

Автономная некоммерческая  
образовательная организация высшего образования  
**«НАУЧНО-ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «СИРИУС»**

Научный центр информационных технологий и искусственного интеллекта

направление «Математическое моделирование процессов и материалов»

К ЗАЩИТЕ ДОПУСТИТЬ  
Руководитель направления  
«Математическое моделирование  
процессов и материалов»

К.Т.Н., Д.Э.Н.,  
(ученая степень, звание)

\_\_\_\_\_ М.В. Ширяев  
« \_\_\_\_ » \_\_\_\_\_ 2025 г.

**«ВЕРИФИКАЦИЯ МОДЕЛИ ТРЕХФАЗНОГО  
НЕСМЕШИВАЮЩЕГОСЯ ТЕЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ  
РЕШЕТОК БОЛЬЦМАНА»**

Магистерская диссертация  
по направлению подготовки 01.04.02, «Прикладная математика и  
информатика»  
направленность (профиль) «Математическое моделирование процессов и  
материалов»

Студент гр. М01ММ-23  
\_\_\_\_\_ Н.Э.Карпинский  
« \_\_\_\_ » \_\_\_\_\_ 2025 г.

Руководитель  
профессор, к.ф.-м.н., доктор  
философии в области  
математического моделирования  
(должность, ученая степень, звание)

\_\_\_\_\_ С.Ю. Малясов  
« \_\_\_\_ » \_\_\_\_\_ 2025 г.

Автономная некоммерческая  
образовательная организация высшего образования  
**«НАУЧНО-ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «СИРИУС»**

Научный центр информационных технологий и искусственного интеллекта  
направление «Математическое моделирование процессов и материалов»

УТВЕРДИТЬ

Руководитель направления  
«Математическое моделирование  
процессов и материалов»

К.Т.Н., Д.Э.Н.,  
(ученая степень, звание)

\_\_\_\_\_ М.В. Ширяев  
« \_\_\_\_ » \_\_\_\_\_ 2025 г.

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

на выполнение магистерской диссертации  
по направлению подготовки 01.04.02, «Прикладная математика и  
информатика»  
направленность (профиль) «Математическое моделирование процессов и  
материалов»

**Карпинский Никита Эдуардович**

1. Тема: «Верификация модели трехфазного несмешивающегося течения с использованием решеток Больцмана».
2. Цель: верификации трёхфазной модели несмешивающегося течения, основанной на методе решеточных уравнений Больцмана (LBM).
3. Задачи:
  - Изучение статей для создания набора тестов для последующей верификации модели.
  - Анализ тестов, сортировка по уровням в соответствии с методологией верификации по Оберкампфу.
  - Построение тестовых сценариев в библиотеке Palabos.
  - Верификация написанных сценариев в соответствии с методологией Оберкампфа.

4. Рабочий график (план) выполнения магистерской диссертации:

№	Перечень заданий	Сроки выполнения
1.	Написание литературного обзора	09.01.2025-14.03.2025
2.	Изложение материалов и методов	17.03.2025-30.04.2025
3.	Представление полученных результатов	02.05.2025-20.05.2025

Дата выдачи: «\_\_\_» \_\_\_\_\_ 20\_\_\_ г.

Руководитель:

\_\_\_\_\_ С.Ю. Малясов

Задание принял к исполнению:

Студент группы \_\_\_\_\_

\_\_\_\_\_ Н.Э. Карпинский  
(подпись) ФИО

«\_\_\_» \_\_\_\_\_ 20\_\_\_ г.

## Реферат

Выпускная квалификационная работа, 102 страницы, 5 рисунков, 8 таблиц, 13 источников, 3 приложения.

МЕТОД РЕШЕТОК БОЛЬЦМАНА, ТРЕХФАЗНОЕ ТЕЧЕНИЕ, НЕСМЕШИВАЮЩИЕСЯ ЖИДКОСТИ, ПСЕВДОПОТЕНЦИАЛЬНАЯ МОДЕЛЬ, ШАН-ЧЕН, КАПИЛЯРНЫЕ ЯВЛЕНИЯ, ВЕРИФИКАЦИЯ МОДЕЛИ, МНОГОУРОВНЕВАЯ ВЕРИФИКАЦИЯ

В магистерской диссертации рассматривается задача верификации численной модели трёхфазного несмешивающегося течения, реализованной на основе метода решёток Больцмана в библиотеке Palabos.

Актуальность работы обусловлена необходимостью моделирования многофазных процессов, протекающих в пористых средах и капиллярных структурах, что имеет широкое применение в таких областях, как нефтегазовая промышленность и гидрогеология.

Целью исследования является отбор, реализация и численная проверка верификационных тестов для трёхфазной модели на основе псевдопотенциального подхода Шан–Чена (Shan–Chen). В основе подхода лежит локальное взаимодействие между фазами через псевдопотенциальные силы, что позволяет моделировать поверхностное натяжение, смачивание и поведение линий пересечения трех фаз.

В работе проведён систематический анализ современных публикаций, содержащих верификационные тесты для трёхфазных моделей. Отобранные тесты классифицированы по методологии многоуровневой верификации Оберкампа, предполагающей последовательную проверку корректности модели от элементарных уравнений до воспроизведения сложных физических сценариев. Реализация проведена на основе библиотеки Palabos, в которой были адаптированы и дополнены необходимые структуры и вычислительные процедуры для трёхфазных взаимодействий.

В рамках работы реализованы три теста: «Жидкая линза» (Liquid Lens), «Вложенная капля» (Nested Droplet) и тест на вязкость. Каждый из них соответствует определённому уровню верификации и охватывает различные физические аспекты – равновесие трёхфазного интерфейса, проверку закона Лапласа на двух вложенных границах и динамику спонтанного распада флюидной смеси. Численные результаты подтверждают корректность реализованной модели и её способность воспроизводить основные межфазные явления.

Практическая значимость работы заключается в возможности использования реализованной модели для моделирования капиллярных процессов и вытеснения флюидов в пористых средах. Разработанный численный прототип может быть применён как в академических исследованиях, так и при построении инженерных симуляторов в задачах энергетики и геофизики.

## The Abstract

Final qualification thesis, 102 pages, 5 figures, 8 tables, 13 references, 3 appendices.

LATTICE BOLTZMANN METHOD, THREE-PHASE FLOW, IMMISCIBLE FLUIDS, PSEUDOPOTENTIAL MODEL, SHAN–CHEN, CAPILLARY PHENOMENA, MODEL VERIFICATION, MULTILEVEL VERIFICATION

This master’s thesis addresses the verification of a numerical model for three-phase immiscible flow implemented using the lattice Boltzmann method in the Palabos computational framework.

The relevance of the study is due to the growing need for accurate modeling of multiphase processes occurring in porous media and capillary structures, which are widely encountered in fields such as petroleum engineering and hydrogeology.

The aim of the study is the selection, implementation, and numerical verification of

test cases for a three-phase model based on the pseudopotential approach by Shan–Chen. This approach is built upon the concept of local interphase interactions through pseudopotential forces, enabling the modeling of surface tension, wettability, and triple-line behavior without requiring additional governing equations.

The thesis includes a systematic review of current scientific publications containing verification tests for three-phase models. Selected tests were classified according to the Oberkampf methodology of multilevel verification, which prescribes a step-by-step evaluation of the model’s correctness — from fundamental equations to the reproduction of complex physical phenomena. Implementation was carried out using the Palabos library, in which the necessary data structures and numerical routines for three-phase interactions were adapted and extended.

Three verification tests were implemented as part of the work: the Liquid Lens, the Nested Droplet, and a viscosity test. Each of these corresponds to a **specific verification** level and covers different physical aspects — equilibrium of a three-phase interface, validation of the Laplace law across two nested boundaries, and the dynamics of spontaneous phase separation in fluid mixtures. The numerical results confirm the correctness of the implemented model and its capability to capture essential interfacial phenomena.

The practical significance of the work lies in the potential application of the developed model for simulating capillary processes and fluid displacement in porous structures. The resulting numerical prototype can be used both in academic research and in the development of engineering simulators for applications in energy and geophysical sciences.

## Сокращения, обозначения, термины и определения

В настоящей работе используются следующие сокращения:

- LBM — метод решёток Больцмана (Lattice Boltzmann Method)
- BGK — модель Бхатнагара–Гросса–Крука (Bhatnagar–Gross–Krook model)
- D2Q9 — двумерная решётка с девятью дискретными скоростями
- D3Q19 — трёхмерная решётка с девятнадцатью дискретными скоростями
- D3Q27 — трёхмерная решётка с двадцатью семью дискретными скоростями
- SC — модель Шан–Чен (Shan–Chen model)
- VOF — метод объема жидкости (Volume of Fluid method)
- CNE — уравнение Кана–Хиллиарда (Cahn–Hilliard Equation)
- ACE — уравнение Аллена–Кана (Allen–Cahn Equation)
- EDM — метод точной разности (Exact Difference Method)

# ОГЛАВЛЕНИЕ

Введение.....	10
<b>1 обзор литературы .....</b>	<b>13</b>
<b>1.1 Метод решеток Больцмана (LBM) на решетке D2Q9.....</b>	<b>13</b>
1.1.1 Дискретное кинетическое уравнение .....	13
1.1.2 Геометрия решетки D2Q9.....	13
1.1.3 Модель столкновений BGK .....	14
1.1.4 Функция равновесия.....	14
1.1.5 Связь с уравнениями Навье-Стокса .....	15
1.1.6 Алгоритм расчета .....	15
1.1.7 Расширение до решетки D3Q19 .....	15
1.1.8 Расширение до решетки D3Q27 .....	16
<b>1.2 Многофазные методы в LBM.....</b>	<b>17</b>
1.2.1 Метод цветового градиента (Color-gradient) .....	17
1.2.2 Метод свободной энергии (Free-energy) .....	22
1.2.3 Метод фазового поля (Phase-field) .....	27
1.2.4 Метод объема жидкости (LB-VOF).....	33
1.2.5 Метод Шан-Чен (Shan-Chen) .....	38
1.2.6 Анализ методов и их сравнение .....	44
<b>1.3 Методология верификации по Оберкампфу.....</b>	<b>46</b>
1.3.1 Представление работы Оберкампфа .....	46
1.3.2 Ключевые определения .....	47
1.3.3 Общие принципы верификации .....	47
1.3.4 Уровни верификационных тестов.....	48
<b>1.4 Статьи с тестами.....</b>	<b>49</b>
1.4.1 Трёхфазная модель решёток Больцмана на основе свободной энергии с настраиваемыми поверхностными натяжениями и контактными углами.....	49
1.4.2 Моделирование трёхфазного реактивного течения на поровом уровне в случайной пористой среде методом решёток Больцмана .....	50
1.4.3 Численное прогнозирование относительной проницаемости трёх фаз в пористой среде на поровом уровне методом решёток Больцмана.....	52
1.4.4 Механизмы вытеснения одной жидкости другой в сети капиллярных каналов .....	53
1.4.5 Трёхмерный решатель потоков на основе метода решёток Больцмана для трёхфазного / трёхкомпонентного течения.....	54
1.4.6 Исследование поведения отрыва фазы при самопроизвольной имбибии в трёхмерной модели порового горлышка методом решёток Больцмана с псевдопотенциалом.....	56

<b>2</b>	<b>материалы и методы</b>	<b>58</b>
<b>2.1</b>	<b>Отобранные тесты</b>	<b>58</b>
2.1.1	Одиночная капля (Young-Laplace)	58
2.1.2	Трёхмерная капля у смачиваемой поверхности	58
2.1.3	Трёхфазная линза (Liquid Lens)	59
2.1.4	Фазовое разделение трёх компонентов	59
2.1.5	Слоистое течение в трехмерном канале	60
2.1.6	Вложенная капля (Nested Droplet)	60
2.1.7	Подъём пузыря в трёхфазной системе	61
2.1.8	Вытеснение линейной нефтяной перемычки	61
2.1.9	Поршневое вытеснение (Piston-type)	62
2.1.10	Отрыв (Snap-off) в поровом горлышке	62
2.1.11	Последовательное вытеснение газ–нефть–вода	63
<b>2.2</b>	<b>Постановка тестов в Palabos</b>	<b>64</b>
2.2.1	Тест на вязкость	64
2.2.2	Тест «Жидкая линза» («Liquid Lens»)	66
2.2.3	Тест «Вложенная капля» («Nested Droplet»)	68
<b>3</b>	<b>результаты</b>	<b>72</b>
<b>3.1</b>	<b>Результаты теста вязкости</b>	<b>72</b>
3.1.1	Вывод	72
3.1.2	Результаты расчетов	72
<b>3.2</b>	<b>Результаты теста «Жидкая линза»</b>	<b>72</b>
3.2.1	Результаты теста	72
3.2.2	Вывод	73
<b>3.3</b>	<b>Результаты теста «Вложенная капля»</b>	<b>75</b>
3.3.1	Результаты теста	75
3.3.2	Вывод	76
	<b>Заключение</b>	<b>78</b>
	<b>Список использованных источников</b>	<b>80</b>
	<b>Приложение А. Код теста на вязкость</b>	<b>82</b>
	<b>Приложение Б. Код теста «Жидкая линза»</b>	<b>87</b>
	<b>Приложение В. Код теста «Вложенная капля»</b>	<b>103</b>

## Введение

Моделирование многофазных течений в пористых средах и капиллярных структурах представляет собой одну из наиболее актуальных задач вычислительной гидродинамики, имеющую широкое применение в таких отраслях, как нефтегазовая промышленность и гидрогеология. Особый интерес представляет исследование трёхфазных систем, в которых одновременно взаимодействуют три несмешивающиеся жидкости – в частности, вода, нефть и газ. Корректное описание таких систем требует учёта сложных интерфейсных явлений, включая поверхностное натяжение, смачивание, а также динамику фазовых границ в условиях резких перепадов плотностей и вязкостей.

Метод решёток Больцмана (LBM), основан на дискретизации кинетического уравнения Больцмана. По причине относительной простоты реализации численного решения уравнений методом параллельных вычислений и гибкости метода в описании границ фаз, LBM является востребованным инструментом для численного изучения капиллярных и поверхностных эффектов в сложных геометриях. Одной из ключевых задач при разработке модели несмешивающегося течения фаз является ее верификация. Верификация модели — это установление степени соответствия между численным решением и эталонными результатами, полученных в ходе аналитического решения, экспериментальных данных или представлениях о физике процессов. Отсутствие надлежащей верификации может привести к некорректным выводам при интерпретации результатов моделирования.

Одна из методологий верификации была предложена У. Л. Оберкампом и Т. Трукано в контексте инженерного моделирования. Эта методология позволяет структурировать процесс верификации, разбивая его на последовательные уровни — от элементарных тестов до комплексных

сценариев. Такая многоуровневая структура способствует объективной оценке точности и устойчивости численного метода.

Целью настоящей работы является подбор тестов для верификации трёхфазной модели несмешивающегося течения, реализованной на основе метода решёток Больцмана в библиотеке Palabos, с последующей реализацией тест-кейсов и проведением верификации по методологии Оберкампфа. Для достижения данной цели в ходе работы были поставлены и решены следующие задачи:

- изучение научных публикаций, содержащих верификационные тесты для трёхфазных моделей;
- анализ и отбор релевантных тестов с учётом критериев воспроизводимости, физической достоверности и соответствия задачам верификации;
- классификация тестов по уровням согласно методологии Оберкампфа;
- численная реализация отобранных сценариев в библиотеке Palabos;
- проверка полученных результатов на соответствие ожидаемым эффектам и их анализ.

Научная значимость работы заключается в применении формализованного подхода к верификации сложной трёхфазной модели, что способствует повышению надёжности и воспроизводимости численных исследований. Прикладное значение определяется возможностью использования результатов в инженерных расчётах, связанных с проектированием и оптимизацией процессов многофазного вытеснения, фильтрации и капиллярного захвата. Используемые методы и постановки могут быть адаптированы для более сложных задач, включая моделирование реактивных потоков и взаимодействие с пористыми структурами.



# 1 ОБЗОР ЛИТЕРАТУРЫ

## 1.1 Метод решеток Больцмана (LBM) на решетке D2Q9

### 1.1.1 Дискретное кинетическое уравнение

Метод решёток Больцмана (LBM) основан на решении дискретного кинетического уравнения, описывающего эволюцию функции распределения  $f_i(x, t)$ , определённой вдоль дискретных направлений решётки:

$$f_i(x + e_i \Delta t, t + \Delta t) = f_i(x, t) + \Omega_i(f(x, t)), \#(1)$$

где  $f_i$  – функция распределения,

$e_i$  – дискретные скорости,

$\Omega_i$  – оператор столкновения.

Макроскопические параметры:

$$\rho = \sum_i f_i, \rho u = \sum_i f_i e_i, \#(2)$$

### 1.1.2 Геометрия решетки D2Q9

Модель D2Q9 включает 9 дискретных направлений, включая нулевое (покой), ортогональные и диагональные направления. Весовые коэффициенты  $\omega_i$  подбираются таким образом, чтобы обеспечить изотропию и корректное восстановление макроскопических уравнений гидродинамики.

$$e_i = \begin{cases} (0,0), & i = 0; \\ (\pm 1,0), (0, \pm 1), & i = 1..4; \\ (\pm 1, \pm 1) & i = 5..8; \end{cases}$$

Весовые коэффициенты скоростей:

$$\omega_0 = \frac{4}{9}, \omega_{1..4} = \frac{1}{9}, \omega_{5..8} = \frac{1}{36}.$$

### 1.1.3 Модель столкновений BGK

В модели Бхатнагара–Гросса–Крука (BGK) столкновения аппроксимируются возвращением к локальному равновесию с заданным временем релаксации  $\tau$ :

$$\Omega_i = -\frac{1}{\tau}(f_i - f_i^{eq}), \quad \#(3)$$

где  $f_i^{eq}$  – локальная равновесная функция распределения,

$\tau$  – время релаксации.

Итоговое уравнение метода имеет вид:

$$f_i(x + e_i \Delta t, t + \Delta t) = f_i(x, t) - \frac{\Delta t}{\tau}(f_i - f_i^{eq}). \#(4)$$

### 1.1.4 Функция равновесия

Для численного восстановления уравнений Навье–Стокса функция распределения аппроксимируется вблизи равновесия. При малых числах Маха используется разложение функции Максвелла–Больцмана по малым скоростям:

$$f_i^{eq} = \omega_i \rho \left[ 1 + \frac{e_i u}{c_s^2} + \frac{(e_i u)^2}{2c_s^4} - \frac{u^2}{2c_s^2} \right], \#(5)$$

где  $c_s = \frac{1}{\sqrt{3}}$  – скорость звука в решетке.

### 1.1.5 Связь с уравнениями Навье-Стокса

Связь между микроскопической динамикой LBM и макроскопическими уравнениями Навье–Стокса устанавливается с помощью многошкального разложения Чепмена–Энскога. Это позволяет выразить тензор напряжений  $\Pi_{\alpha\beta}$  как:

$$\Pi_{\alpha\beta} = \sum_i e_{i\alpha} e_{i\beta} \left[ f_i^{eq} + \left(1 - \frac{1}{2\tau}\right) f_i^{(1)} \right], \#(6)$$

что приводит к определению кинематической вязкости:

$$\nu = c_s^2 \left( \tau - \frac{1}{2} \right) \Delta t. \#(7)$$

### 1.1.6 Алгоритм расчета

На каждом временном шаге метод LBM выполняет два этапа:

- Столкновение (обновление функции распределения):

$$f_i^* = f_i - \frac{\Delta t}{\tau} (f_i - f_i^{eq}), \#(8)$$

- Течение (перенос по направлению дискретной скорости):

$$f_i^*(x) \rightarrow f_i(x + e_i \Delta t). \#(9)$$

### 1.1.7 Расширение до решетки D3Q19

Решётка D3Q19, в сравнении с D2Q9 - очевидное расширение решетки с двумерного в трехмерное пространство, данная решетка является компромиссом между вычислительной стоимостью и изотропией, хорошо подходит для большинства 3D задач.

Множество скоростей. Всего 19 векторов: один покоя, шесть «линейных» и двенадцать «диагональных».

$$e_i = \begin{cases} (0,0,0), & i = 0; \\ (\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1), & i = 1 \dots 6; \\ (\pm 1, \pm 1, 0), (\pm 1, 0, \pm 1), (0, \pm 1, \pm 1), & i = 7 \dots 18. \end{cases} \#(10)$$

Весовые коэффициенты:

$$w_0 = \frac{1}{3}, w_{1..6} = \frac{1}{18}, w_{7..18} = \frac{1}{36}.$$

Пример расчёта макроскопических величин:

$$\rho = \sum_{i=0}^{18} f_i, \rho \mathbf{u} = \sum_{i=0}^{18} f_i \mathbf{e}_i \#(11)$$

### 1.1.8 Расширение до решетки D3Q27

D3Q27 обеспечивает лучшую изотропию, в то время как D3Q19 — компромисс между численной эффективностью и физической точностью.

Множество скоростей:

$$\mathbf{e}_i = (e_{ix}, e_{iy}, e_{iz}), e_{i\alpha} \in \{-1, 0, 1\}, i = 0 \dots 26.$$

Скорости и их веса для решетки D3Q27:

Таблица 1-Вектора скорости и их веса

Вектор скорости	Кол-во векторов в наборе	Вес скорости $w_i$
(0,0,0)	1	8/27
(±1,0,0)	6	2/27
(±1, ±1, 0)	12	1/54
(±1, ±1, ±1)	8	1/216

## 1.2 Многофазные методы в LBM

### 1.2.1 Метод цветового градиента (Color-gradient)

#### 1.2.1.1 Функция распределения

В многофазной модели с цветовым градиентом каждая фаза описывается собственной функцией распределения  $f_i^{(k)}$ , эволюция которой подчиняется уравнению Больцмана с зависимым от фазы временем релаксации.

Уравнение (12) показывает, как после переноса вдоль дискретной скорости  $e_i$  распределение частиц  $f_i^{(k)}$  возвращается к локальному равновесию за время  $\tau^{(k)}$ , сохраняя массу и импульс [1]:

$$f_i^{(k)}(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i^{(k)}(\mathbf{x}, t) - \frac{\Delta t}{\tau^{(k)}} [f_i^{(k)} - f_i^{(k),eq}], \#(12)$$

где  $f_i^{(k)}$  – распределение «частиц» цветной фазы  $k$  вдоль направления  $i$ ,

$\mathbf{e}_i$  – дискретные скорости,

$\Delta t$  – шаг времени,

$\tau^{(k)}$  – время релаксации, отражающее вязкость фазы,

$f_i^{(k),eq}$  – равновесное распределение;

#### 1.2.1.2 Макроскопические поля

Чтобы связать мезоуровень с гидродинамикой, берём нулевой и первый моменты распределения. Они дают локальные плотности фаз и барицентрическую скорость смеси, необходимую для вычисления сил и давления.

Макроскопическая плотность  $\rho$  и барицентрическая скорость  $u$  определяются как нулевой и первый моменты распределения по скоростям:

$$\rho = \sum_i f_i^{(k)}, \rho u = \sum_i f_i^{(k)} e_i \#(13)$$

### 1.2.1.3 Равновесное распределение

Разложение равновесного распределения до второго порядка по малой скорости гарантирует аппроксимацию уравнений Навье–Стокса с точностью порядка  $O(u^2)$ , что необходимо для корректного воспроизведения макроскопической гидродинамики. Весовые коэффициенты  $w_i$ , зависящие от симметрии конкретной решётки (прим. D2Q9), определяют вклад каждой дискретной скорости  $\mathbf{e}_i$  и выполняют ключевую роль в обеспечении изотропности межфазного натяжения — одного из важнейших факторов при моделировании интерфейсных явлений.

$$f_i^{(k),eq} = w_i \rho^{(k)} \left[ 1 + \frac{3}{c_s^2} \mathbf{e}_i \cdot \mathbf{u} + \frac{9}{2c_s^4} (\mathbf{e}_i \cdot \mathbf{u})^2 - \frac{3}{2c_s^2} |\mathbf{u}|^2 \right], \#(14)$$

где  $w_i$  — весовые коэффициенты,

$c_s$  — скорость звука в решётке.

Разложение до второго порядка по скорости  $u$  позволяет сохранить точность восстановления уравнений Навье–Стокса в рамках численного моделирования.

### 1.2.1.4 Расчет градиента цвета

Граница между жидкостями определяется вектором, называемым вектором цвета  $\mathbf{C}$ , который направлен перпендикулярно к межфазному переходу. Градиент цвета  $\mathbf{C}$  служит для определения положения и ориентации интерфейса между фазами. Этот вектор вычисляется как градиент скалярной функции, описывающей распределение фаз. Точность его расчёта принципиально важна, поскольку от неё зависит симметричность межфазных сил и уровень нежелательных численных возмущений, известных как паразитные токи.

Формула Rothman–Keller:

$$\mathbf{C}(\mathbf{x}) = \sum_i \mathbf{e}_i [\rho^{(r)}(\mathbf{x} + \mathbf{e}_i) - \rho^{(b)}(\mathbf{x} + \mathbf{e}_i)]. \#(15)$$

Изотропная формула:

$$\mathbf{C}(\mathbf{x}) = \sum_i \mathbf{e}_i \xi_i [\rho^{(r)}(\mathbf{x} + \mathbf{e}_i) - \rho^{(b)}(\mathbf{x} + \mathbf{e}_i)]. \#(16)$$

$$\xi_i = \begin{cases} \frac{1}{3}, & i = 1 \dots 4 \\ \frac{1}{12}, & i = 5 \dots 8 \\ 0, & i = 0 \end{cases} \#(17)$$

$\mathbf{C}$  – градиент цвета, его нормаль задаёт направление кривизны, а  $\xi_i$  –корректирует вес, убирая анизотропию четвёртого порядка.

Коллизия BGK. После вычисления  $f_i^{(k),eq}$  стандартный BGK-оператор релаксирует каждую краску отдельно, сохраняя внутреннюю вязкость:

$$(\Omega_i^{(k)})^{(1)} = -\frac{1}{\tau^{(k)}} [f_i^{(k)} - f_i^{(k),eq}] \#(18)$$

где  $\Omega_i^{(1)}$  – однокомпонентная релаксация, не включающая межфазные силы,  $\tau^{(k)}$  – выбирается так, чтобы  $\nu(k) = c_s^2 (\tau^k - 1/2)\Delta t$ .

#### 1.2.1.5 Поверхностное натяжение

Для моделирования поверхностного натяжения вводится добавочный вклад в уравнение на основе величины градиента цвета. Это создаёт результирующее давление, пропорциональное кривизне интерфейса, что соответствует уравнению Юнга–Лапласа. Таким образом обеспечивается

стабилизация межфазной границы и формирование корректного давления.

$$(\Omega_i^{(k)})^{(2)} = \frac{A_k |c|}{2 c_s^2} \left[ w_i \frac{(\mathbf{e}_i \cdot \mathbf{c})^2}{c \cdot c} - B_i \right], \#(19)$$

где  $A_k$  – коэффициент, напрямую задающий межфазное натяжение  $\sigma$ ,

$B_i$  – набор постоянных, обеспечивающих изотропный тензор,

$\tau$  – интерполированное время релаксации.

$$A = \begin{cases} \frac{3}{4} \frac{\sigma}{(\rho^{(r)} + \rho^{(b)})\tau}, & \text{градиент (4a),} \\ \frac{9}{2} \frac{\sigma}{(\rho^{(r)} + \rho^{(b)})\tau}, & \text{градиент (4b).} \end{cases} \#(20)$$

#### 1.2.1.6 Перекрашивание

Перекрашивание реализуется через модификацию распределений вблизи интерфейса на основе направления градиента цвета. Используемая функция перекраски ориентирована вдоль нормали к границе и управляется параметром  $\beta$ , определяющим толщину переходной зоны между фазами и устойчивость интерфейса.

#### 1.2.1.7 Интерполяция вязкости

В случае, когда вязкости фаз существенно различаются, используется специальная интерполяционная схема, в которой релаксационное время в ячейке рассчитывается в зависимости от её фазовой принадлежности («цвета» ячейки). Это позволяет задать плавный переход вязкости вблизи границы раздела и обеспечивает корректное моделирование сдвиговых напряжений вблизи интерфейса.

$$P = \frac{\rho^{(r)} - \rho^{(b)}}{\rho^{(r)} + \rho^{(b)}} \#(21)$$

$$\frac{1}{\tau(\mathbf{x}) - \frac{1}{2}} = \frac{1 + P}{2\left(\tau_r - \frac{1}{2}\right)} + \frac{1 - P}{2\left(\tau_b - \frac{1}{2}\right)}, \#(21)$$

где  $P$  – скалярное поле, линейно переходящее от  $+1$  (*red*) до  $1$  (*blue*),  
 $\tau(\mathbf{x})$  – гармоническая интерполяция.

Интерполяция времени релаксации осуществляется с использованием гармонического среднего, зависящего от параметра цвета  $P$ , отражающего локальное соотношение плотностей фаз. Это обеспечивает физически непрерывный переход вязкости.

#### 1.2.1.8 Силовые воздействия

Метод градиента цвета допускает введение внешнего ускорения (прим. силы тяжести) двумя математически эквивалентными способами: либо путём модификации равновесной функции распределения за счёт добавления скорости с учётом силы, либо через явное включение источниковой функции  $S_i^{(k)}$  в уравнение эволюции. Оба подхода направлены на корректное воспроизведение движения жидкости под действием внешних сил.

Влияние внешних сил может быть учтено либо через сдвиг скорости  $u \rightarrow u + \tau a$  в функции равновесия, либо через добавление источникового члена  $S_i^{(k)}$  в уравнение. Оба подхода обеспечивают корректное воспроизведение гидростатического равновесия и ускорений.

$$S_i^{(k)} = \frac{\omega_i \rho^{(k)} (e_i a^{(k)})}{c_s^2} \#(22)$$

## 1.2.2 Метод свободной энергии (*Free-energy*)

### 1.2.2.1 Свободная энергия системы

Свободная энергия системы — это скалярная функциональная величина, характеризующая термодинамическое состояние многофазной жидкости. Её аналитическая форма определяет устойчивые конфигурации плотностей (или концентраций) всех компонентов смеси [2]. Минимизация свободной энергии приводит к появлению химических потенциалов и межфазных сил, которые затем используются в уравнениях гидродинамики и фазового перехода, таких как уравнение Навье–Стокса и уравнение Кона–Хиллиарда.

Метод свободной энергии использует функционал свободной энергии, зависящий от плотности и концентраций фаз, минимизация которого приводит к устойчивому распределению фаз. Общая форма свободной энергии включает объемную и межфазную части:

$$F = \int_{\Omega} [f_B(\rho, C_{l1}, C_{l2}) + f_I(\nabla\rho, \nabla C_{l1}, \nabla C_{l2})] dV, \#(23)$$

где  $\Omega$  — объём расчётной области,

$f_B$  — однородная часть, формирует три минимума (газ и две жидкости),

$f_I$  — межфазная энергия, придающая интерфейсам конечную толщину и натяжение.

Минимизация этого функционала приводит к выводу выражений для химических потенциалов, используемых в уравнениях движения и фазовой эволюции.

### 1.2.2.2 Объёмная часть

Вклад свободной энергии, связанный с внутренним состоянием вещества, учитывает предпочтительные значения плотностей фаз и описывается через потенциал с несколькими минимумами. Такой потенциал моделирует стремление каждой фазы к своей равновесной плотности в объёме, вдали от межфазных границ.

$$f_B = \frac{\lambda_1}{2} [\Psi_{\text{EOS}}(\rho) - \Psi_0] + \frac{\lambda_2}{2} C_{l1}^2 (1 - C_{l1})^2 + \frac{\lambda_3}{2} C_{l2}^2 (1 - C_{l2})^2, \quad (24)$$

где  $\lambda_1, \lambda_2, \lambda_3$  — определяют "жесткость" потенциала, то есть глубину энергетических минимумов,

$\Psi_0$  — константа, обеспечивающая равенство свободных энергий при фазовом равновесии,

$C_{l1}, C_{l2}$  — относительные концентрации жидкостей 1 и 2 (от 0 до 1).

Модифицированное уравнение Карнахана–Старлинга, позволяющее добиться высокой плотностной контрастности между фазами.

$$\Psi_{\text{EOS}} = \rho \left[ C - a\rho - \frac{8RT(-6 + b\rho)}{(-4 + b\rho)^2} + RT \ln \rho \right], \#(25)$$

где  $a, b$  — параметры Карнахана–Старлинга; вместе с  $T$  задают контраст плотностей  $\frac{\rho_l}{\rho_g}$ ,

$R$  — газовая постоянная в решёточных единицах,

$C$  — выбран так, чтобы обеспечить общую касательную к минимумам  $\Psi_{\text{EOS}}$ .

Этот член свободной энергии отвечает за стабильные значения плотности или концентрации каждой из трёх фаз в отсутствии градиентов. Его форма определяет, сколько устойчивых фаз может сосуществовать, и

контролирует глубину потенциальных «долин», соответствующих равновесным значениям.

### 1.2.2.3 Межфазная часть

Вторая составляющая свободной энергии содержит члены, зависящие от пространственных производных порядковых параметров. Эти градиентные слагаемые задают штрафы за резкие изменения плотности или концентрации, тем самым формируя плавный переход между фазами и определяя конечную толщину интерфейса. Значение коэффициентов при градиентах напрямую связано с поверхностным натяжением между фазами.

$$f_I = \frac{\kappa_1}{2} |\nabla\rho|^2 + \frac{\kappa_2}{2} |\nabla C_{l1}|^2 + \frac{\kappa_3}{2} |\nabla C_{l2}|^2, \#(26)$$

где  $\kappa_1, \kappa_2, \kappa_3$  — коэффициенты, напрямую связанные с соответствующими поверхностными энергиями.

Параметры  $\kappa_i$  напрямую влияют на величину межфазной энергии, определяя степень "штрафа" за резкие градиенты и толщину переходной зоны.

### 1.2.2.4 Переход к двум порядковым параметрам

Для упрощения описания трёхфазной смеси вводятся два независимых порядковых параметра: один описывает суммарную плотность (массу), другой — распределение между фазами, аналогично переменной спина в моделях фазовых переходов. При этом предполагается, что сумма всех компонент смеси сохраняется, то есть  $C_g + C_{l1} + C_{l2} = 1$ . Переход к двум независимым полям позволяет упростить описание трёхфазной системы, сохранив возможность независимого контроля над межфазными натяжениями и равновесным распределением фаз.

$$\rho, \phi = \frac{C_{l1} - C_{l2}}{2} \chi, \#(27)$$

где  $\rho$  — плотность,

$\phi$  — комбинированное поле,

$\chi$  — масштаб (амплитуда) поля  $\phi$ , выбирается для симметрии жидкостей.

В формуле (27)  $\phi$  описывает фазовый контраст между жидкостями, а  $\chi$  служит масштабом симметрии между ними.

После подстановки новых переменных в функционал свободной энергии возникает смешанный градиентный член  $\nabla\rho \cdot \nabla\phi$ , который играет важную роль: он обеспечивает независимое управление поверхностными натяжениями между каждой парой фаз.

$$f_I = \left[ \frac{\kappa_1}{2} + \frac{\kappa_2 + \kappa_3}{8(\rho_g - \rho_l)^2} \right] |\nabla\rho|^2 + \frac{\kappa_2 + \kappa_3}{8\chi^2} |\nabla\phi|^2 + \frac{\kappa_3 - \kappa_2}{4\chi(\rho_g - \rho_l)} \nabla\rho \cdot \nabla\phi, \#(28)$$

где  $\rho_g, \rho_l$  — коэкзистентные плотности жидкой и газовой фаз,

знак перед  $(\kappa_3 - \kappa_2)$  управляет предпочтительным смачиванием (треугольник Неймана).

#### 1.2.2.5 Химические потенциалы

В результате вариационного дифференцирования свободной энергии по соответствующим порядковым параметрам получают химические потенциалы — величины, определяющие термодинамические силы, стремящиеся минимизировать энергию системы. Эти потенциалы непосредственно входят в уравнение Кона–Хиллиарда и определяют направление и интенсивность переноса массы между фазами, а также формируют межфазные силы в уравнениях движения.

$$\mu_\rho = \frac{\delta F}{\delta \rho}, \mu_\phi = \frac{\delta F}{\delta \phi}, \#(29)$$

где  $\mu_\rho$  – определяет капиллярный вклад в тензор давления,

$\mu_\phi$  – отвечает за диффузионный поток и релаксацию интерфейса.

Потенциалы (29) выступают в качестве термодинамических сил в уравнениях Кона–Хиллиарда и Навье–Стокса.

#### 1.2.2.6 Континуальные уравнения

Система Навье–Стокса и Кона–Хиллиарда обеспечивает динамику фаз с учётом сил поверхностного натяжения через давление и химические потенциалы.

$$\begin{aligned} \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) &= 0, \quad (\text{сохранение массы}) \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) &= -\nabla \cdot \mathbf{P} + \nabla \cdot [\eta (\nabla \mathbf{u} + \nabla \mathbf{u}^T)], \quad (\text{движение}) \#(30) \\ \partial_t \phi + \nabla \cdot (\phi \mathbf{u}) &= M \nabla^2 \mu_\phi, \quad (\text{фазовое разделение}) \end{aligned}$$

где  $\mathbf{P}$  – тензор давления =  $\rho \nabla \mu_\rho + \phi \nabla \mu_\phi$  (из равновесия),

$\eta$  – динамическая вязкость (обычно зависит от  $\rho$ ),

$M$  – подвижность, определяет время релаксации интерфейса.

#### 1.2.2.7 LB-дискретизация

Для численной реализации модели используется отдельное дискретизованное описание:

- I. Уравнение движения жидкости решается с помощью энтропийной версии метода решёток Больцмана, которая обладает высокой устойчивостью при большом контрасте плотностей между фазами (до  $\frac{\rho^l}{\rho^g} \sim 103$ ).
- II. Уравнение эволюции порядкового параметра (уравнение Кона–Хиллиарда), описывающее диффузию и перемешивание фаз, дискретизируется с использованием стандартной модели релаксации BGK.

Такой подход позволяет точно и устойчиво моделировать динамику многофазной среды с выраженными межфазными переходами.

Для дискретизации используются две независимые решётки:

- $f_i$  – описывает плотность и скорость через энтропийную модель LBM,
- $g_i$  – описывает фазовое поле через BGK-дискретизацию уравнения Кона–Хиллиарда.

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i - \Delta t \Omega_i^H + \Delta t F_i^{\text{EDM}}, \#(31)$$

где  $\Omega_i^H$  – оператор столкновения,

$F_i^{\text{EDM}}$  – источник, учитывающий давление  $\nabla \cdot P$ .

Таблица 2- Характеристики решёток и численных схем в модели свободной энергии

Сетка	Переменная	Эволюция	Схема
$f_i$	$\rho, u$	$f_i \rightarrow f_i^{eq}(\rho, u)$	Entropic LB
$g_i$	$\phi$	$g_i \rightarrow g_i^{eq}(\phi, u)$	BGK

### 1.2.3 Метод фазового поля (*Phase-field*)

#### 1.2.3.1 Функция распределения

В фазово-полевым (phase-field) [3] подходе метод решёток Больцмана использует две функции распределения: одна отвечает за эволюцию гидродинамики, другая — за поведение порядкового параметра, описывающего распределение фаз. Эти распределения построены таким образом, чтобы при вычислении их моментов возвращались основные макроскопические величины: плотность, импульс и давление. Это обеспечивает вывод уравнений движения жидкости (уравнений Навье–Стокса) из микроскопической динамики.

Распределение  $f_i$  используется для описания гидродинамики и обеспечивает вывод уравнений Навье–Стокса. Его равновесная форма:

$$f_i^{eq} = w_i \left[ \frac{p_0}{c_s^2} + \rho \left( \frac{\mathbf{e}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{|\mathbf{u}|^2}{2c_s^2} \right) \right], \#(32)$$

где  $w_i$  — весовые коэффициенты решётки,

$c_s$  — скорость звука в решётке,

$p_0$  включает капиллярный вклад  $\psi - \phi\mu$ .

Давление определяется как:

$$p_0 = c_s^2 \rho + \psi(\phi) - \phi\mu, \#(33)$$

где  $\psi(\phi)$  — объемная часть свободной энергии,

$\mu$  — химический потенциал.

В зависимости от цели моделирования (необходимость сохранения массы), используется либо схема Кан–Хиллиарда, обеспечивающая сохранение фаз, либо схема Аллена–Кана, допускающая локальную релаксацию без строгого сохранения.

- для уравнения Кана–Хиллиарда (СНЕ) используется схема, сохраняющая массу:

$$g_i^{eq} = w_i \left[ \phi + \frac{\mathbf{e}_i \cdot \mathbf{u} \phi}{c_s^2} + \eta\mu \right]. \#(34a)$$

- добавочный член  $\eta\mu$  нужен, чтобы восстановить четвёртый порядок СНЕ.
- для уравнения Аллена–Кана (АСЕ) — схема проще, но не обеспечивает сохранение массы фаз.

$$g_i^{eq} = w_i \left[ \phi + \frac{\mathbf{e}_i \cdot \mathbf{u}\phi}{c_s^2} \right] \#(34b)$$

Минимизация  $F$  определяет равновесные значения порядкового параметра  $\phi$  и формирует движущие силы, которые затем войдут в уравнения

### 1.2.3.2 Свободная энергия

Как и в методе свободной энергии, фазово-полевая модель основывается на задании функционала свободной энергии, от которого зависят равновесные состояния фаз. Минимизация этой энергии определяет устойчивую пространственную конфигурацию фазового параметра  $\phi$  – переменной, описывающей принадлежность каждой точки к одной из фаз. Функциональная производная по  $\phi$  даёт химический потенциал, который затем используется как источник диффузионных и капиллярных сил в уравнениях движения и эволюции фаз.

$$F = \int_{\Omega} \left[ \underbrace{\frac{k}{2} |\nabla\phi|^2}_{\text{градиентная энергия}} + \underbrace{\psi(\phi)}_{\text{объёмная энергия } \psi(\phi)} \right] dV, \#(35)$$

где  $\phi(x, t)$  – порядковый параметр, плавно меняющийся от  $\phi_a$  (фаза А) до  $\phi_B$  (фаза В),

$k$  – коэффициент градиентной энергии, задающий толщину интерфейса,

$\Omega$  – расчетная область,

$\psi(\phi) = \beta(\phi - \phi_A)^2(\phi - \phi_B)^2$  – двойной потенциал (двойной колодец), определяющий устойчивость фаз,

$\beta$  – глубина двойного потенциала  $\psi$ , определяет поверхностное натяжение  $\gamma$ ,

Формула (34) выводит химический потенциал — вариационную производную  $F$  – который будет источником капиллярных сил.

$$\mu = -k\nabla^2\phi + \psi'(\phi), \#(36)$$

где  $\mu$  – химический потенциал (движущая сила диффузии),

$\psi'(\phi)$  – производная объемной энергии.

### 1.2.3.3 Макроскопические уравнения

Эволюция фаз в фазово-полевым подходе описывается уравнением для порядкового параметра  $\phi$ . Используются два варианта:

- уравнение Кана–Хиллиарда (сохраняющее массу фаз) описывает перенос вещества за счёт градиента химического потенциала:

$$\partial_t \phi + \nabla \cdot (\phi \mathbf{u}) = \nabla \cdot (M_\phi \nabla \mu), \#(37)$$

где  $M$  – мобильность: задаёт скорость диффузионного выравнивания,

левая часть – перенос  $\phi$  потоком, правая – диффузия по градиенту  $\mu$ .

- уравнение Аллена–Кана применяется в задачах, где сохранение массы не является строго необходимым:

$$\partial_t \phi + \mathbf{u} \cdot \nabla \phi = M_\phi [\nabla^2 \phi - \psi'(\phi)] - M_\phi |\nabla \phi| \nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right), \#(38)$$

где первое слагаемое в квадратных скобках — классический локальный АСЕ, второй член обеспечивает сохранение массы при высоких градиентах, моделируя движение интерфейса с учётом кривизны.

В обоих случаях уравнение включает слагаемые, описывающие перенос фаз под действием потока жидкости и стремление параметра  $\phi$  к равновесному значению под действием химического потенциала.

### 1.2.3.4 Дискретизация

Для численного решения фазово-полевой модели метод решёток Больцмана использует две решётки:

- функция распределения  $f_i$  отвечает за воспроизведение уравнений Навье–Стокса для движения жидкости
- функция  $g_i$  описывает поведение фазового поля  $\phi$

Таблица 3- Поля, уравнения и релаксационные параметры в фазово-полевой модели

Поле	Уравнение	Распределение	Релаксация
Скорость/плотность	Навье-Стокс	$f_i$	$\tau_f$
Порядок $\phi$	СНЕ/АСЕ	$g_i$	$\tau_g$

Дискретизация уравнений выполняется по методу ВГК с двумя распределениями:

Эволюция  $f_i$  (гидродинамика):

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i - \frac{\Delta t}{\tau_f} (f_i - f_i^{eq}) + \Delta t F_i, \#(39)$$

где  $\mathbf{e}_i$  – дискретные скорости,

$\tau_f$  – коллизийное время, определяет вязкость  $\nu = c_s^2 (\tau_f - 1/2) \Delta t$ .

Эволюция  $g_i$  (фазовое поле):

$$g_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = g_i - \frac{\Delta t}{\tau_g} (g_i - g_i^{eq}) \#(40)$$

где  $\tau_g$  – коллизийное время поля  $\phi$ : мобильность  $M_\phi = \eta c_s^2 (\tau_g - 1/2) \Delta t$ .

Эволюция этих распределений (с учётом столкновений и переноса по решётке) позволяет получить на макроскопическом уровне соответствующие дифференциальные уравнения, описывающие динамику флюида.

### 1.2.3.5 Капиллярная сила

Капиллярные силы, возникающие на межфазных границах, выводятся из свободной энергии как производные по пространственным координатам. Эти силы, пропорциональные градиенту химического потенциала, вносятся в уравнение импульса как дополнительный вклад, аналогичный объёмной силе. Это позволяет моделировать поверхностное натяжение и кривизну интерфейса в рамках единой модели.

Капиллярная сила, возникающая из свободной энергии:

$$\mathbf{F} = \mu \nabla \phi, \#(41)$$

вносится в уравнение импульса как внешний вклад и моделирует поверхностное натяжение. В распределении она реализуется через источник:

$$F_i = w_i \frac{(\mathbf{e}_i - \mathbf{u})}{c_s^2} \cdot \mathbf{F}. \#(42)$$

### 1.2.3.6 Макроскопические величины

Макроскопические характеристики, такие как плотность, скорость и фазовое поле, восстанавливаются из распределений  $f_i$  и  $g_i$  путём вычисления их моментов. Это обеспечивает связь между микроскопическим описанием через распределения и наблюдаемыми физическими величинами в каждой точке пространства.

$$\rho = \sum_i f_i, \rho \mathbf{u} = \sum_i \mathbf{e}_i f_i + \frac{\Delta t}{2} \mathbf{F}, \phi = \sum_i g_i \#(43)$$

Таблица 4- Связь физических величин с параметрами численной реализации в фазово-полевой модели

Физическая величина	Связь с параметрами
Толщина интерфейса	$D = \sqrt{\frac{8k}{\beta}} / (\phi_A - \phi_B)$

Поверхностное натяжение	$\gamma = \frac{(\phi_A - \phi_B)^3}{6\sqrt{2k/\beta}}$
Вязкость	$\nu = c_s^2(\tau_f - 1/2)\Delta t$
Мобильность	$M_\phi = \eta c_s^2(\tau_g - 1/2)\Delta t$

## 1.2.4 Метод объема жидкости (LB-VOF)

### 1.2.4.1 Функция распределений

Основное уравнение метода решёток Больцмана описывает, как частицы каждой фазы перемещаются вдоль дискретных направлений решётки и со временем стремятся к равновесному распределению. Такое приближение реализуется посредством релаксации функции распределения к её равновесной форме в течение характерного временного масштаба, связанного с вязкостью среды [4].

Каждая фаза в модели описывается собственной функцией распределения  $f_k^{(\alpha)}$ , подчиняющейся уравнению Больцмана в форме:

$$f_k^{(\alpha)}(\mathbf{x} + \mathbf{e}_k \Delta t, t + \Delta t) = f_k^{(\alpha)}(\mathbf{x}, t) - \frac{\Delta t}{\tau^{(\alpha)}} [f_k^{(\alpha)} - f_k^{(\alpha),eq}], \#(44)$$

где  $f_k^{(\alpha)}$  – распределение для фазы  $\alpha$ ,

$\mathbf{e}_k$  – дискретные скорости,

$\Delta t$  – шаг по времени решётки,

$\tau^{(\alpha)}$  – время релаксации (через  $\nu^{(\alpha)}$ ),

$f_k^{(\alpha),eq}$  – равновесное распределение, определяется разложением функции Максвелла-Больцмана.

Переход от микроскопического описания к макроскопическим величинам осуществляется через вычисление моментов: сумма всех

распределений даёт локальную плотность, а их взвешенная сумма по направлениям скорости — импульс.

$$\rho^{(\alpha)} = \sum_k f_k^{(\alpha)}, \quad \rho^{(\alpha)} \mathbf{u}^{(\alpha)} = \sum_k \mathbf{e}_k f_k^{(\alpha)}, \#(45)$$

где  $\rho^{(\alpha)}$  — плотность фазы  $\alpha$ ,

$\mathbf{u}^{(\alpha)}$  — скорость фазы  $\alpha$ ,

$f_k^{(\alpha)}$  — распределение для фазы  $\alpha$ .

Суммарные поля, где (46) — суммарная плотность смеси, а (47) — результирующий импульс смеси, получаемый как сумма импульсов фаз:

$$\rho = \rho^{(d)} + \rho^{(c)}, \#(46)$$

$$\rho \mathbf{u} = \rho^{(d)} \mathbf{u}^{(d)} + \rho^{(c)} \mathbf{u}^{(c)}. \#(47)$$

#### 1.2.4.2 Равновесное распределение

Форма равновесного распределения выводится из классического распределения Максвелла–Больцмана с использованием разложения по малой скорости. Такое приближение гарантирует, что на крупномасштабном уровне уравнения движения флюида (уравнения Навье–Стокса) корректно восстанавливаются из кинетического описания.

Равновесное распределение получается с помощью разложения функции Максвелла–Больцмана по малым скоростям:

$$f_k^{(\alpha),eq} = w_k \left[ \rho^{(\alpha)} + \frac{3}{c_s^2} \rho^{(\alpha)} (\mathbf{e}_k \cdot \mathbf{u}) + \frac{9}{2c_s^4} \rho^{(\alpha)} (\mathbf{e}_k \cdot \mathbf{u})^2 - \frac{3}{2c_s^2} \rho^{(\alpha)} |\mathbf{u}|^2 \right], \#(48)$$

где  $w_k$  — весовой коэффициент узла  $k$ ,

$c_s$  — скорость звука в решётке,

$u$  – барицентрическая скорость смеси.

Используется барицентрическая скорость для обеспечения консистентности уравнений импульса.

#### 1.2.4.3 Метод VOF в решетке

В модели LB-VOF интерфейс между фазами отслеживается не напрямую, а через поле объёмной доли. Это скалярное значение показывает, какая часть ячейки заполнена дисперсной фазой, и позволяет точно учитывать форму и движение свободной поверхности.

$$\varepsilon_d = \frac{m_d}{\rho_{\text{cell}}}, \varepsilon_c = 1 - \varepsilon_d, \#(49)$$

где  $m_d$  – масса дисперсной (d) фазы в ячейке,

$\rho_{\text{cell}}$  – средняя плотность смеси в ячейке,

$\varepsilon_d$  – объёмная доля дисперсной фазы, определяемая как отношение массы к общей плотности в ячейке,

$\varepsilon_c$  – дополняющая доля непрерывной фазы,

$\varepsilon_d, \varepsilon_c$  – используются для классификации ячеек как заполненных, интерфейсных или пустых для каждой фазы.

Объёмная доля определяется как отношение массы дисперсной фазы к полной массе в ячейке. Это даёт возможность эффективно хранить информацию о форме интерфейса. Передача массы между ячейками рассчитывается через поток фазы через грань — величину, зависящую от локальных распределений. Такой подход обеспечивает согласованную эволюцию объёмной доли и распределения частиц.

Передача массы между ячейками осуществляется на основе разности распределений и объёмной доли:

$$\Delta m_k^{(\alpha)} = [f_{\tilde{k}}(\mathbf{x} + \mathbf{e}_k \Delta t, t) - f_k(\mathbf{x}, t)] \times \frac{\varepsilon^{(\alpha)}(\mathbf{x} + \mathbf{e}_k \Delta t) + \varepsilon^{(\alpha)}(\mathbf{x})}{2}, \#(50)$$

где  $f_{\tilde{k}}$  — распределение, приходящее из противоположного направления,  $\varepsilon^{(\alpha)}$  — объёмная доля фазы  $\alpha$  в соответствующих ячейках.

Для того чтобы определить, полностью ли заполнена ячейка или находится на границе между фазами, используется пороговое условие: если масса фазы превышает заданный уровень, ячейка считается заполненной; если значительно меньше — пустой. Это позволяет эффективно локализовать интерфейс и применять соответствующие численные схемы.

$$\begin{cases} m^\alpha > (1 + k)\rho^{(\alpha)} \rightarrow \text{ячейка заполняется фазой } \alpha, \\ m^\alpha < k\rho^{(\alpha)} \rightarrow \text{ячейка опустошается,} \end{cases} \#(51)$$

где  $m^\alpha$  — масса фазы  $\alpha$  в ячейке,

$k$  — численная толерантность для устойчивого перехода.

#### 1.2.4.4 Вязкость и релаксация

В модели решёток Больцмана вязкость жидкости задаётся косвенно — через параметр релаксации  $\tau$ , определяющий скорость достижения локального равновесия. Эта связь позволяет управлять вязкостными свойствами жидкости напрямую, без явного задания коэффициента вязкости.

Связь между временем релаксации и вязкостью:

$$\tau^{(\alpha)} = \frac{3\nu^{(\alpha)}}{c_s^2} + \frac{\Delta t}{2}, \#(52)$$

где  $\nu^{(\alpha)}$  — кинематическая вязкость фазы  $\alpha$ ,

$c_s$  — скорость звука в решётке.

Это выражение получается из второго момента уравнения Больцмана.

#### 1.2.4.5 Гравитация и внешние силы

Воздействие внешних сил, таких как гравитация или давление, реализуется через изменение импульса в уравнении Больцмана. Это осуществляется путём добавления специального корректирующего члена в распределение, который приводит к желаемому ускорению жидкости. Такой способ позволяет гибко учитывать влияние внешней среды.

Сила вводится через поправку Форсера (Guo forcing scheme):

$$\Delta \mathbf{j}^{(\alpha)} = \rho^{(\alpha)} \mathbf{F} \Delta t \Rightarrow F_k = 3w_k \rho^{(\alpha)} (\mathbf{e}_k \cdot \mathbf{F}), \#(53)$$

где  $F$  – удельная сила (прим. гравитация  $\rho g$ ),

$F_k$  – проекция силы в направлении  $k$  в пространстве скоростей.

#### 1.2.4.6 Капиллярное давление

Капиллярное давление, возникающее на границе между фазами, включает два слагаемых:

- основной член пропорционален кривизне интерфейса и поверхностному натяжению (уравнение Юнга–Лапласа),
- дополнительный член учитывает контраст вязкостей между фазами, что особенно важно при моделировании жидкостей с сильно различающейся вязкостью. Это позволяет уменьшить численные артефакты (паразитные токи) и стабилизировать симуляцию.

Капиллярное давление включает кривизну и вязкостную коррекцию:

$$\Delta P = \sigma \mathcal{K} + 2(\mu_d - \mu_c) \mathbf{n} \cdot (\nabla \mathbf{u}) \cdot \mathbf{n}, \#(54)$$

где  $\Delta P$  – капиллярное давление на интерфейсе между фазами,

$\sigma$  – коэффициент поверхностного натяжения,

$\mathcal{K}$  – кривизна интерфейса,

$\mu_d, \mu_c$  – динамические вязкости фаз,

$\mathbf{n}$  – единичная нормаль к интерфейсу.

Нормаль к интерфейсу определяется как единичный вектор, направленный вдоль градиента объёмной доли. Кривизна рассчитывается как отрицательная дивергенция этого вектора, что соответствует методу центральной симметричной точечной аппроксимации.

Кривизна поверхности:

$$\mathbf{n} = \frac{\nabla \varepsilon}{|\nabla \varepsilon|}, \mathcal{K} = -\nabla \cdot \mathbf{n}, \#(55)$$

Где  $\varepsilon$  – объёмная доля дисперсной фазы.

#### 1.2.4.7 Давление и обновление

В изотермическом приближении, используемом в LBM, давление определяется через уравнение состояния, аналогичное уравнению для идеального газа. Оно зависит от плотности жидкости и квадрата скорости звука в решётке. Такое приближение позволяет корректно описывать давление без необходимости вводить температуру.

Изотермическое уравнение состояния:

$$P = \rho c_s^2, \#(56)$$

где  $P$  – макроскопическое давление,

$\rho$  – суммарная плотность смеси.

### 1.2.5 Метод Шан-Чен (Shan-Chen)

#### 1.2.5.1 Функция распределений

Для учёта межфазного взаимодействия в модели каждая жидкость описывается собственной функцией распределения, которая эволюционирует согласно модифицированному уравнению Больцмана. В правую часть

уравнения добавляется вклад внешней силы, реализующий межфазные силы притяжения.

$$f_i^{(\sigma)}(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i^{(\sigma)}(\mathbf{x}, t) - \frac{\Delta t}{\tau^{(\sigma)}} \left[ f_i^{(\sigma)}(\mathbf{x}, t) - f_i^{(\sigma),eq}(\rho^{(\sigma)}, \mathbf{u}^{(\sigma)*}) \right], \#(57)$$

где  $f_i^{(\sigma)}$  – распределение частиц фазы  $\sigma$  по направлению  $i$ ,

$\mathbf{e}_i$  – дискретные скорости,

$\Delta t$  – шаг по времени решётки; обычно принимается равным 1,

$\tau^{(\sigma)}$  – время релаксации,

$f_i^{(\sigma),eq}$  – равновесное распределение,

$\mathbf{u}^{(\sigma)*}$  – скорректированная скорость с поправкой по методу Гуо, учитывающая внешнее воздействие.

Весовые коэффициенты и дискретные скорости соответствуют стандартной двумерной решётке D2Q9. Релаксационное время  $\tau$ , зависящее от вязкости каждой компоненты, управляет скоростью восстановления к равновесному состоянию. Влияние силы на равновесную скорость учитывается сдвигом на половину временного шага по методу Гуо.

$$\mathbf{u}^{(\sigma)*} = \mathbf{u}^{(\sigma)} + \frac{\Delta t}{2\rho^{(\sigma)}} \mathbf{F}^{(\sigma)}, \#(58)$$

где  $\mathbf{u}^{(\sigma)}$  – истинная макроскопическая скорость фазы  $\sigma$ ,

$\rho^{(\sigma)}$  – плотность фазы  $\sigma$ ,

$\mathbf{F}^{(\sigma)}$  – совокупная сила, действующая на фазу.

#### 1.2.5.2 Равновесное распределение

Равновесная функция распределения получена путём приближённого разложения максвелловского распределения по малой скорости до второго порядка. Такая форма равновесного распределения обеспечивает корректный

переход к макроскопическим уравнениям Навье–Стокса второго порядка точности при усреднении по времени и пространству.

$$f_i^{(\sigma),eq} = w_i \rho^{(\sigma)} \left[ 1 + \frac{\mathbf{e}_i \cdot \mathbf{u}^{(\sigma)*}}{c_s^2} + \frac{(\mathbf{e}_i \cdot \mathbf{u}^{(\sigma)*})^2}{2c_s^4} - \frac{|\mathbf{u}^{(\sigma)*}|^2}{2c_s^2} \right], c_s^2 = \frac{1}{3}, \#(59)$$

где первое слагаемое соответствует линейному отклику, второе – квадратичному по скорости, третье обеспечивает сохранение энергии при малых скоростях,

$w_i$  – весовые коэффициенты,

$c_s$  – скорость звука в решётке,

$|\mathbf{u}^{(\sigma)*}|$  – модуль скорректированной скорости.

Плотность и скорость каждой фазы вычисляются из моментов:

$$\rho^{(\sigma)} = \sum_i f_i^{(\sigma)}, \rho^{(\sigma)} \mathbf{u}^{(\sigma)} = \sum_i \mathbf{e}_i f_i^{(\sigma)} + \frac{\Delta t}{2} \mathbf{F}^{(\sigma)}. \#(60)$$

- Первый момент (сумма значений) возвращает плотность;
- Второй момент (сумма с весом  $\mathbf{e}_i$ ) даёт импульс;

### 1.2.5.3 Межфазное взаимодействие

Главной идеей модели Shan–Chen является введение межфазного взаимодействия через силу, вычисляемую как функция локальной плотности и плотности в соседних ячейках. Это взаимодействие реализуется как притяжение между частицами одной или разных фаз и приводит к возникновению поверхностного натяжения.

$$\mathbf{F}_{ff}^{(\sigma)} = -g_{\sigma\bar{\sigma}} \psi^{(\sigma)}(\mathbf{x}) c_s^2 \sum_i w_i \psi^{(\bar{\sigma})}(\mathbf{x} + \mathbf{e}_i) \mathbf{e}_i, \#(61)$$

где  $g_{\sigma\bar{\sigma}}$  – коэффициент взаимодействия между фазами  $\sigma$  и  $\bar{\sigma}$  (знак коэффициента определяет характер взаимодействия: отрицательное значение вызывает притяжение между фазами и образование интерфейса),

$\psi^{(\sigma)}$  – псевдопотенциал фазы  $\sigma$ .

Функция псевдопотенциала определяется как:

$$\psi(\rho) = 1 - \exp(-\rho). \#(62)$$

- При малых  $\rho$  – даёт линейное приближение идеального газа.
- При больших  $\rho$  – функция насыщается, моделируя «сгущение» жидкости.

Такое взаимодействие создаёт отклонение от идеального газа и соответствует неидеальному уравнению состояния, позволяющему моделировать фазовые переходы и образование интерфейса.

$$P(\rho) = c_s^2 \rho + \frac{1}{2} c_s^2 \sum_{\sigma, \bar{\sigma}} g_{\sigma\bar{\sigma}} \psi^{(\sigma)} \psi^{(\bar{\sigma})}, \#(63)$$

- Первое слагаемое – идеальный вклад,
- Второе – нелинейная поправка,

чтобы обеспечить существование двух равновесных плотностей и, как следствие, интерфейсов фаз.

#### *1.2.5.4 Взаимодействие с твердой стенкой*

Для задания смачивания твёрдых поверхностей в модели Shan–Chen вводится дополнительное условие взаимодействия между жидкой фазой и стенкой. Эта сила моделирует притяжение или отталкивание жидкости от границы, что позволяет управлять контактным углом и воспроизводить физически реалистичную линию тройного контакта.

$$\mathbf{F}_{\text{ads}}^{(\sigma)} = -g_{\sigma w} \psi^{(\sigma)}(\mathbf{x}) c_s^2 \sum_i w_i \psi(\rho_w) s(\mathbf{x} + \mathbf{e}_i) \mathbf{e}_i, \#(64)$$

где  $g_{\sigma w}$  – коэффициент адгезии фазы к стенке,

$\rho_w$  – параметр «фиктивной плотности» стенки, управляет смачивающими свойствами ( $\rho_w$  не имеет прямой физической интерпретации и используется для управления смачиваемостью границы через псевдопотенциал),

$s(\mathbf{x})$  – индикатор твёрдого узла (1 – стена, 0 – жидкость).

Взаимодействие жидкости с твёрдой поверхностью моделируется аналогично межфазному взаимодействию путём введения псевдопотенциала на стенке. Уровень смачиваемости контролируется через величину граничного параметра

#### 1.2.5.5 Полная сила и обновление распределений

Полный силовой вклад, действующий на ячейку, формируется как сумма трёх составляющих:

- межмолекурного притяжения между фазами,
- взаимодействия с твёрдой стенкой (в случае граничных ячеек),
- внешних сил, таких как гравитация.

Эти силы учитываются в эволюции распределений с помощью корректирующего члена, что позволяет реализовать динамику многофазной системы на уровне решётки.

Общая сила  $\mathbf{F}^{(\sigma)}$  действующая на компоненту  $\sigma$ , включает три вклада: межфазное взаимодействие с другими жидкостями, взаимодействие с твёрдой стенкой и внешние силы (прим. гравитацию).

$$\mathbf{F}^{(\sigma)} = \mathbf{F}_{\text{ff}}^{(\sigma)} + \mathbf{F}_{\text{ads}}^{(\sigma)} + \mathbf{F}_{\text{body}}^{(\sigma)} \#(65)$$

где  $\mathbf{F}_{\text{body}}^{(\sigma)}$  – внешние силовые поля (гравитация, давления и т.п.).

### 1.2.5.6 Связь релаксации с вязкостью

В LBM-модели вязкость жидкости не задаётся напрямую, а определяется через время релаксации  $\tau$ . Эта величина контролирует, насколько быстро система возвращается к равновесному состоянию после возмущения. Кинематическая вязкость для каждой компоненты рассчитывается по формуле (66), позволяющей точно управлять реологическими свойствами фаз.

$$\nu^{(\sigma)} = c_s^2 \left( \frac{1}{s_v^{(\sigma)}} - \frac{1}{2} \right) \Delta t, \#(66)$$

где  $s_v^{(\sigma)}$  – элемент матрицы релаксации  $S$  для моды сдвига (MRT).

В многокомпонентной реализации с MRT (Multiple-Relaxation-Time) элемент  $s_v^{(\sigma)}$  напрямую определяет вязкость компоненты и может быть настроен независимо от других параметров.

### 1.2.5.7 Контактные углы и поверхностное натяжение

В точке соприкосновения трёх фаз (линии тройного контакта) устанавливается равновесие сил, что позволяет аналитически выразить контактные углы между фазами на основе уравнения Неймана.

Уравнение Неймана, описывающее равновесие сил на линии тройного контакта, позволяет установить связь между контактными углами и параметрами межфазного взаимодействия. Это обеспечивает численное управление смачиваемостью и геометрией фазовых границ на твёрдой поверхности.

Уравнение Неймана устанавливает равновесие межфазных сил на линии тройного контакта и позволяет аналитически выразить контактные углы через параметры взаимодействия.

$$\cos \theta_{\sigma\bar{\sigma}} = \frac{2(g_{\sigma w} - g_{\bar{\sigma}w})}{g_{\sigma\bar{\sigma}}(\rho_{eq} - \rho_{dis})}, \#(67)$$

где  $\theta_{\sigma\bar{\sigma}}$  – равновесный контактный угол между фазами  $\sigma$  и  $\bar{\sigma}$  на стенке,

$\rho_{eq}, \rho_{dis}$  – плотности жидкой и газовой фаз на плато изотермы.

Дополнительно, модель воспроизводит закон Лапласа, согласно которому избыточное давление внутри капли обратно пропорционально радиусу кривизны, что подтверждает корректность моделирования капиллярных эффектов.

$$\Delta p = \frac{\gamma}{R}, \#(68)$$

где  $\Delta p$  – избыточное давление внутри капли,

$\gamma$  – эффективное поверхностное натяжение (можно извлечь из численного эксперимента на капле).

### ***1.2.6 Анализ методов и их сравнение***

Среди рассмотренных подходов к моделированию трёхфазных несмешивающихся течений в рамках метода решёток Больцмана (LBM) — моделей свободной энергии (free-energy), фазового поля (phase-field), метода отслеживания объёмной доли (LB-VOF) и модели цветового градиента (color-gradient) — псевдопотенциальный подход Шан–Чена (Shan–Chen) был выделен, как наиболее предпочтительный для реализации, в качестве обоснования выбора необходимо подчеркнуть преимущества подхода Шан–Чена над остальными:

- I. Метод Shan–Chen обеспечивает полную локальность всех расчётов, что существенно упрощает реализацию и масштабирование модели в многомерной сетке, особенно в среде Palabos, изначально ориентированной на локальные коллизии и стриминг. Отсутствие

необходимости в решении уравнений Пуассона (как в методе свободной энергии) или дополнительных уравнений эволюции поля порядкового параметра (как в фазовом поле) существенно снижает вычислительную сложность и повышает устойчивость симуляции за счёт полной локальности операций.

II. Shan–Chen модель обладает высокой гибкостью в задаче межфазных взаимодействий. Благодаря введению сил псевдопотенциала между фазами обеспечивается простое и управляемое задание межфазного натяжения и смачивания. Это критически важно при реализации трёхфазных моделей, где необходимо независимо регулировать силы взаимодействия между каждой парой фаз и корректно задавать контактные углы на твёрдых границах. В отличие от модели цветового градиента, модель Shan–Chen позволяет напрямую задать силу притяжения между фазами, сохраняя физическую интерпретируемость параметров.

III. В сравнении с методом LB-VOF, где требуется сложная интерполяция и реализация реконструкции интерфейса, Shan–Chen сохраняет межфазную границу на основе гладких распределений плотности, что особенно удобно при моделировании тройных линий и фазовых капель произвольной формы. Кроме того, подход Shan–Chen естественным образом поддерживает сжатие фаз (compressibility effects), что важно для капиллярных явлений и фазовых переходов.

Таблица 5-Сравнение методов

Критерий	Color-gradient	Free-energy	Phase-Field	LB-VOF	Shan-Chen
Локальность взаимодействия	Да	Нет	Нет	Да	Да
Поддержка трех и более фаз	Да	Да	Да	Нет	Да
Явное задание межфазного натяжения	Да	Да	Да	Да	Косвенно: через коэффициент $G$

Сохранение массы фаз	Да	Ограниченно	Кан-Хиллиард: да	Да	Да
Необходимость решения уравнений 4-го порядка	Нет	Да	Да	Нет	Нет
Устойчивость при контрасте плотностей	Высокая	Средняя	Средняя	Высокая	Средняя
Сложность реализации и настройки	Средняя	Высокая	Высокая	Низкая	Низкая
Контроль контактного угла	Частичный (эмпирический)	Да (точный)	Да (вариационный)	Ограничен	Эвристический
Тип межфазной модели	Градиентная	Энергетическая	Энергетическая	Геометрическая	Псевдопотенциальная

Учитывая все вышеуказанные аспекты – локальность, гибкость в управлении межфазными силами, численную устойчивость и программную доступность – псевдопотенциальная модель Shan–Chen является обоснованным выбором для построения трёхфазной модели несмешивающегося течения на базе LBM и библиотеки Palabos.

### **1.3 Методология верификации по Оберкампфу**

#### ***1.3.1 Представление работы Оберкампфа***

Монография [5] объединяет математические основы, практические руководства и перечень примеров.

Валидность книги, как источника информации, подкрепляется использованием работы в промышленной и академической стандартизации. Положения книги вошли в стандарты NASA-STD-7009, ASME V&V-10/20 и рекомендацию AIAA G-077-1998; этим же руководствовались комиссии DoE и DoD при экспертной оценке ядерных и аэрокосмических кодов.

### ***1.3.2 Ключевые определения***

У.Л.Оберкампф и Т.Трукано описывают верификацию как «процесс определения того, корректно ли реализованная модель отражает концептуальное описание и насколько точно решается сама модель».

- Фокус внимания – выявление, количественная оценка и минимизация ошибок в алгоритмах, программном коде и численном решении.
- Верификация — это процесс накопления доказательств, а не разовая процедура: «substantiation», то есть предоставление убедительных свидетельств корректности, должно вестись на всех стадиях разработки и сопровождения кода.

Валидация, напротив, оценивает, насколько «модель соответствует реальному миру в рамках предполагаемой области применения». Тем самым верификация и валидация образуют последовательную цепочку:

1. сначала доказываемся правильность численного решения идеализированной (концептуальной) модели;
2. далее проверяется, описывает ли эта модель физику интересующей системы.

### ***1.3.3 Общие принципы верификации***

В качестве общих принципов верификации можно выделить несколько пунктов:

1. Определяются два вида ошибок:
  - Признанные (acknowledged) ошибки — неизбежные отклонения (конечная сетка  $h$ , шаг  $\Delta t$ , остатки итераций и т. д.) — которые оцениваются количественно.
  - Непризнанные (unacknowledged) ошибки — «баги» (ошибки программирования, ввода данных, сбои компилятора). Их нельзя оценить количественным критерием оценки, поэтому требуются строгие процедуры тестирования кода.

2. Надёжным доказательством является только демонстрация. Оберкампф подчёркивает важную роль качественно спланированных тест-кейсов и сравнений с эталонными значениями — аналитическими решениями или полуаналитическими числовыми эталонами высокой точности.
3. Процедуры должны быть воспроизводимыми и подробно описаны. Оберкампф сравнивает верификацию с «подготовкой юридического дела» — нужно собрать и сохранить доказательную базу, чтобы любое стороннее лицо смогло повторить расчёты и прийти к тем же выводам.

### *1.3.4 Уровни верификационных тестов*

Для сложных моделей авторы предлагают building-block approach — движение от простого к сложному с разбиением на четыре уровня (tiers). Эта концепция определяет «лестницу Оберкампфа».

Таблица 6 – Уровни верификации Оберкампфа и их описание

Уровень	Описание уровня	Цель и характер ошибок
Unit problems / Code benchmarks	Уровень предполагает проверку «элементарных» задач, каждая содержит один физический механизм и простую геометрию.	Проверяется, что конкретный численный алгоритм реализован корректно, выявляются ошибки в коде и оценивается базовая сходимость сетки.
Benchmark cases (building-block tests)	Специально созданные экспериментальные конфигурации, где взаимодействуют 2–3 физических процесса; геометрия является упрощенной для простоты реализации и скорости расчетов, но сохраняет ключевые особенности подсистемы.	Оценивается качество моделирования совместных эффектов, большинство исходных данных измеряется, погрешности эксперимента задокументированы.
Subsystem cases	Реальные компоненты системы, но в условиях, упрощённых по сравнению с полноценной установкой.	Модель проверяется на приближенной к реальной сложности. Некоторые граничные/начальные условия известны не полностью → возникают неопределённости входных параметров
Complete	Полностью реализованная	После прохождения всех

system	система в условиях «реального мира»(максимально приближенных).	ранее перечисленных уровней тестов, проверяется, что модель остаётся предсказательной на полной системе, несмотря на скудность измерений и сильную нелинейную связь процессов.
--------	--	--

Оберкамфф подчёркивает, что каждый уровень является фильтром ошибок предыдущего: пока код не прошёл низшие тесты, unit- и benchmark-этапы – не имеет смысла двигаться к более сложным тестам.

## 1.4 Статьи с тестами

### *1.4.1 Трёхфазная модель решёток Больцмана на основе свободной энергии с настраиваемыми поверхностными натяжениями и контактными углами*

В данной работе [6] разрабатывается и анализируется тернарная модель, в основе которой лежит метод решёток Больцмана, предназначенную для описания взаимодействия трёх несмешивающихся жидкостей, в том числе с учётом смачивания на твёрдых границах.

Особенностью предложенного подхода является возможность независимого и аналитически управляемого задания межфазных натяжений и углов смачивания. Модель использует формализм Ландау свободной энергии, позволяющий строго вывести химический потенциал, тензор давления и граничные условия из вариационного принципа. В рамках статьи проведена верификация модели по трём ключевым направлениям:

- Двухслойные эмульсии и линзы – тест на воспроизведение аналитических соотношений между поверхностными натяжениями через углы Неймана.

- Контактные углы – сравнение предсказанных углов смачивания с численными значениями, полученными в симуляции жидкостей в квадратной ячейке.
- Фазовое разделение – демонстрация воспроизведения закономерностей фазового разделения (spinodal decomposition) при различных концентрационных конфигурациях.

Авторы подробно разбирают процедуру инверсии углов смачивания — то есть определение параметров граничного взаимодействия между жидкостью и твёрдым телом на основе заданных контактных углов. Это важно для построения физических симуляций с точно контролируемыми условиями смачивания.

Также показана возможность обобщения модели на произвольное число компонентов, что расширяет её применимость к многокомпонентным системам с более сложной динамикой, таким как жидкие инфузионные поверхности или сложные эмульсии.

#### ***1.4.2 Моделирование трёхфазного реактивного течения на поровом уровне в случайной пористой среде методом решёток Больцмана***

Авторы статьи [7] представляют численное исследование трёхфазного реактивного течения в случайной пористой среде с использованием метода решёток Больцмана (LBM). Работа направлена на моделирование сложных взаимодействий между жидкой водой, неводной жидкостью и газовой фазой в условиях, типичных для подземных инженерных систем, включая процессы переноса, фазового взаимодействия и химических реакций.

Основой численного метода является гибридная трёхфазная модель, объединяющая преимущества метода цветового градиента и псевдопотенциального подхода Shan–Chen, что обеспечивает точное разрешение интерфейсов и регулируемую капиллярную силу. Такая комбинация позволяет точно отслеживать межфазные границы, реализовать

регулируемое межфазное натяжение и учитывать капиллярные силы, одновременно обеспечивая стабильность при высоком контрасте плотностей.

Модель дополнена:

- модулем химической реакции, в котором концентрационное поле взаимодействует с твёрдой фазой через источник в уравнении переноса, моделируя процессы растворения и реактивного изменения пористой структуры;
- геометрией пористой среды, сформированной по методу случайной укладки частиц (random packing), что обеспечивает реалистичную геометрию порового пространства.

Для верификации модели выполнен:

- тест капиллярного впитывания и вытеснения с учётом капиллярного давления;
- анализ формы и поведения капель в статических условиях;
- имитацию реактивного распространения жидкости и изменения структуры поровой среды (прим. растворения порового твердого вещества).

Численные эксперименты демонстрируют способность модели воспроизводить ключевые характеристики трёхфазного взаимодействия, включая динамику контакта трёхфазного угла, растворение твёрдой фазы, вызванное химической реакцией с проникающей жидкостью и влияние геометрической неоднородности на фронт реакционного вытеснения. Работа подчёркивает необходимость учёта мультифизических процессов и геометрической сложности для точного моделирования подповерхностных реактивных систем.

### *1.4.3 Численное прогнозирование относительной проницаемости трёх фаз в пористой среде на поровом уровне методом решёток*

#### *Больцмана*

В данной работе [8] рассматривается трёхфазное течение в пористой среде с использованием метода решёток Больцмана (LBM) на основе модели цветового градиента (color-gradient model). Исследование направлено на численное моделирование процессов вытеснения нефти и газа водой в геометрически сложной пористой структуре, что имеет важное прикладное значение для нефтедобычи, геохимии и подповерхностной гидрологии.

Ключевые особенности модели:

- Модель построена на усовершенствованной трёхфазной схеме color-gradient, включающей устойчивую реализацию межфазных сил.
- Используется механизм цветового градиента, обеспечивающий чёткое разделение фаз и независимое управление межфазным натяжением для каждой пары.
- Реализация перекраски (recoloring) с сохранением симметрии фаз и строгим выполнением закона сохранения массы.

Геометрия пористых сред моделируется на основе либо реальных микротомографических изображений, либо сгенерированных геометрий, что позволяет контролировать пористость и топологию структуры. Авторы исследуют влияние капиллярных и вязкостных сил на эффективность вытеснения нефти и газа при водонагнетании, анализируя:

- относительную проницаемость;
- распределение фаз в порах;
- форму межфазной границы (включая динамику тройных линий).

Для верификации модели выполнен:

- Верификация модели через тест жидкой линзы – оценка соответствия равновесных углов условиям Неймана.

- Симуляция трёхфазного вытеснения в поровом канале с анализом формы фронта и поведения фазовых границ.
- Исследование капиллярного захвата и остаточной насыщенности при варьировании межфазного натяжения между фазами.

Полученные численные результаты подтверждают, что модель адекватно описывает поведение трёхфазной системы, включая воспроизведение реалистичных углов смачивания и корректную динамику межфазных границ. Работа подчёркивает эффективность метода цветового градиента для моделирования многокомпонентных течений в пористых средах, где капиллярные эффекты играют ключевую роль

#### ***1.4.4 Механизмы вытеснения одной жидкости другой в сети капиллярных каналов***

В работе [9] исследуется поведение несмешивающихся жидкостей при вытеснении в пористых средах, сочетая экспериментальные наблюдения и численное моделирование. Целью исследования является выявление механизмов, управляющих геометрией фронта вытеснения в двухфазной системе — типично нефти и воды — в условиях, характерных для процессов фильтрации и нефтедобычи.

Авторы вводят концептуальное разделение режимов вытеснения на основе соотношения капиллярного числа  $Ca$  и отношения вязкостей вытесняющей и вытесняемой фаз:

- поршневое вытеснение (piston-like),
- капиллярное вытеснение (capillary fingering),
- вязкостная нестабильность (viscous fingering).

Путём систематических экспериментов на микромоделльных ячейках (двухмерных стеклянных матрицах с регулярным и случайным распределением пор) показано, что форма фронта зависит от взаимодействия

вязкости, капиллярных сил и геометрии пористой среды. Ключевыми параметрами являются:

- соотношение размеров пор и горлышек;
- граничные условия при подаче жидкости;
- значения межфазного натяжения и смачивания.

Исследование проводится на микромоделльных ячейках — двумерных матрицах с контролируемой геометрией пор. Эксперименты дополняются численным моделированием с помощью метода инвазии-перколяции (без решения уравнений течения), что позволяет изолировать влияние отдельных параметров.

#### ***1.4.5 Трёхмерный решатель потоков на основе метода решёток Больцмана для трёхфазного / трёхкомпонентного течения***

В статье [10] представлена численно устойчивая и физически корректная трёхмерная схема моделирования многокомпонентных несжимаемых течений, основанная на решателе потока метода решёток Больцмана (Lattice Boltzmann Flux Solver, LBFS). В отличие от классического LBM, предложенный подход интегрирует механизм реконструкции потока, характерный для конечнообъёмных методов, с кинетической природой решёток Больцмана. Это позволяет значительно повысить устойчивость численного решения, особенно при моделировании многофазных интерфейсов и резких градиентов плотности.

Модель основана на:

- трёхмерной решётке D3Q27, обеспечивающей улучшенную изотропию и точность дискретизации в сравнении с D3Q19;
- численном флюкс-решении, построенном на модифицированном уравнении Капицы–Фридрихса, применяемом на границах ячеек;
- явной схеме Рунге–Кутты третьего порядка для временной интеграции, обеспечивающей численную стабильность.

Для моделирования межфазных взаимодействий применяется модифицированный метод цветового градиента (color-gradient), включающий перекраску (recoloring), позволяющую точно отслеживать положение и кривизну межфазной границы. Регулировка поверхностного натяжения осуществляется за счёт параметров перераспределения плотности и импульса между фазами, что обеспечивает контроль капиллярных эффектов.

Для верификации разработанного алгоритма выполнены следующие численные тесты:

- статическая сферическая капля — проверка соблюдения соотношения Лапласа между кривизной и давлением;
- столкновение капель — моделирование слияния и отражения на основе кинетических критериев;
- восстановление формы капли после деформации.
- всплытие пузырька в поле силы тяжести — демонстрация динамической устойчивости при наличии плотностного контраста;
- межфазная неустойчивость Рэлея–Тейлора (Rayleigh–Taylor) — ключевой тест на устойчивость интерфейса при встречных потоках фаз разной плотности.

Результаты численного моделирования демонстрируют высокую точность восстановления формы интерфейсов, устойчивость при низких числах Вебера и Рейнольдса, а также точное соблюдение закона сохранения массы. Модель сохраняет симметрию интерфейсов и остаётся устойчивой при высоких плотностных контрастах.

Таким образом, предложенная схема расширяет применимость метода решёток Больцмана к задачам трёхмерной многофазной гидродинамики, сочетающим сложную топологию, поверхностные силы и низкие числа Рейнольдса, и представляет собой перспективную альтернативу классическим LBM и конечнообъёмным схемам.

#### ***1.4.6 Исследование поведения отрыва фазы при самопроизвольной имбибиции в трёхмерной модели порового горлышка методом решёток Больцмана с псевдопотенциалом***

В данной статье [11] исследуется механизм snap-off — отрыва капли вытесняемой жидкости под действием капиллярных сил в пористой среде. Для численного моделирования используется модифицированный метод решёток Больцмана на основе цветового градиента (color-gradient LBM), позволяющий точно воспроизводить межфазную границу и учитывать капиллярные эффекты в условиях сложной геометрии пор.

Snap-off — это процесс, при котором вытесняемая жидкость (прим. нефть) фрагментируется на капли под действием капиллярных сил в узких горлышках пор. Этот процесс имеет критическое значение для остаточной нефтенасыщенности и оценки эффективности методов повышения нефтеотдачи.

Метод включает в себя:

- процедуру перекраски (recoloring), сохраняющую симметрию фаз и массу,
- управление межфазным натяжением и углами смачивания через параметры интерфейсной модели,
- генерация геометрии с переменным сечением каналов.

Для верификации модели выполнены следующие тесты:

- воспроизведение устойчивой тройной линии контакта в узком капилляре,
- моделирование фронта вытеснения при различных значениях капиллярного числа ( $Ca$ ) и углах смачивания,
- анализ поведения при snap-off в одиночных и последовательных поровых каналах.

Результаты показывают:

- существование критического значения капиллярного числа ( $Ca$ ), при котором возникает отрыв капли,
- чувствительность процесса snap-off к смачиваемости поверхности и геометрии пор,
- образование остаточных капель в поровых ловушках, критичных для остаточной нефтенасыщенности.

## 2 МАТЕРИАЛЫ И МЕТОДЫ

### 2.1 Отобранные тесты

#### 2.1.1 *Одиночная капля (Young-Laplace)*

**Уровень Оберкампа:** Unit / Code benchmark

**Физическая постановка:** Проверяется выполнение закона Лапласа для сферической капли:  $\Delta p = 2\sigma/R$ , где  $\sigma$  — коэффициент поверхностного натяжения,  $R$  — радиус капли для одной сферической капли фазы А внутри фазы В.

**Геометрия домена и начальные условия:** Двухмерная расчётная область с периодическими границами; капля инициализируется как распределение плотности с гиперболическим профилем.

**Граничные условия:** Периодические условия по всем направлениям.

**Критерии успешности.** Относительное отклонение численного  $\Delta p$  от аналитического значения не превышает 1 %..

**Источник:** Semprebon, 2016. [6]

#### 2.1.2 *Трёхмерная капля у смачиваемой поверхности*

**Уровень Оберкампа:** Unit / Code benchmark

**Физическая постановка:** Газовая капля размещена между двумя смачиваемыми стенками, оценивается точность воспроизведения статического контактного угла, обусловленного межфазными силами и свойствами стенок.

**Геометрия домена и начальные условия:** Расчётная область ограничена двумя твёрдыми параллельными стенками; капля инициализируется в форме полусферы.

**Граничные условия:** Стены – (отражающее) bounce-back, остальные – периодические.

**Критерии успешности:** Отклонение измеренного угла от теоретического не превышает двух градусов.

**Источник:** Yuan Yu, 2019. [12]

### ***2.1.3 Трёхфазная линза (Liquid Lens)***

**Уровень Оберкампа:** Unit / Code benchmark

**Физическая постановка:** Лёгкая жидкость (нефть) располагается между двумя более тяжёлыми (вода и газ), формируя тройной контакт. Проверяется корректность воспроизведения равновесия межфазных сил на тройной линии в соответствии с векторной формой уравнения Неймана.

**Геометрия домена и начальные условия:** Прямоугольная область с периодическими границами; фазы инициализируются секторально.

**Граничные условия:** Периодические по всем направлениям.

**Критерии успешности:** Векторная сумма поверхностных напряжений на тройной линии отклоняется не более чем на 3 %.

**Источник:** Semprebou, 2016. [6]

### ***2.1.4 Фазовое разделение трёх компонентов***

**Уровень Оберкампа:** Building-block test

**Физическая постановка:** Однородная смесь трёх жидкостей разделяется на домены в процессе спинодального распада. Тест воспроизводит динамику спинодального распада — ключевого механизма разделения однородной смеси на устойчивые фазы. Используется логарифмический анализ роста доменов.

**Геометрия домена и начальные условия:** Периодический квадрат, в каждой точке равное содержание трёх фаз + шум  $\leq 1\%$ .

**Граничные условия:** Периодические по всем направлениям.

**Критерии успешности:** Логарифмическое масштабирование размера доменов даёт наклон  $0.33 \pm 0.02$ .

**Источник:** Semperebon, 2016. [6]

### *2.1.5 Слоистое течение в трехмерном канале*

**Уровень Оберкампа:** Building-block test

**Физическая постановка:** По оси канала движутся три горизонтальных слоя (газ–нефть–вода). Сочетает сдвиг и капиллярные силы. Тест оценивает устойчивость интерфейсов в условиях наложения сдвигового профиля и капиллярных сил, характерного для микроканальных течений.

**Геометрия домена и начальные условия:** Канал, три равных по высоте слоя, профиль Пуазейля.

**Граничные условия:** Стенки — отражающие (bounce-back),  $x$  — периодические,  $z$  — циклический сдвиг.

**Критерии успешности:** Интерфейсы сохраняются вдоль 95 % длины канала, профиль скорости — точность 2 %.

**Источник:** Zhu, 2021. [8]

### *2.1.6 Вложенная капля (Nested Droplet)*

**Уровень Оберкампа:** Building-block test

**Физическая постановка:** Капля нефти окружена оболочкой газа и погружена в воду. Проверяется согласованность давления на двух

интерфейсах вложенной капли, что является прямым тестом для капиллярной реализации модели Шан–Чен.

**Геометрия домена и начальные условия:** Две окружности радиусов  $R_1 < R_2$  в квадратном периодическом домене.

**Граничные условия:** Периодические условия по всем направлениям.

**Критерии успешности:** Ошибка давления на внутреннем и внешнем интерфейсах  $< 1.5 \%$ .

**Источник:** Semprebon, 2016. [6]

### *2.1.7 Подъём пузыря в трёхфазной системе*

**Уровень Оберкампа:** Solution verification

**Физическая постановка:** Газовый пузырь поднимается через интерфейс нефть–вода под действием гравитации. Тест проверяет параболическую траекторию движения центра пузыря а также ограничение паразитных скоростей вдоль границы.

**Геометрия домена и начальные условия:** Вертикальный столб: снизу нефть, сверху вода. Пузырь в нефти.

**Граничные условия:** Низ – bounce–back (отражение), верх – губка, боковые – периодические.

**Критерии успешности:** Траектория  $Ct^2 \pm 10\%$ , паразитная скорость  $< 5 \times 10^{-3} c_s$ .

**Источник:** Zheng, 2024. [13]

### *2.1.8 Вытеснение линейной нефтяной перемычки*

**Уровень Оберкампа:** Solution verification

**Физическая постановка:** Вода вытесняет узкую полосу нефти через капиллярный барьер.

**Геометрия домена и начальные условия:** Двухмерный канал шириной  $W$ , по центру нефтяная полоса длиной  $L > W$ .

**Граничные условия:** Вход – линейно нарастающее давление, выход – фиксированное давление; стенки – гидрофобные.

**Критерии успешности:** Погрешность воспроизведения порогового капиллярного давления  $< 4\%$ .

**Источник:** Lenormand, 1983. [9]

### ***2.1.9 Поршневое вытеснение (Piston-type)***

**Уровень Оберкампа:** Solution verification

**Физическая постановка:** Вода как «поршень» выдавливает нефть, образуя фронт, оценивается стабильность фронта при капиллярных числах  $Ca < 10^{-3}$ . Проверяется фронт вытеснения при низких капиллярных числах, характерных для вытеснения в микрокапиллярах.

**Геометрия домена и начальные условия:** Канал с нефтью, подача воды с одного конца.

**Граничные условия:** Вход – постоянная скорость, выход – нулевая производная, стенки – несмачиваемые.

**Критерии успешности:** Ширина фронта сохраняется 80 % до  $t = 2000$ ; без неустойчивостей.

**Источник:** Lenormand, 1983. [9]

### ***2.1.10 Отрыв (Snap-off) в поровом горлышке***

**Уровень Оберкампа:** Subsystem benchmark

**Физическая постановка:** Газовая капля отделяется в горлышке при нагнетании воды. Тест демонстрирует топологическое изменение фазового

распределения в поровом пространстве — один из ключевых механизмов захвата газа

**Геометрия домена и начальные условия:** Геометрия импортирована из микротомографии, горло шириной  $d$ , камера – шириной  $3d$ . Газ заполняет камеру, вода – горло.

**Граничные условия:** Вход воды – постоянный дебит, остальные стенки – несмачиваемые.

**Критерии успешности:** Время snap-off и объём оторванной газовой нити совпадают с данными из статьи при тех же параметрах в пределах 8%.

**Источник:** Zheng, 2024. [13]

### *2.1.11 Последовательное вытеснение газ–нефть–вода*

**Уровень Оберкампа:** Subsystem benchmark

**Физическая постановка:** Три фазы последовательно перемещаются в микро-канале.

**Геометрия домена и начальные условия:** Канал длиной 30 мм. Инициализация фаз в заданной последовательности газ–нефть–вода с фиксированной длиной каждого сегмента.

**Граничные условия:** На входе давление воды растёт линейно. Стенки смешанной смачиваемости.

**Критерии успешности:** Пороговое давление перехода  $\Delta P_{gw}$  и  $\Delta P_{ow}$  предсказаны с точностью  $\pm 5\%$ .

**Источник:** Расширенная на три фазы версия двухфазного теста из Lenormand, 1983. [9]

## 2.2 Постановка тестов в Palabos

### 2.2.1 Тест на вязкость

#### 2.2.1.1 Уровень верификации по Оберкампфу

Данный тест относится к первому уровню — верификация кода (code verification), так как проверяет корректность численного решения на основании известного аналитического решения задачи затухающего сдвига. Это классический unit-тест, основанный на сравнении численно полученного значения кинематической вязкости с её аналитическим выражением, вычисленным из параметра релаксации.

#### 2.2.1.2 Цель теста

Проверка точности реализации вязкости в модели решёток Больцмана (LBM) для трёх различных компонентов: воды, нефти и газа. Для каждой фазы измеряется численное значение кинематической вязкости  $\nu$ , которое затем сравнивается с теоретическим значением, заданным через параметр релаксации  $\tau$  в модели BGK.

#### 2.2.1.3 Физическая постановка

В двумерной области с периодическими границами по обоим направлениям задаётся синусоидальный профиль скорости по горизонтали:

$$u_x(y, t = 0) = u_0 \sin(ky), u_y = 0, \#(69)$$

где  $u_0$  — начальная амплитуда,

$k = \frac{2\pi}{L}$  — волновое число,

$L$  — высота домена,

$\rho_0 = 1.0$  — плотность считается постоянной.

В рамках линейной гидродинамики и отсутствия внешних сил такое течение должно затухать экспоненциально:

$$u_x(y, t) = u_0 \exp(-\nu k^2 t) \sin(ky). \#(70)$$

Вычисленная энергия затухает экспоненциально, что соответствует теоретическому предсказанию о рассеивании энергии вязкими силами в отсутствии внешних воздействий.

$$E(t) = \langle u_x^2 \rangle \sim \exp(-2\nu k^2 t). \#(71)$$

#### 2.2.1.4 Начальные и граничные условия

Начальные условия: скорость по оси  $x$  задана в виде  $u_0 \sin(ky)$ , компонента  $u_y = 0$ ; плотность  $\rho = 1.0$ .

Граничные условия: периодические по обоим направлениям.

#### 2.2.1.5 Численная реализация

- Используется классическая двумерная решётка D2Q9 с BGK-динамикой.
- Параметры симуляции:  $n_x = n_y = 100$ ,  $u_0 = 0.01$ , шаги до  $t = 5000$ .
- Для каждой из трёх фаз используется своё значение  $\tau$  :

- I. Вода:  $\tau = 0.7, \nu_{\text{ref}} = \frac{\tau - 0.5}{3} \approx 0.0667$
- II. Нефть:  $\tau = 0.8, \nu_{\text{ref}} = 0.1$
- III. Газ:  $\tau = 0.9, \nu_{\text{ref}} = 0.1333$

#### 2.2.1.6 Методы постобработки

На каждом временном шаге с периодичностью 50 итераций рассчитывается средняя кинетическая энергия:

$$E(t) = \langle u_x^2 \rangle. \#(72)$$

Вычисляется логарифм энергии:

$$\ln E(t) = -2\nu k^2 t + \text{const}. \#(73)$$

По методу наименьших квадратов (линейной регрессии) определяется наклон прямой  $m$ , и далее извлекается измеренная вязкость:

$$v_{\text{num}} = -\frac{m}{2k^2} \cdot \#(74)$$

### 2.2.1.7 Критерии успешности

Относительная ошибка:

$$\varepsilon_{\text{rel}} = \frac{|v_{\text{num}} - v_{\text{ref}}|}{v_{\text{ref}}} \cdot \#(75)$$

Тест считается пройденным, если  $\varepsilon_{\text{rel}} < 1\%$ .

## 2.2.2 Тест «Жидкая линза» («Liquid Lens»)

### 2.2.2.1 Уровень верификации по Оберкампфу

Данный тест относится ко второму уровню — верификация решения (solution verification) согласно классификации Оберкампфа. Он направлен на качественную и количественную оценку способности модели воспроизводить корректную форму интерфейса и значение контактного угла в трёхфазной системе с известной симметрией и стабильной геометрией. Верификация проводится путём визуального анализа распределения плотностей и численного измерения контактного угла между фазами.

#### 2.2.2.2 Цель теста

Целью теста является воспроизведение устойчивой геометрической конфигурации трёхфазного интерфейса в виде «жидкой линзы», а также корректное моделирование межфазных взаимодействий, включая восстановление ожидаемых контактных углов между фазами. Это необходимо для подтверждения физической достоверности реализации модели Шан–Чена (Shan–Chen) для трёх взаимодействующих компонентов.

#### 2.2.2.3 Физическая постановка

В двухмерной прямоугольной области размещается сферическая капля одной фазы (fluid 1), находящаяся на границе между двумя другими фазами: нижняя (fluid 2) и верхняя (fluid 3). Система представляет собой статичную

жидкую линзу, в которой капля удерживается межфазными взаимодействиями и взаимодействием с границами. Первоначально плотность капли внутри радиуса  $R$  и плотность окружающих фаз инициализируются с флуктуациями.

Изначально плотности задаются как:

- Внутри капли (фаза 1):  $\rho_1 = 1.0 + \delta$ ,
- Ниже капли (фаза 2):  $\rho_2 = 1.0 + \delta$ ,
- Выше капли (фаза 3):  $\rho_3 = 1.0 + \delta$ ,

где  $\delta \sim 10^{-3}$  — случайные флуктуации.

#### 2.2.2.4 Начальные и граничные условия

- Границы по  $x$  (левая и правая): периодические условия.
- Нижняя граница ( $y = 0$ ): отражающая граница (bounce-back).
- Верхняя граница: демпфирующая «sponge-зона» для поглощения волн и предотвращения отражений.
- Начальные условия: плотности инициализируются как описано выше, скорости во всех ячейках – нулевые.

#### 2.2.2.5 Численная реализация

Прямоугольная область:  $500 \times 500$  узлов.

Радиус капли:  $R = 60$ .

Используется решётка D2Q9 с моделью Forced Shan–Chen.

Каждая фаза моделируется на отдельной решётке с собственной динамикой:

- Fluid 1:  $\omega_1 = 1.0$
- Fluid 2:  $\omega_2 = 0.8$
- Fluid 3:  $\omega_3 = 1.2$

Коэффициенты межфазного взаимодействия:

- $G_{12} = 1.3$ ,
- $G_{13} = 1.5$ ,
- $G_{23} = 1.4$ .

Инициализация фаз выполняется функциями «applyIndexed» с выбором плотности по геометрии. На верхней границе применяется «Sponge-зона» с функцией «ViscositySpongeZone2D».

#### 2.2.2.6 Критерий успешности

- Поддержание устойчивой формы линзы в течение всей симуляции.
- Отсутствие неустойчивости или несимметричности плотности
- Значения контактных углов в пределах физических отклонений от ожидаемых (45–135 градусов).

### 2.2.3 Тест «Вложенная капля» («Nested Droplet»)

#### 2.2.3.1 Уровень верификации по Оберкампфу

Тест относится ко второму уровню — верификация решения (solution verification) в классификации Оберкампфа. Он направлен на качественную оценку стабильности интерфейсов и способности модели воспроизводить устойчивую геометрию фаз при наличии вложенной капли одной жидкости, окружённой второй фазой и погружённой в третью. Тест не имеет аналитического решения, но оценивает корректность взаимодействий между фазами на основе физически ожидаемого распределения плотности и формы.

#### 2.2.3.2 Цель теста

Проверка способности трёхфазной модели Шан–Чена адекватно воспроизводить вложенную конфигурацию фаз: нефть внутри газа, находящихся во внешней фазе воды. Такое распределение моделирует важный физический сценарий, возникающий, в частности, при эмульсиях, капиллярной инверсии или в задачах о капле в оболочке.

Цель — достижение устойчивого конфигурационного равновесия и симметрии фаз.

### 2.2.3.3 Физическая постановка

В двумерной прямоугольной области размещается центральная круглая капля нефти радиуса  $R_{oil}$ , окружённая кольцевым слоем газа с внешним радиусом  $R_{gas}$ . Внешняя фаза — вода, заполняющая остальное пространство. Все три фазы инициализируются с различными базовыми плотностями и флуктуациями:

- Нефть (фаза 1):  $\rho \approx 0.8$  при  $r \leq R_{oil}$ ;
- Газ (фаза 2):  $\rho \approx 0.4$  при  $R_{oil} < r < R_{gas}$ ;
- Вода (фаза 3):  $\rho \approx 1.0$  при  $r \geq R_{gas}$ ;

Флуктуации плотности составляют  $\delta\rho \sim 10^{-4}$  и необходимы для устойчивости

симуляции.

### 2.2.3.4 Начальные и граничные условия

- Начальные условия: плотности задаются по геометрическим областям, скорости во всех точках равны нулю.
- Границы по  $x$  (левая и правая): периодические.
- Границы по  $y$  (верх и низ): задаются как «bulk», с BGK-динамикой при  $\tau = 1.0$ , инициализированы как  $\rho = 1.0$ ,  $u = 0$ .
- Специальные граничные условия (bounce-back) не применяются, поскольку симуляция представляет статическое равновесие.

### 2.2.3.5 Численная реализация

- Размер сетки:  $400 \times 400$  узлов.
- Положение центра:  $(c_x, c_y) = (200, 200)$ .
- Радиусы:  $R_{oil} = 60$ ,  $R_{gas} = 120$ .
- Используемая решётка: D2Q9 с моделью Forced Shan–Chen.
- Каждая фаза моделируется на отдельной решётке с BGK-динамикой и  $\omega = 1.0$ .

- Коэффициенты межфазного взаимодействия:

$$\text{I. } G_{oil-gas} = 2.2,$$

$$\text{II. } G_{oil-water} = 2.5,$$

$$\text{III. } G_{water-gas} = 2.8.$$

- Механизм взаимодействия фаз реализован через `ShanChenMultiComponentProcessor2D`.

### 2.2.3.6 Методы постобработки

Для двумерного случая закон Лапласа связывает разность давлений между внутренностью капли и окружающей средой с поверхностным натяжением и радиусом кривизны интерфейса:

$$\Delta P = \frac{\sigma}{R}, \#(76)$$

где  $\Delta P$  — разность давлений внутри и снаружи капли,

$\sigma$  — поверхностное натяжение,

$R$  — радиус капли.

Переходя к практической реализации, формулу можно переписать как:

$$\sigma = R \cdot \Delta P.$$

### 2.2.3.7 Критерии успешности

- Интерфейсы между фазами устойчивы и симметричны.
- Геометрическая структура «капля в капле» сохраняется на протяжении симуляции.
- Отсутствие паразитных токов и разрывов интерфейса.

### 2.2.3.8 Алгоритм численной верификации

Для анализа корректности реализации закона Лапласа была реализована следующая последовательность шагов:

- Измерение радиуса  $R$ . Центр капли определялся заранее. Радиус вычислялся по изолинии плотности нефти  $\rho = \frac{\rho_{oil} + \rho_{gas}}{2}$  как среднее расстояние от центра до точек этой изолинии.
- Вычисление давления внутри и снаружи капли. Давление вычислялось как локальное термодинамическое давление:

$$P = \rho \cdot c_s^2 + \frac{1}{2} G \psi^2,$$

где  $c_s^2 = \frac{1}{3}$ ,

$\psi = \psi(\rho)$  – псевдопотенциал,

$G$  – коэффициент межфазного взаимодействия.

- Вычисление  $\sigma = R \cdot \Delta P$ . По измеренному  $R$  и  $\Delta P$  на каждой итерации вычислялось численное значение  $\sigma$ , сравниваемое между различными сетками.

Таким образом, можно заключить, что тест Nested Droplet прошёл второй уровень верификации по методологии Оберкампа. Проведённый численный эксперимент демонстрирует высокую точность капиллярного взаимодействия модели на сетке  $1200 \times 1200$ , и показывает допустимое поведение при уменьшении размера сетки до  $600 \times 600$ , но недостаточную точность при дальнейшем её снижении.

## 3 РЕЗУЛЬТАТЫ

### 3.1 Результаты теста вязкости

#### 3.1.1 Вывод

Тест затухания синусоидального сдвига является надёжным и воспроизводимым способом проверки корректной реализации вязкости в LBM. Он не зависит от сложных межфазных взаимодействий или геометрии, и поэтому идеально подходит для unit-тестирования отдельных физических механизмов в многофазной модели

#### 3.1.2 Результаты расчетов

Для реализации данного тестового сценария был написан код с использованием языка C++ и библиотеки Palabos (см. приложение А).

В таблице приведены численно измеренные значения кинематической вязкости для трёх фаз (вода, нефть, газ), сравнение с аналитическими значениями и относительная ошибка. Все тесты успешно пройдены, так как относительное отклонение не превышает установленного порога в 1%.

Таким образом, реализованная модель корректно воспроизводит вязкость для каждого из компонентов, подтверждая правильность реализации BGK-динамики и параметров дискретизации. Данный unit-тест может считаться успешно верифицированным.

Таблица 7-Результаты теста затухания сдвига

Фаза	$\tau$	$\nu_{ref}$	$\nu_{num}$	Отн.ошибка
Вода	0.7	0.06667	0.06668	0.0271%
Нефть	0.8	0.10000	0.10002	0.0209%
Газ	0.9	0.13333	0.13335	0.0118%

### 3.2 Результаты теста «Жидкая линза»

#### 3.2.1 Результаты теста

Для реализации данного тестового сценария был написан код с использованием языка C++ и библиотеки Palabos (см. приложение Б).

Одним из возможных вариантов для начальной верификации по методологии Оберкампа является визуальный анализ. Соответственно, для визуального анализа теста линзы были выгружены в Paraview vti-файлы с выходными данными расчетов модели и из этих данных можно сделать вывод, что при достижении квазистационарного состояния в симуляции формируется симметричная структура линзы относительно вертикальной оси. Форма капли сохраняется на протяжении длительного времени без заметных деформаций, что визуально подтверждает установившееся равновесие.

Для количественного подтверждения пространственной симметрии проведён анализ параметра (72):

$$\Delta = |x_L - x_R|, \#(77)$$

где  $x_L$  и  $x_R$  – координаты симметричных точек интерфейса относительно центра купола линзы.

На итерациях 560–580 зафиксированы следующие характеристики:

- Значения  $\Delta$  колеблются в пределах 7.4–18.8,
- В течение нескольких шагов (по 4 измерения на итерацию) наблюдается устойчивое удержание формы линзы,
- Отсутствует направленный тренд к росту или снижению параметра  $\Delta$ , что свидетельствует о локальной стабилизации решения.

### **3.2.2 Вывод**

На основании проведённого визуального анализа и численной оценки симметрии в интервале 560–580 временных шагов можно сделать вывод о достижении устойчивого состояния жидкой линзы и корректной реализации физики межфазного взаимодействия. Это позволяет считать данный тест успешно пройденным в рамках второго уровня верификации по Оберкампу.

Положение фаз в момент установившегося равновесия системы при расчетах.

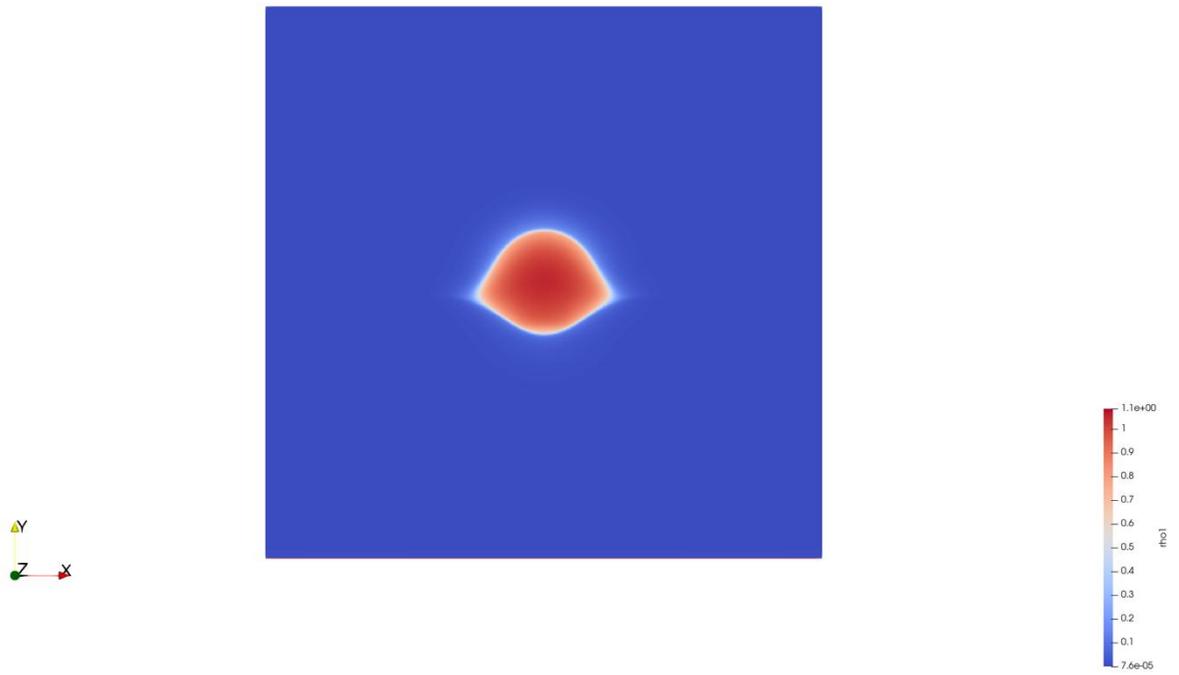


Рисунок 1-Расположение фазы 1

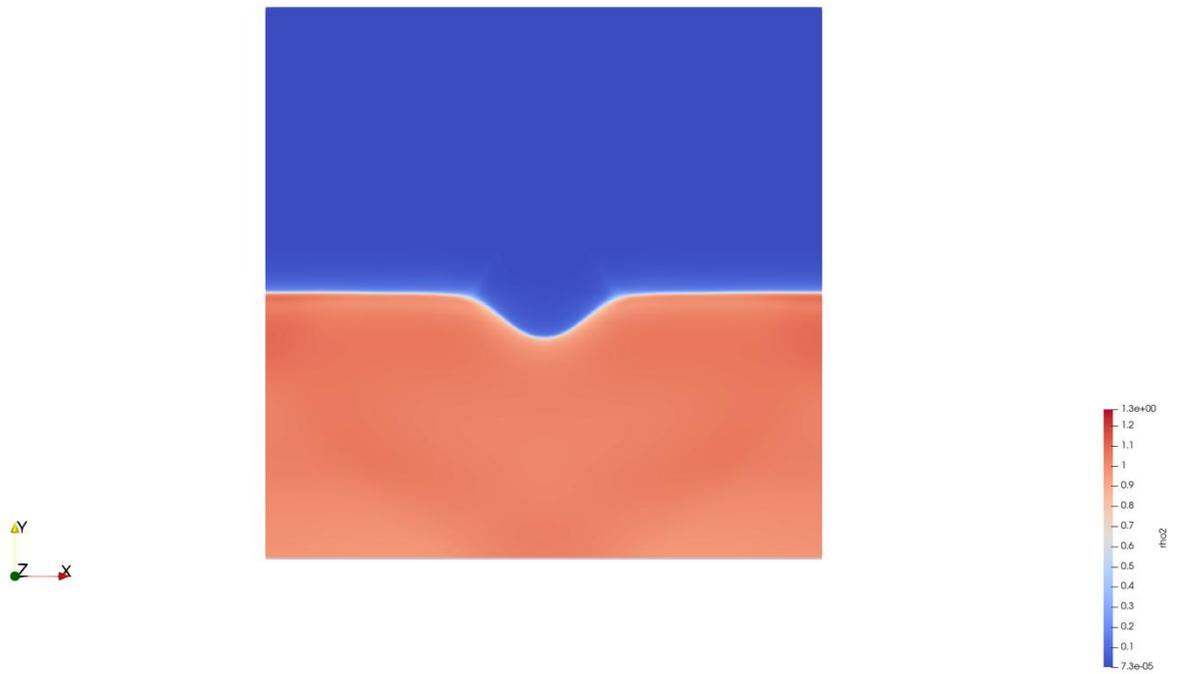


Рисунок 2-Расположение фазы 2

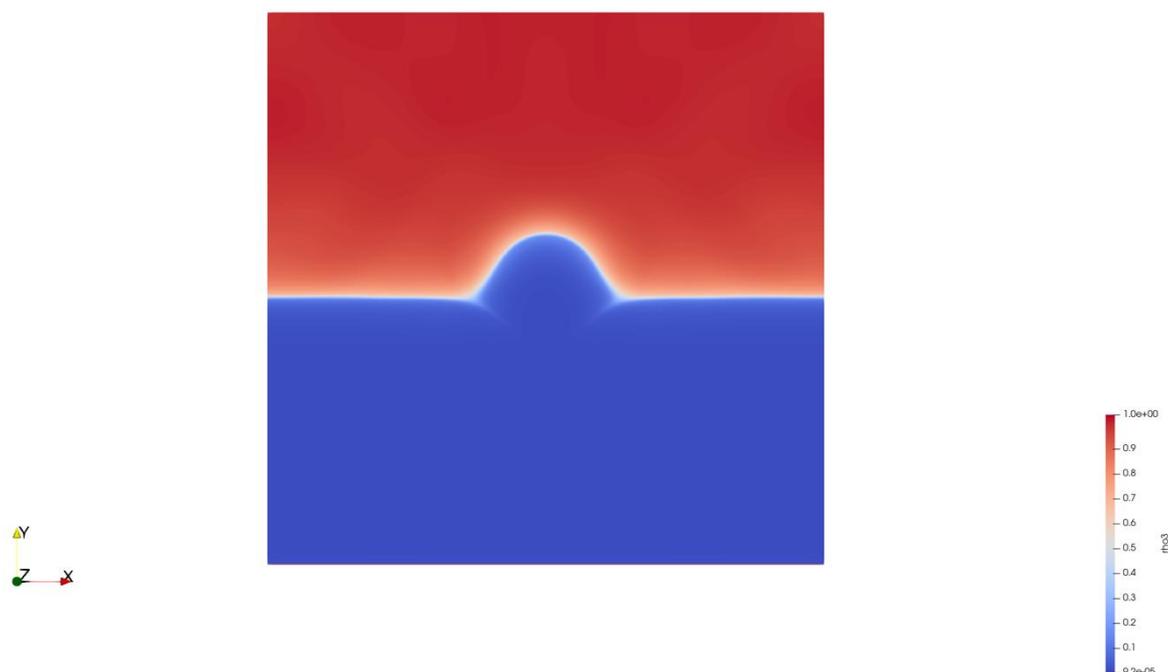


Рисунок 3-Расположение фазы 3

### 3.3 Результаты теста «Вложенная капля»

#### 3.3.1 Результаты теста

Для реализации данного тестового сценария был написан код с использованием языка C++ и библиотеки Palabos (см. приложение В).

Сравнение проводилось на трёх уровнях пространственного разрешения, где расчеты на самой большой сетке принимались за референсные.

- $1200 \times 1200$  (референс),
- $600 \times 600$ ,
- $300 \times 300$ .

Средние значения  $\sigma$  по итерациям 310–440 представлены в таблице.

Таблица 8-Результаты расчетов на разных размерностях сетки

Размер сетки	Среднее значение $\sigma = R \cdot \Delta P$	Отклонение от $1200 \times 1200$
$1200 \times 1200$	$\bar{\sigma} = 18.0141$	Референсное значение

600 × 600	$\bar{\sigma} = 17.8075$	1.15%
300 × 300	$\bar{\sigma} = 14.5316$	19.37%

### 3.3.2 Вывод

- Модель корректно воспроизводит физический закон Лапласа при достаточно высоком разрешении (600 × 600 и выше);
- Сходимость по сетке демонстрирует уменьшение ошибки с увеличением пространственного разрешения.
- При низком разрешении (300 × 300) наблюдаются значительные численные флуктуации, появление пустых значений, нестабильные изолинии плотности и нарушение линейной зависимости между  $\Delta P$  и  $1/R$ .

Положение фаз в момент установившегося равновесия системы при расчетах на сетке размером 600\*600.

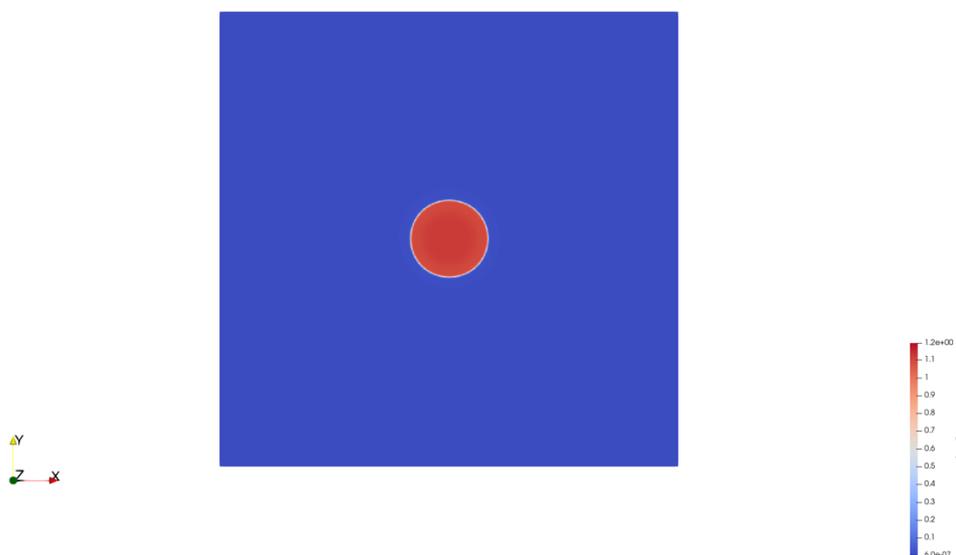


Рисунок 4-Фаза нефти в момент равновесия

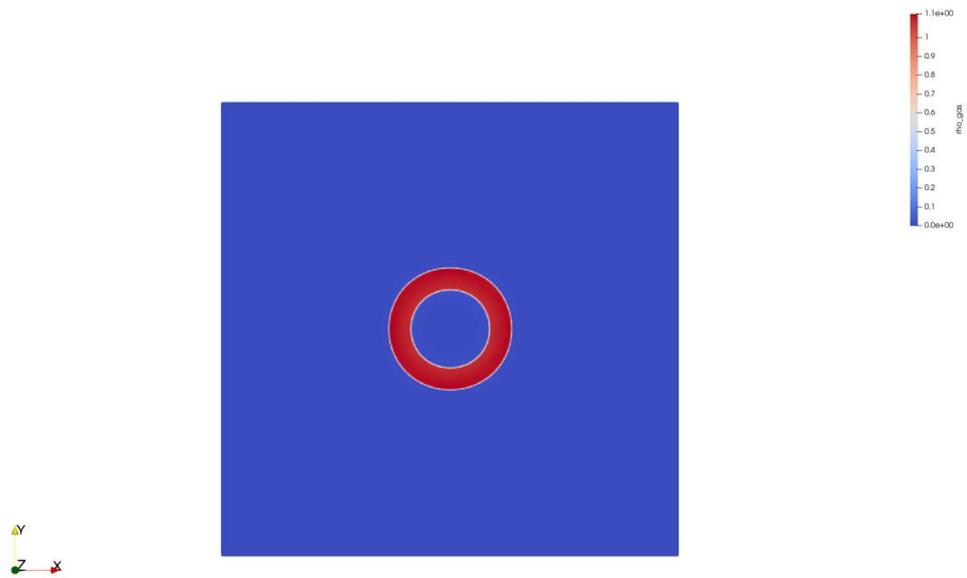


Рисунок 5-Фаза газа в момент равновесия

## Заключение

Целью настоящей работы являлся подбор и реализация тестов для верификации трёхфазной модели несмешивающегося течения, построенной на основе метода решёток Больцмана и реализованной в библиотеке Palabos. Проверка модели проводилась согласно методологии Оберкампа, предполагающей многоуровневую структуру верификационных задач.

Для достижения поставленной цели в ходе работы были успешно решены следующие задачи:

- проведён обзор актуальных научных публикаций, содержащих тесты для верификации трёхфазных моделей,
- выполнен анализ и отбор релевантных тестов с учётом воспроизводимости, физической достоверности и применимости к модели,
- произведена классификация тестов по уровням согласно подходу Оберкампа (unit, building-block, solution verification и др.),
- реализованы три верификационных теста в библиотеке Palabos, охватывающие ключевые аспекты трёхфазного течения,
- проведён качественный и количественный анализ полученных результатов и их соответствия ожидаемым физическим эффектам

В рамках данной работы были реализованы три сценария, которые соответствуют разным уровням верификации и охватывают важные физические механизмы. В частности:

- тест «Жидкая линза» подтвердил способность модели воспроизводить равновесие на тройной линии в трёхфазной системе,
- тест «Вложенная капля» обеспечил количественную проверку закона Лапласа на двух вложенных интерфейсах,
- тест на вязкость продемонстрировал корректную реализацию спонтанного разделения фаз в изначально однородной системе.

Каждый из реализованных тестов прошёл успешную визуальную и численную верификацию, что позволяет сделать обоснованный вывод о корректной работе модели Shan–Chen в трёхфазной постановке и её применимости к дальнейшему численному моделированию сложных многокомпонентных течений.

Научная значимость работы заключается в формализации и реализации многоуровневого подхода к верификации трёхфазной модели, а также в адаптации библиотеки Palabos к целевым задачам с сохранением физической интерпретируемости параметров.

Таким образом, выполненная работа не только подтвердила корректность реализации выбранной модели, но и заложила основу для её дальнейшего расширения и практического применения в рамках вычислительной гидродинамики многофазных систем.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Zahid, F. Review of the Color Gradient Lattice Boltzmann Method for Simulating Multi-Phase Flow in Porous Media: Viscosity, Gradient Calculation, and Fluid Acceleration / F. Zahid, J. A. Cunningham // *Fluids*. – 2025. – Vol. 10. – Review of the Color Gradient Lattice Boltzmann Method for Simulating Multi-Phase Flow in Porous Media. – № 5. – P. 128.
2. Ternary Free-Energy Entropic Lattice Boltzmann Model with a High Density Ratio / M. Wöhrwag, C. Semperebon, A. Mazloomi Moqaddam [et al.] // *Physical Review Letters*. – 2018. – Vol. 120. – № 23. – P. 234501.
3. A brief review of the phase-field-based lattice Boltzmann method for multiphase flows / H. Wang, X. Yuan, H. Liang [et al.] // *Capillarity*. – 2019. – Vol. 2. – № 3. – P. 33-52.
4. Bagheri, M. An improved three-dimensional lattice Boltzmann-volume of fluid (LB-VOF) method for simulation of laminar liquid jet breakup with high density ratio / M. Bagheri, M. Siavashi, S. Mousavi // *Computers & Fluids*. – 2023. – Vol. 263. – P. 105969.
5. Oberkampf, W. L. Verification and validation in computational fluid dynamics / W. L. Oberkampf, T. G. Trucano.
6. Semperebon, C. Ternary free-energy lattice Boltzmann model with tunable surface tensions and contact angles / C. Semperebon, T. Krüger, H. Kusumaatmaja // *Physical Review E*. – 2016. – Vol. 93. – № 3. – P. 033305.
7. Zhang, D. Lattice Boltzmann simulation of three phase reactive flow in random porous media at pore-scale / D. Zhang, S. Li, Y. Li // *Applied Thermal Engineering*. – 2021. – Vol. 194. – P. 117140.
8. Pore-scale numerical prediction of three-phase relative permeability in porous media using the lattice Boltzmann method / X. Zhu, S. Wang, Q. Feng [et al.] // *International Communications in Heat and Mass Transfer*. – 2021. – Vol. 126. – P. 105403.
9. Lenormand, R. Mechanisms of the displacement of one fluid by another in a network of capillary ducts / R. Lenormand, C. Zarcone, A. Sarr // *Journal of Fluid Mechanics*. – 1983. – Vol. 135. – № 1. – P. 337.
10. Three-dimensional lattice Boltzmann flux solver for three-phase/component flow / D. Zhang, Y. Li, L. Gong [et al.] // *Physics of Fluids*. – 2024. – Vol. 36. – № 8. – P. 083350.
11. Investigating snap-off behavior during spontaneous imbibition in 3D pore-throat model by pseudopotential lattice Boltzmann method / J. Zheng, X. Qi, W. Gong [et al.] // *Advances in Water Resources*. – 2024. – Vol. 190. – P. 104751.

12. A versatile lattice Boltzmann model for immiscible ternary fluid flows / Y. Yu, H. Liu, D. Liang, Y. Zhang // *Physics of Fluids*. – 2019. – Vol. 31. – № 1. – P. 012108.
13. Investigating snap-off behavior during spontaneous imbibition in 3D pore-throat model by pseudopotential lattice Boltzmann method / J. Zheng, X. Qi, W. Gong [et al.] // *Advances in Water Resources*. – 2024. – Vol. 190. – P. 104751.

## Приложение А. Код теста на вязкость

```
#include <cmath>

#include <vector>

#include <string>

#include <iostream>

#include "palabos2D.h"

#include "palabos2D.hh"

using namespace plb;

using std::vector;

using std::string;

using std::cout;

using std::endl;

typedef double T;

#define DESCRIPTOR descriptors::D2Q9Descriptor

template<typename T, template<typename U> class Descriptor>

class VelocitySinInitializer : public

OneCellIndexedWithRandFunctional2D<T,Descriptor> {

public:

VelocitySinInitializer(T u0_, T k_, T rho0_)

: u0(u0_), k(k_), rho0(rho0_) {}
```

```

VelocitySinInitializer<T,Descriptor>* clone() const override {
    return new VelocitySinInitializer(*this);
}

void execute(plint, plint iY, T, Cell<T,Descriptor>& cell) const override {
    T u = u0 * std::sin(k * static_cast<T>(iY));
    iniCellAtEquilibrium(cell, rho0, Array<T,2>(u, 0.));
}

private:
    T u0, k, rho0;
};

int main(int argc, char* argv[]) {
    plbInit(&argc, &argv);

    const plint nx = 100, ny = 100;
    const T rho0 = 1.0;
    const T u0 = 0.01;
    const T L = static_cast<T>(ny);
    const T k = 2.0 * M_PI / L;
    const plint maxIter = 5000;
    const plint statIter = 50;
    const T tolRel = 0.01;

```

```

vector<string> phaseNames = {"Water", "Oil", "Gas"};

vector<T> tauList = {0.7, 0.8, 0.9};

bool allPassed = true;

for (size_t p = 0; p < phaseNames.size(); ++p) {

    T tau = tauList[p];

    T omega = 1.0 / tau;

    T nu_ref = (tau - T(0.5)) / T(3.0);

    cout << "=== Testing viscosity for phase: " << phaseNames[p]
         << " (tau=" << tau << ", nu_ref=" << nu_ref << ") ===" << endl;

    MultiBlockLattice2D<T,DESCRIPTOR> lattice(
        nx, ny, new BGKdynamics<T,DESCRIPTOR>(omega)
    );

    lattice.periodicity().toggle(0,true);

    lattice.periodicity().toggle(1,true);

    applyIndexed(
        lattice,
        Box2D(0,nx-1, 0,ny-1),
        new VelocitySinInitializer<T,DESCRIPTOR>(u0, k, rho0)

```

```

);

lattice.initialize();

vector<T> times, lnE;

for (plint iter = 0; iter <= maxIter; ++iter) {

    lattice.collideAndStream();

    if (iter % statIter == 0) {

        // Compute average kinetic energy  $E = \langle u_x^2 \rangle$ 

        std::unique_ptr<MultiTensorField2D<T,2>> velocity(

            computeVelocity(lattice)

        );

        T sum = 0;

        for (plint iX = 0; iX < nx; ++iX) {

            for (plint iY = 0; iY < ny; ++iY) {

                Array<T,2> u = velocity->get(iX,iY);

                sum += u[0]*u[0];

            }

        }

        T E = sum / (nx * ny);

        times.push_back(static_cast<T>(iter));

        lnE. push_back(std::log(E));

    }
}

```

```

}

int M = times.size();

T sum_t=0, sum_lnE=0, sum_t2=0, sum_tlnE=0;

for (int i = 0; i < M; ++i) {

    sum_t += times[i];

    sum_lnE += lnE[i];

    sum_t2 += times[i]*times[i];

    sum_tlnE += times[i]*lnE[i];

}

T m = ( M*sum_tlnE - sum_t*sum_lnE )

    / ( M*sum_t2 - sum_t*sum_t );

T nu_num = - m / (2.0 * k * k);

cout << "Measured nu = " << nu_num

    << ", Reference nu = " << nu_ref << endl;

// PASS/FAIL

T errRel = std::fabs(nu_num - nu_ref) / nu_ref;

if (errRel <= tolRel) {

    cout << "*** " << phaseNames[p] << " viscosity test PASSED (rel.err="

        << errRel << ") ***" << endl;

} else {

```

```

    cout << "*** " << phaseNames[p] << " viscosity test FAILED (rel.err="
        << errRel << ") ***" << endl;

    allPassed = false;

}

cout << endl;

}

return allPassed ? EXIT_SUCCESS : EXIT_FAILURE;

}

```

### **Приложение Б. Код теста «Жидкая линза»**

```

#include <cstdlib>

#include <iostream>

#include <vector>

#include "palabos2D.h"

#include "palabos2D.hh"

#include "boundaryCondition/spongeZones2D.h"

#include "boundaryCondition/neumannCondition2D.h"

#include <fstream>

using namespace plb;

using namespace std;

```

```

typedef double T;

#define DESCRIPTOR descriptors::ForcedShanChenD2Q9Descriptor

template <typename T, template <typename U> class Descriptor>
class LiquidLensInitializer : public OneCellIndexedWithRandFunctional2D<T,
Descriptor> {
public:
    LiquidLensInitializer(plint nx_, plint ny_, T cx_, T cy_, T R_) : nx(nx_),
ny(ny_), cx(cx_), cy(cy_), R(R_) {}

    LiquidLensInitializer<T, Descriptor> *clone() const override {
        return new LiquidLensInitializer<T, Descriptor>(*this);
    }

    void execute(plint iX, plint iY, T rand_val, Cell<T, Descriptor> &cell) const
    override {
        // Фазы: 1 - капля, 2 - снизу, 3 - сверху

        T densityFluctuations = 1.e-3;

        T almostNoFluid = 1e-4;

        Array<T,2> zeroVelocity(0., 0.);

        T dx = iX - cx, dy = iY - cy;

        T r2 = dx*dx + dy*dy;

```

```

T rho = almostNoFluid;

// Fluid 1 — круглая капля
if (r2 < R*R) {
    rho = 1.0 + rand_val * densityFluctuations;
}

iniCellAtEquilibrium(cell, rho, zeroVelocity);
}

private:

plint nx, ny;

T cx, cy, R;

};

template <typename T, template <typename U> class Descriptor>
class LiquidLensBackInitializer : public
OneCellIndexedWithRandFunctional2D<T, Descriptor> {
public:
    // phaseFlag = 2 — снизу (Fluid 2); phaseFlag = 3 — сверху (Fluid 3)
    LiquidLensBackInitializer(plint nx_, plint ny_, T cx_, T cy_, T R_, int
phaseFlag_) :
        nx(nx_), ny(ny_), cx(cx_), cy(cy_), R(R_), phaseFlag(phaseFlag_) {}
    LiquidLensBackInitializer<T, Descriptor> *clone() const override {
        return new LiquidLensBackInitializer<T, Descriptor>(*this);
    }
};

```

```

}

void execute(plint iX, plint iY, T rand_val, Cell<T, Descriptor> &cell) const
override {

    T densityFluctuations = 1.e-3;

    T almostNoFluid = 1e-4;

    Array<T,2> zeroVelocity(0., 0.);

    T dx = iX - cx, dy = iY - cy;

    T r2 = dx*dx + dy*dy;

    T rho = almostNoFluid;

    if (r2 >= R*R) {

        if (phaseFlag == 2 && iY < cy) {

            rho = 1.0 + rand_val * densityFluctuations;

        } else if (phaseFlag == 3 && iY >= cy) {

            rho = 1.0 + rand_val * densityFluctuations;

        }

    }

    iniCellAtEquilibrium(cell, rho, zeroVelocity);

}

private:

    plint nx, ny;

    T cx, cy, R;

```

```

    int phaseFlag;

};

void setupLiquidLens(
    MultiBlockLattice2D<T, DESCRIPTOR> &fluid1,
    MultiBlockLattice2D<T, DESCRIPTOR> &fluid2,
    MultiBlockLattice2D<T, DESCRIPTOR> &fluid3,
    plint nx, plint ny,
    T cx, T cy, T R)
{
    T wallRho = 1.0;

    defineDynamics(fluid1, Box2D(0, nx-1, 0, 0), new BounceBack<T,
DESCRIPTOR>(wallRho));

    defineDynamics(fluid2, Box2D(0, nx-1, 0, 0), new BounceBack<T,
DESCRIPTOR>(wallRho));

    defineDynamics(fluid3, Box2D(0, nx-1, 0, 0), new BounceBack<T,
DESCRIPTOR>(wallRho));

    defineDynamics(fluid1, Box2D(0, nx-1, ny-1, ny-1), new BGKdynamics<T,
DESCRIPTOR>(1.0));

    defineDynamics(fluid2, Box2D(0, nx-1, ny-1, ny-1), new BGKdynamics<T,
DESCRIPTOR>(1.0));

    defineDynamics(fluid3, Box2D(0, nx-1, ny-1, ny-1), new BGKdynamics<T,
DESCRIPTOR>(1.0));
}

```

```
    initializeAtEquilibrium(fluid1, Box2D(0, nx-1, ny-1, ny-1), T(1.0),  
Array<T,2>(0.0, 0.0));
```

```
    initializeAtEquilibrium(fluid2, Box2D(0, nx-1, ny-1, ny-1), T(1.0),  
Array<T,2>(0.0, 0.0));
```

```
    initializeAtEquilibrium(fluid3, Box2D(0, nx-1, ny-1, ny-1), T(1.0),  
Array<T,2>(0.0, 0.0));
```

```
    applyIndexed(fluid1, Box2D(0, nx-1, 0, ny-1), new LiquidLensInitializer<T,  
DESCRIPTOR>(nx, ny, cx, cy, R));
```

```
    applyIndexed(fluid2, Box2D(0, nx-1, 0, ny-1), new  
LiquidLensBackInitializer<T, DESCRIPTOR>(nx, ny, cx, cy, R, 2));
```

```
    applyIndexed(fluid3, Box2D(0, nx-1, 0, ny-1), new  
LiquidLensBackInitializer<T, DESCRIPTOR>(nx, ny, cx, cy, R, 3));
```

```
    fluid1.initialize();
```

```
    fluid2.initialize();
```

```
    fluid3.initialize();
```

```
}
```

```
void writeVTK(  

```

```
    MultiBlockLattice2D<T, DESCRIPTOR> &fluid1,  

```

```
    MultiBlockLattice2D<T, DESCRIPTOR> &fluid2,  

```

```

MultiBlockLattice2D<T, DESCRIPTOR> &fluid3,

plint iter

) {

    T dx = 1.0;

    VtkImageOutput2D<T> vtkOut(createFileName("vtk", iter, 6), dx);

    auto rho1 = computeDensity(fluid1);
    auto rho2 = computeDensity(fluid2);
    auto rho3 = computeDensity(fluid3);

    Box2D domain = fluid1.getBoundingBox();

    auto rhoTotal = *rho1;
    add(rhoTotal, *rho2, rhoTotal, domain);
    add(rhoTotal, *rho3, rhoTotal, domain);

    vtkOut.writeData<float>(*rho1, "rho1", dx);
    vtkOut.writeData<float>(*rho2, "rho2", dx);
    vtkOut.writeData<float>(*rho3, "rho3", dx);
    vtkOut.writeData<float>(rhoTotal, "rho_total", dx);

    vtkOut.writeData<2, float>(*computeVelocity(fluid1), "velocity1", dx);
}

```

```

void computeContactAngle(
    MultiBlockLattice2D<T, DESCRIPTOR>& fluid,
    plint nx, plint ny, plint iter)
{
    static std::ofstream out("contact_angle.csv", std::ios::app);

    const T rhoIso = 0.5;
    const T eps = 0.005;

    std::vector<std::pair<T,T>> pts;
    auto rhoField = computeDensity(fluid);
    for (plint iX = 1; iX < nx-1; ++iX) {
        for (plint iY = 1; iY < ny-1; ++iY) {
            T r = rhoField->get(iX,iY);
            if (std::abs(r - rhoIso) < eps) {
                pts.emplace_back((T)iX, (T)iY);
            }
        }
    }

    if (pts.size() < 20) return;

    T cx = nx/2.0;

```

```

std::vector<std::pair<T,T>> leftPts, rightPts;

for (auto &p : pts) {
    (p.first < cx ? leftPts : rightPts).push_back(p);
}

if (leftPts.size() < 5 || rightPts.size() < 5) return;

auto fitEdge = [&](const std::vector<std::pair<T,T>>& P, bool takeMaxX){
    auto v = P;

    std::sort(v.begin(), v.end(), [&](auto &a, auto &b){
        return takeMaxX ? a.first < b.first : a.first > b.first;
    });

    const size_t N = std::min<size_t>(v.size(), 10);

    T sumX=0, sumY=0;

    for (size_t i = 0; i < N; ++i) {
        sumX += v[i].first;

        sumY += v[i].second;
    }

    T x0 = sumX / N, y0 = sumY / N;

    T Sx=0, Sy=0, Sxx=0, Sxy=0;

    for (size_t i = 0; i < N; ++i) {
        T x = v[i].first - x0;

        T y = v[i].second - y0;

```

```

    Sx += x;

    Sy += y;

    Sxx += x*x;

    Sxy += x*y;

}

T m = Sxy / Sxx;

T angle = std::atan(m) * 180.0 / M_PI;

if (angle < 0) angle += 360.0;

return std::tuple<T,T,T>(x0, y0, angle);

};

T xL, yL, angleL;

T xR, yR, angleR;

std::tie(xL, yL, angleL) = fitEdge(leftPts, true);

std::tie(xR, yR, angleR) = fitEdge(rightPts, false);

// Средний контактный угол (Юнга)

T contactAngle = (180.0 - angleL + angleR) / 2.0;

out << iter << ", "

    << xL << ", " << yL << ", " << angleL << ", "

```

```

    << xR << "," << yR << "," << angleR << ","
    << contactAngle << "\n";
}

```

```

void computeMass(
    MultiBlockLattice2D<T, DESCRIPTOR>& fluid,
    const string& filename,
    plint iter
) {
    static ofstream out(filename, ios::app);
    auto rho = computeDensity(fluid);
    Box2D domain = fluid.getBoundingBox();
    T sum = 0;
    for (plint iX = domain.x0; iX <= domain.x1; ++iX) {
        for (plint iY = domain.y0; iY <= domain.y1; ++iY) {
            sum += rho->get(iX, iY);
        }
    }
    out << iter << "," << sum << std::endl;
}

```

```

void computeSymmetry(
    MultiBlockLattice2D<T, DESCRIPTOR>& fluid,

```

```

plint nx, plint ny,

const string& filename,

plint iter
) {

    static ofstream out(filename, ios::app);

    const T rhoIso = 0.5;

    const T eps = 0.005;

    vector<pair<T,T>> pts;

    auto rho = computeDensity(fluid);

    for (plint iX = 1; iX < nx-1; ++iX) {

        for (plint iY = 1; iY < ny-1; ++iY) {

            T r = rho->get(iX, iY);

            if (fabs(r - rhoIso) < eps) pts.emplace_back((T)iX, (T)iY);

        }

    }

    if (pts.size() < 20) return;

    // разделение по центру

    T cx = nx/2.0;

    vector<pair<T,T>> left, right;

    for (auto &p : pts) (p.first < cx ? left : right).push_back(p);

    if (left.size() < 5 || right.size() < 5) return;

    auto fitSlope = [&](const vector<pair<T,T>>& P) {

        T x0=0, y0=0;

```

```

for (auto &p:P) { x0+=p.first; y0+=p.second; }

x0/=P.size(); y0/=P.size();

T Sxx=0, Sxy=0;

for (auto &p:P) {

    T dx = p.first - x0;

    T dy = p.second- y0;

    Sxx += dx*dx;

    Sxy += dx*dy;

}

return atan2(Sxy, Sxx) * 180.0/M_PI;

};

T aL = fitSlope(left);

T aR = fitSlope(right);

T diff = fabs(aL - aR);

out << iter << "," << aL << "," << aR << "," << diff << std::endl;

}

```

```

int main(int argc, char *argv[])

{

    plbInit(&argc, &argv);

    global::directories().setOutputDir("./tmp/");

```

```

// Сетка

plint nx = 600;

plint ny = 600;

T cx = nx/2.0, cy = ny/2.0;

T R = 60.0; // Радиус капли

const T omega1 = 1.0, omega2 = 0.8, omega3 = 1.2;

const T G12 = 1.3, G13 = 1.5, G23 = 1.4;

MultiBlockLattice2D<T, DESCRIPTOR> fluid1(nx, ny, new
ExternalMomentRegularizedBGKdynamics<T, DESCRIPTOR>(omega1));

MultiBlockLattice2D<T, DESCRIPTOR> fluid2(nx, ny, new
ExternalMomentRegularizedBGKdynamics<T, DESCRIPTOR>(omega2));

MultiBlockLattice2D<T, DESCRIPTOR> fluid3(nx, ny, new
ExternalMomentRegularizedBGKdynamics<T, DESCRIPTOR>(omega3));

fluid1.periodicity().toggle(0, true);

fluid2.periodicity().toggle(0, true);

fluid3.periodicity().toggle(0, true);

std::vector<std::vector<T>> speciesG = {
    { 0.0, G12, G13 },
    { G12, 0.0, G23 },

```

```

    { G13, G23, 0.0 }

};

std::vector<T> constOmegaValues = {omega1, omega2, omega3};

vector<MultiBlockLattice2D<T, DESCRIPTOR>*> blockLattices = {&fluid1,
&fluid2, &fluid3};

plint processorLevel = 1;

integrateProcessingFunctional(
    new ShanChenMultiComponentProcessor2D<T, DESCRIPTOR>(speciesG,
constOmegaValues),
    Box2D(0, nx-1, 1, ny-2), blockLattices, processorLevel
);

setupLiquidLens(fluid1, fluid2, fluid3, nx, ny, cx, cy, R);

// === Sponge-зона сверху (демпфирование волн на верхней границе) ===
plint spongeWidth = 100;

Array<plint, 4> spongeCells = {0, 0, 0, spongeWidth};

Array<T, 4> translation    = {0., 0., 0., 0.5};

Array<T, 4> scale          = {1., 1., 1., 0.12};

T bulkOmega = omega3;

```

```

integrateProcessingFunctional<T, DESCRIPTOR>(
    new ViscositySpongeZone2D<T, DESCRIPTOR>(
        nx, ny, bulkOmega, spongeCells, translation, scale
    ),
    Box2D(0, nx - 1, ny - spongeWidth, ny - 1),
    fluid3
);

const plint maxIter = 10000;

const plint saveIter = 10;

for (plint iT = 0; iT < maxIter; ++iT) {

    if (iT % saveIter == 0) {

        writeVTK(fluid1, fluid2, fluid3, iT);

        computeContactAngleBetween1and3(fluid1, nx, ny, iT);

        computeMass(fluid1, "mass_fluid1.csv", iT);

        computeMass(fluid2, "mass_fluid2.csv", iT);

        computeMass(fluid3, "mass_fluid3.csv", iT);

        computeSymmetry(fluid1, nx, ny, "symmetry_fluid1.csv", iT);

    }
}

```

```

    fluid3.collideAndStream();

    fluid2.collideAndStream();

    fluid1.collideAndStream();

}

pcout << "Liquid lens simulation done!" << endl;

}

```

### **Приложение В. Код теста «Вложенная капля»**

```

#include <cstdlib>

#include <iostream>

#include <vector>

#include "palabos2D.h"

#include "palabos2D.hh"

#include "boundaryCondition/spongeZones2D.h"

#include "boundaryCondition/neumannCondition2D.h"

#include <fstream>

using namespace plb;

using namespace std;

typedef double T;

#define DESCRIPTOR descriptors::ForcedShanChenD2Q9Descriptor

```

```

template <typename T, template <typename U> class Descriptor>
class NestedDropletInitializer : public OneCellIndexedWithRandFunctional2D<T,
Descriptor> {
public:
    NestedDropletInitializer(
        plint nx_, plint ny_,
        T cx_, T cy_,
        T R_oil_, T R_gas_
    ):
        nx(nx_), ny(ny_), cx(cx_), cy(cy_),
        R_oil(R_oil_), R_gas(R_gas_)
    {}

    NestedDropletInitializer<T,Descriptor>* clone() const override {
        return new NestedDropletInitializer<T,Descriptor>(*this);
    }

    // phaseFlag = 1 (нефть), 2 (газ), 3 (вода)
    void setPhaseFlag(int flag) {
        phaseFlag = flag;
    }
}

```

```

void execute(plint iX, plint iY, T rand_val, Cell<T,Descriptor>& cell) const
override {

    const T densityFluctuations = 1.e-4;

    const T almostNoFluid    = 1e-6;

    Array<T,2> zeroVelocity(0., 0.);

    T dx = iX - cx;

    T dy = iY - cy;

    T r2 = dx*dx + dy*dy;

    T rhoLiq = 1.0;

    T rhoGas = 0.4;

    T rhoOil = 0.8;

    T rho = almostNoFluid;

    // Внутренний круг (r <= R_oil) → нефть
    if (r2 < R_oil*R_oil && phaseFlag==1) {
        rho = rhoOil + (rand_val - 0.5)*densityFluctuations;
    }

    // Оболочка газа: между R_oil <= r < R_gas

```

```

else if (r2 >= R_oil*R_oil && r2 < R_gas*R_gas && phaseFlag==2) {
    rho = rhoGas + (rand_val - 0.5)*densityFluctuations;
}

// ФОН — вода: r >= R_gas
else if (r2 >= R_gas*R_gas && phaseFlag==3) {
    rho = rhoLiq + (rand_val - 0.5)*densityFluctuations;
}

iniCellAtEquilibrium(cell, rho, zeroVelocity);
}

```

private:

```

    plint nx, ny;

    T cx, cy;

    T R_oil, R_gas;

    int phaseFlag;
};

```

void setupNestedDroplet(

```

    MultiBlockLattice2D<T,DESCRIPTOR>& fluidOil,
    MultiBlockLattice2D<T,DESCRIPTOR>& fluidGas,
    MultiBlockLattice2D<T,DESCRIPTOR>& fluidWater,

```

```

    plint nx, plint ny,

    T cx, T cy,

    T R_oil, T R_gas
) {

    fluidOil .periodicity().toggle(0, true);

    fluidGas .periodicity().toggle(0, true);

    fluidWater.periodicity().toggle(0, true);

    defineDynamics(fluidOil,          Box2D(0,nx-1,0,0),          new
BGKdynamics<T,DESCRIPTOR>(1.0));

    defineDynamics(fluidGas,          Box2D(0,nx-1,0,0),          new
BGKdynamics<T,DESCRIPTOR>(1.0));

    defineDynamics(fluidWater,       Box2D(0,nx-1,0,0),          new
BGKdynamics<T,DESCRIPTOR>(1.0));

    defineDynamics(fluidOil,          Box2D(0,nx-1,ny-1,ny-1),new
BGKdynamics<T,DESCRIPTOR>(1.0));

    defineDynamics(fluidGas,          Box2D(0,nx-1,ny-1,ny-1),new
BGKdynamics<T,DESCRIPTOR>(1.0));

    defineDynamics(fluidWater,       Box2D(0,nx-1,ny-1,ny-1),new
BGKdynamics<T,DESCRIPTOR>(1.0));

    initializeAtEquilibrium(fluidOil,  Box2D(0,nx-1,0,0),          T(1.0),
Array<T,2>(0.0,0.0));

```

```
    initializeAtEquilibrium(fluidGas,      Box2D(0,nx-1,0,0),      T(1.0),  
Array<T,2>(0.0,0.0));
```

```
    initializeAtEquilibrium(fluidWater,   Box2D(0,nx-1,0,0),      T(1.0),  
Array<T,2>(0.0,0.0));
```

```
    initializeAtEquilibrium(fluidOil,     Box2D(0,nx-1,ny-1,ny-1),T(1.0),  
Array<T,2>(0.0,0.0));
```

```
    initializeAtEquilibrium(fluidGas,     Box2D(0,nx-1,ny-1,ny-1),T(1.0),  
Array<T,2>(0.0,0.0));
```

```
    initializeAtEquilibrium(fluidWater,   Box2D(0,nx-1,ny-1,ny-1),T(1.0),  
Array<T,2>(0.0,0.0));
```

```
    auto initOil = new NestedDropletInitializer<T,DESCRIPTOR>(nx, ny, cx, cy,  
R_oil, R_gas);
```

```
    initOil->setPhaseFlag(1);
```

```
    auto initGas = new NestedDropletInitializer<T,DESCRIPTOR>(nx, ny, cx, cy,  
R_oil, R_gas);
```

```
    initGas->setPhaseFlag(2);
```

```
    auto initWater = new NestedDropletInitializer<T,DESCRIPTOR>(nx, ny, cx,  
cy, R_oil, R_gas);
```

```
    initWater->setPhaseFlag(3);
```

```
    applyIndexed(fluidOil, Box2D(0,nx-1,0,ny-1), initOil);
```

```
    applyIndexed(fluidGas, Box2D(0,nx-1,0,ny-1), initGas);
```

```
    applyIndexed(fluidWater, Box2D(0,nx-1,0,ny-1), initWater);
```

```

fluidOil .initialize();

fluidGas .initialize();

fluidWater.initialize();

}

```

```

void writeVTK(

    MultiBlockLattice2D<T,DESCRIPTOR>& fluidOil,

    MultiBlockLattice2D<T,DESCRIPTOR>& fluidGas,

    MultiBlockLattice2D<T,DESCRIPTOR>& fluidWater,

    plint iter

) {

    T dx = 1.0;

    VtkImageOutput2D<T> vtkOut(createFileName("vtk", iter, 6), dx);

    auto rhoOil = computeDensity(fluidOil);

    auto rhoGas = computeDensity(fluidGas);

    auto rhoWater = computeDensity(fluidWater);

    Box2D domain = fluidOil.getBoundingBox();

    auto rhoTotal = *rhoOil;

    add(rhoTotal, *rhoGas, rhoTotal, domain);

```

```

add(rhoTotal, *rhoWater, rhoTotal, domain);

vtkOut.writeData<float>(*rhoOil, "rho_oil", dx);
vtkOut.writeData<float>(*rhoGas, "rho_gas", dx);
vtkOut.writeData<float>(*rhoWater, "rho_water", dx);
vtkOut.writeData<float>(rhoTotal, "rho_total", dx);

vtkOut.writeData<2,float>(*computeVelocity(fluidOil), "vel_oil", dx);
vtkOut.writeData<2,float>(*computeVelocity(fluidGas), "vel_gas", dx);
vtkOut.writeData<2,float>(*computeVelocity(fluidWater), "vel_water", dx);
}

void computeContactAngle_OilOnWater(
    MultiBlockLattice2D<T,DESCRIPTOR>& fluidOil,
    plint nx, plint ny,
    plint iter
) {
    auto rho = computeDensity(fluidOil);
    T rhoIso = 0.5*(0.8 + 1.0);

    vector<plint> xList, yList;
    for (plint iX=1; iX<nx-1; ++iX) {
        for (plint iY=1; iY<ny-1; ++iY) {

```

```

T val = rho->get(iX, iY);

if (fabs(val - rhoIso) < 1.e-2) {

    xList.push_back(iX);

    yList.push_back(iY);

}

}

}

if (xList.size()<5) return;

plint leftIdx = min_element(xList.begin(), xList.end()) - xList.begin();
plint rightIdx = max_element(xList.begin(), xList.end()) - xList.begin();

plint xL = xList[leftIdx], yL = yList[leftIdx];

plint xR = xList[rightIdx], yR = yList[rightIdx];

T dyL = (rho->get(xL, yL+1) - rho->get(xL, yL-1)) / 2.0;
T dxL = (rho->get(xL+1, yL) - rho->get(xL-1, yL)) / 2.0;
T angleL = atan2(dyL, dxL) * 180.0 / M_PI;

T dyR = (rho->get(xR, yR+1) - rho->get(xR, yR-1)) / 2.0;
T dxR = (rho->get(xR+1, yR) - rho->get(xR-1, yR)) / 2.0;
T angleR = atan2(dyR, dxR) * 180.0 / M_PI;

T contactAngle = (180.0 - fabs(angleL) + fabs(angleR)) / 2.0;

```

```
ofstream out("contact_angle.csv", ios::app);  
out << iter << "," << angleL << "," << angleR << "," << contactAngle << endl;  
}
```

```
int main(int argc, char* argv[]) {  
    plbInit(&argc, &argv);  
    global::directories().setOutputDir("./tmp/");
```

```
    plint nx = 400;
```

```
    plint ny = 400;
```

```
    T cx = nx/2.0, cy = ny/2.0;
```

```
    T R_oil = 60.0;
```

```
    T R_gas = 120.0;
```

```
    const T omegaOil = 1.0;
```

```
    const T omegaGas = 1.0;
```

```
    const T omegaWater = 1.0;
```

```
    T G_oil_gas = 2.2;
```

```
    T G_oil_water = 2.5;
```

```
T G_water_gas = 2.8;
```

```
std::vector<std::vector<T>> speciesG = {  
    { 0.0,    G_oil_gas, G_oil_water },  
    { G_oil_gas, 0.0,    G_water_gas },  
    { G_oil_water, G_water_gas, 0.0    }  
};
```

```
std::vector<T> constOmegaValues = {omegaOil, omegaGas, omegaWater};
```

```
MultiBlockLattice2D<T,DESCRIPTOR>    fluidOil(    nx,    ny,    new  
ExternalMomentRegularizedBGKdynamics<T,DESCRIPTOR>(omegaOil) );
```

```
MultiBlockLattice2D<T,DESCRIPTOR>    fluidGas(    nx,    ny,    new  
ExternalMomentRegularizedBGKdynamics<T,DESCRIPTOR>(omegaGas) );
```

```
MultiBlockLattice2D<T,DESCRIPTOR>    fluidWater(nx,    ny,    new  
ExternalMomentRegularizedBGKdynamics<T,DESCRIPTOR>(omegaWater) );
```

```
fluidOil .periodicity().toggle(0, true);
```

```
fluidGas .periodicity().toggle(0, true);
```

```
fluidWater.periodicity().toggle(0, true);
```

```
vector<MultiBlockLattice2D<T,DESCRIPTOR>*>    blockLattices    =    {  
&fluidOil, &fluidGas, &fluidWater };
```

```
plint processorLevel = 1;
```

```
integrateProcessingFunctional(  

```

```

    new    ShanChenMultiComponentProcessor2D<T,DESCRIPTOR>(speciesG,
constOmegaValues),

    Box2D(1, nx-2, 1, ny-2),

    blockLattices,

    processorLevel

);

    setupNestedDroplet(fluidOil, fluidGas, fluidWater, nx, ny, cx, cy, R_oil,
R_gas);

const plint maxIter = 5000;

for (plint iter = 0; iter <= maxIter; ++iter) {

    if (iter % 100 == 0) {

        writeVTK(fluidOil, fluidGas, fluidWater, iter);

        computeContactAngle_OilOnWater(fluidOil, nx, ny, iter);

    }

    fluidWater.collideAndStream();

    fluidGas.collideAndStream();

    fluidOil.collideAndStream();

}

pcout << "Nested Droplet (3-phase) simulation finished!" << std::endl;

return 0;

```

}