліцевою матрицею порядку N введення даних і виведення результатів обчислень можна виконати [103] за такою схемою:



$$z_N \quad 0, p, q \;\; d_1^{(0)},\; h_1^{(0)},\; w_1^{(0)}, 0, 0$$

$$z_{N-1} \;\; b_1, 0, 0,\; d_2^{(0)},\; h_2^{(0)},\; w_2^{(0)}, 0, 0$$

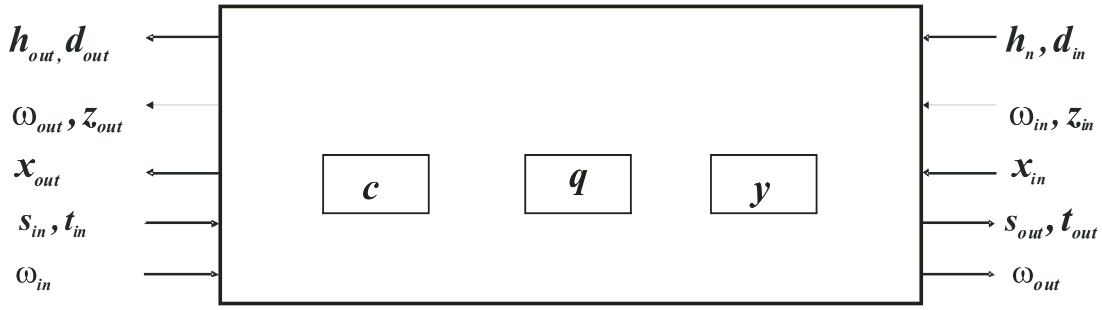
$$z_{N-2} \;\; b_2, 0, 0,\; d_3^{(0)},\; h_3^{(0)},\; w_3^{(0)}, 0, 0$$

. . . .

$$z_0, b_N\, 0, 0,\; d_N^{(0)},\; h_N^{(0)},\; z_N^{(0)}, x_N^{(0)}.$$

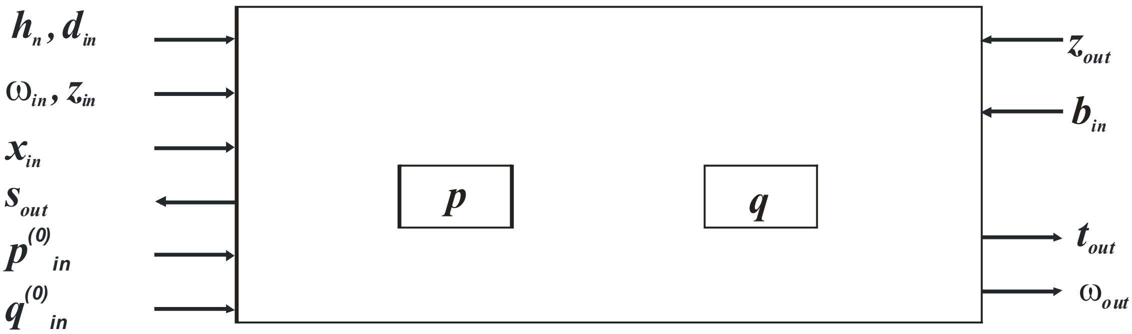
Систолічний масив складається з N комірок, позначених відповідно $v_1, v_2, ..., v_N$. Використовується лише два типи комірок: перший відповідає коміркам v_i для $1 \le i \le N-1$, інший – лише комірці, яка дещо відрізняється від решти.

Комірка першого типу має вигляд прямокутника, що приймає та передає інформацію через ліву та праву межі:



Зліва є три входи s_{in}, t_{in}, w_{in} і п'ять виходів $h_{out}, d_{out}, w_{out}, z_{out}, x_{out}$. Справа розміщені п'ять входів та три виходи для приймання і передавання відповідно величин $h_{in}, d_{in}, w_{in}, z_{in}, \hat{x}_{in}$ і $s_{out}, t_{out}, w_{out}$. Комірка має три регістри для зберігання даних між послідовними тактами роботи.

Розглянемо тепер будову іншого типу комірки. Символічно комірка P_0 зображена на схемі як прямокутник, що має на лівій межі три входи і один вихід $b_{in}, p_{in}^{(0)}, q_{in}^{(0)}$ та z_{out}, а на правій границі – п'ять входів $h_{in}, d_{in}, w_{in}, z_{in}, \hat{x}_{in}$ і три виходи $s_{out}, t_{out}, w_{out}$. Комірка P_0 має два регістри для зберігання нормованих множників p_k, q_k; початкові і вихідні на момент опрацювання значення цих регістрів позначені p_{in}, q_{in} і p_{out}, q_{out}



Згідно з [18], загальний час роботи такого систолічного масиву для розв'язання системи складатиме *5(nl+1)(n+1)-6* тактів.

Отже, розглянуті алгоритми розв'язання систем лінійних алгебричних рівнянь можна успішно використовувати на багатопроцесорних обчислювальних комплексах. Для p-процесорних ЕОМ при p J n досягається повна завантаженість усіх процесорів для кожного алгоритму. Характерно, що ефективність і прискорення практично збігаються для концепції необмеженого паралелізму та ЕОМ з MIMD-архітектурою.

5.6. Аналіз обчислювальної стійкості алгоритмів розв'язання систем лінійних алгебричних рівнянь з $\lambda-$матрицями

Стійкість числового алгоритму. Погано обумовленою може бути не сама задача, а лише алгоритм, вибраний для її розв'язання. Якщо обчислений розв'язок сильно 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 [11]; [12] .

Нехай обчислювальна задача з початковими даними $A(\lambda)$ розв'язується з допомогою деякого точного алгоритму g. Результат K розв'язання задачі запишемо у вигляді: $K = g(A(\lambda))$.

При комп'ютерній реалізації алгоритму g всі його операції будуть замінені машинними псевдоопераціями, а сам алгоритм – деяким машинним алгоритмом $K = g_t(A(\lambda))$, результат виконання якого запишемо

$$X_t(\lambda) = g_t(A(\lambda)).$$

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

$$X_t(\lambda) = g(A_t(\lambda)).$$

Різницю $K = X_t(\lambda) - X(\lambda)$ називають еквівалентним збуренням, яке також характеризує похибку розв'язання задачі. Останню рівність запишемо у вигляді: $X_t(\lambda) = g(A(\lambda) + K)$. $X_t(\lambda)$ можна розглядати як розв'язок тієї ж задачі зі збуреними на K вхідними даними. Для отримання кількісної оцінки впливу похибок заокруглення використовують так званий зворотний аналіз похибок [97].

Для аналізу похибок заокруглення, що виникають на ЕОМ, використаємо методику Дж.Х. Уілкінсона [97].

Запис чисел в обчислювальних системах. Два режими обчислень використовують при реалізації алгоритмів на ЕОМ – з фіксованою та плаваючою комою. Останній є основним для комп'ютера. У цьому режимі кожне ненульове число x представляється парою чисел a та b: $x = h^b a$, де h – основа машинної арифметики, ціле b - порядок числа, a – мантиса, така що

$$\frac{1}{h} \text{ J } |a| \text{ J } 1.$$

Для представлення мантиси виділяється t розрядів (кількість знаків після коми для запису a). Найчастіше в обчислювальних системах використовується двійкова система числення, тому ще говорять, що число x має t-бітову довжину.

Запис $z = fl\left(z * y\right)$ означає, що стандартне число z з плаваючою комою одержане зі стандартних чисел x та y в результаті виконання на ЕОМ деякої арифметичної операції (додавання, віднімання, множення або ділення) в режимі з плаваючою комою. Для сучасних ЕОМ похибки окремих арифметичних операцій є невеликими: $z = fl\left(x * y\right)$ є $\left(x * y\right)\left(1 + e\right)$, де $\left|e\right|$ J h^{-t}.

Така оцінка має місце при режимі блокування заокруглення, який найпоширеніший на сучасних комп'ютерах. На довільній ЕОМ для деякої множини чисел в околі машинного нуля справедлива рівність $\varepsilon = -1$, незалежно від кількості розрядів t.

Отже, для дійсних чисел справджуються оцінки:

$$\begin{cases} |e| \text{ J } h^{-t}, & \text{для } fl(x) \text{№} 0, \\ e = -1, & \text{для } fl(x) = 0, \text{ при } x \text{№} 0 \end{cases}$$

Для обчислень з фіксованою комою (крапкою) використовують позначення $z = fi\left(x * y\right)$. Вважається, що операції додавання і віднімання в цьому режимі не мають похибок [97] $z = fi\left(x \pm y\right)$ є $\left(x \pm y\right)$, а ділення і множення породжують малі похибки заокруглення:

$z = fi\left(x * y\right)$ є $\left(x * y\right) + e$, $z = fi\left(x \text{ ё } y\right)$ є $\left(x \text{ ё } y\right) + e$, де $\left|e\right|$ J h^{-t+1}.

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

Деякі результати з теорії похибок. Нехай $B\left(\lambda\right)$ – прямокутна матриця, яка перетворюється в процесі реалізації деякого числового методу. Припустимо, що за математичним алгоритмом будується послідовність:

$$B\left(\lambda\right) = B_0\left(\lambda\right),\ B_1\left(\lambda\right),\ B_2\left(\lambda\right),\ldots,\ B_n\left(\lambda\right),$$

де $B_i\left(\lambda\right) = P_i\left(\lambda\right)B_{i-1}\left(\lambda\right)$ $\left(i = 1, 2, \ldots, n\right)$ і матриці $P_i\left(\lambda\right)$ – невироджені.

Тоді

$$B_n(\lambda) = P(\lambda) B(\lambda). \tag{5.24}$$

Отже, матриця $B_n(\lambda)$ – результат точного множення матриці $P(\lambda)$ на матрицю $B(\lambda)$. Тобто побудуємо послідовність:

$$B_i(\lambda) = W_i(\lambda) B_{i-1}(\lambda) + \mu_{i-1} \;\; (i = 1,2,...,n), \tag{5.25}$$

$W_i(\lambda)$ – матриці, реально одержані в процесі обчислень, μ_i – матриця похибок від множення $P_i(\lambda)$ на $B_{i-1}(\lambda)$. Далі одержимо:

$$B_i(\lambda) = W_n(\lambda) B_{n-1}(\lambda) + \mu_{n-1} = W_n(\lambda) W_{n-1}(\lambda)...W_1(\lambda) \times$$
$$\times \left(B(\lambda) + W_1(\lambda)\mu_0 + W_1(\lambda) W_2(\lambda)\mu_1 + W_1(\lambda) W_2(\lambda)...W_m(\lambda)\mu_{n-1} \right).$$

Позначивши:

$$\left. \begin{array}{l} P(\lambda) = W_n(\lambda) W_{n-1}(\lambda)...W_1(\lambda) \\ Q_n = W_1(\lambda)\mu_0 + W_1(\lambda) W_2(\lambda)\mu_1 + W_1(\lambda) W_2(\lambda)...W_m(\lambda)\mu_{n-1} \end{array} \right\}, \tag{5.26}$$

отримаємо

$$B_n(\lambda) = P(\lambda)[B(\lambda) + Q_n]. \tag{5.27}$$

Аналізуючи (5.24) та (5.27), приходимо до висновку, що знайдену в процесі обчислень матрицю $B_n(\lambda)$ одержимо як результат точного множення збуреної матриці $B(\lambda) + Q_n$ на матрицю $P(\lambda)$.

ТВЕРДЖЕННЯ. *Припустимо, що деякий масив вхідних даних* $B\{P_1(\lambda), P_2(\lambda), ..., P_n(\lambda)\}$ *обробляється за алгоритмом* ρ*, який складається з етапів* $\rho_1, \rho_2, ..., \rho_k$ *так, що*

$$B_t(\lambda) = \rho_t\left(B(\lambda)\right). \tag{5.28}$$

Тоді процесу ρ_t *реалізації алгоритму* ρ *на ЕОМ відповідають еквівалентні збурення* ψ *масиву вхідних даних* $B(\lambda)$ *такі, що*

$$\psi = \prod_{i=1}^{k}\left(1 + \psi^{(i)}\right) - 1 \tag{5.29}$$

ДОВЕДЕННЯ. Опираючись на метод зворотного аналізу заокруглення похибок, машинній реалізації відношення (5.28) поставимо у відповідність точну реалізацію (5.29), але зі збуреними вхідними даними:

$$B^{(i)}\left(\lambda\right)=\rho_i\left(B^{(i-1)}\left(\lambda\right)\left[1+\psi^{(i)}\right]\right) \tag{5.30}$$

З урахуванням (5.30) для комп'ютерної реалізації першого етапу запишемо: $B^{(1)}\left(\lambda\right)=\rho_1\left(B^{(0)}\left(\lambda\right)\left[1+\psi^{(1)}\right]\right)$. Для другого етапу: $B^{(2)}\left(\lambda\right)=\rho_2\left(B^{(1)}\left(\lambda\right)\left[1+\psi^{(2)}\right]\right)$. Продовжуючи цей процес на k-ому етапі отримаємо:

$$B^{(k)}\left(\lambda\right)=\rho_k\left(B^{(k-1)}\left(\lambda\right)\left[1+\psi^{(k)}\right]\right)=\rho_k\left(\left[1+\psi^{(k)}\right]\right)\psi_{k-1}\left(B^{(k-2)}\left(\lambda\right)\left[1+\psi^{(k-1)}\right]\right)\right).$$

Якщо реалізація довільного обчислювального алгоритму ρ_t означає виконання хоча б однієї арифметичної операції, то для відповідного збурення ε мають місце співвідношення

$$\left(1+\gamma\right)\rho\left(X\left(\lambda\right)\right)=\rho_t\left(X\left(\lambda\right)\left[1+\varepsilon\right]\right), \tag{5.31}$$

де $\left|\gamma\right|>0$ – відносна похибка алгоритму.

Виходячи з (5.31), запишемо:

$$B^{(2)}\left(\lambda\right)=\rho_2(B^{(1)}\left(\lambda\right)[1+\psi^{(2)}])=\rho_2([1+\psi^{(2)}])\rho_1(B\left(\lambda\right)[1+\psi^{(1)}]))=$$
$$=\rho_2(\rho_1(B\left(\lambda\right)[1+\eta^{(2)}][1+\psi^{(1)}])),$$

$$B^{(3)}\left(\lambda\right)=\rho_3(B^{(1)}\left(\lambda\right)[1+\psi^{(3)}])=\rho_3([1+\psi^{(3)}])\rho_2\left(\rho_1\left(B\left(\lambda\right)[1+\eta^{(2)}][1+\psi^{(1)}]\right)\right)=$$

$$=\rho_3\rho_2\left([1+\psi^{(3)}]\right)(\rho_1(B\left(\lambda\right)[1+\eta^{(2)}][1+\psi^{(1)}]))=\rho_3\left(\rho_2\left(\rho_1\left(B\left(\lambda\right)[1+\eta^{(3)}][1+\eta^{(2)}][1+\psi^{(1)}]\right)\right)\right)$$

L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L

$$B^{(k)}\left(\lambda\right)=\rho_k\left(\rho_{k-1}\left(...\rho_1\left(B\left(\lambda\right)[1+\eta^{(k)}][1+\eta^{(k-1)}]...[1+\eta^{(2)}][1+\psi^{(1)}]\right)\right)...\right),$$

де $\left|\eta^{(i)}\right|>0$ – відносні похибки окремих етапів алгоритмів і $\left\|\eta^{(i)}\right\|\le\left\|\psi^{(i)}\right\|$, оскільки відносна похибка алгоритму не може бути менша від похибок вхідних даних. Тому, якщо

$$\rho\left(B\left(\lambda\right)\right)=\rho_k\left[\ \rho_{k-1}\left[...\rho_1\left[B\left(\lambda\right)\left(1+\eta^{(k)}\right)\left(1+\eta^{(k-1)}\right)...\left(1+\eta^{(2)}\right)\left(1+\psi^{(1)}\right)\right]\ \right]\ \right],$$

то $\rho_t(B(\lambda)(1+\psi)) = \rho_k(\rho_{k-1}(...\rho_1(\prod_{i=2}^{k}[1+\eta^{(i)}][1+\psi^{(1)}])).$

Звідси матимемо: $\psi = \prod_{i=2}^{k}[1+\eta^{(i)}][1+\psi^{(1)}]-1.$ З урахуванням того, що $\left\|\eta^{(i)}\right\| \le \left\|\psi^{(i)}\right\|$, з останньої рівності маємо:

$$\|\psi\|_j \le \prod_{i=2}^{k}\left(1+\left\|\eta^{(i)}\right\|_j\right)\left(1+\left\|\psi^{(1)}\right\|_j\right)-1 \le \prod_{i=2}^{k}\left(1+\left\|\psi^{(i)}\right\|_j\right)\left(1+\left\|\psi^{(1)}\right\|_j\right)-1 = \prod_{i=1}^{k}\left(1+\left\|\psi^{(i)}\right\|_j\right)-1 .$$

Що й треба було довести.

Наслідок. Якщо реалізація деякого обчислювального алгоритму ρ_t на ЕОМ означає виконання хоча б однієї арифметичної операції на кожному з k етапів так, що

$$\left.\begin{array}{l} (1+\gamma)\rho(x) = \rho_t\left(x[1+\varepsilon]\right) \\ \|\varepsilon\| \le \|\gamma\| \end{array}\right\}, \tag{5.32}$$

то для сумарної похибки заокруглення на ЕОМ γ і відповідного еквівалентого збурення ε справджуються нерівності $\|\beta\| \le \prod_{i=1}^{k}(1+\left\|\psi^{(i)}\right\|)-1$.

Проведемо тепер детальний аналіз обчислювальної стійкості раніше запропонованих алгоритмів засобами зворотного аналізу похибок. Для цього використаємо деякі результати, одержані Дж.Х. Уілкінсоном [97], В.В.Воєводіним [16] та іншими дослідниками стійкості обчислювальних методів. Сформулюємо їх без доведення.

Аналізуючи похибки заокруглення, часто доводиться використовувати оцінки вигляду:

$$\left(1-\alpha^{-t}\right)^k \le 1+\varepsilon \le \left(1+\alpha^{-t}\right)^k. \tag{5.33}$$

Для спрощення у всіх практичних застосуваннях припускається, що величина k обмежена зверху, а саме $k\varepsilon^{-t} < 0.1$.

Це обмеження для кожного прийнятного значення t спричинено, в першу чергу, обмеженістю оперативної пам'яті ЕОМ [12]. Враховуючи (5.33), можна записати:

$$\left.\begin{array}{l}(1+\alpha^{-t})^k < 1+1.05\cdot\alpha^{-t} \\ (1-\alpha^{-t})^k > 1-1.05\cdot\alpha^{-t}\end{array}\right\}.$$

Аналіз похибок схеми розрізання для розв'язання систем з $\lambda-$матрицями. Як і в попередніх випадках, реалізація методу має кілька етапів. Тому спочатку проведемо аналіз похибок для кожного етапу окремо, а потім виведемо остаточну оцінку.

ТЕОРЕМА 5.6.1. *Якщо при реалізації алгоритму схеми розрізання для СЛАР на ЕОМ у режимі плаваючої коми скалярні добутки обчислюються з одинарною точністю, то еквівалентні збурення задовольняють нерівності:*

$$\left\|\alpha_{i,j}^{(0)}\right\| \le C_0\cdot nl\cdot\max\left\{\|A\|,\left\|Y_2-C_{21}C_{11}^{-1}C_{12}\right\|,\left\|C_{22}-C_{21}C_{11}^{-1}Y_1\right\|,\left\|Y_1-C_{12}Z\right\|\right\}h^{-t+1}.$$

ДОВЕДЕННЯ. За алгоритмом розрізання на першому етапі обчислюється $C_{11}^{-1}C_{12}$ та $C_{11}^{-1}Y_1$. Матриця C_{11} - клітково-тепліцева, причому з блочно-трикутним заповненням. Обчислення $C_{11}^{-1}C_{12}$ та $C_{11}^{-1}Y_1$ можна виконати, розв'язуючи блочно-трикутну систему з $nl+l$ правими частинами. Отже, еквівалентні збурення a_{ij} елементів a_{ij} для обчислень з одинарною точністю задовольнятимуть нерівності $\left\|a_{i,j}^{(1)}\right\| \le 3.015\cdot nl\cdot\|A\|h^{-t+1}$.

На другому етапі обчислюються $C_{22}-C_{21}C_{11}^{-1}Y_1$ та $Y_2-C_{21}C_{11}^{-1}Y_1$. Якщо матриці множаться за класичним способом, то для обчислень з одинарною точністю одержимо такі еквівалентні збурення $a_{i,j}^{(*)}$, що

$$\left\|a_{i,j}^{(*)}\right\| \le 2.01nl\max\left\{\left\|D_{22}^{*}\right\|,\left\|B_{2}^{*}\right\|\right\}h^{-t+1}.$$

Третій крок – розв'язання системи лінійних алгебричних рівнянь порядку nl з щільним заповненням, а тому для обчислень з одинарною точністю:

$$\left\|a_{i,j}^{(3)}\right\| \le 3.015\cdot nl\cdot\left\|A^{(3)}\right\|h^{-t+1}.$$

Перш за все на четвертому етапі визначаються матриці $Y_1-C_{12}Z$. Комп'ютерне обчислення зумовлює такі ж похибки, як і на другому кроці. Потім необхідно ще розв'язати на комп'ютері систему з матрицею C_{11} порядку

$n^2 l$ із клітково-трикутним заповненням. Отже, виникнуть такі ж еквіваленті збурення, як і на першому кроці для обчислень з одинарною точністю:

$$\left\|a_{i,j}^{(4)}\right\| \le 3.015 \cdot n l \cdot \max\left\{\|A\|, \|Y_1 - C_{12} Z\|\right\} h^{-t+1}.$$

Визначимо тепер еквівалентні збурення, які відповідають реалізації на комп'ютері всього алгоритму. Якщо всі обчислення проводяться з одинарною точністю, то

$$\left\|a_{i,j}^{(0)}\right\| \le C_0 \cdot n l \cdot \max\left\{\|A\|, \|Y_2 - C_{21} C_{11}^{-1} C_{12}\|, \|C_{22} - C_{21} C_{11}^{-1} Y_1\|, \|Y_1 - C_{12} Z\|\right\} h^{-t+1},$$

що й треба було довести.

ТЕОРЕМА 5.6.2 *Якщо комп'ютерна реалізація алгоритму схеми розрізання для системи рівнянь у режимі плаваючої коми скалярні добутки обчислюються з подвійною точністю, то еквівалентні збурення задовольняють нерівності*

$$\left\|a_{i,j}^{(0)}\right\| \le C_0 \cdot \max\left\{\|A\|, \|Y_2 - C_{21} C_{11}^{-1} C_{12}\|, \|C_{22} - C_{21} C_{11}^{-1} Y_1\|, \|Y_1 - C_{12} Z\|\right\} h^{-t+1}.$$

ДОВЕДЕННЯ. Матриця C_{11} - блочно-трикутна квадратна матриця клітково-тепліцевого характеру. Добутки $C_{11}^{-1} C_{12}$ та $C_{11}^{-1} Y_1$ можна обчислити, розв'язавши блочно-трикутну систему з $2ln + 1$ правими частинами. Для обчислень з подвійною точністю еквівалентні збурення $a_{i,j}^{(1)}$ елементів $a_{i,j}$, що відповідають першому етапу алгоритму, задовольняють нерівность

$$\left\|a_{i,j}^{(1)}\right\| \le 3.015 \|A\| h^{-t+1}.$$

Множення матриць на другому етапі зумовлює еквівалентні збурення $a_{i,j}^{(2)}$

$$\left\|a_{i,j}^{(2)}\right\| \le 2.01 \max\left\{\|C_{22} - C_{21} C_{11}^{-1} C_{12}\|, \|Y_2 - C_{21} C_{11}^{-1} Y_1\|\right\} h^{-t+1}.$$

На третьому етапі, коли шукають розв'язок системи лінійних алгебричних рівнянь порядку $2ln$ зі щільним заповненням, виникають еквівалентні збурення

$$\left\|a_{i,j}^{(3)}\right\| \le 3.015 \|A^*\| h^{-t+1}.$$

Визначаючи матрицю $Y_1 - C_{12} Z$, одержимо похибки такі самі, як і на другому етапі $\left\|a_{i,j}^{(4)}\right\| \le 2.01 \max\left\{\|C_{22} - C_{21} C_{11}^{-1} C_{12}\|, \|Y_2 - C_{21} C_{11}^{-1} Y_1\|\right\} h^{-t+1}.$

Обчислення розв'язку системи з матрицею C_{11} порядку $(2nl+1)n$ із клітково-трикутним заповненням зумовлює такі ж еквівалентні збурення, як і на першому кроці $\left\|a_{i,j}^{(5)}\right\| \le 3.015\left\{\|A\|,\|Y_1 - C_{12}Z\|\right\}h^{-t+1}$.

Повній комп'ютерній реалізації алгоритму для обчислень з подвійною точністю відповідають збурення

$$\left\|a_{i,j}^{(0)}\right\| \le C_0 \cdot \max\left\{\|A\|,\|Y_2 - C_{21}C_{11}^{-1}C_{12}\|,\|C_{22} - C_{21}C_{11}^{-1}Y_1\|,\|Y_1 - C_{12}Z\|\right\}h^{-t+1},$$

що й треба було довести.

Отже, за кількістю арифметичних операцій і за малістю еквівалентних збурень алгоритм схеми розрізання придатний для використання на ЕОМ. При співрозмірних еквівалентних збуреннях він має вищу швидкість, ніж запропоновані прямі алгоритми.

Висновки до п'ятого розділу

В даному розділі розглядаються застосування методів комп'ютерної алгебри для математичного моделювання динамічної моделі Леонтьєва.

Запропоновано новий метод розв'язування розріджених систем лінійних алгебричних рівнянь з блочними елементами. Проведено підрахунок кількостей операцій числової реалізації алгоритму множення матриць. Проведено порівняння ефективності запропонованого алгоритму та блочного методу прогонки з погляду комп'ютерної алгебри.

Розглянуто кліткові обчислювальні методи розв'язування систем лінійних алгебричних рівнянь.

Описано ефективний новий підхід до розв'язання систем лінійних алгебраїчних рівнянь у моделі Леонтьєва з прямокутними $\lambda-$матрицями. Створено оптимізаційну модель з матрицями міжгалузевого балансу та запропоновано ефективний обчислювальний метод реалізації цієї моделі. Проаналізовано та розширено деякі підходи до обчислювальних алгоритмів.

Запропоновано новий підхід до розв’язання погано обумовлених систем лінійних алгебричних рівнянь у моделі Леонтьєва. Підраховано арифметичні операції СЛАР при чисельній реалізації алгоритму на ЕОМ. Наведено спосіб обчислення числа обумовленості матриці.

Проведено дослідження паралельності описаних алгоритмів розв'язання матричних систем з $\lambda -$матрицями для двох найбільш часто використовуваних моделей – концепції необмеженого паралелізму [12], [103], а також для комп'ютерів з конкретною архітектурою – багатопроцесорною обчислювальною системою типу MIMD. Встановлено хороші можливості для застосування алгоритмів розв'язання матричних систем з $\lambda -$матрицями на паралельних обчислювальних системах.

Проаналізовано обчислювальну стійкість запропонованого алгоритму розв'язання систем лінійних алгебричних рівнянь у моделі Леонтьєва. Охарактеризовано складність алгоритму та показано його ефективність з точки зору комп'ютерної алгебри.

Отримані результати опубліковано у працях [81; 82; 83] та доповідались на конференції [85].

ВИСНОВКИ

Одержані в роботі результати є важливими для теорії СЛАР з матрицями для моделей Леонтьєва. Вони, також, можуть бути застосовані в теорії результатів, при розрахунку і оптимізації економіко-математичних моделей, в задачах лінійного і параметричного програмування. Подібні задачі зустрічаються в задачах хімічної кінетики, при розв'язанні задач синтезу великих електронних схем, в задачах будівельної механіки, в динамічному програмуванні.

1. Розроблено стійкі методи розв'язування розріджених СЛАР із блочними елементами. Проведено підрахунок кількості записів при символьному розв'язанні та арифметичних операцій при чисельній реалізації алгоритму множення матриць. Охарактеризовано складність алгоритму з точки зору комп'ютерної алгебри. Проведено порівняння пропонованого алгоритму та існуючих методів.
2. Створено алгоритми розв'язування СЛАР з багатовимірними матрицями для динамічної моделі В. Леонтьєва. Розвинуто скінченно-різницевий підхід до розв'язування систем лінійних алгебричних рівнянь із двовимірними матрицями. Зведено системи лінійних алгебраїчних рівнянь з мірними матрицями до системи з числовими елементами та підраховано кількість арифметичних операцій.
3. Розроблено новий підхід розв'язування кліткових алгоритмів для систем лінійних алгебраїчних рівнянь з блочними елементами в моделях В.Леонтьєва. Описано алгоритм для трьохдіагональної блочної системи лінійних алгебричних рівнянь. Розглянуто блочний варіант другого алгоритму відсічних систем, а також проведено підрахунок кількості операцій, потрібних для їх реалізації.
4. Визначено умови розв'язності прямокутних та розріджених систем лінійних рівнянь з матрицями в моделях Леонтьєва. Розглянуто новий підхід до розв'язання СЛАР Леонтьєва із прямокутними матрицями.

Створено оптимізаційну модель з матрицями міжгалузевого балансу та запропоновано ефективний обчислювальний метод реалізації цієї моделі.

5. Розглянуто новий підхід до розв'язування погано обумовлених СЛАР у моделі Леонтьєва. Виконано аналіз складності числової реалізації алгоритму розв'язування СЛАР на ЕОМ. Проаналізовано обчислювальну стійкість запропонованого алгоритму розв'язання СЛАР у моделі Леонтьєва. Охарактеризовано складність алгоритму та показано його ефективність з точки зору комп'ютерної алгебри.
6. Теоретичне обґрунтування розроблених методів підтверджується числовими та експериментальними прикладами.

Проведено тестування:

- ✓ алгоритмів розв'язання щільно заповнених числових систем лінійних алгебричних рівнянь середньої розмірності;
- ✓ кліткових алгоритмів розв'язання числових систем лінійних алгебричних рівнянь;
- ✓ алгоритмів розв'язання деяких типів розріджених числових систем лінійних алгебричних рівнянь;
- ✓ алгоритмів розв'язання систем лінійних алгебричних рівнянь з поліноміальними елементами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ахо А. Построение и анализ вычислительных алгоритмов / Ахо А., Хопкрофт Дж., Ульман Дж. – М.: Мир, 1979. – 320 с.
2. Ашманов С.А. Введение в математическую экономику / С.А. Ашманов. – М.: Наука, 1984. – 296 с.
3. Ашманов С.А. Линейное программирование / С.А. Ашманов. – М.: Наука, 1981. – 340 с.
4. Ашманов С.А. Математические модели и методы в экономике / С.А. Ашманов. – М.: Изд-во Моск. ун-та, 1980. – 199 с.
5. Бейкер Дж. Апроксимации Паде / Дж. Бейкер, П. Грейвс-Моррис. – М.: Мир, 1996. – 502 с.
6. Беллман Р. Введение в теорию матриц / Р. Беллман. – М.: Наука, 1969. – 368 с.
7. Боднар Д.И. Ветвящиеся цепные дроби / Д.И. Боднар. – К.: Наук. думка, 1986. –176 с.
8. Босикова И.И. Решение систем линейных алгебраических уравнений с двумерными $\lambda-$матрицами / И.И. Босикова // Кибернетика и системный анализ. – 2001. – № 4. – С. 175-179.
9. Босикова И.И. Метод решения систем линейных алгебраических уравнений с m-мерными $\lambda-$матрицами / И.И. Босикова // Кибернетика и системный анализ. – 2002. – № 1. – С. 37-47.
10. Босікова І.І. Прямі методи розв'язування систем лінійних алгебраїчних рівнянь з $\lambda-$матрицями: автореф. дис. на здобуття наук. ступеня канд. фіз.-мат. наук : спец. 01.01.07 "Обчислювальна математика" / І.І. Босікова. – Львів, 2002. – 16 с.
11. Боюн В.П. Скінченно-різницеві підходи до оптимізації обчислень матриць із складними елементами і множення їх на довільній вектор /

В.П. Боюн // Зб. наук. праць. – К.: Ін-т кібернетики ім. В.М.Глушкова, 1999. – С.61-64.

12. Валях В.Е. Последовательно-параллельные вычисления / В.Е. Валях. – М.: Мир, 1985. – 456 с.
13. Воеводин В.В. Вычислительные основы линейной алгебры / В.В. Воеводин. – М.: Наука, 1977. – 303 с.
14. Воеводин В.В. Вычислительные процесы с теплицевыми матрицами / В.В. Воеводин, Е.Е. Тыртышников. – М.: Наука, 1987. – 319 с.
15. Воеводин В.В. Линейная алгебра / В.В. Воеводин. – С.-П.: Лань, 2008. – 416 с.
16. Воеводин В.В. Математические модели и методы в параллельных процессах / В.В. Воеводин. – М.: Наука, 1986. – 296 с.
17. Воеводин В.В. Матрицы и вычисления / В.В. Воеводин, Ю.А. Кузнецов. – М.: Наука, 1984. – 320 с.
18. Воеводин В.В. Ошибки округления и устойчивость в прямых методах линейной алгебры / В.В. Воеводин. – М.: Изд-во МГУ, 1969. – 187 с.
19. Воеводин В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин. – С.-П.: БХВ–Петербург, 2002. – 608 с.
20. Воеводин В.В. Численные методы алгебры. Теория и алгоритмы / В.В. Воеводин. – М.: Наука, 1976. – 312 с.
21. Волошин О.Ф. Алгоритм послідовного аналізу варіантів для розв'язування міжгалузевої моделі Леонтьєва–Форда / О.Ф. Волошин, Н.Б. Чорней // Вісник Київ. ун-ту. Сер. фіз.-мат. наук. – 1999. – №1. – С. 171-175.
22. Гантмахер Ф.Р. Теория матриц / Ф.Р. Гантмахер. – Москва: Наука, 1967. – 324 с.
23. Гельфонд А.О. Исчисление конечных разностей / А.О. Гельфонд. – М.: Наука, 1967. – 375 с.
24. Гольштейн Е.Г. Методы оптимизации в экономико-математическом моделировании / Е.Г. Гольштейн. – М.: Наука, 1991. – 448 с.

25. Горбачук О.Л. Матриці та системи лінійних рівнянь / Горбачук О.Л., Комарницька Л.І., Матурін Ю.П.. – Дрогобич: Редакційно-видавничий відділ ДДПУ, 2007. – 50 с.

26. Григорків В.С. Моделювання економіки. Частина 2: навч. посібник / В.С. Григорків. – Чернівці: Рута, 2006. – 100 с.

27. Дейнека В.С. Модели и методы решения задач в неоднородных средах / В.С. Дейнека, И.В. Сергиенко. – К.: Наук. думка, 2001. – 606 с.

28. Дмитришин Р.В. Оптимизация электронных схем на ЭВМ / Р.В. Дмитришин. – К.: Техника, 1987. – 224 с.

29. Дэвэнпорт Д. Компьютерная алгебра / Дэвэнпорт Д., Сирэ И., Турнье Э.. – М.: Мир, 1991. – 352 с.

30. Иванилов Ю.П. Математические модели в экономике / Ю.П. Иванилов, А.В. Лотов. – М.: Физматлит, 1979. – 304 с.

31. Икрамов Х.Д. К вопросу об анализе ошибок округления для неортогональных методов / Х.Д. Икрамов // Журнал вычислительной математики и математической физики. – 1982. – Т. 20, №2. – С. 259-268.

32. Икрамов Х.Д. Численное решение матричных уравнений / Х.Д. Икрамов. – М.: Наука, 1984. – 192 с.

33. Интриллигатор М. Математические методы оптимизации и экономическая теория / М. Интриллигатор. – М.: Прогресс, 1975. – 607 с.

34. Жуков С.А. Математичні методи та моделі в економіці: навч. посібник / Жуков С.А., Остапчук В.С., Сторубльов О.І.. – К.: Видавничо-поліграфічний центр “Київський університет”, 2002. – 231 с.

35. Заборовець М.О. Сучасні методи розв'язування систем лінійних алгебраїчних рівнянь / Заборовець М.О., Левченко Ф.А., Охріменко М.Г.. – К.: КНЕУ, 2005. – 76 с.

36. Здрок В.В. Моделювання економічної динаміки / В.В. Здрок, І.М. Паславська. – Л.: ЛНУ ім. І. Франка, 2007. – 243 с.

37. Кабак А.Ф. Економіко-математичні методи і моделі: навч. посібник / А.Ф. Кабак. – К.: ІЗМН, 1996. – 164 с.

38. Карлин С. Математические методы в теории игр, программировании и экономике / С. Карлин. – М.: Мир, 1964. – 836 с.

39. Канатников А.Н. Линейная алгебра / А.Н. Канатников, А.П. Крищенко. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 336 с.

40. Киселев Б.Н. Условия существования решения в упрощенных аналогах модели Леонтьева / Б.Н. Киселев // Анализ и применение математических моделей экономической динамики. – Новосибирск: Наука, 1990. – С. 116-120.

41. Ковальчук О.Я. Алгоритми для систем з тепліцевими λ – матрицями та їх застосування: автореф. дис. на здобуття наук. ступеня канд. фіз.-мат. наук : спец. 01.05.02 "Математичне моделювання та обчислювальні методи" / О.Я. Ковальчук. – Інститут кібернетики імені В.М.Глушкова, Київ, 2005. – 20 с.

42. Колемаев В.А. Математическая экономика / В.А. Колемаев. – М.: ЮНИТИ, 1998. – 240 с.

43. Колоколов А.А. Методы дискретной оптимизации: учеб. пособие / А.А. Колоколов. – Омск, ОмГУ, 1984. – 76 с.

44. Кострикин А.И. Линейная алгебра и геометрия / А.И. Кострикин, Ю.И. Манин. – М.: МГУ, 1980. – 320 с.

45. Кублановская В.Н. Численные методы и вопросы организации вычислений / В.Н. Кублановская, В.П. Ильин. – Ленинград: Наука, 1987. – 180 с.

46. Кудін В.І. Аналіз моделі Леонтьєва при нечітко заданій множині обмежень на змінні / В.І. Кудін // Вісник Київського університету. Серія: фізико-математичні науки. – К. – 2006. Випуск 1. – С. 161-165.

47. Кудін В.І. Про загальні розв'язки систем лінійних алгебраїчних рівнянь / В.І. Кудін // Вісник Київського університету. Серія: фізико-математичні науки. – К. – 2004. Випуск 3. – С. 228-232.

48. Ланкастер П. Теория матриц / П. Ланкастер; пер. с англ. С.П. Демушкина. – Москва: Главная редакция физико-математической литературы, 1982. – 272 с.

49. Лотов А.В. Введение в экономико-математическое моделирование / А.В. Лотов. – М.: Наука, 1984. – 392 с.

50. Лэсдон Л.С. Оптимизация больших систем / Л.С. Лэсдон. – М.: Наука, 1975. – 432 с.

51. Ляшенко І.М. Економіко-математичні методи та моделі сталого розвитку / І.М. Ляшенко. – К.: Вища школа, 1999. – 236 с.

52. Ляшенко І.М. Основи математичного моделювання економічних, екологічних та соціальних процесів / Ляшенко І.М., Коробова М.В., Столяр А.М.. – Тернопіль: Навчальна книга “Богдан”, 2007. – 304 с.

53. Макаров В.Л. Математическая теория экономической динамики и равновесия / В.Л. Макаров, А.М. Рубинов. – М.: Наука, 1973. – 336 с.

54. Малашонок Г.И. О параллельных матричных алгоритмах в компьютерной алгебре / Г.И. Малашонок, Ю.Д. Валеев, М.С. Зуев // Вестн. Тамб. ун-та. Сер.: Естеств. и техн. науки. Тамбов – 2005. – Т.10. вып. 1. – С. 102-104

55. Мальцев А.И. Основы линейной алгебры / А.И. Мальцев. – М.: Наука, 1970. – 400 с.

56. Марчук Г.И. Методы вычислительной математики / Г.И. Марчук. – М.: Наука, 1980. – 536 с.

57. Михлин С.Г. Некоторые вопросы теории погрешностей / С.Г. Михлин. – Л.: Изд-во Ленинградского университета, 1988. – 333 с.

58. Молчанов И.Н. Машинные методы решения прикладных задач. Алгебра. Приближение функций / И.Н. Молчанов. – К.: Наукова думка, 1987. – 288 с.

59. Натансон И.П. Краткий курс высшей математики / И.П. Натансон. – М.: Наука, 1968. – 680 с.

60. Недашковський М.О. Ознаки збіжності матричних гіллястих ланцюгових дробів / М.О. Недашковський // Математичні методи та фізико-механічні поля. – Львів, 2003. Т.46, №4. – С. 50-56.

61. Недашковський М.О. Методи та алгоритми комп'ютерної алгебри для систем лінійних алгебраїчних рівнянь з поліноміальними елементами: автореф. дис. на здобуття наук. ступеня докт. фіз.-мат. наук : спец. 01.05.02 "Математичне моделювання та обчислювальні методи" / М.О. Недашковський. – Харків, 1995. – 43с.

62. Недашковський М.О. Обчислення з λ – матрицями / М.О. Недашковський, О.Я. Ковальчук. – К.: Наук. думка, 2007. – 294 с.

63. Недашковський М.О. Розв'язування систем лінійних алгебраїчних рівнянь методом гіллястих ланцюгових дробів / М.О. Недашковський, В.Я. Скоробагатько // В кн. Теоретичні та прикладні питання алгебри і рівнянь – К: Наук. думка, 1977. – С. 84-92.

64. Недашковський М.О. Узагальнені динамічні міжгалузеві моделі / М.О. Недашковський, Л.М. Семчишин // Вісник Тернопільського національного економічного університету. Випуск 1. – Тернопіль: Економічна думка, 2009. – С. 169-187.

65. Недашковський М.О. Швидка схема розв'язання для систем лінійних алгебраїчних рівнянь з λ – матрицями / М.О. Недашковський // Доп. НАН України. Серія А. –1995. – № 4. – С. 23 - 29.

66. Недашковский Н.А. Вычислительные алгоритмы для линейных балансовых моделей межотраслевого эколого-экономического взаимодействия. / Н.А. Недашковский, Т.И. Крошка // Кибернетика и системный анализ. К. – 2010. – №1. – С. 17-28.

67. Недашковский Н.А. О решении систем алгебраических уравнений с λ – матрицами / Н.А. Недашковский // Журнал вычислительной математики и математической физики. – М.: Вычислительный центр им. А.А. Дородницына РАН , 1988. – Т. 28, №3. – С. 439-443.

68. Недашковский Н.А. О решении систем алгебраических уравнений с полиномиально-численным заполнением / Н.А. Недашковский // Кибернетика и системный анализ. – К., 1996. – №6. – С. 173-183.
69. Недашковский Н.А. Прямые методы решения систем линейных алгебраических уравнений с комплексными $\lambda -$матрицами / Н.А. Недашковский, И.И. Босикова // Кибернетика и системный анализ. – К., 2000. – № 2. – С. 144-156.
70. Недашковский Н.А. Решение матричных уравнений ветвящимися цепными дробями / Н.А. Недашковский, О.Я. Ковальчук // Кибернетика и системный анализ. – К., 2004. – №1. – С. 22-33.
71. Недашковский Н.А. Решение систем алгебраических уравнений с полиномиальными матрицами / Н.А. Недашковский // Доклады АН УССР. Серия А. – 1984. –№ 10. – С. 68-73.
72. Овчинников П.П. Вища математика / П.П. Овчинников, Ф.П. Яремчук, В.М. Михайленко. – К.: Техніка, 2003. – 600 с.
73. Панкратьев Е.В. Компьютерная алгебра. Факторизация многочленов / Е.В. Панкратьев. – М.: Изд-во МГУ, 1998. – 164 с.
74. Писанецки С. Технология разреженных матриц / С. Писанецки. – М.: Мир, 1988. – 410 с.
75. Пономаренко О.І. Основи математичної економіки / О.І. Пономаренко, М.О. Перестюк, В.М. Бурим. – К.: Інформтехніка, 1995. – 320 с.
76. Семчишин Л.М. Динамічні математичні моделі в економіці / Л.М. Семчишин // Вісник Тернопільського національного економічного університету. Випуск 3. – Тернопіль: Економічна думка, 2008. – С. 123-129.
77. Семчишин Л.М. До розв’язання розріджених систем лінійних алгебраїчних рівнянь / Л.М. Семчишин // Матеріали XII міжнародної наукової конференції імені академіка М. Кравчука, 15-17 травня 2008 року, Київ: ТОВ "Задруга", 2008. – С. 786

78. Семчишин Л.М. До розв'язування систем лінійних алгебраїчних рівнянь з $m-$мірними $\lambda-$матрицями / Л.М. Семчишин // м. Київ, Інститут математики НАН України, 27-29 серпня 2009 р., // www.imath.kiev.ua/~congress2009/

79. Семчишин Л.М. Економіко-математичні моделі в багаторівневому управлінні / Л.М. Семчишин // Матеріали II міжнародної наукової конференції імені Я. С. Підстригача, 25-29 травня 2008 року, Львів – 2008. – Т.3. – С. 46-48

80. Семчишин Л.М. Кліткові алгоритми для систем лінійних алгебраїчних рівнянь з блочними елементами в моделях В. Леонтьєва / Л.М. Семчишин // Комп'ютерна математика. Випуск 2. – Київ: Інститут кібернетики ім. В.М. Глушкова Національної академії наук України, 2009. – С. 24-35.

81. Семчишин Л.М. Розв’язання розріджених систем лінійних алгебраїчних рівнянь з блочними елементами / Л.М. Семчишин // Фізико-математичне моделювання та інформаційні технології. Випуск 6. – Львів: Центр математичного моделювання інституту прикладних проблем механіки і математики ім. Я. С. Підстригача НАН України, 2007. – С. 128-135.

82. Семчишин Л.М. Розв’язування погано обумовлених систем лінійних алгебраїчних рівнянь у моделі Леонтьєва / Л.М. Семчишин // Вісник Тернопільського державного технічного університету. Тернопіль – 2009. Том 14. №4 – С. 168-175

83. Семчишин Л.М. Розв’язування прямокутних та розріджених систем лінійних рівнянь з $\lambda-$матрицями в моделі Леонтьєва / Л.М. Семчишин // Математичне та комп’ютерне моделювання. Серія: фізико-математичні науки: збірник наукових праць / Кам’янець-Подільський національний університет, Інститут кібернетики імені В.М. Глушкова Національної академії наук України. – Випуск 1. Кам’янець-Подільський – 2009. – С. 121-133

84. Семчишин Л.М. Розв’язування систем лінійних алгебраїчних рівнянь із багатовимірними $\lambda-$матрицями для динамічної моделі Леонтьєва / Л.М.

Семчишин // Фізико-математичне моделювання та інформаційні технології. Випуск 10. – Львів: Центр математичного моделювання інституту прикладних проблем механіки і математики ім. Я.С. Підстригача НАН України, 2009. – С. 123-131

85. Семчишин Л.М. Розв'язування систем лінійних алгебраїчних рівнянь із двовимірними λ–матрицями для динамічної моделі Леонтьєва / Л.М. Семчишин // Матеріали XIII міжнародної наукової конференції імені академіка М. Кравчука, 13-15 травня 2010 року, Київ, НТУУ, 2010. – Т.2. – С. 243

86. Семчишин Л.М. Узагальнена динамічна модель Леонтьєва / Л.М. Семчишин // Матеріали І Міжнародної науково-методичної конференції 1 – 4 квітня 2009 року, Чернівці: ДрукАрт, 2009. – С. 360-362

87. Сопронюк Ф.О. Спостережуваність сингулярно збурених систем зі зміною вимірності фазового простору / Ф.О. Сопронюк, Є.Ф. Сопронюк // Науковий вісник Чернівецького університету. Серія: Комп'ютерні системи і компоненти – 2010. – Том 1, випуск 1. – С. 58-61

88. Сергиенко И.В. Исследование устойчивости и параметрический анализ дискретных оптимизационных задач / И.В. Сергиенко, Л.Н. Козерацкая, Т.Т. Лебедева. – К.: Наук. думка, 1995. – 453 с.

89. Сергиенко И.В. Математические модели и метод решения задачи дискретной оптимизации / И.В. Сергиенко. – Киев: Наук. думка, 1988. – 472 с.

90. Скоробогатько В.Я. Теория ветвящихся цепных дробей и её применение в вычислительной математике / В.Я. Скоробогатько – М.: Наука. Главная редакция физико-математической литературы, 1983. – 312 с.

91. Скорняков Л.А. Элементы алгебры / Л.А. Скорняков. – М.: Наука, 1980. – 240 с.

92. Тер-Крикоров А.М. Оптимальное управление и математическая экономика / А.М. Тер-Крикоров. – М.: Наука, 1977. – 216 с.

93. Тинберхен Я. Математические модели экономического роста / Я. Тинберхен, Х. Бос; пер. с англ. М.М. Голанского. – М.: Прогресс, 1967. – 176 с.

94. Тихонов А.Н. Методы решения некорректных задач / А. Н. Тихонов, В.Я. Арсенин. – М.: Наука, 1974. – 224 с.

95. Тыртышников Е.Е. Матричный анализ и линейная алгебра / Е.Е. Тыртышников. – М.: Физматлит, 2007. – 480 с.

96. Тьюарсон Р. Разреженные матрицы / Р. Тьюарсон; пер. с англ. Э.М. Пейсаховича, под ред. Х.Д. Икрамова. – М.: Мир, 1977. – 172 с.

97. Уилкинсон Дж.Х. Алгебраическая проблема собственных значений / Дж.Х. Уилкинсон. – М.: Наука, 1970. – 564 с.

98. Уилкинсон Дж.Х. Справочник алгоритмов на языке Ангол. Линейная алгебра / Дж.Х. Уилкинсон, К. Райнш. – М.: Машиностроение, 1976. – 389 с.

99. Фаддеев Д.К. Вычислительные основы линейной алгебры / Д.К. Фаддеев, В.Н. Фаддеева. – М.: Физматгиз, 1963. – 734 с.

100. Фаддеев Д.К., Фаддеева В.Н. Вычислительные методы линейной алгебры. – М.: Физматгиз, 1960. – 330с.

101. Форсайт Дж. Машинные методы вычислений / Дж. Форсайт, М. Малькольм, К. Моулер. – М.: Мир, 1980. – 330с.

102. Химич А.Н. Оценки полной погрешности решения систем линейных алгебраических уравнений для матриц произвольного ранга / А.Н. Химич. – Киев, Компьютерная математика. – 2002. – №2. – с. 41-49.

103. Химич А.Н. Параллельные алгоритмы решения задач вычислительной математики / [Химич А.Н., Молчанов И.Н. и др.]. – К.: Наукова думка, 2008. – 248 с.

104. Хорн Р. Матричный анализ: пер. с англ. / Р. Хорн, Ч. Джонсон. – М.: Мир, 1989. – 655с.

105. Цегелик Г.Г. Чисельні методи / Г.Г. Цегелик. – Л.: Видавничий центр ЛНУ імені Івана Франка, 2004. – 408 с.

106. Чорней Н.Б. Дослідження алгоритму послідовного аналізу варіантів для розв'язання міжгалузевої моделі Леонтьєва — Форда / Н.Б. Чорней // Вісник Київського університету. – 1999. – Сер. фіз.-мат. н., № 3. – С. 259–262.

107. Шахно С.М. Чисельні методи лінійної алгебри / С.М. Шахно. – Л.: Видавничий центр ЛНУ імені І. Франка, 2007. – 245 с.

108. Шор Н.З. Квадратичные экстремальные задачи и дифференцируемая оптимизация / Н.З. Шор, С.И. Стеценко. – К.: Наук. думка, 1989. – 208 с.

109. Эстербю О. Прямые методы для разреженных матриц / О. Эстербю, З.Златев. – М.: Мир, 1987. – 120 с.

110. Acevedo J. A hybrid parametric and stochastic programming approach for mixed-integer linear problems under uncertainty / J. Acevedo, N. Pistikopoulos // Industrial Engineering Chemical Research. – 1997. – Vol. 36. – P. 2262-2270.

111. Adler I. A geometric view of parametric linear programming / I. Adler, R. Monteiro. – Algorithmica. – 1992. – № 8. – P. 161-176.

112. Bareiss E.H. Sylvester's identity and multistep integreserving Gaussian elimination / E.H. Bareiss // Math. Comp. – 1968. – Vol. 22. – P. 565-578.

113. Borodin A. The computational complexiti of algebraic and numeric problems / A. Borodin, I. Munro. – New York: American Elsevier, 1975.

114. Cabay S. Systems of Linear Equations with Dense Univariate Polynomial Coefficients / S. Cabay, B. Domzy // Journal of the Association for Computing Machinery. – 1987. – Vol. 34, № 3. – P.646-660.

115. Cambini A. Linear fractional and bicriteria linear fractional programs / A. Cambini, L. Martein // Third International Conference on Generalized Convexity "International Workshop on Generalized Concavity, Fractional Programming and Economic Applications". – Pisa. – 1988. – P. 155-166.

116. Gal T. Survey of Solved and Open Problems in the Degeneracy Phenomenon / T. Gal, H. Kruse, P. Zörnig // Math. Program. – 1988. – 42, No 1. – P. 125-133

117. Chanas S. An algorithm for solving bicriterial linear programming problems with parametrical coefficients in the objective functions / S. Chanas, D. Kuchta // Annals of Operations Research. – 1998. – Vol. 81. – P. 63-72.

118. Chanas S. FPLP – a package for fuzzy and parametric linear programming problems / S. Chanas, D. Kuchta, Z. Nowak // Interactive fuzzy optimization. – Berlin.: Springer– Verlag, 1991. – P. 188-198.

119. Chvata V. Linear programming / V. Chvata. – New York, 1983.

120. Coppersmith D. In application of factoring / D. Coppersmith, J. Davenport // J. Symbolic Comp. – 1985. – № 1. – P. 241-243.

121. Dax Achiya The Convergence of Linear Stationary Iterative Processes for Solving Singular Unstructured Systems of Linear Equations / Dax Achiya // SIAM Rev. – 1990. – 32, No 4. – P. 611-635.

122. Golub G. Parallel block schemes for large scale least squares computations / G. Golub, R. Plemmons, A. Sameh // Center for supercomputing research and develepment report № 574, University of Illinois. – 1986.

123. Computational methods in linear algebra / [Goult R.J., Hoskins R.F., Milner J.A., Pratt M.J.] – London: Stanley Thornes, 1974.

124. Goult R.J., Hoskins R.F., Milner J.A., Pratt M.J. Computational methods in linear algebra. Goult R.J., Hoskins R.F., Milner J.A., Pratt M.J. – London.: Stanley Thornes, 1974.

125. Jenkins L. Parametric methods in integer linear programming / L. Jenkins // Annals of Operations Research. – 1990. – Vol. 27. – P. 77-96.

126. John F. Lectures on advanced numerical analysis / F. John. – New York: Gordon and Brench, 1966.

127. Kellison S.G. Fundamentals of numerical analysis / S.G. Kellison. – Irwin: Homewood, III., 1975.

128. Lipson J.D. Symbolic metods for the computer solution of linear equations with aplications to flowgraphs / J.D. Lipson // Proccedings of the 1968 Summer Institute Symbolic Mathematical Computations (Boston). – New York: IBM, 1969. – P. 233-303.

129. Loukakis E. Parameterisation algorithms for the integer linear programs in binary variables / E. Loukakis, A. Muhlemann // European Journal of Operational Research. – 1984. – Vol. 17. – P. 104-115.

130. Macclelian M.T. The exact solution of systems of linear equations with polynomial coefficients / M.T. Macclelian // ACM. – 1973. – Vol. 0, № 4. – P.563-588.

131. Macclelian M.T. A comparison of algorithms for the exact solution of linear equations / M.T. Macclelian // ACM Trans Math Softw. –1977. – Vol. 3, № 2. – P. 1-25.

132. Magnanti T. Parametric linear programming and anti-cycling pivoting rules / T. Magnanti, J. Orlin // Mathematical Programming – 1988. – Vol. 41. – P. 317-325.

133. Moenk R.T. Appoximate algoritms to derive exact solutions to sustems of linear equations / R.T. Moenk, J.H. Carter // Symbolic and Algebraic Computation TURSAM (Marseille, France, 6, June). – New York: Springer-Verlag. – 1979. – P. 65-73.

134. Murty K. Computational complexity of parametric linear programming / K.Murty // Mathematical Programming. – 1980. – Vol. 19. – P. 213-219.

135. Papadimitriou I. Decomposition d'une matrice de Leontief par l'analyse des correspondances / I. Papadimitriou // Cah. Anal. Damies, 1987. – 12, No 2. – P. 147–168.

136. Rountree S. Parametric integer linear programming: A synthesis of branch and bound with cutting planes / S. Rountree, B. Gillett // European Journal of Operational Research. – 1982. – Vol. 10 – P. 183-189.

137. Savage J.E. The complexity of computing / J.E. Savage. –New York: Wiley, 1976.

138. Sazaki T. Efficient Gaussian elimination method for sumbolic determinants and linear systems / T. Sazaki, H. Murao // ACM Transactions on Mathematical Software. – 1982. – № 8. – P. 227-289.

139. Smit J. Consellation free algorithm, with factoring capabilities, for the efficient solution of large sparse sets of equations / J. Smit // Proceedinds of the 1981 ASM Symposium on symbolic Algebraic Computation, ACM Inc. – New York. – 1981. – P. 146-154.

140. Sodini C. Equivalence and parametric analysis in linear fractional programming / C. Sodini // Third International Conference on Generalized Convexity "International Workshop on Generalized Concavity, Fractional Programming and Economic Applications". – Pisa. – 1988. – P. 143-154.

141. Van Tu T. Optimization over the efficient set of a parametric multiple objective linear programming / T. Van Tu // European Journal of Operational Research. – 2000. – Vol. 122. – P. 570-583.

142. Van Dooren P.M. Numerical Analysis 2000: Linear Algebra - Linear Systems and Eigenvalues / P.M. Van Dooren. – Vol. 3, 2001. – P. 526.

143. Westlake J.R. Handbook of numerical matrix inversion and solution of linear equations / J.R. Westlake. – New York:Wiley, 1968.

144. Winograd S. Arithmetic complexity of computations / S. Winograd. – Philadelphia, Pennsylvania.: Society for Industrial and Applied Mathematics, 1980.

145. Young D.M. Iterative solution of large linear systems / D.M. Young. – New York: Academic Press, 1971.

146. Zhang X. A note on the continuity of solutions of parametric linear programs / X. Zhang, D. Liu // Mathematical Programming. – 1990. – Vol. – P. 143-153.

ДОДАТКИ

Додаток А

Тестування алгоритмів розв'язання щільно заповнених числових систем лінійних алгебраїчних рівнянь середньої розмірності

Опис тестування функції ESSEMP

Система лінійних рівнянь Дж.Х.Уілкінсона

Для перевірки наростання похибок заокруглення в методах виключення невідомих за рахунок росту проміжних елементів в процесі перетворення матриці Дж.Х.Уілкінсон запропонував систему з такою матрицею:

$$A_W = \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ -1 & 1 & \cdots & 0 & 1 \\ \cdots & \cdots & \cdots & \cdots & 1 \\ -1 & -1 & \cdots & 1 & 1 \\ -1 & -1 & \cdots & -1 & 1 \end{pmatrix}$$

У методах виключення з вибором провідного елемента по стовпцях через ріст елементів у процесі перетворень при подібному заповненні матриці досягається похибка заокруглення порядку $n2^n$. Тут n - порядок системи.

Для спрощення аналізу точності одержаних значень невідомих x_i права підібрана так, щоб точний розв’язок був $x_i = i$ для всіх $i = 1,2,\ldots,n$.

Для розв'язання систем лінійних алгебричних рівнянь з числовими елементами в середовищі MatLab написана і протестована функція Essemp. Ця функція реалізує другий алгоритм відсічених систем і написана за допомогою об’єктно-орієнтованої макромови MatLab

Для спрощення її можливого використання поданий її текст разом з блоком формуванням системи лінійних алгебричних рівнянь, яка має щойно описану матрицю Дж.Х. Уілкінсона.

```
function [] =Essemp_Wilkinson_Test( Dimension )
%-----------------------------------------------------------------------
%   << E S S E M P >> - процедура для рішення невироджених систем
%   лінійних алгебричних рівнянь Ax=b з багатьма правими частинами
%    Вхідні параметри:
%    N - кількість невідомих системи
%    Np  кількість правих частин системи
```

```
%     X  - одномірний масив розміру RBxKB для зберігання обчислених
%     значень невідомих;
%     Y - одномірний робочий масив довжини N.
%     Det - значення  визначника системи
%------------------------------------------------------------------
%   Ввід початкових даних тестової системи
clc
N=0;
while N<=36
N=N+12
for i=1 : N
   Sum=0;
   for j=1 : N
      if (i<j)  A(i,j)=0.0; end
      if (i>j) A(i,j)=-1.0; end
      A(i,i)=1.0;
      A(i,N)=1.0;
      Sum=Sum+A(i,j)*j;
   end
   A(i,N+1)=Sum;
end
% Тіло програми
N1 =N+1;
Np=1;
B=zeros(N);
X=zeros(N);
Y=zeros(N);
Det=1.0;
P=0;
Piv=0;
Sum=0;
for m=1 : N
       M1 =m-1;
       M2 =m-2;
       MP1=m+1;
       Piv=0.0;
       iv =m;
       for i=m : N
            P=A(i,m);
            if (m>1)
              for j=1 : M1 P=P-A(i,j)*X(j,1); end
            end
            if abs(Piv)<abs(P)  Piv=P; iv=i; end
            B(i,m)=P;
       end
        for j=1 : m
            Sum=B(m,j);
            B(m,j)=B(iv,j);
```

```
            B(iv,j)=Sum;
        end
        for j=1 : N1
            Sum=A(m,j);
            A(m,j)=A(iv,j);
            A(iv,j)=Sum;
         end
       Det=Det*B(m,m);
         if m<N  for i=MP1 : N B(i,m)=B(i,m)/B(m,m); end
         end
         if m>1  Y(M1)=B(m,M1); end
         if  m>2 for jr=1 : M2
                    j=m-jr-1;
                    Y(j)=B(m,j);
                    js=j+1;
                    for i=js : M1 Y(j)=Y(j)-B(i,j)*Y(i); end
                   end
          end
          for j=MP1 : N+Np
              P=A(m,j);
              if m>1  for i=1 : M1  P=P-A(i,j)*Y(i); end
          end
              B(m,j)=P/B(m,m);
          end
          X(m,1)=B(m,MP1);
          if m>1  for ir=1 :M1
                    i=m-ir;
                    X(i,1)=B(i,MP1);
                    is=i+1;
                    for  j=is : m  X(i,1)=X(i,1)-B(i,j)*X(j,1); end
                   end
          end
          if m>=N  for j=2 : Np
                       X(m,j)=B(m,N+j);
                       for ir=1 : M1
                         i=m-ir;
                         X(i,j)=B(i,N+j);
                         is=i+1;
                       for jj=is : m X(i,j)=X(i,j)-B(i,jj)*X(jj,j); end
                      end
            end
          end
   end
N
Det
disp('X=')
   for j=1:Np
         X_i= X(:,j);
```

```
    end
  X_i'
end
end
```

Утворена так система розв'язувалася за допомогою функції ESSEMP, а також стандарними методами, включеними до складу пакета MatLab 2007b. Текст цієї невеличкої програми

```
unction [] =MatLab_Wilkinson_Test( Dimension )
%-----------------------------------------------------------------------
%        процедура для тестування  методів лінійної алгебри пакету MatLab
%        на ріст похибки в проміжних обчисленнях за допомогою тестової
%        матриці Уілкінсона
%-----------------------------------------------------------------
%   Ввід початкових даних тестової системи
clc
N=0;
while N<=36
N=N+12
N1 =N+1;
Np=1;
for i=1 : N
   Sum=0;
   for j=1 : N
      if (i<j)  A(i,j)=0.0; end
      if (i>j) A(i,j)=-1.0; end
      A(i,i)=1.0;
      A(i,N)=1.0;
      Sum=Sum+A(i,j)*j;
   end
   B(i)=Sum;
end
 X=B'\A
end
end
```

Результати тестування функції ESSEMP, скопійовані з вікна MatLab і подані в таблиці.

Порядок n **системи**	**Назва функції**	**Значення невідомих** x_i
12	MatLab	0.0189 0.0213 0.0233 0.0248 0.0255 0.0254 0.0241 0.0215 0.0174 0.0117 0.0041 -0.0141
	ESSEMP	1 2 3 4 5 6 7 8 9 10 11 12

24	MatLab	0.0061 0.0062 0.0063 0.0064 0.0064 0.0065 0.0065 0.0065 0.0064 0.0064 0.0062 0.0061 0.0059 0.0056 0.0053 0.0049 0.0045 0.0039 0.0034 0.0027 0.0019 0.0011 0.0002 -0.0059
	ESSEMP	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
36	MatLab	0.0027 0.0027 0.0027 0.0028 0.0028 0.0028 0.0028 0.0028 0.0028 0.0028 0.0028 0.0028 0.0028 0.0027 0.0027 0.0027 0.0026 0.0026 0.0025 0.0024 0.0024 0.0023 0.0022 0.0021 0.0019 0.0018 0.0017 0.0015 0.0013 0.0012 0.0010 0.0008 0.0005 0.0003 0.0000 -0.0027
	ESSEMP	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
48	MatLaB	0.0015 0.0014 0.0014 0.0014 0.0014 0.0013 0.0013 0.0013 0.0012 0.0012 0.0012 0.0011 0.0011 0.0010 0.0010 0.0009 0.0008 0.0008 0.0007 0.0006 0.0006 0.0005 0.0004 0.0003 0.0002 0.0001 0.0000 -0.0015
	ESSEMP	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48

Перша тестова система Уілкінсона-Райнша

Слід відзначити, що для тестування якості алгоритмів розв'язання систем лінійних алгебричних рівнянь у відомому довіднику Дж. Уілкісона і К. Райнша [97] запропоновано систему з такою матрицею:

$$A_{WR1} = \begin{bmatrix} 74 & 80 & 18 & -11 & -4 & -8 \\ 14 & -69 & 21 & 28 & 0 & 7 \\ 66 & -72 & -6 & 7 & 1 & 4 \\ -12 & 66 & -30 & -23 & 3 & -3 \\ 3 & 8 & -7 & -4 & 1 & 0 \\ 4 & -12 & 4 & 4 & 0 & 1 \end{bmatrix}$$

Ця матриця, незважаючи на невеликий порядок, є надзвичайно погано обумовленою: її спектральне число обумовленості при використанні евклідової матричної норми дорівнює $3,66 \times 10^6$.

Права підібрана так, щоб точний розв’язок був $x_i = i$ для всіх $i = 1,2,\ldots,n$. Утворена так система розв'язувалася за допомогою функції

ESSEMP, а також стандарними методами, включеними до складу пакета MatLab 2007b. Текст цієї програми

```
function [] =MatLab_WR_1_Test( Dimension )
%----------------------------------------------------------------------
%      процедура для методів лінійної алгебри пакету MatLab
%    за допомогою першої тестової матриці Уілкінсона-Райнша
%-----------------------------------------------------------------
%   Ввід початкових даних тестової системи
clc
N=6;
A=[-74   80   18   -11   -4   -8;
       14  -60   21    28     0   7;
       66   -72  -6      7     1   4;
      -12    66  -30  -23    3  -3;
        3      8   -7    -4     1   0;
        4    -12   4     4     0   1];
 for i=1 : N
    Sum=0;
    for j=1 : N
        Sum=Sum+j*A(i,j);
    end
    B(i)=Sum;
 end
 X=B'\A
end
```

Для зручності обчислені в обох випадках результати записані до наступної таблиці.

Назва функції	**Обчислені значення невідомих**					
	x_1	x_2	x_3	x_4	x_5	x_6
MatLab	-0.1194	-0.3213	0.2688	0.2149	-0.0187	0.0315
ESSEMP	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000

Друга тестова система Уілкінсона-Райнша

В цьому тесті для розв'язання Дж. Уілкісоном і К. Райншем запропонована система лінійних рівнянь з такою матрицею:

$$A_{WR2} = \begin{bmatrix} 36 & -630 & 3360 & -7560 & 7560 \\ -630 & 14700 & -88200 & 211680 & -220500 \\ 3360 & -88200 & 564480 & -1411200 & 1512000 \\ -7560 & 211680 & 1411200 & 3628800 & 3969000 \\ 7560 & 220500 & 1512000 & 3969000 & 4410000 \end{bmatrix}.$$

Ця система утворена з матриці, оберненої до матриці Гільберта шостого порядку. Права підібрана так, щоб точний розв’язок був $x_i = n - i + 1$ для всіх $i = 1, 2, \ldots, n$.

Утворена так система розв'язувалася за допомогою функції ESSEMP, а також стандарними методами, включеними до складу пакета MatLab 2007b. Текст цієї тестової програми на вхідній мові пакета:

```
function [] =MatLab_WR_2_Test( Dimension )
%-----------------------------------------------------------------------
%     процедура для методів лінійної алгебри пакету MatLab
%    за допомогою другої тестової матриці Уілкінсона-Райнша
%-----------------------------------------------------------------
%     Ввід початкових даних тестової системи
clc
N=5
A=[36       -630      3360      -7560       7560;
     -630    14700   -88200    211680    -220500;
     3360   -88200   564480  -1411200   1512000;
     -7560  211680  1411200 3628800    3969000;
      7560  220500  1512000 3969000    4410000];
for i=1 : N
   Sum=0;
   for j=1 : N
     Sum=Sum+(N+1-j)*A(i,j);
   end
   B(i)=Sum;
end
 disp('X=')
X=B'\A ;
X
end
```

Для зручності порівняння результати обчислені в обох випадках записані до наступної таблиці.

Назва функції	Обчислені значення невідомих				
	x_1	x_2	x_3	x_4	x_5
MatLab	0.0000	0.0127	0.0858	0.2229	0.2461
ESSEMP	5.0000	4.0000	3.0000	2.0000	1.0000

Третя тестова система Уілкінсона-Райнша.

В даній тестовій системі лінійних алгебричних рівнянь використана симетрична матриця 6-го порядку з таким заповненням:

$$A_{WR3} = \begin{pmatrix} 8 & 1 & -2 & 0 & -2 & 5 \\ 1 & 6 & -3 & 2 & 0 & 6 \\ -2 & -3 & 8 & -5 & -6 & 0 \\ 0 & 2 & -5 & 5 & 1 & -2 \\ -2 & 0 & -6 & 1 & 6 & -3 \\ 5 & 6 & 0 & -2 & -3 & 8 \end{pmatrix}$$

Дана матриця має кратні власні значення і є погано обумовленою. Матриця взята із згадуваного вже довідника Уілкінсона-Райнша [97]. Для простоти аналізу права частина системи підібрана так, щоб точний розв'язок був $x_i = i$ для всіх $i = 1, 2, \ldots, n$.

Створена таким чином система розв'язувалася за допомогою функції ESSEMP, а також стандарними методами, включеними до складу пакета MatLab 2007b. Текст цієї програми на вхідній мові пакета

```
function [] =MatLab_WR_3_Test( Dimension )
%-----------------------------------------------------------------------
%    процедура для методів лінійної алгебри пакету MatLab
%    за допомогою третьої тестової матриці Уілкінсона-Райнша
%--------------------------------------------------------------------
%   Ввід початкових даних тестової системи
clc
N=6
A=[8   1  -2   0   -2   5 ;
     1   6  -3   2    0   6 ;
    -2 -3    8  -5  -6   0;
     0 -2   -5   5   1  -2;
    -2  0   -6   1   6  -3;
     5  6    0  -2  -3   8];
for i=1 : N
   Sum=0;
```

```
    for j=1 : N
      Sum=Sum+j*A(i,j);
    end
    B(i)=Sum;
end
X=B'\A;
X
end
```

Результати, обчислені в обох випадках, записані до наступної таблиці.

Назва функції	Обчислені значення невідомих					
	x_1	x_2	x_3	x_4	x_5	x_6
MatLab	0.0899	0.1159	-0.0701	0.0253	0	0.1312
ESSEMP	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000

Четверта тестова система Уілкінсона-Райнша

В цьому тесті використано систему лінійних алгебричних рівнянь з матрицею, яка має верхню форму Хессенберга:

$$A_{WR4} = \begin{pmatrix} 4 & 1 & 2 & 3 & 5 & 6 & 1 \\ 2 & 3 & 1 & -4 & 1 & 2 & 3 \\ 0 & -2 & 1 & 3 & 1 & 4 & 1 \\ 0 & 0 & 3 & 1 & 2 & 2 & 1 \\ 0 & 0 & 0 & -1 & 2 & 1 & 3 \\ 0 & 0 & 0 & 0 & -1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & -2 & 1 \end{pmatrix}$$

Ця система, наведена в довіднику Уілкінсона і Райнша, є погано обумовленою. Права підібрана таким чином, щоб точний розв'язок був $x_i = n - i + 1$ для всіх $i = 1,2,\ldots,n$.

Створена система розв'язувалася за допомогою функції ESSEMP, а також стандарними методами, включеними до складу пакета MatLab 2007b. Текст програми на вхідній мові пакета такий

```
function [] =MatLab_WR_4_Test( Dimension )
%---------------------------------------------------------------------
%     процедура для методів лінійної алгебри пакету MatLab
%    за допомогою четвертої тестової матриці Уілкінсона-Райнша
 %-----------------------------------------------------------------
```

```
%   Ввід початкових даних тестової системи
clc
N=7
A=[4  1  2   3  5  6  1 ;
    2  3  1  -4  1  2  3 ;
    0  -2  1   3  1  4  1;
    0  0  3   1  2  2  1;
    0  0  0  -1  2  1  3;
    0  0  0   0  -1  1  2;
    0  9  9   9   9 -2  1];
for i=1 : N
   Sum=0;
   for j=1 : N
     Sum=Sum+(N+1-j)*A(i,j);
   end
   B(i)=Sum;
end
X=B'\A;
X
end
```

Обчислені в обох випадках результати записані до наступної таблиці.

Назва функції	**Обчислені значення невідомих**						
	x_1	x_2	x_3	x_4	x_5	x_6	x_7
MatLab	0.0115	0.0456	0.0503	0.0473	0.0571	0.0111	0.0118
ESSEMP	7.0000	6.0000	5.0000	4.0000	3.0000	2.0000	1.0000

Таким чином, у розглянутих випадках погано обумовлених систем функція ESSEMP, яка програмно реалізує метод відсічених систем має суттєві переваги порівняно з набором програм лінійної алгебри, включених до пакету MatLab.

Додаток Б

Тестування кліткових алгоритмів розв'язання числових систем лінійних алгебричних рівнянь

Опис тестування функцією ESSEMP

Для розв'язання систем лінійних алгебричних рівнянь з числовими елементами в середовищі MatLab написана і протестована функція Essemp.

Для спрощення її можливого використання поданий її текст разом з блоком формуванням так званої e-діагональної системи лінійних алгебраїчних рівнянь, яка детально буде описана далі.

```
function [] =Bl_Essemp_Epsilon( Dimension )
%-----------------------------------------------------------------------
%   << B L E S S E M  P >> - процедура для розв'язання невироджених систем
%    лінійних алгебричних рівнянь блочним варіантом алгоритму відсічених
систем.
%     Вхідні параметри:
%     RB - розміри кліток,на які розбита вихідна матриця системи
%     рівнянь  Ax=b;
%     KB - кількість блоків при розбитті системи;
%     X  - одномірний масив розміру RBxKB для зберігання обчислених
%     значень невідомих;
%     Y - одномірний робочий масив довжини N.
%    -------------------------------------------------------------------
clc
KB=2; % - кількість блоків системи
RB=12; % - розмір блоків
N_Dim=RB*KB; % обчислення загального порядку системи
%   Ввід початкових даних тестової системи
Epsilon=1;
for NumEps=1:6
Epsilon=Epsilon/10
for i=1 : N_Dim
    for j=1 : N_Dim
        A_sys(i,j)=1.0;
    end
    A_sys(i,i)=1+Epsilon;
    for j=N_Dim+1 : N_Dim+RB
        A_sys(i,N_Dim+j)=0;
    end
    A_sys(i,N_Dim+1)=N_Dim+Epsilon;
end
% Тіло програми
Sum=zeros(RB);
```

```
N=KB;
N1 =N+1;
Np=1;
for i=1 : KB
   for j=1 : KB+1
      i_bgn=(i-1)*RB+1;
      i_end=i*RB;
      j_bgn=(j-1)*RB+1;
      j_end=j*RB;
      A(i,j)={A_sys(i_bgn:i_end,j_bgn:j_end)};
      B(i,j)={zeros(RB)};
      X(i,j)={zeros(RB)};
   end
   Y(i)={zeros(RB)};
end
for m=1 : N
        M1 =m-1;
        M2 =m-2;
        MP1=m+1;
        Piv=zeros(RB);
        P=zeros(RB);
        iv =m;
        for i=m : N
            P=A{i,m};
            if (m>1)
              for j=1 : M1 P=P-X{j,1}*A{i,j}; end
            end
            if abs(det(Piv))<abs(det(P))  Piv=P; iv=i; end
            B{i,m}=P;
        end
        for j=1 : m
            Sum=B{m,j};
            B{m,j}=B{iv,j};
            B{iv,j}=Sum;
        end
        for j=1 : N1
            Sum=A{m,j};
            A{m,j}=A{iv,j};
            A{iv,j}=Sum;
        end
              if m<N  for i=MP1 : N B{i,m}=B{m,m}^(-1)*B{i,m}; end
         end
         if m>1  Y{M1}=B{m,M1}; end
         if  m>2 for jr=1 : M2
                    j=m-jr-1;
                    Y{j}=B{m,j};
```

```
                js=j+1;
                for i=js : M1 Y{j}=Y{j}-B{i,j}*Y{i}; end
              end
        end
        for j=MP1 : N+Np
            P=A{m,j};
            if m>1  for i=1 : M1  P=P-Y{i}*A{i,j}; end
        end
            B{m,j}=B{m,m}^(-1)*P;
        end
        X{m,1}=B{m,MP1};
        if m>1  for ir=1 :M1
                i=m-ir;
                X{i,1}=B{i,MP1};
                is=i+1;
                for  j=is : m  X{i,1}=X{i,1}-B{i,j}*X{j,1}; end
              end
        end
        if m>=N  for j=2 : Np
                   X{m,j}=B{m,N+j};
                   for ir=1 : M1
                     i=m-ir;
                     X{i,j}=B{i,N+j};
                     is=i+1;
                   for jj=is : m X{i,j}=X{i,j}-B{i,jj}*X{jj,j}; end
                  end
          end
        end
  end
% Вивід результату
 disp('X=');
  for i=1 : KB disp(X{i,1}); end
end
end
```

EPSILON-діагональна система лінійних алгебричних рівнянь.

Дана система лінійних алгебричних рівнянь має матрицю вигляду:

$$A_{\varepsilon} = \begin{pmatrix} 1+\varepsilon & 1 & 1 & \ldots & 1 \\ 1 & 1+\varepsilon & 1 & \ldots & 1 \\ 1 & 1 & 1+\varepsilon & \ldots & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & 1 & 1 & \ldots & 1+\varepsilon \end{pmatrix}$$

Спектральне число обумовленості для даної матриці A_e є пропорційним величині $1/\varepsilon$. Отже, для достатньо малих значень ε система стає погано обумовленою. Якщо покласти $a_{i,n+1} = n + e$ (n - порядок системи) , то легко перевірити, що $x_i = 1 \ (i = 1,2,\ldots,n)$ є точним розв`язком даної системи.

Результати тестування функції ESSEMP для $n = 20$ скопійовані з вікна MatLab і подані в наступній таблиці

Значення e	**Значення невідомих** x_i
0.1000	1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.000
0.0100	1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.000
1.0000e-003	1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.000
1.0000e-004	1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.000
1.0000e-005	1.0001 1.0001 1.0001 1.0001 1.0001 1.0001 1.0001 1.0001 1.0001 1.0001 1.0001 1.0001 1.0000 0.9999 0.9999 0.9998 0.9999 1.0001 0.9997 1.0001 0.9999 0.9999 0.9999 0.9999
1.0000e-006	0.9948 0.9948 0.9948 0.9948 0.9948 0.9948 0.9948 0.9948 0.9948 0.9948 0.9948 0.9948 1.0222 0.9993 0.9993 0.9888 0.9993 1.0243 1.0076 0.9909 1.0118 1.0118 0.9951 1.0118

Таким чином, невідомі обчислюються достатньо точно і для великих значень спектрального числа обумовленості.

Система лінійних рівнянь з матрицею Дж.Х. Уілкінсона.

Для перевірки наростання похибок заокруглення в методах виключення невідомих за рахунок росту проміжних елементів в процесі перетворення матриці знову скористаємося вже описаною матрицею Дж.Х. Уілкінсона:

$$A_W = \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ -1 & 1 & \cdots & 0 & 1 \\ \cdots & \cdots & \cdots & \cdots & 1 \\ -1 & -1 & \cdots & 1 & 1 \\ -1 & -1 & \cdots & -1 & 1 \end{pmatrix}$$

Як уже згадувалося в попередньому додатку, при подібному заповненні матриці в методах виключення невідомих досягається похибка заокруглення порядку $n2^n$. Тут n - порядок системи.

Для спрощення аналізу точності одержаних значень невідомих x_i права підібрана так, щоб точний розв'язок був $x_i = i$ для всіх $i = 1, 2, \ldots, n$.

Результати тестування функції ESSEMP скопійовані з вікна MatLab і подані в таблиці.

Порядок n системи	**Розмір блоків**	**Значення невідомих x_i**
24	1	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
24	12	1.0e+005 *0.0008 0.0016 0.0031 0.0063 0.0125 0.0250 0.0499 0.0997 0.1992 0.3984 0.7967 1.593412.8269 13.6853 14.4336 14.9931 15.2380 14.9794 13.9656 11.9449 8.9174 5.8899 6.8899 -53.7955

Одержані результати добре узгоджуються з результатами тестування набору алгоритмів MatLab у Додатку А. Для розмірів блока більших 1 в BLESSEMP використовуються функції MatLab, в яких спостерігається накопичення похибок заокруглення за рахунок значного росту абсолютних величин проміжних результатів обчислень, що характерно для даної матриці.

Додаток В

Тестування алгоритмів розв'язання деяких типів розріджених числових систем лінійних алгебричних рівнянь

Опис тестування функції FC_Three_Diag_Sys.

Для перевірки алгоритму розв'язання 3-х діагональних систем лінійних алгебричних рівнянь методом ланцюгових дробів була використана система рівнянь такого вигляду:

$$\begin{pmatrix} 1.5 & 1 & \cdots & 0 & 0 \\ -1 & 1.5 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1.5 & 1 \\ 0 & 0 & \cdots & -1 & 1.5 \end{pmatrix} \begin{pmatrix} \dfrac{x_1}{x_2} \\ \cdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} \dfrac{3}{0} \\ \cdots \\ 0 \\ 0 \end{pmatrix}$$

Це несиметрична система рівнянь, без діагонального домінування із середнім значенням спектрального числа обумовленості.

Для розв'язання систем лінійних алгебричних рівнянь з числовими елементами в середовищі MatLab написана і протестована функція FC_Three_Diag_Sys. Ця функція реалізує алгоритм розв'язування систем лінійних алгебричних рівнянь методом ланцюгових дробів і написана за допомогою об'єктно-орієнтованої макромови MatLab.

Для спрощення її можливого використання поданий її текст разом з блоком формуванням системи лінійних алгебричних рівнянь, яка має щойно описану матрицю.

```
unction [] =FC_Three_Diag_Sys( )
 %   Розв'язування 3-х діагональних систем лінійних алгебричних рівнянь
 %           Ax=b
 %   за допомогою матричних ланцюгових дробів
    clc
    n=25;
   %  формування тестової системи лінійних рівнянь
     for i=1 : n
        for j=1: n
           A(i,j)=0;
```

```
            if (i==j)   A(i,j)=1.5;  end
            if(i==j+1) A(i,j)=-1; end
            if(j==i+1) A(i,j)=1; end
        end
         b(i)=0;
    end;
    b(1)=3;
  %, обчислення X(1) і решти невідомих
   D(n)=A(n,n);
   i=n;
   while (i>1);
       i = i-1;
       D(i)=A(i,i)-A(i+1,i)*A(i,i+1)/D(i+1);
   end;
   x(1)=b(1)/D(1);
   i=1;
   while (i<n)
     i=i+1;
     x(i)=-A(i,i-1)*x(i-1)/D(i);
  end
 x
 end
```

Результати тестування функції FC_Three_Diag_Sys для $n = 25$ скопійовані з вікна MatLab і подані в таблиці:

Значення n	**Значення невідомих** x_i									
25	1.5000	0.7500	0.3750	0.1875	0.0938	0.0469	0.0234	0.0117	0.0059	0.0029
	0.0015	0.0007	0.0004	0.0002	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000					

Нескладна перевірка показує високу точність запропонованого методу розв'язання трьох діагональних систем методом ланцюгових дробів.

Опис тестування функції ESSELS.

Для перевірки алгоритму розв'язання стрічкового варіанту алгоритму відсічених систем була використана система рівнянь такого вигляду:

$$
\left(\begin{array}{ccccccc}
1+\varepsilon & 1 & 0 & \ddots & 0 & 0 & 0 \\
\hline
1 & 1+\varepsilon & 1 & 0 & \ddots & 0 & 0 \\
-1 & 1 & 1+\varepsilon & 1 & 0 & \ddots & 0 \\
0 & -1 & 1 & \ddots & \ddots & \ddots & \ddots \\
\ddots & \ddots & \ddots & \ddots & 1+\varepsilon & 1 & 0 \\
0 & \ddots & 0 & -1 & 1 & 1+\varepsilon & 1 \\
0 & 0 & \ddots & 0 & -1 & 1 & 1+\varepsilon
\end{array}\right)
\left(\begin{array}{c}
x_1 \\
\hline
x_2 \\
x_3 \\
\cdots \\
x_{n-2} \\
x_{n-1} \\
x_n
\end{array}\right)
=
\left(\begin{array}{c}
2+\varepsilon \\
\hline
3+\varepsilon \\
2+\varepsilon \\
\cdots \\
2+\varepsilon \\
2+\varepsilon \\
1+\varepsilon
\end{array}\right)
$$

Легко бачити, що точним розв'язком системи будуть значення $x_i = 1 \ (i = 1,2,\ldots,n)$

Це несиметрична система рівнянь, без діагонального домінування із значенням спектрального числа обумовленості $n_A = 6.6837\text{e}+010$.

Для розв'язання систем лінійних алгебричних рівнянь з числовими елементами в середовищі MatLab написана і протестована функція ESSELS. Ця функція реалізує алгоритм розв'язування систем лінійних алгебричних рівнянь методом відсічених систем і написана за допомогою об'єктно-орієнтованої макромови MatLab.

Для спрощення її можливого використання поданий її текст разом з блоком формуванням системи лінійних алгебричних рівнянь, яка має щойно описану матрицю.

```
function [] =Essels( Dimension )
%   << E S S E L S >> - процедура для розв'язання стрічкових систем
%   лінійних алгебричних рівнянь.
%   Написана для MatLab 2010 року за алгоритмом відсічених систем
%   Вхідні параметри:
%   A - двовимірний масив розмірності Nx(LN+1) для зберігання
%   вихідних елементів системи Ax=b;
%   N - кількість невідомих системи;
%   N1- параметр рівний N+1;
%   CountOvDiag - параметр рівний кількості наддіагоналей матриці ;
%   CountUndDiag - параметр рівний кількості наддіагоналей матриці ;
%   B - двохмірний робочий масив розмірності Nx(N+1);
%   Y - одномірний робочий масив довжини N.
%   Формування вхідних даних системи Ax=b
clc
N=70
```

```
CountOvDiag=1;
CountUndDiag=2;
N1 =N+1;
Np=1;
Epsilon=0.001;
for i=1 : N
   if(i>1) A(i-1,i)=1.0;end
   if(i<N) A(i+1,i)=1.0; end
   if(i>2)  A(i,i-2)=-1.0; end
   A(i,i)=1.0+Epsilon;
   A(i,N+1)=2+Epsilon;
end
     A(2,N1)=3+Epsilon;
    A(N,N1)=1+Epsilon;
%  Власне алгоритм програми
N1 =N+1;
Np=1;
for i=1 : N
   for j=1: N
      B(i,j)=0.0;
   end
end
for m=1 : N
      if m>1  M1=m-1;end
      if m>2  M2 =m-2; end
       MP1=m+1;
       NKN=m+CountOvDiag;
       if (NKN>=N+1)  NKN=N+1; end
       NKP=m+CountUndDiag;
       if (NKP>=N)     NKP=N;     end
       for i=m : NKP
            P=A(i,m);
            if (m>1)
               if NKP<M1 NM=M1-NKP;else NM=1;end
               for j=NM : M1 P=P-A(i,j)*X(j); end
            end
            B(i,m)=P;
       end
       if(m<N) for i=MP1 : NKP  B(i,m)=B(i,m)/B(m,m); end
       end
       if(m>1) Y(M1)=B(m,M1); end
       if(m>2) for jr=1 : M2  j=m-jr-1; Y(j)=B(m,j); js=j+1;
                      if(js+CountUndDiag<=M1) MKP=js+CountUndDiag; else MKP=M1; end
                      for i=js : MKP  Y(j)=Y(j)-B(i,j)*Y(i); end
          end
       end
       for j=MP1 : N1
           P=A(m,j);
```

```
            if (m>1) for i=1:M1 P=P-A(i,j)*Y(i);end
            end
            B(m,j)=P/B(m,m);
        end
        X(m)=B(m,MP1);
        if(m>1) for ir=1 : M1
                    i=m-ir;
                    X(i)=B(i,MP1);
                    is=i+1;
                    if(is+CountOvDiag<=m)  MKN=is+CountOvDiag; else MKN=m;  end
                    for j=is : MKN   X(i)=X(i)-B(i,j)*X(j);  end

           end
        end

end
 cond(A)
  X
end
```

Для порівняння із штатними програмами лінійної алгебри пакета MatLab була також написана невелика програма MatLab_Band наступного змісту

```
function [] =MatLab_Band( Dimension )
%    процедура для розв'язання стрічкових систем
%    лінійних алгебричних рівнянь засобами MatLab.
%   %   Формування вхідних даних системи Ax=b
clc
N=70
CountOvDiag=1;
CountUndDiag=2;
N1 =N+1;
Np=1;
Epsilon=0.001;
for i=1 : N
   if(i>1) A(i-1,i)=1.0;end
   if(i<N) A(i+1,i)=1.0; end
   if(i>2)  A(i,i-2)=-1.0; end
   A(i,i)=1.0+Epsilon;
   B(i)=2+Epsilon;
end
    B(2)=3+Epsilon;
    B(N)=1+Epsilon;
%  Власне алгоритм програми
 cond(A)
Z=B'\A
end
```

Результати порівняння обох програм подані в наступній таблиці.

Значення $n = 70$	**Значення невідомих** x_i									
Функція MatLab_Band	0.0106	0.0177	0.0177	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142
	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142
	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142
	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142
	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142
	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142
	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0177	0.0177	0.0106
Функція ESSELS	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Таким чином, запропонований алгоритм для даної тестової системи середньої розмірності має суттєві переваги порівняно із стандартними функціями пакета MatLab.

Додаток Д

Тестування алгоритмів розв'язання систем лінійних алгебричних рівнянь з поліноміальними елементами

Опис тестування функції Lambda_Matr_Sys_Ex_1

Для перевірки функціональної придатності описаних алгоритмів алгоритмів зупинимось на розв’язанні конкретної системи з поліноміальними елементами:

$$\begin{pmatrix} \lambda+1 & 2\lambda+1 & \lambda+2 \\ \lambda+2 & \lambda+1 & \lambda+3 \\ \lambda+1 & \lambda+1 & 2\lambda+1 \end{pmatrix}\begin{pmatrix} x_1(\lambda) \\ x_2(\lambda) \\ x_3(\lambda) \end{pmatrix}=\begin{pmatrix} 3\lambda+4 \\ \lambda+2 \\ \lambda+1 \end{pmatrix}.$$

Система вибрана нескладною для можливої перевірки результатів на папері та кращого їх осмислення.

Для її розв'язання вибраний алгоритм зведення до числової системи лінійних алгебричних рівнянь спеціального вигляду із наступним застосування модифікованого варіанту алгоритму відсічених систем. Ця функція написана за допомогою об'єктно-орієнтованої макромови MatLab.

```
function[] = Lambda_Matr_Sys_Ex_1_Numb
   % Розв'язування лінійних систем алгебричних рівнянь
   %      AX=b
   % з поліноміальними елементами шляхом зведення до числових лінійних
   % систем спеціального вигляду
     clc
  A_Lambda(:, :, 1)=[ 1 1 2 -4;
                      2 1 3 -2;
                      1 1  1 -1];
  A_Lambda(:, :, 2)=[1 2 1 -3;
                     1 1 3  -1;
                     1 1 2 -1];
  L=1; % максимальний порядок поліноміальних елементів системи
  N=3; % порядок системи алгебричних рівнянь
 %  початок процедури
 N1=N+1;
 Ni=L*N*(N+1)+N;
 NIP1=Ni+1;
 for i=1 : Ni
    for j=1 : NIP1
```

```
        A(i,j)=0.0;
       end
   end
     K=0;
     K1=0;
   while K+(L+1)*N<=Ni
     for m=1: L+1
       for i=1 : N
           for j=1: N1
               A(K+(m-1)*N+i,K1+j)=A_Lambda(i,j,m);
           end
       end
     end
    K=K+N;
    K1=K1+N1;
    end
 KN=L*N-1;
 KP=L*N-1;
 % A
 N=Ni;
 N1=N+1;
 Np=1;
 % .ESSEMP(At,Ni,Nip1,Yt,IER);
 B=zeros(N);
 X=zeros(N);
 Y=zeros(N);
 P=0;
 Piv=0;
 Sum=0;
 for m=1 : N
         M1 =m-1;
         M2 =m-2;
         MP1=m+1;
         Piv=0.0;
         iv =m;
         for i=m : N
              P=A(i,m);
              if (m>1)
                for j=1 : M1 P=P-A(i,j)*X(j,1); end
              end
              if abs(Piv)<abs(P)  Piv=P; iv=i; end
              B(i,m)=P;
         end
          for j=1 : m
              Sum=B(m,j);
```

```
            B(m,j)=B(iv,j);
            B(iv,j)=Sum;
        end
        for j=1 : N1
            Sum=A(m,j);
            A(m,j)=A(iv,j);
            A(iv,j)=Sum;
         end
             if m<N  for i=MP1 : N B(i,m)=B(i,m)/B(m,m); end
         end
         if m>1  Y(M1)=B(m,M1); end
         if  m>2 for jr=1 : M2
                    j=m-jr-1;
                    Y(j)=B(m,j);
                    js=j+1;
                    for i=js : M1 Y(j)=Y(j)-B(i,j)*Y(i); end
                  end
          end
          for j=MP1 : N+Np
               P=A(m,j);
               if m>1  for i=1 : M1  P=P-A(i,j)*Y(i); end
          end
               B(m,j)=P/B(m,m);
          end
          X(m,1)=B(m,MP1);
          if m>1  for ir=1 :M1
                    i=m-ir;
                    X(i,1)=B(i,MP1);
                    is=i+1;
                    for  j=is : m  X(i,1)=X(i,1)-B(i,j)*X(j,1); end
                  end
          end
          if m>=N  for j=2 : Np
                       X(m,j)=B(m,N+j);
                       for ir=1 : M1
                         i=m-ir;
                         X(i,j)=B(i,N+j);
                         is=i+1;
                       for jj=is : m X(i,j)=X(i,j)-B(i,jj)*X(jj,j); end
                     end
            end
          end
end
for i=1:N  X_(i)= X(i,1); end
  disp('X=');
```

```
disp(X_);
end
```

Розв’язок для системи n-го порядку з поліноміальними елементами порядку не вище l шукається у вигляді:

$$x(\lambda) = \sum_{j=0}^{nl} \lambda^j X_{nl-j} \Bigg/ \sum_{j=0}^{nl} \lambda^j y_{nl-j},$$

де $X_1, X_2, ..., X_{nl}$ – вектори розмірності n, а $y_1, y_2, \ldots, y_{nl}$ - скаляри .

В результаті обчислень у вікні отримані такі коефіцієнти числових векторів

X=

5.0000 -3.0000 -3.0000 -1.0000 12.0000 -5.0000 -5.0000 -1.0000 9.0000 -5.0000 -2.0000 0.0000 1.0000 -2.0000 0

На основі проведених обчислень можна виписати для даної системи такі розв'язки:

$$x_3(l) = \frac{5l^3 - 3l^2 - 3l + 1}{l^3 - 2l^2 + 1},\ x_2(l) = \frac{12l^3 + 5l^2 - 5l - 1}{l^3 - 2l^2 + 1};$$

$$x_1(l) = \frac{9l^3 + 5^2 - 2l}{l^3 - 2l^2 + 1}.$$

Неважко перевірити, що знайдені розв'язки задовольняють даній системі лінійних алгебричних рівнянь.