МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В. Н. КАРАЗІНА
НАУКОВО-НАВЧАЛЬНИЙ ІНСТИТУТ
КОМП'ЮТЕРНОЇ ФІЗИКИ та ЕНЕРГЕТИКИ
Кафедра комп'ютерної фізики

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи
за освітньо-кваліфікаційним рівнем магістр

«Використання багатоядерних обчислювальних систем
в моделюванні випадкових процесів»

Виконав: студент 2 курсу, групи НФ-64
Самойленко А.О.
спеціальність:
105 Прикладна фізика та наноматеріали
освітня програма: Комп'ютерна фізика
Керівник:
кандидат фізико-математичних наук:
Рогова С.Ю.
Рецензент:

Харків – 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В.Н. Каразіна

Навчально-науковий інститут комп'ютерної фізики та енергетики
Кафедра комп'ютерної фізики
Рівень вищої освіти (освітньо-кваліфікаційний рівень) магістр
Спеціальність: 105 Прикладна фізика та наноматеріали
Освітня програма: Комп'ютерна фізика

ЗАТВЕРДЖУЮ

Завідувач кафедри
_____ Костянтин НЄМЧЕНКО
підпис

"_____" _____ 2020 року

ЗАВДАННЯ
на дипломну роботу магістра студентові

Самойленко Антон Олександрович

1. Тема роботи «Використання багатоядерних обчислювальних систем в моделюванні випадкових процесів»

керівник роботи Рогова Світлана Юріївна

затверджені наказом по університету від 06 грудня 2020 № 0210 – 05/2267

2. Строк подання студентом роботи 12 травня 2020 р.

3. Перелік питань, які потрібно розглянути:

Розробити оригінальні методі використання багатоядерних обчислювальних систем в моделюванні випадкових процесів. Зокрема, провести моделювання процесів переносу тіла в одновимірних наноструктурах – нанодротах – з метою визначення теплових властивостей цих структур. Вирішення цього завдання дозволяє оптимізувати процеси енергоспоживання в сучасних пристроях наноелектроніки. Розробити обчислювальні алгоритми, необхідне

програмне забезпечення та отримати результати, які можна буде використати в дослідженнях реальних матеріалів.

4.План роботи

№ з/п	Назви етапів роботи	
1	Вибір та затвердження теми	
2	Робота над літературою та іншими джерелами	
3	Написання вступу до роботи та першого розділу	
4	Написання другого розділу	
5	Написання третього розділу	
6	Завершення написання дипломної роботи(висновки и література)	
7	Подача науковому керівнику, доробка зауважень	
8	Відгук, рецензія, підготовка до захисту	
9	Попередній захист на кафедрі	
10	Захист	

5. Дата видачі завдання « »_____2020 р

Студент _____ Антон САМОЙЛЕНКО

підпис

Керівник роботи _____ Світлана РОГОВА

підпис

РЕФЕРАТ

Антон Самойленко «Використання багатоядерних обчислювальних систем в моделюванні випадкових процесів»
Кваліфікаційна робота магістра з прикладної фізики. – Х.: ХНУ імені В. Н.Каразіна, 2020. – 82 с.

В роботі проведено моделювання теплового переносу квазічастинок в одновимірних наноструктурах. Вирішення цього завдання дозволило визначити коефіцієнт теплопровідності одновимірних наноматеріалів – нанодротів і нанотрубок, визначити вплив на цей коефіцієнт температури і кількості домішок, за рахунок розсіяння або поглинання.

Моделювання руху частинок і їх взаємодії проводилося методом Монте-Карло, а для прискорення великої кількості розрахунків застосовувалися методи паралельних обчислень з використанням технології CUDA.

У дипломі розроблено обчислювальні алгоритми, необхідне програмне забезпечення та отримані результати, які будуть використані для пояснення спостережуваних теплових властивостей наноречовин.

КЛЮЧОВІ СЛОВА: НАНОДРОТИ, НАНОТРУБКИ, ФОНОНИ, ТЕПЛОВИЙ ПЕРЕНОС, ТЕПЛОПРОВІДНІСТЬ, МЕТОД МОНТЕ-КАРЛО

РЕФЕРАТ

Антон Самойленко «Использование многоядерных вычислительных систем в моделировании случайных процессов»
Квалификационная работа магистра прикладной физики. – Х.: ХНУ имени В. Н.Каразина, 2020. – 82 с.

В работе проведено моделирование теплового переноса квазичастиц в одномерных наноструктурах. Решение этой задачи позволяет определить коэффициент теплопроводности одномерных наноматериалов – нанопроволок и нанотрубок, определить влияние на этот коэффициент температуры и количества рассеивающих и поглощающих центров.

Моделирование движения частиц и их взаимодействия проводилось методом Монте-Карло, а для ускорения большого количества вычислений применялись методы распараллеливания с использованием технологии CUDA.

В дипломе разработаны вычислительные алгоритмы, необходимое программное обеспечение и получены результаты, которые будут использованы для объяснения наблюдаемых теплопроводящих свойств нановеществ.

КЛЮЧЕВЫЕ СЛОВА: НАНОПРОВОЛОКИ, НАНОТРУБКИ, ФОНОНЫ, ТЕПЛОВОЙ ПЕРЕНОС, ТЕПЛОПРОВОДНОСТЬ, МЕТОД МОНТЕ–КАРЛО

ABSTRACT

Anton Samoilenko “The use of multi-core processors computing systems in modeling of random processes”
Qualification work of the Master of Applied Physics. – Kharkiv: V.N. Karazin Kharkiv national university, 2020. – 82 p.

In this work, we simulate the thermal transport of quasiparticles in one-dimensional nanostructures. The solution to this problem allows one to determine the thermal conductivity coefficient of one-dimensional nanomaterials – nanowires and nanotubes, and to determine the effect of temperature and the number of scattering and absorbing centers on this coefficient.

The motion of particles and their interactions were simulated by the Monte Carlo method, and parallelization methods using CUDA technology were used to accelerate the large number of calculations.

The diploma has developed computational algorithms, the necessary software, and obtained results that will be used to explain the observed heat-conducting properties of nanoparticles.

KEYWORDS: NANOWIRES, NANOTUBES, PHONONES, HEAT TRANSPORT, HEAT CONDUCTIVITY, MONTE CARLO METHOD,

ЗМІСТ

ВСТУП……... 9
РОЗДІЛ 1. ВИКОРИСТАННЯ БАГАТОЯДЕРНИХ ОБЧИЛЮВАЛЬНИХ СИСТЕМ ТА ПАРАЛЛЕЛЬНИХ ОБЧИСЛЕНЬ В ФІЗИЦІ (АНАЛІТИЧНИЙ ОГЛЯД)…….. 11
1.1 Концепція універсального обчислювального пристрою CUDA…… 11
1.2. Пам’ять в CUDA…… 15
1.3. Бібліотека cuRand для генератора випадкових чисел…… 18
1.4 Метод розкладання по матрицях Гелл-Мана……... 23
РОЗДІЛ 2. ФІЗИЧНІ ОСОБЛИВОСТІ ТЕПЛОПРОВІДНІСТНИХ ВЛАСТИВОСТЕЙ НАНОДРОТІВ……. 31
2.1 Вплив розміру зразка на теплопровідність…….. 32
2.2. Тепловий менеджмент…….. 38
2.3. Розгляд одновимірної моделі……... 41
РОЗДІЛ 3. МОДЕЛЮВАННЯ ПЕРЕНЕСЕННЯ ТЕПЛА ФОНОНАМИ В НАНОДРОТАХ…… 50
3.1. Вплив довжини вільного пробігу фононі на теплопровідність…… 50
3.2 Програмна реалізація та її результати…….. 61
ВИСНОВКИ…….. 70
ПЕРЕЛІК ПОСИЛАНЬ…… 71
ДОДАТОК А……. 75
ДОДАТОК Б……. 76
ДОДАТОК В……. 77
ДОДАТОК Г……. 78
ДОДАТОК Ґ…….. 79

ПЕРЕЛІК СКОРОЧЕНЬ ТА АБРЕВІАТУР

1D – одновимірний;

NW – нанодріт;

NT – нанотрубка;

GPU – графічний процесор (англ. Graphics Processing Unit);

CPU – центральний процесор (англ. Central Processing Unit);

CUDA – програмно-апаратна архітектура паралельних обчислень

(Англ. Compute Unified Device Architecture);

SM – потоковий багатопроцесор;

DRAM – динамічна пам’ять з довільним доступом;

MFP – довжина вільного пробігу;

MD – молекулярна динаміка;

SVR – співвідношення поверхні та об’єму;

SC – напівпровідники;

RT – кімнатна температура.

ВСТУП

Актуальність теми. З удосконаленням технології виробництва та розробки наноструктур, в останні десятиріччя були розроблені та синтезовані одновимірні (1D) матеріали, такі як нанодроти (NW), нанотрубки (NT) та квантові дроти [1, 2]. Завдяки своїм унікальним властивостям, які істотно відрізняються від об'ємних речовин, 1D матеріали викликали зростаючий науковий інтерес [1,3,4] і мають численні застосування в галузях електронної, оптико-електронної промисловості та в пристроях перетворення енергії, зокрема, в термоелектричних та фотовольтаїчних.

До таких одновимірних матеріалів, зокрема, відносяться кристалічні напівпровідникові нанодроти – перспективний матеріал, який використовують в мініатюрній електронній техніці аж до наномасштабів. Широко використовуються також і металеві нанодроти, які знайшли застосування в важливих технологічних додатках для передачі тепла, квантового переносу, або, наприклад, при створенні матеріалів з ефектом пам'яті форми.

Таким чином, розв'язання задач про визначення коефіцієнта теплопровідності одновимірних наноструктур є необхідною умовою для подальшого розвитку нанотехнологій та їх використання в техніці, зокрема, перетворення енергії.

Для досягнення цієї мети при виконанні дипломної роботи були визначені **задачі**:

– проаналізувати сучасний стан і шляхи розвитку нанотехнологій;

– проаналізувати особливості технологій, що використовуються для паралельних обчислень;

– розробити алгоритм, що реалізує модель руху фононів в одновимірному зразку з урахуванням випадкових розсіювань або поглинань;

– застосувати розроблені алгоритми для визначення коефіцієнта теплопровідності одновимірних систем, зокрема, його залежності від параметрів задачі.

Об’єкт дослідження – теплові властивості наноречовин.

Предмет дослідження – паралельні обчислення для моделювання процесів тепло перенесення за рахунок фононів в одновимірних наноструктурах.

Методи дослідження. Теоретичні положення роботи базуються на математичних методах теорії ймовірності та випадкових процесів.

Наукова новизна одержаних результатів.

– розроблено новий алгоритм паралельних обчислень випадкових процесів розсіювання фононів в одновимірних наноструктурах;

– визначені залежності коефіцієнта фононної теплопровідності нанодротів від параметрів задачі.

Практичне значення отриманих результатів:

– обґрунтовано принципову можливість використання методу Монте – Карло для визначення коефіцієнту теплопровідності нанодротів.

– розроблено методику визначення теплових властивостей одновимірних наноструктур .

Структура і обсяг роботи. Магістерська робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи становить 82 сторінки, із них основного тексту 59 сторінок, 2 таблиці, 23 рисунка за текстом, список використаних джерел із 58 найменувань на 4 сторінках, 5 додатків на 8 сторінках.

РОЗДІЛ 1. ВИКОРИСТАННЯ БАГАТОЯДЕРНИХ ОБЧИЛЮВАЛЬНИХ СИСТЕМ ТА ПАРАЛЛЕЛЬНИХ ОБЧИСЛЕНЬ В ФІЗИЦІ (АНАЛІТИЧНИЙ ОГЛЯД)

Паралельні обчислення давно вже перестали бути спеціальними технологіями, якими користувалися лише в дослідницьких центрах. Сьогодні методи паралельних обчислень в тій чи іншій мірі використовуються майже в кожній обчислювальній машині. Удосконалення елементної бази комп'ютерів в поєднанні з паралельною обробкою даних дозволило істотно підвищити продуктивність обчислень.

Одна з популярних технологій високопродуктивних обчислень використовує GPU. Спочатку цей клас пристроїв розроблявся для обробки графіки. Техніка удосконалювалася, GPU нарощували продуктивність і в якийсь момент виявилося, що GPU можна успішно використовувати не тільки для задач комп'ютерної графіки але і як математичний співпроцесор для CPU машини, отримуючи при цьому істотний приріст в продуктивності.

Цей клас технологій отримав назву GPGPU (General-Purpose Graphics Processing Units) – використання графічного процесора для загального призначення. Популярний представник цього класу – CUDA, вона була вперше представлена компанією NVIDIA в 2007 році[1].

1.1 Концепція універсального обчислювального пристрою CUDA

Програму на CUDA C можна логічно розділити на дві частини. Перша частина (керуюча) виконується на CPU, а друга частина (обчислювальна) виконується на GPU. CPU в термінах CUDA називається хостом (host), GPU називається пристроєм (device). Мовою CUDA C частина коду, яка повинна виконуватися на GPU, називається ядром (kernel), вона описується у вигляді функції.

На функції, виконувані на GPU, накладається ряд обмежень: вони не можуть містити рекурсію, не можуть мати змінне число аргументів і не можуть мати статичні (static) змінні всередині себе.

Специфікатори змінних, які служать для вказівки типу використовуваної пам’яті GPU, наведені в таблиці 1.1.

Таблиця 1.1

Специфікатори змінних в CUDA

Специфікатор	Знаходиться в	Доступний для	Вид доступу
__device__	пристрої	пристрія	тільки читання
__constant__	пристрої	пристрія та хоста	читання/запис
__shared__	пристрої	блока потоків	читання/запис

На їх використання також накладається ряд обмежень:

- специфікатор $__device__$ являється аналогом $const$ на центральному процесорі;
- специфікатор $__shared__$ застосовується для задання колективної пам’яті та не може ініціалізуватися при оголошенні. Він вимагає явної синхронізації потоків всередині одного блоку після будь-якої операції з такою пам’яттю;
- запис в $__constant__$ пам’ять здійснюється тільки через спеціальні функції з CPU;
- ніякі специфікатори не можна застосовувати до полів структур.

Директивою для виклику ядра являється така конструкція:

$$kernelName \ll\ gridSize, blockSize, sharedMemSize, cudaStream \gg.$$

Деякі змінні при виклику ядра можна опускати, наприклад $sharedMemSize$ та $cudaStream$. Тут:

- $kernelName$ – це ім'я функції ядра специфікатору $__global__$;

- $gridSize$ – це розмірність сітки для блоків (dim3), виділена для розрахунків;
- $blockSize$ – розмір блоку (dim3), виділеного для розрахунків;
- $sharedMemSize$ – розмір додаткової пам'яті, що виділяється при запуску ядра;
- $cudaStream$ – змінна $cudaStream_t$, що задає потік, в якому буде проведено виклик;

Також варто повідомити про вбудовані змінні:

- $gridDim$ – розмірність гріду, має тип dim3. Дозволяє дізнатися його розмір, який було виділено при поточному виклику ядра;
- $blockDim$ – розмірність блоку, має тип dim3. Дозволяє дізнатися розмір блоку, виділеного при поточному виклик ядра;
- $blockIdx$ – індекс поточного блоку в обчисленні на GPU, має тип uint3;
- $threadIdx$ – індекс поточного треду в обчисленні на GPU, має тип uint3;
- $warpSize$ – розмір варпу, має тип int.

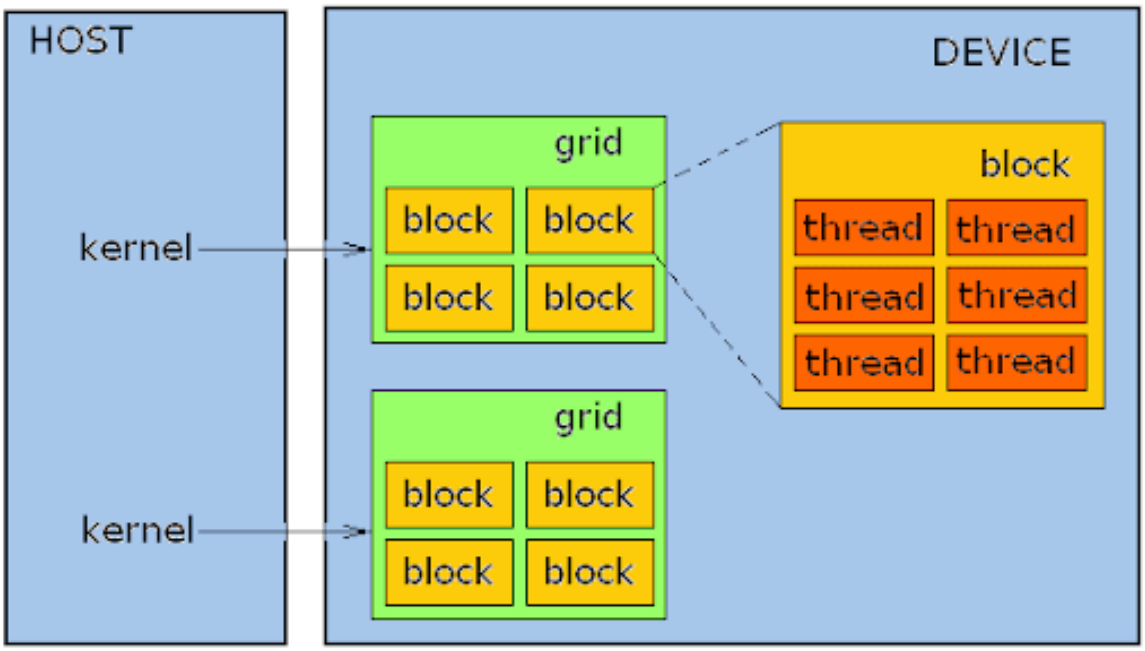


Рис. 1.1. Схема організації паралельних процесів

Запуск ядра породжує процеси (треди) на GPU відповідно до заданих параметрів (рис. 1.1). Всі процеси, породжені запуском ядра, в термінології CUDA називається грідом (grid). Грід складається з блоків (block), а блок, в свою чергу, з тредів (thread). Тред – це елементарний паралельний процес. Треди в блоках і блоки в гріді можуть бути представлені у вигляді одно-, дво- або тривимірній решітці.

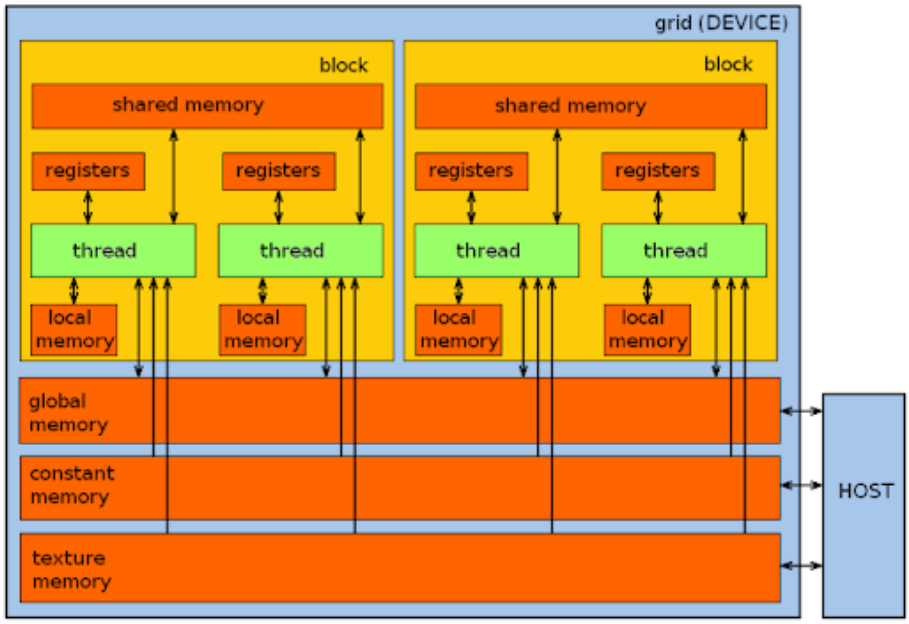


Рис.1.2. Схема організації пам’яті CUDA-програми

При конфігуруванні топології процесів необхідно враховувати апаратні особливості. Тут слід ввести поняття варпу. Варп (warp) – це група тредів. Полуварп (half-warp) – половина тредів варпу. Все треди одного варпу виконуються одночасно і синхронно (SIMD) на своєму мультипроцесорі. При доступі треду до основної пам’яті GPU її частини можуть кешуватися в локальній пам’яті даного мультипроцесора. Якщо всі дані, які потрібні треду полуварпа, будуть знаходитися в цьому кеші, то це може підвищити продуктивність.

GPU має складно організовану пам’ять (рис.1.2), крім основної (або глобальної) пам’яті кожен мультипроцесор володіє своєю пам’яттю.

1.2. Пам’ять в CUDA

Робота з пам’ттю в CUDA – це дуже важливе питання як з теоретичної, так і з практичної точки зору. Адже для написання програм, які будуть задіяти пам’ять GPU, потрібно розуміти основні команди роботи з пам’яттю і знати її ієрархічну структуру.

Фізично пам’ять GPU розділяється на DRAM і на мультипроцесорну, розміщену безпосередньо на самому графічному процесорі.

У CUDA є всього шість типів пам’яті:

- Регістри.
- Локальна пам’ять.
- Колективна пам’ять.
- Глобальна пам’ять.
- Константна пам’ять.
- Текстурна пам’ять.

Найбільш простим видом пам’яті являється реєстрова пам’ять. Кожен потоковий мультипроцесор містить 8192 або 16384 32-бітових регістрів. На етапі компіляції вони розподіляються між тредами блоку, тим самим впливаючи на кількість виконуваних блоків одним мультипроцесором. Кожен тред отримує в своє монопольне користування кілька регістрів, які доступні як для читання, так і для запису (read / write). Тред не має доступу до регістрів інших тредів, але свої регістри доступні йому протягом виконання даного ядра.

Оскільки регістри розташовані безпосередньо в потоковому мультипроцесорі, то вони володіють максимальною швидкістю доступу. Якщо наявних регістрів не вистачає, то для розміщення локальних даних (змінних) тредом використовується, так звана, локальна пам’ять, розміщена в DRAM. Тому доступ до локальної пам’яті характеризується дуже високою латентністю – від 400 до 600 тактів.

Локальна пам’ять – це область в глобальній пам’яті, виділена компілятором для зберігання локальних значень потоків. Локальна пам’ять використовується для зберігання локальних даних потоків в разі нестачі регістрів або оголошення локальних масивів всередині ядра без ключового слова $__shared__$.

Наступним типом пам’яті являється колективна (shared) пам'ять. Ця пам’ять розташована безпосередньо в мультипроцесорі, але вона виділяється на рівні блоків – кожен блок отримує в своє розпорядження одну й ту саму кількість колективної пам’яті. Всього кожен мультипроцесор містить 16 Кбайт колективної пам’яті, і від того скільки пам’яті потрібно блоку залежить кількість блоків, яке може бути запущено на одному мультипроцесорі.

Колективна пам’ять володіє дуже великою латентністю, яка доступна всім тредім блоку як для читання, так і для запису.

Глобальна пам’ять – це звичайна пам’ять DRAM, яка виділяється за допомогою спеціальних функцій на CPU. Все треди сітки можуть як читати, так і записувати в глобальну пам’ять. Оскільки вона розташована на центральному процесорі, а не на графічному, то вона має високу латентність (від 400 до 600 тактів). В таблиці Додатку А описані вищевказані типи пам’яті з їх характеристиками.

1.2.1. Робота з глобальною пам’яттю в CUDA

Хоч глобальна пам’ять являється основною в CUDA, однак вона при цьому й сама повільна (вона повільно звертається за випадковою адресою, відсутній кеш і, відповідно, кожен раз дані треба зчитувати заново). Процес роботи з глобальною пам’яттю дуже схожий на звичайну роботу з пам’яттю на центральному процесорі:

1. ініціалізація (виділення);
2. заповнення (копіювання);
3. використання;
4. зворотне копіювання;

5. звільнення.

Варто вказати на базові функції CUDA для роботи з глобальною пам’яттю, так як вони являються невід’ємною частиною програм.

- $cudaError_t\ cudaMalloc(void\ **\ devPtr, size_t\ size)$ – виділення пам’яті на відеокарті. Тут $devPtr$ – вказівник на ділянку пам’яті на пристрої, $size$ – розмір виділеної пам’яті в байтах. Повертає $cudaSuccess$ при вдалому виділення пам’яті, або ж $cudaErrorMemoryAllocation$ при помилці.
- $cudaError_t\ cudaFree\ (void\ *\ devPtr)$ – звільнення пам’яті на пристрої, тут $devPtr$ являється вказівником на ділянку пам’яті;
- $cudaError_t\ cudaMemcpy\ (void\ *\ dst, const\ void\ * src, size_t countm, enum\ cudaMemcpyKind\ kind)$ – функція копіювання даних між хостом та пристроєм. Тут dst та src – вказівники на джерело та приймач відповідно, $count$ – розмір копійованих даних в байтах, $kind$ – напрямок копіювання. $kind$ має всього 4 значення:

1) $cudaMemcpyHostToDevice$ – копіювання даних з хоста до пристрою;
2) $cudaMemcpyDeviceToHost$ – копіювання даних з пристрою до хосту;
3) $cudaMemcpyHostToHost$ – копіювання даних з хоста до хосту;
4) $cudaMemcpyDeviceToDevice$ – копіювання даних з пристрою до пристрою;

Функція $cudaError_t\ cudaMemcpy$ повертає:

- $cudaSuccess$ – при вдалому копіюванні;
- $cudaErrorInvalidValue$ – при невірних параметрах аргументу (наприклад, розмір копіювання негативний);
- $cudaErrorInvalidDevicePointer$ – невірний вказівник пам’яті в відеокарті;
- $cudaErrorInvalidMemcpyDirection$ – неправильний напрямок (наприклад, переплутано джерело і місце-призначення копіювання).

1.2.2 Робота с колективною пам’яттю в CUDA

Колективна пам’ять знаходиться на кожному SM і використовується спільно всіма тредами всередині блоку. Колективна пам’ять, як правило, вимагає використання команди явною синхронізації $__syncthreads()$ після операцій копіювання та зміни. Для підвищення пропускної здатності колективна пам’ять розбита на 16 осередків, доступ до яких йде незалежно. Банки складаються з 32-бітних слів, причому ті слова, що йдуть поспіль, розташовані в тих блоках, що також йдуть поспіль. Якщо виникає ситуація, що кілька тредів звертаються до одного й того ж осередку (не елементу, а саме осередку), то їх звернення обробляються послідовно (відбувається конфлікт осередків), проте якщо все треди звертаються до одного й того ж елементу, то конфлікту не відбувається.

Константна пам’ять використовується тоді, коли в ядро необхідно передати багато різних даних, які будуть однаково використовуватися всіма тредами ядра.

$__constant__\ float\ constData\ [256]$ – оголошення глобальної змінної з ім’ям $constData$ для використання в якості константної пам’яті. $cudaMemcpyToSymbol\ (constData, hostData, sizeof\ (data), 0,$ $cudaMemcpyHostToDevice)$ – копіювання даних з центрального процесора в константну пам’ять. Використання константної пам’яті всередині ядра нічим не відрізняється від використання звичайної глобальної змінної на хості.

1.3. Бібліотека cuRand для генератора випадкових чисел

Бібліотека cuRAND надає засоби, що зосереджуються на простому та ефективному генеруванні високоякісних псевдовипадкових та квазівипадкових чисел. Псевдовипадкова послідовність чисел задовільняє більшість статистичних властивостей істинно випадкової послідовності, але породжується детермінованим алгоритмом. Квазівипадкова послідовність n-вимірних точок генерується детермінованим алгоритмом, призначеним для рівномірного заповнення n-вимірного простору.

Бібліотека cuRAND складається з двох фрагментів: бібліотеки на стороні хоста та файлу заголовку пристрою. Бібліотека на стороні хоста обробляється як будь-яка інша бібліотека процесора: користувачі підключають файл заголовка $/include/curand.h$, щоб отримати об'явлення функцій, а потім посилання на бібліотеку. Випадкові числа можна генерувати на пристрої або на процесорі хоста. Для генерації на пристрої виклики бібліотеки відбуваються на хості, але фактична робота генерування випадкових чисел відбувається на пристрої. Отримані випадкові числа зберігаються в глобальній пам'яті на пристрої. Потім користувачі можуть викликати власні ядра для використання випадкових чисел, або вони можуть скопіювати випадкові числа назад на хост для подальшої обробки. Для генерації процесора хоста всі роботи виконуються на ньому, а випадкові числа зберігаються у пам'яті хоста.

Другий фрагмент cuRAND – це файл заголовка пристрою $/include/curand_kernel.h$. Цей файл визначає функції пристрою для налаштування станів генератора випадкових чисел та генерації послідовностей випадкових чисел. Код може включати цей файл заголовку, а написані ядра можуть викликати функції пристрою, визначені у файлі заголовку. Це дозволяє генерувати випадкові числа та негайно споживати їх ядрами, не вимагаючи запису випадкових чисел та їх зчитування з глобальної пам'яті.

1.3.1. Використання API на стороні хоста

Щоб використовувати API на стороні хоста, код повинен включати файл заголовку бібліотеки $curand.h$ та динамічно зв'язуватися з бібліотекою cuRAND. Бібліотека використовує середу виконання CUDA, тому код також повинен буди на ній же. CUDA Driver API, в свою чергу, не підтримується cuRAND.

Випадкові числа виробляються генераторами. Генератор в cuRAND інкапсулює весь внутрішній стан, необхідний для створення послідовності псевдовипадкових чи квазівипадкових чисел. Нормальна послідовність операцій така:

1. Створюється новий генератор потрібного типу за допомогою $curandCreateGenerator$ ().
2. Встановлюються параметри генератора; наприклад, можна використовувати $curandSetPseudoRandomGeneratorSeed$ () для встановлення *seed*.
3. Виділяється пам’ять на пристрої за допомогою $cudaMalloc$ ().
4. Створюються випадкові числа за допомогою $curandGenerate$ () або іншої функції генератору.
5. Використовуються результати.
6. За бажанням генерується більше випадкових чисел із більшою кількістю викликів $curandGenerate$ ().
7. Очистка за допомогою $curandDestroyGenerator$ ().

Щоб генерувати випадкові числа на процесорі хоста, на першому етапі треба викликати $curandCreateGeneratorHost$ () та на третьому етапі виділити буфер пам’яті хоста для отримання результатів. Усі інші виклики працюють однаково, незалежно від того, генеруються випадкові числа на пристрої, чи на процесорі хоста.

Можна створювати кілька генераторів одночасно. Кожен генератор інкапсулює окремий стан і не залежить від усіх інших. Послідовність чисел, вироблених кожним генератором, являється детермінованою. Враховуючи однакові параметри налаштування, одна й та ж послідовність буде створена при кожному запуску програми. Генерація випадкових чисел на пристрої призведе до тієї ж послідовності, що їх генерування на процесорі хоста.

Треба зауважити, що $curandGenerate$ () запускає і повертає ядро асинхронно. Якщо використовується інше ядро в іншому потоці, і воно повинно використовувати результати $curandGenerate$ (), то потрібно або викликати $cudaThreadSynchronize$ () або використовувати підпрограми управління потоком / управління подіями, щоб гарантувати, що ядро випадкової генерації закінчило виконання перед запуском нового ядра.

Також невірно передавати вказівник пам’яті хоста генератору, який працює на пристрої, і невірно передавати вказівник пам’яті пристрою генератору, який працює на процесорі. Поведінка в цих випадках не визначена.

Генератори випадкових чисел створюються шляхом передачі типу $curandCreateGenerator$(). У cuRAND існує дев'ять типів генераторів випадкових чисел, які поділяються на дві категорії.

Після створення генератори випадкових чисел можна визначити, використовуючи загальні параметри *seed*, *offset* та *order*.

Параметр *seed* – це 64-бітове ціле число, яке ініціалізує початковий стан генератора псевдовипадкових чисел. Один й той же *seed* завжди дає однакову послідовність результатів.

Параметр *offset* використовується для зміщення вперед в послідовності. Якщо offset = 100, то перше генероване випадкове число буде 100-м у послідовності. Це дозволяє декільком запусків однієї програми продовжувати генерувати результати з однієї послідовності без перекриття.

Параметр *order* використовується для вибору порядку упорядкування результатів у глобальній пам’яті.

1.3.2. Використання API на стороні пристрою

Щоб використовувати API пристрою, треба підключити файл $curand_kernel.h$ у файлах, що визначають ядра, які використовують функції пристрою cuRAND. API пристрою включає в себе функції псевдовипадкової генерації для і квазівипадкової генерації.

Функції псевдовипадкових послідовностей підтримують генерацію бітів та генерацію з дистрибутивів. Це можна зробити, наприклад, за допомогою генераторів XORWOW, який реалізований з використанням алгоритму XORWOW, члена сімейства генераторів псевдовипадкових чисел xor-shift, та MRG32k3a, який, в свою чергу, являється членом сімейства генераторів псевдовипадкових чисел Combined Multiple Recursive. В такому випадку буде:

$__device__\ unsigned\ int$

$curand\ (curandState_t\ *\ state)$

Після виклику $curand_init\ ()$, $curand\ ()$ повертає послідовність псевдовипадкових чисел з періодом, більшим за 2^{190}. Якщо $curand\ ()$ викликається з однаковим початковим станом кожного разу, а стан не змінюється між викликами $curand\ ()$, тоді завжди генерується однакова послідовність.

$__device__\ void$

$curand_init\ ($

$unsigned\ long\ long\ seed, unsigned\ long\ long\ sequence,$

$unsigned\ long\ long\ offset, curandState_t\ *\ state)$

Функція $curand_init\ ()$ встановлює початковий стан, призначений стороною, що її викликала, використовуючи заданий *seed*, номер послідовності та зміщення в межах послідовності. Різні *seed* гарантовано дають різні стартові стани та різні послідовності. Один й той же, в свою чергу, завжди виробляє той самий стан й ту саму послідовність.

Послідовності, згенеровані різними *seed*, зазвичай не мають статистично корельованих значень, але деякі варіанти *seed* можуть дати статистично корельовані послідовності. Послідовності, згенеровані одним й тим же *seed* та різними порядковими номерами, не матимуть статистично корельованих значень.

Для найвищої якості паралельного генерування псевдовипадкових чисел кожному експерименту слід призначити унікальний *seed*. В рамках експерименту кожному потоку обчислень слід присвоїти унікальний порядковий номер. Якщо експеримент охоплює декілька запусків ядра, то рекомендується, щоб треди між запусками ядра отримували один й той же *seed*, а порядкові номери присвоювали монотонно зростаючим способом. Якщо запускається однакова конфігурація потоків, випадковий стан може

зберігатися у глобальній пам’яті між запусками, щоб уникнути часу встановлення стану.

1.4 Метод розкладання по матрицях Гелл-Мана

Даний метод полягає в перетворенні рівняння для матриці щільності в векторний вигляд. Однак він містить дані про симетрії системи, відповідно до теорії некомутативних груп Лі і розкладанням по матрицях Гелл-Мана [6].

Узагальненими матрицями Гелл-Мана називають $n^2–1$ матриць, що представляють собою генератори групи *SU(n)*, $n \geq 2$. Вони є узагальненням добре відомих матриць Гелл-Мана, генераторів групи *SU(3)* і матриць Паулі, генераторів групи *SU(2)*.

Алгоритм їх побудови може бути легко описаний за допомогою допоміжних матриць E_{jk}. Всі елементи матриці E_{jk}, крім елемента *(j, k)* рівного 1, нульові, тобто

$$(E_{jk})_{lm} = \delta_{jl}\delta_{km}, \tag{1.1}$$

де δ_{xy} – символ Кронекера. Далі вводиться три набора матриць. Перший набір становлять симетричні матриці λ_{jk}^{s}:

$$\lambda_{jk}^{s} = E_{jk} + E_{kj}, \tag{1.2}$$

де $1 \leq j < k \leq n$. Другий набір утворений антисиметричними матрицями λ_{jk}^{a}:

$$\lambda_{jk}^{a} = -i(E_{jk} - E_{kj}) , \tag{1.3}$$

де i – уявна одиниця, $1 \leq j < k \leq n$. Третій набір включає в себе діагональні матриці λ_l виду

$$\lambda_l = \sqrt{\frac{2}{l(l+1)}}\left(\sum_{j=1}^{l} E_{jj} - lE_{l+1,l+1}\right), \tag{1.4}$$

$1 \leq l \leq n - 1$. Легко знайти, що загальне число узагальнених матриць Гелл-Мана

$$\frac{n(n-1)}{2}+\frac{n(n-1)}{2}+(n-1)=n^2-1$$

Узагальнені матриці Гелл-Мана можуть бути використані для побудови базису простору ермітових (самоспряжених) матриць. Для цього до перерахованої множини необхідно додати будь-яку матрицю, пропорційну одиничній матриці *I*. Один з варіантів побудови базису полягає в наступному. Вважається, що $E_{n+1,n+1}=0$, а ряд діагональних матриць λ_l триває аж до $l \le n$, тобто $\lambda_n = \sqrt{\frac{2}{n(n+1)}}I$.

Для використання елементів побудованого базису в програмній реалізації виявляється зручною наступна їх нумерація:

$$\begin{aligned}
&\Lambda_1=\Lambda_{11},\Lambda_2=\Lambda_{12},\dots,\Lambda_n=\Lambda_{1n},\\
&\Lambda_{n+1}=\Lambda_{21},\Lambda_{n+1}=\Lambda_{22},\dots,\Lambda_{2n}=\Lambda_{2n},\\
&\dots\\
&\Lambda_{n(n-1)+1}=\Lambda_{n1},\dots,\Lambda_{n^2}=\Lambda_{nn}
\end{aligned} \tag{1.5}$$

де

$$\Lambda_{jk}=\begin{cases}\lambda_{jk}^{s}, & j<k\\ \lambda_{kj}^{a}, & j>k\\ \lambda_j, & j=k\end{cases} \tag{1.6}$$

Всі матриці Λ_k мають властивість ортогональності по сліду, яке може бути використане для розкладання довільної ермітової матриці по базису Λ_k:

$$Tr(\Lambda_j\Lambda_k)=\begin{cases}2\delta_j\delta_k, & (j\neq n)\wedge(k\neq n)\\ \frac{2}{n+1}\delta_j\delta_k, & (j=n)\vee(k=n)\end{cases} \tag{1.7}$$

В основі наступного методу рішення рівняння для матриці щільності [5] $\frac{\partial \rho}{\partial t} = \frac{1}{t\hbar}[H, \rho] + \frac{\gamma}{2}(2a\rho a^{+} - a^{+}a\rho - \rho a^{+}a)$ лежить ідея розкладання шуканої матриці по базису Λ_k:

$$\rho(t) = \sum_{a=1}^{\pi^2} \rho_a(t)\Lambda_a \tag{1.8}$$

Таким чином, завдання зводиться до розрахунку динаміки вектору коефіцієнтів розкладання $\{\rho_1, \rho_2, \dots, \rho_{n^2}\}$, а не самої матриці щільності ρ. Даний метод має ряд переваг, головний з яких полягає в тому, що даний підхід враховує фізичну специфіку розв'язуваної задачі – автоматично зберігає ермітовість шуканої матриці щільності ρ. Крім того, коефіцієнти розкладання по базису Λ_k являються дійсними. Аналогічне розкладання гамильтоніану *H* по базису Λ_k та дійсність всіх коефіцієнтів розкладання дозволяють повністю відмовитися від використання комплексних чисел в програмній реалізації, що в свою чергу призводить до її спрощення та економії використовуваних ресурсів.

Виконання розкладання ρ, *H* та дисипативного доданка рівняння для матриці щільності по базису Гелл-Мана Λ_k призводить до наступної системи рівнянь для вектору коефіцієнтів розкладання $\{\rho_1, \rho_2, \dots, \rho_{n^2}\}$:

$$\frac{\partial \rho_a}{\partial t} = \sum_{\mu,\nu} f_{\alpha\mu\nu} + H_\mu \rho_\nu + \sum_\nu \Gamma_{\alpha\nu}\rho_\nu\,,\ 1 \le \alpha \le n^2 \tag{1.9}$$

Тут структурні константи $f_{\alpha\mu\nu}$ та $\Gamma_{\alpha\nu}$ визначаються наступними виразами

$$f_{\alpha\mu\nu} = \frac{1}{2i} Tr(\Lambda_\alpha[\Lambda_\mu \Lambda_\nu]) \tag{1.10}$$

$$\Gamma_{\alpha\nu} = \frac{\gamma}{4} Tr(\Lambda_\nu a^{+}[\Lambda_\alpha a] - a\Lambda_\nu[\Lambda_a a^{+}]) \tag{1.11}$$

Проводячи елементарні перетворення, можна переписати вираз для структурних констант (1.10) у вигляді

$$f_{\alpha\mu\nu} = ImTr(\Lambda_\alpha \Lambda_\mu \Lambda_\nu) \tag{1.12}$$

1.4.1 Алгоритм розпаралелювання методу розкладання по матрицях Гелл-Мана

Основні витрати обчислювальних ресурсів в описаному методі пов'язані з обчисленням подвійної суми $\sum_{\mu,\nu}$ ... правої частині системи (1.9).

Для обчислення кожної з таких сум потрібно ~$3n^4$ операцій додавання і множення. Загальна кількість операцій, необхідних для обчислення подвійних сум, що відносяться до різних рядків системи, складає ~$3n^6$. Число операцій, необхідних для обчислення всіх n^2 сум, пов'язаних з дисипативної частиною, на два порядки менше та становить ~$2n^4$.

Однак, завдяки особливим властивостям структурних констант $f_{\alpha\mu\nu}$ (1.12), багато з них виявляються рівними нулю, що дозволяє виключити з підсумовування $\sum_{\mu,\nu}$ ... переважну більшість доданків. До згаданих вище властивостям структурних констант $f_{\alpha\mu\nu}$ відносяться їх антисиметричність по перестановці будь-якої пари індексів і дійсність їх значень (1.12). Аналітичні розрахунки показують, що при фіксованому значенні індексу α число структурних констант $f_{\alpha\mu\nu}$, відмінних від нуля, змінюється в межах від 0 до *½(n-1)(n+3)* для непарних *n* або до *½(n^2+2n-4)* для парних значень *n* (n≥8). Таким чином, число ненульових доданків в сумі $\sum_{\mu,\nu}$ ... являється величиною порядку ~ $n^2/2$, і для її обчислень потрібно ~ $3n^2/2$, що на два порядки менше, ніж потрібно при звичайному підсумовуванні. Загальна кількість незалежних ненульових структурних констант $f_{\alpha\mu\nu}$ становить *1/6 (n-1)(5n^2-4n-6)*, ~ $5n^2/6$.

Використання описаних фактів здатне збільшити продуктивність обчислювального процесу і стало основою для розробленої програми.

Один з можливих та найбільш простих способів реалізації підсумовування $\sum_{\mu,\nu}$ ... з виключенням нульових доданків полягає в тому, що у відповідність кожному рядку α системи ставиться таблиця, звана таблицею індексів множення, що має структуру таблиці 1.2.

Таблиця 1.2

Вид таблиці індексів множення

μ	ν	$f_{\alpha\mu\nu}$
μ_1	ν_1	$f_{\alpha\mu_1\nu_1}$
μ_2	ν_2	$f_{\alpha\mu_2\nu_2}$
μ_3	...	...

k-ий рядок такої таблиці вказує, що в суму має увійти доданок $f_{\alpha\mu_k\nu_k}H_{\mu_k}\rho_{\nu_k}$. Побудова таблиць індексів множення відбувається в момент запуску програми. Для обчислення значень $f_{\alpha\mu\nu}$ використовується алгоритм, який застосовує аналітичні вирази і не вимагає роботи з матрицями, що робить процес ініціалізації таблиць індексів множення дуже швидким. Самі таблиці індексів множення виконані у вигляді масиву структур *MultiplicationIndex*:

```
__global__ struct MultiplicationIndex
{
    int x, y;
    REAL v;
};
```

Поля x, y та v грають відповідно роль μ, ν та $f_{\alpha\mu\nu}$. З огляду на антисиметричні властивості структурних констант $f_{\alpha\mu\nu}$ досить зберігати лише половини таблиць індексів таблиць множення.

Аналогічні міркування можна провести і для суми $\sum_\nu$ ... . Кожна таблиця індексів множення тепер складатиметься з двох стовпців і містити один або два записи. Таким чином, кількість операцій, необхідних для обчислення однієї суми, знижується на два порядки, з ~$2n^2$ до ~2. Індекси множення суми $\sum_\nu$ ... також можна зберегти в таблицю. Так як в цьому випадку потрібно

задіяти один індекс v, то значення індексу μ виставляємо рівним -1, що дозволить під час рахунку визначати, до якої з двох сум відноситься доданок.

Додаткове прискорення програми досягається шляхом розпаралелювання процесу обчислення груп рядків правої частини системи на графічних процесорах. Кількість рядків в групі має вибиратися з урахуванням технічних характеристик графічного процесора (його тактової частоти, обсягу пам'яті, максимального числа потоків). На стадії ініціалізації програми визначається число графічних процесорів в системі, їх характеристики, й приймається рішення про розміри груп, після чого, відповідно до прийнятого рішення, їм розсилаються відповідні таблиці індексів множення. При обчисленні на GPU одному рядку групи у відповідність ставиться один тред GPU. Реалізація всіх перерахованих вище ідей для обчислення правих частин на GPU представлена в Додатку Б.

Функція $CalculateRightPart$ утворена двома циклами, які обчислюють суми $\sum_{\mu,\nu} \ldots$ та $\sum_{\nu} \ldots$. Ознакою другої суми є негативний індекс x. Використовується відповідна рядку n таблиця індексів множення $aMultiplicationIndices\,[n]$, побудована при ініціалізації програми.

Процес ініціалізації програми умовно може бути розділений на три основних етапи:

1. створення субкомунікаторов, які об'єднують процеси, виконувані на одному вузлі;
2. вибір кожним процесом відеокарти, використовуваної для обчислень;
3. поділ навантаження між відеокартами.

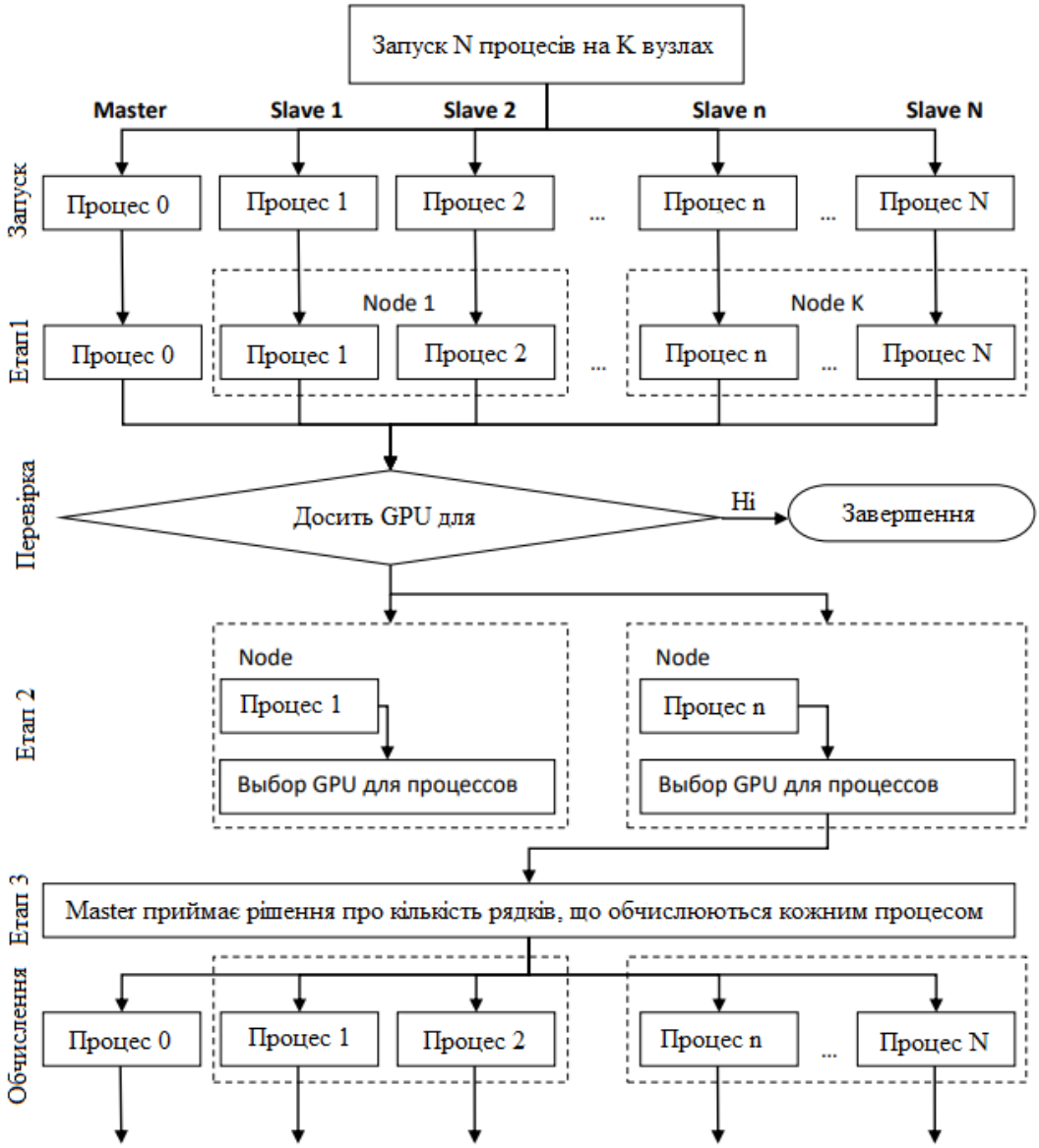


Рис. 1.3. Схема процесу ініціалізації програми для методу розкладання по матрицях Гелл-Мана.

Для об'єднання в групи процесів, запущених на одному вузлі, використовується об’єкт комунікатора і функція MPI [6] MPI_Comm_split, що забезпечує швидкий та простий спосіб одночасного створення декількох комунікаторів. Алгоритм формування груп може бути описаний таким чином:

1. всі процеси, крім процесу Master (має ранг 0 в комунікаторі MPI_COMM_WORLD), «забарвлюються» в чорний колір.

Використовуючи «забарвлення» процесів та функцію MPI_Comm_split, створюємо комунікатори;

2. всі «незабарвлені» процеси продовжують виконання циклу 2 – 5;
3. «незабарвлений» процес з рангом 0, використовуючи свій поточний комунікатор, розсилає всім учасникам групи ім'я вузла і «забарвлюється»;
4. члени групи беруть ім’я вузла. Якщо прийняте ім’я збігається з ім’ям вузла приймача, то процес «забарвлюється»;
5. використовуючи «забарвлення» процесів та функцію MPI_Comm_split, створюємо комунікатори. Переходимо до кроку 2.

Додаток В містить фрагмент вихідного коду програми, що реалізує описаний алгоритм формування субкомунікаторов.

На другому етапі ініціалізації програми кожен процес з рангом 0 в субкомунікаторі вузла (node master) збирає інформацію про встановлені на вузлі відеокарти. Для цього використовуються стандартні функції CUDA $cudaGetDeviceCount$ та $cudaGetDeviceProperties$. На основі значення поля clockRate структури cudaDeviceProp вибирається необхідне число найбільш високопродуктивних відеокарт і здійснюється їх прив'язка до процесів. Відповідний фрагмент вихідного коду міститься в Додатку Г.

На третьому етапі ініціалізації процес Master приймає рішення про розділення рядків системи між процесами для забезпечення рівномірного завантаження системи. Найпростіше рішення цього завдання може бути отримано в припущенні, що обчислення кожного рядка вимагає однакової кількості операцій, а швидкодія відеокарти визначається її тактовою частотою. В такому випадку, для забезпечення рівномірного завантаження, число рядків, що обчислюються на i-ій відеокарті, має бути пропорційно її тактовій частоті і буде дорівнювати $\approx n^2 \cdot clockRate[i] / \sum_k clockRate[k]$.

РОЗДІЛ 2. ФІЗИЧНІ ОСОБЛИВОСТІ ТЕПЛОПРОВІДНІСТНИХ ВЛАСТИВОСТЕЙ НАНОДРОТІВ

З удосконаленням технологій виготовлення нано та інженерних технологій в останні десятиліття були розроблені та синтезовані одновимірні (1D) матеріали, такі як нанодроти (NW), нанотрубки та квантові дроти [10,11]. Завдяки унікальним властивостям, відмінним від об'ємних матеріалів, 1D матеріали викликають зростаючий науковий інтерес та мають широке застосування в галузі електронних, оптоелектронних та приладів перетворення енергії. Серед них кристалічні напівпровідникові NW являються дуже перспективними матеріалами в сучасній мініатюризації пристроїв у напрямку до нанорозмірних. Вони охоплюють широкий спектр матеріалів – від одноелементних напівпровідників (таких як Si, Ge та Bi), складних напівпровідників (таких як ZnO, GaN, SiC, Bi_2Te_3, GaAs та InP), до композиційних систем (таких як Si/Ge, SiO_2/SiC і ZnS/Si), в яких Si, Ge та їх складені NW отримали численні дослідження для перспективних застосувань та інтеграції з поточною промисловістю напівпровідникової електроніки.

Теплопровідність нанорозмірних матеріалів сильно відрізняється від об'ємних. Розуміння та прогнозування теплопровідності NW відіграють важливу роль у просуванні наступних двох найбільш фундаментальних застосувань: управління теплом, яке має принципове та технологічне значення завдяки широкому застосуванню в контролі теплового потоку, сучасних фононічних обчисленнях, нових рішень з матеріалами для розсіювання тепла та нових термоелектричних матеріалах, де ефективність перетворення фіксується безрозмірною показником достовірності $ZT = S^2\sigma T/k$. Тут S – коефіцієнт Зеєбека, T – температура, σ – електропровідність, k – теплопровідність. Таким чином, бажана низька теплопровідність разом з високим коефіцієнтом термоенергії для кращих показників термоелектрики. Термоелектрика в NW привернула широку увагу через їх надзвичайно покращений термоелектричний показник достовірності, порівняно з

об'ємними аналогами, що в основному викликано значним зниженням теплопровідності. Наприклад, ранній експеримент Хохбаума та інших виявив, що *ZT* у кремнієвих нанодротах може наблизитись до цільності при кімнатній температурі через надзвичайне 100-кратне зниження теплопровідності над об'ємними значеннями в кремнієвих нанодротах з шорсткою поверхнею, зберігаючи при цьому чудові електричні властивості. Було запропоновано багато підходів для подальшого зниження теплопровідності NW для оптимізації термоелектричних характеристик, таких як введення домішків розсіювання, діркова структура та шорсткі поверхні.

З іншого боку, останні досягнення в галузі нанофабрикації дозволяють контролювати геометрію та розмірність NW з атомною точністю. Отже, пропонуються різні шляхи наноструктурування для побудови різноманітних NW, таких як NW сплавів, надрешіткові NW, NW з серцевинною оболонкою, пористі NW, резонансна та зігнута структури. У цих NW виявлено різні способи теплового переносу та зниження теплопровідності. Наприклад, когерентний фонон, знайдений як у NW із надрешіткою, так і у NW із серцевинною оболонкою, може використовуватися для зниження теплопровідності NW. Більше того, різні методи наноструктурування можуть призвести до значного зниження теплопровідності NW, перебуваючи під роздільними механізмами розсіювання. На сьогоднішній день не існує систематичного огляду різних методів наноструктуризації та поведінки теплового переносу в NW, який був би цінним для огляду існуючих досліджень та вказівок до майбутніх робіт.

2.1. Вплив розміру зразка на теплопровідність

Фонони являються домінуючими теплоносіями в напівпровідникових твердих тілах і мають незначний внесок у тепловий перенос у металах. Встановлено, що розміри системи мають суттєвий вплив на розсіювання фононів та поведінку теплового переносу в низькорозмірних матеріалах, особливо для NW. З іншого боку, безперервна мініатюризація транзисторів і,

як наслідок, збільшення щільності потужності створюють серйозні проблеми з термічним управлінням, що ще більше ускладняється невеликими характерними розмірами пристроїв, які також роблять тепловий перенос в NW важливим явищем для дослідження. Далі, узагальнені останні дослідження впливу розміру, включають вплив довжини, вплив діаметра та вплив орієнтації на теплопровідність NW.

2.1.1. Вплив довжини зразка

Теплопровідність ґратчастого матеріалу, в основному, описувалася в значенні середньої довжини вільного пробігу (MFP) фононів. Коли масштаб довжини стає порівнянним з MFP тих фононів, які вносять значний внесок у тепловий перенос, то може спостерігатися явне зниження теплопровідності. Таким чином, тривала залежність теплопровідності в NW цікава, змістовна і дуже важлива для вивчення. З молекулярної динаміки (MD) та розрахунків Грін-Кубо, Вольц і Чен виявили [12], що Si NW мають слабку залежність від довжини. Охара та інші [13] також повідомили, що MFP фононів Si NW являється досить коротким, нижче 5 нм, коли діаметр NW менше 6.5 нм. Отримані результати свідчать про те, що MFP фононів Si NW набагато коротша, ніж у об’ємного Si, в якому більше половини теплопровідності при кімнатній температурі вносять фонони з MFP довше 1 мкм. Тим часом Ян з іншими [14] продемонстрували за допомогою нерівноважного моделювання MD, що теплопровідність k Si NW розходиться з довжиною як $k \sim L^{\beta}$, навіть коли довжина до 1.1 мкм, що набагато довше, ніж MFP фонону, що вказує на порушення закону Фур'є в низькорозмірних матеріалах, як показано на рис. 2.1(а). Розрізна залежність довжини, виявлена в теоретичному дослідженні, пов'язана з тим, що взаємодії фонон-фонон недостатньо для індукції дифузійного процесу, викликаючи наддифузійний перенос фонону. Однак при експериментальному вимірюванні через сильний ефект розсіювання фононів шорсткою поверхнею та дефекти експериментальних зразків цю розбіжну залежність довжини важко спостерігати. Нещодавно і експериментальні, і

теоретичні роботи підтвердили цей результат. Більше того, Чжоу і Ху [15] знайшли інший тип Si NW з теплопровідністю, нечутливою до довжини, завдяки здавлюванню відтворюючих фононів. Здійснюючи нерівноважне моделювання та моделювання MD Грін-Кубо, вони повідомили, що теплопровідність Si NW в полікристалічній формі може досягати рекордно низького значення, істотно нижче межі Казиміра. Сильна локалізація полікристалічних NW призводить до помітного зниження ефективного MFP для теплоносіїв, включаючи як відтворюючі, так і дифузійні, і, таким чином, прориваючи межу Казиміра та слабку залежність довжини.

Існує також ряд досліджень, які свідчать про сильну залежність теплопровідності від довжини та балістичного теплового переносу в NW. Сяо та інші дослідники встановили [16], що балістична теплопровідність може зберігати понад 8.3 мкм у SiGe сплаву NW з низькою теплопровідністю для широкого діапазону структурних змін та концентрацій сплавів. Існує несподівано низький відсоток (~0,04%) низькочастотних (нижче 0.3 ТГц) фононів, що здійснюють процес теплопровідності в SiGe NW, що призводить до надзвичайно великої балістичної відстані переносу. Їх подальша робота також повідомляла про балістичну теплопровідність у мікронних масштабах у Si/Ge, неоднорідно пов’язаних NW, демонструючи низьку теплопровідність у малих масштабах. Неоднорідно пов’язані NW подовжують MFP більше 5 мкм при кімнатній температурі. Така MFP не тільки в 25 разів довша, ніж у Si або Ge, але і довше, ніж у найкращих теплових провідників, таких як алмаз або графен.

2.1.2. Вплив діаметру зразка

У 1D структурах NW ефективна теплопровідність являється функцією як поверхневої (граничної), так і внутрішнього фонон-фонон розсіювання, як повідомляється у Si NW, алмазних NW, GaAs NW та Cu NW. Внаслідок унікального розсіювання поверхні в NW, визначеного характером його структури, значення теплопровідності можна зменшити навіть на один

порядок нижче межі Казиміра. Малий поперечний переріз у NW передбачає більш значне зниження теплопровідності внаслідок поверхневого розсіювання порівняно з таким у тонких плівках через більш високе співвідношення поверхні та об'єму (SVR) у NW. Наприклад, ранні експерименти та моделювання на Si або Si/Ge NW підтвердили, що теплопровідність при кімнатній температурі сильно залежить від діаметрів, виявляючи тенденцію до зменшення із збільшенням поверхневих розсіювань. Нещодавно Лі та інші експериментально продемонстровали [17], що ця залежність все ще діє при високій температурі до 700 К. При високій температурі теплопровідність гладких Si NW також проявляє залежність класичного діаметра від 40 до 120 нм, і зростає внесок високочастотних фононів (оптичні фонони) у міру зменшення діаметра та підвищення температури.

З іншого боку, інші механізми розсіювання, такі як фононно-поверхневе розсіювання та розсіювання фонон-домішки, також впливали б на ступінь залежності діаметра. Наприклад, Хохбаум з іншими вченими експериментально встановили [18], що після травлення на поверхні теплопровідність шорстких Si NW має не тільки меншу величину, але й виявляє слабку залежність від діаметра. Крім того, Мартін та інші обчислили [19] частотно-залежну швидкість розсіювання фононів з теорії збурень і виявили, що теплопровідність грубих Si NW проявляє квадратичну залежність від діаметра D та шорсткості Δ як $(D/\Delta)^2$.

Для інших видів NW також повідомлялося про аналогічну залежну від діаметра теплопровідність NW, таких як ZnO NW, PbTe NW, Bi_2Te_3 NW та Bi NW. У більшості NW теплопровідність зазвичай демонструє різке зниження принаймні на порядок порівняно з об'ємними значеннями завдяки посиленому фононно-поверхневому розсіюванню. Тим часом, також спостерігаються деякі унікальні залежні від діаметра явища теплового переносу та фононні дії, що спостерігаються у NW. Буй з іншими експериментально встановили [20], що в індивідуальній теплопровідності ZnO NW приблизно показана лінійна залежність від площі поперечного перерізу NW в діапазоні вимірюваного

діаметру ~50 – 210 нм. Крім того, за допомогою симуляції MD, Чен та інші знайшли [21] просту універсальну лінійну залежність теплопровідності від SVR для Si NW з малим діаметром незалежно від геометрії поперечного перерізу. Як високоефективний термоелектричний матеріал, теплопровідність Bi_2Te_3 NW експериментально вивчається Рохо та іншими [22]. В основному, спостерігалося помітне зниження теплопровідності, більше ніж на 70%, коли діаметр NW був зменшений на один порядок (з 300 нм до 25 нм). Що ще цікавіше, аналіз на основі кінетично-колективної моделі показує, що поверхневе розсіювання має значний вплив на акустичні фонони і значною мірою змінило MFP низькочастотних фононів.

Більше того, деякі дослідження повідомили про аномальну залежність від діаметра теплопровідності у надтонких NW, тобто теплопровідність збільшується зі зменшенням діаметру, що відрізняється від звичайного розуміння або навіть суперечить йому. Наприклад, Пономарьова та інші в моделюваннях MD виявили [23], що теплопровідність збільшується зі зменшенням діаметру при дуже малому діаметрі (< 1.5 нм). Вони пов'язували це явище з ефектом утримання фононів у надтонких NW. Нещодавно Чжоу з іншими [24] провели систематичне дослідження цієї аномально зростаючої тенденції теплопровідності в надзвичайно тонких NW з використанням симуляцій MD з різними потенціалами. Як показано на рис. 2.1 (б), теплопровідність NW демонструє немонотонну залежність від діаметру, а теплопровідність найтоншого можливого Si NW досягає надвисокого рівня, що більше, ніж на порядок, ніж у об’ємного матеріалу. Конкуруючий механізм між гідродинамічним фононним потоком та граничним розсіюванням відповідає за цю аномальну залежність. У надтонких NW нормальний (N) процес (збереження енергії та імпульсу) низькочастотних акустичних фононів домінує в тепловому переносі, який спричиняє гідродинамічний потік фононів у Si NW без розсіювання.

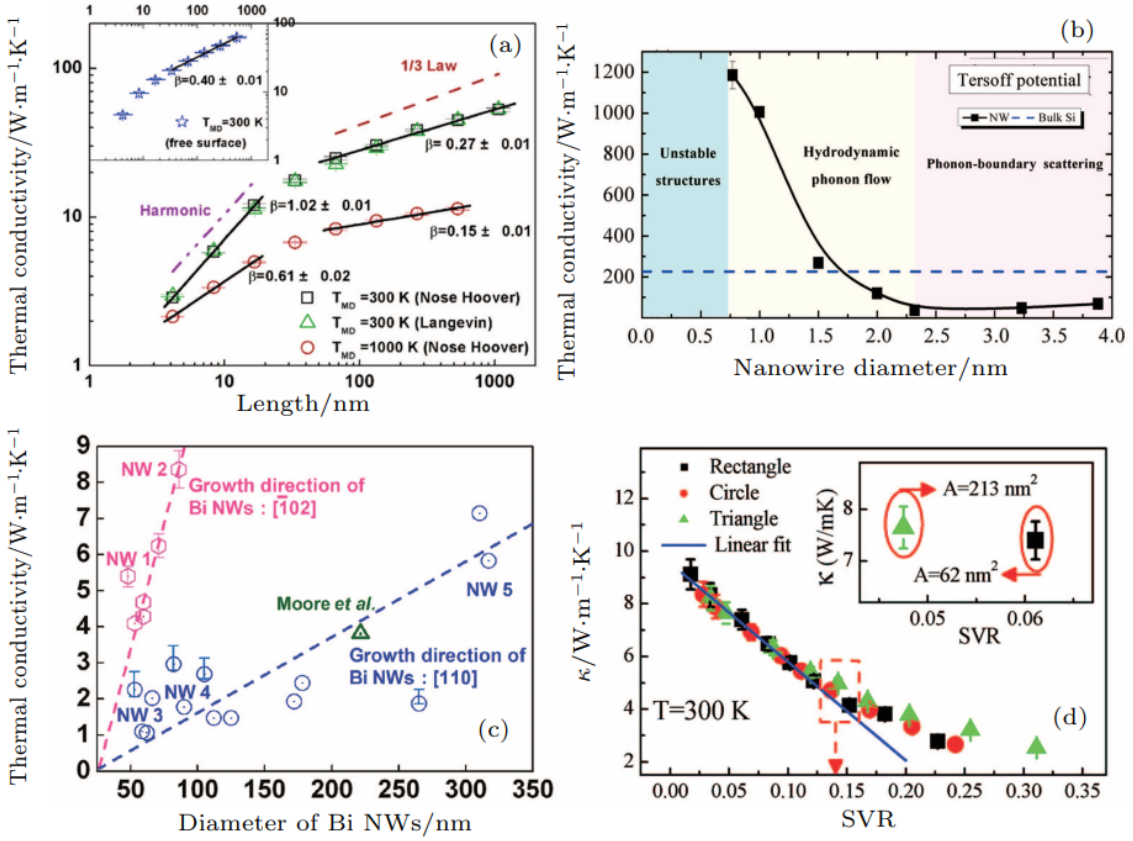


Рис. 2.1. Вплив розміру на теплопровідність NW. (а) Теплопровідність Si NW при кімнатній температурі (з фіксованою поперечною граничною умовою) проти поздовжньої довжини з різними тепловими ваннами. Вставка - вільний (поперечний) випадок граничної умови. (б) Теплопровідність Si NW, залежної від діаметра, обчислена за розрахунками Грін-Кубо з потенціалом Терссофа. (в) Залежна від діаметру теплопровідність Bi NW з напрямками росту ‹102› (рожевий шестикутник) та ‹110› (синій круг) при 300 K. SVR для різних геометрій при 300 K. (г) Теплопровідність ‹100› Si NW відносно SVR для різних геометрій при 300 K.

2.1.3. Вплив поперечного перерізу

Для управління поверхневими розсіюваннями та тепловим переносом у NW були розроблені різні поперечні перерізи. Використовуючи моделювання MD, Чен та інші виявили [21], що теплопровідність Si NW змінюється в залежності від різної геометрії поперечного перерізу (рис. 2.1 (г)), а NW з

трикутним перерізом мають найнижчу теплопровідність під однаковою площею поперечного перерізу. Цікаво, що теплопровідність також монотонно зменшується зі збільшенням SVR, і в межах скромної площі поперечного перерізу (більше 20 $нм^2$) спостерігається проста універсальна лінійна залежність теплопровідності від SVR.

Крім зовнішньої поверхні, фононно-поверхневе розсіювання також може бути індуковане до внутрішніх NW. Чен та інші запропонували [25] ідею знизити теплопровідність Si NW шляхом введення невеликого отвору в центрі перетину NW для побудови структури нанотрубки Si (Si NT). Дуже невеликий отвір (лише на 1% зменшення площі поперечного перерізу) може викликати зменшення теплопровідності при кімнатній температурі на 35%. Більше того, з однаковою площею поперечного перерізу теплопровідність Si NT становить лише близько 33% від тієї, що становить Si NW при кімнатній температурі. Це пояснюється тим, що більше режимів локалізації сконцентровано на внутрішній та зовнішній поверхнях Si NT. Використовуючи високочутливу систему вимірювань, Вінгер з іншими експериментально виявили [26] субаморфну теплопровідність у надтонких кристалічних Si NT. Кристалічні Si NT з товщиною оболонки ~5 нм мають низьку теплопровідність $\sim 1.1 Вт \cdot м^{-1} \cdot К^{-1}$. Мабуть, це значення нижче граничної межі розсіювання і навіть приблизно на 30% нижче вимірюваного значення для аморфних Si NT з аналогічною геометрією.

2.2. Тепловий менеджмент

Контрольована теплопровідність NW також забезпечує ідеальну платформу та матеріали для термічного управління наноматеріалами не тільки для застосування тепловідведення, але й для вивчення унікального явища теплового переносу. Найголовніше, що різні механізми розсіювання можуть домінувати лише в тепловому переносі в певному діапазоні частоти. Наприклад, високочастотні фонони чутливі до домішок / легування та дефектів / пористого розсіювання в різних NW. Крім того, низькочастотні

фонони можуть бути розсіяні взаємодією у надрешітках NW, аморфним розладом у полікристалічному NW та когерентним резонансом у структурах ядро–оболонки NW. Тому в NW теплопровідністю або переносом фононів можна керувати добре контрольованим чином. Наприклад, використовуючи атомне моделювання, явища теплового випрямлення в NW були повідомлені Чжаном та іншими [27]. У градуйованому Si NW (рис. 2.2 (а)) вони виявили, що коефіцієнт теплового випрямлення зростає немонотонно з геометричною асиметрією, як показано на рис. 2.2 (б). Коефіцієнт виявляється різким реверсом, коли кут вершини градуйованого NW приблизно дорівнює 90°. Ця аномальна поведінка походить від відповідності / невідповідності між коливальними спектрами у двох кінцях, як показано співвідношення S_+ / S_- . Інші, в свою чергу, повідомили [28] про теплову ректифікацію в асиметричних двосегментних Si NW. Їх результати показують, що теплопровідність і розсіювання поверхні в чистих NW призводять до теплової ректифікації. Крім того, Лю та інші запропонували [29] градуйований NW із перерізом ядро-оболонка, в якому виявляється очевидна термічна ректифікація. Вони стверджують, що це явище походить від стоячих хвиль, що перешкоджають поширенню фононних хвиль, що схоже на фононні когерентні резонанси в Si–Ge структури NW ядро-оболонка.

Більше того, останнім часом деякі роботи займаються використанням NW для поліпшення теплопередачі та стабільності системи кипіння потоку, яка використовується для охолодження систем з високим термічним навантаженням, таких як електростанції та вбудовані електричні пристрої. Наприклад, Лі з іншими [30] успішно синтезували Si NW на місці у паралельних мікроканальних масивах кремнію та інтегрували NW в мікроканальні тепловідводи. Нарешті, вони встановили, що така установка може значно підвищити теплопередачу киплячого потоку та задавити нестабільність кипіння потоку у NW, яка виступає в якості покриття для системи кипіння потоку. Крім того, Шим з іншими встановили [31], що вирівняні кремнієві NW виявляють покращені властивості тепловіддачі

порівняно зі звичайними випадковими кремнієвими NW. Вирівнювання може значно збільшити критичний тепловий потік за рахунок ефективної подачі теплоносія та забезпечити високу стабільність в системах з надзвичайно високим нагріванням.

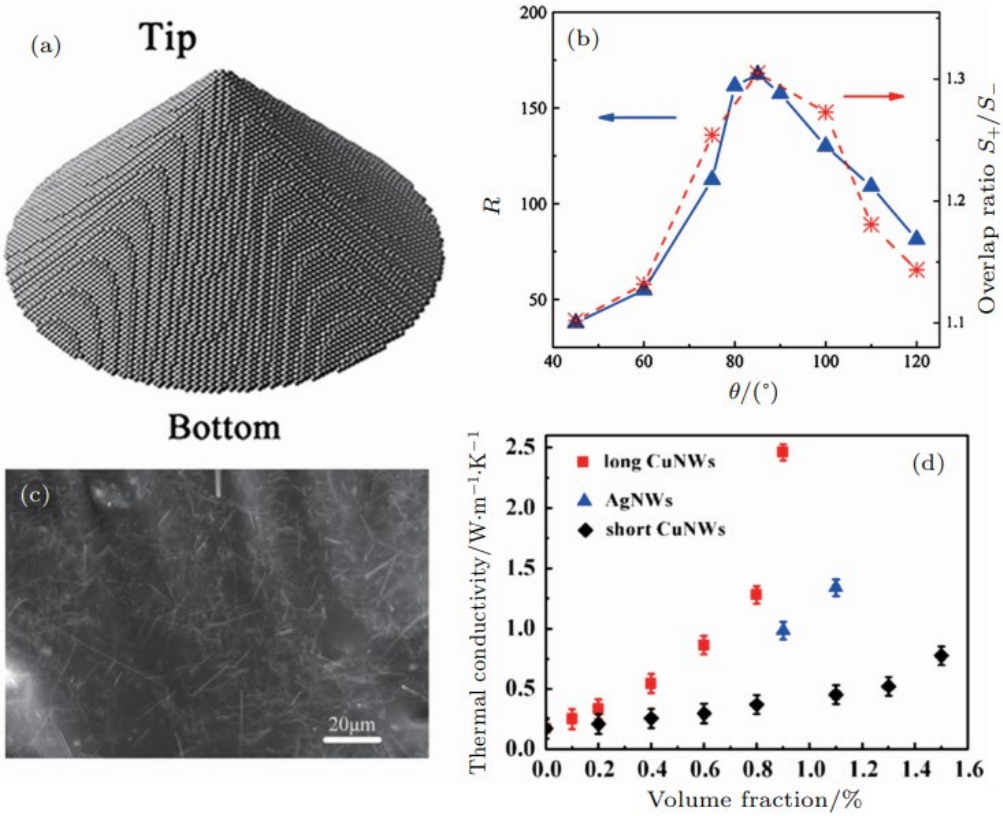


Рис. 2.2. Термічне управління NW. (а) Схематичний малюнок градуйованих Si NW. (б) Коефіцієнт випрямлення (ліва вертикальна вісь) і нормалізований S_+ / S_- коефіцієнт щільності перекриття станів фононів (права вертикальна вісь) як функція кута. (в) SEM-зображення коротко-CuNW-поліакрилатних композитів. (г) Експериментальна теплопровідність металевих NW-поліакрилатних композитів з різними об'ємними частками при 298 К.

Тим часом, експериментально досліджені металеві покриття NW для проточного кипіння. Моршед та інші з'ясували [32], що покриття Cu NW може збільшити однофазну швидкість передачі тепла до ~25%, тоді як у режимі потокового кипіння посилення досягало ~56% при падінні тиску, що зросло

на ~20% у одно-фазовому режимі. Більше того, Ван та інші встановили [33], що Cu NW, як матеріали для заповнення, можуть підвищити міжфазний тепловий перенос. Високе співвідношення пропорцій сприятливе для досягнення низького порогу проникнення для нанокомпозитів, а значення теплопровідності 2.46 Вт· м$^{-1}$· К$^{-1}$ було отримано при наднизькій частці навантаження, яка була збільшена на 1350% порівняно із звичайною матрицею (рис. 2.2 (в) та 2.2 (г)). Хсу та інші [34] запропонували систему для особистого управління теплом з використанням металевого полотна, вбудованого в NW, яке може зменшити відходи тепла. Металеві NW утворюють електропровідну мережу, яка не тільки є високо теплоізоляційною, оскільки відображає інфрачервоне випромінювання людського тіла, але також дозволяє закону Джоуля доповнювати пасивну ізоляцію.

2.3. Розгляд одновимірної моделі

Представимо спрощені моделі зразків теплопровідності, щоб проілюструвати принципи радіаційного переносу тепла, пояснити фізичне походження декількох дивовижних геометричних ефектів та надати засоби перевірки успішної роботи розрахунку Монте-Карло.

Дві паралельні нескінченні чорні плоскі поверхні у вакуумі, розділені на відстань d, утримуються відповідно при температурах T_1 і T_2 (рис. 2.3). Для того, щоб визначити температуру між цими площинами, вважаємо нескінченно малий чорний термометр в рівновазі з полем випромінювання фотона в деякій точці P. Термометр отримує випромінювання з обох площин і досягає рівноважної температури $T_р$, коли він отримує стільки ж енергії, скільки він поглинає. Суцільний кут, що підлягає будь-якій поверхні в точці P, завжди дорівнює 2π, так що отримана випромінювана потужність буде незалежною від позиції P. Отже, температура, виміряна термометром при P, буде однаковою протягом усього об’єму, обмеженого поверхнями, і задається

$$T_P^4 = \frac{1}{2}(T_1^4 + T_2^4). \qquad (2.1)$$

Отже, на поверхнях повинні бути переривчасті стрибки температури, як показано на рис. 2.3 (а). Температура T_P не залежить від розміру термометра. Однак, якщо ввести другий термометр, вони впливатимуть один на одного, якщо тільки обидва не будуть нескінченними. На рис. 2.3 (б) показаний температурний профіль для крайнього випадку нескінченної площини чорного термометра, вставленої між двома чорними поверхнями в положенні, що містить точку *P*. Температура площини термометра також T_p, як дало рівняння (2.1). Другий нескінченно малий термометр вимірює різні температури, T_L або T_R, залежно від того, розташований він ліворуч або праворуч від точки P.

$$T_L^4 = \frac{1}{2}(T_1^4 + T_P^4),$$

$$T_R^4 = \frac{1}{2}(T_P^4 + T_2^4). \qquad (2.2)$$

Стрибки температури тепер також відбуватимуться з обох боків площини вставленого термометра.

Тепер розглянемо два термометри кінцевого розміру між площинами. Вони будуть схильні кидати тіні один на одного, тобто кожен термометр бачить суцільний кут менше 2π від однієї з площин. Натомість, кожен також отримає випромінювання від іншого термометра, який знаходиться за деякої проміжної температури. Тому різниця температур ΔT буде спостерігатися між двома термометрами, що залежатиме від фізичного розміру двох термометрів, а також від їх розташування. Коли термометри невеликі, вони кидають невеликі тіні, і різниця температур буде невеликою. Якщо ми спробуємо виразити ΔT з точки зору ефективної теплопровідності радіаційного поля, то менші термометри повідомлять більшу теплопровідність. Нескінченно малі термометри давали б $\Delta T = 0$ або нескінченну теплопровідність. Звичайно, термін теплопровідність є помилковим в даному випадку.

Результати, отримані для двох нескінченних площин, залишаються незмінними, якщо ідеально дзеркальні стінки вставлені перпендикулярно площинам, щоб представити зразок теплопровідності з ідеально дзеркальними

поверхнями. Температурний профіль вздовж цього зразка обумовлений виключно термометрами. Якщо стіни не ідеально блискучі, а частково дифузно розсіяні, то температурний профіль визначається частково термометрами і частково стінками.

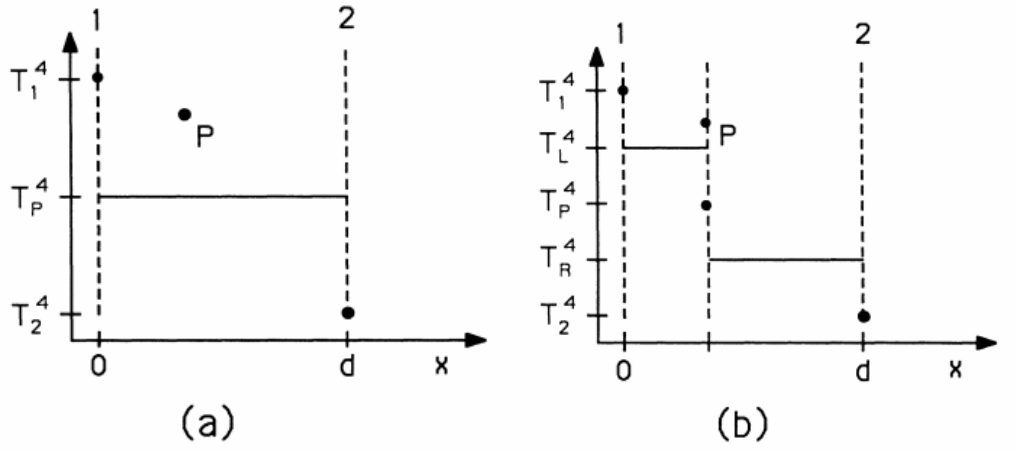


Рис. 2.3. (а) Дві нескінченні площини (штрихові лінії) при температурах T_1 та T_2 розташовані при x = 0 та x = d. Точка P лежить десь в об'ємі, обмеженому двома площинами. Температура в точці P не залежить від її розташування між площинами. У площинах відбуваються переривчасті стрибки температури. (б) Третя чорна площина вставлена між іншими двома, щоб вона містила точку P. Тепер додаткові стрибки температури відбудуться на лівій грані вставленої площини, а також на правій. Температура, виміряна нескінченно малим термометром, тепер залежить від того, розташований він праворуч (T_R) або ліворуч (T_L) від вставленої площини.

Для того, щоб забезпечити точно розчинну модель фононної випромінювальної теплопередачі, яка включає вплив приєднаних термометрів і дифузне розсіювання фононів на вільних поверхнях зразку, треба розглянути спрощене представлення геометрії зразків, що використовуються в експериментах. Заміняємо прямокутний прутоподібний зразок на дзеркальний циліндричний зразок і відображаємо дифузно розсіяну поверхню зразка поряд термізуючих поверхонь (або листів), перпендикулярних осі окулярного

циліндра. Це представлення зразків називається «моделлю аркушів». На рис. 2.4. (а) показано циліндричне тверде тіло діаметром D, безрозмірне співвідношення довжини до діаметру L, площа поперечного перерізу $A = \pi D^2/4$ та з ідеально дзеркальними поверхнями. Хоча в моделі використовується циліндр кругового перерізу, результати не змінюються для прямокутних перерізів, що мають однакову площу A. N листів, які представляють поверхню зразка, позначені від 1 до N, тоді як кінці зразка представляємо за аркуші з позначкою 0 та N + 1. I-му аркушу надається випромінювання, яке пов'язане з ймовірністю дифузного розсіювання f поверхонь зразка в експериментах, і тому для наочної зручності це називається як f_i. I-й лист при температурі T_i потім фіксує частку f_i всього фононного випромінювання, що потрапляє на нього, і споживає потужність $f_i \sigma T_i^4$ у кожному напрямку. Оскільки листи термізуються, фононове випромінювання, поглинене кожним листом, підлягає кутовому розподілу $cos\theta$ відповідно до закону Ламберта щодо випромінювання чорних тіл, де θ – кут відносно норми поверхня аркуша. Оскільки кожен лист відбивається до нескінченності стінками дзеркального циліндра, ми фактично маємо одновимірну модель теплопередачі. Зокрема, тепловий потік уздовж циліндру листів рівномірний за його перетином, оскільки ігнорується еластична анізотропія в цій моделі. Тому кут випромінювання θ не має значення, і потрібно враховувати лише ту сторону листа, з якої випромінюється фонон.

Обмежуємо кінці циліндра, уточнюючи, що 0-й аркуш не випромінюється ліворуч і що (N+1)-аркуш не випромінюється праворуч. Якщо прийняти обидва кінці чорними ($f_0 = f_{N+1} = 1$), тоді не потрібно відслідковувати численні відбиття від торців циліндру, і обговорення теплопередачі набуває дуже простої форми. Можна визначити константу фонона Стефана-Больцмана аналогічно тій, що використовується для випромінювання фотонів

$$\sigma \equiv \frac{\pi^2 k^4}{40\hbar^3} \frac{1}{3} \overline{\sum_{i=1}^{3} \frac{1}{v_i^2}}, \qquad (2.3)$$

де v_i – групові швидкості трьох різних фононних режимів, а смуга позначає середню за напрямами поширення (для випадку пружно-анізотропного кристала). Для кремнію σ становить близько 18 мВт / см^{2}K^4 .

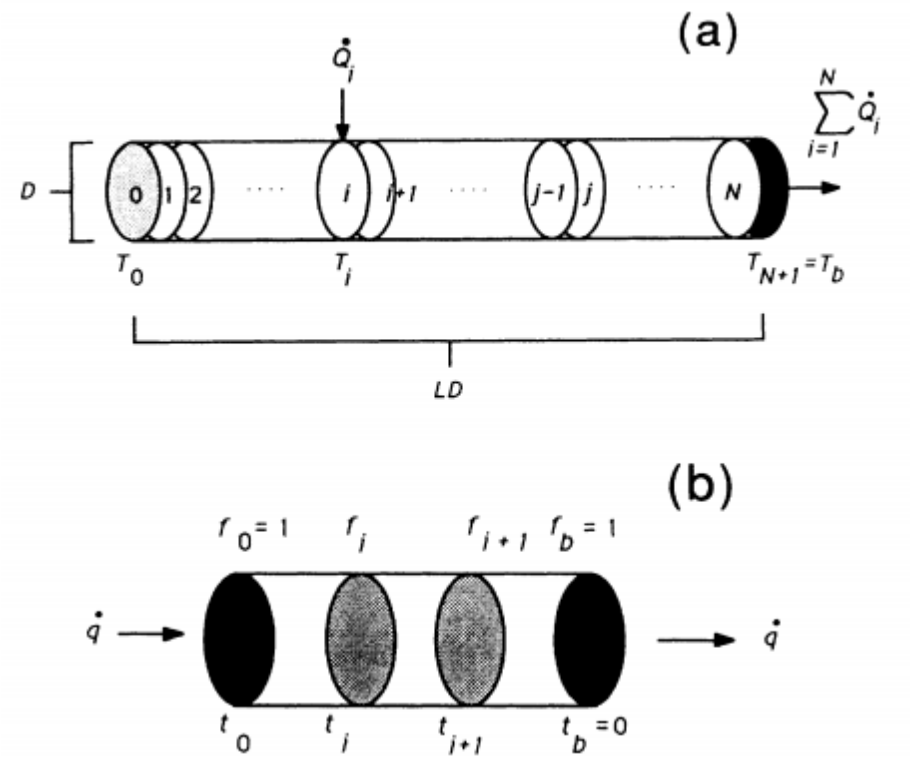


Рис. 2.4. (а) Одновимірна модель радіаційної теплопередачі. Кожен елемент поверхні зразка, включаючи затиснуту ділянку, представлений термоізолюючим листом, перпендикулярним до протилежної дзеркальної трубки з площею поперечного перерізу A, яка поглинає частку f_i випромінювання, що проходить через неї. Все тепло проходить через крайній правий елемент, який представляє точку, в якій зразок притискається до кріостату. (б) Скорочена модель 1D передачі тепла для обчислення падіння температури між елементами i та $i+1$ моделі.

Нехай $\dot{Q}$ – зовнішня теплова потужність, що подається на i-й лист (для $\dot{Q}_i > 0$), або тепло, отримане з i-го листа (для $\dot{Q}_i < 0$). Нехай a'_{ij} дорівнює добутку ймовірності викиду з листа j та поглинання в аркуші i без поглинання будь-яким із листів між i та j, або

$$a'_{ij} = f_i f_j (1 - f_{j-1}) \cdots (1 - f_{i+1}) \text{ для } j > i \qquad (2.4)$$

і $a'_{ij} = a'_{ji}$. Система листів знаходиться в рівновазі, коли сума променевої енергії, отриманої від усіх інших листів, і потужності, що надходить від навколишнього середовища, дорівнює потужності, випромінюваної кожним листом. Тобто

$$A\sigma \sum_{j(\neq 1)} a'_{ij} T_j^4 + \dot{Q}_i = \delta_i \sigma f_i A T_i^4 \qquad (2.5)$$

Для i = 0 до N, де δ_i = 2 для 1≤ i ≤ N та δ_i = 1 для i = 0 та N + 1. Ця система рівнянь має рішення з усіма $T_i^4 > 0$ лише тоді, коли $\sum_{i=0}^{N+1} \dot{Q}_i = 0$. Отже, ми можемо усунути одне з цих N + 2 рівнянь з цим обмеженням, і найприродніше це зробити шляхом ідентифікації (N + 1)-го аркуша з основою зразка при певній фіксованій температурі T_b, через яку всі теплові входи залишають зразок. Тоді $T_b \equiv T_{N+1}$, $f_b \equiv f_{N+1}$, і $\dot{Q}_{N+1} = -\sum_{i=0}^{N} \dot{Q}_i$ і ми усуваємо базу від подальшого розгляду. Ми можемо переписати решту N рівнянь у матричній формі як

$$A\sigma \underline{a}(T^4 - T_b) = \dot{Q}, \qquad (2.6)$$

де $a_{ij} = - a'_{ij}$ ($i \neq j$), $a_{ij} = 2f_i$ ($i \neq 0$) та $a_{00} = 1$. Використовується ідентичність $\sum_{i=0}^{N+1} a_{ij} = 0$ для кожного i, що виражає той факт, що фонон, випромінюваний з однієї поверхні, з часом повинен бути поглинений десь в іншому місці. Елементи T^4 – це T_i^4, а елементи $\dot{Q}$ – це $\dot{Q}_i$.

Якщо падіння температури вздовж зразка невелике порівняно з T_b, зручно перетворити рівняння (2.5) від лінійного в T^4 до лінійного в T. Визначаємо понижену температуру $t_i = (T_i - T_b) / T_b$ так, що $T_i^4 = T_b^4(1 + t_i)^4 \approx T_b^4 + 4T_b^4 t_i$ для $t_i << 1$. Переставляємо для отримання

$$t = \underline{a}^{-1} \dot{q},$$

де $\dot{q} = \dot{Q}_i/4\sigma T_b^4 A$ і $\underline{a}$ такі ж, як і раніше. Знижена температура в основі, t_b, тепер дорівнює нулю. Коли тільки $\dot{q} \equiv \dot{q}_0$ ненульове, проблема легко вирішується. Розглянемо листи i та $i + 1$ окремо. Через 1D-характер проблеми, чиста $\dot{q}$ повинна бути однаковою між будь-якими двома листами. Середня температура цих двох елементів невідома, оскільки додавання рівномірної температури до всіх елементів не призводить до відсутності чистого теплового потоку. Однак чистий радіаційний обмін між i та $i + 1$, а отже, і різниця температур $t_i - t_{i+1}$, ідентична значно спрощеної та легко вирішуваної моделі рис. 2.4 (б), для якої $f_0 = f_b = 1$, $f_1 = f_i$, та $f_2 = f_{i+1}$. Усуваємо крайній правий лист і отримуємо

$$\underline{a} = \begin{pmatrix} 1 & -f_i & f_{i+1}(1-f_i) \\ -f_i & 2f_i & -f_i f_{i+1} \\ -f_{i+1}(1-f_i) & -f_i f_{i+1} & 2f_{i+1} \end{pmatrix}. \qquad (2.7)$$

У цьому випадку $\underline{a}$ може бути легко інвертований, і ми знаходимо

$$t_i - t_{i+1} = \dot{q}(a_{10}^{-1} - a_{20}^{-1}) = \dot{q}\left(\frac{f_i + f_{i+1} - f_i f_{i+1}}{(2-f_i)(2-f_{i+1})}\right). \qquad (2.8)$$

Цей результат стосується всіх i.

Для імітації вибірки рівномірного f, нехай $N \to \infty$ та $F_i = fL / N$ для $i \neq 0$, $N + 1$. Застосовуючи рівняння (2.8), знаходимо падіння температури $\dot{q}fL/2N$ між сусідніми листами для $i \neq 0$, N, що для N / LD листів на одиницю довжини дає рівномірний градієнт температури $\dot{q}f/2D$. На кінцях, однак, $t_0 - t_1 = t_N = \dot{q}/2$, оскільки $f_0 = 1$ та $f_1 \to 0$, $f_N \to 0$ з $f_{N+1} = f_b = 1$. Оскільки аркуші розташовуються довільно близько в межі, ми маємо переривчасті стрибки $\dot{q}/2$ на кожному кінці з лінійним профілем між ними.

Для чистих відшліфованих зразків площа під затискачами термометра розсіюватиме випромінювання набагато ефективніше, ніж рівна площа оголеної поверхні. На рис. 2.5 показано типовий зразок з термометрами на H і

C та нагрівачем у верхній частині, що має стінки ймовірність дифузного розсіювання *f*.

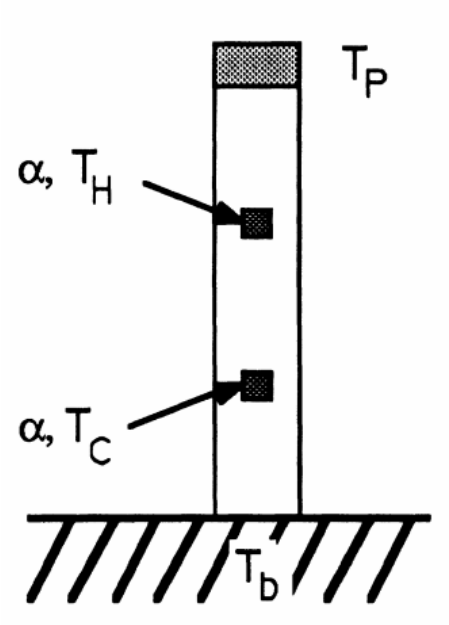


Рис. 2.5. Типові місця розташування обігрівачів, затискачів термометра та радіатора основи на зразку. Двонагрівальний метод вимірювання теплопровідності підводить тепло при *H* або *C* і вимірює $T_{\text{р}}$. Двотермометровий метод підводить тепло до *P* і вимірює T_{H} і T_{C}. Заштриховані квадрати на *H* і *C* представляють площу під затискачем, яка розсіює фонони набагато сильніше, ніж чиста поверхня. Частина *a* фононів, що проходять вниз по зразку, розсіяно дифузно по хомутах.

Без затискача частина зразка представлена так як тільки що описана, тоді як затискачі представлені окремими аркушами з $i = h, c$ та з $f_{\text{h}} = f_{\text{c}} = a$, де *a* – ймовірність того, що затискач розсіює випромінювання під час проходження через зразок. Знову застосуємо (2.8) з $i = h - c, h, c - 1, c$ та зазначимо

$$f_{h-1} = f_{h+1} = f_{c-1} = f/N \to 0,$$

находимо стрибок $\dot{q}a/2(2-a)$ між елементами $h - 1$ та h, а також між h та $h + 1$, з тим же результатом для *c*. Таким чином, вимірювана різниця температур $T_{\text{h}} - T_{\text{c}}$ включає внесок

$$\frac{\dot{Q}a}{4\sigma T_b^3 A(2-a)}$$

від самих затискачів, що в моделі аркушів не залежить від їх поділу.

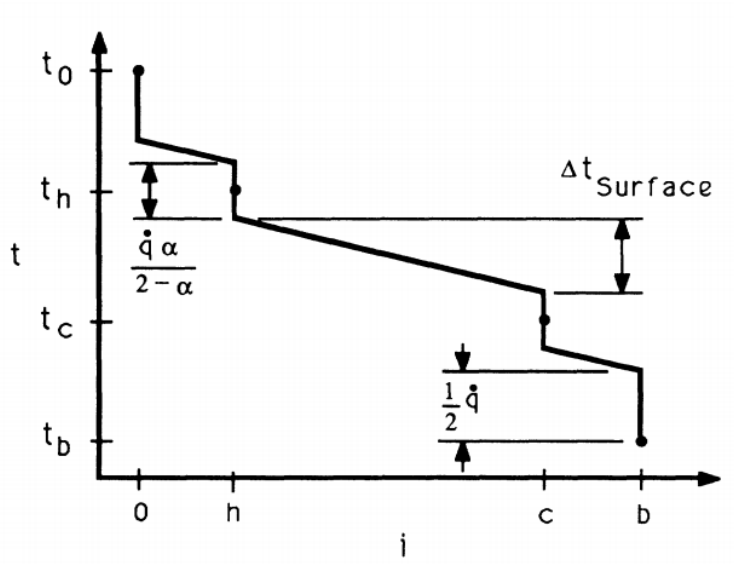


Рис. 2.6. Лінійний температурний профіль у моделі листів, що представляють зразок з ненульовою дифузною ймовірністю розсіювання *f* на поверхні та із затискачами, які розсіюють частку *a* = 1/3 фононів, що проходять повз них.

Температурний профіль зразка з $f \neq 0$ і з приєднаними термометрами показаний на рис. 2.6. Лінійний профіль з градієнтом $\dot{q}f/2D$ має дискретні стрибки на кінцях, незалежних від довжини зразка або наявності затискачів, а також стрибки на затискачах, пропорційних їх площі у вигляді частки від загального перерізу зразка. Попередні дослідники припускали лінійний профіль уздовж усієї вибірки, нехтуючи стрибками на кінцях та на затискачах. Для кінцевих зразків високої зернистості це призводить до серйозних помилок при перетворенні вимірювання ($T_h - T_c$) у кількісну характеристику (*f*), як $f \rightarrow 0$. Нехтування стрибками температури на кінцях дає розрахунковий *f*, який занадто низький; нехтування стрибками температури на затискачах дає розрахунковий *f*, який занадто високий.

РОЗДІЛ 3. МОДЕЛЮВАННЯ ПЕРЕНЕСЕННЯ ТЕПЛА ФОНОНАМИ В НАНОДРОТАХ

3.1. Вплив довжини вільного пробігу фононі на теплопровідність

Вже давно було запропоновано, що зниження теплопровідності за рахунок зниження довжини вільного пробігу фонону являється одним з найкращих способів поліпшити поточні показники термоелектрики. Вимірюючи теплопровідність та питому теплопровідність нанодротів та тонких плівок, можна показати різні способи збільшення розсіювання фононів від низькотемпературних до експериментів з кімнатною температурою. Показано, що граючи з геометрією (звуження, періодичні структури, нано-включення), від балістичної до дифузійної межі, фононний тепловий перенос може бути сильно змінений в монокристалічних напівпровідних структурах; середнє значення фононного вільного шляху внаслідок цього зменшується.

Протягом двох останніх десятиліть матеріали, структуровані на нанорозмірних покриттях з низькою теплопровідністю, привернули багато уваги завдяки їх високо потенційним термоелектричним застосуванням [35-38]. Дійсно, як запропонували Дресселгауз та інші в 1993 році [39,40], наноструктура може значно підвищити термоелектричну ефективність. Ця ефективність термоелектричних матеріалів визначається безрозмірною показником достовірності $ZT = S^2T\sigma/k$, де S – коефіцієнт Зеєбека, σ електропровідність, k – теплопровідність. Якщо зосередитись лише на теплових властивостях, ZT буде збільшено за рахунок зменшення фононного теплового переносу. Це відображає той факт, що джерело тепла та джерело холоду повинні бути максимально термічно ізольованими, щоб запобігти занадто сильному витоку тепла, оскільки це погіршить загальну продуктивність термоелектричних модулів.

Для електронів теплові та електронні властивості пов'язані законом Відемана-Франца. Таким чином, єдиним способом зниження теплового

переносу є зниження фононної теплопровідності. Існують два основні напрямки для впливу на фононні властивості: некогерентні та когерентні ефекти. З одного боку, некогерентними ефектами керує середня довжина вільного пробігу фонову (MFP), тоді зменшення отримується додаванням центрів розсіювання, зменшенням розміру, шорсткою поверхнею або додаванням нано-включень. З іншого боку, когерентні ефекти проявлятимуться через зменшення групової швидкості (сплющення смуги), відкриття проміжків смуги або деструктивні перешкоди. Дві значні шкали довжини керують фізикою фононів, включаючи некогерентні або когерентні ефекти: середня довжина вільного пробігу (Λ) фонону та домінуюча довжина хвилі фонону, задана:

$$\lambda_{dom} = \frac{h\nu_s}{2.82k_BT} \tag{3.1}$$

де h – постійна Планка, k_B – константа Больцмана і ν_s – швидкість звуку. Ця довжина дійсно залежить від температури, зростаючи зі зниженням температури (рис. 3.1).

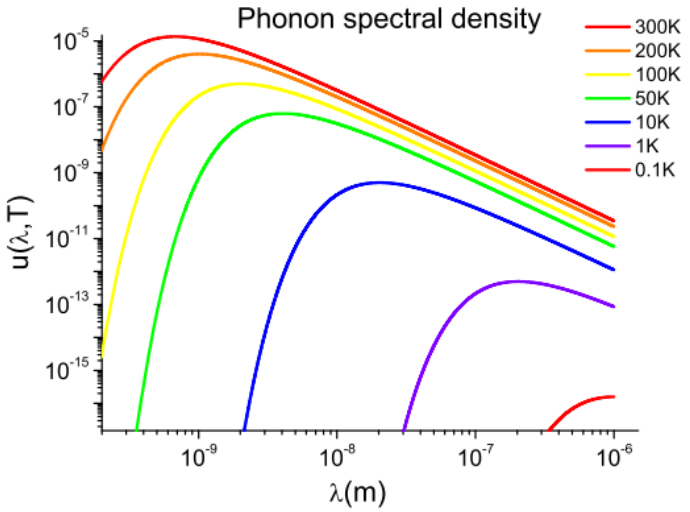


Рис. 3.1. Спектральна щільність фонону як функція довжини хвилі для різних рівноважних температур чорного тіла. Домінуюча довжина хвилі фонону для кожної температури відповідає максимуму кривих. Як правило, домінуюча довжина хвилі фонону становить порядку 1 нм при 300 К і 100 нм при 1 К.

3.1.1 Балістична межа: тепловий перенос у Si нанодротах

У маломірних системах поверхневе розсіювання фононів відіграє вирішальну роль у розумінні теплового переносу. Так званий режим Казимира виникає тоді, коли об'ємна MFP об'ємного фонону набагато більша за фактичний розмір досліджуваної наносистеми. У цьому випадку MFP фонону встановлюється лише розсіюванням по поверхнях. Ця межа досягається при досить низькій температурі, коли інші механізми розсіювання не домінують (взаємодія фонон-фонон, взаємодія електрон-фонон, розсіювання на домішки тощо). В результаті ефекти, пов'язані з шорсткістю поверхні, мають істотний вплив на фононний перенос [44, 45]. Ці ефекти можуть бути простими некогерентними розсіяннями, коли довжина хвилі фонону набагато більша за середню швидкість або при зниженні температури може стати залежною від довжини хвилі. Дійсно, якщо довжина хвилі більша за середню шорсткість, на поверхні можуть виникати дзеркальні відбиття фононів. Якщо бере участь пружне розсіювання, енергія зберігається, що призводить до збільшення MFP.

relevant length scales	transport regime	models	relevant formulae
$\lambda_{dom}(T) < \Lambda_{ph}(T) < L$	Diffusive regime (incoherent)	Kinetic model	$K = \frac{1}{3} C \upsilon_s \Lambda_{ph}$
$\lambda_{dom}(T) < L \sim \Lambda_{ph} < \Lambda_{ph\text{-}bulk}$	Casimir regime (incoherent)	Casimir model	$K_{Cas} = \beta \Lambda_{Cas} T^3$
$\lambda_{dom}(T) \sim L < \Lambda_{ph}$	Ziman regime (partly coherent)	Ziman model	$\Lambda_{ph} = \frac{1+p}{1-p} \Lambda_{Cas}$
$L < \lambda_{dom}(T) < \Lambda_{ph}$	Ballistic regime (coherent)	Landauer model	$T = \frac{1}{1 + 3L / 4\Lambda_{eff}}$

Рис. 3.2. Класифікація різних режимів переносу фононів у нанорозмірних системах при досить низькій температурі, як правило, нижче 30 К для кремнію. L – довжина вибірки, Λ_{ph} – середня довжина вільного пробігу нееластичного фонону, а λ_{dom} – домінуюча довжина хвилі фонону. Щодо межі Ландауера, параметр T позначає коефіцієнт пропускання між нанодротом та його тепловим резервуаром.

3.1.2 Нормальний перенос фононів у нанодротах Si при низькій температурі

Для розуміння теплового переносу при низькій температурі необхідний короткий виклад моделей Казимира та Зімана. У моделі Казимира [41, 43] теплопровідність задається:

$$K_{Cas} = 3.2 \cdot 10^3 \left(\frac{2\pi^2 k_B^4}{5\hbar^3 v_s^3}\right)^{\left(\frac{2}{3}\right)} \frac{e \cdot w \Lambda_{Cas}}{L} T^3 = \beta_{Cas} T^3 \qquad (3.2)$$

де k_B – константа Больцмана, ħ – константа Планка і v_s відноситься до швидкості звуку. $\Lambda_{Cas} = 1.12\sqrt{e \cdot w}$ – MFP Казиміра з фононами у нанодроті, що має прямокутний переріз, e відноситься до товщини та w до середньої ширини нанодроту; L – його довжина. Ця формула виходить, коли середня довжина вільного пробігу фонону обмежена лише поперечним перерізом дроту, тобто межовим розсіюванням. Дійсно, у цій межі передбачається, що кожен фонон, що потрапив на шорстку поверхню, поглинається повторно випромінюваним у будь-який бік та енергію відповідно до закону випромінювання чорного тіла (рис. 3.1). Іншими словами, у цій межі Казимира поверхня вважається нескінченно шорсткою незалежно від довжини хвилі фонона.

По мірі зниження температури домінуюча довжина хвилі фонона фактично зростає (рівняння (3.1)) і з часом стає більшою за амплітуду шорсткості; тоді дзеркальні відбиття стають більш імовірними і ними не можна нехтувати, як у моделі Казимира. Щоб врахувати цю тенденцію, середню довжина вільного пробігу Казимира слід замінити у рівняння (3.2) на ефективний MFP Λ_{eff}, запропонований Займаном та його колегами [42, 43]:

$$\Lambda_{eff} = \frac{1+p}{1-p} \Lambda_{Cas} \qquad (3.3)$$

де p – параметр, що описує ймовірність того, що фонон може бути дзеркально відображений на поверхнях. Значення цього параметра буде залежати як від

довжини хвилі фононів, так і від шорсткості поверхні дроту. Суто дифузійний випадок Казимира задається при $p = 0$, а чисто дзеркальний, коли $p = 1$, буде називатися балістичною межею; MFP розходиться до нескінченності. Модель Зімана враховує значення p, яке відрізняється від нуля; при зниженні температури середня довжина вільного пробігу фонону може бути більше, ніж переріз нанодроту. Різні режими переносу фононів та пов'язані з ними моделі узагальнені в таблиці, наведеній на рис.3.2.

Як висновок, тепловий перенос в кремнієвих нанодротах, що мають перетин 100 нм, буде добре описаний моделлю Казимира, а потім при більш низькій температурі моделлю Зімана. Дійсно, довжина L менше, ніж середня довжина вільного пробігу об'ємного фонону $\Lambda_{ph\text{-}bulk}$ в Si, та на той же порядок величини, що і домінуюча довжина хвилі фонона. При ще більш низькій температурі повністю балістичну межу слід описати моделлю типу Ландауера, де тепловий перенос задається коефіцієнтами пропускання між нанодротом і тепловою ванною. На даний момент експериментальних даних у цьому режимі немає.

3.1.3 Блокування фононів з S-подібними шиканами в Si нанодротах

Теплопровідність нанодроту була виміряна за допомогою поздовжньої технології 3ω з високочутливою термометрією на основі нітриду ніобію (NbN) [46, 44, 45, 47]. Si нанодроти виготовляються з кремнію на ізоляційних підкладках методом електронного променя. Загальна довжина кожної конструкції була спеціально встановлена на 10 мкм, щоб легко порівняти теплові вимірювання всіх видів нанодротів (рис. 3.3). Переріз нанодротів становить від 100 нм до 200 нм. Центральна симетрія конструкцій необхідна для методу 3ω, який використовується для вимірювання теплопровідності [46].

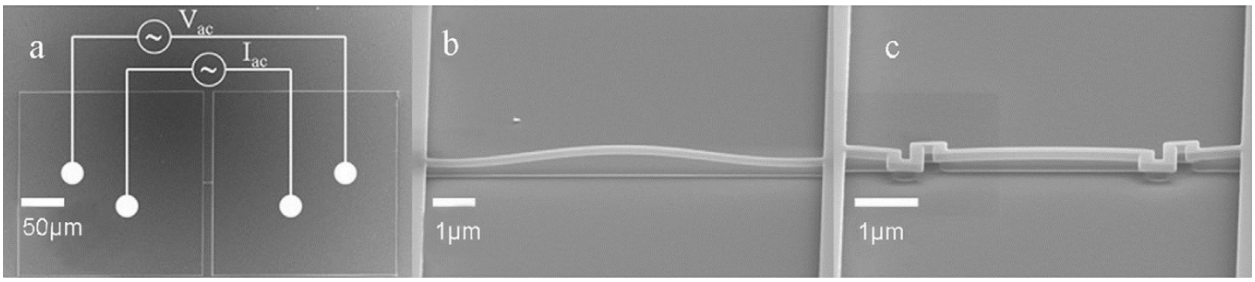


Рис. 3.3. SEM-зображення: (а) підвісного нанодрота, пов'язаного з двома колодками, де електричні з’єднання дозволяють проводити 3ω вимірювання; (б) прямий нанодріт; (в) відповідає виду збоку S-подібного нанодроту.

Температурний профіль виглядає як парабола з найгарячішою точкою, розташованою в центрі нанодроту. Тепло, що утворюється в кожній точці підвісної конструкції, тече від центру до теплової бані з обох сторін. Теплопровідність, як правило, вимірюється за допомогою потужності кількох десятків фемтоватів, що розсіюються в перетворювачі нітриду ніобію, нанесеному поверх нанодротів. Ця потужність створює в середньому градієнт температури менше 1 мК. Значення вимірюваної теплопровідності приблизно в декількох універсальних значеннях теплопровідності $K_0 = \pi^2 k_B^2 T/3h \sim 10^{-12} T$ Вт/К^2; значення, яке використовується для нормалізації даних.

Рівняння (3.2) дає правильний порядок величини для теплопровідності, оскільки значення *K*, близьке до 10^{-11} Вт/К, прогнозується при 3 К, а значення $1{,}5 \times 10^{-11}$ Вт / К вимірюється по прямому нанодроту. Для пояснення зміни температури теплопровідності необхідно враховувати внесок дзеркального відбиття, вводячи середню довжину вільного пробігу Зімана, а отже, і параметр *p*. Треба підкреслити, що внесок балістичних фононів у тепловий перенос є вагомим. Запропоноване рішення для управління теплом не порушить дифузійний перенос, а вплине лише на балістичний. Були зроблені специфічні підвісні нанодроти, включаючи подвійні вигини шиканових конструкцій довжиною 400 нм; загальна розширена довжина S-подібного нанодроту стане ідентична довжині прямого нанодроту (рис. 3.3 (в) та рис. 3.4).

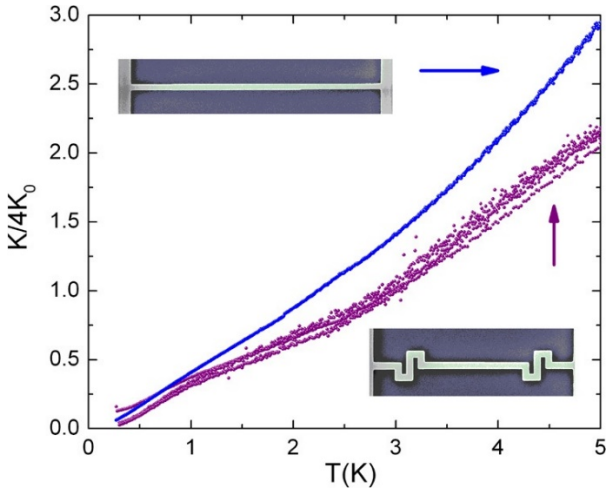


Рис. 3.4. Теплопровідність нанодроту довжиною 5 мкм нормалізується до чотирьохкратного загального значення теплопровідності проти температури. Теплопровідність прямого нанодроту набагато більша, ніж теплопровідність нанодротів S-форми. Наявність серпантинового перерізу зупиняє передачу балістичних фононів, зменшуючи загальний середній вільний шлях фонону.

S-подібні нанодроти мають набагато нижчу теплопровідність, ніж прямі, як це видно на рис. 3.4. Якби перенос був дифузійним, це перетяжка взагалі не мала б впливу на теплову провідність. Власне, суттєве скорочення, що спостерігається (зменшення більш ніж на 30%), свідчить про два суттєві факти: по-перше, це ілюструє внесок балістичних фононів у тепловий перенос, а по-друге, включаючи шикан в тепловому провіднику, це дуже ефективний спосіб зменшення теплового переносу. Фонони, що мають довгу середню довжину вільного пробігу (більше, ніж середня довжина вільного пробігу Казиміра, Λ_{Cas}), будуть заблоковані та відбиті шиканом, що діє як дзеркало для хвиль акустичної площини довгих хвиль. І навпаки, короткохвильові фонони, які більше розсипаються нанодротовими поверхнями, не будуть сильно впливати на шикан, їх середня довжина вільного пробігу вже встановлена секцією нанодроту, як у режимі Казимира. Ця концепція успішно використовується для теплової ізоляції підвісної платформи, призначеної для чутливих теплових вимірювань [49].

3.1.4 Знижений тепловий перенос у гофрованих нанодротах

Проілюструємо, як можна ввести фононне зворотне розсіювання або багаторазове розсіювання за допомогою наноінженерних поверхонь. У цьому відношенні фононні кристали, визначені як регулярна модуляція геометрії фононного хвилеводу, за останні десятиліття привернули значну увагу як перспективний засіб зменшення переносу фононів. У цих зразках періодичні наноструктури використовуються (рис. 3.5) для індукції зворотного розсіювання фононів і некогерентно знижують теплопровідність, а не когерентно, як у справжнього термокристала. За допомогою такого типу інженерних гофрованих нанодротів фононний перенос може бути зменшений навіть нижче межі Казимира [48, 50].

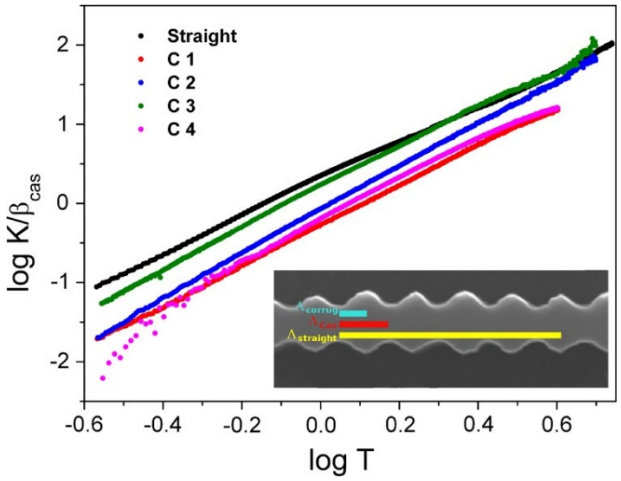


Рис. 3.5. Нормалізована теплопровідність проти температури для прямого нанодроту (прямого) та чотирьох гофрованих нанодротів (з C1 до C4) на подвійному логарифмічному графіку. Коефіцієнт нормалізації β_{Cas} відповідає граничному значенню Казимира теплопровідності в 1 К. Гофрований нанодріт C_i може мати набагато меншу теплопровідність, ніж гладкий нанодріт. MFP фонону різко знижується, навіть нижче межі Казимира. Як ілюстрацію, на вставці порівнюється середня довжина вільного пробігу у прямому нанодроті (жовтий колір), межі Казимира (червоний колір) та вимірюваному у гофрованому нанодроті (синій колір). Гофри значно скорочують середній вільний шлях фонону.

Щоб порівняти тепловий перенос між різними геометріями нанодротів, теплопровідність повинна бути нормалізована до геометричного коефіцієнта, який називається коефіцієнтом Казіміра β_{Cas}, згідно з рівняннями (3.2) і (3.3). Рис. 3.5 повідомляє про теплопровідність, побудовану як K / β_{Cas}.

Ця нормалізація дає закон про енергетичну потужність і дозволяє безпосередньо порівняти тепловий перенос кожного нанодроту. Дійсно, від рівнянь (3.2) і (3.3) нормована електропровідність може бути виражена як:

$$\log\left(\frac{K}{\beta_{Cas}}\right) = a + b\log(T) \tag{24}$$

де

$$\left(\frac{K}{\beta_{Cas}}\right) = \frac{1+p}{1-p}T^{b} \tag{25}$$

З цих двох рівнянь можна витягнути коефіцієнт закону потужності b зміни температури теплопровідності, значення MFP Λ_{eff} та параметр p [48, 50].

Теплопровідність гофрованих нанодротів явно нижче теплопровідності прямих нанодротів, навіть якщо закон про потужність не був змінений; вона дуже близька до 3 ($b \approx 2.6$). Наявність штучної періодичної наноструктури збільшує розсіювання фононів (зворотне розсіювання, захоплення фононів тощо). Шорсткість поверхні не змінюється, але середня довжина вільного пробігу явно знижується. Це можна зобразити методом променевого відстеження за допомогою чисельних моделювань Монте-Карло. Дійсно, багаторазове розсіювання балістичних фононів можна побачити через корпускулярну картину, що пояснює той факт, що фонони якимось чином потрапляють у ядро нанодроту. Як наслідок, MFP фонону може бути навіть меншим, ніж межа Казимира, тобто менший за середній діаметр нанодроту. Це характеризується необхідністю розміщення від'ємного параметра p для відповідності даним.

3.1.5 У дифузійній межі: нано включення обмежують теплопровідність у напівпровідниках

Інженерія теплових властивостей фонону можлива також за допомогою напівпровідника при кімнатній температурі (*RT*). Порівняно з балістичними межами, робота на *RT* у дифузійному межі матиме значні наслідки для реальних застосувань для управління нагріванням, охолодженням або збиранням енергії.

Оскільки, з одного боку, напівпровідник занадто теплопровідний (близько 100 Вт / мК), з іншого боку, його коефіцієнт Зеєбека, як правило, досить високий (вище 200 мкВ / К). Тоді отримати ефективний термоелектричний матеріал, що має високий *ZT* на основі напівпровідника, можливо, значно зменшивши його теплопровідність.

Підвищення термоелектричної ефективності можливо завдяки різниці середньої довжини вільного пробігу між фононами (~100 нм при 300 К) та електронами (~1 нм при 300 К). Вводячи в даному матеріалі структурний розлад на шкалі нанометрів, можна індукувати фононну дифузію, не впливаючи на носіїв заряду, обґрунтовуючи інноваційну концепцію матеріалу електронного кристала-фононного скла.

Дійсно, у напівпровідниковій матриці фонони найбільше сприяють тепловому переносу, маючи середній вільний шлях ~100 нм при кімнатній температурі та домінуючу довжину хвилі 1 нм при *RT*. З метою маніпулювання та контролю переносу фононів в SC, інженерія матеріалів може потім дотримуватися двох різних підходів: або шляхом гри з довжиною хвилі фонона [51 – 54], або із середньою довжиною вільного пробігу фонону [50, 53, 54]. При кімнатній температурі найбільш ефективним ефектом буде зменшення середньої довжини вільного пробігу фонону з використанням геометричного підходу розсіювання. Дійсно, як це було недавно експериментально продемонстровано, довжина хвилі фонону набагато менше при *RT* для наноструктурації для створення значних ефектів [53, 54].

Найкраща високомасштабна застосовність очікується від об'ємного 3D SC матеріалу, який був розроблений в низьких розмірах для теплового управління. Однією з найважливіших концепцій являється електронно-кришталеве фононне скло, що вбудоване напівпровідниковою системою, що показує великі переваги та потенціал [55]. Для кристалічних напівпровідників ідеальною моделлю була б бездефектна висококристалічна матриця, що містить нанорозмірні включення з широким розподілом діаметру. Теоретично можна передбачити, що оскільки відстань включень порівнянна із середньою довжиною вільного пробігу в матриці, то нановключення можуть забороняти ефективний перенос фонону. Дійсно, впроваджуючи більше процесів розсіювання фононів, очікується отримання матеріалу скляного фононового кристала на основі германію.

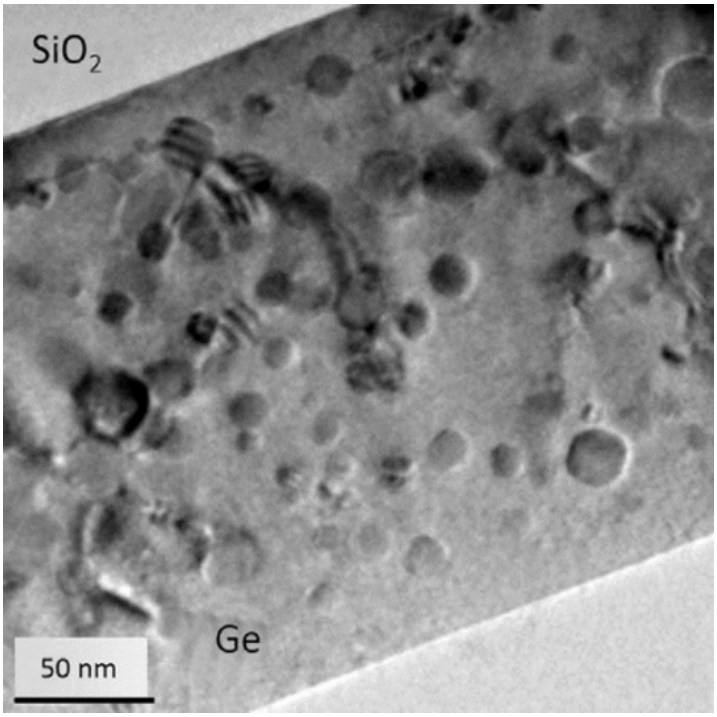


Рис. 3.6. TEM-зображення германійної матриці зі сферичними нано-включеннями GeMn при 8% Mn. Наявність численних включень значно збільшить розсіювання фононів, не порушивши електронного переносу. Відстань між нано-включеннями дійсно менше середнього MFP.

Тонкі плівки однокристалічної германієвої матриці вбудовані з випадково розподіленими нано-включеннями Ge_3Mn_5, вирощеними епітаксією на підкладці германію (рис. 3.6). Вважається, що нано-включення майже сферичні з розподілом діаметра від 5 до 50 нм, залежно від параметрів росту. Крім того, атоми марганцю в матриці Ge діють як допант *p*-типу, що забезпечує хорошу електропровідність матеріалу.

Експериментальні вимірювання теплопровідності матеріалів Ge:Mn були виконані за допомогою високочутливої диференціальної 3ω техніки [56]. Експериментальні результати 3ω вимірювань показали сильне зменшення теплопровідності порівняно з масивом Ge при кімнатній температурі для тонких плівок Ge:Mn, що містять різні концентрації Mn.

Основним експериментальним висновком являється зменшення більш ніж в десять разів відносно об'ємного значення, коли в матриці Ge є 8% Mn. Виміряна теплопровідність нижче 8 Вт / мК, що має бути порівняно з об'ємною теплопровідністю, тобто 65 Вт / мК. Зниження повністю приписується наявності рівномірно розподілених фононних розсіювачів, що значно зменшують середній вільний шлях фонону.

3.2 Програмна реалізація та її результати

Перш ніж починати, потрібно переконатися, що інструментарій CUDA може знаходити та правильно спілкуватися з апаратним забезпеченням, що підтримує CUDA. Для цього потрібно скласти і запустити деякі включені зразкові програми. Після налаштування інструментів розробки CUDA в системі, у папці зі зразками програм за замовчуванням, потрібно запустити програму *deviceQuery*, щоб перевірити правильність конфігурації апаратного та програмного забезпечення та дізнатися його характеристики, які знадобляться при самому програмуванні (рис. 3.7). Це, наприклад, загальний обсяг глобальної та константної пам’яті, колективної пам’яті на блок та максимальна кількість тредів на блок.

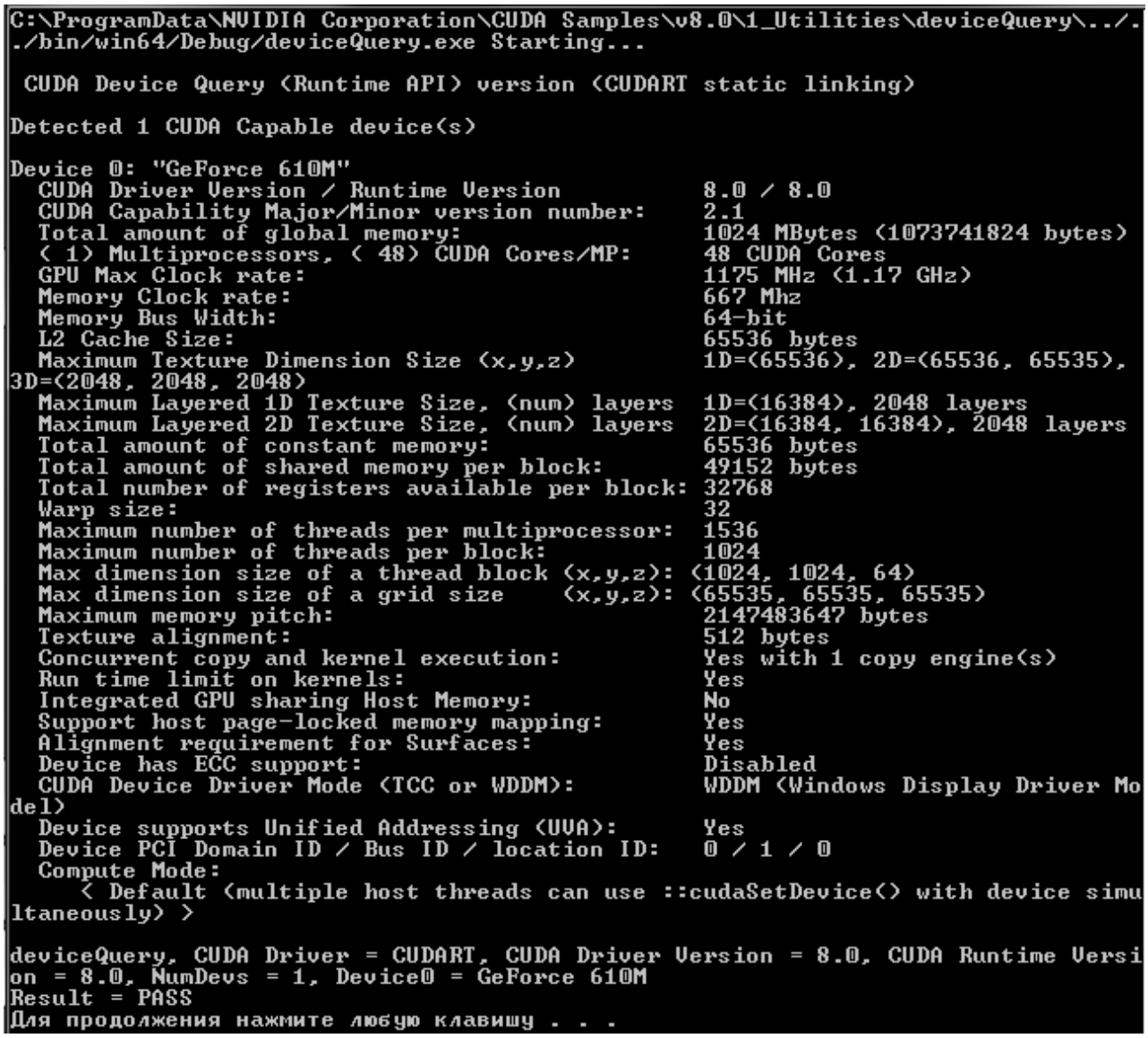


Рис. 3.7 Дійсні результати від зразка програми CUDA за замовчуванням *deviceQuery.*

Після цього починається розробка методу використання багатоядерних обчислювальних систем в моделюванні процесів переносу тіла в одновимірних наноструктурах. Для наглядного показу працездатності задається 10 стінок: 8, які фонони повинні перетнути, та 2, які їх поглинають. Паралельно буде проводитися 2 такі експерименти. А їх результати для зрівняння чистоти дослідження будуть приведені в виді діаграми. Програмний код наведений в додатку Ґ.

3.2.1 Розгляд програмної частини

В самому початку потрібно домовитися, що кількість тредів, оскільки були з'ясовані характеристики пристрою, можна задати як $\#define\ THREADS_PER_BLOCK\ 1024$ та регулювати завдяки зміненню кількості блоків, які попередньо задаються рівними 2 *(*$\#define\ BLOCKS\ 2$) для лише двох експериментів.

Керуюча частина коду, обробляюча вихідні дані, буде знаходитися на хості. А та частина з обчисленнями ймовірностей та симуляцією ходу фонону між стінок нанодроту (без додаткових модифікацій та S-подібного шикану) виконується вже на стороні GPU.

Тому в функції $main$ (), одразу після оголошень, виділення пам’яті та створення об’єкту класу $ofstream$ йде виклик допоміжної функції використання CUDA для паралельного обчислення двох експериментів. Тут, передусім, потрібно обрати, на якому графічному процесорі працювати ($cudaStatus = cudaSetDevice(0)$), а вже далі виділяти буфери GPU, як аналог виділення пам’яті на хості, з перевіркою на помилки при надзвичайних випадках. Потім викликається ядро із заданої кількістю експериментів та часток, діючих в кожному з них. В даному випадку, як вже було сказано, кількість часток дорівнює максимальній кількості тредів у блоці.

В самому ж ядрі й виконуються основні обчислення. Індекс кожного фонону задається як сума індексу треду в блоці та добутку індексу самого блоку на його розмірність. Полегшений приклад правильного індексування наведено на рис. 3.8.

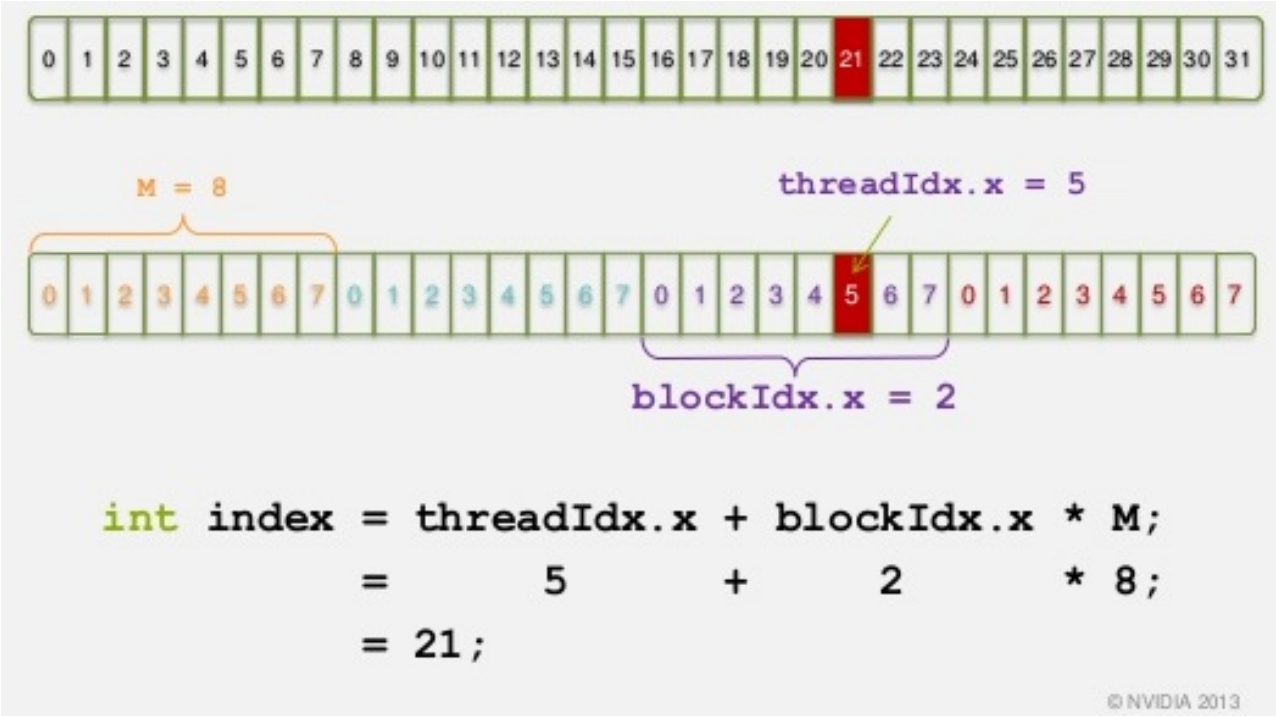


Рис. 3.8. Приклад індексації тредів в блоці та знаходження особистого індексу одного з них.

Далі йде визначання стану для послідовності *seed* з багатопроцесорним лічильником, щоб при кожному новому запуску генератору псевдовипадкових чисел послідовність результатів була різна:

$curandState\ state;$

$curand_init((unsigned\ int)clock()\ +\ index, 0, 0, \&state);$

Потім можна проводити обчислення кількості фононів, які пройдуть через всі стінки нанодроту. Це відбувається у циклі while при відмінних ймовірностях p (від 0.1 до 1 з шагом в 0.1) в різних експериментах. В Додатку Ґ відображений випадок с ймовірністю в 50%.

Після закінчення виконання цією частини йде перевірка помилок пристрія $cudaGetLastError()$, яка повертає останню помилку, та $cudaDeviceSynchronize()$, яка очікує закінчення роботи ядра та повертає будь-які помилки, що виникли під час його запуску.

Пізніше дані копіюються з пам’яті пристрою до пам’яті хоста, де вже підраховується кількість фононів, які при переносі повернулися до першої

стінки, та кількість фононів, які перебралися до останньої, для всіх паралельних експериментів.

3.2.3 Тестування результатів програмної частини

Графік зрівняння кінцевих результатів для двох експериментів можна побачити на рис. 3.9.

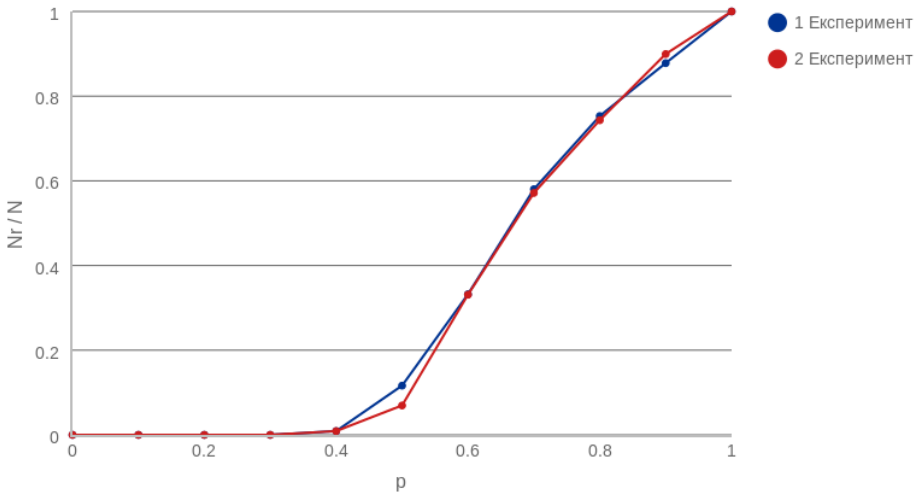


Рис. 3.9. Графік, який показує кількість фононів, які дійшли до останньої стінки (*Nr / N*) при двох експериментах при різних ймовірностях *p*.

Після проведення експериментів теплового переносу фононів через 16 розсіюючих центрів з різними ймовірностями можна побачити, що зі збільшенням кількості фононів похибка розсіювання зменшується (рис. 3.9)

На Рис. 3.10 наведено результат тестування збіжності методики, що використається. Цю перевірку необхідно робити при кожному значення ймовірності *p*, тому що кількість частинок *N*, яку треба використати для збіжності суттєво залежить від *p*.

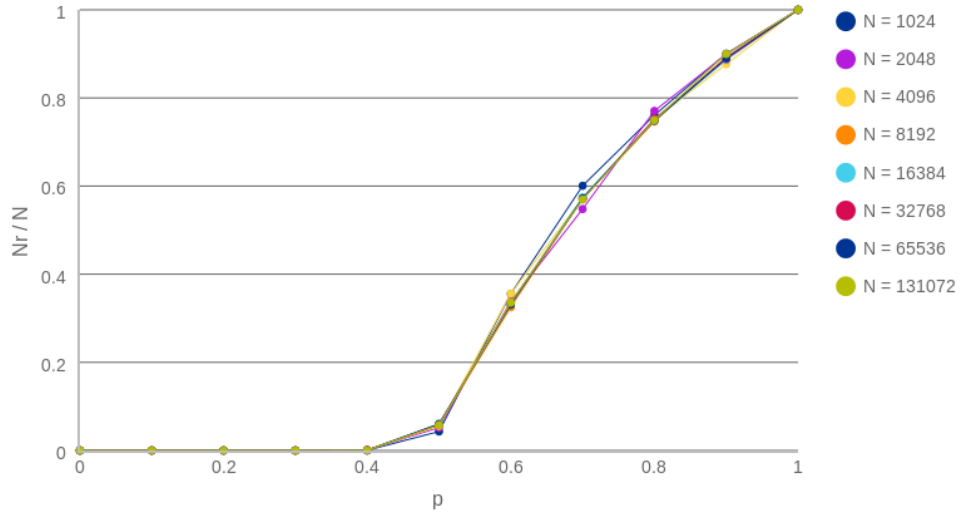


Рис. 3.10. Діаграма можливих розбіжностей в обчисленнях при різній кількісті *Nr / N* та при різних *p*.

Рис.3.10 демонструє досить швидку збіжність результатів при всіх значеннях ймовірності *p*.

Проте, якщо кількість фононів буде однакова велика, а кількість розсіюючих центрів постійно буде збільшуватися в експериментах, то розбіжність вимірювань почне згладжуватися лише при ймовірності $p > 50\%$, а до того ця характеристика буде значно зменшувати теплоперенос (рис. 3.11).

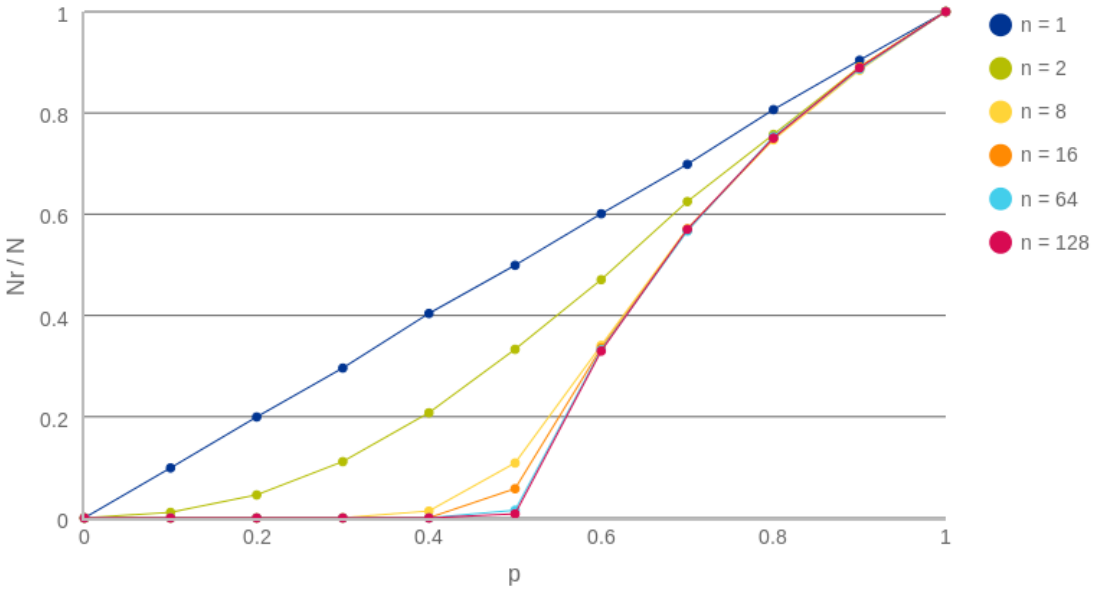


Рис. 3.11. Залежність *Nr / N* до кількості розсіюючих центрів *n*.

Рис. 3.11 також відображає поведінку потоку тепла в різних граничних випадках. Так, в разі одного центру розсіяння відношення Nr/N дорівнює ймовірності p, тому залежність на графіку лінійна. У разі, коли $p < ½$, тобто ймовірність зворотного розсіювання більше ймовірності прямого розсіювання, при збільшенні числа розсіюючих центрів n всі фонони в результаті повинні повернутися назад, що й відображено на Рис. 3.11. На рисунку також відображено той факт, що в разі, коли $p = 1$, то незалежно від числа n розсіюючих центрів все фонони повинні дійти до правого кінця, тобто $Nr/N = 1$ при будь-яких n.

Рис. 3.12 відображає залежність числа фононів Nr від кількості розсіюючих центрів n при різних значеннях ймовірності p. Рисунок також демонструє різну поведінку в залежності від співвідношення заданого значення p і значення ½, яке відповідає однаковим розсіюванням вперед і назад. При $p < ½$ відношення Nr/N різко зменшується з ростом n. При зворотної нерівності $p > ½$ це відношення виходить практично на плато.

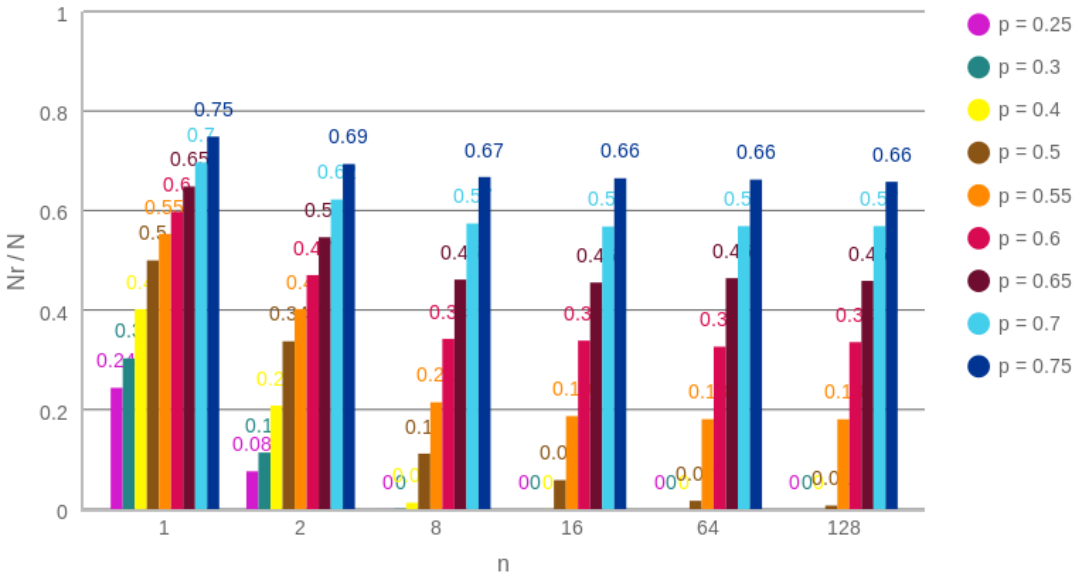


Рис. 3.12. Діаграма залежності Nr / N до кількості розсіюючих центрів n при даних p.

3.2.4. Порівняння з результатами фізичних експериментів

Результати роботи, що наведені на Рис. 3.11 і Рис. 3.12, дозволяють провести порівняння здобутих результатів з експериментальними вимірами теплопровідності в нанопровідниках з послідовно розташованими дефектами.

На Рис. 3.13 показано порівняння результатів розрахунків з результатами експериментів з дослідження сегментованих нанодротів [57] по вимірюванню числа фононів N, які пройшли через наноструктуру без сегментів до числа фононів N_r, які пройшли через наноструктуру зі структурними сегментами в кількості n. Вид сегментованих нанодротів показаний на вставці до Рис. 3.13.

Рис 3.13 демонструє якісне узгодження залежності розрахункових і експериментальних результатів від числа сегментів, а також можливість підбору величини p для досягнення кількісного узгодження.

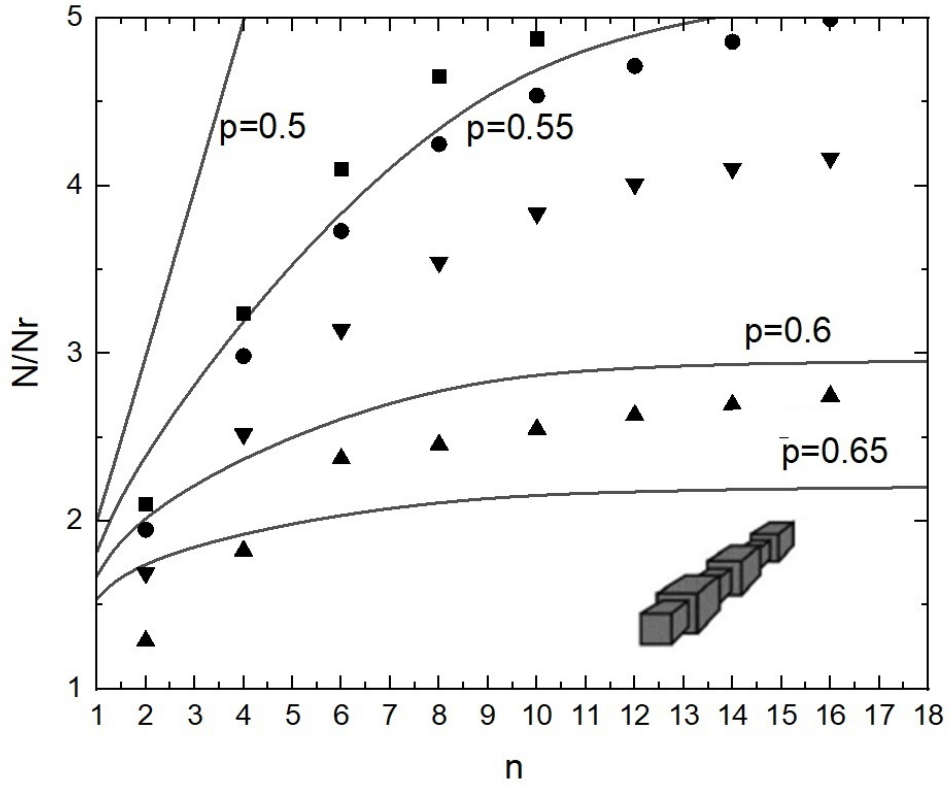


Рис. 3.13. Порівняння залежності розрахункових і експериментальних [57] результатів від кількості сегментів в сегментованих нанодротах.

На Рис. 3.14 показано порівняння результатів розрахунків з результатами експериментів з вивчення нанодротів з *S*-подібними дефектами форми [58] по вимірюванню числа фононів *N*, які пройшли через наноструктуру без дефектів до числа фононів *N*r, які пройшли через наноструктуру з двома структурними сегментами, тобто $n = 2$. Вигляд нанодротів з і без *S*-подібних структурних дефектів показаний на вставці до Рис. 3.14.

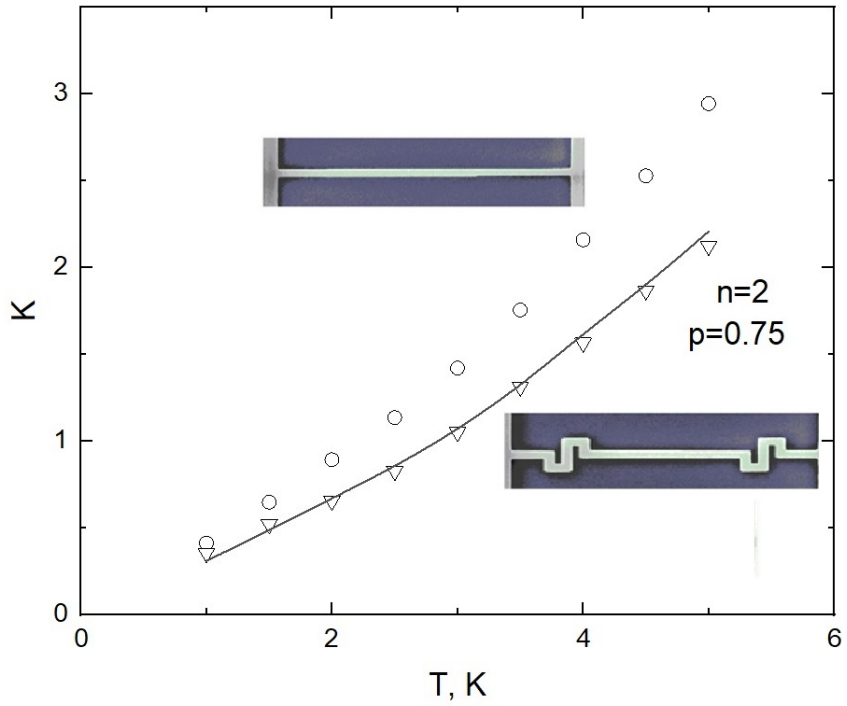


Рис. 3.14. Порівняння залежності розрахункових і експериментальних [58] результатів з теплопровідності у нанодротах без дефектів (○) та з двома S – образними дефектами (▽).

Точками на рис. 3.14 позначені експериментальні дані нормованого коефіцієнта теплопровідності для $n = 0$ (○) і для $n = 2$ (▽). Суцільна лінія відповідає розрахунку, в яких дані для прямого нанодроту множаться на фактор N_r/N, який отриманий для конкретного значення ймовірності $p = 0.75$. Рисунок демонструє задовільну згоду розрахунків з експериментальними даними.

ВИСНОВКИ

Мета та задачі, поставлені в цій роботі, були виконані. А саме:

- розглянуті можливості застосування багатоядерних обчислювальних систем в моделюванні випадкових процесів;
- проведено огляд проблем та переваг при переході від звичайних методів до методів паралельних обчислень;
- в якості найбільш якісного прикладу приведена NVIDIA CUDA та перераховані її переваг та можливостей;
- були розглянуті фізичні особливості теплопровіднистих властивостей нанодротів;
- розроблено програмну симуляцію теплового переносу вздовж нанодроту методом паралельних обчислень;
- проведено порівняння з результатами експериментальних даних.

Розроблена програма та обчислювальний алгоритм призначені для розв’язку питань теплового переносу одновимірної моделі в складних наноструктурах.

Результати роботи буде використано в реальних дослідженнях перенесення тепла в наноматеріалах з ціллю оптимізації процесів енергоспоживання в сучасних пристроях наноелектроніки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Джейсон Сандерс, Эдвард Кэндрот. Технология CUDA в примерах. Введение в программирование графических процессоров. – ДМК Пресс, 2011г.

[2] CUDA Toolkit Documentation – http://docs.nvidia.com/cuda

[3] CUDA Toolkit Documentation: cuRAND – http://docs.nvidia.com/cuda/curand

[4] NVIDIA CUDA Programming Guide / NVIDIA Corporation, 2018

[5] Скалли М.О., Зубайри М.С. Квантовая оптика. Под ред. В.В. Самарцева. – М.: ФИЗМАТЛИТ, 2003. - 512 с

[6] Некрасов К.А., Поташников С.И, Боярченков А.С., Купряжкин А.Я. Параллельные вычисления на графических процессорах. Учебное пособие. - Издательство Уральского университета Редакционно-издательский отдел ИПЦ УрФУ, 2015 г.

[7] В.П. Гергель, Р.Г. Стронгин. Основы параллельных вычислений для многопроцессорных вычислительных систем. - Н.Новгород, ННГУ, 2 изд, 2003.

[8] Accelerator: Using Data Parallelism to Program GPUs for General-Purpose Uses / Tarditi D., Puri S., Oglesby J – Microsoft Research, 2006 – 11 с.

[9] Денисенко М.В., Муняев В.О., Сатанин А.М. Применение Распределенных вычислительных систем и технологии CUDA для моделирования физических процессов. - Н.Новгород, ННГУ им. Н.И. Лобачевского, 2017 г.

[10] Wang Z L 2013 Nanowires and Nanobelts: Materials, Properties and Devices (Place: Springer Science & Business Media) Volume 1

[11] Yogeswaran U and Chen S M 2008 Sensors 8 290

[12] Volz S G and Chen G 1999 Appl. Phys. Lett. 75 2056

[13] Wang S C, Liang X G, Xu X H and Ohara T 2009 J. Appl. Phys. 105

[14] Yang N, Zhang G and Li B 2010 Nano Today 5 85

[15] Zhou Y and Hu M 2016 Nano Lett. 16 6178

[16] Hsiao T K, Chang H K, Liou S C, Chu M W, Lee S C and Chang C W 2013 Nat. Nanotechnol. 8 534

[17] Lee J, Lee W, Lim J, Yu Y, Kong Q, Urban J J and Yang P 2016 Nano Lett. 16 4133
[18] Hochbaum A I, Chen R, Delgado R D, Liang W, Garnett E C, Najarian M, Majumdar A and Yang P 2008 Nature 451 163
[19] Martin P, Aksamija Z, Pop E and Ravaioli U 2009 Phys. Rev. Lett. 102 125503
[20] Bui C T, Xie R, Zheng M, Zhang Q, Sow C H, Li B and Thong J T 2012 Small 8 738
[21] Chen J, Zhang G and Li B 2011 J. Chem. Phys. 135 204705
[22] Rojo M M, Abad B, Manzano C V, Torres P, Cartoixa X, Alvarez F X and Martin-Gonzalez M 2017 Nanoscale 9 6741
[23] Ponomareva I, Srivastava D and Menon M 2007 Nano Lett. 7 1155
[24] Zhou Y, Zhang X and Hu M 2017 Nano Lett. 17 1269
[25] Chen J, Zhang G and Li B 2010 Nano Lett. 10 3978
[26] Wingert M C, Kwon S, Hu M, Poulikakos D, Xiang J and Chen R 2015 Nano Lett. 15 2605
[27] Zhang Z, Chen Y, Xie Y and Zhang S 2016 Appl. Therm. Eng. 102 1075
[28] Cartoixa X, Colombo L and Rurali R 2015 Nano Lett. 15 8255
[29] Liu Y Y, Zhou W X, Tang L M and Chen K Q 2014 Appl. Phys. Lett. 105 203111
[30] Li D, Wu G S, Wang W, Wang Y D, Liu D, Zhang D C, Chen Y F, Peterson G P and Yang R 2012 Nano Lett. 12 3385
[31] Shim D I, Choi G, Lee N, Kim T, Kim B S and Cho H H 2017 ACS Appl. Mater. Inter. 9 17595
[32] Morshed A K M M, Yang F, Yakut Ali M, Khan J A and Li C 2012 Appl. Therm. Eng. 32 68
[33] Wang S, Cheng Y, Wang R, Sun J and Gao L 2014 ACS Appl. Mater. Inter. 6 6481
[34] Hsu P C, Liu X, Liu C, Xie X, Lee H R, Welch A J, Zhao T and Cui Y

2015 Nano Lett. 15 365

[35] D.G. Cahill, W.K. Ford, K.E. Goodson, G.D. Mahan, A. Majumdar, H.J. Maris, R. Merlin, S.R. Phillpot, J. Appl. Phys. 93 (2003) 793.

[36] A. Shakouri, Annu. Rev. Mater. Res. 41 (2011) 399.

[37] D.G. Cahill, P.V. Braun, G. Chen, D.R. Clarke, S.H. Fan, K.E. Goodson, P. Keblinski, W.P. King, G.D. Mahan, A. Majumdar, H.J. Maris, R. Merlin, S.R. Phillpot,
E. Pop, L. Shi, J. Appl. Phys. 1 (2014) 011305.

[38] S. LeBlanc, S.K. Yee, M.L. Scullin, C. Dames, K.E. Goodson, Renew. Sustain. Energy Rev. 32 (2014) 313.

[39] L.D. Hicks, M.S. Dresselhaus, Phys. Rev. B 47 (1993) 16631.

[40] M.S. Dresselhaus, G. Chen, M.Y. Tang, R.G. Yang, H. Lee, D.Z. Wang, R.F. Ren, J.P. Fleurial, P. Gogna, Adv. Mater. 19 (2007) 1.

[41] H.B.G. Casimir, Physica 5 (6) (1938).

[42] J.M. Ziman, Electrons and Phonons, Clarendon Press, Oxford, UK, 2001.

[43] R. Berman, E.L. Foster, J.M. Ziman, Proc. R. Soc. Lond. Ser. A 231 (1955) 130.

[44] J.S. Heron, T. Fournier, N. Mingo, O. Bourgeois, Nano Lett. 9 (2009) 1861.

[45] J.S. Heron, T. Fournier, N. Mingo, O. Bourgeois, Nano Lett. 10 (2010) 2288.

[46] O. Bourgeois, T. Fournier, J. Chaussy, J. Appl. Phys. 101 (2007) 016104.

[47] O. Bourgeois, E. André, C. Macovei, J. Chaussy, Rev. Sci. Instrum. 77 (2006) 126108.

[48] C. Blanc, J.-S. Heron, T. Fournier, O. Bourgeois, Appl. Phys. Lett. 105 (2014) 043106.

[49] S. Alaie, D.F. Goettler, K. Abbas, M.F. Su, C.M. Reinke, I. El-Kady, Z.C. Leseman, Rev. Sci. Instrum. 84 (2013) 105003.

[50] C. Blanc, A. Rajabpour, S. Volz, T. Fournier, O. Bourgeois, Appl. Phys. Lett. 103 (2013) 043109.

[51] N. Zen, T.A. Puurtinen, T.J. Isotalo, S. Chaudhuri, I.J. Maasilta, Nat. Commun. 5 (2014) 3435.

[52] M. Nomura, J. Nakagawa, Y. Kage, J. Maire, D. Moser, O. Paul, Appl. Phys. Lett. 106 (2015) 143102.

[53] J. Maire, R. Anufriev, H. Han, S. Volz, M. Nomura, arXiv:1508.04574.

[54] M.R. Wagner, B. Graczykowski, J.S. Reparaz, A. El Sachat, M. Sledzinska, F. Alzina, C.M. Sotomayor Torres, arXiv:1511.07398.

[55] W. Kim, J. Zide, A. Gossard, D. Klenov, S. Stemmer, A. Shakouri, A. Majumdar, Phys. Rev. Lett. 96 (2006) 045901.

[56] Y.Q. Liu, D. Tainoff, M. Boukhari, J. Richard, A. Barski, P. Bayle-Guillemaud, E. Hadji, O. Bourgeois, IOP Conf. Ser., Mater. Sci. Eng. 68 (2014) 012005.

[57] D.L.Nika, A.I.Cocemasov, A.A.Balandin, Physics Review B, 2013, 88,035428.

[58] O. Bourgeois, D. Tainoff, A.Tavakoli, Y. Liu, C. Blanc, M.Boukhari, A. Barski, E. Hadji, C.R.Physique 17 (2016) 1154-1160.

ДОДАТОК А

Тип пам’яті	Розташування	Кешу-вання	Доступ	Рівень доступу	Час життя	Можливі операції	Швидкість роботи
регістри	Мульти-процесор	ні	R/w	у треді	тред	читання/ запис	висока
локальна	DRAM	ні	R/w	у треді	тред		низька
колективна	Мульти-процесор	ні	R/w	всі треди блоку	блок	читання/ запис	висока
глобальна	DRAM	ні	R/w	Всі треди та CPU (весь грід)	виділя-ється CPU	читання/ запис	низька
константна	DRAM	так	R/o	Все треди та CPU (весь грід)	виділя-ється CPU	тільки читання	висока
текстурна	DRAM	так	R/o	Всі треди і CPU (весь грід)		тільки читання	висока

ДОДАТОК Б

```
__constant__ REAL s_rGamma2;            // Коефіцієнт дисипації
__constant__ int s_nLinesCount;         // Число обчислюваних на GPU рядків
__constant__ REAL *s_vHamiltonian;      // Гамільтоніан системи

__global__ void CalculateRightPart(const REAL *d_vDensity0, REAL
*d_vDensity, const REAL *d_vDensityArg, REAL *d_vDensityArgTemp, REAL
h1, REAL h2)
{
	int n = blockIdx.x * blockDim.x + threadIdx.x;

	if (n < s_nLinesCount)
	{
		REAL rValue = 0;

		// Масив індексів множення
		Index *pIndices = s_aMultiplicationIndices[n];
		Index *pIndicesNext = s_aMultiplicationIndices[n + 1];

		// Дисипативна частина
		while (pIndices->x < 0)
		{
			rValue += pIndices->v * d_vDensityArg[pIndices->y];

			++pIndices;
		}
		rValue = rValue * s_rGamma2;

			// Подвійна сума
			while (pIndices < pIndicesNext)
			{
				rValue += pIndices->v * s_vHamiltonian[pIndices->x] *
					d_vDensityArg[pIndices->y];
				rValue -= pIndices->v * s_vHamiltonian[pIndices->y] *
					d_vDensityArg[pIndices->x];

				++pIndices;
			}
		...
	}
}
```

ДОДАТОК В

```
// g_nProcRank – ранг процесу в MPI_COMM_WORLD
MPI_Comm commNode, commNodeTemp;

// «Колір» процесу
int nColor = !g_nProcRank;
MPI_SAFE_CALL(MPI_Comm_split(MPI_COMM_WORLD, nColor,
g_nProcRank, &commNode));

// Обробляємо «нефарбовані» процеси
while (!nColor)
{
	char sNodeNameBuf[MPI_MAX_PROCESSOR_NAME];

	MPI_SAFE_CALL(MPI_Comm_rank(commNode, &nNodeProcRank));

	// node master
	if (nNodeProcRank == 0)
	{
		strcpy(sNodeNameBuf, sNodeName);
	}

	MPI_SAFE_CALL(MPI_Bcast(sNodeNameBuf,
MPI_MAX_PROCESSOR_NAME, MPI_CHAR, 0, commNode));

// Приймаємо рішення про "фарбування"
nColor = !strcmp(sNodeNameBuf, sNodeName);
MPI_SAFE_CALL(MPI_Comm_split(commNode, nColor, g_nProcRank, &
commNodeTemp));

MPI_SAFE_CALL(MPI_Comm_free(&commNode));
MPI_SAFE_CALL(MPI_Comm_dup(commNodeTemp, &commNode));
MPI_SAFE_CALL(MPI_Comm_free(&commNodeTemp));

}
```

ДОДАТОК Г

```
// nNodeProcRank - ранг процесу в комунікаторі вузла commNode

// Пристрої вузла (сортовані за величиною тактової частоти)
int *devices = (int*)malloc(nNodeDeviceCount * sizeof(int));
int *rates = (int*)malloc(nNodeDeviceCount * sizeof(int));

if (nNodeProcRank == 0)
{
	for (int i = 0; i < nNodeDeviceCount; i++)
	{
		cudaDeviceProp prop;

		CUDA_SAFE_CALL(cudaGetDeviceProperties(&prop, i));

		int j;
		for (j = i - 1; j >= 0 && rates[j] < prop.clockRate; j--)
		{
			devices[j + 1] = devices[j];
			rates[j + 1] = rates[j];
		}

		devices[j + 1] = i;
		rates[j + 1] = prop.clockRate;
	}
}

MPI_SAFE_CALL(MPI_Bcast(devices, nNodeDeviceCount, MPI_INT, 0,
commNode));

MPI_SAFE_CALL(MPI_Bcast(rates, nNodeDeviceCount, MPI_INT, 0,
commNode));

// Вибираємо пристрої
CUDA_SAFE_CALL(cudaSetDevice(devices[nNodeProcRank]));
```

ДОДАТОК Ґ

```
#include "cuda_runtime.h"
#include "device_launch_parameters.h"
#include <curand_kernel.h>
#include <stdio.h>
#include <iostream>
#include <fstream>

#define MAX 100
#define THREADS_PER_BLOCK 1024
#define BLOCKS 2
#define GATES 10
#define PROBABILITY 50

cudaError_t RunWithCuda(int* end, unsigned int size);

__global__ void kernel_launch(int* end, int* rand);

using namespace std;

int main() {

	int *end;
	int size = THREADS_PER_BLOCK * BLOCKS * sizeof(int);

	ofstream result;

	end = (int *)malloc(size);

	cudaError_t cudaStatus = RunWithCuda(end, size);
	if (cudaStatus != cudaSuccess) {
		fprintf(stderr, "RunWithCuda failed!");
		return 1;
	}

	int first = 0;
	int last = 0;
	float firstRatio = 0.0;
	float lastRatio = 0.0;

	result.open("Result.txt", ios::app);
```

```
if (result.is_open())
{
	for (int bl = 0; bl < BLOCKS; bl++) {
		for (int thr = 0; thr < THREADS_PER_BLOCK; thr++) {
			if (end[thr + bl * THREADS_PER_BLOCK] == 0) {
				first += 1;
			}
			else if (end[thr + bl * THREADS_PER_BLOCK] == GATES-1)
{
				last += 1;
			}
		}
	}
	firstRatio = (float)first / (THREADS_PER_BLOCK * BLOCKS);
	lastRatio = (float)last / (THREADS_PER_BLOCK * BLOCKS);

	result << "Probability: " << PROBABILITY << "%" << endl;
	result << "n = " << GATES - 2 << endl;
	result << "N = " << THREADS_PER_BLOCK * BLOCKS << endl << endl;
	result << "N on the first wall: " << first<< " or Nl/N = " << firstRatio << endl;
	result << "N on the last wall: " << last << " or Nr/N = " << lastRatio << endl;
	result << "-----------------------------------" << endl;

	}

	result.close();

	free(end);
	// cudaDeviceReset must be called before exiting in order for profiling and
tracing tools such as Nsight and Visual Profiler to show complete traces.
	cudaStatus = cudaDeviceReset();
	if (cudaStatus != cudaSuccess) {
		fprintf(stderr, "cudaDeviceReset failed!");
		return 1;
	}

	return 0;
}

cudaError_t RunWithCuda(int* end, unsigned int size)
{
	int *d_end, *d_rand;
	cudaError_t cudaStatus;
```

```
	cudaStatus = cudaSetDevice(0);
	if (cudaStatus != cudaSuccess)
	{
		fprintf(stderr, "cudaSetDevice failed!  Do you have a CUDA-capable
GPU installed?");
		goto Error;
	}

	cudaStatus = cudaMalloc((void**)&d_end, size);
	if (cudaStatus != cudaSuccess)
	{
		fprintf(stderr, "cudaMalloc failed!");
		goto Error;
	}

	cudaStatus = cudaMalloc((void**)&d_rand, size);
	if (cudaStatus != cudaSuccess)
	{
		fprintf(stderr, "cudaMalloc failed!");
		goto Error;
	}

kernel_launch << <BLOCKS, THREADS_PER_BLOCK >> > (d_end, d_rand);

	cudaStatus = cudaGetLastError();
	if (cudaStatus != cudaSuccess)
	{
		fprintf(stderr, "addKernel launch failed: %s\n",
cudaGetErrorString(cudaStatus));
		goto Error;
	}

	// cudaDeviceSynchronize waits for the kernel to finish, and returns any
errors encountered during the launch.
	cudaStatus = cudaDeviceSynchronize();
	if (cudaStatus != cudaSuccess)
	{
		fprintf(stderr, "cudaDeviceSynchronize returned error code %d after
launching addKernel!\n", cudaStatus);
		goto Error;
	}

	cudaStatus = cudaMemcpy(end, d_end, size, cudaMemcpyDeviceToHost);
```

```
if (cudaStatus != cudaSuccess)
	{
		fprintf(stderr, "cudaMemcpy failed!");
		goto Error;
	}

Error:
	cudaFree(d_end);
	cudaFree(d_rand);

	return cudaStatus;
}

__global__ void kernel_launch(int* end, int* rand)
{

	int index = threadIdx.x + blockIdx.x * blockDim.x;
	curandState state;
	curand_init((unsigned int)clock() + index, 0, 0, &state);

	end[index] = 1;

	while (end[index] < GATES - 1)
	{

		rand[index] = curand(&state) % MAX;

		if (rand[index] >= PROBABILITY)
		{
			end[index] -= 1;
		}
		else
		{
			end[index] += 1;
		}
		if (end[index] == 0)
		{
			break;
		}
	}
}
```