

ПАРАЛЕЛЬНИЙ АЛГОРИТМ РОЗВ'ЯЗУВАННЯ ЗАДАЧ ТЕОРІЇ ПРУЖНОСТІ

© Гомон К. О., Дияк І. І., Копитко М. Ф., 2017

Для розв'язування задач теорії пружності методом декомпозиції області запропоновано алгоритм з використанням паралельних обчислень. Глобальна система рівнянь для всієї області не формується, а подається через локальні матриці та вектори для підобластей з використанням булевих матриць зв'язності. Систему лінійних алгебраїчних рівнянь розв'язують модифікованим методом спряжених градієнтів. Розроблений алгоритм реалізований у вигляді програми мовою C++ з використанням бібліотеки паралельного виконання MPI. Наведено результати апробації запропонованого підходу на модельному прикладі.

Ключові слова: задача теорії пружності, метод декомпозиції області, паралельні обчислення, метод скінченних елементів, паралельний метод спряжених градієнтів.

Domain decomposition algorithm for solving problems of elasticity based on parallel computing is considered. The global system of equations for the entire domain is not formed and is represented by local matrices and vectors for subdomains using Boolean matrices of connectivity. The system of linear equations is solved by modified conjugate gradient method. The algorithm is implemented with C++ using parallel MPI library. The results of testing proposed approach for modeling example are included.

Key words: problem of the theory of elasticity, domain decomposition method, parallel computing, finite element method, parallel conjugate gradient method.

Вступ

Найпоширенішими і практично важливими класами задач теорії пружності є так звані плоскі задачі: плоска деформація і плоский напружений стан. У цих задачах всі шукані функції (переміщення u_1, u_2 , деформації e_{11}, e_{22}, e_{12} , напруження s_{11}, s_{22}, s_{12}) залежать тільки від двох координат x_1, x_2 . Як відомо, ці обидві задачі зводяться до ідентичної математичної постановки, що дає змогу використовувати для їх розв'язування той самий підхід. Залежно від складності геометрії об'єкта й умов навантаження використовують відповідні методи розв'язування задачі [1–3]. Однією із основних вимог до сучасних числових алгоритмів є можливість їх ефективного розпаралелювання та організації розподілених обчислень [4, 5]. Побудову такого підходу розглянуто в цій роботі.

Постановка задачі

Розглянемо двовимірну область Ω з границею Γ (рис. 1). На частині Γ_q границі прикладене розподілене навантаження q_1^0, q_2^0 , а на іншій частині Γ_u задані переміщення u_1^0, u_2^0 . У тілі діють масові сили $\mathbf{P}^T = [p_1, p_2]$.

Математична постановка плоскої задачі теорії пружності зводиться до розв'язання системи рівнянь:

рівнянь рівноваги

$$\frac{\partial s_{11}}{\partial x_1} + \frac{\partial s_{12}}{\partial x_2} + p_1 = 0, \quad \frac{\partial s_{21}}{\partial x_1} + \frac{\partial s_{22}}{\partial x_2} + p_2 = 0, \quad \frac{\partial^2 s_{11}}{\partial x_1^2} + \frac{\partial^2 s_{22}}{\partial x_2^2} = 0; \quad (1)$$

співвідношень Коші

$$e_{11} = \frac{\partial u_1}{\partial x_1}, \quad e_{22} = \frac{\partial u_2}{\partial x_2}, \quad e_{12} = \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}; \quad (2)$$

закону Гука

$$\boldsymbol{\sigma} \equiv \begin{bmatrix} S_{11} \\ S_{22} \\ S_{12} \end{bmatrix} = \mathbf{D} \begin{bmatrix} e_{11} \\ e_{22} \\ e_{12} \end{bmatrix} = \mathbf{D}\boldsymbol{\varepsilon}; \quad (3)$$

кінематичних граничних умов

$$\mathbf{u} \equiv \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} u_1^0 \\ u_1 \\ u_2^0 \\ u_2 \end{bmatrix} \quad \text{на } \Gamma_u; \quad (4)$$

статичних граничних умов

$$\mathbf{q} \equiv \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} s_{11}n_1 + s_{12}n_2 \\ s_{21}n_1 + s_{22}n_2 \end{bmatrix} = \begin{bmatrix} q_1^0 \\ q_2^0 \end{bmatrix}, \quad (5)$$

де (n_1, n_2) – вектор одиничної нормалі до границі Γ_q .

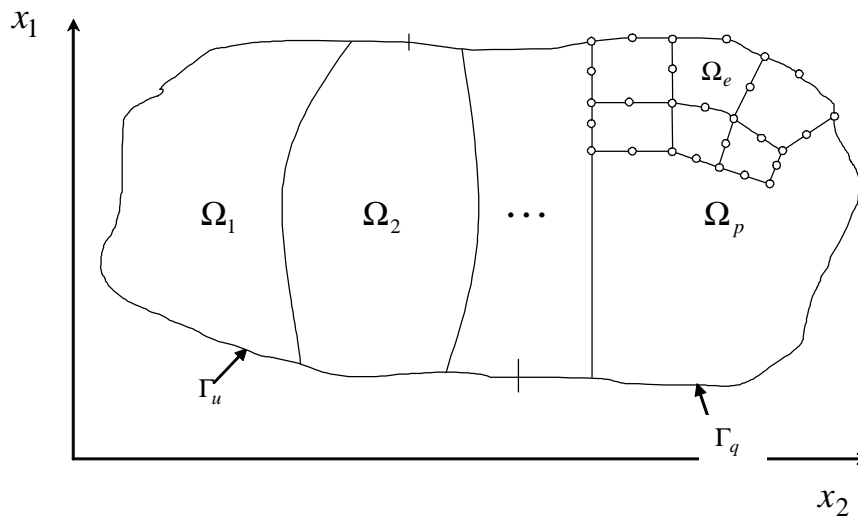


Рис. 1. Область Ω розв'язування задачі

У випадку плоскої деформації матриця \mathbf{D} у формулі (3) має вигляд

$$\mathbf{D} = \frac{E(1-n)}{(1+n)(1-2n)} \begin{bmatrix} 1 & n/(1-n) & 0 \\ n/(1-n) & 1 & 0 \\ 0 & 0 & (1-2n)/2(1-n) \end{bmatrix}, \quad (6)$$

де E – модуль пружності Юнга; n – коефіцієнт Пуассона. Для плоского напруженого стану матриця закону Гука має вигляд

$$\mathbf{D} = \frac{E}{(1-n^2)} \begin{bmatrix} 1 & n & 0 \\ n & 1 & 0 \\ 0 & 0 & (1-n)/2 \end{bmatrix}. \quad (7)$$

Для двовимірних задач механіки деформівного твердого тіла на основі принципу мінімуму потенціальної енергії Лагранжа розв'язування крайової задачі (1)–(5) еквівалентно мінімізації функціонала

$$F(\mathbf{u}) = \frac{1}{2} \int_{\Omega} (s_{11}e_{11} + s_{12}e_{12} + s_{22}e_{22}) d\Omega - \int_{\Omega} (p_1u_1 + p_2u_2) d\Omega - \int_{\Gamma_q} (q_1^0u_1 + q_2^0u_2) d\Gamma \quad (8)$$

у просторі функцій $H_A = \{\mathbf{u}(x_1, x_2) \in H^1(\Omega) : \mathbf{u} = 0, (x_1, x_2) \in \Gamma_u\}$.

Метод розв'язування

Для розв'язування задачі (1)–(5) або мінімізації функціонала (8) розроблено багато методів знаходження аналітичного або наближеного (числового) розв'язку. У цій роботі використовується метод скінченних елементів (МСЕ) з біквадратичними ізопараметричними апроксимаціями на чотирикутниках (рис. 1). Особливістю цього методу є те, що для подання апроксимованої функції і взаємного відображення стандартного квадрата $\Omega^* = \{x, h : -1 \leq x, h \leq 1\}$ на скінченний елемент Ω_e використовуються ті самі вузлові точки та базисні функції.

Шуканий вектор \mathbf{u} подамо на елементі Ω^* у вигляді

$$\mathbf{u} = \mathbf{N}_e \mathbf{u}_e, \quad (9)$$

де \mathbf{N}_e матриця, елементами якої є базисні функції,

$$\mathbf{N}_e = \begin{bmatrix} j_1 & 0 & j_2 & 0 & \mathbf{L} & j_8 & 0 \\ 0 & j_1 & 0 & j_2 & \mathbf{L} & 0 & j_8 \end{bmatrix} \quad (10)$$

$$\mathbf{u}_e^T = [u_1^1, u_2^1, u_1^2, u_2^2, \mathbf{L}, u_1^8, u_2^8]. \quad (11)$$

Використовуючи матричні позначення, запишемо функціонал (8) у вигляді

$$F(\mathbf{u}) = \frac{1}{2} \int_{\Omega} \mathbf{u}^T \mathbf{B}^T \mathbf{D} \mathbf{B} \mathbf{u} d\Omega - \int_{\Omega} \mathbf{u}^T \mathbf{P} d\Omega - \int_{\Gamma_q} \mathbf{u}^T \mathbf{q}^0 d\Gamma, \quad (12)$$

де \mathbf{B} – матриця похідних зі співвідношень Коші (2)

$$\mathbf{B} = \begin{bmatrix} \partial/\partial x_1 & 0 \\ 0 & \partial/\partial x_2 \\ \partial/\partial x_2 & \partial/\partial x_1 \end{bmatrix}. \quad (13)$$

Представимо область Ω у вигляді підобластей Ω_e і запишемо інтеграл по області Ω як суму інтегралів за скінченними елементами Ω_e у формулі для функціонала (12). Перейшовши від інтегрування за змінними x_1, x_2 до інтегрування на стандартному елементі Ω^* за змінними x, h і використавши представлення переміщень (9), отримаємо новий вираз для функціонала

$$F = \frac{1}{2} \sum_{e=1}^M \left[\int_{\Omega^*} \mathbf{u}_e^T \mathbf{N}_e^T \mathbf{B}^{*T} \mathbf{D} \mathbf{B}^* \mathbf{N}_e \mathbf{u}_e \det Je \, dx dh - \int_{\Omega^*} \mathbf{u}_e^T \mathbf{N}_e^T \mathbf{P}^* \det Je \, dx dh - \int_{\Gamma_q^*} \mathbf{u}_e^T \mathbf{N}_e^T \mathbf{q}^{*0} \, dg \right], \quad (13a)$$

де $\det Je$ якобіан переходу від координат x_1, x_2 до x, h . Винісши вектори коефіцієнтів \mathbf{u}_e за знак інтегралів і записавши умову мінімуму функціонала (13), одержимо систему лінійних рівнянь для визначення вузлових переміщень

$$\sum_{e=1}^M [\mathbf{K}_e \mathbf{u}_e - \mathbf{F}_p^e - \mathbf{F}_q^e] = 0, \quad (14)$$

де \mathbf{K}_e матриця жорсткості елемента Ω_e ; \mathbf{F}_p^e – внесок від масових сил; \mathbf{F}_q^e – внесок від зовнішнього навантаження у вектор правих частин.

Систему алгебраїчних лінійних рівнянь (СЛАР) (14) можна переписати у звичнішому матричному вигляді

$$\mathbf{A} \mathbf{x} = \mathbf{b}. \quad (15)$$

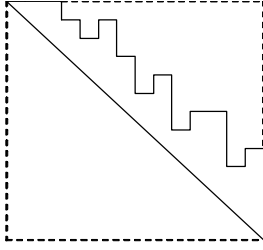


Рис. 2. Профіль ненульових елементів матриці СЛАР

Структура матриці \mathbf{A} і правої частини \mathbf{b} залежить від сітки розбиття області Ω на скінченні елементи Ω_e , особливо загальної нумерації вузлів. Матрицю \mathbf{A} можна розглядати як квадратну матрицю $N \times N$ (N – загальна кількість невідомих у вузлах), як блочно-діагональну матрицю або як матрицю рядкової будови, в якій ширина ненульових елементів дорівнює максимальному розміру блока. Від способу представлення матриці системи (15) залежить вибір методу її розв’язування, а, відповідно, і час розв’язування задачі загалом. У нашому підході матриця \mathbf{A} представляється як верхній профіль ненульових елементів (рис. 2) і зберігається по стовпцях як одновимірний масив. Це дає змогу значно зекономити оперативну пам’ять і зменшує тривалість матричних операцій.

Організація паралельних обчислень

У разі використання однопроцесорного комп’ютера всі етапи розв’язування задачі (1)-(5) виконуються послідовно: введення вхідних даних; побудова розбиття області Ω на скінченні елементи Ω_e , обчислення локальних (елементних $\mathbf{K}_e, \mathbf{F}_p^e, \mathbf{F}_q^e$) матриць; формування матриць системи (15) \mathbf{A}, \mathbf{b} , розв’язування СЛАР (15). У цій роботі пропонується паралельний алгоритм розв’язування задачі на n_p процесорах. Область Ω подається як об’єднання n_p підобластей (рис. 1), тобто $\Omega = \Omega_1 \cup \Omega_2 \cup \dots \cup \Omega_{n_p}, \Omega_i \cap \Omega_j = \emptyset, i \neq j, i, j = 1, n_p$. У кожній підобласті Ω_i будується сітка МСЕ з використанням i -го процесу. Кожна підобласть Ω_i містить N_i вузлів і M_i

елементів. Загальна кількість елементів у області Ω дорівнює $M = \sum_{i=1}^{n_p} M_i$ і вузлів $N < \sum_{i=1}^{n_p} N_i$.

У запропонованому підході глобальна система рівнянь для всієї області не формується. Алгоритм ґрунтується на поданні матриць і векторів для всієї області через локальні матриці та вектори для підобластей з використанням булевих матриць зв’язності, які відображають співвідношення між локальними степенями вільності та незалежними невідомими для всієї області.

Позначимо через $\tilde{\mathbf{A}}_i$ матрицю жорсткості, $\tilde{\mathbf{x}}_i$ – вектор невідомих, $\tilde{\mathbf{b}}_i$ – вектор правих частин підобласті Ω_i без внесків із сусідніх підобластей, а через $\bar{\mathbf{A}}_i, \bar{\mathbf{x}}_i, \bar{\mathbf{b}}_i$ ті самі величини, але з внесками із сусідніх підобластей у глобально розподіленому форматі. Глобальну матрицю жорсткості \mathbf{A} можна подати, використовуючи булеву матрицю $\mathbf{C}_i \in \mathbf{Z}^{N_i \times N}$ (матрицю зв’язності), формулою

$$\mathbf{A} = \sum_{i=1}^{n_p} \mathbf{C}_i^T \tilde{\mathbf{A}}_i \mathbf{C}_i. \quad (16)$$

Аналогічно глобальні вектори \mathbf{x} і \mathbf{b} можна представити через глобально розподілені вектори $\bar{\mathbf{x}}_i$ і $\bar{\mathbf{b}}_i$

$$\bar{\mathbf{x}}_i = \mathbf{C}_i \mathbf{x}, \quad \bar{\mathbf{b}}_i = \mathbf{C}_i \mathbf{b} \quad (17)$$

або через локально розподілені вектори $\tilde{\mathbf{x}}_i$ і $\tilde{\mathbf{b}}_i$

$$\mathbf{x} = \sum_{i=1}^{n_p} \mathbf{C}_i^T \tilde{\mathbf{x}}_i, \quad \mathbf{b} = \sum_{i=1}^{n_p} \mathbf{C}_i^T \tilde{\mathbf{b}}_i. \quad (18)$$

Відповідно локально розподілені вектори $\tilde{\mathbf{x}}_i$ і $\tilde{\mathbf{b}}_i$, використовуючи формули (16), (17), можна перетворити на глобально розподілені вектори $\bar{\mathbf{x}}_i$ і $\bar{\mathbf{b}}_i$ за формулами

$$\bar{\mathbf{x}}_i = \mathbf{C}_i \sum_{i=1}^{n_p} \mathbf{C}_i^T \tilde{\mathbf{x}}_i, \quad \bar{\mathbf{b}}_i = \mathbf{C}_i \sum_{i=1}^{n_p} \mathbf{C}_i^T \tilde{\mathbf{b}}_i. \quad (19)$$

Необхідно зазначити, що саме обчислення за формулами (19) потребують комунікації між процесами, тобто здійснення обміну між процесами – частинами векторів $\tilde{\mathbf{x}}_i$ і $\tilde{\mathbf{b}}_i$, які необхідні для обчислення $\bar{\mathbf{x}}_i$ і $\bar{\mathbf{b}}_i$. Позначимо процедуру перетворення $\tilde{\mathbf{x}}_i$ у $\bar{\mathbf{x}}_i$ і $\tilde{\mathbf{b}}_i$ у $\bar{\mathbf{b}}_i$ як

$$\bar{\mathbf{x}}_i \equiv \sum_{\Leftrightarrow}^{\partial\Omega_i} \tilde{\mathbf{x}}_i, \quad \bar{\mathbf{b}}_i \equiv \sum_{\Leftrightarrow}^{\partial\Omega_i} \tilde{\mathbf{b}}_i. \quad (20)$$

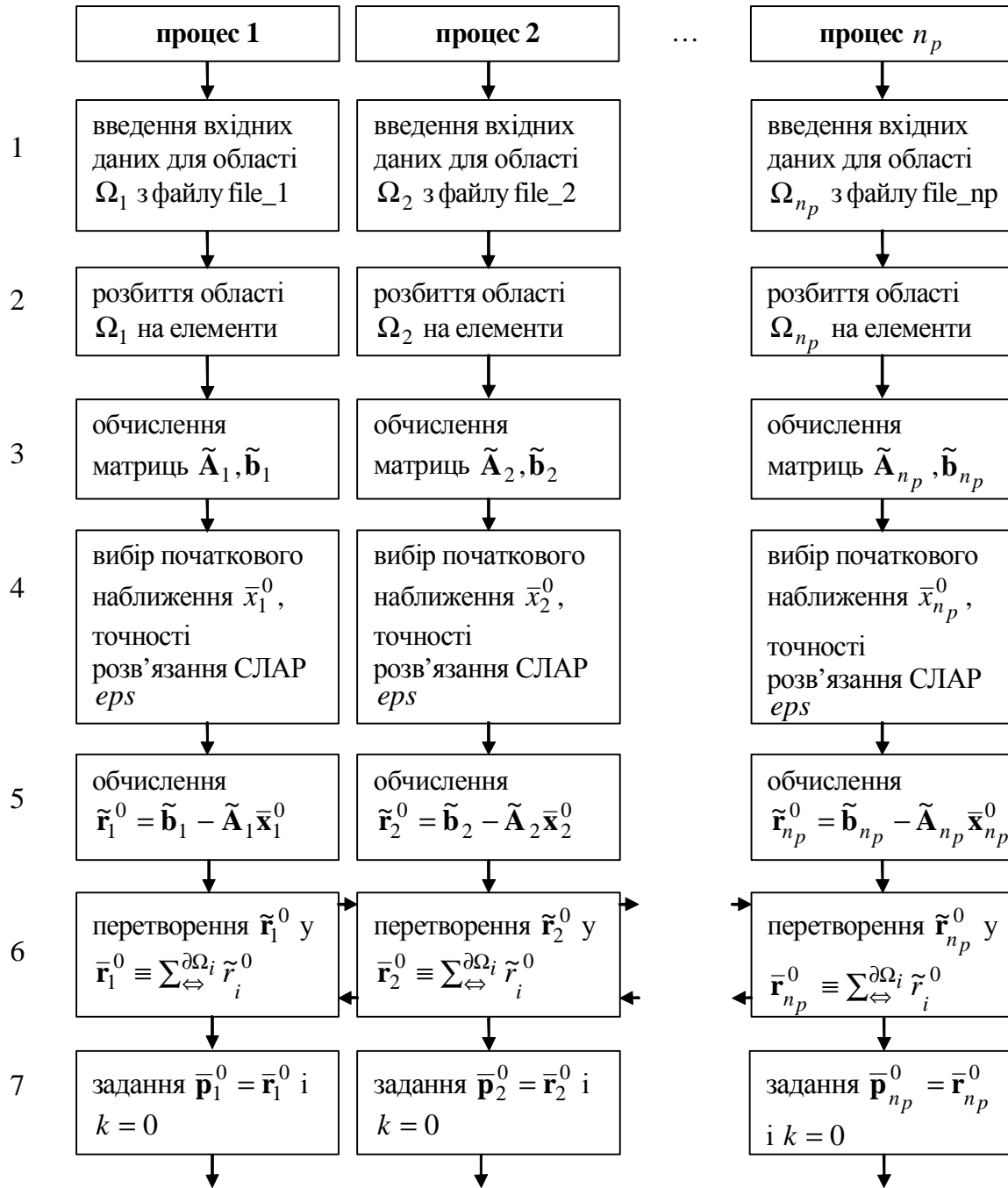


Рис. 3. Блок-схема паралельного розв'язування задачі з використанням n_p процесів

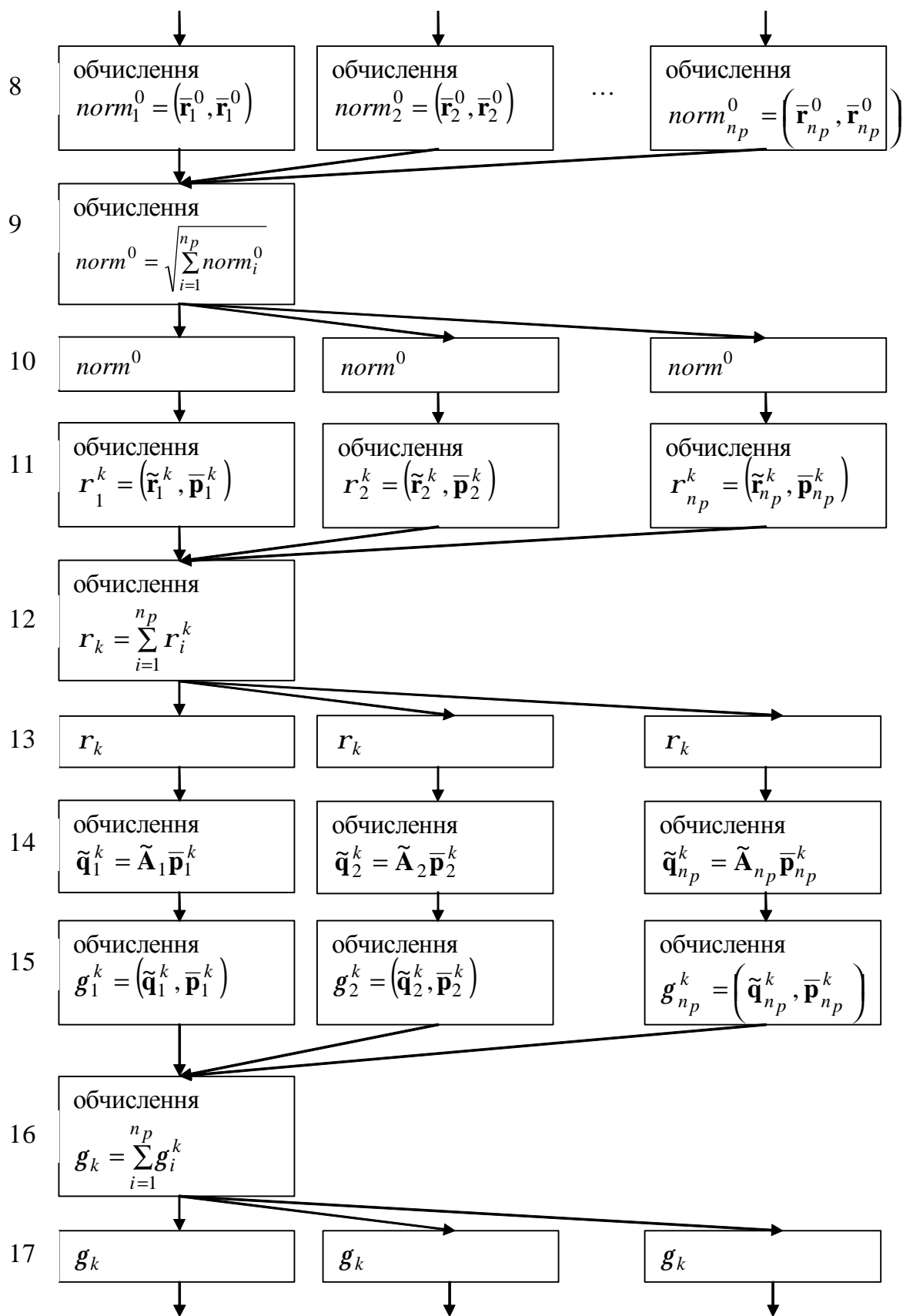


Рис. 4. Продовження блок-схеми з рис. 3 паралельного розв'язування задачі з використанням n_p процесів

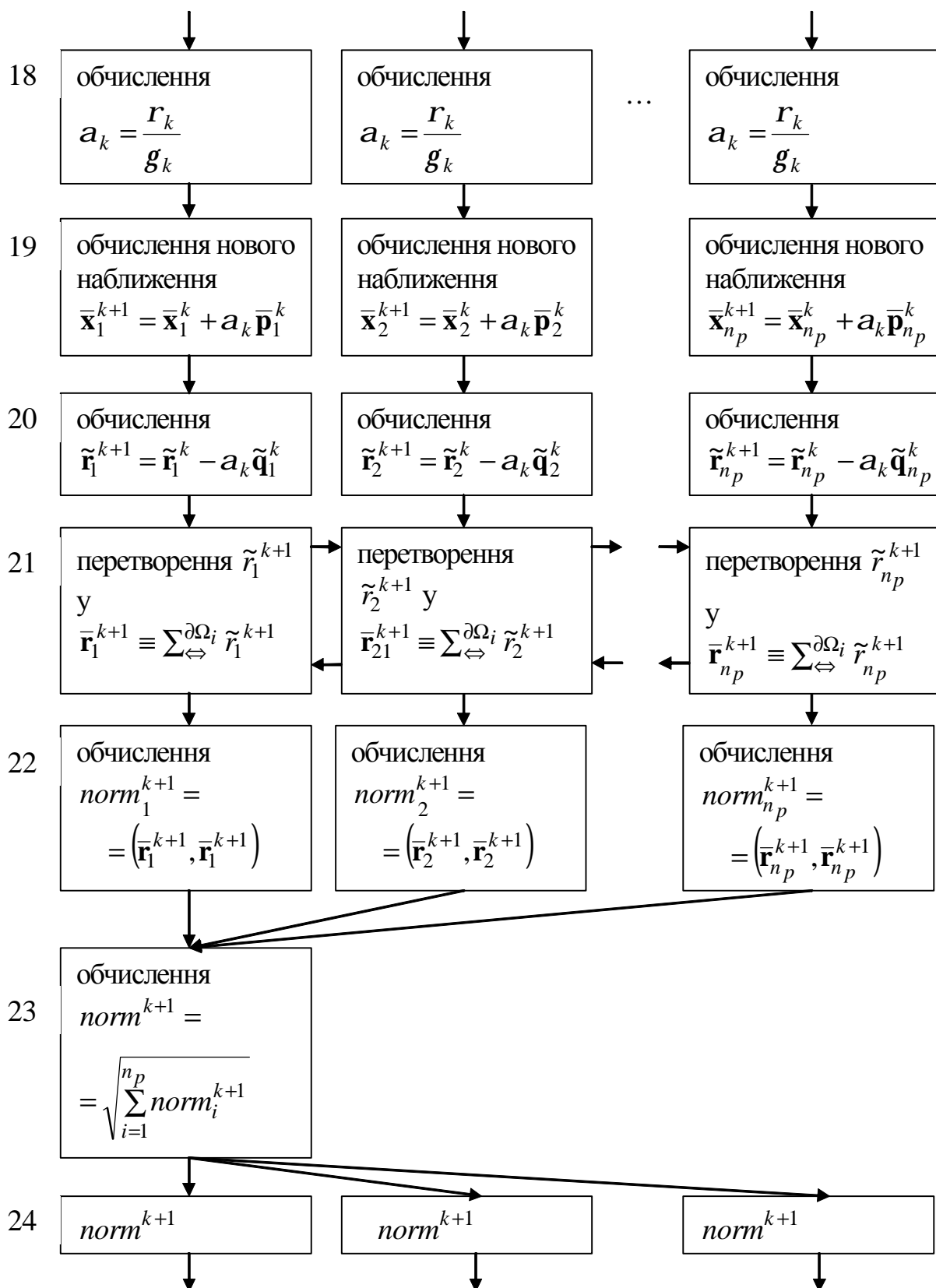


Рис. 5. Продовження блок-схеми з рис. 3 паралельного розв'язування задачі з використанням n_p процесів

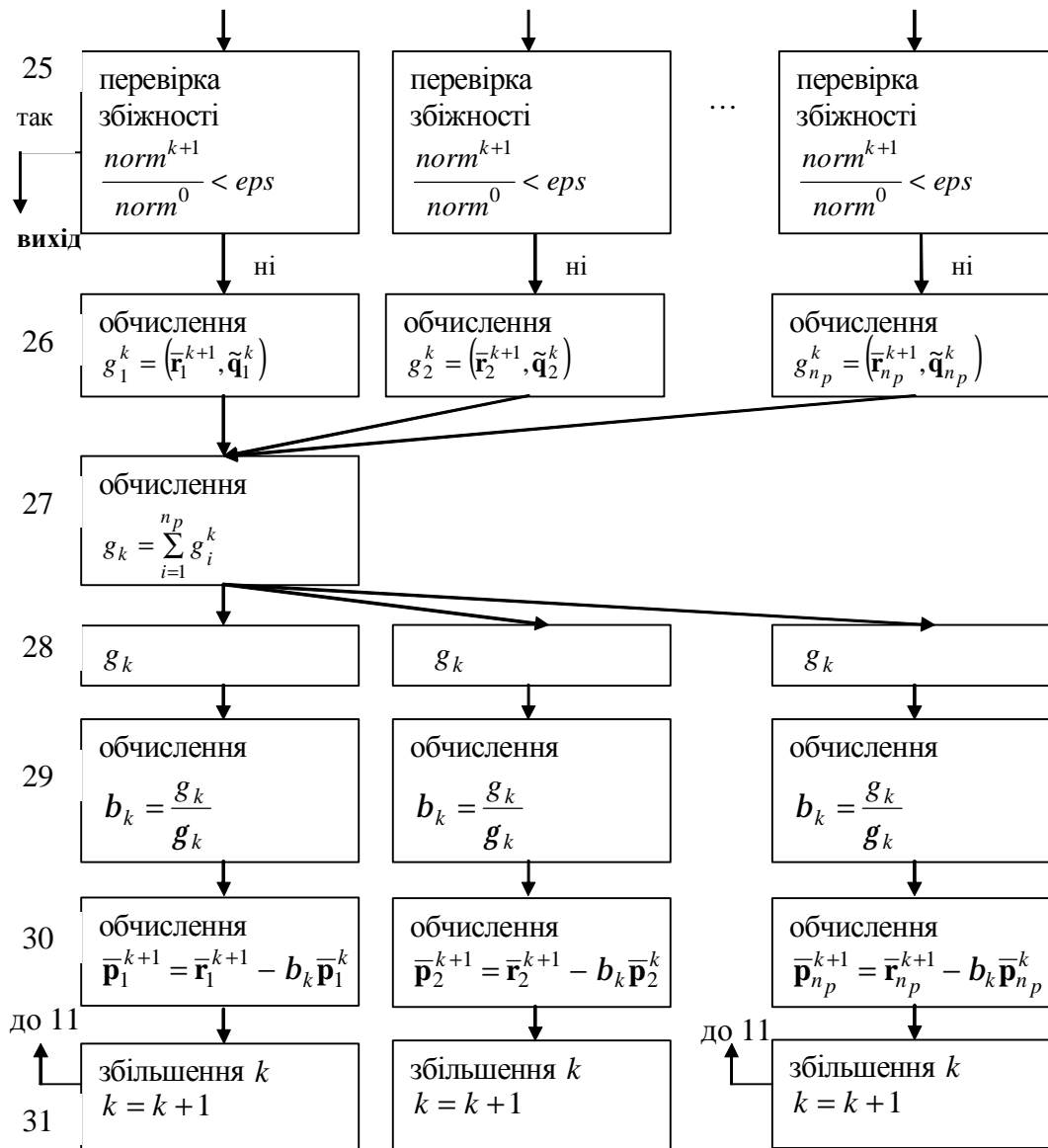


Рис. 6. Закінчення блок-схеми з рис. 3 паралельного розв'язування задачі з використанням n_p процесів

На рис. 3–6 зображено розроблений авторами алгоритм паралельного розв'язування задачі (1)–(5) з використанням n_p процесів. Для його реалізації розроблено програму мовою C++ з використанням бібліотеки паралельного виконання завдань Message Passing Interface (MPI) [6]. Основна частина алгоритму – розв'язування СЛАР модифікованим методом спряжених градієнтів (МСГ) (кроки 4–31). Як видно з блок-схеми, більшість операцій виконує паралельно i -й процес, тобто кожний процес “відповідає” за свою задачу для області Ω_i . Комунікація між процесами відбувається тільки на певних кроках алгоритму. Зокрема, кроки алгоритму 6 і 21 реалізують операцію (20), тобто пересилання одновимірних масивів між процесами за допомогою функції `MPI_Sendrecv()` [6]. Саме ця операція є найзатратнішою в алгоритмі й залежить від розбиття області Ω на підобласті Ω_i і від кількості вузлів на спільних границях підобластей. На кроках алгоритму 9, 12, 16, 23, 27 виконується передавання скалярних величин з усіх процесів на один з одночасним їх підсумуванням за допомогою функції `MPI_Reduce()`. Наступні кроки алгоритму 10, 13, 17, 24, 28 здійснюють розсилання обчислених скалярних даних з одного процесу на всі за допомогою функції `MPI_Bcast()`.

Тестовий приклад

Для апробації розробленого алгоритму паралельного розв'язування задачі (1)–(5) розглянемо плоску деформацію консольної балки (рис. 7), рівномірно навантаженої на верхній границі ($q_2 = 3\text{кН} / \text{м}^2$). Модуль Юнга матеріалу $E = 2,1 \cdot 10^6 \text{кН} / \text{м}^2$, коефіцієнт Пуассона $\nu = 0,3$.

Розрахунки виконано для різних розбиттів області Ω як на скінченні елементи, так і на підобласті. Оскільки здійснювалось тестування алгоритму, то вибір кількості елементів зумовлювався не достовірністю результатів (для цієї задачі достатньо значно менше елементів), а можливістю дослідити ефективність саме алгоритму паралельних обчислень і специфіки використання вузлів кластера. Для розв'язування задачі використано кластер Львівського національного університету імені Івана Франка, який складається з 14 обчислювальних вузлів і сервера. Обчислювальні вузли – AMD® Athlon II 2x із частотою 2,9 ГГц, 8 Гб ОЗП, 500 Гб HDD. GPU – Nvidia GTS450 (CUDA 2.1). Операційна система: Scientific Linux 6.2 (ядро 3.6.6). Усі обчислювальні вузли для обміну даними між паралельними процесами користувачських задач об'єднані мережею Ethernet 1 Гбіт/с.

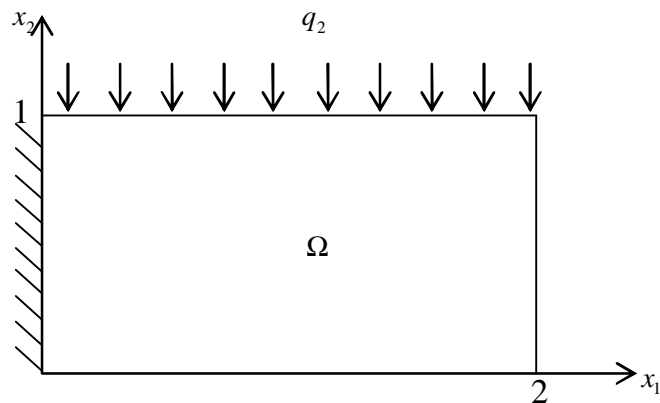


Рис. 7. Задача про плоску деформацію консольної балки

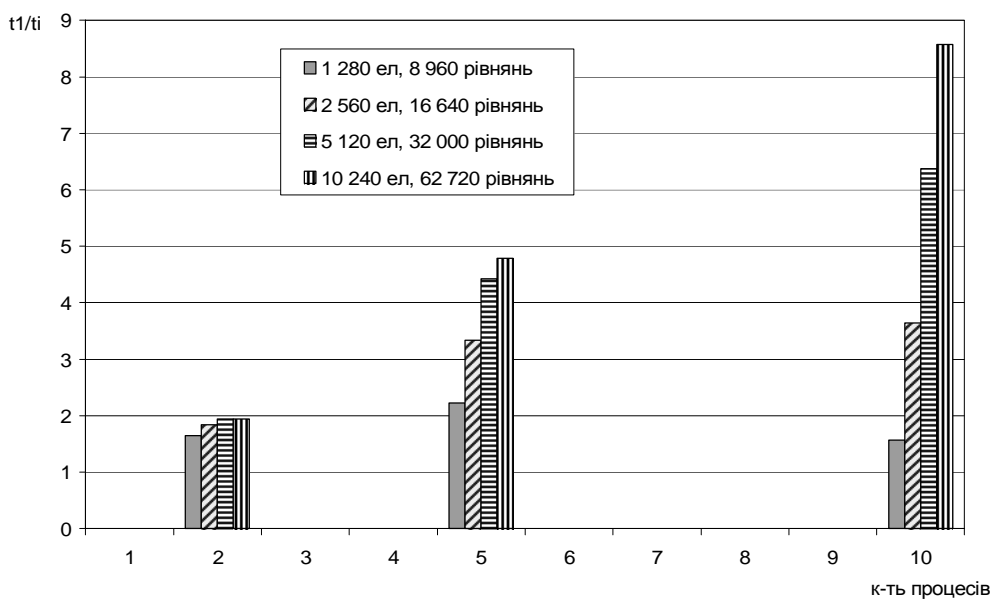


Рис. 8. Прискорення (t_1/t_i) паралельного алгоритму для різної кількості процесів

Дослідження здійснено на одному (послідовний алгоритм), п'ятьох і десятих вузлах. Для цього область Ω розділили на відповідну кількість підобластей. Розбиття всієї області й кожної підобласті здійснювалось так, щоб розмірність СЛАР була однаковою у всіх випадках, тобто для послідовного і паралельного алгоритмів. На рис. 8 показано ефективність алгоритму для різного розбиття області на скінченні елементи і відповідної розмірності СЛАР. Зображено прискорення t_1/t_i алгоритму для різної кількості процесів, де t_1 – час розв'язування задачі на одному процесорі (послідовний алгоритм), а t_i – час розв'язування задачі з використанням i процесорів. Як видно з наведених даних, чим більша розмірність СЛАР, тим ефективніше використовувати паралельний алгоритм. Для малих систем (<10 000 рівнянь) вигащ від задіяння десяти процесів менший, ніж 2, а для ще менших розмірностей СЛАР взагалі доцільно використовувати послідовний алгоритм (час на обміни між процесами перевищує час обчислень). Також необхідно відзначити, що для 10240 скінченних елементів (62720 рівнянь) час на введення даних, побудову сітки МСЕ – 0,02 %, обчислення і формування матриць \mathbf{A} , \mathbf{b} – 0,12 %, розв'язування СЛАР – 99,86 %. Тобто ефективність алгоритму визначається методом розв'язування СЛАР.

Висновки

Розроблений паралельний алгоритм розв'язування задач теорії пружності дає змогу ефективно поєднувати метод декомпозиції області з паралельними обчисленнями на довільній кількості процесорів. Використання модифікованого методу спряжених градієнтів для розв'язування СЛАР не вимагає формування глобальної матриці жорсткості для всієї області, що істотно зменшує загальний час розрахунку задачі. Як видно з результатів тестового прикладу, залежно від розміру задачі час обчислень зменшився від двох до дев'яти разів.

1. Григоренко А. Я. Применение метода декомпозиции области с использованием гибридных аппроксимаций для решения задач теории упругости / А. Я. Григоренко, И. И. Дьяк, И. И. Прокопъшин // Прикладная механика. – 2008. – Т. 44, № 11. – С. 18–29. 2. Дьяк І. І. Чисельне дослідження плоскої задачі теорії пружності методом граничних елементів / І. І. Дьяк // *Мат. методи та фіз.-мех. поля.* – 1997. – Т. 40. – С. 60–64. 3. Дьяк І. Числова ефективність гібридних скінченно-граничноелементних аппроксимаций задач теорії пружності на підставі методу декомпозиції області / І. Дьяк, І. Макар, І. Прокопъшин // *Вісник Львівського університету.* – 2007. – № 12 : Серія: прикладна математика та інформатика. – С. 93–100. 4. Хіміч О. М. Гібридний алгоритм розв'язування лінійних систем зі стрічковими матрицями прямими методами / О. М. Хіміч, А. Ю. Баранов // *Комп'ютерна математика: зб. наук. праць.* – 2013. – Вип. 2. – С. 80–87. 5. Хіміч О. М. Технологія грид-обчислень для математичного моделювання процесів в'язкого руйнування / О. М. Хіміч, В. В. Полякко, О. В. Попов, В. А. Сидорук, О. В. Чистяков // *Штучний інтелект.* – 2014. – № 4. – С. 101–110. 6. A High Performance Message Passing Library [Електронний ресурс]. – Режим доступу: <http://www.open-mpi.org> – 10.03.2017.