

УДК 519.612

М.Ю. Савченко

Інститут проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України, Київ

## АНАЛІЗ ШВИДКОСТІ ЗБІЖНОСТІ АСИНХРОННИХ ІТЕРАЦІЙНИХ МЕТОДІВ РІШЕННЯ СИСТЕМ ЛІНІЙНИХ РІВНЯНЬ

У даній статті представлений клас ітераційних методів рішення систем лінійних алгебраїчних рівнянь. Дані умови збіжності цього класу методів. Представлений критерій оцінки швидкості збіжності різних ітераційних методів. Наведена класифікація паралельних ітераційних алгоритмів, показані переваги асинхронних ітераційних методів, їхні види, а також критерій для визначення збіжності.

**Ключові слова:** ітераційні методи, паралельні програми, асинхронні ітераційні методи, детерміновані алгоритми.

### Вступ

Рішення систем лінійних алгебраїчних рівнянь (СЛАР) є одною з найбільш розповсюджених і важливих задач обчислювальної математики. Як відомо, існують різноманітні способи розв'язання СЛАР, зокрема їх ділять на прямі та ітераційні методи. Використання перших на практиці показує, що при комп'ютерних обчисленнях вони часто виконуються з похибками, що спричиняє похибки і в фінальних результатах. Ітераційні ж методи базуються на початкових наближеннях (наближених розв'язках), які встановлюються перед початком обчислень, і не накопичують похибку в процесі рішення, що особливо актуально для великих систем, коли різко збільшується загальна кількість операцій.

### Методи розв'язання СЛАР

Розглянемо СЛАР виду

$$Ax = b, x \in R^n, \quad (1)$$

де  $A = (A_{i,j})_{1 \leq i, j \leq n}$  – це квадратна неодиначна матриця,  $b$  – вектор вигляду  $b = (b_1, \dots, b_n)^T$ .  $x = A^{-1}b$  є точним розв'язком системи (1). Дуже часто цей точний розв'язок неможливо отримати через різні типи похибок, таких як похибки округлення результатів.

Як було сказано вище, для рішення системи (1) існує два класи методів: прямі та ітераційні. Прямі методи доходять до розв'язку після ряду скінченного числа елементарних операцій. Точний розв'язок теоретично отримується, якщо ми впевнені у тому, що під час обчислень не було похибок округлення. В прямих методах число елементарних операцій може бути наперед відомо, це число не залежить від точності розв'язку.

Під час виконання ітераційних методів відбувається послідовне наближення розв'язку.

Розв'язком є границя ряду  $\{x^{(k)}\}_{k \in N}$ :

$$\lim_{k \rightarrow \infty} x^{(k)} = A^{-1}b. \quad (2)$$

Ітераційний процес завершується, коли отримується задана точність розв'язку.

Згідно з [1] ітераційні методи мають ряд переваг. Вони вимагають зберігання в пам'яті не всієї матриці, а лише декількох векторів з  $n$  компонентами, причому іноді елементи матриці можна не зберігати, а обчислювати при потребі. Також похибки фінальних результатів не накопичуються, так як точність обчислень, як правило, залежить лише від попередньої ітерації. Ітераційні методи рішення систем рівнянь зручні для реалізації на багатопроекторних обчислювальних системах, відносно легко розпаралелюються на рівні груп компонентів вектора невідомих.

Проте ітераційні методи не позбавлені недоліків, серед яких: алгоритми рішення лінійних рівнянь за допомогою ітераційних методів, як правило, складніші за прямі методи; складно попередньо визначити об'єм обчислень; швидкість збіжності ітерацій може бути дуже повільною; при використанні паралельних версій вимагається синхронізація для взаємозв'язку між процесами і гарантування коректності їх виконання.

Лінійні ітераційні методи можуть бути виражені у формі

$$x^{(k+1)} = Tx^{(k)} + c \quad (3)$$

з відомим початковим наближенням  $x^{(0)}$ . Методи Якобі, Гауса-Зейделя, релаксації та Річардсона є прикладами лінійних ітераційних методів.

Якщо відображення ітераційної матриці  $T$  не залежить від поточної ітерації  $k$ , то метод називають стаціонарним. В протилежному випадку – нестаціонарним.

### Збіжність ітераційних методів

Згідно з [2] лінійний ітераційний метод збігається (досягає розв'язку СЛАР (1)), якщо при будь-якому  $x^{(0)} \in R^n$  виконується умова  $\lim_{k \rightarrow +\infty} x^{(k)} = A^{-1}b$ . В цьому випадку спектральний радіус

$$\rho(T) < 1, \tag{4}$$

а також будь-яка норма матриці  $\| \cdot \|$  є такою, що  $\|T\| < 1$ . При цьому  $x^*$  є розв'язком системи (1).

Розглянемо швидкість збіжності лінійних ітераційних методів. Нехай дано збіжний ітераційний метод, ітераційна матриця  $T$  якого задовольняє вираз (4). Отже,  $\lim_{k \rightarrow +\infty} x^{(k)} = x^*$ . Нехай  $\varepsilon^{(k)}$  – вектор похибок на ітерації  $k$ :

$$\varepsilon^{(k)} = x^{(k)} - x^*, \tag{5}$$

тоді маємо

$$\varepsilon^{(k)} = T\varepsilon^{(k-1)} = T^k\varepsilon^{(0)}. \tag{6}$$

Виберемо таке  $\varepsilon$ , що  $\rho(T) + \varepsilon < 1$ . Тоді

$$\forall k \geq K, \|\varepsilon^{(k)}\| \leq (\rho(T) + \varepsilon)^k \|\varepsilon^{(0)}\|. \tag{7}$$

Отже, швидкість збіжності лінійного ітераційного методу з ітераційною матрицею  $T$  визначається спектральним радіусом від  $T$ . Чим менше  $\rho(T)$ , тим швидше збігається метод.

Розглянемо методи Якобі, Гауса-Зейделя та релаксації (SOR). Їх покомпонентний вигляд буде, відповідно

$$x_i^{(k+1)} = \left( b_i - \sum_{j \neq i} A_{i,j} x_j^{(k)} \right) / A_{i,i}, \tag{8}$$

$$x_i^{(k+1)} = \left( b_i - \sum_{j < i} A_{i,j} x_j^{(k+1)} - \sum_{j > i} A_{i,j} x_j^{(k)} \right) / A_{i,i}, \tag{9}$$

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \frac{\omega \left( b_i - \sum_{j < i} A_{i,j} x_j^{(k+1)} - \sum_{j > i} A_{i,j} x_j^{(k)} \right)}{A_{i,i}}, \tag{10}$$

де  $\omega$  – параметр релаксації.

Неважко пересвідчитись, що спектральний радіус ітераційної матриці для методу Гауса-Зейделя є меншим, отже, швидкість збіжності для нього є вищою за метод Якобі.

### Блочні ітераційні методи

Всі три розглянуті вище методи працюють над пошуком розв'язку лише покомпонентно. Існують також так звані блочні версії цих методів, у яких структура алгоритмів така ж, проте обчислення і реалізація проводяться іншим чином. Для цих методів матриця  $A$ , вектори  $x$  та  $b$  діляться на блоки:

$$A = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \dots & \dots & \dots & \dots \\ A_{n1} & A_{n2} & \dots & A_{nn} \end{pmatrix}, \tag{11}$$

$$x = \begin{pmatrix} X_1 \\ X_2 \\ \dots \\ X_n \end{pmatrix}, b = \begin{pmatrix} B_1 \\ B_2 \\ \dots \\ B_n \end{pmatrix}.$$

Перевагою блочного методу Якобі є те, що кількість ітерацій значно зменшується, проте він вимагає рішення кількох лінійних підсистем, що не завжди є простою задачею. Реалізація блочних версій методів Гауса-Зейделя та SOR просто вимагає використання останніх версій компонентів попередніх блоків і попередніх версій наступних блоків (як і для покомпонентних версій). Метод SOR також вимагає використання параметру релаксації  $\omega$ .

### Асинхронні ітераційні методи

Ітераційні методи, зокрема їх блочні версії, можуть бути розпаралелені для виконання на розподілених обчислювальних системах. Цим, як правило, досягається істотне збільшення їх ефективності.

Виділяють три класи паралельних ітераційних алгоритмів:

1. Синхронні ітерації – синхронні обміни даними (CICO).
2. Синхронні ітерації – асинхронні обміни даними (CIAO).
3. Асинхронні ітерації – асинхронні обміни даними (AIAO).

На рис. 1 – 3 наведені схеми виконання програм для цих трьох класів на двопроцесорній ЕОМ.

Як бачимо, клас AIAO дозволяє уникнути простоявання процесорів, завдяки цьому істотно зменшується час виконання програм.

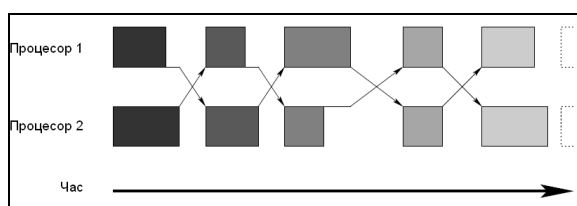


Рис. 1. Синхронні ітерації – синхронні обміни даними (CICO)

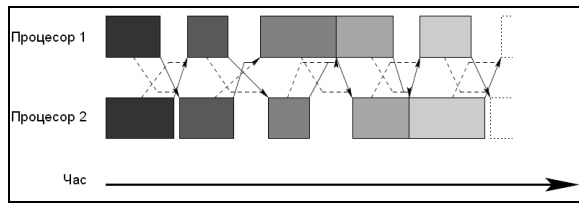


Рис. 2. Синхронні ітерації – асинхронні обміни даними (СІАО)

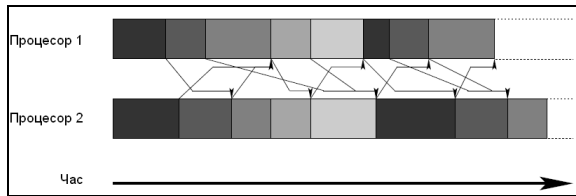


Рис. 3. Асинхронні ітерації – асинхронні обміни даними (АІАО)

Розглянемо паралельний синхронний блочний алгоритм Якобі. Його відносно просто реалізувати, беручи за основу послідовну версію. В ньому кожен процесор відповідає за обчислення свого блоку, і після чергової ітерації всі процесори надсилають їхній локальний розв'язок «сусідам», що його потребують.

Паралельний асинхронний алгоритм Якобі у порівнянні з його послідовною версією має лише дві групи відмінностей: керування обмінами даних та визначення збіжності. В асинхронній версії процесор бере останню версію даних, отриманих від «сусідів», якщо нові дані були отримані під час попередньої ітерації. Для збільшення ефективності в таких алгоритмах впроваджується перевірка «Чи прийшли нові дані?», то ж немає необхідності чекати завершення обчислень на всіх інших процесорах. Як було наведено вище, також необхідно визначати досягнення збіжності алгоритмом. Згідно з [3] асинхронний ітераційний метод збігається до  $x^*$  тоді, коли зважена максимальна норма для всіх  $x, y \in E$ :

$$\begin{aligned} & \|T(x, y) - x^*\|_{\gamma, \infty} \leq \\ & \leq \gamma \max(\|x - x^*\|_{\gamma, \infty}, \|y - x^*\|_{\gamma, \infty}) \end{aligned} \quad (12)$$

та існує  $\gamma \in [0, 1]$  й  $x^* \in E$  таке, що  $T(x^*, x^*) = x^*$ .

Згідно з [4] перевагами асинхронних ітераційних алгоритмів є, зокрема:

- Усунення ефектів «вузьких місць» при обміні даними між процесорами. Якщо між процесорами  $G$  та  $H$  відбулось зниження пропускної здатності каналу зв'язку, то в асинхронній версії алгоритму інші процесори, обчислення яких на пряму не залежать від результатів з  $G$  та  $H$ , не будуть чекати на дані від них.

- Зменшується витрата часу на синхронізацію. Процесор може рахувати далі, не чекаючи на завершення тої ж ітерації на інших процесорах.

- Такі алгоритми добре пристосовані до обчислювальних систем, для яких синхронізація є надзвичайно небажаною.

- У випадках, якщо частина даних була втрачена, ці алгоритми легко перезавантажити без значних витрат у часі, так як для кожен процесор працює зі своїм зразком даних.

- Показники збіжності покращуються завдяки слідуванню принципу Гауса.

## Детерміновані асинхронні обчислення

Потрібно відрізнити детерміновані та недетерміновані асинхронні обчислення. При детермінованих – завчасно відомий алгоритм, по якому відбуваються наближення компонент вектора невідомих. Проробивши двічі одну й ту ж послідовність обчислень, прийдемо до однакового результату. У випадку недетермінованих обчислень вказати завчасно алгоритм, по якому будуть відбуватися послідовні наближення, неможливо.

Детермінований асинхронний процес може бути стаціонарним та нестаціонарним. У випадку першого – ітераційна матриця, за допомогою якої формуються компоненти вектора невідомих, є незмінною в часі, для нестаціонарних – змінюється від ітерації до ітерації.

Детерміновані обчислення можуть бути реалізовані в одно- та багатопроцесорних системах. В однопроцесорному виконанні можлива реалізація завчасно розроблених алгоритмів, що можуть бути організовані з метою забезпечення збіжності, підвищення швидкості збіжності обчислень і т.д. Згідно з [5], в багатопроцесорних системах детерміновані обчислення можуть бути т.зв. природними та штучними. Перші організуються в результаті різного обчислювального навантаження, яка вимагається від кожного з процесорів (для реальних практичних задач виділити однакові об'єми обчислень для кожного процесора практично неможливо). При цьому кожному процесору виділяється співвідносне обчислювальне навантаження. Другі виникають у зв'язку з різним об'ємом навантаження, що свідомо виділяється кожному процесору.

Отже, асинхронним методом  $i$ -го порядку будемо називати такі обчислення, при яких максимальне здійснене наближення хоча б однієї компоненти вектора невідомих на кожній з асинхронних ітерацій дорівнює  $i+1$ . Порядок асинхронізму обчислювального методу визначається числом  $i+1$ . Для прикладу, асинхронний метод нульового порядку є звичайний метод, при якому за одну ітерацію число наближень

кожної з компонент вектора невідомих збільшується на одиницю.

При асинхронному методі першого порядку за одну ітерацію число наближень компонент збільшується на одиницю чи двійку.

Нехай задане СЛАР виду (1),  $\det A \neq 0$ . Для рішення (1) побудуємо ітераційний алгоритм

$$Vx^{K+1} = Vx^K - \mu(Ax^K - b), \quad (13)$$

що перетворимо до вигляду

$$x^{K+1} = x^K - \mu V^{-1} Ax^K + \mu V^{-1} b = \alpha x^K + f, \quad (14)$$

де  $\alpha = E - \mu V^{-1} A$ ,  $f = \mu V^{-1} b$ ,  $K$  – номер поточної ітерації.

Вибором  $\mu$  та  $V$  можна побудувати багато різних ітераційних алгоритмів.

Нехай матриця  $b$  така, що ітераційний метод (14) певним чином оптимізується (наприклад, по швидкості збіжності).

З метою підвищення продуктивності обчислювальної системи можлива організація асинхронних обчислень. Але це призведе до відхилення обчислень від алгоритму (14).

Збільшення продуктивності обчислень не має погіршувати властивості алгоритмів, бо час рішення задачі може збільшитись, не дивлячись на збільшення продуктивності обчислювальних засобів.

Що б відповісти на питання доцільності організації таких обчислень, необхідно скласти ітераційну матрицю, що відповідатиме цьому алгоритму, й провести порівняльний аналіз отриманої матриці.

Для аналізу детермінованих асинхронних обчислень можна використовувати ті ж методи, що застосовуються для звичайних ітераційних алгоритмів. Це обумовлюється тим, що на кожній ітерації асинхронних обчислень може бути побудований

оператор переходу. Аналізуючи ці оператори, можемо оптимальним чином ітераційні параметри, що мінімізують норми цих операторів або норми похибки рішення.

## Висновок

В даній статті розглянуті ітераційні методи та їх асинхронні аналоги, представлені їх критерії збіжності та оцінки швидкості збіжності.

Наведені класифікація паралельних ітераційних алгоритмів, переваги асинхронних ітераційних алгоритмів, а також розглянуті детерміновані асинхронні алгоритми.

## Список літератури

1. Турчак Л.И. Основы численных методов: Учебное пособие. – 2-е изд., перераб. и доп. / Л.И. Турчак, П.В. Плотников. – М.: ФИЗМАТЛИТ, 2002. – 304 с.
2. Bahi J.M. Parallel iterative algorithms: from sequential to grid computing / J.M. Bahi, S.C. Vivier, R. Couturier. – Chapman and Hall / CRC, 2008.
3. Chazan D. Chaotic relaxation / D. Chazan, W. Miranker // Linear Algebra Appl., 2: 199–222, 1969.
4. Kung H.T. Synchronized and asynchronous algorithms for multiprocessors / H.T. Kung // In J. F. Traub Ed., Algorithm and Complexity: New Directions and Recent Results, New York, 1976. Academic Press.
5. Белецкий В.Н. Многопроцессорные и параллельные структуры с организацией асинхронных вычислений / В.Н. Белецкий. – К.: Наук. думка, 1988. – 239 с.

Надійшла до редколегії 16.06.2014

**Рецензент:** канд. техн. наук, старш. наук співр. О.А. Чемерис, Інститут проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України, Київ.

## АНАЛИЗ СКОРОСТИ СХОДИМОСТИ АСИНХРОННЫХ ИТЕРАЦИОННЫХ МЕТОДОВ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ

М.Ю. Савченко

*В этой статье представлен класс итерационных методов решения систем линейных алгебраических уравнений. Даны условия сходимости этого класса методов. Представлен критерий оценки скорости сходимости разных итерационных методов. Приведена классификация параллельных итерационных алгоритмов, показаны преимущества асинхронных итерационных методов, их виды, а также критерий для нахождения сходимости.*

**Ключевые слова:** итерационные методы, параллельные программы, асинхронные итерационные методы, детерминированные алгоритмы.

## CONVERGENCE SPEED ANALYSIS OF ASYNCHRONOUS ITERATIVE METHODS FOR SOLVING LINEAR SYSTEMS

M.Yu. Savchenko

*This paper gives a comparison of iterative methods for solving linear systems of equations. The convergence conditions for these methods are proposed. The criterion for measuring convergence speed is offered. This paper also gives a classification list for the parallel iterative algorithms, their advantages and the way to detect the convergence of asynchronous iterative algorithms.*

**Keywords:** iterative methods, parallel programs, asynchronous iterative methods, deterministic algorithms.