

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ имени М.В. ЛОМОНОСОВА  
ФИЗИЧЕСКИЙ ФАКУЛЬТЕТ

На правах рукописи  
УДК 004.942

Бикулов Дмитрий Александрович

**МОДЕЛИРОВАНИЕ ЯВЛЕНИЙ ПЕРЕНОСА В ПОРИСТЫХ СРЕДАХ  
НА ГИБРИДНЫХ СУПЕРКОМПЬЮТЕРНЫХ СИСТЕМАХ**

05.13.18 — «Математическое моделирование,  
численные методы и комплексы программ»

ДИССЕРТАЦИЯ  
на соискание учёной степени  
кандидата физико-математических наук

Научный руководитель:  
доцент, кандидат технических наук  
Евгений Александрович Грачёв

Москва — 2015

# ОГЛАВЛЕНИЕ

<b>ВВЕДЕНИЕ . . . . .</b>	<b>4</b>
<b>ГЛАВА 1. Аналитический обзор литературы . . . . .</b>	<b>11</b>
1.1 Метод решеточных уравнений Больцмана (РМБ) . . . . .	11
1.1.1 Обзор распространенных шаблонов скоростей . . . . .	13
1.1.2 Обзор аппроксимаций интеграла столкновений . . . . .	16
1.1.3 Учет внешней силы . . . . .	18
1.1.4 Граничные условия . . . . .	19
1.2 РМБ для моделирования многокомпонентного течения . . . . .	22
1.3 РМБ для моделирования анизотропной диффузии . . . . .	25
1.4 РМБ для моделирования теплопереноса . . . . .	25
1.5 Оптимизация потребления памяти в РМБ . . . . .	26
1.6 Заключение к Главе 1 . . . . .	29
<b>ГЛАВА 2. Анализ суперкомпьютерных технологий . . . . .</b>	<b>30</b>
2.1 Классификация Флинна архитектур вычислительных систем . . . . .	31
2.2 Классификация вычислительных систем по доступу к памяти . . . . .	32
2.3 Программные инструменты для параллельных и распределенных вычислений . . . . .	35
2.3.1 Message Passing Interface (MPI) . . . . .	35
2.3.2 Compute Unified Device Architecture (CUDA) . . . . .	36
2.4 Совместное использование CUDA и MPI . . . . .	41
2.5 Заключение к Главе 2 . . . . .	44
<b>ГЛАВА 3. Описание математических моделей и их реализаций . . . . .</b>	<b>45</b>
3.1 Метод решеточных уравнений Больцмана для моделирования процессов переноса .	46
3.1.1 Моделирование однокомпонентного течения . . . . .	46
3.1.2 Моделирование двухкомпонентного течения . . . . .	49
3.1.3 Моделирование диффузии . . . . .	52
3.1.4 Моделирование теплопроводности . . . . .	54
3.1.5 Граничные условия . . . . .	54
3.2 Метод дискретных элементов . . . . .	57
3.3 Оценка производительности программного комплекса . . . . .	58
3.4 Верификация программного модуля . . . . .	62

3.4.1 Течение Пуазейля . . . . .	63
3.4.2 Абсолютная проницаемость цилиндрической трубы . . . . .	64
3.4.3 Лобовое сопротивление шара . . . . .	66
3.4.4 Тест Лапласа . . . . .	67
3.4.5 Коалесценция двух капель . . . . .	69
3.4.6 Тест смачивания . . . . .	69
3.4.7 Диффузия от множества источников . . . . .	73
3.5 Заключение к Главе 3 . . . . .	74
<b>ГЛАВА 4. Моделирование естественных систем . . . . .</b>	<b>75</b>
4.1 Определение абсолютной проницаемости пористой среды . . . . .	75
4.2 Построение петрофизических зависимостей . . . . .	80
4.3 Моделирование изменения проницаемости проппантных упаковок под нагрузкой . .	81
4.4 Определение компонентных проницаемостей . . . . .	84
4.5 Заключение к Главе 4 . . . . .	90
<b>ЗАКЛЮЧЕНИЕ . . . . .</b>	<b>91</b>
<b>СПИСОК ЛИТЕРАТУРЫ . . . . .</b>	<b>93</b>

# **ВВЕДЕНИЕ**

## **Актуальность проблемы**

Высокопроизводительное моделирование процессов переноса со сложными границами требуется во многих научных, медицинских и инженерных задачах. В медицине моделирование диффузии зачастую выступает в качестве части более крупной модели целого физиологического процесса или патологического состояния. В нефтяной промышленности моделирование позволяет лучше понимать фильтрационные характеристики месторождений на микроуровне, сопоставлять их с лабораторными исследованиями, и, в конечном счете, производить оценку свойств месторождения.

Метод решеточных уравнений Больцмана (РМБ) выступает в роли алгоритмической платформы для моделирования различных процессов переноса. Его отличительными особенностями являются универсальность, внутренняя параллельность и адаптируемость. Под универсальностью понимается возможность применения РМБ для моделирования однокомпонентного течения, многокомпонентного течения, анизотропной диффузии и теплопроводности без существенных изменений в алгоритме. При этом метод высокопараллелен по данным (т.е. количество исполняемых потоков достигает сотен тысяч) и может быть реализован для расчетов на графических ускорителях (GPU), в том числе и гибридных архитектурах с множеством GPU. Адаптируемость позволяет добавлять в модель дополнительные эффекты: силу тяжести, фазовые переходы и т.д.

Использование GPU в научных расчетах позволяет в ряде случаев достигать увеличения производительности на порядок и более. При этом массивно-параллельные вычисления могут осуществляться на обычной рабочей станции: в современных графических ускорителях количество вычислительных ядер достигает 2880, объем доступной на устройстве оперативной памяти — 24 Гб, энергопотребление не превышает 300 Ватт.

Суперкомпьютерные системы с гибридной архитектурой стablyно присутствуют в списке top500 [99]. Три из десяти систем в списке top500 за ноябрь 2014 года используют графические ускорители. В списке самых энергоэффективных суперкомпьютеров green500 [98] за ноябрь 2014 года девять из десяти используют графические ускорители, причем восемь из них используют GPU Nvidia K20, Nvidia K20x или Nvidia K40m, поддерживающие технологию Compute

Unified Device Architecture (CUDA). CUDA представляет собой программно-аппаратную платформу, позволяющую производить вычисления общего назначения на поддерживающих ее графических ускорителях.

Таким образом, моделирование процессов переноса в сложнопостроенных средах и использование для этого соответствующего высокопроизводительного программного комплекса, оптимизированного для расчетов на гибридных суперкомпьютерных архитектурах, является, безусловно, актуальной задачей. Рассматриваемые в данной работе алгоритмы и методики оптимизации использования памяти позволяют производить массивное моделирование с почти линейным ростом производительности в зависимости от числа использованных графических ускорителей вплоть до 45 GPU.

## **Цель работы**

Целью данной диссертационной работы является создание программного комплекса — инструмента моделирования процессов переноса, оптимизированного для работы на гибридных суперкомпьютерах, его верификация и реализация примеров использования для задач, обладающих внутренней параллельностью.

## **Задачи работы**

Основными задачами данной диссертационной работы являются:

1. Анализ существующих алгоритмов моделирования процессов переноса в пористых средах на основе метода решеточных уравнений Больцмана.
2. Анализ методов оптимизации алгоритма для выполнения на графических ускорителях.
3. Создание методологии моделирования процессов переноса, разработка на ее основе соответствующего высокопроизводительного комплекса программ.
4. Верификация комплекса программ на основе стандартных тестов и известных точных аналитических решений.
5. Применение реализованного программного комплекса для определения физических характеристик: абсолютной проницаемости сплошных флюидонасыщенных сред, фазовых (компонентных) проницаемостей для эмульсий и других флюидов в сложнопостроенной пористой среде.

## **Основные результаты, выносимые на защиту**

Основными результатами данной диссертационной работы являются:

1. Методология моделирования процессов переноса с использованием множества графических ускорителей на гибридных суперкомпьютерных системах. Созданный и верифицированный на тестовых примерах соответствующий программный комплекс, позволяющий производить моделирование процессов переноса.
2. Алгоритмизация и реализация численных методов для моделирования процессов переноса на основе единого инструментария — метода решеточных уравнений Больцмана.
3. Механическая модель проппантной упаковки под нагрузкой (проппант — гранулярный материал, часто песок, используемый в нефтедобывающей промышленности в качестве расклинивающего агента для удержания трещин при гидроразрыве пласта).
4. Методика совместного использования моделей порового пространства (как на полученных на основе рентгеновской компьютерной микротомографии, так и созданных на основе алгоритмических моделей) и гидродинамической модели.

## **Научная новизна работы**

1. Впервые создан комплекс программ, позволяющий в полуавтоматическом режиме производить численное моделирование процессов переноса в сложных геометриях и оптимизированный для работы на гибридных суперкомпьютерах (суперкомпьютер «Ломоносов»).
2. Разработана методология высокопроизводительного параллельного моделирования в сложных естественных системах (нерегулярные пористые среды) процессов, отличающихся локальностью по данным, на примере процессов переноса.
3. Создана механическая модель проппантной упаковки под нагрузкой, разработана и использована на практике методика взаимодействия механической и гидродинамической моделей.
4. Получена и проверена на лабораторных данных эмпирическая оценка минимально допустимого разрешения рентгеновской компьютерной микротомографии геологических (каменных) образцов, необходимого для корректного моделирования фильтрации в пористой среде на основе томографии реальных образцов.

## **Практическая значимость работы**

1. Реализованный комплекс программ позволяет в полуавтоматическом режиме производить численное моделирование процессов переноса в сложных геометриях, в том числе в геологических средах.
2. Предложенная методология моделирования, использующая вычисления на графических ускорителях, существенно сокращает время проведения каждого вычислительного эксперимента.
3. Дополнительные оптимизации потребления памяти, использованные в программном комплексе, позволяют рассматривать модели с большими линейными размерами.
4. Использование метода решёточных уравнений Больцмана в качестве платформы позволяет учитывать дополнительные эффекты в численных экспериментах (например, фазовые переходы или силу тяжести).
5. Созданный программный интерфейс позволяет производить потоковое выполнение расчётов, которое полезно в случае исследования статистических зависимостей фильтрационных характеристик среды от её параметров.

## **Личный вклад автора**

Все исследования, результаты которых изложены в диссертационной работе, проведены лично автором в процессе научной деятельности. Материал из совместных публикаций, использованный в работе, по большей части состоит из оригинальных результатов автора.

## **Соответствие диссертации паспорту научной специальности**

Содержание и результаты работы соответствуют паспорту специальности 05.13.18, а именно соответствуют областям исследований:

1. Разработка, обоснование и тестирование эффективных вычислительных методов с применением современных компьютерных технологий.
2. Реализация эффективных численных методов и алгоритмов в виде комплексов проблемно-ориентированных программ для проведения вычислительного эксперимента.
3. Комплексные исследования научных и технических проблем с применением современной технологии математического моделирования и вычислительного эксперимента.

## **Апробация работы**

Результаты, представленные в данной диссертации, докладывались на российских и международных конференциях:

1. III Балтийская школа-семинар «Петрофизическое моделирование осадочных пород» (г. Санкт-Петербург, 2014)
2. XX Всероссийская конференция «Теоретические основы и конструирование численных алгоритмов решения задач математической физики», посвященная памяти К.И. Бабенко (Дюрсо Краснодарского края, 2014)
3. Всероссийская научная конференция «Научный сервис в сети Интернет: все грани параллелизма» (Дюрсо Краснодарского края, 2013)
4. Международная конференция студентов, аспирантов и молодых ученых «Ломоносов 2013» (Москва, 2013)
5. Международная конференция студентов, аспирантов и молодых ученых «Ломоносов 2012» (Москва, 2012)
6. GPU Technology Conference 2012, (San Jose, California, USA, 2012)

Помимо этого, результаты докладывались на научных семинарах:

1. Семинар «Суперкомпьютерные технологии в науке, образовании и промышленности» на базе НОЦ «Суперкомпьютерные технологии» под руководством академика В.А. Садовничего.
2. Семинар отделения теоретической физики ИЯИ РАН по программированию.
3. Семинар кафедры компьютерных методов физики Физического факультета МГУ им. М.В. Ломоносова.
4. Семинар «Математические методы в естественных науках» на базе Физического факультета МГУ им. М.В. Ломоносова под руководством профессора А.Н. Боголюбова.
5. Семинар «Вычислительные методы и математическое моделирование» 11 отдела ИМП им. М.В. Келдыша РАН под руководством член-корреспондента РАН Ю.П. Попова и профессора М.П. Галанина.
6. Научно-методологический семинар НИВЦ МГУ под руководством профессора А.В. Тихонравова.

## **Публикации**

Материалы диссертации опубликованы в 11 печатных работах, из них 3 статьи в рецензируемых журналах из списка ВАК (из них одна в печати), 1 статья в рецензируемом зарубежном журнале, 6 статей в сборниках.

Публикации в журналах из списка ВАК:

1. Бикулов Д. Эффективная реализация метода решеточных уравнений Больцмана для гибридных суперкомпьютерных систем // Вычислительные методы и программирование: Новые вычислительные технологии. — 2015. — Т. 16. — С. 205–214.
2. Бикулов Д., Сенин Д. Реализация метода решеточных уравнений Больцмана без хранимых функций распределения для gpru // Вычислительные методы и программирование: Новые вычислительные технологии. — 2013. — Т. 14. — С. 370–374.
3. Бикулов Д. и др. Реализация метода решеточных уравнений Больцмана для расчетов на gpru-кластере // Вычислительные методы и программирование: Новые вычислительные технологии. — 2012. — Т. 13. — С. 13–19.

Публикация в рецензируемом зарубежном журнале:

4. Bikulov D., Saratov A., Grachev E. Prediction of the permeability of proppant packs under load // International Journal of Modern Physics C. — 2015. — Р. 1–18.

Публикации в сборниках:

1. Бикулов Д. Эффективная реализация метода решеточных уравнений Больцмана для гибридных суперкомпьютерных систем // CUDA Альманах. — 2015. — С. 15–15.
2. Бикулов Д. и др. Определение проницаемости проппантных упаковок с помощью метода решеточных уравнений Больцмана // Тезисы докладов XX Всероссийской конференции «Теоретические основы и конструирование численных алгоритмов решения задач математической физики», посвященная памяти К.И.Бабенко (Дюрсо, 15-21 сентября 2014). — М: Институт прикладной математики им. М.В.Келдыша Москва, 2014.
3. Макаров С., Бикулов Д. и др. Компьютерное моделирование динамики системы клеток головного мозга в условиях ишемического инсульта // Суперкомпьютерные технологии в науке, образовании и промышленности. — Т. 5. — 2013. — С. 61–68.
4. Бикулов Д. Математическая модель пульсового течения крови // CUDA Альманах. — 2013. — С. 11–11.
5. Макаров С., Бикулов Д. и др. Компьютерное моделирование развития ишемического инсульта с помощью технологии CUDA // Труды международной суперкомпьютерной конференции «Научный сервис в сети Интернет: все грани параллелизма» (23 - 28 сентября 2013 г., г. Новороссийск). — Изд-во МГУ Москва, 2013.
6. Бикулов Д. Реализация модели пульсового течения крови в сосуде с эластичными стенками с использованием технологии NVidia CUDA // XX Международная конференция студентов, аспирантов и молодых ученых Ломоносов-2013. — Секция вычислительной математики и кибернетики. — Издательский отдел факультета вычислительной математики и кибернетики МГУ им. М.В. Ломоносова. Москва, 2013. — С. 66–67.

7. Бикулов Д. Реализация метода решеточных уравнений Больцмана для расчетов на гри кластере // XIX Международная конференция студентов, аспирантов и молодых ученых Ломоносов-2012. — Секция вычислительной математики и кибернетики. — Издательский отдел факультета вычислительной математики и кибернетики МГУ имени М.В. Ломоносова Москва, 2012. — С. 118–119. Москва.

## **Структура и объем диссертационной работы**

Диссертация состоит из титульного листа, оглавления, введения, четырех глав, заключения, списка литературы, содержащего 100 наименований. Общий объем диссертации составляет **100** страниц, в том числе 40 рисунков и 11 таблиц.

**ВВЕДЕНИЕ.** Сформулированы цель, актуальность, задачи, основные результаты, научная новизна, практическая значимость диссертационного исследования, приведен список опубликованных по теме работ. Приведено краткое содержание диссертации.

**ГЛАВА 1.** Аналитический обзор основных аспектов современных модификаций решеточного уравнения Больцмана для моделирования многокомпонентного течения, диффузии и теплопроводности соответственно. Методики оптимизации использования памяти алгоритмом.

**ГЛАВА 2.** Анализ основных суперкомпьютерных технологий, необходимых для использования высокопроизводительных вычислительных машин с гибридной архитектурой (с множеством графических ускорителей): Message Passing Interface (MPI) и Compute Unified Device Architecture (CUDA), а также методика их совместного использования, алгоритм присвоения уникальных графических вычислителей (GPU) разным процессам в случае мульти-GPU вычислительных узлов. Классификации вычислительных систем Флинна и по организации доступа к памяти.

**ГЛАВА 3.** Модификации метода решеточных уравнений Больцмана (РМБ) и оптимизации использования памяти, реализованные в программном комплексе. Описание модели проппантной упаковки под нагрузкой. Оценка производительности программного комплекса. Описания и результаты тестовых вычислительных экспериментов, предназначенных для верификации программного комплекса. Сравнение производительности CPU и GPU реализаций.

**ГЛАВА 4.** Применение реализованного программного комплекса для решения реальных задач. Описания и результаты вычислительных экспериментов.

**ЗАКЛЮЧЕНИЕ.** Кратко перечислены основные результаты диссертационной работы.

# ГЛАВА 1. Аналитический обзор литературы

В главе сделан аналитический обзор основных аспектов современного применения метода решеточных уравнений Больцмана для моделирования одно- и многокомпонентных течений, теплопереноса и диффузии.

В § 1 рассматриваются основные уравнения и понятия метода решеточных уравнений Больцмана: шаблоны скоростей (D2Q9, D3Q7, D3Q15, D3Q19), аппроксимации интеграла столкновений  $\Omega$  (однорелаксационная (SRT), с двумя временами релаксации (TRT) и многорелаксационная (MRT)), основные варианты учета внешней силы, три основных варианта граничных условий на краях непроницаемых областей (нарешеточный отскок, межрешеточный отскок и два интерполяционных метода) и их особенности. В § 2-4 приводятся детальные описания модификаций решеточного уравнения Больцмана для моделирования многокомпонентного течения (метод псевдопотенциала, метод цветового градиента и метод свободной энергии), диффузии и теплопроводности соответственно. В § 5 рассматриваются аспекты оптимизации использования памяти алгоритмом: алгоритм с дополнительным массивом для одного направления, алгоритм со сжатой сеткой, обменный алгоритм, AA-шаблон, трюк с неявным обменом и реализация без хранимых значений функции распределения.

Содержание данной главы представляет собой аналитический обзор литературы по теме моделирования с помощью метода решеточных уравнений Больцмана и оптимальной с точки зрения объёма требуемой памяти алгоритмизации данного метода.

## 1.1. Метод решеточных уравнений Больцмана (РМБ)

Метод решеточных уравнений Больцмана (РБМ) представляет собой один из универсальных методов вычислительной физики. Он используется для моделирования множества процессов в различных областях: динамики разреженных газов [87], однокомпонентного течения [16, 88, 91, 92] (в том числе на неструктурированных сетках [94]), тетраэдрических сетках [89], развития стеноза артерий [10], течения неニュтоновских жидкостей [9], течения в микроканалах [46], роста кристаллов в перенасыщенном растворе [37], течения жидкости в поровом пространстве [6, 7, 36] поведения двухфазных систем [44, 95, 96], моделирования диффузии [78, 93] и теплопереноса [34]. Отличительными особенностями данного метода являются высокая параллельность

по данным, простота реализации и широкие возможности моделирования: для моделирования анизотропной диффузии [78] и теплопроводности [34] и течения в сложной геометрии [7] используется тот же алгоритм и основные уравнения.

Вычислительный объем в методе разбивается на пространственные ячейки, чаще всего кубические, принадлежащие одному из двух классов: проницаемая или непроницаемая. В данной работе рассматриваются кубические пространственные сетки. Среда моделируется в виде набора квазичастиц, перемещающихся между ячейками вдоль заранее заданных дискретных направлений. Перемещение частиц возможно только между проницаемыми ячейками, а в граничащих с непроницаемыми ячейками применяется локальное краевое условие. Совокупность заданных направлений квазичастиц в РМБ называется шаблоном скоростей и обозначается  $DdQq$ , где  $d$  — размерность пространства (2 для плоского случая и 3 для трехмерного), а  $q$  — число выбранных направлений скоростей в шаблоне. Наиболее часто используются [16, 65, 82, 84] шаблоны D2Q9 в плоском случае и D3Q7, D3Q15, D3Q19 в трехмерном случае.

Для описания ансамбля квазичастиц в РМБ используются значения функции распределения  $f_i$ , где  $i$  — номер соответствующего направления скорости в шаблоне. Эволюция  $f_i$  задается решеточным уравнением Больцмана (РУБ):

$$|f(\mathbf{r} + \mathbf{e}\delta t, t + \delta t)\rangle = |f(\mathbf{r}, t)\rangle + \Omega(|f(\mathbf{r}, t)\rangle) \quad (1.1)$$

где  $\Omega(|f(\mathbf{r}, t)\rangle)$  — интеграл столкновений в общем виде,  $|f(\mathbf{r} + \mathbf{e}\delta t, t + \delta t)\rangle = (f_0(\mathbf{r} + \mathbf{e}_0\delta t, t + \delta t), \dots, f_{q-1}(\mathbf{r} + \mathbf{e}_{q-1}\delta t, t + \delta t))^T$  — вектор значений функции распределения после шага распространения,  $|f(\mathbf{r}, t)\rangle = (f_0(\mathbf{r}, t), \dots, f_{q-1}(\mathbf{r}, t))^T$  — вектор значений функции распределения до шага столкновения, векторы  $\mathbf{e}_i$  задаются в выбранном шаблоне скоростей.

Одна итерация, таким образом, разделяется на два шага: столкновение (collsision) (1.2a) и распространение (propagation) (1.2b).

$$|f^*(\mathbf{r}, t)\rangle = |f(\mathbf{r}, t)\rangle + \Omega(f(\mathbf{r}, t)) \quad (1.2a)$$

$$|f(\mathbf{r} + \mathbf{e}\delta t, t + \delta t)\rangle = |f^*(\mathbf{r}, t)\rangle \quad (1.2b)$$

Шаг столкновения полностью локален и выполняется независимо в каждой ячейке пространства. На шаге распространения для расчетов требуются значения функции распределения в соседних ячейках, количество которых зависит от выбора решетки направлений. Локальные плот-

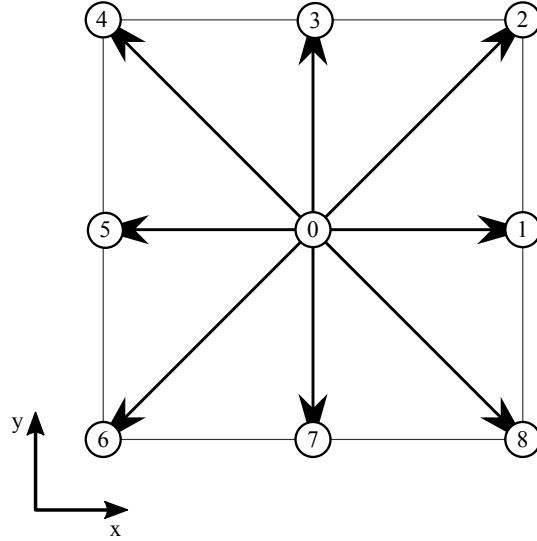


Рисунок 1.1: Решетка направлений D2Q9.

нность  $\rho$  и скорость  $\mathbf{u}$  среды рассчитываются на основе значений функции распределения в данной ячейке (1.3).

$$\rho = \sum_{i=0}^{q-1} f_i \quad (1.3a)$$

$$\rho\mathbf{u} = \sum_{i=0}^{q-1} \mathbf{e}_i f_i \quad (1.3b)$$

### 1.1.1. Обзор распространенных шаблонов скоростей

Для плоского случая наиболее часто используется шаблон D2Q9: восемь ненулевых направлений и одно нулевое (Рис. 1.1). Векторы скоростей для D2Q9 задаются следующим образом:

$$\mathbf{e} = \begin{bmatrix} 0 & 1 & 1 & 0 & -1 & -1 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & -1 & -1 & -1 \end{bmatrix}^T \quad (1.4)$$

В (1.4) каждой строке соответствует вектор  $\mathbf{e}_i$ , так что для нулевого направления  $\mathbf{e}_0 = (0, 0)$ , для первого направления  $\mathbf{e}_1 = (1, 0)$  и т.д.

Порядок перечисления направлений в шаблоне не важен, но влияет на лаконичность программного кода при реализации. Здесь и далее направления скоростей упорядочены так, чтобы выполнялось условие:  $\mathbf{e}_i = -\mathbf{e}_{i+(q-1)/2}$ , где  $i \in [1, (q-1)/2]$ ,  $q$  — количество направлений в шаблоне DdQq.

Весовые коэффициенты  $w$  для шаблона D2Q9:

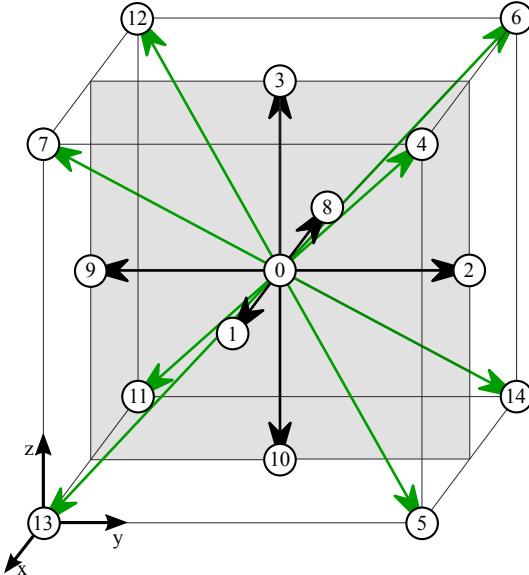


Рисунок 1.2: Решетка направлений D3Q15.

$$\mathbf{w} = \left[ \frac{4}{9} \quad \frac{1}{9} \quad \frac{1}{36} \quad \frac{1}{9} \quad \frac{1}{36} \quad \frac{1}{9} \quad \frac{1}{36} \quad \frac{1}{9} \quad \frac{1}{36} \right]^T \quad (1.5)$$

Шаблоны D2Q4 и D2Q5 не могут использоваться для моделирования течения жидкостей из-за того, что при их использовании невозможно восстановить уравнение Навье-Стокса из РУБ [54].

В трехмерном случае наиболее часто встречаются шаблоны D3Q15 и D3Q19. В D3Q15 используется 14 ненулевых направлений и одно нулевое (Рис. 1.2). Векторы скоростей для D3Q15 задаются следующим образом:

$$\mathbf{e} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & -1 & 1 & -1 & 0 & 0 & -1 & -1 & 1 & -1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & -1 & 0 & -1 & 0 & -1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & 1 & -1 & 1 & 1 & 0 & 0 & -1 & -1 & 1 & -1 & -1 \end{bmatrix}^T \quad (1.6)$$

Весовые коэффициенты для шаблона D3Q15:

$$\mathbf{w} = \left[ \frac{16}{72} \quad \frac{8}{72} \quad \frac{8}{72} \quad \frac{8}{72} \quad \frac{1}{72} \quad \frac{1}{72} \quad \frac{1}{72} \quad \frac{1}{72} \quad \frac{8}{72} \quad \frac{8}{72} \quad \frac{8}{72} \quad \frac{1}{72} \quad \frac{1}{72} \quad \frac{1}{72} \quad \frac{1}{72} \right]^T \quad (1.7)$$

В шаблоне D3Q19 используется 18 ненулевых направлений и одно нулевое (Рис. 1.3). Этот шаблон наиболее часто встречается в работах по моделированию гидродинамики в трехмерной среде. Векторы скоростей для D3Q19 задаются следующим образом:

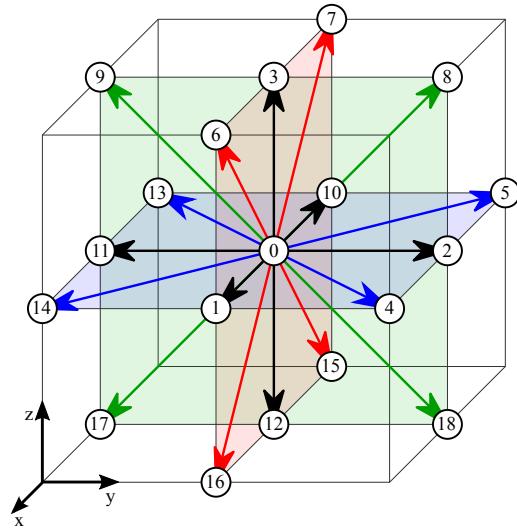


Рисунок 1.3: Решетка направлений D3Q19.

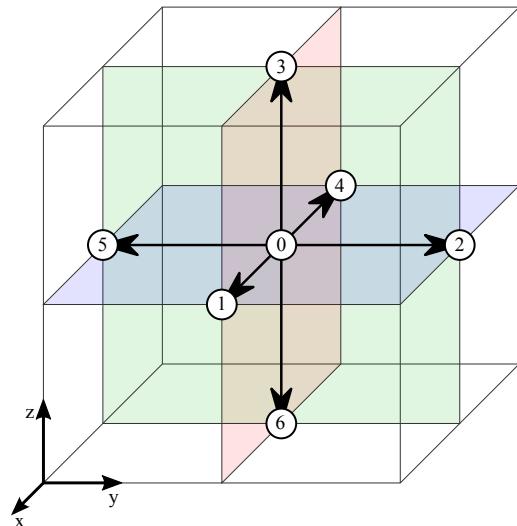


Рисунок 1.4: Решетка направлений D3Q7.

$$\mathbf{e} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & -1 & 0 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & -1 & 1 \end{bmatrix}^T \quad (1.8)$$

Весовые коэффициенты для шаблона D3Q19:

$$\mathbf{w} = \left[ \frac{12}{36} \quad \frac{2}{36} \quad \frac{2}{36} \quad \frac{2}{36} \quad \frac{1}{36} \quad \frac{1}{36} \quad \frac{1}{36} \quad \frac{1}{36} \quad \frac{1}{36} \quad \frac{2}{36} \quad \frac{2}{36} \quad \frac{2}{36} \quad \frac{1}{36} \right]^T \quad (1.9)$$

Для моделирования диффузии в трехмерном случае оказывается достаточно [78] шаблона D3Q7 (Рис. 1.4). Векторы скоростей для него задаются следующим образом:

$$\mathbf{e} = \begin{bmatrix} 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}^T \quad (1.10)$$

Весовые коэффициенты  $w$  для шаблона D3Q7:

$$\mathbf{w} = \left[ \frac{1}{4} \quad \frac{1}{8} \quad \frac{1}{8} \quad \frac{1}{8} \quad \frac{1}{8} \quad \frac{1}{8} \quad \frac{1}{8} \right]^T \quad (1.11)$$

### 1.1.2. Обзор аппроксимаций интеграла столкновений

Бхатнагар, Гросс и Крук [5]) предложили в своей работе аппроксимацию интеграла столкновений в виде линейного выражения с одним релаксационным параметром (Single Relaxation Time, SRT)  $\tau$ . Эта аппроксимация широко используется из-за малой вычислительной сложности и простоты реализации:

$$\Omega(f_i) = \frac{f_i^{eq} - f_i}{\tau_f} \quad (1.12)$$

где  $f_i^{eq}$  — равновесные значения функции распределения. Они аппроксимируются [14, 64, 80] уравнением (1.13) и характеризуют равновесное состояние, к которому со временем стремятся частицы внутри каждой ячейки.

$$f_i^{eq}(\rho, \mathbf{u}) = w_i \rho \left[ 1 + \frac{(\mathbf{e}_i \mathbf{u})}{c_s^2} + \frac{(\mathbf{e}_i \mathbf{u})^2}{2c_s^4} - \frac{(\mathbf{u}, \mathbf{u})}{2c_s^2} \right] \quad (1.13)$$

где  $c_s$  — решеточная скорость звука, а  $w_i$  — весовые коэффициенты. И скорость звука, и весовые коэффициенты зависят от выбранного шаблона скоростей.

Параметр релаксации  $\tau$  связан [64, 90, 95] с кинематической вязкостью:

$$\nu = c_s^2 (\tau - 0.5) \quad (1.14)$$

Другая широко используемая аппроксимация интеграла столкновений — т.н. многорелаксационный интеграл столкновений (Multiple Relaxation Times, MRT) [16]. В отличие от SRT, приближение MRT позволяет моделировать среды со значениями параметра релаксации  $\tau$ , близкими к 0.5, а полученные с помощью MRT перколяционные характеристики (в частности, абсолютная проницаемость) не зависят от вязкости среды [45, 59, 60]. Многорелаксационное приближение интеграла столкновений сложнее в реализации.

Основная идея MRT состоит в вычислении интеграла столкновений в пространстве выбранного набора моментов, при этом каждому моменту соответствует свой параметр релаксации. Целочисленный матричный оператор  $M$ ,  $\det(M) \neq 0$  переводит вектор значений функции распределения  $|f\rangle$  в соответствующий вектор моментов  $|m\rangle$ :

$$|m\rangle = M |f\rangle \quad (1.15a)$$

$$|f\rangle = M^{-1} |m\rangle \quad (1.15b)$$

Матрица  $M$  строится с помощью алгоритма Грама-Шмидта[16], ее строки ортогональны. Уравнение эволюции значений функции распределения  $f_i$  в случае MRT записывается следующим образом:

$$|f(\mathbf{r} + \mathbf{e}\delta t, t + \delta t)\rangle = |f(\mathbf{r}, t)\rangle - \mathbf{M}^{-1} \mathbf{S}(|m\rangle - |m^{eq}\rangle) \quad (1.16)$$

где  $\mathbf{S} = \text{diag}(s_0, \dots, s_{q-1})$  — диагональная матрица столкновений,  $|m\rangle = (m_0, \dots, m_{q-1})^T$  — вектор моментов,  $\mathbf{M}$  — ортогональная матрица перехода  $|m\rangle = \mathbf{M}|f\rangle$ ,  $|f(\mathbf{r} + \mathbf{e}\delta t, t + \delta t)\rangle = (f_0(\mathbf{r} + \mathbf{e}_0\delta t, t + \delta t), \dots, f_{q-1}(\mathbf{r} + \mathbf{e}_{q-1}\delta t, t + \delta t))^T$  — вектор значений функции распределения после шагов столкновения и распространения,  $|f(\mathbf{r}, t)\rangle = (f_0(\mathbf{r}, t), \dots, f_{q-1}(\mathbf{r}, t))^T$  — вектор значений функции распределения до шага столкновения,  $q$  — количество выделенных направлений скоростей в шаблоне.

Параметр релаксации для каждого момента выбирается независимо. Совокупность всех параметров релаксации формирует диагональную матрицу  $\hat{S}$  [16]:

$$\hat{S} = \text{diag}(0, 1.19, 1.4, 0, 1.2, 0, 1.2, 0, 1.2, \omega, 1.4, \omega, 1.4, \omega, \omega, 1.98, 1.98, 1.98) \quad (1.17)$$

где  $\omega = \frac{1}{\tau}$ ,  $\tau$  — параметр релаксации ( $\tau \in (0.5, \infty)$ ), аналогичный одноименному в SRT, также связанный с кинематической вязкостью  $\nu$ :

$$\nu = \frac{1}{3} (\omega^{-1} - 0.5) \quad (1.18)$$

Вектор равновесных моментов  $m^{eq}$  задается как:

$$m^{eq} = (\rho^{eq}, \epsilon^{eq}, \varepsilon^{eq}, j_x^{eq}, q_x^{eq}, j_y^{eq}, q_y^{eq}, j_z^{eq}, q_z^{eq}, 3p_{xx}^{eq}, 3\pi_{xx}^{eq}, p_{\omega\omega}^{eq}, \pi_{\omega\omega}^{eq}, p_{xy}^{eq}, p_{yz}^{eq}, p_{xz}^{eq}, m_x^{eq}, m_y^{eq}, m_z^{eq})^T$$

где  $\rho^{eq} = \sum_{i=0}^q f_i$ ,  $\epsilon^{eq} = -11\rho + 19\mathbf{j}^2$ ,  $\varepsilon^{eq} = \omega_\varepsilon \rho + \omega_{\varepsilon j} \mathbf{j}^2$ ,  $\mathbf{j}^{eq} = \mathbf{e}|f\rangle$ ,  $\mathbf{q}^{eq} = \frac{2}{3}\mathbf{j}$ ,  $p_{xx}^{eq} = \frac{1}{3}(2j_x^2 - j_y^2 - j_z^2)$ ,  $p_{\omega\omega}^{eq} = j_y^2 - j_z^2$ ,  $p_{xy}^{eq} = j_x j_y$ ,  $p_{yz}^{eq} = j_y j_z$ ,  $p_{xz}^{eq} = j_x j_z$ ,  $\pi_{xx}^{eq} = \omega_{xx} p_{xx}$ ,  $\pi_{\omega\omega}^{eq} = \omega_{xx} p_{\omega\omega}$ ,  $\mathbf{m}^{eq} = 0$ .

Апроксимация SRT может быть восстановлена из MRT при задании следующих значений параметров:  $\omega_\varepsilon = 3$ ,  $\omega_{\varepsilon j} = -\frac{11}{2}$  и  $\omega_{xx} = -0.5$ . Таким образом, SRT является частным случаем MRT.

Промежуточный вариант между SRT и MRT — аппроксимация интеграла столкновений с двумя параметрами релаксации (Two Relaxation Times, TRT) [25, 30]. Основная идея состоит в том, что конструируются по две составляющих для каждого значения функции распределения  $f_i$ , симметричная (1.19a) и антисимметричная (1.19b).

$$f_i^s = \frac{1}{2}(f_i + f_{-i}) \quad (1.19a)$$

$$f_i^a = \frac{1}{2}(f_i - f_{-i}) \quad (1.19b)$$

При этом исходные значения функции распределения представляют собой сумму симметричной и антисимметричной составляющих:

$$f_i = f_i^s + f_i^a \quad (1.20)$$

Интеграл столкновений записывается следующим образом:

$$\Omega(f_i) = \frac{f_i^{seq} - f_i^s}{\tau_s} + \frac{f_i^{aeq} - f_i^a}{\tau_a} \quad (1.21)$$

В случае  $\tau_s = \tau_a$  TRT сводится к SRT.

Кинематическая вязкость в TRT рассчитывается как [39]:

$$\nu = c_s^2 (\tau_s - 0.5) \quad (1.22)$$

### 1.1.3. Учет внешней силы

Существует несколько различных моделей учета внешней силы в методе решеточных уравнений Больцмана [11, 28, 55, 68, 71]. Они отличаются способами учета поправки к значениям функции распределения, возникающей из-за действия объемной силы. В некоторых работах изменяют только равновесное распределение скорости [68], другие включают дополнительный член в интеграл столкновений [50], третьи используют оба подхода одновременно [11, 28].

Самой простой моделью является учет объемной силы в виде введения поправки к значениям функции распределения на шаге столкновения:

$$|f(\mathbf{r},t)^*\rangle = \Omega(f(\mathbf{r},t)) + |F\rangle \quad (1.23)$$

где  $|F\rangle = (F_1, F_2, \dots, F_{q-1})^T$  — поправка, вызванная действием объемной силы.

Поправка рассчитывается на основе внешней силы  $\mathbf{F}$  [50]:

$$F_i = -3w_i \rho \frac{\mathbf{e}_i \mathbf{F}}{c^2} \quad (1.24)$$

В более поздних работах [68]  $F_i$  записывается в более общем виде:

$$F_i = -3w_i \left( \frac{\mathbf{e}_i - \mathbf{u}}{c^2} + 3 \frac{\mathbf{e}_i \mathbf{u}}{c^4} \mathbf{e}_i \right) \mathbf{F} \quad (1.25)$$

В некоторых работах [71] учет внешней силы производится за счет сдвига равновесного распределения скорости:

$$\mathbf{u}^{eq} = \mathbf{u} + \frac{\tau \mathbf{F}}{\rho}. \quad (1.26)$$

В последнее время чаще всего используются модели [11], модифицирующие и равновесную скорость, и интеграл столкновений:

$$F_i = -\frac{4}{c^2} \left( 1 - \frac{1}{2\tau} \right) \mathbf{F} \mathbf{e}_i \quad (1.27a)$$

$$\mathbf{u}^{eq} = \frac{\sum \mathbf{e}_i f_i}{\sum f_i} + \frac{\mathbf{F} \Delta t}{2\rho} \quad (1.27b)$$

Позже была опубликована [28] усовершенствованная версия данного подхода:

$$F_i = w_i \left( 1 - \frac{1}{2\tau} \right) \left( \frac{\mathbf{e}_i - \mathbf{u}}{c_s^2} + 3 \frac{\mathbf{e}_i \mathbf{u}}{c_s^4} \mathbf{e}_i \right) \mathbf{F} \quad (1.28a)$$

$$\mathbf{u}^{eq} = \frac{\sum \mathbf{e}_i f_i}{\sum f_i} + \frac{\mathbf{F} \Delta t}{2\rho} \quad (1.28b)$$

#### 1.1.4. Границные условия

Границные условия в методе решеточных уравнений Больцмана можно разделить на два типа: на краях непроницаемых областей и на границах вычислительного объема.

Наиболее часто на краях непроницаемых областей используются граничные условия без проскальзывания [70, 79]: межрешеточный отскок (Midgrid Bounceback), нарешеточный отскок (Ongrid Bounceback) и различные промежуточные интерполяционные схемы [8, 19, 42, 53]. Межрешеточный отскок соответствует ситуации, когда непроницаемая граница располагается точно

между ячейками и имеет второй порядок точности [40], нарешеточный отскок описывает ситуацию, когда положение непроницаемой стенки совпадает с границами ячеек.

Существует несколько вариантов интерполяционных схем, рассматривающих произвольное расположение непроницаемой границы. Наиболее известные из них [38]: модель Филипповой и Хэнеля (ФХ) [19, 53], модель Боузиди-Фирдауса-Лалемманда (БФЛ) [8] и безинтерполяционная модель [42].

В модели ФХ неизвестные значения функции распределения после шагов столкновения и распространения  $f_{-i}(\mathbf{r}_A, t + 1)$  в точке  $A$ , рассчитываются из известных  $f_i(\mathbf{r}_A, t)$ ,  $f_i(\mathbf{r}_E, t)$ ,  $f_i(\mathbf{r}_F, t)$  после шага столкновения (см. Рис. 1.5):

$$f_{-i}(\mathbf{r}_A, t + 1) = (1 - \chi)f_i(\mathbf{r}_A, t) + \chi f_i^*(\mathbf{r}_B, t) \quad (1.29a)$$

$$f_i^*(\mathbf{r}_B, t) = w_i \rho(\mathbf{r}_A) \left[ 1 + 3(\mathbf{e}_i, \mathbf{u}_{AB}) + \frac{9}{2}(\mathbf{e}_i, \mathbf{u}_{AB})^2 - \frac{3}{2}\mathbf{u}_{AB}^2 \right] \quad (1.29b)$$

Весовой фактор  $\chi$  и скорость  $\mathbf{u}_{AB}$  определяются [53] как:

$$\begin{cases} \mathbf{u}_{AB} = \mathbf{u}(\mathbf{r}_E, t), \chi = \frac{2q-1}{\tau-1} & \text{если } q < \frac{1}{2} \\ \mathbf{u}_{AB} = \frac{q-1}{q}\mathbf{u}(\mathbf{r}_E, t) + \mathbf{u}(\mathbf{r}_C), \chi = \frac{2q-1}{\tau} & \text{если } q \geq \frac{1}{2} \end{cases} \quad (1.30)$$

где  $\mathbf{u}(\mathbf{r}_C)$  — скорость границы. В случае неподвижной границы  $\mathbf{u}(\mathbf{r}_C) = 0$ .

В модели БФЛ для  $q < \frac{1}{2}$  для построения неизвестных значений функции распределения используется нарешеточный отскок, но в качестве известных значений функции распределения берется ее интерполяция в точке  $D$ , расстояние от которой до точки  $A$  с учетом отскока равно единице. В случае  $q \geq \frac{1}{2}$  берется значение функции распределения после отскока в точке  $D$  и строится интерполяция в точке  $A$  на основе данных из  $D, E, F$  (см. Рис. 1.5). В случае линейной интерполяции уравнения принимают вид (значения функции распределения берутся после этапа столкновения):

$$\begin{cases} f_{-i}(\mathbf{r}, t + 1) = q(2q + 1)f_i(\mathbf{r}, t) \\ \quad + (1 + 2q)(1 - 2q)f_i(\mathbf{r} - \mathbf{e}_i, t) & \text{если } q < \frac{1}{2} \\ f_{-i}(\mathbf{r}, t + 1) = \frac{1}{2q(2q + 1)}f_i(\mathbf{r}, t) \\ \quad + \frac{2q - 1}{q}f_{-i}(\mathbf{r}, t) \\ \quad + \frac{1 - 2q}{1 + 2q}f_{-i}(\mathbf{r} - \mathbf{e}_i, t) & \text{если } q \geq \frac{1}{2} \end{cases} \quad (1.31)$$

В случае квадратичной интерполяции уравнения для подели БФЛ принимают вид:

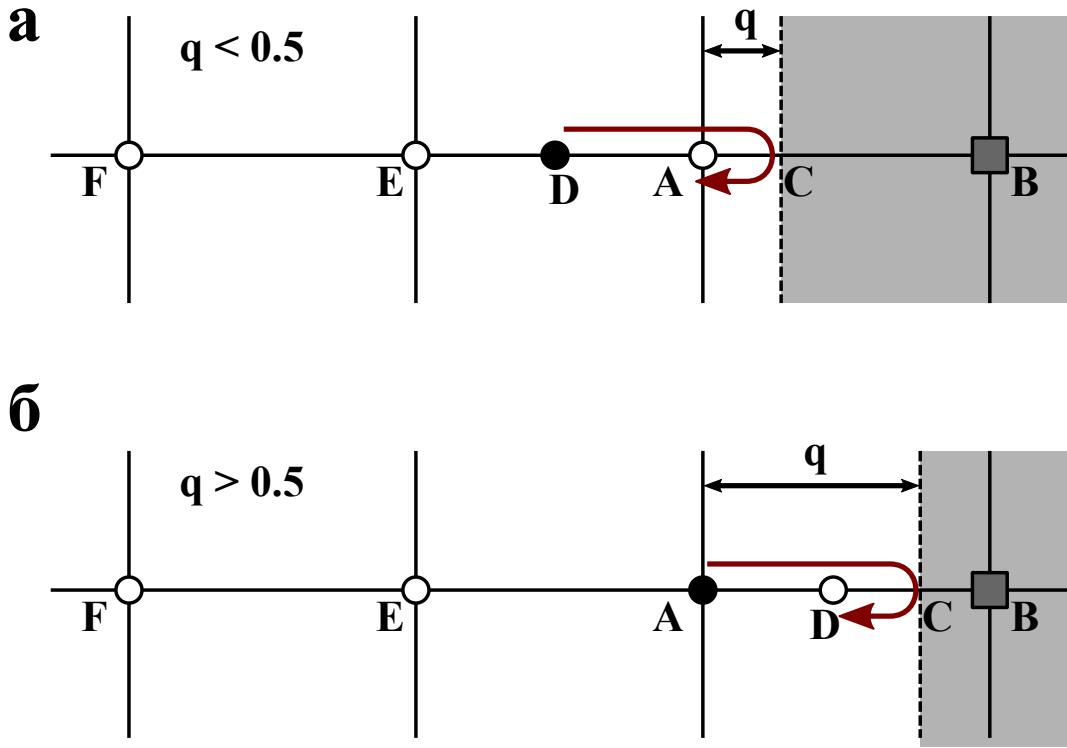


Рисунок 1.5: Ячейки и положения границы в случае положения непроницаемой стенки, не совпадающей с границами ячеек.

$$\begin{cases} f_{-i}(\mathbf{r}, t+1) = 2qf_i(\mathbf{r}, t) \\ \quad + (1-2q)f_i(\mathbf{r} - \mathbf{e}_i, t) & \text{если } q < \frac{1}{2} \\ f_{-i}(\mathbf{r}, t+1) = \frac{1}{2q}f_i(\mathbf{r}, t) \\ \quad + \frac{2q-1}{2q}f_{-i}(\mathbf{r}, t) & \text{если } q \geq \frac{1}{2} \end{cases} \quad (1.32)$$

В случае  $q = 0.5$  (см. Рис. 1.5) интерполяционные схемы переходят в межрешеточный отскок, а в случае  $q = 1$  — в нарешеточный.

На границе вычислительного объема одними из наиболее часто используемых граничных условий являются периодические. В этом случае неизвестные значения функции распределения берутся из противоположного слоя на шаге распространения, а расчетный объем представляет собой гипертор.

Были предложены [56] граничные условия с фиксированной плотностью (в случае изотермического РМБ это эквивалентно фиксированному давлению). В случае сетки D3Q19 на гранях, перпендикулярных оси Z, для этого решается система уравнений относительно неизвестных  $f_i$  и  $u_z$  ( $u_x$  и  $u_y$  на границе полагаются равными нулю):

$$\begin{aligned}
0 &= \sum f_i e_{i_x} \\
0 &= \sum f_i e_{i_y} \\
\rho u_z &= \sum f_i c_{i_z} \\
\rho &= \sum f_i \\
f_3 - f_{10} &= \rho u_z / 3 \\
f_6 - f_{15} &= \rho u_z / 6 + N \\
f_7 - f_{16} &= \rho u_z / 6 - N \\
f_8 - f_{17} &= \rho u_z / 6 + M \\
f_9 - f_{18} &= \rho u_z / 6 - M
\end{aligned} \tag{1.33}$$

где  $M$  и  $N$  — поправочные слагаемые [56]. Данная система уравнений решается относительно неизвестных значений функции распределения  $f_i$ .

Аналогичный подход используется при выводе граничных условий с фиксированной скоростью [31], для этого решается система уравнений относительно неизвестных на границе  $f_i$  и  $\rho$  при заданной скорости  $\mathbf{u} = (u_x, u_y, u_z)$ . Для граней, перпендикулярных оси  $Z$  ( $M$  и  $N$  — поправочные слагаемые):

$$\begin{aligned}
\rho u_x &= \sum f_i e_{i_x} \\
\rho u_y &= \sum f_i e_{i_y} \\
\rho u_z &= \sum f_i e_{i_z} \\
\rho &= \sum f_i \\
f_3 - f_{10} &= \rho u_z / 3 \\
f_6 - f_{15} &= \rho u_z / 6 + N \\
f_7 - f_{16} &= \rho u_z / 6 - N \\
f_8 - f_{17} &= \rho u_z / 6 + M \\
f_9 - f_{18} &= \rho u_z / 6 - M
\end{aligned} \tag{1.34}$$

## 1.2. РМБ для моделирования многокомпонентного течения

Существует несколько подходов к моделированию многокомпонентного течения с помощью метода решеточных уравнений Больцмана: метод цветового градиента (Color Gradient) [44, 47, 48], подход псевдопотенциала (или Шен-Чен, по фамилиям исследователей, предложивших его) [33, 47, 68], и метод свободной энергии (Free Energy) [47, 69, 72].

Метод цветового градиента используется для моделирования поведения строго двух несмешивающихся жидкостей [44, 47, 48]. В этой модели для простоты дальнейших рассуждений компоненты маркируются как «синяя» (все ее переменные для нее обозначаются индексом  $b$ ) и красная (индекс  $r$ ). Покомпонентное уравнение эволюции значений функции распределения, как и однокомпонентном течении, состоит из двух этапов, столкновения и распространения:

$$|f^k(\mathbf{r}, t)^*\rangle = \Omega^k(|f^k(\mathbf{r}, t)\rangle) \quad (1.35)$$

$$|f^k(\mathbf{r} + \mathbf{e}, t + 1)\rangle = |f^k(\mathbf{r}, t)^*\rangle \quad (1.36)$$

где вектор значений функции распределения обозначен как  $|f^k(\mathbf{r}, t)\rangle$ ,  $k$  обозначает красную или синюю жидкость.

Интеграл столкновений состоит из трех частей:

$$\Omega^k(\cdot) = \Omega_3^k (\Omega_1^k(\cdot) + \Omega_2^k(\cdot))$$

$\Omega_1^k(\cdot)$  — обычный покомпонентный интеграл столкновений, описывающий покомпонентное перераспределение фиктивных частиц внутри ячеек.  $\Omega_2^k(\cdot)$  отвечает за взаимодействие частиц разных компонент на основе цветового градиента — специально введенной функции на границе раздела компонент. Вторая часть интеграла столкновений имеет смысл только в приграничных ячейках. Второй интеграл контролирует поверхностное натяжение.

Последняя часть  $\Omega_3^k(\cdot)$  отвечает за сохранение тонкого интерфейса между компонентами, устраняет перемешивание и называется «перекрашиванием». Перекрашивание позволяет контролировать толщину переходного слоя между компонентами, но порождает паразитные токи на границе раздела компонент и может приводить к «прилипанию» к сетке в случае медленных течений [43]. Тем не менее, данный подход позволяет моделировать течение компонент с различиями в плотностях до 1000 раз [44].

В подходе Шена и Чена [33, 47, 68] взаимодействие между компонентами флюида учитывается через изменение интеграла столкновений и равновесной функции распределения. Уравнение эволюции значений функции распределения:

$$|f^k(\mathbf{r}, t)^*\rangle = \Omega^k(|f^k(\mathbf{r}, t)\rangle) \quad (1.37)$$

$$|f^k(\mathbf{r} + \mathbf{e}, t + 1)\rangle = |f^k(\mathbf{r}, t)^*\rangle \quad (1.38)$$

Интеграл столкновений записывается в однорелаксационном приближении:

$$\Omega^k(f_i^k(\mathbf{r}, t)) = f_i^k(\mathbf{r}, t) - \frac{1}{\tau^k} \left( f_i^k(\mathbf{r}, t) - f_i^{k, eq}(\mathbf{r}, t) \right)$$

В уравнении для равновесных значений функции распределения:

$$f_i^{k, eq}(\mathbf{r}, t) = \rho^k w_i \left( 1 + 3\mathbf{e}_i \mathbf{u}^{k, eq} + \frac{9}{2} (\mathbf{e}_i \mathbf{u}^{k, eq})^2 - \frac{3}{2} (\mathbf{u}^{k, eq})^2 \right)$$

взаимодействие компонент между собой  $\mathbf{F}^k$  учитывается через равновесную скорость  $\mathbf{u}^{k, eq} = \mathbf{u} + \frac{\tau^k \mathbf{F}^k}{\rho^k}$ . Существует несколько различных вариантов задания потенциала, определяющего  $\mathbf{F}^k$ , поэтому метод Шена и Чена иногда называют методом псевдопотенциала.

Метод псевдопотенциала в настоящее время является одной из самых популярных моделей многокомпонентного течения в рамках метода решеточных уравнений Больцмана. С одной стороны, это вызвано простотой реализации и вычислительной эффективностью, с другой стороны — возможностями учета дополнительных эффектов через модифицирование формы потенциала. Основным недостатком данной модели можно считать размытый граничный слой между компонентами и сильные паразитные токи в нем.

Метод функционала свободной энергии [47, 69, 72] построен на основе теории фазовых переходов, для описания явлений на границе раздела фаз используется функционал свободной энергии. С точки зрения реализации в рамках методов решеточных уравнений Больцмана, вводятся функции распределения  $f$  и  $g$ :

$$|f(\mathbf{r} + \mathbf{e}\delta t, t + \delta t)\rangle = |f(\mathbf{r}, t)\rangle + \frac{1}{\tau_f} (|f^e q(\mathbf{r}, t)\rangle - |f(\mathbf{r}, t)\rangle) \quad (1.39a)$$

$$|g(\mathbf{r} + \mathbf{e}\delta t, t + \delta t)\rangle = |g(\mathbf{r}, t)\rangle + \frac{1}{\tau_g} (|g^e q(\mathbf{r}, t)\rangle - |g(\mathbf{r}, t)\rangle) \quad (1.39b)$$

Равновесные распределения записываются следующим образом [72]:

$$f_i^{eq}(\rho, \mathbf{u}) = F_i w_i \rho \left[ \frac{(\mathbf{e}_i \mathbf{u})}{c_s^2} + \frac{(\mathbf{e}_i \mathbf{u})^2}{2c_s^4} - \frac{(\mathbf{u}, \mathbf{u})}{2c_s^2} \right] \quad (1.40a)$$

$$g_i^{eq}(\rho, \mathbf{u}) = w_i \left[ \frac{\Gamma \mu}{c_s^2} + \phi \left( \frac{(\mathbf{e}_i \mathbf{u})}{c_s^2} + \frac{(\mathbf{e}_i \mathbf{u})^2}{2c_s^4} - \frac{(\mathbf{u}, \mathbf{u})}{2c_s^2} \right) \right] \quad (1.40b)$$

где  $M = \Gamma(\tau_g - 0.5)$ ,  $M$  — параметр модели,  $\phi = \langle 1 | g \rangle$ ,  $\mu$  входит в соотношение  $\sum_{i=0}^{q-1} g_i \mathbf{e}_i \mathbf{e}_i^T = \Gamma \mu \mathbf{I} + \phi \mathbf{u} \mathbf{u}^T$ , а значение коэффициента  $F_i$  зависит от выбранного шаблона.

### 1.3. РМБ для моделирования анизотропной диффузии

Уравнение диффузии-адвекции имеет вид:

$$\frac{\partial \phi}{\partial t} + \frac{\partial}{\partial x_j} (\phi v_j) = \frac{\partial}{\partial x_j} \left( D_{ij} \frac{\partial \phi}{\partial x_j} \right) + S \quad (1.41)$$

где  $S$  описывает источники (стоки) диффузирующего вещества,  $D_{ij}$  — тензор диффузии,  $\mathbf{u} = (v_x, v_y, v_z)$  — скорость среды, индексы  $i$  и  $j$  используются для обозначений векторов трехмерного пространства, по парным индексам предполагается суммирование. В начальный момент времени концентрация  $\phi$  известна во всем пространстве.

В общем виде решеточное уравнение Больцмана для уравнения диффузии-адвекции конструируется следующим образом [78]:

$$|f(\mathbf{r} + \mathbf{e}\delta t, t + \delta t)\rangle = |f(\mathbf{r}, t)\rangle - L |f(\mathbf{r}, t)\rangle + \Theta |\omega\rangle \quad (1.42)$$

$$L = M^{-1} S M (|\omega\rangle + U_e |\omega\rangle) \langle 1| - I \quad (1.43)$$

$$U_e = \text{diag}(\mathbf{u}\mathbf{e}_0, \dots, \mathbf{u}\mathbf{e}_{q-1}) \quad (1.44)$$

где  $|\omega\rangle$  — вектор параметров релаксации,  $L$  - оператор столкновений,  $\Theta$  — оператор учета внешних источников,  $S$  — оператор столкновений,  $\langle 1| = (1, \dots, 1)$ ,  $I$  — единичная матрица.

Преимущество использования метода решеточных уравнений Больцмана для решения уравнения диффузии заключается в простоте реализации на основе уже существующей реализации РМБ.

### 1.4. РМБ для моделирования теплопереноса

Метод решеточных уравнений Больцмана может быть адаптирован для численного решения уравнения теплопроводности [34]. В литературе эта модель в рамках РМБ называется методом пассивного скаляра. При этом его следует рассматривать как инструмент, позволяющий получить численное решение задачи распространения тепла. Данная модель рассматривает случай однородного неизменного коэффициента теплопроводности и не учитывает фазовые переходы.

Рассмотрим уравнение теплопроводности следующего вида:

$$\frac{\partial T}{\partial t} + \mathbf{u} \nabla T = \nabla \cdot (\alpha \kappa \nabla T) \quad (1.45)$$

Ему соответствует уравнение эволюции значений «функции распределения температуры» (при этом используется однорелаксационная аппроксимация интеграла столкновений):

$$|g(\mathbf{r} + \mathbf{e}, t + 1)\rangle = |g(\mathbf{r}, t)\rangle - \frac{1}{\tau_T} (|g(\mathbf{r}, t)\rangle - |g^{eq}(\mathbf{u}, T)\rangle) \quad (1.46)$$

$$g_i^{eq} = \omega_i T \left( 1 + \mathbf{e}_i \mathbf{u} + \frac{1}{2} (\mathbf{e}_i \mathbf{u})^2 - \frac{1}{2} u^2 \right) \quad (1.47)$$

$$T = \langle 1 | g \rangle \quad (1.48)$$

где  $\tau_T$  — параметр релаксации, через него определяется теплопроводность.

Компоненты  $g_i^{eq}$  равновесного вектора  $|g^{eq}(\mathbf{r}, t)\rangle = (g_0^{eq}(\mathbf{r}, t), \dots, g_{q-1}^{eq}(\mathbf{r}, t))^T$  равны:

$$g_i^{eq}(T, \mathbf{u}) = w_i T \left[ 1 + \frac{(\mathbf{e}_i \mathbf{u})}{c_s^2} + \frac{(\mathbf{e}_i \mathbf{u})^2}{2c_s^4} - \frac{(\mathbf{u}, \mathbf{u})}{2c_s^2} \right] \quad (1.49)$$

Температура рассчитывается аналогично давлению в моделировании однокомпонентного течения:

$$T = \sum_{i=0}^{q-1} g_i^{eq} \quad (1.50)$$

Коэффициент теплопроводности  $\kappa$  определяется на основе параметра релаксации:

$$\kappa = \frac{1}{3} (\tau_t - 0.5) \quad (1.51)$$

Данный прием позволяет получать численное решение уравнения теплопроводности, в том числе со сложными геометриями. Взаимодействие с гидродинамической моделью осуществляется через локальную скорость среды.

## 1.5. Оптимизация потребления памяти в РМБ

Одним из основных ограничительных факторов метода решеточных уравнений Больцмана является большое потребление памяти. Например, в случае шаблона D3Q19 на каждую ячейку требуется минимум 77 байт: 76 на хранение 19 значений функции распределения с одинарной точностью и 1 байт на хранение статуса проницаемости ячейки.

Шаг столкновения заключается в изменении значений функции распределения внутри ячейки, он локален и может выполняться параллельно. Шаг распространения представляет собой запись  $f_i$  в соседние ячейки, он имеет зависимости по данным от соседних ячеек и требует рекурсивного обхода. Для того, чтобы избежать некорректного перезаписывания значений функции распределения и при этом сделать шаг распространения более простым, зачастую используют два массива («старый» и «новый»): значения функции распределения после столкновения хранятся в «старом» массиве, а записываются с учетом сдвига после распространения в «новый» массив. После

завершения обоих шагов «старый» и «новый» массивы меняются местами. Таким образом, на одну ячейку приходится  $2 * 76 + 1 = 153$  байта в случае шаблона D3Q19, наивной реализации с двумя массивами и использования чисел одинарной точности для хранения значений функции распределения.

Хранение двух дублирующих массивов данных крайне неэффективно с точки зрения использования памяти, которая выступает ограничивающим фактором. Существует несколько приемов [74], позволяющих сократить объем требуемой памяти: алгоритм с дополнительным массивом для одного направления [82], алгоритм со сжатой сеткой [62], обменный алгоритм [51], АА-шаблон [2] и трюк с неявным обменом [67].

Алгоритм с дополнительным массивом для одного направления [82] представляют собой эволюцию алгоритма со «старым» и «новым» массивами, значения функции распределения сгруппированы по направлениям и хранятся в отдельных массивах. Но «новый» массив создается только для одного направления. После шага столкновения  $f_i$  записываются в прямом порядке. Затем последовательно для каждого направления в шаблоне (кроме нулевого) рассчитывается шаг распространения, причем значения функции распределения со смещением после шага распространения записываются в «новый» массив. Как только распространение для данного направления завершено, происходит обмен указателей на «новый» массив и массив  $f_i$  для данного направления. Шаг повторяется для следующего направления и т.д. Оба шага производятся независимо для каждой ячейки и могут осуществляться параллельно, выполнение распространения для всех направлений — последовательно.

Алгоритм со сжатой сеткой [62] также представляют собой эволюцию алгоритма со «старым» и «новым» массивами. В данном алгоритме используется только один основной массив и по дополнительному слою на каждое из направлений: два массива, наложенных друг на друга со смещением на одну ячейку вдоль каждого из направлений используемого шаблона. На шаге распространения в зависимости от номера итерации  $f_i$  записываются со смещением на  $(1,1,1)$  (для нечетных итераций) или на  $(-1,-1,-1)$  (для четных итераций). При этом необходимо соблюдать очередность обхода ячеек таким образом, чтобы соответствующие значения функции распределения не оказались перезаписанным до того, как они были распространены в соседние ячейки. На границах расчетной области перезаписи не происходит за счет использования дополнительных слоев вдоль каждого из направлений.

В обменном алгоритме [51] используется только один массив. После шага столкновения до шага распространения значений функции распределения записываются в обратном порядке (Рис. 1.6, слева). Распространение в данном случае представляет собой попарный обмен значениями функции распределения между соответствующими проницаемыми ячейками (Рис. 1.6, в центре), при этом на границах с непроницаемыми ячейками автоматически применяется граничное условие межрешеточный отскок. Шаг распространения в данной схеме производится

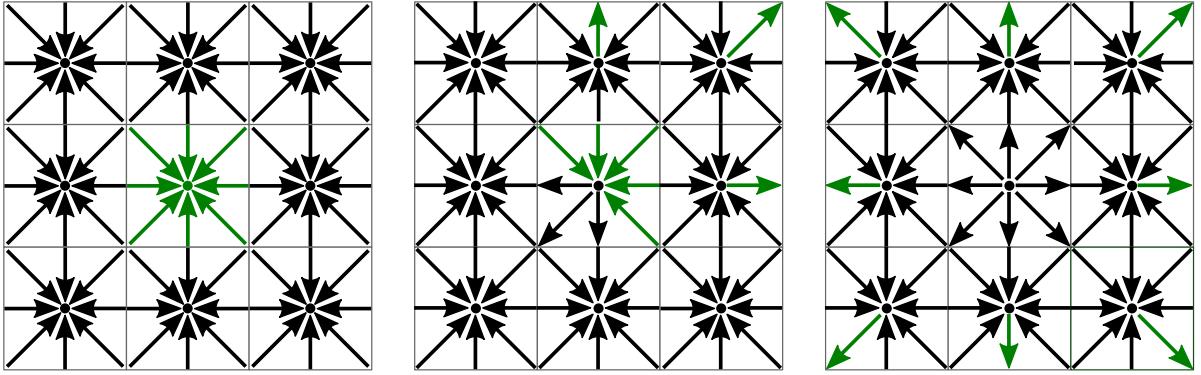


Рисунок 1.6: Обменный алгоритм: после столкновения до распространения (слева), в процессе распространения (в центре), после распространения из центральной ячейки (справа).

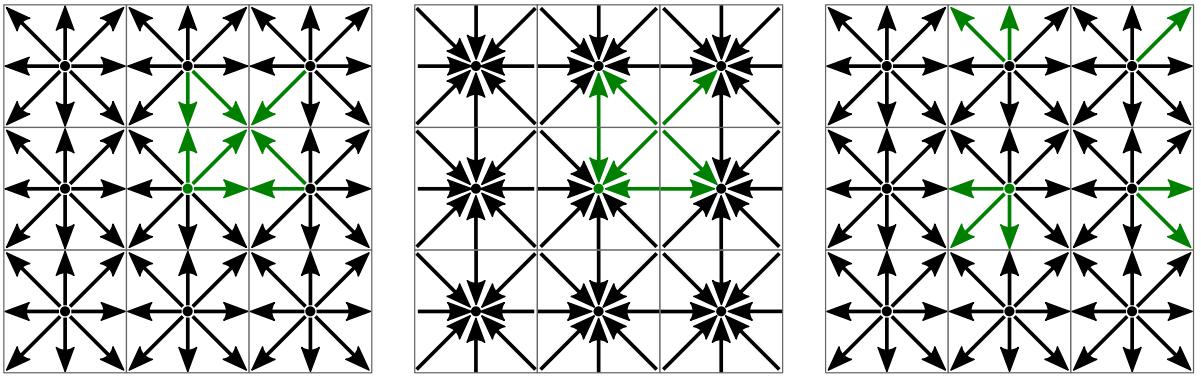


Рисунок 1.7: Трюк с неявным обменом: до столкновения (слева), после столкновения (в центре), после распространения (справа).

независимо для каждой пары значений  $f_i$  и может осуществляться параллельно. После шага распространения  $f_i$  оказываются записаны в прямом порядке (Рис. 1.6, справа).

В АА-шаблоне также используется только один массив, шаги столкновения и распространения перегруппированы в зависимости от номера итерации. На каждой четной итерации производитель только расчет столкновения и запись значений функции распределения в прямом порядке. На каждой нечетной итерации чтение  $f_i$  производится с учетом пропущенного на предыдущей итерации распространения, после шага столкновения  $f_i$  записываются также с учетом распространения. Поэтому на четных итерациях происходит только столкновение, а на нечетных: распространение-столкновение-распространение. Шаги на обеих итерациях производятся независимо для каждой ячейки и могут осуществляться параллельно.

Трюк с неявным обменом [67], как и два предыдущих алгоритма, использует только один массив, разделенный на две половины. В отличие от АА-шаблона, четные и нечетные итерации не отличаются. Значения функции распределения в данном алгоритме хранятся как бы «смещеными» на половину ячейки (Рис. 1.7, слева), при этом противоположным  $f_i$  хранятся в разных половинах массива на соответствующих местах. После шага столкновения значения функции

распределения записываются в обратном порядке (Рис. 1.7, в центре). Распространение в данном случае заключается в обмене указателей на половины массива (Рис. 1.7, справа). Столкновение производится независимо для каждой ячейки и могут осуществляться параллельно, а распространение получается автоматически в результате обмена указателей на половины.

Отдельно следует отметить реализацию решеточных уравнений Больцмана без хранения значений функции распределения [81]. Данная реализация подходит только для частного случая однорелаксационного приближения интеграла столкновений с значением параметра  $\tau = 1$ . При этом каждый раз при столкновении происходит полная релаксация, и значения функции распределения можно однозначно восстановить на основе значений скорости и плотности среды в ячейке. Поэтому, вместо хранения  $q$  значений  $f_i$  достаточно хранить два набора по 4 числа одинарной точности: плотность среды и три компонента скорости, что более, чем в два раза сокращает объем потребляемой памяти в случае шаблона D3Q19.

## 1.6. Заключение к Главе 1

В главе сделан аналитический обзор основных аспектов применения метода решеточных уравнений Больцмана (РМБ) для моделирования одно- и многокомпонентных течений, теплопереноса и диффузии. Все рассмотренные модификации метода для моделирования явлений переноса основаны на решеточном уравнении Больцмана, а потому имеют схожую алгоритмизацию и стратегии оптимизации потребления памяти.

Показано, что для алгоритм РМБ во всех рассмотренных случаях разделяется на два шага: столкновение и распространение. Столкновение производится независимо параллельно в каждой ячейке. На шаге распространения значения функции распределения в ячейке зависят от данных в соседних ячейках, и в общем случае шаг не может выполняться параллельно. Проведен анализ алгоритмов, позволяющих выполнять шаг распространения параллельно и независимо в каждой ячейке, выделены алгоритмы, не требующие для работы дополнительной памяти.

Все рассмотренные модификации РМБ могут быть реализованы для расчетов на вычислительных системах с массивно-параллельной архитектурой, в том числе с вычислениями на графических ускорителях (на гибридных суперкомпьютерных системах).

## ГЛАВА 2. Анализ суперкомпьютерных технологий

В главе проведен анализ основных суперкомпьютерных технологий, необходимых для использования высокопроизводительных вычислительных машин с гибридной архитектурой (с множеством графических ускорителей). Понятие «гибридная архитектура» относительно, в данной работе оно относится к системам, в которых помимо традиционного центрального процессора установлены и используется специализированные графические ускорители (Graphics Processing Unit, GPU). Первыми устройствами, поддерживающими вычисления на CUDA, были видеокарты. Поэтому, несмотря на то, что некоторые современные графические ускорители (например, серия Tesla) уже утратили возможности видеокарт по рендерингу и выводу изображения на экран (у них даже не распаян видеовыход), в данной работе они иногда будут называться видеокартами, т.к. до сих пор имеют с ними много общего во внутреннем строении. Различие между суперкомпьютерами и кластерами (раздел 2.2) введено специально для того, чтобы подчеркнуть важность высокоскоростного соединения между вычислительными узлами.

В § 1 на основе классификации Флинна архитектур вычислительных систем рассматриваются особенности архитектур MISD (Multiple Instruction, Single Data), MIMD (Multiple Instruction, Multiple Data), SISD (Single Instruction, Single Data) и SIMD (Single Instruction, Multiple Data). На примере суперкомпьютера «Ломоносов» показывается, что вычислительные системы на разных уровнях могут соответствовать разным классам по таксономии Флинна: суперкомпьютер «Ломоносов» имеет MIMD архитектуру на верхнем уровне, причем каждый из вычислительных узлов соответствует гибридной MIMD-SIMD архитектуре. В § 2 приводится общая классификация вычислительных систем по организации доступа к памяти. В § 3 перечисляются основные программные инструменты, относящиеся к суперкомпьютерным вычислениям и использованные в данной работе: Message Passing Interface (MPI) и Compute Unified Device Architecture (CUDA). В § 4 предложена методика совместного использования CUDA и MPI для создания универсальных программных комплексов, ориентированных на работу на гибридных суперкомпьютерных системах. Приводится универсальный алгоритм распределения задач между графическими ускорителями в случае, если на каждый вычислительный узел приходится по несколько GPU (на суперкомпьютере «Ломоносов» в каждом вычислительном узле в разделе gpus установлено по два графических ускорителя).

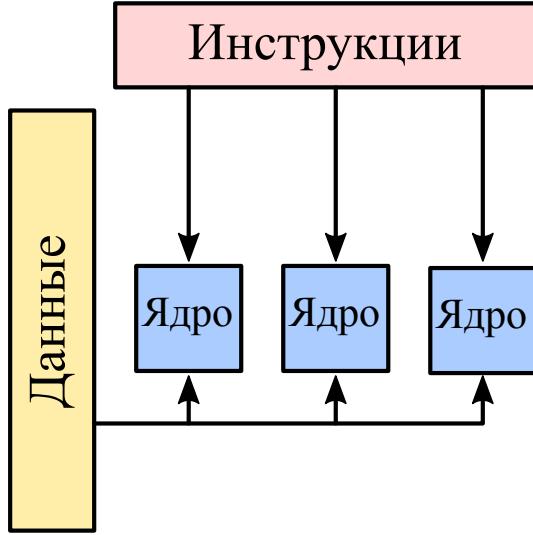


Рисунок 2.1: Архитектура MISD: множество инструкций выполняются на одном потоке данных в единицу времени.

Содержание данной главы представляет собой аналитический обзор особенностей архитектур суперкомпьютерных систем и два типа их классификации, а также обзор программно-аппаратных технологий, использованных в данной работе при создании комплекса программ.

## 2.1. Классификация Флинна архитектур вычислительных систем

Согласно классификации Флинна [20, 21], архитектуры вычислительных систем можно разделить на четыре больших класса: MISD (Multiple Instruction stream and Single Data stream), SISD (Single Instruction stream and Single Data stream), MIMD (Multiple Instruction stream and Multiple Data stream) и SIMD (Single Instruction stream and Multiple Data stream). Эта классификация носит условный характер, и современные компьютеры на разных уровнях могут иметь разную архитектуру. Например, суперкомпьютер «Ломоносов» имеет MIMD архитектуру на верхнем уровне, причем каждый из вычислительных узлов соответствует гибридной MIMD-SIMD архитектуре [73].

В архитектуре MISD (Рис. 2.1) в единицу времени выполняется множество инструкций на одном потоке данных. В реальных задачах более подходящими оказываются другие архитектуры, поэтому компьютеры с такой архитектурой не распространены. Сюда можно отнести отказоустойчивые компьютеры, многократно дублирующие одни и те же вычисления для обнаружения и арбитражной коррекции ошибок. Некоторые исследователи относят к данному классу вычислительные системы с систолической структурой [21].

В архитектуре SISD в единицу времени выполняется одна команда на одном потоке данных. В данном случае нет параллельности ни по инструкциям, ни по данным, и подобная схема вычислений не является параллельной. Компьютеры с такой архитектурой имеют одноядерный про-

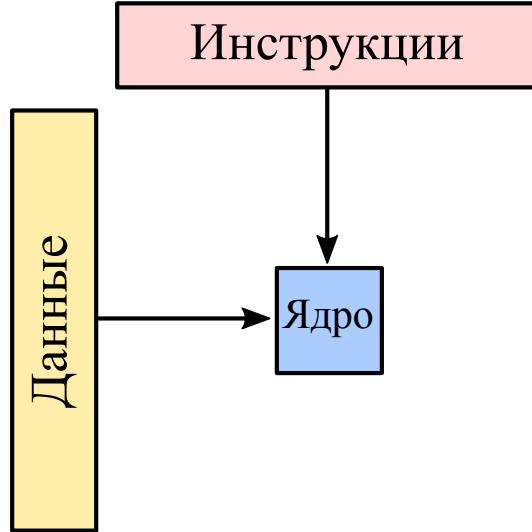


Рисунок 2.2: Архитектура SISD: одна инструкция выполняется на одном потоке данных в единицу времени.

цессор, последовательно выполняющий одиночные команды. Для SISD выполняются все принципы фон Неймана.

В архитектуре MIMD в единицу времени выполняется множество команд на множестве потоков данных. К этому классу традиционно относят многоядерные и многопроцессорные компьютеры, суперкомпьютеры, вычислительные кластеры. Это наиболее универсальная архитектура вычислительных систем.

В архитектуре SIMD в единицу времени параллельно выполняется одна и та же команда на множестве потоков данных. Эта архитектура сейчас широко используется для параллельной обработки больших массивов данных на суперкомпьютерах и кластерах. Современные графические ускорители (Graphics Processing Unit, GPU) построены на этой архитектуре и наиболее эффективно работают в задачах с высокой параллельностью, одной из которых является моделирование с помощью метода решеточных уравнений Больцмана.

## 2.2. Классификация вычислительных систем по доступу к памяти

По организации и доступу к оперативной памяти многопроцессорные системы разделяются на системы с общей памятью и системы с распределенной памятью [21].

В системах с общей памятью все процессоры имеют доступ к памяти как к общему адресному пространству. Это удобно для написания программ и обеспечения взаимодействия между процессорами, но сложно с точки зрения построения подобных систем. Примером параллельного компьютера с общей памятью может служить современный стационарный компьютер или смартфон: все ядра процессора имеют одинаковый доступ к оперативной памяти.

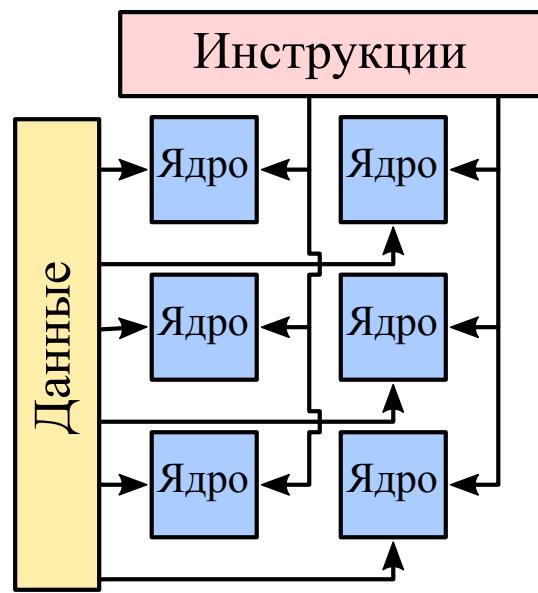


Рисунок 2.3: Архитектура MIMD: множество инструкций выполняется на множестве потоков данных в единицу времени.

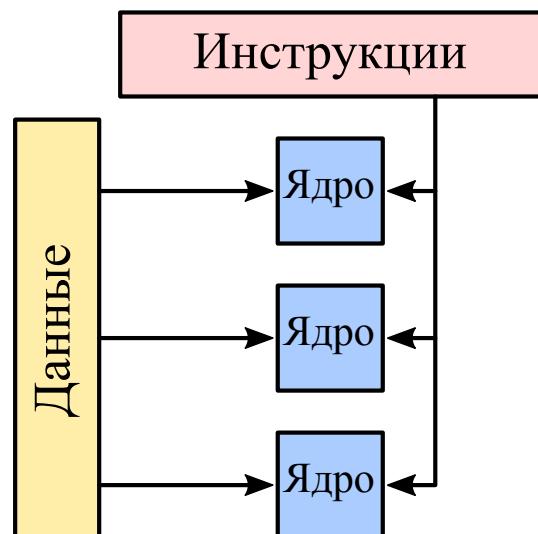


Рисунок 2.4: Архитектура SIMD: одна инструкция выполняется на множестве потоков данных в единицу времени.

В системах с распределенной памятью каждый процессор имеет собственную локальную память, доступную только ему. Например, на каждом вычислительном узле суперкомпьютера «Ломоносов» установлены несколько многоядерных процессоров, имеющих доступ к локальной оперативной памяти — это система с общей памятью. Но процессоры разных вычислительных узлов не имеют доступ к оперативной памяти друг друга — это система с распределенной памятью.

Системы с распределенной памятью хорошо масштабируются, дешевле и проще машин с общей памятью при одинаковой производительности. К недостаткам можно отнести сужение набора алгоритмов, допускающих эффективную реализацию на таких компьютерах из-за задержек на синхронизацию данных между узлами (если это необходимо). Message Passing Interface (MPI) служит наиболее распространенным инструментом для написания и выполнения параллельных программ на системах с распределенной памятью.

Также стоит отметить, что системы с распределенной памятью можно условно разделить на три типа: суперкомпьютеры, кластеры и грид-сети. В суперкомпьютерах для соединения отдельных вычислительных узлов используются специализированные высокоскоростные сети, имеющие высокую пропускную способность и низкую задержку. Наиболее популярной сетью для суперкомпьютеров в настоящее время является Infiniband. Кластер компьютеров также может выглядеть для пользователя как единый вычислительный комплекс, но узлы в нем соединяются с помощью менее скоростного Ethernet. Тем не менее, уже существует стандарт Ethernet, обеспечивающий пропускную способность, сравнимую с Infiniband. Грид-сети отличает низкая пропускная способность соединения и негарантированная доступность вычислительных узлов.

Выбор оптимальной архитектуры вычислительной системы зависит от природы моделируемой задачи. В случае моделирования гидро- и газодинамики с помощью метода решеточных уравнений Больцмана задача параллельна по данным, может быть эффективно реализована на архитектуре SIMD. Большой объем данных требует вычислений на суперкомпьютере с большим объемом оперативной памяти. Параллельность по данным позволяет эффективно использовать системы с распределенной памятью, геометрически разбивая вычислительный объем на фрагменты. Необходимость синхронизации граничных условий между фрагментами задачи требует высокоскоростного соединения между вычислительными узлами. Высочайшая параллельность (сотни тысяч потоков) позволяет эффективно использовать возможности вычислений на графических ускорителях.

## 2.3. Программные инструменты для параллельных и распределенных вычислений

В данном разделе рассмотрены основные программные инструменты для параллельных и распределенных вычислений применительно к языку программирования C++.

### 2.3.1. Message Passing Interface (MPI)

Message Passing Interface (MPI) [22, 26, 27] — стандарт и программный интерфейс для организации параллельного выполнения программ, в первую очередь на системах с распределенной памятью. Предоставляет разнообразные функции для обмена сообщениями между выполняемыми на разных узлах процессами, их синхронизации, построения виртуальной топологии сети и т.д. Состоит из двух компонентов: клиентской библиотеки для использования программистом и серверного комплекса программ, обеспечивающих параллельный запуск целевой программы. Чаще всего используется на суперкомпьютерах. Существует множество реализаций этого интерфейса, одни из самых популярных — OpenMPI и MPICH2, обе установлены на суперкомпьютере «Ломоносов».

В случае систем с распределенной памятью, вычислительные узлы работают независимо друг от друга и не имеют доступа к памяти друг друга. Поэтому возникают как минимум две общие задачи: межузловой обмен сообщениями и синхронизация выполнения. MPI предоставляет инструменты решения упомянутых проблем. Важными особенностями этого программного интерфейса также являются его переносимость на другие вычислительные системы (важно, чтобы серверная часть была на них установлена), эффективность (настроенный на суперкомпьютере MPI содержит информацию о топологии сети и учитывает ее при параллельном запуске программ) и универсальность (MPI работает как на системах с распределенной памятью, так и на многопроцессорных многоядерных системах с общей памятью). MPI легко разворачивается на локальном комплексе: для этого нужны общая сетевая папка, установленный на каждом узле MPI и список IP адресов машин, на которых параллельно будет запущена программа.

При вызове программы, написанной с использованием MPI, явно указывается количество процессов, которое надо запустить, и количество узлов, на которых эти процессы будут запущены. MPI эффективно использует разные ядра многоядерных процессов, поэтому на один вычислительный узел обычно можно ставить более одного процесса с увеличением производительности (меньше задержки из-за передачи данных). Для параллельных вычислений под MPI разрабатывается и многократно запускается единая программа, в которой, с помощью вызовов специальных функций, можно узнать номер копии работающей программы, называемый рангом, и варьировать алгоритм работы в зависимости от него. Такой подход иногда называют SPMD (Single Program and Multiple Data Stream), то есть одна программа и множественные потоки данных.

Межпроцессное взаимодействие (в т.ч. обмен сообщениями и синхронизация процессов) между разными копиями также может осуществляться посредством MPI.

MPI распространен, надежен, повышает эффективность параллельного выполнения программ за счет учета топологии сети суперкомпьютера. Он упрощает разработку и верификацию программного кода, оставляя за кулисами детали межпроцессного взаимодействия. Однако, следует помнить, что он не отказоустойчив: при досрочном завершении любого из запущенных процессов, остальные тут же получают сигнал об этом и также аварийно завершаются. В данной работе MPI используется для синхронизации данных между процессами, запущенными на разных вычислительных узлах.

### 2.3.2. Compute Unified Device Architecture (CUDA)

Compute Unified Device Architecture (CUDA) — программно-аппаратная платформа, предлагающая возможность использовать графические ускорители (Graphics Processing Unit, GPU) компании NVIDIA для проведения расчетов общего назначения. При написании программ используется специальный Си-подобный синтаксис, за компиляцию исходных файлов (обычно имеют расширение .cu) отвечает специальный компилятор nvcc. Библиотека функций и компилятор распространяются бесплатно и доступны для загрузки на сайте NVIDIA, доступны под Windows и Mac OS и основные дистрибутивы Linux. CUDA установлена на суперкомпьютере «Ломоносов».

Изначально GPU предназначались для обработки компьютерной графики, что является массивно параллельной задачей, т.е. допускающей параллельное выполнение в сотнях тысяч параллельных потоков. Поэтому исторически GPU рассчитаны на массивно-параллельные вычисления вычисления. Современные графические ускорители могут полноценно использоваться для высокопроизводительных вычислений общего назначения (general-purpose graphics processing units, GPGPU) [15, 83, 86].

Само строение графических ускорителей предполагает, что наиболее эффективно на них работают алгоритмы, рассчитанные на SIMD модель обработки. На GPU низкие накладные расходы на создание потоков («нитей»), переключение контекста происходит почти мгновенно, а количество нитей обычно достигает сотен тысяч и позволяет производить массивно-параллельные вычисления. Каждая нить выполняет одну и ту же функцию, которая называется ядром и представляет собой обычную функцию со специальным модификатором перед объявлением. Существует иерархия группировки нитей: нити объединяются в блоки, а блоки объединяются в сеть. И блоки, и сеть могут быть одномерными, двухмерными и трехмерными. Эта иерархия позволяет для каждой нити понять ее глобальный номер: внутри нитей доступны координаты нити в блоке и координаты блока в вычислительной сетке [83].

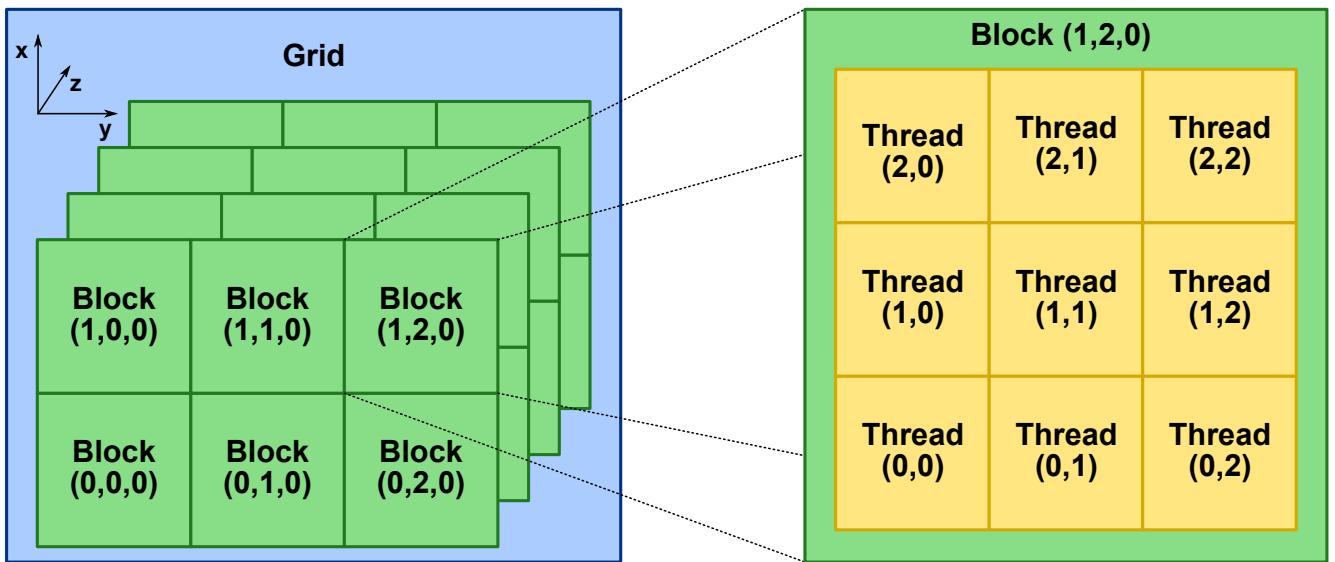


Рисунок 2.5: Иерархия нитей в вычислительной модели CUDA. Нити выполнения объединяются в блоки (справа), блоки могут иметь одно-, двух- и трехмерную структуру. Блоки объединяются в сеть, которая также может иметь одно-, двух- и трехмерную структуру.

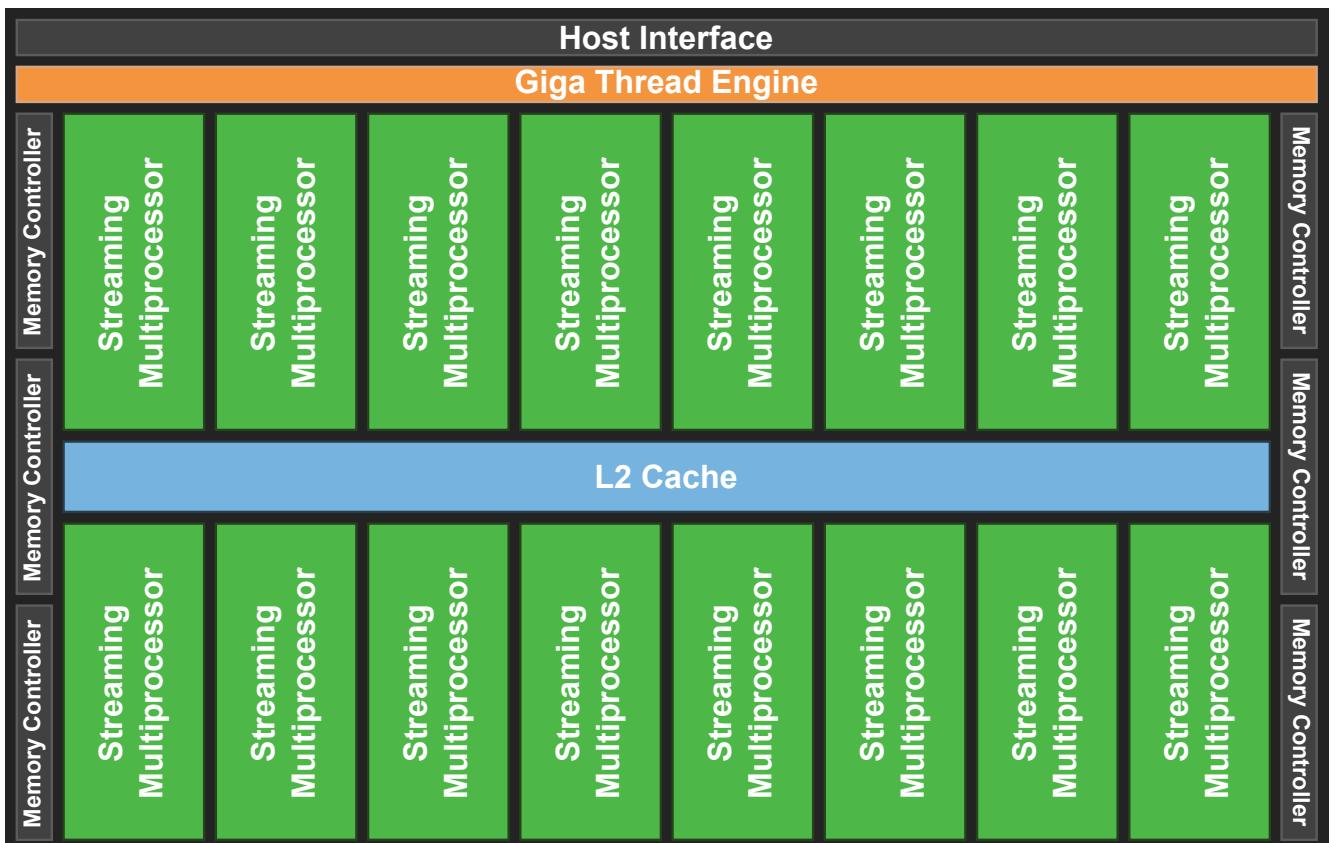


Рисунок 2.6: Общее строение графического ускорителя [86]. Количество потоковых мультипроцессоров (Streaming Multiprocessor) варьируется в зависимости от модели видеокарты.

При вычислениях нити каждого блока объединяются в варпы по 32 нити (warps). После распределения блоков по потоковым мультипроцессорам (о них рассказано ниже), в каждый момент времени лишь нити двух варпов (в архитектуре Kepler уже четырех) выполняются физически параллельно на одном мультипроцессоре. Порядок группировки нитей по варпам и порядок выполнения варпов на одном мультипроцессоре контролируются автоматически и недоступны для ручного управления, но есть несколько свойств, которые гарантируются: в варпе могут быть нити только одного блока, все варпы одного блока выполняются на одном потоковом мультипроцессоре, но на одном мультипроцессоре может считаться несколько блоков.

Различные поколения графических ускорителей от NVIDIA отличаются архитектурой и вычислительными возможностями, т.н. Compute Capability (CC). В настоящий момент последней выпущенной архитектурой является Kepler, CC 3.5. На суперкомпьютере «Ломоносов» установлены вычислители Tesla X2070, имеющие архитектуру Fermi и CC 2.0. Рассмотрим детали обеих архитектур более подробно.

GPU с архитектурой Fermi содержат до 512 вычислительных ядер (шейдерных процессоров) [86]. Каждое ядро, в свою очередь, включает в себя модуль для целочисленных расчетов (arithmetic logic unit, ALU) и модуль для вещественнонозначных расчетов (floating-point unit, FPU). Ядра объединены в группы по 32 элемента, которые называются потоковыми мультипроцессорами (streaming multiprocessor, SM). Помимо шейдерных процессоров в каждом SM расположены 16 банков доступа к общей памяти (Load/Store Units, LD/ST), четыре модуля расчетов специальных функций (Special Function Units, SFU), два планировщика варпов (Warp Schedulers), два модуля управления инструкциями (Dispatch Units), 32 768 32-битных регистров, четыре текстурных модуля, однородный кэш (Uniform Cache), текстурный кэш (Texture Cache), геометрический движок (PolyMorph Engine) и 64 Кб памяти, распределенной между кэшем L1 и общей памятью (Shared Memory), для которой доступны два специализированных режима: 48 Кб для L1 кэша и 16 Кб для общей памяти и наоборот, 16 Кб и 48 Кб соответственно. Общий для всех потоковых мультипроцессоров кэш L2 имеет размер 768 Кб.

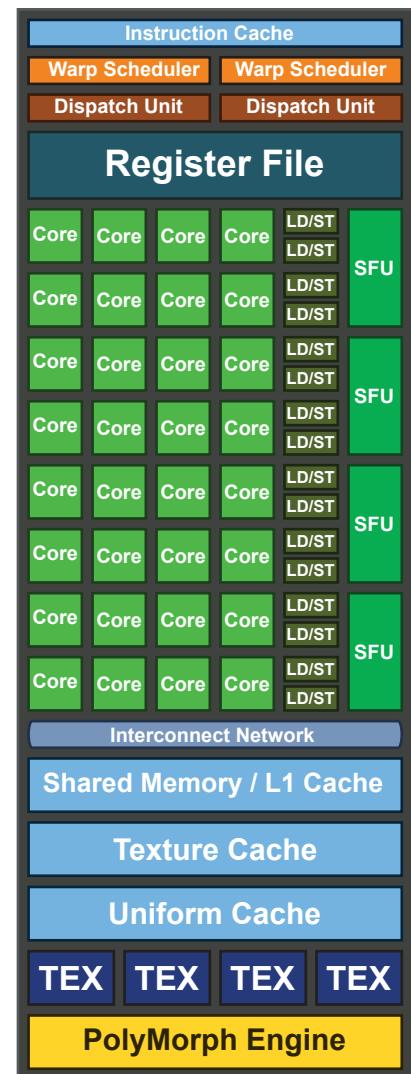


Рисунок 2.7: Строение потокового мультипроцессора поколения Fermi [15].

Основной идеей при создании новой архитектуры было повышение энергоэффективности вычислителей, это привело к значительным изменениям в потоко-

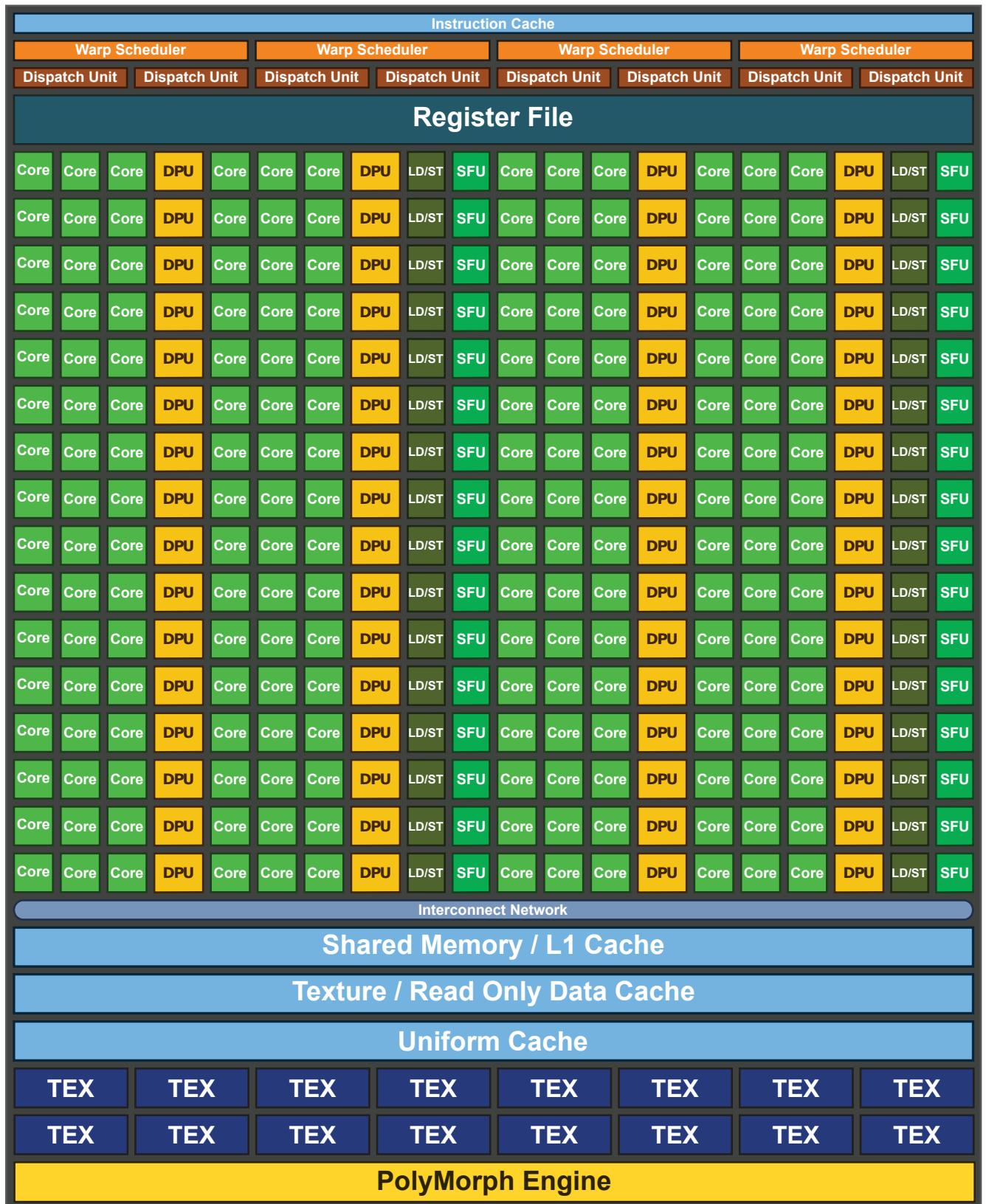


Рисунок 2.8: Схема потокового мультипроцессора поколения Kepler [15].

Таблица 2.1: Сводная таблица характеристик архитектур Fermi и Kepler

Параметр	Fermi	Kepler
Ядер в мультипроцессоре	32	192
Банков памяти	16	32
Блоков расчета специальных функций	4	32
Планировщиков варпов	2	4
Модуль управления инструкциями	2	8
Модуль вычислений двойной точности	нет	64
32-битных регистров	32 768	65 536
Однородный кэш		
Текстурный кэш		
Общая память/L1 кэш	64 Кб	
L2 кэш	768 Кб	1536 Кб

вых мультипроцессорах. GPU с архитектурой Kepler содержат в себе до 2880 вычислительных ядер. С другой стороны, производительность каждого ядра уменьшилась по сравнению ядрами в Fermi. Шейдерные процессоры организованы по группам в 192 элемента, которые называются потоковыми мультипроцессорами нового поколения (Next Generation Streaming Multiprocessor, SMX). Помимо шейдерных процессоров в каждом SM расположены 32 банка доступа к общей памяти, 32 модуля расчетов специальных функций, 64 модуля вычислений с двойной точностью (Double Precision Unit, DPU), четыре планировщика варпов и восемь модулей управления инструкциями, 65 536 32-битных регистров, однородный кэш (Uniform Cache), текстурный кэш (Texture Cache) и 64 Кб RAM, распределенных между кэшем L1 и общей памятью, для которой также доступны три специализированных режима: по 32 Кб на кэш и память, 48 Кб для L1 кэша и 16 Кб для общей памяти и наоборот, 16 Кб и 48 Кб соответственно. Общий для всех потоковых мультипроцессоров кэш L2 теперь размер 1536 Кб. Благодаря увеличенному числу модулей управления инструкциями, на каждый такт SMX может выполнить по две инструкции на 4 варпах, причем вычисления двойной точности могут выполняться одновременно с вычислениями одинарной точности.

В CUDA используется многоуровневая система видов памяти [15], включающая: глобальную память (Global Memory), текстурную память (Texture Memory), константную память (Constant Memory), локальную память (Local Memory), общую память (Shared Memory) и регистры (Registers). Также существуют два уровня кэширования L1 и L2 и два специализированных неизменяемых кэша для констант и текстур. Кэши используются автоматически и не доступны для управления (единственное исключение — существует возможность задания размера кэша L1).

Глобальная память — самая большая по объему и самая медленная. Физически глобальная память расположена в DRAM-памяти видеокарты. Максимальный размер в архитектуре Fermi

составлял 6 Гб, а для архитектуры Kepler уже выпущен вычислитель Tesla K80 с 24 Гб DRAM памяти. Метод решеточных уравнений Больцмана является требовательным по потреблению памяти, т.к. для распространенной трехмерной модели однокомпонентного течения с схемой D3Q19 в каждой ячейке необходимо хранить 19 чисел типа float, а количество вычислительных ячеек в экспериментах может измеряться миллиардами. Поэтому развитие GPU в сторону увеличения DRAM устройств позволяет моделировать все большие объемы без задержек на пересылку данных между несколькими видеокартами. К глобальной памяти имеют доступ все нити.

Текстурная и константная память также физически располагаются в DRAM, но кэшируются в специальных текстурном и константном кэшах соответственно. Отличие текстурной памяти состоит в нескольких дополнительных возможностях, полезных для работы с текстурами: предположения о 2D-локальности данных, аппаратной фильтрации и др. В некоторых случаях она может вести себя медленнее глобальной памяти. Объем константной памяти ограничен 64 Кб на устройство, скорость обращения к ней за счет кэширования на потоковых мультипроцессорах более чем на порядок превосходит скорость доступа к глобальной памяти. К текстурной и константной памяти имеют доступ все нити.

Общая память располагается непосредственно на потоковых мультипроцессорах, доступ к ней более чем на порядок быстрее доступа к глобальной памяти, но и размер значительно меньше. Часто ее используют в виде управляемого вручную кэша L1, но для многих задач хватает автоматического кэширования. К общей памяти имеют доступ только нити одного блока, это ограничение вызвано физическим расположением памяти.

Регистры — самая быстрая память на GPU. Память под все переменные, объявленные в ядрах (в том числе и статически созданные массивы) по умолчанию выделяется в регистровой памяти. К регистровой памяти имеет доступ только та нить, в которой она была выделена. В случае, если память для переменной не может быть выделена в регистровой памяти (например, из-за ее переполнения лимита регистров на ядро), попадают в локальную память. Локальная память — специальная область глобальной памяти, поэтому попадание в локальную память снижает производительность из-за задержек при доступе. На современных GPU доступ к локальной памяти кэшируется в L1 и L2.

## 2.4. Совместное использование CUDA и MPI

Для полного моделирования гидродинамики методом решеточных уравнений Больцмана с сеткой D3Q19 на микромасштабе в большинстве случаев требуется объем памяти, значительно превосходящий размер глобальной памяти одной видеокарты. С другой стороны, РМБ является высокопараллельным алгоритмом с локальными данными и может быть разбит на геометрические фрагменты, каждый из которых рассчитывается параллельно на выделенной видеокарте, данные

Таблица 2.2: Характеристики типов иерархической памяти в CUDA. Запись в текстурную и константную память (отмечены звездочкой) доступна только с хоста.

Название	Расположение	Доступ	Область видимости
глобальная	DRAM	чтение и запись	все нити и хост
текстурная	DRAM	чтение*	все нити и хост
константная	DRAM	чтение*	все нити и хост
локальная	DRAM	чтение и запись	внутри нити
общая	на чипе	чтение и запись	все нити одного блока
регистры	на чипе	чтение и запись	внутри нити

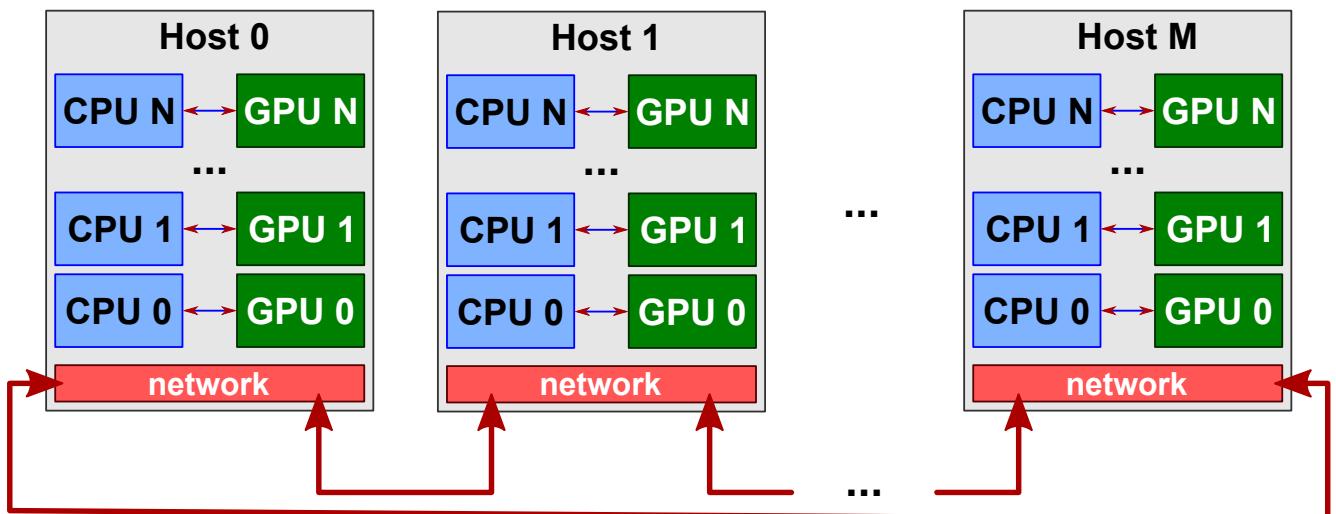


Рисунок 2.9: Схема типичного расчетного модуля для множества видеокарт.

синхронизируются с соседними фрагментами на каждой итерации. Для построения такой схемы в реализованном комплексе программ совместно используются технологии CUDA и MPI. CUDA обеспечивает массивно-параллельное выполнение на уровне отдельных вычислительных узлов, синхронизация необходимых данных между отдельными фрагментами задачи производится посредством MPI.

В последнее время опубликован ряд работ на тему эффективной реализации различных вариантов метода решеточных уравнений Больцмана с использованием вычислений на графических ускорителях [3, 52, 81, 95], в том числе и на суперкомпьютерах с множеством GPU (Multi-GPU кластерах) [18, 32, 57, 58, 76, 82], но в большинстве из них не уделяется особое внимание оптимизации размера потребляемой алгоритмом памяти. Совместное использование MPI и CUDA становится де-факто стандартом переносимых суперкомпьютерных научных вычислений [13, 15, 23, 49, 61, 66, 77].

На суперкомпьютере «Ломоносов» используется восемь различных типов узлов [73], два типа из которых содержат графические сопроцессоры Tesla X2070. Основные характеристики этих

узлов: два Xeon E5630 2.53 GHz с 12 Гб RAM + два Tesla X2070 с 6 Гб DRAM на каждом GPU для первого типа, и два Xeon E5630 2.53 GHz с 24 Гб RAM + два Tesla X2070 с 6 Гб DRAM на каждом GPU для второго типа. Все вычислительные узлы соединены между собой посредством QDR InfiniBand.

На каждом вычислительном узле из раздела GPU располагается по два графических ускорителя. Численные эксперименты показали, что использование многопоточности внутри узлов не дает сильного прироста производительности по сравнению с случаем запуска двух MPI процессов, но при этом усложняет структуру кода. По этой причине для использования всех доступных видеокарт, с помощью MPI запускается по два процесса на каждый узел. При такой конфигурации запуска важно, чтобы каждый процесс задействовал отдельную видеокарту, в то время как номера процессов могут быть распределены по вычислительным узлам случайным образом с единственным условием: не более двух на один узел. Для корректного задания соответствия номера видеокарты номеру процесса предлагается следующий алгоритм:

---

**Алгоритм 1:** Задание неповторяющихся номеров видеокарт. В качестве мастер-процесса может выступать любой процесс, например, нулевой.

---

```
for каждый процесс do
    послать мастер-процессу имя хоста;
end
if мастер-процесс then
    принять все имена хостов в один массив;
    задать номер видеокарты по правилу: если имя хоста уже встречалось в массиве ранее, то
        1, иначе 0;
    послать номера видеокарт всем процессам;
end
for каждый процесс do
    получить номер GPU;
    задать номер GPU;
end
```

---

Для прямой пересылки сообщений между несколькими GPU можно использовать GPUDirect, если эта технология поддерживается оборудованием. Если же она недоступна, используется следующая схема:

---

**Алгоритм 2:** Обмен сообщениями между GPU с помощью MPI

---

```
for каждый MPI процесс do
    CUDA: скачать данные с GPU на хост;
    CPU: упаковать данные;
    MPI: послать данные;
    MPI: принять данные;
    CPU: распаковать данные;
    CUDA: загрузить данные с хоста на GPU;
end
```

---

## 2.5. Заключение к Главе 2

В главе сделан обзор основных суперкомпьютерных технологий, необходимых для использования высокопроизводительных вычислительных машин с гибридной архитектурой (с множеством графических ускорителей). Показано, что метод решеточных уравнений Больцмана из-за локальности и внутренней параллельности по данным подходит для реализации как на системах с общей памятью, так и на системах с распределенной памятью.

Вычисления в РМБ соответствуют вычислительным системам класса SIMD в классификации Флинна архитектур вычислительных систем, том числе имеющих гибридную архитектуру (с расчетами на графических ускорителях).

Предложен алгоритм распределения задач по графическим ускорителям в случае, если на каждый вычислительный узел приходится более одной GPU (gpu-раздел суперкомпьютера «Ломоносов»). Предложена методология двухуровневого распараллеливания вычислений для массивно-параллельного моделирования задач с локальностью по данным: на каждом отдельном узле выполняется параллельный расчет с помощью Compute Unified Device Architecture (CUDA), обмен сообщениями между узлами осуществляется по Message Passing Interface (MPI). Рассмотрено совместное использование этих технологий для создания универсальных программных комплексов, ориентированных на работу на гибридных суперкомпьютерных системах.

## **ГЛАВА 3. Описание математических моделей и их реализации**

В данной главе на основе предложенной методологии рассматриваются детали модификаций решеточных уравнений Больцмана (РМБ), использованных при создании программного комплекса, рассчитанного на суперкомпьютерное моделирование однокомпонентного течения, многокомпонентного течения, процессов диффузии и теплопереноса. Описывается механическая модель упаковки эластичных шаров под нагрузкой, которая основывается на методе дискретных элементов и реализована в рамках данной работы с использованием вычислений на графических ускорителях. Рассматриваются результаты оценки масштабируемости реализованного программного комплекса и результаты тестов, верифицирующих реализованный программный комплекс.

В § 1 детально описываются модификации метода решеточных уравнений Больцмана для моделирования однокомпонентного течения, двухкомпонентного течения, диффузии и теплопереноса, реализованные в программном комплексе. Приводятся необходимые значения констант и схемы выбранных шаблонов скоростей. Приводится вывод использованных граничных условий. В § 2 приводится описание механической модели проппантной упаковки под нагрузкой (проппант — гранулярный материал, используемый в нефтедобывающей промышленности). В § 3 производится оценка производительности программного комплекса и сравнение производительности реализаций, производящих расчеты только на графическом ускорителе (GPU) и только на центральном процессоре (CPU). В § 4 приводятся описания и результаты тестовых вычислительных экспериментов, предназначенных для верификации программного комплекса: течение Пуазейля в цилиндрической трубе, определение абсолютной проницаемости цилиндрической трубы, определение лобового сопротивления шара, оценка избытка давления внутри шарообразной капли (формула Лапласа), оценка соответствия фактического и заданного в модели углов смачивания, коалесценция двух капель под действием сил поверхностного натяжения, многоточечная анизотропная диффузия.

Содержание данной главы представляет собой подробное описание моделей, реализованных в программном комплексе, особенностей их оптимизации для расчетов на гибридном суперкомпьютере, а также постановки и параметры верификационных тестовых вычислительных экспериментов, направленных на проверку программного комплекса.

## 3.1. Метод решеточных уравнений Больцмана для моделирования процессов переноса

### 3.1.1. Моделирование однокомпонентного течения

В данной работе для моделирования многокомпонентного течения используется многорелаксационная аппроксимация интеграла столкновений [16] (MRT). В отличие от однорелаксационной аппроксимации [5], MRT лишена таких недостатков, как зависимость проницаемости от вязкости среды и численная неустойчивость при значениях параметра релаксации, близком к 0.5 [59].

Уравнение эволюции значений функции распределения  $f_i$  в случае MRT и шаблона D3Q19 (см. Рис. 1.3):

$$|f(\mathbf{r} + \mathbf{e}\delta t, t + \delta t)\rangle = |f(\mathbf{r}, t)\rangle - \mathbf{M}^{-1}\mathbf{S}(|m\rangle - |m^{eq}\rangle) \quad (3.1)$$

где  $\mathbf{S} = \text{diag}(s_0, \dots, s_{q-1})$  — диагональная матрица столкновений,  $|m\rangle = (m_0, \dots, m_{q-1})^T$  — вектор моментов,  $\mathbf{M}$  — ортогональная матрица перехода  $|m\rangle = \mathbf{M}|f\rangle$ ,  $|f(\mathbf{r} + \mathbf{e}\delta t, t + \delta t)\rangle = (f_0(\mathbf{r} + \mathbf{e}_0\delta t, t + \delta t), \dots, f_{q-1}(\mathbf{r} + \mathbf{e}_{q-1}\delta t, t + \delta t))^T$  — вектор значений функции распределения после шагов столкновения и распространения,  $|f(\mathbf{r}, t)\rangle = (f_0(\mathbf{r}, t), \dots, f_{q-1}(\mathbf{r}, t))^T$  — вектор значений функции распределения до шага столкновения,  $q = 19$  — количество выделенных направлений скоростей в шаблоне, векторы  $\mathbf{e}_i$  задают решетку направлений (см. Рис. 1.3 и (3.3)). Далее без потери общности полагается  $\delta t = 1$ .

Уравнение (3.1) разделяется на шаг столкновения (3.2a) и шаг распространения (3.2b):

$$|f^*(\mathbf{r}, t)\rangle = |f(\mathbf{r}, t)\rangle - \mathbf{M}^{-1}\mathbf{S}(|m\rangle - |m^{eq}\rangle) \quad (3.2a)$$

$$|f(\mathbf{r} + \mathbf{e}\delta t, t + \delta t)\rangle = |f^*(\mathbf{r}, t)\rangle \quad (3.2b)$$

Для расчета локальных плотности  $\rho$  и скорости  $\mathbf{u}$  среды используются нулевой и первый моменты соответственно:

$$\rho = \langle \mathbf{1} | f \rangle$$

$$\rho \mathbf{u} = \langle \mathbf{e} | f \rangle$$

где  $\langle \mathbf{1} | = (1, \dots, 1)$  — единичный вектор,  $\langle \mathbf{e} | = (\mathbf{e}_0, \dots, \mathbf{e}_{q-1})$  — векторы направлений скоростей в шаблоне.

Выбран следующий порядок направлений в шаблоне D3Q19 (направлениям  $i = 1, \dots, 9$  соответствуют противоположные направления  $(i + 9)$  для  $i \in [1, 9]$ ):

$$\mathbf{e} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & -1 & 0 & -1 & 0 & -1 & -1 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}^T \quad (3.3)$$

Весовые коэффициенты для направлений из (3.3):

$$w = \left[ \frac{12}{36} \quad \frac{2}{36} \quad \frac{2}{36} \quad \frac{2}{36} \quad \frac{1}{36} \quad \frac{1}{36} \quad \frac{1}{36} \quad \frac{1}{36} \quad \frac{2}{36} \quad \frac{2}{36} \quad \frac{2}{36} \quad \frac{1}{36} \right]^T \quad (3.4)$$

Иногда может быть удобно обозначать противоположные направления отрицательным индексом, например:  $\mathbf{e}_1 = -\mathbf{e}_{-1} = -\mathbf{e}_{10}$ .

Матрица перехода  $\mathbf{M}$  из пространства значений функции распределения в пространство значений моментов построена с использованием процесса Грама-Шмидта, ортогональна и задается следующим образом [16]:

$$\mathbf{M} =$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -30 & -11 & -11 & -11 & 8 & 8 & 8 & 8 & 8 & 8 & -11 & -11 & -11 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & -4 & -4 & -4 & 1 & 1 & 1 & 1 & 1 & 1 & -4 & -4 & -4 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 \\ 0 & -4 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 4 & 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & -1 & 0 & -1 & 0 & -1 & -1 & 0 & 0 & -1 & 1 \\ 0 & 0 & -4 & 0 & 1 & 1 & 0 & 0 & 1 & -1 & 0 & 4 & 0 & -1 & -1 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & -1 & 0 & 0 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & -4 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 4 & 0 & 0 & -1 & -1 & -1 & -1 \\ 0 & 2 & -1 & -1 & 1 & 1 & 1 & 1 & -2 & -2 & 2 & -1 & -1 & 1 & 1 & 1 & 1 & -2 & -2 \\ 0 & -4 & 2 & 2 & 1 & 1 & 1 & 1 & -2 & -2 & -4 & 2 & 2 & 1 & 1 & 1 & 1 & -2 & -2 \\ 0 & 0 & 1 & -1 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 1 & -1 & 1 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & -2 & 2 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & -2 & 2 & 1 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix}$$

Она соответствует следующему базису:

$$\begin{aligned}
|\phi_0\rangle_\alpha &= \|\mathbf{e}_\alpha\|^0 \\
|\phi_1\rangle_\alpha &= 19 \|\mathbf{e}_\alpha\|^2 - 30 \\
|\phi_2\rangle_\alpha &= 0.5(21 \|\mathbf{e}_\alpha\|^4 - 53 \|\mathbf{e}_\alpha\|^2 + 24) \\
|\phi_3\rangle_\alpha &= e_{\alpha_x} \\
|\phi_4\rangle_\alpha &= e_{\alpha_y} \\
|\phi_5\rangle_\alpha &= e_{\alpha_z} \\
|\phi_6\rangle_\alpha &= (5 \|\mathbf{e}_\alpha\|^2 - 9)e_{\alpha_x} \\
|\phi_7\rangle_\alpha &= (5 \|\mathbf{e}_\alpha\|^2 - 9)e_{\alpha_y} \\
|\phi_8\rangle_\alpha &= (5 \|\mathbf{e}_\alpha\|^2 - 9)e_{\alpha_z} \\
|\phi_9\rangle_\alpha &= 3e_{\alpha_x}^2 - \|\mathbf{e}_\alpha\|^2 \\
|\phi_{10}\rangle_\alpha &= e_{\alpha_y}^2 - e_{\alpha_z}^2 \\
|\phi_{11}\rangle_\alpha &= e_{\alpha_x}e_{\alpha_y} \\
|\phi_{12}\rangle_\alpha &= e_{\alpha_y}e_{\alpha_z} \\
|\phi_{13}\rangle_\alpha &= e_{\alpha_x}e_{\alpha_z} \\
|\phi_{14}\rangle_\alpha &= (3 \|\mathbf{e}_\alpha\|^2 - 5)(3e_{\alpha_x}^2 - \|\mathbf{e}_\alpha\|^2) \\
|\phi_{15}\rangle_\alpha &= (3 \|\mathbf{e}_\alpha\|^2 - 5)(e_{\alpha_y}^2 - e_{\alpha_z}^2) \\
|\phi_{16}\rangle_\alpha &= (e_{\alpha_y}^2 - e_{\alpha_z}^2)e_{\alpha_x} \\
|\phi_{17}\rangle_\alpha &= (e_{\alpha_z}^2 - e_{\alpha_x}^2)e_{\alpha_y} \\
|\phi_{18}\rangle_\alpha &= (e_{\alpha_x}^2 - e_{\alpha_y}^2)e_{\alpha_z}
\end{aligned}$$

Оператор столкновений представляет собой диагональную матрицу, составленную из покомпонентных параметров релаксации [16]:

$$\hat{S} = \text{diag}(0, 1.19, 1.4, 0, 1.2, 0, 1.2, 0, 1.2, \omega, 1.4, \omega, 1.4, \omega, \omega, 1.98, 1.98, 1.98) \quad (3.5)$$

где  $\omega = \frac{1}{\tau}$ , а  $\tau$  — параметр релаксации, который связан с кинематической вязкостью соотношением:

$$\nu = \frac{1}{3}(\tau - 0.5) \quad (3.6)$$

Равновесные значения моментов  $m^{eq}$  при заданных скорости и плотности среды образуют вектор:

$$m^{eq} = (\rho^{eq}, \epsilon^{eq}, \varepsilon^{eq}, j_x^{eq}, q_x^{eq}, j_y^{eq}, q_y^{eq}, j_z^{eq}, q_z^{eq}, 3p_{xx}^{eq}, 3\pi_{xx}^{eq}, p_{\omega\omega}^{eq}, \pi_{\omega\omega}^{eq}, p_{xy}^{eq}, p_{yz}^{eq}, p_{xz}^{eq}, m_x^{eq}, m_y^{eq}, m_z^{eq})^T$$

где  $\rho^{eq} = \sum_{i=0}^{q-1} f_i$ ,  $\epsilon^{eq} = -11\rho + 19\mathbf{j}^2$ ,  $\varepsilon^{eq} = \omega_\varepsilon \rho + \omega_{\varepsilon j} \mathbf{j}^2$ ,  $\mathbf{j}^{eq} = \sum_{i=0}^{q-1} \mathbf{e}_i f_i$ ,  $\mathbf{q}^{eq} = \frac{2}{3}\mathbf{j}$ ,  $p_{xx}^{eq} = \frac{1}{3}(2j_x^2 - j_y^2 - j_z^2)$ ,  $p_{\omega\omega}^{eq} = j_y^2 - j_z^2$ ,  $p_{xy}^{eq} = j_x j_y$ ,  $p_{yz}^{eq} = j_y j_z$ ,  $p_{xz}^{eq} = j_x j_z$ ,  $\pi_{xx}^{eq} = \omega_{xx} p_{xx}$ ,  $\pi_{\omega\omega}^{eq} = \omega_{xx} p_{\omega\omega}$ ,  $\mathbf{m}^{eq} = 0$ .

В данной работе используются следующие значения свободных параметров [16, 41]:  $\omega_\varepsilon = 0$ ,  $\omega_{\varepsilon j} = -\frac{475}{63}$  и  $\omega_{xx} = 0$ .

В случае, если для моделирования требуется учет объемной силы, в данной работе применяется простая модель с изменением равновесного распределения скоростей [71]. Для этого при расчете равновесных значений моментов вместо  $\mathbf{u}$  используется смещенное значение  $\mathbf{u}^F$ :

$$\mathbf{u}^F = \mathbf{u} + \frac{\tau \mathbf{F}}{\rho} \quad (3.7)$$

### 3.1.2. Моделирование двухкомпонентного течения

Для моделирования двухкомпонентного течения используется модель цветового градиента метода решеточных уравнений Больцмана [43, 44, 48]. К основным преимуществам данной модели относят возможность моделирования двух несмешивающихся жидкостей с разными плотностями [44], а недостатком является ее чисто имитационная природа, не сводящаяся напрямую к физическим уравнениям.

В данной модели рассматривается два типа жидкости, которые для простоты в дальнейшем называются «красная» (ее параметры имеют индекс  $r$ ) и «синяя» (ее параметры имеют индекс  $b$ ). Модель имеет название «цветовой градиент» потому, что на шаге, моделирующем разделение компонент, используется т.н. «цветовой градиент», характеризующий направление наибольшей концентрации каждой из компонент. Используется сетка D3Q19.

Уравнение эволюции значений функции распределения для каждой из компонент аналогично однокомпонентному случаю:

$$|f^k(\mathbf{r} + \mathbf{e}\delta t, t + \delta t)\rangle = |f^k(\mathbf{r}, t)\rangle + |\Omega(\rho^k, \mathbf{u}^k, \cdot)\rangle \quad (3.8)$$

$$|\Omega(\rho^k, \mathbf{u}^k, \cdot)\rangle = |\Omega^3(\rho^k, \mathbf{u}^k, \cdot)\rangle (|\Omega^1(\rho^k, \mathbf{u}^k, \cdot)\rangle + |\Omega^2(\rho^k, \mathbf{u}^k, \cdot)\rangle) \quad (3.9)$$

где  $|f^k(\mathbf{r} + \mathbf{e}\delta t, t + \delta t)\rangle = (f_0^k(\mathbf{r} + \mathbf{e}_0\delta t, t + \delta t), \dots, f_{q-1}^k(\mathbf{r} + \mathbf{e}_{q-1}\delta t, t + \delta t))^T$ ,  $\mathbf{e}_i$  — дискретные направления скоростей,  $|\Omega^1\rangle$  — покомпонентный интеграл столкновений,  $|\Omega^2\rangle$  — «перемешивание»,  $|\Omega^3\rangle$  — «перекрашивание»,  $k$  — индекс жидкости («красная» или «синяя»).

Используется шаблон D3Q19. Порядок направлений в шаблоне (3.3) и весовые коэффициенты (3.4) задаются аналогично однокомпонентному течению.

Покомпонентный интеграл столкновений  $|\Omega^{1,k}\rangle$  записывается аналогично случаю однорелаксационного приближения [48]:

$$|\Omega^{1,k}\rangle = \omega^k (|f^{k,eq}\rangle - |f^k\rangle) \quad (3.10)$$

где  $\omega = \frac{1}{\tau}$  — эффективный параметр релаксации. Он варьируется для в граничном слое в случае жидкостей с разными вязкостями:

$$\omega = \begin{cases} \omega^r, & \text{если } \psi > \delta \\ f^r(\psi), & \text{если } \delta \geq \psi > 0 \\ f^b(\psi), & \text{если } 0 \geq \psi > -\delta \\ \omega^b, & \text{если } -\delta \geq \psi \end{cases} \quad (3.11)$$

где  $\omega^r$  и  $\omega^b$  — параметры релаксации красной и синей жидкости соответственно,  $\psi = \frac{\rho^r - \rho^b}{\rho^r + \rho^b}$  — параметр «наполнения» ячейки жидкостью одного из типов. Модуль параметра наполнения отличается от единицы только в приграничном слое. При этом:

$$\begin{aligned} f^r(\psi) &= \chi + \eta\psi + \kappa\psi^2 \\ f^b(\psi) &= \chi + \lambda\psi + \nu\psi^2 \\ \chi &= 2(\omega^r\omega^b)/(\omega^r + \omega^b) \\ \eta &= 2(\omega^r - \chi)/\delta \\ \kappa &= -\eta/(2\delta) \\ \lambda &= 2(\chi - \omega^b)/\delta \\ \nu &= \lambda/(2\delta) \end{aligned}$$

Равновесная функция распределения для каждого значения  $f_i^k$  также записывается аналогично однокомпонентному случаю:

$$f_i^{k,eq}(\rho_k, \mathbf{u}^k) = \rho^k \left( \phi_i^k + w_i \left( 3\mathbf{e}_i \mathbf{u}^k + \frac{9}{2}(\mathbf{e}_i \mathbf{u}^k)^2 - \frac{3}{2}(u^k)^2 \right) \right)$$

где  $\mathbf{u}^k$  — скорость,  $\rho^k$  — плотность  $k$ -ой жидкости,  $w_i$  — весовые коэффициенты,  $\phi_i^k$  — параметры плотности:

В приведенных выше формулах  $\delta$  — свободный параметр. В данной работе он выбран равным 0.001.

Оператор перемешивания  $|\Omega^{2,k}\rangle$  имеет вид [48]:

$$|\Omega^{2,k}\rangle = (\Omega_0^{2,k}, \dots, \Omega_{q-1}^{2,k})^T \quad (3.13)$$

$$\Omega_i^{2,k} = f_i^k + \frac{A^k}{2} |\mathbf{F}| \left( w_i \frac{(\mathbf{F}\mathbf{e}_i)^2}{F^2} - B_i \right) \quad (3.14)$$

где  $A_k$  и  $B_i$  — некоторые свободные параметры. Эти параметры подбираются таким образом, чтобы полученное в итоге уравнение соответствовало макроскопическим физическим уравнениям и варьируются у разных авторов. В данной работе используются:

$$A^r = A^b = 0.0001 \quad (3.15)$$

$$B = \begin{bmatrix} -\frac{1}{3} & \frac{1}{18} & \frac{1}{18} & \frac{1}{18} & \frac{1}{36} & \frac{1}{36} & \frac{1}{36} & \frac{1}{36} & \frac{1}{36} & \frac{1}{18} & \frac{1}{18} & \frac{1}{18} & \frac{1}{36} & \frac{1}{36} & \frac{1}{36} & \frac{1}{36} & \frac{1}{36} \end{bmatrix}^T$$

Цветовой градиент  $\mathbf{F}$  задается следующим соотношением [48]:

$$\mathbf{F} = 3 \sum_i w_i \mathbf{e}_i \frac{\rho^r(\mathbf{r} + \mathbf{e}_i) - \rho^b(\mathbf{r} + \mathbf{e}_i)}{\rho^r(\mathbf{r} + \mathbf{e}_i) + \rho^b(\mathbf{r} + \mathbf{e}_i)} \quad (3.16)$$

Оператор перекрашивания контролирует разделение двух компонент и отвечает за поверхностные явления  $|\Omega^{3,k}\rangle$  имеет вид [48]:

$$\Omega_i^{3,r} = \frac{\rho^r}{\rho} f_i + \beta \frac{\rho^r \rho^b}{\rho^2} \cos(\varphi_i) \sum_k f_i^{k,eq} \quad (3.17)$$

$$\Omega_i^{3,b} = \frac{\rho^b}{\rho} f_i - \beta \frac{\rho^r \rho^b}{\rho^2} \cos(\varphi_i) \sum_k f_i^{k,eq} \quad (3.18)$$

где  $f_i = f_i^r + f_i^b$ ,  $\rho = \rho^r + \rho^b$ ,  $\cos(\varphi_i) = \frac{\mathbf{e}_i \cdot \mathbf{F}}{|\mathbf{e}_i| |\mathbf{F}|}$ ,  $\beta = 0.99$  — параметр, контролирующий толщину слоя между компонентами.

### 3.1.3. Моделирование диффузии

Для моделирования диффузии используется подход, схожий с многорелаксационным интегралом столкновений [78]. Используется решетка скоростей D3Q7 (см. Рис. 1.4).

Уравнение эволюции значений функции распределения  $|f(\mathbf{r}, t)\rangle$  следующим образом [78]:

$$|f(\mathbf{r} + \mathbf{e}\delta t, t + \delta t)\rangle = |f(\mathbf{r}, t)\rangle - \mathbf{L}|f(\mathbf{r}, t)\rangle + \Theta|w\rangle \quad (3.19)$$

$$\mathbf{L} = \mathbf{M}^{-1}\mathbf{S}\mathbf{M}(|w\rangle + \mathbf{U}_e|w\rangle)\langle 1| - \mathbf{I} \quad (3.20)$$

$$\mathbf{U}_e = \text{diag}(\mathbf{u}\mathbf{e}_0, \dots, \mathbf{u}\mathbf{e}_{q-1}) \quad (3.21)$$

где  $|w\rangle$  — вектор весовых коэффициентов,  $\Theta$  — оператор учета внешних источников,  $\mathbf{S}$  — оператор столкновений,  $\langle 1| = (1, \dots, 1)$  — единичный вектор,  $\mathbf{I}$  — единичная матрица,  $\mathbf{u}$  — скорость среды,  $\mathbf{e}_i$  — шаблонные скорости,  $\langle 1| = (1, 1, 1, 1, 1, 1, 1)$  — единичный вектор.

При этом используются обозначения:

$$|f(\mathbf{r}, t)\rangle = (f_0(\mathbf{r}, t), \dots, f_{q-1}(\mathbf{r}, t))^T$$

$$|f(\mathbf{r} + \mathbf{e}, t + \delta t)\rangle = (f_0(\mathbf{r} + \mathbf{e}_0, t + \delta t), \dots, f_{q-1}(\mathbf{r} + \mathbf{e}_{q-1}, t + \delta t))^T$$

где  $|f(\mathbf{r} + \mathbf{e}, t + \delta t)\rangle$  — значения функции распределения после столкновения и распространения, векторы  $\mathbf{e}_i$  задают решетку направлений,  $|f(\mathbf{r}, t)\rangle$  — вектор значений функции распределения до шага столкновения.

Представленное выше уравнение эволюции соответствует уравнению диффузии адвекции:

$$\frac{\partial \phi}{\partial t} + \frac{\partial(\phi \nu_j)}{\partial x_j} = \frac{\partial}{\partial x_i} \left( D_{ij} \frac{\partial \phi}{\partial x_j} \right) + S \quad (3.22)$$

$$\phi(0, \mathbf{r}) = \phi(\mathbf{r}) \quad (3.23)$$

Тензор диффузии  $D_{ij}$  задается положительной симметричной матрицей, начальные значения  $\phi(\mathbf{r})$  известны. Выбран следующий порядок шаблонных скоростей  $\mathbf{e}_i$ :

$$\mathbf{e} = \begin{bmatrix} 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}^T \quad (3.24)$$

Вектор весовых коэффициентов  $|w\rangle$  с учетом выбранного порядка шаблонных скоростей:

$$|w\rangle = \left( \frac{1}{4}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8} \right)$$

Матрица перехода  $M$  задается как [78]:

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 6 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 2 & -1 & -1 & 2 & -1 & -1 \\ 0 & 0 & 1 & -1 & 0 & 1 & -1 \end{bmatrix} \quad (3.25)$$

Она соответствует следующему базису [78]:

$$\begin{aligned} |\phi_0\rangle_\alpha &= \|\mathbf{e}_\alpha\|^0 \\ |\phi_1\rangle_\alpha &= e_{\alpha_x} \\ |\phi_2\rangle_\alpha &= e_{\alpha_y} \\ |\phi_3\rangle_\alpha &= e_{\alpha_z} \\ |\phi_4\rangle_\alpha &= 6 - 7 \|\mathbf{e}_\alpha\|^2 \\ |\phi_5\rangle_\alpha &= 3 \|\mathbf{e}_x\|^2 - \|\mathbf{e}_\alpha\|^2 \\ |\phi_6\rangle_\alpha &= \|\mathbf{e}_y\|^2 - \|\mathbf{e}_z\|^2 \end{aligned}$$

Матрица столкновений включает в себя тензор диффузии  $D_{ij}$ :

$$S^{-1} = \begin{bmatrix} \tau_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \tau_{xx} & \tau_{xy} & \tau_{xz} & 0 & 0 & 0 \\ 0 & \tau_{xx} & \tau_{xy} & \tau_{xz} & 0 & 0 & 0 \\ 0 & \tau_{zx} & \tau_{zy} & \tau_{zz} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \tau_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \tau_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \tau_6 \end{bmatrix} \quad (3.26)$$

где  $\tau_{ij} = 0.5\delta_{ij} + D_{ij}$ ,

В случае изотропной диффузии матрица  $D_{ij}$  имеет диагональный вид:  $D_{ij} = \delta_{ij}$ . В данной работе используется анизотропный тензор диффузии:

$$\begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{bmatrix} = \begin{bmatrix} \frac{1}{10} & 0 & 0 \\ 0 & \frac{2}{5} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.27)$$

Параметры  $\tau_0$ ,  $\tau_4$ ,  $\tau_5$  и  $\tau_6$  не влияют на решение уравнение, только на величину ошибки [78], для простоты положим  $\tau_0 = \tau_4 = \tau_5 = \tau_6 = 1$ .

### 3.1.4. Моделирование теплопроводности

Для моделирования диффузии используется подход пассивного скаляра [34], построенный на основе РМБ с однорелаксационным интегралом столкновений [5]. В качестве шаблона скоростей взят D3Q19.

Уравнение эволюции «функции распределения температуры» в методе массивного скаляра задается аналогично однорелаксационному приближению интеграла столкновений:

$$|g(\mathbf{r},t)^*\rangle = |g(\mathbf{r},t)\rangle - \frac{1}{\tau_T} (|g(\mathbf{r},t)\rangle - |g^{eq}(\mathbf{r},t)\rangle) \quad (3.28)$$

$$|g(\mathbf{r} + \mathbf{e}, t + \delta t)\rangle = |g(\mathbf{r},t)^*\rangle \quad (3.29)$$

При этом компоненты  $g_i^{eq}$  равновесного вектора  $|g^{eq}(\mathbf{r},t)\rangle = (g_0^{eq}(\mathbf{r},t), \dots, g_{q-1}^{eq}(\mathbf{r},t))^T$  имеют вид:

$$g_i^{eq} = w_i T \left( 1 + 3\mathbf{e}_i \mathbf{u} + \frac{9}{2} (\mathbf{e}_i \mathbf{u})^2 - \frac{3}{2} u^2 \right) \quad (3.30)$$

Вектор весовых коэффициентов  $|w\rangle = (w_0, \dots, w_{q-1})^T$  задается аналогично случаю для однокомпонентного течения (3.4), как и  $|\mathbf{e}\rangle$  (3.3).

Температура рассчитывается как нулевой момент функции распределения ( $\langle 1 | = (1, \dots, 1)$  — единичный вектор):

$$T = \langle 1 | g \rangle \quad (3.31)$$

Коэффициент теплопроводности  $\kappa$  определяется на основе параметра релаксации:

$$\kappa = \frac{1}{3} (\tau - 0.5) \quad (3.32)$$

Данный прием позволяет получать численное решение уравнения теплопроводности, в том числе со сложными геометриями. Взаимодействие с гидродинамической моделью осуществляется через локальную скорость среды.

### 3.1.5. Границные условия

После шага распространения в ячейках, граничащих с непроницаемыми, оказываются неизвестны некоторые из значений функции распределения. Для их доопределения используются

краевые условия, наиболее известные из которых — краевые условия без проскальзывания [79]: нарешеточный отскок (on-grid bounceback), межрешеточный отскок (midgrid bounceback), различающиеся положением непроницаемой стенки. В первом случае стенка проходит через центр крайней непроницаемой ячейки, во втором — между проницаемой и непроницаемой ячейками. Универсальная «искривленная граница» (curved boundary) [42] обобщает оба случая и позволяет задавать любое положение стенки.

В данной работе используется межрешеточный отскок, т.к. он обладает вторым порядком точности по пространственным координатам [40] и не требует хранения дополнительной информации о граничных ячейках. В случае межрешеточного отскока неизвестные значения функции распределения  $f_i$  приравниваются «противоположным» значениям  $f_{-i}$  в данной ячейке до шага распространения. Таким образом,  $f_i = f_{-i}$  для всех неизвестных  $f_i$ , где противоположные направления обозначаются отрицательными индексами:  $\mathbf{e}_i = -\mathbf{e}_{-i}$ .

На границах вычислительного объема наиболее часто используются граничные условия с заданным давлением [56] или скоростью [31]. В реализованной модели используются граничные условия с заданным давлением, заданный градиент давления в дальнейшем используется для расчета абсолютной проницаемости.

Рассмотрим случай границ, перпендикулярных оси Z. В общем виде система уравнений выглядит следующим образом (поперечные скорости  $u_x = 0$  и  $u_y = 0$  на границе положим нулевыми, плотность  $\rho$ , связанная через уравнение состояния  $p = \frac{\rho}{3}$  с давлением  $p$ , задана):

$$\begin{aligned}
 0 &= \sum_{i=0}^{q-1} f_i e_{i_x} \\
 0 &= \sum_{i=0}^{q-1} f_i e_{i_y} \\
 \rho u_z &= \sum_{i=0}^{q-1} f_i e_{i_z} \\
 \rho &= \sum_{i=0}^{q-1} f_i \\
 f_3 - f_{10} &= \rho u_z / 3 \\
 f_6 - f_{15} &= \rho u_z / 6 + N \\
 f_7 - f_{16} &= \rho u_z / 6 - N \\
 f_8 - f_{17} &= \rho u_z / 6 + M \\
 f_9 - f_{18} &= \rho u_z / 6 - M
 \end{aligned} \tag{3.33}$$

где  $\mathbf{e}_i = (e_{i_x}, e_{i_y}, e_{i_z})$ ,  $N$  и  $M$  - поправочные слагаемые [56].

На нижней границе не определены значения  $f_3, f_6, f_7, f_8, f_9$ . На верхней —  $f_{12}, f_{15}, f_{16}, f_{17}, f_{18}$ .

Комбинируя уравнения для компоненты скорости  $u_z$  и плотности  $\rho$  на нижней границе  $z = 0$  получаем выражение для определения  $u_z$ :

$$u_z = 1 - \frac{1}{\rho}(f_0 + 2(f_{12} + f_{15} + f_{16} + f_{17} + f_{18}) + f_1 + f_2 + f_4 + f_5 + f_{11} + f_{12} + f_{13} + f_{14}) \quad (3.34)$$

Аналогично для верхней границы:

$$u_z = \frac{1}{\rho}(f_0 + 2(f_3 + f_6 + f_7 + f_8 + f_9) + f_1 + f_2 + f_4 + f_5 + f_{11} + f_{12} + f_{13} + f_{14}) - 1 \quad (3.35)$$

В случае граничных условий с заданной скоростью  $u_z$  на границах считается заданной, дальнейшие рассуждения остаются без изменений.

Решая оставшуюся систему уравнений,

$$\begin{aligned} 0 &= \sum_{i=0}^{q-1} f_i e_{i_x} \\ 0 &= \sum_{i=0}^{q-1} f_i e_{i_y} \\ f_3 - f_{10} &= \rho u_z / 3 \\ f_6 - f_{15} &= \rho u_z / 6 + N \\ f_7 - f_{16} &= \rho u_z / 6 - N \\ f_8 - f_{17} &= \rho u_z / 6 + M \\ f_9 - f_{18} &= \rho u_z / 6 - M \end{aligned} \quad (3.36)$$

получим неизвестные значения функции распределения на границах. Для нижней границы получаем:

$$\begin{aligned} f_3 &= f_{12} + \rho u_z / 3 \\ f_6 &= (f_{10} + f_{13} - f_{14} - f_1 - f_4 + f_5) / 2 + f_{15} + \rho u_z / 6 \\ f_7 &= (-f_{10} - f_{13} + f_{14} + f_1 + f_4 - f_5) / 2 + f_{16} + \rho u_z / 6 \\ f_8 &= (f_{11} + f_{13} + f_{14} - f_2 - f_4 - f_5) / 2 + f_{17} + \rho u_z / 6 \\ f_9 &= (-f_{11} - f_{13} - f_{14} + f_2 + f_4 + f_5) / 2 + f_{18} + \rho u_z / 6 \end{aligned} \quad (3.37)$$

Аналогично для верхней границы:

Таблица 3.1: Численные параметры модели механики

Параметр	Символ	Значение
коэффициент упругости	$k$	0.1
искусственная вязкость	$\eta$	0.05
начальный размер упаковки	$B_x \times B_y \times B_z$	$600 \times 600 \times 600$
начальная пористость	—	33.3 %

$$\begin{aligned}
 f_{12} &= f_3 - \rho u_z / 3 \\
 f_{15} &= (-f_{10} - f_{13} + f_{14} + f_1 + f_4 - f_5) / 2 + f_6 - \rho u_z / 6 \\
 f_{16} &= (f_{10} + f_{13} - f_{14} - f_1 - f_4 + f_5) / 2 + f_7 - \rho u_z / 6 \\
 f_{17} &= (-f_{11} - f_{13} - f_{14} + f_2 + f_4 + f_5) / 2 + f_8 - \rho u_z / 6 \\
 f_{18} &= (f_{11} + f_{13} + f_{14} - f_2 - f_4 - f_5) / 2 + f_9 - \rho u_z / 6
 \end{aligned} \tag{3.38}$$

Для моделей диффузии и теплопроводности на границах расчетного объема использовались периодические граничные условия.

### 3.2. Метод дискретных элементов

Метод дискретных элементов применяется в данной работе для создания искусственных сред с точно известной структурой. В частности, для моделирования поведения упаковок проппанта с хорошей сферичностью и упругостью. Для упрощения модели не рассматривается разрушение зерен при нагрузке, а взаимодействие зерен описывается линейным законом упругости. Проппант — гранулярный материал, часто песок, используемый в нефтедобывающей промышленности в качестве расклинивающего агента для удержания трещин при гидроразрыве пласта. В данной работе рассматривается специальный класс полимерного проппанта, отличающегося от остальных сферической формой и устойчивостью к разрушению.

Проппант моделируется в виде квазиэластичных шаров массы  $m_i$  с заданным диаметром  $D_i$  и начальной позицией  $r_i$ . Вычислительный объем имеет размеры  $B_x \times B_y \times B_z$  (см. Рис. 3.1), на границах установлены упругие стенки. Параметры модели представлены в Таб. 3.1. Сжатие проппантной упаковки моделируется постепенным изменением размера упаковки вдоль оси X.

Зерна взаимодействуют друг с другом и со стенками. Два зерна взаимодействуют тогда и только тогда, когда расстояние между их центрами меньше суммы радиусов:  $|\mathbf{r}_n - \mathbf{r}_m| < R_n + R_m$ . В таком случае сила взаимодействия между зернами  $n$  и  $m$  задается уравнением:

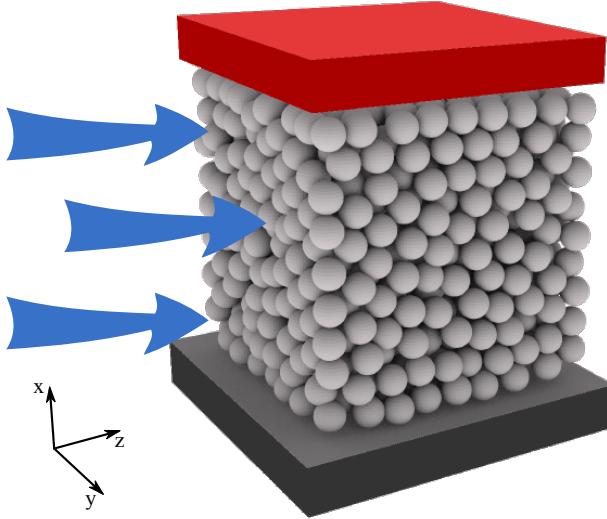


Рисунок 3.1: Пример упаковки шаров одинакового радиуса. Стрелками показано направление течения. Для наглядности на схеме не показаны боковые стенки.

$$\begin{aligned}\mathbf{F}_n &= k(R_n + R_m - |\mathbf{r}_n - \mathbf{r}_m|) \frac{\mathbf{r}_n - \mathbf{r}_m}{|\mathbf{r}_n - \mathbf{r}_m|} - \eta \mathbf{v}_n \\ \mathbf{F}_m &= k(R_n + R_m - |\mathbf{r}_n - \mathbf{r}_m|) \frac{\mathbf{r}_m - \mathbf{r}_n}{|\mathbf{r}_n - \mathbf{r}_m|} - \eta \mathbf{v}_m\end{aligned}\quad (3.39)$$

где  $\eta$  — свободный параметр (искусственная вязкость), гарантирующий установление положений центров шаров,  $\mathbf{v}$  — скорости центров зерен, а параметр упругости  $k$  определяется из лабораторных данных. В данной работе используются значения  $k = 0.1$  и  $\eta = 0.05$ .

Параметр  $k$  подбирается таким образом, чтобы лабораторная кривая нагрузки проппантной упаковки соответствовала результатам численного эксперимента. Эффективная нагрузка на проппант рассчитывается как:

$$P_x = \sum_i^N (\mathbf{F}_i, \mathbf{x}) \quad (3.40)$$

где  $N$  — общее число зерен проппанта,  $\mathbf{x} = (1, 0, 0)^T$ .

### 3.3. Оценка производительности программного комплекса

Для увеличения скорости вычислений все расчеты в программном комплексе реализованы на видеокартах (GPU) с помощью технологии CUDA, требуется Compute Capability не ниже 2.0. Одна итерация расчетов состоит из трех шагов: столкновение, распространение, применение граничных условий. На краях проницаемых областей используется граничное условие без проскальзывания: межрешеточный отскок.

В шаблоне D3Q19 используется 19 дискретных направлений скоростей, в схеме D3Q7 — семь дискретных направлений. В случае использования вещественных чисел одинарной точности на одну расчетную ячейку требуется 1 байт для хранения типа (проницаемая или непроницаемая), 76 или 28 байт для хранения значений функции распределения для шаблонов D3Q19 и D3Q7 соответственно. В случае двух компонент объем памяти, необходимый для хранения значений функции распределения, удваивается. Максимальный размер DRAM памяти на GPU сейчас достигает 24 Гб, но такие устройства еще массово не распространены. На суперкомпьютере «Ломоносов» установлены вычислители с 6 Гб, что достаточно для моделирования объема около  $340^3$  в случае простой реализации с дублированием массивов на шаге распространения. Этого недостаточно для решения множества реальных задач.

Для оптимизации потребления памяти было использовано два приема: обменный алгоритм [74] и хранение значений функции распределения только в проницаемых ячейках. Обменный алгоритм позволяет избежать дублирования массива на шаге распространения, а хранение значений функций распределения только в проницаемых ячейках сокращает размеры массивов в случае объемов с малой пористостью. Наконец, для массивно-параллельных вычислений используется блочная декомпозиция объема, при которой отдельные фрагменты области рассчитываются на различных видеокартах в отдельных процессах.

Суть обменного алгоритма заключается в записи значений функции распределения после шага столкновения в обратном порядке и попарный обмен этими значениями между соседними ячейками на шаге распространения (Рис. 1.6). Важно, что обмен осуществляется параллельно независимо для каждой пары направлений в соседних ячейках.

Для хранения значений функции распределения только в проницаемых ячейках в момент загрузки программы все проницаемые ячейки нумеруются, также сохраняется карта соответствия  $\mathfrak{M} : i \rightarrow (x_i, y_i, z_i)$ . Составляется редуцированный массив значений  $f_i$  (и, если требуется, плотностей  $\rho$ ) только в проницаемых ячейках. После этого в вычислениях используется редуцированный массив, для вычисления соседних ячеек используется карта  $\mathfrak{M}$ .

Использование обменного алгоритма за счет исключения дополнительного массива сокращает потребление памяти практически вдвое, с  $(156 + 1)$  до  $(76 + 1)$  байт на одну ячейку для шаблона D3Q19 в случае использования чисел одинарной точности для хранения значений функции распределения (1 байт для хранения статуса ячейки: проницаемая или непроницаемая). Выигрыш от хранения значений функции распределения только в проницаемых ячейках зависит от доли проницаемых ячеек (то есть от пористости  $\varphi$ ):  $((76 + 3 \cdot 4) \cdot \varphi + 4 + 1)$ , где  $(3 \cdot 4 \cdot \varphi + 4)$  байт используется для хранения матрицы соответствия между полным и редуцированным массивами значений функции распределения, а сами значения хранятся только в проницаемых ячейках:  $76 \cdot \varphi$ . Хранение значений только в проницаемых ячейках имеет смысл в основном на небольших относительных пористостях (Рис. 3.2).

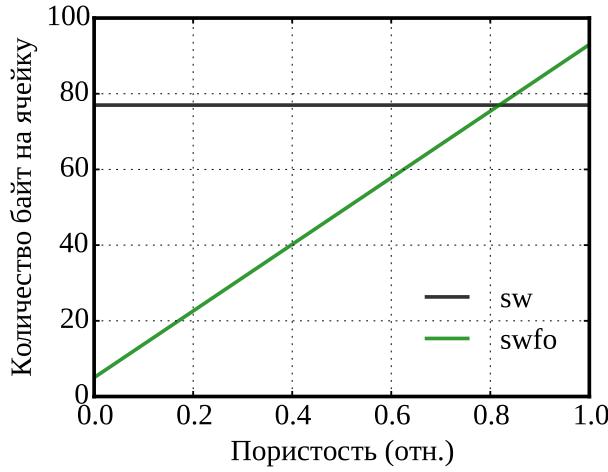


Рисунок 3.2: Требуемое количество байт на ячейку в случае использования обменного алгоритма без оптимизации хранения значений функции распределения только в проницаемых ячейках (sw) и с применением этой оптимизации (swfo).

При блочном разбиении расчетная область разделяется на равные слои перпендикулярно оси Z (Рис. 3.3). Размер фрагментов (и, соответственно, количество GPU) подбираются на основе минимума по максимально доступным объемам памяти на каждом из массива ускорителей. Каждый процесс на каждой итерации на шаге распространения синхронизирует значения функции распределения в граничных ячейках с соседними процессами. Этапы столкновения и применения граничных условий рассчитываются независимо на каждом узле. Для синхронизации данных между процессами в случае блочной декомпозиции вычислительного объема в данной работе используется MPI. Подобный подход используется в [35, 57, 58, 76].

Было произведена серия измерений производительности вычислительного модуля для расчетов однокомпонентного течения в зависимости от числа задействованных видеокарт. Вплоть до 45 GPU на суперкомпьютере «Ломоносов» виден линейный рост количества рассчитанных ячеек в единицу времени (Рис. 3.4а) вплоть до 10 видеокарт, далее наблюдается загиб тренда из-за задержек при пересылке данных между узлами. Здесь в качестве расчетного объема использовался пустой куб с размерами  $450^3$  ячеек.

Те же измерения производительности расчета однокомпонентного течения от числа задействованных видеокарт были произведены при расчете образца LV60A из коллекции Imperial College, результаты представлены на Рис. 3.4б. По сравнению с предыдущим случаем, в данном случае меньше количество проницаемых ячеек и, соответственно, большая производительность самого модуля.

В целом, зависимость производительности от числа использованных графических ускорителей контролируется двумя факторами: максимальным по всем задействованным графическим ускорителям временем расчета на одной видеокарте и задержками на пересылку данных по MPI.

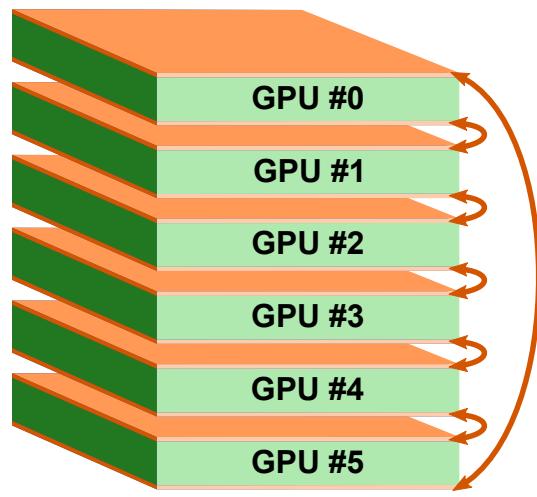
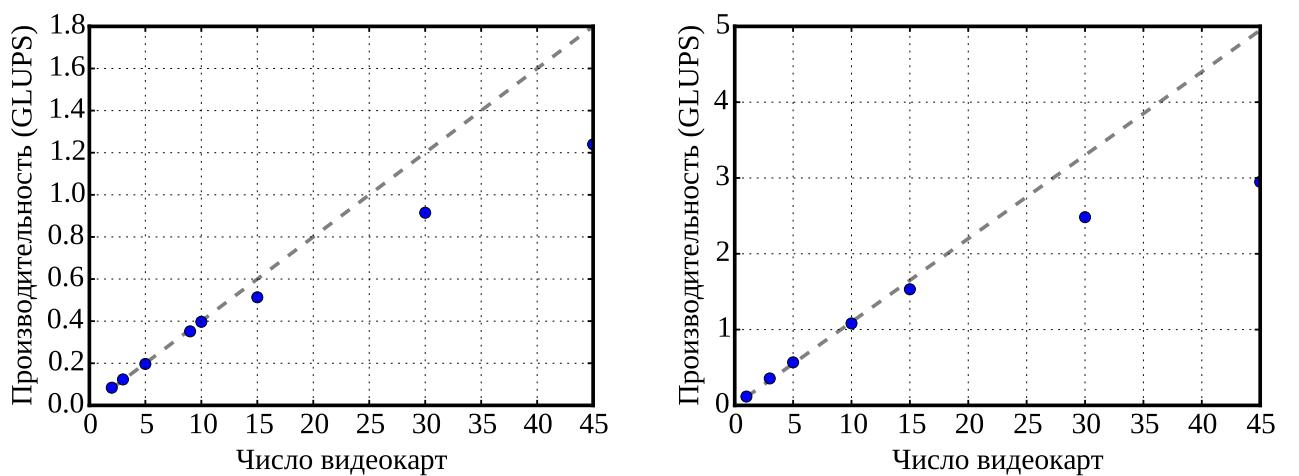


Рисунок 3.3: Одномерное разбиение области интереса на слои и распределение по нескольким процессам. Обмен значениями функций распределения на границах между соседними фрагментами по MPI показан стрелками.



(а) Пустой расчетный объем (состоит только из проницаемых ячеек).

(б) Образец LV60A из коллекции Imperial College.

Рисунок 3.4: Зависимость производительности расчетного модуля однокомпонентного течения от числа видеокарт.

Из-за того, что данными обмениваются только «соседние» процессы, можно считать, что задержки на пересылку данных не меняются при увеличении числа GPU. С другой стороны, чем выше минимальная производительность GPU (то есть чем меньше время расчета шага на одном GPU), тем сильнее оказывается фактор задержек на пересылку данных. Поэтому, при фиксированном суммарном расчетном объеме при увеличении числа задействованных графических ускорителей в результате геометрической декомпозиции на каждый из ускорителей приходится меньший объем вычислений, время расчета столкновения и распространения внутри GPU уменьшается, задержки на пересылку данных становятся заметными (как это видно на Рис. 3.4a и Рис. 3.4b). В вырожденном случае, когда на один графический ускоритель уже приходится недостаточное для его полной загрузки количество ячеек, производительность перестает расти при дополнительном увеличении числа GPU. В невырожденном случае зависимость производительности  $P$  от числа видеокарт  $N_{GPU}$  выражается следующим соотношением:

$$P = \frac{A}{\frac{B}{N_{GPU}} + C} \quad (3.41)$$

где  $A$  — нормировочный коэффициент, параметр  $B$  характеризует время расчета на одной видеокарте, параметр  $C$  характеризует задержки на пересылку данных по сети посредством MPI.

Для оценки ускорения расчетов на GPU по сравнению с CPU была реализована отдельная версия расчетного модуля для моделирования однокомпонентного течения со всеми вышеперечисленными оптимизациями потребления памяти, но выполняющая весь цикл вычислений на CPU. Ускорение массивно-параллельной реализации на GPU по сравнению с однопоточной реализацией на CPU составило 20 раз (расчетный объем составил  $450 \times 450 \times 400$  вокселей, относительная доля проницаемых ячеек 30%). По сравнению с многопоточной реализацией на CPU с использованием OpenMP (Open Multi-Processing) ускорение расчетов на GPU составило 5 раз на том же объеме. Сравнительные эксперименты производились на рабочей станции с GPU GeForce GTX 770 Ti и центральным процессором Intel Core i5-3330.

Реализованный программный комплекс может работать на любом суперкомпьютере с гибридной архитектурой без дополнительных изменений исходного кода. Для компиляции и исполнения на суперкомпьютере должны быть установлены только CUDA и MPI.

### 3.4. Верификация программного модуля

Для проверки корректности работы реализованного программного комплекса была проведена серия тестовых экспериментов, результаты которых были проверены на основе известных точных аналитических зависимостей.

Таблица 3.2: Сводная таблица параметров вычислительного эксперимента по моделированию течения Пуазейля

Параметр	Символ	Значение
Размер области	$N_x \times N_y \times N_z$	$300 \times 300 \times 300$
Параметр релаксации	$\nu$	1.0
Радиусы трубы	$R$	40, 60, 80, 100
Длина трубы	$L$	$N_z = 300$
Градиент давления	$\Delta p$	0.006

### 3.4.1. Течение Пуазейля

Тест предназначен для расчетного модуля однокомпонентного течения. Размер расчетной области равен  $N_x \times N_y \times N_z = 300 \times 300 \times 300$ . Цилиндрическая труба задается в соответствии с правилом: если координаты центра  $i$ -ой ячейки  $(x_i, y_i, z_i)$  удовлетворяют условию  $(x_i - \frac{N_x}{2})^2 + (y_i - \frac{N_y}{2})^2 < R^2$ , то ячейка является проницаемой, иначе ячейка считается непроницаемой.

В качестве краевого условия на границах с непроницаемыми ячейками используется межрешеточный отскок. Вязкость жидкости единичная  $\nu = 1.0$ . На гранях, перпендикулярных оси  $Z$ , использованы граничные условия с фиксированным давлением, на остальных гранях заданы периодические граничные условия. Течение направлено вдоль оси  $Z$ .

В случае течения Пуазейля модуль скорости среды в попечном сечении в зависимости от расстояния до центра трубы образует параболический профиль:

$$v(r) = \frac{\Delta p}{4\eta L} (R^2 - r^2)$$

где  $\eta$  — динамическая вязкость, которая связана с кинематической вязкостью соотношением  $\eta = \rho\nu$ ,  $\Delta p$  — градиент давления.

Таким образом, в случае кубической сетки уравнение имеет вид:

$$v(r) = \frac{\Delta p}{4\eta L} \left( R^2 - \left( \left( x_i - \frac{N_x}{2} \right)^2 + \left( y_i - \frac{N_y}{2} \right)^2 \right) \right)$$

где  $\Delta p = p_0 - p_{N_z-1}$ .

Проведены эксперименты для нескольких значений радиуса трубы  $R = 40, 60, 80, 100$ . Длина трубы  $L$  во всех экспериментах равна длине всего образца и составляет  $N_z = 300$  пикселей. Количество итераций во всех экспериментах равно и составляет 50 тысяч.

Полученный в численном эксперименте профиль скорости демонстрирует отличное соответствие точному решению (см. Рис. 3.5). Малые значения диаметров труб не рассматривались из-

за ошибки, вносимой дискретным представлением стенки трубы в случае накладывания ее на трехмерную сетку.

Данный численный эксперимент позволяет верифицировать поведение вязкой жидкости и корректность сшивки данных в случае геометрической декомпозиции объема и произведения вычислений на нескольких видеокартах с синхронизацией по MPI.

### 3.4.2. Абсолютная проницаемость цилиндрической трубы

Тест предназначен для расчетного модуля однокомпонентного течения. Размер расчетной области равен  $N_x \times N_y \times N_z = 300 \times 300 \times 300$ . Цилиндрическая труба задается в соответствии с правилом: если координаты центра  $i$ -ой ячейки  $(x_i, y_i, z_i)$  удовлетворяют условию  $(x_i - \frac{N_x}{2})^2 + (y_i - \frac{N_y}{2})^2 < R^2$ , то ячейка является проницаемой, иначе ячейка считается непроницаемой.

В качестве краевого условия на границах с непроницаемыми ячейками используется межрешеточный отскок. Вязкость жидкости единичная  $\nu = 1.0$ . На гранях, перпендикулярных оси  $Z$ , использованы граничные условия с фиксированным давлением, на остальных гранях заданы периодические граничные условия. Течение направлено вдоль оси  $Z$ .

Согласно закону Дарси, абсолютная проницаемость  $k_p$  для медленных течений может быть получена из формулы:

$$Q = -\frac{k_p A}{\mu} \frac{\Delta p}{L} \quad (3.42)$$

где  $A = N_x \cdot N_y$  —поперечное сечение течения,  $\Delta p$  — разница давлений на верхней и нижней границе (течение вдоль оси  $Z$ ),  $\mu = \rho\nu$  — динамическая вязкость,  $L = N_z$  — длина образца

С другой стороны, из закона Пуазейля известно течение через цилиндрическую трубу:

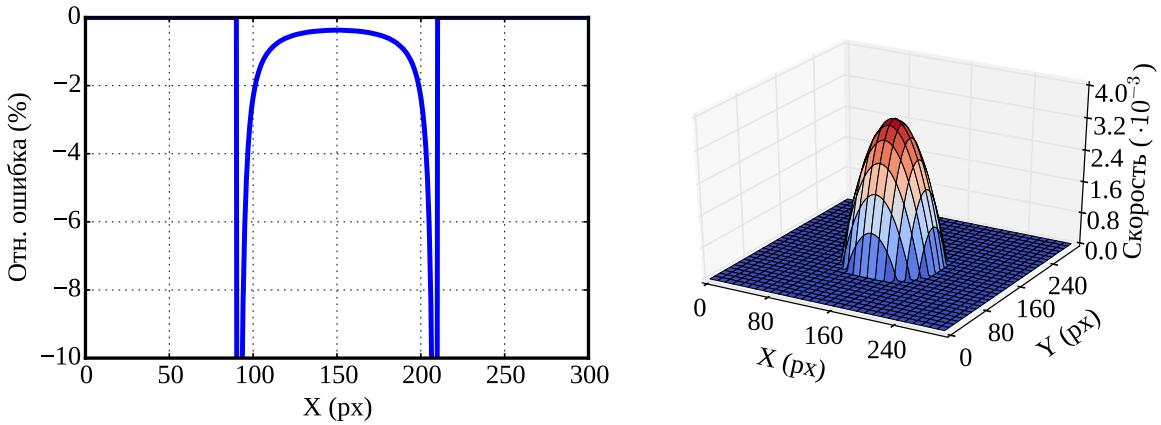
$$Q = -\frac{\pi R^4}{8\mu} \frac{\Delta p}{L} \quad (3.43)$$

Из уравнений (3.42) и (3.43) следует, что:

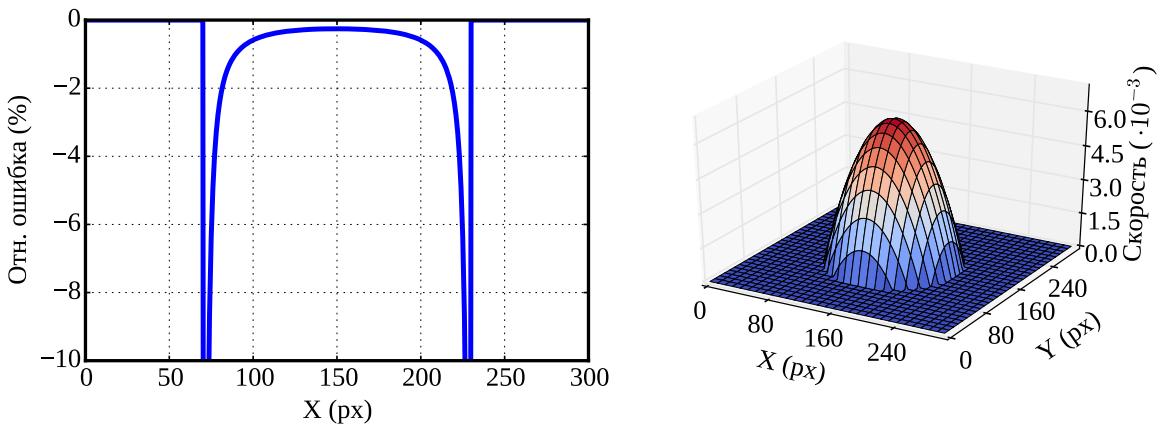
$$k_p = \frac{\pi R^4}{8A} \quad (3.44)$$

В численном эксперименте были посчитаны абсолютные проницаемости для 11 значений радиусов труб (Таб. 3.3). Результаты представлены в Таб. 3.4 и на Рис. 3.6 и демонстрируют хорошее соответствие точным решениям.

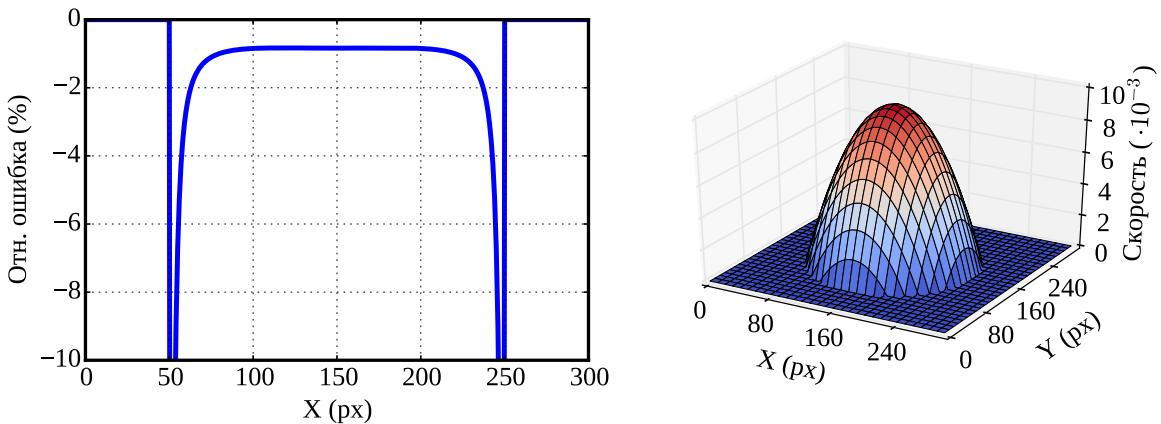
Данный численный эксперимент позволяет верифицировать процедуру расчета абсолютной проницаемости и корректность сшивки данных в случае геометрической декомпозиции объема



(a) профиль скорости (слева) и сравнение с точным решением (справа) для  $R = 60$



(b) профиль скорости (слева) и сравнение с точным решением (справа) для  $R = 80$



(c) профиль скорости (слева) и сравнение с точным решением (справа) для  $R = 100$

Рисунок 3.5: Полученные в ходе численного моделирования поперечные профили скорости для различных радиусов цилиндрической трубы и их относительное отклонение от точного решения для течения Пуазейля (максимальная ошибка на границах трубы составляет во всех случаях 45%, ось ординат на графиках ошибки ограничена для наглядности).

Таблица 3.3: Сводная таблица параметров вычислительного эксперимента по определению абсолютной проницаемости цилиндрической трубы

Параметр	Символ	Значение
Размер области	$N_x \times N_y \times N_z$	$300 \times 300 \times 300$
Параметр релаксации	$\nu$	1.0
Радиусы трубы	$R$	10, 10, ..., 100, 140
Длина трубы	$L$	$N_z = 300$
Градиент давления	$\Delta p$	0.006

Таблица 3.4: Сводная таблица результатов вычислительного эксперимента по определению абсолютной проницаемости цилиндрической трубы

Радиус	Аналитическая $k_p$ (мД)	Численная $k_p$ (мД)	Относительная ошибка (%)
10	44	40	-9.00
20	698	672	-3.68
30	3534	3450	-2.39
40	11170	11029	-1.27
50	27271	26921	-1.28
60	56549	55987	-0.99
70	104763	103965	-0.76
80	178722	177454	-0.71
90	286278	284761	-0.53
100	436332	431970	-1.00
140	1676214	1660000	-0.97

и произведения вычислений на нескольких видеокартах с синхронизацией по MPI: при любом количестве используемых видеокарт результат должен оставаться одним и тем же.

### 3.4.3. Лобовое сопротивление шара

Тест предназначен для расчетного модуля однокомпонентного течения. Размер расчетной области равен  $N_x \times N_y \times N_z = 300 \times 300 \times 300$ . Ячейки, составляющие непроницаемый шар, задаются согласно правилу  $(x_i - \frac{N_x}{2})^2 + (y_i - \frac{N_y}{2})^2 + (z_i - \frac{N_z}{2})^2 > R^2$ . Остальные ячейки считаются проницаемыми. Аналогично тесту с течением Пуазейля, здесь  $x_i$ ,  $y_i$  и  $z_i$  — координаты центра ячейки, а  $R$  — радиус шара. В эксперименте использованы радиусы шаров  $R = 30, 50, 70$ .

На гранях, перпендикулярных оси  $Z$ , используются граничные условия в виде заданной скорости течения. Причем при  $z = 0$  и  $z = N_z$  скорости равны и составляли  $u_z = 0.125 \cdot 10^{-3}, 0.25 \cdot 10^{-3}, 0.5 \cdot 10^{-3}, 1 \cdot 10^{-3}, 2 \cdot 10^{-3}, 3 \cdot 10^{-3}, \dots, 8 \cdot 10^{-3}$ , причем  $u_x = u_y = 0$ . На гранях, перпендику-

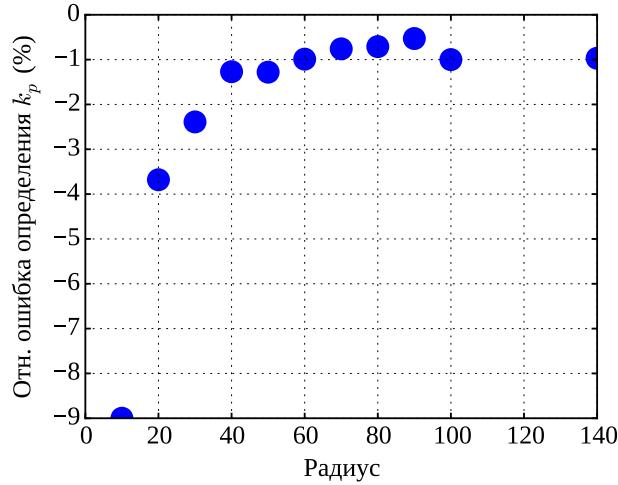


Рисунок 3.6: Относительная ошибка определения абсолютной проницаемости цилиндрической трубы для разных значений радиуса  $R$ .

Таблица 3.5: Сводная таблица параметров вычислительного эксперимента по расчету лобового сопротивления шара

Параметр	Символ	Значение
Размер области	$N_x \times N_y \times N_z$	$300 \times 300 \times 300$
Параметр релаксации	$\nu$	1.0
Радиус трубы	$R$	10, 90, 150
Градиент давления	$\Delta p$	0.006

лярных осях  $X$  и  $Y$  заданы периодические граничные условия. Для расчета чисел Рейнольдса используется соотношение  $Re = \frac{2Ru_x}{\nu}$ .

Согласно известной из теории Стокса аналитической зависимости лобового сопротивления от числа Рейнольдса:

$$C_d = \frac{24}{Re}$$

Полученная зависимость коэффициента лобового сопротивления шара от числе Рейнольдса действительно имеет вид  $C_d \sim \frac{1}{Re}$  (см. Рис. 3.7). Максимальная отклонение от теоретической зависимости составило около 5%.

#### 3.4.4. Тест Лапласа

Тест предназначен для расчетного модуля двухкомпонентного течения. Размер расчетной области равен  $N_x \times N_y \times N_z = 300 \times 300 \times 300$ . На всех гранях заданы периодические граничные условия. В начальный момент времени шар радиуса  $R$  красной жидкости (индекс  $r$ )

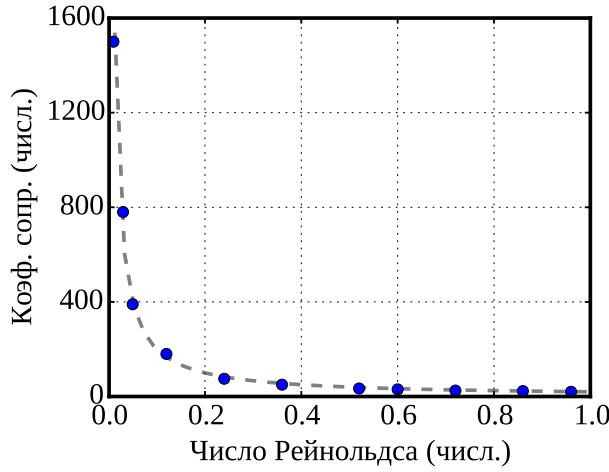


Рисунок 3.7: Сравнение аналитической зависимости значений коэффициента лобового сопротивления сопротивления шара для различных чисел Рейнольдса (линия) и результатов численного моделирования (точки).

Таблица 3.6: Сводная таблица параметров вычислительного эксперимента по расчету разницы давлений снаружи и внутри шаровой капли (тест Лапласа).

Параметр	Символ	Значение
Размер области	$N_x \times N_y \times N_z$	$300 \times 300 \times 300$
Параметр релаксации	$\tau^r, \tau^b$	1.0, 1.0
Поверхностное натяжение	$A^r, A^b$	0.01, 0.01
Параметры соотношения плотностей	$\alpha^r, \alpha^b$	0.2, 0.19
Начальный радиус капли	$R$	20, 30, ..., 100
Коэффициент разделения компонент	$\beta$	0.99

плавает внутри пространства синей жидкости (ее параметры обозначены с индексом  $b$ ). Соотношение плотностей задано через параметры  $\alpha^r = 0.2$ ,  $\alpha^b = 0.19$ , поверхностное натяжение  $A^r = A^b = 0.01$ , коэффициент разделения компонент  $\beta = 0.99$ . Параметры релаксации обеих жидкостей — единичные,  $\tau^r = \tau^b = 1.0$ .

Закон Юнга-Лапласа предсказывает обратную зависимость между разницей давлений внутри и вне шаровой капли для разных радиусов капель из-за силы поверхностного натяжения:

$$\Delta p = \frac{\sigma}{R} \quad (3.45)$$

где  $\sigma$  — коэффициент поверхностного натяжения.

Полученный в численном эксперименте зависимость разности давлений снаружи и внутри шара демонстрирует отличное соответствие предсказанной аналитически зависимости и действительно имеет вид  $\Delta p \sim \frac{1}{R}$  (см. Рис. 3.8). На рисунке показаны результаты во внутренних безраз-

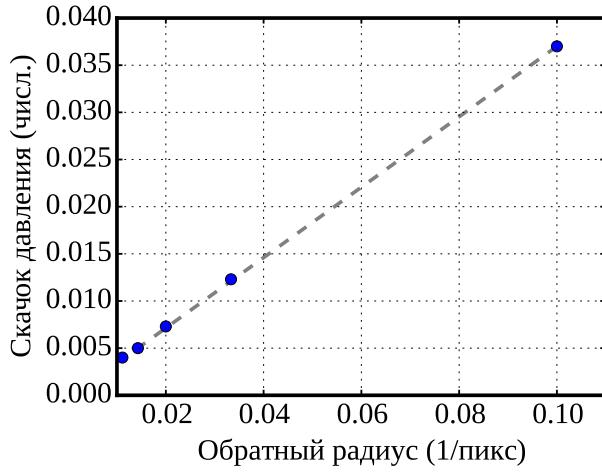


Рисунок 3.8: Зависимость разности давлений от радиуса.

мерных единицах модели. При пересчете в реальные физические величины необходимо учесть «разрешение» образа на пиксель.

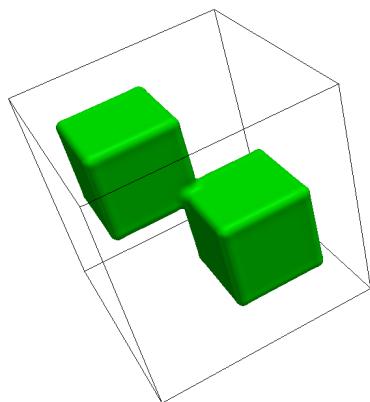
### 3.4.5. Коалесценция двух капель

Тест предназначен для расчетного модуля двухкомпонентного течения. Размер расчетной области равен  $N_x \times N_y \times N_z = 160 \times 160 \times 160$ . На всех гранях заданы периодические граничные условия. Соотношение плотностей задано через параметры  $\alpha^r = 0.001$ ,  $\alpha^b = 0.001$ , поверхностное натяжение  $A^r = A^b = 0.001$ . Параметры релаксации обеих жидкостей единичные,  $\tau^r = \tau^b = 1.0$ . В начальный момент времени каждая капля имеет кубическую форму со стороной 64 пикселя и касается второй капли углом. По всем граням вычислительного объема заданы периодические граничные условия.

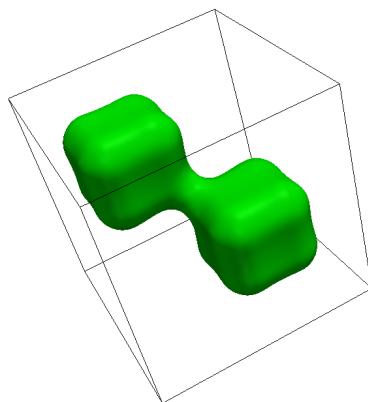
Также был проведен эксперимент по склеиванию под действием сил поверхностного натяжения двух кубических капель, касающихся в начальный момент времени углами (Рис. 3.9). В данном teste опосредованно проверяется корректность сшивки результатов при расчете на нескольких видеокартах, действие сил поверхностного натяжения и изотропность выбранного при моделировании цветового градиента: в состоянии покоя результирующая капля принимает сферическую форму. При выбранных в модели значениях параметров не возникает разрушения капли. Тем не менее, оно возможно при выборе других значений параметров модели.

### 3.4.6. Тест смачивания

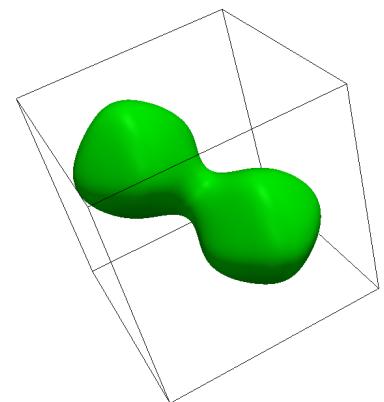
Тест предназначен для расчетного модуля двухкомпонентного течения. Размер расчетной области равен  $N_x \times N_y \times N_z = 256 \times 256 \times 64$ . На всех гранях заданы периодические граничные условия. На границах раздела проницаемых и непроницаемых областей в качестве краевого



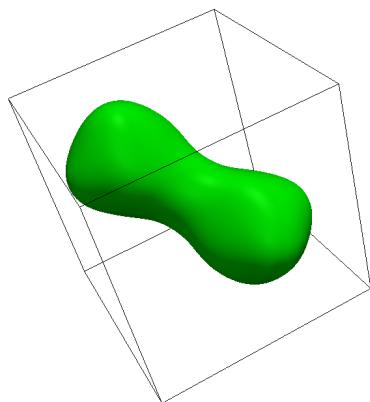
(a) Форма капель при  $t = 10$



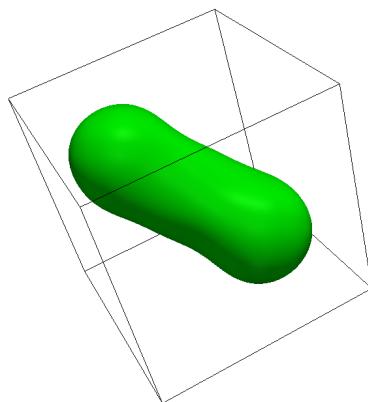
(b) Форма капель при  $t = 100$



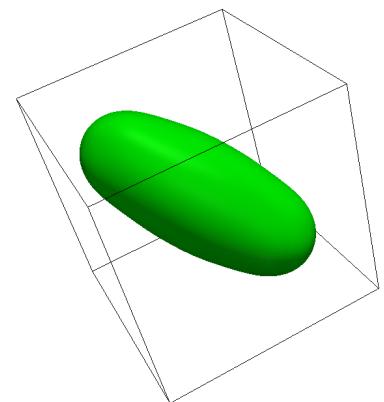
(c) Форма капель при  $t = 300$



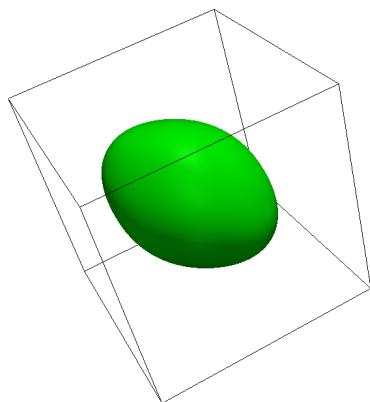
(d) Форма капель при  $t = 500$



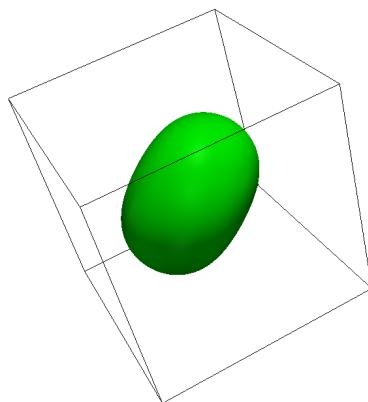
(e) Форма капель при  $t = 750$



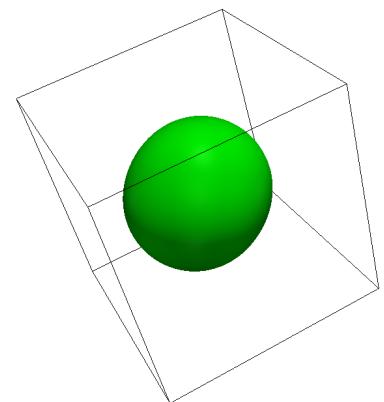
(f) Форма капель при  $t = 1000$



(g) Форма капель при  $t = 1500$



(h) Форма капель при  $t = 2000$



(i) Форма капель при  $t = 3000$

Рисунок 3.9: Склейивание двух кубических капель по действием силы поверхностного натяжения. В начальный момент времени каждая капля имеет кубическую форму и касается второй капли.

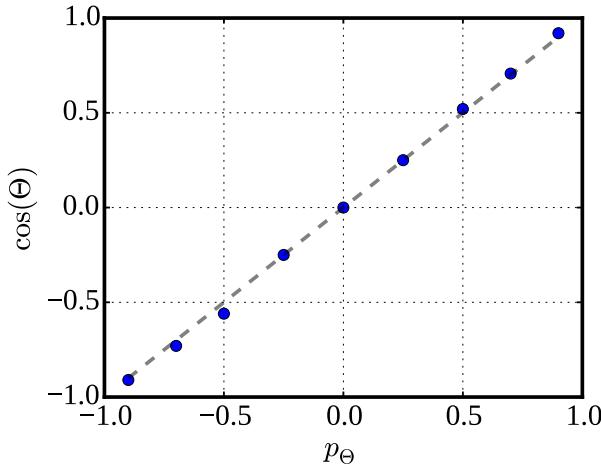


Рисунок 3.10: Зависимость косинуса угла смачивания от  $p_\Theta$ . Небольшие отклонения от линейной зависимости при вычислении угла смачивания связаны в том числе с дискретностью сетки.

условия используется межрешеточный отскок. Соотношение плотностей задано через параметры  $\alpha^r = 0.001$ ,  $\alpha^b = 0.001$ , поверхностное натяжение  $A^r = A^b = 0.001$ . Параметры релаксации обеих жидкостей единичные,  $\tau^r = \tau^b = 1.0$ . В начальный момент времени капля имеет кубическую форму со стороной 32 пикселя и лежит на непроницаемой подложке. По всем граням вычислительного объема заданы периодические граничные условия.

Проверяется корректность задания угла смачивания. В методе цветового градиента смачивание учитывается с помощью параметра  $p_\Theta = \cos(\Theta)$ . Он управляет дополнительной силой, действующей в граничных с непроницаемыми ячейками:

$$\mathbf{F} = \sum_q p^r c_q D_q \rho_i^r - \sum_q p^b c_q D_q \rho_i^b \quad (3.46)$$

В эксперименте задаются различные значения параметра  $p_\Theta$ , затем рассматривается фактический угол смачивания в установившемся режиме.

Полученные в результате численного эксперимента значения углов смачивания соответствуют заданным 3.10. Действительно, полученные фактические значения углов смачивания в целом равны заданным в модели. Отклонения в точках  $\pm 0.5$  вызваны неточностью определения фактического угла смачивания из-за пикселизованных образов поля плотности.

Для иллюстрации (Рис. 3.11) также приведены формы капель на плоской поверхности для различных значений коэффициентов смачивания.

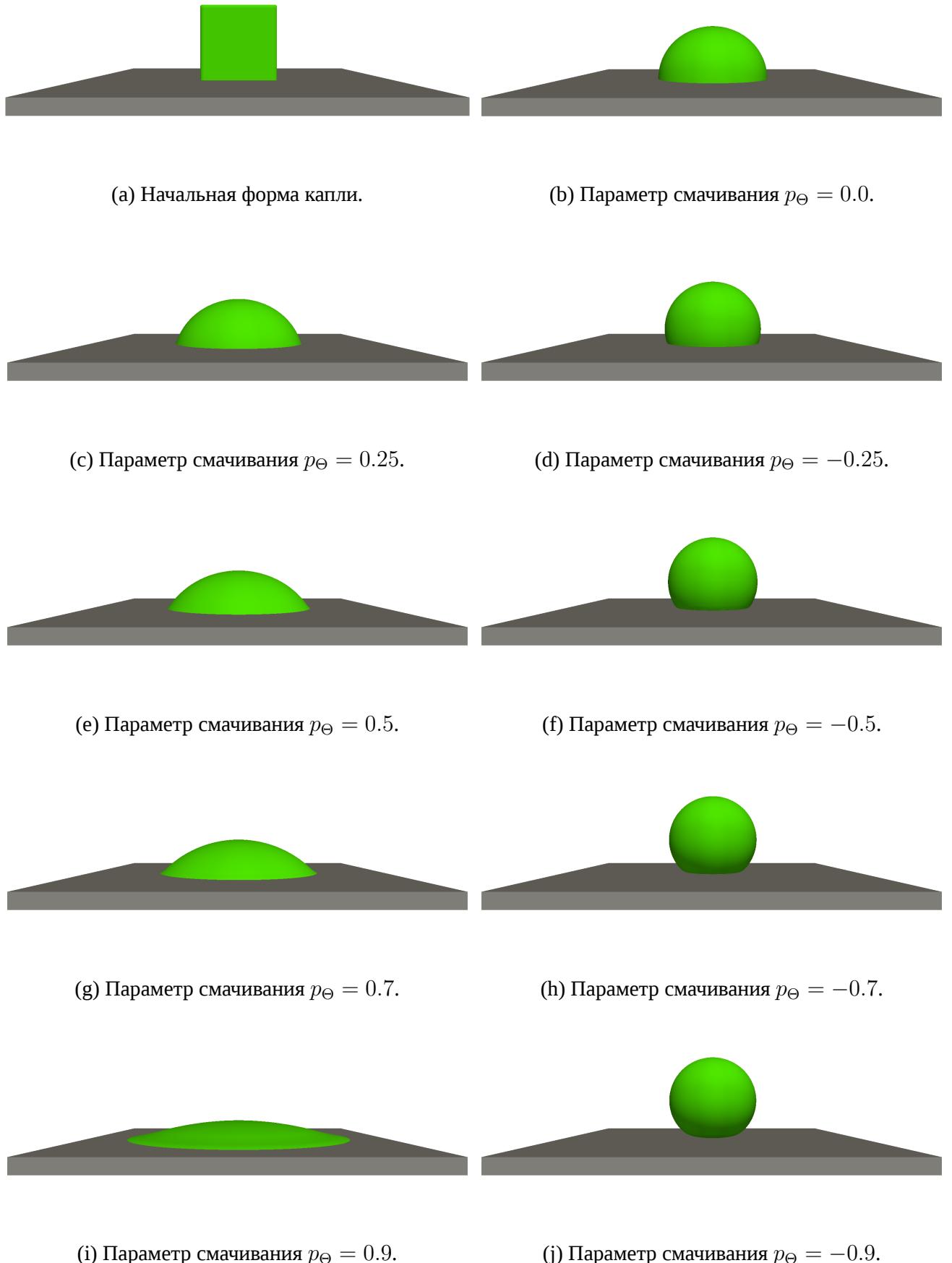


Рисунок 3.11: Установившиеся формы капли для различных значений коэффициента смачивания  $p_\Theta$ .

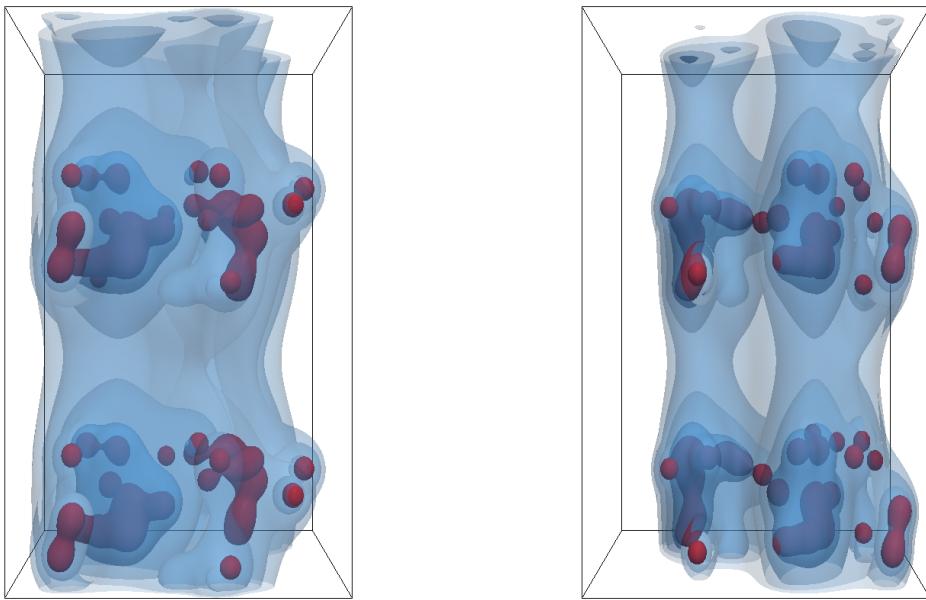


Рисунок 3.12: Поверхности одинаковой концентрации в случае анизотропной диффузии с множеством источников.

### 3.4.7. Диффузия от множества источников

Тест предназначен для расчетного модуля моделирования диффузии. Размер расчетной области равен  $N_x \times N_y \times N_z = 320 \times 320 \times 320$ . На всех гранях заданы периодические граничные условия. Используется анизотропный тензор диффузии:

$$D = \text{diag}(0.1, 0.4, 1) \quad (3.47)$$

В начальный момент времени случайным выбрано  $M >> 1$  точечных источников диффузанта. Поверхности одинаковой концентрации показаны на Рис. 3.12, вид спереди (слева) и вид сбоку (справа).

Точное моделирование диффузии от множества источников может быть полезно при моделировании различных сложных процессов в живых тканях, таких как ишемический инсульт [97].

### **3.5. Заключение к Главе 3**

В данной главе на основе предложенной методологии моделирования рассмотрены детали реализации программного комплекса, рассчитанного на высокопроизводительное моделирование на суперкомпьютерах с гибридной архитектурой. Предложены и описаны параллельные алгоритмы, не требующие использования дополнительной памяти.

Модель однокомпонентного течения основана на РМБ с многорелаксационной аппроксимацией интеграла столкновений, модель двухкомпонентного течения основывается на подходе цветового градиента с однорелаксационной аппроксимацией покомпонентного интеграла столкновений, модель диффузии также основана на РМБ с многорелаксационной аппроксимацией интеграла столкновений, а модель теплопереноса основывается на подходе пассивного скаляра. Предложена и описана механическая модель упаковки проппанта (расклинивающего агента) под нагрузкой, основанная на методе дискретных элементов. Все модели реализованы для расчетов на гибридных суперкомпьютерах с множеством графических ускорителей.

Проанализирована зависимость производительности программного комплекса для моделирования процессов переноса в зависимости от числа задействованных видеокарт на суперкомпьютере «Ломоносов», получен почти линейный рост производительности вплоть до 45 видеокарт.

Произведено сравнение производительности GPU и CPU реализаций программного модуля для моделирования однокомпонентного течения. В области  $450 \times 450 \times 400$  вокселей реализация с вынесенными на GPU расчетами показала в 5 раз большую производительность по сравнению с многопоточной реализацией на CPU, и в 20 раз большую производительность по сравнению с однопоточной реализацией на CPU.

Приведены описания и результаты модельных вычислительных экспериментов, предназначенные для верификации программного комплекса: течение Пуазейля в цилиндрической трубе, определение абсолютной проницаемости цилиндрической трубы, определение лобового сопротивления шара, оценка избытка давления внутри шарообразной капли (формула Лапласа), оценка соответствия фактического и заданного в модели углов смачивания, коалесценция двух капель под действием сил поверхностного натяжения, многоточечная анизотропная диффузия.

## ГЛАВА 4. Моделирование естественных систем

В данной главе продемонстрировано применение реализованного программного комплекса для решения реальных задач: определения абсолютной проницаемости пористой геологической среды, определения долгосрочной проницаемости проппантной упаковки под нагрузкой и определения компонентных проницаемостей пористой среды. Проведено сравнение с данными, полученными в ходе лабораторных экспериментов по определению абсолютной проницаемости каменных геологических образов по газу, фазовых проницаемостей естественных геологических пористых сред для водонефтяных эмульсий.

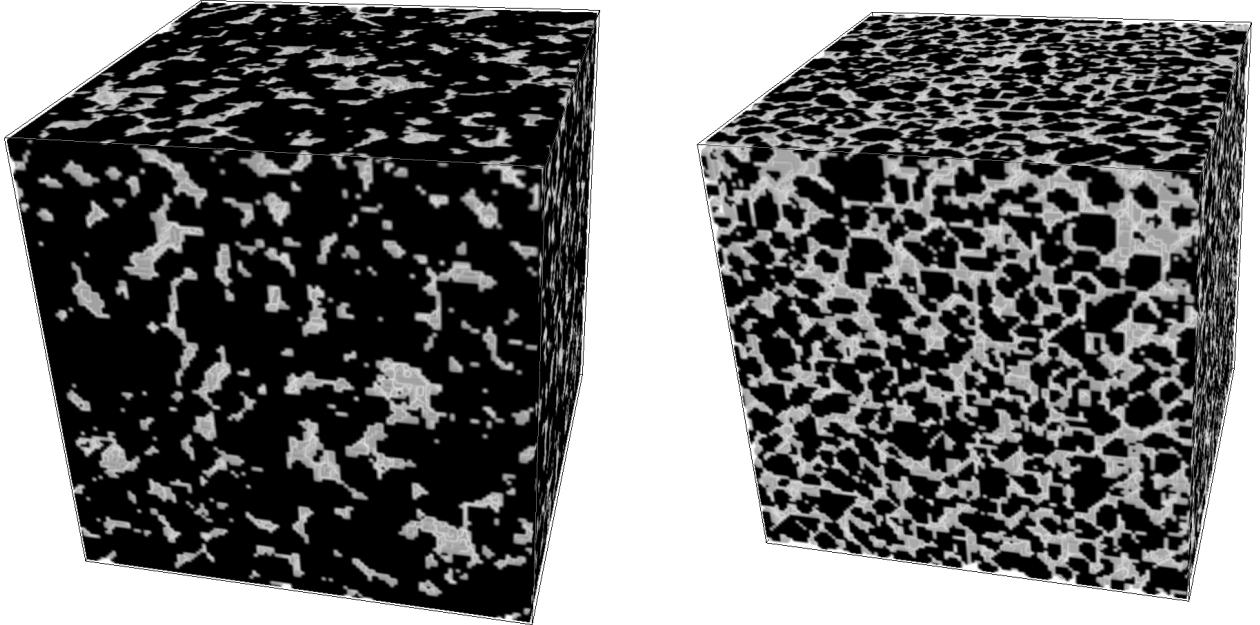
В § 1 описывается вычислительный эксперимент по определению абсолютной проницаемости пористой геологической среды и полученные результаты для коллекции каменных образцов коллекторов месторождений углеводородов Imperial College [63]. В § 2 описывается вычислительный эксперимент по построению зависимости проницаемости от пористости для коллекции образцов и рассматриваются полученные результаты. В § 3 описывается вычислительный эксперимент по определению зависимости абсолютной проницаемости проппантных упаковок от приложенной к ним нагрузки и рассматриваются полученные результаты, рассмотрено пять различных вариантов распределений зерен проппанта по диаметрам. В § 4 рассматриваются результаты численного эксперимента по определению покомпонентной абсолютной проницаемости для каждой компоненты двухкомпонентного течения через пористую среду с разными соотношениями объемов компонент.

### 4.1. Определение абсолютной проницаемости пористой среды

Традиционно коэффициент абсолютной проницаемости  $k_p$  определяется из закона Дарси для медленных течений [7]:

$$Q = -\frac{k_p A}{\mu} \frac{\Delta p}{L} \quad (4.1)$$

где  $Q = A\varphi\bar{u}_z$  — объемный расход жидкости,  $\varphi = \frac{M}{N_x N_y N_z}$  — пористость,  $\bar{u}_z = \frac{\sum_{i=0}^M (\mathbf{u}_z(x_i, y_i, z_i))}{M}$  — модуль средней скорости течения вдоль направления по всем протекаемым ячейкам,  $M$  — общее число протекаемых ячеек,  $A$  — площадь поперечного сечения,  $\Delta p$  — градиент давления,  $L$  — длина образца. Из этого уравнения следует, что:



(a) Образец Berea. Линейные размеры  $400^3$  пикселей, разрешение томографии 5.345 мкм на один пиксель.

(b) Образец LV60A. Линейные размеры  $450^3$  пикселей, разрешение томографии 10.002 мкм на один пиксель.

Рисунок 4.1: Примеры образцов коллекторов месторождений углеводородов, предоставленных Imperial College. Черным обозначена порода, прозрачным — пустотное пространство.

$$k_p = -\frac{Q\mu L}{A\Delta p} \quad (4.2)$$

Принимая во внимание, что  $\mu = \rho\nu$ ,  $\nu = c_s^2(\tau - 0.5)$ , и разрешение томографии составляет  $a$  мкм, уравнение (4.2) принимает вид:

$$k_p = -a^2 \frac{\varphi \bar{u}_z \rho c_s^2 (\tau - 0.5) N_z}{\Delta p} \quad (4.3)$$

Для каждого образца из коллекции Imperial College [63] (см. Рис. 4.1) был произведен расчет абсолютной проницаемости. Продолжительность каждого вычислительного эксперимента зависела от образца, но в среднем не превышала 2 часов на одной GPU. Значения проницаемости приведены в миллидарси. Дарси — широко распространенная в нефтедобыче внесистемная единица, приближенно равна  $1 \text{ мкм}^2$ . Во всех экспериментах рассматривалось течение вдоль оси Z, для каждого образца известны размеры  $N_x \times N_y \times N_z$  и разрешение  $a$  мкм. На плоскостях  $z = 0$  и  $z = N_z - 1$  были заданы граничные условия с фиксированным давлением  $p_0$  и  $p_{N_z-1}$  соответственно ( $p_0 > p_{N_z-1}$ ), на остальных плоскостях использованы периодические граничные условия. Течение направлено вдоль оси Z. Течение считается установившимся в том случае, ес-

ли максимальное отклонение значения абсолютной проницаемости за последние 5000 итераций не превышает 0.001%.

Известно заранее заданное давление на границах, поэтому уравнение для определения абсолютной проницаемости можно записать следующим образом:

$$k_p^1 = a^2 \frac{\varphi \bar{u}_z \rho c_s^2 (\tau - 0.5) N_z}{p_0 - p_{N_z-1}} \quad (4.4)$$

Градиент давления для каждого образца корректировался автоматически в ходе вычислений так, чтобы абсолютные значения средней скорости флюида в модели находились в диапазоне  $(10^{-5}, 10^{-3})$ , а отклонение значения плотности от начального не превышало 1%.

Ограничения на порядок значений скорости и плотности флюида вызваны тем, что для малых значений скорости существенную роль начинает играть вычислительная ошибка (значения функций распределения хранятся в массивах одинарной точности для экономии памяти на графических ускорителях). Для больших значений скорости не выполняется закон Дарси для медленных течений, а большое изменение плотности противоречит предположению о «почти несжимаемости» жидкости, используемом при выводе РМБ [75].

При численном определении абсолютной проницаемости необходимо учитывать достаточность разрешения томографии для определения этой величины. Численные эксперименты показывают, что достоверными оказываются результаты с безразмерной проницаемостью 10 и выше (по порядку величины). Учитывая, что  $k_p$  при пересчете из внутренних безразмерных параметров умножается на значение разрешения  $a$ :

$$k_p = a^2 k_p^{lu} \quad (4.5)$$

Учитывая вышесказанное, предлагается эмпирическая оценка минимально необходимого разрешения, которая составляет порядка  $a_{min} = \sqrt{\frac{k_p}{10}}$  (см. Таб /reftab:results-ic-res). Если же разрешение оказывается грубее ( $a > a_{min}$ ), то полученная с помощью численного эксперимента абсолютная проницаемость может быть недостоверной. Среди коллекции Imperial College есть три образца с недостаточным разрешением томографии (C2, S3 и S4) согласно этому критерию, для которых, действительно, посчитанная абсолютная проницаемость значительно (на фоне остальных) отличается от лабораторной, либо в результате расчетов оказалась равной нулю (см. Рис. 4.2).

При определении фильтрационных характеристик геологических образцов месторождений углеводородов на основе численного моделирования течения в изображении, полученном с помощью компьютерной томографии, следует учитывать как разрешение томографии, так и её качество. Например, при разрешении 5 мкм границы проницаемых областей могут быть определены с точностью до половины пикселя, что приводит к ошибке определения диаметра каналов  $\pm 5$

Таблица 4.1: Линейные размеры в пикселях и разрешение в микрометрах на пиксель моделей образцов из коллекции Imperial College. В последней колонке указана оценка минимального разрешения  $a_{min}$ , необходимого для расчетного модуля. Если  $a > a_{min}$ , то результат моделирования недостоверный, такие образцы помечены звездочкой.

Образец	$N_x$ (пикс)	$N_y$ (пикс)	$N_z$ (пикс)	Разрешение образца (мкм)	Минимально необходимое разрешение $a_{min}$ (мкм)
A1	300	300	300	3.85	26.87
Berea	400	400	400	5.345	11.34
C1	400	400	400	2.85	10.50
C2*	400	400	400	5.345	2.69
F42A	450	450	450	9.996	76.81
F42B	450	450	450	10.002	72.32
F42C	450	450	450	10.002	70.99
LV60A	450	450	450	10.002	59.41
LV60B	450	450	425	8.851	56.12
LV60C	450	450	450	10.002	44.05
S1	300	300	300	8.683	12.95
S2	300	300	300	4.956	19.74
S3*	300	300	300	9.1	4.73
S4*	300	300	300	8.96	5.09
S5	300	300	300	3.997	21.57
S6	300	300	300	5.1	33.13
S7	300	300	300	4.803	26.39
S8	300	300	300	4.892	36.29
S9	300	300	300	3.398	14.91

мкм, или 1 воксель. Для узких каналов в 2-3 вокселя в диаметре эта ошибка является существенной, поэтому для низкопроницаемых образцов с преобладанием узких каналов (распределение каналов по диаметрам может быть получено в лаборатории) требуется томография с более высоким разрешением. Качество томографии характеризуется наличием артефактов (в том числе кольцевых) и шума на томографии, которые могут быть частично устранены за счет дополнительной постобработки изображения или учёта скорректированной аппаратной функции томографа.

Недостаточное разрешение томографии приводит к численным ошибкам в узких каналах (3 вокселя и менее в диаметре, см. раздел 3.4.1) и к ошибкам сегментации порового пространства (из-за округления до разрешения томографии при сегментации порового пространства). Недостаточное качество томографии приводит к появлению артефактов и дополнительному шуму на результирующем изображении. Все это следует учитывать при решении практических задач.

Таблица 4.2: Значения абсолютной проницаемости в миллидарси: лабораторные данные, данные предоставленные Imperial College и полученные в ходе вычислительного эксперимента с помощью метода решеточных уравнений Больцмана в данной работе. Последняя колонка в таблице — относительная ошибка определения абсолютной проницаемости с помощью РМБ по сравнению с лабораторными данными.

Образец	Абсолютная проницаемость, данные лаборатории (миллидарси)	Абсолютная проницаемость, данные Imperial College (миллидарси)	Абсолютная проницаемость, данные РМБ (миллидарси)	Относительная ошибка РМБ (%)
A1	7220	8076	5638.24	21.91
Berea	1286	1111	1307.44	1.67
C1	1102	556	1212.77	10.05
C2	72.333	158	229.02	216.62
F42A	59000	61000	56461.80	4.30
F42B	52300	48000	56301.61	7.65
F42C	50400	44800	52022.80	3.22
LV60A	35300	27200	29506.50	16.41
LV60B	31500	29000	30129.80	4.35
LV60C	19400	23200	28857.14	48.75
S1	1678	1486	1563.82	6.80
S2	3898	3951	3554.26	8.82
S3	224	281	0.00	-
S4	259	169	0.00	-
S5	4651	5369	5983.61	28.65
S6	10974	11282	12368.09	12.70
S7	6966	7926	6887.79	1.12
S8	13169	13932	13680.80	3.89
S9	2224	3640	2168.95	2.48

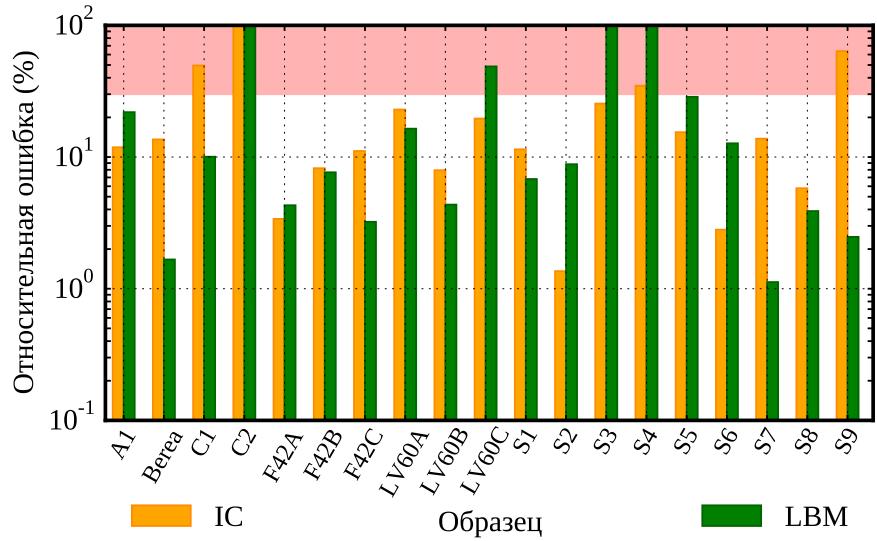


Рисунок 4.2: Относительные ошибки коэффициентов абсолютной проницаемости, предоставленных Imperial College и полученных в данной работе, по сравнению с лабораторными данными.

В целом, полученные с помощью численного моделирования методом решеточных уравнений Больцмана на томографических изображениях значения абсолютной проницаемости согласуются с лабораторными данными (см. Таб. 4.2 и Рис. 4.2). Посчитанные Imperial College значения абсолютной проницаемости были получены с помощью метода Pore-Network [17]. Видно, что РМБ лучше работает в образцах с крупными разрешенными порами, в то время как сеточный алгоритм обладает преимуществом на образцах с мелкими порами. Скорее всего, это вызвано особенностями этапа построения сеточной модели в Pore-Network, сглаживающего дефекты сегментации томографии на мелком масштабе.

## 4.2. Построение петрофизических зависимостей

В случае больших образцов или ограниченных вычислительных ресурсов, оказывается сложно произвести моделирование в полноразмерном образе. В таком случае удобно производить серию экспериментов на фрагментах меньшего размера, и затем строить зависимость абсолютной проницаемости от пористости, оценивая по ней проницаемость полного образца с известной пористостью. Этот подход предполагает гомогенность полного образца, репрезентативность выбранных фрагментов и дает только оценку абсолютной проницаемости.

Для демонстрации методики был проведен вычислительный эксперимент по определению абсолютной проницаемости для 89 случайных фрагментов, вырезанных из образа Berea

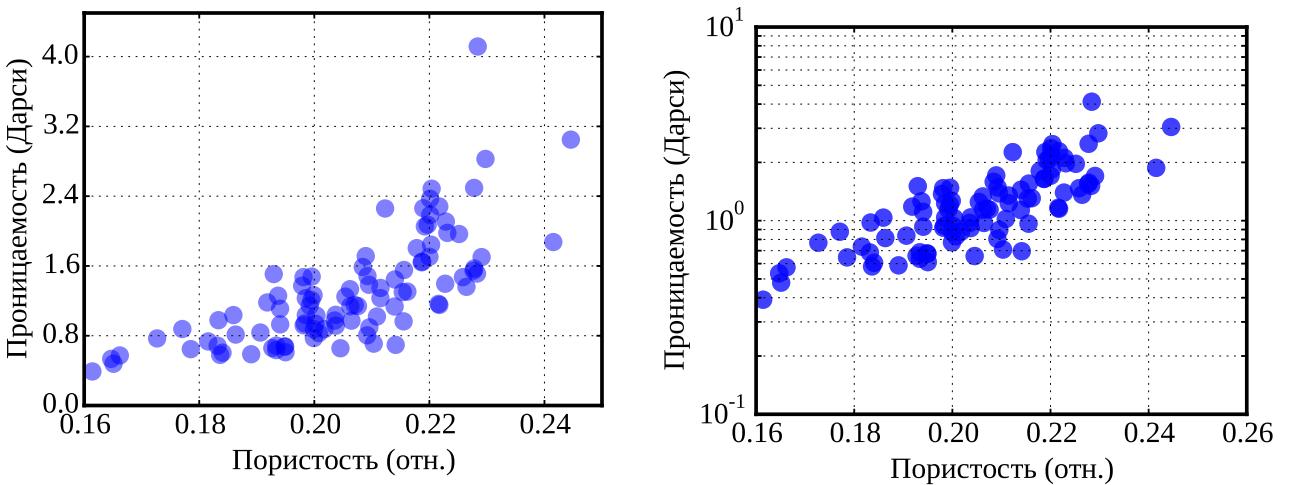


Рисунок 4.3: «Облако» зависимости проницаемости от пористости для образца Berea из коллекции Imperial College: линейная ось ординат (слева) и логарифмическая ось ординат (справа).

Sandstone [4] из коллекции Imperial College с полным размером  $1024^3$  вокселей. Размер фрагментов одинаковый,  $300^3$  воксели. Расположение фрагментов выбиралось случайным образом. Продолжительность каждого вычислительного эксперимента составила около 1 часа на одной GPU. Результат представлен на Рис. 4.3.

### 4.3. Моделирование изменения проницаемости проппантных упаковок под нагрузкой

Механическое поведение проппантных упаковок под нагрузкой моделируется с помощью подхода, основанного на методе дискретных элементов. Ускорение вычислительного модуля осуществлялось за счет массивно-параллельных вычислений на графическом ускорителе (GPU), для этого использовалась технология Compute Unified Device Architecture (CUDA). Производительности одного GPU оказалось достаточно для моделирования, поэтому дальнейшее расширение поддержки на одновременное использование нескольких GPU не производилось.

Гидродинамическое моделирование течения в пористой среде, образованной проппантной упаковкой под нагрузкой и полученной в механической модели, осуществлялось с помощью метода решеточных уравнений Больцмана. В этом расчетном модуле также используется ускорение за счет переноса существенной части расчетов на видеокарту, но производительности одного GPU оказывается недостаточно. Поэтому в вычислительном эксперименте использовалось 25 GPU в разделе гриди гибридного суперкомпьютера «Ломоносов» [100].

Абсолютная проницаемость также определяется на основе закона Дарси для медленных течений (4.2). Размер расчетной области составил от  $550 \times 550 \times 550$  до  $350 \times 550 \times 550$ . Адаптация

полученных в результате эксперимента кривых на основе лабораторных данных производится за счет параметра эластичности зерен проппанта в механической модели. В вычислительном эксперименте рассматривается диапазон от 400 Дарси при нагрузке 10 МПа до 30 Дарси при нагрузке в 69 МПа в пересчете на физические величины. Без нагрузки пористость всех упаковок составляет около 33%.

Исследовались зависимости проницаемости от нагрузки для пяти различных распределений диаметров зерен (Таб. 4.3). Для упрощения расчетов, но без потери общности предполагается, что итоговые образы проппантных упаковок имеют физическое разрешение 1 мкм на воксель. Для того, чтобы пересчитать полученные результаты на другое разрешение изображения, достаточно умножить результат на квадрат этого разрешения в микрометрах. Полученный результат будет проницаемостью в миллидарси. Для каждого из пяти распределений диаметров зерен было проведено по 5 экспериментов для каждого значения нагрузки, суммарно 225 вычислительных экспериментов.

Всего выбрано 9 различных значений нагрузки через равные промежутки в диапазоне деформации исходной упаковки вдоль оси  $X$  от  $B_x \times B_y \times B_z = 600 \times 600 \times 600$  до  $400 \times 600 \times 600$  включительно. Для того, чтобы уменьшить влияние выравнивания зерен у стенок на границах расчетной области, из полученных образов для непосредственных гидродинамических расчетов извлекается внутренний фрагмент без 25-пиксельного внешнего слоя. Таким образом, в гидродинамическую модель загружаются образы с размерами от  $550 \times 550 \times 550$  до  $350 \times 550 \times 550$ .

На верхней  $z = N_z - 1$  и нижней  $z = 0$  границах заданы граничные условия с фиксированным давлением. На остальных гранях заданы периодические граничные условия. На краях между проницаемыми и непроницаемыми ячейками используется условие «межрешеточный отскок».

Полученные в результате вычислительного эксперимента зависимости значений абсолютной проницаемости от нагрузки для разных распределений радиусов представлены на Рис. 4.6 (А). Относительная разность между проницаемостью упаковки с одинаковыми зернами и другими типами распределений диаметров представлена на Рис. 4.6 (Б). Аналогичные зависимости от значения пористости представлены на Рис. 4.7, (А) и (Б).

На Рис. 4.4 представлено сравнение полученных в ходе вычислительного эксперимента значений абсолютной проницаемости упаковки шаров и определенных согласно уравнению Козени-Кармана для упаковок шаров одинакового радиуса [12]:

$$k_p = \frac{R^2 \varphi^3}{45(1 - \varphi)^2} \quad (4.6)$$

где  $\varphi$  — пористость упаковки,  $R$  — радиус шаров (зерен проппанта).

На малых значениях деформации (соответственно, больших значениях пористости) видно хорошее соответствие значений, полученных из уравнения Козени-Кармана и результатов числен-

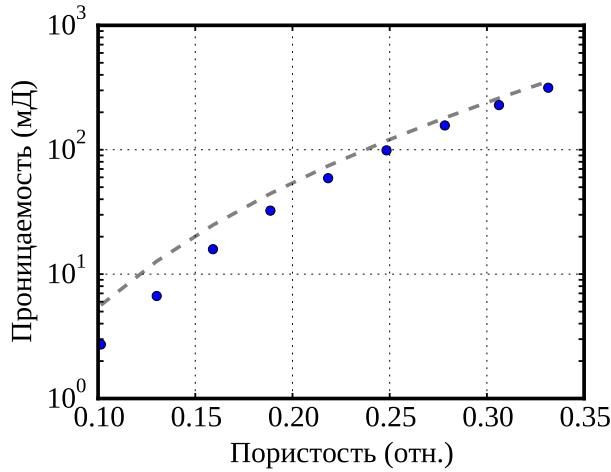


Рисунок 4.4: Определенная по закону Кармана проницаемость упаковки шаров одинакового радиуса (пунктирная линия) и результаты, полученные в ходе вычислительного эксперимента (показаны точками).

ного эксперимента. На значительных деформациях (малых пористостях) поровое пространство уже не может быть аппроксимировано насыпной моделью шаров одинакового радиуса, заметно отклонение от аналитической кривой.

Наибольшую проницаемость демонстрируют проппантные упаковки с одинаковыми диаметрами зерен. Это ожидаемый результат, т.к. при больших нагрузках и деформации проппантной упаковки, одинаковые зерна образуют устойчивые периодические структуры (Рис. 4.8). Так, на Рис. 4.8 можно видеть периодические колебания пористости около стенок и проникающие вглубь упаковки. По этой же причине упаковки с нормальным распределением диаметров менее проницаемы, т.к. регулярность нарушается быстрее. В остальных случаях регулярность нарушается еще быстрее. Таким образом, простая механическая модель позволяет лучше понять особенности подбора соотношений диаметров зерен проппанта.

На Рис. 4.9 показаны распределения количества зерен по числу непосредственных соседей. Для каждого распределения представлены значения среднего числа соседей и проницаемости образца в миллидарси. На гистограммах (C) и (D) можно видеть два пика в распределении, они соответствуют двум диаметрам дуальных распределений. На (A) показано распределение соседей для зерен одинакового радиуса, поэтому виден только один пик. Случай (B) почти не отличается он (A), но более размыт из-за нормального распределения диаметров зерен проппанта. В общем случае, абсолютная проницаемость обратно пропорциональна среднему числу соседей (которое, в свою очередь, характеризует плотность упаковки), но может незначительно варьироваться из-за статистической природы сгенерированных изображений.

На Рис. 4.10 показаны средние полные деформации зерен проппанта. В данной механической модели не рассматривается разрушение зерен при сильной нагрузке. Полные деформации поз-

Таблица 4.3: Численные значения параметров, использованных при моделировании проппантных упаковок. Количество зерен  $N$  выбрано таким образом, чтобы без нагрузки пористость упаковки составляла около 33.3 % и  $N$  было кратно 128. Во всех случаях средний диаметр равен 28 пикселей.

Распределение диаметров	Число зерен	Пористость	Описание
одинаковые зерна	12288	33.16 %	$D = 28$
нормальное распределение	11520	33.48 %	$\mu = 28$ и $\sigma = 4.2$
два пропорциональных диаметра	9088	33.61 %	$D_1 = 18.67$ и $D_2 = 37.33$
два некратных диаметра I	10752	33.28 %	$D_1 = 22$ и $D_2 = 34$
два некратных диаметра II	11264	33.14 %	$D_1 = 23.2$ и $D_2 = 32.8$

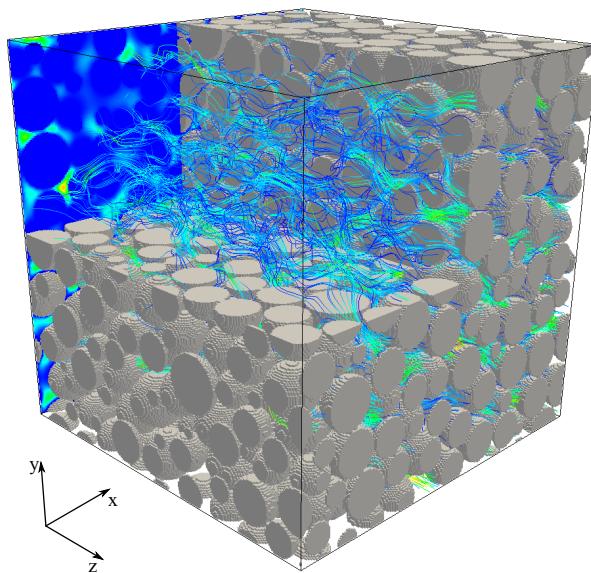


Рисунок 4.5: Линии тока демонстрируют течение в упаковке. Направление течения совпадает с осью  $Z$ . Цвет соответствует скорости среды, часть объема удалена для наглядности.

воляют оценить количество гипотетически разрушенных зерен и расхождение результатов вычислительного и лабораторного экспериментов, которое может возникать из-за этого.

#### 4.4. Определение компонентных проницаемостей

Для образца S6 из коллекции Imperial College были посчитаны фазовые (компонентные) проницаемости для разных значений соотношений объемов компонент: от 0 (весь объем занимает «синяя» компонента) до 0.82 (82% проницаемого объема занимает «красная» компонента). Использовались следующие параметры модели:  $\alpha^r = 0.001$ ,  $\alpha^b = 0.001$ , поверхностное натяжение  $A^r = A^b = 0.001$ , смачиваемость, параметры релаксации  $\tau^r = 0.8$ ,  $\tau^b = 1.5$  (то есть компоненты имели разные вязкости), косинус угла смачивания задан  $p = 0.7$ .

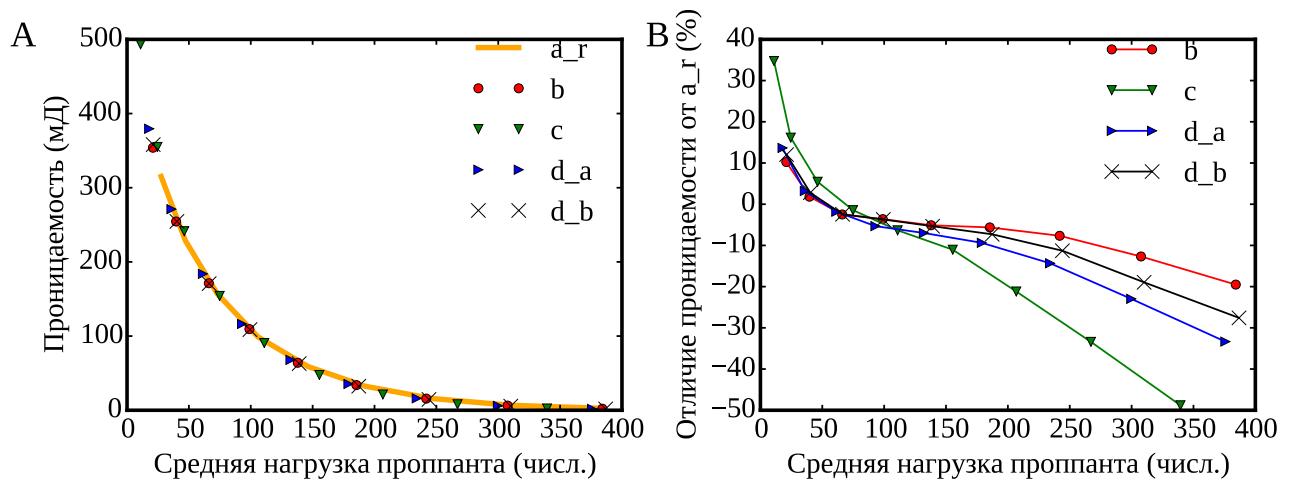


Рисунок 4.6: График зависимости проницаемости для разных нагрузок (А) и относительная разность между распределением с одинаковыми диаметрами зерен («a<sub>r</sub>» на (А)) и другими распределениями (Б). На всех графиках: (b) нормальное распределение, (c) два кратных диаметра, (d<sub>a</sub>) два некратных диаметра I, (d<sub>b</sub>) два некратных диаметра II.

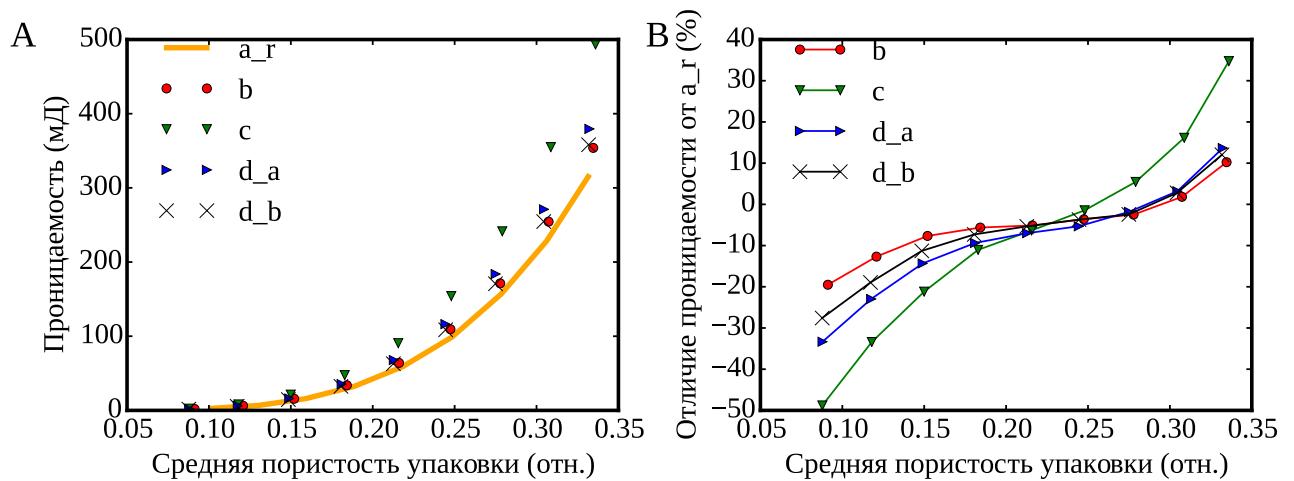


Рисунок 4.7: График зависимости проницаемости для разных пористостей (А) и относительная разность между распределением с одинаковыми диаметрами зерен («a<sub>r</sub>» на (А)) и другими распределениями (Б). На всех графиках: (b) нормальное распределение, (c) два кратных диаметра, (d<sub>a</sub>) два некратных диаметра I, (d<sub>b</sub>) два некратных диаметра II.

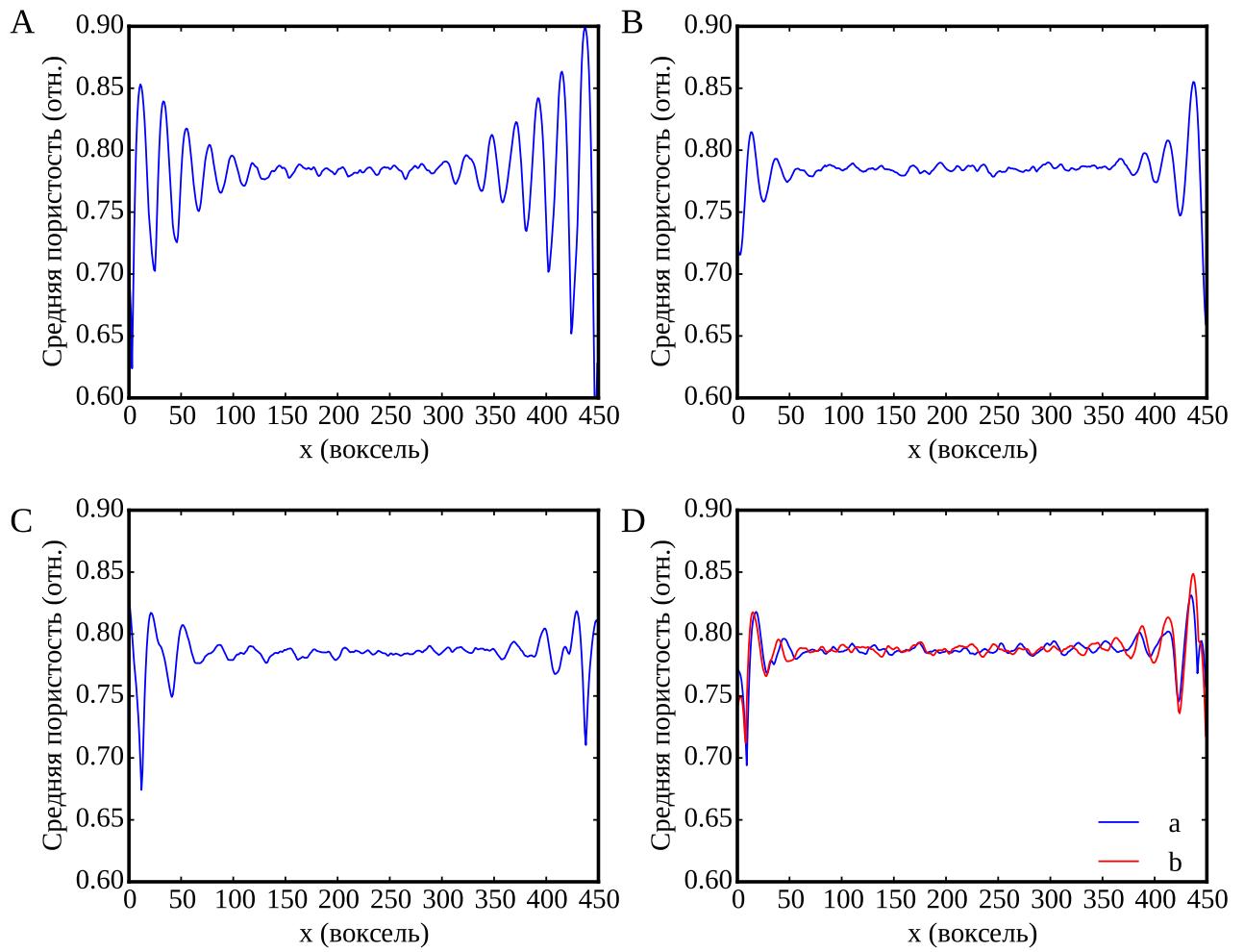


Рисунок 4.8: Графики средней по слою пористости вдоль оси  $X$ . Средняя пористость в точке  $x_i$  рассчитывается как средняя величина по плоскости, перпендикулярной оси  $X$  и проходящей через эту точку: (А) одинаковые зерна, (В) нормальное распределение, (С) два пропорциональных диаметра, (Д) два некратных диаметра I (а) и II (б). Все величины получены для случая  $N_x = 450$ .

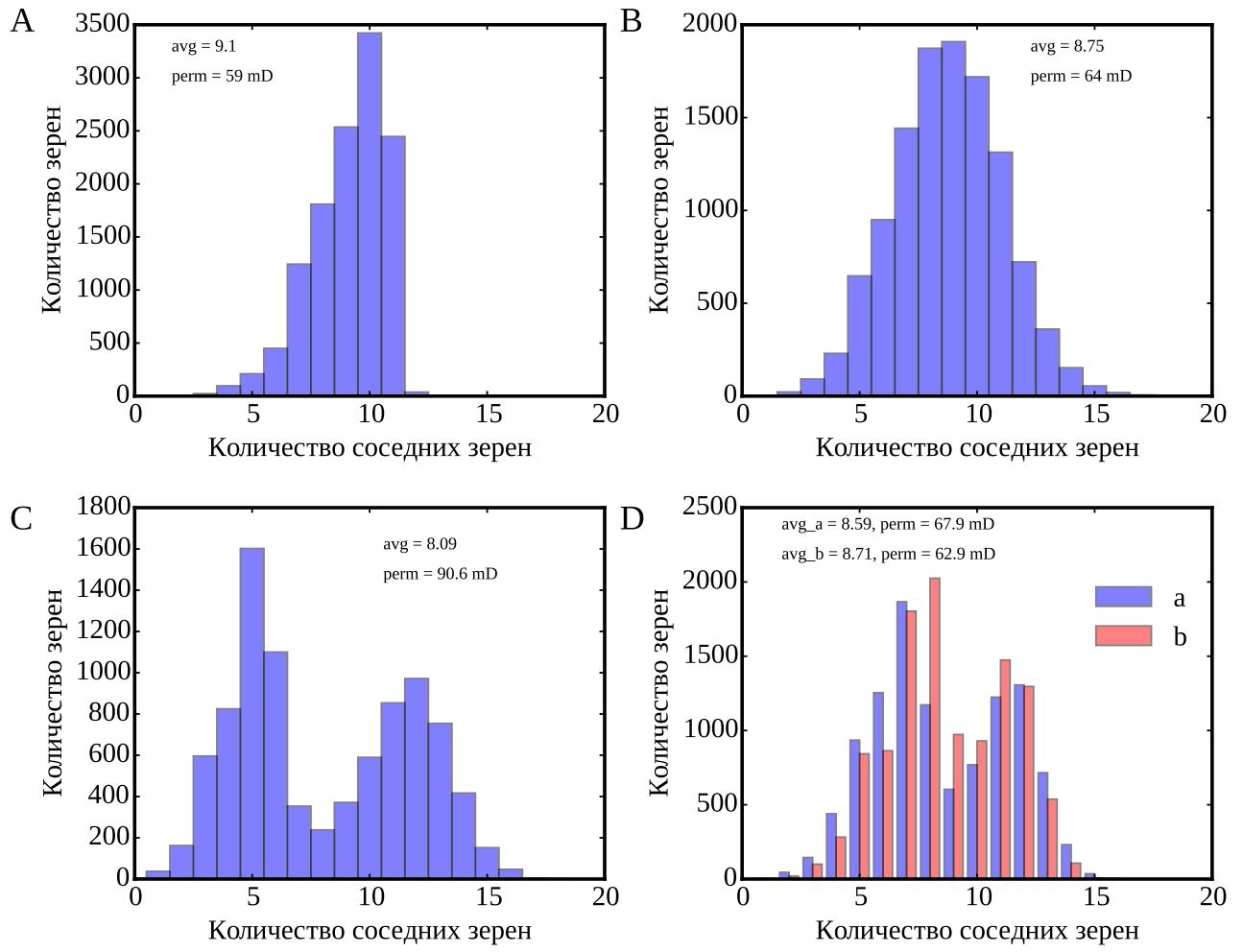


Рисунок 4.9: Гистограммы распределений количества соседей: (A) одинаковые зерна, (B) нормальное распределение, (C) два пропорциональных диаметра, (D) два некратных диаметра I (a) и II (b). Все величины получены для случая  $N_x = 450$ .

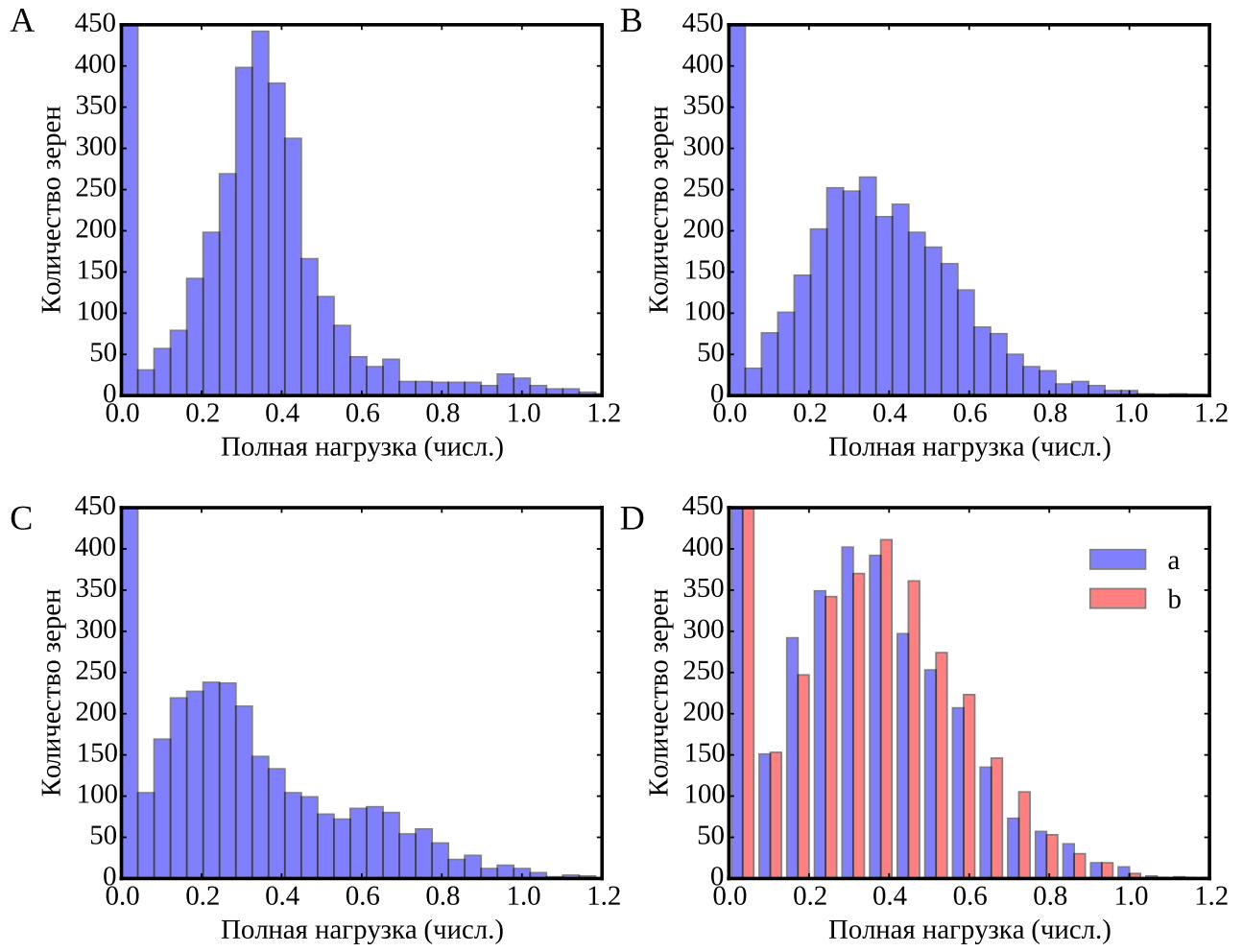


Рисунок 4.10: Гистограммы средней полной деформации: (A) одинаковые зерна, (B) нормальное распределение, (C) два пропорциональных диаметра, (D) два некратных диаметра I (a) и II (b). Все величины получены для случая  $N_x = 450$ .

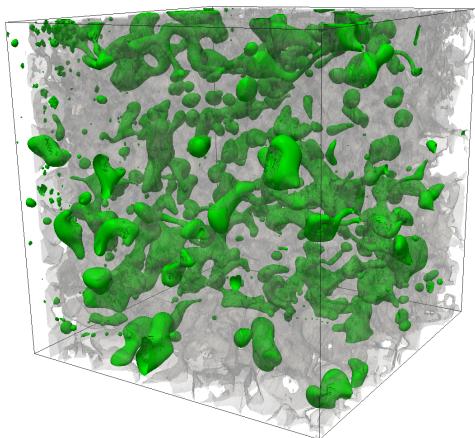


Рисунок 4.11: Течение водонефтяной эмульсии в пористой среде, полученной в результате рентгеновской компьютерной микротомографии геологического образца. Вторая компонента (вода) сделана прозрачной для наглядности.

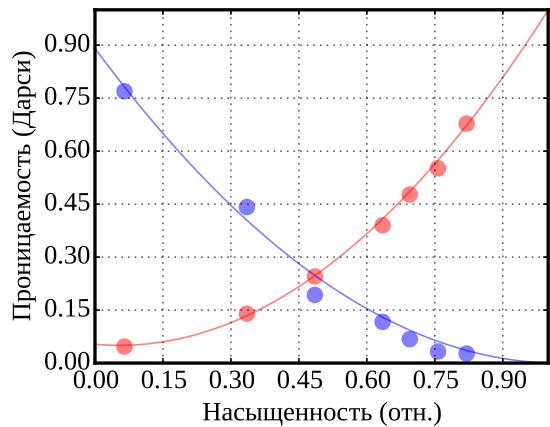


Рисунок 4.12: Нормированные компонентные проницаемости для разных соотношений объемов компонент (где 0 — вся область заполнена «синей» компонентной (вода), 1 — вся область заполнена «красной» компонентой (нефть)).

Рассматривалось течение вдоль оси Z. Для того, чтобы достичь необходимого соотношения объемов компонент, в начальный момент времени в вычислительный объем, заполненный «синей» жидкостью, накидывались сферические капли «красной» жидкости до тех пор, пока заданное соотношение не было получено. Капли не пересекались между собой. Течение осуществлялось за счет действия объемной силы [85] (см. Рис. 4.11), на всех границах расчетного объема использовали периодические граничные условия, по краям области вдоль оси Z добавлены непроницаемые стенки. Для сшивки верхней и нижней границ расчетного объема (перпендикулярных оси Z), до и после образца были добавлены [29] дополнительные пустоты по 30 вокセルей вдоль оси Z. Таким образом, расчетный объем составил  $300 \times 300 \times 360$  воксели при фактических размерах образца  $300 \times 300 \times 300$ . Течение считается установленвшимся в том случае, если максимальное отклонение значения абсолютной проницаемости за последние 5000 итераций не превышает 0.1%.

Нормированные результаты моделирования представлены на Рис. 4.12. Продолжительность каждого вычислительного эксперимента (пара точек при заданной насыщенности) составила около 8 часов на одной GPU. Полученная форма кривых в целом соответствует лабораторной, полученной на каменных образцах для водонефтяной эмульсии [24]. Такая постановка эксперимента по определению относительных компонентных проницаемостей называется стационарной [1]. Для постановки точного эксперимента в модели порового пространства недостаточно информации о минеральном составе, реальной смачиваемости и т.д.

## **4.5. Заключение к Главе 4**

В данной главе продемонстрировано применение реализованного программного комплекса для решения реальных задач: определения абсолютной проницаемости геологических образцов, определения долгосрочной проницаемости проппантной упаковки под нагрузкой и определения компонентных проницаемостей пористой среды. Проведено сравнение с данными, полученными в ходе лабораторного эксперимента.

Полученная в ходе вычислительных экспериментов абсолютная проницаемость геологических образцов месторождений углеводородов из коллекции Imperial College отличается от лабораторных значений в большинстве случаев не более, чем на 20%. Получена эмпирическая оценка минимально допустимого разрешения томографического изображения, необходимого для расчетного модуля на основе оценки проницаемости. Показаны результаты вычислительного эксперимента по построению зависимости проницаемости от пористости для коллекции образцов. Представлены результаты численного эксперимента по определению зависимости абсолютной проницаемости проппантных упаковок от приложенной к ним нагрузки для различных распределений диаметров шаров. Для упаковки шаров с одинаковыми диаметрами произведено сравнение со значениями, полученными из уравнения Козени-Кармана. Представлены результаты численного эксперимента по определению фазовых (покомпонентных) проницаемостей для двухкомпонентного течения через пористую среду с различными соотношениями объемов компонент.

## ЗАКЛЮЧЕНИЕ

Использование метода решеточных уравнений Больцмана в качестве алгоритмической платформы позволило создать программный комплекс для моделирования различных процессов переноса (течение, диффузия и теплоперенос) на основе схожих алгоритмов. Реализованный программный комплекс оптимизирован для высокопроизводительной работы на гибридных суперкомпьютерных системах с множеством графических ускорителей (GPU). В качестве такой системы для массивно-параллельных расчетов используется раздел при суперкомпьютера «Ломоносов».

На суперкомпьютере «Ломоносов» получен почти линейный рост производительности в зависимости от числа использованных видеокарт вплоть до 45 GPU. Сравнение производительности GPU и CPU реализаций программного модуля в случае области  $450 \times 450 \times 400$  вокселей показало в 5 раз большую производительность реализации на GPU по сравнению с многопоточной реализацией на CPU, и в 20 раз большую производительность по сравнению с однопоточной реализацией на CPU.

Реализованный программный комплекс был верифицирован на модельных примерах: течении Пуазейля в цилиндрической трубе, определении абсолютной проницаемости цилиндрической трубы, определении лобового сопротивления шара, оценке избытка давления внутри шарообразной капли (формула Лапласа), оценке соответствия фактического и заданного в модели углов смачивания, коалесценции двух капель под действием сил поверхностного натяжения, многоточечной анизотропной диффузии.

Реализованный программный комплекс применен для решения реальных задач: определению абсолютной проницаемости геологических образцов месторождений углеводородов на основе их компьютерной томографии, построения зависимости проницаемости от пористости, определению зависимости абсолютной проницаемости проппантных упаковок от приложенной к ним нагрузки для различных распределений диаметров шаров, определению фазовых (покомпонентных) проницаемостей. Получена эмпирическая оценка минимально допустимого разрешения томографического изображения, необходимого для расчетного модуля на основе оценки проницаемости. Для насыпной модели шаров с одинаковыми диаметрами показано хорошее соответствие полученной в ходе вычислительного эксперимента зависимости абсолютной проницаемости от пористости со значениями, полученными из уравнения Козени-Кармана.

Основными результатами данной диссертационной работы являются:

1. Методология моделирования процессов переноса с использованием множества графических ускорителей на гибридных суперкомпьютерных системах. Созданный и верифицированный на тестовых примерах соответствующий программный комплекс, позволяющий производить моделирование процессов переноса.
2. Алгоритмизация и реализация численных методов для моделирования процессов переноса на основе единого инструментария — метода решеточных уравнений Больцмана.
3. Механическая модель проппантной упаковки под нагрузкой (проппант — гранулярный материал, часто песок, используемый в нефтедобывающей промышленности в качестве расклинивающего агента для удержания трещин при гидроразрыве пласта).
4. Методика совместного использования моделей порового пространства (как на полученных на основе рентгеновской компьютерной микротомографии, так и созданных на основе алгоритмических моделей) и гидродинамической модели.

Реализованный программный комплекс может применяться не только в нефтегазовой сфере, но и для моделирования биологических, медицинских и инженерных задач. Схожие подходы и идеология моделирования может быть применена не только к методу решеточных уравнений Больцмана, но и к моделированию большинства физических явлений, обладающих локальностью по данным.

## СПИСОК ЛИТЕРАТУРЫ

1. *Alemán M. A., Ramamohan T. R., Slattery J. C.* The difference between steady-state and unsteady-state relative permeabilities // Transport in Porous Media. — 1989. — Т. 4, № 5. — С. 449—493. — (Дата обр. 09.03.2015).
2. *Bailey P., Myre J., Walsh D. C. S., David J. Lilja, Saar O. M.* Accelerating Lattice Boltzmann Fluid Flow Simulations Using Graphics Processors // (Parallel Processing, 2009. ICPP '09. International Conference). — Vienna, Austria, 2009. — С. 550—557.
3. *Banari A., Janßen C., Grilli S. T., Krafczyk M.* Efficient GPGPU implementation of a lattice Boltzmann model for multiphase flows with high density ratios // Computers & Fluids. — 2014. — Апр. — Т. 93. — С. 1—17.
4. Berea Sandstone Imperial College. — URL: [https://workspace.imperial.ac.uk/earthscienceandengineering/Public/external/Research/PERM/Berea%5C\\_1024%5C\\_2.7745.zip](https://workspace.imperial.ac.uk/earthscienceandengineering/Public/external/Research/PERM/Berea%5C_1024%5C_2.7745.zip).
5. *Bhatnagar P. L., Gross E. P., Krook M.* A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems // Phys. Rev. — 1954. — Т. 94, № 3. — С. 511.
6. *Bikulov D., Saratov A., Grachev E.* Prediction of the permeability of proppant packs under load // International Journal of Modern Physics C. — 2015. — С. 1—18.
7. *Boek E.* Pore Scale Simulation of Flow in Porous Media Using Lattice-Boltzmann Computer Simulations // SPE. — 2010.
8. *Bouzidi M., Firdaouss M., Lallemand P.* Momentum transfer of a Boltzmann-lattice fluid with boundaries // Phys. Fluids. — 2001. — Т. 13, № 11. — С. 3452.
9. *Boyd J., Buick J., Green S.* A second-order accurate lattice Boltzmann non-Newtonian flow model // J. Phys. Math. Gen. — 2006. — Т. 39, № 46. — С. 14241—14247.

10. *Boyd J., Buick J., Cosgrove J. A., Stansell P.* Application of the lattice Boltzmann model to simulated stenosis growth in a two-dimensional carotid artery // *Phys. Med. Biol.* — 2005. — Т. 50, № 20. — С. 4783—4796.
11. *Buick J. M., Greated C. A.* Gravity in a lattice Boltzmann model // *Phys. Rev. E*. — 2000. — Т. 61, № 5.
12. *Carman P.* Permeability of saturated sands, soils and clays // *The Journal of Agricultural Science*. — 1939. — Т. 29, № 02. — С. 262—273.
13. *Chang L.-C., El-Araby E., Dang V. Q., Dao L. H.* GPU acceleration of nonlinear diffusion tensor estimation using CUDA and MPI // *Neurocomputing*. — 2014. — Июль. — Т. 135. — С. 328—338.
14. *Chen S., Doolen G. D.* Lattice Boltzmann method for fluid flows // *Annu. Rev. Fluid Mech.* — 1998. — Т. 30, № 1. — С. 329—364.
15. *Cheng J.* Professional cuda c programming. — Indianapolis, IN : John Wiley, Sons, 2014.
16. *d'Humieres D., Ginzburg I., Krafczyk M., Lallemand P., Luo L.-S.* Multiple-relaxation-time lattice Boltzmann models in three dimensions // *Philos. Trans. R. Soc. Math. Phys. Eng. Sci.* — 2002. — Т. 360, № 1792. — С. 437—451.
17. *Dong H., Blunt M. J.* Pore-network extraction from micro-computerized-tomography images // *Physical Review E*. — Т. 80, № 3.
18. *Feichtinger C., Habich J., Köstler H., Rüde U., Aoki T.* Performance Modeling and Analysis of Heterogeneous Lattice Boltzmann Simulations on CPU-GPU Clusters // *Parallel Computing*. — 2014. — Дек.
19. *Filippova O., Hanel D.* Grid Refinement for Lattice-BGK Models // *J. Comput. Phys.* — 1998. — № 147. — С. 219—228.
20. *Flynn M.* Some computer organizations and their effectiveness // *Computers, IEEE Transactions on*. — 1972. — Т. 100, № 9. — С. 948—960.
21. *Flynn M. J., Rudd K. W.* Parallel architectures // *ACM Computing Surveys (CSUR)*. — 1996. — Т. 28, № 1. — С. 67—70.
22. *Gabriel E., Fagg G. E., Bosilca G., Angskun T., Dongarra J. J., Squyres J. M., Sahay V., Kambadur P., Barrett B., Lumsdaine A. [и др.]* Open MPI: Goals, concept, and design of a next generation MPI implementation // *Recent Advances in Parallel Virtual Machine and Message Passing Interface*. — Springer, 2004. — С. 97—104.

23. *Galizia A., D'Agostino D., Clematis A.* An MPI–CUDA library for image processing on HPC architectures // Journal of Computational and Applied Mathematics. — 2015. — Янв. — Т. 273. — С. 414—427.
24. *Ghassemi A., Pak A.* Numerical study of factors influencing relative permeabilities of two immiscible fluids flowing through porous media using lattice Boltzmann method // Journal of Petroleum Science and Engineering. — 2011. — Апр. — Т. 77, № 1. — С. 135—145.
25. *Ginzburg I., d'Humières D., Kuzmin A.* Optimal Stability of Advection-Diffusion Lattice Boltzmann Models with Two Relaxation Times for Positive/Negative Equilibrium // J. Stat. Phys. — 2010. — Июнь. — Т. 139, № 6. — С. 1090—1143.
26. *Graham R. L., Shipman G. M., Barrett B. W., Castain R. H., Bosilca G., Lumsdaine A.* Open MPI: A high-performance, heterogeneous MPI // Cluster Computing, 2006 IEEE International Conference on. — IEEE, 2006. — С. 1—9.
27. *Graham R. L., Woodall T. S., Squyres J. M.* Open MPI: A flexible high performance MPI // Parallel Processing and Applied Mathematics. — Springer, 2006. — С. 228—239.
28. *Guo Z., Zheng C., Shi B.* Discrete lattice effects on the forcing term in the lattice Boltzmann method // Phys. Rev. E. — 2002. — Т. 65, № 4.
29. *Guodong J., Patzek T., Silin D.* Direct Prediction of the Absolute Permeability of Unconsolidated and Consolidated Reservoir Rock. — (Дата обр. 30.08.2014).
30. *Hammou H., Ginzburg I., Boulerhcha M.* Two-relaxation-times Lattice Boltzmann schemes for solute transport in unsaturated water flow, with a focus on stability // Adv. Water Resour. — 2011. — Т. 34, № 6. — С. 779—793.
31. *Hecht M., Harting J.* Implementation of on-site velocity boundary conditions for D3Q19 lattice Boltzmann simulations // J. Stat. Mech. Theory Exp. — 2010. — Т. 2010, № 01. — P01018.
32. *Hong P.-Y., Huang L.-M., Lin L.-S., Lin C.-A.* Scalable multi-relaxation-time lattice Boltzmann simulations on multi-GPU cluster // Computers & Fluids. — 2014. — Дек.
33. *Huang H., Li Z., Liu S., Lu X.-y.* Shan-and-Chen-type multiphase lattice Boltzmann study of viscous coupling effects for two-phase flow in porous media // Int. J. Numer. Methods Fluids. — 2009. — Т. 61, № 3. — С. 341—354.
34. *Ikeda M., Rao P., Schaefer L.* A thermal multicomponent lattice Boltzmann model // Computers & Fluids. — 2014. — Т. 101. — С. 250—262.
35. *Jacobsen D. A., Thibault J. C., Senocak I.* An MPI-CUDA implementation for massively parallel incompressible flow computations on multi-GPU clusters // 48th AIAA Aerospace Sciences Meeting and Exhibit. Т. 16. — 2010.

36. *Jiang Z., Wu K., Couples G. D., Ma J.* The Impact of Pore Size and Pore Connectivity on Single-Phase Fluid Flow in Porous Media // *Adv. Eng. Mater.* — 2011. — Т. 13, № 3. — С. 208—215.
37. *Kang Q.* Lattice Boltzmann model for crystal growth from supersaturated solution // *Geophys. Res. Lett.* — 2004. — Т. 31, № 21.
38. *Kao P.-H., Yang R.-J.* An investigation into curved and moving boundary treatments in the lattice Boltzmann method // *J. Comput. Phys.* — 2008. — Май. — Т. 227, № 11. — С. 5671—5690.
39. *Khirevich S., Ginzburg I., Tallarek U.* Coarse- and fine-grid numerical behavior of MRT/TRT lattice-Boltzmann schemes in regular and random sphere packings // *J. Comput. Phys.* — 2015. — Янв. — Т. 281. — С. 708—742.
40. *Kutay M. E., Aydilek A. H., Masad E.* Laboratory validation of lattice Boltzmann method for modeling pore-scale flow in granular materials // *Comput. Geotech.* — 2006. — Т. 33, № 8. — С. 381—395.
41. *Lallemand P., Luo L.-S.* Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability // *Phys. Rev. E.* — 2000. — Т. 61, № 6. — С. 6546.
42. *Lallemand P., Luo L.-S.* Lattice Boltzmann method for moving boundaries // *J. Comput. Phys.* — 2003. — Т. 184, № 2. — С. 406—421.
43. *Latva-Kokko M., Rothman D.* Static contact angle in lattice Boltzmann models of immiscible fluids // *Phys. Rev. E.* — 2005. — Т. 72, № 4.
44. *Leclaire S., Reggio M., Trépanier J.-Y.* Isotropic color gradient for simulating very high-density ratios with a two-phase flow lattice Boltzmann model // *Comput. Fluids.* — 2011. — Т. 48, № 1. — С. 98—112.
45. *Li H., Pan C., Miller C.* Pore-scale investigation of viscous coupling effects for two-phase flow in porous media // *Phys. Rev. E.* — 2005. — Т. 72, № 2.
46. *Lim C. Y., Shu C., Niu X. D., Chew Y. T.* Application of lattice Boltzmann method to simulate microchannel flows // *Phys. Fluids.* — 2002. — Т. 14, № 7. — С. 2299—3009.
47. *Liu H., Kang Q., Leonardi C. R., Jones B. D., Schmieschek S., Narváez A., Williams J. R., Valocchi A. J., Harting J.* Multiphase lattice Boltzmann simulations for porous media applications—a review // arXiv preprint arXiv:1404.7523. — 2014.
48. *Liu H., Valocchi A. J., Kang Q.* Three-dimensional lattice Boltzmann model for immiscible two-phase flow simulations // *Phys. Rev. E.* — 2012. — Т. 85, № 4.

49. *Lu F., Song J., Yin F., Zhu X.* Performance evaluation of hybrid programming patterns for large CPU/GPU heterogeneous clusters // Computer Physics Communications. — 2012. — Июнь. — Т. 183, № 6. — С. 1172—1181.
50. *Luo L. S.* Lattice-gas automata and lattice Boltzmann equations for two-dimensional hydrodynamics: дис. ... канд. / Luo Li Shi. — Georgia Institute of Technology, 1993.
51. *Mattila K., Hyväläluoma J., Rossi T., Aspnäs M., Westerholm J.* An efficient swap algorithm for the lattice Boltzmann method // Computer Physics Communications. — 2007. — Т. 176, № 3. — С. 200—210.
52. *McClure J., Prins J., Miller C.* A novel heterogeneous algorithm to simulate multiphase flow in porous media on multicore CPU–GPU systems // Computer Physics Communications. — 2014. — Июль. — Т. 185, № 7. — С. 1865—1874.
53. *Mei R., Luo L. S., Shyy W.* An Accurate Curved Boundary Treatment in the Lattice Boltzmann Method // J. Comput. Phys. — 1999. — № 155. — С. 307—330.
54. *Mohamad A. A.* Lattice Boltzmann Method. — London : Springer London, 2011.
55. *Mohamad A., Kuzmin A.* A critical evaluation of force term in lattice Boltzmann method, natural convection problem // Int. J. Heat Mass Transf. — 2010. — Т. 53, 5-6. — С. 990—996.
56. *Narváez A., Harting J.* Evaluation of pressure boundary conditions for permeability calculations using the lattice-Boltzmann method // ArXiv Prepr. ArXiv10052322. — 2010.
57. *Obrecht C., Kuznik F., Tourancheau B., Roux J.-J.* The TheLMA project: Multi-GPU implementation of the lattice Boltzmann method // Int. J. High Perform. Comput. Appl. — 2011. — Авг. — Т. 25, № 3. — С. 295—303.
58. *Obrecht C., Kuznik F., Tourancheau B., Roux J.-J.* Multi-GPU implementation of the lattice Boltzmann method // Comput. Math. Appl. — 2011. — Т. 65, № 2. — С. 252—261.
59. *Pan C., Luo L.-S., Miller C. T.* An evaluation of lattice Boltzmann schemes for porous medium flow simulation // Comput. Fluids. — 2006. — Т. 35, 8-9. — С. 898—909.
60. *Peng Y., Liao W., Luo L.-S., Wang L.-P.* A Comparative Study of the Lattice Boltzmann and Pseudo-Spectral Methods for Decaying Homogeneous Isotropic Turbulence // 47th AIAA Aerospace Sciences Meeting. — Orlando, Florida, 2009. — (Дата обр. 29.03.2015).
61. *Pereira C. M., Mól A. C., Heimlich A., Moraes S. R., Resende P.* Development and performance analysis of a parallel Monte Carlo neutron transport simulation program for GPU-Cluster using MPI and CUDA technologies // Progress in Nuclear Energy. — 2013. — Май. — Т. 65. — С. 88—94.

62. *Pohl T., Kowarschik M., Wilke J., Iglberger K., Rüde U.* Optimization and profiling of the cache performance of parallel lattice Boltzmann codes // Parallel Processing Letters. — 2003. — Т. 13, № 04. — С. 549—560.
63. *Pore-scale Modelling I. C. C. on Micro-CT images.* — URL: <http://dx.doi.org/10.6084/m9.figshare.1189262>.
64. *Qian Y. H., d'Humières D., Lallemand P.* Lattice BGK models for Navier-Stokes equation // EPL Europhys. Lett. — 1992. — Т. 17, № 6. — С. 479.
65. *Raabe D.* Overview of the lattice Boltzmann method for nano- and microscale fluid dynamics in materials science and engineering // Model. Simul. Mater. Sci. Eng. — 2004. — Т. 12, № 6. — R13—R46.
66. *Rakić P., Milašinović D., Živanov Ѽ., Suvajdžin Z., Nikolić M., Hajduković M.* MPI–CUDA parallelization of a finite-strip program for geometric nonlinear analysis: A hybrid approach // Advances in Engineering Software. — 2011. — Май. — Т. 42, № 5. — С. 273—285.
67. *Schönherr M., Geier M., Krafczyk M.* 3D GPGPU LBM implementation on non-uniform grids // (nternational Conference on Parallel Computational Fluid Dynamics 2011, Parallel CFD 2011). — 2011.
68. *Shan X., Chen H.* Simulation of nonideal gases and liquid-gas phase transitions by the lattice Boltzmann equation // Phys. Rev. E. — 1994. — Т. 49, № 4.
69. *Shao J. Y., Shu C., Huang H. B., Chew Y. T.* Free-energy-based lattice Boltzmann model for the simulation of multiphase flows with density contrast // Phys. Rev. E. — 201419. — марта. — Т. 89, № 3.
70. *Succi S.* The Lattice-Boltzmann Equation. — Oxford university press, Oxford, 2001.
71. *Sukop M. C., Thorne D. T.* Lattice Boltzmann modeling an introduction for geoscientists and engineers. — Berlin; New York : Springer, 2006.
72. *Swift M. R., Osborn W. R., Yeomans J. M.* Lattice Boltzmann simulation of nonideal fluids // Physical Review Letters. — 1995. — Т. 75, № 5. — С. 830. — (Дата обр. 24.03.2015).
73. *Vetter J.* Contemporary High Performance Computing. — CRC, 2013.
74. *Wittmann M., Zeiser T., Hager G., Wellein G.* Comparison of different propagation steps for lattice Boltzmann methods // Comput. Math. Appl. — 2013. — Т. 65, № 6. — С. 924—935.
75. *Wolf-Gladrow D.* Lattice-Gas Cellular Automata and LAttice Boltzmann Models. — Springer Berlin, 2000.
76. *Xian W., Takayuki A.* Multi-GPU performance of incompressible flow computation by lattice Boltzmann method on GPU cluster // Parallel Comput. — 2011. — Февр.

77. Yang C.-T., Huang C.-L., Lin C.-F. Hybrid CUDA, OpenMP, and MPI parallel programming on multicore GPU clusters // Computer Physics Communications. — 2011. — Янв. — Т. 182, № 1. — С. 266—269.
78. Yoshida H., Nagaoka M. Multiple-relaxation-time lattice Boltzmann model for the convection and anisotropic diffusion equation // J. Comput. Phys. — 2010. — Окт. — Т. 229, № 20. — С. 7774—7795.
79. Ziegler D. P. Boundary conditions for lattice Boltzmann simulations // J. Stat. Phys. — 1993. — Т. 71, 5-6. — С. 1171—1177.
80. Zou Q., He X. On pressure and velocity boundary conditions for the lattice Boltzmann BGK model // Phys. Fluids 1994-Present. — 1997. — Т. 9, № 6. — С. 1591—1598.
81. Бикулов Д., Сенин Д. Реализация метода решеточных уравнений Больмана без хранимых функций распределения для GPU // Вычислительные методы и программирование: Новые вычислительные технологии (Электронный научный журнал). — 2013. — Т. 14. — С. 370—374.
82. Бикулов Д., Сенин Д., Демин Д., Дмитриев А., Грачев Н. Реализация метода решеточных уравнений Больцмана для расчетов на GPU-клUSTERе // Вычислительные методы и программирование: Новые вычислительные технологии (Электронный научный журнал). — 2012. — Т. 13. — С. 13—19.
83. Боресков А., Харlamov А. Основы работы с технологией CUDA. — Москва : ДМК Пресс, 2010.
84. Грачев Н. Е., Дмитриев А. В., Сенин Д. С. Моделирование динамики газа при помощи решеточного метода Больцмана // Вычислительные Методы И Программирование. — 2011. — Т. 12, № 1. — С. 227—231.
85. Двухкомпонентное течение через пористую среду (видео). — URL: <https://vimeo.com/90556223>.
86. Джораев А. Гибридные вычислительные системы на основе GPU для задач биоинформатики // Компьютерные исследования и моделирование. — 2010. — Т. 2, № 2. — С. 163—167.
87. Додулад О. И., Клосс Ю. Ю., Рябченков В. В., Черемисин Ф. Г. Система программных модулей для вычисления интеграла столкновений Больцмана // Вычислительные методы и программирование. — 2011. — Т. 12, № 1. — С. 40—47.
88. Захаров А., Сенин Д., Грачев Е. Моделирование течений методом решеточных уравнений Больцмана со многими временами релаксации // Вычислительные методы и программирование. — 2014. — Т. 15, № 4. — С. 644—657.

89. Клосс Ю. Ю., Мартынов Д. В. Решение кинетического уравнения Больцмана с помощью тетраэдрических сеток на кластерной архитектуре // вычислительные методы и программирование. — 2012. — Т. 13. — С. 91.
90. Кривовичев Г. В. Исследование устойчивости явных конечно-разностных решеточных кинетических схем Больцмана // Вычислительные Методы И Программирование. — 2012. — Т. 13. — С. 332—340.
91. Кривовичев Г. В. О расчете течений вязкой жидкости методом решеточных уравнений Больцмана // Компьютерные исследования и моделирование. — 2013. — Т. 5, № 2. — С. 165—178.
92. Кривовичев Г. В. Модифицированный вариант метода решеточных уравнений Больцмана для расчета течений вязкой несжимаемой жидкости // Компьютерные исследования и моделирование. — 2014. — Т. 6, № 3. — С. 364—381.
93. Кривовичев Г. Анализ устойчивости решеточных схем Больцмана для решения уравнения диффузии // вычислительные методы и программирование. — 2013. — Т. 14. — С. 175.
94. Кривовичев Г. О решеточной схеме Больцмана для расчетов на неструктурированных сетках // вычислительные методы и программирование. — 2013. — Т. 14. — С. 525.
95. Куперштог А. Л. Трехмерное моделирование двухфазных систем типа жидкость–пар методом решеточных уравнений Больцмана на GPU // Вычислительные Методы И Программирование. — 2012. — Т. 13. — С. 130—138.
96. Куперштог А. Метод решеточных уравнений Больцмана для моделирования двухфазных систем типа жидкость–пар // Сборник научных статей «Современная наука». — 2010. — Т. 2010, № 2. — С. 56—63.
97. Макаров С., Бикулов Д., Грачев Е., Кочетов А., Губский Л. Компьютерное моделирование развития ишемического инсульта с помощью технологии CUDA // Труды международной суперкомпьютерной конференции "Научный сервис в сети Интернет: все грани параллелизма" (23 - 28 сентября 2013 г., г. Новороссийск). — Изд-во МГУ Москва, 2013.
98. Список Green500, редакция за ноябрь 2014 года. — URL: <http://www.green500.org/lists/green201411>.
99. Список Top500, редакция за ноябрь 2014 года. — URL: <http://www.top500.org/lists/2014/11/>.
100. Суперкомпьютерный комплекс Московского Университета. — URL: <http://parallel.ru/cluster>.