

DOI 10.20535/2411-1031.2025.13.1.328901

УДК 004.925.8:004.43.032.24

АНАСТАСІЯ КАЛЮЖНЯК,

ОЛЕКСІЙ КУДІН

ДИСКРЕТНІ МОДЕЛІ ГЕОМЕТРИЧНИХ ОБ'ЄКТІВ В ПАРАЛЕЛЬНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ

Використання потужних обчислювальних систем визначають обчислювальні експерименти як нові та важливі методи досліджень. Враховуючи те, що традиційні аналітичні методи візуалізації математичних моделей мають визначальний характер, все ж виникає потреба в застосуванні сучасних математичних теорій, які дозволяють розширити можливості прикладних математичних досліджень. Об'єктом дослідження є процес розробки функціонального проектування складних геометричних моделей. Предметом даного наукового дослідження є використання паралельних методів побудови поверхонь дискретних моделей деяких геометричних об'єктів. Методами дослідження є: застосування апарату аналітичної геометрії, математичного аналізу, теорії R -функцій, паралельної архітектури та чисельних методів. Метою дослідження є вирішення актуальної науково-технічної проблеми, а саме: збільшити ефективність при проектуванні програми побудови дискретних моделей, за рахунок застосування паралельної архітектури. Дане програмне забезпечення може бути використано в скінченно-елементному аналізі складних технічних систем.

Для досягнення поставленої мети було поставлено наступні задачі: проаналізувати та розглянути відомі на сьогодні методи і підходи, які відносяться до побудови дискретних моделей в складних обчислювальних системах. Розробити відповідний метод та візуалізувати математичні моделі з використанням функціонального підходу. Модифікувати метод «Маршируючих кубів». Реалізувати в паралельних архітектурах візуалізацію дискретних моделей використавши сучасні технології і бібліотеки програмування, такі як, OpenMP та MPI. Проведення тестових експериментальних розрахунків, які доводять ефективність запропонованого даного алгоритму.

Враховуючи, що точність геометричних моделей важлива для безпеки складних технічних систем, застосування паралельних методів побудови дискретних моделей може суттєво вплинути на забезпечення надійності та безпеки при розробці і випробуванні високотехнологічних продуктів, зокрема в таких галузях, як ракетобудування. У процесі створення моделей, які використовуються для проектування ракетних комплексів і інших високотехнологічних систем, важливо досягти високої точності, швидкості обчислень та надійності результатів. Розроблені паралельні методи обчислень дозволяють знизити ризики помилок та оптимізувати процеси проектування, що є важливим для забезпечення безпеки в цій критичній сфері.

Ключові слова: MPI, OpenMP, R -функції, граничний елемент, метод “Маршируючі куби”, паралельні розрахунки, функціональний підхід.

Постановка проблеми. Існують сучасні обчислювальні методи, які замінюють вихідний неперервний об'єкт його дискретною моделлю. Дані об'єкти можна візуалізувати замкненими та обмеженими підмножинами деякого евклідового простору. Розробка дискретних моделей складних технічних об'єктів особливо актуальна в таких галузях: будівництво, суднобудівництво, ракетобудівництво, медицина та машинобудівництво.

На сьогодні процеси автоматизації призводять до необхідності розв'язку задач візуалізації геометричних моделей та розробки паралельної архітектури в машинобудуванні,

при вирішенні інженерних задач задля пришвидшення продуктивності обчислювальних систем та наступного супроводу в майбутньому програмного забезпечення. Актуальні задачі, які виникають перед інженерами та розробниками, це проведення випробувань та обчислювальних експериментів дослідних зразків складних систем шляхом заміщення фізичних експериментів на віртуальні.

Це дасть змогу зменшити витрати на натуральні експерименти, ресурси та час на проєктування. Наявні на даний час системи мають свої недоліки, особливо в часовій ефективності, тому потрібна реалізація відповідного програмного забезпечення з використанням паралельної архітектури та методу “Маршируючих кубів”. Використання геометричного моделювання дозволяє замінити дорогі дослідження та експерименти зразків продукції шляхом аналізу математичних відношень. Тому, дане дослідження є актуальним, так як моделювання геометричних об’єктів автоматизує проєктувальні роботи, та матиме більший вигаш, якщо збільшити його часову ефективність.

Аналіз досліджень і публікацій. Сьогодні існує безліч методів формального опису геометричних моделей складної форми описаних в роботах авторів, таких як: каркасне моделювання [1], твердотільне геометричне моделювання [1], [2], одним з найбільш ефективних та універсальних методів є функціональний підхід. Так як за основу використовується ідея застосування неявних функцій разом з R -функціями В.Л. Рвачова для проєктування складних форм [3]. Використання функціонального підходу та R -функції допомагають перетворити візуалізацію складної неявної функції на конструювання логіки із “легких блоків”, що в свою чергу зберігає всю виразність математичної мови, що є актуальним та необхідним в наш час. Недоліки застосування даних методів описані в роботі [4].

Теорія функцій, викладена В.Л. Рвачовим дозволяє аналітично описати границю будь-якого геометричного об’єкта [3]. Процес його опису зводиться до задання деякої неявної функції з координатними аргументами, які прийматимуть нульові значення на границі деякої області, додатні значення всередині області, та від’ємні за її межами. Тобто, незалежні і залежні аргументи даних функцій приймають визначені чисельні значення, які групуються в множини, та будуть характеризувати відповідні якісні характеристики даних функцій.

Тому, розглянемо на прикладі функції дійсних аргументів для якої знак величини значення функції встановлюватиметься знаками незалежних аргументів, а не абсолютним значенням аргументів [5]:

$$g_1 = x_1 x_2 x_3 \varphi(x_1 x_2 x_3),$$

де $\varphi(x_1, x_2, x_3) > 0, \forall (x_1, x_2, x_3);$

$$g_2 = x_1 + x_2 + x_3 + \sqrt{x_1^2 + x_2^2 + x_1^2 x_2^2}, \quad (1)$$

Для значення функції можна виписати деякі наступні залежності знаків аргументів і знаків функції зображених в таблиці 1:

Таблиця 1 – Залежність якості функції від якості аргументів

Варіанти знаків значень аргументів			Варіанти знаків для відповідних значень функцій			
x_1	x_2	x_3	g_1	g_2	g_3	g_4
—	—	—	—	—	+	—
—	—	+	—	—	+	+
—	+	—	+	+	+	+
—	+	+	+	+	+	+
+	—	—	+	+	+	+
+	—	+	—	+	+	+
+	+	—	—	+	+	+
+	+	+	+	+	+	+

Визначення множин R -функції обумовлено розглядом пов'язаних з даними функціями множин функції алгебри логіки. Існування даного зв'язку дозволяє застосовувати методи дискретної математики. Отже, булева функція, яка відповідає даній R -функції, називається функцією, яка супроводжуватиме R -функцію. Одна і так ж сама булева функція буде супроводжувати безкінечну кількість разів множини R -функцій [6].

Найбільш ефективно вирішує питання триангуляції метод, який працює з функціональним підходом – “Маршируючі куби” [7]. Проте, необхідна певна модифікація для паралельної архітектури, щоб апроксимація границі деякої кінцевої області, яка отримується із застосуванням даного алгоритму при невеликій кількості кроків, побудувала достатньо якісну геометричну модель.

Даний метод використовує підхід «розділяй і володарюй», для того щоб виділити поверхню в заданому логічному кубі, який створений з восьми пікселів у кожній з двох його суміжних сторін. Проектування можна розділити на два етапи: розбиття деякої області в просторі на дискретну множину комірок, виконання пошуку комірок, що перетинатимуться з шуканою поверхнею та його апроксимація у знайдених комірках. Дані підзадачі незалежні. Розглянемо їх детальніше.

На першому етапі можемо зіштовхнутися з двома проблемами: як розбити області на комірки, та яким чином обрати комірки, що перетинаються з поверхнею, яка шукається. Схема роботи методу “Маршируючих кубів” зображена на рис. 1.

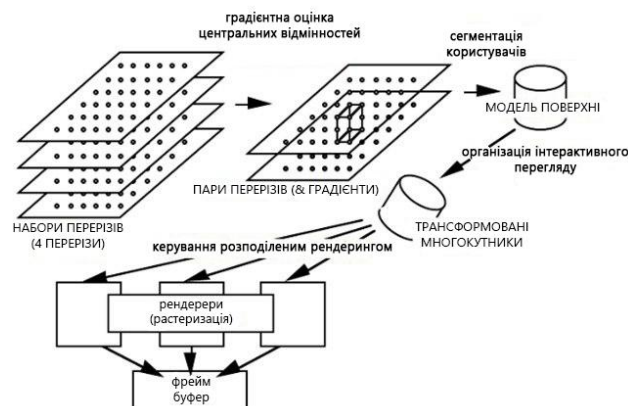


Рисунок 1 – Робота методу “Маршируючих кубів”

Як тільки область буде розбита на комірки, то значення функції, в загальних випадках, що задаватимуть поверхню, будуть відомі тільки у вершинах даних комірок. Тому, комірка буде головною структурною одиницею у всіх алгоритмах на даному етапі.

Можливі варіанти задання таблицею, яка основана на регулярній сітці, тоді проблема розбиття області на комірки відразу зникає, що й було використано в даному дослідженні, а отже зважаючи на однозначність рішення – дана комірка повинна бути паралелепіпедом – щоб мати значення функції у вершинах комірки. Також необхідно врахувати розмір такої комірки, якщо буде з великим, то можлива велика втрата точності. Так як при великому розмірі комірки, частини поверхні, що шукатимуться, не будуть видні. Розмір обирається не менше допустимої похибки візуалізації поверхні, що шукається.

Існуючі альтернативні варіанти даного методу описані в роботі [7]: “Маршируючі тетраєдри”, “Маршируючі октаєдри” та “Маршируючі призми” не суттєво відрізняються та мають майже однакову похибку, вибір полягає у формі триангуляції. Алгоритм “Маршируючих кубів” широко використовується з практичної точки зору, бо ймовірність появи помилок достатньо мала. Наприклад, при проектуванні тестових поверхонь топологічна неточність не виявилася жодного разу, а отже проаналізувавши дані методи, вибір зупиняється на даному алгоритмі.

Метою дослідження є вирішення актуальної науково-технічної проблеми, а саме: збільшити ефективність при проектуванні програми побудови дискретних моделей, за рахунок застосування паралельної архітектури. Дане програмне забезпечення може бути використано в скінченно-елементному аналізі складних технічних систем.

Виклад основного матеріалу. Об'єктом дослідження є процес розробки функціонального підходу для проектування складних геометричних моделей. Методами дослідження виступає застосування апарату аналітичної геометрії, математичного аналізу, теорії R -функцій, паралельної архітектури та обчисленнях.

Наведемо деякі ключові аспекти використання MPI:

- *Можливість паралельного обміну даними:* MPI надає потужні розроблені функції для обміну даними між вузлами. Це особливо корисно методах скінчених елементів (МСЕ), де завдання може бути розподілене на багато вузлів. При цьому MPI дозволяє ефективно обмінюватися даними між цими вузлами.

- *Розподіл завдань:* MPI дає можливість розподілити задачі між різними вузлами кластера або обчислювального середовища. Можливо створити програму, де кожен вузол виконає свої обчислення та обмін даними з наявними іншими вузлами за допомогою повідомлень.

- *Масштабованість:* дана бібліотека добре масштабується, це дозволяє використовувати його в МСЕ з великою кількістю вузлів та обчислювальною потужністю.

- *Синхронізація:* даний стандарт також надає засоби для синхронізації розрахунків між вузлами. Можливо використовувати різні типи синхронізації, наприклад, бар'єри або точки синхронізації, для координації обчислень на різних вузлах.

- *Профілювання та діагностика:* MPI має інструменти для діагностики та профілювання паралельних програм, це допоможе виявити проблеми та оптимізувати їх продуктивність.

- *Підтримка сучасних мов програмування:* MPI має бібліотеки, які підходять для таких мов програмування, як C/C++, Python та ін., що робить його напрочуд універсальним для різних можливих сценаріїв розробки.

А от використання стандарту зі спільною пам'яттю OpenMP у МСЕ дозволить спростити створення паралельних версій програм та покращує їх продуктивність. Це стандарт для багатопотокового проектування на мовах програмування C/C++ та Fortran. Починаючи з 2000-х років, OpenMP став популярним інструментом для реалізації паралельних програм, які можуть працювати на багатоядерних архітектурах та багатопроцесорних системах, і вирішують завдання з обчисленнями швидше. В МСЕ часто необхідно вирішувати громіздкі системи лінійних та нелінійних рівнянь, що вимагає ітераційних методів. Бібліотека OpenMP для паралельного виконання ітерацій використовує deal.II та libMesh при розпаралелюванні обчислень.

Стандарт OpenMP використовує велику кількість потоків, які не знищуються між паралельними і послідовними областями, а керування передається заданій кількості потоків в паралельній області, або головному потоку деякої послідовної області. Стала підтримка великої кількості потоків дає вигоду у ефективності, так як створення потоків у кожній паралельній частині є повільною операцією.

Загалом, розробка паралельної архітектури пропонує значні переваги для геометричного моделювання МСЕ, що включає покращену обчислювальну ефективність, можливість обробляти складніші та більші об'єкти та прискорену конвергенцію перехідних рішень. Дані досягнення вже були продемонстровані в різних областях, що підкреслює широку застосованість та потенціал паралельних реалізацій, вибір того чи іншого стандарту паралельної архітектури залежить від характеристик система та вимог до неї.

Для проведення обчислювального експерименту використовувалася комп'ютерна система з розподіленою пам'яттю з наступними характеристиками:

- два сервери Intel (R) Core i7-4770 CPU @ 3.40GHz, 8 ядер;
- графічний процесор Intel HD Graphics 520;
- операційна система Linux (Ubuntu версії 20.4).

Середовищем розробки виступає QT Creator версії 6.71, в якій реалізовано значну кількість бібліотек для розробки графічного інтерфейсу. Для спільної пам'яті використовувався один з серверів. Задля зручного запуску програмного забезпечення до графічного інтерфейсу була додана можливість перемикання між методами з розподіленою та загальною пам'яттю, що зображено на рис. 2.

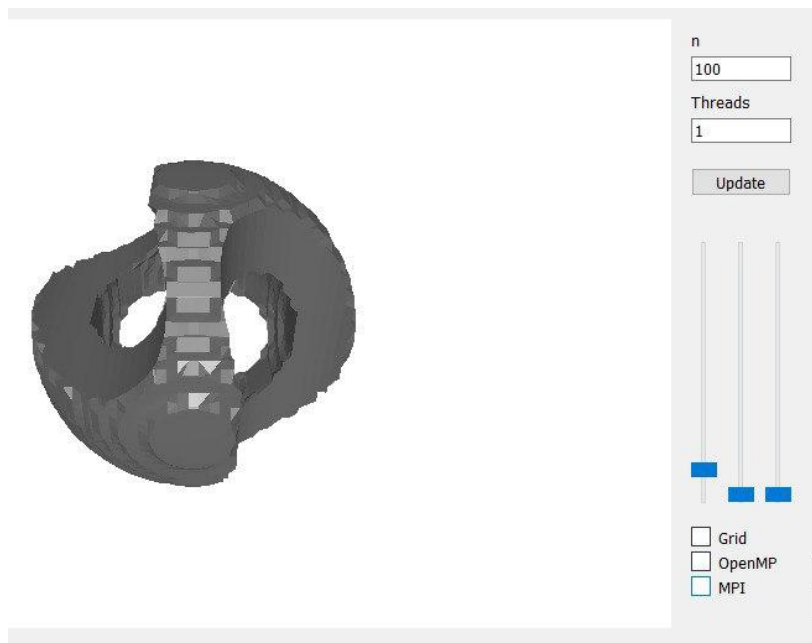


Рисунок 2 – Графічний інтерфейс застосунку

Як видно з рис. 3, задача проєктування розподіляється на декілька вузлів зі стандартом MPI. Кожен потік коду працює над різними розділами загального зображення паралельно з мінімальним відхиленням.

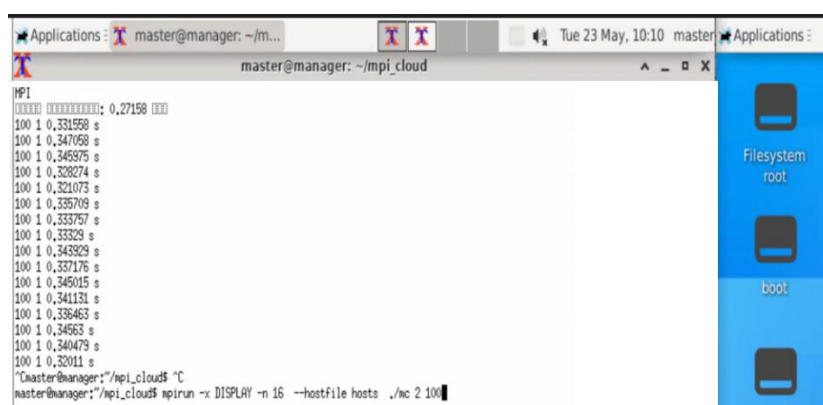


Рисунок 3 – Робота кластеру в розподіленій комп'ютерній системі

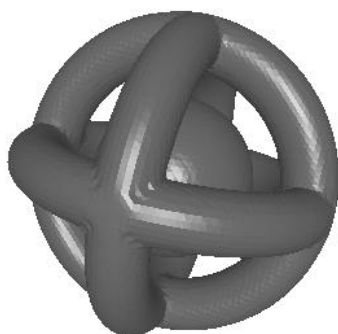
Задля демонстрації ефективності роботи в паралельній архітектурі з використанням бібліотек OpenMP, MPI та методу “Маршируючих кубів” розглянемо проєктування геометричного об’єкту “Пішак” (див. рис.4). Функціонально, модель описується наступним чином:

$$\begin{aligned}
G(x, y, z) &= \frac{7}{16}(\sqrt{x^2 + z^2} - 4)^2 - y^2 \\
G_2(x, y) &= 9 - x^2 - z^2 \\
G_3(y) &= y(7 - y) \\
G_4(x, y, z) &= 1 - x^2 - (7 - y)^2 - z^2 \\
G_5(x, z) &= 2 - x^2 - 9(6 - y)^2 - z^2 \\
P_8 &= (q_1(x, y, z) \wedge q_2(x, z) \wedge q_3(y) \vee q_4(x, y, z) \vee q_5(x, z))
\end{aligned} \tag{2}$$

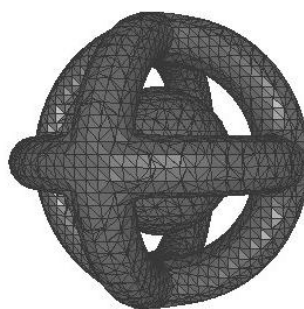
Рисунок 4 – Візуалізація функції P_8

Для ефективного та точного проведення обчислювального експерименту, функціонально було побудовано модель “Атом” (див. рис. 5) та “Скульптура” (див. рис. 6). Перша модель описана наступним чином:

$$\begin{aligned}
G_1(x, y, z) &= 10(\sqrt{x^2 + z^2} - x^2 - y^2 - z^2) - 24 \\
G_2(x, y, z) &= 10(\sqrt{y^2 + z^2} - x^2 - y^2 - z^2) - 24 \\
G_3(x, y, z) &= 10(\sqrt{x^2 + y^2} - x^2 - y^2 - z^2) - 24 \\
G_4(x, y, z) &= 9 - x^2 - y^2 - z^2 \\
G_5(x, y, z) &= q_1(x, y, z) + q_2(x, y, z) + \sqrt{q_1(x, y, z)^2 + q_2(x, y, z)^2} \\
G_6(x, y, z) &= q_3(x, y, z) + q_5(x, y, z) + \sqrt{q_3(x, y, z)^2 + q_5(x, y, z)^2} \\
P_9 &= q_4(x, y, z) + q_6(x, y, z) + \sqrt{q_4(x, y, z)^2 + q_6(x, y, z)^2}
\end{aligned} \tag{3}$$



а



б

Рисунок 5 – Візуалізація моделі P_9 :

а – дискретна модель без сітки; б – дискретна модель із сіткою

Дані математичні моделі проєктуються з різними значеннями параметру n , які можна спостерігати на рис. 6. Можемо бачити, що при невеликому значенні розбиття n можливо зробити деякі висновки з приводу форми геометричної моделі, також можемо зробити висновки щодо якості проєктованого об'єкта. Реалізація за допомогою функцій моделі “Скульптура” виглядає наступним чином:

$$G(x, y, z) = G_1(x, y, z) \wedge \neg G_2(x, y, z) \wedge \neg G_3(x, y, z), \quad (4)$$

де $G_1(x, y, z) = r_1^2 - (x - x_0)^2 - (y - y_0)^2 - (z - z_0)^2$ – сфера, що зовні;

$G_2(x, y, z) = r_2^2 - (x - x_0)^2 - (y - y_0)^2 - (z - z_0)^2$ – сфера, що всередині;

$G_3(x, y, z) = G_4(x, y, z) \vee G_5(x, y, z) \vee G_6(x, y, z)$ – об'єднання ортогональних циліндрів;

$G_4(x, y, z) = 1 - \left(\frac{y}{b}\right)^2 - \left(\frac{z}{c}\right)^2$ – ортогональний циліндр (перший);

$G_5(x, y, z) = 1 - \left(\frac{x - dx}{a}\right)^2 - \left(\frac{z - dz}{c}\right)^2$ – ортогональний циліндр (другий);

$G_6(x, y, z) = 1 - \left(\frac{x}{a}\right)^2 - \left(\frac{y - dy}{b}\right)^2$ – ортогональний циліндр (третій).

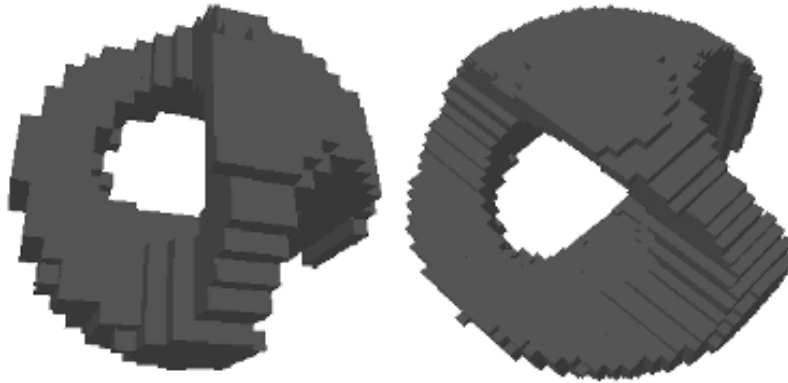


Рисунок 6 – Візуалізація моделі “Скульптура”

Потрібно зауважити, що при збільшених значеннях n , візуалізація воксельної моделі вимагатиме збільшених часових витрат, а також дещо збільшений обсяг пам'яті. Щоб оптимізувати даний алгоритм та вирішити проблему, потрібно виконати наступні кроки:

Реалізувати триангуляцію поверхні, яка базується на наявній тривимірній решітці M , тобто виконати перехід від об'єкта $X(\Omega)$ до гранично-елементного об'єкта $B(\Omega)$.

І вже наступним кроком застосовувати паралельні розрахунки. Для паралельних обчислень рівномірну ґратку простору M необхідно розділити на деяку кількість підґраток M_i , де $i = 1, k$, k – кількість потоків.

Модифікація методу “Маршируючих кубів” для паралельної архітектури необхідна, для того щоб апроксимація границі деякої кінцевої області, яка отримується із застосуванням даного алгоритму при невеликій кількості кроків, побудувала достатньо якісну поверхню геометричної моделі.

В тестуванні зпроєктованої автоматизованої системи паралельних алгоритмів було виконано візуалізацію описаних вище моделей.

Як залежить час роботи реалізованого алгоритму від кількості обчислювальних потоків при візуалізації моделей “Атом”, “Скульптура” та “Пішак” з розбиттям $n = 150$ можемо побачити на рис. 7-9.

Запрограмована на мові C++ дана модель в паралельних архітектурах показала такі результати:

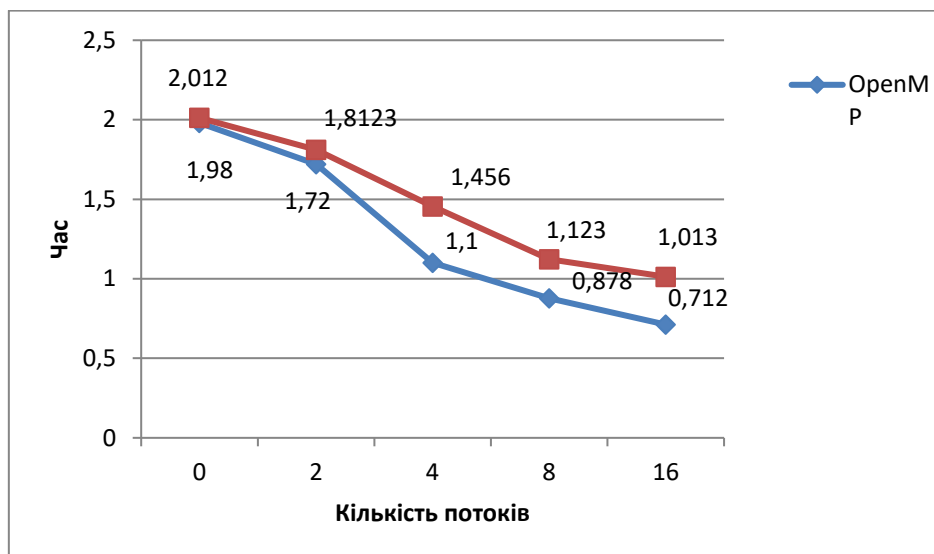


Рисунок 7 – Залежність часу роботи від кількості потоків моделі “Атом”

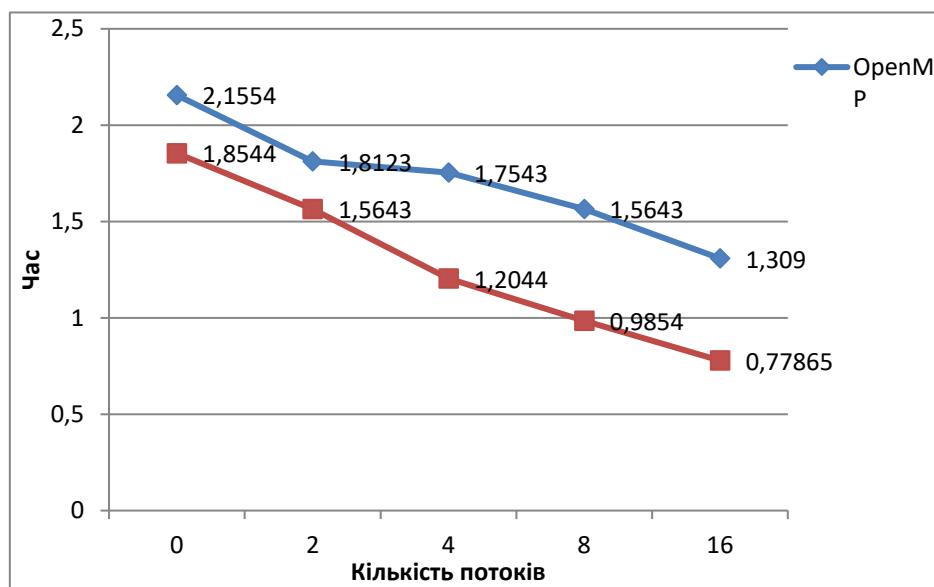


Рисунок 8 – Залежність часу роботи від кількості потоків моделі “Скульптура”

Проаналізувавши результати цих графіків, можемо побачити, що зі збільшенням кількості обчислювальних потоків, швидкість роботи в паралельній архітектурі пришвидшується в двох випадках, але робота в розподіленій системі має більшу ефективність. Це можна пояснити таким фактом, що в MPI скалярне поле розділяється на декілька частин, і виконується для кожної частини в окремих процесах.

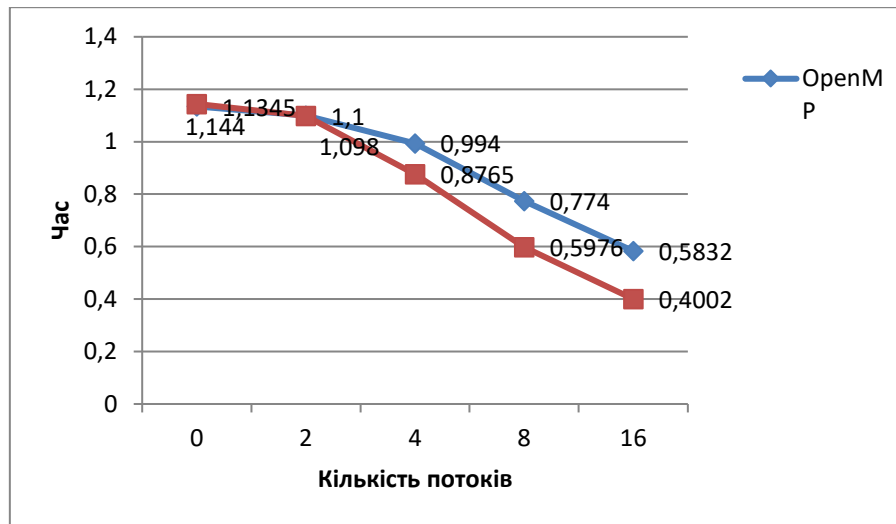


Рисунок 9 – Залежність часу роботи від кількості потоків моделі “Пішак”

Висновки. В рамках даного дослідження запропоновано алгоритми та підходи; досліджено альтернативні методи геометричного моделювання, та зроблено висновок, що застосування функціонального підходу є більш ефективним. Завдання перераховані в чіткій причинно-наслідковій послідовності із зазначенням результату виконання кожного з них, який, у свою чергу, служить вхідною інформацією для вирішення наступного завдання. Дослідження показує чіткий і логічний ланцюжок завдань від проєктування геометричних фігур та застосування методу “Маршрутуючих кубів” до практичної реалізації їх в паралельних архітектурах. Реалізація кожного із завдань ілюструється прикладом його вирішення. Пропоноване дослідження є продовженням пошуків щодо ефективних засобів автоматизації проєктувальних робіт. Перевагою запропонованого складу завдань, порівняно з роботами [1], [3], [8], є побудова більш якісних моделей, з найменшою похибкою, та наявною часовою ефективністю.

Підсумовуючи, можна зробити висновок, що на даний час актуальною є проблема побудови таких геометричних моделей та систем скінченно-елементного аналізу, які дозволяли б користувачеві формально описувати постановку широких класів задач математичної фізики і методів їх розв’язання, а з іншого, давали можливість враховувати особливості паралельної розподіленої архітектури сучасних комп’ютерних систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] M. Song, X. Zhang, and M. Lind, “Automatic identification of maintenance significant items in reliability centered maintenance analysis by using functional modeling and reasoning”, *Computers & Industrial Engineering*, vol. 182, pp. 1-19, 2023, doi: <https://doi.org/10.1016/j.cie.2023.109409>.
- [2] L. Amorim, T. Goveia, R. Mesquita, and I.J. Baratta, “GPU Finite Element Method Computation Strategy Without Mesh Coloring”, *Optoelectronics and Electromagnetic Applications*, vol. 19, no. 2, pp. 252-264, 2020, doi: <https://doi.org/10.1590/2179-10742020v19i2851>.
- [3] O. Hniedzovskyi, O. Kudin, Y. Belokon, D. Kruglyak, and S. Ilin, “Designing an object-oriented architecture for the finite element simulation of structural elements”, *Eastern-European Journal of Enterprise Technologies*, vol. 6, no. 2 (120), pp. 78-84, 2022, DOI: <https://doi.org/10.15587/1729-4061.2022.268018>.
- [4] Mathematical and computer modelling of engineering systems: Collective monograph,

V.S. Hudramovich, Eds. Riga, Latvia: Baltija Publishing, 2020.

- [5] A.F. Bulat, V.I. Dyrda, S.M. Grebenyuk, and M.I. Klymenko, "Numerical Simulation of Viscoelastic Deformation of Rubber Shock Absorbers Based on the Exponential Law", *Strength of Materials*, no. 54, iss. 5, pp. 776-784, 2022, doi: <https://doi.org/10.1007/s11223-022-00454-8>.
- [6] C. Suchocki, and S. Jemioło, "On Finite Element Implementation of Polyconvex Incompressible Hyperelasticity: Theory, Coding and Applications", *Int. J. of Computational Methods*, vol. 17, no 8, pp 1-36, 2020, doi: <https://doi.org/10.1142/S021987621950049X>.
- [7] A. Akoushideh, and A. Shahbahrani, "Performance Evaluation of Matrix-Matrix Multiplication using Parallel Programming Models on CPU Platforms". *PREPRINT (Vers. 1) available at Research Square*, pp 1-23, 2022, doi: <https://doi.org/10.21203/rs.3.rs-2135830/v1>.
- [8] А.В. Калюжняк, "Підвищення ефективності обчислень у розподілених комп'ютерних системах за допомогою бібліотеки MPI", *Прикладні питання математичного моделювання*, вип. 6, № 1, с. 66-73, 2023, doi: <https://doi.org/10.32782/mathematical-modelling/2023-6-1-7>.

Стаття надійшла до редакції 07.03.2025

REFERENCES

- [1] M. Song, X. Zhang, and M. Lind, "Automatic identification of maintenance significant items in reliability centered maintenance analysis by using functional modeling and reasoning", *Computers & Industrial Engineering*, vol. 182, pp. 1-19, 2023, doi: <https://doi.org/10.1016/j.cie.2023.109409>.
- [2] L. Amorim, T. Goveia, R. Mesquita, and I.J. Baratta, "GPU Finite Element Method Computation Strategy Without Mesh Coloring", *Optoelectronics and Electromagnetic Applications*, vol. 19, no. 2, pp. 252-264, 2020, doi: <https://doi.org/10.1590/2179-10742020v19i2851>.
- [3] O. Hniedzovskyi, O. Kudin, Y. Belokon, D. Kruglyak, and S. Ilin, "Designing an object-oriented architecture for the finite element simulation of structural elements", *Eastern-European Journal of Enterprise Technologies*, vol. 6, no. 2 (120), pp. 78-84, 2022, DOI: <https://doi.org/10.15587/1729-4061.2022.268018>.
- [4] Mathematical and computer modelling of engineering systems: Collective monograph, V.S. Hudramovich, Eds. Riga, Latvia: Baltija Publishing, 2020.
- [5] A.F. Bulat, V.I. Dyrda, S.M. Grebenyuk, and M.I. Klymenko, "Numerical Simulation of Viscoelastic Deformation of Rubber Shock Absorbers Based on the Exponential Law", *Strength of Materials*, no. 54, iss. 5, pp. 776-784, 2022, doi: <https://doi.org/10.1007/s11223-022-00454-8>.
- [6] C. Suchocki, and S. Jemioło, "On Finite Element Implementation of Polyconvex Incompressible Hyperelasticity: Theory, Coding and Applications", *Int. J. of Computational Methods*, vol. 17, no 8, pp 1-36, 2020, doi: <https://doi.org/10.1142/S021987621950049X>.
- [7] A. Akoushideh, and A. Shahbahrani, "Performance Evaluation of Matrix-Matrix Multiplication using Parallel Programming Models on CPU Platforms". *PREPRINT (Vers. 1) available at Research Square*, pp 1-23, 2022, doi: <https://doi.org/10.21203/rs.3.rs-2135830/v1>.
- [8] A.V. Kaliuzhniak, "Computation efficiency enhancement in distributed computer systems with the MPI library assistance", *Applied problems of mathematical modeling*, vol. 6, no. 1, pp. 66-73, 2023, doi: <https://doi.org/10.32782/mathematical-modelling/2023-6-1-7>.

ANASTASIIA KALIUZHNIAK,
OLEKSII KUDIN

DISCRETE MODELS OF GEOMETRIC OBJECTS IN PARALLEL COMPUTING SYSTEMS

The usage of powerful computer systems defines computational experiments as novel and significant research methods enabling to solve complex. Taking into account the fact that traditional analytical methods for visualizing mathematical models are of a determinative nature, there is still a need to apply modern mathematical theories which in their turn will enable to expand the possibilities of applied mathematical research. The object of the research is the process of developing functional design of complex geometric models. The subject of the research is the usage of parallel methods for constructing the surfaces of discrete models of some geometric objects. The methods of the research are: application of the apparatus of analytical geometry, mathematical analysis, the theory of R-functions, parallel architecture and numerical methods. The purpose of the research is to solve current scientific and technical problems, in particular to increase the efficiency in the design of the programme for building discrete models, which can be implemented in the finite element analysis of complex technical systems by means of using parallel architecture.

The following tasks were set to achieve the goal: analysis and review of currently known methods and approaches related to the construction of discrete models in complex computing systems. Development of the appropriate method and visualization of mathematical models based on the functional approach. Modification of the "Marching Cubes" method. Implementation in parallel architectures resorting to modern technologies and programming libraries, such as OpenMP and MPI, and conducting test experimental calculations that prove the efficiency of the proposed algorithm.

Considering the importance of geometric model accuracy for the safety of complex technical systems, the application of parallel methods for building discrete models can significantly impact the reliability and safety in the development and testing of high-tech products, particularly in fields such as rocket engineering. In the process of creating models used for designing rocket systems and other advanced technologies, it is crucial to achieve high precision, computational speed, and reliability of results. The developed parallel computation methods help reduce the risk of errors and optimize the design processes, which is vital for ensuring safety in this critical area.

Keywords: MPI, OpenMP, R-functions, boundary element, method "Marching cubes", parallel calculations, functional representation.

Калюжняк Анастасія Вікторівна, PhD, доцент кафедри комп'ютерних наук, Запорізький національний університет, Запоріжжя, Україна, ORCID 0000-0002-4837-7566, anastasia.korgun@gmail.com.

Кудін Олексій Володимирович, кандидат фізико-математичних наук, професор кафедри програмної інженерії, Запорізький національний університет, Запоріжжя, Україна, ORCID 0000-0002-5917-9127, avk256@gmail.com.

Kaliuzhniak Anastasiia, PhD, Associate Professor at the Department of Computer Science, Zaporizhzhia National University, Zaporizhzhia, Ukraine.

Kudin Oleksii, PhD, Professor at the Department of Software Engineering, Zaporizhzhia National University, Zaporizhzhia, Ukraine.