

Решеточные газы, решеточное уравнение Больцмана

Отчёт по четвертому этапу группового проекта

Команда №4: Абакумова Олеся Максимовна (НФИбд-02-22)

Астраханцева Анастасия Александровна (НФИбд-01-22)

Ганина Таисия Сергеевна (НФИбд-01-22)

Ибатулина Дарья Эдуардовна (НФИбд-01-22)

Содержание

1	Введение	3
1.1	Цель проекта	3
1.2	Цель и задачи четвертого этапа проекта	3
1.3	Актуальность	3
2	Основная часть	5
2.1	Итоги трёх этапов проекта	5
2.1.1	Этап 1. Теоретическое обоснование и формулировка задачи	5
2.1.2	Этап 2. Изучение алгоритмов и структур данных	5
2.1.3	Этап 3. Реализация модели и проведение численных экспериментов	6
2.2	Коллективная работа над проектом	7
2.2.1	Распределение ролей и организация взаимодействия	7
2.2.2	Коллективное обсуждение решений	8
2.3	Программная реализация	8
2.3.1	Выбор языка и инструментов	8
2.3.2	Архитектура кода	8
2.3.3	Объяснение выбора архитектуры	9
2.4	Анализ результатов: что получилось и почему	9
2.4.1	Корректность работы модели	9
2.4.2	Визуализация	10
2.4.3	Достижения и ограничения	10
2.5	Качество работы кода: подробный разбор	10
2.6	Коллективное обсуждение итогов и самооценка деятельности	13
2.6.1	Обсуждение трудностей и путей их решения	13
2.6.2	Индивидуальный и командный вклад	14
2.6.3	Самооценка	14
2.7	Выводы и рекомендации	14
3	Заключительная часть	16
3.1	Заключение	16
4	Список литературы	17

1 Введение

1.1 Цель проекта

Разработать и проанализировать модель на основе решеточного уравнения Больцмана для описания течений газа.

1.2 Цель и задачи четвертого этапа проекта

Главная цель: провести комплексный анализ полученных результатов моделирования, оценить качество реализации модели НРР, обсудить итоги работы в команде и представить проект на защите.

Задачи этапа:

1. Анализировать результаты тестирования модели
2. Обсудить в команде итоги работы
3. Оценить качество реализации проекта, в частности, написанного кода
4. Подготовить материалы для защиты проекта
5. Провести самооценку деятельности

1.3 Актуальность

В современных научных и инженерных задачах моделирование сложных физических процессов становится всё более востребованным инструментом. Одной

из таких задач является моделирование течения газа, где традиционные методы требуют значительных вычислительных ресурсов и часто оказываются непрактичными для задач большого масштаба или с высокой степенью детализации. В этом контексте решеточные методы (LGA и LBE) приобретают особую актуальность, так как позволяют упростить вычисления, сохраняя при этом физическую достоверность моделируемых процессов [1].

Наш проект был посвящён построению и анализу модели на основе решёточного уравнения Больцмана (модель НРР). Работа над проектом велась в составе команды №4 (Абакумова Олеся, Астраханцева Анастасия, Ганина Таисия, Ибатулина Дарья). Проект был разбит на три этапа, каждый из которых имел свои задачи и результаты. В данном отчёте подробно рассмотрим четвертый этап – коллективное обсуждение итогов и самооценку нашей деятельности.

2 Основная часть

2.1 Итоги трёх этапов проекта

2.1.1 Этап 1. Теоретическое обоснование и формулировка задачи

На этом этапе мы:

1. Изучили основные принципы решеточных моделей газа (LGA) и решеточного уравнения Больцмана (LBE).
2. Ознакомились с базовыми моделями: HPP, FHP-I, FHP-III, а также с расширенной моделью с 9 направлениями скорости.
3. Проанализировали физические свойства этих моделей: симметрию, анизотропию, возможность сохранения импульса и энергии, введение температуры.
4. Изучили область применимости этих моделей — от моделирования жидкостей до фазовых переходов и химических взаимодействий.

Этот этап дал нам прочную теоретическую основу для дальнейших шагов, позволив критически осмыслить достоинства и ограничения моделей, и выбрать для реализации наиболее подходящую — модель HPP.

2.1.2 Этап 2. Изучение алгоритмов и структур данных

На втором этапе:

1. Мы подробно проанализировали алгоритмы распространения и столкновения частиц в решеточных моделях.
2. Были рассмотрены способы кодирования состояний узлов с помощью битовых операций, что обеспечило компактность и быстродействие реализации.
3. Сравнили особенности моделей HPP, FHP-I, FHP-III и LBM, включая их алгоритмическую сложность, вычислительные затраты, физическую реалистичность.
4. Сделали обоснованный выбор в пользу модели HPP, как стартовой точки: она проще в реализации, но при этом позволяет продемонстрировать ключевые принципы решеточного моделирования.

2.1.3 Этап 3. Реализация модели и проведение численных экспериментов

На завершающем техническом этапе:

1. Мы реализовали модель HPP на языке Julia, выбрав его за высокую производительность, лаконичность и удобство для научных вычислений.

В коде предусмотрены функции для:

- создания решетки и добавления частиц;
- применения периодических граничных условий;
- реализации правил столкновений (лобовых и угловых);
- визуализации состояния системы;
- расчета макроскопических параметров (число частиц, импульс);
- автоматического запуска тестов и создания анимаций.

2. Было реализовано три тестовых сценария, проверяющих:

- корректность перемещения одной частицы;
- корректность лобового столкновения;
- поведение при угловом столкновении четырёх частиц.
- Созданы анимации (GIF), наглядно демонстрирующие ход моделирования, а также сохранение физических величин во времени [2].

2.2 Коллективная работа над проектом

2.2.1 Распределение ролей и организация взаимодействия

С самого начала работы над проектом мы договорились о чётком распределении ролей:

- Теоретический анализ (описание модели, её математических основ, анализ преимуществ и недостатков) - все участники, с последующим обсуждением на общем собрании.
- Программная реализация - основная ответственность легла на участников с наиболее сильными навыками программирования на Julia (но все члены команды участвовали в обсуждении архитектуры кода).
- Тестирование и визуализация - совместная работа: каждый предлагал тестовые сценарии, участвовал в анализе результатов и подготовке анимаций.
- Оформление отчёта и презентаций - по частям, с последующим коллективным редактированием. Регулярные встречи (очно и онлайн) позволяли быстро обмениваться идеями, выявлять ошибки и корректировать ход работы.

2.2.2 Коллективное обсуждение решений

Каждое важное решение принималось после обсуждения: - Как лучше реализовать периодические граничные условия? - Как структурировать код для удобства тестирования? - Какие тесты наиболее полно проверяют корректность модели? - Как визуализировать результаты, чтобы они были максимально наглядными? Обсуждение сопровождалось демонстрацией промежуточных результатов, разбором кода, анализом возникающих ошибок.

2.3 Программная реализация

2.3.1 Выбор языка и инструментов

Мы выбрали язык Julia за его высокую скорость, лаконичность и развитые средства для научных вычислений и визуализации. Для графиков использовалась библиотека Plots с бэкендом GR.

2.3.2 Архитектура кода

- Параметры модели (размеры решётки, направления движения частиц, цвета для визуализации) задаются в начале, что облегчает настройку и масштабирование модели.
- Создание сетки: реализована функция для создания булевой 3D-матрицы, где третье измерение отвечает за направления движения частиц.
- Добавление частиц: отдельная функция с проверкой корректности координат и направления.
- Периодические граничные условия: реализованы через фиктивные узлы по краям решётки и копирование значений с противоположных физических границ.
- Обработка столкновений: реализована по строгим правилам НРР, с учётом только физически осмысленных столкновений (лобовых и под прямым

углом).

- Распространение частиц: реализовано через создание новой сетки и перенос частиц согласно их направлениям.
- Подсчёт числа частиц и импульса: отдельные функции для контроля сохранения физических величин.
- Визуализация: функция для отрисовки состояния решётки с цветными стрелками, показывающими направления движения частиц.
- Тесты: три ключевых сценария - одиночная частица, лобовое столкновение, столкновение под прямым углом. Для каждого теста создаётся анимация (GIF) [3].

2.3.3 Объяснение выбора архитектуры

- Периодические граничные условия выбраны для проверки корректности распространения частиц: частица, вышедшая за одну границу, должна появляться с противоположной стороны, что важно для моделирования “бесконечных” сред.
- Разделение на функции повышает читаемость и удобство тестирования.
- Визуализация позволяет не только проверить корректность работы кода, но и наглядно продемонстрировать физический смысл происходящих процессов [2].

2.4 Анализ результатов: что получилось и почему

2.4.1 Корректность работы модели

- Тест 1 (одна частица): частица движется по прямой, корректно “переходит” через границы, возвращаясь с противоположной стороны. Число частиц и импульс сохраняются.

- Тест 2 (лобовое столкновение): две частицы встречаются, после столкновения меняют направления строго по правилам НРР. Сохраняются число частиц и импульс.
- Тест 3 (столкновение под прямым углом): четыре частицы встречаются в одной точке, после столкновения разлетаются в новые направления. Все законы сохранения выполняются [1].

2.4.2 Визуализация

Для каждого теста были созданы анимации, позволяющие шаг за шагом наблюдать за эволюцией системы. Это позволило:

1. Убедиться в правильности работы алгоритма.
2. Наглядно продемонстрировать физические процессы (распространение, столкновения, сохранение числа частиц и импульса).

2.4.3 Достижения и ограничения

- Достижения: реализована полностью работоспособная модель НРР, все физические законы выполняются, тесты проходят успешно, визуализация информативна.
- Ограничения: модель НРР обладает известными недостатками (анизотропия, нефизичное поведение на макроуровне), о чём мы честно говорим в отчёте. Для более точного моделирования реальных жидкостей и газов требуются более сложные модели (например, FHP).

2.5 Качество работы кода: подробный разбор

1. Структурированность, читаемость и модульность

Код организован в виде отдельных функций, каждая из которых отвечает за свою задачу:

- создание сетки (`create_grid`),
- добавление частицы (`add_particle!`),
- применение граничных условий (`apply_periodic_boundaries!`),
- обработка столкновений (`collide!`),
- распространение частиц (`propagate!`),
- подсчёт числа частиц (`count_particles`),
- вычисление импульса (`calculate_momentum`),
- визуализация (`plot_grid`),
- запуск тестов (`run_all_tests`).

Преимущества такого подхода:

- Код легко читать и поддерживать.
- Каждая функция имеет чётко определённую ответственность.
- Можно быстро находить и исправлять ошибки, а также расширять функционал.

2. Корректность реализации физических принципов

2.1. Периодические граничные условия

- Реализованы через фиктивные узлы по краям сетки и копирование значений с противоположных физических границ.
- Проверка на одной частице (Тест 1) показывает: частица, выходя за одну границу, корректно появляется с противоположной стороны.
- Это подтверждает правильную реализацию периодичности, что критически важно для моделирования “бесконечной” среды.

2.2. Столкновения частиц

- В функции `collide!` реализованы два типа столкновений: лобовое (две частицы навстречу) и под прямым углом (четыре частицы).
- Логика соответствует теории НРР: направления меняются строго по правилам, обеспечивается сохранение числа частиц и импульса.
- Тесты 2 и 3 (лобовое и угловое столкновения) показывают: после столкновения частицы разлетаются в правильных направлениях.

2.3. Сохранение числа частиц и импульса - На каждом шаге моделирования в тестах вычисляются и выводятся общее число частиц и суммарный импульс. - В анимациях и текстовых аннотациях видно, что эти величины сохраняются, что говорит о физической корректности кода.

3. Гибкость и расширяемость

- Размер сетки, направления движения, цвета для визуализации задаются в начале через константы, что позволяет легко менять параметры модели.
- Код легко масштабируется: можно увеличивать размеры сетки, добавлять новые сценарии тестирования, менять правила столкновений.

4. Визуализация и наглядность

- Для каждого теста создаются анимации (GIF), где визуально отображается движение и столкновения частиц, а также аннотируются число частиц и импульс.
- Использование разных цветов для направлений делает анимации понятными даже без глубокого погружения в код.
- Визуализация помогает не только анализировать корректность, но и презентовать результаты на защите.

5. Тестируемость и автоматизация проверки

В коде реализованы отдельные функции для трёх ключевых тестов:

- одна частица (проверка распространения и граничных условий),
- две частицы (лобовое столкновение),
- четыре частицы (столкновение под прямым углом).

Функция `run_all_tests` позволяет запускать все тесты последовательно, что удобно для автоматической проверки и демонстрации.

6. Обработка ошибок и валидация данных

- Используются `@assert` для проверки корректности координат частиц и направлений при добавлении в сетку.
- Это предотвращает появление некорректных состояний и облегчает отладку.

7. Документированность

- В отчёте приведены подробные комментарии к каждой функции, объясняющие смысл и работу каждого блока кода.
- Описаны все используемые параметры и логика работы алгоритма.

8. Ограничения и честная самооценка

- В отчёте честно указаны ограничения самой модели НРР (анизотропия, нефизичность на макроуровне), а не только кода.
- Это говорит о критическом отношении к результату и понимании границ применимости [3].

2.6 Коллективное обсуждение итогов и самооценка деятельности

2.6.1 Обсуждение трудностей и путей их решения

- Сложности с граничными условиями: на этапе реализации периодических границ возникли ошибки, которые были обнаружены при коллективном

просмотре анимаций и исправлены после обсуждения.

- Проверка корректности столкновений: не сразу удалось реализовать все случаи столкновений, потребовалось несколько итераций и обсуждений.
- Временные ограничения: благодаря командной работе и чёткому плану удалось уложиться в сроки.

2.6.2 Индивидуальный и командный вклад

- Каждый участник внёс вклад в теорию, программирование, тестирование и оформление отчёта.
- Совместная работа позволила каждому освоить новые навыки: кто-то углубился в программирование на Julia, кто-то - в физику решеточных моделей, кто-то - в визуализацию данных.
- В процессе обсуждений и совместного поиска решений повысился уровень взаимопонимания и доверия в команде.

2.6.3 Самооценка

- Цели этапа полностью достигнуты: реализована модель, проведено тестирование, оформлен подробный отчёт.
- Качество реализации: код структурирован, документирован, легко расширяется для новых задач.
- Командная работа: эффективное распределение задач, взаимопомощь, своевременное решение проблем.
- Личные достижения: каждый участник повысил свою квалификацию в области научного моделирования и командной работы.

2.7 Выводы и рекомендации

Работа над проектом позволила нам:

- Освоить современные методы моделирования физических процессов.
- Получить опыт командной работы, научиться планировать и реализовывать сложные проекты.
- Научиться анализировать и визуализировать результаты моделирования.
- Критически оценивать свои достижения и ограничения [4].

Наш проект продемонстрировал, что даже относительно простые решеточные модели позволяют точно и эффективно описывать поведение газов на микроскопическом уровне. Мы прошли путь от теории к практике, реализовав рабочую систему и подтвердив её корректность с помощью численных экспериментов.

Возможное продолжение работы:

- Модернизация модели: переход от HPP к FHP-III, чтобы устранить анизотропию и повысить физическую достоверность.
- Добавление термодинамики: переход к модели с 9 направлениями и введение температуры.
- Моделирование фазовых переходов: добавление сил взаимодействия между частицами.
- Переход к LBE: для описания вязких течений и сложных граничных условий.
- Параллелизация: реализация на GPU с помощью CUDA.jl для повышения производительности [4].

3 Заключительная часть

3.1 Заключение

Проект был реализован на высоком уровне благодаря слаженной работе команды, глубокому анализу теории и тщательному тестированию программной реализации. Каждый участник внёс значимый вклад, а коллективное обсуждение позволило не только добиться отличного результата, но и получить ценный опыт научного сотрудничества.

4 Список литературы

1. Медведев Д.А.и.др. Моделирование физических процессов и явлений на ПК: Учеб. пособие. Новосибирск: Новосибирский государственный университет, 2010. С. 101.
2. Чашин Г.С. Метод решёточных уравнений Больцмана: моделирование изотермических низкоскоростных течений: 99. Институт прикладной математики им. М.В. Келдыша РАН, 2021.
3. Hardy J., Pomeau Y., Pazzis O. de. Time evolution of a two-dimensional classical lattice system // Physical Review Letters. 1973. Т. 31, № 5. С. 276–279.
4. Succi S. The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. Oxford University Press, 2001.