

ДОСЛІДЖЕННЯ ПАРАЛЕЛЬНИХ МЕТОДІВ РОЗВ'ЯЗАННЯ СИСТЕМ НЕЛІНІЙНИХ РІВНЯНЬ

Вінницький національний технічний університет

Анотація

Розглянуто розробку програмного комплексу для чисельного розв'язання систем нелінійних рівнянь великої розмірності. Розглянуто питання аналізу предметної області, здійснено математичне моделювання, синтезовано потоковий граф алгоритму, розроблено діаграми компонентів та діяльності програмного забезпечення. Обґрунтовано вибір мови C++ та технології OpenMP, створено програмну реалізацію паралельного методу Ньютона та проведено тестування ефективності. Програмна реалізація дозволяє суттєво зменшити час виконання обчислень для систем великої розмірності, забезпечуючи високу точність результатів. Одержані результати можливо використовувати в інженерному моделюванні та наукових розрахунках для прискорення розв'язання складних нелінійних задач.

Ключові слова: системи нелінійних рівнянь, паралельні обчислення, метод Ньютона, MPI, OpenMP, масштабованість, прискорення.

Abstract

The development of a software complex for the numerical solution of large-scale systems of nonlinear equations is considered. The analysis of the subject area is considered, mathematical modeling is carried out, the flow graph of the algorithm is synthesized, diagrams of components and software activities are developed. The choice of C++ language and OpenMP technology is justified, a software implementation of the parallel Newton's method is created, and efficiency testing is carried out. The software implementation allows for significantly reducing the computation time for large-scale systems while ensuring high accuracy of results. The results obtained can be used in engineering modeling and scientific calculations to accelerate the solution of complex nonlinear problems.

Keywords: systems of nonlinear equations, parallel computing, Newton's method, MPI, OpenMP, scalability, speedup.

Вступ

Актуальність теми зумовлена необхідністю підвищення ефективності розв'язання складних систем нелінійних рівнянь, які вимагають значних обчислювальних ресурсів та часу при використанні традиційних послідовних методів. В умовах стрімкого поширення багатоядерних архітектур та обчислювальних кластерів, адаптація класичних алгоритмів (таких як метод Ньютона чи ітераційні методи) до паралельного виконання дозволяє суттєво прискорити отримання результатів. Дослідження методів розпаралелювання є критично важливим для оптимізації роботи з великими масивами даних, забезпечення масштабованості програмних рішень та мінімізації витрат часу на моделювання високотехнологічних процесів.

Метою роботи є підвищення швидкодії та ефективності розв'язання систем нелінійних рівнянь шляхом розробки, програмної реалізації та аналізу паралельних алгоритмів.

Постановка задачі дослідження

Задачі дослідження полягають у вирішенні наступних питань:

- проаналізувати існуючі чисельні методи розв'язання СНР;
- розробити алгоритми паралельного виконання для методу Ньютона;
- виконати програмну реалізацію алгоритмів з використанням технологій паралельного програмування;

- провести порівняльний аналіз продуктивності послідовних та паралельних методів, оцінити їх масштабованість та прискорення на практичних прикладах.

Виклад основного матеріалу

Системи нелінійних рівнянь є невід'ємною частиною математичного моделювання. Класичний метод Ньютона забезпечує квадратичну збіжність, проте вимагає перерахунку матриці Якобі та розв'язання системи лінійних алгебраїчних рівнянь на кожній ітерації [1 – 3].

Для систем великої розмірності ($n \geq 1000$) ці операції стають вузьким місцем. Основна мета застосування технологій паралелізму в даній роботі – декомпозиція цих трудомістких етапів. Сучасні засоби розробки, зокрема мова C++ та бібліотека OpenMP, надають ефективний інструментарій для організації паралелізму в системах зі спільною пам'яттю. Це дозволяє уникнути складного керування потоками вручну та забезпечити переносимість коду [4, 5].

У роботі розроблено паралельну програмну систему, яка реалізує метод Ньютона. Алгоритм базується на ітераційній формулі:

$$J(x^{(k)})\Delta x^{(k)} = -F(x^{(k)}), \quad (1.1)$$

де J – матриця Якобі, яку необхідно обчислити та обернути (через розв'язання СЛАР) на кожному кроці.

Математичне моделювання та синтез потокового графу дозволили виділити основні рівні паралелізму:

- обчислення Якобіана, кожен рядок матриці обчислюється незалежно, використано декомпозицію за даними (Data Decomposition), де потоки ділять між собою діапазон індексів рядків;
- розв'язання СЛАР, реалізовано паралельний метод Гаусса; на етапі прямого ходу операції виключення змінних для рядків під головним елементом виконуються одночасно різними потоками;
- векторні операції, оновлення наближення та перевірка критерію збіжності також розпаралелені.

Для підвищення швидкодії програмної реалізації використано технологію OpenMP.

Оптимізація досягалася за рахунок використання директиви `#pragma omp parallel for` з планувальником `schedule` для балансування навантаження при обчисленні функцій різної складності, мінімізації накладних витрат на створення потоків шляхом використання паралельних регіонів та оптимізації роботи з пам'яттю для покращення локальності даних [3, 6].

Реалізація програми здійснена мовою C++, що забезпечує високий контроль над ресурсами пам'яті, необхідний для роботи з великими матрицями [7].

Для візуалізації архітектури розроблено UML-діаграму компонентів, яка відображає модулі обчислення Якобіана, розв'язувача СЛАР та керування ітераціями, а також UML-діаграму діяльності, що демонструє точки розгалуження та синхронізації потоків [4, 8].

Аналіз результатів тестування програми виконано на системі з 2000 рівнянь при різній кількості потоків (табл. 1).

Результати тестування підтвердили коректність алгоритму та ефективність обраного підходу. Спостерігається суттєве зменшення часу виконання зі збільшенням кількості потоків. На 12-потоковій конфігурації досягнуто скорочення часу виконання з 261 с. до 41 с., що відповідає прискоренню у 6,3 рази. Таким чином, розроблений паралельний метод забезпечує:

- підвищення швидкодії обчислень для задач великої розмірності;
- ефективне використання ресурсів багатоядерних процесорів;
- можливість динамічного масштабування залежно від апаратного забезпечення.

Подальше вдосконалення може передбачати використання блочних алгоритмів для кращого використання кеш-пам'яті або адаптацію під гібридні системи.

Таблиця 1 – Результати вимірювання продуктивності паралельного алгоритму

Кількість потоків (p)	Час виконання (Tp), с	Прискорення ($Sp=T1/Tp$)	Ефективність ($Ep=Sp/p$)
1	261,33	1,00	1,00
2	123,81	2,11	1,06
4	69,45	3,76	0.94
8	42,25	6,19	0.77
12	41,44	6,31	0.53

Висновки

Досліджено та програмно реалізовано паралельний метод Ньютона для розв'язання систем нелінійних рівнянь. Розроблена система демонструє ефективну організацію обчислювального процесу завдяки використанню технології OpenMP у середовищі C++. Створено архітектуру, що включає модулі генерації даних, паралельного розрахунку похідних та лінійного розв'язувача.

Програмно реалізовано поставлену задачу з можливістю динамічного вибору кількості потоків. Використання директив OpenMP забезпечило автоматичний розподіл навантаження. Проведено тестування розробленої програми. Результати експериментів на системі з 2000 змінних підтвердили стабільність роботи та досягнення прискорення у 6,3 рази на 12 потоках. Визначено, що отримане програмне рішення має значний потенціал для застосування в наукових та інженерних розрахунках, де критичним є час отримання результату.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Коцовський В. М. Теорія паралельних обчислень: навчальний посібник. Ужгород: ПП «АУТДОР-Шарк», 2021. 188 с.
2. Багатозадачність. URL: <https://uk.wikipedia.org/wiki/Багатозадачність>.
3. Герасимов В. В., Матвєєва Н. О. Розробка програмного забезпечення на платформі Java. Багатопоточне програмування і паралельні обчислення: навчальний посібник. Дніпро, 2020. 174 с.
4. Anthony Williams. C++ Concurrency in Action. 2nd Edition. Manning Publishing, 2019. 592 p. URL: <https://readingtoday.site/download/c-concurrency-in-action-second-edition>
5. Parallel Scaling Guide. Colorado school of mines. URL: https://rc-docs.mines.edu/pages/user_guides/Parallel_Scaling_Guide.html?utm_source=chatgpt.com.
6. Закон Амдала. URL: https://uk.wikipedia.org/wiki/Закон_Амдала.
7. Пропускна здатність. URL: https://uk.wikipedia.org/wiki/Пропускна_здатність.
8. The Unified Modeling Language. URL: <https://www.uml-diagrams.org/>.

Маркевич Софія Андріївна – студентка кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: marksofand@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.

Markevych Sofiia Andriivna – student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: marksofand@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