

**Гоменюк С.І.**

Запорізький національний університет

**Козуб В.Ю.**

Луганський національний університет імені Тараса Шевченка

## ПАРАЛЛЕЛЬНИЙ АЛГОРИТМ ФОРМУВАННЯ МАТРИЦІ ЖОРСТКОСТІ СКІНЧЕННОГО ЕЛЕМЕНТА

У статті досліджуються проектні розрахунки конструкцій, що потребують всебічного аналізу процесів деформування під дією експлуатаційних навантажень. Для розв'язання таких задач зазвичай використовуються чисельні методи, з яких найбільш поширеним є метод скінченних елементів. Точність методу визначається розмірністю розрахункових сіток. Крім того, в розрахункових схемах одночасно можуть використовуватися різні типи скінченних елементів. Одним з етапів розв'язування задач є формування матриць жорсткості скінченних елементів. Причому для елементів одного типу використовуються обчислення за однаковою процедурою і кількість таких обчислень зумовлює час розв'язування задачі. У традиційному підході ці обчислення виконуються послідовно. Для невеликого розміру розрахункових схем достатньо час розв'язування є незначним. У випадку великих розмірів сіток кількість та час розрахунків збільшуються, що потребує оптимізації обчислень з використанням алгоритмів паралельних обчислень. Для паралелізації методу скінченних елементів потрібно організувати управління роботою достатньо великої кількості процесів, упорядкувати обмін даними між процесорами. При виконанні обчислень виникають затримки, зумовлені незавершеністю виконання деяких етапів усіма процесорами. Найповільніша задача з тих, що виконуються паралельно, визначає час виконання програми. При формуванні матриць жорсткості виникає необхідність проводити інтегрування по області скінченного елемента. При використанні моментної схеми скінченних елементів переміщення і деформації апроксимуються однаковими поліномами, що спрощує обчислення інтегралів. Ця процедура закінчується формуванням блоку матриці жорсткості, причому для сусідніх скінченних елементів цей блок є частково спільним. У системі із загальною пам'яттю обмін інформацією між процесорами відбувається за допомогою змінних, що зберігаються в загальній пам'яті. Для забезпечення детермінованості виконання програми реалізовано поступове розпаралелювання програми. Розроблено алгоритми паралельного програмування для побудови матриць жорсткості скінченних елементів та розрахунку напружено-деформованого стану для пакету програм «МІРЕЛА+». Паралельна обробка даних реалізована за допомогою бібліотеки OpenMP, що забезпечує більшу швидкість розробки через зручність використання. На основі обчислювальних методів експериментів встановлено використання алгоритмів паралельних обчислень на початку розв'язування задач традиційними скінченноелементними алгоритмами для великорозмірних сіток.

**Ключові слова:** метод скінченних елементів, матриця жорсткості, паралельні обчислення, OpenMP, напружено-деформований стан.

**Постановка проблеми.** Традиційна процедура побудовання матриць жорсткості скінченних елементів має матричне формулювання та зводиться до обчислення інтегралів від функцій, що задані в місцевій системі координат [1, 2]. Інтеграли обчислюються за допомогою квадратурних процедур, які потребують великої кількості одноманітних розрахунків, що значно збільшує час розв'язування задач. Для розрахункових сіток великих розмірів час рішення задачі складається з етапів формування матриці жорсткості та розв'язування системи алгебраїчних рівнянь. Прискорення виконання цих етапів дозволить

зменшити затрати машинного часу. Одним з підходів до прискорення обчислень є використання паралельних алгоритмів [3, 4].

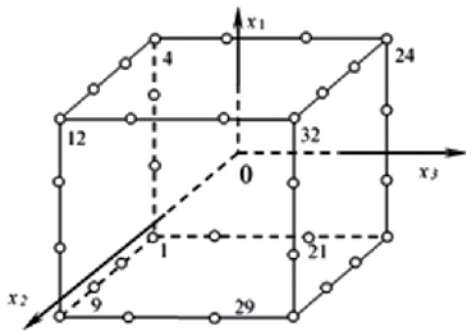
**Аналіз останніх досліджень і публікацій.** Використання в високопродуктивних комп'ютерів для проектних розрахунків конструкцій дозволяє цей процес значно прискорити за рахунок паралелізації обчислень. Для підвищення продуктивності паралелізації методу скінченних елементів використовуються різні підходи. Одними з таких способів є методи, що базуються на використанні блочних алгоритмів прямих методів дослідження та розв'язування систем лінійних алгебраїчних

рівнянь з розрідженими симетричними матрицями (стрічковими, профільними, блочно-діагональними) [5-8]. Такі підходи потребують ефективних методів забезпечення передобумовленості матриць системи, що іноді ставить нетривіальні завдання.

Інші підходи спрямовані на дослідження ефективності декомпозиції розрахункової області [9-10]. За методами декомпозиції задача розділяється на декілька підзадач меншої розмірності, розв'язання яких можна виконати паралельно. При чому можна використати відповідні методи рішення систем рівнянь [11-13]. Метод декомпозиції передбачає різні підходи до розділення на підобласті та способами моделювання контакту поверхонь між собою. В кожному випадку необхідно формулювати додаткові граничні умови та формувати розрахункову сітку скінченних елементів на кожній області окремо.

**Метою роботи** є розробка паралельного алгоритму побудови матриць жорсткості скінченних елементів.

**Виклад основного матеріалу.** Розглянемо побудову матриць жорсткості скінченного елемента з кубічною апроксимацією переміщень на основі моментної схеми скінченних елементів [14].



де 
$$b_{(\mu\nu\eta)}^s = \frac{\partial^{(\mu+\nu+\eta)} z_s}{\partial x_1^\mu \partial x_2^\nu \partial x_3^\eta} \Big|_{x_1=x_2=x_3=0}$$

Функцію змінення об'єму представимо також у вигляді ряду [16]:

$$\theta = \sum_{\alpha=0}^{l-1} \sum_{\beta=0}^{m-1} \sum_{\gamma=0}^{n-1} \xi^{(\alpha\beta\gamma)} \psi^{(\alpha\beta\gamma)}, \quad (9)$$

де коефіцієнти розкладання  $\xi^{(\alpha\beta\gamma)}$  визначаються співвідношенням:

$$\xi^{(\alpha\beta\gamma)} = \frac{\partial^{(\alpha+\beta+\lambda)} u_{i,j} g^{ij}}{\partial x_1^\alpha \partial x_2^\beta \partial x_3^\gamma} \Big|_{x_1=x_2=x_3=0}, \quad (10)$$

Аналіз кожної з компонент  $e_{ij}^{(pqr)}$  показав, що деякі коефіцієнти  $u_i^{(pqr)}$  не входять до розкладання для апроксимації переміщень. Ті коефіцієнти деформацій  $e_{ij}^{(pqr)}$ , які містять хоча б один з членів, відсутній в (4), повинні бути опущені в розкладанні. Таким чином встановлюється відповідний зв'язок між коефіцієнтами  $e_{ij}^{(pqr)}$  та  $u_i^{(pqr)}$ . Для апроксимаційних поліномів встановлюється співвідношення для інших компонент тензора деформацій.

Підставляючи (8), (9) у варіацію енергії пружної деформації отримуємо:

$$\delta W = \iiint_V \{\delta e\}^T \{\psi_{(ij)}\} 2\mu g^{ik} g^{jl} \{\psi_{(kl)}\}^T \{e_{kl}\} dV + \iiint_V \{\delta \xi\}^T \{\psi_{(\theta)}\} \lambda g^{ij} \{\psi_{(\theta)}\}^T \{\xi\} dV \quad (11)$$

$$\delta W = \{\delta e_{ij}\}^T [H^{ijkl}] \{e_{kl}\} + \{\delta \xi\}^T [H^\theta] \{\xi\},$$

де  $[E^{ijkl}] = \int_- \int_1^1 \int 2\mu \{\psi_{(ij)}\} g^{ik} g^{jl} \{\psi_{(kl)}\}^T \sqrt{g} dx_1 dx_2 dx_3;$

$$[E^\theta] = \int_- \int_1^1 \int \lambda \{\psi_{(\theta)}\} g^{ij} \{\psi_{(\theta)}\}^T \sqrt{g} dx_1 dx_2 dx_3; \quad (12)$$

Враховуючи зв'язок коефіцієнтів розкладання компонентів тензора деформацій та вектора переміщень рівняння (11) має вигляд:

$$\delta W = \delta \{\omega_s\}^T [F_{ij}^s]^T [H^{ijkl}] [F_{kl}^t] \{\omega_t\} + \delta \{\omega_s\}^T [F_{(\theta)}^s]^T [H^{(\theta)}] [F_{(\theta)}^t] \{\omega_t\} \quad (13)$$

Для побудови матриці жорсткості необхідно перейти у виразі (13) від коефіцієнтів  $\omega_s$  до значень переміщень у вузлах скінченного елемента. Між вузовими функціями форми та степеневими функціями для скінченного елемента існує простий зв'язок, який можна виразити у матричній формі

$$N_L = \sum_{i=0}^3 \sum_{j=0}^3 \sum_{k=0}^3 a_L^{ijk} \psi^{ijk}.$$

Переміщення довільної точки елемента:

$$u_i = \sum_{s=1}^n u_i^s \sum_{i=0}^3 \sum_{j=0}^3 \sum_{k=0}^3 a_s^{ijk} \psi^{ijk}.$$

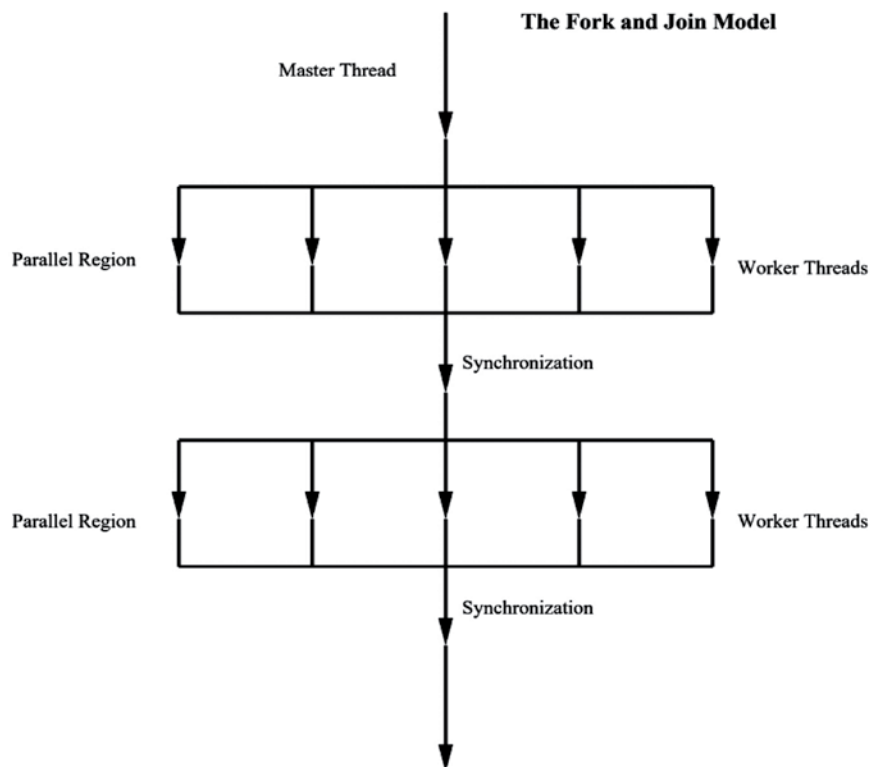


Рис. 2. Модель Fork and Join

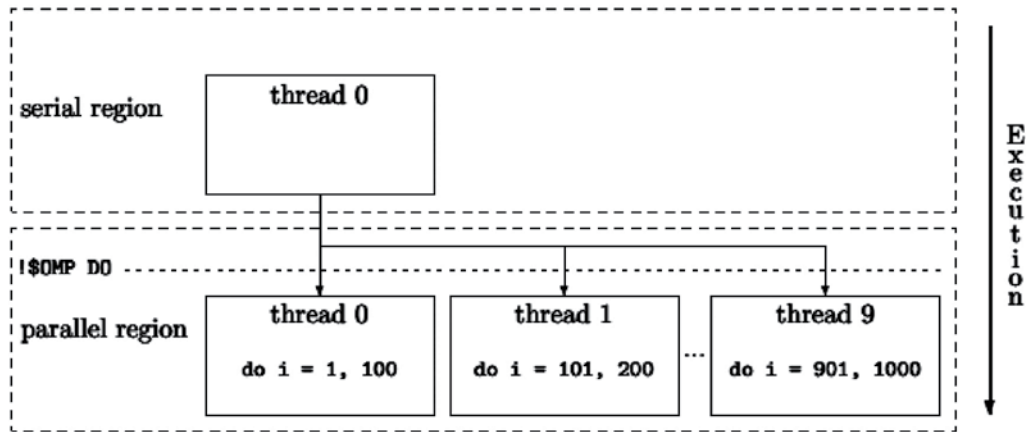


Рис. 3. Графічне зображення загального принципу роботи директиви !\$OMP DO !\$OMP END DO

З урахуванням зв'язку між коефіцієнтами розкладання по степеневим функціям для деформацій, вузлових переміщень та функції зміни об'єму варіація енергії деформування набуде вигляду:

$$\delta W = \delta\{u_s\}^T [K^{st}] \{u_t\} + \delta\{u_s\}^T [K_\delta^{st}] \{u_t\}.$$

Для визначення матриці жорсткості необхідно обчислити інтеграли що входять в рівняння (11), (12). Цю процедуру можна виконати за паралельним алгоритмом.

Для побудови паралельного алгоритму використано бібліотеку OpenMP. OpenMP являє собою набір директив компілятора, процедур та змінних середовища, призначених для паралельного програмування на машинах зі спільною пам'яттю. Паралельне виконання за моделлю Fork and Join лягає в основу архітектури бібліотеки (рис. 2).

Усі програми OpenMP починаються як єдиний процес, який називається головним потоком. Цей головний потік виконується послідовно доки не зустрінеться паралельна область. У цей момент головний потік «розгалужується» на кілька паралельних робочих потоків. Інструкції в паралельній області виконуються групою робочих потоків. У кінці паралельної області потоки синхронізуються та об'єднуються в єдиний головний потік. Щоб максимізувати продуктивність об'єднано вкладені цикли обчислення інтегралу. Потік керування для циклу додає накладні витрати на необхідну логіку. Об'єднання вкладених циклів в один цикл зменшує ці витрати.

```
!$OMP PARALLEL
!$OMP DO
do 12121 ijk=1,kt123
IK=MOD((ijk-1)/KT2/KT3,KT1)+1
```

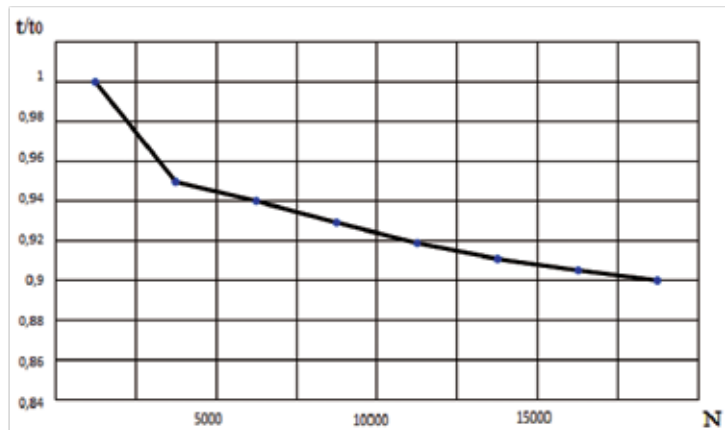


Рис. 4. Порівняння швидкості виконання для сіток з різною кількістю скінченних елементів

```
JK=MOD((ijk-1)/KT3,KT2)+1
KK=MOD((ijk-1),KT3)+1
```

```
....
12121 CONTINUE
!$OMP END DO
!$OMP END PARALLEL
```

Пара директив !\$OMP DO !\$OMP END DO забезпечують паралельне виконання циклу. Таким чином цикл do-loop розподіляється між різними потоками: кожен потік обчислює свою частину ітерацій. Наприклад, якщо використовується 10 потоків, то, як правило, кожен потік виконує обчислення 100 ітерацій циклу do-loop: потік 0 обчислює від 1 до 100, потік 1 від 101 до 200 і так далі (рис. 3).

В якості прикладу для перевірки та тестування представлено розрахунок квадратної тришарової плити під дією нормального навантаження, прикладеного до верхньої поверхні плити

$$p_3^{(n)} = \bar{p}_3^{(n)} \sin(\pi x_1/a) \sin(\pi x_2/a).$$

Відносний розмір у плані  $a/h=7$ , де  $h$  – загальна товщина пакета шарів. На бічних гранях плити задовольняються граничні умови типу Нав'є, яким у рамках побудованих моделей відповідають шарнірно рухливе закріплення контуру. Кожний шар являє собою композитний матеріал, армований односпрямованими волокнами. У зовнішніх шарах ( $k = 1, 3$ ) напрямок волокон збігається з віссю  $x_l$ , механічні й теплофізичні характеристики наступні:

$$E_1 = 172 \cdot 10^3 \text{ МПа}; E_2 = E_3 = 6,9 \cdot 10^3 \text{ МПа}; G_{12} = G_{13} = 3,45 \cdot 10^3 \text{ МПа}; \\ G_{23} = 1,38 \cdot 10^3 \text{ МПа}; \nu_{21} = \nu_{31} = \nu_{23} = 0,25.$$

Характеристики внутрішнього шару ( $k = 2$ ) виходять заміною нижніх індексів  $1 \leftrightarrow 2$ , тобто в цьому шарі волокна орієнтовані під кутом  $90^\circ$  до осі  $x_l$ . Товщина внутрішнього шару  $h_2 = 0,5 h$ , товщини зовнішніх шарів  $h_1 = 0,4h$ ,  $h_3 = 0,1h$ .

Для розрахунку використано обчислювальний комплекс «МІРЕЛА+» із застосуванням моментної схеми скінченних елементів. Тестування виконувалось на пристрої з Intel i7-9750H (6 ядер та 12 потоків) та 16 ГБ RAM.

На рис. 4 представлено порівняльні результати обчислень, де  $t_0$  – час розрахунку за традиційною схемою,  $t$  – час роботи з паралельним алгоритмом.

**Висновки.** Розроблена методика формування матриць жорсткості на основі використання паралельних обчислень інтегральних квадратурних складників дозволяє прискорити процес розв'язування задач механіки конструкцій. Використання системи паралельних обчислень на основі схеми зі спільною пам'яттю підвищує ефективність розробленого пакету прикладних програм «МІРЕЛА+».

#### Список літератури:

1. Бурман З.И., Артюхин Г.А., Зархин Б.Я. Программное обеспечение матричных алгоритмов и метода конечных элементов в инженерных расчетах. М.: Машиностроение. 1988. 256 с.
2. Воеводин В.В. Математические модели и методы в параллельных процессах. М.; Наука, 1986. 296 с.
3. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем. М.; Мир, 1991. 368 с.
4. Are Magnus Bruaset, Aslak Tveito. Numerical Solution of Partial Differential Equations on Parallel Computers. Springer Berlin Heidelberg, 2006.
5. Жданов А.И., Богданова Е.Ю. Об одной вычислительной реализации блочного метода Гаусса-Зейделя. *Вестн. Сам. гос. тех. ун-та. Сер. Физ.-мат. науки.* 2016. Т. 20, № 4. С. 730-738.
6. Wriggers P. Computational Contact Mechanics. Berlin-Heidelberg: Springer-Verlag. 2006. 520 p.
7. Попов А.В. Параллельные алгоритмы решения линейных систем с разреженными симметричными матрицами. *Проблеми програмування.* 2008. № 2-3. С. 111-118.
8. Исследование некоторых параллельных алгоритмов решения задач линейной алгебры на MIMD-компьютерах / Зубатенко В.С., Майстренко А.С., Молчанов И.Н. и др. *Искусственный интеллект.* 2006. № 3. С. 129-138.
9. C. Vollaire, L. Nicolas and A. Nicolas. Parallel computing for the Finite element method. *Eur. Phys. J. Appl. Physics.* 1998. 1(03). P. 305-314.
10. Каропова Е.Д., Шайдунов В.В., Вдовенко М.С. Параллельные реализации метода конечных элементов для краевой задачи для уравнений мелкой воды. *Вестник ЮУрГУ. Серия «Математическое моделирование и программирование».* 2009. Т. 17(150), вып. 3. С. 73-85.
11. Matteo Frigo, Steven G. Johnson. FFTW Home Page. 1997. URL: <http://www.fftw.org>.
12. P. Jarzebski, K. Wisniewski, R. L. Taylor. On parallelization of the loop over elements in FEAP // *Comput Mech.* 2015. 56. P. 77-86.
13. Fialko S (2010) PARFES: a method for solving finite element linear equations on multi-core computers. *Adv Eng Softw.* 41. P. 1256-1265.
14. Метод конечных элементов в механике твердых тел / под общей ред. А. С. Сахарова и И. Альтенбаха. К.: Вища школа, 1982. 480 с.
15. Киричевский В.В., Сахаров А.С. Нелинейные задачи термомеханики конструкций из слабосжимаемых эластомеров. К.: Будівельник, 1992. 216 с.
16. Метод конечных элементов в вычислительном комплексе «МІРЕЛА+» / под ред. В.В. Киричевского. К.: Наукова думка. 2005. 402 с.

#### Homeniuk S.I., Kozub V.Yu. PARALLEL ALGORITHM FOR FORMATION OF THE STIFFNESS MATRIX OF A FINITE ELEMENT

*Calculations of structures require a comprehensive analysis of deformation processes under the action of operational loads. Numerical methods are usually used to solve such problems, of which the finite element*

*method is the most common. The accuracy of the method is determined by the size of the calculation grids. In addition, different types of finite elements can be used simultaneously in calculation schemes. One of the stages of problem solving is the formation of stiffness matrices of finite elements. In the traditional approach, these calculations are performed sequentially. For a small size of calculation schemes, the solving time is quite insignificant. In the case of large grid sizes, the number and time of calculations increase, which requires optimization of calculations using parallel calculation algorithms. For the parallelization of the finite element method, it is necessary to organize the management of the work of a sufficiently large number of processes, to organize the exchange of data between processes. When performing calculations, there are delays caused by incomplete execution of some stages by all processors. The slowest parallel task determines the execution time of the program. When forming stiffness matrices, it is necessary to perform integration over the area of the finite element. When using the moment scheme of finite elements, displacements and deformations are approximated by the same polynomials, which simplifies the calculation of integrals. This procedure ends with the formation of a stiffness matrix block, and this block is partially shared by neighboring finite elements. In a system with shared memory, information is exchanged between processors using variables stored in shared memory. To ensure the determinism of program execution, gradual program parallelization is implemented. Parallel programming algorithms have been developed for constructing stiffness matrices of finite elements and calculating the stress-strain state for the «MIRELA+» program package. Parallel data processing is implemented using the OpenMP library, which provides faster development due to ease of use. On the basis of computational methods of experiments, the use of parallel computing algorithms at the beginning of solving problems by traditional finite element algorithms for large grids has been established.*

**Key words:** *finite element method, stiffness matrix, parallel computing, OpenMP, stress-strain state.*