УДК 004.8

С. О. Шиманська
В.О. Денисюк

РЕАЛІЗАЦІЯ ПАРАЛЕЛЬНОГО АЛГОРИТМУ РОЗВ'ЯЗАННЯ СИСТЕМИ НЕЛІНІЙНИХ РІВНЯНЬ МЕТОДОМ НЬЮТОНА

Вінницький національний технічний університет

Анотація

Розглянуто розробку паралельного алгоритму розв'язання системи нелінійних рівнянь методом Ньютона з використанням технології MPI. Проведено аналіз існуючих методів чисельного розв'язання нелінійних систем для досягнення високої точності та продуктивності, обґрунтовано вибір засобів програмної реалізації, розроблено UML-діаграми основних компонентів програмного модуля. У роботі створено паралельну програмну реалізацію методу Ньютона з використанням MPI та проведено тестування її швидкодії на різних наборах даних і різній кількості процесів. Застосування отриманих результатів дозволяє підвищити ефективність і масштабованість алгоритмів чисельного аналізу у високопродуктивних обчислювальних системах.

Ключові слова: *метод Ньютона, MPI, паралельні обчислення, нелінійні рівняння, продуктивність.*

Abstract

The development of a parallel Newton's method algorithm for solving systems of nonlinear equations using MPI technology is considered. The study analyzes existing numerical approaches for solving nonlinear systems in order to achieve high performance and accuracy, justifies the choice of software tools, and introduces UML diagrams of the software module. A parallel implementation of Newton's method using MPI is developed and its performance is evaluated on various datasets and different numbers of processes. The results demonstrate the potential for improving computational speed and scalability in high-performance computing environments.

Keywords: *Newton's method, MPI, parallel computing, nonlinear equations, performance.*

Вступ

Актуальність розробки паралельного алгоритму розв'язання системи нелінійних рівнянь методом Ньютона зумовлена швидким зростанням задач, що потребують високої точності та значних обчислювальних ресурсів. Такі задачі широко застосовуються у моделюванні фізичних процесів, оптимізаційних методах, машинному навчанні, аналізі складних технічних і економічних систем.

Метод Ньютона, незважаючи на свою ефективність, вимагає значного обсягу обчислень для знаходження частинних похідних, обчислення та розв'язання системи лінійних рівнянь на кожній ітерації. Використання паралельних обчислювальних архітектур дозволяє розподілити операції між декількома процесами, що суттєво скорочує час виконання і дає можливість працювати з великими системами рівнянь.

Паралельні комунікаційні бібліотеки, зокрема MPI, забезпечують можливість ефективної взаємодії між процесами, що робить їх придатними для реалізації паралельного методу Ньютона. Це є критично важливим у сучасних високопродуктивних обчислювальних системах.

Метою роботи є розробка та дослідження паралельного алгоритму методу Ньютона для розв'язання системи нелінійних рівнянь з використанням MPI з метою підвищення продуктивності та зменшення часу обчислень.

Постановка задачі дослідження

Задачі дослідження полягають у вирішенні наступних питань:

- розробка паралельної архітектури алгоритму методу Ньютона;
- оптимізація обчислення частинних похідних і елементів матриці Якобі;

- реалізація програмного модуля, що виконує паралельні ітерації методу Ньютона;
- забезпечення ефективної взаємодії між MPI-процесами під час формування та розв'язання лінійних підзадач;
- тестування програми та аналіз отриманих результатів.

Виклад основного матеріалу

Системи нелінійних рівнянь широко використовуються в математичному моделюванні, фізиці, хімії, економіці, технічних задачах керування та оптимізації [1–3]. Одним з найефективніших методів їх розв'язання є метод Ньютона, який забезпечує квадратичну збіжність за умови коректного вибору початкового наближення та обчислення матриці Якобі [4].

Класичний метод Ньютона передбачає:

- формування матриці Якобі;
- розв'язання системи лінійних рівнянь на кожній ітерації, $J(xk)\cdot\Delta x = -F(xk)$;
- оновлення наближення, $x(k+1) = x(k) + \Delta x$.

Ключовим обчислювальним навантаженням методу Ньютона є дві операції: формування матриці Якобі та розв'язання відповідної лінійної системи рівнянь на кожній ітерації. Обидві ці процедури мають високу обчислювальну складність і значною мірою визначають загальний час виконання алгоритму. У разі збільшення розмірності задачі або ускладнення функцій, що входять до системи, обсяг необхідних обчислень зростає нелінійно.

Побудова матриці Якобі вимагає обчислення часткових похідних від кожної функції за кожною змінною. Таким чином, для системи з n рівнянь необхідно обчислити $\square^2$ похідних. У випадку, коли функції містять тригонометричні, експоненціальні або комбіновані вирази, значення кожної похідної потребує виконання низки дорогих обчислювальних операцій. Враховуючи, що на кожній ітерації методу Ньютона матриця Якобі обчислюється заново, це значно збільшує загальний час виконання алгоритму, особливо при великій кількості ітерацій або високій точності розв'язку [5].

Другим ресурсомістким етапом є розв'язання системи лінійних рівнянь, де застосовується метод Гаусса або інший чисельний метод. Обчислювальна складність класичного алгоритму Гаусса становить $O(n^3)$, що є серйозним обмеженням при зростанні розмірності задачі. Навіть у випадку середніх розмірів системи ця частина алгоритму може займати домінуючий відсоток часу.

Такі особливості роблять метод Ньютона природним кандидатом для розпаралелювання. Векторні операції при обчисленні елементів матриці Якобі не залежать одна від одної, а операції над рядками під час методу Гаусса також частково можуть виконуватися паралельно. Застосування паралельних технологій дає можливість розподілити ці незалежні обчислення між кількома ядрами процесора, що дозволяє значно зменшити час виконання на одній ітерації.

Технологія OpenMP забезпечує зручний та гнучкий механізм організації паралельних обчислень у межах багатоядерних систем. Вона дозволяє користуватися директивами компілятора для розподілу роботи між потоками без суттєвої зміни структури програмного коду. Найчастіше для прискорення обчислень використовується директива #pragma omp parallel for яка дозволяє розбити цикл на незалежні частини й обчислювати значення елементів матриці Якобі або окремі операції методу Гаусса паралельно.

У результаті використання OpenMP саме найважчі частини алгоритму - обчислення похідних, формування матриці Якобі та операції виключення змінних у методі Гаусса - виконуються з істотним прискоренням. Це дозволяє зменшити час роботи алгоритму при збереженні точності методу Ньютона, роблячи його застосування ефективним навіть для систем великої розмірності.У роботі виконано аналіз підходів паралелізації:

1. Паралелізація обчислення нев'язки F(x). Обчислення значень системи рівнянь розподіляється між процесами, що дозволяє обраховувати частини вектора F(x) незалежно.
2. Паралелізація побудови матриці Якобі. Елементи Якобі можуть формуватися у різних процесах окремо, оскільки частинні похідні не залежать одна від одної.

3. Паралельне розв’язання лінійної системи. Розглянуто два варіанти: передача частин матриці та використання колективних MPI-операцій; розподіл матриці між процесами з наступним паралельним методом Гауса.

4. Взаємодія MPI-процесів. Синхронізація та обмін: розподіл даних, збір проміжних обчислень, колективні операції scatter/gather/broadcast [6-8].

Аналіз результатів тестування програми виконано для різної кількості змінних (табл.1-3).

Таблиця 1 – Час виконання програми на різних вхідних даних

Кількість процесів	Загальна кількість змінних		
	1000	10 000	100 000
2	32.1830	176.721	1429.174
4	26.7219	181.791	1437.933
8	36.1743	196.992	1470.197

Таблиця 2 – Коефіцієнти прискорення

Загальна кількість змінних	1000	10 000	100 000
Коефіцієнт прискорення	0.7861	0.9572	0.9812

Таблиця 3 – Коефіцієнти ефективності

Кількість процесів	2	4	8
Коефіцієнт ефективності	0.4263	0.2331	0.1272

Проаналізувавши отримані коефіцієнти прискорення та коефіцієнти ефективності для різних обсягів вхідних даних і кількості процесів (табл. 1–3), можна зробити такі висновки:

• при малих вхідних даних значення коефіцієнтів прискорення та ефективності залишаються низькими, тому збільшення кількості процесів у цьому випадку практично не впливає на загальний час виконання програми;

• збільшення кількості процесів під час обчислень із великими вхідними даними забезпечує лише незначне підвищення коефіцієнта прискорення, водночас призводячи до суттєвого зменшення коефіцієнта ефективності через зростання накладних витрат паралелізації.

Отже, на зміну таких характеристик, як коефіцієнт прискорення, коефіцієнт ефективності та загальний час виконання алгоритму методу Ньютона, впливають насамперед розмір задачі, кількість задіяних процесів, а також особливості обчислювальних операцій, пов’язаних із формуванням матриці Якобі та розв’язанням відповідної системи лінійних рівнянь. Застосування більш оптимізованих чисельних методів або іншого підходу до паралелізації може як покращити, так і погіршити отримані показники.

Висновки

У ході дослідження розглянуто метод Ньютона як один із найважливіших методів для розв’язання систем нелінійних рівнянь, проаналізовано його переваги та обмеження, а також основні сфери застосування. Розглянуто можливості паралелізації обчислень та побудовано алгоритмічну структуру паралельного методу Ньютона.

Розроблено програмну реалізацію паралельного алгоритму з використанням MPI, наведено UML-діаграми та описано ключові компоненти. Проведено тестування продуктивності для різних розмірів систем та різної кількості процесів, виконано аналіз коефіцієнтів прискорення та ефективності.

Встановлено, що збільшення кількості процесів призводить до помірного росту прискорення, проте ефективність зменшується через комунікаційні накладні витрати та складність операцій лінійної алгебри. Паралельний метод Ньютона є особливо ефективним для великих задач, де він дозволяє суттєво знизити загальний час обчислень.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Мартинюк А. Паралельні обчислення: навчальний посібник. Вінниця: ВНТУ, 2021. 78 с. URL: https://mpa.vntu.edu.ua/fdb/838/Lec_CITCSHI/Tema_2.pdf .
2. Корочкін О. В., Русанова О. В. Паралельні та розподілені обчислення. Вибрані розділи : навч. посібник. Київ : КПІ ім. Ігоря Сікорського, 2020. 123 с.
3. Márcio Matheus de Lima Barboza. Newton’s Method Applied to Nonlinear Boundary Value Problems: A Numerical Approach University of Rio Grande do Norte Caicó, 2024. 96 p.
4. Gropp W., Lusk E., Skjellum A. Using MPI: Portable Programming with the Message Passing Interface. MIT Press, 2014.
5. MPI Standard Documentation. URL: https://www.mpi-forum.org .
6. Saad Y. Iterative Methods for Sparse Linear Systems. SIAM, 2003.
7. Burden R., Faires J. Numerical Analysis. Cengage Learning.
8. Джеймс А. Шардт, Майкл Джессі Чонолес. Короткий посібник зі стандартної мови об`єктного моделювання. John Wiley and Sons Ltd. 2003. 432 с.

Шиманська Софія Олександрівна – студентка кафедри комп`ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: sofismk7@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.

Shymanska Sofiia Olexandrivna – student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: sofismk7@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