МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**
Кафедра дискретной математики и информационных технологий

ИССЛЕДОВАНИЕ МАСШТАБИРУЕМОСТИ ПРОГРАММНОЙ СИСТЕМЫ МОДЕЛИРОВАНИЯ ПРОЦЕССОВ ПЕРЕНОСА ПРИ РЕАЛИЗАЦИИ НА СОПРОЦЕССОРАХ NVIDIA

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 271 группы
направления 09.04.01 — Информатика и вычислительная техника
факультета КНиИТ
Щербакова Ильи Алексеевича

Научный руководитель
доцент, к. ф.-м. н. _____ А. Д. Панфёров

Заведующий кафедрой
доцент, к. ф.-м. н. _____ Л. Б. Тяпаев

Саратов 2022

ВВЕДЕНИЕ

В настоящее время уровень развития вычислительной техники позволяет выполнять прямое численное моделирование многих сложных физических процессов «из первопринципов» для решения научных проблем и практических прикладных задач. В определенном смысле на стоимость численного эксперимента точной математической модели требуется меньше затрат, чем на проведение натурного эксперимента. Такое имитационное моделирование максимально требовательно к вычислительным ресурсам и может быть продуктивным только при их эффективном использовании.

Основной проблемой при работе с современными высокопроизводительными вычислительными системами является то обстоятельство, что для их архитектур характерен многоуровневый параллелизм и гетерогенность. Только умение полноценно и эффективно использовать эти особенности открывает путь к получению новых интересных результатов. Важна вся совокупность факторов: корректная физико-математическая постановка задачи, алгоритм решения, эффективные высокоточные численные методы, выбор алгоритмического языка и компилятора, поддерживающих современные технологии программирования, эффективные библиотеки. Правильные решения на каждом из этих шагов могут обеспечить разумное время выполнения программного модуля и корректное решение задачи.

Целью представляемой выпускной квалификационной работы является адаптация и развитие системы моделирования процессов переноса носителей заряда в графене для её эксплуатации на гибридных вычислительных системах с несколькими вычислительными ускорителями.

Для достижения поставленной цели были поставлены следующие задачи:

- Исследовать варианты реализации вычислительной модели;
- Исследовать инструменты программирования сложных вычислительных задач на графических ускорителях NVIDIA;
- Провести тестирование и сравнение различных вариантов программных решений с использованием выбранных инструментов;
- Проанализировать полученные результаты.

В первой главе рассматриваются аппаратная архитектура вычислительных ускорителей компании NVIDIA и программная технология CUDA, необ-

ходимые для использования их возможностей. В настоящее время линейка этих ускорителей абсолютно доминирует в отрасли высокопроизводительных вычислений. На июнь 2022 года в списке TOP500 присутствуют 169 гетерогенных систем, использующих наряду с универсальными CPU вычислительные ускорители различных типов. В 146 системах — это ускорители NVIDIA различных поколений. Все семь систем из TOP500, эксплуатируемые в России, используют ускорители этого типа. СГУ располагает мощной вычислительной системой с четырьмя ускорителями NVIDIA A100 последнего поколения, на которой проводилось тестирование разработанного программного решения.

Во второй главе представлена физической модель для описания процесса действия внешнего электрического поля на носители заряда в графене. Она реализована в двух вариантах: первый основывается на стандартном приближении безмассовых фермионов MLF (massless fermions), второй вариант использует TBM (tight binding model) позволяя расширить область применения модели. Обе версии модели представляются в форме систем обыкновенных дифференциальных уравнений (ОДУ). Различия определяются зависимостями коэффициентов уравнений от параметров моделируемого процесса.

В третьей главе рассматриваются и выбираются инструменты для решения системы ОДУ и распараллеливания процесса в двумерном пространстве параметров.

Четвертая глава представляет описание программной реализации системы моделирования для использования на гетерогенном мультипроцессоре с несколькими вычислительными ускорителями NVIDIA и результатов её тестирования. Работоспособность и эффективность реализованного программного решения оценивается по ряду параметров в условиях использования различных версий модели и различных параметрах моделируемого процесса. Основное внимание уделяется эффективности использования нескольких вычислительных ускорителей.

Магистерская работа состоит из введения, 4 разделов, заключения, списка использованных источников и 2 приложений. Общий объем работы — 74 страницы, из них 48 страниц — основное содержание, включая 42 рисунка, список использованных источников информации — 30 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

В первой главе описывается программная технология CUDA, которая является разработкой NVIDIA. CUDA предоставляет удобные инструменты, чтобы написать приложения для ускорителей этой компании. Архитектура CUDA является современной для видеокарт и развивается вместе с выходом новых поколений графических процессоров и используется в подавляющем большинстве суперкомпьютеров с видеоускорителями из списка TOP-500.

Во второй главе приведен вид решаемой системы ОДУ, определена зависимость переменных коэффициентов этой системы от параметров моделируемого процесса и времени. Определены различия представления коэффициентов модели в версиях на основе приближения MLF (massless fermions) и TBM (tight binding model). Определена процедура воспроизведения функции распределения носителей заряда по состояниям в двумерном импульсном пространстве. Представлены интегральные выражения, определяющие наблюдаемые процессы переноса заряда через фенкцию распределения.

В третьей главе приводятся основные подходы и инструменты, использованные для реализации программы на CPU и GPU. Для CPU входят следующие библиотеки:

- GSL;
- MPI.

Используемые библиотеки для реализации на GPU:

- Boost;
- Odeint;
- OpenMP;
- Thrust.

Описывается организация работы с вычислительным кластером СГУ, приводятся основные очереди для вычислительных задач. Указываются технические характеристики вычислительного узла, а также программное обеспечение, установленное на узле кластера, необходимое для корректного запуска задачи. Демонстрируется сценарный скрипт для запуска вычислительной задачи в специальном формате, представляется вывод диагностической программы с полными характеристиками GPU.

В четвёртой главе представляется описание программной реализации системы моделирования для использования на гетерогенном мультипроцес-

соре с несколькими вычислительными ускорителями NVIDIA и проведено тестирование её работы в процессе моделирования с реалистическими физическими параметрами. Алгоритм моделирования требовал обеспечивать возможность получать решение задачи Коши для миллионов различных вариаций параметров, определяющих поведение коэффициентов используемой системы ОДУ. Ключевым было требование реализации вычислительного процесса на базе технологии CUDA. Это обусловлено необходимостью обеспечить эффективное использование программного комплекса на современных высокопроизводительных кластерных системах, построенных с использованием вычислительных ускорителей NVIDIA.

Для решения данной задачи было принято решение выполнять декомпозицию физической области (сетки в двумерном импульсном пространстве) на этапе подготовки задания на CPU исходя из доступного количества ускорителей. Такое решение позволило эффективно распределять вычислительную нагрузку между любым количеством графических ускорителей вычислительного узла.

Для того чтобы распределить вычислительную нагрузку между несколькими графическими устройствами, необходимо распределить между ними данные векторов, соответствующие внутренним элементам вычислительной сетки. Для этого реализована процедура формирования линейного массива из элементов двумерной вычислительной сетки. А уже этот линейный массив делится на заданное число блоков, оформляемых в виде векторов с использованием функционала библиотеки Thrust. Выбор такого разбиения позволяет нам совершать одинаковое количество операций интегрирования на каждом GPU. Поскольку вектора в ходе итерационного процесса не модифицируются, то их нужно разместить в памяти CPU и GPU только один раз перед основным стартом работы процесса интегрирования того или иного метода.

Тестирование и демонстрация проводились с использованием реалистичных параметров моделируемого процесса. Область в импульсном пространстве, для которой воспроизводилось состояние функции распределения после завершения действия внешнего поля, была определена заранее. Для варьирования вычислительной сложности задачи использовались два варианта равномерной покрывающей сетки с количеством узлов, отличающимся в 4 раза. На этом этапе использовались два метода интегрирования — RKCK и RKDP,

реализованных в библиотеке Odeint.

По полученным результатам сделали вывод, что метод RKCK оказывается быстрее при всех сочетаниях рассматривавшихся параметров. На основании этого было принято решение в дальнейшем использовать только его. Такое ограничение никак не сказывается на общности представляемых далее результатов.

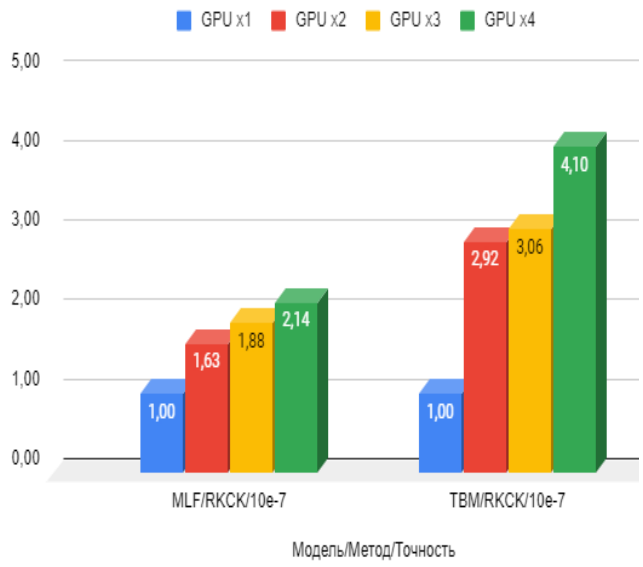

Рисунок 1 – Ускорение при точности 10^{-7}

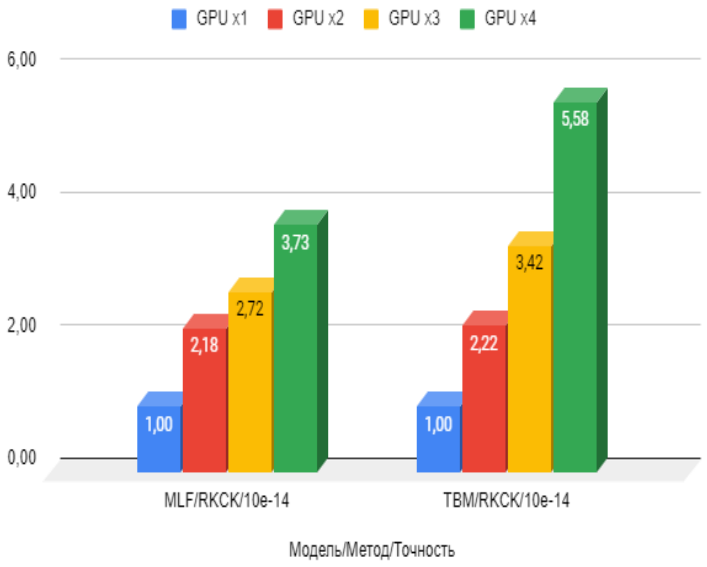

Рисунок 2 – Ускорение при точности 10^{-14}

При увеличении количества задействованных GPU ожидаемо наблюдалась тенденция уменьшения времени подсчёта, которая продемонстрирована на рисунках 1, 2. Особенно хорошо отзывается на количество доступных ускорителей модель TBM. При этом модель MLF показала лучшие абсолютные временные показатели на всех тестах и при использовании различных методов решения и при разной размерности матричной сетки. Это определяется простотой аналитического представления параметров (переменных коэффициентов системы ОДУ) по сравнению с TBM версией модели.

Поскольку исходная версия программы моделирования разрабатывалась для классических CPU и вычислительные узлы с GPU могут быть не всегда доступны, было проведено сравнительное тестирование реализаций программ для CPU и GPU при двух наборах параметров.

Полученные результаты подтверждают, что вычислительные ускорители оправдывают свое название и действительно обеспечивают кратное уменьшение времени решения вычислительно сложных задач. При использовании

1 GPU мы можем получить прибавку по скорости выполнения от 8 до 57 раз в зависимости от набора параметров. При использовании 4 GPU достижимо ускорение от 13 до 50 раз. При этом важную роль играет настройка вычислительных параметров процесса моделирования и версии модели. Так при задании высокой точности промежуточных вычислений 10^{-14} модель MLF на одном GPU обгоняет реализацию на CPU. Однако модель TBM начинает сравниваться и обгонять CPU при количестве использованных GPU 2 и более. Причина этого связана с архитектурой современных GPU, в которых вычисления с одинарной точностью и с двойной точностью выполняются раздельно на специализированных ядрах, и ядер для вычислений с двойной точностью в два раза меньше. Это обстоятельство проявляет себя по-разному в версиях модели MLF и TBM в связи с отмечавшейся выше разницей в их вычислительной сложности. Можно отметить, что фактическая производительность и CPU и GPU сильно зависит от задаваемой точности вычислений. Повышение требований к точности вычислений приводит к падению производительности примерно на порядок для CPU и существенно больше при использовании GPU.

Целевым назначением системы моделирования является воспроизведение характеристик процессов переноса заряда. Процедуры вычисления поверхностных плотности заряда и компонентов тока, возникающих в модели в результате действия внешнего электрического поля, выполняются с использованием численного интегрирования. Интегрирование выполняется с использованием промежуточных результатов, полученных на этапе воспроизведения функции распределения, и не требует существенных вычислительных ресурсов. Оно было реализовано в форме постобработки отдельной программой.

Таблица 1 – Плотность сетки

Модель	512x512 $10e^{-7}$	512x512 $10e^{-14}$	1024x1024 $10e^{-7}$	1024x1024 $10e^{-14}$
MLF	0,00140312	0,00140312	0,00140309	0,00140309
TBM	0,00140626	0,00140627	0,00140623	0,00140623

В таблице 1 представлены результаты вычисления плотности носителей заряда в расчете на одну элементарную ячейку для реалистичного набора параметров поля. При построении таблицы 1 использовались разные версии физической модели (MLF и TBM) при различных настройках процедуры моделирования.

Из представленных данных следует, что в условиях использования заданных физических параметров выбор версии модели приводит к различию в воспроизводимом значении плотности носителей на уровне 0.2%. Варьирование же параметров вычислительной процедуры (плотность используемой расчетной сетки, требования к точности на этапе решения систем ОДУ) практически не влияют на результат. В наглядной форме это представлено в таблице 2.

Таблица 2 – Плотность сетки

Модель	512x512 $10e^{-7}$	512x512 $10e^{-14}$	1024x1024 $10e^{-7}$	1024x1024 $10e^{-14}$
MLF	100%	100%	99,99759%	99,99749%
TBM	100%	100%	99,99759%	99,99757%

Это открывает возможность оптимизировать процедуру моделирования путем снижения требований к точности промежуточных вычислений. Исследование такой возможности и конкретных границ её использования выходит за рамки представленного исследования. В рамках выполнявшейся работы выбор параметров вычислительной процедуры определялся в основном требованиями корректного нагрузочного тестирования.

Таблица 3 – Перенос заряда

Модель	512x512 $10e^{-7}$	512x512 $10e^{-14}$	1024x1024 $10e^{-7}$	1024x1024 $10e^{-14}$
MLF (j_1)	-4,9204E-05	-4,9204E-05	-4,9203E-05	-4,9202E-05
MLF (j_2)	7,0810E-13	2,4645E-18	-7,6937E-14	-1,3278E-11
TBM (j_1)	-4,9296E-05	-4,9297E-05	-4,9294E-05	-4,9295E-05
TBM (j_2)	-3,3406E-05	-3,3405E-05	-3,3406E-05	-3,3404E-05

Выбор оптимальных параметров вычислительной процедуры для конкретных физических параметров моделируемого процесса является отдельной задачей и выходит за рамки представляемого исследования.

Значения компонент тока также вычислялись и представлены в таблице 3. Здесь результат более интересен. Если первые компоненты тока (в этом

направлении действует внешнее поле) также демонстрируют высокую степень совпадения вне зависимости от выбора версии модели и настроек процесса моделирования, то вторые компоненты тока при смене версии модели ведут себя по-разному. Это интересный результат, который будет являться предметом рассмотрения разработчиков модели.

ЗАКЛЮЧЕНИЕ

В работе представлены результаты адаптации программной системы моделирования поведения графена во вешних электрических полях с произвольной зависимостью от времени для её использования на высокопроизводительных гетерогенных вычислительных системах на основе GPU ускорителей компании NVIDIA. Реализована возможность декомпозиции модели по числу доступных в системе ускорителей. Проведено подробное исследование скоростных характеристик разработанных программ в зависимости от используемых алгоритмов интегрирования, требований к точности результатов, числа задействованных GPU.

Проведено сравнительное тестирования версий программ для GPU и CPU. Продемонстрировано, что использование GPU эффективно и обеспечивает кратное уменьшение времени решения задачи.

Разработаны программные реализации для двух версий физической модели: классического приближения свободных фермионов и более точной модели полного учета взаимодействия ближайших соседей по атомной решетке. Проведено тестирование ресурсных требований этих двух вариантов. Первый из них более прост и менее требователен к ресурсам, но имеет ограничения по физическим параметрам, для которых возможно моделирование. Второй вариант модели сложнее и более требователен к вычислительным ресурсам. Поэтому его использование имеет смысл только в тех диапазонах параметров задачи, когда неточность результатов использования первого варианта модели становится существенной. Проведенные численные эксперименты позволили определить границы применимости рассматривавшихся моделей.

Поставленные задачи были решены в полном объёме. Представленные результаты показывают высокую эффективность использования GPU ускорителей при моделировании процессов переноса заряда в рассматриваемом материале. Это обеспечивает в перспективе возможность детального исследования характеристик таких процессов, анализа их особенностей и поиска новых практических приложений.

Основные источники информации

1 *Novoselov, K. S.* Two-Dimensional gas of massless Dirac fermions in graphene / K. S. Novoselov, A. K. Geim. — Nature, 2005. — Pp. 197–200.

2 *Smolyansky, S. A.* Residual currents generated from vacuum by an electric field pulse in 2+1 dimensional qed models / S. A. Smolyansky, D. V. Churochkin, V. V. Dmitrievand A. D. Panferov, B. Kampfer // *EPJ Web Conf.* — 2017. — Pp. 138–139.

3 *K. Ahnert D. Demidov, M. Milansky.* Solving ordinary differential equationson GPUs / M. Milansky. K. Ahnert, D. Demidov. — Cham: Springer, 2014. — 412 pp.

4 *Боресков, А.В.* Основы работы с технологией CUDA / А.В. Боресков, А.А. Харламов. — Москва: ДМК Пресс, 2010. — 312 с.

5 *Cash, J. R.* A variable order runge-kutta method for initial value problems with rapidly varying right-hand sides / J. R. Cash, A. H. Karp // *ACM Transactions on Mathematical Software*. — 1990. — Pp. 201–222.

6 *Теплов, А.М.* Об одном подходе к сравнению масштабируемости параллельных программ // Вычислительные методы и программирование. / А.М. Теплов. — М.: Изд-во МГУ, 2014. — С. 697–711.

7 *Богданов, П.Б.* Применение планировщика для эффективного обмена данными на суперкомпьютерах гибридной архитектуры с массивно-параллельными ускорителями // Вычислительные методы и программирование / П.Б. Богданов, А.А. Ефремов, А.В. Горобец, С.А. Суков. — М.: Изд-во МГУ, 2013. — С. 122–134.

8 *Боум, А.* Квантовая механика: основы и приложения / А. Боум. — Москва: М.: Мир, 1990. — 710 с.

9 Решение систем дифференциальных уравнений с помощью CUDA| Engraver Weblog [Электронный Ресурс]. — URL: https://engraver.wordpress.com/2010/10/22/ode-solving-by-cuda/ (Дата обращения 19.12.2021). Загл. с экр. Яз. рус.

10 *Dormand, J. R.* A family of embedded runge-kutta formulae / J. R. Dormand, P. J. Prince // *Journal of Computational and Applied Mathematics*. — 1980. — Pp. 19–26.