Лекція 6. Чисельні методи розв’язання систем лінійних алгебраїчних рівнянь

1. Системи лінійних алгебраїчних рівнянь.
   1. Основні поняття.
   2. Прямі методи розв’язання СЛАР.
      1. Метод Гаусса.
         1. Метод Гаусса з вибором головного елементу.
         2. Метод Гаусса-Жордана.
         3. Матрична форма метода Гаусса. Розкладання матриці на множники (LU-розклад).
      2. Метод квадратного кореня (Холецького).
      3. Метод прогонки.
   3. Ітераційні методи розв’язання СЛАР.
      1. Метод простої ітерації.
      2. Метод Гаусса-Зейделя.
2. Системи нелінійних рівнянь.

# Системи лінійних алгебраїчних рівнянь

## Основні поняття

Лінійні системи. Багато практичних задач зводяться до розв’язання системи лінійних рівнянь.

Запишемо систему *n* лінійних алгебраїчних рівнянь з *n* невідомими:

|  |  |
| --- | --- |
|  | (1) |

Сукупність коефіцієнтів цієї системи запишемо у виді таблиці

|  |  |
| --- | --- |
|  |  |

Ця таблиця *n*2 елементів, що складається з *n* рядків і *n* стовпців, називається квадратною матрицею порядку *n*. Якщо подібна таблиця містить *mn* елементів, розташованих в *m* рядках і *n* стовпцях, то вона називається прямокутною матрицею.

Використовуючи поняття матриці *А*, систему рівнянь (1) можна записати у векторно-матричному виді:

|  |  |
| --- | --- |
| *A***x** = **b** | (2) |

де **х** і **b** – вектор-стовпець невідомих і вектор-стовпець правих частин відповідно:

|  |  |
| --- | --- |
|  |  |

чи, в компактнішому записі,

|  |  |
| --- | --- |
| **x** = {*x*1, *x*2, …, *xn*}, **b** = {*b*1, *b*2, …, *bn*} |  |

У ряді випадків виходять системи рівнянь з деякими спеціальними видами матриць. Ось деякі приклади таких матриць:

|  |  |
| --- | --- |
|  |  |

Тут:

*А* – симетрична матриця (її елементи розташовані симетрично відносно головній діагоналі (*aij* = *aji*));

*B* – верхня трикутна матриця з рівними нулю елементами, розташованими нижче діагоналі;

*C* – клітинна матриця (її ненульові елементи складають окремі групи (клітини));

*F* – стрічкова матриця (її ненульові елементи складають «стрічку», паралельну діагоналі (в даному випадку стрічкова матриця *F* одночасно є також трьохдіагональною));

*Е* – одинична матриця (окремий випадок діагональної);

*О* – нульова матриця.

Визначником (детермінантом) матриці *А* *n*-го порядку називається число D, рівне

|  |  |
| --- | --- |
|  | (3) |

Тут індекси *α*, *β*, ..., *ω* пробігають усі можливі *n*! перестановок номерів 1,2,..., *n*; *k* – число інверсій в цій перестановці (під інверсією розуміється обмін двох індексів місцями; за допомогою таких обмінів перестановка *α*, *β*, ..., *ω* виходить з перестановки 1, 2, ..., *n*.)

Необхідною і достатньою умовою існування єдиного рішення системи лінійних рівнянь є умова *D* ≠ 0. У разі рівності нулю визначника системи матриця називається виродженою; при цьому система лінійних рівнянь(1) або не має рішення, або має їх нескінченну множину.

На практиці, особливо при обчисленнях на комп'ютері, коли відбуваються округлення або відкидання молодших розрядів чисел, далеко не завжди можна отримати точну рівність визначника нулю. При D ≈ 0 рішення дуже чутливо до зміни коефіцієнтів системи.

Таким чином, малі погрішності обчислень або початкових даних можуть привести до істотних погрішностей в рішенні. Такі системи рівнянь називаються погано обумовленими.

Відмітимо, що умова D ≈ 0 є необхідною для поганої обумовленості системи лінійних рівнянь, але не достатньою. Наприклад, система рівнянь *n*-го порядку з діагональною матрицею з елементами *aii* = 0.1 не є погано обумовленою, хоча її визначник малий (*D* = 10 –*n*). Строгий критерій поганої обумовленості системи лінійних рівнянь можна знайти в повніших курсах по чисельних методах.

Про методи рішення лінійних систем. Методи рішення систем лінійних рівнянь діляться на дві групи – прямі і ітераційні. Прямі методи використовують кінцеві співвідношення (формули) для обчислення невідомих. Вони дають рішення після виконання заздалегідь відомого числа операцій. Ці методи порівняно прості і найбільш універсальні, тобто придатні для розв’язання широкого класу лінійних систем.

В той же час прямі методи мають і ряд недоліків. Як правило, вони вимагають зберігання в оперативній пам'яті комп'ютера відразу усієї матриці, і при великих значеннях *n* витрачається багато місця в пам'яті. Далі, прямі методи зазвичай не враховують структуру матриці при великому числі нульових елементів в розріджених матрицях (наприклад, клітинних або стрічкових) ці елементи займають місце в пам'яті машини, і над ними проводяться арифметичні дії. Виключенням тут є спеціалізовані методи для таких рівнянь, наприклад, метод прогонки. Істотним недоліком прямих методів є також накопичення погрішностей в процесі рішення, оскільки обчислення на будь-якому етапі використовують результати попередніх операцій. Це особливо небезпечно для великих систем, коли різко зростає загальне число операцій, а також для погано обумовлених систем, дуже чутливих до погрішностей. У зв'язку з цим прямі методи використовуються зазвичай для не занадто великих (*n* < 1000) систем з щільно заповненою матрицею і не близьким до нуля визначником.

Відмітимо ще, що прямі методи рішення лінійних систем іноді називають точними, оскільки рішення виражається у вигляді точних формул через коефіцієнти системи. Проте точне рішення може бути отримане лише при точному виконанні обчислень (і, зрозуміло, при точних значеннях коефіцієнтів системи). На практиці ж при використанні комп'ютерів обчислення проводяться з погрішностями. Тому неминучі погрішності і в остаточних результатах.

Ітераційні методи – це методи послідовних наближень. У них необхідно задати деяке наближене рішення – початкове наближення. Після цього за допомогою деякого алгоритму проводиться один цикл обчислень, що називається ітерацією. В результаті ітерації знаходять нове наближення. Ітерації проводяться до отримання рішення з необхідною точністю. Алгоритми рішення лінійних систем з використанням ітераційних методів зазвичай складніші в порівнянні з прямими методами. Об'єм обчислень заздалегідь визначити важко.

Проте ітераційні методи у ряді випадків прийнятніше. Вони вимагають зберігання в пам'яті машини не усієї матриці системи, а лише декількох векторів з *n* компонентами. Іноді елементи матриці можна зовсім не зберігати, а обчислювати їх в міру необхідності. Погрішності остаточних результатів при використанні ітераційних методів не накопичуються, оскільки точність обчислень в кожній ітерації визначається лише результатами попередньої ітерації і практично не залежить від раніше виконаних обчислень. Ці достоїнства ітераційних методів роблять їх особливо корисними у разі великого числа рівнянь, а також погано обумовлених систем. Слід зазначити, що при цьому збіжність ітерацій може бути дуже повільною; тому шукаються ефективні шляхи її прискорення.

Ітераційні методи можуть використовуватися для уточнення рішень, отриманих за допомогою прямих методів. Такі змішані алгоритми зазвичай досить ефективні, особливо для погано обумовлених систем.

## Прямі методи розв’язання СЛАР

### Вступні зауваження

Одним із способів рішення системи лінійних рівнянь є правило Крамера, згідно з яким кожне невідоме представляється у вигляді відношення визначників. Запишемо його для системи

|  |  |
| --- | --- |
|  |  |

Тоді

|  |  |
| --- | --- |
|  |  |

Можна спробувати використати це правило для розв’язання систем рівнянь довільного порядку. Проте при великому числі рівнянь потрібно буде виконати величезне число арифметичних операцій, оскільки для обчислень *n* невідомих необхідно знайти значення визначників, число яких *n* + 1. Кількість арифметичних операцій можна оцінити так

|  |  |
| --- | --- |
|  |  |

Тому правило Крамера можна використати лише для розв’язання систем, що складаються з декількох рівнянь.

Відомий також метод рішення лінійної системи з використанням зворотної матриці. Система записується у виді *А***х** = **b** (див. (2)). Тоді, помноживши обидві частини цього векторного рівняння зліва на зворотну матрицю *А*–1, отримуємо **х** = *A*–1**b**. Проте якщо не використати економічних схем для обчислення зворотної матриці, цей спосіб також непридатний для практичного розв’язання лінійних систем при великих значеннях п із-за великого об'єму обчислень.

Найбільш поширеними серед прямих методів є метод виключення Гауса і його модифікації. Нижче розглядається застосування методу виключення для розв’язання систем лінійних рівнянь, а також для обчислення визначника і знаходження зворотної матриці.

### Метод Гауса

Він оснований на приведенні матриці системи до трикутного виду. Це досягається послідовним виключенням невідомих з рівнянь системи. Спочатку за допомогою першого рівняння виключається *х*1 з усіх наступних рівнянь системи. Потім за допомогою другого рівняння виключається *х*2 з третього і усіх наступних рівнянь. Цей процес, що називається прямим ходом методу Гауса, триває до тих пір, поки в лівій частині останнього (*n*-го) рівняння не залишиться лише один член з невідомим *хn*, тобто матриця системи буде приведена до трикутного виду.

Зворотний хід методу Гауса полягає в послідовному обчисленні шуканих невідомих: вирішуючи останнє рівняння, знаходимо єдине в цьому рівнянні невідоме *хn*. Далі, використовуючи це значення, з попереднього рівняння обчислюємо *хn*–1 і т. д. Останнім знайдемо *х*1 з першого рівняння.

Відмітимо, що описані процедури застосовні лише для систем з невиродженою матрицею. Інакше (за умови, що обчислення проводяться точно) за допомогою методу Гауса можна відповісти на питання, чи має система нескінченну безліч рішень або не має жодного. Проте ці випадки ми надалі розглядати не будемо, припускаючи, що матриця системи невироджена.

Розглянемо застосування методу Гауса для системи

|  |  |
| --- | --- |
|  | (4) |

Для виключення *х*1 з другого рівняння додамо до нього перше, помножене на –*a*21/*a*11. Потім, помноживши перше рівняння на –*a*31/*a*11 і додавши результат до третього рівняння, також виключимо з нього *х*1. Отримаємо рівносильну (4) систему рівнянь виду

|  |  |
| --- | --- |
|  | (5) |

Тепер з третього рівняння системи (5) треба виключити *х*2. Для цього помножимо друге рівняння на –*а*'32/*a*'22 і додамо результат до третього. Отримаємо

|  |  |
| --- | --- |
|  | (6) |

Матриця системи (6) має трикутний вигляд. На цьому закінчується прямий хід методу Гауса.

В процесі виключення невідомих доводиться виконувати операції ділення на коефіцієнти *а*11, *а*'22 і т. д. Тому вони мають бути відмінні від нуля. Інакше необхідно відповідним чином переставити рівняння системи. Перестановка рівнянь має бути передбачена в обчислювальному алгоритмі при його реалізації на комп'ютері.

Зворотний хід розпочинається з рішення третього рівняння системи (6):

|  |  |
| --- | --- |
|  |  |

Використовуючи це значення, можна знайти *x*2 з другого рівняння, а потім *х*1 з першого:

|  |  |
| --- | --- |
|  |  |

Аналогічно будується обчислювальний алгоритм для лінійної системи з довільним числом рівнянь. Загальні формули прямого ходу можна вивести наступним чином.

Нехай дана система *n* лінійних алгебраїчних рівнянь з *n* невідомими:

|  |  |
| --- | --- |
|  | (6.1) |

Припустимо, що *a*11 ≠ 0. В протилежному випадку можна поміняти місцями перше рівняння з рівнянням, в якому коефіцієнт при невідомому *x*1 відмінний від нуля.

**Прямий хід**

Крок *k* = 1

Розділимо перше рівняння системи (6.1) на *a*11. Воно прийме вид

|  |  |
| --- | --- |
| де | (6.2) |

Помножимо розрішуюче рівняння (6.2) на *a*21 і віднімемо отримане рівняння з другого рівняння системи (6.1). Аналогічно перетворимо решту рівнянь. В результаті цих операцій система запишеться так:

|  |  |
| --- | --- |
| де | (6.3) |

Якщо який-небудь з коефіцієнтів *aj*1 виявиться рівним нулю, то *j*-рівняння системи (6.1) увійде в систему (6.3) без змін, тобто

Крок *k* = 2

Тепер, залишивши без змін перше рівняння системи (6.3), можна зробити розрішуючим друге рівняння системи і застосувати описану процедуру до системи з *n*– 1 рівнянь, виключивши невідоме *x*2 з третього і наступних рівнянь. Отримаємо систему виду

|  |  |
| --- | --- |
| де | (6.4) |

Крок *k*

Коли виконані перші *k* – 1 кроків і вже виключені змінні *x*1, *x*2, …, *xk*– 1 тоді для виключення матимемо скорочену систему *n*– *k*+ 1 рівнянь, яка не містить змінних, виключених на попередніх кроках:

|  |  |
| --- | --- |
|  | (6.5) |

Тепер розрішуючим буде *k*-те рівняння системи і необхідно виконати процедуру виключення невідомого *xk* з (*k*+ 1)-ого і наступних рівнянь. Коефіцієнти рівнянь перераховуються за наступними формулами:

|  |  |
| --- | --- |
|  | (6.6) |

Після *k*-го кроку скорочена система рівнянь матиме вид:

|  |  |
| --- | --- |
|  | (6.7) |

Крок *n*

В результаті виконання прямого ходу система прийме наступний вид:

|  |  |
| --- | --- |
|  | (6.8) |

**Зворотній хід**

Значення невідомих знаходять за формулами зворотного ходу:

|  |  |
| --- | --- |
|  | (6.9) |

### Метод Гаусса з вибором головного елемента

Основним обмеженням методу Гаусса є припущення про те, що всі елементи *аkk*(*k*– 1) на які проводиться ділення на кожному кроці методу, відмінні від нуля. Елемент *аkk*(*k*– 1) називається *ведучим елементом* на *k*-му кроці виключення. Слід мати на увазі, що навіть якщо якийсь ведучий елемент не дорівнює нулю, а просто близький до нього, в процесі обчислень може відбуватися значне накопичення похибок. Уникнути цього дозволяє *метод Гаусса з вибором головного елемента*. Ідея методу полягає в тому, щоб на черговому кроці виключати ту невідому, при якій стоїть найбільший за модулем коефіцієнт. Отже, ведучим елементом тут вибирається *головний*, тобто *найбільший за модулем* елемент матриці. Тим самим, якщо detА ≠ 0, то в процесі обчислень не відбуватиметься ділення на нуль.

Для зменшення помилок округлювання чинять так. Серед елементів першого рядка *аkj*(*k*– 1) кожної проміжної матриці (6.5) вибирають найбільший за модулем елемент *max*|*аkj*(*k*– 1)|, *j* = *k*, *k* +1,..., *n* і роблять цей елемент ведучим. Вказаний спосіб виключення називається методом Гаусса з *вибором головного елемента по рядку*. Він еквівалентний застосуванню звичайного методу Гаусса до системи, в якій на кожному кроці виключення проводиться відповідна перенумерація змінних.

Застосовується також метод із *вибором головного елемента по стовпцю* (рис. 1, а). Він еквівалентний застосуванню звичайного методу Гаусса до системи, в якій на кожному кроці виключення проводиться відповідна перенумерація рівнянь.

|  |  |
| --- | --- |
|  |  |

Проте головний елемент можна вибирати і серед всіх елементів неперетвореної частини матриці (рис. 1, б):

|  |  |
| --- | --- |
|  |  |

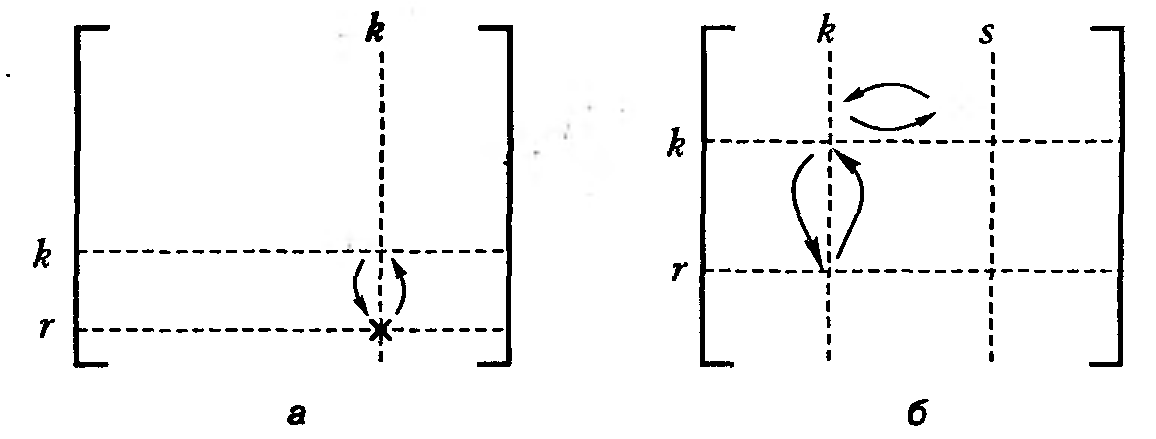


Рис. 1. Вибір головного елемента: а– серед елементів стовпця матриці; *б* – серед елементів не перетвореної частини матриці.

Складність алгоритму (кількість арифметичних операцій) оцінюється величиною

|  |  |
| --- | --- |
|  |  |

### Метод Гауса-Жордана

Метод є модифікацією методу Гауса. Метод Жордана-Гауса відрізняється від методу Гауса, тільки послідовністю виключення змінних при виконанні зворотного ходу рішення. Після закінчення прямого ходу, у верхній трикутній матриці продовжують виключати невідомі. У результаті отримують одиничну матрицю і рішення на місці вільного вектора (над ним звичайно необхідно виконувати ті самі перетворення, що і над матрицею системи).

### Матрична форма метода Гауса. Розкладання матриці на множники (LU-розклад)

Якщо представити алгоритм методу Гауса у матричних позначеннях, то можна побачити, що він відповідає розкладенню матриці **A** на добуток більш простих матриць.

Результат обчислень 1-го кроку виключення змінних системи можна представити в матричному виді

|  |  |
| --- | --- |
| **A**(1)**x** = **b**(1), | (6.10) |

де

|  |  |
| --- | --- |
| , |  |

а коефіцієнти , (*i*, *j* = 2, 3, …, *n*) обчислюються за формулами

|  |  |
| --- | --- |
|  |  |

Введемо матрицю

|  |  |
| --- | --- |
|  |  |

Як неважко перевірити, справедливі рівності

**A**(1) = **M1A**, **b**(1) = **M1b1**

тобто перетворення системи (2) до виду (6.10) еквівалентне множенню лівої і правої частин системи на матрицю **M1**.

На *k*-му кроці матрицю системи **A**(*k*) можна представити як добуток **M***k*∙…∙**M**2**M**1**A**. В результаті прямого ходу отримують систему рівнянь

|  |  |
| --- | --- |
| **A**(*n*– 1)**x** = **b**(*n*– 1), | (6.11) |

де матриця якої **A**(*n*– 1) = **M***n –*1∙…∙**M**2**M**1**A**. є верхня трикутна з одиничною головною діагоналлю. Ця матриця звичайно позначається

|  |  |
| --- | --- |
| **U** = **A**(*n*– 1). | (12) |

А саму матрицю вихідної системи **A** можна представити у виді добутку двох трикутних матриць – нижньої трикутної **L**, і верхньої трикутної **U**.

|  |  |
| --- | --- |
| **A** = **LU** | (13) |

Де **L** = **M**1–1**M**2–1∙…∙**M***n*–1 з (12).

Наприклад, для матриці 3-го порядку:

|  |  |
| --- | --- |
|  |  |

При використанні LU-розкладу метод Гаусса можна реалізувати інакше. Нехай задані матриця **А** і вектор **b**. Спочатку проводиться розкладання **А** в добуток двох трикутних матриць, **А** = **LU**. Початкова система набуває вигляду **LUx** = **b**, її розв'язання рівносильно послідовному розв'язанню двох систем рівнянь

|  |  |
| --- | --- |
| **Ly** = **b**,  **Ux** = **у** | (14) |

з трикутними матрицями, звідки й знаходять шуканий вектор **х**. Розкладання **A** = **LU** відповідає прямому ходу методу Гаусса, а розв'язання системи (14) – зворотному ходу.

Трикутні матриці **L** та **U**, на які розкладається матриця **А**, на однопроцесорних ЕОМ одержують не виконанням процедури матричних перетворень, що ґрунтуються на використанні послідовності елементарних нижніх трикутних матриць **L***k*, а за допомогою прямих рекурентних формул перетворення, отриманих на основі формул множення матриць.

|  |  |
| --- | --- |
|  |  |

при цьому *lsk* = 0 для *k* > *s*, *ukj* = 0 для *k* > *j* і *lss* =1.

Елементи нижньої **L** і верхньої **U** матриць обчислюються за формулами:

|  |  |
| --- | --- |
|  | (15) |

Формули (14) зводяться до

|  |  |
| --- | --- |
|  | (16) |

Складність LU-розкладання, що виконується на однопроцесорній ЕОМ, оцінюється величиною

|  |  |
| --- | --- |
|  |  |

до якої додається складність розв'язку двох систем лінійних рівнянь із трикутними матрицями (16), що складає приблизно *n*2 операцій множення/ділення.

У порівнянні з методом Гаусса основною перевагою LU-розкладання, яка випливає з формул (14), (16), є можливість за знайдених **L** і **U** та різних векторах правої частини **b** за результатами попереднього розкладання багатократно повторювати тільки зворотний хід розв'язання системи лінійних рівнянь.

Приклад

|  |
| --- |
|  |

### Метод квадратного кореня (Холецького)

Метод Холецького використовується для розв'язання систем лінійних рівнянь з симетричними додатними матрицями (*аij* = *aji*) У цих випадках приймається, що *ukk* = *lkk*, *usj* = *ljs*

З добутку двох матриць

|  |  |
| --- | --- |
|  |  |

за правилами матричного множення знаходимо співвідношення між елементами матриць:

|  |  |
| --- | --- |
|  |  |

За формулами (15) маємо

|  |  |
| --- | --- |
|  | (17) |

Тому перетворення Холецького для симетричних матриць набуває такого вигляду:

|  |  |
| --- | --- |
|  | (18) |

### Метод прогону

Він є модифікацією методу Гауса для окремого випадку розріджених систем – системи рівнянь з трьохдіагональною матрицею. Такі системи виходять при моделюванні деяких інженерних завдань, а також при чисельному рішенні крайових завдань для диференціальних рівнянь.

Запишемо систему рівнянь у виді.

|  |  |
| --- | --- |
|  | (19) |

На головній діагоналі матриці цієї системи стоять елементи *b*1, *b*2 … *bn*, над нею – елементи *c*1, *с*2, ..., *cn*–1 i, під нею – елементи *а*2, *а*3,..., *аn* (при цьому зазвичай усі коефіцієнти *bi* не дорівнюють нулю). Інші елементи матриці дорівнюють нулю.

Метод прогону складається з двох етапів — прямого прогону (аналога прямого ходу методу Гаусcа) і зворотного прогону (аналога зворотного ходу методу Гаусcа). Прямий прогін полягає в обчисленні прогонних коефіцієнтів *Аi*, *Bi*, за допомогою яких кожне невідоме *xi* виражається через *xi*+1.

|  |  |
| --- | --- |
|  | (20) |

З першого рівняння системи (19) знайдемо

|  |  |
| --- | --- |
|  |  |

З іншого боку, по формулі (20) *х*1 = *А*1*х*2 + *В*1. Прирівнюючи коефіцієнти в обох виразах для *x*1 отримуємо

|  |  |
| --- | --- |
|  | (21) |

Аналогічно обчислюються прогонні коефіцієнти для будь-якого номера *i*:

|  |  |
| --- | --- |
|  | (22) |

Зворотний прогін полягає в послідовному обчисленні невідомих *xi*. Спочатку треба знайти *хn*.

|  |  |
| --- | --- |
|  |  |

Далі, використовуючи формули (20) і вичислені раніше по формулах (21), (22) прогонні коефіцієнти, послідовно обчислюємо усі невідомі *xn*– 1, *xn*– 2,..., *х*1.

Приклад

|  |
| --- |
| Розглянемо алгоритм знаходження розв’язку СЛАР з тридіагональною матрицею для такої системи:  *n* = 3. Прямий хід виконується *n* – 1 раз (*n* – 1 = 3 – 1 = 2)  *Пряма прогонка.*  За формулами прямого ходу маємо:  Крок *i* = 1.  Крок *i* = 2.  *Зворотна прогонка.* |

## Ітераційні методи розв’язання СЛАР

### Уточнення розв’язку

Розв’язок, що отримуються за допомогою *прямих методів*, зазвичай містять *погрішності*, *викликані округленнями* при виконанні операцій над числами з плаваючою точкою на комп'ютері з обмеженим числом розрядів. У ряді випадків ці погрішності можуть бути значними, і необхідно знайти спосіб їх зменшення. Розглянемо тут один з *методів*, що дозволяє *уточнити рішення*, отримане за допомогою прямого методу.

Знайдемо розв’язок системи лінійних рівнянь.

|  |  |
| --- | --- |
| *А***х** = **b**. | (23) |

Нехай за допомогою деякого прямого методу отриманий наближений розв’язок **х**(0) (тобто наближені значення невідомих *х*1(0), *х*2(0),..., *xn*(0)), що називається початковим або нульовим наближенням до розв’язку.

Підставляючи цей розв’язок в ліву частину системи (23), отримуємо деякий стовпець правих частин **b**(0), відмінний від **b**:

|  |  |
| --- | --- |
| *А***х**(0) = **b**(0). | (24) |

Введемо позначення: **Δх**(0) – похибка отриманого розв’язку, **r**(0) – нев'язка, тобто

|  |  |
| --- | --- |
| **Δx**(0) = **x** – **x**(0), **r**(0) = *A***x**(0) – **b** = **b**(0) – **b**. | (25) |

Віднімаючи рівність (24) з рівності (23), з урахуванням позначень (25) отримуємо

|  |  |
| --- | --- |
| *A***Δx**(0) = –**r**(0). | (26) |

Розв’язуючи цю систему, знаходимо значення похибки **Δх**(0), яке використовуємо в якості поправки до наближеного розв’язку **х**(0), обчислюючи таким чином новий наближений розв’язок **х**(1) (чи наступне наближення до розв’язку):

|  |  |
| --- | --- |
| **х**(1) = **х**(0) + **Δх**(0). |  |

Таким же способом можна знайти нову поправку до розв’язку **Δх**(1) і наступне наближення

|  |  |
| --- | --- |
| **х**(2) = **х**(1) + **Δх**(1). |  |

і т. д. Процес триває до тих пір, поки чергове значення похибки (поправки) **Δх**(*k*) не стане досить малим, тобто поки чергові наближені значення невідомих х1(*k*+ 1), х2(*k*+ 1),...,х*n*(*k*+ 1) не відрізнятимуться від попередніх значень х1(*k*), х2(*k*),...,х*n*(*k*) на малу величину.

Розглянутий процес уточнення розв’язку є фактично ітераційним методом розв’язування системи лінійних рівнянь. При цьому відмітимо, що для знаходження чергового наближення, тобто на кожній ітерації, розв’язуються системи рівнянь виду (26) з однією і тією ж матрицею, що є матрицею початкової системи (23), при різних правих частинах. Це дозволяє будувати економічні алгоритми. Наприклад, при використанні методу Гаусса скорочується об'єм обчислень на етапі прямого ходу.

*Розв’язання СЛАР* за допомогою розглянутого методу (а також розв’язання систем лінійних рівнянь іншими *ітераційними методами*, розв’язок ітераційними методами рівнянь іншого виду і їх систем) зводиться до наступного.

(1) Вводяться початкові дані, наприклад, *коефіцієнти рівнянь* і допустиме *значення* *похибки*.

(2) Задаються *початкові наближення* значень невідомих (вектор-стовпець **х**(0)). Вони або вводяться в комп'ютер, або обчислюються яким-небудь чином (зокрема, шляхом розв’язання системи рівнянь за допомогою прямого методу).

(3) Потім організовується *циклічний обчислювальний процес*, кожен цикл якого є однією ітерацією – перехід від попереднього наближення **х**(*k*– 1) до наступного **х**(*k*). Якщо виявляється, що зі збільшенням числа ітерацій наближений розв’язок прагне до точного:

|  |  |
| --- | --- |
|  |  |

то ітераційний метод називають таким, що *сходиться*.

На практиці наявність збіжності і досягнення необхідної точності зазвичай визначають приблизно, поступаючи таким чином. При малій (із заданою допустимою похибкою) зміні **х** на двох послідовних ітераціях, тобто при малій відмінності **х**(*k*) від **х**(*k*– 1), процес припиняється, і відбувається виведення значень невідомих, отриманих на останній ітерації.

Можливі різні підходи до визначення малості відмінності **х** на двох послідовних ітераціях. Наприклад, якщо задана допустима похибка *ε* > 0, то критерієм закінчення ітераційного процесу можна вважати виконання однієї з трьох нерівностей:

|  |  |
| --- | --- |
|  | (27) |
|  | (28) |
|  | (29) |

Тут в першому випадку відмінність векторів х(*k*) і х(*k*– 1) «на *ε*» розуміється в сенсі малості модуля їх різниці, в другому – в сенсі малості різниць усіх відповідних компонент векторів, в третьому – в сенсі малості відносних різниць компонент. Якщо система не є погано обумовленою, то в якості критерію закінчення ітераційного процесу можна використати і умову малості нев'язки, наприклад

|  |  |
| --- | --- |
|  | (30) |

Відмітимо, що в розглянутому алгоритмі не передбачений випадок відсутності збіжності. Для запобігання непродуктивним витратам машинного часу в алгоритм вводять лічильник числа ітерацій і при досягненні їм деякого заданого значення рахунок припиняють.

### Метод простої ітерації

Цей метод широко використовується для чисельного розв’язання рівнянь і їх систем різних видів. Розглянемо застосування методу простої ітерації до розв’язання систем лінійних алгебраїчних рівнянь.

Запишемо початкову систему рівнянь у векторно-матричному виді (2) і виконаємо ряд тотожних перетворень:

|  |  |
| --- | --- |
| **Ax**=**b**; **0**= **b**–**Ax**; **x**= **b**–**Ax**+ **x**;  **x**= (**b**–**Ax**)τ+ **x**; **x**= (**E**–τ**A**)**x**+ τ**b**;  **x**= **αx**+ **β**, | (31) |

де *τ* ≠ 0 – деяке число, **Е** — одинична матриця, **α** = **Е** – *τ***А**, **β**=τ**b**. Отримана система (31) еквівалентна початковій системі і служить основою для побудови методу простої ітерації.

Виберемо деяке початкове наближення **х**(0) і підставимо його в праву частину системи (31):

|  |  |
| --- | --- |
| **х**(1) = **αх**(0) + **β**. |  |

Оскільки **х**(0) не є рішенням системи, в лівій частині (31) знайдеться деякий стовпець х(1), в загальному випадку відмінний від **х**(0). Отриманий стовпець **х**(1) розглядатимемо в якості наступного (першого) наближення до розв’язку. Аналогічно, по відомому *k*-му наближенню можна знайти (*k* + 1)-е наближення:

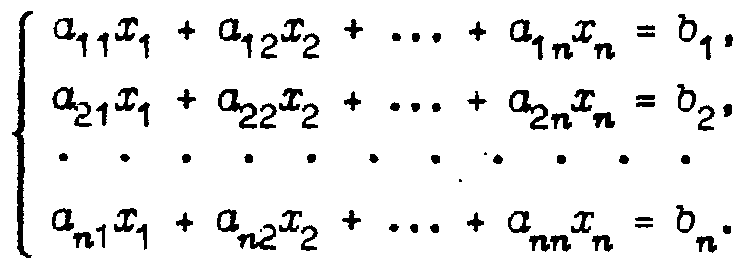
|  |  |
| --- | --- |
| **х**(*k*+ 1) = **αх**(*k*) + **β**. *k* = 0,1,2,… | (32) |

Формула (32) і виражає собою метод простої ітерації. Для її застосування треба задати невизначений доки параметр *τ*. Від значення *τ* залежить, чи сходитиметься метод, а якщо буде, то яка буде швидкість збіжності, тобто як багато ітерацій треба виконати для досягнення необхідної точності. Зокрема, справедлива наступна теорема.

Теорема. *Нехай* det **А** ≠ 0. *Метод простої ітерації* (32) *сходиться тоді і тільки тоді, коли усі власні числа матриці* **α** = **E** – *τ***A** *по модулю менше одиниці.*

Для деяких типів матриці **А** можна вказати правило вибору *τ*, що забезпечує збіжність методу і оптимальну швидкість збіжності. У найпростішому ж випадку *τ* можна покласти рівним деякому постійному числу, наприклад, 1, 0.1 і т. д.

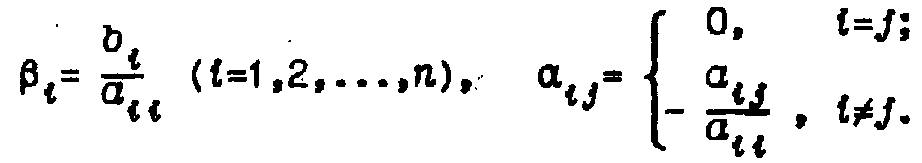
Ще один спосіб *приведення системи* (1)



до виду (31).

Якщо діагональні коефіцієнти *aii*≠ 0 (*i*= 1,2,…*n*), то систему (1) можна представити в такому приведеному виді:

|  |  |
| --- | --- |
|  | (6.12) |

де 

Достатньою умовою збіжності ітерацій в даному випадку є:

ǁ**α**ǁ < 1

де ǁ**α**ǁ — будь-яка норма матриці **α**.

Приклад

|  |
| --- |
| Знайти розв’язок системи рівнянь  методом ітерацій з точністю 10–2  *Розв’язання*  Приведемо задану систему до виду (6.12)    Тепер запишемо послідовність ітерацій    Для приведеної матриці    достатня умова збіжності виконується по *m-*нормі, оскільки    В якості початкового наближення візьмемо вектор-стовпець вільних членів приведеної матриці    Знайдемо перше наближення      Похибку результату можна оцінити використовуючи нерівність      Таким чином **x**(1) дає значення кореня з похибкою, яка не перевищує величини    Далі послідовно знаходимо    Відмітимо, що точний розв’язок системи *x*1 = 1, *x*2 = –1, *x*3 = 1. |

### Метод Гауса-Зейделя

Одним з найпоширеніших ітераційних методів, що відрізняється простотою і легкістю програмування, є метод Гауса-Зейделя.

Проілюструємо спочатку цей метод на прикладі рішення системи:

|  |  |
| --- | --- |
|  | (33) |

Припустимо, що діагональні елементи *а*11, *а*22, *а*33 відмінні від нуля (інакше можна переставити рівняння). Виразимо невідомі *x*1, *x*2 і *х*3 відповідно з першого, другого і третього рівнянь системи (33):

|  |  |
| --- | --- |
|  | (34) |
|  | (35) |
|  | (36) |

Задамо деякі початкові (нульові) наближення значень невідомих: *х*1 = *х*1(0), *х*2 = *х*2(0), *х*3 = *х*3(0). Підставляючи ці значення в праву частину виразу (34), отримуємо нове (перше) наближення для *x*1:

|  |  |
| --- | --- |
|  |  |

Використовуючи це значення для *x*1 і наближення *х*3(0) для *х*3, знаходимо з (35) перше наближення для *x*2:

|  |  |
| --- | --- |
|  |  |

І нарешті, використовуючи вичислені значення *х*1 = *х*1(1), *х*2 = *х*2(1) находимо з допомогою виразу (36) перше наближення для *х*3:

|  |  |
| --- | --- |
|  |  |

На цьому закінчується перша ітерація розв’язання системи (34) - (36). Використовуючи тепер значення *х*1(1), *х*2(1), *х*3(1) можна таким же способом провести другу ітерацію, в результаті якої будуть знайдені другі наближення до розв’язку: *х*1(2), *х*2(2), *х*3(2) і т. д.

Наближення з номером *k* можна вичислити, знаючи наближення з номером *k* – 1, як:

|  |  |
| --- | --- |
|  |  |

Ітераційний процес триває до тих пір, поки значення *х*1(*k*), *х*2(*k*), *х*3(*k*) не стануть близькими із заданою похибкою до значень *х*1(*k*– 1), *х*2(*k*– 1), *х*3(*k*– 1).

Розглянемо тепер систему *n* лінійних рівнянь з *n* невідомими. Запишемо її у виді:

|  |  |
| --- | --- |
|  |  |

Тут також будемо, припускати, що усі діагональні елементи відмінні від нуля. Тоді відповідно до методу Гауса-Зейделя *k*-е наближення до рішення можна представити у виді

|  |  |
| --- | --- |
|  | (37) |

Ітераційний процес триває до тих пір, поки усі значення *хi*(*k*) не стануть близькими *хi*(*k*– 1), тобто як критерій завершення ітерації використовується одна з умов (27) – (29), (30).

Для збіжності ітераційного процесу (37) достатньо, щоб модулі діагональних коефіцієнтів для кожного рівняння системи були не менше сум модулів усіх інших коефіцієнтів (переважання діагональних елементів):

|  |  |
| --- | --- |
|  | (38) |

При цьому хоч би для одного рівняння нерівність повинна виконуватися строго. Ці умови є достатніми для збіжності методу, але вони не є необхідними, тобто для деяких систем ітерації сходяться і при порушенні умов (38).

Приклад

|  |
| --- |
| Знайти розв’язок системи рівнянь  методом Гауса-Зейделя  *Розв’язання*  Виразимо невідомі *x*1, *x*2 і *х*3 відповідно з першого, другого і третього рівнянь системи  Взявши нульове наближення *x*(0) = (0, 0, 0), знайдемо перше:  Аналогічно знаходяться й наступні наближення:  Складові вектора **x**(2) = (71/72, 71/72, 71/72) близькі до **x** = (1, 1, 1). |

|  |  |
| --- | --- |
|  | () |

# Джерела

1. Турчак Л. И., Плотников П. В. Основы численных методов: Учебное пособие. — 2-е изд., перераб. и доп. — М.: ФИЗМАТЛИТ, 2003. — 304 с.

2. Фельдман Л.П., Петренко А.І., Дмитрієва О.А. Підручник Чисельні методи в інформатиці. — К.: Видавнича група BHV, 2006. —480 с.

3. Амосов А.А., Дубинский Ю.А., Копченова Н. В. Вычислительные методы для инженеров: Учеб. пособие. — М.: Высш. шк., 1994. — 544 с.

4. Ракитин В.И., Первушин В.Е. Практическое руководство по методам вычислений с приложением программ для персональных компьютеров: Учеб. пособие. — М.: Высш. шк., 1998. —383 с.

5. Мусіяка В.Г. Основи чисельних методів механіки: Підручник. — К.: Вища освіта, 2004.— 240 с.