

ДОСЛІДЖЕННЯ ІТЕРАЦІЙНИХ ПАРАЛЕЛЬНИХ АЛГОРИТМІВ РОЗВ'ЯЗАННЯ РОЗРІДЖЕНИХ ЛІНІЙНИХ СИСТЕМ РІВНЯНЬ

Вінницький національний технічний університет

Анотація

Розглянуто питання розробки та аналізу ітераційних алгоритмів для розв'язання великих розріджених систем лінійних алгебраїчних рівнянь (СЛАР), включаючи паралельні методи обчислення. У роботі досліджено властивості розріджених матриць та підходи до організації ітераційних процесів Якобі, Гаусса–Зейделя та релаксації SOR. Створено програмну реалізацію на мові Python з використанням бібліотек SciPy та multiprocessing для прискорення обчислень на багатоядерних процесорах. Проведено експериментальне дослідження ефективності алгоритмів, що підтвердило підвищення продуктивності при використанні паралельних обчислень для великих систем.

Ключові слова: розріджені матриці, метод Якобі, метод Гаусса–Зейделя, метод релаксації SOR, паралельні обчислення, ітераційні методи.

Abstract

The article considers the development and analysis of iterative algorithms for solving large sparse systems of linear algebraic equations (SLAE), including parallel computing methods. The properties of sparse matrices and approaches to organizing the iterative processes of Jacobi, Gauss-Seidel, and Successive Over-Relaxation (SOR) are investigated. A software implementation in Python using SciPy and multiprocessing libraries was created to accelerate calculations on multi-core processors. An experimental study of the efficiency of the algorithms was conducted, confirming the performance increase when using parallel computing for large-scale systems.

Keywords: sparse matrices, Jacobi method, Gauss-Seidel method, SOR relaxation method, parallel computing, iterative methods.

Вступ

Актуальність дослідження ітераційних паралельних алгоритмів для розв'язування розріджених лінійних систем рівнянь зумовлена стрімким розвитком обчислювальної техніки та зростанням складності задач у сучасній науці та інженерії [1-5]. На сьогоднішній день математичне моделювання складних фізичних процесів, проектування інженерних споруд, задачі обробки великих даних (Big Data) та аналізу складних мереж вимагають розв'язання систем лінійних алгебраїчних рівнянь (СЛАР) надвеликої розмірності, де кількість невідомих може сягати мільйонів. Характерною особливістю таких систем є їхня розрідженість — стан, при якому переважна більшість елементів матриці є нульовими.

Традиційні прямі методи розв'язання, такі як метод виключення Гаусса, стають неефективними для подібних систем через квадратичну або кубічну складність та значні витрати пам'яті, оскільки в процесі обчислень розріджена структура матриці зазвичай втрачається. У зв'язку з цим, ітераційні методи, такі як методи Якобі, Гаусса–Зейделя та метод верхньої релаксації (SOR), стають основним інструментом обчислень. Вони дозволяють працювати безпосередньо з ненульовими елементами матриці, зберігаючи її компактну структуру та забезпечуючи поступове наближення до розв'язку з заданою точністю.

Метою роботи є дослідження та програмна реалізація методів Якобі, Гаусса–Зейделя та SOR, а також аналіз можливостей їх прискорення шляхом паралелізації.

Постановка задачі дослідження

Задачі дослідження включають:

- аналіз предметної області розріджених систем та ітераційних методів;
- математичне моделювання ітераційних алгоритмів та обґрунтування вибору засобів розробки;
- побудову UML-діаграм класів та активності алгоритмів;
- програмну реалізацію оптимізованого паралельного алгоритму та аналіз результатів тестування.

Виклад основного матеріалу

Аналіз розріджених матриць та методів зберігання. Розрідженою вважається матриця, у якій кількість ненульових елементів не перевищує 5–10% від загальної кількості. Використання традиційних (щільних) форматів зберігання для таких систем призводить до неефективного використання пам'яті та обчислювальних ресурсів. Для оптимізації застосовуються спеціальні формати, такі як CSR (Compressed Sparse Row), що зберігають лише ненульові елементи [4, 6].

Ітераційні алгоритми. Основними дослідженими методами є:

- метод Якобі, у якому кожне нове значення обчислюється незалежно на основі попередньої ітерації, що робить його ідеальним для паралелізації;
- метод Гаусса–Зейделя оновлює значення відразу після обчислення, що забезпечує швидшу збіжність, але ускладнює паралелізацію через послідовні залежності;
- метод SOR (Successive Over-Relaxation), який є покращеною версією метода Гаусса–Зейделя з параметром релаксації ω для прискорення процесу збіжності.

Паралелізація обчислень. Для паралельної реалізації обрано багатопроцесорну модель (Shared Memory) з використанням модуля multiprocessing у Python. Найкраще для паралелізації підходить метод Якобі, де рядки матриці розподіляються між окремими обчислювальними потоками. Кожна ітерація включає незалежне обчислення компонентів вектора розв'язку з наступною синхронізацією результатів.

Програмна реалізація та UML-моделювання. Система реалізована у процедурному стилі з використанням бібліотек NumPy та SciPy. Структура модулів включає matrix_utils для генерації даних, iterative_methods для основних розрахунків та parallel_solver для паралельної версії Якобі. Процес виконання ітераційного циклу з перевіркою умови збіжності $\|x(k+1) - x(k)\| < \varepsilon$ відображено на побудованій діаграмі активності [7].

Аналіз результатів тестування

Тестування проведено на розрідженій матриці розміром $n = 1500$ із щільністю 0.003 (табл.1).

Таблиця 1 – Результатів роботи алгоритмів

| Метод | Ітерацій | Час виконання, с | Нев'язка | Відносна похибка |
|---------------------|----------|------------------|-----------|------------------|
| Якобі (послідовний) | 20 | 0.0012 | 9.994e-06 | 5.007e-08 |
| Якобі (паралельний) | 20 | 18.6183 | 9.994e-06 | 5.007e-08 |
| Гаусса–Зейделя | 14 | 0.1936 | 0.997e-06 | 0.542e-08 |
| SOR | 20 | 0.2634 | 4.102e-06 | 1.915e-08 |

Аналіз результатів показав:

- метод Гаусса–Зейделя продемонстрував найкращу збіжність (лише 14 ітерацій) та найвищу точність;
- паралельний метод Якобі на невеликих розмірностях виявився повільнішим за послідовний через високі накладні витрати на створення процесів та обмін даними;
- ефективність паралелізації стає помітною лише при значному зростанні розмірності системи, коли обсяг обчислень перевищує час на синхронізацію.

Висновки

У роботі досліджено та реалізовано ітераційні методи розв'язання розріджених СЛАР. Встановлено, що для діагонально домінуючих систем найбільш ефективним за кількістю ітерацій є метод Гаусса–Зейделя. Паралелізація за допомогою модуля multiprocessing є доцільною для задач великої розмірності, оскільки на малих об'ємах даних домінують накладні витрати на організацію паралельного виконання. Розроблене програмне забезпечення може бути використане як основа для побудови складних обчислювальних систем в інженерних та наукових задачах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Коцовський В. М. Теорія паралельних обчислень: навч. посібник. Ужгород: ПП «АУТДОР-Шарк», 2021. 188 с. URL: <https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/38994/1/%D0%9D%D0%B0%D0%B2%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9%20%D0%BF%D0%BE%D1%81%D1%96%D0%B1%D0%BD%D0%B8%D0%BA.pdf> .
2. Багатозадачність. URL: <https://uk.wikipedia.org/wiki/Багатозадачність>.
3. Семеренко, В. П. Технології паралельних обчислень : навч. посіб. Вінниця: ВНТУ, 2018. – 104 с. URL: https://www.researchgate.net/publication/334710599_V_P_Semerenco_TEHNOLOGII_PARALELNIH_OBCISLEN_Ministerstvo_osviti_i_nauki_Ukraini_Vinnickij_nacionalnij_tehnicnij_universitet
4. Parallel Scaling Guide. URL: https://rc-docs.mines.edu/pages/user_guides/Parallel_Scaling_Guide.html?utm_source=chatgpt.com .
5. Технології розподілених систем та паралельних обчислень: конспект лекцій: для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 122 Комп'ютерні науки/ Київ. нац. ун-т буд-ва і архіт. Київ: КНУБА, 2024. 98 с. URL: <https://repository.knuba.edu.ua/server/api/core/bitstreams/9b73ce1f-b611-4baf-bf35-048210ecc627/content>
6. Пропускна здатність. URL: https://uk.wikipedia.org/wiki/Пропускна_здатність.
7. The Unified Modeling Language. URL: <https://www.uml-diagrams.org/>.

Князький Денис Віталійович – студент кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: denisknyazkyi@gmail.com;

Денисюк Валерій Олександрович – канд. техн. наук, доцент, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, м.Вінниця, e-mail: vad64@i.ua.

Knyazkyi Denys Vitaliyovych – student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: denisknyazkyi@gmail.com;

Denysiuk Valerii Olexandrovich – Ph.D., Assistant Professor, Assistant Professor of the Chair of Computer Science, Vinnytsia National Technical University, Vinnytsia, e-mail: vad64@i.ua