

УДК 519.63

# СПОСІБ РОЗПАРАЛЕЛЮВАННЯ ОБЧИСЛЕНЬ ПРИ МОДЕЛЮВАННІ ТЕПЛОВИХ ПОЛІВ З ВИКОРИСТАННЯМ АДАПТИВНИХ СІТОК

**С. О. Лук'яненко**

Доктор технічних наук, професор,  
завідуючий кафедри\*

E-mail: lukian@aprodos.kpi.ua

**І. Ю. Михайлова**

Аспірант\*

E-mail: imikh@aprodos.kpi.ua

**Т. А. Лемкін\***

E-mail: taraslemkin@gmail.com

\*Кафедра автоматизації проектування  
енергетичних процесів та систем

Національний технічний університет України  
«Київський політехнічний інститут»  
пр. Перемоги, 37, м. Київ, Україна, 03056

*У статті запропоновано спосіб організації паралельних обчислень при моделюванні процесу лазерного опромінення пластини, що описується нестационарним тривимірним рівнянням теплопровідності. Використано метод розщеплення по координатах з адаптивною сіткою. На базі отриманих результатів зроблено висновок про ефективність проведеного розпаралелювання розрахунку на адаптивній сітці*

*Ключові слова: паралельні обчислення, адаптивні числові методи, метод розщеплення по координатах*

*В статье предлагается способ организации параллельных вычислений при моделировании процесса лазерного облучения пластины, который описывается нестационарным трехмерным уравнением теплопроводности. Использован метод расщепления по координатам с адаптивной сеткой. На основе полученных результатов сделан вывод об эффективности проведенного распараллеливания расчета на адаптивной сетке*

*Ключевые слова: параллельные вычисления, адаптивные численные методы, метод расщепления по координатам*

## 1. Вступ

Зважаючи на підвищення вимог до продуктивності комп'ютерних систем, поширення процесорів нового покоління, що мають декілька ядер, та необхідність вивчення процесів у режимі реального часу постає проблема прискорення виконання числових розрахунків, що направлені на комп'ютерне моделювання цих процесів. Таким чином, актуальною є задача якомога повнішого використання обчислювальних можливостей багатоядерних процесорів, що дозволяють виконувати розрахунки у декілька потоків у паралельному режимі. При цьому операційна система, виконуючи команди розробника програмного забезпечення, розподіляє програмні потоки по різних ядрах. Завдяки цьому з'являється можливість значно збільшити продуктивність обчислень. Це, у свою чергу, дозволяє скоротити час розрахунку не втрачаючи точність.

У роботі розглядається алгоритм розпаралелювання розрахунків у методі розщеплення по координатах за використання адаптивної сітки за допомогою звернень до Windows API.

## 2. Постановка завдання

Для дослідження ефективності паралельного алгоритму використаємо задачу моделювання темпера-

турного поля пластини під впливом лазерного опромінення. Математичною моделлю фізичного процесу розподілу температурного поля на поверхні та всередині пластини є тривимірне рівняння теплопровідності [1].

Для розв'язування задачі використаємо метод скінчених різниць [2]. На першому етапі для дискретизації області визначення застосуємо алгоритм побудови адаптивної сітки [3], у якому кожен часовий крок для оцінки похибки виконується п'ять разів із різними значеннями кроків [4]. На другому етапі методу скінчених різниць для переходу від диференціального рівняння в частинних похідних (ДРЧП) до систем лінійних алгебричних рівнянь (СЛАР) використаємо метод розщеплення по координатах [5]. Згідно нього, кожен часовий крок складається з шести етапів для знаходження розв'язку на наступному часовому шарі. Результат виконання кожного наступного етапу залежить від результату, отриманого на попередньому.

Необхідно побудувати такий алгоритм розпаралелювання розрахунків, який зменшує час виконання обчислень за рахунок відокремлення незалежних етапів розрахунку для виконання в окремих потоках.

## 3. Теоретичне обґрунтування

Введемо величину, що оцінює у скільки разів зміниться час розрахунку задачі при використанні

паралельного алгоритму ( $T_{par}$ ) відносно до однопоточного алгоритму ( $T_{oth}$ ). Назвемо її коефіцієнтом ефективності розпаралелювання і позначимо  $\eta$ .

$$\eta = \frac{T_{par}}{T_{oth}}$$

Очевидно, з підвищенням ефективності  $\eta$  зменшується. Нехай  $n$  – кількість задач,  $m$  – кількість доступних потоків,  $t_i$  – час, потрібний для виконання  $i$ -тої задачі. У послідовному режимі час виконання усіх задач дорівнює  $\sum_{i=1}^n t_i$ . Припустимо, що задачі створюються одночасно або різниця між створенням задач

набагато менша ніж час виконання кожної з них і нею можна знехтувати. У випадку, коли  $n \leq m$ , час виконання усіх задач у паралельному режимі буде визначатись як максимальний час, потрібний для виконання  $i$ -тої задачі, тобто  $\max_{i=1..n}(t_i)$ . Очевидно, що  $\max_{i=1..n}(t_i) < \sum_{i=1}^n t_i$ .

Тоді коефіцієнт ефективності розпаралелювання  $\eta = \frac{\max_{i=1..n}(t_i)}{\sum_{i=1}^n t_i} < 1$ .

Розглянемо випадок, коли  $n > m$ . Сформуємо вектори  $\vec{A}$  та  $\vec{B}$  таким чином, що вектор  $\vec{A}$  міститиме інтервали часу, потрібні для виконання кожної з перших  $m$  задач, вектор  $\vec{B}$  – решти  $(n - m)$  задач. Позначимо  $a_{min} = \min(A[i]), i=1..m$ ,  $\vec{A}'$  – вектор  $\vec{A}$ , без елемента  $a_{min}$ ,  $b$  – час виконання задачі, що буде поставлена на виконання при знаходженні вільного потоку, вектор  $\vec{B}'$  – вектор  $\vec{B}$  без елемента  $b$ , вектор  $\vec{A}$  – вектор, що отриманий з елементів вектору  $\vec{A}'$ :

- 1) відніманням від кожного елемента значення  $a_{min}$ ;
- 2) додаванням  $m$  у вектор значення  $b$ .

Коли вектор  $\vec{B}$  не містить жодного елемента, задача приймає вигляд, розглянутий вище, а час, потрібний для завершення всіх задач рівний значенню максимального елемента вектора  $\vec{A}$ . Для розрахунку часу, потрібного для виконання у паралельних потоках усіх задач, можна сформулювати наступну рекурсивну формулу:

$$T(\vec{A}, \vec{B}) = a_{min} + T(\vec{A}', \vec{B}),$$

$$T(\vec{A}, \emptyset) = \max_{i=1..m}(a_i).$$

Очевидно, що в даному випадку за умови багатопоточності ( $m > 1$ ) хоча б один раз виконається вибір значення  $a_{min}$ . Отже, можна стверджувати, що  $T_{par} < T_{oth}$ , звідки випливає, що  $\eta < 1$ , а отже розпаралелювання розрахунків дає вигравш у часі.

#### 4. Алгоритм розпаралелювання

Для реалізації розпаралелювання роботи програми моделювання необхідно розробити паралельний алгоритм роботи методу в цілому. Виконання розрахунку для одного шару з певними параметрами адаптивної сітки будемо вважати атомарною задачею, тобто такою, що неможливо розділити на паралельні підпроцеси,

оскільки, як згадувалось вище, цей процес складається з шести етапів, кожен з яких залежить від попереднього.

На противагу цьому, для оцінки похибки та автоматичного підбору просторових та часового кроків у алгоритмі побудови адаптивної сітки передбачено п'ять незалежних етапів розв'язування ДРЧП методом розщеплення по координатах з різними кроками сіток. Тому доцільно виконувати ці обчислення в паралельних потоках. З іншого боку, після проведення цих обчислень потоки необхідно синхронізувати, оскільки обчислення коефіцієнтів для зміни кроків проводиться на основі результатів додаткових розрахунків, що виконуються паралельно. Хоча синхронізація потоків і не є оптимальною операцією з точки зору розпаралелювання обчислень, вона, по-перше, є необхідною, а по-друге, при виконанні обчислень у більш ніж одному потоці, максимальні сумарні витрати часу на синхронізацію завжди менші або рівні витратам часу на виконання обчислень в одному потоці, оскільки наступна задача в черзі чекатиме не повний час виконання попередніх задач, а той, що виконається швидше.

Алгоритм розпаралелювання процесу розв'язування задачі представлений на рис. 1.

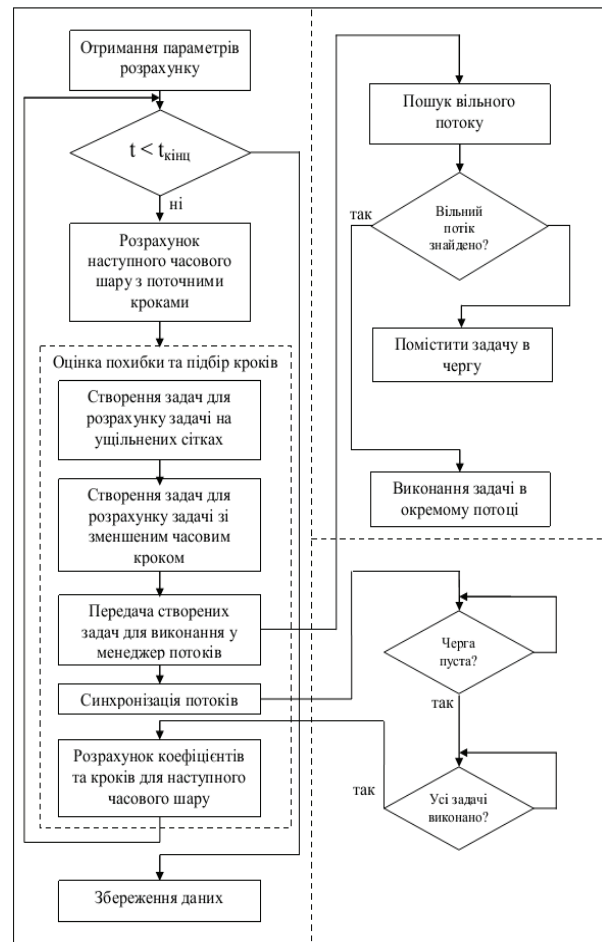


Рис. 1. Схема паралельного алгоритму

#### 5. Програмна реалізація

Програмний продукт було розроблено на мові програмування Delphi з використанням системних

звернень до Windows API [6 – 10]. Такий підхід дає можливість передачі на виконання у окремому потоці будь-яких методів та функцій. Єдиним обмеженням, що накладається, є подібність сигнатур методів, а саме – кожен метод повинен приймати лише один вказівник на параметр, що може представляти собою певну структуру даних. У цій структурі даних зберігається вказівник на процедуру/функцію, яка повинна виконуватися у багатопотоковому режимі та вказівники на параметри, необхідні для роботи цієї процедури/функції. В даному випадку передається вказівник на функцію, що реалізує перехід з одного часового шару на інший. А в параметрах записані параметри лазерного променя та температурний розподіл у даний момент часу на необхідній для даного розрахунку сітці.

Розроблено модуль автоматичного керування потоками, що надає користувачу наступні можливості:

1. постановки задачі в чергу на виконання;
2. зняття задачі з черги;
3. перевірку стану виконання задачі;
4. виконання синхронізації всіх потоків, або певної групи потоків;
5. отримання списку виконаних задач на поточний момент часу.

Якщо при постановці завдання на виконання керуючий модуль не може знайти вільного потоку для виконання, він розміщує задачу у кінець черги. Задачі з цієї черги можуть бути або скасовані користувачем або автоматично поставлені на виконання у порядку надходження при звільненні одного з потоків.

Після закінчення розв'язання задачі, протокол виконаних задач записується у файл для подальшого аналізу. Цей файл містить інформацію про те, в якому з потоків виконувалась задача, моменти часу постановки її в чергу, початку та завершення виконання задачі, та іншу метаінформацію, потрібну для проведення аналізу ефективності застосування розпаралелювання.

### 6. Аналіз проведених експериментів

Експериментальні обчислення проводились на ЕОМ з чотирьохядерним процесором AMD Phenom II X4 з тактовою частотою 3,2 ГГц. Під час проведення експерименту збирались наступні дані:

1. загальний час розрахунку;
2. сумарний час виконання частини задачі, що може бути розпаралелена для однопоточного випадку;
3. сумарний час виконання паралельної частини для багатопоточного випадку;
4. середня завантаженість процесора.

Експерименти проводились із різним значенням допустимої похибки для однакової задачі моделювання. Значення похибки вказані в табл. 1. Кожен експеримент виконувався двічі: для однопоточного та багатопоточного режиму.

Після того, як дані про виконання були зібрані, було проведено аналіз та розрахунок наступних параметрів:

1. коефіцієнт ефективності розпаралелювання  $\eta$ ;
2. коефіцієнт паралельного пришвидшення  $1/\eta$ ;
3. коефіцієнт загального пришвидшення розрахунку (відношення загального часу виконання в однопоточному режимі до цього ж часу із застосуванням мультиточних обчислень).

Результуючі дані відображені в табл. 2.

Таблиця 1

### Значення допустимих похибок та розмірності СЛАР

№ експерименту	Значення допустимої похибки	Кількість часових кроків	Середня розмірність СЛАР
1	0,1	23	25
2	0,05	40	28
3	0,025	84	28
4	0,01	208	29

Таблиця 2

### Зведені дані проведення експериментів

№	К-сть потоків	Завантаженість процесора (%)	Загальний час виконання, с	Час виконання паралельних операцій, с	$\eta$	$1/\eta$	Загальне пришвидшення
1	1	27	54	14,389	0,653	1,533	1,102
	4	94	49	9,391			
2	1	26	147	45,071	0,601	1,665	1,147
	4	98	129	27,069			
3	1	29	260	80,552	0,591	1,692	1,15
	4	91	227	47,586			
4	1	25	325	100,045	0,59	1,694	1,144
	4	93	284	59,103			

Зі зменшенням значення допустимої похибки збільшується розмірність СЛАР, які потрібно розв'язувати на кожному етапі методу розщеплення, а отже збільшується і час розрахунку. З табл. 2 видно, що коефіцієнт ефективності розпаралелювання в останніх трьох експериментах майже однаковий, але відрізняється від першого випадку. Це можна пояснити тим, що зі збільшенням кількості однотипних операцій (експерименти 2-4, розмірність масивів більша) процесор виконує ці операції швидше в порівнянні з випадком, коли операції змінюються порівняно часто (експеримент 1, розмірність масивів менша), оскільки в такому разі витрачається час на передачу іншого потоку команд на виконання і ці потоки частіше змінюються.

З табл. 2 також видно, що найменше загальне пришвидшення було досягнуто у 1 експерименті. Це можна пояснити тим, що на початку роботи програми виконуються порівняно ресурсномісткі операції заповнення початкових значень усіх масивів та структур, що впливає на загальний час роботи програми. Оскільки при цьому кількість кроків є найменшою, то розпаралелювання не може нівелювати витрачений на початку роботи програми час.

### 7. Висновки

У результаті проведених досліджень було розроблено паралельний алгоритм моделювання, створено програмний продукт, що реалізовує цей алгоритм,

проведено експерименти, аналіз результатів яких показав, що при розпаралелюванні досягається пришвидшення розрахунку. Зокрема етап обчислення похибки та прийняття кроків для наступного часового шару скорочується щонайменше у 1,5 рази. Загальний

час моделювання зменшується всього у 1,15 рази, що пояснюється тим, що значну частину процесорного часу займає формування масивів для подальшого відображення результатів, які записуються на жорсткий диск.

---

#### Література

1. Тихонов, А. Н. Уравнения математической физики [Текст] / А. Н. Тихонов, А. А. Самарский – 5-е изд. – М. : Наука, 1977. – 735 с.
2. Головка, Л. Ф. Моделирование процесса бесконтактной лазерной деформации адаптивным методом [Текст] / Л. Ф. Головка, С. О. Лукьяненко, И. Ю. Михайлова, В. А. Третьяк // Электронное моделирование / Институт проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины. – 2011. – Вып. 3, Том 33. – С. 71–84.
3. Лук'яненко, С.О. Адаптивні обчислювальні методи моделювання об'єктів з розподіленими параметрами [Текст] / С.О. Лук'яненко – К. : ІВЦ «Видавництво «Політехніка», 2004. – 236 с.
4. Лазерні технології та комп'ютерне моделювання : монографія / За ред. Л. Ф. Головка, С. О. Лук'яненка. – К. : Вістка, 2009. – 296 с.
5. Марчук, Г. И. Методы расщепления [Текст] / Г. И. Марчук – М. : Наука. Гл. ред. физ.-мат. лит., 1988. – 264 с.
6. Process and Thread Functions (Windows) [Virtual Resource] Access Mode : URL:<http://msdn.microsoft.com/en-us/library/windows/desktop/ms684847.aspx> – Title from Screen. – Date of Access: 18 March 2013.
7. Task Parallelism (Task Parallel Library) [Virtual Resource] Access Mode : URL: <http://msdn.microsoft.com/en-us/library/dd537609.aspx> – Title from Screen. – Date of Access: 18 March 2013.
8. Synchronization Functions (Windows) [Virtual Resource] Access Mode : URL: <http://msdn.microsoft.com/en-us/library/windows/desktop/ms686360.aspx> – Title from Screen. – Date of Access: 18 March 2013.
9. Grama, A. Introduction to Parallel Computing [Text] / A. Grama – Pearson Education, 2003. – 636 p.
10. Heroux, M. A. Parallel Processing for Scientific Computing [Text] / M. A. Heroux, P. Raghavan, H. D. Simon - Software, Environments and Tools, 2006. – 397 p.