


Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Московский авиационный институт

(национальный исследовательский университет)»


На правах рукописи

Семенов Сергей Александрович

УДК 519.688

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ АЛГОРИТМОВ
МОЛЕКУЛЯРНО-ДИНАМИЧЕСКОГО МОДЕЛИРОВАНИЯ НАНОСИСТЕМ
НА ГРАФИЧЕСКИХ ПРОЦЕССОРАХ

Специальность 05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Специальность 05.13.18 – Математическое моделирование, численные методы и комплексы программ

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
доктор физико-математических наук,
профессор
Ревизников Дмитрий Леонидович

Научный консультант:
кандидат физико-математических наук,
Сластушенский Юрий Викторович

Москва 2017

Содержание

Введение.....	3
Глава 1. Постановка задачи молекулярно-динамического моделирования на графических процессорах.....	14
1.1 Архитектура графических процессоров NVIDIA.....	14
1.2 Метод молекулярной динамики.....	18
1.3 Методы аппроксимации уравнений движения.....	20
1.4 Потенциалы межчастичного взаимодействия.....	22
1.5 Методы термостатирования.....	40
1.6 Геометрическое построение углеродных наноструктур.....	46
Выводы к главе 1.....	52
Глава 2. Программирование алгоритмов молекулярно-динамического моделирования наносистем на графических процессорах.....	53
2.1 Принципы разработки программного обеспечения.....	53
2.1.1 Отображение расчётной области на структуру памяти GPU: гибридная модель.....	53
2.1.2 Реализация потенциала Бреннера второй модификации на GPU.....	56
2.1.3 Интегрирование и представление разностной схемы.....	57
2.1.4 Структура данных для параллельного счёта.....	58
2.1.5 Редукция данных для определения макропараметров.....	59
2.1.6 Отображение параллельных процессов на архитектуру GPU.....	60
2.2 Сложность вычислений и затраты памяти.....	68
2.3 Методы анализа ускорения вычислений.....	70
2.4 Наследование в технологии CUDA.....	72
2.5 Технология разработки программного обеспечения.....	76
2.6 Визуализация вычислений в реальном времени.....	78
Выводы к главе 2.....	86
Глава 3. Компьютерные эксперименты. Анализ эффективности распараллеливания вычислений.....	89
3.1 Верификация программного обеспечения.....	89
3.2 Скорость вычисления многочастичных потенциалов на графических процессорах.....	94
3.3 Измерение производительности вычислений.....	101
3.3.1 Схема компьютерного эксперимента.....	101
3.3.2 Сравнение эффективности решения задач на различных GPU.....	103
3.4 Решение задачи термостатирования.....	106
Выводы к главе 3.....	112
Глава 4. Методы моделирования теплопроводности наносистем.....	114
4.1 Особенности теплопереноса в наноструктурах.....	114
4.2 Моделирование теплопроводности наноструктур. Идентификация и анализ режимов теплопроводности.....	118
4.2.1 Моделирование распространения тепла в листе графена.....	119
4.2.2 Моделирование распространения тепла в нанотрубке.....	134
4.3 Описание аномальной теплопроводности с использованием дробно-дифференциальных уравнений.....	143
Выводы к главе 4.....	146
Заключение.....	147
Глоссарий.....	149
Список литературы.....	150

Введение

Растущие требования к производительности современных компьютерных систем приводят к уменьшению размеров транзисторов, а также стимулируют проводить многоуровневую интеграцию компонент и вводить быстрое исполнение инструкций, кэшируемую память, многопоточность. Упростив инструкции одного ядра, уменьшив потребление энергии и увеличив общее количество ядер, графические процессоры (GPU – Graphics Processing Unit) вышли сегодня вперёд по производительности на масштабных задачах по сравнению с современными процессорами (CPU – Central Processing Unit). Кроме того, использование GPU позволяет существенно расширить возможности вычислительных систем.

Относительно простая архитектура GPU ставит разработчиков перед необходимостью использовать определённые вычислительные модели, которые не приводят к рекурсиям, не имеют множественных зависимостей данных, численно устойчивы и имеют достаточную точность в 32-битном представлении.

Архитектура GPU позволяет масштабировать задачи как внутри одной видекарты, так и внутри набора видеокарт, или набора отдельных вычислительных машин. Технология NVIDIA SLI позволяет объединить до 3 плат в одно вычислительное устройство. Серверная система Tesla позволяет установить до 4 графических плат на одной вычислительной машине. В подобных машинах содержится более 8 тыс. вычислительных ядер. В работе [1] используются технологии MPI, OpenMP для запуска решения задач на нескольких вычислительных машинах, содержащих CPU и GPU. Показаны преимущества и недостатки различных способов взаимодействия GPU на примере решения уравнений Навье-Стокса.

Программирование GPU разбилось на два подхода: полный контроль аппаратных ресурсов и универсальный язык разных платформ. Первый подход предоставляет программисту инструменты для достижения высокой производительности за счёт использования иерархии кэш памяти и поточного выполнения инструкций. При этом требуются знания по архитектуре процессорных чипов, об ограничениях шины передачи данных, об очередности исполнения вычислительных потоков, о доступности и загруженности аппаратных ресурсов. В

работе [2] приводится сравнение технологий OpenACC и CUDA для решения задач UPACS на GPU и показано, что использование CUDA с полным контролем ресурсов превосходит на 50% по производительности OpenACC. Подобные результаты отмечены в работах [3, 4], а также при сравнении технологий OpenCL. В работе [5] получено ускорение до 15 раз при использовании технологии OpenCL вместо OpenMP. В работе [6] проводилось сравнение в различных тестах SHOC, FFT, GEMM, Rodinia реализованных в различных библиотеках CUBLASS, MAGMA различных видеокарт Tesla, GeForce, отмечено преимущество использования технологии CUDA перед OpenCL и видеокарт GeForce 480 GTX перед Tesla C2050.

Эффективное практическое применение современных GPU началось с 2008 года. На видеокартах производства 2006 года проводилось моделирование астрофизических задач с помощью программных комплексов GraCCA [7], GRAPE-6A. На тот момент видеокарты могли обеспечивать ускорение в 2 раза по сравнению с CPU, но имели хорошие перспективы развития, низкую стоимость и меньшее энергопотребление в пересчёте на вычислительное ядро.

Применение специфичных алгоритмов позволяет увеличивать производительность в рамках GPU по сравнению с прямой реализацией математических формул. В работе [8] рассматривается простой алгоритм SFCPACK, который обеспечивает прирост эффективности сортировки на 8% по сравнению с более сложным GPART. Также показано многократное ускорение при использовании сортировки по сравнению с неотсортированными данными.

Различные прикладные математические алгоритмы оказываются нетривиальными для реализации на GPU. В работе [9] рассматривается эффективная реализация генератора псевдослучайных чисел для задач молекулярного моделирования. В связи с различиями CPU и GPU адаптация разработанных ранее методов оказывается затруднительной, так как требуется генерации большого количество случайных чисел за микросекунды. Необходимо подобрать алгоритм, который должен иметь большой период, быть быстрым и генерировать числа высокого статистического качества. С учётом всех требований LCG, Hybrid Taus и алгоритм Фибоначи имеют хорошую производительность и расходуют небольшое

количество памяти. Но, принимая во внимание статистическое качество, лучшими оказываются Hybrid Taus и алгоритм Фибоначи для молекулярного моделирования на GPU.

Графические процессоры активно применяются в вычислительной гидродинамике. В работе [10] вычисления проводятся на GPU производства NVIDIA с количеством ячеек сетки от 10^5 до 10^7 . Полученное увеличение производительности при решении задачи вязкого течения жидкости на плоской пластине достигает соответственно значений от 46 до 56 раз на Tesla S2050 по сравнению с CPU Xeon X5670 2.93 GHz.

Графические процессоры дают преимущество в задачах молекулярной динамики. В работе [11] рассматривается эффективная реализация силы ReaxFF в реакционной молекулярной динамике. Достигнуто ускорение расчётов до 16 раз по сравнению с CPU. Проведено обширное исследование по подбору структур хранения данных, алгоритмов, оптимизаций, чтобы получить приемлемые результаты вычислений на GPU известными ранее методами.

Точное моделирование наноструктур методом молекулярной динамики достигается выбором сложных потенциалов межатомного взаимодействия, что усложняет реализацию алгоритмов. Их производительность существенно зависит от способа доступа к памяти, вида математических выражений и аппаратных возможностей. В работе [12] показано какие инструменты необходимо задействовать, чтобы получить ускорения для решения задач с потенциалом Терсоффа и погруженного атома. Рассматривается переупаковка трехмерных данных в четырехмерные типы для выравнивания памяти. Достигнуто ускорение расчётов до 40 раз по сравнению с CPU.

Для проведения молекулярно-динамического моделирования требуется решить большой спектр задач, но группы исследователей выбирают для улучшения только один класс алгоритмов, чего недостаточно для получения общей картины. Поэтому в работе [13] портировали разработанный код ReaxFF в более мощный программный комплекс LAMMPS. При встраивании разнородных модулей возникает множество задач совместимости различных языков, так, например, вычислительные алгоритмы в

большинстве случаев реализуются на языке FORTRAN, а программные комплексы – в виде целостных приложений на языках C, C++.

Разработка одинаковых алгоритмов с использованием GPU ведется разными командами разработчиков, программы получают похожие названия: PuReMD-GPU [11], User-Reax [14] и GMD-Reax [15], при этом они имеют разную производительность.

В вычислительных задачах в основном используются GPU производства NVIDIA в виду того, что компания выпустила удобное средство разработки CUDA (Compute Unified Device Architecture), которое представляет новую программную высокоуровневую модель. Чтобы алгоритм эффективно выполнялся на видеокарте, необходимо его представить в виде параллельных операций над тысячами блоками данных. В CUDA потоки объединяются в блоки одинакового размера. Программист выбирает необходимое или оптимальное количество блоков, которое запускается на устройстве. Эффективное выполнение блоков требует понимания вычислительной конфигурации аппаратного устройства. Например, видеокарта 480GTX содержит 480 потоковых процессоров. Каждый мультипроцессор выполняет 32 потока одновременно, что составляет мультипоток (warp). Например, при организации 3 блоков по 128 потоков необходимо задействовать 12 мультипотоков. Минимальное количество потоков выполнения равно размеру мультипотока. Все 32 потока в мультипотоке выполняют одну инструкцию.

Важным является вопрос специфичного доступа к памяти. Большие объёмы данных хранятся в глобальной памяти, данные необходимые вычислительным потокам в блоке для скоростного обмена загружаются в разделяемую память. Повышает скорость запросов к памяти специфичная текстурная память, которая имеет механизмы кеширования, предотвращает ошибки выхода за пределы памяти, а также обеспечивает линейную интерполяцию.

Память GPU имеет большую пропускную способность и большую задержку при обращении по заданному адресу. Во время обращения к памяти может формироваться множество запросов. Если потоки из вычислительного блока обращаются к близким адресам памяти, то несколько запросов могут быть сгруппированы в один и

исполнены параллельно. Это свойство используется для повышения скорости доступа.

Интенсивный рост производительности вычислительной техники и, прежде всего, многопроцессорных вычислительных систем делает возможным в настоящее время эффективно моделировать структуры, состоящие из сотен миллионов структурных элементов.

Особенности молекулярно-динамического моделирования наносистем на графических процессорах обсуждаются в статье [105]. В таких задачах расчёт действующих на атомы сил требует значительных вычислительных затрат. В больших системах обычно используется модель короткодействия, когда область влияния одного атома ограничивается радиусом усечения. На графических процессорах эффективно разбивать область моделирования на ячейки. Такая модель уменьшает количество вычислений для N_a атомов с $O(N_a^2)$ до $O(N_a)$ и допускает независимые параллельные расчёты. Подобные вычисления каждого шага моделирования выполнялись в работе [16] полностью на GPU. Оценивалась производительность двух подходов: построения списка соседей и ячеистой модели движения [9], достигалась производительность в 160 шагов вычислений за секунду.

В настоящее время известен целый ряд эффективных реализаций для сравнительно простых потенциалов межчастичного взаимодействия. В этой связи отметим такие программные комплексы, как HOOMD [9], AMBER [17], LAMMPS [18-20], GROMACS [21], NAMD [22-23], CHARMM [24], ACEMD [25], Desmond [26], Espresso [27], Folding@Home [24]. Все они используют GPU для ускорения молекулярно-динамических расчётов. В ранних работах по распараллеливанию молекулярно-динамических расчетов на GPU [28] не учитывается многочастичный характер потенциала и наличие ограничивающих функций. Повышенный интерес в настоящее время вызывает реализация многочастичных потенциалов, таких, например, как потенциал Бреннера, на графических процессорах [29]. Это требует разработки специальных параллельных алгоритмов и программной реализации на GPU, которые адаптированы к специфике таких потенциалов. Разработка и применение методов, позволяющих снизить вычислительную сложность до $O(N)$

позволяют увеличить количество моделируемых элементов на порядок. Ещё более важной задачей является перенос алгоритмов, имеющих сложность $O(N)$, на параллельные вычислители так, чтобы они выполнялись параллельно и давали вычислительную сложность $O(\frac{N}{P})$. Деление на P обеспечивается выделением памяти, поэтому актуальной является не только алгоритмическая достижимость параллельных вычислений, но и практическая реализация на заданном аппаратном обеспечении.

Исходя из вышесказанного можно сформулировать цели, задачи и основные положения диссертационной работы.

Актуальность темы.

В последнее время все больший интерес вызывает проведение массивных вычислений с использованием графических процессоров (GPU). Известны варианты применения графических ускорителей к задачам линейной алгебры, математической физики, вычислительной аэрогидродинамики и т.д. В отличие от универсальных процессоров (CPU) видеочипы предназначены, в первую очередь, для параллельных вычислений с большим количеством арифметических операций. Упрощение инструкций одного ядра, уменьшение потребления энергии и увеличение общего количества потоковых процессоров обуславливают преимущество графических ускорителей над традиционными высокопроизводительными вычислительными комплексами. Немаловажным является и существенное снижение требований к инфраструктуре вычислительных систем, для проведения массивных параллельных расчетов достаточно оснащение персонального компьютера соответствующей видеокартой.

Одной из перспективных областей применения графических процессоров является молекулярно-динамическое моделирование. Внутренне присущий этому направлению параллелизм вычислительного процесса находит адекватное отражение в архитектуре видеокарт. В настоящее время существует целый ряд эффективных реализаций ускорения вычислений для сравнительно простых потенциалов межчастичного взаимодействия. В этой связи можно отметить известные программные комплексы молекулярно-динамического моделирования HOOMD,

AMBER, LAMMPS, GROMACS, NAMD, CHARMM, ACEMD, Desmond, Espresso, Folding@Home. Все они оснащены средствами, позволяющими использовать GPU для ускорения расчётов. Однако предсказательной возможности простых потенциалов зачастую недостаточно для адекватного воспроизведения широкого спектра свойств наносистем. Необходимо использовать многочастичные потенциалы, отражающие реальные свойства межчастичных связей. Это обуславливает актуальность разработки специальных подходов к программной реализации параллельных алгоритмов на GPU, адаптированных к специфике таких потенциалов. При этом большое значение имеет технологичность разработки и модификации программ данного класса, возможность интеграции с имеющимися программными комплексами, визуализация результатов вычислительного эксперимента в реальном времени. Данным вопросам посвящена диссертационная работа.

Целью работы является разработка технологии программирования алгоритмов молекулярно-динамического моделирования наноструктур со сложными потенциалами межчастичного взаимодействия на графических процессорах. Для этого необходимо решение следующей группы **задач**.

- ✓ Анализ существующих алгоритмов молекулярно-динамического моделирования с позиций поиска оптимального отображения вычислительных процессов на архитектуру графических процессоров.
- ✓ Выработка подходов к размещению данных в памяти видеокарты с целью минимизации количества перекрестных запросов при параллельных вычислениях.
- ✓ Разработка методов наследования классов в технологии CUDA, обеспечивающих возможность оснащения программы новыми потенциалами межчастичного взаимодействия без модификации основного кода.
- ✓ Разработка методов интеграции программного обеспечения с имеющимися программными комплексами молекулярно-динамического моделирования.
- ✓ Разработка методов и средств визуализации результатов вычислительного процесса в реальном времени.

- ✓ Применение разработанной технологии программирования для создания авторского комплекса программ молекулярно-динамического моделирования наносистем.
- ✓ Проведение вычислительных экспериментов, направленных на анализ эффективности вычислительного процесса.
- ✓ Разработка подходов к моделированию теплопроводности наноструктур с использованием созданного программного обеспечения.
- ✓ Проведение вычислительных экспериментов по моделированию процессов теплопереноса в наноструктурах. Выявление режимов аномальной теплопроводности.
- ✓ Разработка методов согласования молекулярно-динамического моделирования и макромасштабного описания процессов аномальной теплопроводности с использованием дробно-дифференциальных уравнений.

Научная новизна.

Разработаны методы построения программного обеспечения для молекулярно-динамического моделирования наносистем на графических процессорах, включающие методы отображения вычислительных процессов на архитектуру видеокарт, методы размещения данных в памяти видеокарты, методы наследования классов в технологии CUDA, обеспечивающие возможность оснащения программ новыми потенциалами межчастичного взаимодействия без модификации основного кода.

Предложены подходы к повышению эффективности параллельных вычислений на графических процессорах, включающие использование гибридной модели поиска ближних частиц, составление и обновление списка соседних атомов с целью минимизации коллизий памяти, оптимальное распределение операций по вычислительным потокам, выделение дополнительной памяти для создания копий координат взаимодействующих атомов.

С использованием созданного по разработанной технологии программного обеспечения исследованы вопросы моделирования теплопроводности углеродных наносистем. Для описания аномальных режимов теплопроводности разработан подход, основанный на сочетании методов молекулярной динамики и дробно-

дифференциального исчисления. Предложен алгоритм определения параметров макроскопической модели по данным молекулярно-динамического моделирования. Таким образом, установлена связь между различными масштабами в описании аномальной теплопроводности.

Достоверность и обоснованность результатов, полученных в ходе диссертационного исследования, обеспечивается согласованностью результатов проведённых вычислительных экспериментов с лабораторными измерениями, а также результатами моделирования с использованием других программных комплексов, полученными независимыми друг от друга способами.

Практическая ценность Представленная в диссертации технология программирования даёт возможность повысить эффективность разработки программного обеспечения для моделирования динамических систем на графических процессорах, использовать и расширять уже существующие программные модули, осуществлять интеграцию разрабатываемых программных средств с имеющимися программными комплексами.

Разработанные в диссертации методы и средства математического моделирования имеют высокую значимость с точки зрения перспектив их применения для исследования диффузионных и тепловых процессов в наносистемах.

Результаты диссертационного исследования могут быть использованы при составлении образовательных курсов по методам программирования для графических процессоров и математическому моделированию наносистем.

Апробация работы.

Материалы диссертации докладывались и обсуждались на следующих российских и международных форумах:

VIII Международная конференция по неравновесным процессам в соплах и струях (NPNJ'2012), 25-31 мая 2012 г., Алушта.

11-я Международная конференция «Авиация и космонавтика — 2012», 13-15 ноября 2012 г., Москва.

55-я научная конференция МФТИ, 19-25 ноября 2012 г., Долгопрудный.

XVII Международная конференция по вычислительной механике и современным прикладным программным системам (ВСПССС`2013), 22-31 мая 2013 г., Алушта.

15-я Международная конференция «Авиация и космонавтика – 2016», 14-18 ноября 2016 г., Москва.

16-я Международная конференция «Авиация и космонавтика – 2017», 20-24 ноября 2017 г., Москва.

Публикации.

По теме диссертации опубликовано 11 работ [105-115], из них 4 статьи в научных журналах из перечня ВАК РФ для представления основных научных результатов диссертаций на соискание учёных степеней доктора и кандидата наук. Список публикаций приведён в конце работы.

Структура и содержание работы.

Диссертация состоит из введения, четырёх глав, заключения, списка литературы, глоссария. В работе содержится 20 таблиц, 95 рисунков и 115 библиографических ссылок. Общий объём работы составляет 160 страниц.

Во введении диссертационной работы даётся постановка задачи, рассматриваются области применения нанотрубок, определяется схема компьютерного моделирования. Обосновывается преимущество видеопроцессоров при вычислениях динамики молекул. Даётся обзор литературы по применению графических процессоров в молекулярно-динамическом моделировании наноструктур и обоснование актуальности темы диссертационной работы, сформулированы цели и задачи, отмечена научная новизна, представлены данные по апробации работы и перечислены авторские публикации по теме.

В первой главе работы дано описание задач молекулярно-динамического моделирования и способов их реализации на графических процессорах. GPU состоят из множества вычислительных ядер, которые выполняют простые функции, и работают параллельно при выполнении одинакового кода.

Во второй главе представлена технология программирования, которая позволяет эффективно моделировать углеродные наноструктуры на графических

процессорах. Рассматривается ячеистая модель пространства, которая уменьшает сложность вычисления до $O(N)$. Ячеистая модель расширена до гибридной модели, сочетающей в себе ячеистую модель и систему списков соседних атомов. При параллельных вычислениях такое расширение позволяет избежать перекрёстных запросов памяти. Приводится расчёт вычислительной сложности; показано, что наиболее длительные операции при достаточном объёме памяти имеют сложность $O(N)$. Приводится оценка масштабного ускорения; так как изменить количество графических процессоров возможно при замене видеокарты, то основным методом оценки эффективности является измерение времени выполнения при увеличении объёма данных. Так как данные находятся в памяти графической карты, то результаты моделирования возможно отображать на мониторе в реальном времени.

С использованием разработанной технологии алгоритмы реализованы в программном комплексе, который предоставляет расширение архитектуры CUDA. Вычислительные потоки объединяются в блоки и запускаются в соответствии с требованиями алгоритмов. Данные для вычислительных потоков распределяются, сортируются, переразмещаются в памяти таким образом, что не происходит перекрёстных запросов.

В третьей главе даются результаты различных компьютерных экспериментов по моделированию наноструктур на графических процессорах. Проведён анализ вычислительной сложности, затрат памяти и производительности CPU и GPU.

В четвёртой главе исследуется задача моделирования теплопроводности в наносистемах с использованием разработанного программного комплекса. По центру образцов подводится тепловой импульс, после чего моделируется его распространение вдоль наноструктуры. Затем определяется характер распространения тепла. Для описания обнаруженных аномальных режимов предложено использовать мелко-дифференциальную математическую модель уравнения теплопроводности.

В заключении перечисляются основные результаты, полученные в ходе выполнения диссертационной работы.

Глава 1. Постановка задачи молекулярно-динамического моделирования на графических процессорах

1.1 Архитектура графических процессоров NVIDIA

В отличие от современных универсальных центральных процессоров (CPU), видеочипы предназначены для параллельных вычислений с большим количеством арифметических операций. Большое число транзисторов графических процессоров (GPU) работает по прямому назначению – обработке массивов данных, а не управляет исполнением (flow control) немногочисленных последовательных вычислительных потоков. В современных CPU присутствует небольшое количество ядер – 4 арифметическо-логических устройства (ALU), блок управления (Control), устройство кэширования (Cache), оперативная память (DRAM). В современных GPU устанавливают большое количество ядер, несколько блоков управления и кэширования, большой объём общей памяти. С начала 2000-х годов GPU вычисляли цвета пикселей на экране с помощью программируемых арифметических устройств и пиксельных шейдеров (pixel shader). Пиксельный шейдер получает на входе координаты (x, y) точки на экране и дополнительную информацию, а на выходе должен выдать конечный цвет этой точки. В качестве дополнительной информации могут выступать входные цвета, текстурные координаты или иные атрибуты, передаваемые шейдеру на этапе его выполнения. Так как арифметические действия над входными цветами и текстурами контролируются программистом, то в качестве данных могут выступать не только цвета. В ноябре 2006 года NVIDIA выпустила серию видеокарт GeForce на основе архитектуры CUDA. В отличие от предыдущих поколений GPU, в которых вычислительные потоки подразделялись на вершинные и пиксельные шейдеры, в архитектуру CUDA включен унифицированный шейдерный конвейер, позволяющий программе общего назначения задействовать любое арифметическо-логическое устройство (ALU). Поскольку NVIDIA рассчитывала, что новое семейство графических процессоров будет использоваться для вычислений общего назначения, то ALU были сконструированы с учетом требований IEEE (англ. Institute of Electrical and Electronics Engineers) к арифметическим операциям над

числами с плавающей точкой одинарной точности. Кроме того, был разработан набор команд, ориентированный на вычисления общего назначения, а не только на графику. Устройствам GPU разрешен произвольный доступ к памяти для чтения и записи, а также доступ к программно-управляемой кэш-памяти. Эти средства были добавлены в архитектуру CUDA с целью создать GPU, который отлично справлялся бы с вычислениями общего назначения.

В качестве первой доступной видеокарты, которая использовалась для моделирования, приведём одну из самых первых с поддержкой CUDA GeForce 8600 GTS. Она имеет следующие характеристики: память GDDR3 256 Мб, 4 мультипроцессора по 8 потоков, частоту памяти 1050 МГц, пропускную способность шины 128 бит. Это значит, что видеокарта способна выполнять 32 шага цикла вычислений за один такт. Если функция в цикле не приводит к рассинхронизации вычислительных потоков, то это дает преимущество в 32 раза по сравнению с 1 ядром процессора.

Архитектура вычислителей CUDA состоит из параллельных экземпляров ядра. Эти параллельные экземпляры называются вычислительными блоками. Исполняющая среда CUDA позволяет расщепить блоки на вычислительные потоки (thread). Запуск вычислительного ядра выглядит следующим образом:

```
//numBlocks, numThreads  
function<<< numBlocks, numThreads >>>(dev_a, dev_b, dev_c);
```

где *numBlocks*, *numTreads* – количество вычислительных блоков и потоков, *dev_a*, *dev_b*, *dev_c* – некоторые аргументы, передаваемые ядру. Первый параметр обозначается количество вычислительных блоков, второй параметр в угловых скобках говорит, сколько вычислительных потоков создать для каждого блока. В этом примере конфигурируется *numBlocks* блоков, состоящих из *numTreads* вычислительных потоков, всего получается *numBlocks*numTread* параллельных вычислителей.

Для программирования на GPU производства NVIDIA используется технология CUDA [30]. Согласно архитектуре CUDA вычислительные потоки организованы в блоки (рис. 1.1).

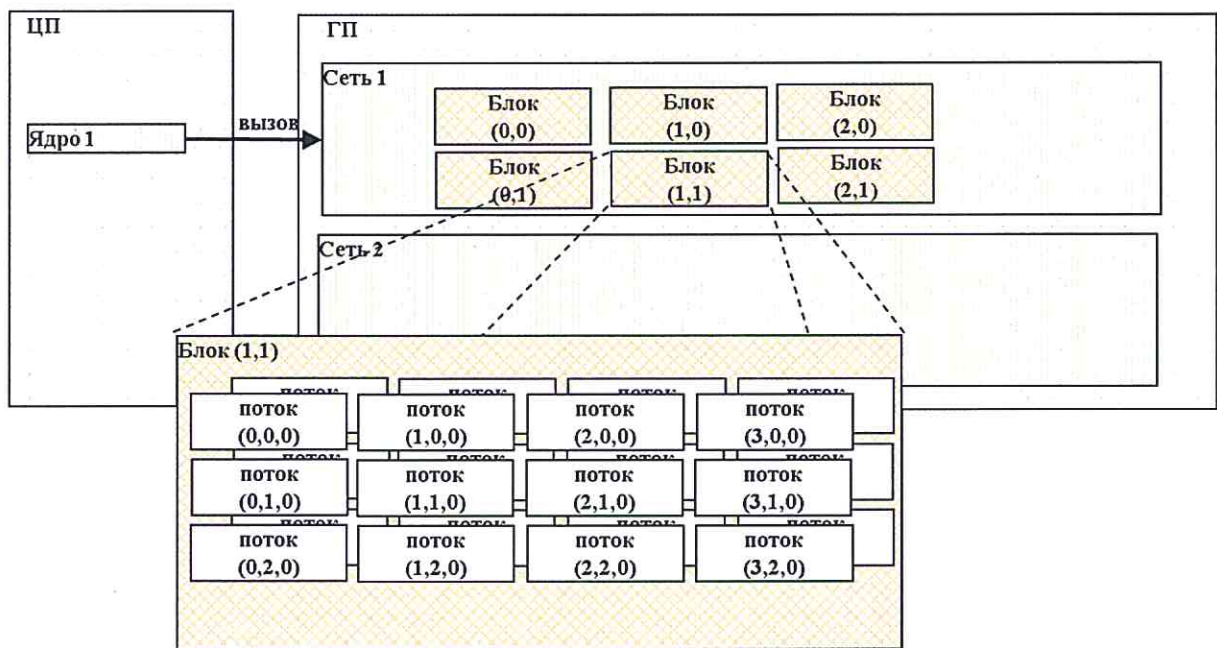


Рисунок 1.1. Логическое представление вычислительных блоков

Каждый блок имеет не менее 32 потоков. Все потоки в блоке выполняют одинаковый набор инструкций, однако ветвления внутри программы могут нарушать синхронность исполнения.

Особенностью программирования на GPU является доступ к памяти. Устройства имеют большую пропускную способность памяти, но и большую задержку при обращении по заданному адресу. Во время обращения к памяти может формироваться множество запросов. Если потоки из вычислительного блока обращаются к близким адресам памяти, то несколько запросов могут быть сгруппированы в один и исполнены параллельно. Это свойство используется для повышения эффективности вычислительного процесса в задачах молекулярной динамики.

Технология CUDA описывает работу графических процессоров в виде блоков выборки текстур (рис.1.2). В блоке выборки текстур реализована надстройка для работы с памятью и потоковые мультипроцессоры. Потоковый мультипроцессор состоит из кэша инструкций, кэша констант, блока выборки инструкций, разделяемой памяти, графических ядер, блока специальных функций, регистровой памяти. Система расписаний обрабатывает данные из блока инструкций и запросы вычислительных потоков на доступ к памяти. Процесс вычисления запускается только тогда, когда есть

необходимые ресурсы. Во время запроса на доступ к памяти одними вычислительными потоками в блоке инструкций может формироваться новый набор инструкций, также в этот момент могут исполняться потоки, которым не нужен доступ к памяти или данные для которых уже получены.

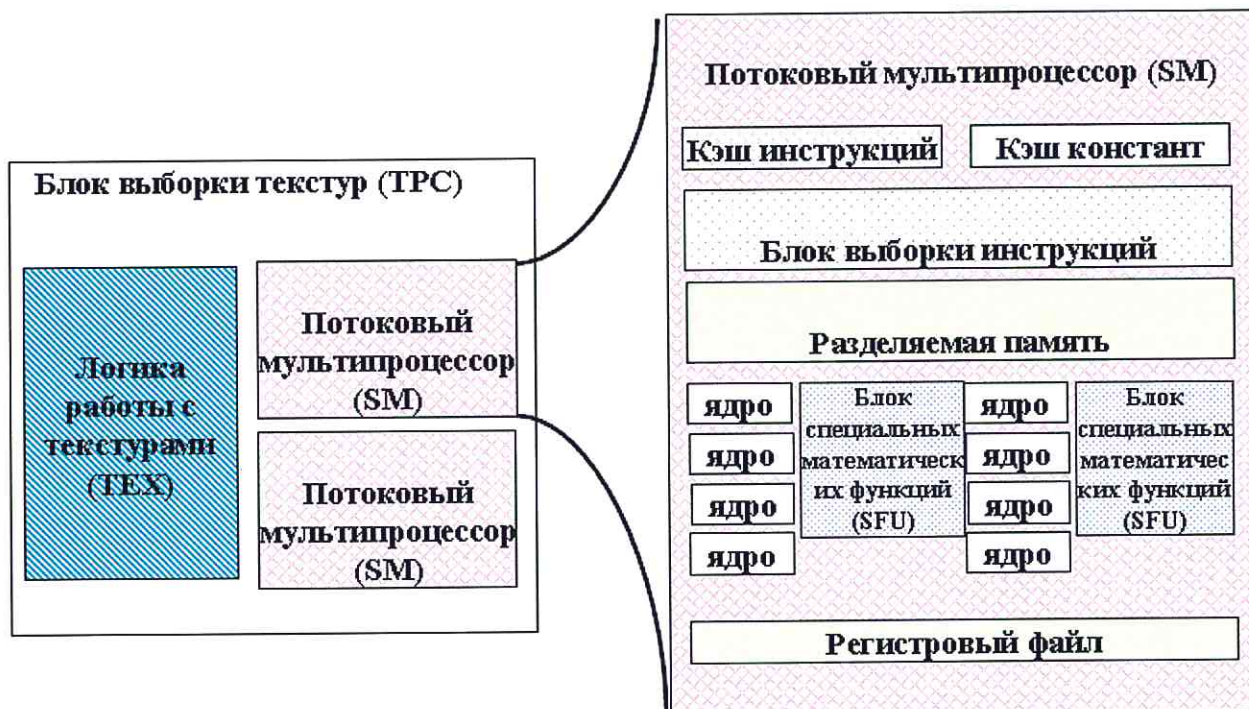


Рисунок 1.2. Схематическое изображение потокового мультипроцессора

Язык CUDA C предоставляет в распоряжение программы разделяемую память (shared memory). Компилятор CUDA C создает копию разделяемых переменных (`__shared__`) в каждом блоке, и все вычислительные потоки используют их совместно. Буферы разделяемой памяти находятся в самих GPU, а не в общей памяти (DRAM – динамическое запоминающее устройство с произвольной выборкой). Таким образом, задержки при доступе к разделяемой памяти существенно меньше, чем к глобальной памяти (DRAM). Чтобы вычислительные потоки взаимодействовали, необходим механизм синхронизации. Например, если вычислительный поток А записывает какое-либо значение в разделяемую память, то вычислительный поток В, использующий это значение, не должен приступать к работе, пока вычислительный поток А не завершил операцию.

Помимо глобальной и разделяемой памяти NVIDIA предоставляет и другие виды памяти. Константная память имеет ограничения по сравнению с глобальной, но даёт большую производительность. Если обращения к константной памяти из всех потоков (thread) происходят к одним и тем же данным только для чтения, то трафик между памятью и процессором уменьшается за счет трансляции результатов операции чтения на половину вычислительного блока (half warp) и за счет кэша памяти на кристалле. Доступ к памяти является узким местом во многих алгоритмах, поэтому наличие механизма, позволяющего улучшить производительность доступа, необходимо для эффективных вычислений.

Память графической карты изменяет скорость работы алгоритма в зависимости от способа доступа. Поэтому на графической плате реализовано несколько видов памяти. Глобальная память имеет возможность динамического случайного доступа (DRAM) и может хранить большой объём данных, но вносит большие задержки. Считывание одного байта из глобальной памяти происходит за 400-600 тактов. Но технология DRAM памяти позволяет делать параллельный доступ, если выполнен запрос к последовательным блокам. В таком случае доступ к памяти происходит с максимальной пропускной способностью. Перед выполнением инструкции мультипотока аппаратное обеспечение проверяет, в какие блоки памяти будет запрос и назначает последовательность считывания.

1.2 Метод молекулярной динамики

Молекулярная динамика (MD) – это техника моделирования, в которой координаты и скорости частиц изменяются со временем по закону Ньютона под действием сил межатомного взаимодействия. Для численного решения системы уравнений движения задаются начальные значения и определённые граничные условия.

Молекулярная динамика может рассматриваться как вариант решения задачи N тел. Идея заключается в том, чтобы получить новые положения частиц с течением времени. В методе могут использоваться сложные потенциалы взаимодействия

атомов, на ближнем и дальнем расстоянии потенциалы могут выбираться разные в зависимости от физической структуры вещества.

Пусть в системе имеется N атомов, атомы находятся в потенциальном поле E_{ij} . Для моделирования взаимодействия атомов необходимо решить систему N уравнений:

$$m_i \ddot{\mathbf{r}}_i = \sum_{j \neq i}^N (-\nabla E_{ij}), \quad (1.1)$$

где \mathbf{r}_i – радиус-вектор атома с номером i , m_i – масса атома, E_{ij} – межатомный потенциал для частицы i и j . Схема вычислительного процесса показана на рис. 1.3.



Рисунок 1.3. Схема метода молекулярной динамики (MD)

В системе N атомов, которые занимают объем V (рис. 1.4) можно определить полную энергию $E = K + U$, где K – кинетическая энергия:

$$K = \sum_{i=1}^N \frac{3}{2} m_i \dot{\mathbf{r}}_i^2(t), \quad (1.2)$$

и U – потенциальная энергия:

$$U = U(x(t)). \quad (1.3)$$

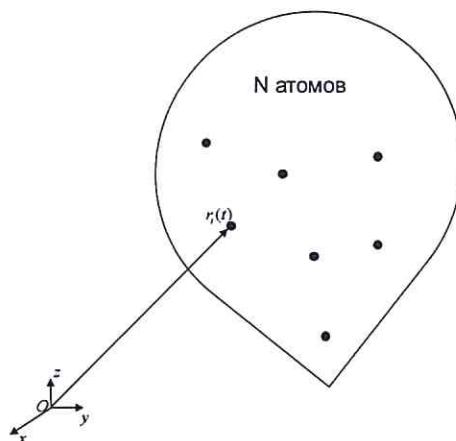


Рисунок 1.4. Пример системы для молекулярно-динамического моделирования

Метод молекулярно динамики основан на численном решении уравнений движения. Частицы, находящиеся в выделенном объёме, взаимодействуют друг с другом посредством заданного потенциала. Выбор потенциала взаимодействия частиц является наиболее важным этапом построения численной модели. Для решения уравнений движений используются конечно-разностные методы аппроксимации.

1.3 Методы аппроксимации уравнений движения

Для интегрирования уравнений динамики частиц в молекулярно-динамическом моделировании наиболее распространение нашли схемы второго порядка точности. Дадим краткое описание этих схем:

Алгоритм с перешагиванием записывается выражениями:

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{v}(t + \frac{1}{2} \Delta t) \Delta t. \quad (1.4)$$

$$\mathbf{v}(t + \frac{1}{2} \Delta t) = \mathbf{v}(t - \frac{1}{2} \Delta t) + \mathbf{a}(t) \Delta t. \quad (1.5)$$

где \mathbf{r} – радиус-вектор частицы, \mathbf{v} – скорость, \mathbf{a} – ускорение.

В этом методе скорости вычисляются в момент времени $t + \frac{1}{2} \Delta t$, и используются для вычисления координаты в момент $t + \Delta t$. Таким образом, шаг по

скорости перешагивает через момент времени по координате. Преимуществом такого метода является явное вычисление скорости, недостатком является то, что координата и скорость вычисляются в разные моменты времени. Скорость в момент времени t можно определить выражением:

$$\mathbf{v}(t) = \frac{1}{2} \left[\mathbf{v}(t - \frac{1}{2} \Delta t) + \mathbf{v}(t + \frac{1}{2} \Delta t) \right]. \quad (1.6)$$

Для вывода схемы интегрирования нужно записать:

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{v}(t) \Delta t + \frac{1}{2} \mathbf{a}(t) \Delta t^2, \quad (1.7)$$

$$\mathbf{r}(t - \Delta t) = \mathbf{r}(t) - \mathbf{v}(t) \Delta t + \frac{1}{2} \mathbf{a}(t) \Delta t^2. \quad (1.8)$$

Суммируя эти выражения, получается:

$$\mathbf{r}(t + \Delta t) = 2\mathbf{r}(t) - \mathbf{r}(t - \Delta t) + \mathbf{a}(t) \Delta t^2. \quad (1.9)$$

Алгоритм Верле использует координаты и ускорения в момент времени t и координаты в момент времени $t + \Delta t$, чтобы вычислить новые координаты в момент времени $t + \Delta t$. Преимущество такого алгоритма в том, что он прост и требует немного памяти для хранения данных на каждом шаге. В качестве недостатка алгоритма можно отметить его недостаточную точность. Алгоритм Верле даёт третий порядок точности вычисления координаты и второй порядок точности вычисления скорости.

Наибольшее распространение получил метод скоростей Верле, который даёт координаты, скорости, ускорения в момент времени t . Схема записывается в виде:

$$\begin{aligned} \mathbf{r}(t + \Delta t) &= \mathbf{r}(t) + \mathbf{v}(t) \Delta t + \frac{1}{2} \mathbf{a}(t) \Delta t^2, \\ \mathbf{a}(t + \Delta t) &= \frac{\mathbf{F}(t)}{m}, \\ \mathbf{v}(t + \Delta t) &= \mathbf{v}(t) + \frac{1}{2} [\mathbf{a}(t) + \mathbf{a}(t + \Delta t)] \Delta t. \end{aligned} \quad (1.10)$$

В алгоритме явно используется скорость, для чего приходится выделять дополнительную память. В такой записи на момент $t + \Delta t$ для вычисления скорости необходимо хранить новые и предыдущие ускорения. Можно преобразовать запись схемы так, что скорости будут обновляться в момент $t + \frac{1}{2} \Delta t$, и в таком случае предыдущие ускорения на момент $t + \Delta t$ не потребуются:

$$\mathbf{v}(t + \frac{1}{2}\Delta t) = \mathbf{v}(t) + \frac{1}{2}\Delta t \cdot \mathbf{a}(t). \quad (1.11)$$

Выражение для координат:

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \Delta t \cdot \mathbf{v}(t + \frac{1}{2}\Delta t). \quad (1.12)$$

Ещё раз рассчитываются скорости:

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t + \frac{1}{2}\Delta t) + \frac{1}{2}\Delta t \cdot \mathbf{a}(t + \Delta t). \quad (1.13)$$

Это позволяет сократить затраты памяти и уменьшить количество дополнительных преобразований с ускорениями.

1.4 Потенциалы межчастичного взаимодействия

Простейшими потенциалами взаимодействия являются парные потенциалы, наиболее известный из которых – потенциал Леннарда-Джонса. Расчёт реальных структур, однако, требует использования более сложных многочастичных потенциалов, как, например, потенциал Бреннера.

В общем случае действие межатомных потенциалов распространяется на бесконечное удаление. При моделировании действие потенциала ограничивают радиусом усечения r_c . Но простое усечение даёт скачок энергии на его границе, поэтому применяется функция усечения, которая сглаживает действие потенциала. Выбирается расстояние r_s так, что в интервале $[r_s, r_c]$ потенциал и производная потенциала являются гладкими и стремятся к нулю. Если расстояние между атомами i и j превышает r_c , то программа моделирования не учитывает действие атома j на атом i . Таким образом, количество операций для N атомов уменьшается от $N(N-1)$ до $N \cdot m$, где обычно $m \ll N$.

Потенциал Леннарда-Джонса (LJ) для двух атомов на расстоянии x записывается выражением:

$$u(x) = -\frac{A}{x^6} + \frac{B}{x^{12}}, \quad (1.14)$$

что эквивалентно записи:

$$u(x) = 4\varepsilon \left[-\left(\frac{\sigma}{x}\right)^6 + \left(\frac{\sigma}{x}\right)^{12} \right]. \quad (1.15)$$

Равновесное расстояние x_0 запишется:

$$x_0 = 2^{\frac{1}{6}} \sigma = \left(\frac{2B}{A} \right)^{\frac{1}{6}}, \quad (1.16)$$

а высота барьера ε будет:

$$\varepsilon = \frac{A^2}{4B}. \quad (1.17)$$

где A и B – притягивающая и отталкивающая константы потенциала Леннарда-Джонса.

Однако для более точного описания широкого спектра свойств наносистем необходимо использовать многочастичные потенциалы, отражающие реальные свойства межчастичных связей. В качестве типичного представителя таких потенциалов в настоящей работе рассматривается потенциал Бреннера. С одной стороны, его использование позволяет исследовать довольно широкий класс наносистем, с другой – на примере этого потенциала можно отчетливо проследить основные проблемы молекулярно-динамического моделирования на графических процессорах.

Потенциал Бреннера второй модификации.

Этот потенциал учитывает разрывы в ковалентной связи и составлен с учётом возникающих изменений в гибридизации атома в рамках классического потенциала. Он даёт мощный инструмент для моделирования сложных химических процессов в большинстве многоатомных систем. Этот усовершенствованный потенциал содержит улучшенные аналитические выражения для функций и расширенную базу данных по сравнению с первой модификацией потенциала Бреннера. Этот потенциал лучше описывает параметры системы: энергию связи, длину связи и коэффициент упругости для углеводородных молекул, а также свойства упругости, энергию внутренних дефектов и поверхностную энергию для алмаза.

Общий вид аналитической функции, используемой для выражения потенциальной энергии межмолекулярного взаимодействия, первоначально получен Абеллем из теории псевдопотенциальной энергии [31]. В локальном базисе невозмущенных атомных орбиталей энергия химической связи E_b может быть записана как простая сумма из ближайших соседей в следующем виде:

$$E^B = \sum_i \sum_{j>i} (V^R(r_{ij}) - b_{ij}V^A(r_{ij})). \quad (1.18)$$

где $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ – расстояние между i -ым и j -ым атомами.

Функции $V^R(r)$ и $V^A(r)$ парного взаимодействия определяют все межатомные отталкивания (ядро-ядро) и притяжения между валентными электронами соответственно. Переменная r_{ij} – расстояние между парой ближних соседних атомов i и j , и b_{ij} – порядок связи между атомами i и j , который получается из теории Хакла или подобных теорий уровней электронных оболочек.

Отмечено, что локальная координата N есть величина первоначально управляющая значением порядка связи, и, используя решетку Бете, получается простое выражение для регулярных структур:

$$b \propto (N)^{-\frac{1}{2}}. \quad (1.19)$$

В общем виде выражение (1.18) Абелля-Терсоффа используется для полной потенциальной энергии. Следуя ранним выражениям связи углеводородов [32] и согласно последующим работам [33], эмпирическая функция порядка связи записывается как сумма следующих слагаемых:

$$b_{ij} = \frac{1}{2}(b_{ij}^{\sigma-\pi} + b_{ji}^{\sigma-\pi}) + \pi_{ij}^{RC} + b_{ij}^{DH}. \quad (1.20)$$

Значения для функции $b_{ij}^{\sigma-\pi}$ и $b_{ji}^{\sigma-\pi}$ зависят от расположения атома и угла связи между атомами i и j соответственно.

Значение π_{ij}^{RC} зависит от того, имеют ли атомы i и j связь между собой или связаны как часть системы атомов. Значение выражения b_{ij}^{DH} зависит от двугранного угла в двойной связи углерод-углерод. Это выражение вместе с (1.18) используется для определения энергии связи благодаря ковалентным связям любого набора атомов водорода и углерода.

Согласно Терсоффу [34-47], в первое выражение для углеводородов включены члены выражения Морзе для парного взаимодействия в уравнении (1.18). Однако было определено, что эта форма слишком ограничена для одновременного заполнения констант равновесных расстояний, энергий и сил для связей углерод-углерод. Более того, эта форма имела тот недостаток, что оба члена стремились к конечному

значению при уменьшении расстояния между атомами, что ограничивало возможность моделирования процессов, включающие сильные столкновения атомов.

Во второй модификации потенциала формы:

$$V_A(r_{ij}) = f^c(r_{ij}) \sum_{n=1,3} B_n e^{-\beta_n r_{ij}} \quad (1.21)$$

и

$$V^R(r_{ij}) = f^c(r_{ij}) \left(1 + \frac{Q}{r_{ij}}\right) A e^{-\alpha r_{ij}} \quad (1.22)$$

используются для парных членов. Индекс n относится к сумме в уравнении (1.21), и r есть скаляр расстояния между атомами. Ограниченная функция Кулона, использованная для отталкивания парного взаимодействия (1.22), стремится к бесконечности при приближении межатомного расстояния к нулю. Член притяжения (1.21) обладает достаточной гибкостью для одновременного представления свойств связи, которые не могут быть представлены членами Морзе. Функция f^c ограничивает расстояние ковалентных взаимодействий. Параметр подстановки для углерода предполагает значение единицы для f^c для ближайших соседей и ноль для всех остальных межатомных расстояний.

База данных, используемая для подстановки значений парного взаимодействия и значения порядка связи b_{CC} , состоит из констант равновесных расстояний, энергий и сил натяжения для одинарных (из алмаза), сопряженных двойных (из графита), двойных (из этилена) и тройных (из этана) связей. Значения для этих переменных определяются следующим образом. Во-первых, расстояния, соответствующие минимальной энергии, берутся из стандартных значений в литературных источниках; они даются в таблице 1.1.

Таблица 1.1. Данные, используемые в схеме подбора.

Образец	ΔH (eV) ^a	Энергия нулевого уровня (eV) ^b	Энергия расщепления (eV) ^c	Энергия связи (eV)	Длина связи (Å)	коэффициент упругости (10 ⁵ дин см ⁻¹)
Алмаз	-	-	7.36 ^d	3.68	1.54	4.73 ^e
Графит	-	-	7.40 ^d	4.93	1.42	6.98 ^f
SC	-	-	4.70 ^d	1.57	1.93 ^d	-
FCC	-	-	2.86 ^d	0.46	2.18 ^d	-
Ацетилен	2.356	0.701	17.57	8.51	1.20	16.00 ^g

Этилен	0.629	1.339	24.41	6.30	1.33	9.56 ^f
Этан	-0.717	1.955	30.84	3.69	1.54	4.79 ^f
Циклогексан	-0.868	4.496	76.45	3.69	1.54	4.79 ^f
Бензол	1.041	2.677	59.29	5.36	1.39	7.85 ^f
Связь С-Н	-	-	-	4.526	1.09	-

^a из [38]

^b вычислено в предположении гармонических колебаний

^c вычислено в предположении $\Delta H_{\text{carbon}} = 7.37 \text{ eV}$, $\Delta H_{\text{hydrogen}} = 2.239 \text{ eV}$

^d из [39-40]

^e вычислено из модуля объёмной деформации алмаза $4.4 \times 10^{12} \text{ дин} \cdot \text{см}^{-2}$

^f из выражения (7) $a = 4.99682 \times 10^5 \text{ \AA}^3 \text{ дин} \text{ см}^{-1}$, $b = 0.52554 \text{ \AA}$

^g из [41]

Значения для коэффициента упругости получены из выражения правила Бадгера:

$$K = a(r_e - b)^{-3}, \quad (1.23)$$

где K – коэффициент упругости, r_e – расстояние для минимальной энергии связи, a и b – изменяемые параметры. Значения последних двух параметров определены из коэффициента упругости и расстояний, соответствующих минимальной энергии для углерод-углеродных одинарных и тройных связей. Они представлены в таблице 1.1. Уравнение (1.23) используется для определения непрерывного набора коэффициентов упругости, включая оставшиеся значения из таблицы 1.1.

Для того чтобы уменьшить количество переменных, необходимых для подстановки парных членов, используются первоначальные предположения для переменных A_{CC} , Q_{CC} , α_{CC} , B_{CC1} , B_{CC2} , β_{CC1} и β_{CC2} . Значения для B_{CC3} и β_{CC3} определяются из условия равновесия одинарной связи. Если задать значения для порядка связи b_{CC} для одинарной связи, тогда можно найти B_{CC3} , и все три значения B_{CCn} нормируются этим значением. Каждый порядок связи для оставшихся трёх равновесных длин связи тогда определяется через приравнивание к нулю производных уравнений (1.21) и (1.22) при всех расстояниях. Значения для оставшихся энергий связи и коэффициента упругости тогда вычисляются по методу наименьших квадратов для значений из таблицы 1.1. Согласующиеся значения для порядка связи для случаев координированности (количество соседних атомов) из шести или двенадцати вычисляются так, чтобы дать минимальную энергию связи

соответственно кубической и гранецентрированной кристаллическим решеткам. В конце этого процесса каждые из b_{CC} , B_{CCn} нормируются так, чтобы порядок связи был равен единице, а энергия связи была равна значению для углерода в алмазе. Окончательные значения параметров, полученные таким способом, представлены в таблице 1.2.

Таблица 1.2. Параметры для углерод-углеродных парных членов.

Параметр	Значение (C – C)	Значение (C – H)	Значение (H – H)
Q	0.3134602960833 Å	0.340776 Å	0.370471487045 Å
A	10953.544162170 эВ	149.94099 эВ	32.817355747 эВ
α	4.7465390606595 Å ⁻¹	4.1025498 Å ⁻¹	3.536298648 Å ⁻¹
B ₁	12388.79197798 эВ	32.355187 эВ	29.632593 эВ
B ₂	17.56740646509 эВ	0	0
B ₃	30.71493208065 эВ	0	0
β_1	4.7204523127 Å ⁻¹	1.4344581 Å ⁻¹	1.71589217 Å ⁻¹
β_2	1.4332132499 Å ⁻¹	0	0
β_3	1.3826912506 Å ⁻¹	0	0
D _{min}	1.7 Å	1.3 Å	1.1 Å
D _{max}	2.0 Å	1.8 Å	1.7 Å

Аналитические выражения для каждого члена уравнения (1.18) подбираются одновременно по дискретным значениям порядка связи, как определено выше, и по другим параметрам твердого и молекулярного углерода, как описано ниже. Первый член (1.20) дается выражением:

$$b_{ij}^{\sigma-\pi} = \left[1 + \sum_{k \neq i, j} f_{ik}^c(r_{ik}) G(\cos(\theta_{ijk})) e^{\lambda_{ijk}} + P_{ij}(N_i^C, N_i^H) \right]^{-\frac{1}{2}}. \quad (1.24)$$

Как и в уравнениях (1.21) и (1.22), индексы соответствуют номеру атома, функция $f^c(r)$ обеспечивает учёт только ближних соседних атомов. Функция P представляется бикубическим сплайном, и значения N_i^C и N_i^H показывают количество соседних атомов углерода и водорода соответственно для атома i . Они определяются как суммы:

$$N_i^C = \sum_{\substack{\text{carbon atom} \\ k \neq i, j}} f_{ik}^c(r_{ik}), \quad (1.25)$$

$$N_i^H = \sum_{\substack{\text{hydrogen atom} \\ l \neq i, j}} f_{il}^c(r_{il}). \quad (1.26)$$

Для твердого углерода значения λ и функция P берутся нулевыми (с одной особенностью, рассмотренной ниже). Последняя функция может быть представлена как корректирующая в аналитическом выражении функции порядка связи для твердого вещества, что необходимо для точного моделирования молекулярной энергии связи. Аналогичное выражение даётся для $b_{ji}^{\sigma-\pi}$ заменой индексов i и j в формуле (1.24).

Таблица 1.3. Параметры угловой функции для углерода.

θ, rad	$G_c(\cos(\theta_{ijk}))$	$\frac{dG_c}{d(\cos(\theta_{ijk}))}$	$\frac{d^2G_c}{d(\cos(\theta_{ijk}))^2}$	$\gamma(\theta_{ijk})$
0	8	—	—	1
$\frac{\pi}{3}$	2.0014	—	—	0.416335
$\frac{\pi}{2}$	0.37545	—	—	0.271856
0.6082π	0.09733	0.40000	1.98000	—
$\frac{2\pi}{3}$	0.05280	0.17000	0.37000	—
π	-0.001	0.10400	0.00000	—

Функция $G_c(\cos(\theta_{ijk}))$ в формуле (1.24) отображает вклад каждого ближнего соседнего атома в эмпирический порядок связи в зависимости от косинуса угла связи между атомами i и k и атомами i и j . Эта функция определена следующим образом. Кристаллическая решетка алмаза и графитовых слоев состоят только из углов 109.47° и 120° соответственно. Формула (1.24) с учетом значения b_{cc} на рисунке 1.5 (верхний график) дает значения для $G_c(\cos(\theta_{ijk}))$ для каждого из этих углов. Разница энергии между линейной молекулой C_3 и молекулой, согнутой под углом 120° , как дают вычисления функционала плотности [42], используется для нахождения значения $G_c(\cos(\theta = 180^\circ))$. Поэтому в кубической кристаллической решетке угол связи между ближайшими соседями 90° и 180° , а значение $G_c(\cos(\theta = 180^\circ))$ вместе со значением порядка связи для кубической кристаллической решетки используется для нахождения значения $G_c(\cos(\theta = 90^\circ))$. Окончательно гранцентрированная кристаллическая решетка содержит углы 60° , 120° , 180° и 90° . Значение для

$G_c(\cos(\theta = 60^\circ))$, следовательно, может быть определено из $G_c(\cos(\theta))$, заданной выше, и b_{cc} для гранцентрированной кристаллической решетки. Это приближение для выбора углового взаимодействия неприемлемо при увеличении угла равновесной связи. Вместо этого используется функция (1.30) для всех гибридизаций атомов. Это позволяет изменять угол связи, влияющий на энергетический потенциал непрерывным образом, как в химических реакциях. Более того, предполагается, что выбор $G_c(\cos(\theta))$ таким способом даёт значения, которые монотонно уменьшаются при увеличении угла. Такое поведение может быть интерпретировано благодаря парному отталкиванию валентных электронов с внешних оболочек.

Для завершения аналитического выражения угловой функции $G_c(\cos(\theta))$ используется полиномиальный сплайн пятого порядка по $\cos(\theta)$, который определяется в трех интервалах $0^\circ < \theta < 109.47^\circ$, $109.47^\circ < \theta < 120^\circ$, $120^\circ < \theta < 180^\circ$. Поэтому шесть коэффициентов – три значения для функции и три для её производной необходимы в каждом интервале для определения угловой функции. Для значений угла θ между 109.47° и 120° полином заполняется значениями $G_c(\cos(\theta))$, при θ равном 109.47° и 120° заполняется значениями, определенными выше как первая и вторая производная $G_c(\cos(\theta))$ по $\cos(\theta)$ для этих двух углов. Значения для второй производной $G_c(\cos(\theta))$ для углов 109.47° и 120° выбираются соответственно модулям упругости c_{11} для алмаза и поверхностной упругости c_{11} для графита. Первая производная выбирается так, чтобы убрать нежелательные колебания в сплайнах. Данные, необходимые для конструирования сплайна, представлены в таблице 1.3.

Две добавки необходимо сделать в угловую функцию, определенную выше, для полиномиального сплайна. Первая определяется тем, что в диапазоне $120^\circ < \theta < 180^\circ$ необходимо немного меньшее значение $G_c(\cos(\theta = 180^\circ))$, чем определено выше, для определения первой производной, которая одновременно убирает нежелательные колебания и даёт удовлетворительное значение для модуля изгиба для ацетилена. В дальнейшем используется значение -0.01 для $G_c(\cos(\theta = 180^\circ))$, энергия связи для молекулы C_3 восстанавливается выбором значения функции $P_{cc}(1,0)$. Первая и вторая

производные $G_c(\cos(\theta = 180^\circ))$ определяются подстановкой модуля изгиба ацетилена и исключением нежелательных колебаний сплайна соответственно. Эти две производные вместе со значением функции в узлах достаточны для определения коэффициентов сплайна. Значения функции и производных $G_c(\cos(\theta = 180^\circ))$, необходимые для составления коэффициентов сплайна, сведены в таблице 1.3.

Для угла θ между 0° и 109.47° сплайн заполняется значениями первой и второй производной $G_c(\cos(\theta = 109.47^\circ))$ и значениями $G_c(\cos(\theta))$ угла θ равного 90° , 60° , 0° . Следующее значение даёт гладкую функцию по углу при угле меньше 60° . Анализ энергии маленьких колец углеводородов как бы то ни было показывает, что значения $G_c(\cos(\theta))$, определённые выше для углов равных 90° и 60° , слишком большие для не полностью связанных атомов углерода (они определены выше для восьми и двенадцати связанных атомов углерода соответственно). Для определения обоих случаев, когда атомы имеют неполные связи и лишние связи, определяется второй сплайн $\gamma_c(\cos(\theta))$ для углов меньших 109.47° , который связан с $G_c(\cos(\theta))$. Эта функция содержит значение первой и второй производной при θ равном 109.47° для начальной функции $G_c(\cos(\theta))$, но имеет меньшие значения для углов 90° , 60° , 0° , собранных в таблице 1.3. Уточнённая угловая функция между 109.47° и 0° для i -ого атома углерода даётся выражением:

$$g_c = G_c(\cos(\theta)) + Q(N'_i)[\gamma_c(\cos(\theta)) - G_c(\cos(\theta))], \quad (1.27)$$

где функция Q определяется

$$Q_i(N'_i) = \begin{cases} 1 & N'_i < 3.2 \\ \frac{1 + \cos(2\pi(N'_i - 3.2))}{2} & 3.2 < N'_i < 3.7 \\ 0 & N'_i > 3.7 \end{cases} \quad (1.28)$$

Количество N'_i связанных с атомом i определяется как

$$N'_i = N_i^C + N_i^H, \quad (1.29)$$

где N_i^C и N_i^H определяются формулами (1.25) и (1.26) соответственно. Соединяя два сплайна таким способом, гарантируется, что общая функция имеет непрерывные производные независимо от значения N'_i . Угловая функция при N'_i равном 3.7 и

больше показана на рис. 1.5 сплошной линией. Уточнённая функция для N'_i меньше 3.2 показана тонкой линией на рис. 1.5.

Для вычисления угловой функции в программе удобно использовать её выражение через косинусы:

$$G_c(\cos(\theta_{ijk})) \approx \sum_{n=0}^5 a_n (\cos(\theta_{ijk}))^n . \quad (1.30)$$

Также нужно помнить, что в диапазоне $0^\circ < \theta < 109.47^\circ$ угловая функция даёт слишком большие значения для атомов с меньшим количеством связей. Поэтому используется уточненная функция, которая имеет такое же выражение:

$$g_c(\cos(\theta_{ijk})) \approx \sum_{n=0}^5 a_n (\cos(\theta_{ijk}))^n . \quad (1.31)$$

Параметры полинома пятого порядка представлены в таблице 1.4.

Таблица 1.4. Параметры полинома косинусов для угловых функций.

функция	угол	a ₀	a ₁	a ₂	a ₃	a ₄	a ₅
$g_c(\cos(\theta))$	$0^\circ < \theta < 109.47^\circ$	0.27186	0.48922	-0.43286	-0.56140	1.27111	-0.03793
$G_c(\cos(\theta))$	$0^\circ < \theta < 109.47^\circ$	0.281695	1.062743	2.1363075	2.5334145	1.5544035	0.3862485
$G_c(\cos(\theta))$	$109.47^\circ < \theta < 120^\circ$	0.69669	5.54440	23.4320	55.9476	69.8760	35.3117
$G_c(\cos(\theta))$	$120^\circ < \theta < 180^\circ$	0.28160	1.06200	2.13400	2.53000	1.55200	0.38560

График угловой функции, построенный по этому полиному, показан на рис. 1.5.

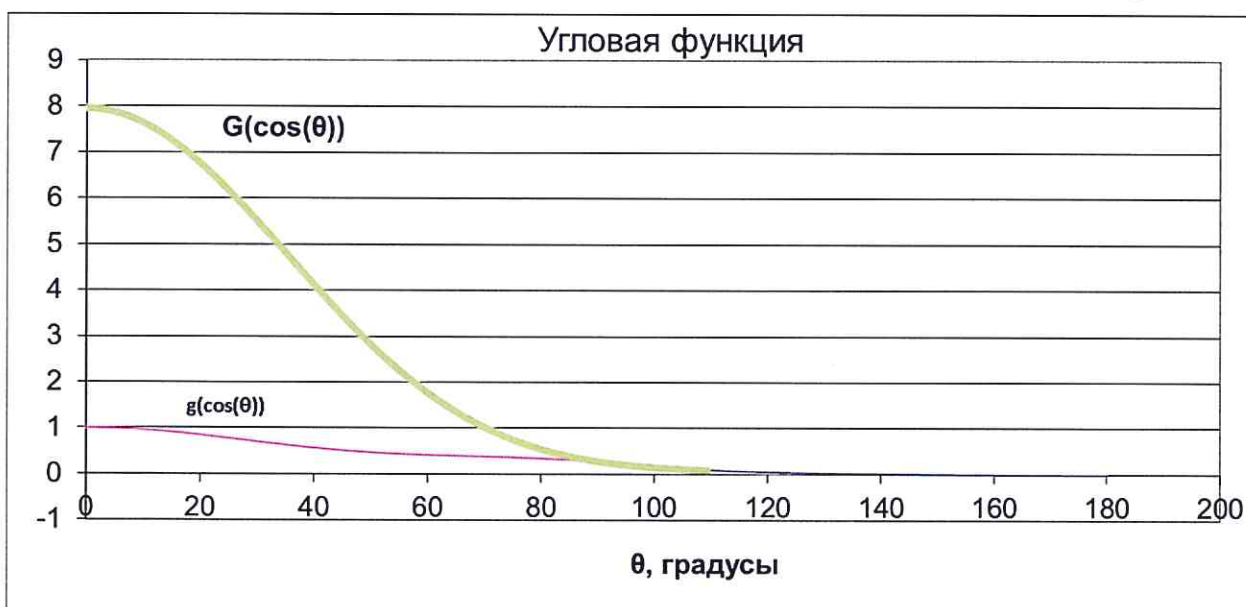


Рисунок 1.5. График угловой функции

График угловой функции $G_c(\cos(\theta))$ имеет перегибы, что приводит в резкому численному скачку на границах диапазона, поэтому на интервале $0^\circ < \theta < 120^\circ$ используется экспоненциальный сплайн:

$$G_c(\theta_{ijk}) \approx A + B \cos(\theta_{ijk}) + Ce^{p \cos(\theta_{ijk})} + De^{-p \cos(\theta_{ijk})}. \quad (1.32)$$

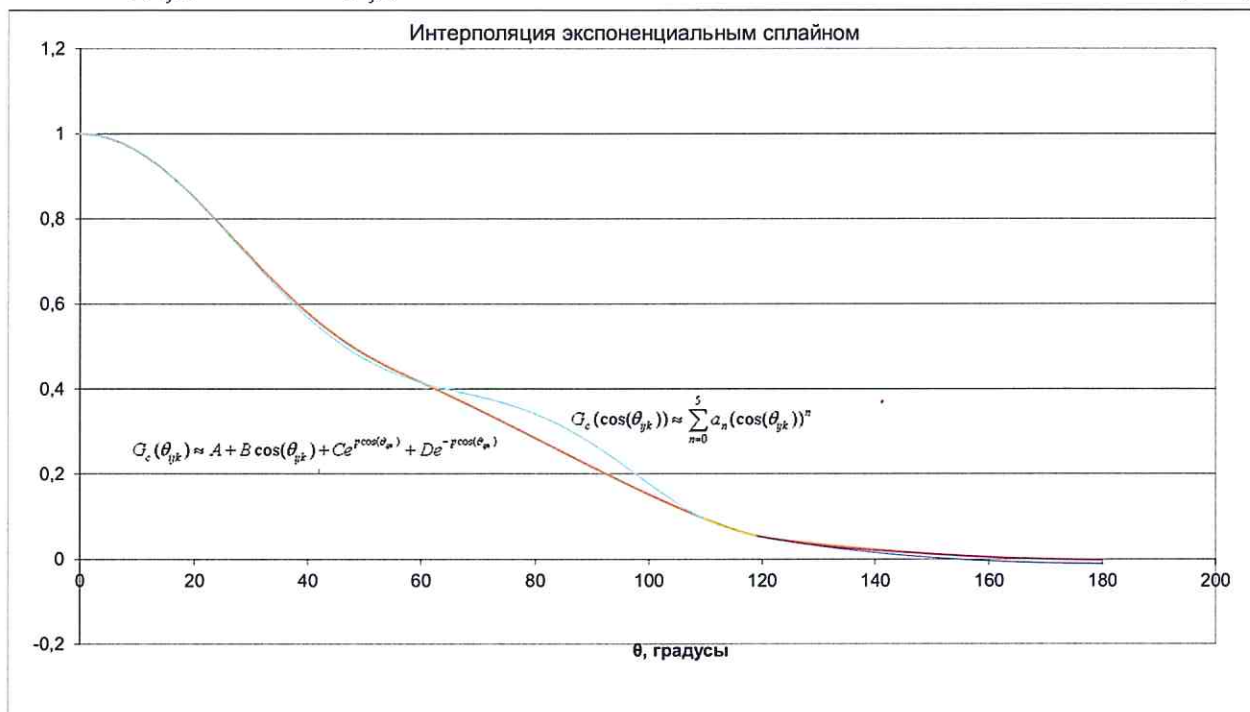


Рисунок 1.6. График угловой функции

Член π_{ij}^{RC} в уравнении (1.20) отображает влияние энергии основных уровней и энергии π -связи. Как известно из первой формы потенциала [32], этот член необходим для правильного описания орбитального строения, вакантных энергетических уровней в алмазе и для подсчёта пространственных эффектов связи – таких, которые обуславливают разные свойства связей углерод-углерод в графите и бензоле. Эта функция дается трикубическим сплайном F :

$$\pi_{ij}^{RC} = F_{ij}(N_i^l, N_j^l, N_{ij}^{conj}), \quad (1.33)$$

который зависит от общего числа соседних связанных атомов i и j и определяется выражением (1.29), тогда как N_{ij}^{conj} зависит от локальных связей.

Выражение N_{ij}^{conj} даётся следующей функцией:

$$N_{ij}^{conj} = 1 + \left[\sum_{k \neq i, j}^{carbon} f_{ik}^c(r_{ij}) F(X_{ik}) \right]^2 + \left[\sum_{l \neq i, j}^{carbon} f_{jl}^c(r_{jl}) F(X_{jl}) \right]^2, \quad (1.34)$$

где

$$F(x_{ik}) = \begin{cases} 1 & x_{ij} < 2 \\ \frac{1 + \cos(2\pi(x_{ij} - 2))}{2} & 2 < x_{ij} < 3 \\ 0 & x_{ij} > 3 \end{cases} \quad (1.35)$$

и

$$x_{ij} = N_k^l - f_{ik}^c(r_{ik}). \quad (1.36)$$

Если все атомы углерода, которые связаны в пары i и j , имеют 4 или более соседей, уравнения (1.34)-(1.36) дают значение единицы для N_{ij}^{conj} , и связи между этими атомами не рассматриваются как часть связанной системы. Если число связанных соседних атомов уменьшается, N_{ij}^{conj} становится больше единицы, и это показывает конфигурацию связанной системы. В дальнейшем, выражения (1.34)-(1.36) устанавливаются для различных конфигураций, которые могут привести к связанности. Например, значение N_{ij}^{conj} для связи углерод-углерод в графите равно девяти, тогда как для бензола равно трём. Она даёт существенную дополнительную гибкость для подбора энергий связанных систем. Эти выражения позволяют учитывать эффекты связности структуры в пределах взаимодействия ближайших соседей, не решая полную систему уравнений взаимодействующих атомов. В дальнейшем изменения координированности в разрывах и образовании связей вычисляются на каждом шаге. Это приближение значительно уменьшает время вычисления, так как включает связи первого порядка.

Дискретные значения для функции, определяемой уравнением (1.33), выбираются для удовлетворения энергиям статических структур, затем используется трикубический сплайн для интерполяции значений между ними. Значения этой функции собраны в таблице 1.5.

Таблица 1.5. Значения функции F_{ss} .

i	j	k	$F(i, j, k)$	Вещество
1	1	1	0.105000	Ацитилен
1	1	2	-0.0041775	$H_2C=C=CH$
1	1	3-9	-0.0160856	C_4
2	2	1	0.09444957	$(CH_3)_2C=C(CH_3)_2$
2	2	2	0.02200000	Бензол

2	2	3	0.03970587	Средние значения из F(2,2,2) и F(2,2,9)
2	2	4	0.03308822	
2	2	5	0.02647058	
2	2	6	0.01985293	
2	2	7	0.01323529	
2	2	8	0.00661764	
2	2	9	0.0	
0	1	1	0.04338699	C ₂ H
0	1	2	0.0099172158	C ₃
0	2	1	0.0493976637	CCH ₂
0	2	2	-0.011942669	CCH(CH ₂)
0	3	1-9	-0.119798935	H ₃ CC
1	2	1	0.0096495698	H ₂ CCH
1	2	2	0.030	H ₂ C=C=CH ₂
1	2	3	-0.0200	C ₆ H ₅
1	2	4	-0.0233778774	Среднее из F(1,2,3) и F(1,2,6)
1	2	5	-0.0267557548	
1	2	6-9	-0.030133632	Углеродные вакансии
1	3	2-9	-0.124836752	H ₃ C-CCH
2	3	1-9	-0.044709383	Алмазные вакансии

Значения функции F_{cc} , не представленные в таблице, равны 0. Также выполняются соотношения $F(i, j, k) = F(j, i, k)$, $F(i, j, k > 9) = F(i, j, 9)$, $F(i > 3, j, k) = F(3, j, k)$, и $F(i, j > 3, k) = F(i, 3, k)$.

Производные представлены как центральные конечные разности, таблица 1.6.

Таблица 1.6. Значения производных функции F_{cc} .

Производная	Значение
$\partial F(2,1,1)/\partial i$	-0.052500
$\partial F(2,1,5-9)/\partial i$	-0.054376
$\partial F(2,3,1)/\partial i$	0.00000
$\partial F(2,3,2-6)/\partial i$	0.062418
$\partial F(2,2,4-8)/\partial k$	-0.006618
$\partial F(2,3,7-9)/\partial i$	0.062418
$\partial F(1,1,2)/\partial k$	-0.060543
$\partial F(1,2,4)/\partial k$	-0.020044
$\partial F(1,2,5)/\partial k$	-0.02044

Член b_{ij}^{DH} в выражении (1.20) дается в виде:

$$b_{ij}^{DH} = T_{ij}(N_i^l, N_j^l, N_{ij}^{conj}) \left[\sum_{k \neq i, j} \sum_{l \neq i, j} (1 - \cos^2(\Theta_{ijkl})) f_{ik}^c(r_{ik}) f_{jl}^c(r_{jl}) \right], \quad (1.37)$$

где

$$\cos \Theta_{ijkl} = (\mathbf{e}_{jik}, \mathbf{e}_{ijl}). \quad (1.38)$$

Функция $T_{ij}(N_i^l, N_j^l, N_{ij}^{conj})$ представляется трикубическим сплайном, а функции \mathbf{e}_{jik} и \mathbf{e}_{ijl} – единичные векторы по направлению векторного произведения $\mathbf{R}_{ji} \times \mathbf{R}_{ik}$ и $\mathbf{R}_{ji} \times \mathbf{R}_{jk}$ соответственно, где вектор \mathbf{R} соединяет выбранные атомы. Это уравнение представляет стандартный метод описания сил, возникающих при вращении двугранного угла двойной углеродной связи в аналитическом выражении порядка связи. Значение этой функции равно нулю для плоских систем и равно единице для угла 90° . Далее функция T_{ss} определяет барьер вращения этой связи. Для не двойных углеродных связей эта функция параметризована нулевым значением. Значение T_{ss} для несвязных систем подбирается из ограничения этилена на вращения. Значение T_{ss} для двойных связей для двугранных углов 90° определяется как длина, соответствующая минимальной энергии связи, 1.45 \AA . Это значение выбрано, потому что это есть средняя длина связи для гипотетических структур, теоретически выведенная в работе [43]. Такие структуры как графит содержат атомы с тройными связями, но двугранные углы равны 90° , а не 0° . Значение T_{ss} представлено в таблице 1.7.

Таблица 1.7. Значения функции T_{ss} .

i	j	k	$T(i, j, k)$	Вещество
2	2	1	-0.070280085	Ацетилен
2	2	9	-0.00809675	Твердый углерод

Полная схема выбора параметров, описанная выше, становится значительно проще, если предполагать только взаимодействия ближних соседей. Способ для лучшего определения непрерывной функции, как бы то ни было, затруднителен. Приближение, которое используется здесь, такое же, как использовал Терсофф. Значение $f^c(r)$ определяется следующим выражением:

$$f^c = \begin{cases} 1, & r_{ij} < D_{ij}^{\min} \\ \frac{1}{2} \left(1 + \cos \frac{\pi (r_{ij} - D_{ij}^{\min})}{D_{ij}^{\max} - D_{ij}^{\min}} \right), & D_{ij}^{\min} < r_{ij} < D_{ij}^{\max} \\ 0, & r_{ij} > D_{ij}^{\max} \end{cases}, \quad (1.39)$$

где D_{ij}^{\max} и D_{ij}^{\min} определяют расстояние, на котором функция изменяется от единицы до нуля.

Потенциал Бреннера задаёт поле, в котором находится каждый атом в структуре. Значение силы будет равно минус градиенту общего потенциала:

$$\mathbf{F}_i = -\nabla_i E_{total}. \quad (1.40)$$

Радиус-вектора i -го атома \mathbf{r}_i , расстояние между атомами i и j равно $r_{ij} = |\mathbf{r}_{ij}|$, а

вектор $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$.

$$\nabla_i E_B = \nabla_i f_{ij}^c (V_{ij}^R - b_{ij} V_{ij}^A) + f_{ij}^c (\nabla_i V_{ij}^R - \nabla_i b_{ij} V_{ij}^A - b_{ij} \nabla_i V_{ij}^A). \quad (1.41)$$

Все производные берутся как производные от сложной функции и зависят от

$$\nabla_i r_{ij} = \nabla_i (|\mathbf{r}_{ij}|) = \nabla_i ((\mathbf{r}_i - \mathbf{r}_j)^2)^{\frac{1}{2}} = \frac{1}{2} ((\mathbf{r}_i - \mathbf{r}_j)^2)^{-\frac{1}{2}} \cdot 2 \cdot (\mathbf{r}_i - \mathbf{r}_j) = \frac{\mathbf{r}_{ij}}{r_{ij}}. \quad (1.42)$$

Далее производные не расписываются полностью, а используются предыдущие шаги.

1. Градиент функции усечения:

$$\nabla_i f_{ij}^c = \begin{cases} 0, r_{ij} < D_{ij}^{\min} \\ -\nabla_i r_{ij} \cdot \frac{1}{2} \frac{\pi}{D_{ij}^{\max} - D_{ij}^{\min}} \sin \frac{\pi(r_{ij} - D_{ij}^{\min})}{D_{ij}^{\max} - D_{ij}^{\min}}, D_{ij}^{\min} < r_{ij} < D_{ij}^{\max} \\ 0, r_{ij} > D_{ij}^{\max} \end{cases}. \quad (1.43)$$

2. Градиент от притягивающей и отталкивающей компонент потенциала:

$$\nabla_i V_{ij}^R = -(\alpha + \alpha \frac{Q}{r_{ij}} + \frac{Q}{r_{ij}^2}) A e^{-\alpha r_{ij}} \cdot \nabla_i r_{ij}, \quad (1.44)$$

$$\nabla_i V_{ij}^A = \sum_{n=1}^3 B_n \beta_n e^{-\beta r_{ij}} \cdot \nabla_i r_{ij}. \quad (1.45)$$

3. Градиент от функции связи состоит из нескольких частей:

$$\nabla_i b_{ij} = \frac{1}{2} \nabla_i (b_{ij}^{\sigma-\pi} + b_{ji}^{\sigma-\pi}) + \nabla_i \pi_{ij}^{RC} + \nabla_i b_{ij}^{DH}. \quad (1.46)$$

3.1 Вклад порядка связей:

$$\nabla_i b_{ij}^{\sigma-\pi} = -\frac{1}{2} \left[1 + \sum_{k \neq i, j} f_{ik}^c(r_{ik}) G(\cos(\theta_{ijk})) e^{\lambda_{ijk}} + P_{ij}(N_i^C, N_i^H) \right]^{\frac{3}{2}} \times \left[\sum_{k \neq i, j} \nabla_i f_{ik}^c(r_{ik}) G(\cos(\theta_{ijk})) e^{\lambda_{ijk}} + \sum_{k \neq i, j} f_{ik}^c(r_{ik}) \nabla_i G(\cos(\theta_{ijk})) e^{\lambda_{ijk}} + \sum_{k \neq i, j} f_{ik}^c(r_{ik}) G(\cos(\theta_{ijk})) \nabla_i e^{\lambda_{ijk}} + \nabla_i P_{ij}(N_i^C, N_i^H) \right] \quad (1.47)$$

Угол θ_{ijk} обозначается индексами i, j, k в таком порядке, что сила действует на атом i .

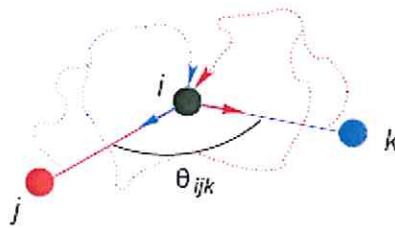


Рисунок 1.7. Расположение атомов, которые образуют угол θ_{ijk}

$$\nabla_i G(\cos(\theta_{ijk})) \approx \sum_{n=1}^5 n a_n (\cos(\theta_{ijk}))^{n-1} \cdot \nabla_i \cos(\theta_{ijk}). \quad (1.48)$$

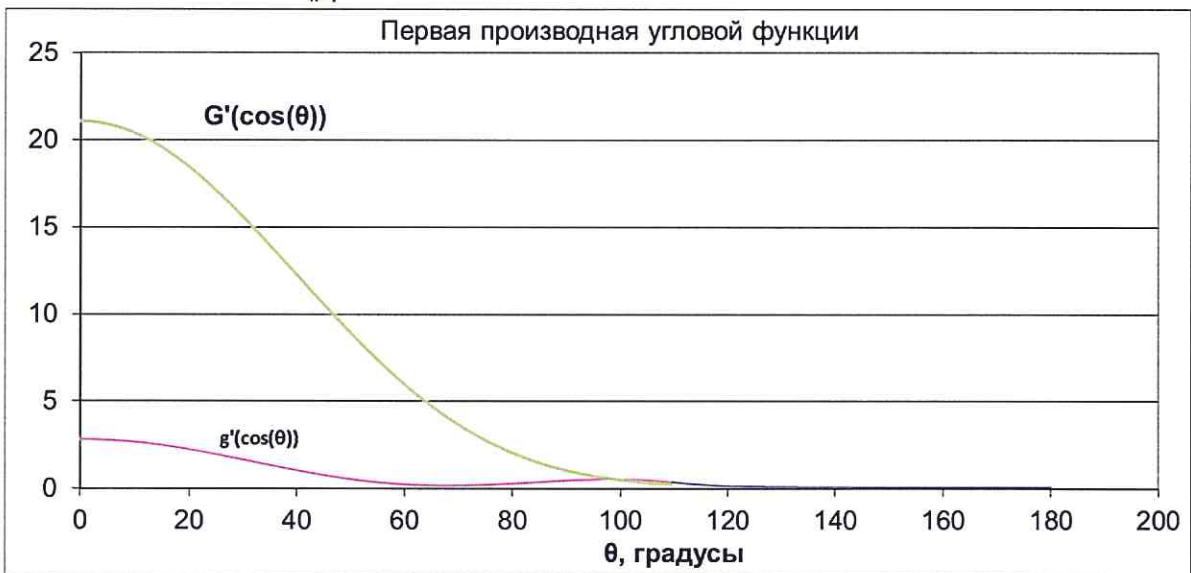


Рисунок 1.8. График первой производной угловой функции $G_c(\cos(\theta))$

Вид угловой функции, представленной в виде тригонометрического сплайна, ближе к реальности, чем экспоненциального сплайна.

$$\nabla_i \cos(\theta_{ijk}) = \nabla_i \frac{(\mathbf{r}_{ik}, \mathbf{r}_{ij})}{r_{ik} r_{ij}} = \frac{\nabla_i (\mathbf{r}_i - \mathbf{r}_k)(\mathbf{r}_i - \mathbf{r}_j)}{r_{ik} r_{jk}} - (\mathbf{r}_{ik}, \mathbf{r}_{jk}) \left(\frac{\nabla_i r_{ik}}{r_{ik}^2 r_{ij}} + \frac{\nabla_i r_{ij}}{r_{ij}^2 r_{ik}} \right) =$$

$$= \frac{\mathbf{r}_{ij} + \mathbf{r}_{ik}}{r_{ik} r_{jk}} - (\mathbf{r}_{ik}, \mathbf{r}_{jk}) \left(\frac{\nabla_i r_{ik}}{r_{ik}^2 r_{ij}} + \frac{\nabla_i r_{ij}}{r_{ij}^2 r_{ik}} \right), \quad (1.49)$$

$$\nabla_i e^{\lambda_{ijk}} = 0, \quad (1.50)$$

$$\nabla_i P_{ij}(N_i^C, N_i^H) = \frac{\partial P_{ij}}{\partial N_i^C} \nabla_i N_i^C + \frac{\partial P_{ij}}{\partial N_i^H} \nabla_i N_i^H, \quad (1.51)$$

$$\nabla_i N_i^C = \sum_{k \neq i, j}^{\text{carbon atom}} \nabla_i f_{ik}^c(r_{ik}), \quad (1.52)$$

$$\nabla_i N_i^H = \sum_{l \neq i, j}^{\text{hydrogen atom}} \nabla_i f_{il}^c(r_{il}). \quad (1.53)$$

3.2 Следует отметить, что $\nabla_i b_{ji}^{\sigma-\pi}$ не получается из $\nabla_i b_{ij}^{\sigma-\pi}$ простой перестановкой индексов:

$$\nabla_i b_{ji}^{\sigma-\pi} = -\frac{1}{2} \left[1 + \sum_{k \neq i, j} f_{jk}^c(r_{jk}) G(\cos(\theta_{jik})) e^{\lambda_{ijk}} + P_{ji}(N_j^C, N_j^H) \right]^{\frac{3}{2}} \times \left[\sum_{k \neq i, j} f_{jk}^c(r_{jk}) \nabla_i G(\cos(\theta_{jik})) e^{\lambda_{jik}} \right], \quad (1.54)$$

$$\nabla_i G(\cos(\theta_{jik})) \approx \sum_{n=1}^5 n a_n (\cos(\theta_{jik}))^n \cdot \nabla_i \cos(\theta_{jik}), \quad (1.55)$$

$$\nabla_i \cos(\theta_{jik}) = \nabla_i \frac{(\mathbf{r}_{jk}, \mathbf{r}_{ji})}{r_{jk} r_{ji}} = \frac{\nabla_i (\mathbf{r}_j - \mathbf{r}_k)(\mathbf{r}_j - \mathbf{r}_i)}{r_{jk} r_{ji}} - (\mathbf{r}_{jk}, \mathbf{r}_{ji}) \frac{\nabla_i r_{ji}}{r_{ji}^2 r_{jk}} =$$

$$= \frac{\mathbf{r}_{jk}}{r_{jk} r_{ji}} - (\mathbf{r}_{jk}, \mathbf{r}_{ji}) \frac{\nabla_i r_{ji}}{r_{ji}^2 r_{jk}}. \quad (1.56)$$

3.3 Вклад уточняющей материал функции:

$$\nabla_i \pi_{ij}^{RC} = \nabla_i F_{ij}(N_i^l, N_j^l, N_{ij}^{conj}) = \frac{\partial F_{ij}}{\partial N_i^l} \nabla_i N_i^l + \frac{\partial F_{ij}}{\partial N_j^l} \nabla_i N_j^l + \frac{\partial F_{ij}}{\partial N_{ij}^{conj}} \nabla_i N_{ij}^{conj}, \quad (1.57)$$

$$\nabla_i N_i^l = \nabla_i N_i^C + \nabla_i N_i^H, \quad (1.58)$$

$$\nabla_i N_j^l = 0, \quad (1.59)$$

$$\nabla_i N_{ij}^{conj} = 2 \left[\sum_{k \neq i, j}^{\text{carbon}} f_{ik}^c(r_{ij}) F(X_{ik}) \right] \left(\sum_{k \neq i, j}^{\text{carbon}} \nabla_i f_{ik}^c(r_{ij}) F(X_{ik}) + f_{ik}^c(r_{ij}) \nabla_i F(X_{ik}) \right), \quad (1.60)$$

$$\nabla_i F(X_{ik}) = \frac{dF}{dX_{ik}} \nabla_i X_{ik} = \frac{dF}{dX_{ik}} \nabla_i (-f_{ik}(r_{ik})), \quad (1.61)$$

$$\frac{dF}{dX_{ij}} = \begin{cases} 0 & x_{ij} < 2 \\ -\pi \sin(2\pi(x_{ij} - 2)) & 2 < x_{ij} < 3 \\ 0 & x_{ij} > 3 \end{cases}. \quad (1.62)$$

3.4 Вклад энергии конформации поворотной изомерии:

$$\begin{aligned} \nabla_i b_{ij}^{DH} = & \nabla_i T_{ij}(N_i^l, N_j^l, N_{ij}^{conj}) \left[\sum_{k \neq i, j} \sum_{l \neq i, j} (1 - \cos^2(\Theta_{ijkl})) f_{ik}^c(r_{ik}) f_{jl}^c(r_{jl}) \right] + \\ & + T_{ij}(N_i^l, N_j^l, N_{ij}^{conj}) \nabla_i \left[\sum_{k \neq i, j} \sum_{l \neq i, j} (1 - \cos^2(\Theta_{ijkl})) f_{ik}^c(r_{ik}) f_{jl}^c(r_{jl}) \right], \end{aligned} \quad (1.63)$$

$$\nabla_i T_{ij}(N_i^l, N_j^l, N_{ij}^{conj}) = \frac{\partial T_{ij}}{\partial N_i^l} \nabla_i N_i^l + \frac{\partial T_{ij}}{\partial N_j^l} \nabla_i N_j^l + \frac{\partial T_{ij}}{\partial N_{ij}^{conj}} \nabla_i N_{ij}^{conj}, \quad (1.64)$$

$$\begin{aligned} \nabla_i \left[\sum_{k \neq i, j} \sum_{l \neq i, j} (1 - \cos^2(\Theta_{ijkl})) f_{ik}^c(r_{ik}) f_{jl}^c(r_{jl}) \right] = & \sum_{k \neq i, j} \sum_{l \neq i, j} (1 - \cos^2(\Theta_{ijkl})) \nabla_i f_{ik}^c(r_{ik}) \nabla_i f_{jl}^c(r_{jl}) + \\ & + \sum_{k \neq i, j} \sum_{l \neq i, j} (1 - \cos^2(\Theta_{ijkl})) f_{ik}^c(r_{ik}) \nabla_i f_{jl}^c(r_{jl}) + \sum_{k \neq i, j} \sum_{l \neq i, j} (-2 \cos(\Theta_{ijkl}) \nabla_i \cos(\Theta_{ijkl})) f_{ik}^c(r_{ik}) f_{jl}^c(r_{jl}) \end{aligned} \quad (1.65)$$

Двугранный угол образуется между плоскостями, которые натянуты на вектора $\mathbf{r}_{ij}, \mathbf{r}_{il}$ и $\mathbf{r}_{jk}, \mathbf{r}_{ji}$.

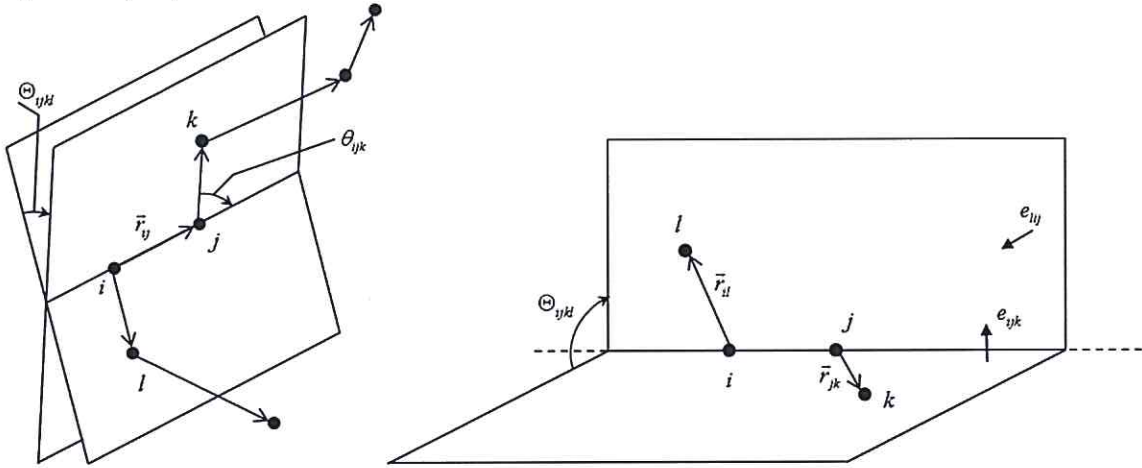


Рисунок 1.9. Двугранный угол Θ_{ijkl}

$$\cos(\Theta_{ijkl}) = \frac{\mathbf{r}_{il} \times \mathbf{r}_{ij}}{r_{il} r_{ij} \sin(\theta_{ijl})} \cdot \frac{\mathbf{r}_{jk} \times \mathbf{r}_{ji}}{r_{jk} r_{ji} \sin(\theta_{jik})} = \left[\frac{(\mathbf{r}_{jk}, \mathbf{r}_{il})}{r_{jk} r_{il}} + \cos(\theta_{ijl}) \cos(\theta_{jik}) \right] \frac{-1}{\sin(\theta_{ijl}) \sin(\theta_{jik})} \quad (1.66)$$

$$\begin{aligned}
\nabla_i \cos(\Theta_{ijkl}) &= \nabla_i \left[\frac{\mathbf{r}_{il} \times \mathbf{r}_{ij}}{r_{il} r_{ij} \sin(\theta_{ijl})} \frac{\mathbf{r}_{jk} \times \mathbf{r}_{ji}}{r_{jk} r_{ji} \sin(\theta_{jik})} \right] = \nabla_i \left[\left(\frac{(\mathbf{r}_{jk}, \mathbf{r}_{il})}{r_{jk} r_{il}} + \cos(\theta_{ijl}) \cos(\theta_{jik}) \right) \frac{-1}{\sin(\theta_{ijl}) \sin(\theta_{jik})} \right] = \\
&= \left[\nabla_i \frac{(\mathbf{r}_{jk}, \mathbf{r}_{il})}{r_{jk} r_{il}} + \nabla_i (\cos(\theta_{ijl}) \cos(\theta_{jik})) \right] \frac{-1}{\sin(\theta_{ijl}) \sin(\theta_{jik})} + \left(\frac{(\mathbf{r}_{jk}, \mathbf{r}_{il})}{r_{jk} r_{il}} + \cos(\theta_{ijl}) \cos(\theta_{jik}) \right) \frac{-\nabla_i}{\sin(\theta_{ijl}) \sin(\theta_{jik})} = \\
&= \left[\frac{\mathbf{r}_{jk}}{r_{jk} r_{il}} - \frac{\nabla_i r_{il}}{r_{jk} r_{il}^2} + \nabla_i \cos(\theta_{ijl}) \cos(\theta_{jik}) + \cos(\theta_{ijl}) \nabla_i \cos(\theta_{jik}) \right] \frac{-1}{\sin(\theta_{ijl}) \sin(\theta_{jik})} + \\
&+ \cos(\Theta_{ijkl}) \left[\frac{\cos(\theta_{ijl})}{\sin(\theta_{ijl})} \nabla_i \cos(\theta_{ijl}) + \frac{\cos(\theta_{jik})}{\sin(\theta_{jik})} \nabla_i \cos(\theta_{jik}) \right]
\end{aligned} \tag{1.67}$$

Здесь также нужно обратить внимание, что градиенты косинусов будут иметь разные выражения для разных углов и не получаются простой заменой индексов.

Представленный потенциал межмолекулярной связи удовлетворяет как твердым веществам, так и молекулярным образованиям. Доказательством валидации потенциала является предсказание новых свойств за пределами базы подбора.

1.5 Методы термостатирования

Статистические величины моделируются в молекулярной динамике в соответствии с некоторым распределением. Физическая система находится в различных состояниях, а вся их совокупность составляет статистический ансамбль. В МД используется несколько видов таких ансамблей.

Ансамбль представляет собой набор всех возможных состояний системы, которые отличаются микроскопическими параметрами, но имеют одно макро- или термодинамическое состояние. Можно выделить основные ансамбли систем.

Микроканонический ансамбль (NVE). Термодинамическое состояние характеризуется заданным набором частиц (N), заданным объёмом (V) и заданной энергией (E). Этот ансамбль соответствует замкнутой системе, в которой нет потерь энергии и количество частиц остается постоянным.

Канонический ансамбль (NVT). Такая система характеризуется заданным набором частиц (N), заданным объёмом (V) и заданной температурой (T). В системе может происходить обмен тепла с термостатом, количество частиц сохраняется.

Изобарически-изотермический ансамбль (NPT). Такая система характеризуется заданным набором частиц (N), заданным давлением (P) и заданной температурой (T). Чтобы поддерживать давление постоянным, изменяется объём системы. Температура поддерживается термостатом.

Канонический ансамбль Гиббса (μVT). Термодинамическое состояние характеризуется заданным химическим потенциалом (μ), заданным объёмом (V) и заданной температурой (T). В такой системе изменяется энергия и количество частиц.

Мгновенное значение температуры системы частиц определяется выражением:

$$\sum_{i=1}^N \frac{|\mathbf{p}_i|^2}{2m_i} = \frac{k_b T}{2} (3N - N_c), \quad (1.68)$$

где N_c – количество ограничений, и тогда $3N - N_c = N_{df}$ определяет полное число степеней свободы, \mathbf{p}_i – импульс i -ой частицы. Средняя температура $\langle T \rangle$ равна макроскопической температуре.

Обычно в молекулярно-динамическом моделировании применяют NVE ансамбль. В таком случае момент p и угловой момент L сохраняются. Но при использовании граничных условий угловой момент не сохраняется.

Микроканонический ансамбль (NVT) в большинстве случаев не удовлетворяет лабораторным экспериментам. Но этот метод необходим при численном моделировании систем при заданной температуре. Также метод корректирует численные ошибки моделирования.

Простым способом изменить температуру является масштабирование скоростей. Если скорости в момент времени t умножить на коэффициент λ , то температура $T(t)$ изменится:

$$\Delta T = \frac{1}{2} \sum_{i=1}^N 2 \frac{m_i (\lambda \mathbf{v}_i)^2}{N_{df} k_b} - \frac{1}{2} \sum_{i=1}^N 2 \frac{m_i (\mathbf{v}_i)^2}{N_{df} k_b}, \quad (1.69)$$

$$\Delta T = (\lambda^2 - 1) T(t), \quad (1.70)$$

$$\lambda = \sqrt{\frac{T_0}{T(t)}}, \quad (1.71)$$

где $T(t)$ – текущая температура, вычисленная из кинетической энергии, T_0 – установленная температура.

Таким образом, умножая скорости на коэффициент $\lambda = \sqrt{\frac{T_0}{T(t)}}$ на каждом шаге моделирования, контролируется температура. В этом подходе возникает проблема в том, что невозможны флуктуации температуры, которые характерны для канонического ансамбля.

Более физичный подход предложен в работе [44]. Система погружается во внешнюю ванну с заданной температурой T_0 . Скорости масштабируются на каждом шаге, при этом температура меняется с заданной скоростью:

$$\frac{dT(t)}{dt} = \frac{1}{\tau}(T_0 - T(t)), \quad (1.72)$$

где τ связывающий параметр, который определяет степень погружения системы в ванну. Это даёт экспоненциальный переход системы к заданной температуре. Изменение температуры на каждом шаге будет:

$$\Delta T = \frac{\Delta t}{\tau}(T_0 - T(t)). \quad (1.73)$$

Тогда коэффициент масштабирования скоростей будет:

$$\lambda^2 = 1 + \frac{\Delta t}{\tau} \left(\frac{T_0}{T(t)} - 1 \right). \quad (1.74)$$

На практике связывающий параметр τ выбирается эмпирическим путем. При $\tau \rightarrow \infty$ термостат Берендсена не действует, и выполняется микроканонический ансамбль NVE. В таком случае температурные флуктуации не достигают заданного уровня канонического ансамбля NVT. С другой стороны маленькие значения τ приведут к нефизично малым флуктуациям температуры. При выборе параметра τ равном Δt термостат Берендсена просто масштабирует скорости. Обычно используется $\tau \approx 1 ps$ для конденсированных систем.

Масштабирование скоростей может быть выражено:

$$\langle \mathbf{v} \rangle(t + \Delta t) = \mathbf{v}(t + \Delta t) \sqrt{\omega + (1 - \omega) \frac{T_0}{T(t)}}, \quad (1.75)$$

такое же выражение следует использовать для ускорений:

$$\langle \mathbf{a} \rangle(t + \Delta t) = \mathbf{a}(t + \Delta t) \sqrt{\omega + (1 - \omega) \frac{T_0}{T(t)}}, \quad (1.76)$$

где ω – параметр термостата, $0 \leq \omega \leq 1$. При уменьшении параметра ω уменьшаются флуктуации температуры. При $\omega = 0$ температура будет приближаться к заданной температуре термостата, при $\omega = 1$ термостат действовать на систему не будет. Параметр термостата выражается через время реакции термостата t_r :

$$\omega = 1 - e^{-\frac{t_r}{\Delta t}}. \quad (1.77)$$

Термостат обычно используется в начале моделирования для приведения системы в равновесие потенциальной и кинетической энергии. Рекомендуется увеличивать время реакции термостата от 0.5 фемтосекунд до 100 фемтосекунд по мере приближения системы к равновесию. При использовании термостата ансамбль не является каноническим, но температура поддерживается около заданной T_0 .

Термостат Берендсена эффективен для приведения системы к заданной температуре, но если система достигла равновесия, более важным является канонический ансамбль. Необходимо использовать другой метод, интегрировать тепловую ванну в саму систему добавлением искусственной переменной \tilde{s} , ассоциированной с массой $Q > 0$ и скоростью $\dot{\tilde{s}}$. Величина Q определяет связанность между резервуаром и системой и влияет на температурные флуктуации. Искусственная переменная \tilde{s} играет роль параметра, масштабирующего время:

$$d\tilde{t} = \tilde{s} dt. \quad (1.78)$$

Координаты в новой системе даются прежними выражениями:

$$\tilde{\mathbf{r}} = \mathbf{r}, \dot{\tilde{\mathbf{r}}} = \tilde{s}^{-1} \dot{\mathbf{r}}, \tilde{s} = s, \dot{\tilde{s}} = \tilde{s}^{-1} \dot{s}. \quad (1.79)$$

Лагранжиан расширенной системы определяется выражением:

$$L = \sum_i \frac{m_i}{2} \tilde{s}^2 \dot{\tilde{\mathbf{r}}}_i^2 - U(\tilde{\mathbf{r}}) + \frac{1}{2} Q \dot{\tilde{s}}^2 - g k_b T_0 \ln \tilde{s}. \quad (1.80)$$

Первые два члена Лагранжиана отражают разницу кинетической и потенциальной энергий системы. Дополнительные члены отражают кинетическую и потенциальную энергию искусственной добавки, которая выбирается для приведения системы к каноническому ансамблю, где $g = N_{df}$ для формализма Нозе-Хувера [45-48] и $g = N_{df} + 1$ для формализма Нозе. Уравнения движения запишутся в следующем виде:

$$\ddot{\tilde{\mathbf{r}}}_i = \frac{\tilde{\mathbf{F}}_i}{m_i \tilde{s}^2} - \frac{2\dot{\tilde{s}}\dot{\tilde{\mathbf{r}}}_i}{\tilde{s}}, \quad (1.81)$$

$$\ddot{\tilde{s}} = \frac{1}{Q\tilde{s}} \left(\sum_i m_i \tilde{s}^2 \dot{\tilde{\mathbf{r}}}_i^2 - gk_b T_0 \right). \quad (1.82)$$

Эти выражения отражают канонический ансамбль с расширенными параметрами $(\tilde{r}, \tilde{p}, \tilde{t})$. Однако энергия системы не сохраняется. Вместе с флуктуацией \tilde{s} появляется перенос тепла между системой и тепловой ванной. Это отражается в уравнениях движения, которые описывают канонический ансамбль.

Уравнения Нозе детерминированные и обратимые по времени. Но при эволюции \tilde{s} , которая задается уравнением второго порядка, нагрев может привести к периодическим колебаниям температуры. Уравнения движения могут быть переписаны в терминах реальной системы:

$$s = \tilde{s}, \dot{s} = \tilde{s}\dot{\tilde{s}}, \ddot{s} = \tilde{s}^2\ddot{\tilde{s}} + \tilde{s}\dot{\tilde{s}}^2, \quad (1.83)$$

$$\mathbf{r} = \tilde{\mathbf{r}}, \dot{\mathbf{r}} = \tilde{s}\dot{\tilde{\mathbf{r}}}, \ddot{\mathbf{r}} = \tilde{s}^2\ddot{\tilde{\mathbf{r}}} + \tilde{s}\dot{\tilde{\mathbf{r}}}^2, \quad (1.84)$$

производя замену:

$$\gamma = \frac{\dot{s}}{s}. \quad (1.85)$$

Тогда уравнения движения системы можно записать:

$$\ddot{\mathbf{r}}_i = \frac{\mathbf{F}_i}{m_i} - \gamma\dot{\mathbf{r}}_i, \quad (1.86)$$

$$\dot{\gamma} = -\frac{k_B N_{df}}{Q} T(t) \left(\frac{q}{N_{df} T(t)} - 1 \right), \quad (1.87)$$

где

$$T(t) = \sum_{i=1}^N \frac{m_i \dot{\mathbf{r}}_i^2}{N_{df} k_B} \quad (1.88)$$

и $N_{df} = 3N$.

В алгоритме термостатирования необходимо правильно выбрать фиктивные массы Q и энергию расширенной системы E_e . С одной стороны, слишком большие значения Q могут приводить к потере контроля температуры. Хотя любое конечное значение массы гарантирует канонический ансамбль, если Q большое, каноническое распределение будет достигаться через очень долгое время моделирования. С другой стороны, слишком малые значения Q могут привести к высокочастотным

температурным колебаниями. Переменная \tilde{s} может осциллировать с большой частотой и изменять количество физических степеней свободы.

Более подходящий выбор параметра связи системы может быть сделан как:

$$\dot{\gamma} = -\frac{1}{\tau_{NH}^2} \frac{T(t)}{T_0} \left(\frac{q}{N_{df}} \frac{T_0}{T(t)} - 1 \right), \quad (1.89)$$

с эффективным временем релаксации:

$$\tau_{NH}^2 = \frac{Q}{N_{df} k_B T_0}. \quad (1.90)$$

Время релаксации можно оценить по частоте колебаний среднего $\langle \tilde{s} \rangle$ при отклонении на $\Delta \tilde{s}$.

Стандартный метод интегрирования уравнений Нозе-Хувера с применением схемы скоростей Верле состоит из обратимых по времени выражений:

$$\mathbf{v}(t + \frac{\Delta t}{2}) = \mathbf{v}(t) + \frac{\Delta t}{2} (\mathbf{a}(t) - \gamma(t) \mathbf{v}(t)), \quad (1.91)$$

$$\gamma(t + \frac{\Delta t}{2}) = \gamma(t) - \frac{\Delta t}{2} \frac{1}{\tau_{NH}^2} \frac{T(t)}{T_0} \left(\frac{q}{N_{df}} \frac{T_0}{T(t)} - 1 \right),$$

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \Delta t \cdot \mathbf{v}(t + \frac{1}{2} \Delta t),$$

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t + \frac{\Delta t}{2}) + \frac{\Delta t}{2} (\mathbf{a}(t) - \gamma(t + \frac{\Delta t}{2}) \mathbf{v}(t + \frac{\Delta t}{2})),$$

$$\gamma(t + \Delta t) = \gamma(t + \frac{\Delta t}{2}) - \frac{\Delta t}{2} \frac{1}{\tau_{NH}^2} \frac{T(t + \Delta t)}{T_0} \left(\frac{q}{N_{df}} \frac{T_0}{T(t + \Delta t)} - 1 \right).$$

Алгоритм использует явную схему интегрирования и легко реализуется на графических процессорах.

Термостат Ланжевена предполагает, что в систему помещено множество фантомных частиц. В этом случае система описывается уравнениями Ланжевена, в которые включены дополнительные члены трения, шума:

$$m_i \mathbf{a}_i = m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i - \frac{1}{\tau_d} m \mathbf{v}_i + \sqrt{\frac{2k_B T m_i}{\tau_d}} \mathbf{R}_i(t), \forall i = 1..N_{at}, \quad (1.92)$$

где $k_B T$ определяет тепловую энергию системы, τ_d – время затухания, $\mathbf{R}_i(t)$ – определяет гауссов процесс с нулевым средним значением, такой что:

$$\begin{aligned} \langle \mathbf{R}_i(t) \rangle &= 0 \\ \langle \mathbf{R}_i(t) \mathbf{R}_i(t') \rangle &= \delta(t - t') \end{aligned} \quad (1.93)$$

Уравнения Ланжевена дают корректное физическое описание многочастичным взаимодействиям с тепловой ванной и поддерживают заданную температуру T . Параметр τ_d задаёт время обмена тепла между системой и ванной. Если задать малое значение τ_d , то броуновская динамика будет доминировать над ньютоновской, что приведёт к неверным результатам. Если задать большое значение τ_d , то система не придёт в равновесие.

1.6 Геометрическое построение углеродных наноструктур

Нанотрубки представляют собой собранные в цилиндрическую конфигурацию атомы углерода. Цилиндр может быть закручен вдоль своей оси, тогда говорят о хиральности нанотрубки и обозначают двумя числами (n, m) . В зависимости от этих чисел нанотрубку называют «кресло», «зигзаг», «хиральная». Симметрия и электрические свойства одностенной нанотрубки полностью описываются этими двумя числами [49].

Нанотрубка представляет собой свёрнутую графеновую плоскость. Эта плоскость состоит из шестиугольников, в вершинах которых находятся атомы углерода. Диаметр атома углерода 0.91Å , расстояние между атомами углерода $a_{cc} = 1.42\text{Å}$ в графеновой плоскости или 1.44Å в нанотрубке. В зависимости от количества связей, которые образует углерод с соседним атомом углерода, расстояние может быть 1.44Å – одинарная связь, 1.42Å – полуторная связь, 1.4Å – двойная связь, 1.36Å – тройная связь, 1.32Å – четверная связь. При скручивании плоскости шестиугольники могут смещаться по спирали, что дает различные варианты нанотрубок. Они будут различаться по геометрической форме и диаметру.

В качестве базовых векторов [50] выбираются \mathbf{a}_1 и \mathbf{a}_2 , рис. 1.10:

$$\mathbf{a}_1 = x \frac{\sqrt{3}}{2} \mathbf{a} + y \frac{1}{2} \mathbf{a}, \quad (1.94)$$

$$\mathbf{a}_2 = x \frac{\sqrt{3}}{2} \mathbf{a} - y \frac{1}{2} \mathbf{a}, \quad (1.95)$$

где x и y – единичные векторы по осям X и Y . Длины векторов обозначены:

$$a = |\mathbf{a}_1| = |\mathbf{a}_2| = \sqrt{3} a_{cc}. \quad (1.96)$$

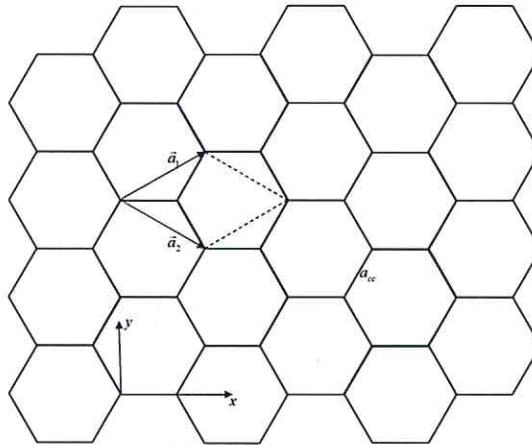


Рисунок 1.10. Графеновая плоскость

Все точки плоскости могут быть выражены через базисные вектора комбинацией $na_1 + ma_2$, где n и m – целые числа.

При скручивании определяется хиральный вектор $\vec{OD} = \vec{C}_h$ [50]. Однослойные нанотрубки сворачиваются таким образом, что начало вектора т. О и конец т. D совпадают. Он может быть выражен через базисные векторы:

$$\mathbf{C}_h = na_1 + ma_2 \equiv (n, m), \text{ где } -\frac{n}{2} < m \leq n, n > 0, \quad (1.97)$$

где n и m – целые числа. (n, m) – краткая запись хирального вектора. Следует отметить, что базисные векторы a_1 и a_2 не ортогональны. Если $m=n$, такие нанотрубки называются «armchair». Если $m=0$, такие нанотрубки называются «zigzag», так как края одного кольца шестиугольников образуют зигзаг. Для хирального вектора, учитывая $L^2 = (\mathbf{C}_h, \mathbf{C}_h)$, определяется:

$$L = |\mathbf{C}_h| = (n^2 + m^2 + nm)^{\frac{1}{2}} a_{cc}. \quad (1.98)$$

При вычислениях используется, что $(a_1, a_1) = (a_2, a_2) = a^2$ и $(a_1, a_2) = \frac{a^2}{2}$. Для нанотрубок $a=2.49 \text{ \AA}$, если $a_{cc} = 1.44 \text{ \AA}$.

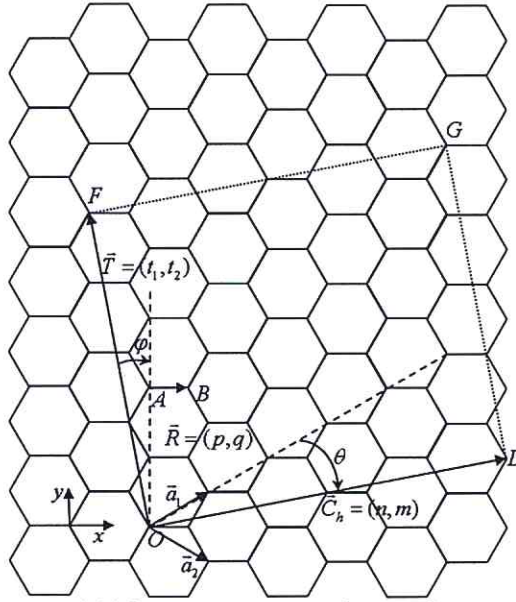


Рисунок 1.11. Закручивание графеновой плоскости

Так как нанотрубка закручивается по хиральному вектору, то диаметр равен длине окружности (длина хирального вектора) поделённой на π :

$$d_t = \frac{L}{\pi}. \quad (1.99)$$

Угол хиральности θ определяется как угол между базисным вектором a_1 и хиральным вектором \bar{C}_h (рис. 1.11). Хиральный угол положительный, если откладывается по часовой стрелке от базисного вектора a_1 , и отрицательный, если откладывается против часовой стрелки. Вычисляется угол по теореме косинусов:

$$(C_h, a_1) = |C_h| |a_1| \cos \theta, \quad (1.100)$$

$$\theta = \text{sgn}(m) \arccos\left(\frac{2n+m}{2\sqrt{n^2+m^2+nm}}\right) = \arctan\left(\frac{\sqrt{3}m}{2n+m}\right),$$

где

$$\text{sgn}(m) = \begin{cases} 1, m > 0 \\ -1, m < 0 \end{cases}. \quad (1.101)$$

Угол хиральности может находиться в диапазоне $-\frac{\pi}{6} < \theta < \frac{\pi}{6}$. Для нанотрубок типа «armchair» угол равен $\frac{\pi}{6}$, для «zigzag» – 0.

Можно выбирать начало базисных векторов в центре шестиугольника, а можно – в одном из узлов решетки. Повороты во всех секторах будут эквивалентны поворотам точек и векторов в первом секторе.

Моделирование нанотрубки. Создание координат шестиугольника.

Координаты одной точки шестиугольника в декартовом трехмерном пространстве – $(a_{cc}, 0, 0)$ (рис. 1.12).

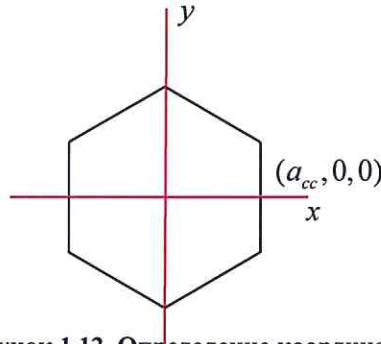


Рисунок 1.12. Определение координат

Для получения следующей пары используется матрица поворота предыдущих координат:

$$\begin{pmatrix} \cos \frac{\pi}{3} & -\sin \frac{\pi}{3} \\ \sin \frac{\pi}{3} & \cos \frac{\pi}{3} \end{pmatrix}. \tag{1.102}$$

Координаты точек получаются:

$$\begin{pmatrix} \cos \frac{\pi}{3} & -\sin \frac{\pi}{3} \\ \sin \frac{\pi}{3} & \cos \frac{\pi}{3} \end{pmatrix} (a, 0)^T = \begin{pmatrix} a \cos \frac{\pi}{3} \\ a \sin \frac{\pi}{3} \end{pmatrix} = \begin{pmatrix} \frac{a}{2} \\ \frac{\sqrt{3}a}{2} \end{pmatrix}, \tag{1.103}$$

$$\begin{pmatrix} \cos \frac{\pi}{3} & -\sin \frac{\pi}{3} \\ \sin \frac{\pi}{3} & \cos \frac{\pi}{3} \end{pmatrix} \begin{pmatrix} a \cos \frac{\pi}{3} \\ a \sin \frac{\pi}{3} \end{pmatrix} = \begin{pmatrix} a \cos^2 \frac{\pi}{3} - a \sin^2 \frac{\pi}{3} \\ a \cos \frac{\pi}{3} \sin \frac{\pi}{3} - a \sin \frac{\pi}{3} \cos \frac{\pi}{3} \end{pmatrix} = \begin{pmatrix} a \cos \frac{2\pi}{3} \\ a \sin \frac{2\pi}{3} \end{pmatrix} = \begin{pmatrix} -\frac{1}{2}a \\ \frac{\sqrt{3}}{2}a \end{pmatrix}, \tag{1.104}$$

$$\begin{pmatrix} \cos \frac{\pi}{3} & -\sin \frac{\pi}{3} \\ \sin \frac{\pi}{3} & \cos \frac{\pi}{3} \end{pmatrix} \begin{pmatrix} a \cos \frac{2\pi}{3} \\ a \sin \frac{2\pi}{3} \end{pmatrix} = \begin{pmatrix} a \cos \frac{2\pi}{3} \cos \frac{\pi}{3} - a \sin \frac{2\pi}{3} \sin \frac{\pi}{3} \\ a \cos \frac{2\pi}{3} \sin \frac{\pi}{3} + a \cos \frac{2\pi}{3} \sin \frac{\pi}{3} \end{pmatrix} = \begin{pmatrix} a \cos \frac{3\pi}{3} \\ a \sin \frac{3\pi}{3} \end{pmatrix} = \begin{pmatrix} -a \\ 0 \end{pmatrix}, \tag{1.105}$$

Далее выполняется повторение шестиугольников на заданную длину и ширину.

Так как плоскость можно составить повторением лишь четырех атомов, то они повторяются на заданную длину трубки через тройной интервал C-C.

Матрица повторения по x:

$$\begin{pmatrix} 1 & 3a \\ 0 & 1 \end{pmatrix}. \quad (1.106)$$

Повторение атомов по оси y такое же, как и по оси OX, но с интервалом $\sqrt{3}a$.

Матрица повторения по оси OY:

$$\begin{pmatrix} 1 & 0 \\ \sqrt{3}a & 1 \end{pmatrix}. \quad (1.107)$$

Получили множество точек графенового листа. Это множество нужно повернуть на угол хиральности. Хиральность нанотрубки задается двумя параметра (n, m).

Матрица поворота плоскости при хиральности:

$$\begin{pmatrix} \frac{vx}{vlen} & \frac{vy}{vlen} \\ -\frac{vy}{vlen} & \frac{vx}{vlen} \end{pmatrix} \text{ где } vx = \frac{(n+m)3a}{2}, vy = \frac{(n-m)\sqrt{3}a}{2}, vlen = \sqrt{vx^2 + vy^2}. \quad (1.108)$$

После поворота плоскости происходит ее закручивание в трубку, например, вдоль оси OY.

Матрица преобразования координат из плоскости в закрученную спираль (матрица Якоби для преобразования координат):

$$\begin{pmatrix} R \cos \frac{x}{R} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & R \sin \frac{x}{R} \end{pmatrix}, \quad (1.109)$$

где R – радиус спирали, $vlen = \sqrt{vx^2 + vy^2}$ – длина окружности, следовательно $R = \frac{vlen}{2\pi}$.

Построение фуллеренов.

Топология молекулы C_{60} представляет усеченный икосаэдр, где каждый из 60 атомов связан с тремя соседними, что создаёт набор трехвалентных связей. В отличие от графита фуллерены содержат пятиугольники вдоль граней шестиугольников. Тогда

в соответствии с теоремой Эйлера сферических многогранников для закрытой геометрии число вершин v связано с числом рёбер e и числом граней f :

$$v + f = e + 2, \quad (1.110)$$

где v равно числу атомов, e равно числу связей. Можно вычислить число граней как

$$f = \frac{5}{6}p + h + 2, \text{ где } p \text{ равно числу пятиугольников, } h \text{ равно числу шестиугольников в}$$

фуллерене. Для закрытой структуры из v атомов и трёхвалентными связями выполняется соотношение: количество пятиугольников – 12, шестиугольников – $\frac{v}{2} - 10$. Для молекулы C_{60} получается 90 связей углерод-углерод, 12 пятиугольников и 20 шестиугольников.

Распределение 12 пятиугольников в молекуле C_{60} приводит к 1812 изомерам. Только один изомер, в котором пятиугольники равномерно распределены между шестиугольниками, энергетически стабилен. Это I_h -симметрия известна под названием бакминстерский фуллерен, и это единственный стабильный C_{60} фуллерен. Фуллерены с таким пяти-, шестиугольным строением подчиняются правилу изолированных пятиугольников (IRP). Однако известны заряженные фуллерены, для которых стабильная форма может не подчиняться этому правилу.

Так как в C_{60} есть пятиугольники, то молекула обладает двумя типами связей. Эти связи состоят из шестидесяти связей пятиугольник-шестиугольник длиной 1.45-1.49Å, и тридцати связей шестиугольник-шестиугольник длиной 1.38-1.40Å. Связи пятиугольник-шестиугольник формируются рёбрами пятиугольника и шестиугольника, связи шестиугольник-шестиугольник формируются рёбрами шестиугольника и шестиугольника. Эти связи являются в первом случае одинарными, во втором – двойными. Молекула C_{60} содержит 240 валентных электронов, которые находятся на 60 π -орбиталях, и имеет 180 степеней свободы, которые состоят из 6 степеней перемещения и вращений и 174 степеней колебаний, и включает 46 собственных частот. Свойства образования, строения, стабильности, энергии абсорбции и перераспределения за 25 лет интенсивных исследований полностью до сих пор не изучены.

Выводы к главе 1

В главе 1 дана постановка задачи молекулярно-динамического моделирования на графических процессорах. Графические карты имеют собственную архитектуру вычислителей, которые объединяются в блоки и grids. Изначально они были рассчитаны на обработку видео и вычисления с матрицами. Для выполнения задач общего назначения требуется адаптация математических моделей под многопоточную систему исполнения и разноуровневую систему памяти. Для этого были рассмотрены возможности GPU и возникающие к ним требования при решении задачи молекулярной динамики. В молекулярной динамике наиболее сложной вычислительной операцией является расчёт сил, действующих на атом. Для этого выбирается потенциал многочастичного взаимодействия, который учитывает необходимые эффекты. Простые потенциалы хорошо подходят для вычисления на графических процессорах, но такие потенциалы описывают ограниченный набор свойств наноструктур. Для реализации сложных многочастичных потенциалов необходимо разбить процедуру вычисления на отдельные части таким образом, чтобы задействовать параллельные потоки вычисления и аппаратные функции.

Глава 2. Программирование алгоритмов молекулярно-динамического моделирования наносистем на графических процессорах

2.1 Принципы разработки программного обеспечения

2.1.1 Отображение расчётной области на структуру памяти GPU: гибридная модель

Особенности доступа к памяти при программировании на GPU показаны на рис.

2.1.

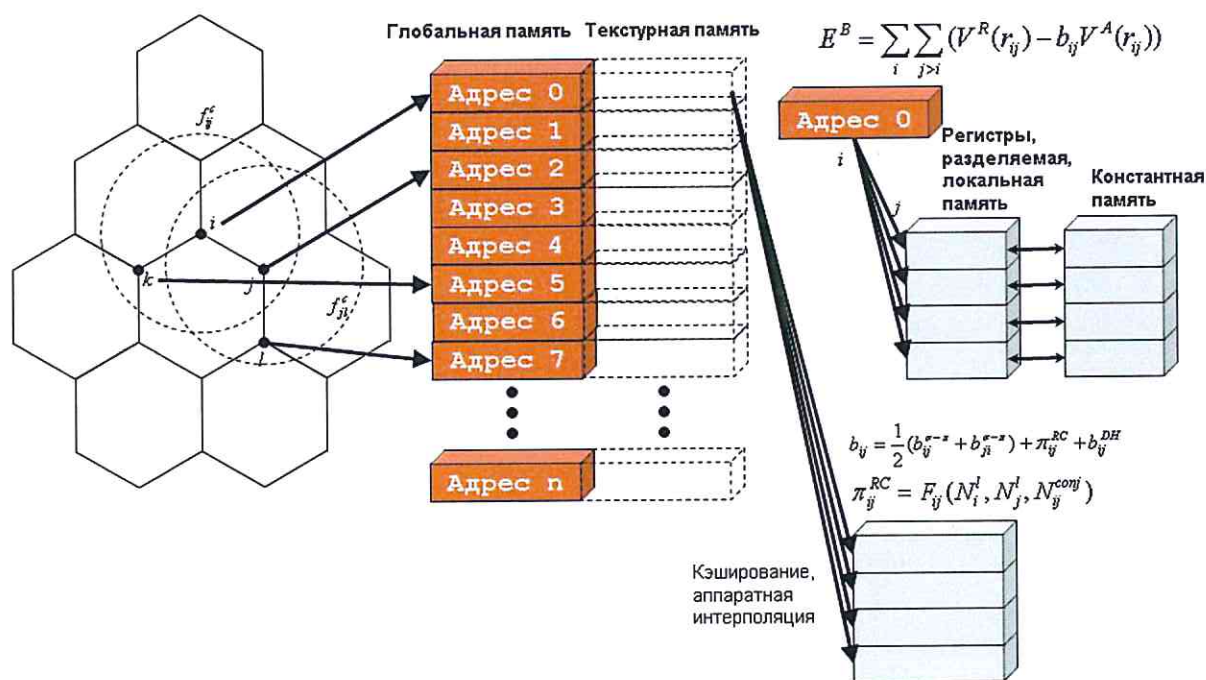


Рисунок 2.1. Схема использования памяти GPU при вычислениях сил в потенциальном поле

Для определения действующей на атом результирующей силы, вообще говоря, необходимо рассчитать вклады, вносимые всеми атомами системы. Однако структура большинства потенциалов межчастичного взаимодействия позволяет учитывать не все атомы, а лишь соседние по отношению к рассматриваемому атому. Таким образом, у атома возникает определенный набор соседей, для которых необходимо провести вычисление сил. Здесь можно выделить два основных способа [51]: составлять список соседей или разбить пространство на ячейки. В первом случае с использованием ограничивающих сфер выбираются все соседние частицы, которые вносят вклад в

результатирующую силу. При этом необходимо обновлять данный список на каждом шаге и для каждого атома.

Согласно второму способу пространство разбивается на ячейки, определяемые ограничивающей функцией (рис. 2.2). В этом случае учитываются дополнительные атомы, которые практически не вносят вклад в результирующую силу. Однако данный подход обладает важным достоинством – конфигурация ячеек известна заранее и не изменяется в процессе моделирования.

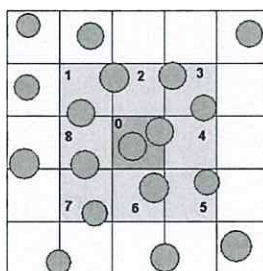


Рисунок 2.2. Ячейчатая модель расчётной области, соседние ячейки пронумерованы одним из способов

Составление списка по всему набору частиц требует $O(N^2)$ операций, но применение ячейчатой модели уменьшает вычислительную сложность до $O(N)$. На реализацию такой модели требуется дополнительная память для хранения данных ячеек. Это накладывает основное ограничение на их количество ячеек, так как объём равен произведению. После размещения атомов в ячейках пространства достаточно проверить локальную окрестность ячейки i -ой частицы, чтобы выделить необходимые j -ые частицы для расчёта потенциала. В выражении (1.18) такую выборку необходимо проделывать множество раз, что приводит к перекрёстным запросам из массива координат и замедлению.

После выполнения такой операции у каждого атома есть локальный набор в дополнительной памяти, который не приводит к перекрёстным запросам.

Применительно к распараллеливанию на графических процессорах более эффективным оказывается использование гибридной модели расчётной области. При этом для частиц, находящихся в некоторой ячейке, можно учитывать только взаимодействие с частицами из соседних ячеек. Расчётная область представляется ячейками, атомы размещаются в ячейках. Проводится дополнительная процедура, в которой для i -ой частицы проверяется локальная окрестность и составляется список

соседних атомов. В дальнейших процедурах используются отдельные списки соседних атомов для вычисления выражений сил взаимодействия.

Совместив два подхода для выбора соседних атомов, удалось достичь сложности вычислений $O(N)$ и избавиться от перекрёстных запросов памяти.

В случае сложного потенциала необходимо рассчитывать угловые и силы кручения. Для этого необходимо зафиксировать атом, его соседний атом и проверять ближайших соседей, которые удовлетворяют формуле плоского и двугранного угла. Это приводит к считыванию данных по одному и тому же адресу несколько раз, что даёт коллизии доступа к памяти.

Алгоритм изменён так, что в первом проходе по ячейкам фиксируется атом и составляется список его соседей:

```
makeList<<< numBlocks, numThreads >>>((float4*)sortedPos,  
                                         cellStart,  
                                         cellEnd,  
                                         numParticles,  
                                         list,  
                                         (float4*)listPos);
```

где *numBlocks*, *numTreads* – количество вычислительных блоков и потоков, *sortedPos* – массив отсортированных радиус-векторов атомов, *cellStart* – массив начальных атомов в ячейках, *cellEnd* – массив конечных атомов в ячейках, *numParticles* – количество атомов, *list* – массив номеров ближайших соседей, *listPost* – массив радиус-векторов ближайших соседей.

Гибридная модель позволяет эффективно представлять пространство моделирования за счёт сочетания ячеистой модели, так как не нужно делать цикл по всем атомам, и техники списков соседей.

На втором этапе используются составленные списки ближайших атомов для вычисления угловых сил и сил кручения:

```
coeff<<< numBlocks, numThreads >>>((float4*)sortedPos,  
                                     numParticles,  
                                     (float4*)bondCoeff,  
                                     (float4*)listPos,  
                                     list);
```

где *bondCoeff* – массив коэффициентов связи в потенциале взаимодействия.

Следует отметить, что максимальный размер списка ближайших соседей равен 8, по числу валентных связей атома. Таким образом, размеры массива можно задать

перед исполнением, чего требует компилятор видеокарты. У каждого атома есть список номеров ближайших атомов и их координаты, что разрешает коллизии памяти и допускает параллельное исполнение мультипотокков.

Для определения соответствия частицы ячейке трёхмерные координаты преобразуются хеш-функцией:

$$hash_i = \left[\frac{z_i}{cellSize.z} \right] \cdot Sy \cdot Sx + \left[\frac{y_i}{cellSize.y} \right] \cdot Sx + \left[\frac{x_i}{cellSize.x} \right], \quad (2.1)$$

где x_i, y_i, z_i компоненты радиус-вектора $\mathbf{r}_i = \mathbf{r}_i(x, y, z)$ i -ой частицы, Sx, Sy – количество ячеек, $cellSize.x, cellSize.y, cellSize.z$ – размеры ячеек (рис. 2.3).

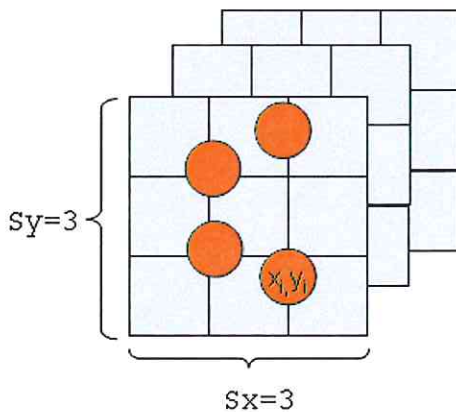


Рисунок 2.3. Иллюстрация ячеистой модели

Хеш-функция является простой, что позволяет ей быстро выполняться на графических процессорах, и достаточной, чтобы определить близкое расположение атомов в линейном массиве; координаты атомов в трёхмерном пространстве преобразуются в линейное расположение. Размер ячейки выбирается по размеру радиуса усечения. Таким образом, для вычисления потенциала требуется проверить лишь соседние ячейки ячеистой модели или соседние диапазоны в массиве данных, отсортированных согласно хеш-значениям.

2.1.2 Реализация потенциала Бреннера второй модификации на GPU

Потенциал межчастичного взаимодействия полностью реализуется на графической карте.

На шаге расчёта потенциала выполняется операция:

$$E^B = \sum_i \sum_{j>i} (V^R(r_{ij}) - b_{ij} V^A(r_{ij})). \quad (2.2)$$

Вызов основного кода выполняется следующей строкой:


```

BrenerKernel<<<numBlocks, numThreads>>>( (float4*)sortedPos,
                                           (float4*)sortedVel,
                                           (float4*)newVel,
                                           (float4*)newAcel,
                                           gridParticleIndex,
                                           numParticles,
                                           (float4*)bondCoeff,
                                           deltaTime,
                                           list,
                                           (float4*)listPos,
                                           iteration,
                                           dNose);

```

где *numBlocks*, *numTreads* – количество вычислительных блоков и потоков, *sortedPos* – массив сортированных радиус-векторов атомов, *newVel*, *newAcel* – массивы вычисляемых на данном шаге скоростей и ускорений, *gridParticleIndex* – массив индексов частиц после сортировки, *numParticles* – количество атомов, *bondCoeff* – массив коэффициентов связи в потенциале взаимодействия, *deltaTime* – шаг по времени, *list* – массив номеров ближайших соседей, *listPost* – массив радиус векторов ближайших соседей, *iteration* – номер шага по времени, *dNose* – параметр термостата Нозе-Хувера.

Подготовленная система хранения координат позволяет записывать выражения, входящие в потенциал, в виде математических формул.

2.1.3 Интегрирование и представление разностной схемы

На шаге интегрирования выполняется операция:

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta t \cdot \mathbf{v}_i(t) + \frac{1}{2} \Delta t^2 \mathbf{a}_i(t). \quad (2.3)$$

Данные, необходимые для одного шага одного вычислительного потока, располагаются в глобальной памяти видеокарты (DRAM), так как нужно хранить большой объём координат, скорости, ускорений всех атомов. Параметры же потенциала копируются в константную память (`__constant__`) каждого вычислительного блока (`warp`).

В программе интегрирование запускается следующей строчкой:

```

thrust::for_each(
    thrust::make_zip_iterator(thrust::make_tuple(d_pos4, d_vel4, d_acel4)),
    thrust::make_zip_iterator(thrust::make_tuple(d_pos4+numParticles,
    d_vel4+numParticles, d_acel4+numParticles)),
    integrate_functor(deltaTime));

```

где *deltaTime* – шаг интегрирования по времени, *d_pos4* – массив координат атомов, *d_vel4* – массив скоростей атомов, *d_acel4* – массив ускорений атомов, *numParticles* – количество атомов.

Интегрирование осуществляется итератором `thrust`, который автоматически определяет необходимое количество потоков вычисления (`thread`), ресурсы памяти и оптимальные параметры запуска. Вызов сторонней библиотеки параллельного программирования даёт возможность пользоваться постоянно развивающимися прикладными модулями.

2.1.4 Структура данных для параллельного счёта

Так как число атомов может быть очень большим (до 10^9), то размещать данные по координатам, скоростям и ускорениям необходимо в глобальной памяти видеокарты. В процессе вычисления силы способ доступа к памяти является хаотичным. Чтобы упорядочить расположение блоков, проводится сортировка атомов. Если данные для вычисления силы, действующие на один атом, попадут в последовательные блоки памяти и не будут блокироваться вычислениями сил других атомов, то считывание данных будет происходить одновременно.

Выделение памяти выполняется процедурами CUDA:

```
//coordinates
if (m_bUseOpenGL)
{
    m_posVbo = createVBO(memSize);
    registerGLBufferObject(m_posVbo, &m_cuda_posvbo_resource);
}
else
{
    checkCudaErrors(cudaMalloc((void **)&m_cudaPosVBO, memSize));
}
//velocities
allocateArray((void**)&m_dVel, memSize);

//acceleration
allocateArray((void**)&m_dAcel, memSize);
```

Координаты атомов располагаются в объектах OpenGL в случае графической работы или в глобальной памяти видеокарты в случае консольной работы.

Список соседних атомов копируется в разделяемую память (`__shared__`) каждого вычислительного блока.

Параметры материала копируются в константную память (`__constant__`).

Коэффициенты потенциала для расчёта действующей на каждый атом силы копируются в дополнительную глобальную память.

2.1.5 Редукция данных для определения макропараметров

Метод редукции реализован в двух процедурах. Первая процедура выполняет вычисления по формуле (1.88):

```

data [tid] = (i < numParticles) ? mass* Mol *dot(vel, vel) /3/numParticles : 0;
__syncthreads ();

for (int s = blockDim.x/2; s>0; s>>=1){
    if (tid <s)
        data [tid]+=data [tid+s];

    __syncthreads ();
}

if (tid == 0)
    output [blockIdx.x] = data [0];
    
```

Во второй процедуре выполняются оставшиеся шаги, как показано на рис. 2.4.

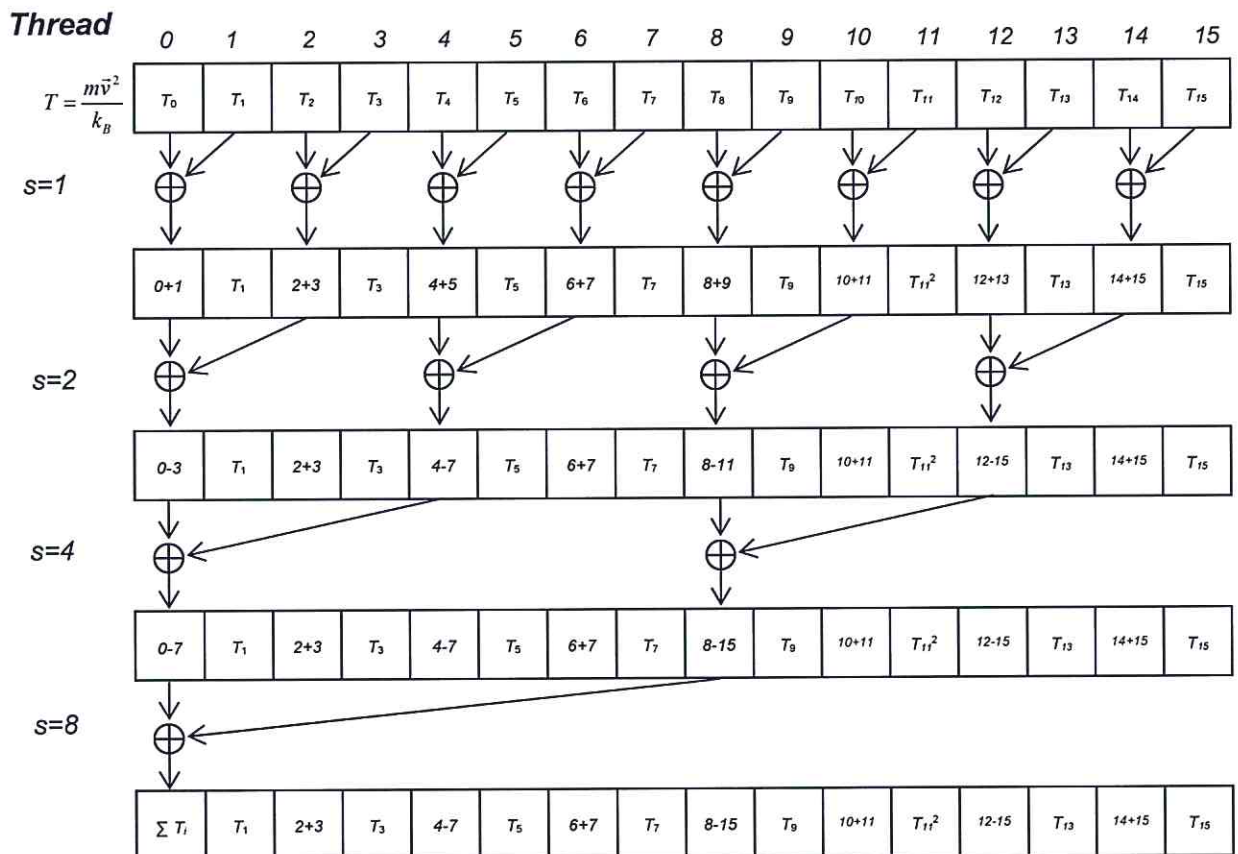


Рисунок 2.4. Иллюстрация модели параллельной редукции

Каждый вычислительный поток суммирует 2 значения и записывает результат в свою ячейку. Процесс получения глобального значения проходит в несколько этапов, число которых равно $\log_2 N$, что обеспечивает сложность $O(\frac{N}{P} \log N)$, где P – количество ядер графического процессора.

2.1.6 Отображение параллельных процессов на архитектуру GPU

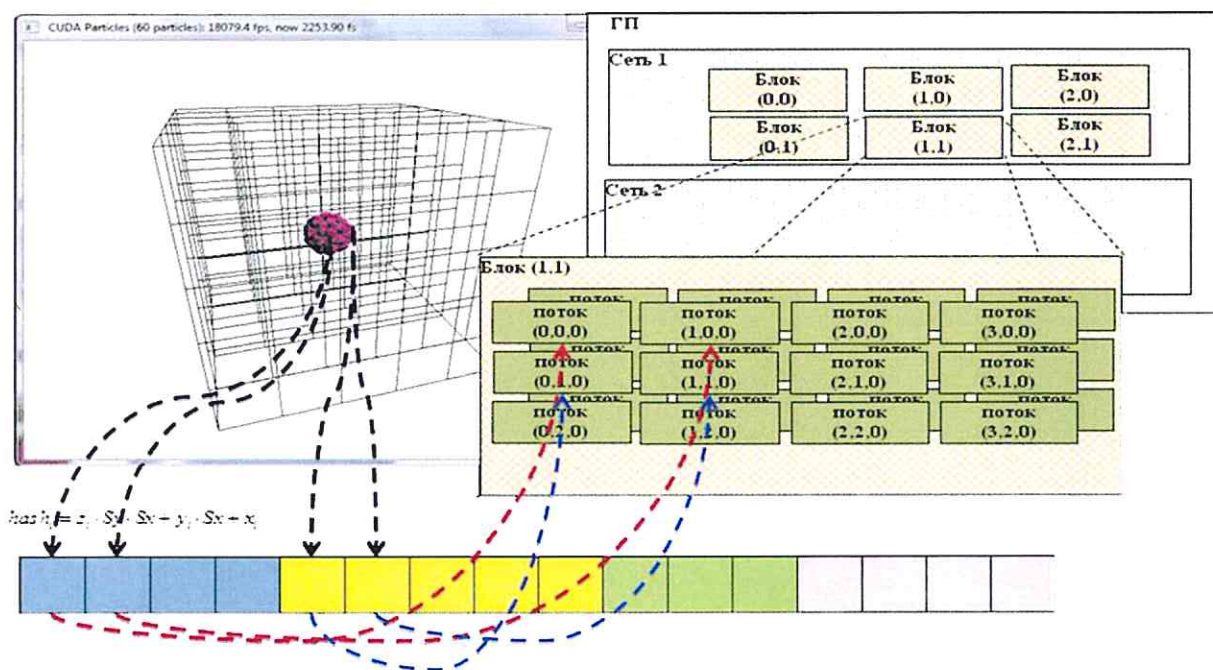


Рисунок 2.5. Иллюстрация отображения расчётной области на архитектуру GPU

Задача отображения неоднородно взаимодействующих процессов на архитектуру вычислительной машины ставится в основном с целью уменьшения времени обмена данными [52-54]. В качестве инструмента отображения в этих работах выбирается MPI, назначение которого – распределить вычисления между узлами. А исследуется и оптимизируется объём и время передачи данных между ними.

В данной работе предлагается более общее отображение задачи на архитектуру графических процессоров. Таким образом есть набор взаимодействующих процессов, в каждом процессе проводится решение уравнения движения. Вычислительный процесс оперирует данными (координатами частиц), которые хранятся в памяти.

Один процесс решает уравнение для одной частицы, но для решения требуются координаты и других частиц. Назначим для каждого вычислительного процесса вычислительный поток. Таким образом количество вычислительных потоков и блоков устанавливается перед каждым выполнением процедуры не менее необходимого количества вычислительных процессов.

Для примера в таблице 2.1 собраны параметры видеокарты Tesla M2050, характеризующие максимальное количество вычислительных потоков и блоков. Большее количество вычислительных потоков, чем поддерживает видеокарта в каждой размерности, задать технически нельзя.

Таблица 2.1. Вычислительные параметры Tesla M2075.

Максимальное количество вычислительных потоков в блоке [x,y,z]	[1024, 1024, 64]
Максимальное количество вычислительных блоков [x,y,z]	[65535, 65535, 65535]
Максимальное количество элементов	$2.8823 \cdot 10^{17}$

С одной стороны необходимо задать такое количество потоков, чтобы каждый из них решал задачу, было минимальное количество простаивающих потоков. С другой стороны количество потоков имеет технические ограничения. Взаимодействие потоков замедляет их выполнение. Поэтому выделим дополнительную память каждому потоку и сделаем копии необходимых для расчёта координат, чтобы процесс решения системы уравнений движения выполнялся вычислительными потоками без взаимодействия.

Далее рассмотрим детали каждого вычислительного процесса.

Рассмотрим количество операций $K = t \cdot n$, где $t = 1..1000000$ – количество итераций в конечно-разностной схеме, $n = 1..100000$ – количество частиц, тогда $K = 1..10^{11}$. Время расчёта равно количеству операций поделенному на ускорение:

$T = \frac{K}{S} = K(f + \frac{1-f}{p})$, где f – доля последовательных вычислений, $p = 32, 64, 128, 256,$

512, 1024 в случае вычислений на графических процессорах.

Рассмотрим вычисление сил в потенциальном поле Леннарда-Джонса:

$$\nabla U_{ij} = 4\varepsilon \left[-12 \left(\frac{\sigma}{r_{ij}} \right)^{12} + 6 \left(\frac{\sigma}{r_{ij}} \right)^6 \right] \cdot \frac{\mathbf{r}_{ij}}{r_{ij}^2}, \quad (2.4)$$

где r_{ij} – расстояние между атомами i и j , σ – расстояние, на котором энергия взаимодействия становится равной нулю, ε – глубина потенциальной ямы. Параметры σ и ε являются характеристиками атомов соответствующего вещества.

За счёт увеличения количества памяти расчёт межатомной силы можно полностью распараллелить, тогда $f = 0$, и зависимость $T = K(f + \frac{1-f}{p}) \rightarrow \frac{K}{p}$ условного времени расчётов представлена на рисунке 2.6.

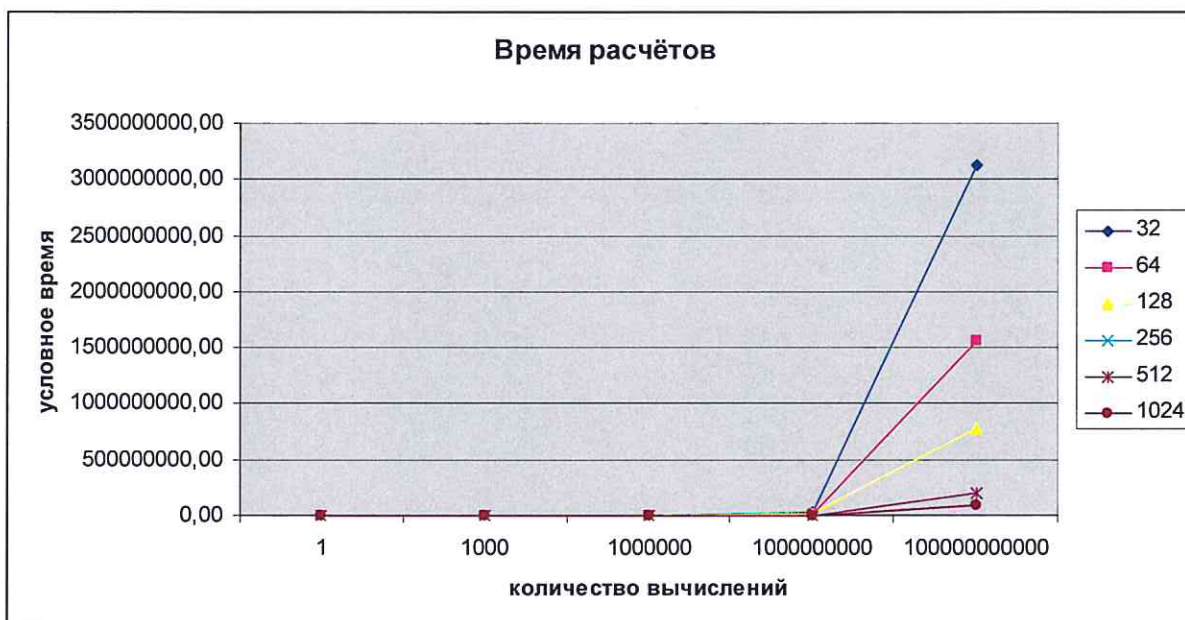


Рисунок 2.6. Условное время расчётов в зависимости от количества процессоров

Эффективное ускорение достигается за счёт полного распараллеливания $f \sim 0$. При $f \neq 0$ количество процессоров влияет в меньшей степени, чем близость f к 0. Процесс вычислений можно представить как последовательный конвейер вычислений, замедляющийся в том случае, когда количества процессоров недостаточно для выполнения. Для того чтобы определить степень распараллеливания процесса вычислений, рассмотрим количество вычислений в итерационном процессе (рис. 2.7).

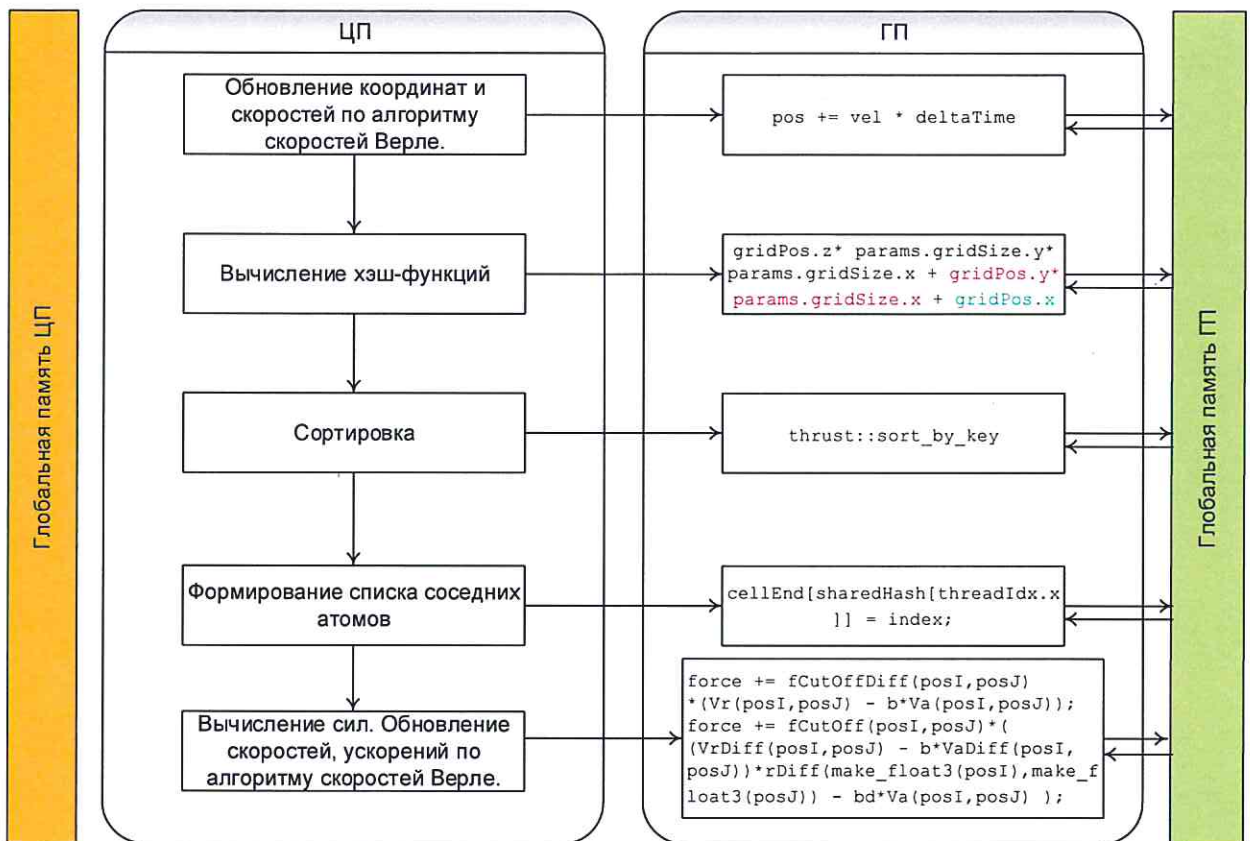


Рисунок 2.7. Диаграмма исполнения одного шага МД процесса

На рисунке 2.7 показано взаимодействие функций в разработанной программе [105]. Инициализация параметров и запуск вычислений на GPU выполняется центральным процессором. Все расчёты проводятся в графической памяти и на графических процессорах. Данные организованы в специальных переменных, состоящих из 4 компонент, переменная координат имеет компоненты из 3 пространственных составляющих и массы атома. Переменная скорости имеет 3 компоненты скорости и значение температуры, переменная ускорения имеет 3 компоненты скорости и значение потенциальной энергии. Такая организация переменных соответствует архитектуре графических карт по выравниванию памяти.

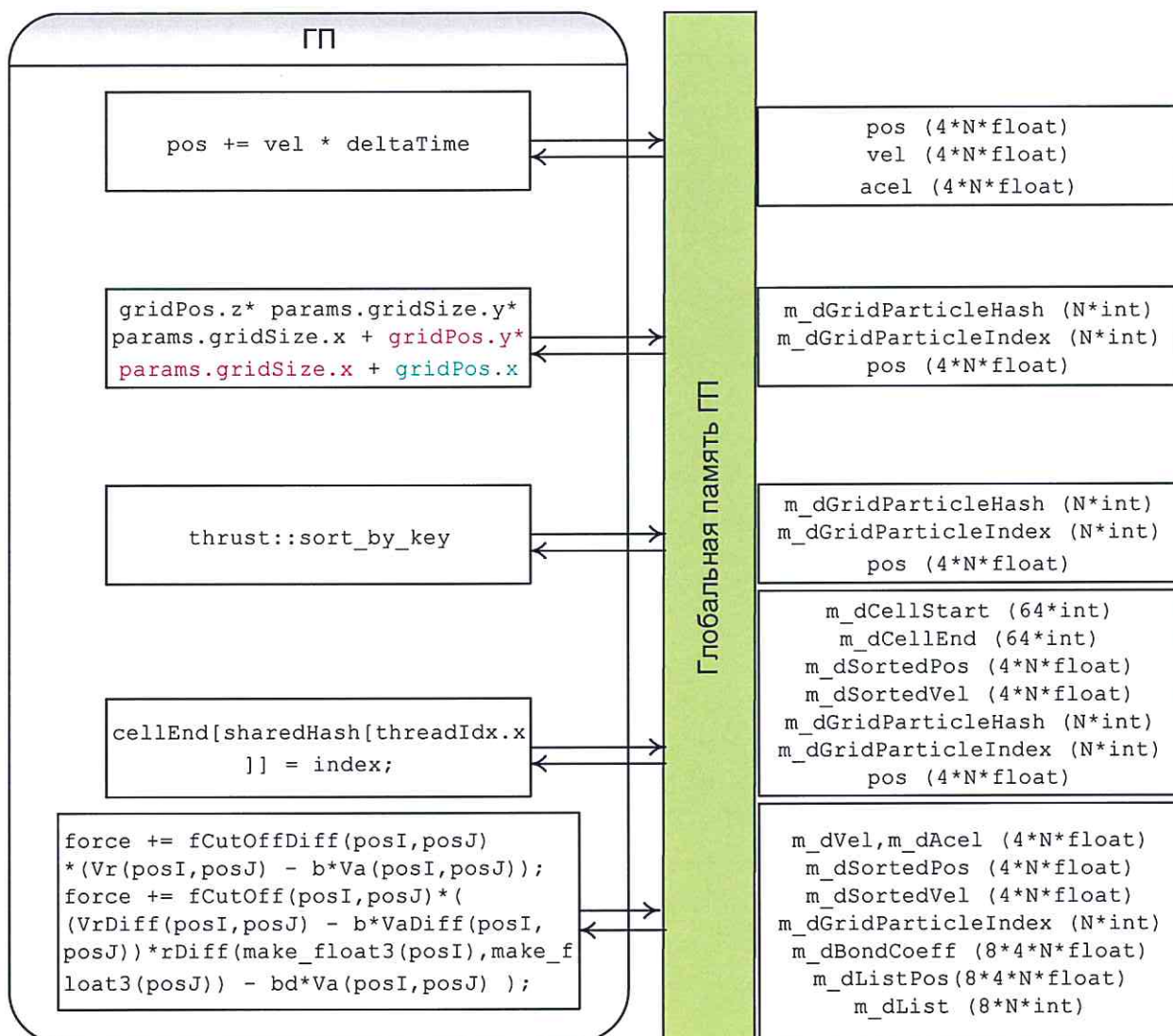


Рисунок 2.8. Объём памяти функций одного шага МД процесса. Отмечены переменные, зависящие от количества элементов

На рисунке 2.8 показано использование памяти в процедурах. Отмечены только переменные, которые зависят от количества элементов. Увеличение количества атомов в системе увеличивает объём памяти в системе для функции обновления координат и скоростей в $12*N$ раз, для функции вычисления хеш-функций – в $6*N$ раз, для функции сортировки – в $6*N$ раз, для функции формирования списка соседних атомов – в $14*N$ раз, для функции вычисления сил – в $89*N$ раз.

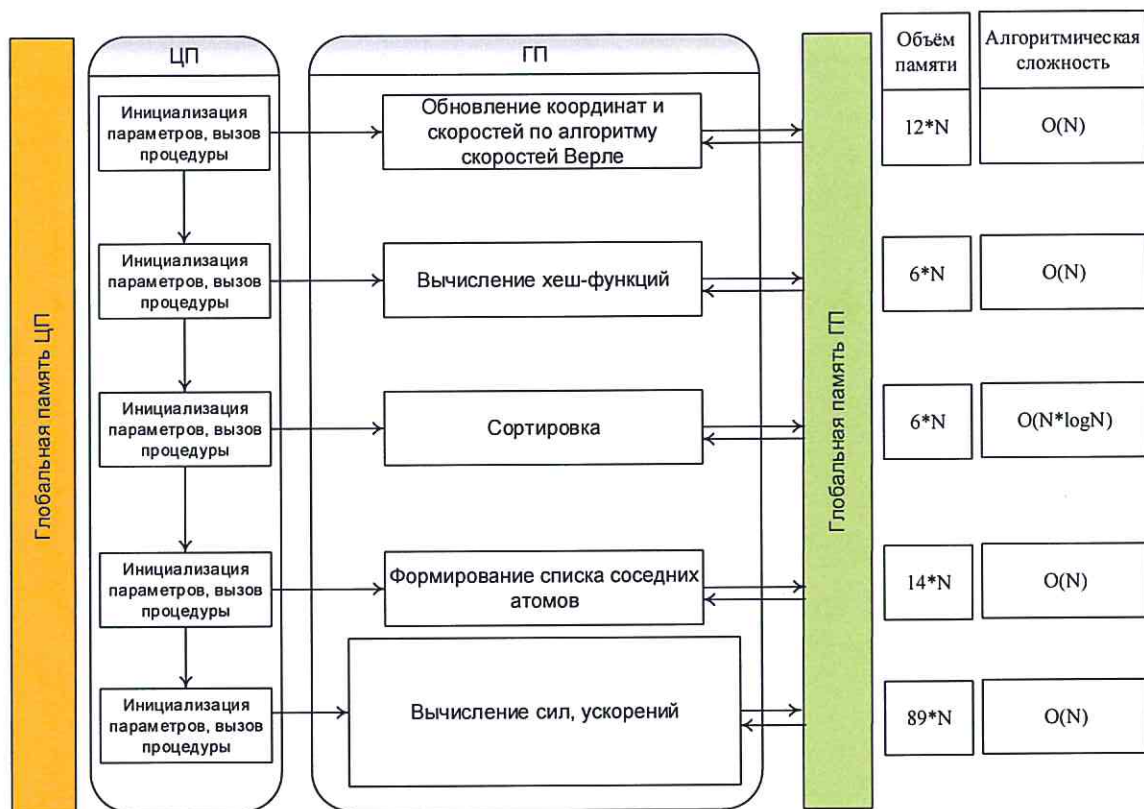


Рисунок 2.9. Алгоритмическая сложность этапов моделирования

Как показано на рис. 2.9, на центральном процессоре выполняются только последовательные части и вызовы процедур, подготовка и копирование данных на устройство, задание конфигурации вычислений на GPU и его запуск. Справа показано использование памяти в процедурах, а также их вычислительная сложность.

Для построения диаграммы использованных процессоров рассмотрим систему из $N_{\text{ит}} = 1000$ атомов и $p = 1024$ процессоров.

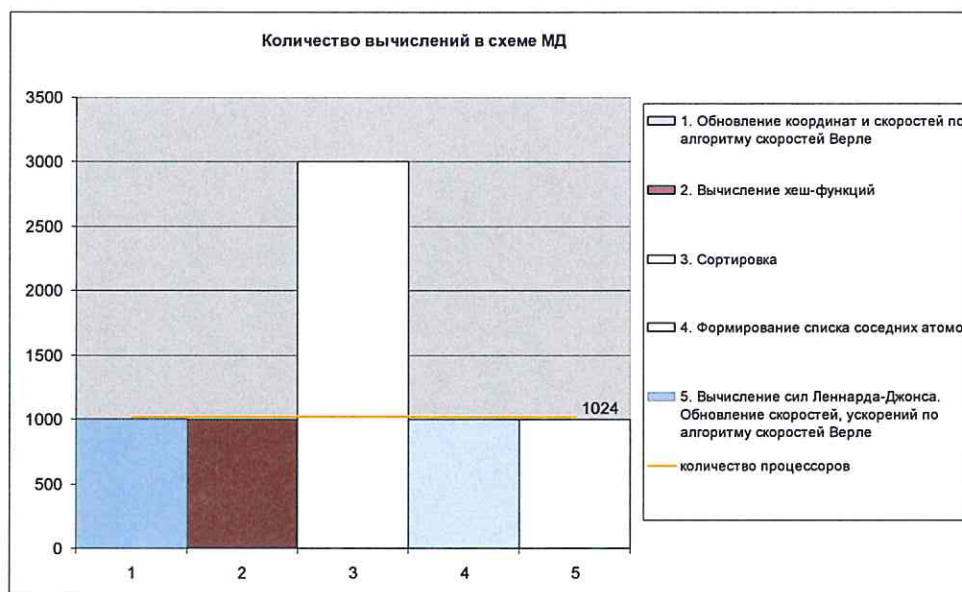


Рисунок 2.10. Профиль параллелизма на одном шаге МД при использовании потенциала Леннарда-Джонса

Из рисунка 2.10 видно, что при наличии количества процессоров большим, чем количество атомов, вычисления не проходят полностью параллельно. В столбце 3 количество вычислений превосходит количество процессоров в 3 раза, так как сложность этапа сортировки пропорциональна не $O(N)$, а $O(N \log N)$. В таком случае условное время вычислений равно 7 единиц. Время вычислений оказалось больше, чем при использовании закона Амдала и подстановке доли параллельных вычислений и количества используемых процессоров.

Рассмотрим вычисление сил в потенциальном поле Бреннера (1.41). Шаги итерационного процесса выполняются аналогично рис. 2.7 с заменой потенциала Леннарда-Джонса на потенциал Бреннера. Количество вычислений в итерационном процессе представлено на рис. 2.11. В столбце 5 количество вычислений превосходит количество процессоров в 4 раза, так как сложность этапа вычисления сил пропорциональна $4N$. В таком случае условное время вычислений равно 10 единиц.

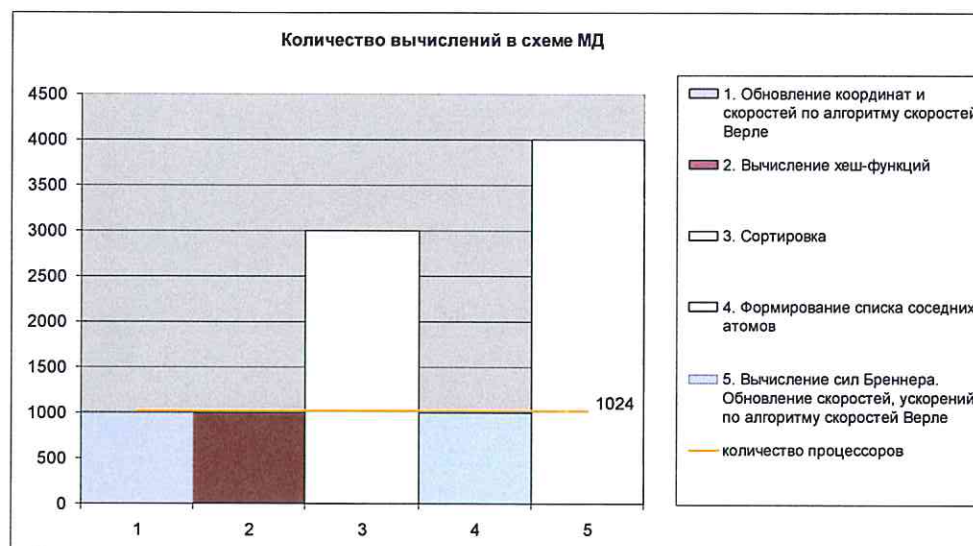


Рисунок 2.11. Количество вычислений на одном шаге МД при использовании потенциала Бреннера

Распределение количества вычислений в потоке вычислений не полностью отражает затраченное время, так как ускорение вычислений при параллельных расчётах достигается за счёт использования дополнительной памяти. Использование значительного объёма дополнительной памяти, характеризующейся степенной зависимостью от количества атомов приведёт к пропорциональному замедлению. Количество использованной памяти показано на рисунке 2.12.

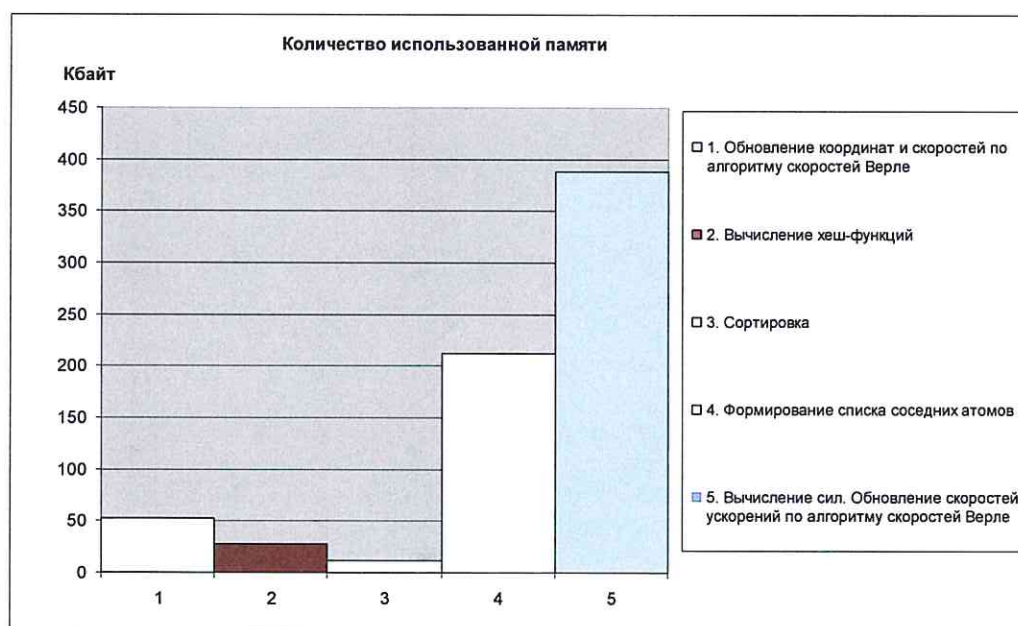


Рисунок 2.12. Количество использованной памяти на одном шаге МД

Из рисунка 2.11 и 2.12 видно, что, например, 2 и 4 этапы имеют одинаковое количество потоков вычислений и имеют сравнимое количество математических

операций, но 5-ый этап будет выполняться в несколько раз дольше, так как потребуются обращения к большому количеству данных.

2.2 Сложность вычислений и затраты памяти

Схема скоростей Верле разбивается на две части: выполнение операций обновления координат и скоростей и вычисление сил, действующих на атомы, и результирующих ускорений.

Доступ к памяти улучшается за счёт сортировки элементов. Вычислительные потоки записывают результаты в один массив одновременно, при следующем обращении к памяти они используют весь массив необходимых значений.

Вычисление сил разбивается на два этапа: вычисление сложных коэффициентов потенциала межатомного взаимодействия с их сохранением в дополнительную память и использование готовых коэффициентов для суммирования сил со стороны всех атомов. Такой подход позволяет избежать блокирования доступа к памяти для параллельных потоков вычислений.

Основной задачей разработки алгоритма вычисления потенциала Бреннера на GPU является обеспечение наиболее эффективного доступа к памяти. В разработанном программном комплексе для этого выделяется дополнительная память и переменные, которые хранят копии координат до 8 соседних атомов. Из функции вычисления сил выносятся расчёт коэффициентов, так что вычислительно сложная часть операций перенесена в специальную функцию. Затратная по времени функция составления списка соседних атомов оперирует значениями хеш-функции и дополнительной памятью, после построения списка расположение координат атомов оказывается выровненным в памяти, и не возникает перекрёстного доступа на следующих этапах. Можно сформулировать следующее утверждение.

Утверждение. Алгоритм расчёта атомных сил с выделением дополнительной памяти для хранения списка взаимодействующих атомов имеет вычислительную сложность $O(\frac{N}{P})$, где P – количество ядер графического процессора.

Доказательство.

Согласно предлагаемому алгоритму количество потоков устанавливается равным количеству элементов N . При вычислениях на GPU имеет место пропорциональная зависимость затрат времени от загруженности памяти. Использование ограничивающей функции фиксирует объём памяти в зависимости от s – числа ковалентных связей, и не зависящее от количества атомов в системе. Учитывая введение дополнительной памяти в количестве m ячеек, вычисления по формуле (1.18) с известными значениями b_{ij} потребуют $s \cdot m \cdot N$ условных единиц времени. Вычисления по формуле (1.20) за счёт функции ограничения потребуют $s^3 \cdot N$ тактов. Общее число условных единиц времени составит $s^3 \cdot N + s \cdot m \cdot N$. За счёт реализованных методов у каждого потока вычислений есть набор данных и инструкций, одному потоку вычислений нет необходимости ожидать, пока другой поток подготовит данные для расчёта (рис.2.13). N потоков выполняются на P процессорах, что даёт сложность $(s^3 + sm) \frac{N}{P}$ условных единиц времени.

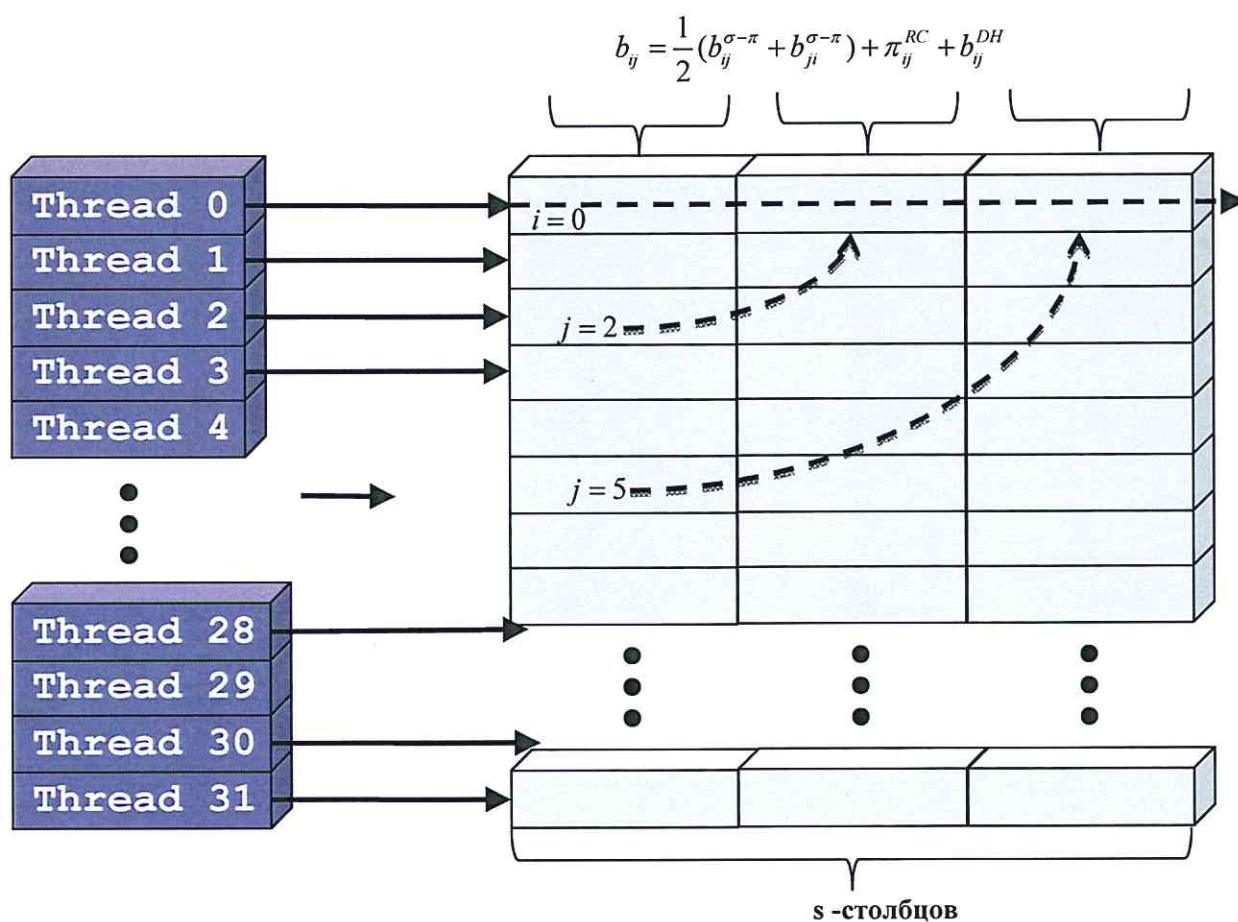


Рисунок 2.13. Организация бесконфликтных запросов памяти при вычислении потенциала Бреннера

В качестве примера, позволяющего продемонстрировать методы работы с памятью, рассмотрим моделирование фуллерена (рис. 2.14). В фуллерене каждый атом имеет три соседних, таким образом, задействованы все параметры потенциала: угловая функция, двугранные углы, корректирующие значения связности. В отличие от пространственно вытянутого листа графена такая структура имеет компактное расположение атомов.

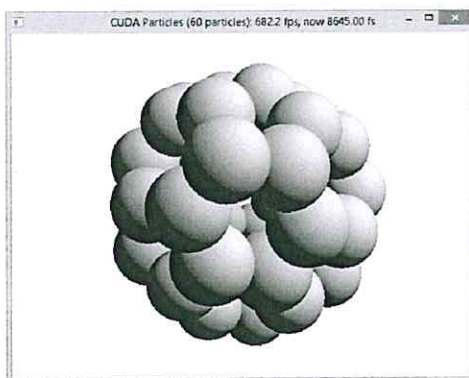


Рисунок 2.14. Моделирование фуллерена

Применение хеш-функции и сортировки при условии использования 16 вычислительных потоков в блоке и 64 ячейках памяти позволило уменьшить количество перекрёстных запросов к памяти на 13%.

2.3 Методы анализа ускорения вычислений

Изменяя количество процессоров и замеряя время выполнения программы, можно вычислить ускорение. В общем виде ускорение можно записать как:

$$s = \frac{T(1)}{T(p)}, \quad (2.5)$$

где $T(1)$ – время решения задачи на 1 процессоре (собственное время), $T(p)$ – время решения задачи на p процессорах.

Достижимое ускорение расчётов даёт закон Амдала [55]:

$$s = \frac{1}{f + \frac{1-f}{p}}, \quad (2.6)$$

где f – доля последовательных вычислений, p – количество процессоров. Для отдельных этапов вычислений на графических процессорах f равно или близко к 0.

Время выполнения программы на p процессорах:

$$T(p) = T_s + \frac{T_p}{p}, \quad (2.7)$$

где T_s – время выполнения последовательного кода, T_p – время выполнения параллельного кода, и сумма $T_s + T_p = T(1)$, а $f = \frac{T_s}{T(1)}$ – доля последовательных вычислений.

Если выразить из формулы (2.6) долю последовательных вычислений:

$$\frac{1}{s} = f + \frac{1-f}{p} \Rightarrow \frac{1}{s} - \frac{1}{p} = f - \frac{f}{p} \Rightarrow f_{KF} = \frac{\frac{1}{s} - \frac{1}{p}}{1 - \frac{1}{p}}, \quad (2.8)$$

где f_{KF} – метрика Карпа-Флэтта [56], показывающая экспериментальное значение распараллеливания кода. Чем меньше значение f_{KF} , тем лучше может быть распараллелена программа.

Решая на вычислительной системе из 1024 процессоров три больших задачи, для которых доля последовательного кода f лежала в пределах от 0,4 до 0,8%, Густафсон [57] получил значения ускорения по сравнению с однопроцессорным вариантом равные соответственно 1021, 1020 и 1016. Согласно закону Амдала для данного числа процессоров и диапазона f , ускорение не должно было превысить величины порядка 201.

Чтобы оценить степень ускорения вычислений, когда объём последних увеличивается с ростом количества процессоров в системе (при постоянстве общего времени вычислений), Густафсон рекомендует использовать выражение, предложенное Е. Барсисом (Ed Barsis):

$$s = f + (1-f) \cdot p = p + (1-p) \cdot f. \quad (2.9)$$

В формулах ускорения входит количество процессоров, при увеличении количества процессоров время расчётов уменьшается. Для определения того, насколько время уменьшается, введена метрика эффективности [58]:

$$e = \frac{T(1)}{p \cdot T(p)}. \quad (2.10)$$

В практических измерениях ускорений не всегда возможно получить время выполнения программы на одном процессоре, так как мощность вычислений

достигает миллиардов элементов. Ещё сложнее получить время выполнения, если для расчётов используются графические процессоры, их архитектура не позволяет задать одно ядро, а в общем случае, когда объём задачи превосходит количество ядер, то нет возможности задать какое-либо количество ядер отличное от максимально доступного на графической плате. Для таких случаев введена метрика скорости [59]:

$$R(n) = \frac{n}{T(n, p)}, \quad (2.11)$$

где n – количество элементов в задаче, $T(n)$ – время решения задачи из n элементов на 1 процессоре, $T(n, p)$ – время решения задачи из n элементов на p процессорах. В таком случае ускорение:

$$s(n, p) = \frac{T(n)}{T(n, p)} = \frac{T(n)}{p} R(n, p). \quad (2.12)$$

2.4 Наследование в технологии CUDA

Программный комплекс конструируется из классов. Основной класс описывает систему частиц:

```
// BodySystem abstract base class
template <typename T>
class BodySystem
{
public: // methods
    BodySystem(int numBodies) {}
    virtual ~BodySystem() {}

    virtual void update(T deltaTime) = 0;

    virtual T *getArray(BodyArray array) = 0;
    virtual void setArray(BodyArray array, const T *data) = 0;

    virtual unsigned int getCurrentReadBuffer() const = 0;

    virtual unsigned int getNumBodies() const = 0;

    virtual void synchronizeThreads() const {};

protected: // methods
    BodySystem() {} // default constructor

    virtual void _initialize(int numBodies) = 0;
    virtual void _finalize() = 0;
};
```

От него наследуются классы, которые реализованы на CPU:

```
// CPU Body System
```



```

template <typename T>
class BodySystemCPU : public BodySystem<T>
{
public:
    BodySystemCPU(int numBodies);
    virtual ~BodySystemCPU();

    virtual void update(T deltaTime);

    virtual T *getArray(BodyArray array);
    virtual void setArray(BodyArray array, const T *data);

    virtual unsigned int getNumBodies() const
    {
        return m_numBodies;
    }

protected: // methods
    BodySystemCPU() {} // default constructor

    virtual void _initialize(int numBodies);
    virtual void _finalize();

    void _integrateNBodySystem(T deltaTime);

protected: // data
    int m_numBodies;
    bool m_bInitialized;

    T *m_pos;
    T *m_vel;
    T *m_force;
};

```

и GPU:

```

// CUDA BodySystem: runs on the GPU
template <typename T>
class BodySystemCUDA : public BodySystem<T>
{
public:
    BodySystemCUDA(unsigned int numBodies,
                  unsigned int numDevices,
                  unsigned int blockSize,
                  bool usePBO,
                  bool useSysMem = false);
    virtual ~BodySystemCUDA();

    virtual void update(T deltaTime);

    virtual T *getArray(BodyArray array);
    virtual void setArray(BodyArray array, const T *data);

    virtual unsigned int getCurrentReadBuffer() const
    {
        return m_pbo[m_currentRead];
    }
}

```

```

    virtual unsigned int getNumBodies() const
    {
        return m_numBodies;
    }

protected: // methods
    BodySystemCUDA() {}

    virtual void _initialize(int numBodies);
    virtual void _finalize();

protected: // data
    unsigned int m_numBodies;
    unsigned int m_numDevices;
    bool m_bInitialized;

    // Host data
    T *m_hPos[2];
    T *m_hVel;

    DeviceData<T> *m_deviceData;
};

```

Пусть ставится задача добавить новый потенциал в программу. Требуется написать только новый класс, реализующий новый потенциал, и не затрагивать остальной основной код.

Тогда основной класс можно представить следующим образом:

```

class Pair : protected BodySystem {
public:

    virtual void evaluateForces(lattice, forces);
    // general child-class methods
    virtual void compute(int, int) = 0;
};

```

Наследуется класс, вычисляющий потенциала Бреннера, от Pair для CPU реализации:

```

class PairREBO : public Pair {
public:

    virtual ~PairREBO();
    virtual void compute(int, int);
};

```

на GPU:

```

class PairREBOCuda : public PairREBO
{
    public:

```

```

        void compute(int, int);
        void settings(int, char **);
        void init_list(int, class NeighList *);
protected:

        void allocate();
        class CudaNeighList *cuda_list; // CUDA neighbor list
};

```

Потенциал Леннарда-Джонса имеет простую структуру, по сравнению с потенциалом Бреннером, а наследуется таким же способом на CPU:

```

class PairLJ : public Pair {
public:

    PairLJ();
    virtual ~PairLJ();
    virtual void compute(int, int);
};

```

и GPU:

```

class PairLJCuda : public PairLJ
{
public:
    void compute(int, int);
    void settings(int, char **);
    void init_list(int, class NeighList *);
protected:

    void allocate();
    class CudaNeighList *cuda_list; // CUDA neighbor list
};

```

Функция вычисления сил объявляется в основном классе и затем переопределяется в каждом классе-наследнике. Запуск производится на хостовой части:

```

void PairREBOCuda::compute()
{
    Pair_REBO_Kernel <<< grid, threads >>>
}

```

Новый потенциал добавляется подобным образом, CPU реализация выглядит следующим образом:

```

class PairNew : public Pair {
public:

    PairNew();
    virtual ~PairNew();
};

```

```
virtual void compute(int, int);
};
```

и на GPU:

```
class PairNewCuda : public PairNew
{
    public:
        void compute(int, int);
        void settings(int, char **);
        void init_list(int, class NeighList *);
    protected:
        void allocate();
        class CudaNeighList *cuda_list; // CUDA neighbor list
};
```

В потенциалах вызывается класс, который подготавливает список соседей. Он может быть общий для всех потенциалов или иметь расширение для нового потенциала. Реализация такого класса имеет вид:

```
class CudaNeighList : protected BodySystem {
    public:
        int index; // index of which neighbor list is required for a potential
    protected:
        int maxatoms; // size of allocated atom arrays
};
```

Вычислительное ядро на GPU объявляется шаблонной функцией, которая принимает любой потенциал:

```
template <class Potential>
__global__ Pair_Kernel () {
    i = threadIdx;
    forces[i] += potential.calcForce();
}

template __global__ void Pair_Kernel (float*, float*, size_t);
```

2.5 Технология разработки программного обеспечения

Процесс программирования алгоритмов молекулярно-динамического моделирования на графических картах разбивается на три этапа: выбор основы проекта, добавление основного функционала, добавление внешнего функционала. На первом этапе выбирается шаблон CUDA, который соответствует поставленной задаче.

Производитель NVIDIA предоставил широкий набор шаблонов; выбор одного из них обеспечит длительную поддержку и полную совместимость с аппаратным обеспечением и операционными системами. На втором этапе в проект включается основной функционал. Рекомендуется включать процедуры и дополнительные файлы так, чтобы они собирались линковщиком базового проекта. На третьем этапе в проект добавляется дополнительный функционал на уровне кода или совместимых модулей. Описанный процесс можно представить на схеме (рис. 2.15).

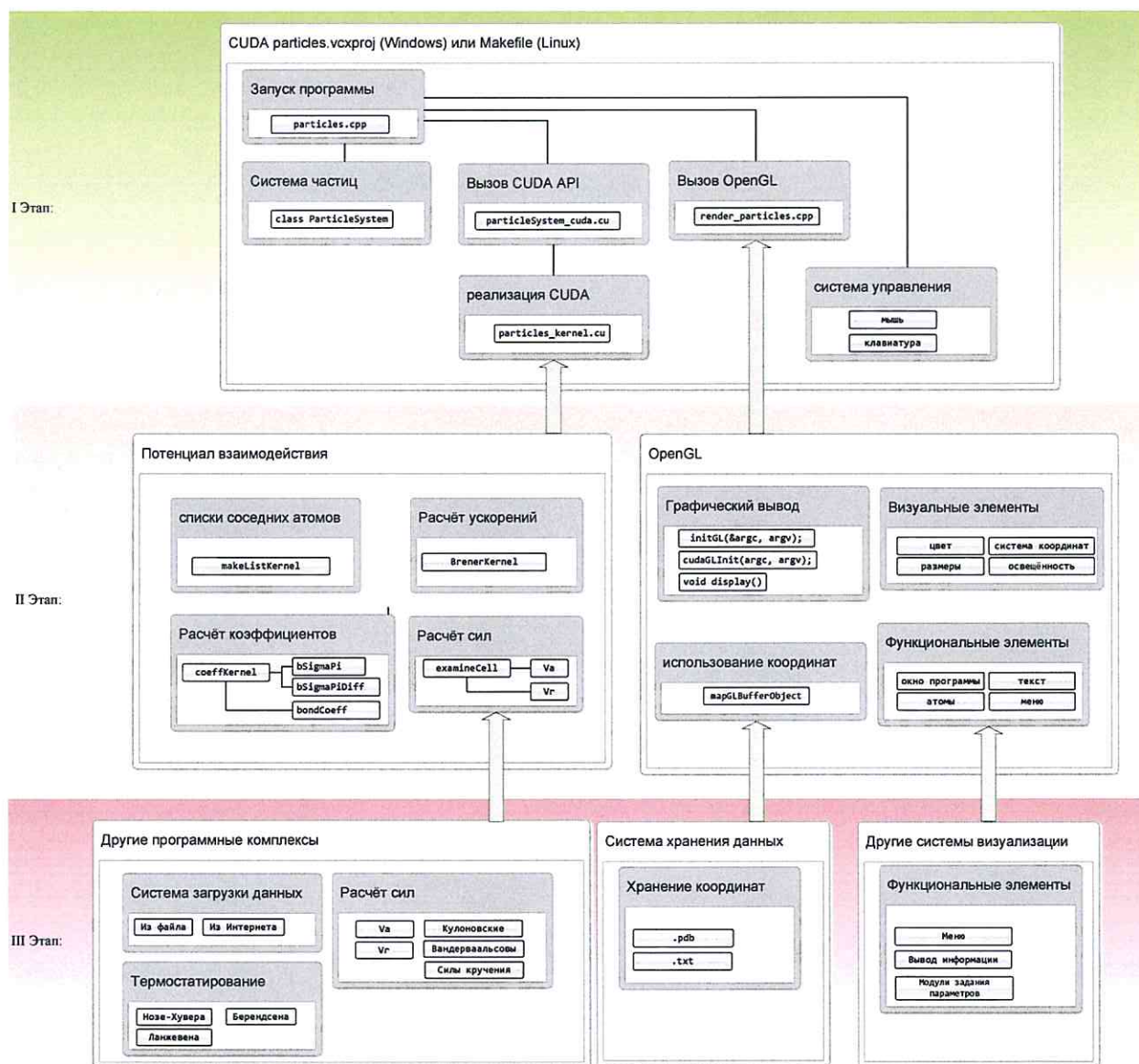


Рисунок 2.15. Схема процесса разработки взаимодействующих компонент программы

Архитектура разработанного программного комплекса содержит модули CUDA, OpenGL, системы ввода-вывода баз данных, системы создания, редактирования

структур, системы эволюции. Структуры нанообразцов могут создаваться изначально, могут редактироваться в процессе моделирования, могут загружаться из базы данных. Координаты записываются в переменные, состоящие из 4 компонентов: 3 пространственных составляющих и массы атома. Атомы имеют координаты в пространстве и не требуют дополнительных сведений по связям. Связи между атомами визуально не отображаются. Связность двух атомов определяется за счёт параметров межчастичного потенциала. Такой подход удобен как для хранения данных, так и для исключения ошибок моделирования.

В качестве примера приведена технология создания программы молекулярно-динамического моделирования наноструктур; на первом этапе за основу проекта берётся шаблон `CUDA particles`, предоставляющий базовый набор инструментов моделирования многочастичного взаимодействия. На втором этапе в проект добавляется функционал геометрического построения, межатомного взаимодействия, выделения памяти, создания списков соседних частиц, отображения структур. На третьем этапе добавляется код модуля генерации начального распределения в основной проект. Проект целиком собирается в рамках базового компоновщика.

2.6 Визуализация вычислений в реальном времени

Одна из компонент программных комплексов молекулярно-динамического моделирования осуществляет анимацию и вывод на экран. Известные системы, такие как LAMMPS и NAMD, реализуют вычислительный блок отдельно от графического отображения.

В рамках предлагаемой технологии разработки программ отображение результатов и процесса моделирования добавляется на втором этапе. Реализацию алгоритмов отрисовки и систему взаимодействия данных необходимо заложить в процессе проектирования, так как переменные используются во всех процедурах комплекса. Это принципиальное решение – хранить данные в подходящих структурах, чтобы их вывести на экран, а также чтобы была возможность работы на различных вычислительных машинах.

Визуализация моделирования важна, поэтому одна из задач реализации алгоритмов молекулярно-динамического моделирования – это 3D отображение. В наноинженерии принципиально именно трёхмерное отображение структур, что даёт первоначальное представление о правильности моделирования.

Программные комплексы работают в разных операционных системах. Научные программы и программы для физического моделирования в основном работают под Linux системами. Программы для графического отображения данных в основном работают под Windows системами. Поэтому актуальна задача визуализировать полученные данные расчётных программ. Основным инструментом, который позволяет отображать графику под Linux и Windows, является OpenGL. Чаще всего программы сначала сохраняют данные в файлы на диске и затем считывают их и отображают на экране. Такой процесс может занимать несколько дней. В разработанной программе за счёт получения данных в памяти графической карты результаты могут выводиться на экран в реальном времени. Для вывода на экран также используется система OpenGL. Для сравнения на рисунках представлены различные образцы наноструктур – в программе ovito, VMD и в разработанной программе.

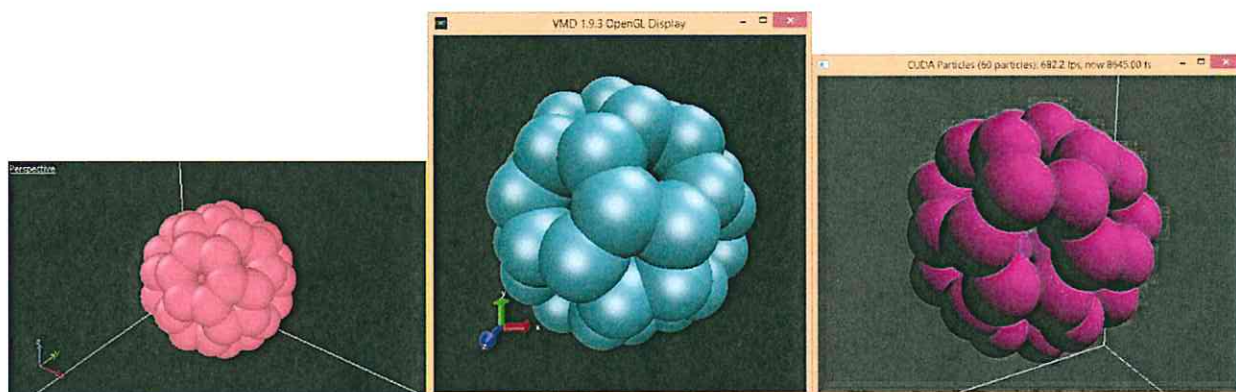


Рисунок 2.16. Вид фуллерена в ovito, VMD, разработанной программе

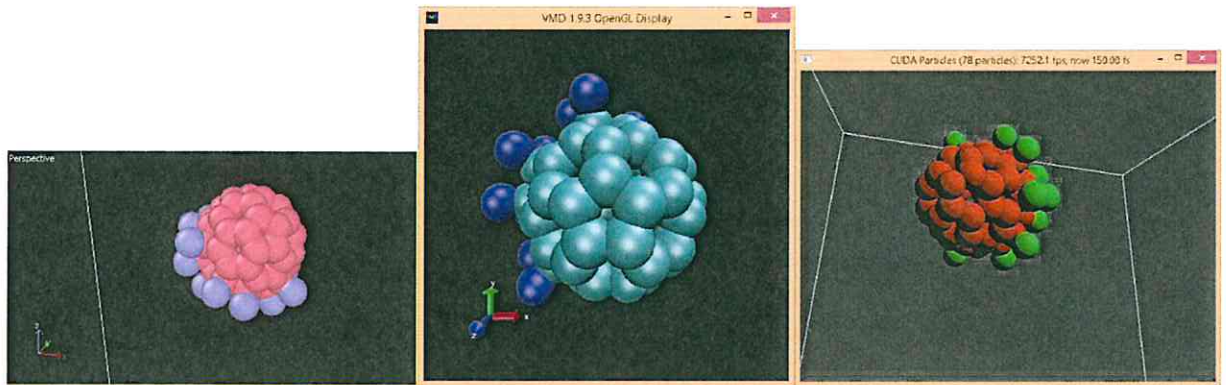


Рисунок 2.17. Вид флуорофуллерепа в ovito, VMD, разработанной программе

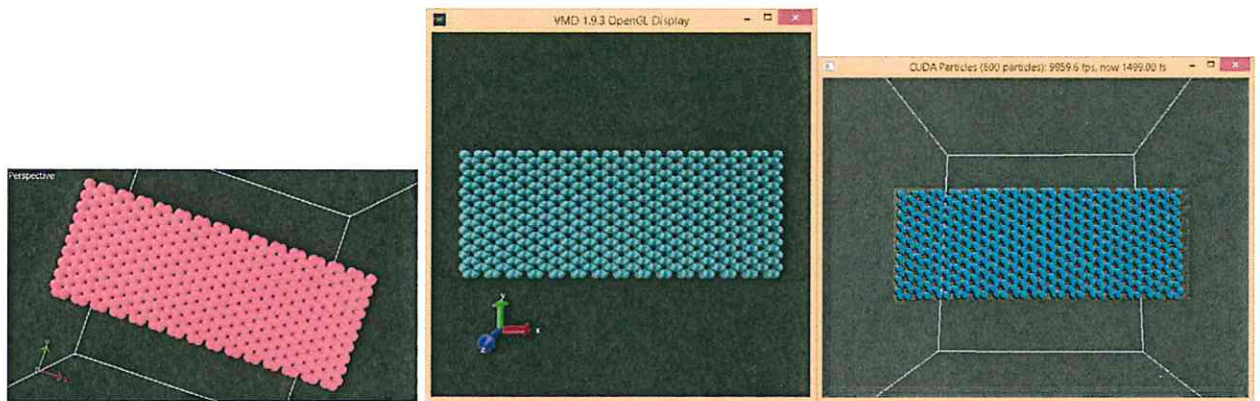


Рисунок 2.18. Вид листа графена в ovito, VMD, разработанной программе

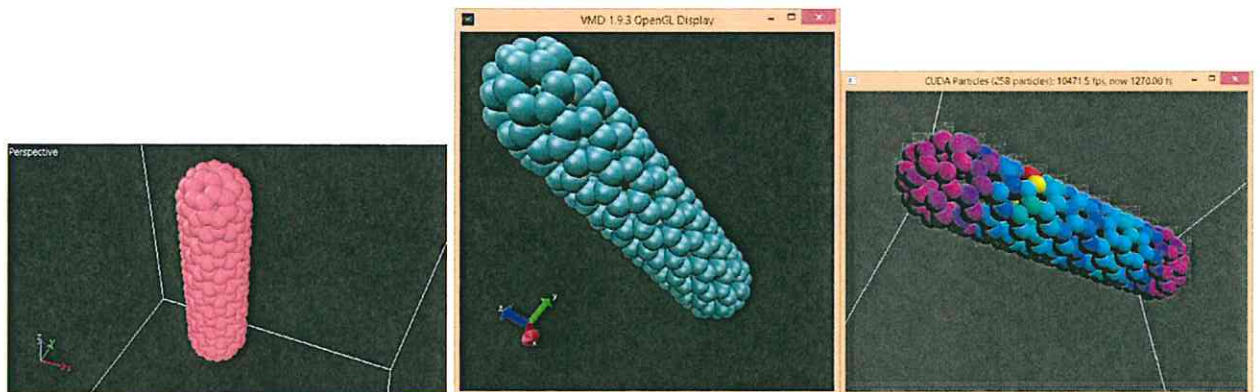


Рисунок 2.19. Вид нанотрубки (9,0) в ovito, VMD, разработанной программе

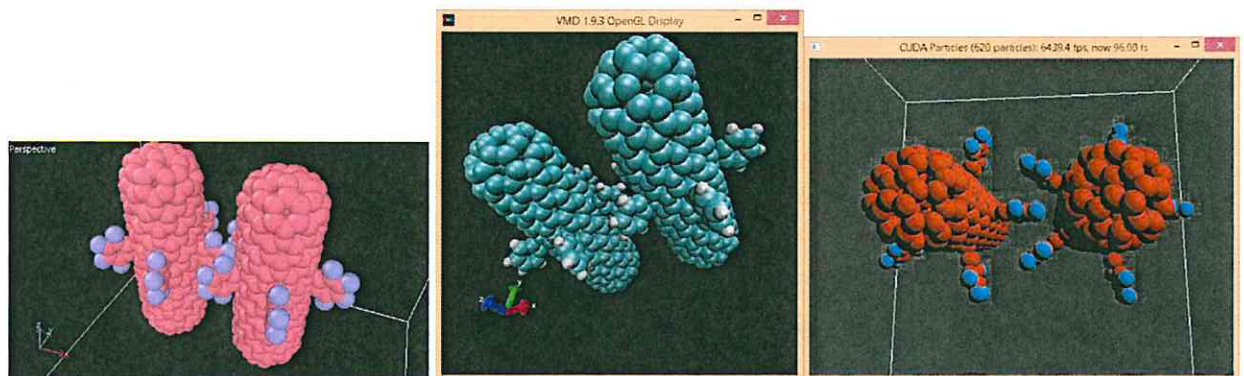


Рисунок 2.20. Вид цепной передачи в ovito, VMD, разработанной программе

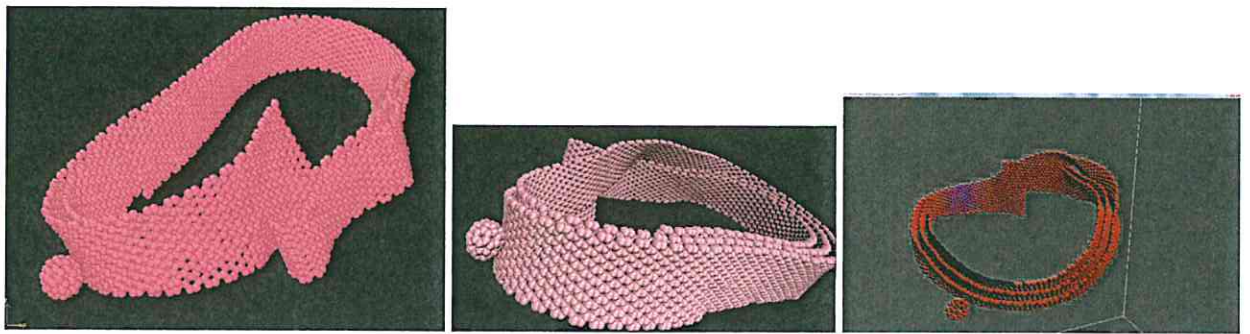


Рисунок 2.21. Вид фуллерена, катящегося по листу графена в ovito, VMD, разработанной программе

Архитектура разработанной программы (рис. 2.22) для работы с 3D графикой состоит из данных, содержащих координаты атомов, наборов компонент OpenGL для отображения и изменения сцены и главного цикла отрисовки, который совпадает с главным циклом вычислений. Таким образом, данные моделирования наноструктур отображаются на экране в момент поступления данных. В этом программа имеет преимущества перед всеми указанными выше комплексами.

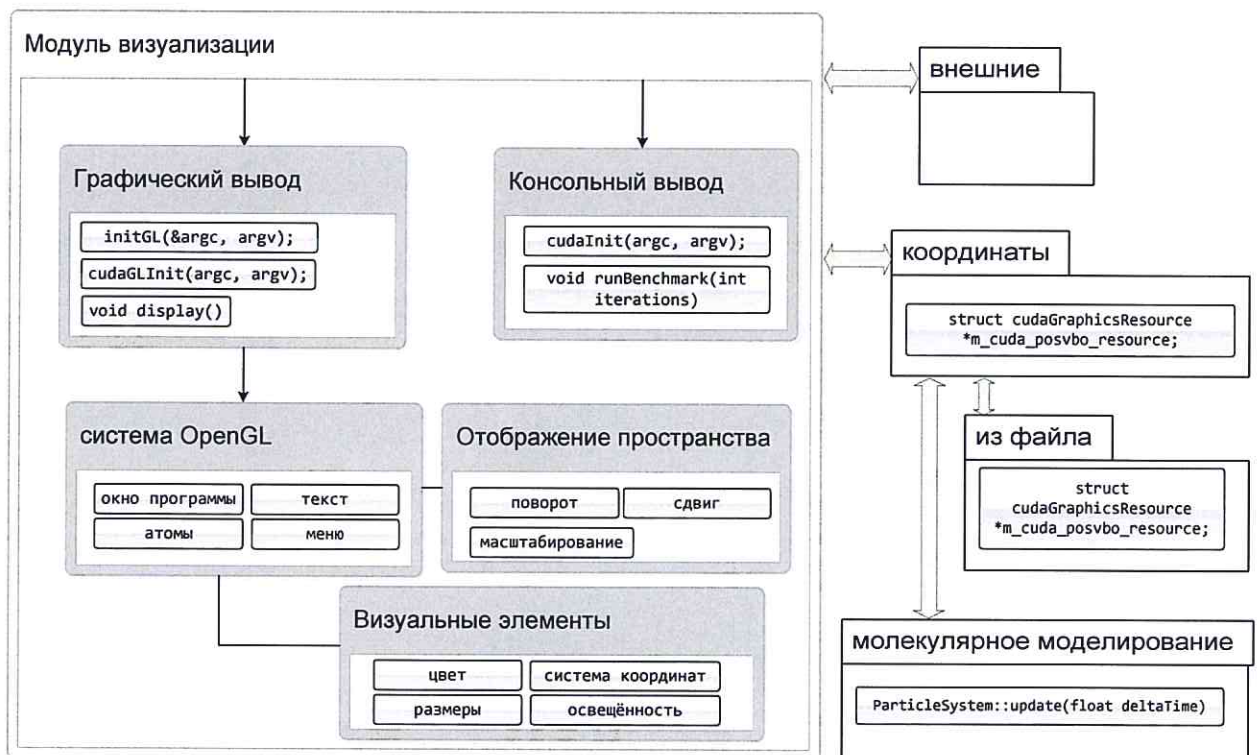


Рисунок 2.22. Архитектура подсистемы визуализации процессов моделирования

Формат хранения данных Proteins Data Bases [60] имеет широкое распространение, поэтому положительной стороной работы программы является отображение таких данных. База поддерживается многими исследователями из разных

областей науки. Структура данных позволяет хранить миллионы атомов. Большинство программ моделирования и отображения поддерживают формат `pdb`.

Атомы нумеруются по порядку, номер атома не несёт информационной составляющей. Ячеистая модель пространства и списки соседних атомов основываются на координатах.

В файле данных атом маркируется символом из периодической таблицы элементов. В программе символ элемента заменяется на массу.

Каждому элементу задаётся цвет для отображения. Цвет может заменяться на градиентную расцветку для отображения температуры.

Элементу задан радиус в ангстремах в соответствии с физическими характеристиками.

Атомы имеют массы и трёхмерные координаты, на экран выводится их проекция в перспективе. Проекцию можно поворачивать, двигать, масштабировать манипулятором типа «мышь». Пространство ограничено кубическим боксом. Поворот проекции происходит относительно центра окна программы вместе с боксом.

Разработанный программный комплекс позволяет отображать процесс молекулярно-динамического моделирования на персональном компьютере при наличии графической видеокарты от NVIDIA.

Фазовый переход листа графена можно наблюдать в молекулярно-динамическом процессе, так как потенциал Бреннера позволяет моделировать структуру вещества во всём диапазоне температур. Фазовые переходы играют важную роль в формировании граней вещества.



Рисунок 2.23. Значения потенциальной энергии в зависимости от температуры

На рисунке видно 3 фазовых перехода при температурах $T_1=2260\text{K}$, $T_2=2750\text{K}$ и $T_3=3520\text{K}$. Изменения потенциальной энергии при этом равны $\Delta U_1=4.08\text{eV/atom}$, $\Delta U_2=3.02\text{eV/atom}$, $\Delta U_3=3.0\text{eV/atom}$.

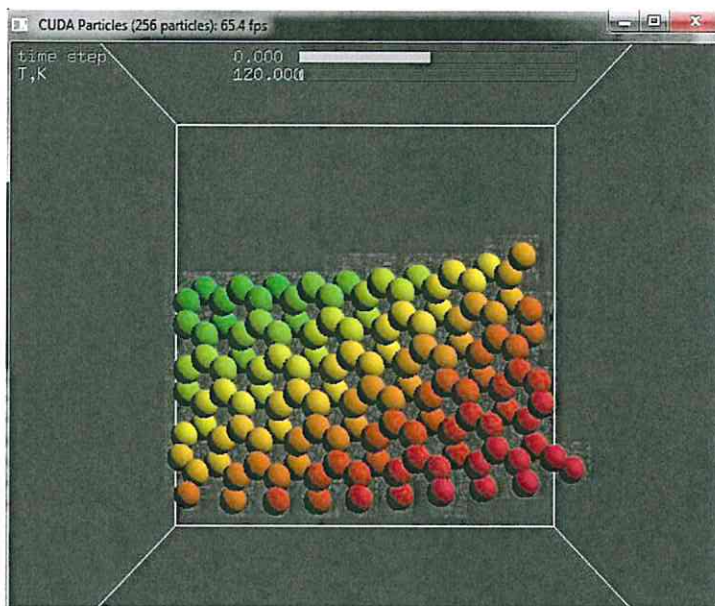


Рисунок 2.24. Начальный вид листа графена при $T=120\text{K}$, $\Delta H_2 = 2,34\text{E}+07$ Дж/кг

Цветом выделена температура: красный цвет показывает горячие области, зелёный – холодные.

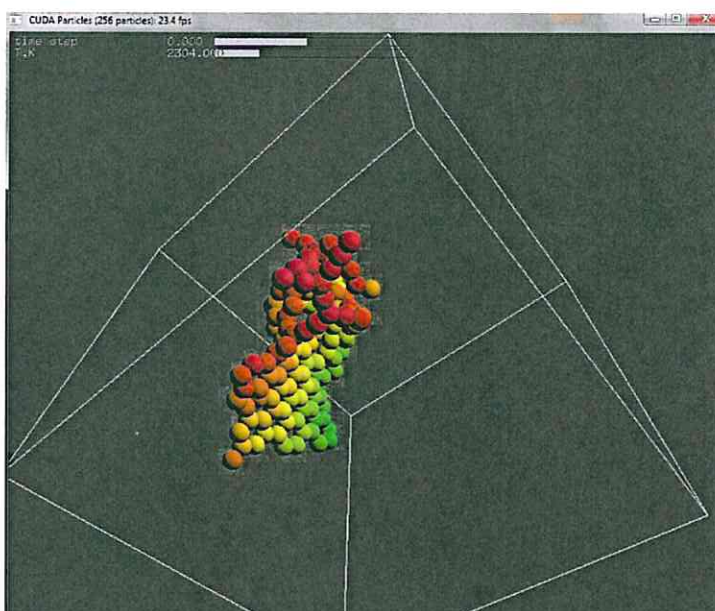


Рисунок 2.25. Вид листа графена при $T=2260\text{K}$, $\Delta H_3 = 2,25\text{E}+07$ Дж/кг

Из литературных источников можно найти значения 1400-2800K для фазового перехода углерода [61-62].

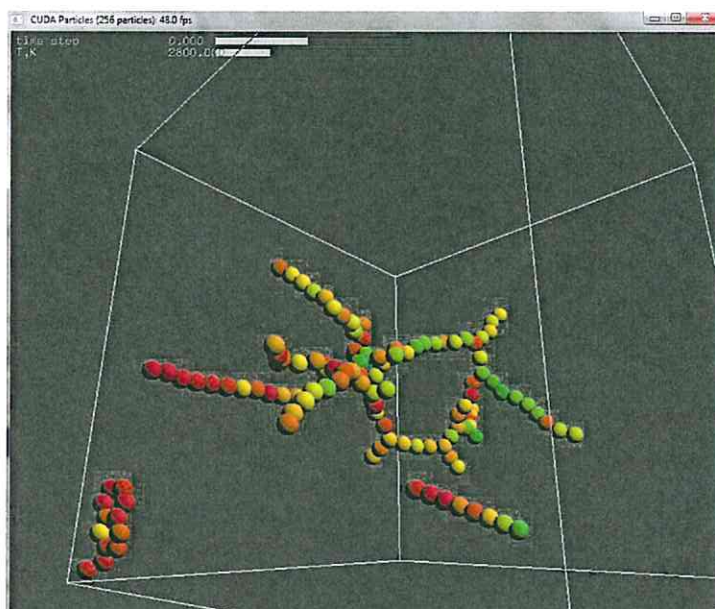


Рисунок 2.26. Вид листа графена при $T=2750\text{K}$, $\Delta H_2 = 2,34\text{E}+07$ Дж/кг

Из литературных источников можно найти значения 2700-3000К для плавления углерода, 2700-3600К для испарения углерода [63].

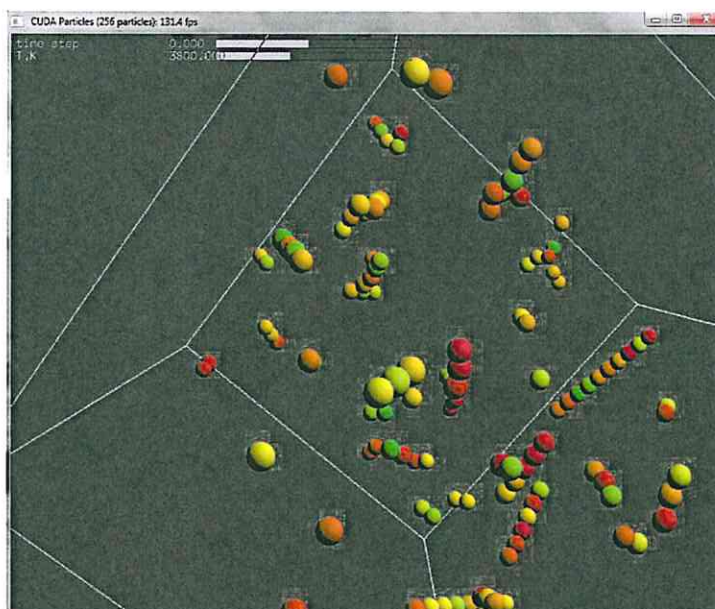


Рисунок 2.27. Вид листа графена при $T=3520\text{K}$, $\Delta H_3 = 2,25\text{E}+07$ Дж/кг

Сообщалось о моделировании цепной передачи углеродных нанотрубок [64], где исследовалась приложенная сила вращения одной шестерёнки и реакция второй. Такая модель получила название «горячих концов», что приводит к тому, что приложенная сила вращения больше, чем измеренная сила вращения этой шестерёнки. Оказалось, что есть величины скоростей вращения, при которых они совпадают у

ведущей и ведомой, а также есть скорости, при которых происходит проскальзывание шестерёнок.

Проведено отображение на экране подобных шестерёнок, состоящих из 580 атомов углерода и 40 атомов водорода (рис 2.28) в системе с холодными концами. В таком случае приложенная сила вращения равна измеренной у ведущей шестерёнки (рис. 2.29). Для удержания ведомой трубки использовалась виртуальная пружина, действующая с силой 1 eV/Angstrom на краях. Скорость вращения изменялась ступеньками 0.05, 0.1, 0.2, 0.05 оборотов за пикосекунду. Вращение трубок происходит без проскальзывания до 200 ГГц при средней температуре 135К.

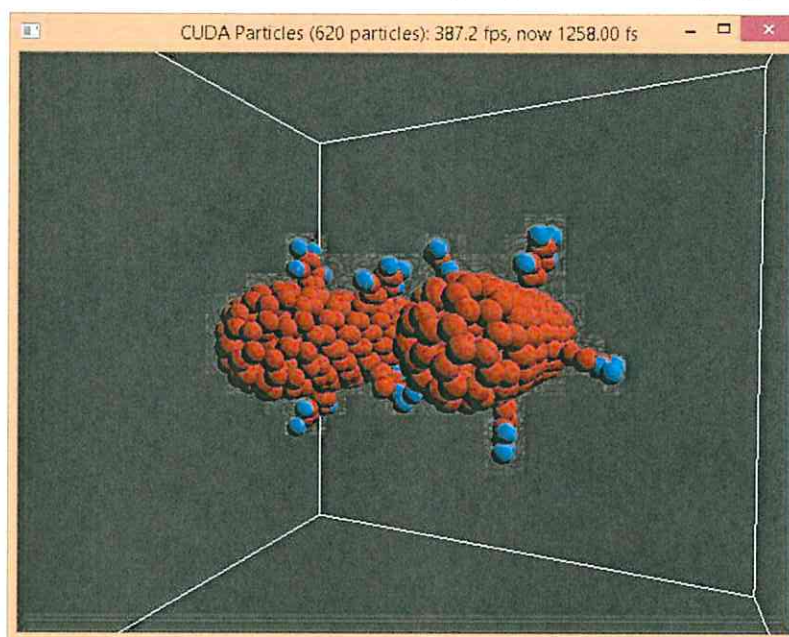


Рисунок 2.28. Закрытые углеродные нанотрубки с бензольными зубцами

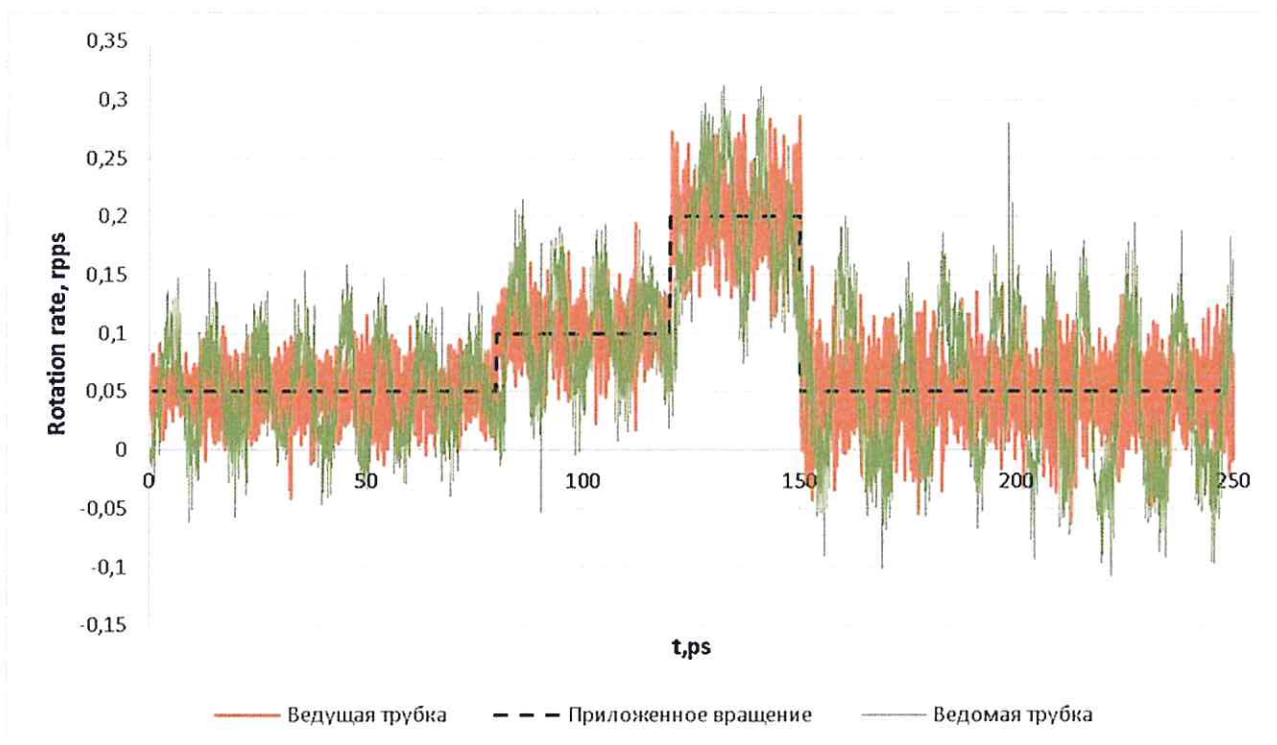


Рисунок 2.29. Скорость вращения и энергия для моделирования цепной передачи с контролем температуры

Визуализация проводилась на аппаратном обеспечении Intel Core i5-4200U, 1,6 ГГц, 2 ядра, 12 Гб RAM, nVidia GeForce 840M под управлением ОС Windows 8.1.

Подобное графическое отображение показывает возможность анимации материалов с механически изменяющейся структурой, что позволит видеть покрытия или механизмы, реагирующие на уровне атомов в зависимости от необходимости в реальном времени.

Выводы к главе 2

В главе 2 представлена технология программирования молекулярно-динамического моделирования различных наноструктур с использованием потенциала Бреннера на графических процессорах. Программирование на GPU сталкивается с трудностями прямого представления математических формул на параллельных вычислителях. Предлагается метод реализации численных алгоритмов решения уравнений движения, в которых используется явная разностная схема и один временной шаг реализуется так, чтобы не возникало ветвлений кода. Представленная

технология выбора алгоритмов молекулярной динамики обеспечивает эффективное исполнение на видеокартах.

Предложены подходы к повышению эффективности параллельных вычислений на графических процессорах, включающие использование гибридной модели поиска ближних частиц. Применение такой модели позволяет достичь одновременно и вычислительной сложности $O(N)$ в наиболее затратной по времени части программного комплекса и независимости параллельных потоков по памяти. Общие положения отображения вычислительной модели на графические вычислители создают основу технологии программирования алгоритмов молекулярно-динамического моделирования наноструктур, а также алгоритмов из класса задач взаимодействия N тел.

Разработаны эффективные алгоритмы параллельного программирования, для которых были созданы разноуровневые классы массивов для хранения и передачи информации на GPU.

В работе предложен эффективный подход к отображению потенциала межатомного взаимодействия на GPU. Основной задачей разработки алгоритма вычисления потенциала Бреннера на GPU является обеспечение наиболее эффективного доступа к памяти. Для этого выделяется дополнительная память и переменные, которые хранят копии координат до 8 соседних атомов. Из функции вычисления сил выносятся расчет коэффициентов, так что вычислительно сложная часть операций перенесена в специальную функцию. Затратная по времени функция составления списка соседних атомов оперирует значениями хеш-функции и дополнительной памятью, после построения списка расположение координат атомов, оказывается выровненным в памяти и не возникает перекрёстного доступа на следующих этапах.

Проведено теоретическое исследование оценки вычислительной сложности реализованных алгоритмов и требуемых затрат памяти в зависимости от количества атомов и используемого числа процессоров.

Разработана архитектура программного комплекса для моделирования наноструктур, которая обеспечивает визуализацию полученных результатов в реальном времени.

Глава 3. Компьютерные эксперименты. Анализ эффективности распараллеливания вычислений

3.1 Верификация программного обеспечения

Проверка структурных параметров.

В таблице 3.1 собраны параметры графена, рассчитанные с использованием разработанной программы, в сравнении с результатами, полученными в других лабораторных и вычислительных экспериментах.

Таблица 3.1. Сравнение структурных параметров графита (графена)

Параметр	Бреннер	Работа [65]	Экспериментальное значение	Рассчитано
$r_{CC}, \text{Å}$	1.420	1.396	1.415 [66]	1.420
Энергия связи, эВ	7.380	7.471	7.454 [66]	7.429
Энергия образования вакансии, эВ	7.5	7.7	7.6 [67]	7.485

Сравнение с аналитическим решением системы двух атомов.

На рис. 3.2 наблюдается совпадение аналитического решения с численным решением для системы двух атомов с использованием потенциала Бреннера. Соответствующая энергия взаимодействия приведена на рис. 3.1.

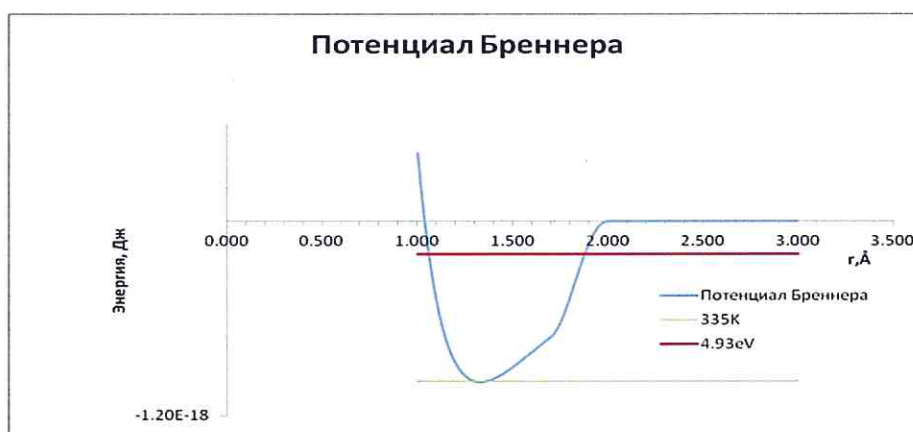


Рисунок 3.1. Энергия двухатомного взаимодействия в потенциале Бреннера

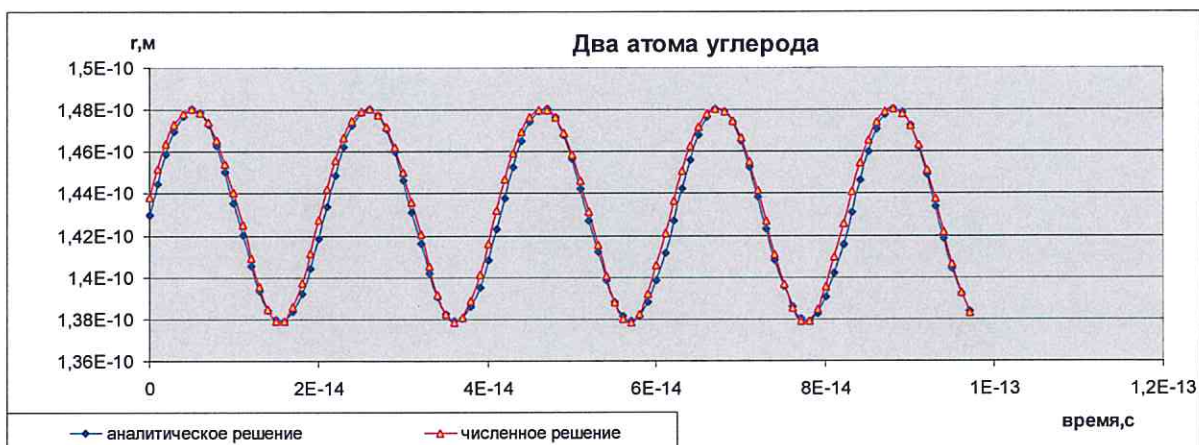


Рисунок 3.2. Колебание расстояния между двумя атомами углерода

Энергия одной связи атомов углерода в графене отмечена красной широкой прямой линией. Зелёной тонкой прямой линией отмечена энергия, соответствующая температуре системы атомов.

Сравнение расчёта температуры с решением LAMMPS.

В общем случае кинетическая энергия системы должна сохраняться. В молекулярном моделировании температура поддерживается искусственной добавкой в уравнения движения. Для корректного вычисления температуры на основе кинетической энергии в нестационарном молекулярно-динамическом процессе в образце необходимо выделить составные части кинетической энергии. Соответствующий метод описывается в статье [68]. Кинетическая энергия разделяется на часть, получаемую из движения системы как твёрдого тела, и часть, получаемую из внутренних возмущений. Для правильного учёта температуры в зависимости от структуры материала требуется дополнительная калибровка. Для более точного подсчёта энергии движения, вибрации и возмущения необходимо рассматривать малые части образца и меньшее время моделирования.

Влияние образца на статистическую температуру рассматривается на примере листа графена типа «кресло» длиной 88 нм и шириной 20Å. Данный образец содержит 6720 атомов углерода.

Лист графена разбивается на полоски (slab) длиной 16 Å и шириной 20Å. Температура рассчитывается как средняя по времени и пространству величина (1.88).



Рисунок 3.3. Распределение температуры вдоль листа графена 88 нм, усреднённое за 10 фемтосекунд

Среднее значение температуры – 40.46K, среднеквадратичное отклонение – 4.73K, доля среднеквадратичного отклонения по отношению к среднему значению составляет 11.70%. При моделировании листа графена в LAMMPS получено среднее значение 41.68K, среднеквадратичное отклонение 3.07K, доля 7.38%.



Рисунок 3.4. Распределение температуры вдоль листа графена 88 нм, усреднённое за 20 фемтосекунд

Среднее значение – 40.53K, среднеквадратичное отклонение – 4.20K, доля – 10.37%. При моделировании листа графена в LAMMPS получено среднее значение 41.28K, среднеквадратичное отклонение 2.98K, доля 7.24%.

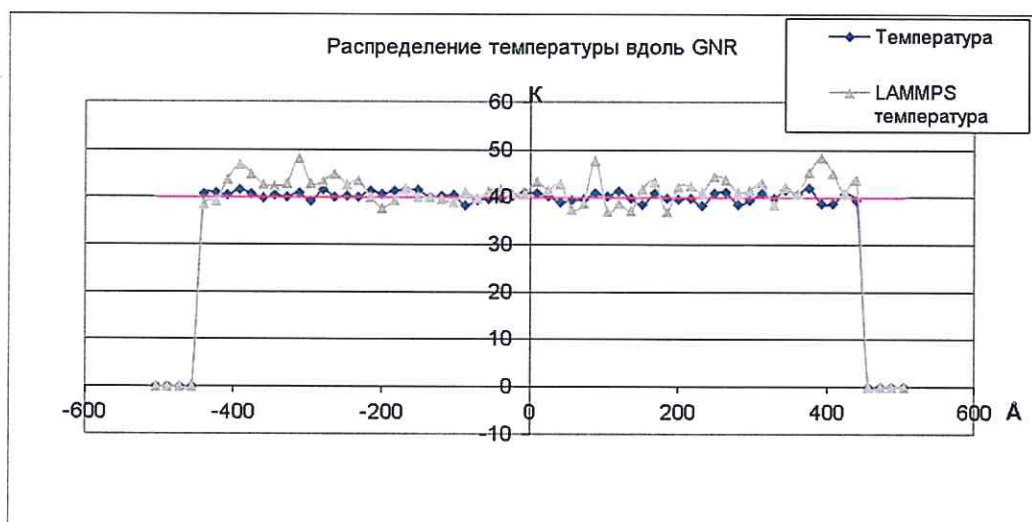


Рисунок 3.5. Распределение температуры вдоль листа графена 88 нм, усреднённое за 100 фемтосекунд
 Среднее значение – 40.23K, среднеквадратичное отклонение – 0.92K, доля – 2.29%. При моделировании листа графена в LAMMPS получено среднее значение 41.80K, среднеквадратичное отклонение 2.71K, доля 6.49%.

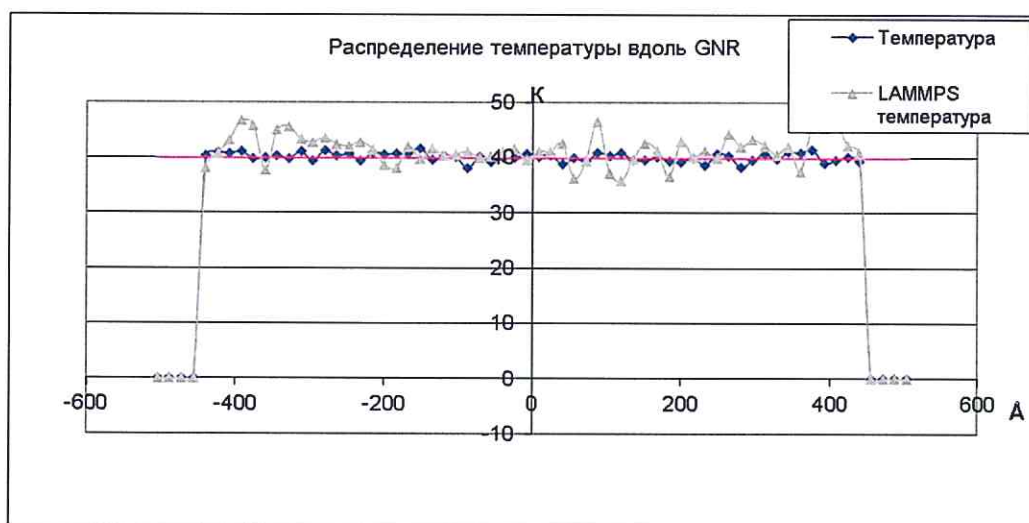


Рисунок 3.6. Распределение температуры вдоль листа графена 88 нм, усреднённое за 110 фемтосекунд
 Среднее значение – 40.18K, среднеквадратичное отклонение – 0.76K, доля – 1.91%. При моделировании листа графена в LAMMPS получено среднее значение 41.54K, среднеквадратичное отклонение 2.59K, доля 6.26%.

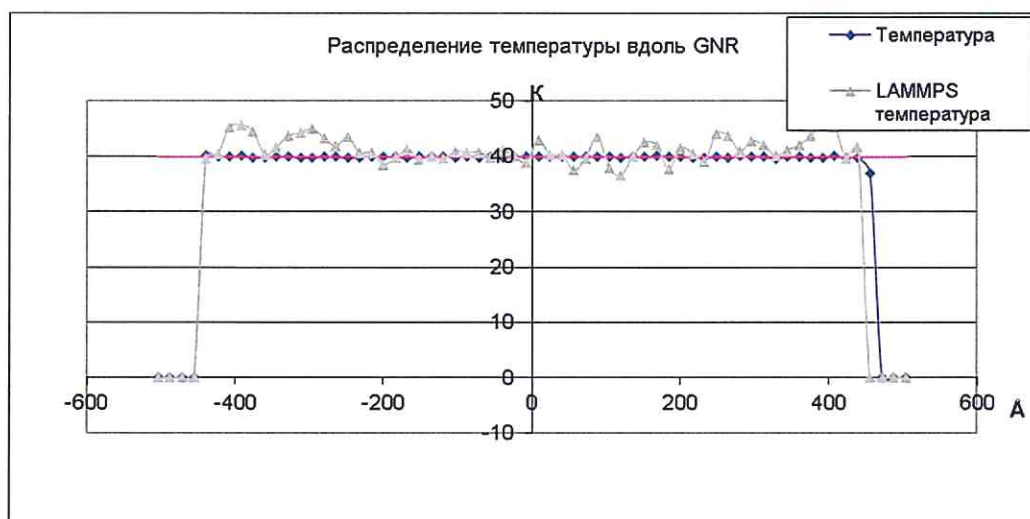


Рисунок 3.7. Распределение температуры вдоль листа графена 88 нм, усреднённое за 1000 фемтосекунд
 Среднее значение – 39.97К, среднеквадратичное отклонение – 0.10К, доля – 0.26%. При моделировании листа графена в LAMMPS получено среднее значение 41.43К, среднеквадратичное отклонение 2.19К, доля 5.30%.

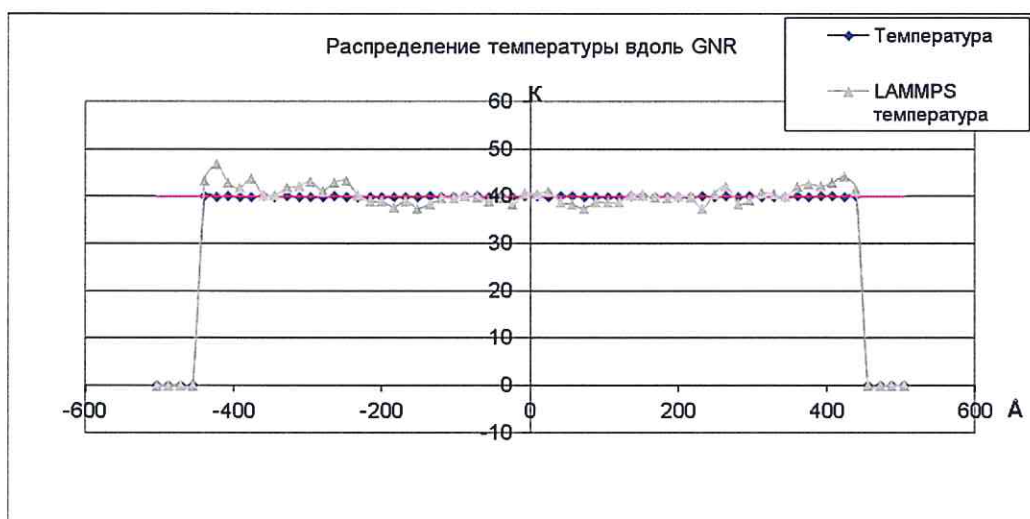


Рисунок 3.8. Распределение температуры вдоль листа графена 88 нм, усреднённое за 1010 фемтосекунд
 Среднее значение – 39.91К, среднеквадратичное отклонение – 0.07К, доля – 0.18%. При моделировании листа графена в LAMMPS получено среднее значение 40.46К, среднеквадратичное отклонение 1.92К, доля 4.75%.

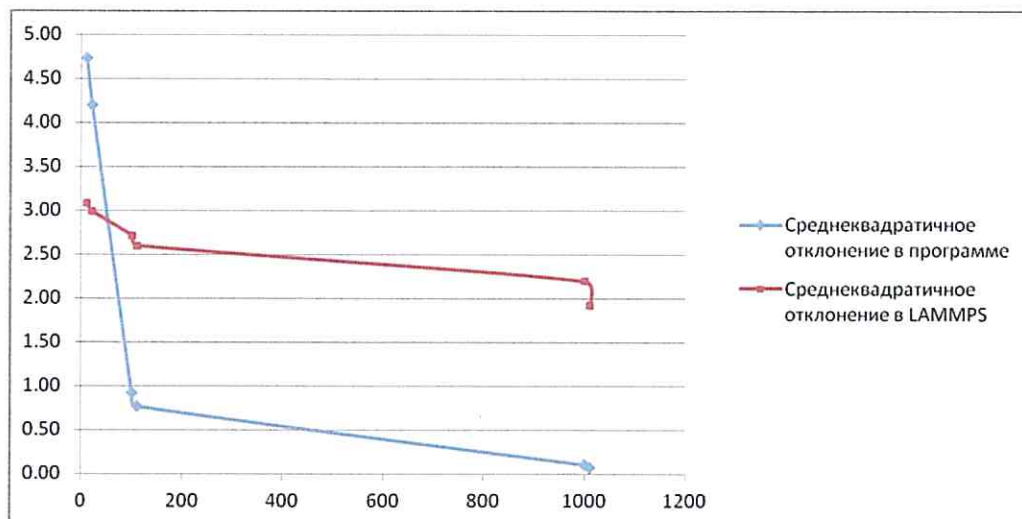


Рисунок 3.9. Уменьшение значения среднеквадратичного отклонения температуры по длине образца от времени усреднения в программе и в LAMMPS

Из рис. 3.9 видно, что увеличение периода усреднения по времени приводит к уменьшению колебаний вычисленного значения температуры по длине образца, причём можно видеть, что программное решение существенно быстрее приходит к заданной точности, чем решение LAMMPS.

3.2 Скорость вычисления многочастичных потенциалов на графических процессорах

Для определения производительности разработанного программного комплекса рассматривается актуальная задача моделирования теплопереноса в углеродных структурах. Важным элементом здесь является определение коэффициента теплопроводности. С этой целью по центру образца подводится тепловой импульс. Замеряется скорость распространения теплового импульса с течением времени. По характеру и скорости распространения определяется коэффициент теплопроводности.

В качестве примера рассматривается лист графена типа «кресло». В течение 1000 шагов лист доводится термостатом до средней температуры 16К. После стабилизации атомам по центру в полоске шириной 20\AA придаётся тепловой импульс 600К. Характерное распределение температуры показано на рис. 3.10. Осреднение проводится по 1000 временных шагов по области длиной 16\AA . Шаг моделирования 1 фемтосекунда. Проводится 10000 шагов моделирования. Выдерживается

микрoканонический ансамбль. Для моделирования листа на графических процессорах написана программа [105].

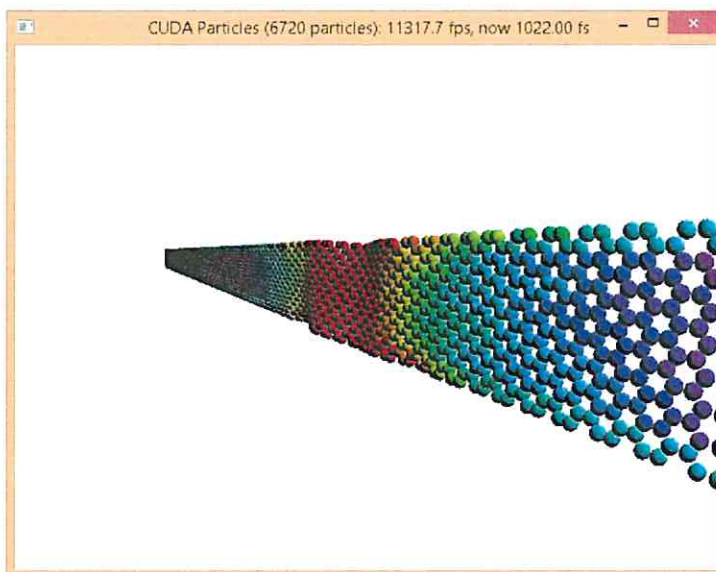


Рисунок 3.10. Распространение теплового импульса от центра листа графена

Молекулярно-динамическое моделирование проводилось на аппаратном обеспечении Intel Xeon CPU E5-2650, 2.00 ГГц, 32 ядра в ОС, 128 Гб RAM, NVIDIA Tesla M2075 (448 ядер), NVIDIA Quadro 4000 (256 ядер) под управлением ОС Centos 6.2. Максимальные значения количества вычислительных потоков в блоке: [1024, 1024, 64]. В таблице 3.2 и на рис. 3.11 показано время, необходимое компьютеру для моделирования образцов различного размера.

Таблица 3.2. Ускорение, достигнутое на GPU в задаче теплопереноса.

Длина образца, нм	Количество атомов	Время расчёта, с LAMMPS	Затраты памяти, Мб LAMMPS	Время расчёта, с Програма	Затраты памяти, Мб Програма	Ускорение, раз	Время расчёта, с Програма	Затраты памяти, Мб Програма
		1 ядро		Tesla M2075 (448 ядер)			Quadro 4000 (256 ядер)	
12,6	960	47,1	3,5	14,8	56,0	3,2	22,3	37,0
25	1920	99,9	5,5	25,6	58,0	3,9	28,9	38,0
50	3840	198,6	20,3	27,3	73,0	7,3	33,6	53,0
100	7680	407,4	134,1	37,6	72,0	10,8	58,8	54,0
200	15360	797,1	1026,9	62,8	189,0	12,0	113,0	170,0
400	30720	1661,9	1168,9	131,2	195,0	12,7	224,8	176,0
800	61440	3086,8	1230,1	276,3	207,0	11,2	524,1	188,0

1600	122880	9666,7	1389,8	524,0	1129,0	11,5	1002,4	1110,0
------	--------	--------	--------	-------	--------	------	--------	--------

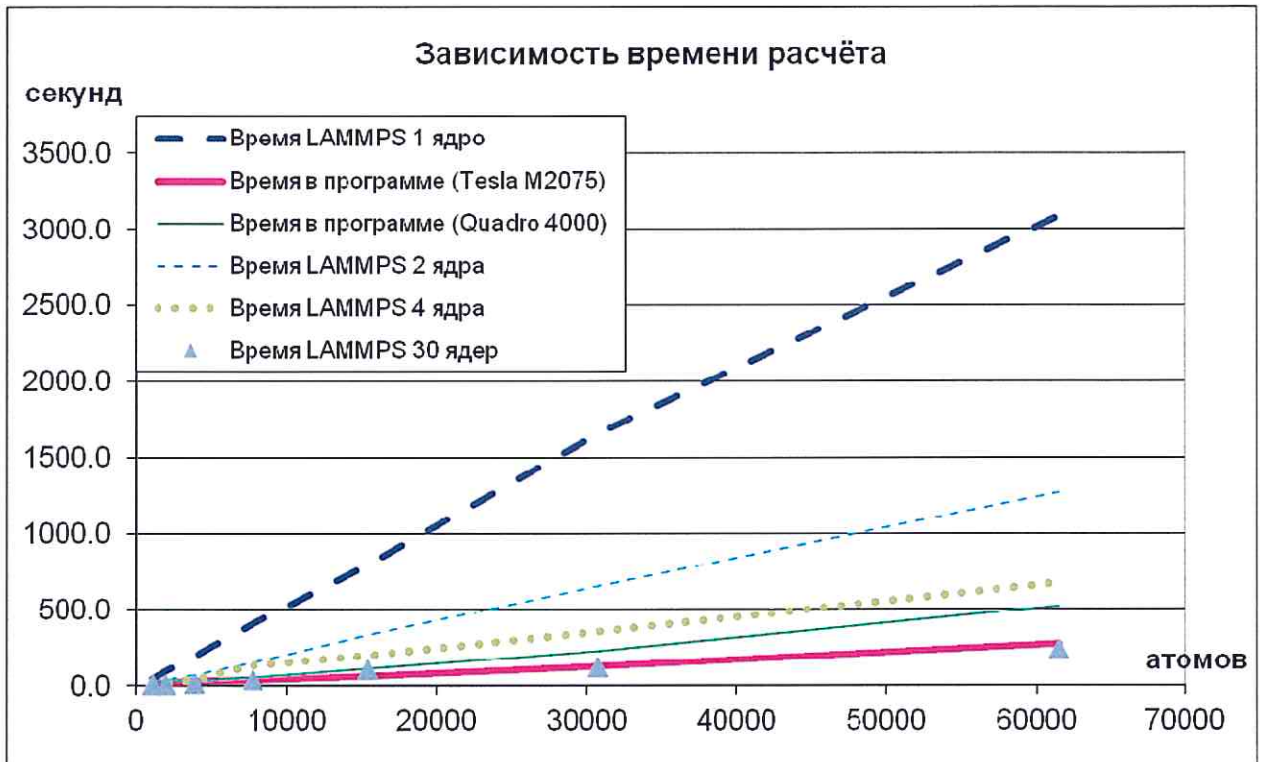


Рисунок 3.11. Зависимость времени расчёта от количества частиц

Представлены данные, полученные с помощью разработанного программного обеспечения и с использованием известного пакета молекулярно-динамического моделирования LAMMPS.

Можно видеть, что разработанный подход с использованием вычислений на графических процессорах одной видеокарты позволяет существенно повысить производительность в сравнении с вычислениями LAMMPS на многопроцессорных машинах вплоть до 32 ядер.

В данной работе исследовались разные методы представления пространства моделирования. Отдельные методы повышали скорость работы или уменьшали сложность [69] представления пространства.

На рис. 3.12 представлено сравнение ячеистой модели движения и гибридного алгоритма; квадратными и треугольными маркерами отмечено время расчёта коэффициентов по формуле (1.20). Ромбическими маркерами и крестиками отмечено время расчёта потенциала Бреннера по формуле (1.18) с известными коэффициентами b_{ij} . Видно, что время вычисления градиента потенциала Бреннера в случае ячеистой

модели росло в зависимости от количества атомов, тогда как в гибридном алгоритме оно постоянно. Для расчёта потенциала не требуются данные по трём и четырём атомам, эти коэффициенты уже известны, поэтому в обоих случаях время не зависит от количества атомов.

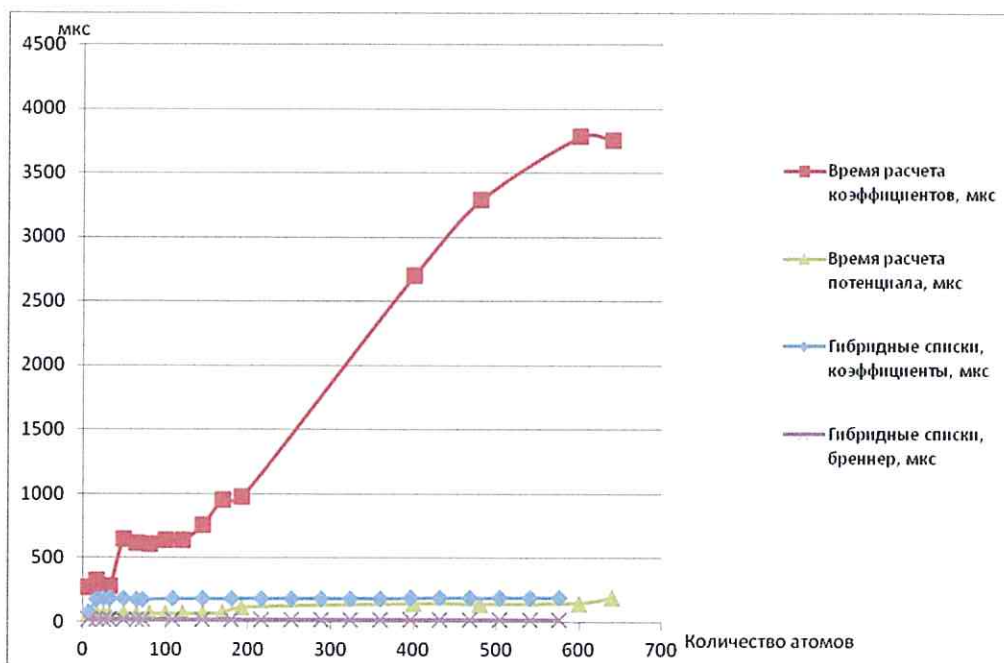


Рисунок 3.12. Время расчёта потенциала Бреннера и его градиента

На рис. 3.13 показана зависимость времени расчётов коэффициентов потенциала Бреннера и самого потенциала Бреннера и его градиента в случае использования гибридного алгоритма молекулярного моделирования. График имеет ступенчатый вид, что характерно для параллельного алгоритма. При увеличении количества частиц в 100 раз время расчётов увеличивается всего в 6 раз. Использовалось аппаратное обеспечение Intel Core i3, 2.93ГГц, 4 ядра, 4Гб RAM, nVidia GeForce GTX 480 (A) с программным обеспечением Windows 7 Pro Cuda 5.0 и Cuda 5.5.

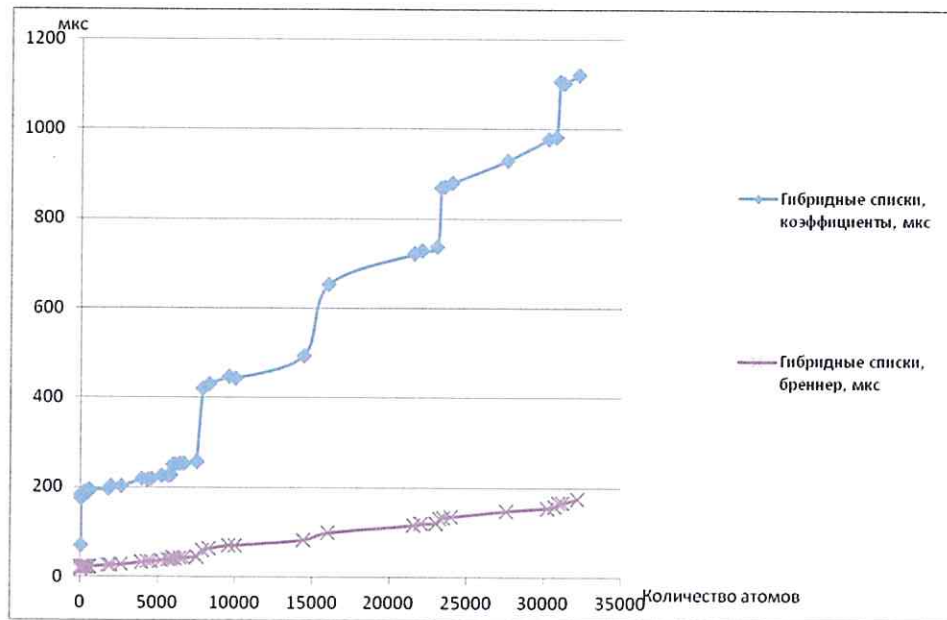
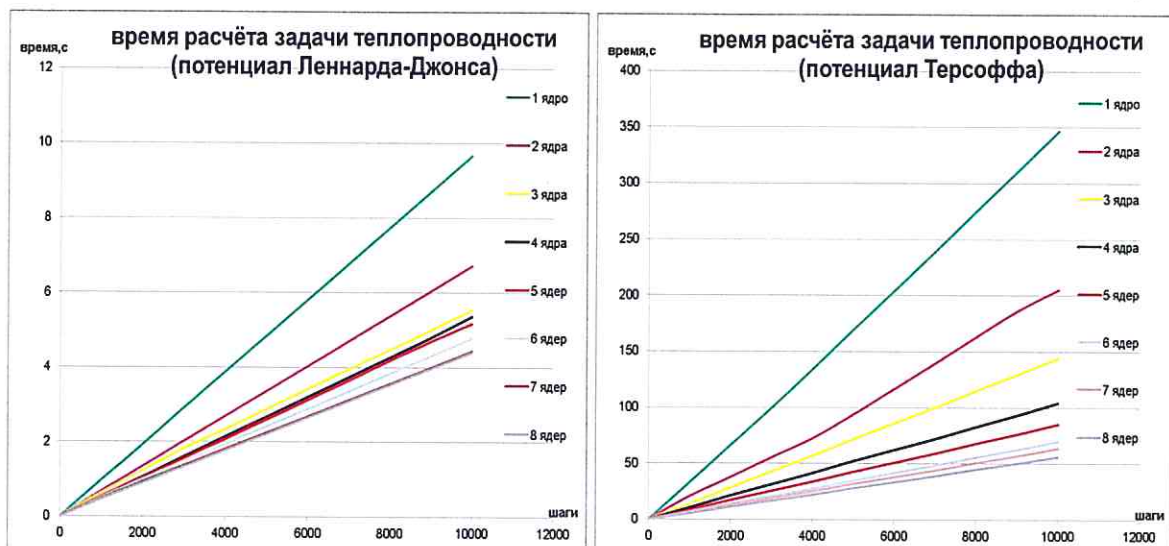


Рисунок 3.13. Время расчёта потенциала Бреннера и его градиента в параллельных вычислениях

Задача теплопроводности листа графена решалась на сервере Intel Xeon CPU E5-2650, 2.00 ГГц, 32 виртуальных ядра в ОС Centos 6, 128Гб RAM, NVIDIA Tesla M2075. Полученные результаты сравнивались с результатами моделирования на центральном процессоре с различным количеством потоков вычислений в программе LAMMPS (рис. 3.14). Также приведено сравнение времени расчёта в программе LAMMPS для различных потенциалов – Леннарда-Джонса, Терсоффа, Бреннера.



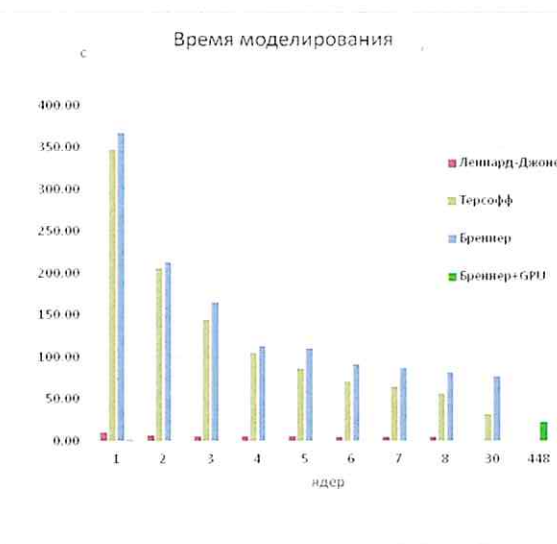
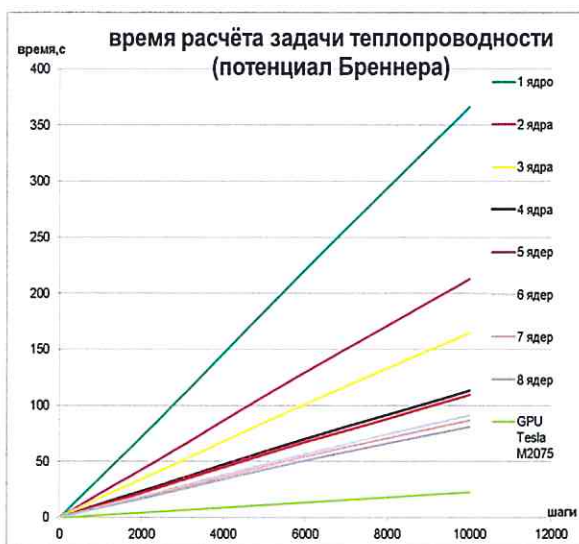


Рисунок 3.14. Время расчёта задачи теплопроводности при использовании потенциала Бреннера и др.

Метрики параллельных вычислений для решаемой задачи теплопроводности листа графена собраны в таблицах 3.3 – 3.5.

Таблица 3.3. Метрики производительности решения задачи теплопроводности в программе LAMMPS с использованием потенциала Леннарда-Джонса.

p	время,с	s (2.5)	e (2.10)	f (2.8)	Амдал (2.6)	Густафсон (2.9)	R(n) (2.11)	b (3.1)
1	9,67	-	-	-	-	-	694,78	-
2	6,73	1,44	0,72	0,39	1,44	1,61	999,12	1,50
3	5,54	1,74	0,58	0,36	1,69	2,22	1212,17	1,80
4	5,36	1,80	0,45	0,41	1,84	2,83	1253,02	2,00
5	5,18	1,87	0,37	0,42	1,95	3,44	1297,99	2,14
6	4,78	2,02	0,34	0,39	2,03	4,05	1405,85	2,25
7	4,44	2,18	0,31	0,37	2,10	4,66	1513,64	2,33
8	4,40	2,20	0,27	0,38	2,14	5,27	1526,20	2,40

Таблица 3.4. Метрики производительности решения задачи теплопроводности в программе LAMMPS с использованием потенциала Терсоффа.

p	время,с	s (2.5)	e (2.10)	f (2.8)	Амдал (2.6)	Густафсон (2.9)	R(n) (2.11)	b (3.1)
1	346,84	-	-	-	-	-	19,37	-
2	205,27	1,69	0,84	0,18	1,85	1,92	32,74	1,90
3	144,68	2,40	0,80	0,13	2,59	2,84	46,45	2,71
4	104,18	3,33	0,83	0,07	3,23	3,76	64,50	3,45
5	85,32	4,06	0,81	0,06	3,79	4,68	78,76	4,13
6	69,97	4,96	0,83	0,04	4,29	5,60	96,04	4,75
7	63,70	5,44	0,78	0,05	4,73	6,52	105,49	5,32
8	56,30	6,16	0,77	0,04	5,13	7,44	119,35	5,85
30	31,83	10,90	0,36	0,06	9,04	27,68	211,09	11,88

Таблица 3.5. Метрики производительности решения задачи теплопроводности в программе LAMMPS и в разработанной программе для вычислений на графических процессорах с использованием потенциала Бреннера.

p	время,с	s (2.5)	e (2.10)	f (2.8)	Амдал (2.6)	Густафсон (2.9)	R(n) (2.11)	b (3.1)
1	366,61	-	-	-	-	-	18,33	-
2	213,09	1,72	0,86	0,16	1,77	1,87	31,54	1,91
3	165,27	2,22	0,74	0,18	2,38	2,74	40,66	2,74
4	113,16	3,24	0,81	0,08	2,88	3,61	59,39	3,50
5	109,86	3,34	0,67	0,12	3,29	4,48	61,17	4,20
6	91,29	4,02	0,67	0,10	3,64	5,35	73,61	4,85
7	86,99	4,21	0,60	0,11	3,93	6,22	77,25	5,44
8	81,04	4,52	0,57	0,11	4,19	7,09	82,92	6,00
30	77,07	4,76	0,16	0,18	6,29	26,23	87,19	12,60
448 GPU	22,87	16,03	0,04	0,06	7,58	389,89	293,84	20,10

Из графиков рис. 3.14 видно, что если продолжить тенденцию увеличения количества процессоров, то время выполнения не будет уменьшаться пропорционально, более того, эффективность масштабирования резко снижается. Значительное улучшение производительности при увеличении количества ядер позволяет достичь использования графических процессоров.

Рассмотрим кристаллическую решётку графена на рисунке 3.15.

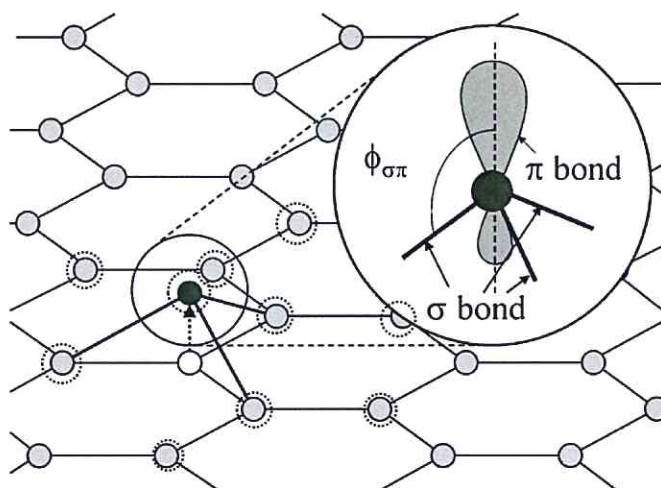


Рисунок 3.15. Кристаллическая решётка графена. Штриховыми кружками отмечены атомы для рассмотрения связей, используемых в формуле потенциалов

В потенциалах Леннарда-Джонса, Терсоффа, Бреннера используется выражение, зависящее от 2, 3, 4-х атомов соответственно. Один атом составляет 3 связи с соседними атомами, выбрать 1 из 3 атомов можно $C^b = C_3^1 = 3$ способами, что

является количеством сочетаний; для выбора угла, составленного из 3 атомов, из 6 возможных соседних есть $C^b = C_6^3 = 20$ способов. Так как в графене вычисление двугранных углов не даёт значительный вклад в сложность вычислений, то время моделирования потенциала Бреннера имеет тот же порядок, что и потенциал Терсоффа. В результате можно составить формулу ускорения, учитывающую информацию о структуре потенциала, в виде:

$$b = \frac{1}{\frac{1}{C^b} + \frac{C^b - 1}{C^b p}}, \quad (3.1)$$

где C^b – количество способов составить связь или угол в потенциале.

Выражение (3.1) предлагается считать авторской метрикой для задач молекулярно-динамического моделирования. В таблицах 3.3 – 3.5 видно, что такое выражение ускорения параллельных вычислений в зависимости от сложностей связей даёт более точное представление об эффективности использования множества ядер.

3.3 Измерение производительности вычислений

3.3.1 Схема компьютерного эксперимента

В данном разделе представлены результаты молекулярно-динамического моделирования на аппаратном обеспечении Intel Core i5-4200U, 1.6 ГГц, 2 ядра, 12 Гб RAM, NVIDIA GeForce 840M (384 ядра) под управлением ОС Windows 8.1. Максимальные значения количества вычислительных потоков в блоке: [1024, 1024, 64].

В качестве примера рассматривается материал графен, который находит все большее применение в различных отраслях современной индустрии. Моделирование позволяет подбирать необходимые размеры листов и структуру, подходящую для прикладных задач и синтеза в лабораториях. Влияние количества потоков в элементарном вычислительном блоке на время расчёта в зависимости от этапа показано в таблице 3.6 (где есть выбор), время измерялось в задаче моделирования листа графена, содержащего 6720 атомов.

Таблица 3.6. Время выполнения функций (микросекунды) в зависимости от параметров инициализации GPU.

Этап	Количество потоков в блоке					
	[32,1,1]	[64,1,1]	[128,1,1]	[256,1,1]	[512,1,1]	[1024,1,1]
Обновление координат и скоростей по алгоритму скоростей Верле	-	-	20	-	-	-
Вычисление хеш-функции	21	23	23	23	24	23
Выравнивание данных	68	75	71	75	72	73
Сортировка	-	-	-	-	-	140
Формирование списка соседних атомов	2795	2600	2600	2855	2820	3045
Вычисление коэффициентов потенциала	2630	2640	2650	2680	3165	-
Вычисление сил, ускорений	260	260	262	263	269	-

В таблице 3.7 подобраны оптимальные параметры конфигурации запуска вычислений на графических процессорах для рассмотренной задачи. Для каждой функции устанавливаются независимые параметры конфигурации. Значения конфигурации соответствуют логическому представлению вычислительных потоков в блоке. Реализованные методы выравнивания памяти и использование объединённых переменных позволяют задавать одномерную конфигурацию блоков, что имеет более простое логическое представление данных.

Таблица 3.7. Оптимальные конфигурации запуска вычислений на графических процессорах.

Этап	Конфигурация вычислительных потоков	Максимальный прирост эффективности, %	Доля вычислительного времени на одном шаге, %
Обновление координат и скоростей по алгоритму скоростей Верле	[128,1,1]	-	0.4
Вычисление хеш-функции	[32,1,1]	12.5	0.4
Выравнивание данных	[32,1,1]	9.3	1.2
Сортировка	[1024,1,1]	-	2.3
Формирование списка соседних атомов	[128,1,1]	14.6	43.1
Вычисление коэффициентов потенциала	[32,1,1]	15.3	44.1
Вычисление сил,	[32,1,1]	3.3	4.3

ускорений			
-----------	--	--	--

Здесь же представлены показатели затрат времени функций в процентах от общего времени одного шага для образца из 6720 атомов. Из таблицы видно, что 91% времени требуется для расчёта потенциала и обновления списка соседних атомов. Отметим, что сортировка – единственный из списка этапов, который имеет вычислительную сложность $O(N \log N)$, но он даёт незначительный вклад в общее время вычислений.

Для процедуры обновления координат подбирается максимальное количество вычислительных потоков в блоке. Подбор оптимального размера элементарного блока вычислений для каждой процедуры, составляющей один этап моделирования, позволяет ускорить общее выполнение программы на 13.3%.

3.3.2 Сравнение эффективности решения задач на различных GPU

Правильное конфигурирование вычислительных потоков на видеокарте даёт многократный прирост производительности. Необходимо распределять данные определённым образом для того, чтобы потоки исполнялись параллельно. Прямое вычисление потенциала многочастичного взаимодействия не всегда позволяет эффективно использовать графические вычислительные ресурсы. Сочетание техник молекулярной динамики позволяет достичь ускорения работы программы моделирования наносистем.

На рис. 3.16 показано влияние программного обеспечения на скорость моделирования графена на стандартном аппаратном обеспечении Intel Core i3, 2.93 ГГц, 4 ядра, 4 Гб RAM, NVIDIA GeForce GTX 480, 1.40 ГГц, 1536 Мб с программным обеспечением Windows 7 Pro CUDA 5.0 и CUDA 5.5. Видно, что программное обеспечение CUDA влияет неоднородно на скорость расчётов: время расчётов потенциала Бреннера увеличилось на 55%, время расчётов коэффициентов увеличилось на 4%. Также можно отметить наблюдающуюся линейную зависимость затраченного времени от количества частиц, хотя в общем случае для расчёта коэффициентов полным перебором требуется $O(N^4)$ операций.

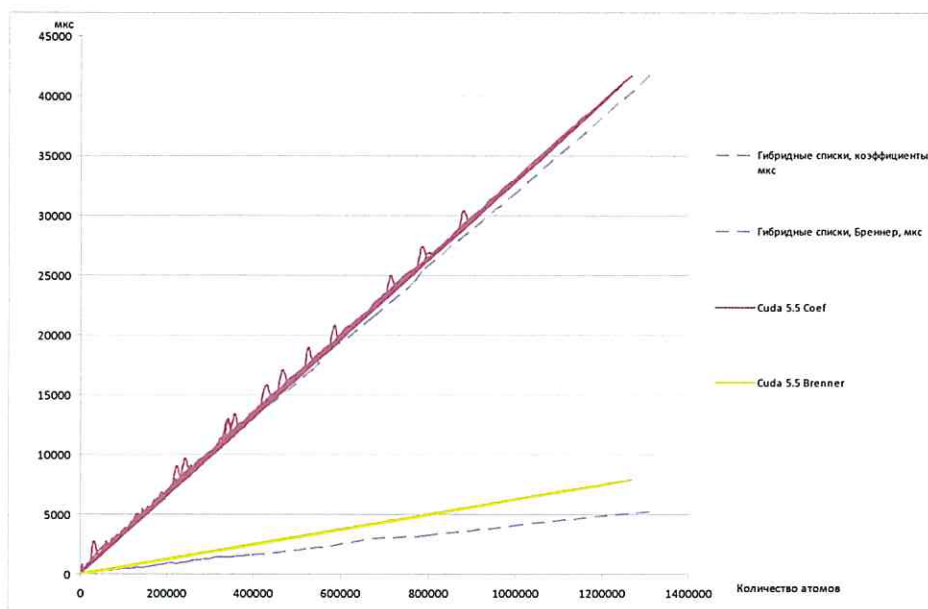


Рисунок 3.16. Время расчёта потенциала Бреннера и его градиента при разных версиях CUDA 5.0 и 5.5

На рис. 3.17 показано сравнение производительности стандартного аппаратного обеспечения с системой Intel Core i7 2600, 3.4 ГГц, 8 ядер, 8 Гб RAM, NVIDIA Quadro 4000, 0.95 ГГц, 1984 Мб с программным обеспечением Windows 7 Enterprise Cuda 5.5. Видно, что аппаратное обеспечение тоже значительно влияет на скорость расчётов: несмотря на повышение мощности CPU время расчётов потенциала Бреннера увеличилось в 2.21 раза, время расчётов коэффициентов увеличилось в 2.73 раза.

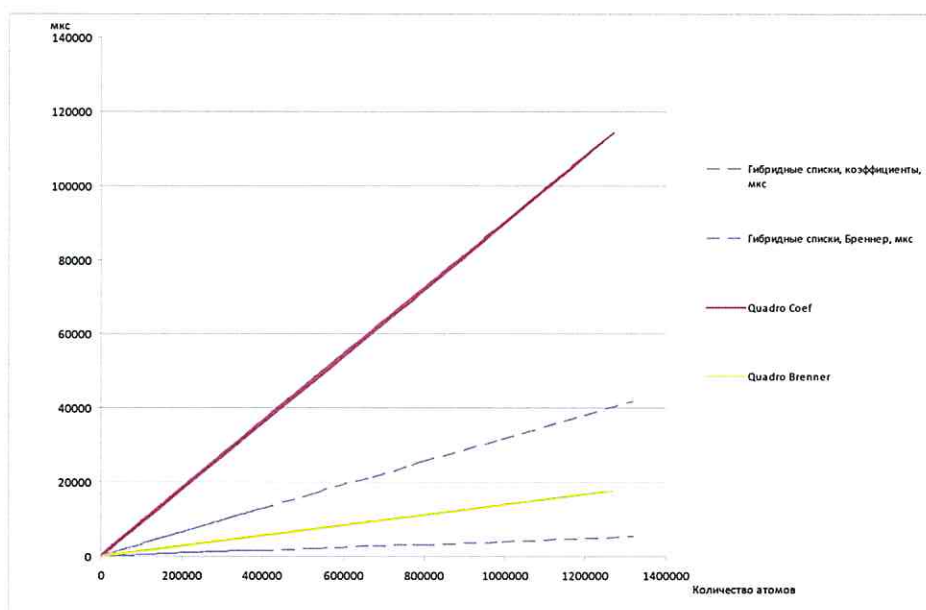


Рисунок 3.17. Время расчёта потенциала Бреннера и его градиента на разных видеокартах GTX480 и Quadro4000

На рис. 3.18 показано сравнение производительности стандартного аппаратного обеспечения и Intel Xeon E5-2650 2.00GHz, 32 ядра, 128Гб RAM, NVIDIA Tesla M2075, 1.15 GHz, 5301 Мб с программным обеспечением CentOS 6.2 Cuda 5.5. Вновь время расчётов потенциала Бреннера увеличилось в 1.36 раза, время расчётов коэффициентов увеличилось в 1.62 раза.

Можно сделать вывод, что в аппаратном обеспечении на скорость расчётов влияет производительность одного графического ядра, а не мощность и количество CPU. Объём графической памяти следует учитывать только при увеличении размера задачи.

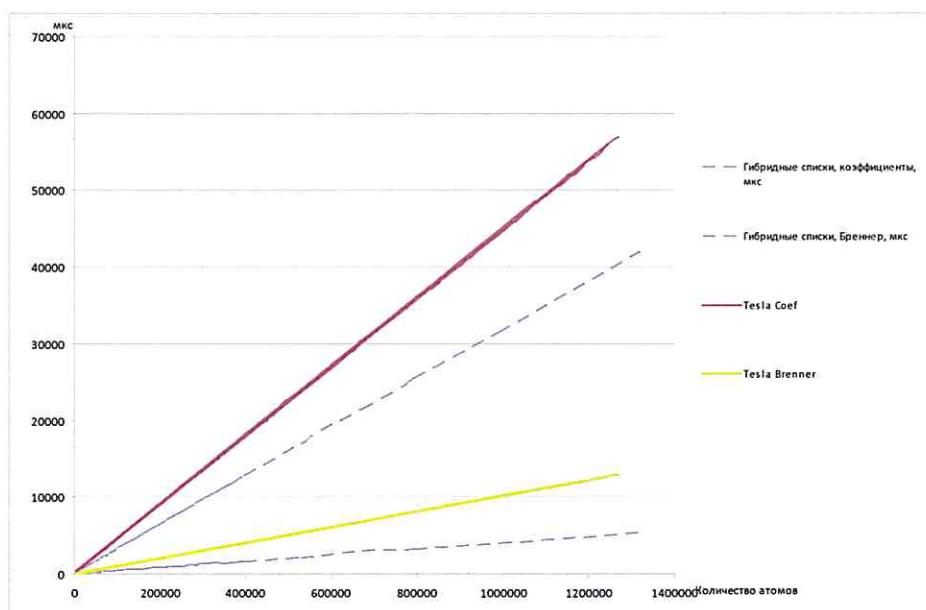


Рисунок 3.18. Время расчёта потенциала Бреннера и его градиента на разных видеокартах GTX480 и Tesla4000

На рис. 3.19 показано увеличение графика с рис. 3.18 в начальной области, где можно видеть ступенчатую структуру графика, что характерно для параллельных вычислений. Для разных вариантов аппаратного обеспечения ширина ступенек получается одинаковой, порядка 7300 атомов для задачи молекулярного моделирования графена.

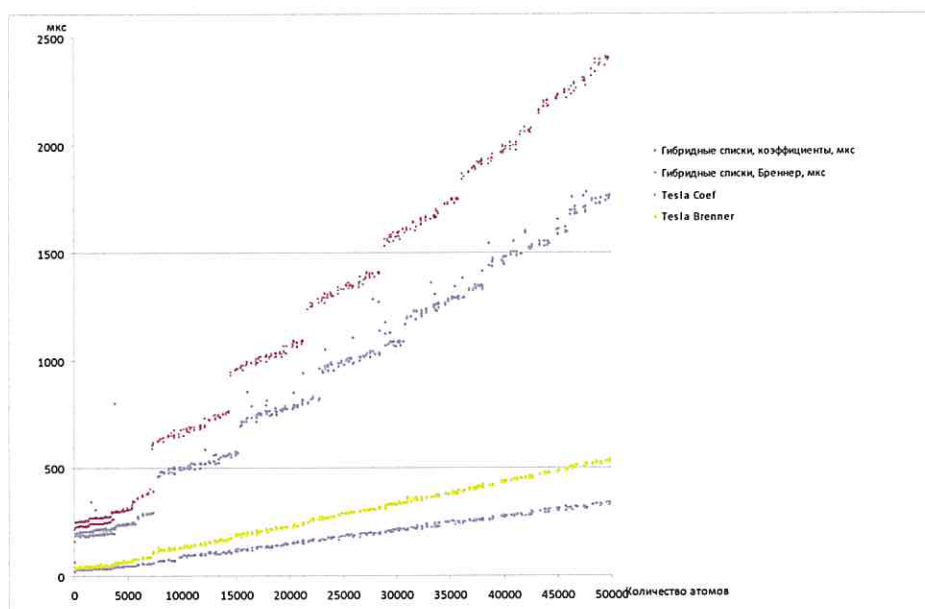


Рисунок 3.19. Время расчёта потенциала Бреннера и его градиента на разных видеокартах GTX480 и Tesla4000 (увеличение)

3.4 Решение задачи термостатирования

В настоящем разделе рассмотрены вопросы термостатирования. Эта задача актуальна для анализа режимов теплопроводности, который проводится в главе 4. Для примера рассматривались следующие материалы: сотовая ячейка графена (6 атомов углерода), лист графена размера $32.2 \times 18.0 \text{ \AA}^2$, фуллерен C_{60} , нанотрубка (14,0) длиной 25 \AA .

Область моделирования представляет собой камеру в виде куба с ребром 40 \AA , так что её объём равен $64 \cdot 10^3 \text{ \AA}^3$, в который уместаются отмеченные структуры (рис. 3.20). Граничные условия заданы неупругим соударением о виртуальные стенки камеры с коэффициентом отражения 0.5. Они не имеют большого значения, так как во время моделирования структуры представляли собой твёрдые вещества и не прикасались к стенкам камеры.

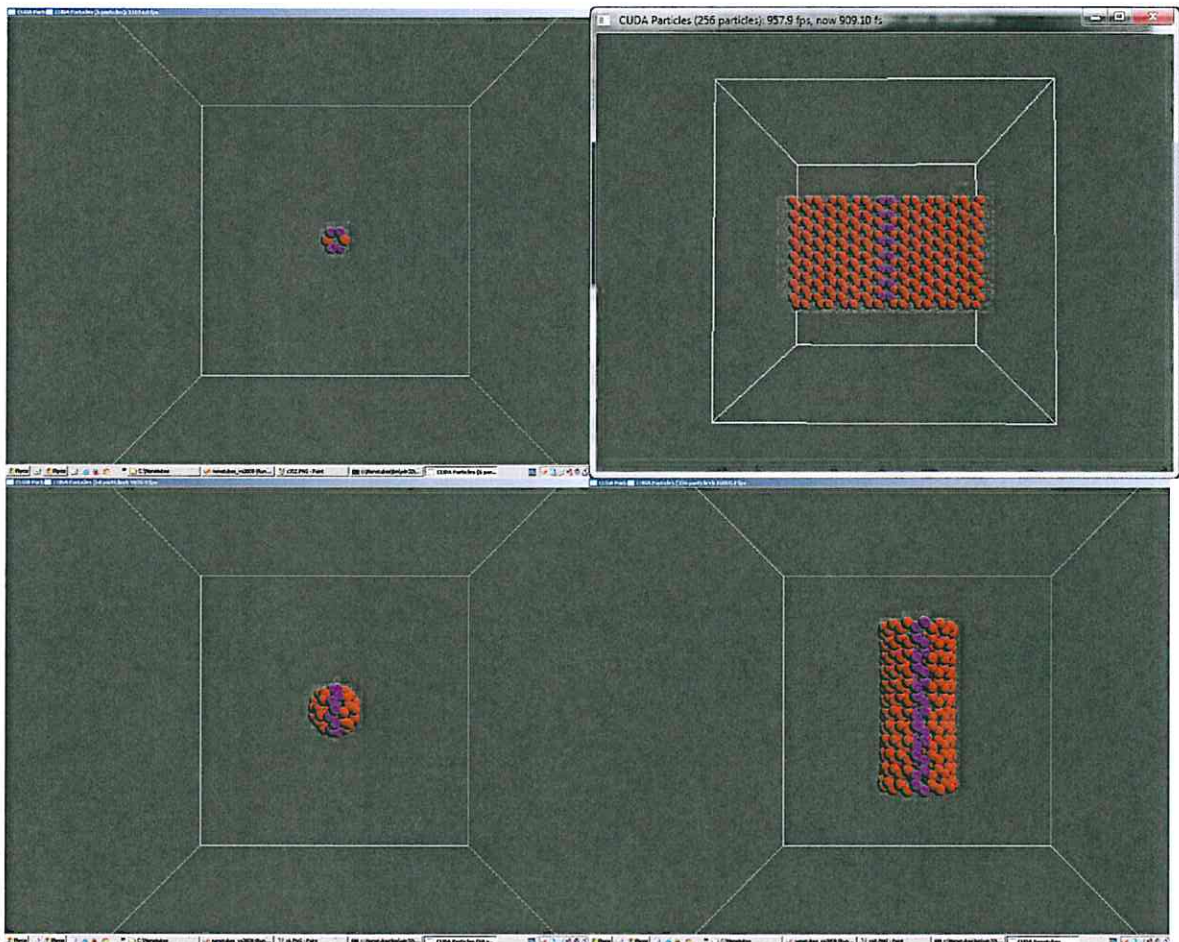


Рисунок 3.20. Образцы углеродных структур для моделирования: ячейка графена, лист графена, фуллерен, нанотрубка (14,0)

Для моделирования были выбраны 4 углеродных наноструктуры, задавались температуры 1К и 300К и выбраны два термостата. Исследовались термостаты Берендсена и Нозе-Хувера, так как они ограничивают нагрев молекулярной системы, препятствуя её распаду, и позволяют исследовать свойства теплопроводности.

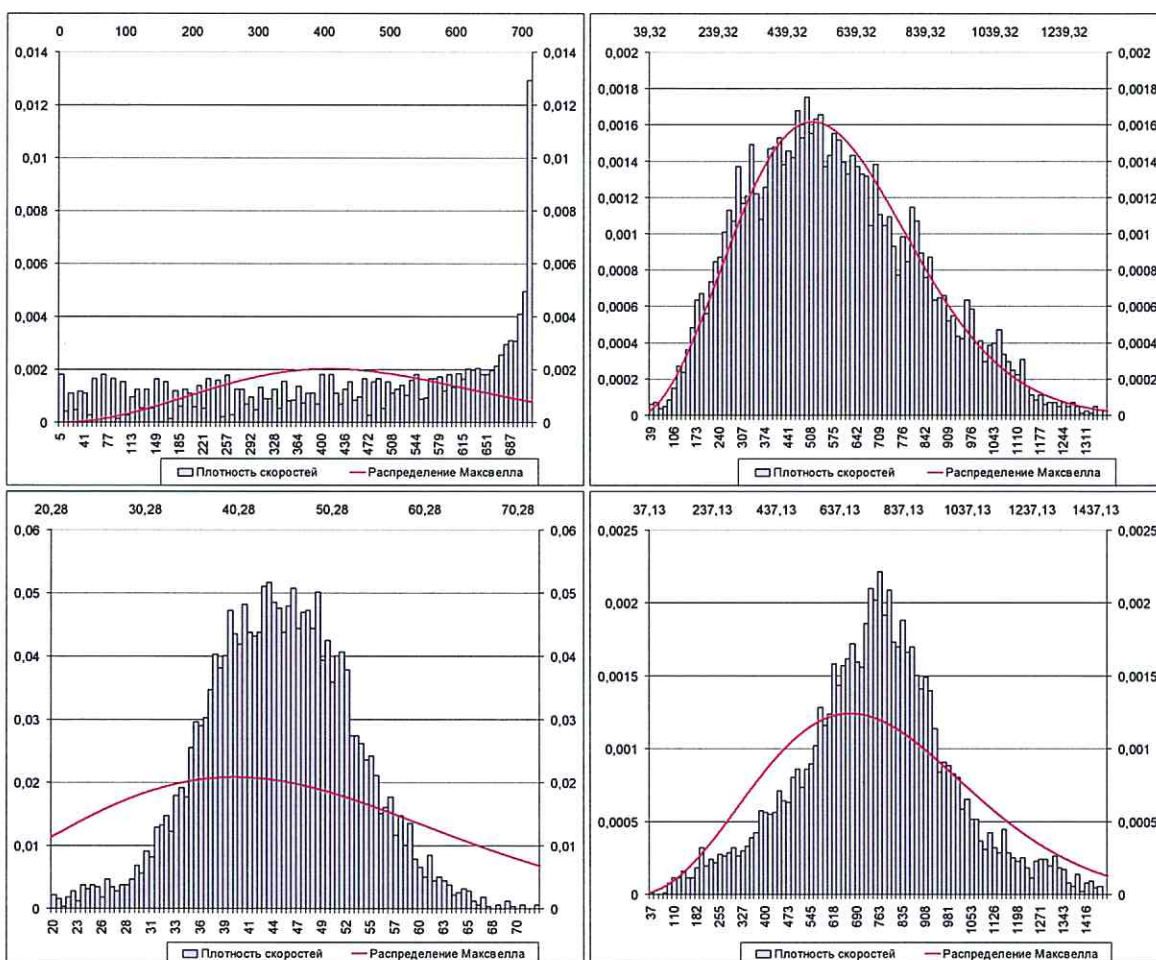
В процессе нагрева образец, ячейка графена из 6 атомов, не разрушается без контроля температуры, что позволяет задавать нулевые начальные скорости. Для остальных образцов заданы случайные начальные скорости для каждого атома в диапазоне $[-227, 227]$ м/с по каждой координате для трёх степеней свободы.

Температура системы атомов вычисляется по формуле (1.88). Параметр для термостата Берендсена из формулы (1.77) выбран как $\frac{t_r}{\Delta t} = 0.2$. Параметр подобран экспериментальным путём; сильно меньшие значения приводят к изменению формы атомных колебаний, а сильно большие значения приводят к таким большим

колебаниям температуры, которые могут разорвать связи атомов в твёрдом веществе. Параметр для термостата Нозе-Хувера из формулы (1.89) выбран как $\tau_{NH} = 100 fs$, он подбирался экспериментально по таким же соображениям.

Молекулярные системы моделировались 10000 шагов по 0.1 фемтосекунде до установления равновесного состояния, потом регистрировались скорости атомов в системе на следующих 10000 шагах моделирования. По ним построены плотности распределения модулей скоростей, и для сравнения нанесено распределение Максвелла с соответствующей температурой.

На рис. 3.21 изображены распределение скоростей (м/с) для ячейки графена из шести атомов углерода без контроля температуры, с применением термостата Берендсена, Нозе-Хувера для двух температур 1К и 300К.



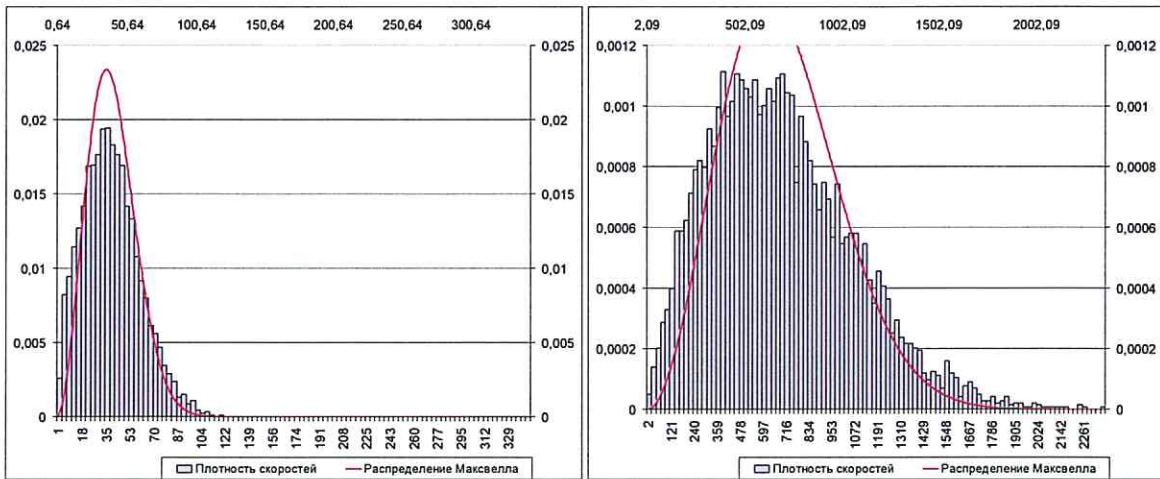
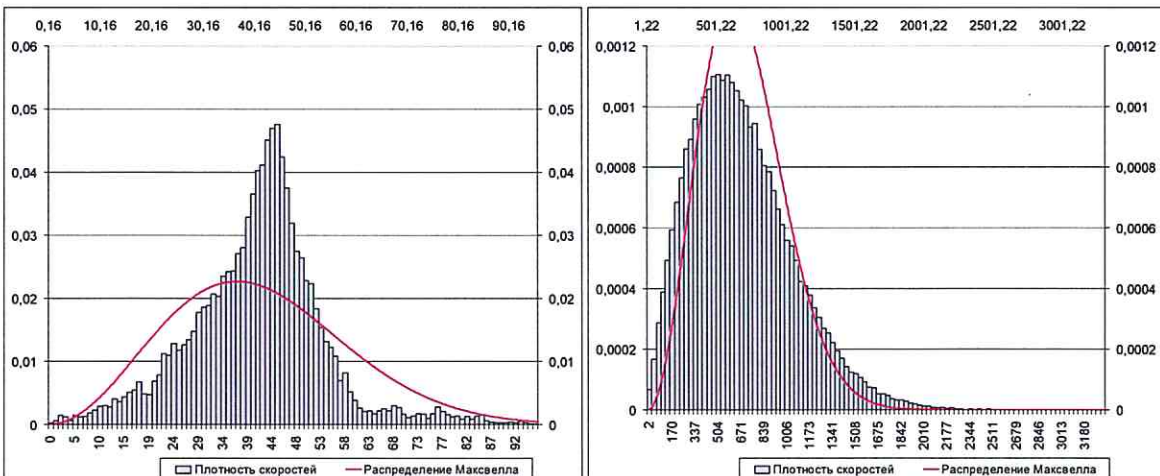


Рисунок 3.21. Распределение скоростей при моделировании ячейки графена

При температуре 1К и случайном задании скоростей без контроля температуры скорости распределяются практически равномерно с небольшим пиком в области больших скоростей. Значения скоростей всех атомов равны между собой на одном шаге моделирования. При температуре 300К и масштабировании скоростей без контроля температуры наблюдается максвелловское распределение скоростей, параметр температуры соответствует заданной. Следует отметить, что использование термостата Нозе-Хувера гораздо точнее задаёт распределение скоростей, чем термостат Берендсена.

На рис. 3.22 изображены распределения скоростей листа графена с применением термостата Берендсена и Нозе-Хувера для двух температур 1К и 300К. Распределения близки к распределению Максвелла, при этом математическое ожидание соответствует заданной температуре.



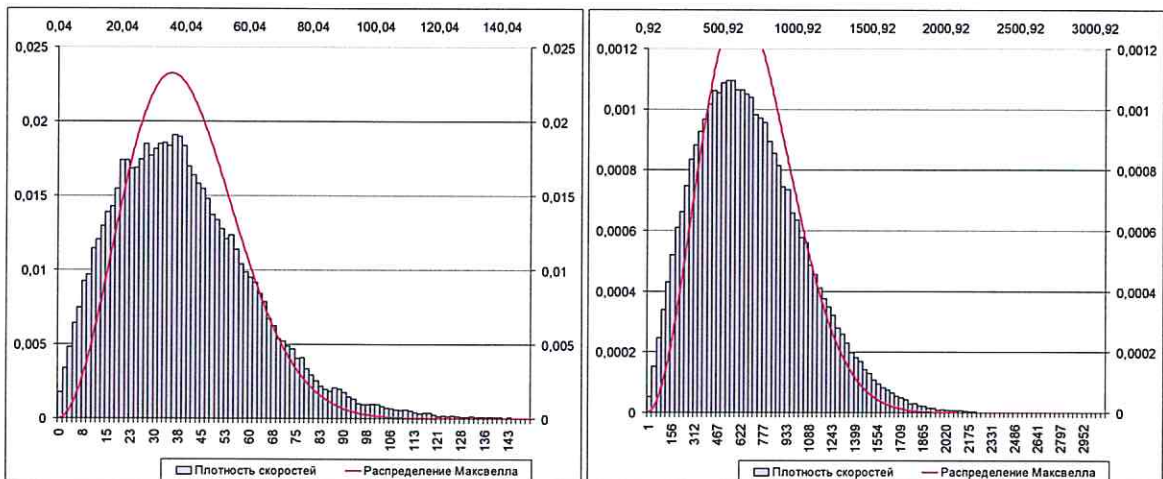
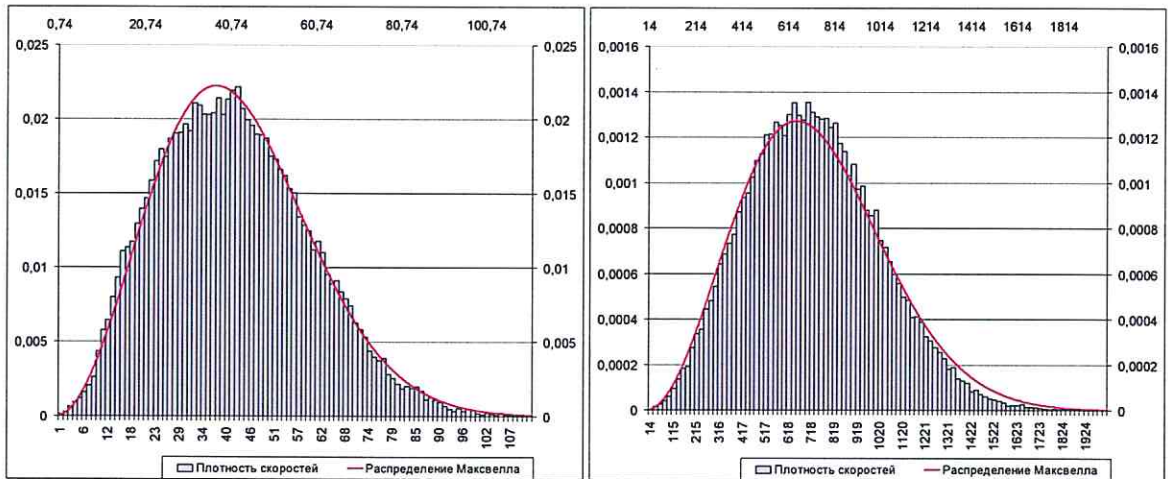


Рисунок 3.22. Распределение скоростей при моделировании листа графена

На рис. 3.23 изображены распределения скоростей фуллерена с применением термостата Берендсена и Нозе-Хувера для двух температур 1К и 300К. Распределения соответствуют распределению Максвелла, но при температуре 1К и использованном термостате Нозе-Хувера часть атомов ещё не приобрела энергии, достаточной для нагрева. Эффект быстро исчезает с повышением температуры.



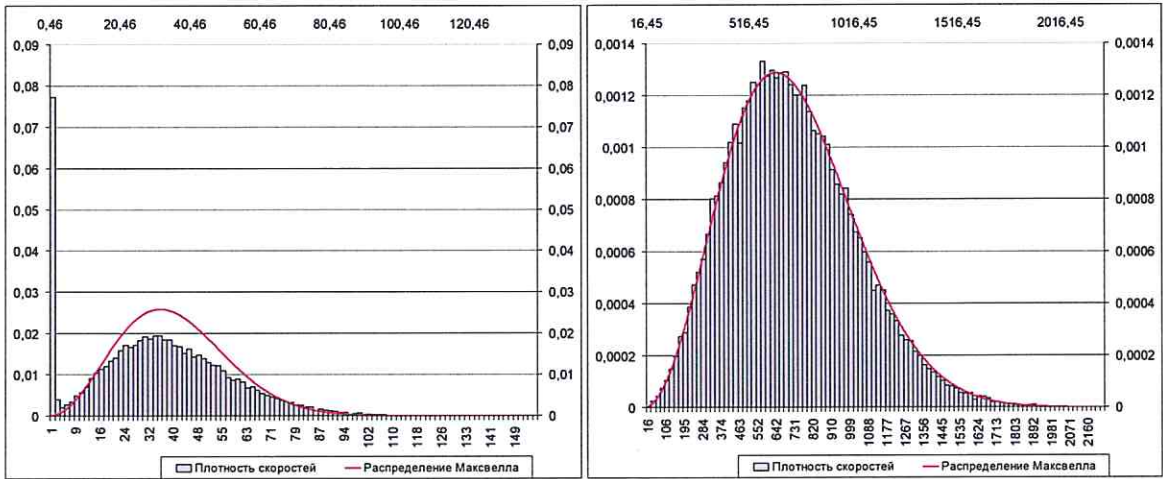


Рисунок 3.23. Распределение скоростей при моделировании фуллерена

На рис. 3.24 изображены распределения скоростей углеродной нанотрубки (14,0) с применением термостата Берендсена и Нозе-Хувера для двух температур 1К и 300К. Распределения при температуре 300К соответствуют распределению Максвелла, распределения при температуре 1К зашумлены.

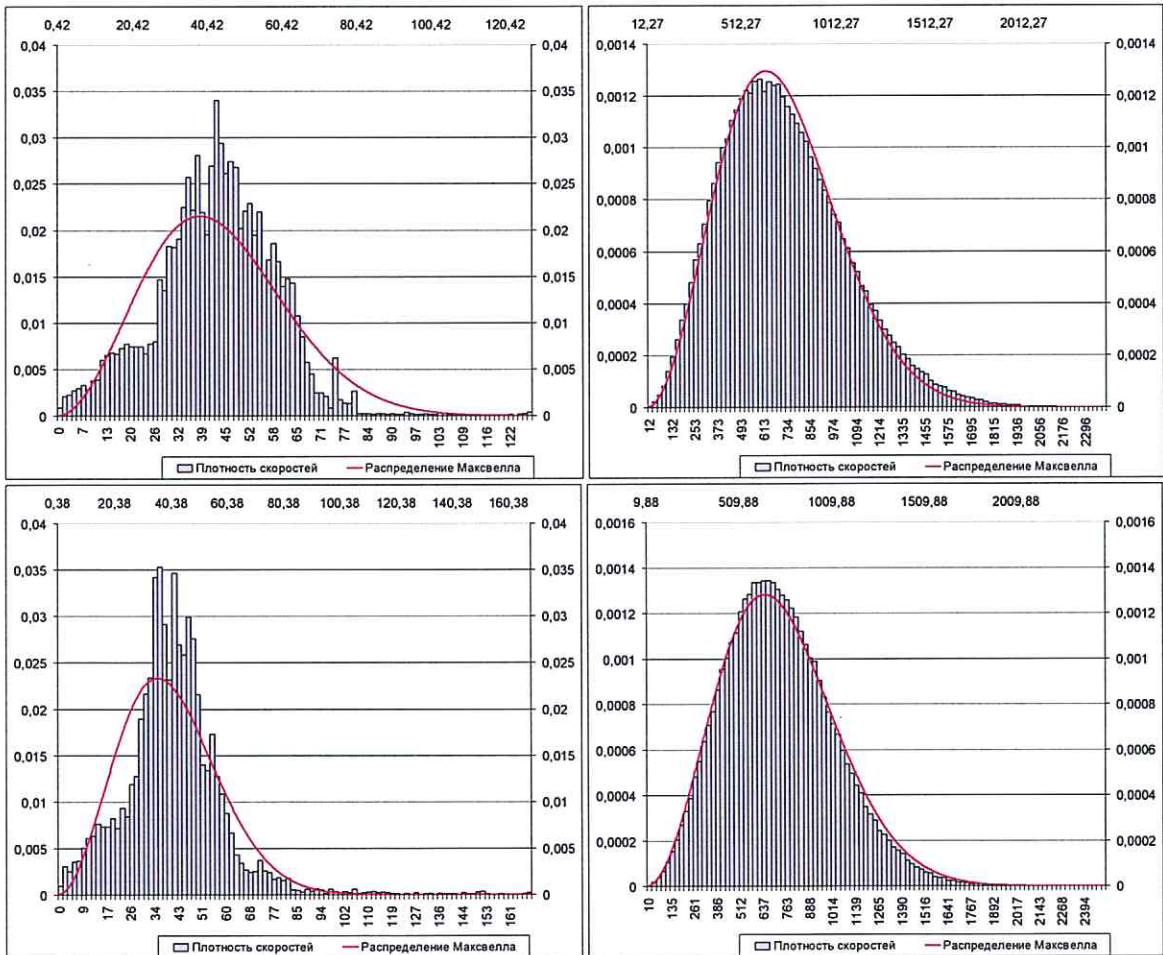


Рисунок 3.24. Распределение скоростей при моделировании углеродной нанотрубки (14,0)

Оба термостата приводят систему к заданной температуре и близкому к максвелловскому распределению скоростей. Термостат Берендесена использует один коэффициент для приведения скоростей к заданному значению. Термостат Нозе-Хувера поддерживает канонический ансамбль с флуктуацией скоростей. Использование термостатов позволяет ограничить колебание скоростей так, чтобы не происходило разрыва ковалентной связи. Рекомендуется для приведения системы к заданному уровню температуры использовать термостат Берендесена, для поддержания заданной температуры – использовать термостат Нозе-Хувера.

Выводы к главе 3

В главе 3 приводятся примеры различных вычислительных экспериментов, выполненных на программном обеспечении, разработанном в рамках предлагаемой технологии. Представлена эффективная реализация молекулярно-динамического моделирования со сложным потенциалом межчастичного взаимодействия на видеокартах.

С использованием разработанного программного комплекса проведено моделирование отдельных известных углеродных структур и вычислены их параметры. Результаты моделирования соответствуют результатам, полученным с использованием программного комплекса LAMMPS, а полученные значения параметров совпадают как с параметрами использованного потенциала, так и близки к значениям, полученным в других программных комплексах и лабораторных экспериментах. При этом следует отметить преимущества разработанного комплекса по скорости и точности моделирования.

Проанализировано влияние архитектуры GPU NVIDIA на скорость вычислений различных задач. Выявлено преимущество перед CPU. Показано, что для эффективных вычислений необходимо учитывать архитектуру графических процессоров и разбивать данные для размещения в различных видах памяти.

Проведено сравнение скорости вычисления потенциалов на разных моделях пространства, показано преимущество гибридной модели по сравнению с ячеистой моделью пространства.

Исследовано влияние скорости вычислений от версии программного обеспечения – на более новых версиях CUDA вычисления проходят быстрее.

Проведено сравнение скорости решения задачи распространения теплового импульса в листе графена для разработанной программы на различных графических процессорах, показано преимущество видеокарт с большей тактовой частотой вне зависимости от производительности центрального процессора.

Показано, что возможен общий прирост скорости моделирования при оптимальном выборе количества вычислительных потоков.

Исследованы вопросы термостатирования наноструктур в ходе молекулярно-динамического моделирования для получения профилей распределения начальных скоростей и температур.

Глава 4. Методы моделирования теплопроводности наносистем

4.1 Особенности теплопереноса в наноструктурах

Физика теплопереноса в наноструктурах отличается от классической и основывается на статистической теории, описывающей динамику молекулярных систем. Для наносистем введено определение специального режима – баллистическо-диффузионного [70], который частично описывает наблюдаемые отклонения. Указанный механизм позволяет учитывать необычное поведение теплопереноса в структуре в зависимости от длины, размера, дефектов, направления. С учётом названных параметров становятся ясными трудности, с которыми сталкиваются в практических измерениях коэффициентов теплопроводности. Данная проблема активно решается последние 25 лет, собрано множество данных, но полного представления о теплопереносе нет. Дальнейший прогресс технологии позволит больше изучить такие материалы, но вопрос теоретического описания данных процессов остаётся открытым.

Перенос тепла в CNT осуществляется за счёт взаимодействия фононов. При низких температурах наблюдается баллистический фононный транспорт. При комнатных температурах происходит переход от квазibalлистического к диффузионному режиму. MWNT показывают зависимость теплопроводности от температуры из-за Ван-дер-Вальсовых взаимодействий слоёв. Графеновые листы проявляют такие же баллистические эффекты переноса тепла [71].

Исследованием теплопроводности углеродных структур активно занимались в последнее время. Разрабатывались различные теории и экспериментальные методы, особое внимание уделялось материалам малой размерности, двумерным материалам, в частности графену. До сих пор дискутируются механизмы переноса тепла и коэффициенты теплопроводности таких веществ.

В последнее время исследователи столкнулись с явлением аномального распространения тепла [72] при решении задач в наноструктурах (в отличие от больших образцов). Имея теоретическое обоснование дробно-дифференциальных законов диффузии, практические измерения теплопроводности наноструктур

сталкиваются с проблемой недостаточной точности применяемых техник на образцах такой длины. Одним из способов моделирования процесса распространения тепла является приложение теплового импульса и его молекулярно-динамическое неравновесное моделирование.

В экспериментах коэффициент теплопроводности чаще получают методом Рамановской спектроскопии. Лист графена закрепляют на подставках, по центру листа направляют лазерный луч (рис. 4.1 из [75]).

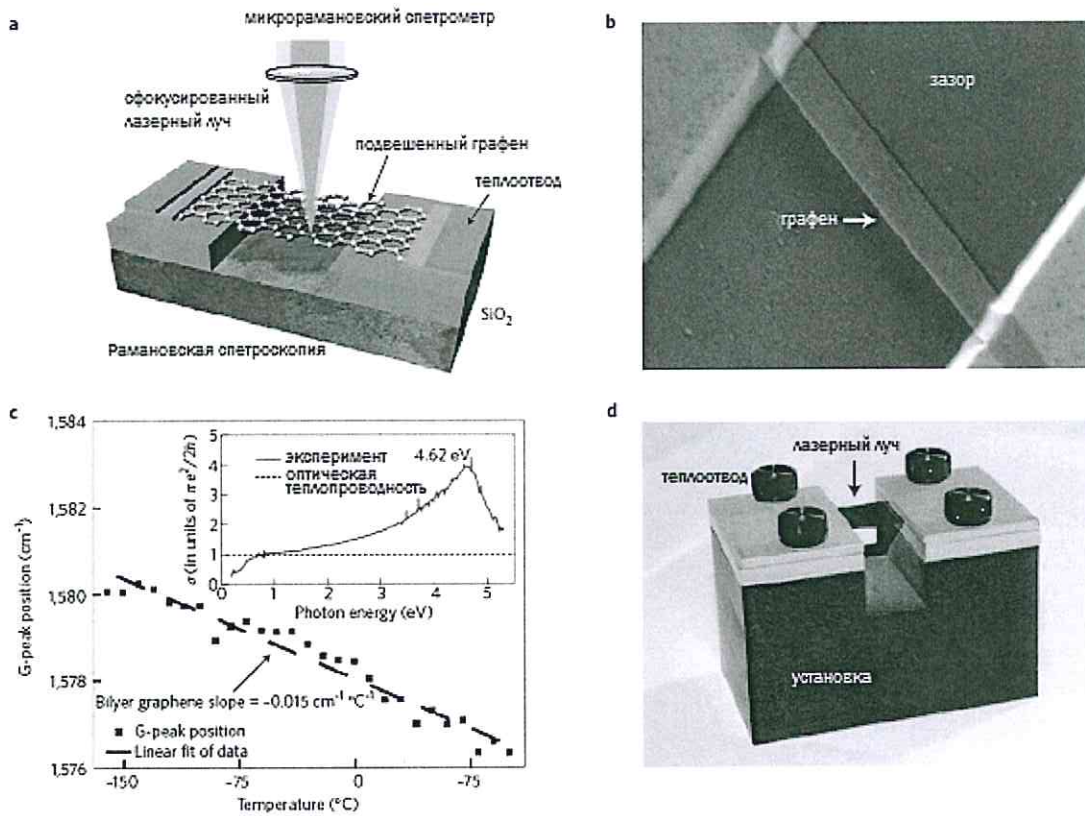


Рисунок 4.1. Схема эксперимента по изучению теплопроводности графена из [75]

Результаты экспериментальных работ собраны в таблице 4.1.

Таблица 4.1. Коэффициент теплопроводности графена в различных экспериментальных исследованиях.

Автор	Лаборатория	Длина, нм	Коэффициент, $\frac{W}{mK}$	Температура, К	Метод	Подложка

Ghosh [74]	Nano-Device Laboratory, Department of Electrical Engineering, University of California–Riverside	300	4100 ± 500	300	Raman optothermal	Suspended; exfoliated
Balandin [75]	Nano-Device Laboratory, Department of Electrical Engineering, University of California–Riverside	3000	4800	300	Raman optothermal	Suspended; exfoliated
Cai, Ruoff [76]	Department of Mechanical Engineering and Texas Materials Institute, University of Texas at Austin, Austin	3000	2500	350	Raman optothermal	Suspended; CVD
Jauregui [77]	Birck Nanotechnology Center, Purdue University, West Lafayette	300	1500	350	Raman optothermal	Suspended; CVD, Au/Cr on SiO ₂ /Si
Faugeras [78]	Laboratoire National des Champs Magnétiques Intenses, CNRS-UJF-UPS-INSA, France	44000	600	660	Raman optothermal	Suspended; exfoliated
Murali [79]	Microelectronics Research Center, Georgia Institute of Technology, Atlanta, Georgia	1000	1100	700	Electrical self-heating	Supported; exfoliated; n < 5
Seol [80]	Department of Mechanical Engineering, The University of Texas at Austin, Austin		600		Electrical	Supported; exfoliated
Ghosh [81]	Nano-Device Laboratory, Department of Electrical Engineering, University of California-Riverside, Riverside, California	300	4100	300	Raman optothermal	Si/SiO ₂
Xu [82]	Graphene Research Center, National University of Singapore, Singapore	9000	1689-1813	300	Thermal bridge	chemical vapor deposition on copper (Cu-CVD)
Chen [83]	Department of Mechanical Engineering and the Texas Materials Institute, The University of Texas at Austin, Austin	9700	2600	350	Raman optothermal	Cu foil using a CVD method
Lee [84]	Department of Physics, Sogang University, Seoul	1700	1800	350	Raman optothermal	SiO ₂

В лабораторных экспериментах, которые использовали данный метод, получен значительный разброс значений коэффициента теплопроводности.

Подготовка экспериментальных установок занимает много времени, поэтому используется моделирование и теоретический расчёт. Это позволяет работать с образцами любой конфигурации и длины, которые имеют идеальную структуру или примеси. Значения коэффициента теплопроводности, полученные моделированием собраны в таблице 4.2, теоретически рассчитанные значения собраны в таблице 4.3.

Таблица 4.2. Коэффициент теплопроводности графена в различных исследованиях по компьютерному моделированию образцов.

Автор	Программа	Длин а, нм	Коэффицие нт, $\frac{W}{mK}$	Темпе ратура , К	Метод
Evans [85]	?	10	8000		Equilibrium MD
Qui [86]	LAMMPS	44	1626	300- 650	Equilibrium MD
Zhang [87]	?	?	2903	300	Equilibrium MD, Green-Kubo
Wei [88]	?	7-25	120-290	300- 800	Nonequilibrium MD
Cao [89]	LAMMPS	2800	2360	300	Nonequilibrium MD
Chen [90]	LAMMPS	6	1606	300	Equilibrium MD
Hu[91]	?	10	2000	400	Nonequilibrium MD
Yu [92]	LAMMPS	600	564	300	Nonequilibrium MD
Pereira [93]	?	21	1015	350	Equilibrium MD

Таблица 4.3. Коэффициент теплопроводности графена в различных исследованиях по теоретическому расчёту.

Автор	Университет	Длина, нм	Коэффи циент, $\frac{W}{mK}$	Темпер атура, К	Метод
Alofi [94]	School of Physics, University of Exeter, United	2900	3000	300	Callaway theory

	Kingdom				
Fugallo [95]	IMPMC, Paris, France	100000	3600	300	BTE
Munoz [96]	Department of Mechanical Engineering & Materials Science, Rice University, Houston, Texas	750	3080	300	Ballistic
Aksamija [97]	Department of Electrical and Computer Engineering, University of Wisconsin-Madison, USA	500	3500	300	BTE
Gu [98]	Department of Mechanical Engineering, University of Colorado, USA	10000	3200	300	BTE
Klemens [99]	Department of Physics and Institute of Materials Science, University of Connecticut, USA	10000	4000	300	BTE
Nika [100]	Department of Electrical Engineering, University of California, Riverside, USA	5000000	4000	300	three-phonon Umklapp processes
Lindsay [101]	Department of Physics, Boston College, Chestnut Hill, Massachusetts, USA	1000000	3000	300	BTE

Полученные коэффициенты теплопроводности графена сильно разбросаны, а более точное их определение оказывается сложной экспериментальной задачей. Фундаментальной причиной сложности такой задачи является чувствительность методов определения к дефектам и условиям измерений. На точность исследований по моделированию теплопроводности влияют начальные условия и время усреднения. Для получения равномерного значения температуры вдоль образцов необходимо рассматривать временные промежутки более 1000 пс, но на коротких образцах 10-100 нм процесс распространения тепла проходит за 10-100пс.

4.2 Моделирование теплопроводности наноструктур. Идентификация и анализ режимов теплопроводности

В вычислительном эксперименте образец приводится в равновесное состояние за 1 пс термостатами Нозе-Хувера и Берендсена (в течение 1000 шагов структура

доводится до заданной средней температуры, один шаг равен 1 фемтосекунде). Затем к нему подводится тепло. Ставится задача определить характер распространения теплового пакета и получить наблюдаемое распределение температуры в продольном направлении наноструктуры в каждый момент времени.

Для этого с помощью разработанного программного комплекса проводится моделирование 10 пс пространственно-временной эволюции атомов наноструктуры (вычисляется 10000 шагов, один шаг равен 1 фемтосекунде). Следует отметить, что для этого необходимо использовать образцы достаточной длины, так как для коротких образцов (сравнимых с длиной свободного пробега фононов при комнатной температуре 80 нм – 100 нм) на достаточных интервалах времени тепло уже дошло до краёв структуры и отразилось, искажая картину распределения температуры. Выдерживается микроканонический ансамбль NVE. Под распространением тепла предполагается распространение фононных волн.

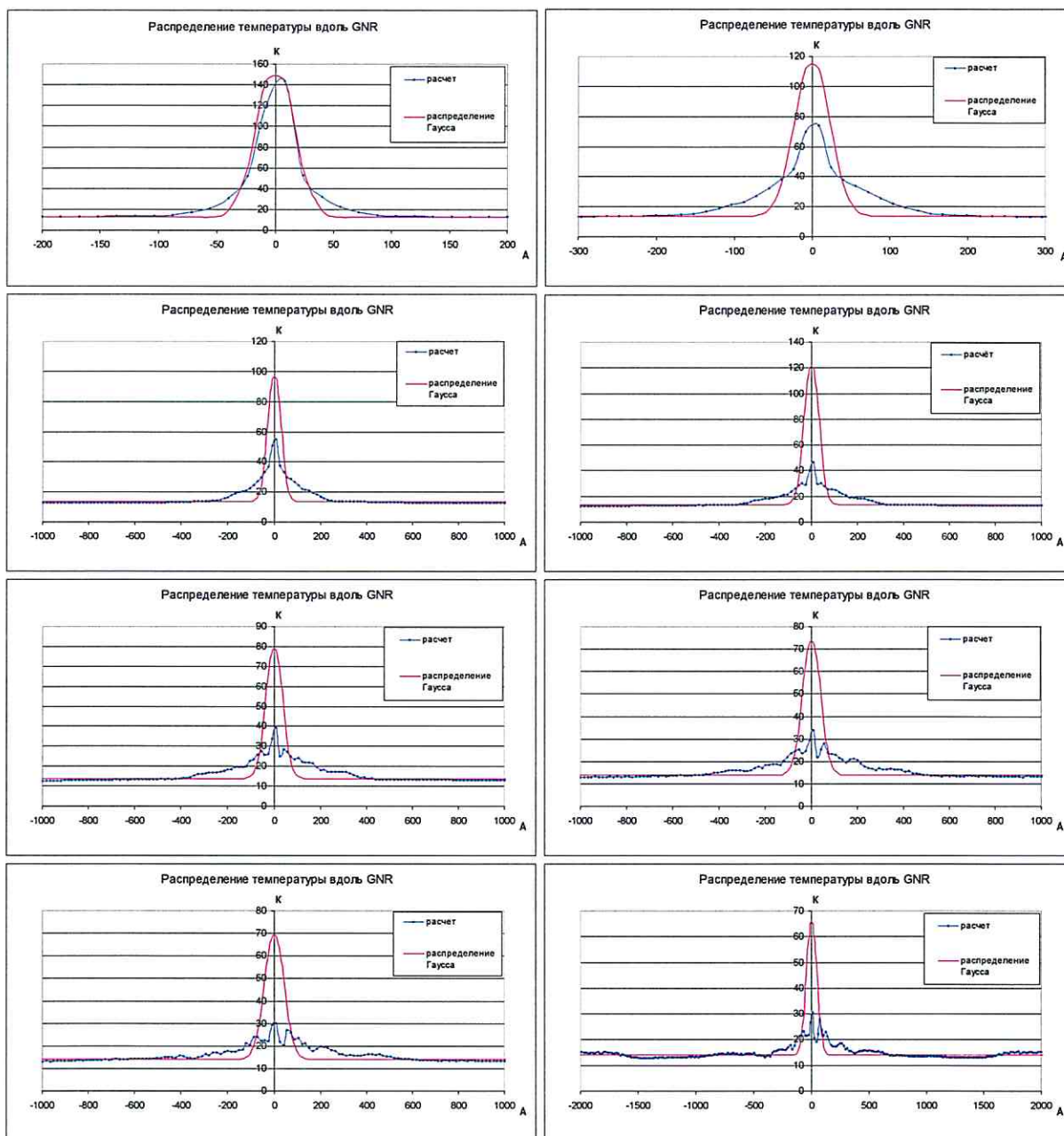
В завершение проводится математическая обработка полученного распределения температуры по образцу. Осреднение температуры проводится по 1 пс (1000 временных шагов) по области длиной 16Å. Полученная картина затем анализируется с целью определения наблюдавшегося режима распространения тепла.

4.2.1 Моделирование распространения тепла в листе графена

Для количественного описания теплопроводности листа графена выбран образец длиной 629.8 нм. Проводилась серия вычислительных экспериментов с варьированием значения начальной средней температуры листа от 16К до 400К. После стабилизации системы термостатами по центру листа в полоске шириной 20Å задавался тепловой импульс, соответствующий температуре 600К, с гауссовым распределением проекций скоростей атомов в каждом случае. Генерация теплового импульса со значительно большими энергиями приводит к распаду образца (при температурах выше температуры кипения графена).

Получившиеся картины распределения температуры по длине образца показаны на рис. 4.2 для начальной средней температуры листа в 16К и на рис. 4.3 для температуры листа в 300К. Приведено по 10 графиков, представляющих результаты

осреднения температуры за каждые 1000 временных шагов; для наглядности приведено также распределение Гаусса, масштабированное по первому пику распределения температуры.



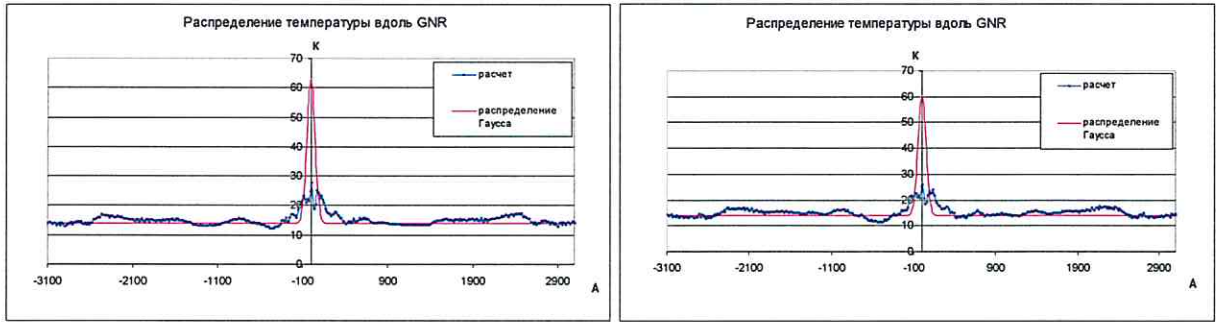
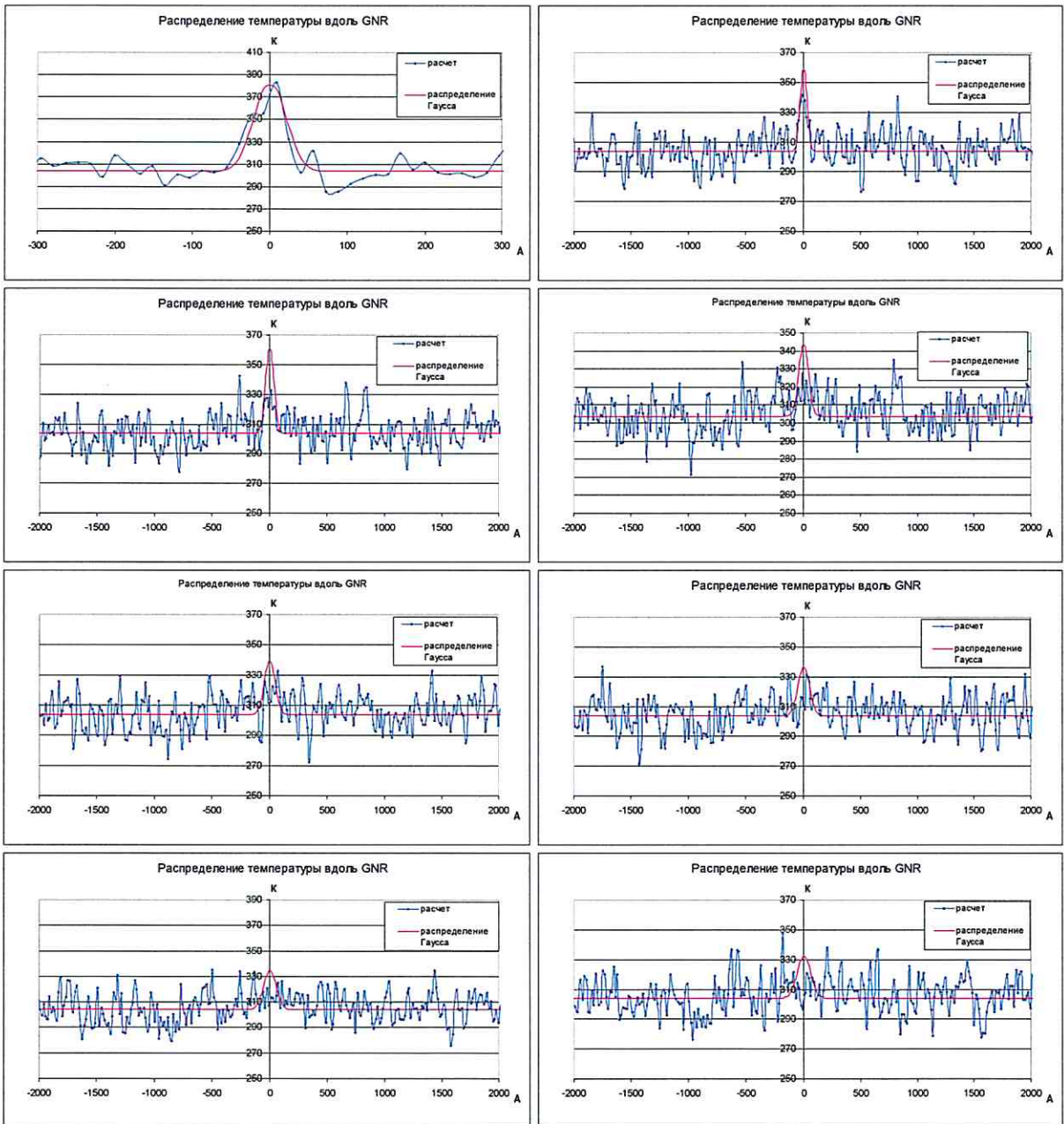


Рисунок 4.2. Распространение теплового импульса в листе графена при температуре листа $T=16\text{K}$. На графиках отмечено среднее значение температуры за 1 пс процесса распространения тепла



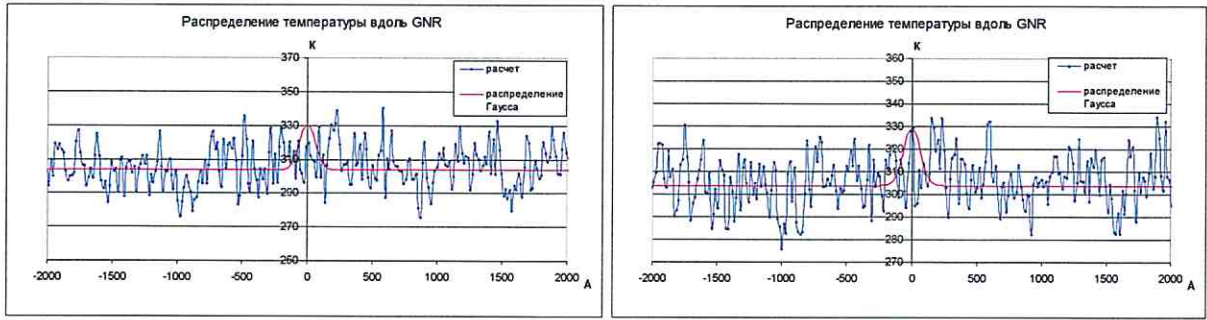


Рисунок 4.3. Распространение теплового импульса в листе графена при температуре листа $T=300\text{K}$

Для математической обработки полученных распределений температуры по образцу возможно применение разных методов.

Проведём анализ полученного распределения температуры по образцу.

В первую очередь, количественно изменение ширины теплового импульса во времени измеряется среднеквадратичным отклонением σ :

$$\sigma^2(t) = \frac{\sum_{i=0}^N (u(\mathbf{r}_i, t) - u_0(\mathbf{r}_i))(\mathbf{r}_i(t) - \mathbf{r}_i^0)^2}{\sum_{i=0}^N (u(\mathbf{r}_i, t) - u_0(\mathbf{r}_i))}, \quad (4.1)$$

где N – количество атомов в системе, $\mathbf{r}_i(t)$ – радиус-вектор i -ого атома в момент t , \mathbf{r}_i^0 – радиус-вектор атома в начальный момент времени,

$$u(\mathbf{r}_i, t) = \frac{m_i |\mathbf{v}_i|^2}{3k_B}, \quad (4.2)$$

а $u_0(\mathbf{r}_i)$ – энергия атома в начальный момент времени.

На рис. 4.4 зелёным цветом отображены рассчитанные по формуле (4.1) значения σ^2 , а красным цветом отображена линейная зависимость $y=1.75x-22.59$, полученная методом наименьших квадратов для этих значений, синим цветом отображено значение ширины гауссового импульса, масштабированного по первому пику распределения температуры. Все зависимости приведены для 10000 шагов моделирования в логарифмическом масштабе для распространения тепла в листе графена при температуре 16K.

Можно видеть, что наблюдается существенное отклонение наблюдаемого режима распространения тепла $\langle r^2 \rangle = \sigma^2 \sim t^p$, $p=1.75$ от нормального $\langle r^2 \rangle = \sigma^2 \sim t$. Таким образом, в листе графена шириной 20\AA и длиной 630 нм при температуре 16K наблюдается аномальная теплопроводность.

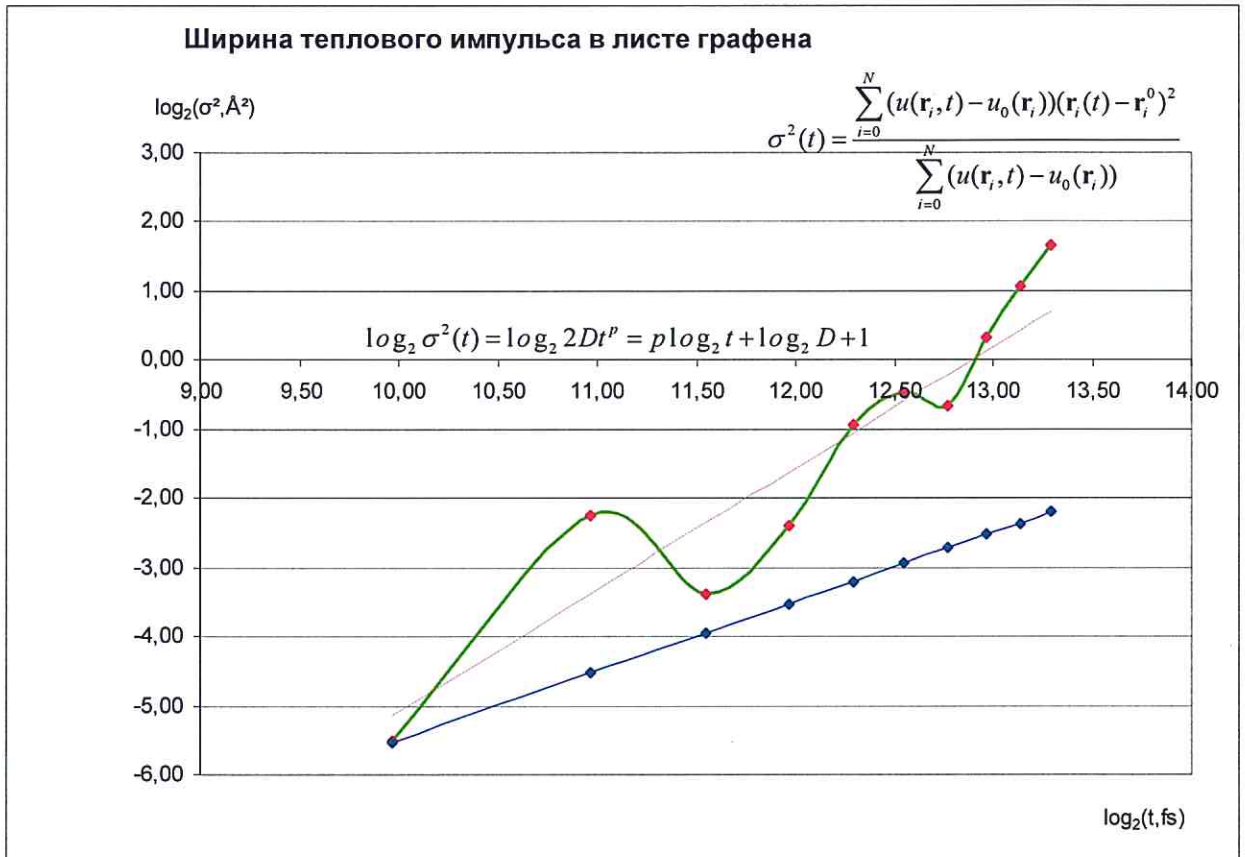


Рисунок 4.4. Распространение теплового импульса в листе графена при $T=16\text{K}$ (логарифмическая шкала)

На практике среднеквадратичное отклонение $\langle \Delta x^2 \rangle(t) = \langle r^2 \rangle = \sigma^2$ даёт в целом верное представление о характере теплопроводности, но может давать неточное количественное представление числом p . Поэтому один из основных вопросов заключается в установлении точного значения этого параметра аномальности по результатам полученного распределения температуры и её динамике. Другим способом определения этого параметра для задачи теплового импульса представляется оценка скорости убывания максимума полученного распределения. Поэтому помимо определения параметра p в зависимости $\langle r^2 \rangle = \sigma^2 \sim t^p$, по результатам эксперимента вычислялся также порядок теплопереноса p' по соотношению:

$$p' = -2 \frac{d(\ln[\max_x u(x, t)])}{d(\ln(t))} \quad (4.3)$$

На рис. 4.5 показан график зависимости температуры в центре образца от времени при начальной температуре 16К. На рис. 4.6 эта зависимость дана в логарифмической шкале. Красным цветом отображена линейная зависимость $y=1.99x-33.81$, полученная методом наименьших квадратов для этих значений температур. Можно видеть, что по-прежнему наблюдается существенное отклонение режима распространения тепла $\langle r^2 \rangle = \sigma^2 \sim t^{p'}$, $p'=1.99$ от нормального $\langle r^2 \rangle = \sigma^2 \sim t$.

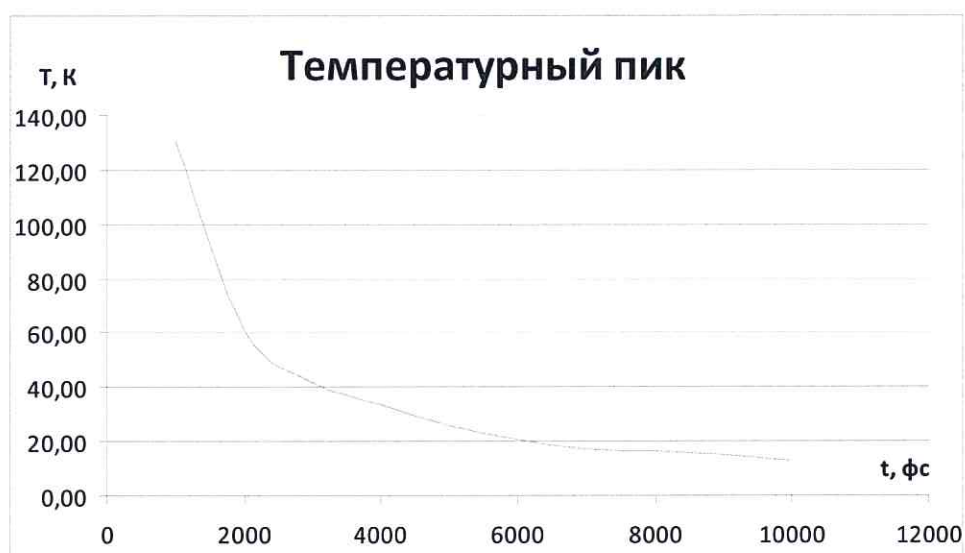


Рисунок 4.5. Спад пика энергии в листе графена при $T=16\text{K}$ с течением времени

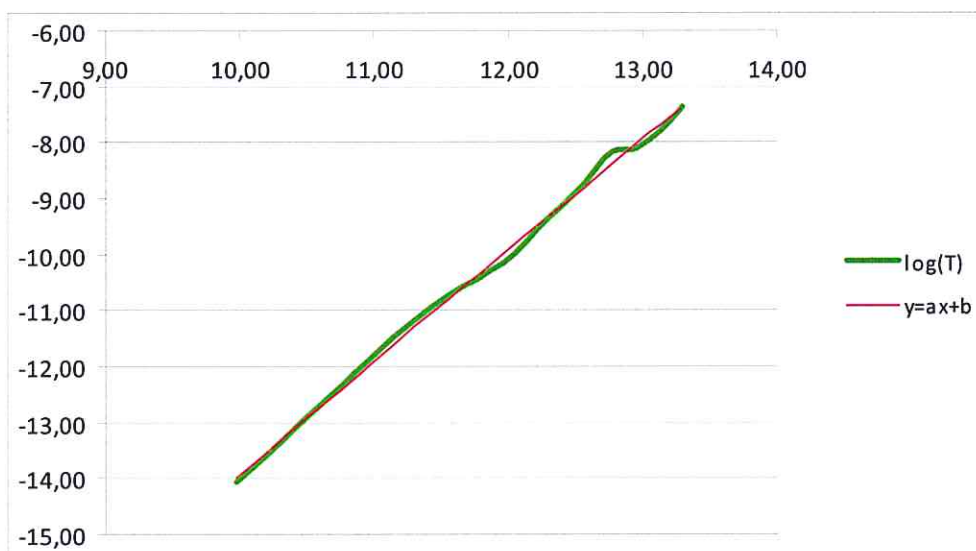


Рисунок 4.6. Спад пика энергии в листе графена при $T=16\text{K}$ с течением времени (логарифмическая шкала)

Использованные выше методы дают оценку характера теплопереноса в случае сильного преобладания пика над соседними температурными регионами. Однако с

повышением температуры образца и усреднением в 1000 фс данные оценки сильно затруднены из-за шума (см. рис. 4.2 для 300К).

Поэтому в работе предлагается третий способ определения параметра p . Необходимо выделить часть импульса, содержащую 68.26% энергии (для простоты) и определить его местоположение на образце. Для нормального распределения это расположение будет соответствовать интервалу $[-\sigma, \sigma]$, для аномальной теплопроводности он будет шире $[-\Omega, \Omega]$.

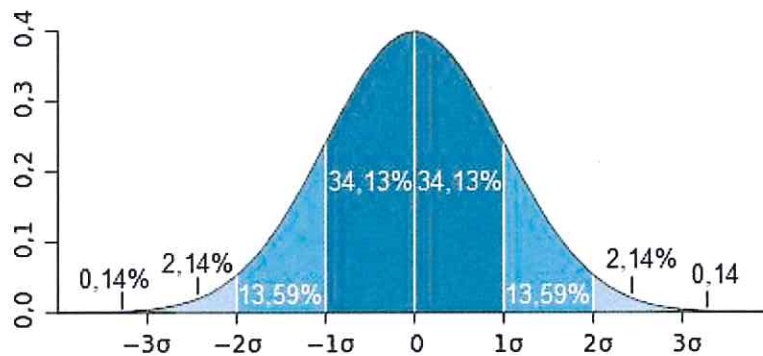


Рисунок 4.7. Плотность вероятности гауссова распределения

Тогда логарифмирование позволит оценить параметр p'' :

$$\log_2 \Omega(t) = \log_2 \sqrt{2D''t^{p''}} = \frac{p''}{2} \log_2 t + \frac{1}{2} \log_2 D'' + \frac{1}{2} \quad (4.4)$$

Подобный способ оценки параметра аномальности p представляется как самый точный из трёх. Также следует по формуле (4.4) оценить параметр D'' .

На рис. 4.8 показаны графики зависимостей доли суммарной энергии теплового импульса, поданного на образец при начальной температуре 16К, от размера взятого интервала $[-x, x]$ для различных моментов времени. Горизонтальная линия соответствует доле энергии в 68.26%, пересечение этой линии с кривыми даёт значение искомого интервала $[-\Omega, \Omega]$.

Доля энергии от x

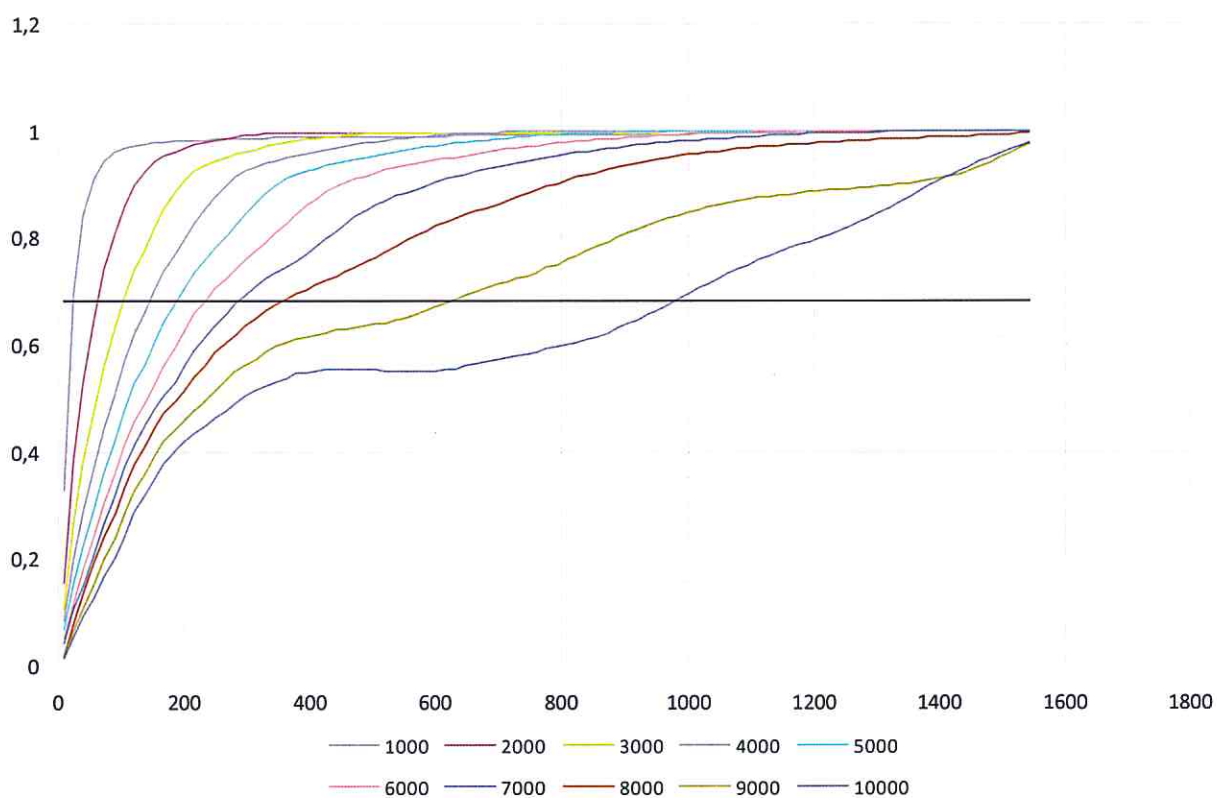


Рисунок 4.8. Зависимость доли энергии теплового импульса от размера интервала $[-x, x]$ при распространении тепла вдоль листа графена при $T=16\text{K}$. Графики соответствуют моментам времени, усреднённые за 1000 фс

На рис. 4.9 дана зависимость искомой Ω от времени в логарифмической шкале. Красным цветом отображена линейная зависимость $y=0.90x-7.37$, полученная методом наименьших квадратов для этих значений Ω . Наблюдается аномальный режим распространения тепла $\langle r^2 \rangle = \sigma^2 \sim t^p, p''=1.81$.

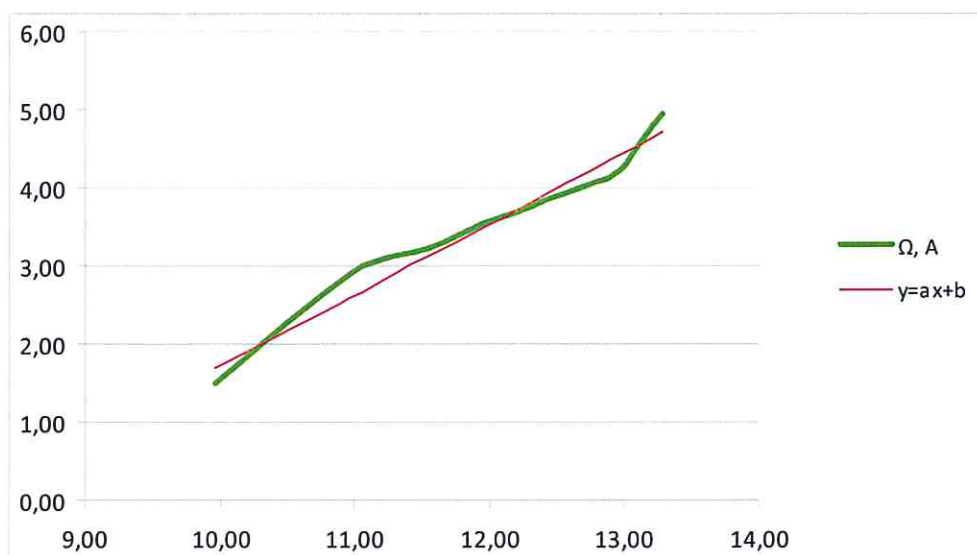


Рисунок 4.9. Зависимость размера интервала $[-\Omega, \Omega]$ от времени при распространении тепла вдоль листа графена при $T=16\text{K}$ (логарифмическая шкала)

В следующем эксперименте подводится тепловой импульс при температуре 300K , что даёт увеличение температуры на 1K . На рис. 4.10 зелёным цветом отображены рассчитанные по формуле (4.1) значения σ^2 , а красным цветом отображена линейная зависимость $y=0.0022x-2.11$, полученная методом наименьших квадратов для этих значений, синим цветом отображено значение ширины гауссового импульса, масштабированного по первому пику распределения температуры. Все зависимости приведены для 10000 шагов моделирования в логарифмическом масштабе для распространения тепла в листе графена при температуре 300K .

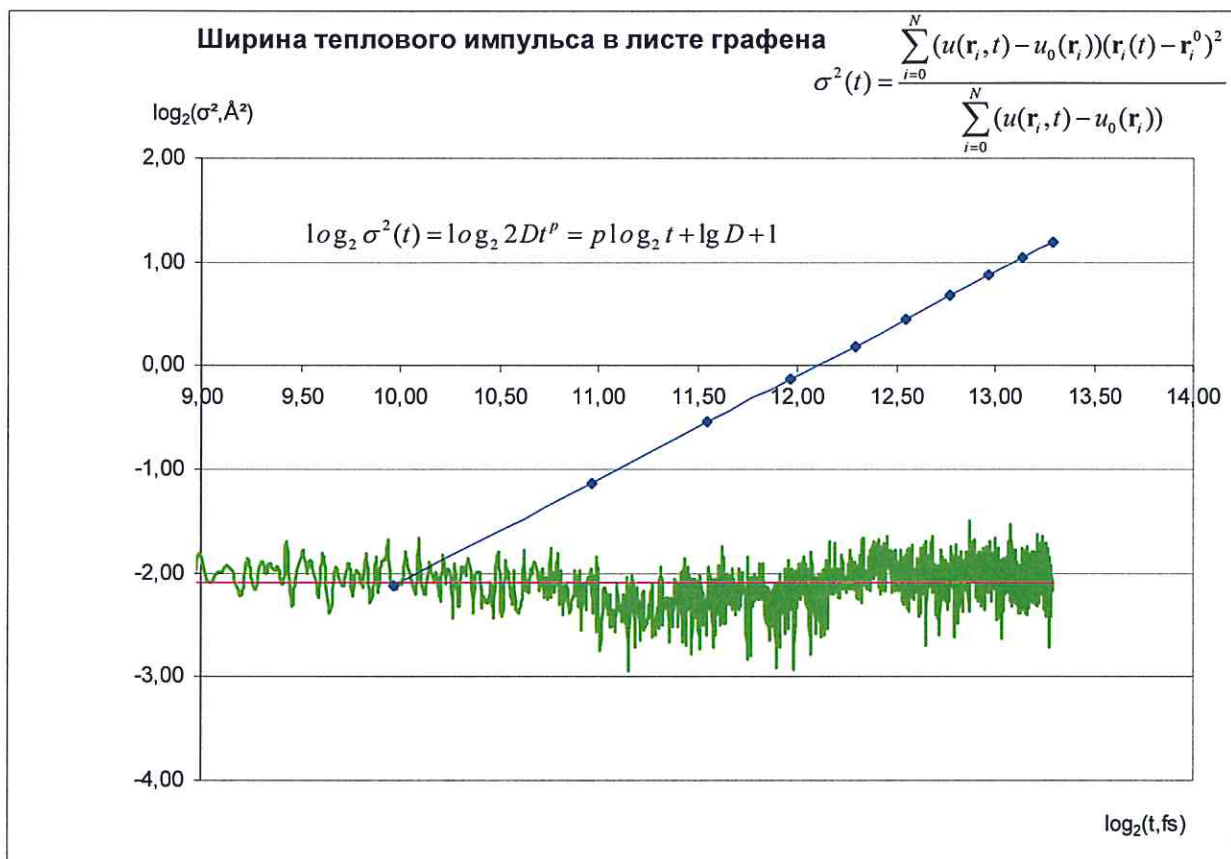


Рисунок 4.10. Распространение теплового импульса в листе графена при $T=300\text{K}$ (логарифмическая шкала)

Из-за уровня зашумлённости данных таким методом определить режим распространения тепла не удаётся.

На рис. 4.11 показан график зависимости температуры в центре образца от времени при начальной температуре 300K . На рис. Рисунок 4.12 эта зависимость дана в логарифмической шкале. Красным цветом отображена линейная зависимость $y=1.89x-30.33$, полученная методом наименьших квадратов для этих значений. Вновь уровень шума не позволяет даже приблизительно определить режим распространения тепла.

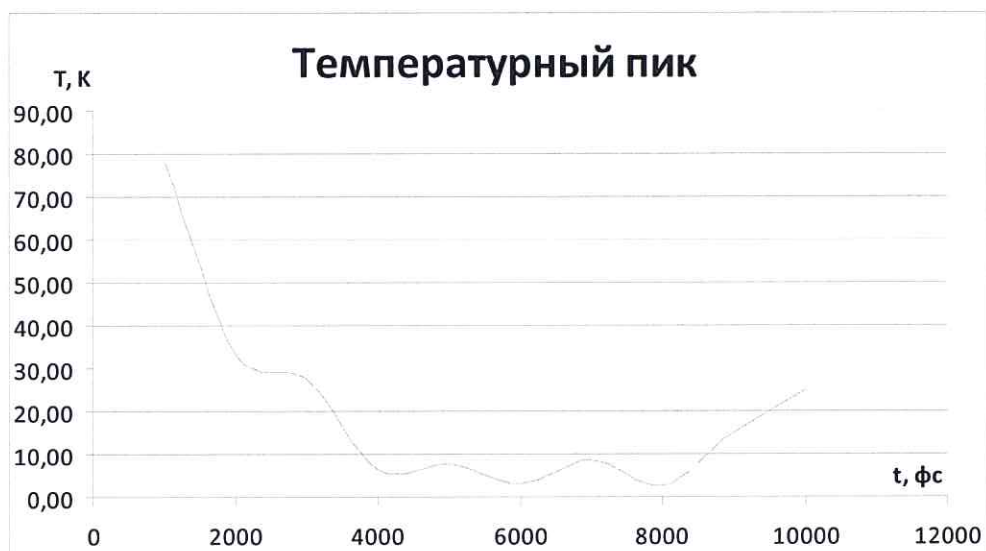


Рисунок 4.11. Спад пика энергии в листе графена при $T=300\text{K}$ с течением времени

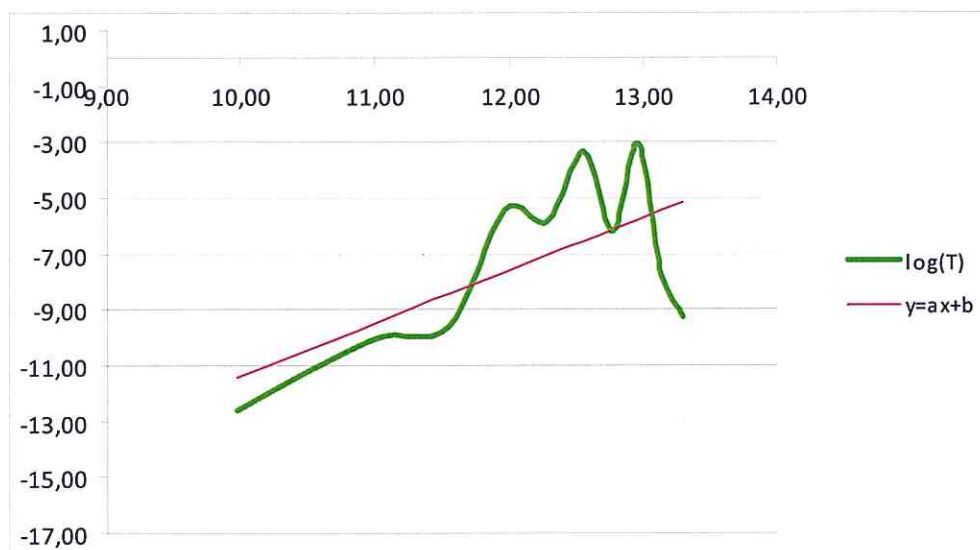


Рисунок 4.12. Спад пика энергии в листе графена при $T=300\text{K}$ с течением времени (логарифмическая шкала)

На рис. 4.13 показаны графики зависимостей доли суммарной энергии теплового импульса, поданного на образец при начальной температуре 300К, от размера взятого интервала $[-x, x]$ для различных моментов времени. Горизонтальная линия соответствует доле энергии в 68.26%, пересечение этой линии графиком даёт значение искомого интервала $[-\Omega, \Omega]$.

Доля энергии от x

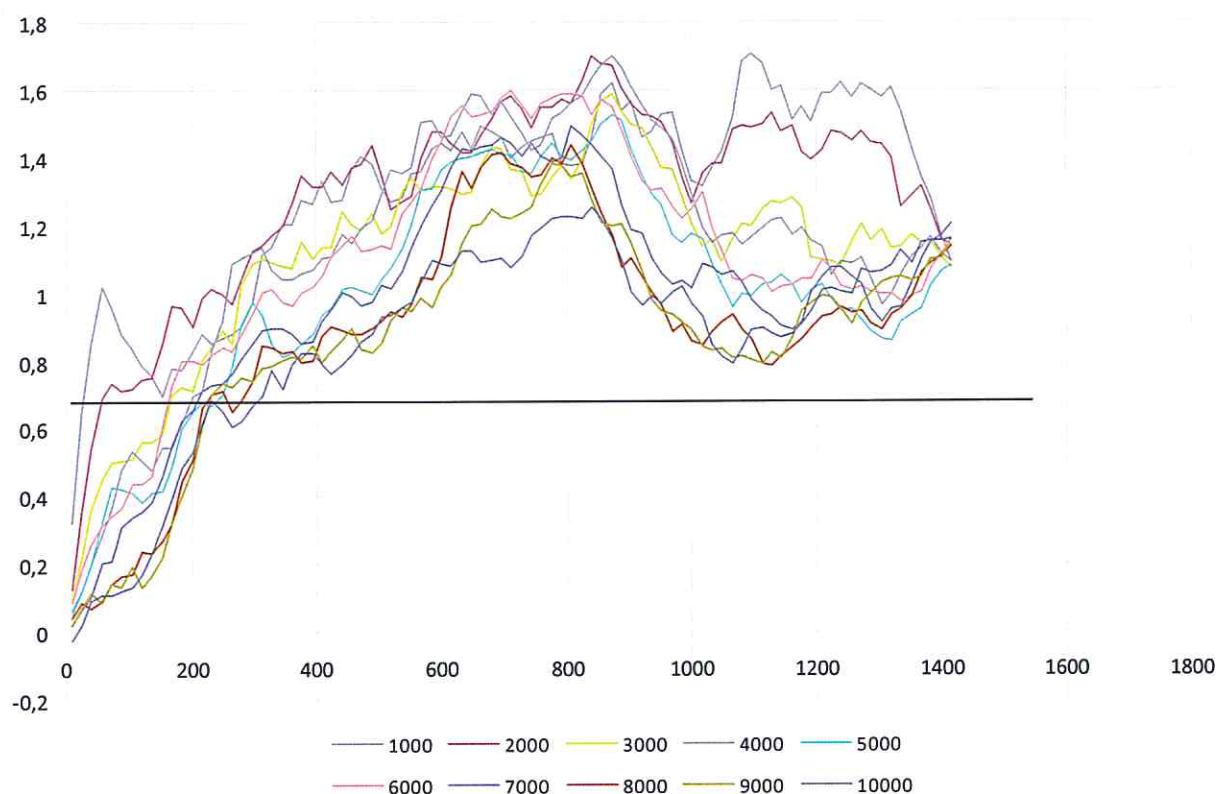


Рисунок 4.13. Зависимость доли энергии теплового импульса от размера интервала $[-x, x]$ при распространении тепла вдоль листа графена при $T=300\text{K}$. Графики соответствуют моменты времени, усреднённые за 1000 фс

На рис. 4.14 дана зависимость искомой Ω от времени в логарифмической шкале. Красным цветом отображена линейная зависимость $y=0.50x+2.45$, полученная методом наименьших квадратов для этих значений Ω при усреднении 10 фс. Наблюдается близкий к нормальному режим распространения тепла $\langle r^2 \rangle = \sigma^2 \sim t$, $p''=1.01$.

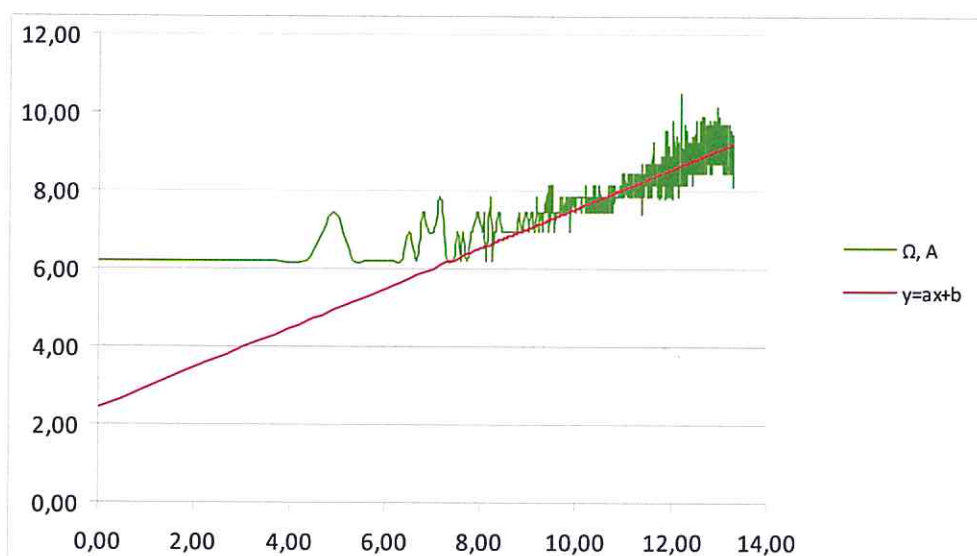


Рисунок 4.14. Зависимость размера интервала $[-\Omega, \Omega]$ от времени при распространении тепла вдоль листа графена при $T=300\text{K}$. Графики отображают моменты времени, усреднённые за 10 фс (логарифмическая шкала)

Подобный анализ режимов теплопроводности, возникающих в листе графена, возможно проделать для целого набора его начальных температур. Была проведена серия вычислительных экспериментов для температур листа в 16К, 20К, 32К, 37К, 46К, 58К, 82К, 128К, 256К, 300К, 400К. Результат математической обработки возникающих в каждом случае распределений температур представлен в табл. 4.4. Значения в пустых ячейках таблицы необходимо уточнить.

Таблица 4.4. Характеристики теплопроводности графена.

Температура, К	16	20	32	37	46	58	82	128	256	300	400
p	1.75	*	*	1.43	1.66	1.58	*	1.39	1.39	0.00	*
p'	1.99	2.69	2.21	2.18	*	2.17	*	*	*	1.89	*
p''	1.81	1.34	1.25	1.03	1.09	0.99	1.09	1.04	0.99	1.01	0.97

Очевидно, что достоверные значения параметра аномальности предоставляются только числом p'' . График значений параметров p'' , полученных по результатам моделирования теплопроводности листа графена в исследованном диапазоне температур от 16К до 400К, представлен на рис. 4.15. При низких температурах (32К и ниже) наблюдается аномальный режим распространения тепла.

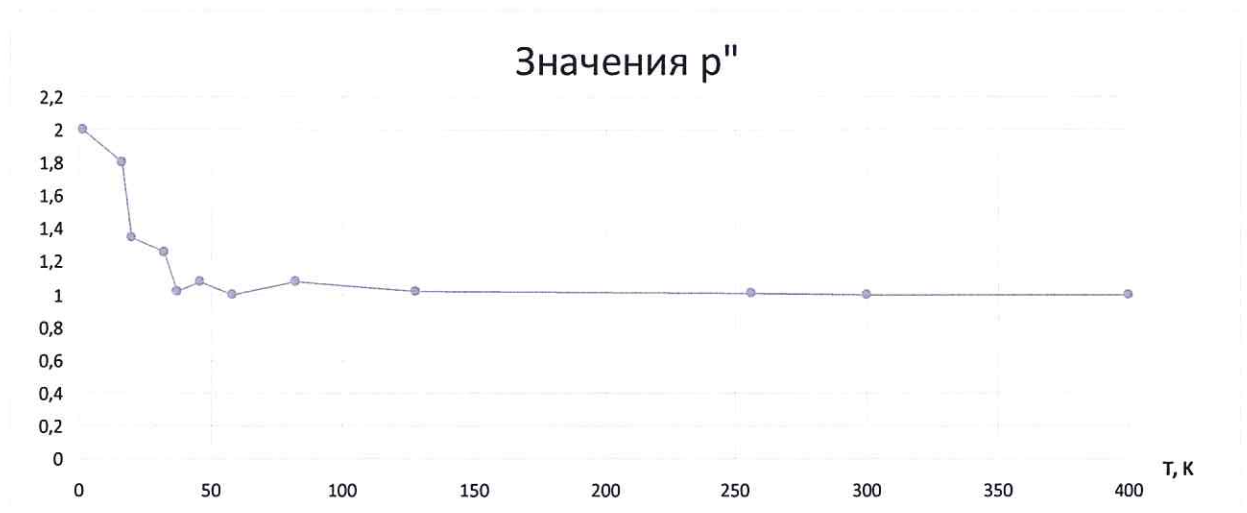


Рисунок 4.15. Значения параметра ρ'' в диапазоне температур от 16К до 400К

По результатам моделирования рассчитаем коэффициент теплопроводности графена при начальной температуре листа в 300К, когда наблюдается нормальный режим распространения тепла.

Сначала вычислим плотность листа графена:

$$m = Nm_a = 6720 \cdot 1.993 \cdot 10^{-26} = 9.5664 \cdot 10^{-22} \text{ кг},$$

$$V = 2lwd = 6898 \cdot 10^{-10} \cdot 20 \cdot 10^{-10} \cdot 3.4 \cdot 10^{-10} = 42196.6 \cdot 10^{-30} \text{ м}^3,$$

$$\rho = \frac{m}{V} = \frac{9.5664 \cdot 10^{-22}}{42196.6 \cdot 10^{-30}} = 2267 \frac{\text{кг}}{\text{м}^3}.$$

Теплоёмкость листа графена при постоянном объёме можно вычислить по следующей формуле (микрoканонический ансамбль):

$$C_V = \frac{(\delta K)^2}{k_B T^2},$$

где $(\delta K(t_k))^2 = \langle K^2(t_k) \rangle - \langle K \rangle^2 = \langle K(t_k) - \langle K \rangle \rangle^2$ – флуктуация кинетической энергии.

$$C_V = \frac{8,5587 \cdot 10^{-37}}{1,38054 \cdot 10^{-23} \cdot 300^2} = 6.8878 \cdot 10^{-19}.$$

Тогда удельная теплоёмкость графена при начальной температуре листа 300К:

$$c = \frac{C_V}{m} = \frac{6.8878 \cdot 10^{-19}}{9.5664 \cdot 10^{-22}} = 720 \frac{\text{Дж}}{\text{кг} \cdot \text{К}}.$$

Аналогично можно подсчитать удельную теплоёмкость и для остальных начальных температур, для которых проводился вычислительный эксперимент. Полученные значения представлены на рис. 4.16.

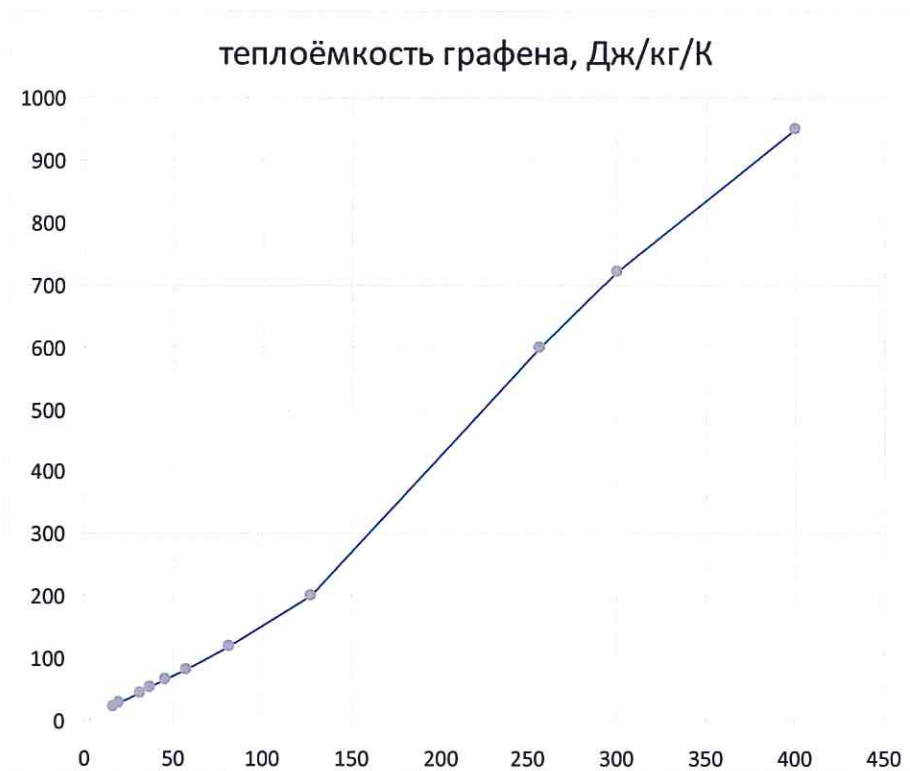


Рисунок 4.16. Удельная теплоёмкость листа графена в зависимости от температуры
 Коэффициент теплопроводности графена тогда вычисляется как:

$$\kappa = c\rho D = 720 \cdot 2267 \cdot 1.37 \cdot 10^{-3} = 2236 \frac{\text{Вт}}{\text{м} \cdot \text{К}},$$

где для значения D используется D'' .

На рис. 4.17 собраны значения коэффициента теплопроводности графена, полученные в разных исследованиях [83] и в результате моделирования в зависимости от начальной температуры листа графена. Приведены аналогично полученные результаты и для листа графена длиной 400 нм.

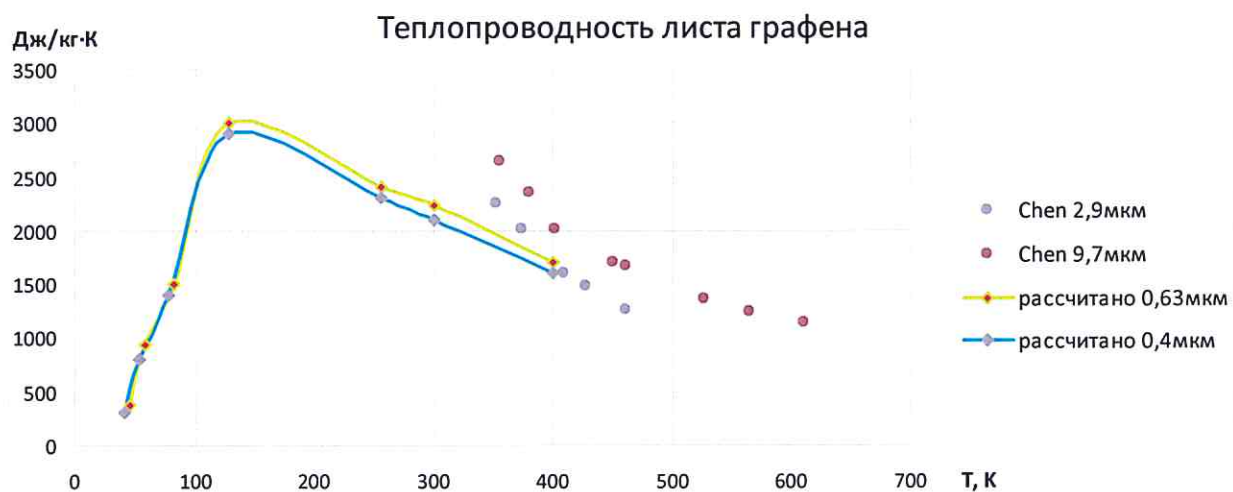


Рисунок 4.17. Зависимость коэффициента теплопроводности графена в диапазоне температур от 16К до 400К

4.2.2 Моделирование распространения тепла в нанотрубке

Для количественного описания теплопроводности нанотрубки (10,0) выбран образец длиной 400 нм. Проводилась серия вычислительных экспериментов с варьированием значения начальной средней температуры трубки от 16К до 400К. В вычислительном эксперименте образец приводится в равновесное состояние за 10 пс термостатами Нозе-Хувера и Берендсена (в течение 10000 шагов структура доводится до заданной средней температуры, один шаг равен 1 фемтосекунде). Затем к нему подводится тепло. После стабилизации системы термостатами по центру листа в полоске шириной 20\AA задавался тепловой импульс, соответствующий температуре 600К, с гауссовым распределением проекций скоростей атомов в каждом случае.

Получившиеся картины распределения температуры по длине образца показаны на рис. 4.18 для начальной средней температуры трубки в 16К и на рис. 4.19 для температуры трубки в 300К. Приведены графики для моментов времени 1 пс и 10 пс, представляющие результаты осреднения температуры за 1000 временных шагов; для наглядности приведено также распределение Гаусса, масштабированное по первому пику распределения температуры.

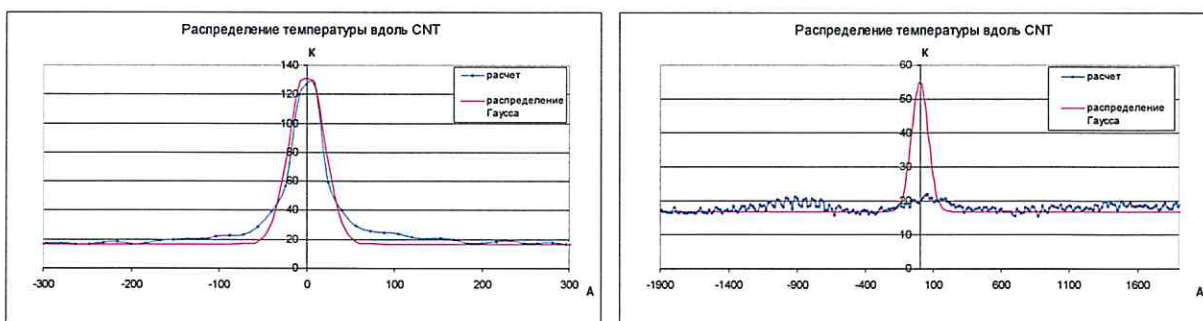


Рисунок 4.18. Распространение теплового импульса в нанотрубке (10,0) при температуре листа $T=16\text{K}$. На графиках отмечено среднее значение температуры за 1 пс и 10 пс процесса распространения тепла

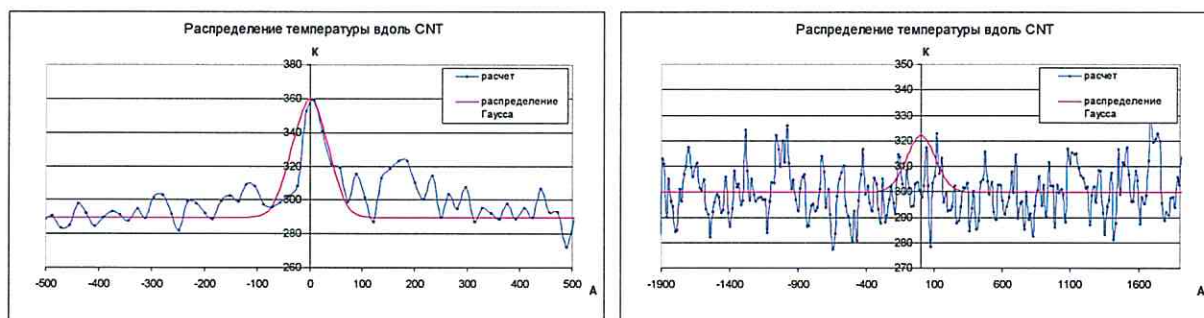


Рисунок 4.19. Распространение теплового импульса в нанотрубке (10,0) при температуре листа $T=300\text{K}$. На графиках отмечено среднее значение температуры за 1 пс и 10 пс процесса распространения тепла

На рис. 4.20 зелёным цветом отображены рассчитанные по формуле (4.1) значения σ^2 , а красным цветом отображена линейная зависимость $y=1.25x-14.79$, полученная методом наименьших квадратов для этих значений, синим цветом отображено значение ширины гауссового импульса, масштабированного по первому пику распределения температуры. Все зависимости приведены в логарифмическом масштабе для распространения тепла в нанотрубке (10,0) при температуре 16К.

Можно видеть, что наблюдается существенное отклонение наблюдаемого режима распространения тепла $\langle r^2 \rangle = \sigma^2 \sim t^p$, $p=1.25$ от нормального $\langle r^2 \rangle = \sigma^2 \sim t$. Таким образом, в нанотрубке (10,0) длиной 400 нм при температуре 16К наблюдается аномальная теплопроводность.

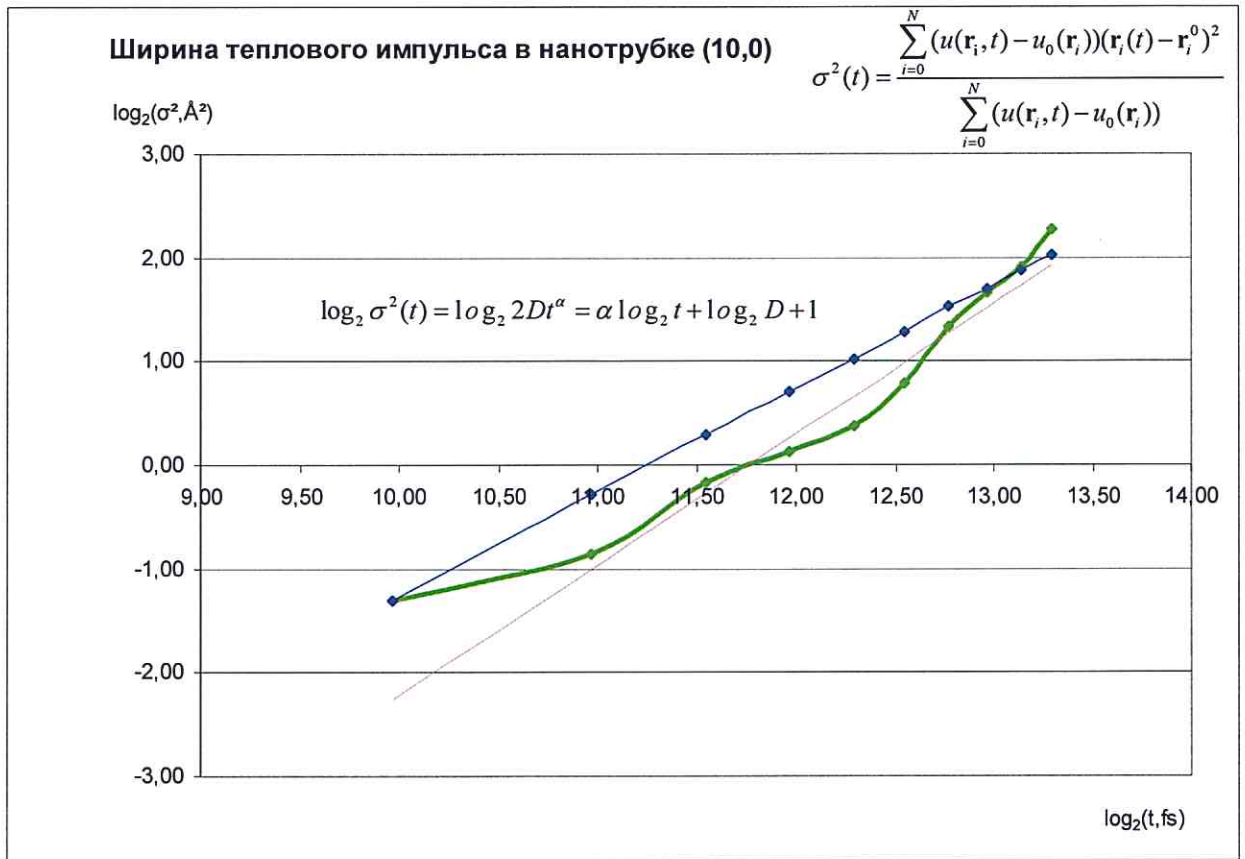


Рисунок 4.20. Распространение теплового импульса в нанотрубке (10,0) при $T=16\text{K}$ (логарифмическая шкала)

На рис. 4.21 показан график зависимости температуры в центре образца от времени при начальной температуре 16K. На рис. 4.22 эта зависимость дана в логарифмической шкале. Красным цветом отображена линейная зависимость $y=2.77x-41.92$, полученная методом наименьших квадратов для этих значений температур. Можно видеть, что по-прежнему наблюдается существенное (сильно завышенное) отклонение режима распространения тепла $\langle r^2 \rangle = \sigma^2 \sim t^{p'}$, $p'=2.77$ от нормального $\langle r^2 \rangle = \sigma^2 \sim t$.

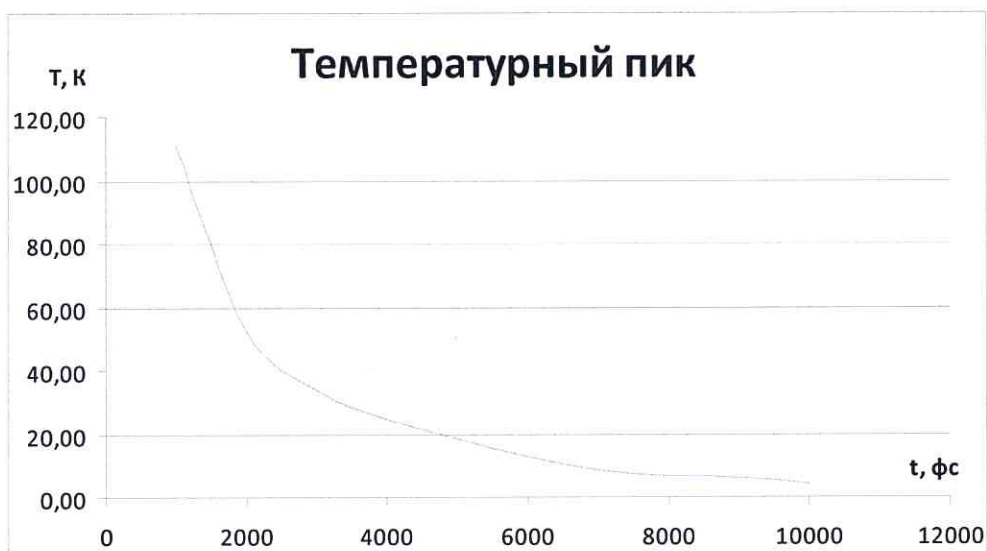


Рисунок 4.21. Спад пика энергии в нанотрубке (10,0) при $T=16\text{K}$ с течением времени

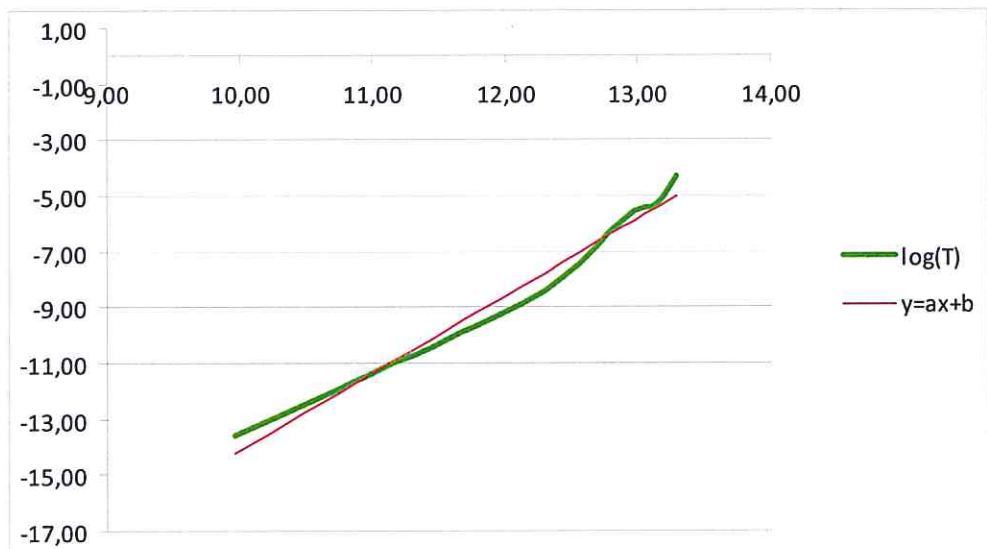


Рисунок 4.22. Спад пика энергии в нанотрубке (10,0) при $T=16\text{K}$ с течением времени (логарифмическая шкала)

На рис. 4.23 показаны графики зависимостей доли суммарной энергии теплового импульса, поданного на образец при начальной температуре 16К, от размера взятого интервала $[-x, x]$ для различных моментов времени. Горизонтальная линия соответствует доле энергии в 68.26%, пересечение этой линии графиком даёт значение искомого интервала $[-\Omega, \Omega]$.

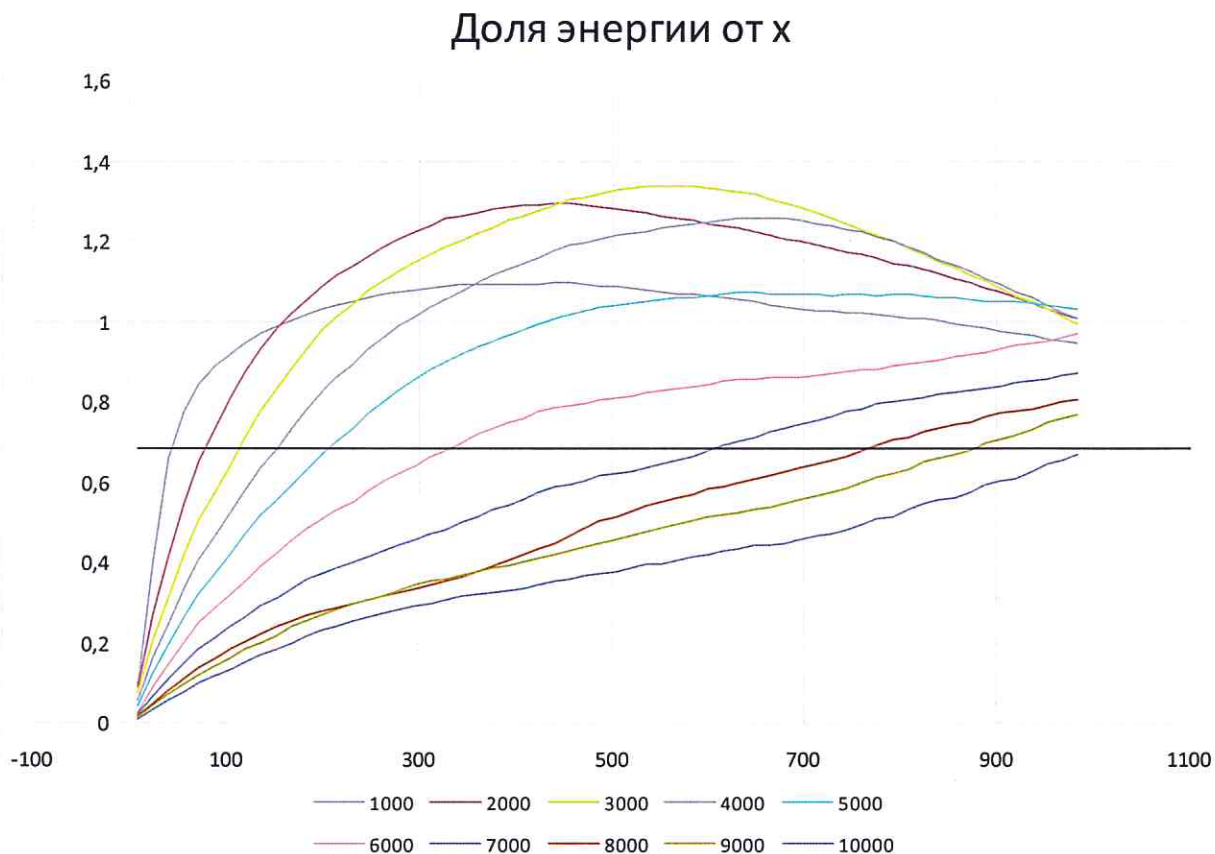


Рисунок 4.23. Зависимость доли энергии теплового импульса от размера интервала $[-x, x]$ при распространении тепла вдоль нанотрубки $(10,0)$ при $T=16\text{K}$. Графики соответствуют моменты времени, усреднённые за 1000 фс

На рис. 4.24 дана зависимость искомой Ω от времени в логарифмической шкале. Красным цветом отображена линейная зависимость $y=0.71x-4.59$, полученная методом наименьших квадратов для этих значений Ω . Наблюдается аномальный режим распространения тепла $\langle r^2 \rangle = \sigma^2 \sim t^{p''}$, $p''=1.42$.

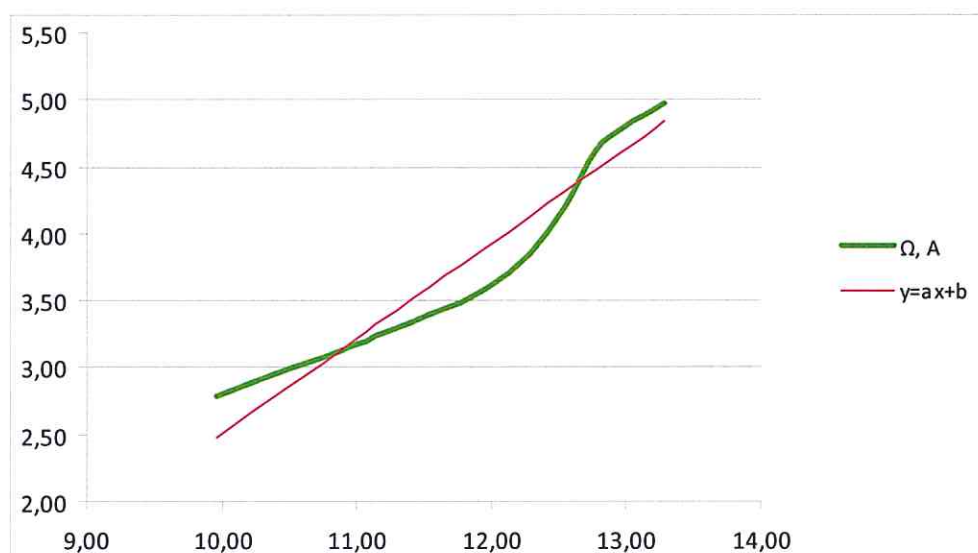


Рисунок 4.24. Зависимость размера интервала $[-\Omega, \Omega]$ от времени при распространении тепла вдоль нанотрубки (10,0) при $T=16\text{K}$ (логарифмическая шкала)

В следующем эксперименте подводится тепловой импульс при температуре 300K. На рис. 4.25 зелёным цветом отображены рассчитанные по формуле (4.1) значения σ^2 , а красным цветом отображена линейная зависимость $y=0.12x+0.75$, полученная методом наименьших квадратов для этих значений, синим цветом отображено значение ширины гауссового импульса, масштабированного по первому пику распределения температуры. Все зависимости приведены в логарифмическом масштабе для распространения тепла в нанотрубке (10,0) при температуре 300K.

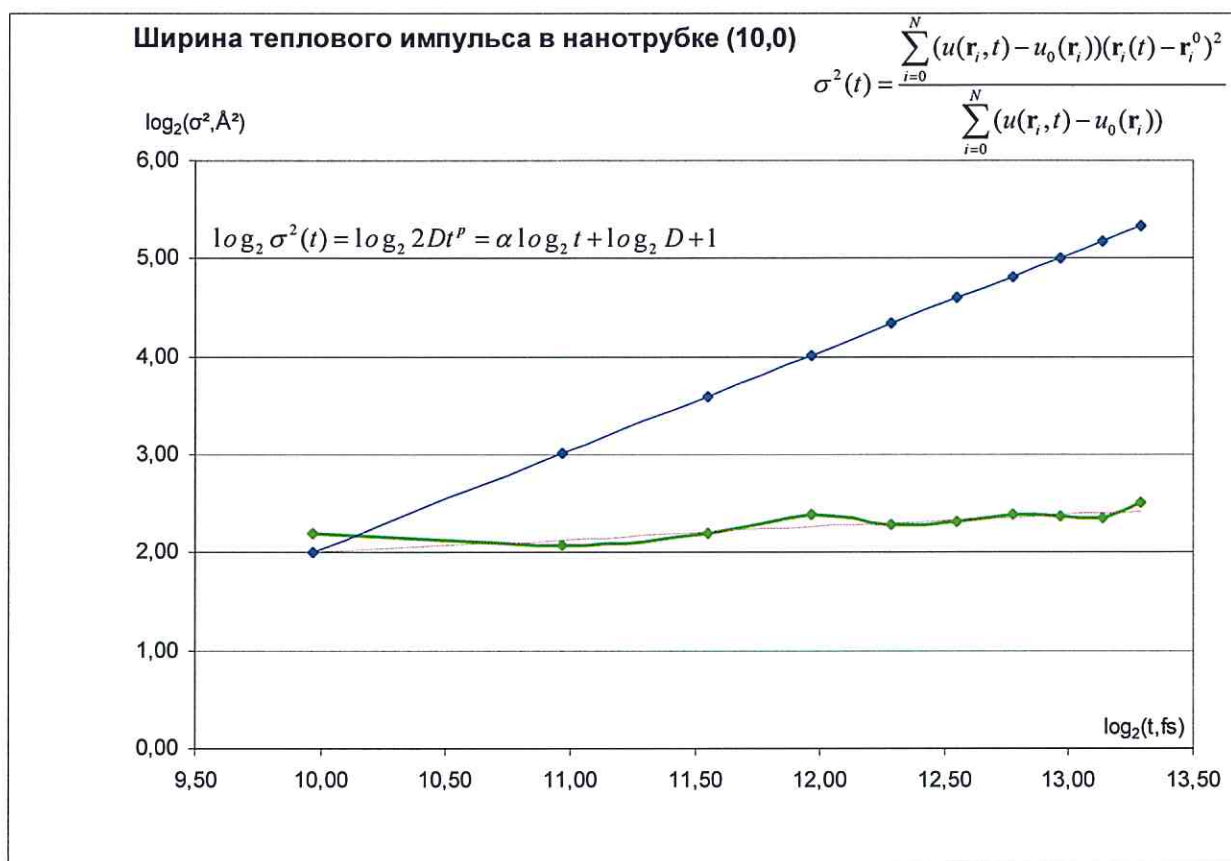


Рисунок 4.25. Распространение теплового импульса в листе графена при $T=300\text{K}$ (логарифмическая шкала)

Из-за уровня зашумлённости данных таким методом определить режим распространения тепла не удаётся.

На рис. 4.26 показан график зависимости температуры в центре образца от времени при начальной температуре 300K . На рис. 4.27 эта зависимость дана в логарифмической шкале. Красным цветом отображена линейная зависимость $y=2.13x-32.80$, полученная методом наименьших квадратов для этих значений. Вновь уровень шума не позволяет даже приблизительно определить режим распространения тепла.

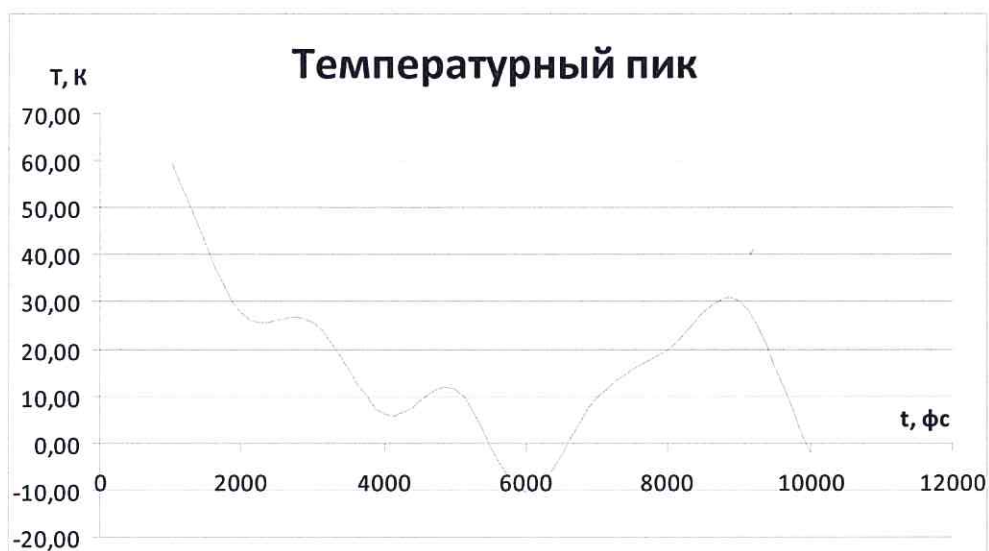


Рисунок 4.26. Спад пика энергии в нанотрубке (10,0) при $T=300\text{K}$ с течением времени



Рисунок 4.27. Спад пика энергии в нанотрубке (10,0) при $T=300\text{K}$ с течением времени (логарифмическая шкала)

На рис. 4.28 показаны графики зависимостей доли суммарной энергии теплового импульса, поданного на образец при начальной температуре 300К, от размера взятого интервала $[-x, x]$ для различных моментов времени. Горизонтальная линия соответствует доле энергии в 68.26%, пересечение этой линии графиком даёт значение искомого интервала $[-\Omega, \Omega]$.

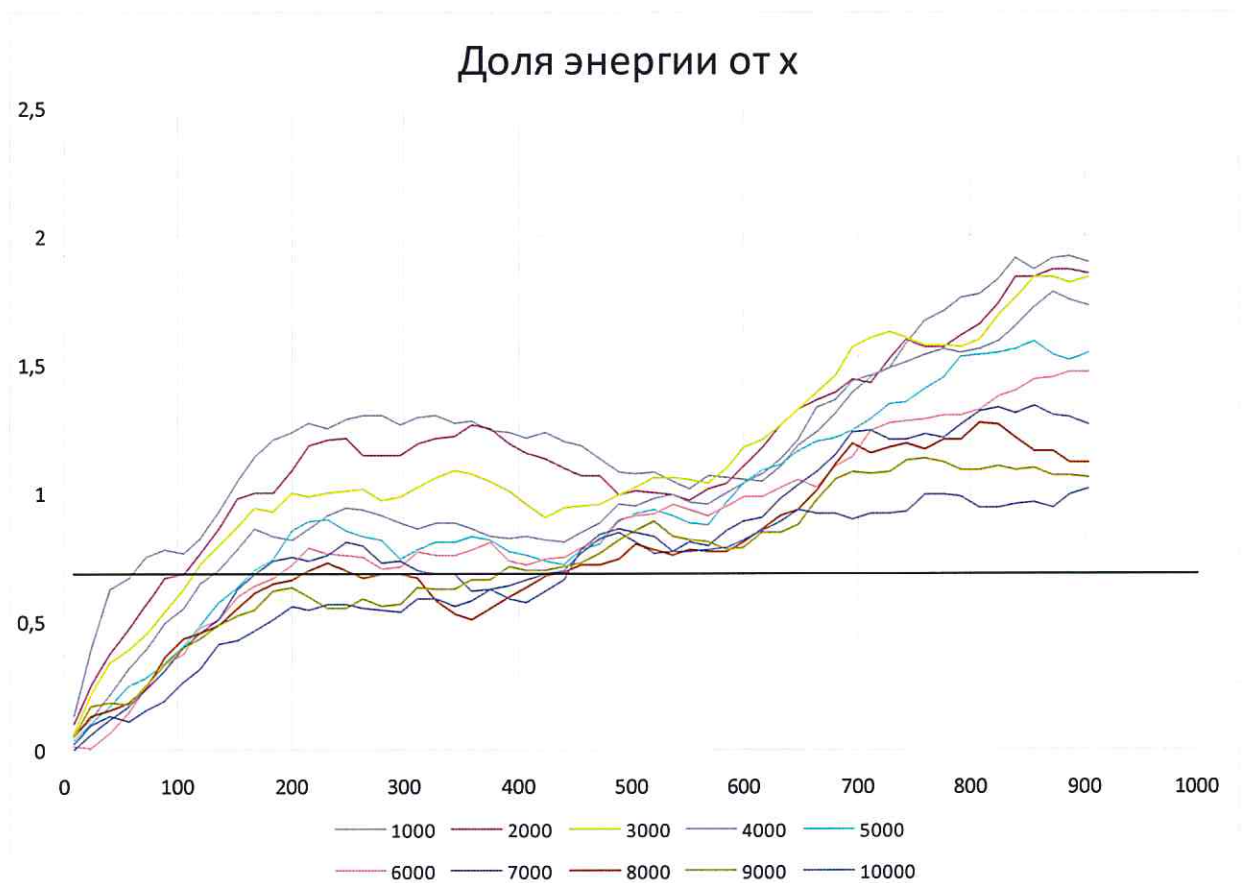


Рисунок 4.28. Зависимость доли энергии теплового импульса от размера интервала $[-x, x]$ при распространении тепла вдоль нанотрубки $(10,0)$ при $T=300\text{K}$. Графики соответствуют моменты времени, усреднённые за 1000 фс

На рис. 4.29 дана зависимость искомой Ω от времени в логарифмической шкале. Красным цветом отображена линейная зависимость $y=0.50x+3.44$, полученная методом наименьших квадратов для этих значений Ω . Наблюдается близкий к нормальному режим распространения тепла $\langle r^2 \rangle = \sigma^2 \sim t, p''=1.01$.

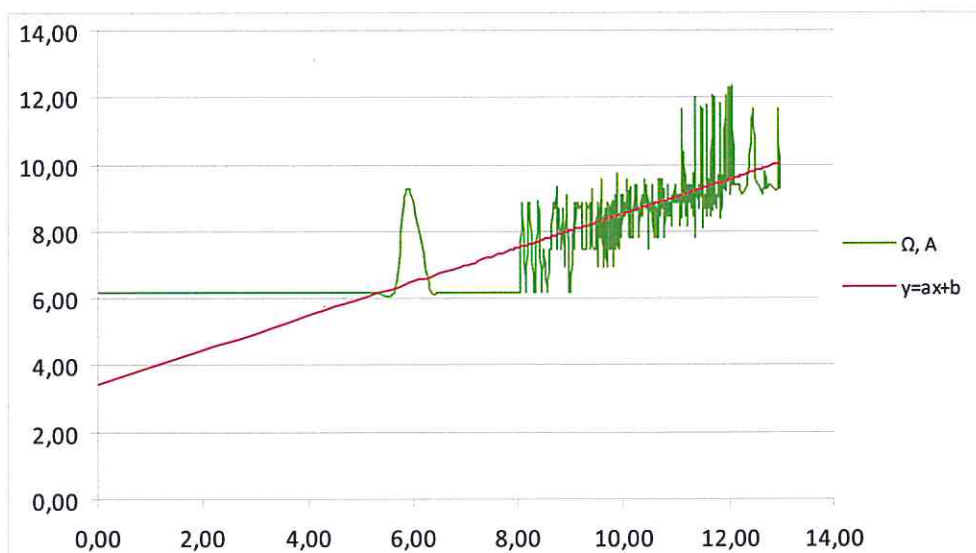


Рисунок 4.29. Зависимость размера интервала $[-\Omega, \Omega]$ от времени при распространении тепла вдоль нанотрубки (10,0) при $T=300\text{K}$. Графики отображают моменты времени, усреднённые за 10 фс (логарифмическая шкала)

Подобный анализ режимов теплопроводности, возникающих в нанотрубках, возможно проделать для целого набора начальных температур. Была проведена серия вычислительных экспериментов для температур трубок в 16К, 24К, 72К, 300К. Результат математической обработки возникающих в каждом случае распределений температур представлен в табл. 4.5. Значения в пустых ячейках таблицы необходимо уточнить.

Таблица 4.5. Характеристики теплопроводности нанотрубки (10,0).

Температура, К	16	24	72	300
ρ	1.25	*	*	0.12
ρ'	2.77	2.83	2.29	2.57
ρ''	1.42	1.23	1.15	1.01

При низких температурах, как и ранее для листов графена, наблюдается аномальный режим распространения тепла.

4.3 Описание аномальной теплопроводности с использованием дробно-дифференциальных уравнений

Явление распространения тепла в исследованных наноструктурах можно описать математической моделью аномальной теплопроводности, по аналогии с

моделью аномальной диффузии [102]. В этом случае явление аномальной теплопроводности представляется следующим уравнением в частных дробных производных:

$$\frac{\partial^\gamma u(x,t)}{\partial t^\gamma} = D \frac{\partial^\alpha u(x,t)}{\partial x^\alpha}, \quad (4.5)$$

где $u(x,t)$ – функция температуры, D – коэффициент температуропроводности ($D > 0$), α и γ – параметры, характеризующие порядок дробных производных по пространству и времени соответственно, из физических соображений выполняются соотношения $0 < \gamma \leq 1$ и $1 < \alpha \leq 2$, в предельном случае $\alpha = 2$, $\gamma = 1$ уравнение (4.5) описывает классическую теплопроводность. Параметр γ в уравнении (4.5) отвечает за появление субдиффузии – при $\alpha = 2$ уравнение (4.5) описывает аномальную теплопроводность с зависимостью $\langle r^2 \rangle = \sigma^2 \sim t^\gamma$. Параметр α отвечает за появление супердиффузии – при $\gamma = 1$ уравнение (4.5) описывает аномальную теплопроводность с зависимостью $\langle r^2 \rangle \sim t^{2/\alpha}$. Комбинация обоих параметров в уравнении (4.5) может описывать аномальную теплопроводность как в режиме субдиффузии, так и в режиме супердиффузии $\langle r^2 \rangle \sim t^{p=2\gamma/\alpha}$ ($0 < p < 2$) – в зависимости от того, какой механизм является преобладающим. Физических механизмов для появления субдиффузии в изучаемых образцах нет, поэтому можно принять $\gamma = 1$ и задать искомое уравнение аномальной теплопроводности как:

$$\frac{\partial u(x,t)}{\partial t} = D \frac{\partial^\alpha u(x,t)}{\partial x^\alpha}. \quad (4.6)$$

Данное уравнение получается из классического дифференциального уравнения теплопроводности заменой производной второго порядка по пространственной координате на дробную производную $\frac{\partial^\alpha u(x,t)}{\partial x^\alpha}$. Здесь выражение $\frac{\partial^\alpha u(x,t)}{\partial x^\alpha}$ представляет собой двустороннюю производную (взвешенная комбинация левосторонней и правосторонней):

$$\frac{\partial^\alpha u(x,t)}{\partial x^\alpha} = C_+ \frac{\partial_+^\alpha u(x,t)}{\partial x^\alpha} + C_- \frac{\partial_-^\alpha u(x,t)}{\partial x^\alpha}, \quad (4.7)$$

где $C_+ \geq 0$, $C_- \geq 0$ и $C_+ + C_- = 1$.

Необходимые односторонние дробные производные произвольного порядка α определяются формулами Римана-Лиувилля для дробной производной Лиувилля и Вейля:

$$\frac{\partial_+^\alpha u(x,t)}{\partial x^\alpha} = \frac{1}{\Gamma([\alpha]+1-\alpha)} \frac{d^{[\alpha]+1}}{dx^{[\alpha]+1}} \int_{-\infty}^x (x-\tau)^{[\alpha]-\alpha} u(\tau,t) d\tau, \quad (4.8)$$

$$\frac{\partial_-^\alpha u(x,t)}{\partial x^\alpha} = \frac{(-1)^{[\alpha]+1}}{\Gamma([\alpha]+1-\alpha)} \frac{d^{[\alpha]+1}}{dx^{[\alpha]+1}} \int_x^{+\infty} (\tau-x)^{[\alpha]-\alpha} u(\tau,t) d\tau. \quad (4.9)$$

Теплопроводность рассматривается на всём одномерном пространстве, т.е. граничных условий нет (решается задача Коши). В качестве начального условия может быть задана любая неотрицательная функция из пространства $L_1(-\infty;+\infty)$, т.е.

$$u(x,0) = f(x) \geq 0, \quad \int_{-\infty}^{+\infty} f(x) dx = U_0 < \infty, \quad \text{в том числе возможно и } u(x,0) = U_0 \cdot \delta(x). \quad \text{В}$$

дискретном представлении $\sum_{n=-l}^{+l} f(n) = U_0 < \infty$, где $2l$ – длина образца, в том числе

можно задать $u(0,0) = U_0 = 600$.

Решение задачи Коши для классического уравнения теплопроводности, т.е. уравнения (4.5) при значении параметров $\alpha = 2$ и $\gamma = 1$, даёт плотность вероятности $u(x,t)/U_0$, представляющую собой распределение Гаусса:

$$u(x,t) = \frac{U_0}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}, \quad \sigma^2 = 2Dt. \quad (4.10)$$

Однако решения уравнения аномальной теплопроводности (4.6) описываются функциями, большинство из которых не имеет явного аналитического выражения. Поэтому для его решения существуют различные численные методы, в том числе конечно-разностные методы и метод случайного блуждания [103].

В качестве примера приведём сравнение распределения температуры, полученного для листа графена при 16К, с решением соответствующего дробно-дифференциального уравнения по аналогии с тем, как это сделано в [14].

Решим уравнение (4.6) с параметрами $\alpha = \frac{2\gamma}{p''} = \frac{2}{1.81} = 1.10$ и $D=0.17$. На рис. 4.30

показано сравнение распределения температуры по образцу по результатам моделирования с такой дробно-дифференциальной моделью в момент времени 7 пс.

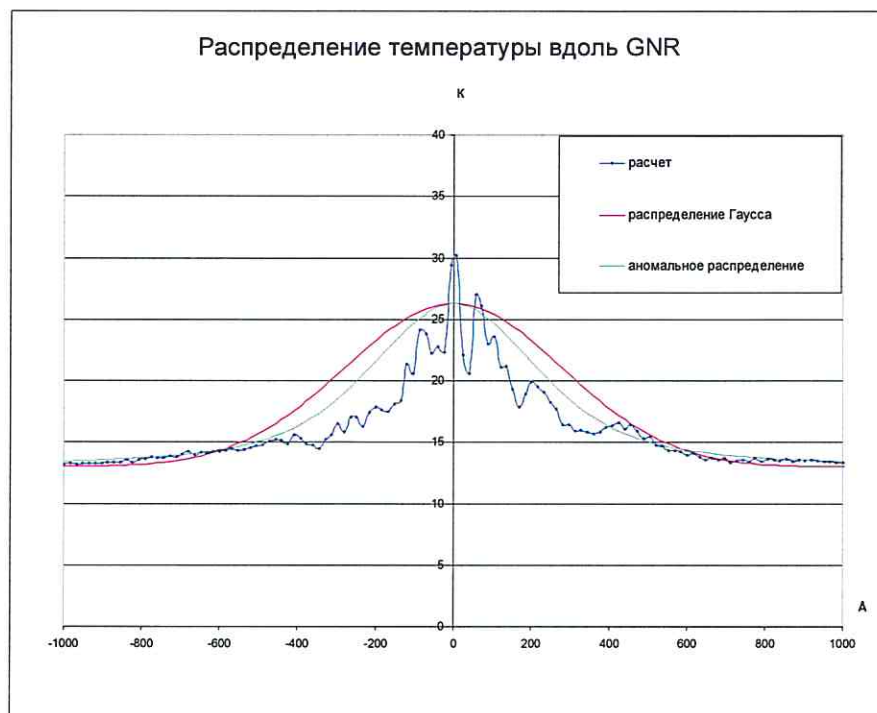


Рисунок 4.30. Графики распределения температуры листа графена при 16К (синий), распределение Гаусса (розовый) и решение соответствующего уравнения аномальной теплопроводности (зелёный) в момент времени 7 пс от начала воздействия

Можно видеть, что аномальная модель теплопроводности описывает возникающее распределение температуры более точно по сравнению с функцией Гаусса.

Выводы к главе 4

В главе 4 были исследованы процессы теплопроводности в наноструктурах, по результатам чего можно сделать следующие выводы.

Результаты молекулярного-динамического (микроскопического) моделирования распространения тепла в исследуемых образцах (листы графена и нанотрубки) свидетельствуют о том, что рассматриваемые процессы проявляют аномальный характер, причём степень аномальности или её отсутствие зависит от заданных начальных температур наноструктур.

Получаемые распределения температуры по образцам во времени могут быть достаточно точно описаны на макроуровне с использованием уравнения аномальной теплопроводности (4.5). Для различных образцов возможно определение параметров соответствующих дробно-дифференциальных уравнений.

Заключение

В ходе выполнения диссертационной работы были получены следующие основные результаты.

1. Разработана технология построения программных средств для молекулярно-динамического моделирования наносистем со сложными потенциалами межчастичного взаимодействия на графических процессорах. Разработаны методы отображения вычислительных процессов на архитектуру видеокарт, способы наследования классов в технологии CUDA, обеспечивающие возможность оснащения программ новыми потенциалами межчастичного взаимодействия без модификации основного кода.

2. Предложены подходы к повышению эффективности параллельных вычислений на графических процессорах, включающие использование гибридной модели расчётной области, методы размещения данных в памяти видеокарты с целью минимизации конфликтов доступа при параллельном обращении, оптимальное распределение операций по вычислительным потокам, выделение дополнительной памяти для создания копий координат взаимодействующих атомов. Дана оценка вычислительной сложности алгоритмов. На представительном ряде задач показана эффективность разработанных подходов.

3. На основе разработанной технологии программирования создан комплекс программ молекулярно-динамического моделирования наносистем на графических процессорах. Разработаны методы и средства визуализации результатов вычислительного процесса в реальном времени, методы интеграции программного обеспечения с имеющимися программными комплексами молекулярно-динамического моделирования.

4. С использованием разработанного программного обеспечения исследованы вопросы моделирования теплопроводности углеродных наносистем. Для описания аномальных режимов теплопроводности предложен подход, основанный на сочетании методов молекулярной динамики и дробно-дифференциального исчисления. Представлен алгоритм определения параметров макроскопической модели по данным

молекулярно-динамического моделирования. Таким образом, установлена связь между различными масштабами в описании аномальной теплопроводности.

Глоссарий

GPU	– Graphics Processing Unit, графический процессор
CPU	– Central Processing Unit, центральный процессор
NVIDIA SLI	– Scalable link interface, масштабируемый интерфейс связи
CUDA	– Compute Unified Device Architecture, архитектура параллельных вычислений от NVIDIA
OpenMP	– Open Multi-Processing, открытый стандарт для распараллеливания программ
OpenGL	– Open Graphics Library, спецификация, определяющая платформонезависимый (независимый от языка программирования) программный интерфейс для написания приложений, использующих двумерную и трёхмерную компьютерную графику
OpenCL	– Open Computing Language, открытый язык вычислений
OpenACC	– Open Accelerators, программный стандарт для параллельного программирования
ReaxFF	– Reactive force field, эмпирическое силовое поле
LAMMPS	– Large-scale Atomic/Molecular Massively Parallel Simulator, свободный пакет для классической молекулярной динамики
NAMD	– Nanoscale Molecular Dynamics, бесплатная программа для молекулярной динамики
ВМСРСС	– Международная конференция по вычислительной механике и современным прикладным программным системам
ALU	– Арифметическо-логическое устройство
DRAM	– Dynamic random access memory, динамическая память с произвольным доступом
GDDR	– Graphics Double Data Rate, подвид энергозависимой динамической памяти с произвольным доступом на графических картах
IEEE	– Institute of Electrical and Electronics Engineers, Институт инженеров электротехники и электроники
MD	– Молекулярная динамика
NVE	– Термодинамическое состояние характеризуется заданным набором частиц (N), заданным объёмом (V) и заданной энергией (E)
NVT	– Термодинамическое состояние характеризуется заданным набором частиц (N), заданным объёмом (V) и заданной температурой (T)
NPT	– Термодинамическое состояние характеризуется заданным набором частиц (N), заданным давлением (P) и заданной температурой (T)
μ VT	– Термодинамическое состояние характеризуется заданным химическим потенциалом (μ), заданным объёмом (V) и заданной температурой (T)
LJ	– Потенциала Леннарда-Джонса
VMD	– Visual Molecular Dynamics, программа визуализации молекул
PDB	– Proteins Data Bases, банк данных трёхмерных структур белков и нуклеиновых кислот
OS	– Операционная система
SWNT	– Одностенная нанотрубка
MWNT	– Многостенная нанотрубка
CNT	– Углеродная нанотрубка
REBO	– Эмпирический реакционный потенциал, учитывающий свойства связи
bcc	– Кубическая сингония
eV	– Электрон Вольт
CVD	– Химическое парофазное осаждение
BTE	– Boltzmann transport equation, кинетическое уравнение Больцмана

Список литературы

1. Jacobsen D.A. , Senocak I. Multi-level parallelism for incompressible flow computations on GPU clusters. // *Parallel Comput.* 2013. V.39(1). P. 1–20.
2. Hoshino T., Maruyama N., Matsuoka S., Takaki R. CUDA vs OpenACC: performance case studies with kernel benchmarks and a memory-bound CFD application. // *13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. CCGrid 2013, IEEE.* 2013. P.136–143.
3. Norman M. , Larkin J., Vose A., Evans K. A case study of CUDA FORTRAN and OpenACC for an atmospheric climate kernel. // *J. Comput. Sci.* 2015. V. 9. P. 1–6.
4. Rueda A.J., Noguera J.M., Luque A. A comparison of native GPU computing versus OpenACC for implementing flow-routing algorithms in hydrological applications. // *Comput. Geosci.* 2016. V. 87. P. 91–100.
5. Книжник А. А., Минкин А. С., Потапкин Б. В. Особенности реализации многочастичных потенциалов межатомного взаимодействия на гетерогенных вычислительных системах. // *Матем. моделирование.* 2013. Т. 25(10). С. 123–136.
6. Кривов М. А., Казеннов А. М. Сравнение вычислительных возможностей графических ускорителей NVidia при решении различных классов задач. // *Труды Всероссийской научно-практической конференции «Применение гибридных высокопроизводительных вычислительных систем для решения научных и инженерных задач», ННГУ, Нижний Новгород.* 2011. С. 18-23.
7. Schive H.-Y. , Chien C.-H., Wong S.-K., Tsai Y.-C., Chiueh T. Graphic-card cluster for astrophysics (GraCCA) – performance tests. // *New Astronomy.* 2008. V. 13(6). P. 418-435.
8. Anderson J. A. , Lorenz C. D., Travesset A. General purpose molecular dynamics simulations fully implemented on graphics processing units. // *Journal of Computational Physics.* 2008. V. 227. P. 5342–5359.
9. Жмуров А. А., Барсегов В. А., Трифонов С. В., Холодов Я. А., Холодов А. С. Эффективные генераторы псевдослучайных чисел при молекулярном моделировании на видеокартах. // *Компьютерные исследования и моделирование.* 2011. Т. 3(1). С. 287-308.

10. Emelyanov V.N. , Karpenko A.G., Kozelkov A.S., Teterina I.V., Volkov K.N., Yalozo A.V. Analysis of impact of general-purpose graphics processor units in supersonic flow modeling. // *Acta Astronautica*. 2017. V. 135. P. 198-207.
11. Kylasa S.B., Aktulga H.M., Gramab A.Y. PuReMD-GPU: A reactive molecular dynamics simulation package for GPUs. // *Journal of Computational Physics*. 2014. V. 272. P. 343–359.
12. Minkin A. S., Knizhnik A. A., Potapkin B. V. GPU implementations of some many-body potentials for molecular dynamics simulations. // *Advances in Engineering Software*. 2016. V. 111. P. 43–51.
13. Thompson A. and H. Cho. Lammmps/reaxff potential. April 2010. [Online]. Available: http://lammmps.sandia.gov/doc/pair_reax.htm
14. Aktulga H.M., Plimpton S.J., Thompson A., LAMMPS/User-Reax/C. [Online] http://lammmps.sandia.gov/doc/pair_reax_c.html.
15. Zheng, X. Li, L. Guo, Algorithms of GPU-enabled reactive force field (ReaxFF) molecular dynamics. // *J. Mol. Graph. Model*. 2013. V. 41. P. 1–11.
16. van Meel J.A., Arnold A., Frenkel D., Zwart S.F.P., Belleman R.G.. Harvesting graphics power for MD simulations. // *Molecular Simulation*. 2008. V.34. P.259-266.
17. Grand S.L. , Götz A.W., Walker R.C. SPFP: Speed without compromise - A mixed precision model for GPU accelerated molecular dynamics simulations. // *Computer Physics Communications*. 2013. V. 184(2). P. 374-380.
18. Plimpton. S. Fast Parallel Algorithms for Short-Range Molecular Dynamics. // *Journal of Computational Physics*. 1995. V. 117(1). P. 1-19.
19. Brown W.M., Wang P., Plimpton S.J., Tharrington S.J. Implementing molecular dynamics on hybrid high performance computers–short range forces. // *Computer Physics Communications*. 2011. V. 182(4). P. 898-911.
20. Tang Y.-H. , Karniadakis G.E. Accelerating dissipative particle dynamics simulations on GPUs: Algorithms, numerics and applications. // *Computer Physics Communications*. 2014. V. 185(11). P. 2809-2822.
21. Páll S. , Hess B. A flexible algorithm for calculating pair interactions on SIMD architectures. // *Computer Physics Communications*. 2013. V. 184(12). P. 2641-2650.

22. Stone J.E. et al. Accelerating molecular modeling applications with graphics processors. // *Journal of Computational Chemistry*. 2007. V. 28(16). P. 2618-2640.
23. Phillips J. , Stone J., Schulten K. Adapting a message-driven parallel application to GPU-accelerated clusters. // *Proceedings of the 2008 ACM/IEEE Conference on Supercomputing, SC*. 2008. P. 1-9.
24. Eastman P. et al. OpenMM 4: A Reusable, Extensible, Hardware Independent Library for High Performance Molecular Simulation. // *Journal of Chemical Theory and Computation*. 2013. V. 9(1).P. 461-469.
25. Harvey M.J. , GiuPoni G., Fabritiis G.D. ACEMD: accelerating biomolecular dynamics in the microsecond time scale. // *Journal of chemical theory and computation*. 2009. V. 5(6). P. 1632-1639.
26. Deng H. , Li X., Liu X., Wang G. Accelerating the Near Non-bonded Force Computation in Desmond with Graphic. // *Processing Units. 40th International Conference on Parallel Processing Workshops*. 2011. P. 191-198.
27. Roehm D. , Arnold A. Lattice Boltzmann simulations on GPUs with ESPResSo. // *The European Physical Journal Special Topics*. 2012. V. 210(1). P. 89-100.
28. Friedrichs M.S. , Eastman P., Vaidyanathan V., Houston M., Legrand S., Beberg A.L., Ensign D.L., Bruns C.M., Pande V.S. Accelerating molecular dynamic simulation on graphics processing units. // *Journal of Computational Chemistry*. 2009. V. 30(6). P. 864–872.
29. Trędak P., Rudnicki W. R., Majewska J. A. Efficient implementation of the many-body Reactive Bond Order (REBO) potential on GPU. // *Journal of Computational Physics*. 2016. V. 321(15). P. 556–570.
30. Боресков А. В., Харламов А. А., Марковский Н. Д., Микушин Д. Н., Мортиков Е. В., Мыльцев А. А., Сахарных Н. А., Фролов В. А. Параллельные вычисления на GPU. Архитектура и программная модель CUDA: Учебное пособие. – М.: МГУ, 2012. 336 с.
31. Abell G. C. Empirical chemical pseudopotential theory of molecular and metallic bonding. // *Physical Review B*. 1985. V. 31. P. 6184-6196.

32. Brenner D. W. Empirical potential for hydrocarbons for use in simulating the chemical vapor deposition of diamond films // *Physical Review B*. 1990. V. 42. P. 9458-9471.
33. Brenner D. W., Shenderova O. A., Harrison J. A. A second-generation reactive empirical bond order (REBO) potential energy expression for hydrocarbons. // *Journal of Physics: Condensed Matter*. 2002. V. 14. P. 783–802.
34. Tersoff J. New empirical model for the structural properties of silicon. // *Physical Review Letters*. 1986. V. 56. P. 632-635.
35. Tersoff J. New empirical approach for the structure and energy of covalent systems. // *Physical Review B*. 1988. V. 37. P. 6991-7000
36. Tersoff J. Empirical interatomic potential for carbon, with applications to amorphous carbon. // *Physical Review Letters*. 1988, V.61, P.2879-2882.
37. Tersoff J. Modeling solid-state chemistry: Interatomic potentials for multicomponent systems. // *Phys. Rev. B*. 1989. V. 39. P. 5566-5568.
38. Lewis G. N., Randall M. *Thermodynamics*. McGraw-Hill, New York. 1961. Appendix A7-8. P. 682.
39. Yin M.T., Cohen M.L. Will Diamond Transform under Megabar Pressures? // *Physical Review Letters*. 1983. V. 50. P. 2006-2009.
40. Yin M.T., Cohen M.L. Structural theory of graphite and graphitic silicon. // *Physical Review B*. 1983. V. 29 P.6996-6998.
41. Harrison W. A. Theory of the two-center bond. // *Physical Review B*. 1983. V. 27. P. 3592-3604.
42. Chelikowski J. R., Chou M. Y. Ab initio pseudopotential–local-density description of the structural properties of small carbon clusters. // *Physical Review B*. 1998. V. 37. P. 6504-6507.
43. Liu A. Y., Cohen M. L., Hass K. C., Tamor M. A. Structural properties of a three-dimensional all-sp² phase of carbon. // *Physical Review B*. 1991. V. 43. P. 6742-6745.
44. Berendsen H.J.C., Postma J.P.M., van Gunsteren W.F., DiNola A., Haak J.R. Molecular dynamics with coupling to an external bath. // *Journal of Chemical Physics*. 1984. V. 81. P. 3684–3690.

45. Nosé S. A unified formulation of the constant temperature molecular dynamics methods. // *Journal of Chemical Physics*. 1984. V. 81. P. 511–519.
46. Nosé S. A molecular dynamics method for simulations in the canonical ensemble. // *Molecular Physics*. 1984. V. 52. P. 255–268.
47. Hoover W.G. Canonical dynamics: Equilibrium phase-space distributions. // *Physical Review A*. 1985. V. 31. P. 1695–1697.
48. Hoover W.G. Constant-pressure equations of motion. // *Physical Review A*. 1986. V. 34. P. 2499–2500.
49. Dresselhaus M. S. Carbon nanotubes synthesis, structure, properties, and applications. Springer, Berlin. 2001. P. 447.
50. Li T.-L. Electronic Structures of Single-Wall Carbon Nanotubes National. // *Nano Communications*. 2007. V. 14(1). P. 42-47.
51. Allen M. P. Introduction to Molecular Dynamics Simulation Computational Soft Matter: From Synthetic Polymers to Proteins. // Lecture Notes, John von Neumann Institute for Computing. NIC Series. 2004. V. 23. P. 1-28.
52. Karlsson C., Davies T., Chen Z. Optimizing Process-to-Core Mappings for Application Level Multidimensional MPI Communications. // *Cluster Computing (CLUSTER)*, 2012 IEEE International Conf. Proceedings (Beijing, China, September, 24–28, 2012). Beijing, 2012. P. 486–494.
53. Zhang J. , Zhai J., Chen W. Process Mapping for MPI Collective Communications. // *Lecture Notes in Computer Science*. 2009. V. 5704. P. 81–92.
54. Chen H., Chen W., Huang J. H. MPIP: an Automatic Profile-Guided Parallel Process Placement Toolset for SMP Clusters and Multiclusters. // *ICS'06 Proceedings of the 20th annual international conference on Supercomputing*. Queensland, 2006. P. 353–360.
55. Amdahl G.M. Validity of the single processor approach to achieving large scale computer capabilities. // *Proceeding AFIPS '67, Spring joint computer conference*. 1967. P. 483-485.
56. Karp A. H. Measuring parallel processor performance. // *Communications of the ACM*. 1990. V. 33(5). P. 539-543.

57. Gustafson J. L. Reevaluating Amdahl's law. // *Communication ACM*. 1988. V. 31(5). P. 532-533.
58. Цилькер Б.Я., Орлов С.А. 2011. Организация ЭВМ и систем: учебник для вузов. СПб.: Питер. 688 с.
59. Moreland K., Oldfoeld R. Formal Metrics for Large-Scale Parallel Performance. // *International Conference on High Performance Computing*. 2015. V. 9137. P. 488-496.
60. Maulika S. P., Himanshu S M. Utilities for Efficient Usage of Large Biological Database. // *Phil. WCEC.S 2010*. V. II.
61. Shenderova O. A., Gruen D. M. Ultra Nanocrystalline Diamond Synthesis, properties and applications. William Andrew, Elsevier, US. 2012. P. 584.
62. Vereshchagin A. L. Phase Diagram of Ultrafine Carbon. // *Combustion, Explosion, and Shock Waves*. 2002. V. 38(3). P. 358-359.
63. Bundy F.P. The pressure-temperature phase transformation diagram for carbon. // *Carbon*. 1996. V. 34(2). P. 141-144.
64. Hany J. , Globus A., Jaffe R., Deardorff G. Molecular dynamics simulations of carbon nanotube-based gears. // *Nanotechnology*. 1997. V. 8. P. 95–102.
65. Steven J. Stuart, Alan B. Tutein, and Judith A. Harrison A reactive potential for hydrocarbons with intermolecular interactions. // *The Journal of Chemical Physics Phys*. 2000. V. 112. P. 6472-6486.
66. Greenwood N. N. and Earnshaw A. *Chemistry of the Elements*. Pergamon, New York. 1984.
67. Kaxiras E. and K. C. Pandey Energetics of defects and diffusion mechanisms in graphite. // *Phys. Rev. Lett*. 1988. V. 61. P. 2693.
68. Ran Xu, Bin Liu, Xiaoqiao He, Dechang Li Kinetic Energy-Based Temperature Computation in Non-Equilibrium Molecular Dynamics Simulation. // *Journal of Computational and Theoretical Nanoscienc*. 2012. V. 9. P. 428–433.
69. Maximova T. , Keasar C. A novel algorithm for non-bonded-list updating in molecular simulations. // *J. Comp. Biol*. 2006. V. 13. P. 1041–1048.
70. Khvesyuk V. I. and A. S. Skryabin. Heat Conduction in Nanostructures. // *High Temperature*. 2017. V. 55(3). P. 434–456.

71. Елецкий А.В., Искандарова И.М., Книжник А.А., Красиков Д.Н. Графен: методы получения и теплофизические свойства. // Успехи физических наук. 2011. Т. 181(3). С. 233-268.
72. Vermeersch B., Carrete J., Mingo N., Shakouri A. Superdiffusive heat conduction in semiconductor alloys. I. Theoretical foundations. // Physical Review B. 2015. V. 91. P. 085202-085202-9
73. Balandin A. A. Thermal properties of graphene and nanostructured carbon materials. // Nat. Mater. 2011. V. 10. P. 569–81.
74. Ghosh S., Bao W., Nika D. L., Subrina S., Pokatilov E. P., Lau C. N. et al. Dimensional crossover of thermal transport in fewlayer graphene. // Nat. Mater. 2010. V. 9. P. 555–8.
75. Balandin A. A. et al. Superior thermal conductivity of single layer graphene. // Nano Lett. 2008. V. 8. P. 902–907.
76. Cai, W. et al. Thermal transport in suspended and supported monolayer graphene grown by chemical vapor deposition. Nano Lett. 2010. V. 10. P. 1645–1651.
77. Jauregui L. A. et al. Thermal transport in graphene nanostructures: Experiments and simulations. // ECS Trans. 2010. V. 28. P. 73–83.
78. Faugeras C. et al. Thermal conductivity of graphene in Corbino membrane geometry. // ACS Nano. 2010. V. 4. P. 1889–1892.
79. Murali R., Yang Y., Brenner K., Beck T. and Meindl J. D. Breakdown current density of graphene nanoribbons. // Appl. Phys. Lett. 2009. V. 94. P. 243114.
80. Seol, J. H. et al. Two-dimensional phonon transport in supported graphene. // Science. 2010. V. 328. P. 213–216 .
81. Ghosh S., Calizo I., Teweldebrhan D., Pokatilov E. P., Nika D. L., Balandin A. A., Bao W., Miao F. and Lau C. N. Extremely high thermal conductivity in graphene: prospects for thermal management application in nanoelectronic circuits. // Appl. Phys. Lett. 2008. V. 92. P. 151911.
82. Xu X. , Pereira L. F. C. , Wang Y., Wu J., Zhang K., Zhao X., Bae S., Tinh Bui C., Xie R., Thong J. T. L., Hong B. H., Loh K. P., Donadio D., Li B., and Özyilmaz B. Length

dependent thermal conductivity in suspended single-layer graphene. // *Nat. Commun.* 2014. V.5. P. 3689.

83. Chen S., Moore A. L., Cai W., Suk J. W. , An J., Mishra C., Amos C., W. Magnuson C., Kang J., Shi L. and Ruof R. S.. Raman Measurements of Thermal Transport in Suspended Monolayer Graphene of Variable Sizes in Vacuum and Gaseous Environments. // *American Chemical Society.* 2011. V. 5(1). P. 321–328.

84. Lee J.-U., Yoon D., Kim H., Lee S. W., Cheong, H. Thermal conductivity of suspended pristine graphene measured by Raman spectroscopy. // *Phys. Rev. B.* 2011. V. 83. P. 081419.

85. Evans W. J., Hu L., Keblinsky P. Thermal conductivity of graphene ribbons from equilibrium molecular dynamics: Effect of ribbon width, edge roughness, and hydrogen termination. // *Appl. Phys. Lett.* 2010. V. 96. P. 203112.

86. Qiu B. and Ruan X. Molecular dynamics simulations of thermal conductivity and spectral phonon relaxation time in suspended and supported graphene. [Online] arXiv:111.4613v1. 2011.

87. Zhang H., Lee G. and Cho K. Thermal transport in graphene and effects of vacancies *Phys. Rev. B.* 2011. V. 84. P. 115460.

88. Wei Z., Ni Z., Bi K., Chen M. and Chen Y. In-plane lattice thermal conductivities of multilayer graphene films. // *Carbon.* 2011. V. 49. P. 2653.

89. Cao A. Molecular dynamics simulation study on heat transport in monolayer graphene sheet with various geometries. // *J. Appl. Phys.* 2012. V. 111. P. 083528.

90. Cheng. L., Kumar S. Thermal transport in graphene supported copper. // *J. Appl. Phys.* 2012. V. 112. P. 043502.

91. Hu J., Ruan X., Chen Y.P. Thermal conductivity and thermal rectification in graphene nanoribbons: A molecular dynamics study. // *Nano Lett.* 2009. V. 9. P. 2730–2735.

92. Yu C., Zhang G. Impacts of length and geometry deformation on thermal conductivity of grapheme nanoribbons. // *Journal of Applied Physics.* 2013. V. 113. P. 044306.

93. Pereira L. F. C., Donadio D. Divergence of the thermal conductivity in uniaxially strained graphene. // *Physical Review B.* 2013. V. 87. P. 125424.

94. Alofi A., Srivastva G.P. Phonon conductivity in graphene. // J. Appl. Phys. 2012. V. 112. P. 013517.
95. Fugallo G. et al. Thermal conductivity of graphene and graphite: collective excitations and mean free paths. // Nano Lett. 2014. V. 14. P. 6109–6114.
96. Munoz E., Lu J., Yakobson B.I. Ballistic thermal conductance of graphene ribbons. // Nano Lett. 2010. V. 10. P. 1652–1656.
97. Aksamija Z., Knezevic I. Lattice thermal conductivity of graphene nanoribbons: Anisotropy and edge roughness scattering. // Appl. Phys. Lett. 2011. V. 98. P. 141919.
98. Gu X.; Yang R. First-principles prediction of phononic thermal conductivity of silicene: A comparison with graphene. // Journal of Applied Physics. 2015. V. 117. P. 025102.
99. Klemens P. G. Theory of the A-plane thermal conductivity of graphene. // J. Wide Bandgap Mater. 2000. V. 7. P. 332–339.
100. Nika D. L. , Pokatilov E. P., Askerov A. S. and Balandin A. A. Phonon thermal conduction in graphene: Role of Umklapp and edge roughness scattering. // Phys. Rev. B. 2009. V. 79. P. 155413.
101. Lindsay L. , Broido D. A. and Mingo N. Flexural phonons and thermal transport in graphene. // Phys. Rev. B. 2010. V. 82. P. 115427.
102. Ревизников Д.Л., Сластушенский Ю.В. Применение дробно-дифференциального исчисления для описания аномальной диффузии. // Вестник Московского авиационного института – М.: Изд-во МАИ. 2011., Т. 18(4). С. 76–82.
103. Сластушенский Ю.В. Модель случайного блуждания для уравнения аномальной диффузии. // Научно-технический вестник Поволжья. 2011. № 5. С. 242–246.
104. Ревизников Д.Л., Сластушенский Ю.В. Численное моделирование аномальной диффузии бильярдного газа в полигональном канале. // Математическое моделирование. 2013. № 5. С. 3–14.
105. Ревизников Д.Л., Семёнов С.А. Особенности молекулярно-динамического моделирования наносистем на графических процессорах. // Программная инженерия. – М.: Издательство «Новые технологии». 2013. № 2. С. 31–35.

106. Семёнов С.А. Использование графических процессоров в молекулярно-динамическом моделировании. // Электронный журнал труды МАИ. 2013. № 65. С. 1-6.
107. Семенов С.А. Реализация гибридного алгоритма молекулярно-динамического моделирования на графических вычислителях. // Научно-технический вестник Поволжья. – Казань. 2013г. № 6. С. 415-422.
108. Семенов С.А., Ревизников Д.Л. Эффективное использование программируемых графических процессоров в задачах молекулярно-динамического моделирования. // Системы и средства информатики. 2017. № 4.
109. Медведев А.А., Масленников А.А., Магидович С.С., Колесник С.С., Семёнов С.А., Особенности построения и использования Veowulf-кластера гибридных вычислительных систем. // Московская молодёжная научно-практическая конференция «Инновации в авиации и космонавтике — 2012». Сборник тезисов докладов. – М.: ООО «Принт-салон». 2012. С. 69-70.
110. Ревизников Д.Л., Семёнов С.А. Оптимизация алгоритмов молекулярно-динамического моделирования для параллельных расчетов наносистем на графических процессорах. // Материалы IX конференции по неравновесным процессам в соплах и струях (NPNJ 2012). –М.: Издательство МАИ. 2012. С. 510-512.
111. Семёнов С.А. Разработка программы молекулярно-динамического моделирования на графических процессорах. // Сборник материалов XI международной конференции «Авиация и космонавтика — 2012». – М. 2012. С. стр. 397-398.
112. Семёнов С.А. Вычисление градиента потенциала Бреннера на видеокартах в задачах молекулярной динамики наноструктур. // Труды 55-й научной конференции МФТИ: Всероссийской научной конференции «Проблемы фундаментальных и прикладных естественных и технических наук в современном информационном обществе», Научной конференции «Современные проблемы фундаментальных и прикладных наук в области физики и астрономии», Всероссийской молодежной научной конференции «Современные проблемы фундаментальных и прикладных наук». Управление и прикладная математика – М.: МФТИ. 2012. Т. 2. С. 83-85.

113. Семёнов С.А. Измерение и сравнение скорости вычислений на *gpu* в задачах молекулярной динамики. // Материалы XVIII международной конференции по Вычислительной механике и современным прикладным программным системам (ВМСППС'2013). – М.: Изд-во МАИ. 2013. С. 134-136.
114. Семёнов С.А., Сластушенский Ю.В. Ускорение вычислений в задачах моделирования динамики частиц с помощью графических процессоров. // Сборник материалов 15-й международной конференции «Авиация и космонавтика — 2016». – М.: Люксор. 2016. С. 397-398.
115. Семенов С.А., Сластушенский Ю.В., Ревизников Д.Л. Численный анализ режимов теплопроводности в углеродных наноструктурах. // Сборник материалов 16-й международной конференции «Авиация и космонавтика – 2017».