

КЕРУВАННЯ СИСТЕМАМИ З РОЗПОДІЛЕНИМИ ПАРАМЕТРАМИ, МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

УДК 519.6

В.А. Сидорук, П.С. Єршов

АДАПТИВНИЙ АЛГОРИТМ РОЗВ'ЯЗАННЯ СИСТЕМ РІВНЯНЬ З БЛОЧНО-ХМАРОЧОСНИМИ МАТРИЦЯМИ

Сидорук Володимир Антонович

Інститут кібернетики ім. В.М. Глушкова НАН України, м. Київ,
wolodymyr.sydoruk@gmail.com

Єршов Павло Сергійович

Інститут кібернетики ім. В.М. Глушкова НАН України, м. Київ,
yershov.pavel.wsk@gmail.com

Зростаючі вимоги до якості проєктних рішень, а також використання нових конструктивних матеріалів викликають необхідність у розв'язанні якісно нових задач. Також завжди існує потреба у виконанні розрахунків складних унікальних конструкцій. Тому зростає необхідність у нових методах і підходах, пов'язаних із побудовою та дослідженням коректних комп'ютерних моделей, які адекватно відображають реальну роботу конструкцій. Використання деталізованих математичних моделей призводить до суттєвого зростання розмірів розрахункових (дискретних) задач, а отже, і відповідних матриць. Зазвичай такі матриці мають розріджену структуру та надвеликі розміри. У результаті виникають проблеми ефективного збереження, декомпозиції та обробки таких даних. Застосовуючи структурну регуляризацію матриць, можна вирішувати наступні завдання: компактне збереження даних; швидкий доступ до великих масивів даних та їх обробка; мінімізація обмінів даними між обчислювальними пристроями. Для задач із розрідженими симетричними матрицями блочно-хмарочосного виду запропоновано адаптивний паралельний алгоритм прямого методу, який забезпечує високу ефективність розпаралелювання і враховує структуру розріджених матриць та їх наповненість даними. Розроблений алгоритм дозволяє виконати розподіл між процесами обчислення з блоками ненульових елементів трикутного розвинення розрідженої матриці таким чином, щоб вони проводилися одночасно більшістю процесів. Отримано оцінки кількості арифметичних операцій, що виконуються алгоритмом, та коефіцієнта прискорення. Також отримано часові характеристики і показники прискорення при розв'язанні низки практичних задач моделювання міцності будівельних конструкцій на різній кількості процесорних ядер із застосуванням різної величини блоків, використовуваних для обчислень.

Ключові слова: математичне моделювання, паралельні алгоритми, змінна розрядність, розріджені матриці.

Вступ

З появою та постійним розвитком паралельних комп'ютерів різної архітектури, які використовуються для математичного моделювання в різних галузях науки та інженерії, дуже великої актуальності набувають проблеми їх ефективного використання та підвищення рівня інтелектуальної інформаційної підтримки користувачів — фахівців у різних предметних областях.

Постійне зростання потужностей суперкомп'ютерів відбувається за рахунок ускладнення архітектури багатопроцесорних обчислювальних систем [1–3] із розподіленою та ієрархічною спільною пам'яттю, зі складною організацією багатопотокових і векторизованих операцій, із застосуванням різного типу прискорювачів, різних видів кеш-пам'яті, систем розпаралелення тощо [4–8].

Використання деталізованих математичних моделей призводить до суттєвого зростання розмірів розрахункових (дискретних) задач, а отже, і відповідних матриць. Зазвичай такі матриці мають розріджену структуру та надвеликі розміри. У результаті виникають проблеми ефективного збереження, декомпозиції та обробки таких даних. Застосовуючи структурну регуляризацію матриць, можна вирішувати наступні завдання: компактне збереження даних; швидкий доступ до великих масивів даних та їх обробка; мінімізація обмінів даними між обчислювальними пристроями.

Розглядаючи проблему в такій постановці, приходимо до дискретних математичних моделей із надвеликими об'ємами даних, у яких кількість степенів свободи може перевищувати 10^7 . Для ефективною реалізації таких задач не вистачає наявних обчислювальних ресурсів сучасних персональних комп'ютерів та робочих станцій.

Сьогодні підвищення продуктивності та ефективності таких обчислень можливе лише за рахунок використання новітніх високопродуктивних обчислювальних систем. Для ефективного використання таких високопродуктивних рішень необхідно розробляти адаптивні алгоритми та програмне забезпечення нового покоління, що враховуватиме апаратні та системні особливості комп'ютерів, а також особливості розв'язуваної задачі.

Наявність практичних задач великих обсягів, які підлягають математичному моделюванню, та велике розмаїття обчислювальної техніки різної архітектури ставлять перед розробниками алгоритмічно-програмного забезпечення нові завдання — створювати адаптивні алгоритми [9, 10] та програми з функціями їх автоматичного налаштування на змінне комп'ютерне середовище (змішана розрядність, багаторозрядна арифметика, змінна топологія міжпроцесорних зв'язків, багаторівневий паралелізм, кешизація обчислень тощо), що забезпечує достовірність комп'ютерних результатів розв'язування задач із наближеними даними при ефективному використанні обчислювальних ресурсів для конкретної задачі.

Задачі аналізу міцності складних конструкцій або їхніх елементів виникають у багатьох галузях народного господарства, зокрема в будівництві. Зростаючі вимоги до якості проектних рішень, а також використання нових конструктивних матеріалів викликають необхідність розв'язання якісно нових задач. Також завжди існує необхідність у виконанні розрахунків складних унікальних конструкцій. Тому зростає потреба в нових методах і підходах, пов'язаних із побудовою та дослідженням коректних комп'ютерних моделей, які адекватно відображають реальну роботу конструкцій.

Математично лінійні стаціонарні задачі розрахунку міцності конструкцій після використання принципу можливих переміщень можуть бути поставлені у нескінченновимірному функціональному просторі можливих переміщень U_0 у ви-

гляді такої варіаційної задачі: знайти вектор-функцію $u \in U_0$, яка для будь-якої вектор-функції $v \in U_0$ (будь-якого можливого переміщення) задовольняє інтегральній тотожності

$$a(u, v) = l(f, v), \quad (1)$$

де симетрична білінійна форма (функціонал) $a(u, v)$ пропорційна потенційній енергії деформації, а лінійна (по v) форма $l(f, v)$ пропорційна роботі прикладених (зовнішніх) зусиль при навантаженні.

Тут і далі $U_0 \equiv U^{(k, m)}(\Omega) = (W_2^k(\Omega))^m$ — простір Соболева m -вимірних вектор-функцій; $\Omega \subset R^n$ — m -вимірна скінченна область із досить гладкою границею Γ ($m = 1, 2, 3$). Також позначимо $f_v \in (L_2(\Omega))^m$ вектор розподілених об'ємних сил; $f_s \in (L_2(\Gamma))^m$ — вектор розподілених поверхневих сил; $\varepsilon_{ij}(u) = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$ — складові тензора деформацій; $\sigma_{ki}(u) = \sigma_{ik}(u) = \sum_{j=1}^m c_{ijkl}(x) \varepsilon_{ij}(u)$ — складові тензора напруг. Припускається, що коефіцієнти пружності $c_{ijkl}(x)$ підпорядковано умовам симетрії $c_{ijkl} = c_{klij} = c_{jikl}$, і, крім того, вони задовольняють умові

$$\sum_{j, k, i=1}^m c_{ijkl} \varepsilon_{ij} \varepsilon_{kl} \geq \mu_0 \sum_{i, j=1}^m \varepsilon_{ij}^2 \quad (\mu_0 = \text{const} > 0). \quad (2)$$

Тоді білінійна та лінійна форми визначаються так:

$$a(u, v) = \int_{\Omega} \sum_{i, j, k, l=1}^m \sigma_{ij}(u) \varepsilon_{kl}(v) dx, \quad l(f, v) = \int_{\Omega} (f_V \cdot v) dx + \int_{\Gamma} (f_S \cdot v) d\gamma,$$

де $(u \cdot v)$ — евклідовий скалярний добуток m -вимірних векторів u та v .

Необхідна умова існування розв'язку задачі (1) — сукупність усіх сил, які прикладено до пружного тіла, статично еквівалентна нулю, тобто головний вектор і головний момент цих сил мають дорівнювати нулю:

$$\int_{\Omega} f_V dx + \int_{\Gamma} f_S d\gamma = 0, \quad l(f, v) = \int_{\Omega} (r \times f_V) dx + \int_{\Gamma} (r \times f_S) d\gamma = 0, \quad (3)$$

де r — радіус-вектор точки області Ω , а $(u \times v)$ — векторний добуток. Але розв'язок задач (1)–(3) щодо переміщень не єдиний — можливий рух тіла (поступальний перенос і жорсткий поворот) як твердого тіла. Для виділення єдиного розв'язку необхідно, щоб виконувались умови

$$\int_{\Omega} u dx = 0, \quad \int_{\Omega} (r \times u) dx = 0, \quad (4)$$

перша з яких виключає довільний поступальний перенос, а друга — довільний жорсткий поворот тіла (конструкції). Для тривимірних моделей умови (3) будуть виконуватись, якщо конструкцію закріплено не менше ніж у трьох точках, які не лежать в одній площині.

Отже, єдиний розв'язок задачі (1)–(3) існує, якщо

$$U_0(\Omega) = \left\{ u \mid u \in U^{(1,m)}(\Omega), \int_{\Omega} u dx = 0, \int_{\Omega} (r \times u) dx = 0. \right.$$

Дискретні моделі

При дискретизації наближені розв'язки відповідних задач шукаються у скінченновимірному підпросторі U_0^h простору U_0 . Використовуються при цьому проєкційні методи, наприклад метод скінченних елементів (МСЕ) та метод скінченних різниць (МСР).

Для отримання дискретної моделі за допомогою МСЕ у формі переміщень зайнята конструкцією область розбивається на скінченні елементи і визначаються вузли та їх степені свободи (зміщення і кути повороту). Степеням свободи відповідають базисні (координатні, апроксимуючі) вектор-функції φ_i ($i = 1, 2, \dots, N$), які задовольняють головним (кінетичним) умовам. Вектор-функції з підпростору U_0^h є кусково-поліноміальними і можуть бути представлені у вигляді лінійної комбінації базисних вектор-функцій:

$$U_h(\chi) = \sum_{i=1}^N x_i \varphi_i(\chi). \quad (5)$$

Тоді відповідна дискретна задача (для лінійного розрахунку міцності конструкції) [5, 6] з урахуванням (4), (5) має вигляд системи лінійних алгебраїчних рівнянь (СЛАР)

$$Ax = b. \quad (6)$$

При побудові алгоритму розв'язування СЛАР із розрідженими матрицями особливе значення має вибір способу зберігання ненульових елементів. Однією з найбільш універсальних схем зберігання розріджених матриць є розріджений рядковий формат. Згідно з ним значення ненульових елементів матриці та відповідні стовпчикові індекси зберігаються у двох масивах — AN та JA . Використовується також масив вказівників IA , що відмічають позиції масивів AN та JA , з яких починається опис чергового рядка. Додаткова компонента в IA містить вказівник першої вільної позиції в AN та JA .

Розріджений рядковий формат являє собою одну з найбільш використовуваних схем зберігання розріджених матриць. Ця схема висуває мінімальні вимоги до пам'яті та в той же час виявляється надзвичайно зручною для кількох важливих операцій над розрідженими матрицями: додавання, множення, перестановок рядків та стовпчиків, транспонування і розв'язування систем із розрідженими матрицями коефіцієнтів як прямими, так й ітераційними методами.

Проте, схеми зберігання даних загального виду, такі як, наприклад, розріджений рядковий формат, не враховують специфіки портрету матриці. Використання інформації про її особливості при побудові схеми дає змогу прискорити доступ до елементів матриці та виконання операцій над ними. Одним із таких прикладів є хмарочосна схема [5].

Часто на практиці, наприклад при розрахунку конструкцій, виникають задачі з симетричними матрицями особливої структури. У верхньому трикутнику таких матриць ненульові значення розміщуються вертикальними смугами по кілька стовпчиків підряд, що на вигляд нагадує хмарочоси. Таким чином, будь-який рядок верхнього трикутника складається з послідовних груп ненульових та нульових елементів. Оскільки ненульові елементи зберігаються групами, зникає необхідність зберігати індекси кожного елемента, що дає змогу зменшити розміри індексних масивів. У цьому полягає суть хмарочосної схеми.

Розглядувана схема є модифікацією розрідженого рядкового формату, поширеного на випадок матриць описаного виду. Кожен рядок матриці задається трьома масивами. Числові значення елементів матриці зберігаються у масиві A , що має довжину, рівну кількості ненульових елементів рядка, починаючи з діагонального. Цілочислові масиви I та J містять довжини груп ненульових і нульових елементів відповідно і мають розмір, рівний числу таких груп у рядку. Їхня поелементна сума для кожного рядка співпадає з порядком матриці. Ще один цілочисловий масив K містить значення кількості груп кожного рядка. Та-

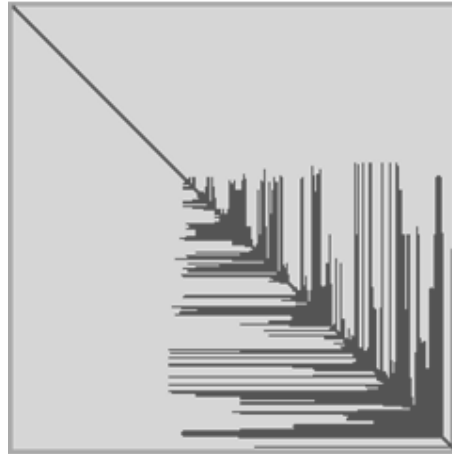


Рис. 1

ким чином, якщо групи елементів мають більше двох елементів, хмарочосна схема вимагає зберігати менше індексів, ніж розріджений рядковий формат, а економія пам'яті тим більша, чим об'ємніші групи ненульових та нульових елементів. Приклад хмарочосної матриці представлено на рис. 1.

Адаптивний алгоритм розв'язання лінійної системи

При розв'язанні СЛАР (6) вирішальну роль у балансуванні завантаження процесів, враховуючи загальну кількість арифметичних операцій, відіграє розподіл даних між процесами при виконанні трикутного розвинення матриці. Розподіл між процесами елементів трикутного розвинення розрідженої матриці визначає також розподіл решти даних, що задіяні в обчисленнях. Залежно від розподілу даних можна одержати ту чи іншу алгоритмічну реалізацію певного методу.

Для алгоритму $U^T U$ -розвинення симетричної матриці блочно-хмарочосної структури використано таку схему обчислень. Розглянемо випадок блочно-розрідженої структури даних. Розглянемо комп'ютер із p CPU та спільною пам'яттю. Нехай A — квадратна матриця порядку n .

Розіб'ємо матрицю A на блоки розмірністю $s*s$. Розміри сітки блоків —

$$N*N, \text{ де } N = \left\lceil \frac{n}{s} \right\rceil, \text{ за умови, що } n \text{ націло ділиться на } s, \text{ інакше } N = \left(\left\lceil \frac{n}{s} \right\rceil + 1 \right).$$

Приклад матриці блочно-хмарочосної структури наведено на рис. 2.

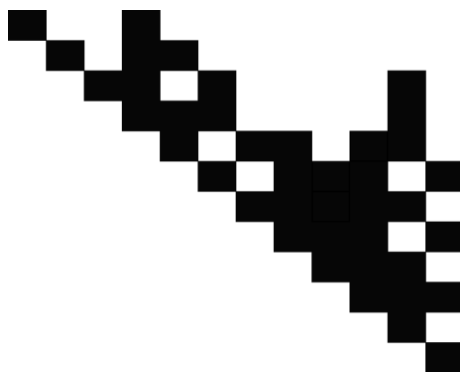


Рис. 2

Для виконання факторизації матриці на k -му кроці блочного розвинення використовуємо таке співвідношення

$$A^k = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} U_{11}^T & 0 \\ U_{12}^T & U_{22}^T \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{pmatrix}, \quad (7)$$

де розміри блоків $A_{11} — s \times s$, $A_{12} — (N - ks) \times s$, $A_{22} — (N - ks) \times (N - ks)$, а блоки A_{12} та A_{22} враховують розріджену структуру матриці A .

Опираючись на (7), реалізуємо метод $U^T U$ -розвинення для блоків на k -му кроці за формулами

$$A_{11} = U_{11}^T U_{11}, \quad (8)$$

$$U_{12} = (U_{11}^T)^{-1} \times A_{12}, \quad (9)$$

$$\tilde{A}_{22} = A_{22} - U_{12}^T U_{12}. \quad (10)$$

Зазначимо, що реалізація (8)–(10) на кожному кроці модифікує тільки ненульові блоки матриці A .

Для подальшого представлення матеріалу введемо додаткові позначення: $NZ(k)$ — кількість ненульових блоків у k -му рядку блоків верхнього трикутника без діагонального блоку, A_{kj}, U_{kj} — ненульові блоки у k -му рядку блоків верхнього трикутника.

Кожен з p потоків, використовуваних в обчисленнях, працюватиме з $\frac{NZ(k)}{p}$ стовпчиками блоків у матриці A .

Паралельний алгоритм $U^T U$ -розвинення симетричної матриці блочно-хмарочосної структури реалізується таким чином.

Факторизуємо діагональний блок

$$A_{kk} = U_{kk}^T U_{kk}. \quad (11)$$

У всіх потоках модифікуємо позадіагональні блоки з A_{12} :

$$U_{kj} = (U_{kk}^T)^{-1} A_{kj}, \quad j = k + 1, \dots, N. \quad (12)$$

У всіх потоках оновлюємо відповідні блоки з A_{22} за такими співвідношеннями

$$\tilde{A}_{ij} = A_{ij} - U_{ki}^T U_{kj}, \quad i = k + 1, \dots, N; \quad j = i, \dots, N. \quad (13)$$

Паралельні алгоритми розв'язання систем із блочно-трикутними матрицями. Одним із етапів розв'язання систем із розрідженою блочно-хмарочосною матрицею є розв'язання систем із блочними трикутними матрицями. Розглянемо відповідний паралельний алгоритм, який дозволяє ефективно реалізувати обчислення, використовуючи декомпозицію даних, що була запропонована вище.

Паралельний алгоритм розв'язання системи з нижньою блочно-трикутною матрицею реалізується таким чином.

Послідовно для кожного i -го ($i = 1, 2, \dots, N - 1$) блочного рядка виконуємо такі макрооперації: розв'язуємо систему з нижньою трикутною матрицею $U_{ii}^T y_i = b_i$ і паралельно у відповідних потоках модифікуємо k -ті частини вектора правих частин $b_k = b_k - U_{ik}^T y_i, k = \overline{i + 1, N}$.

Для останнього рядка розв'язується система $U_{NN}^T y_N = b_N$.

Паралельний алгоритм розв'язання систем із блочною верхньою трикутною матрицею реалізується таким чином:

- виконуємо розв'язання системи з верхньою трикутною матрицею:

$$U_{NN} x_N = y_N.$$

Для $i, i = \overline{1, N-1}$, у зворотному порядку виконуємо мікрооперації:

- у відповідних потоках модифікуємо y_i :

$$y_i^* = y_i - U_{ik} x_k; \quad k = \overline{i+1, N};$$

- виконуємо розв'язання трикутних систем:

$$U_{ii} x_i = y_i^*.$$

Оцінки ефективності та прискорення. Оскільки етап знаходження трикутного розвинення матриці найбільш складний за часом і кількістю операцій, сумарне прискорення програми буде визначатись саме прискоренням алгоритму розвинення. Для оцінки якості паралельних алгоритмів використовуються такі критерії, як коефіцієнти прискорення і ефективності, які відповідно можуть бути визначені таким чином (наприклад, див. [4]):

$$S_p = T_1 / T_p, \quad E_p = S_p / p, \quad (14)$$

де p — кількість процесів, що використовуються для розв'язання задачі; T_p — час розв'язання задачі на комп'ютері з використанням p процесів; T_1 — час розв'язання тієї ж задачі на гіпотетичному послідовному комп'ютері з швидкістю одного процесора і оперативною пам'яттю, яка дорівнює сумарній пам'яті, що використовується p процесами.

Час виконання послідовного алгоритму для знаходження коефіцієнтів (1) можна обчислити так:

$$T_1 = O_1 t.$$

А час виконання паралельного алгоритму — так:

$$T_p = \max \{T_{p,1}, T_{p,2}, \dots, T_{p,p}\}, \quad T_{p,i} = T_{\text{пар},i} + T_{\text{посл},i} + T_{o,i} + T_{c,i}.$$

Тут t — середній час виконання однієї арифметичної операції (додавання або множення) з плаваючою комою; O_1 — кількість таких арифметичних операцій у послідовному алгоритмі; $T_{\text{пар},i}$ — час, що витрачається i -м процесом на паралельні арифметичні операції; $T_{\text{посл},i}$ — час, що витрачається i -м процесом на послідовні арифметичні операції; $T_{o,i}$ — час, що витрачається i -м процесом на обмін даними з іншими процесорними пристроями; $T_{c,i}$ — час, що витрачається i -м процесом на синхронізацію з іншими процесорними пристроями. Для багатопроекторних (багатоядерних) комп'ютерів час T_p можна визначити і так:

$$T_p = O_p t + O_o t_o + O_c t_c,$$

де O_p — максимальна кількість арифметичних операцій з плаваючою комою, що виконуються одним процесом у паралельному алгоритмі; O_o, t_o — максимальний обсяг (кількість машинних слів) обмінів, що виконуються одним процесом, та час, необхідний для обміну одним машинним словом між двома процесами; O_c, t_c — максимальна кількість синхронізацій, що виконуються одним процесом, та час, необхідний для синхронізації двох процесів.

Теорема 1. *Кількість операцій, виконуваних послідовним алгоритмом, дорівнює*

$$O_1 = \frac{Ns^3}{3} + 2s^3 * \sum_{k=1}^{N-1} \left(\frac{NZ(k)^2 + 3NZ(k)}{2} \right).$$

Доведення. Розглянемо алгоритм факторизації блочно-хмарочосної матриці. Враховуючи розбиття на блоки, отримуємо, що для кожного з N блочних рядків необхідно провести розвинення діагонального блоку (11), для кожного з $N-1$ блочних рядків $NZ(k)$ раз обчислюються макрооперації (12) для оновлення позадіагональних блоків верхнього трикутника розвинення та $\frac{NZ(k)^2 + NZ(k)}{2}$ виконати (13) для оновлення всіх блоків відповідної блочної підматриці.

Звідси впливають такі оцінки кількості операцій, виконуваних алгоритмом у процесі обчислень.

- Факторизація діагональних блоків: $\frac{Ns^3}{3}$ операцій.
- Оновлення позадіагональних блоків: $2s^3 \sum_{k=1}^{N-1} NZ(k)$ операцій.
- Оновлення значень у блоках підматриці: $2s^3 \sum_{k=1}^{N-1} \frac{NZ(k)^2 + NZ(k)}{2}$ операцій.

Просумувавши ці величини та спростивши вираз, отримуємо співвідношення

$$O_1 = \frac{Ns^3}{3} + 2s^3 * \sum_{k=1}^{N-1} \left(\frac{NZ(k)^2 + 3NZ(k)}{2} \right).$$

Теорему доведено.

Теорема 2. *Кількість операцій, виконуваних паралельним алгоритмом, дорівнює*

$$O_p = \frac{Ns^3}{3p} + 2s^3 * \sum_{k=1}^{N-1} \left(\frac{NZ(k)^2 + 3NZ(k)}{2p} \right).$$

Доведення. Використавши попередній розподіл даних та інформацію про кількість виконуваних макрооперацій (11)–(13), знайдемо кількість операцій, виконуваних одним потоком при виконанні паралельного алгоритму.

Оскільки для виконання алгоритму задіяно p потоків і архітектура комп'ютера із спільною пам'яттю, це дозволяє для факторизації діагонального блоку використати один з існуючих алгоритмів розвинення щільних матриць на одноузлових комп'ютерах. Кількість операцій, виконуваних одним потоком, тоді буде оцінюватись величиною $\frac{Ns^3}{3p}$ операцій.

Оновлення позадіагональних блоків проводиться незалежно у кожному з p потоків, відповідно виконується $2s^3 \sum_{k=1}^{N-1} \frac{NZ(k)}{p}$ операцій.

Оновлення значень у блоках підматриці також виконується у кожному з p потоків: $2s^3 \sum_{k=1}^{N-1} \frac{NZ(k)^2 + NZ(k)}{2p}$ операцій.

Просумувавши ці величини та спростивши вираз, отримуємо співвідношення

$$O_p = \frac{Ns^3}{3p} + 2s^3 * \sum_{k=1}^{N-1} \left(\frac{NZ(k)^2 + 3NZ(k)}{2p} \right).$$

Теорему доведено.

Теорема 3. Прискорення паралельного алгоритму оцінюється величиною

$$S_p \approx \frac{P}{1 + 2 * \left(\frac{Ns}{p} + \sum_{k=1}^{N-1} \left(\frac{NZ(k)^2 + 3NZ(k)}{2p} \right) \right)} t_c.$$

Доведення. Для знаходження величини прискорення використаємо співвідношення (14). У ньому вже відомі O_1 та O_p . Оскільки алгоритм передбачає виконання в межах одного вузла та використовує спільну пам'ять, величиною O_o можна знехтувати, оскільки фізично обміну даними практично не відбувається. Кількість синхронізацій при виконанні алгоритму можна оцінити

величиною $2 \left(\frac{Ns}{p} + \sum_{k=1}^{N-1} \left(\frac{NZ(k)^2 + 3NZ(k)}{2p} \right) \right)$, оскільки необхідно двічі робити

синхронізації потоків, коли розпочинаємо та завершуємо паралельний регіон. Тут $\frac{Ns}{p}$ — кількість синхронізацій при виконанні розвинення діагонального блоку,

$\sum_{k=1}^{N-1} \left(\frac{NZ(k)^2 + 3NZ(k)}{2p} \right)$ — кількість синхронізацій, що виконуються під час операції оновлення матриці розвинення.

Підставивши всі значення у формулу (14) та провівши відповідні спрощення, отримаємо вираз

$$S_p \approx \frac{P}{1 + 2 * \left(\frac{Ns}{p} + \sum_{k=1}^{N-1} \left(\frac{NZ(k)^2 + 3NZ(k)}{2p} \right) \right)} t_c.$$

Теорему доведено.

Апробація алгоритму. На основі запропонованого паралельного алгоритму було створено програмні модулі, які пройшли апробацію під час розв'язання задач розрахунку міцності будівельних конструкцій. Для тестів було обрано постановки наступних задач: аналіз напружено-деформованого стану фундаменту будівлі; статичний розрахунок напружено-деформованого стану промислової будівлі; аналіз міцності конструкції багатоповерхового будинку.

Апробація проводилась на вузлах 8-вузлового кластера СКІТ-5 [11] з піковою продуктивністю 100 Тфлопс. Кожен вузол має такі характеристики: 192 процесорних ядра (2x AMD EPYC 7642); 256 Гбайт оперативної пам'яті; мережа

передачі даних між вузлами Infiniband HDR — 200 Гбіт/с. Усі вузли інтегровані з загальним сховищем даних кластерного комплексу обсягом 200 Тбайт та працюють під управлінням операційної системи Ubuntu 20.04.

Під час програмної реалізації алгоритму застосовувались: технологія OpenMP — для організації паралелізму між потоками та присвоєння блоків хмарочосів відповідним потокам, а також модулі бібліотеки Intel® Math Kernel Library (Intel® MKL) 2019 — для ефективного виконання математичних операцій; опції компілятора, що дозволяють підвищити ефективність індексних операцій та роботи з пам'яттю на різних рівнях, використовувати при обчисленнях додаткові технології (AVX, SSE2) і проводити операції ділення та знаходження квадратного кореня зі збільшеною розрядністю.

У представлений далі таблиці наведено результати апробації паралельного алгоритму на кількох СЛАР із розрідженими матрицями хмарочосного формату. При цьому для кожної з задач проводилося розв'язання з матрицею вихідної (неоптимізованої) та переупорядкованої (оптимізованої) структури за допомогою алгоритму мінімального степеня; у таблиці дані випадки розрізняються значеннями ширини стрічки та заповнення профілю матриць.

Таблиця

Порядок	Ширина стрічки, елементів	Наповненість ненульовими елементами, %	1	8	16	32	64	128
44 436	4475	21	0,3	0,0621	0,029	0,016	0,026	0,033
44 436	37 580	2	0,26	0,05	0,0275	0,012	0,018	0,087
283 031	19 530	7	32	5,95	2,77	1,3	1,013	1,45
283 031	281 341	1	5,38	0,83	0,54	0,28	0,18	0,35
661 590	34 242	5	98,8	17,6	10,08	5,47	3,45	5,22
661 590	541 257	1	32,45	5,17	3,08	1,57	1,01	1,55

Далі наведено залежності прискорення алгоритму від кількості процесів для представлених у таблиці задач. Випадок із матрицею вихідної структури позначено як «O», а з переупорядкованою структурою — як «I».

Статичний розрахунок напружено-деформованого стану промислової споруди. Загальний вигляд конструкції та використовувану скінченно-елементну сітку представлено на рис. 3.

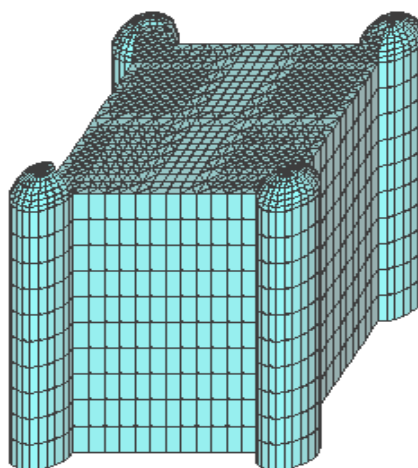


Рис. 3

Конструкцію розбито на 13 876 скінченних елементів. У результаті сформовано скінченно-елементну сітку з 7563 вузлами. Після дискретизації отримано СЛАР із матрицею порядку 44 436. Матриця цієї системи до застосування алгоритму впорядкування має стрічкову структуру (рис. 4, *ліворуч*) зі щільністю (заповненістю) 21 %. Після застосування алгоритму мінімальної степені для зміни впорядкування ненульових елементів матриці вдалося зменшити заповненість до 2 %, хоча при цьому ширина пів стрічки зростає (з 4476 до 27 850). Структуру оптимізованої матриці (до та після упорядкування елементів) представлено на рис. 4, *праворуч*.

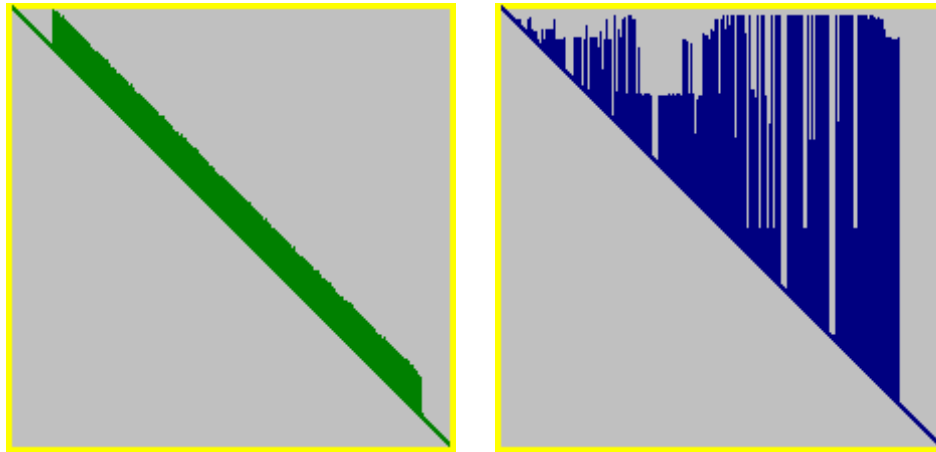


Рис. 4

На рис. 5 показано графіки прискорення при розв'язанні СЛАР із матрицями оптимізованої та оригінальної структур із використанням різної кількості потоків та розміром блоку 128 (залежність прискорення від кількості потоків для СЛАР 44 436-го порядку).

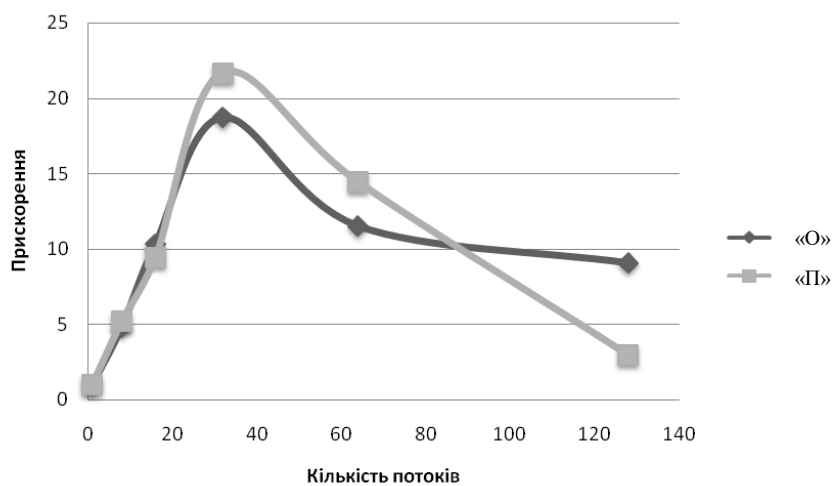


Рис. 5

Статичний розрахунок напруго-деформованого стану багатоповерхового житлового будинку. Загальний вигляд конструкції багатоповерхового житлового будинку та використана скінченно-елементна сітка представлені на рис. 6, *ліворуч*. Конструкція розбита на 121 761 скінченний елемент. Відповідна скінченно-елементна сітка містить 110 265 вузлів.

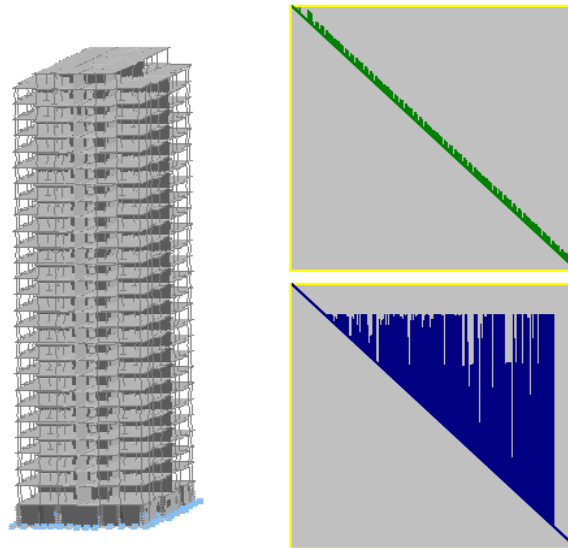


Рис. 6

У результаті дискретизації отримано СЛАР порядку 661 590. Вихідна структура матриці цієї системи мала щільність 5 % і півширину стрічки 34 242 (рис. 6, *зверху праворуч*). Після оптимізації структури матриці щільність впорядкованої структури складала 1 % при пів ширині стрічки, рівній 541 257 (рис. 6, *внизу праворуч*).

На рис. 7 показано графіки прискорень, отриманих при розв'язанні задачі з відповідними розрідженими матрицями на різній кількості потоків із використанням розміру блоку 256 (залежність прискорення від кількості потоків для СЛАР 661 590-го порядку).

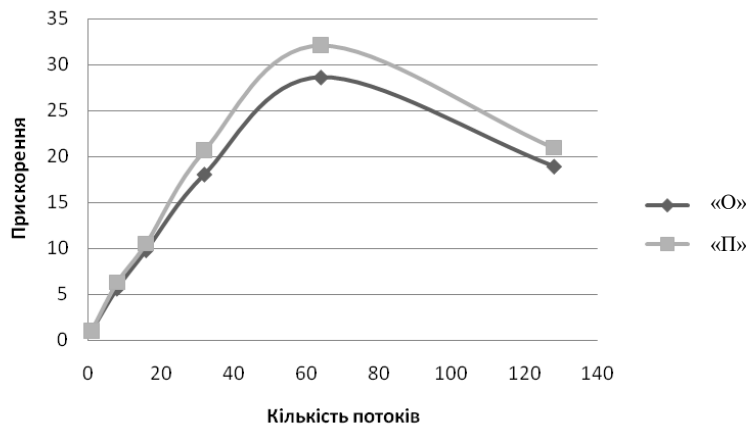


Рис. 7

Аналіз напружено-деформованого стану фундаменту будівлі. Загальний вигляд конструкції фундаменту та використану скінченно-елементну сітку представлено на рис. 8, *внизу*. Конструкція розбита на 94 346 скінченних елементів. Відповідна скінченно-елементна сітка містить 97 412 вузлів.

У результаті дискретизації отримано СЛАР порядку 283 031. Вихідна структура матриці цієї системи мала щільність 7 % і пів ширину стрічки 19 530 (рис. 8, *зверху ліворуч*). Після оптимізації структури матриці щільність впорядкованої структури складала 1 % при пів ширині стрічки, що дорівнює 281 341 (рис. 8, *зверху праворуч*).

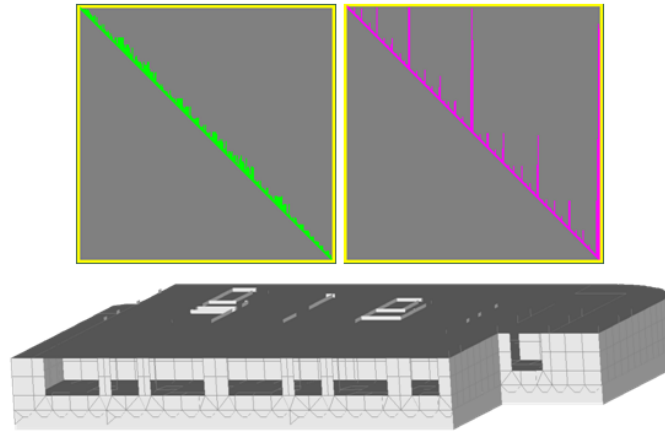


Рис. 8

На рис. 9 показано графіки прискорень, отриманих при розв'язанні задачі, з відповідними розрідженими матрицями на різній кількості потоків із використанням розміру блоку 160 (залежність прискорення від кількості потоків для СЛАР 283 031-го порядку).

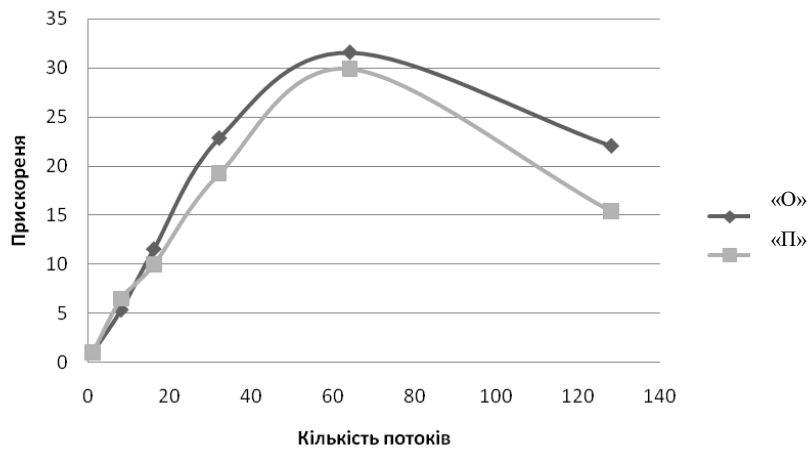


Рис. 9

Наведені в таблиці та на рис. 5, 7, 9 результати демонструють залежність характеристик алгоритму (часу розв'язання задачі, прискорення) від збалансованості завантаження процесів. При цьому чим менший порядок системи, тим складніше досягти збалансованості при нарощуванні числа процесів (див. графік на рис. 5). Також проведені дослідження вказують, що значний вплив на прискорення обчислень має і розмір блоку, з яким виконуються обчислення, оскільки він впливає як на швидкість послідовних операцій, так і на кількість операцій, які можуть бути виконані паралельно.

Висновок

Для задач із розрідженими симетричними матрицями блочно-хмарочосного виду запропоновано адаптивний паралельний алгоритм прямого методу, який забезпечує високу ефективність розпаралелювання, а також враховує структуру розріджених матриць та їхню наповненість даними. Розроблений алгоритм дозволяє провести розподіл обчислень із блоками ненульових елементів трикутного розв'язання розрідженої матриці таким чином, щоб вони проводилися одночасно більшістю процесів.

Отримано оцінки кількості арифметичних операцій, що виконуються алгоритмом, та коефіцієнта прискорення. Також отримано часові характеристики та показники прискорення при розв'язанні низки практичних задач моделювання міцності будівельних конструкцій на різній кількості процесорних ядер із застосуванням різної величини блоків, використовуваних для обчислень.

Подальші дослідження доцільно направити на пошук алгоритмів переупорядкування розріджених матриць, які забезпечували б найвищу ефективність паралельних алгоритмів вирішення задач лінійної алгебри, в першу чергу — за рахунок збалансованості завантаження процесів.

V. Sydoruk, P. Yershov

ADAPTIVE ALGORITHM FOR SOLVING SYSTEMS OF EQUATIONS WITH BLOCK-SKYSCRAPER MATRICES

Volodymyr Sydoruk

V.M. Glushkov Institute of Cybernetics of NAS of Ukraine, Kyiv,
wolodymyr.sydoruk@gmail.com

Pavlo Yershov

V.M. Glushkov Institute of Cybernetics of NAS of Ukraine, Kyiv,
yershov.pavel.wsk@gmail.com

Increasing requirements for the quality of design solutions, as well as the use of new structural materials necessitates the solution of qualitatively new problems. There is always a need to perform calculations of complex unique structures. Therefore, there is a growing need for new methods and approaches related to the construction and study of correct computer models that adequately reflect the real operation of structures. Use of detailed mathematical models leads to a significant increase in the size of computational (discrete) problems, and hence the corresponding matrices. Usually, such matrices have a sparse structure and extremely large sizes. As a result, there are problems of efficient storage, decomposition and processing of such data. Using structural regularization of matrices it is possible to solve the following problems: compact data storage; fast access and processing of large data sets; minimization of data exchanges between computing devices. For the tasks with sparse symmetric matrices of block-skyscraper type, an adaptive parallel algorithm of the direct method is proposed, which provides high parallelization efficiency, takes into account the structure of sparse matrices and their data content. The developed algorithm allows to distribute between the processes of calculations with blocks of non-zero elements of the triangular development of the sparse matrix so that they are carried out simultaneously by most processes. Estimates of the number of arithmetic operations performed by the algorithm and the speedup factor are obtained. Also obtained time characteristics and acceleration rates in solving a number of practical problems of modeling the strength of building structures on different numbers of processor cores using different sizes of blocks used for calculations.

Keywords: mathematical modeling, parallel algorithms, variable precision, sparse matrices.

REFERENCES

1. Sergienko I.V., Molchanov I.N., Khimich A.N. Intelligent technologies of high-performance computing. *Cybernetics and Systems Analysis*. 2010. Vol. 46 (5). P. 833–844. <https://doi.org/10.1007/s10559-010-9265-3>.
2. Dongarra J., Beckman P., Moore T. et al. The international exascale software project roadmap. *The International Journal of High Performance Computing Applications*. 2011. Vol. 25 (1). P. 3–60. doi:10.1177/1094342010391989.
3. Сергієнко І.В., Хімич О.М. Математичне моделювання: Від мелм до екзафлопсів. *Вісник НАН України*. 2019. № 8. С. 37–50.
4. Khimich A.N., Molchanov I.N., Popov A.V., Chistyakova T.V., and Yakovlev M.F. Parallel algorithms to solve problems in calculus mathematics. Kyiv : Naukova Dumka, 2008. 247 с.
5. Khimich A.N., Popov A.V., Polyankova V.V. Algorithms of parallel computations for linear algebra problems with irregularly structured matrices. *Cybernetics and Systems Analysis*. 2011. Vol. 47. P. 973–985. <https://doi.org/10.1007/s10559-011-9377-4>.
6. Baranov A.Yu., Popov A.V., Slobodyan Y.E., and Khimich A.N. Mathematical modeling of building constructions using hybrid computing systems. *Journal of Automation and Information Sciences*. 2017. Vol. 49, N 7. P. 18–32.
7. Khimich A.N., Popov A.V., and Chistyakov O.V. Hybrid algorithms for solving the algebraic eigenvalue problem with sparse matrices. *Cybernetics and Systems Analysis*. 2017. Vol. 53, N 6. P. 937–949. <https://doi.org/10.1007/s10559-017-9996-5>.
8. Khimich O.M., Popov O.V., Chistyakov O.V. et al. A parallel algorithm for solving a partial eigenvalue problem for block-diagonal bordered matrices. *Cybernetics and Systems Analysis*. 2020. Vol. 56. P. 913–923. <https://doi.org/10.1007/s10559-020-00311-z>.
9. Khimich O.M., Chistyakova T.V., Sidoruk V.A. et al. Adaptive computer technologies for solving problems of computational and applied mathematics. *Cybernetics and Systems Analysis*. 2021. Vol. 57. P. 990–997. <https://doi.org/10.1007/s10559-021-00424-z>.
10. Khimich A., Chistyakova T., Sydoruk V., Yershov P. Adaptive algorithms for researching problems in a variable computer environment. *Physico-mathematical modelling and informational technologies*. 2021. Vol. 33. P. 181–185. <https://doi.org/10.15407/fmmit2021.33.081>.
11. http://icybcluster.org.ua/index.php?lang_id=2&menu_id=5.

Отримано 07.12.2022