

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
имени М.В. ЛОМОНОСОВА  
МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

*На правах рукописи*

**Романова Дарья Игоревна**

**Трёхмерное математическое моделирование природных  
склоновых потоков с учетом сложной реологии,  
турбулентности и захвата подстилающего материала**

1.1.9 — Механика жидкости, газа и плазмы

ДИССЕРТАЦИЯ на соискание учёной степени  
кандидата физико-математических наук

Научный руководитель:  
доктор физико-математических наук, профессор  
Эглит Маргарита Эрнестовна

Москва — 2023

## Оглавление

	Стр.
Введение . . . . .	5
<b>Глава 1. Обзор литературы . . . . .</b>	<b>14</b>
1.1 Обзор работ с использованием осредненного по глубине подхода . . . . .	14
1.2 Обзор работ с использованием трёхмерного подхода . . . . .	16
1.3 Обзор используемых для склоновых потоков реологических моделей . . . . .	18
1.4 Обзор существующих моделей захвата подстилающего материала потоком в горах . . . . .	20
<b>Глава 2. Математическая модель . . . . .</b>	<b>25</b>
2.1 Уравнения движения и уравнение неразрывности . . . . .	25
2.2 Реологическая модель и её регуляризация . . . . .	26
2.2.1 Реологическая модель Хершеля-Балкли . . . . .	26
2.2.2 Регуляризация модели Хершеля-Балкли . . . . .	28
2.3 Модели турбулентности . . . . .	30
2.3.1 $k$ - $\epsilon$ модель турбулентности . . . . .	31
2.3.2 $k$ - $\omega$ $SST$ модель турбулентности . . . . .	32
2.4 Начальные условия . . . . .	33
2.5 Граничные условия на свободной поверхности . . . . .	33
2.6 Граничные условия на дне потока . . . . .	34
<b>Глава 3. Трёхмерная модель склонового потока без захвата     склонового материала . . . . .</b>	<b>36</b>
3.1 Определение формы свободной поверхности . . . . .	36
3.2 Результирующая система уравнений . . . . .	39
3.3 Вычислительная модель . . . . .	40
3.3.1 Пакет OpenFOAM . . . . .	40
3.3.2 Используемые методы аппроксимации членов уравнений . . . . .	41
3.3.3 Алгоритм решения системы уравнений . . . . .	42
3.3.4 Методы решения СЛАУ . . . . .	43
3.4 Разработка модуля генерации сетки и начальных условий . . . . .	43

3.5	Примеры применения трехмерной модели без захвата подстилающего материала . . . . .	44
3.5.1	22-ой лавинный очаг горы Юкспор (Хибины) . . . . .	44
3.5.2	Эксперимент по исследованию взаимодействия водоснежного потока с комплексом защитных сооружений (Университет Исландии) . . . . .	52
3.6	Выводы . . . . .	57
<b>Глава 4. Разработка модели разрушения потоком дна и вовлечения донного материала в движение . . . . .</b>		<b>59</b>
4.1	Описание разрабатываемого подхода . . . . .	59
4.2	Гипотеза о захвате . . . . .	61
4.3	Математическая модель разрушения потоком дна и вовлечения донного материала в движение . . . . .	61
4.4	Результатирующая система уравнений . . . . .	63
4.5	Численная модель разрушения потоком дна и вовлечения донного материала в движение . . . . .	64
4.6	Разработка решателя avalancheFoam для расчёта динамики склонового потока с захватом материала на дне в пакете OpenFOAM . . . . .	66
4.7	Примеры применения трехмерной модели с захватом подстилающего материала . . . . .	67
4.7.1	Модельная задача течения потока на склоне постоянного уклона . . . . .	67
4.7.2	Прорыв ледникового озера Малый Азау . . . . .	77
4.8	Выводы . . . . .	83
<b>Глава 5. Оптимизация коэффициентов турбулентных моделей с использованием методов машинного обучения . . . . .</b>		<b>86</b>
5.1	Экспериментальные данные полученные в НИИ Механики МГУ . . . . .	88
5.2	Алгоритм оптимизации коэффициентов моделей турбулентности . . . . .	89
5.3	Разработка модуля автоматической оптимизации MLTFoamOptimization . . . . .	91

	Стр.
5.4 Оптимизация коэффициентов $k$ - $\epsilon$ и $k$ - $\omega$ $SST$ моделей турбулентности . . . . .	92
5.5 Валидация полученных коэффициентов $k$ - $\epsilon$ и $k$ - $\omega$ $SST$ моделей турбулентности . . . . .	94
5.6 Выводы . . . . .	96
<b>Заключение . . . . .</b>	<b>98</b>
<b>Список сокращений и условных обозначений . . . . .</b>	<b>100</b>
<b>Словарь терминов . . . . .</b>	<b>104</b>
<b>Список литературы . . . . .</b>	<b>105</b>
<b>Список рисунков . . . . .</b>	<b>120</b>
<b>Список таблиц . . . . .</b>	<b>123</b>
<b>Приложение А. Библиотека avalancheUtils для генерации расчётной сетки, начальных и граничных условий . . . . .</b>	<b>124</b>
<b>Приложение Б. Код решателя avalancheFoam для многофазного потока неньютоновской среды с учетом захвата донного материала . . . . .</b>	<b>143</b>
<b>Приложение В. Модуль автоматической оптимизации коэффициентов турбулентных моделей MLTFoamOptimization . . . . .</b>	<b>164</b>



## Введение

**Актуальность темы.** В настоящей диссертации рассматриваются различные склоновые потоки. Определим понятие склоновые потоки – это потоки, возникающие внезапно на склонах гор, такие, как, например, снежные лавины, сели, оползни, каменные обвалы, потоки лавы. В настоящей работе в первую очередь рассматриваются плотные снежные лавины, но разрабатываемая модель может быть использована для исследования и других типов потоков.

Склоновые потоки могут представлять большую опасность для людей и различных объектов. Для организации защиты необходимы сведения о динамических параметрах потоков и границах их действия. Одним из источников таких сведений является математическое моделирование.

Существует множество различных видов склоновых потоков:

- снежные лавины (Рис. 1, а)),
- сели,
- оползни,
- паводки,
- водоснежные потоки,
- лахары (грязевые вулканические потоки) (Рис. 1, б)),
- лавовые потоки,
- и др.



а) «Домашняя лавина» вблизи Гляциологической станции МГУ в Азау, 1989 [1]



б) Грязевой вулканический поток (лахар) с вулкана Сент-Хеленс (США), 1982 [2]

Рисунок 1 — Примеры склоновых потоков.

Склоновые потоки классифицируются по трём типам:

- 1 тип — плотные потоки без перемешивания с воздухом (лавины с высокой плотностью, сели, оползни);
- 2 тип — плотность сравнима с плотностью воздуха, перемешивание с воздухом существенно (снежно-пылевые лавины, пирокластические потоки);
- 3 тип — смешанные: плотное ядро и пылевой слой над ним. Перемешивание с воздухом и между слоями существенно.

Для разных типов потоков строились различные модели. Предложенная в настоящей работе модель разрабатывалась для первого типа потоков, но может быть применена и к двум другим типам с некоторыми допущениями.

Сложность моделирования склоновых потоков заключается в следующих их свойствах: трёхмерная структура, неньютоновская реология, наличие свободной поверхности с возможными самопересечениями, турбулентный режим, сложная геометрия склона, а также разрушение потоком материала склона и вовлечение разрушенного материала в движение.

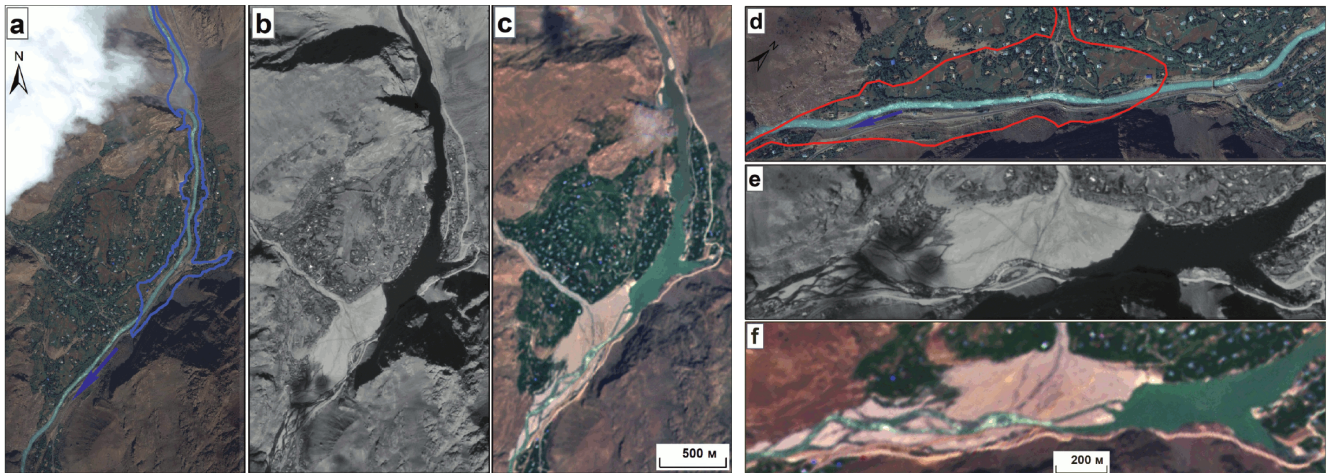
Исследование склоновых потоков — крайне актуальная задача, так как безопасность жизнедеятельности людей в горах напрямую зависит от организации защиты и оценки опасности, которые основываются, в том числе, на результатах численного моделирования.

Селевая катастрофа произошедшая на р. Барсемдара в долине р. Гунт (кишлак Барсем, Горно-Бадахшанская автономная область, Таджикистан) 16–24 июля 2015 г. [3] демонстрирует огромный ущерб приносимый склоновыми потоками. На рис. 2 слева (а) показан вид населённого пункта со спутника до селя и синим цветом очерчена территория, которая будет затоплена, образовавшимся озером. Следующие два снимка (b, c) показывают вид с воздуха на населенный пункт после селевой катастрофы, на них можно видеть образовавшееся после селя озеро. На рисунке справа (d) красным контуром обозначен участок населенного пункта, который будет погребён под селевыми отложениями. На снимках (e, f) можно видеть что на месте жилых строений образовался конус селевых отложений.

Пример данного катастрофического потока говорит о важности организовывать защиту и изучать возможные последствия прорывов ледниковых озёр.

Другим примером актуальности моделирования склоновых потоков является снежная лавина в Шамони (Франция). Здесь в опасной зоне находятся





а) Подпрудное озеро на р. Гунт

б) Конус выноса селя р. Барсемдара

Рисунок 2 — Участок Барсемской селевой катастрофы на космических снимках [3]: а, d — снимки со спутника WorldView-2 от 20.09.2012; б, е — снимки со спутника Канопус-В No 1 от 06.10.2015; с, f — снимки со спутника Sentinel-2A от 07.08.2018.



Рисунок 3 — Противолавинные защитные сооружения в Шамони.

жилые строения и необходимо организовывать защиту. На рис. 3 показаны защитные сооружения двух типов, первые из которых представляют собой бетонные стены — лавинорезы, предназначенные для торможения и отвода в сторону снежного потока. Построенные лавинорезы не выдержали нагрузку снежной массы и были разрушены. Взамен были возведены противолавинные

надолбы (забетонированные холмы), выдерживающие оказываемую на них нагрузку. Данный пример демонстрирует важность расчёта характеристик потока и взаимодействия потока с разрабатываемой защитной конструкцией.

Для организации защиты необходима информация о следующих параметрах склоновых потоков:

- скорости,
- плотности,
- давление в потоке,
- напряжения в потоке и в материале склона,
- размеры по высоте и ширине,
- границы распространения (до остановки),
- время до остановки.

Данная информация позволяет решать следующие задачи организации защиты:

- построение границы опасной зоны;
- скорости, плотности и высоты фронта потока в интересующем месте склона;
- силы, которые будут действовать на объекты в случае удара потока;
- составление карт опасных зон.

Все вышеперечисленные параметры течения могут быть получены при помощи математического моделирования потоков.

Настоящая работа посвящена разработке трёхмерных математических моделей катастрофических потоков на склонах гор.

### **Степень разработанности темы.**

В настоящее время для моделирования склоновых потоков используются в основном модели гидравлического типа (осредненные по глубине). Их достоинством является относительная вычислительная простота, однако они обладают рядом недостатков: не дают распределение параметров потока по глубине, содержат эмпирические коэффициенты, для определения которых требуется калибровка на местности. В связи с развитием вычислительной и измерительной техники появляются возможности построения и использования трёхмерных моделей, основанных на уравнениях механики сплошных сред без использования осреднения по глубине потока.

**Целью** данной работы является разработка трёхмерных математических и численных моделей плотных геофизических потоков на склонах гор, учитывающих сложные реологические свойства материала потока, турбулентный

режим движения, сложный реальный рельеф склона, а также возможное вовлечение потоком в движение донного материала.

Для достижения поставленной цели были решены следующие **задачи**:

1. Сформулирована трехмерная математическая модель склонового потока, включающая выбор реологического соотношения движущегося материала, модели турбулентности при осредненном по Рейнольдсу подходе, а также формулировку граничного условия на нижней границе потока, обусловленного вовлечением массы на этой границе.
2. Сформулирован обобщенный на трехмерный случай критерий разрушения донного материала и вовлечения его в движение.
3. Переформулирована математическая постановка задачи в виде, удобном для создания численной модели.
4. Разработан обработчик и конвертер цифровой модели рельефа (библиотека `avalancheUtils`) для получения расчётной области для проведения компьютерного моделирования потоков на склонах реального рельефа.
5. Разработана процедура реализации модели в пакете `OpenFoam`. Разработаны несколько необходимых дополнительных модулей, отсутствовавших в этом пакете. Разработан решатель `avalancheFoam`, позволяющий моделировать трёхмерные турбулентные потоки неньютоновской среды с учётом разрушения и уноса материала дна и сложного рельефа местности.
6. Произведена валидация модели и разработанного решателя на модельной задаче потока на склоне постоянного уклона.
7. Проведён расчет потоков, для которых имеются данные, полученные в экспериментах, и сравнение результатов расчетов с экспериментальными данными.

**Научная новизна:** В диссертационной работе впервые получены следующие результаты

1. Впервые разработана трёхмерная математическая и численная модель плотных геофизических потоков на склонах гор, учитывающая сложные реологические свойства материала потока, турбулентный режим движения, сложный реальный рельеф склона, а также возможное вовлечение потоком в движение донного материала.
2. Впервые предложен обобщенный на трехмерный случай критерий разрушения донного материала и вовлечения его в движение.

3. Впервые разработана процедура расчета динамики склоновых потоков с использованием предложенной математической модели на базе пакета OpenFoam с добавлением необходимых дополнительных модулей, специально созданных при выполнении диссертационной работы и опубликованных для общего пользования в интернете.
4. На основе расчетов потоков в лотках, экспериментально исследованных учеными в университете Исландии, а также потоков на созданной с участием автора экспериментальной установке в НИИ механики МГУ имени М.В. Ломоносова проведено сравнение результатов, полученных при использовании  $k-\varepsilon$  и  $k-\omega SST$  моделей турбулентности при моделировании склоновых потоков.
5. Разработан алгоритм для оптимизации коэффициентов турбулентной модели по результатам эксперимента на базе методов машинного обучения.

**Практическая значимость** В работе впервые построены трехмерные математические и численные модели склоновых потоков, учитывающие сложную реологию материала потока, турбулентность, реальный рельеф склона и вовлечение в движение подстилающего донного материала. Практическая значимость таких моделей велика, так как с их использованием можно вычислить важные параметры: границы части склона, по которой будет двигаться поток, величины скоростей и других динамических параметров потока, в частности, их распределение вдоль нормали к склону, изменение массы при захвате подстилающего материала, форму отложений при остановке. Все эти данные важны для организации защиты от катастрофических склоновых потоков.

**Методология и методы исследования.** При построении математических моделей использовались законы и уравнения механики сплошных сред.

При выполнении диссертационной работы применялись следующие методы:

1. Численное (компьютерное) моделирование с использованием пакета OpenFoam с включением созданных дополнительных модулей, необходимых для расчетов склоновых потоков;
2. Создание экспериментальной установки и проведение экспериментов с потоками в лотках для проверки математической модели и результатов расчетов, а также для оптимизации коэффициентов моделей турбулентности ;

3. Методы оценки качества численного моделирования;
4. Методы обработки результатов моделирования для изучения особенностей структур течения;
5. Методы оптимизации параметров моделей;
6. Язык программирования C++ для программной реализации модели разрушения и уноса подстилающего материала;
7. Язык программирования Python для программной реализации турбулентной модели на основе нейронной сети.

### **Основные положения, выносимые на защиту:**

1. Разработанная трёхмерная математическая и численная модель плотных геофизических потоков на склонах гор и ее реализация в дополненном пакете OpenFoam дают возможность рассчитать поток на склоне с реальным рельефом с учетом неньютоновских свойств материала потока, турбулентного режима движения и вовлечения потоком донного материала.
2. Сформулированный критерий вовлечения массы на границе поток-подстилающий материал, основанный на сравнении величин интенсивности сдвиговых напряжений на этой границе и предела прочности подстилающего материала, обобщает предложенное ранее условие на величину касательного напряжения.
3. При движении однородного потока по длинному склону постоянного уклона скорость захвата склонового материала потоком стремится к константе, зависящей от величины предельного значения напряжений сдвига подстилающего материала, реологических параметров материала потока и угла склона.
4. Трёхмерное моделирование снежной лавины как потока неньютоновской среды в 22-ом лавинном очаге горы Юкспор (Хибины), а также потока, вызванного прорывом ледникового озера Малый Азау (Центральный Кавказ), демонстрирует возможности разработанной модели.

**Достоверность** результатов диссертации обусловлена использованием классических математических методов механики сплошных сред и общепризнанных вычислительных методов. При проведении численных расчётов проверена сеточная сходимость, и, где это возможно, результаты сравнены с аналитическими и экспериментальными данными.



**Апробация работы.** Результаты диссертации докладывались автором и обсуждались на семинаре НИИ механики МГУ по механике сплошных сред под руководством академика РАН А.Г. Куликовского, профессора В.П. Карликова, член-корр. РАН О.Э. Мельника, профессора А.Н. Осипцова; на семинаре кафедры газовой и волновой динамики механико-математического факультета МГУ под руководством академика Р.Н. Нигматулина, профессора Н.Н. Смирнова и профессора А.В. Звягина; на семинаре кафедры аэромеханики и газовой динамики механико-математического факультета МГУ под руководством профессора К.В. Краснобаева, профессора В.В. Измоденова, профессора В.Д. Котелкина, профессора В.Я. Шкадова; на семинаре отдела гидрологии речных бассейнов Института водных проблем РАН под руководством профессора В.В. Беликова.

По теме диссертации при окончании аспирантуры успешно защищена научно-квалификационная работа. Кроме того, вопросам моделирования склоновых потоков были посвящены курсовые и дипломная работы.

Основные результаты работы докладывались на следующих конференциях:

1. Открытая конференция ИСП РАН имени В.П. Иванникова, Москва, Россия, 1-2 декабря 2022
2. Международная научно-практическая конференция «Лавины и смежные вопросы», Москва, Россия, 27-28 октября 2022
3. ISPRAS Open conference 2021, Москва, Россия, 2-3 декабря 2021
4. Volga neuroscience meeting 2021, Нижний Новгород, Россия, 24-27 августа 2021
5. Международная конференция «Математическое моделирование», Международный авиационно-космический салон МАКС-2021, Россия, 21-22 июля 2021
6. European Geosciences Union General Assembly 2021, Вена, Австрия, 19-30 апреля 2021
7. Конференция «Проблемы механики: теория, эксперимент и новые технологии», Кемеровская обл., пос. Шерегеш; Россия, г. Новосибирск., Россия, 25 февраля - 5 марта 2021
8. ISPRAS OPEN conference 2020, Москва, Россия, 10-11 декабря 2020
9. XVII гляциологический симпозиум «Роль криосферы в прошлом, настоящем и будущем Земли», Санкт-Петербург, Россия, 17-20 ноября 2020



10. Международная конференция «Марчуковские научные чтения 2020» (МНЧ-2020), Академгородок, Новосибирск, Россия, 19-23 октября 2020
11. Международная конференция «Иванниковские чтения», Орёл, Россия, 25-26 сентября 2020
12. XXIV Международная конференция «Нелинейные задачи теории гидродинамической устойчивости и турбулентность», 2020
13. Всероссийская конференция молодых учёных-механиков, YSM-2020, г. Сочи, Россия, 3-13 сентября 2020
14. Ivannikov ISP RAS Open Conference, Moscow, 5-6 December, 2019, Москва, Россия, 5-6 декабря 2019

**Личный вклад.** Основные научные результаты диссертации получены автором лично, являются новыми и соответствуют мировому уровню.

В работе [4] авторский вклад соискателя составляет  $1/3$ , в работе [5] авторский вклад соискателя составляет  $1/4$ , в работе [6] авторский вклад соискателя составляет  $1/2$ .

**Публикации.** Основные результаты по теме диссертации изложены в 7 научных статьях, 1 из которых изданы в журналах, рекомендованных ВАК, 2 — в периодических научных журналах, индексируемых RSCI, 2 — в периодических научных журналах, индексируемых Web of Science и Scopus.

**Объем и структура работы.** Диссертация состоит из введения, 5 глав, заключения и 3 приложений.

Полный объём диссертации составляет 165 страниц, включая 46 рисунков и 10 таблиц. Список литературы содержит 145 наименований.

## Глава 1. Обзор литературы<sup>1</sup>

До середины 20 века сведения о катастрофических склоновых потоках носили главным образом описательный характер. Первое научное изучение склоновых потоков началось с исследования снежных лавин в 1935 г., когда была создана первая снежно-метеорологической служба, которая занималась изучением и предупреждением снежных обвалов [11]. Данная служба была создана после лавинной катастрофы в городе Кировск в Хибинах, в которой под снежными завалами оказалось 239 человека из которых 89 погибли.

Первыми появляются простейшие модели [12–14], в которых поток представляется как материальная точка, движущаяся вниз по склону. Следующими появляются и развиваются по настоящий день модели с использованием осредненных по глубине уравнений [15–25]. Трёхмерные модели [26–29] начали появляться в последнее время в силу развития вычислительной техники.

Обзор различных моделей для потоков в руслах представлен в книге Петрова и Потапова [30].

### 1.1 Обзор работ с использованием осредненного по глубине подхода

Следующими появляются модели гидравлического типа (теория «мелкой воды»), применяемые когда глубина потока  $h$  много меньше его длины  $L$ . Тогда поток описывается уравнениями сплошной среды, осреднёнными по толщине потока или по поперечному сечению, как в теории мелкой воды или в гидравлике рек. В этом подходе вычисляются средние по сечению скорости вдоль склона  $U$  и толщина потока  $h$ . Модели гидравлического типа распространены и в настоящий момент.

---

<sup>1</sup>При подготовке данной главы диссертации использовались следующие публикации автора, в которых, согласно «Положению о присуждении ученых степеней в Московском государственном университете имени М.В.Ломоносова», отражены основные результаты, положения и выводы исследования: [4–10].

Пример простейшей модели гидравлического типа для одномерного потока приведен в (1.1). Более подробное описание уравнений гидравлического типа можно найти в книге [31].

$$\begin{aligned} \frac{\partial h}{\partial t} + \frac{\partial hU}{\partial x} &= 0, \\ \frac{dU}{dt} &= -\frac{1}{h} \frac{\partial P}{\partial x} + g \sin \theta - \frac{\tau}{\rho h}, \\ P &= gh^2 \cos \theta / 2, \quad \tau = \mu_h \rho gh \cos \theta + k\rho U^2 \end{aligned} \quad (1.1)$$

Здесь  $x$  — координата вдоль склона,  $\theta$  — угол наклона склона к горизонту,  $\mu_h$  и  $k$  — коэффициенты «сухого» и «гидравлического» трения, требующие калибровки для конкретного склона.

Можно выделить следующие этапы развития гидравлических моделей склоновых потоков:

- Одномерная модель для движения лавины по широкому склону с вовлечением снега на переднем фронте [15],
- Двумерная модель плотного потока [16; 17] с учетом бокового растекания и поперечного взаимодействия
- Движение по желобам (лоткам) с различными формами поперечного сечения с учётом влияния трения на боковых стенках [18],
- Новый закон сухого трения (переформулировка закона Кулона) [19],
- Захват массы на нижней поверхности [20; 21],
- Учёт сложного рельефа местности и захвата массы на дне [22–24],
- Модель, позволяющая рассчитать движение на скачке дна с учётом размыва представлена в работе [25].

В силу вычислительной простоты гидравлических моделей, они используются для построения карты опасных зон, так как для данной задачи необходимо произвести большое количество расчетов разных очагов. Однако, данный подход обладает большим количеством эмпирических коэффициентов, которые не связаны непосредственно с физическими свойствами движущейся среды и склона, и могут быть оценены только с помощью обратных расчетов потоков, произошедших в данном регионе, для которых имеются данные измерений.

Уравнения мелкой воды реализованы во многих коммерческих программных кодах, используемых для моделирования склоновых потоков, как, например, пакет DAN [32], пакет SamosAT [33; 34], пакет FLATModel [35], пакет RAMMS [36].

Есть также открытые альтернативы данным программным продуктам, как например пакет TITAN2D [37; 38], пакет g.avaflow [39; 40], пакет GERRIS [41], решатель faSavageHutterFoam пакета OpenFOAM [22; 23].

## 1.2 Обзор работ с использованием трёхмерного подхода

Трёхмерное моделирование динамики потоков на склонах гор не получило широкого распространения из-за очевидной вычислительной сложности задачи, так как в длину поток может достигать нескольких километров, а в толщину порядка нескольких десятков метров. В связи с развитием вычислительной техники в последние годы появилась возможность проводить полное трёхмерное моделирование склоновых потоков.

Трёхмерные модели позволяют:

- связать уравнения с физическими свойствами движущейся среды,
- в явной форме учесть реологические свойства материала потока,
- описать зависимость динамических параметров от расстояния от дна.

Для расчета сил, действующих на конструкции при ударе потока, необходимо знать распределение скорости и давления, в том числе, по глубине потока. Модель мелкой воды не может дать таких распределений, так как в ней рассматриваются лишь осреднённые по глубине параметры потока.

Второй важной задачей, для решения которой нужен трёхмерный подход — вывод физически обоснованных формул для членов уравнений, описывающих трение на дне, а также скорость вовлечения потоком подстилающего материала и отложение материала потока на склон. Эмпирические формулы, используемые в настоящее время для склоновых потоков в моделях гидравлического типа, содержат коэффициенты, не связанные непосредственно со свойствами движущегося материала и характеристиками склона. Они определяются для каждого региона только путем обратных расчетов потоков, для которых имеются данные измерения их параметров.

Однако, склоновые потоки обладают сложными физическими свойствами, которые затрудняют трёхмерное моделирование:

- Материал многих потоков обладает сложными реологическими свойствами. Об этом свидетельствует, в частности, тот факт, что лавины и оползни могут останавливаться на наклонных поверхностях;
- Крупномасштабные потоки, как правило, являются турбулентными, для них возникают проблемы осреднения и выбора турбулентной модели;
- Потоки имеют сложную неоднородную трёхмерную структуру;
- Необходимо учитывать сложный рельеф местности;
- Масса потока может увеличиться в несколько раз во время движения. Модель должна учитывать захват потоком материала, лежащего на склоне.

В последнее время начали появляться некоторые трёхмерные модели [26–29], которые, однако, не учитывают всех свойств, характерных для склоновых потоков, а лишь некоторые из них.

Например, расчёт снежной лавины на реальном склоне с помощью метода конечных элементов на основе подхода Петрова-Галёркина представлен в работе Ямагучи и соавторов [26]. В работе склоновый поток представлен бингамовской жидкостью, но движение считается ламинарным и вовлечение снежного покрова, лежащего на склоне, не учитывается.

В работе Оды и соавторов [42] также используется двухфазный подход для описания динамики снежной лавины, только в отличие от работы [26] система уравнений решается конечно-разностным методом.

Модель склонового потока с захватом на основе метода гидродинамики сглаженных частиц, представленная в работе [27], обладает, как и гидравлические модели, большим количеством эмпирических коэффициентов в силу используемого вычислительного метода.

Расчёт снежной лавины как турбулентного потока с учетом реального рельефа местности но без захвата донного материала представлен в первой работе автора [7].

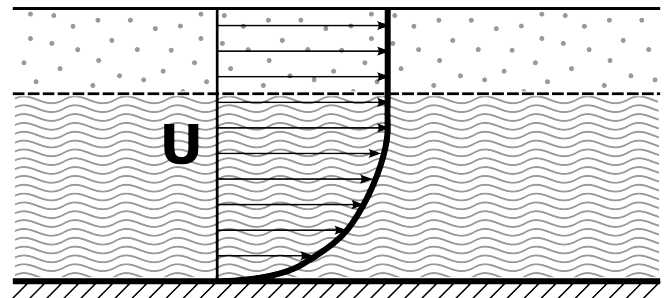
### 1.3 Обзор используемых для склоновых потоков реологических моделей

Для разных типов потоков используются различные реологические модели. Например, для водных потоков (торрентов) — ньютоновская жидкость. Но для лавин, оползней, глинистых селей, лавовых потоков наблюдаются следующие неньютоновские эффекты:

- возможность остановки на наклонном склоне (продемонстрирована на рис. 1.1 а)),
- формирование вблизи верхней поверхности профиля скорости, соответствующего течению без деформирования (кваситвёрдый слой) (показан на рис. 1.1 б)).



а) Остановка лавины на склоне [43]



б) Кваситвёрдый слой

Рисунок 1.1 — Неньютоновские эффекты потоков на склонах гор.

Исследование реологии материалов склоновых потоков проводилось рядом авторов с использованием вискозиметров и на основе измерения профилей скорости в потоках в экспериментальных лотках и в натуральных потоках на специально оборудованных склонах

- для глинистых селей [44],
- для снежных лавин [45; 46].

Для описания склоновых потоков используются самые разные реологические модели:

- Ньютоновская модель среды [47]: линейно вязкая среда,
- Модель Кросса [45]: вязкость, уменьшается при увеличении скорости деформаций;

- Модель Багнольда [48]: квадратичная зависимость касательных напряжений от компонент тензора скоростей деформаций;
- Степенная жидкость [49]: степенная зависимость вязких напряжений от скорости деформации среды;
- Бингамовская жидкость [50]: наличие предела текучести при линейной зависимости вязких напряжений от скорости деформации среды;
- Модель Хершеля-Балкли [28; 44; 45; 50–52]: степенная зависимость вязких напряжений от скорости деформации среды и наличие предела текучести.

Первые три модели могут аппроксимировать профиль скорости, но не описывают остановку потока на наклонной поверхности, в отличие от последней модели Хершеля-Балкли. Данная модель охватывает наибольшее количество течений. Также реологическая модель Хершеля-Балкли позволяет описать вышперечисленные неньютоновские эффекты потоков на склонах гор.

Рассмотрим простое сдвиговое течение, когда поток движется вдоль оси  $x$  и скорость изменяется лишь по глубине (вдоль оси  $z$ ):

$$\begin{aligned} U_x = U_x(z), U_y = U_z = 0, \\ \frac{dU_x}{dz} > 0. \end{aligned} \quad (1.2)$$

Закон Хершеля-Балкли для простого сдвигового течения (1.2) выглядит следующим образом:

$$\begin{aligned} \text{если } \tau \geq \tau_0, \quad \tau = \tau_0 + \kappa s^n, \\ \text{иначе } s = 0. \end{aligned} \quad (1.3)$$

Здесь  $\tau$  — напряжения сдвига,  $\tau_0$  — предел текучести,  $s = \frac{dU_x}{dz}$  — скорость сдвига,  $\kappa$  — коэффициент консистенции,  $n$  — степенной индекс.

Примеры задания коэффициентов реологической модели Хершеля Балкли для различных склоновых потоков:

- Плотная сухая лавина [47]:  
 $\rho = 200 \text{ кг/м}^3, \tau_0 = 0, n = 1, \kappa/\rho = \nu = 10^{-3} \text{ м}^2/\text{с}.$
- Лавина из плотного снега [45]:  
 $\rho = 500 \text{ кг/м}^3, \tau_0/\rho = 2 \text{ м}^2/\text{с}^2, n = 2, \kappa/\rho = 0.89 \cdot 10^{-4} \text{ м}^2.$
- Глинистый поток (концентрация глины  $> 10\%$ ) [44]:  
 $\tau_0/\rho = 1.5 \text{ м}^2/\text{с}^2, n = 0.33, \kappa/\rho = 0.1 \text{ м}^2/\text{с}^{2-n}.$

– Кремниевая лава [50]:

$$\rho = 2600 \text{ кг/м}^3, \tau_0 = 10^5 \text{ Па}, n = 1, \kappa = 10^9 \text{ Па} \cdot \text{с}.$$

– Подводный песчаный оползень [53]:  $\rho = 1985 \text{ кг/м}^3, \tau_0 = 200 \text{ Па}, n = 1, \kappa = 0.001 \text{ Па} \cdot \text{с}.$

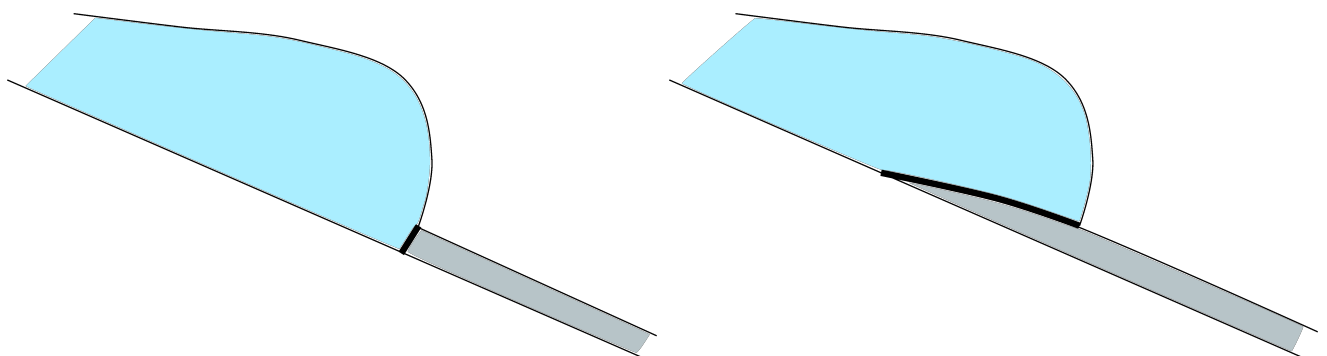
Реологические коэффициенты в данных работах были получены методом аппроксимации профиля скорости, полученного либо в натурном эксперименте по спуску лавинного потока, как, например, в работе [45], или исходя из данных, полученных в ротационном вязкозиметре [52; 54].

Исследование реологических свойств снега проводилось в работах [46; 55; 56], в том числе получены сведения о пределе прочности снега.

#### 1.4 Обзор существующих моделей захвата подстилающего материала потоком в горах

Следующим важным свойством потоков на склонах гор является разрушение и захват материала склона. Выделяют два основных типа захвата:

- Фронтальный захват — предполагается, что захват происходит в узкой зоне, примыкающей к фронту лавины (рис. 1.2 а)).
- Базальный захват — предполагается, что захват происходит по поверхности соприкосновения тела потока и склона (рис. 1.2 б)).



а) Фронтальный захват

б) Базальный захват

Рисунок 1.2 — Различные типы захвата донного материала.

При моделировании с использованием моделей гидравлического типа предлагались различные модели захвата в зависимости от типа потока:

- для плотной снежной лавины [57];
- для снежно-пылевой лавины [58];



– для смешанного типа лавин [21].

Одна из первых моделей движения снежных лавин с захватом была представлена в [59; 60], см. также [20; 21]. В этой модели предполагалось, что вовлечение снежного покрова происходит в узкой зоне, примыкающей к фронту лавины. Это предположение было основано на данных наблюдений, подтвержденных в настоящее время измерениями [61]. Поскольку длина этой зоны значительно меньше длины лавины, ее можно моделировать как «фронт разрушения», где снежный покров претерпевает переход из твердого состояния в жидкое под действием лавины. Этот тип захвата теперь называется фронтальным захватом. Важным допущением является то, что разрушение снежного покрова и вовлечение его в поток происходит при воздействии на него напряжения, равного его прочности на сжатие. Это условие совместно с уравнениями сохранения массы и импульса поперек фронта и осредненными по глубине дифференциальными уравнениями импульса и неразрывности позволило рассчитать динамику потока, включая скорость и высоту фронта.

Фронтальный захват материала используется в гидравлических моделях в виде граничного условия на переднем фронте [15; 17; 21; 59; 60; 62–67].

Для моделирования непрерывного захвата со дна в гидравлических моделях предложены эмпирические формулы захвата со дна (по аналогии с потоками различной физической природы типа размыва речного дна [20; 25; 30; 68]).

Постепенный захват материала из подстилающего слоя может рассматриваться в дополнение или вместо фронтального захвата. В уравнениях в этом случае появляется дополнительный источниковый член. Различные формулы для определения источникового члена были предложены и использованы в работах [15; 58–60; 62; 69–71]. Полевые наблюдения за снежными лавинами показывают, что обычно значительная часть захвата происходит в относительно узкой передней части лавины (в пределах первых десятков метров) [72]. Эту переднюю часть можно либо заменить скачком в математическом моделировании, как это сделано в модели, предложенной в работах [15; 59; 60; 62; 69], либо рассматривать как узкую зону с постепенным захватом. В частности, в [73] предполагается, что границей между движущимся лавинным снегом и подстилающим слоем неподвижного снега является наклонная ко дну ударная волна, в которой снежный покров разрывается и увлекается лавиной. При некоторых дополнительных предположениях такой подход приводит к выражению скорости

захвата через произведение модельного параметра на корень из разницы между толщиной потока и значением толщины потока, при котором давление на дне потока достигает прочности на сдвиг подстилающего снега. Некоторые критические комментарии к этому подходу можно найти в [20; 71]. Постепенный захват из подстилающего слоя может также продолжаться по телу потока за пределами лобовой части, если весь имеющийся донный материал, подверженный эрозии, не захватится лобовой частью. Это называется базальным захватом.

Механизм базального захвата рассмотрен для сухих плотных, пылевых и смешанных лавин в [21; 58; 74–78]. Предполагается, что он подобен перемешиванию в турбулентных струях. Перемешивание во многом обусловлено неустойчивостью на границах потока, приводящей к образованию и обрушению там волн. Известно, что скорость роста возмущений пропорциональна разности скоростей потока и окружающей среды с коэффициентом, зависящим от отношения плотностей [79]. Разумно предположить, что скорость захвата удовлетворяет тому же правилу. Тогда для сухих плотных лавин скорость захвата из подстилающего слоя неподвижного снега пропорциональна скорости движения снега в лавине и отношению плотностей в подстилающем слое снега и в лавине [74; 75].

Исслер и Пастор Перес [80] предложили модель, основанную на допущении, что захват со дна происходит, когда значение напряжения сдвига на дне равно прочности материала дна на сдвиг. Напряжение сдвига на дне и скорость захвата в их модели следует рассчитывать путем решения полных (не осредненных по глубине) уравнений течения. Исслер и Пастор Перес [80] представили численные решения для двумерного ламинарного нестационарного движения линейно-вязких и багнольдовых жидкостей вниз по бесконечно длинному склону с постоянным наклоном.

Базальный тип захвата используется в гидравлической модели [22–24]. Здесь скорость захвата материала пропорциональна скорости потока и напряжению на дне, и обратно пропорциональна удельной энергии эрозии. Такой механизм захвата также рассматривается в работе [81].

В работе [82] подчёркивается важность использования в модели лишь такого типа параметров, которые обусловлены физическими параметрами рассматриваемого явления, такими как свойства движущейся среды или подстилающего материала.

В [83] модель Григоряна–Остроумова анализируется и сравнивается с двумя другими моделями захвата [70; 84; 85], которые также рассматривают границу между потоком и снежным покровом как ударную волну и определяют скорость увлечения из условий скачка параметров потока поперек ударной волны. Это сравнение поучительно, поскольку оно выявляет недостатки всех трех моделей и указывает на наличие улучшенных версий. Однако, проблема эрозии, вероятно, должна решаться в сочетании с реологией течения.

В работе [83] описаны пути улучшения модели эрозии из [71] и двух концептуально связанных беспараметрических моделей захвата. В работе [86] убедительно показано, что реалистичные обратные расчеты нескольких лавинных событий с одним и тем же набором параметров требуют моделирования как захвата, так и осаждения снежной массы.

В статье [86] основное внимание уделяется расчету двух измеренных лавин в Валле-де-ла-Сьон, которые сошли в один и тот же день при очень похожих снежных условиях, но вели себя совершенно по-разному. Ключевым результатом является то, что обе лавины могут быть достаточно хорошо смоделированы с помощью одного набора параметров, если в модель включены как захват, так и отложение. Эти результаты подтверждают и расширяют более раннее открытие [87] о том, что различные лавины могут быть рассчитаны задним числом с меньшим разбросом эмпирических коэффициентов трения, если включен захват. Следует отметить, что формула отложения для модели течения с осреднением по глубине, используемая в [86], была разработана на основе численного моделирования с разрешением по глубине и гранулярной реологии.

Основные проблемы существующих формул, определяющих скорость захвата донного материала:

- В реальных склоновых потоках чаще всего происходит смешанный тип захвата — и фронтальный и базальный одновременно.
- Большинство предложенных формул подразумевают гидравлические модели потока, являются чисто эмпирическими, связывают вовлечение донного материала со средней по глубине скоростью и глубиной потока, а не с динамическими параметрами придонного слоя потока.

В настоящей работе разрабатывается механизм захвата материала, основанный на предположении о пределе прочности донного материала [51] и не содержащий свободных констант. Данный механизм разрабатывается в рамках

трёхмерной модели, позволяющей проводить моделирование с учётом реального рельефа местности, турбулентного режима течения и сложной реологии потока.

## Глава 2. Математическая модель<sup>1</sup>

### 2.1 Уравнения движения и уравнение неразрывности

Склоновый поток рассматривается как турбулентный несжимаемый поток неньютоновской реологии со свободной поверхностью на склоне под действием силы тяжести (см. рис. 2.1).

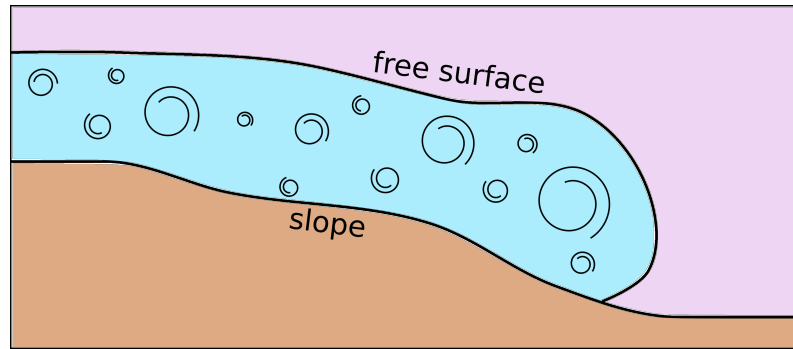


Рисунок 2.1 — Схема склонового потока.

Поток описывается осредненными по Рейнольдсу уравнениями количества движения (2.1).

$$\frac{\partial(\rho\bar{U})}{\partial t} + \nabla \cdot (\rho\bar{U}\bar{U}) = -\nabla\bar{p} + \nabla \cdot \bar{\tau} + \rho\mathbf{g}. \quad (2.1)$$

Здесь  $\rho$  — плотность,  $\bar{U}$  — осреднённая по Рейнольдсу скорость,  $\bar{p}$  — осреднённое по Рейнольдсу давление,  $\mathbf{g}$  — ускорение свободного падения,  $\bar{\tau}$  — тензор напряжений, задаётся по формуле (2.2).

$$\bar{\tau} = \bar{\tau}_m + \bar{\tau}_t, \quad (2.2)$$

где  $\bar{\tau}_m$  — осреднённый по Рейнольдсу тензор вязких молекулярных напряжений,  $\bar{\tau}_t$  — тензор турбулентных напряжений.

Поток считается несжимаемым. Уравнение неразрывности имеет вид (2.3).

$$\nabla \cdot \bar{U} = 0. \quad (2.3)$$

<sup>1</sup>При подготовке данной главы диссертации использовались следующие публикации автора, в которых, согласно «Положению о присуждении ученых степеней в Московском государственном университете имени М.В.Ломоносова», отражены основные результаты, положения и выводы исследования: [4–10].

Система из уравнений (2.1) и (2.3) не замкнута, необходимо задать так называемые реологические соотношения, определяющие связи  $\bar{\tau}_m$  с характеристиками деформирования, а также модель турбулентности.

## 2.2 Реологическая модель и её регуляризация

Для описания динамики склоновых потоков используются различные реологические модели, часть из которых перечислена в разделе 1.3. В разрабатываемом в настоящей работе программном комплексе есть возможность использовать некоторые из перечисленных в 1.3 реологических моделей:

- ньютоновская модель,
- модель Кросса,
- модель Хершеля Балкли.

### 2.2.1 Реологическая модель Хершеля-Балкли

Для описания реологических свойств движущейся жидкости в настоящей работе используется реологическая модель Хершеля-Балкли [28; 44; 45; 50–52], когда среда обладает пределом текучести, задаваемым параметром  $\tau_0$  и степенной зависимостью тензора напряжений от тензора скоростей деформаций.

Для простого сдвигового потока (1.2) можно нарисовать график зависимости напряжения сдвига от скорости сдвига (рис. 2.2). Стандартно выделяют следующие типы жидкостей: ньютоновская, дилатантная, псевдопластическая, Бингамовская и жидкость Хершеля-Балкли. Реологические соотношения Хершеля-Балкли (1.3) позволяют обобщить все вышеперечисленные типы жидкостей. Так при нулевом пределе текучести и единичном показателе степени получается ньютоновская жидкость. При показателе степени больше единицы описывается дилатантная жидкость, а в случае показателя степени меньше единицы - псевдопластическая. При отличном от нуля пределе текучести и единичном показателе степени модель описывает Бингамовскую жидкость.

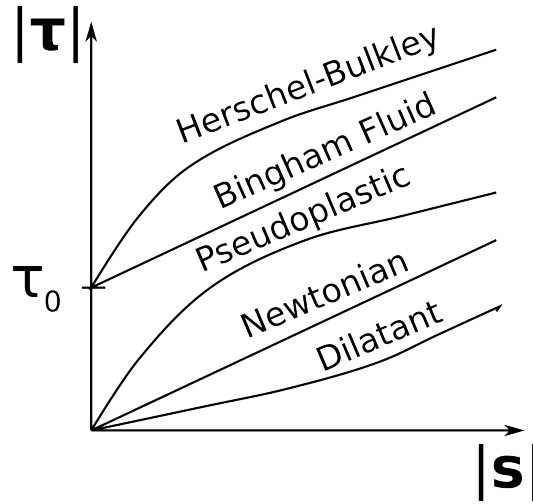


Рисунок 2.2 — Зависимость напряжения сдвига от скорости сдвига в простом сдвиговом потоке для различных реологических моделей.

Для течений общего вида реологические соотношения Хершеля-Балкли выглядят следующим образом (2.4):

$$\begin{aligned} |\boldsymbol{\tau}| < \tau_0, \quad \text{то} \quad \mathbf{s} = 0, \\ |\boldsymbol{\tau}| \geq \tau_0, \quad \text{то} \quad \boldsymbol{\tau} = \frac{\tau_0 + \kappa |\mathbf{s}|^n}{|\mathbf{s}|} \mathbf{s}. \end{aligned} \quad (2.4)$$

Здесь  $|\boldsymbol{\tau}| = \sqrt{0.5\tau_{ij}\tau_{ij}}$  — корень из половины второго инварианта тензора напряжений,  $|\mathbf{s}| = \sqrt{2s_{ij}s_{ij}}$  — корень из удвоенного второго инварианта тензора скоростей деформаций.

Соотношения (2.4) можно переписать в другом виде (2.5), вводя коэффициент эффективной молекулярной вязкости  $\mu_m$ :

$$\boldsymbol{\tau} = 2\mu_m \mathbf{s}, \quad (2.5)$$

Заметим, что в этом разделе речь идет о молекулярных напряжениях  $\boldsymbol{\tau}_m$ , но индекс  $m$  для краткости не пишется.

В настоящей работе рассматривается турбулентный поток, для описания которого используются осреднённые по Рейнольдсу уравнения. Тензор молекулярных напряжений также осредняется, в результате чего появляется неньютоновское дополнительное напряжение.

Гаврилов и Рудяк в своей работе [88] предложили приближение для неньютоновского дополнительного напряжения, возникающего при осреднении

молекулярной вязкости, но при этом пренебрегали пульсациями молекулярной вязкости, предполагая их малыми. Аналогично поступают авторы в работе [89].

За неимением обоснованных гипотез в настоящей работе принимается условие, что реологическое соотношение зависит от осредненных параметров так же, как и от не осредненных.

### 2.2.2 Регуляризация модели Хершеля-Балкли

В терминах эффективной вязкости, когда тензор молекулярных напряжений  $\boldsymbol{\tau}$  выражается по формуле (2.5) как удвоенное произведение эффективной молекулярной вязкости  $\mu_m$  и тензора скоростей деформаций  $\mathbf{s}$ , реологическая модель Хершеля-Балкли записывается в виде (2.6).

$$\begin{aligned} \mu_m &= \infty, & \text{если } |\boldsymbol{\tau}| < \tau_0, \\ \mu_m &= \frac{\tau_0 + \kappa |\mathbf{s}|^n}{|\mathbf{s}|}, & \text{если } |\boldsymbol{\tau}| \geq \tau_0. \end{aligned} \quad (2.6)$$

В выражении (2.6) появляется бесконечная вязкость, с которой невозможно работать, тогда проводится регуляризация модели Хершеля-Балкли [90] при малых скоростях сдвига, когда  $|\mathbf{s}| < s_0$  коэффициент вязкости полагается равным некоторой большой постоянной  $\mu_0$ , как показано на рис. 2.3. Здесь  $s_0$  — параметр регуляризации, удовлетворяющий соотношению (2.7)

$$\mu_0 = \frac{\tau_0 + \kappa s_0^n}{s_0}. \quad (2.7)$$

В случае использования регуляризации коэффициент вязкости  $\mu_0$  действует в «квазитвёрдом» слое, в связи с чем  $\mu_0$  иногда называют коэффициентом «недеформируемой» вязкости.

Регуляризованная модель Хершеля-Балкли записывается следующим образом (2.8):

$$\begin{aligned} \text{если } |\mathbf{s}| < s_0 : \quad & \mu_m = \mu_0, \\ \text{если } |\mathbf{s}| \geq s_0 : \quad & \mu_m = \frac{\tau_0 + \kappa |\mathbf{s}|^n}{|\mathbf{s}|}. \end{aligned} \quad (2.8)$$



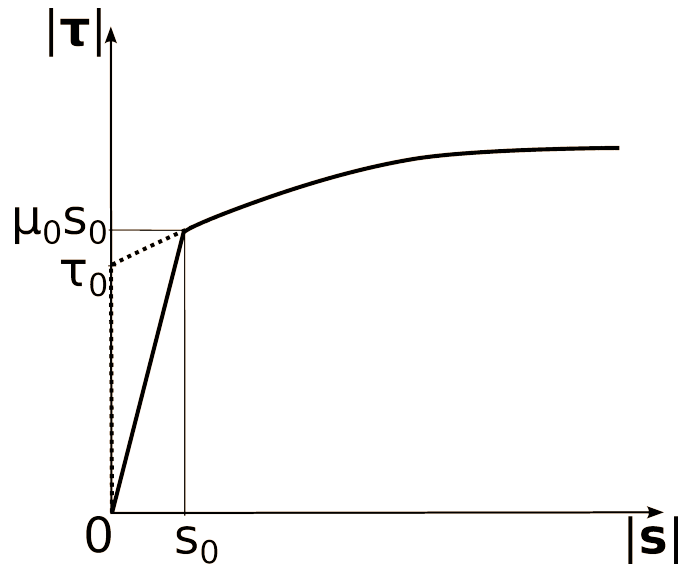


Рисунок 2.3 — Регуляризация реологической модели Хершеля-Балкли.

Коэффициент «недеформируемой» вязкости в регуляризованной модели Хершеля-Балкли берётся очень большим, чтобы получить наибольшее совпадение с исходной моделью.

В полученной модели присутствует ветвление, то есть то или иное действие используется в зависимости от некоторого условия. Это негативно сказывается на скорости расчёта, в силу чего коэффициент вязкости представляется в виде функции минимума между очень большим коэффициентом «недеформируемой» вязкости  $\mu_0$  и коэффициентом вязкости модели Хершеля-Балкли при  $|\mathbf{s}| \geq s_0$ , как показано в выражении (2.9). Предполагается, что  $\mu_0$  настолько большой, что ни в какой другой области, кроме той, где  $\mathbf{s} < s_0$  не достигается такой большой вязкости в течении. Схематично параметр  $\mu_0$  изображён на рис. 2.4.

$$\mu_m = \min\left(\mu_0, \frac{\tau_0 + \kappa |\mathbf{s}|^n}{|\mathbf{s}|}\right). \quad (2.9)$$

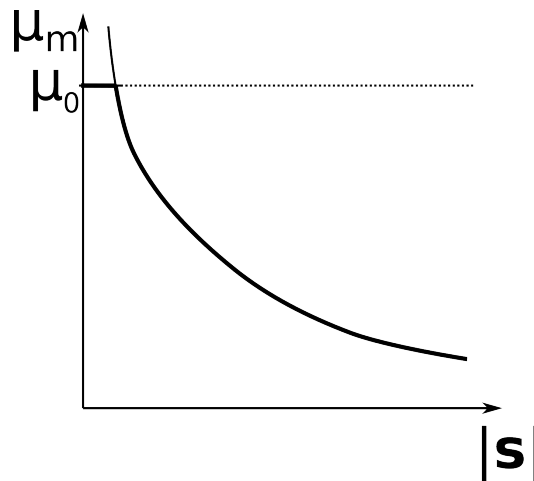


Рисунок 2.4 — Параметр «недеформируемой» вязкости  $\mu_0$ .

### 2.3 Модели турбулентности

При описании турбулентного неьютоновского потока с помощью RANS модели присутствует три основных вопроса:

1. осреднение тензора молекулярных напряжений,
2. применимость стандартных моделей турбулентности для таких потоков,
3. определение значений констант турбулентных моделей.

Поток считается турбулентным, вследствие чего используются осредненные по Рейнольдсу уравнения. Предполагается, что осреднённый по Рейнольдсу тензор молекулярных напряжений имеет тот же вид зависимости от осреднённого тензора скоростей деформаций, как и не осреднённый. Для расчёта турбулентного режима течения используются  $k$ - $\epsilon$  [91–93] и  $k$ - $\omega$  SST [91–94] модели турбулентности.

Первоначально эти модели предлагались для ньютоновских потоков без свободных поверхностей. Однако многие авторы успешно использовали их и для потоков со свободными поверхностями [95–101] а также для неьютоновских потоков [88; 102; 103] — с теми же константами. Отмечено, что условие перехода от ламинарного режима к турбулентному зависит от реологических свойств жидкости [102–104], но для развитого турбулентного течения стандартные константы дают приемлемые результаты. Так, например, при расчётах потоков Бингамовской жидкости и жидкости Хершеля-Балкли, проводившихся в работах [102; 103], были сделаны следующие выводы о влиянии реологической

модели на турбулентные параметры: 1. в ламинарном режиме движение зависит от реологии потока; 2. критическое число Рейнольдса  $Re_{\text{крит}}$  увеличивается тем больше, чем больше предел текучести, который для Бингамовской жидкости характеризуется числом Хедстрёма  $He = \frac{h^2\tau_0}{\rho\nu^2}$ ; 3. Для развитого турбулентного течения можно использовать стандартные коэффициенты турбулентных моделей.

В дальнейшем изложении для краткости знак осреднения по Рейнольдсу (черту над символом) будем опускать.

### 2.3.1 $k$ - $\varepsilon$ модель турбулентности

Турбулентные напряжения  $\boldsymbol{\tau}_t$  вычисляются в соответствии с гипотезой Буссинеска (2.10).

$$\boldsymbol{\tau}_t = 2\mu_t \mathbf{s} - \frac{2}{3}\rho k \mathbf{I}, \quad (2.10)$$

где  $\mu_t$  — коэффициент турбулентной вязкости,  $\mathbf{s}$  — осредненный тензор скоростей деформаций,  $\mathbf{I}$  — единичный тензор.

При использовании  $k$ - $\varepsilon$  турбулентной модели [91] турбулентная вязкость вычисляется по формуле (2.11).

$$\mu_t = \frac{\rho C_\mu k^2}{\varepsilon}, \quad (2.11)$$

где  $k$  — турбулентная кинетическая энергия;  $\varepsilon$  — диссипация турбулентной кинетической энергии,  $C_\mu$  — коэффициент турбулентной модели.

Для замыкания системы уравнений Рейнольдса в систему уравнений добавляются 2 уравнения (2.12) для  $k$  и  $\varepsilon$ .

$$\begin{aligned} \frac{\partial k}{\partial t} + \nabla \cdot (\mathbf{U}k) &= \nabla \cdot (\nu_m \nabla k) - \nabla \cdot \left( \frac{\nu_t}{\sigma_k} \nabla k + P_k \right) - \varepsilon, \\ \frac{\partial \varepsilon}{\partial t} + \nabla \cdot (\mathbf{U}\varepsilon) &= C_{\varepsilon 1} P_k \frac{\varepsilon}{k} - C_{\varepsilon 2} \frac{\varepsilon^2}{k} + \nabla \cdot \left( \frac{\nu_t}{\sigma_\varepsilon} \nabla \varepsilon \right). \end{aligned} \quad (2.12)$$

Здесь  $P_k = \nu_t \nabla \mathbf{U} \cdot [\nabla \mathbf{U} + (\nabla \mathbf{U})^T]$  — скорость диссипации турбулентной кинетической энергии средним течением,  $C_{\varepsilon 1}$ ,  $C_{\varepsilon 2}$ ,  $C_\mu$ ,  $\sigma_k$ ,  $\sigma_\varepsilon$  — коэффициенты модели турбулентности.

Значения коэффициентов стандартно задаются следующими (5.2):

$$C_\mu = 0.09; \quad C_1 = 1.44; \quad C_2 = 1.92; \quad C_3 = 0.0; \quad \sigma_k = 1.0; \quad \sigma_\varepsilon = 1.3. \quad (2.13)$$

### 2.3.2 $k$ - $\omega$ SST модель турбулентности

При использовании  $k$ - $\omega$  SST турбулентной модели [91] турбулентная вязкость вычисляется по формуле (2.14).

$$\mu_t = \frac{\rho a_1 k}{\max(a_1 \omega, b_1 |\mathbf{s}| F_2)}, \quad (2.14)$$

и в систему уравнений добавляются два уравнения (2.15)

$$\begin{aligned} \frac{\partial k}{\partial t} + \nabla \cdot (\mathbf{U}k) &= P_k - \beta^* k \omega + \nabla \cdot ((\mathbf{v}_m + \alpha_k \mathbf{v}_t) \nabla k), \\ \frac{\partial \omega}{\partial t} + \nabla \cdot (\mathbf{U}\omega) &= \gamma |\mathbf{s}|^2 - \beta \omega^2 + \nabla \cdot ((\mathbf{v}_m + \alpha_\omega \mathbf{v}_t) \nabla \omega) + \\ &+ 2(1 - F_1) \rho \alpha_\omega \frac{1}{\omega} \nabla k \cdot \nabla \omega. \end{aligned} \quad (2.15)$$

где  $F_1 = \tanh \left( \left( \min \left( \max \left( \frac{\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega} \right), \frac{4\rho \alpha_\omega k}{CD_{k\omega} y^2} \right) \right)^4 \right)$  — 1-ая функция перемешивания;  $CD_{k\omega} = \max \left( 2\rho \alpha_\omega \frac{1}{\omega} \nabla k \cdot \nabla \omega, 10^{-10} \right)$ ;  $F_2 = \tanh \left( \left( \max \left( \frac{2\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega} \right) \right)^2 \right)$  — 2-ая функция перемешивания;  $P_k = \min(\mathbf{v}_t \nabla \mathbf{U} \cdot [\nabla \mathbf{U} + (\nabla \mathbf{U})^T], 10 \cdot \beta^* k \omega)$  — ограничитель на нарастание турбулентности в режимах стагнации. Подробности можно найти в статьях [91–94] и в руководстве пользователя пакета OpenFOAM [105].

Турбулентная модель содержит восемь коэффициентов:  $\alpha_k$ ,  $\alpha_\omega$ ,  $\beta$ ,  $\gamma$ ,  $\beta^*$ ,  $a_1$ ,  $b_1$ ,  $c_1$ . Из которых  $\alpha_k$ ,  $\alpha_\omega$ ,  $\beta$ ,  $\gamma$  рассчитываются по принципу весового среднего:  $\gamma = \gamma_1 F_1 + \gamma_2 (1 - F_1)$  через  $\alpha_{k1}$ ,  $\alpha_{k2}$ ,  $\alpha_{\omega1}$ ,  $\alpha_{\omega2}$ ,  $\beta_1$ ,  $\beta_2$ ,  $\gamma_1$ ,  $\gamma_2$ . Стандартно константы турбулентной модели задаются следующими значениями (5.4) ([92–94]):

$$\begin{aligned}
\gamma_1 &= 5/9, & \beta_1 &= 3/40, & \alpha_{k1} &= 0.85, & \alpha_{\omega 1} &= 0.5, \\
\gamma_2 &= 0.44, & \beta_2 &= 0.083, & \alpha_{k2} &= 1, & \alpha_{\omega 2} &= 0.856, \\
\beta^* &= 0.09, & a_1 &= 0.31, & b_1 &= 1.0, & c_1 &= 10.0.
\end{aligned}
\tag{2.16}$$

Детали описания модели турбулентности представлены в [106; 107].

## 2.4 Начальные условия

В разных задачах, связанных с моделированием потоков на горных склонах, могут быть разные начальные условия. Например, при моделировании снежных лавин, кроме рельефа склона и свойств снежного покрова, задается так называемая зона зарождения - верхняя часть лавинного очага, где начинается движение снега в виде лавины. Первоначально покоившийся в этой зоне снег под влиянием метеорологических явлений, процессов внутри снежной толщи и других возможных факторов меняет свои свойства и возникает движение. Начальная скорость в этом случае считается равной нулю. При моделировании потоков, возникающих при прорыве горного озера, начальная скорость тоже равна нулю.

Формально в этих случаях параметры  $k$  и  $\varepsilon$  или  $\omega$ , входящие в модели турбулентности, тоже должны быть равны нулю в начальный момент. Однако при расчетах по уравнениям  $k$ - $\varepsilon$  пункта 2.3.1 и  $k$ - $\omega$  SST пункта 2.3.2, соблюдение этого условия невозможно, и в начальный момент они задаются некоторыми ненулевыми малыми величинами [91; 108]. Начальные значения параметров турбулентности почти не влияют на решение задачи, однако от них зависит сходимость на первых шагах расчёта.

## 2.5 Граничные условия на свободной поверхности

Течения со свободными поверхностями представляют собой особенно сложный класс течений с подвижными границами. Положение границы известно только в начальный момент времени, её местоположение в более поздние

моменты времени должно быть вычислено. Если на свободной поверхности не происходит переходов массы, то используются следующие граничные условия на свободной поверхности [109].

Кинематическое граничное условие (2.17): нет потока через свободную поверхность (перемешивания тоже нет) [110; 111].

$$(\mathbf{U} - \mathbf{U}_{fs}) \cdot \mathbf{n}_{fs} = 0, \quad \text{или} \quad \dot{m}_{fs} = 0. \quad (2.17)$$

Динамическое граничное условие (2.18): вектор напряжений непрерывен.

$$(\mathbf{P}_{\mathbf{n}_{fs}})_l = (\mathbf{P}_{\mathbf{n}_{fs}})_g. \quad (2.18)$$

Здесь  $\mathbf{U}$  — скорость потока,  $\mathbf{U}_{fs}$  — скорость движения свободной поверхности,  $\mathbf{n}_{fs}$  — внешняя по отношению к потоку нормаль к свободной поверхности,  $\dot{m}_{fs}$  — поток массы через свободную поверхность,  $(\mathbf{P}_{\mathbf{n}_{fs}})_l$  — вектор напряжений в жидкости, действующий на площадке с нормалью  $\mathbf{n}_{fs}$ ,  $(\mathbf{P}_{\mathbf{n}_{fs}})_g$  — вектор напряжений в газе.

Граничные условия на свободной поверхности для турбулентных параметров [101; 110; 111] задаются нулевыми производными по нормали (2.19):

$$\frac{\partial k}{\partial n_{fs}} = 0, \quad \frac{\partial \varepsilon}{\partial n_{fs}} = 0, \quad \text{или} \quad \frac{\partial \omega}{\partial n_{fs}} = 0. \quad (2.19)$$

## 2.6 Граничные условия на дне потока

На границе поток-склон может происходить процесс разрушения потоком подстилающего материала и вовлечения разрушенного материала в движение. Если этого не происходит, то положение этой границы задано, на ней считаются выполненными условия прилипания  $\mathbf{U} = 0$ . Из закона сохранения количества движения следует, что  $(\mathbf{P}_{\mathbf{n}_s})_l = (\mathbf{P}_{\mathbf{n}_s})_s$  — вектор напряжений непрерывен на границе, так как скачок скорости отсутствует. В качестве граничных условий для турбулентных характеристик на твердой стенке обычно выставляются условия (2.20).

$$k = 0, \quad \frac{\partial \varepsilon}{\partial n_s} = 0, \quad \text{или} \quad \frac{\partial \omega}{\partial n_s} = 0. \quad (2.20)$$

Или граничные условия задаются через пристеночные функции [91; 108; 111].

Если разрушение материала склона и вовлечение его в движение происходит, то граница поток-склон перемещается, скорость этого перемещения определяет скорость притока массы в поток и заранее не известна. Если плотность разрушенного материала равна плотности неразрушенного материала, а скорость на границе поток-склон непрерывна (условие прилипания), то по-прежнему  $(\mathbf{P}_{n_s})_l = (\mathbf{P}_{n_s})_s$  на границе, а скорость перемещения границы  $\mathbf{U}_{entr}$  остается не определенной. Для определения  $\mathbf{U}_{entr}$  нужно задать дополнительное условие физической природы, связанное с механизмом разрушения склонового материала. В этой работе таким условием является условие на величины напряжений на поверхности склона. Подробно механизм разрушения и захвата донного материала описан в Главе 4.

## Глава 3. Трехмерная модель склонового потока без захвата склонового материала<sup>1</sup>

В этой главе дается другая формулировка математической модели, представленной в главе 2. Эта формулировка использована далее при создании численной модели. Кроме того, в этой главе приводятся примеры применения этой модели.

### 3.1 Определение формы свободной поверхности

Склоновый поток рассматривается как турбулентный несжимаемый поток со свободной поверхностью на склоне под действием силы тяжести.

Течения со свободными поверхностями представляют собой особенно сложный класс течений с подвижными границами. Положение границы известно только в начальный момент времени, её местоположение в более поздние моменты времени должно быть вычислено.

Существует много методов для определения формы свободной поверхности, они могут быть классифицированы в две основных группы:

- **Методы отслеживания интерфейса** — методы, которые рассматривают свободную поверхность как резкую границу раздела, движение которой отслеживается. В методах этого типа используются сетки, выровненные по свободной поверхности, которые перестраиваются каждый раз, когда свободная поверхность перемещается. Недостатком данного метода является необходимость использовать маленький шаг по времени при нахождении численного решения.
- **Методы захвата интерфейса** — методы, которые не определяют интерфейс как явную границу. В методах этого типа расчёт ведётся на фиксированной сетке, покрывающей среду по обе стороны от свобод-

---

<sup>1</sup>При подготовке данной главы диссертации использовались следующие публикации автора, в которых, согласно «Положению о присуждении ученых степеней в Московском государственном университете имени М.В.Ломоносова», отражены основные результаты, положения и выводы исследования: [4; 7; 10].



ной поверхности. Форма свободной поверхности определяется путем вычисления доли контрольного объёма по одну сторону от свободной поверхности для тех ячеек, которые пересекает свободная поверхность. Это можно вычислить, вводя безмассовые частицы на свободной поверхности в начальный момент времени и следя за их движением; это называется схемой Marker-and-Cell или MAC, предложенной Харлоу и Уэлшем в 1965 г. [112]. В качестве альтернативы можно решить уравнение переноса для доли ячейки, занятой жидкой фазой (схема Volume-of-Fluid или VoF предложенная Хиртом и Николсоном в 1981 г. [113]).

Для определения формы свободной поверхности в настоящей работе используется метод захвата границы (Volume of Fluid method или VoF) [114; 115].

В VOF-методах для нахождения положения свободной поверхности вводится параметр  $\alpha$ , определяющий заполненность жидкостью каждого контрольного объёма, так что  $\alpha = 1$  в заполненном жидкостью контрольном объёме и  $\alpha = 0$  в заполненном газом контрольном объёме и  $0 < \alpha < 1$  в контрольном объёме, содержащем границу раздела [114] как показано на рис. 3.1. Параметр  $\alpha$  определяет свободную поверхность, и переносится вместе с жидкостью со скоростью  $\mathbf{U}$ , следовательно эволюция  $\alpha$  определяется уравнением переноса (3.1)

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{U}) = 0. \quad (3.1)$$

Метод может быть применен к сложным формам свободной поверхности, включая опрокидывающиеся волны, всплески, отрывы отдельных капель, пузырьки и др. Также достоинством данного метода является вычислительная простота и отсутствие эмпирических моделей для межфазного трения, обмена импульсом и др.

Среда по обе стороны от границы раздела рассматривается как единая среда, свойства которой варьируются в пространстве в зависимости от объемной доли жидкости [114–116]:

$$\rho = \rho_l \alpha + \rho_g (1 - \alpha), \quad \mu = \mu_l \alpha + \mu_g (1 - \alpha), \quad (3.2)$$

где  $\rho_l$  — плотность жидкости,  $\rho_g$  — плотность газа,  $\mu_l$  — вязкость жидкости,  $\mu_g$  — вязкость газа.

0	0	0	0	0	0
0.9	0.8	0	0	0.6	0.7
1	1	0.3	0	0	0
1	1	1	0.5	0	0
1	0.2	1	0.9	0.2	0.3
1	1	1	1	1	1

Рисунок 3.1 — Определение свободной поверхности с помощью метода VoF.

В таблице 1 приведены используемые модели для определения  $\rho_l$ ,  $\mu_l$ ,  $\rho_g$ ,  $\mu_g$ .

Таблица 1 — Плотность и реологические свойства жидкости и газа

gas	liquid
Ньютоновская среда	Среда Хершеля-Балкли
$\mu_g = \text{const}$	$\mu_l = \min\left(\mu_0, \frac{\tau_0 + \kappa \mathbf{s} ^n}{2 \mathbf{s} }\right)$
$\rho_g = \text{const}$	$\rho_l = \text{const}$

Таким образом, в методе VoF свободная поверхность не рассматривается как граница — это просто место, где резко меняются свойства жидкости.

Тогда расчёт динамики в данном методе происходит по обе стороны от свободной поверхности. Вычисления производят в рамках одножидкостной модели для среды со сложными свойствами (3.2), зависящими от  $\alpha$  (см. [114–116]). Для описанной жидкости выполняется уравнение движения (2.1) и уравнение неразрывности (2.3).

### 3.2 Результирующая система уравнений

Объединяя всё вышеописанное, для описания исследуемого гравитационного турбулентного потока неньютоновской реологии со свободной поверхностью на склоне без захвата подстилающего материала используется система уравнений (3.3). К этой системе добавляются уравнения модели турбулентности.

$$\left\{ \begin{array}{l} \nabla \cdot \mathbf{U} = 0, \\ \frac{\partial(\rho \mathbf{U})}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{g}, \\ \rho = \rho_l \alpha + \rho_g (1 - \alpha), \quad \rho_g = \text{const}, \quad \rho_l = \text{const}, \\ \frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{U}) = 0, \\ \boldsymbol{\tau} = 2\mu_{eff} \mathbf{s}, \quad \mu_{eff} = \mu_m + \mu_t, \\ \mu_m = \mu_l \alpha + \mu_g (1 - \alpha), \quad \mu_g = \text{const}, \quad \mu_l = \min\left(\mu_0, \frac{\tau_0 + \kappa |\mathbf{s}|^n}{|\mathbf{s}|}\right). \end{array} \right. \quad (3.3)$$

Отметим, что в этой работе для тензора турбулентных напряжений принимается гипотеза Буссинеска (2.10), согласно которой он представляет собой сумму двух членов

$$\boldsymbol{\tau}_t = 2\mu_t \mathbf{s} - \frac{2}{3} \rho k \mathbf{I}.$$

Так как в несжимаемой среде первый инвариант тензора  $\mathbf{s}$  равен нулю, первое слагаемое называют девиаторной частью тензора турбулентных напряжений, второе — шаровой. В системе (3.3) выражение для  $\boldsymbol{\tau}$  содержит только девиаторную часть тензора турбулентных напряжений. Шаровая часть входит в член с  $p$ , при этом  $p$  есть сумма осреднённого по Рейнольдсу молекулярного давления и величины  $\frac{2}{3} \rho k$ .

### 3.3 Вычислительная модель

#### 3.3.1 Пакет OpenFOAM

Для решения задач течения жидкости на склоне в настоящей главе и далее используется свободно распространяемый пакет OpenFOAM. Для выполнения задач, поставленных в этой работе, потребовалось включить в стандартный пакет новые необходимые дополнительные модули, которые были созданы автором диссертации и опубликованы в интернете.

Для построения расчётной сетки на основе цифровой модели рельефа была разработана библиотека `avalancheUtils`, содержащая python-скрипты для генерации параметров сетки, а также начальных и граничных условий. Описание работы модуля приведено в разделе 3.6. Подробное описание скриптов, их функционала и их код приведены в разделе А. Библиотека доступна в интернете [117].

Для решения задачи течения потока по склону сложного рельефа с учётом неньютоновской реологии, турбулентного режима течения и захвата донного материала был разработан решатель `avalancheFoam`. Решатель, также включает в себя библиотеку `massTransferModel`, описывающую процесс разрушения и захвата потоком подстилающего материала. Описание решателя приведено в этой и следующей главах. Подробное описание кода и процесса разработки приведено в разделе Б. Решатель доступен в интернете [118]. Описание разработанного подхода можно также найти в статьях автора [6; 9].

Скажем несколько слов о самом пакете OpenFOAM и объясним выбор именно данного пакета в качестве основы для разработки программного комплекса.

OpenFOAM — это свободно распространяемое бесплатное программное обеспечение для вычислительной гидро- и аэромеханики с открытым исходным кодом, разрабатываемое в основном компанией OpenCFD Ltd с 2004 года. OpenFOAM обладает широким спектром функций для решения разных задач — от сложных потоков жидкости, включающих химические реакции, турбулентность и теплообмен, до акустики, механики твёрдого тела и электромагнетизма.

Пакет OpenFOAM предоставляет возможности

- препроцессинга (построение и конвертация сеток, создания особых начальных и граничных условий),
- решения задачи (с использованием как стандартных решателей, так и авторских),
- постпроцессинга (обработка и отрисовка данных с использованием различных скриптов, конвертеров).

Пакет также обладает рядом достоинств:

- возможность имплементации новых моделей,
- хорошая задокументированность,
- модульное устройство кода,
- широкое распространение, многочисленность разработчиков и пользователей.

OpenFOAM выпускается каждые шесть месяцев, чтобы включать новейшие усовершенствования, предлагаемые пользователями и разработчиками

Гарантия качества основана на тщательном тестировании. Процесс оценки, проверки и валидации кода включает в себя несколько сотен ежедневных модульных тестов, еженедельный запуск тестового набора среднего размера и запуск крупного тестового набора перед выпуском новых версий.

### 3.3.2 Используемые методы аппроксимации членов уравнений

Для решения задач о движении склонового потока на основе предложенной модели используется решатель `avalancheFoam` пакета OpenFOAM, предназначенный для расчёта задач течения со свободной поверхностью с определением положения свободной поверхности по методу VoF.

Используются следующие аппроксимационные схемы:

- производные по времени  $\frac{\partial}{\partial t}$  аппроксимируются с помощью неявного метода Эйлера;
- поток объёмной доли фазы  $\nabla \cdot (U\alpha)$  аппроксимируется с помощью противопотоковой схемы и интерполяцией ван Леера;
- поток массы смеси  $\nabla \cdot (\rho UU)$  аппроксимируется с помощью противопотоковой схемы с весами;

- дивергенция тензора вязких напряжений  $\nabla \cdot \boldsymbol{\tau}$  аппроксимируется с помощью центральной разностной схемы;
- поток турбулентной кинетической энергии  $\nabla \cdot (\rho U k)$  аппроксимируется противопотоковой схемой;
- поток диссипации турбулентной кинетической энергии  $\nabla \cdot (\rho U \varepsilon)$  используется противопотоковой схема;
- поток специальной диссипации турбулентной кинетической энергии  $\nabla \cdot (\rho \boldsymbol{u} \boldsymbol{\omega})$  используется противопотоковой схема;
- оператор градиента  $\nabla$  аппроксимируется центрально-разностной схемой;
- оператор Лапласа  $\nabla^2$  аппроксимируется центрально-разностной схемой с явной неортогональной коррекцией;
- другие, не перечисленные выше члены описываются с помощью центрально-разностной схемы.

Подробное описание используемых аппроксимационных схем можно найти в книгах [91; 105; 119].

### 3.3.3 Алгоритм решения системы уравнений

Для решения системы уравнений используется алгоритм PIMPLE [91; 105; 119; 120], являющийся комбинацией алгоритмов PISO (Pressure Implicit with Splitting of Operator) [121] и SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) [122].

Алгоритм метода PIMPLE можно описать следующим образом:

1. расчёт скоростей в потоке,
2. вычисление давления,
3. обновление значений скорости и давления,
4. PISO:
  - а) корректировка давления,
  - б) корректировка скорости,
  - в) обновление значений скорости и давления,
5. решение уравнений переноса (объёмной доли фазы, параметров турбулентности).

### 3.3.4 Методы решения СЛАУ

Приведём краткий обзор используемых в настоящей работе методов решения систем линейных алгебраических уравнений (СЛАУ), получаемых в процессе аппроксимации определяющих уравнений (3.3).

Для решения уравнений переноса, а также уравнения движения используется метод бисопряженных градиентов с предобуславливателем в виде диагонального LU разложения. Уравнение на давление решается методом сопряжённых градиентов с предобуславливателем GAMG (геометрически обусловленный многосеточный метод). Подробное описание используемых решателей доступно в [91; 105; 119].

## 3.4 Разработка модуля генерации сетки и начальных условий

Для расчёта потоков на склонах реальной геометрии необходимо учитывать рельеф местности. Учёт рельефа в настоящей работе происходит посредством вычислительной сетки, которая строится из параллелепипедов с основаниями вдоль склона.

Разработанный автором диссертации модуль генерации сетки принимает на вход растровый формат ASCII GRID топографической карты (цифровую модель рельефа (ЦМР)), которая может быть получен, например, из следующих открытых источников:

- ASTER Global Digital Elevation Map Announcement [123]
- USGS TopoView [124]
- ALOS (AW3D) OpenTopography High-Resolution Topography Data and Tools [125]

На рис. 3.2 а) можем видеть цифровую модель рельефа (ЦМР). Используется растровый формат ASCII GRID, в котором высоты записываются в равномерно распределённых вдоль линий широты и долготы точек пространства. На основе ЦМР создается расчётная сетка с ячейками (см. рис. 3.2 б)), основания которых параллельны склону, как показано на рисунке справа.

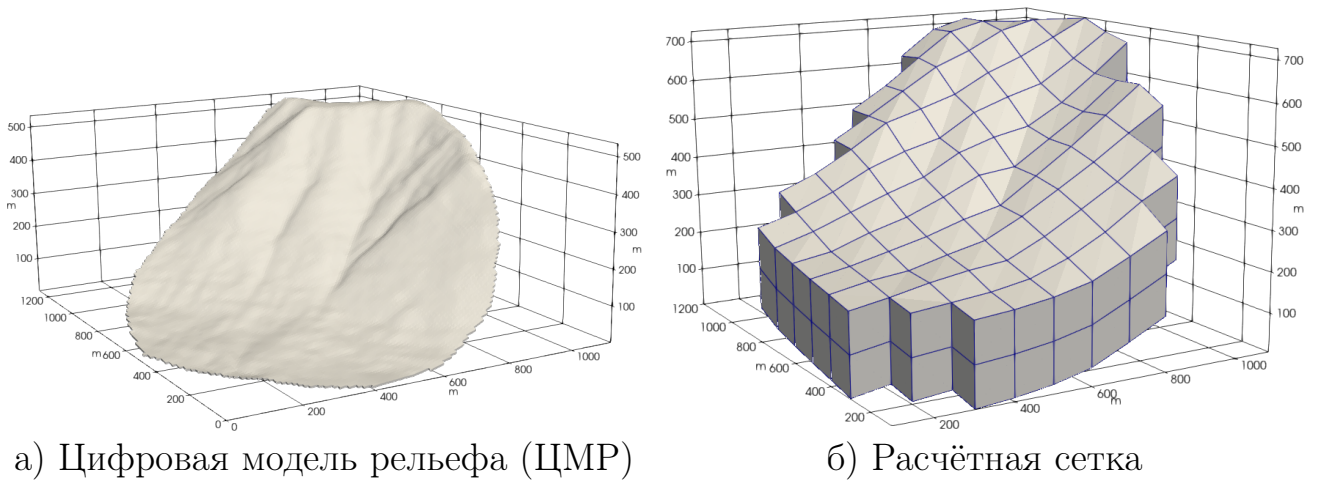


Рисунок 3.2 — Генерация расчётной сетки.

На рис. 3.2 б) размер ячейки составляет 100 метров. Эта сетки была создана для демонстрации работы модуля. Размер ячеек в расчётных сетках 22 лавинного очага варьируется от 0.5 до 5 метров.

Внутри расчётной области, приведённой на рис. 3.2 б), применяется метод VoF. Поток течёт внутри представленной области, граница поток-воздух вычисляется исходя из значения объёмной доли фазы, чаще всего, используя уравнение  $\alpha = 0.5$ .

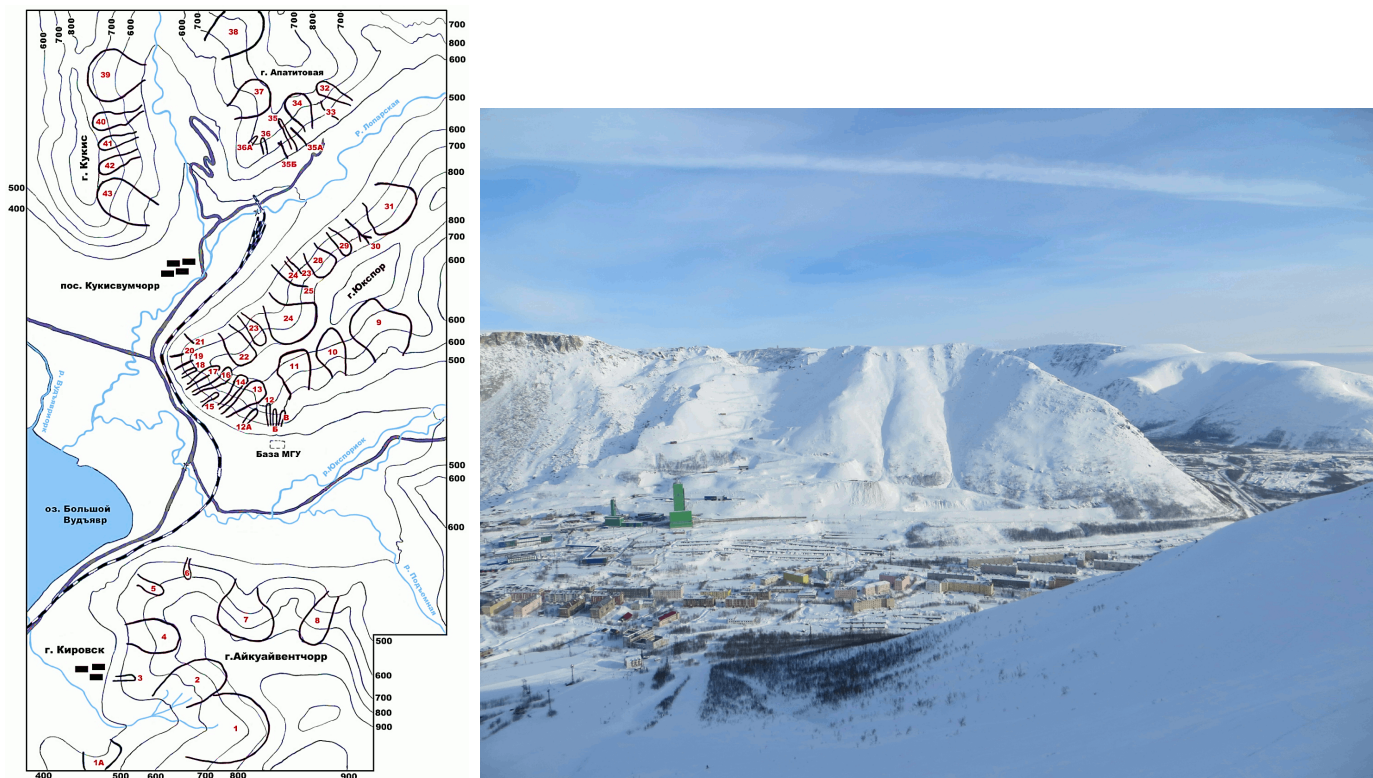
### 3.5 Примеры применения трехмерной модели без захвата подстилающего материала

#### 3.5.1 22-ой лавинный очаг горы Юкспор (Хибины)

Одним из объектов исследования в настоящей работе является 22-ой лавинный очаг горы Юкспор Хибинских гор. Данный лавинный очаг хорошо известен в кругу людей, работающих с лавинами, так как в нем часто сходят катастрофические лавины. Один из примеров — принудительный спуск лавины данном очаге в 22:00 18 февраля 2016 года. Лавина приобрела катастрофический характер и унесла жизни трёх людей, в том числе одного из сотрудников лавинной службы. Были засыпаны железная и автомобильная дороги, выбиты окна в трёх близлежащих домах.



В работе проводится расчёт лавины, сошедшей 5 апреля 1985 года. Паспорт лавины содержит данные о плотности снега, средней глубине снега в зоне зарождения и зоне лавинных отложений, формы зон зарождения и лавинных отложений. Паспорт также содержит информацию о метеорологических условиях, структуре снега, причинах схода лавины, которые не были использованы в настоящей работе.



а) Схема лавинных очагов в районе г. Кировска

б) Фотография 22-ого лавинного очага

Рисунок 3.3 — 22-ой лавинный очаг горы Юкспор (Хибины).

На рис. 3.3 а) показана схема лавинных очагов в районе г. Кировска, на рис. 3.3 б) приведена фотография склона горы Юкспор (Хибины) с 23 (слева) и 22 (справа) лавинными очагами в центре изображения (НИЛ снежных лавин и селей географического факультета МГУ).

На рис. 3.4 представлено фото последствий катастрофической лавины в 22-ом лавинном очаге, сошедшей 18 февраля 2016 года.

Моделирование снежной лавины в данном очаге проводится в настоящей работе.

Захват снега в процессе движения не учитывается, поток считается турбулентным, используется  $k$ - $\epsilon$  модель турбулентности.



Рисунок 3.4 — Фото последствий схода снежной лавины из 22-ого лавинного очага горы Юкспор (Хибины).

### Расчётная область

В качестве расчётной области взят участок склона горы Юкспор (Хибины), включающий 22 лавинный очаг (цифровая модель рельефа предоставлена научно-исследовательской лабораторией снежных лавин и селей географического факультета МГУ). Расчётная область представляет собой трёхмерную область, верхняя и нижняя границы которой коллинеарны склону, и удалены друг от друга по вертикали на 25 м. Склон и лавина находятся внутри этой расчётной области. Сетка содержит 50 слоёв по вертикали (50 ячеек). На рисунке 3.5 показана цифровая модель рельефа 22 лавинного очага (правое русло), зелёным цветом отмечена зона зарождения лавины, голубым — зона лавинных отложений.

Расчётная сетка была создана с помощью разработанного автором модуля преобразования ЦМР (см. раздел 3.6).

Расчётная область содержит 1.5 млн. ячеек.

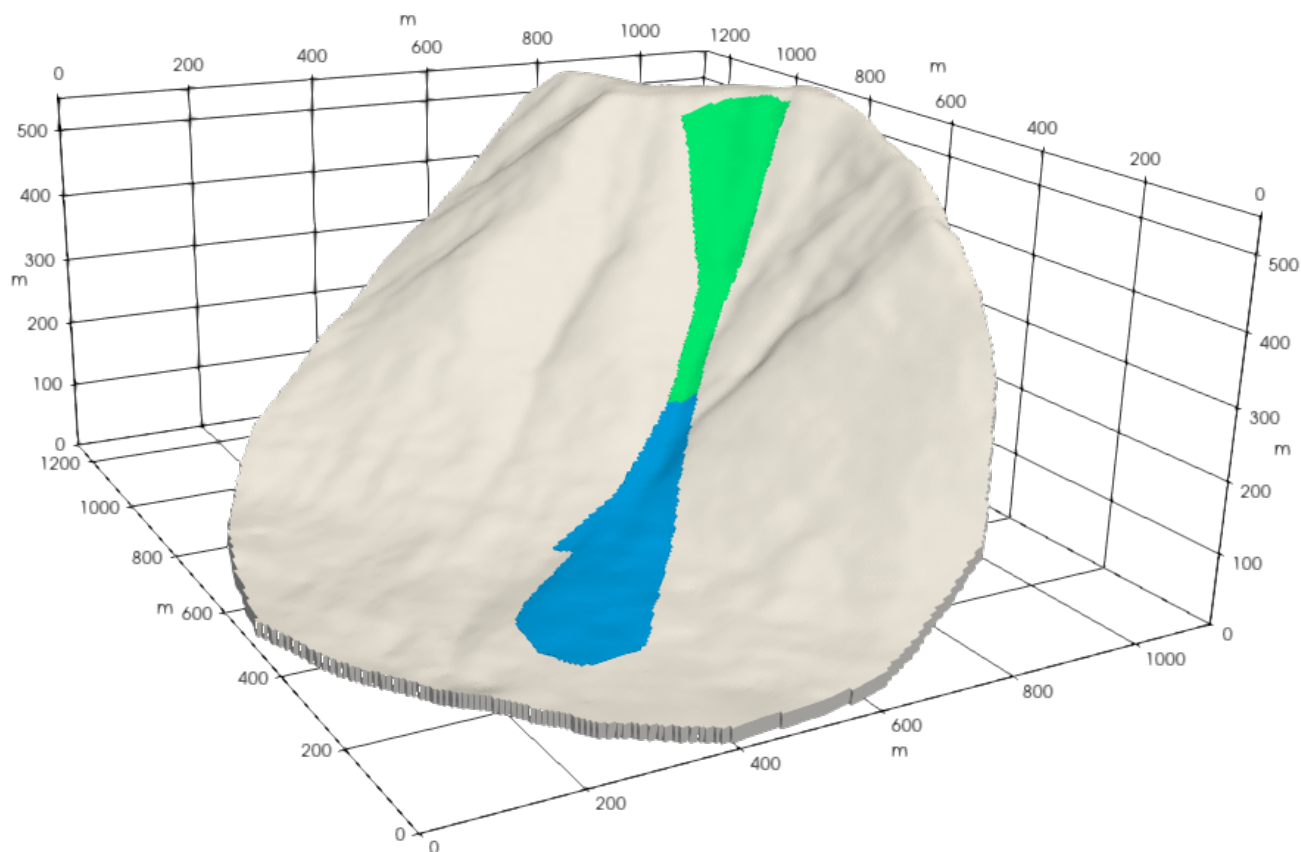


Рисунок 3.5 — Цифровая модель рельефа 22 лавинного очага с зонами зарождения лавины (зелёным) и лавинных отложений (голубой).

### Начальные условия

В задаче расчёта лавинного потока в 22-ом лавинном очаге в начальный момент времени задан неподвижный слой снега толщиной 1.5 метра в зоне зарождения. Слой снега удерживается виртуальной дамбой, которая удаляется в момент начала расчёта, начальная скорость равна 0 во всей расчётной области. В качестве начальных условий для турбулентной кинетической энергии используется значение  $0.1 \text{ м}^2\text{с}^{-2}$  и диссипации турбулентной кинетической энергии используется значение  $0.1 \text{ м}^2\text{с}^{-3}$ , как описано в разделе 2.4. Начальные данные для объёмной доли жидкости определяют толщину слоя снега и его положение в зоне зарождения лавины.

## Граничные условия

Расчётная область для 22-ого лавинного очага содержит 2 типа границ: нижняя и верхняя, далее условно называемые «склон» и «атмосфера», боковые границы расчётной области (вертикальные боковые стенки расчётной области) относятся к атмосфере, так как они удалены от непосредственной области течения потока. На рис. 3.2 можно наглядно увидеть боковые границы расчётной области и верхнюю границу, называемую атмосферой. Граничные условия перечислены в таблице 2:

Таблица 2 — Условия на границах расчётной области в задаче расчёта снежной лавины на горе Юкспор (Хибины)

	«атмосфера»	«склон»
$U$	нулевая производная по нормали	условие прилипания
$p$	нулевая производная по нормали	нулевая производная по нормали
$\alpha$	нулевая производная по нормали	нулевая производная по нормали
$k$	нулевая производная по нормали	нулевая производная по нормали
$\varepsilon$	нулевая производная по нормали	использование пристеночной функции

Более подробную информацию о используемых граничных условиях можно найти в книгах [91; 105; 119].

## Параметры потока

В расчёте использовались следующие константы (3.4) реологической модели Хершеля-Балкли 2.2.2:

$$\rho = 200 \text{ кг/м}^3, \quad \tau_0 = 10^4 \text{ Па}, \quad n = 0.5, \quad \kappa = 1000 \text{ Па} \cdot \text{с}^n. \quad (3.4)$$



Параметр плотности снежной лавины, а также глубина снега в зоне зарождения брались из паспорта лавины 1985 года, содержащего также границы зоны зарождения лавины и зоны лавинных отложений. Остальные параметры не были исследованы в натуре, вследствие чего подбирались в процессе калибровки модели с учетом рекомендуемых в литературе реологических параметров для плотных сухих лавин [45; 51; 80]. В процессе калибровки было произведено 108 расчётов, что заняло 36 часов машинного времени.

## Результаты моделирования

На рис. 3.6 и 3.7 показаны рассчитанные границы потока и величины скоростей на его поверхности в разные моменты времени. Отметим, что границы рассчитывались по методу VOF, к потоку отнесены те ячейки, в которых объемная доля материала потока была больше или равна 0.5. Тёмно серым отмечена натурная зона лавинных отложений.

По данным расчёта были получены результаты:

- Течение происходит в течении 50 секунд;
- Средняя скорость в потоке составляет порядка 25 м/с, максимальная скорость в потоке составила порядка 32 м/с;
- Средняя толщина снега в зоне лавинных отложений, полученная в расчёте, составляет 0.7 м, средняя толщина лавинных отложений в паспорте представленная в паспорте лавины составляет 0.72 м (ошибка моделирования составляет 3%);
- Была получена информация о распределении параметров течения (скорости, давления) в пространстве с течением времени;
- Была получена информация о положении границ потока в различные моменты времени;
- Получена форма зоны лавинных отложений (совпадение с натурными данными составило 80% при оценке пиксельным методом), была рассчитана дальность выброса лавины (точность составила 96% по оценке вдоль линии от самой высокой точки зоны зарождения лавины, до самой нижней точки лавинных отложений).

Дополнительные данные о моделировании можно найти в статье [7].

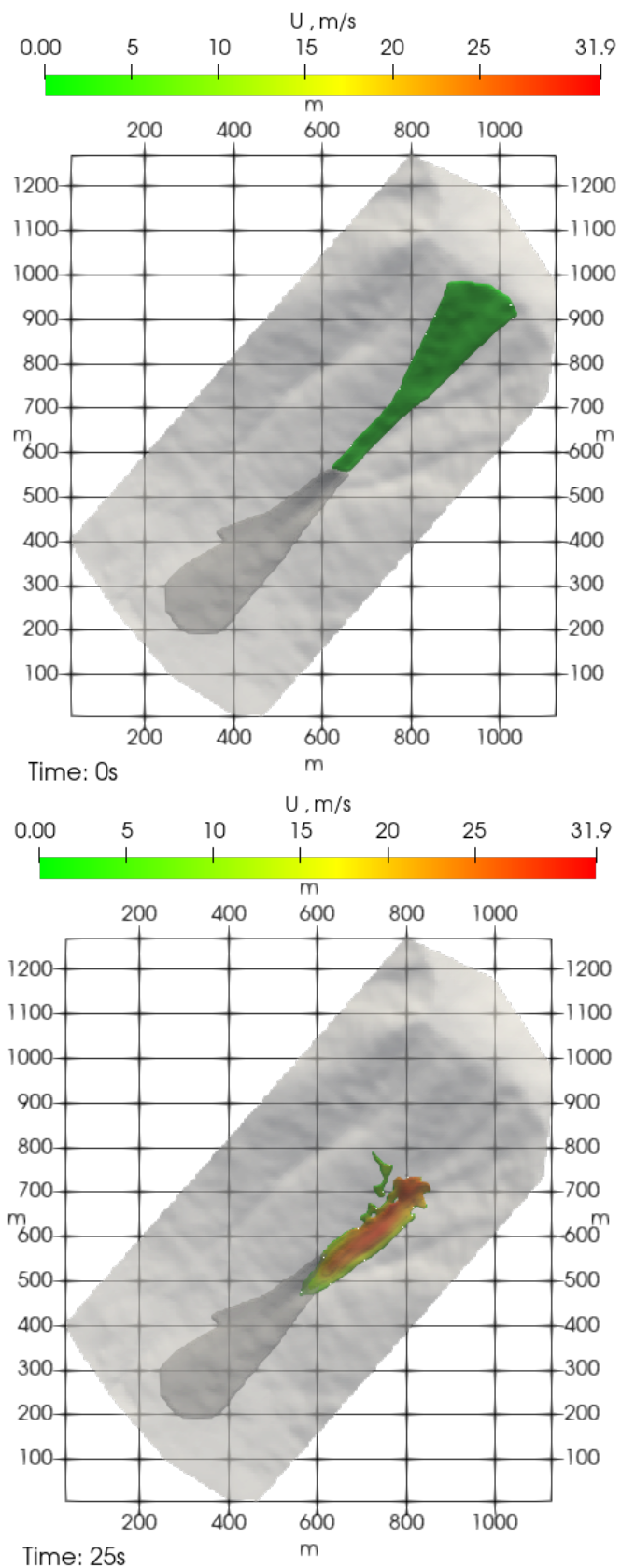


Рисунок 3.6 — Расчёт динамики снежной лавины в 22-ом лавинном очаге горы Юкспор (Хибины).

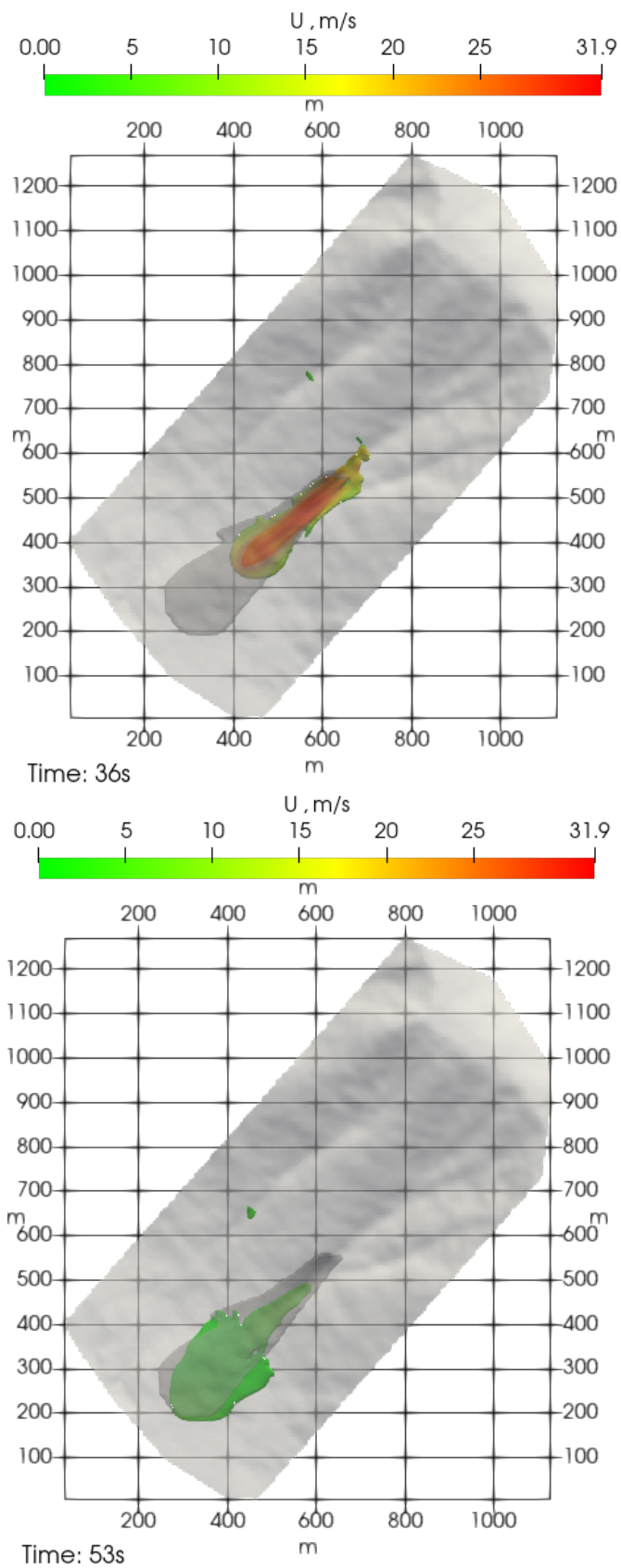


Рисунок 3.7 — Расчёт динамики снежной лавины в 22-ом лавинном очаге горы Юкспор (Хибины).

### 3.5.2 Эксперимент по исследованию взаимодействия водоснежного потока с комплексом защитных сооружений (Университет Исландии)

Эксперимент, поставленный в университете Исландии [126; 127] (далее эксперимент УИ), был спроектирован с целью воспроизвести взаимодействие водоснежного потока с различными препятствиями. В настоящей работе проводится моделирование и сравнение с экспериментом двух вариаций эксперимента: 1) с 3 дамбами и 2) без дамб. Для проведения эксперимента УИ использовалась водопроводная вода, так как она является легко доступным материалом и схожа по плотности с насыщенным водоснежным потоком. В том числе вода позволяет воспроизвести отличия в динамике водоснежных потоков от сжимаемых потоков гранулированных сред с высоким числом Фруда. Эксперимент был спроектирован таким образом, чтобы число Фруда было близко к типичным значениям числа Фруда в природных водоснежных потоках: в диапазоне  $3 < Fr < 5$ . Эксперимент проводился в длинном деревянном желобе с прозрачными стенками из оргстекла, который состоял из четырех секций с разными углами наклона ( $55^\circ$ ,  $27^\circ$ ,  $11.7^\circ$ ,  $0^\circ$  — слева направо). Нижние 5.05 м желоба были горизонтальными, а верхние его части были покрыты ковром для увеличения трения и снижения скорости потока и, следовательно, числа Фруда для получения целевого диапазона чисел Фруда. Защитные конструкции были выполнены из фанеры. Две небольших защитных дамбы и одна основная удерживающая (самая высокая) дамба располагалась на нижней горизонтальной части желоба, как показано на рисунке 3.8.

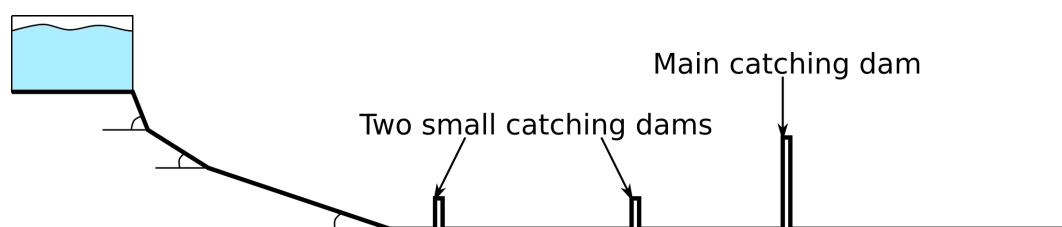


Рисунок 3.8 — Схема экспериментального лотка УИ с дамбами.

На рис. 3.9 показана схема эксперимента без дамб.





Рисунок 3.9 — Схема экспериментального лотка УИ без дамб.

### Расчётная область

Для расчёта эксперимента, поставленного в университете Исландии использовалась двумерная расчётная область толщиной 1 м, содержащая 257 тысяч ячеек.

На рис. 3.10 показана расчётная область в начальный момент расчёта. Синим цветом на рис. 3.10 обозначена жидкость, чёрным обозначен воздух.

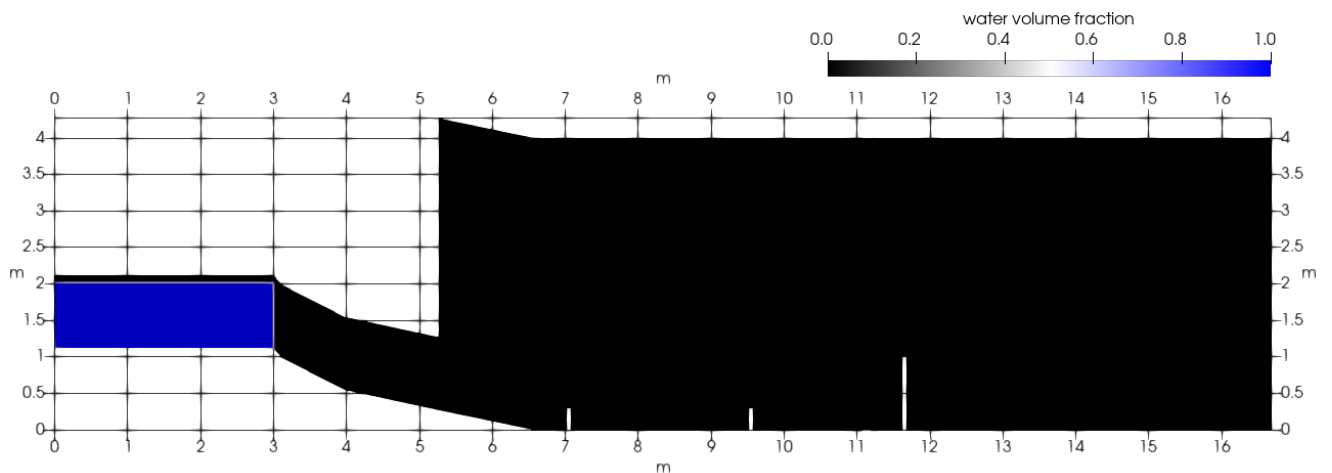


Рисунок 3.10 — Область расчета и начальные условия для моделирования эксперимента УИ с дамбами.

На рис. 3.11 показан начальный момент расчёта эксперимента УИ без дамб, проведённого в настоящей работе. Так же на рис. 3.11 жёлтой линией отмечено сечение замера скорости в потоке.

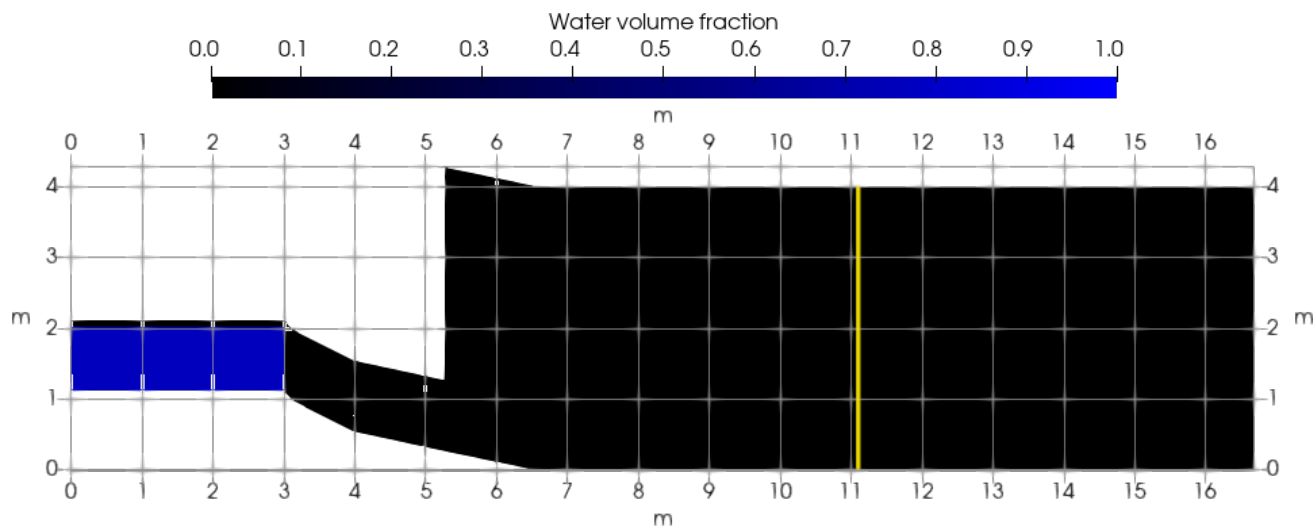


Рисунок 3.11 — Область расчета и начальные условия для моделирования эксперимента УИ без дамб. Жёлтая линия показывает расположение сечения замера скорости потока.

### Начальные условия

Начальные условия заданы таким образом, что вода, глубиной 0.9 м располагается в стартовом резервуаре (обозначена синим на рисунке 3.10 и 3.11) и удерживается виртуальной границей, исчезающий с началом расчёта.

### Граничные условия

В задаче выделены следующие границы: дно лотка (обозначено зелёной линией на рисунке 3.12), верхняя граница лотка (обозначена розовой линией на рисунке 3.12), ковер (обозначен оранжевым на рисунке 3.12), боковые границы расчётной области (передняя и задняя стенки).

В таблице 3 приведены условия на границах расчётной области.

На границе, определяющей ковёр также устанавливается специфическое условие для турбулентной вязкости, определяющее шероховатость поверхности [105].

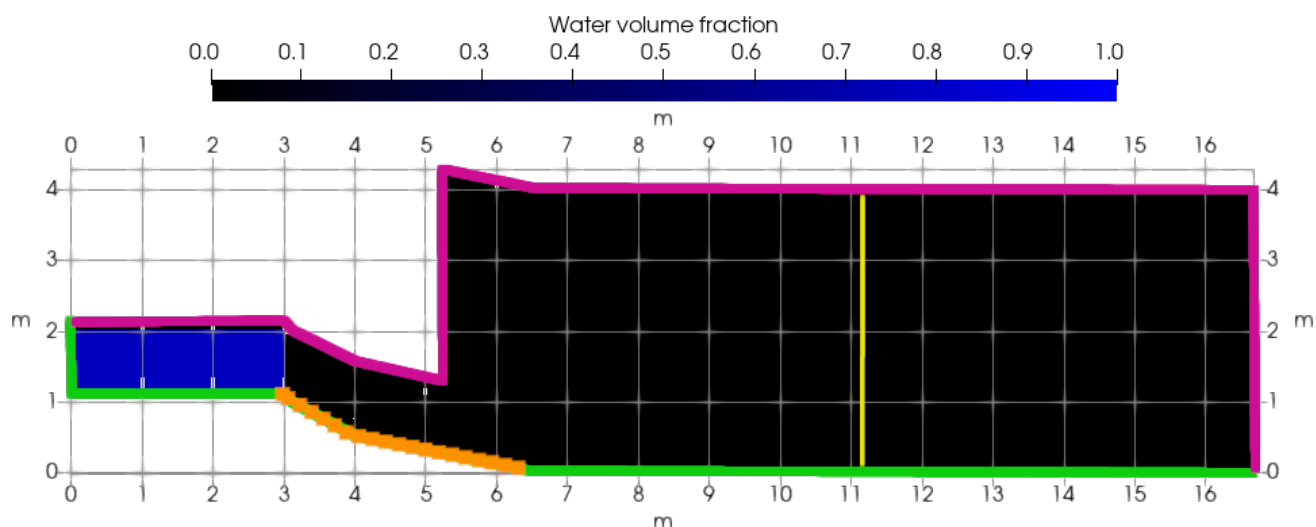


Рисунок 3.12 — Границы расчётной области.

Таблица 3 — Граничные условия в задаче расчёта снежной лавины на горе Юкспор (Хибины)

	атмосфера	боковые стороны	склон, ковёр
$U$	нулевая производная по нормали	нулевая производная по нормали	условие прилипания
$p$	нулевая производная по нормали	нулевая производная по нормали	нулевая производная по нормали
$\alpha$	нулевая производная по нормали	нулевая производная по нормали	нулевая производная по нормали
$k$	нулевая производная по нормали	нулевая производная по нормали	нулевая производная по нормали
$\varepsilon$	нулевая производная по нормали	нулевая производная по нормали	пристеночная функция
$\omega$	нулевая производная по нормали	нулевая производная по нормали	пристеночная функция

### Результаты моделирования

На рис. 3.13 показан момент взаимодействия потока с основной дамбой. Показан момент максимальной высоты всплеска потока на основной дамбе.

Также было проведено моделирование эксперимента, поставленного в Университете Исландии (УИ), при отсутствии защитных сооружений 3.9. На

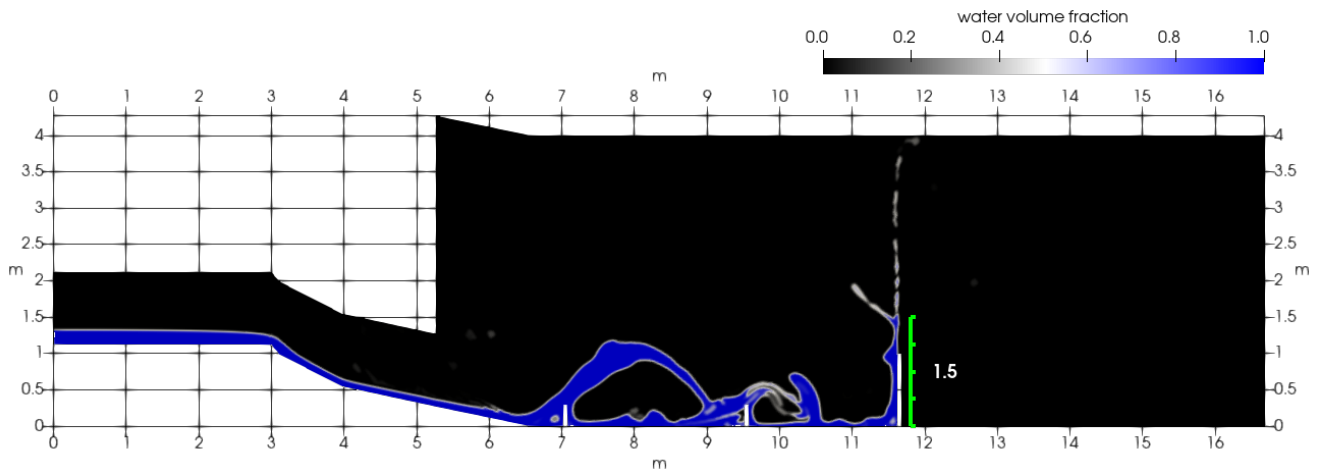


Рисунок 3.13 — Взаимодействие потока с тремя дамбами (результат численного моделирования, проведенного в этой работе). На плотинах наблюдаются всплески и образование струй. Цвета связаны со значениями  $\alpha$  (т. е. относительной части ячейки, занятой жидкостью): синий — 1, черный — 0. Белые линии — ПЛОТИНЫ.

Таблица 4 — Сравнение измеренных и рассчитанных параметров потока.

	Высота первого всплеска на основной дамбе	Время с начала взаимодействия потока с основной дамбой, до окончания переливания потока через дамбу	Объем, удержанный дамбой из 2.7 м <sup>3</sup>
Эксперимент	1.3 м	1.25 с	2.684 м <sup>3</sup>
$k$ - $\omega$ SST модель турбулентности	1.95 м	1.3 с	2.635 м <sup>3</sup>
$k$ - $\epsilon$ модель турбулентности	1.64 м	1.3 с	2.528 м <sup>3</sup>

расстоянии 11.1 м от левого края установки производился замер скорости потока.

Проведено сравнение средней по глубине потока скорости в экспериментальном лотке Университета Исландии при отсутствии дамб, показанное на рис. 3.14, в сечении, находящемся на расстоянии 11.1 м от начала желоба.

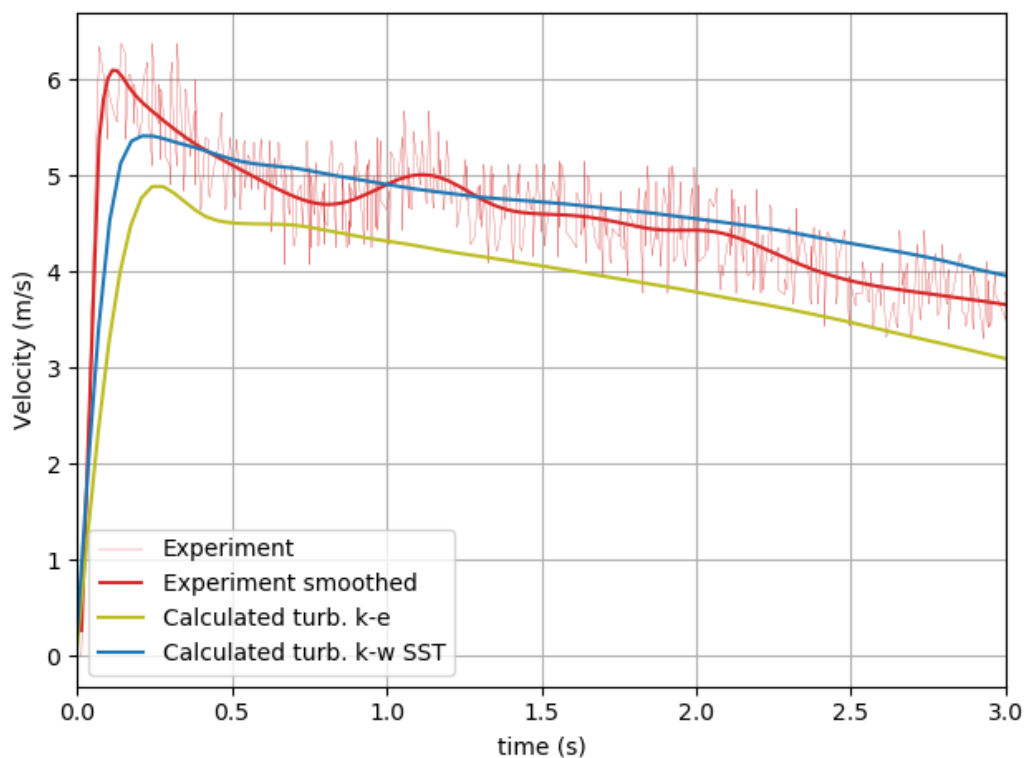


Рисунок 3.14 — Сравнение средней по глубине потока скорости в экспериментальном лотке Университета Исландии при отсутствии дамб, в сечении, находящемся на расстоянии 11.1 м от начала желоба.

### 3.6 Выводы

Проведённое исследование показало состоятельность модели в следующих аспектах:

- Модель позволяет учитывать сложный рельеф местности. Разработанный модуль генерации сетки 3.6 использует стандартную растровую топографическую карту, что упрощает взаимодействие других пользователей с разрабатываемым программным комплексом. Также это избавляет других пользователей от дополнительной предобработки данных.
- Расчёт производится для реологической модели Хершеля-Балкли. Используемые реологические параметры позволили описать остановку потока на склоне с хорошим совпадением с натурными данными.
- Для описания турбулентного режима течения используются  $k-\epsilon$  и  $k-\omega SST$  модели турбулентности.

- Расчёт взаимодействия потока с препятствием показан на примере эксперимента УИ с дамбами. Результаты достаточно хорошо согласуются с экспериментальными данными. Для увеличения точности расчёта в главе 5 проводится оптимизация коэффициентов турбулентной модели.

## Глава 4. Разработка модели разрушения потоком дна и вовлечения донного материала в движение<sup>1</sup>

Многие склоновые потоки при движении вовлекают подстилающий (склоновый) материал. При этом масса потока увеличивается, иногда в несколько раз [61; 72; 87; 128]. Это приводит к увеличению не только размеров потока, но и скорости, дальности выброса, а также сил, действующих на объекты, находящиеся на пути потока. Знание объем массы остановившегося в зоне отложения материала важно при создании защитных сооружений. Прирост массы проверяется сравнением начальной массы и массы отложений. Имеются также работы, в которых описываются результаты измерения баланса массы в процессе движения потока, полученные с использованием радаров [72].

Создание математических моделей склоновых потоков, учитывающих вовлечение подстилающего материала, очень важно. В Главе 1 приведен обзор различных механизмов захвата и обсуждаются модели гидравлического типа. В этой главе строится трехмерная модель, учитывающая базальный захват склонового материала, то есть, происходящий на границе поток-склон.

### 4.1 Описание разрабатываемого подхода

Предполагается, что захват подстилающего донного материала включает два процесса: 1) разрушение структуры этого материала, 2) вовлечение его в движение. Для учёта явления захвата донного материала, а также отложения материала потока на склоне, удобно ввести две дополнительных фазы: неподвижный материал склона, который может быть вовлечён в движение и разрушенный потоком материал склона. Обе этих фазы обладают свойствами, отличными от свойств движущегося материала потока и воздуха. Также необходимо задать условие фазового перехода материала склона в разрушенный

---

<sup>1</sup>При подготовке данной главы диссертации использовались следующие публикации автора, в которых, согласно «Положению о присуждении ученых степеней в Московском государственном университете имени М.В.Ломоносова», отражены основные результаты, положения и выводы исследования: [6; 9].

материал. Основное предположение состоит в том, что разрушение и захват подстилающего материала потоком происходит при превышении сдвиговыми напряжениями на границе поток-склон порогового значения, определяемого пределом прочности донного материала [51].

Когда, например, снег покоится на склоне горы, он неподвижен, его, в том числе, удерживают кристаллические связи, образующиеся в снежной толще в процессе метаморфизма. Под действием напряжений, создаваемых лавиной данные связи разрушаются, ломаются и сами снежные кристаллы. Реологические свойства разрушенного материала отличаются от реологических свойств материала склона. Материал склона обладает высокой вязкостью и пределом текучести, что объясняет его возможность покоиться на склоне. Когда напряжения превышают предел текучести, слой, где это произошло начинает своё медленное движение. Такое поведение характерно для начального момента лавинного потока, когда отрывается снежная плита и начинает свое движение. Далее этот слой рушится и происходит фазовый переход в разрушенный материал. Аналогичные рассуждения можно провести и для грязе-каменного потока. Материалы движущегося потока, покоящегося материала склона и захваченного материала будут обладать различными реологическими свойствами.

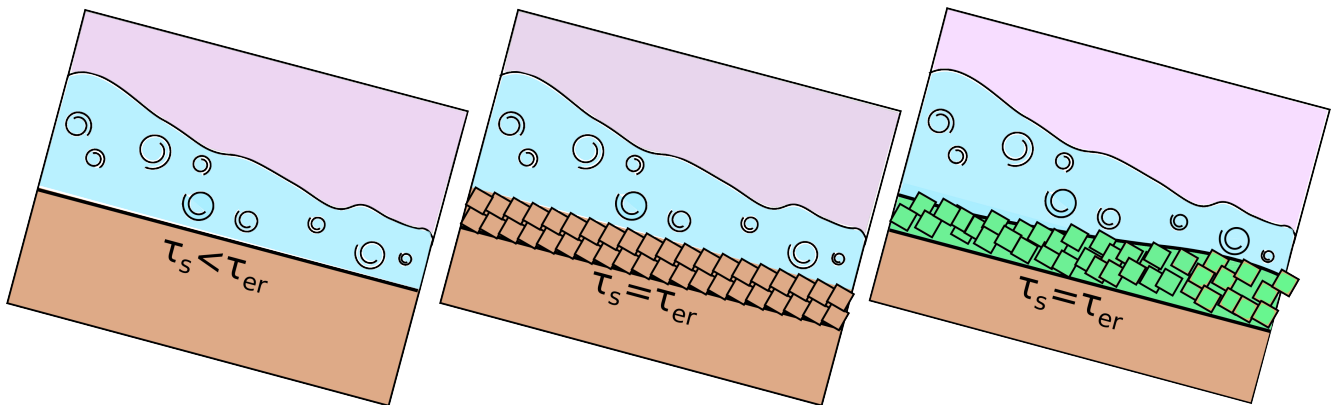


Рисунок 4.1 — Разрушение и захват донного материала.

На рис. 4.1 схематично показан процесс захвата. На самой левой картинке мы видим поток, со сдвиговыми напряжениями на дне не превышающими пороговое значение. На средней картинке изображен процесс разрушения склонового материала, разрушение происходит, когда напряжения на дне соответствуют пороговому значению. На правом рисунке разрушенный материал движется.



## 4.2 Гипотеза о захвате

За основу взята следующая гипотеза о захвате, используемая в работах [32; 51; 80]: если касательное напряжение на границе поток-подстилающий материал достигает предела прочности подстилающего материала, происходит разрушение и вовлечение в движение.

При формулировке гипотезы задавалось условие на касательное напряжение на границе поток-склон. Однако расчёт касательного напряжения на склоне сложной поверхности затруднителен. Поэтому сформулируем условие разрушения как условие на второй инвариант тензора напряжений. Данное обобщение также позволяет учесть фронтальный тип захвата.

Аналогично поступают при формулировке условия возникновения пластических деформаций в теории пластичности, заменяя условие Треска на касательное напряжение на площадке условием Мизеса на второй инвариант тензора напряжений. При подходящем подборе константы для предельного значения этого второго инварианта, получаются результаты, близкие к тем, что получены по более физической теории Треска [109]. Такой же прием используется при обобщении закона Хершеля-Балкли.

Тогда гипотеза о разрушении подстилающего материала и захвате его в движение может быть сформулирована следующим образом: если второй инвариант тензора напряжений на границе поток-подстилающий материал достигает предельного значения, описывающего прочность подстилающего материала, происходит разрушение.

## 4.3 Математическая модель разрушения потоком дна и вовлечения донного материала в движение

Для расчета параметров на границе поток-склон, как и в предыдущей главе, применяется метод VOF. Граница рассматривается как узкая область, в которой свойства среды меняются от свойств подстилающего материала до свойств разрушенного (захваченного) материала. Объемные доли этих материалов обозначаются индексами « $s$ » (склон, slope) и « $e$ » (захваченный, entrained).

$$\begin{aligned}\frac{\partial \alpha_s}{\partial t} + \nabla \cdot (\mathbf{U} \alpha_s) &= -q_{s \rightarrow e}, \\ \frac{\partial \alpha_e}{\partial t} + \nabla \cdot (\mathbf{U} \alpha_e) &= q_{s \rightarrow e}.\end{aligned}\tag{4.1}$$

Здесь  $\mathbf{U}$  — скорость среды;  $\alpha_i$  — объёмная доля  $i$ -ой фазы;  $q_{s \rightarrow e}$  — это скорость изменения объёмной доли материала склона за единицу времени за счет его разрушения и перехода в захваченный материал.

При таком подходе нужно задавать реологические свойства донного материала. Будем считать, что он описывается моделью Хершеля-Балкли. В реологической модели Хершеля-Балкли присутствует предел текучести, который отвечает за движение материала, однако он не может определять процесс разрушения и захвата. Так как в толще разрушаемого материала склона напряжения растут с глубиной, но материал должен лежать на склоне, исходя из постановки задачи. То есть предел текучести материала склона больше его предела прочности при захвате.

В идеализированной постановке в рамках метода VoF захват будет происходить в виде перехода материала склона при его разрушении в фазу захваченного материала на границе потока со склоном, как показано в уравнении (4.2).

$$\text{если } \delta_{s/(f+e)} = 1 \text{ и } |\boldsymbol{\tau}| \geq \tau_{er}, \text{ то } \alpha_s = 0, \alpha_e = 1.\tag{4.2}$$

Тензор напряжений является известной функцией тензора скоростей деформаций  $\mathbf{s}$ . Оперировать последним в расчётах значительно удобнее. Тогда запишем модель захвата как условие на  $\mathbf{s}$ . В численной модели захват определяет межфазный переход  $q_{s \rightarrow e}$ .

$$q_{s \rightarrow e} = \begin{cases} 0, & \text{если } |\mathbf{s}| < s_{er} \text{ или } \delta_{s/(f+e)} = 0, \\ C_{er}, & \text{если } |\mathbf{s}| \geq s_{er} \text{ и } \delta_{s/(f+e)} = 1. \end{cases}\tag{4.3}$$

Здесь  $C_{er}$  — коэффициент, характеризующий скорость захвата,  $s_{er}$  — предел скорости сдвига материала склона, который задаёт его предел прочности.

Коэффициент, характеризующий скорость захвата оценивает её исходя из значения второго инварианта тензора скоростей деформаций (скорости сдвига) на границе склон-поток, предельного значения скорости сдвига материала склона, градиента скорости сдвига и размера ячейки расчётной области:

$$C_{er} = \frac{\chi S}{V \Delta t}, \quad (4.4)$$

$$\chi = \frac{|\mathbf{U}|}{|\mathbf{s}|}. \quad (4.5)$$

где  $\chi$  — характеризует глубину разрушения по формуле (4.5) исходя из полученных в работах [49; 129] данных о линейности профиля скорости в потоке у дна при наличии захвата,  $S$  — единица площади свободной поверхности,  $V$  — элементарный объём, включающий рассматриваемую часть границы,  $\Delta t$  — единица времени.

#### 4.4 Результирующая система уравнений

Для описания исследуемого гравитационного турбулентного потока неньютоновской реологии со свободной поверхностью на склоне с разрушением и захватом подстилающего материала используется система уравнений (4.6). Для определения свободной поверхности потока и границы поток-склон вводятся объёмные доли воздуха ( $a$  — air), материала движущегося потока ( $f$  — flow), разрушаемого материала дна ( $s$  — slope) и захваченного материала ( $e$  — entrained). К системе (4.6) добавляются уравнения модели турбулентности 2.3.1 и 2.3.2.

$$\left\{ \begin{array}{l}
\nabla \cdot \mathbf{U} = 0, \\
\frac{\partial(\rho\mathbf{U})}{\partial t} + \nabla \cdot (\rho\mathbf{U}\mathbf{U}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho\mathbf{g}, \\
\rho = \rho_a\alpha_a + \rho_f\alpha_f + \rho_s\alpha_s + \rho_e\alpha_e, \\
\frac{\partial\alpha_a}{\partial t} + \nabla \cdot (\mathbf{U}\alpha_a) \\
\frac{\partial\alpha_f}{\partial t} + \nabla \cdot (\mathbf{U}\alpha_f) = 0, \\
\frac{\partial\alpha_s}{\partial t} + \nabla \cdot (\mathbf{U}\alpha_s) = -q_{s \rightarrow e}\delta_{s/(f+e)}, \\
\frac{\partial\alpha_e}{\partial t} + \nabla \cdot (\mathbf{U}\alpha_e) = q_{s \rightarrow e}\delta_{s/(f+e)}. \\
\mu_m = \mu_a\alpha_a + \mu_f\alpha_f + \mu_s\alpha_s + \mu_e\alpha_e.
\end{array} \right. \quad \boldsymbol{\tau} = 2\mu_{eff}\mathbf{s}, \quad \mu_{eff} =$$

(4.6)

Здесь  $\delta_{s/(f+e)}$  — дельта функция, определяющая границу разрушаемого материала склона и движущегося потока ( $\delta_{s/e} = 0$  всюду, кроме границы между подстилающим материалом и движущимся потоком, который состоит из материала потока и захваченного материала).  $\delta_{s/(f+e)}$  определяется исходя из объёмной доли материала склона:  $\delta_{s/(f+e)} = 1$  если  $0 < \alpha_s < 1$ .

#### 4.5 Численная модель разрушения потоком дна и вовлечения донного материала в движение

Для программной реализации механизма захвата был разработан решатель `avalancheFoam`, на базе решателя `multiphaseInterFoam`, позволяющий рассчитывать потоки неньютоновских сред с учетом разрушения потоком донного материала и вовлечением его в движение [9]. Отличием нового решателя, от предыдущего является наличие фазового перехода, задающего переход разрушенного материала склона в движущийся захваченный материал. Существуют различные вариации реализаций фазового перехода в многофазных решателях, в настоящей работе была использована идея реализации фазового перехода через процедуру MULES (Multidimensional Universal Limiter for Explicit Solution), описанная в работе [130].

С помощью алгоритма MULES решаются уравнения транспорта объёмных долей фаз. В рамках данного алгоритма источниковые члены имеют вид, представленный в формуле (4.7).

$$\frac{\partial \alpha_i}{\partial t} + \nabla \cdot (\mathbf{U} \alpha_i) = S_{imp_i} \cdot \alpha_i + S_{exp_i}. \quad (4.7)$$

Разделение источникового члена на явный  $S_{exp}$  и неявный  $S_{imp}$  вид сделано с целью увеличения диагонального преобладания матрицы, что в свою очередь снижает число обусловленности.

Тогда уравнения для фаз материала склона и захваченного материала потока можно переписать в виде (4.8).

$$\begin{aligned} \frac{\partial \alpha_e}{\partial t} + \nabla \cdot (\mathbf{U} \alpha_e) &= S_{exp_e}, \\ \frac{\partial \alpha_s}{\partial t} + \nabla \cdot (\mathbf{U} \alpha_s) &= S_{imp_s} \cdot \alpha_s. \end{aligned} \quad (4.8)$$

Где члены  $S_{imp_s}$  и  $S_{exp_e}$  принимают значения представленные в формулах (4.9) и (4.10), соответственно.

$$S_{imp_s} = q_{er}, \quad (4.9)$$

$$S_{exp_e} = -\frac{q_{er}}{\alpha_s} \alpha_s, \quad (4.10)$$

$$q_{er} = \delta_{er} \min(\alpha_s, C_{er}) \quad (4.11)$$

Коэффициент  $\delta_{er}$  был введен (4.12) для реализации условной операции, описанной в (4.3), с целью обеспечить непрерывность члена. Также, используемый вариант реализации условной операции вычислительно более эффективен. Вид  $\delta_{er}$  показан на рисунке 4.2.

$$\delta_{er} = \min(\max((|\mathbf{s}| - s_{er}) \cdot C_{great}, 1) - 1, 1), \quad (4.12)$$

где  $C_{great}$  — большое число.

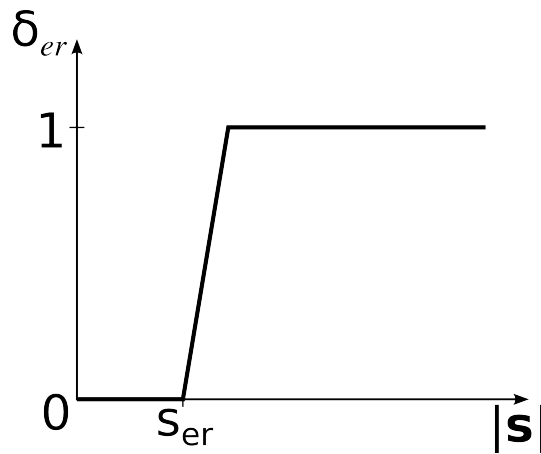


Рисунок 4.2 — Вид коэффициента  $\delta_{er}$ .

#### 4.6 Разработка решателя `avalancheFoam` для расчёта динамики склонового потока с захватом материала на дне в пакете `OpenFOAM`

Для реализации описанного выше фазового перехода была доработана библиотека `interfacialModel`, в неё была включена модель `massTransferModel`, позволяющая использовать различные модели разрушения и захвата материала, а также отложения материала потока. Архитектура полученного решателя с учетом нового модуля приняла вид, показанный на рисунке 4.3.

Подробное описание программной реализации показано в Приложении Б.

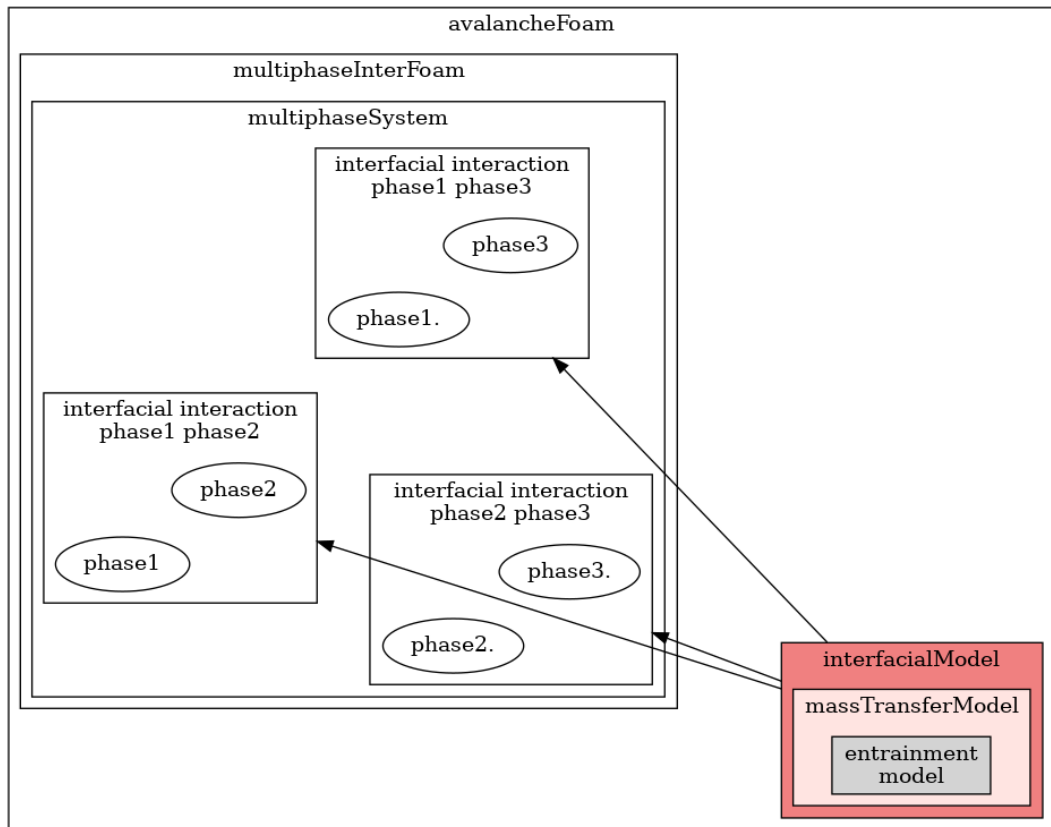


Рисунок 4.3 — Блок-схема архитектуры решателя avalancheFoam.

## 4.7 Примеры применения трехмерной модели с захватом подстилающего материала

### 4.7.1 Модельная задача течения потока на склоне постоянного уклона

Для валидации разработанного решателя avalancheFoam используем численный эксперимент в лотке постоянного угла наклона  $\theta = 30^\circ$ , представленном на рисунке 4.4. Исследуется ламинарный поток неньютоновской жидкости с захватом. Лоток имеет следующие габариты: длина  $L = 1$  м, высота  $H = 0.3$  м, ширина  $B = 0.1$  м. В лоток подаётся жидкость с постоянным расходом: глубина потока на входе  $h = 0.1$  м, скорость на входе  $U = 1$  м/с.

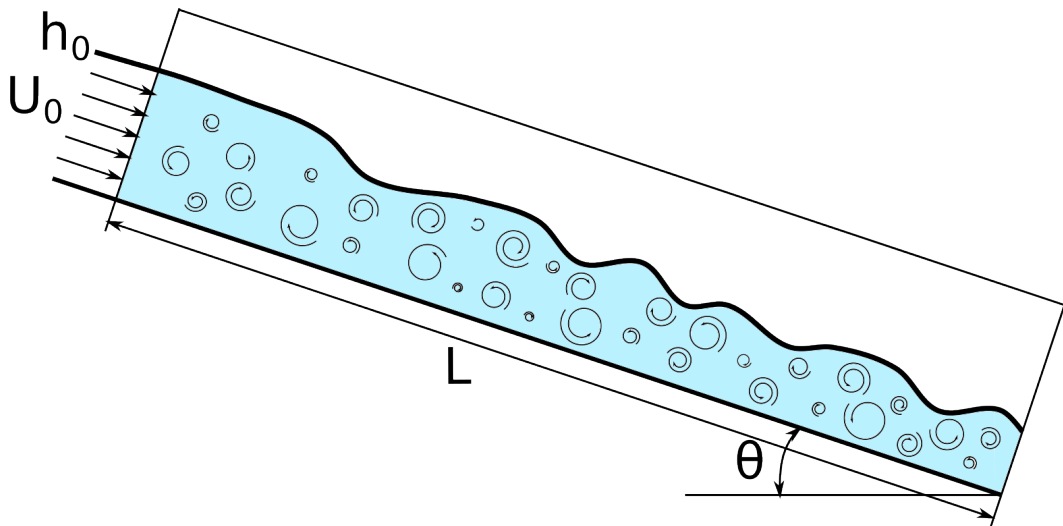


Рисунок 4.4 — Поток в лотке постоянного уклона.

### Начальные условия

На дне лотка лежит неподвижный материал склона (снег) толщиной 0.1 м (изображён горчичным цветом на рисунке 4.5). В начальный момент времени на слое материала склона лежит слой жидкого потока, толщиной 0.1 м, движущийся со скоростью 1 м/с (изображён синим цветом на рисунке 4.5). Бордовым цветом на рисунке 4.5 изображён слой воздуха.



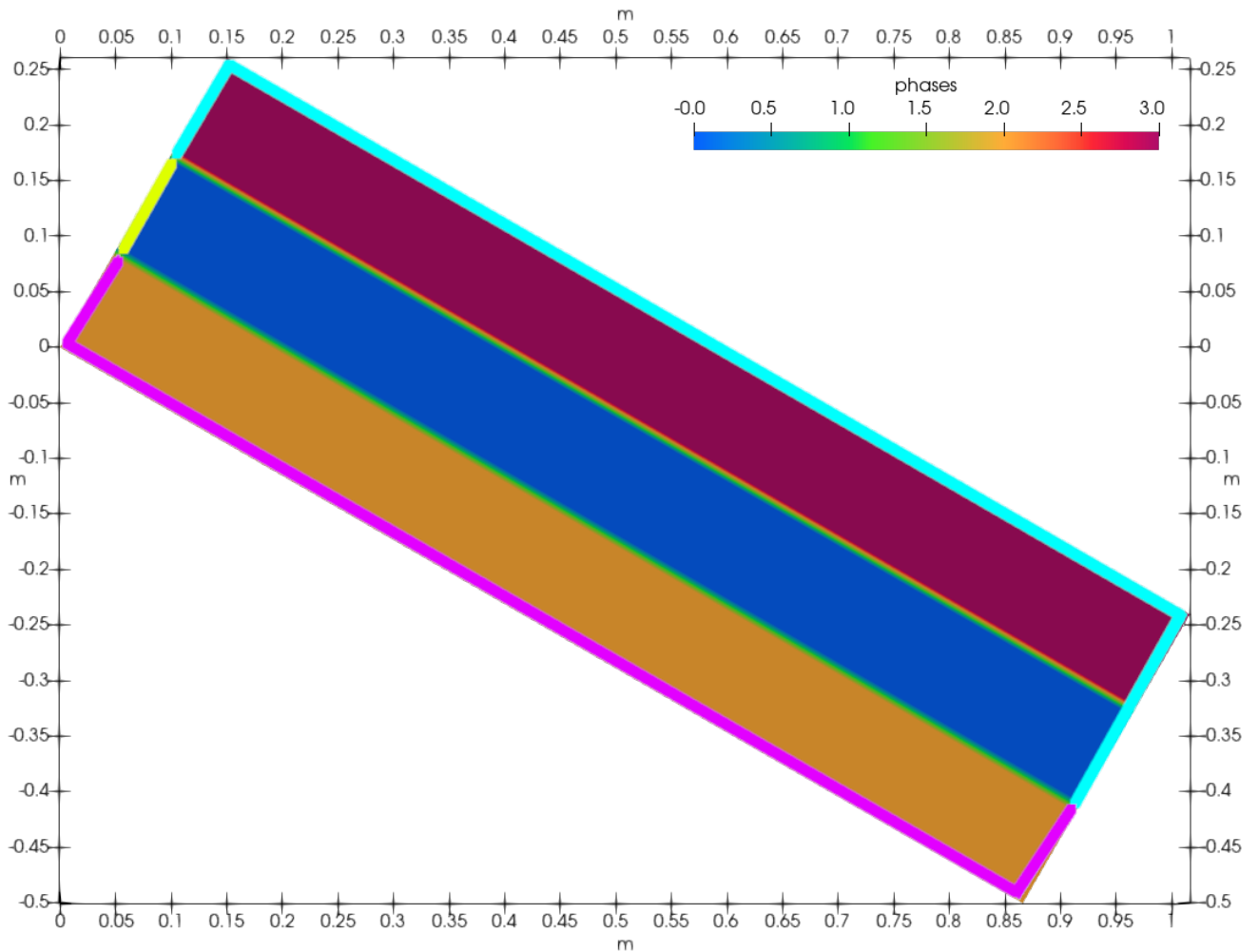


Рисунок 4.5 — Расчётная область численного эксперимента базального захвата в начальный момент времени.

### Граничные условия

Левая граница расчётной области задана стенкой с отверстием в середине, начиная с высоты 0.1 м и заканчивая на высоте 0.2 м (отмечено жёлтым цветом на рисунке 4.5). Через отверстие поступает жидкость со скоростью 1 м/с. Правая сторона расчётной области разделена на 2 части: нижние 0.1 м заданы стенкой (выполнено условие прилипания, для давления — нулевая производная по нормали), остальная часть задана как выходная (обозначена голубым, как и верхняя граница), поток спокойно может вытекать наружу (для скорости задаётся нулевая производная по нормали, давление задано нулём). Стенкой также задана нижняя граница лотка. На рисунке 4.5 малиновым цветом обозначены границы с установленным на них граничным условием стенки. Голубым цветом

на рисунке 4.5 обозначена граница, на которой установлено граничное условие «атмосферы» (выходное, когда поток спокойно вытекает).

## Параметры течения

Данная задача является упрощением задачи захвата покоящегося снега со склона движущейся лавиной. Вследствие этого материал потока, материал склона и захваченный материал представлены реологическими соотношениями Хершеля-Балкли со схожими свойствами. Предел текучести у движущейся массы снега отличается от предела текучести снега, покоящегося на склоне, в силу наличия в последнем дополнительных кристаллических связей, разрушаемых с началом движения, как было описано в разделе 4.1. Реологические параметры снега, приведенные в таблице 5, берутся близкие к общепринятым [45].

Таблица 5 — Реологические свойства в задаче захвата материала потоком в лотке постоянного уклона

	air	flow	entrained	slope
плотность	$\rho_g = 1.3 \text{ кг/м}^3$	$\rho_{snow} = 300 \text{ кг/м}^3$		
вязкость	Ньютоновская среда $\mu_g = 1.7 \cdot 10^{-5} \text{ Па}\cdot\text{с}$	Среда Хершеля-Балкли если $ \mathbf{s}  < s_0$ : $\mu_i = \mu_{i_0}$ , если $ \mathbf{s}  \geq s_0$ : $\mu_i = \frac{\tau_{i_0} + \kappa_i  \mathbf{s} ^n}{ \mathbf{s} }$ .		
$\tau_0$		$10^3 \text{ Па}$	$10^5 \text{ Па}$	
$\kappa$		$0.1 \text{ Па}\cdot\text{с}^n$		
$n$		2		

## Расчёт захвата подстилающего материала

Был произведен расчёт базального захвата материала в лотке постоянного угла наклона  $30^\circ$ .

Жидкость подавалась со скоростью 1 м/с.

Жидкость потока обладала следующими реологическими свойствами: плотность  $\rho = 1000 \text{ кг/м}^3$ , коэффициент кинематической «недеформируемой» вязкости  $\nu_0 = 0.1 \text{ м}^2/\text{с}$ , предел текучести, отнесённый к плотности смеси  $\tau_0 = 0.05 \text{ м}^2/\text{с}^2$ , коэффициент консистенции, отнесённый к плотности смеси  $\kappa = 10^{-4} \text{ м}^2\text{с}^{n-2}$ , степенной индекс  $n = 2$ .

На рисунке 4.6 показано течение в экспериментальном лотке спустя 10 секунд после начала захвата. Горчичным цветом на рисунке 4.6 изображён слой размываемого склонового материала. Синим цветом на рисунке 4.6 изображён слой движущегося потока жидкости. Зелёным цветом на границе материала склона и материала жидкости изображён слой захваченного материала. Бардовым цветом на рисунке 4.6 изображён слой воздуха.

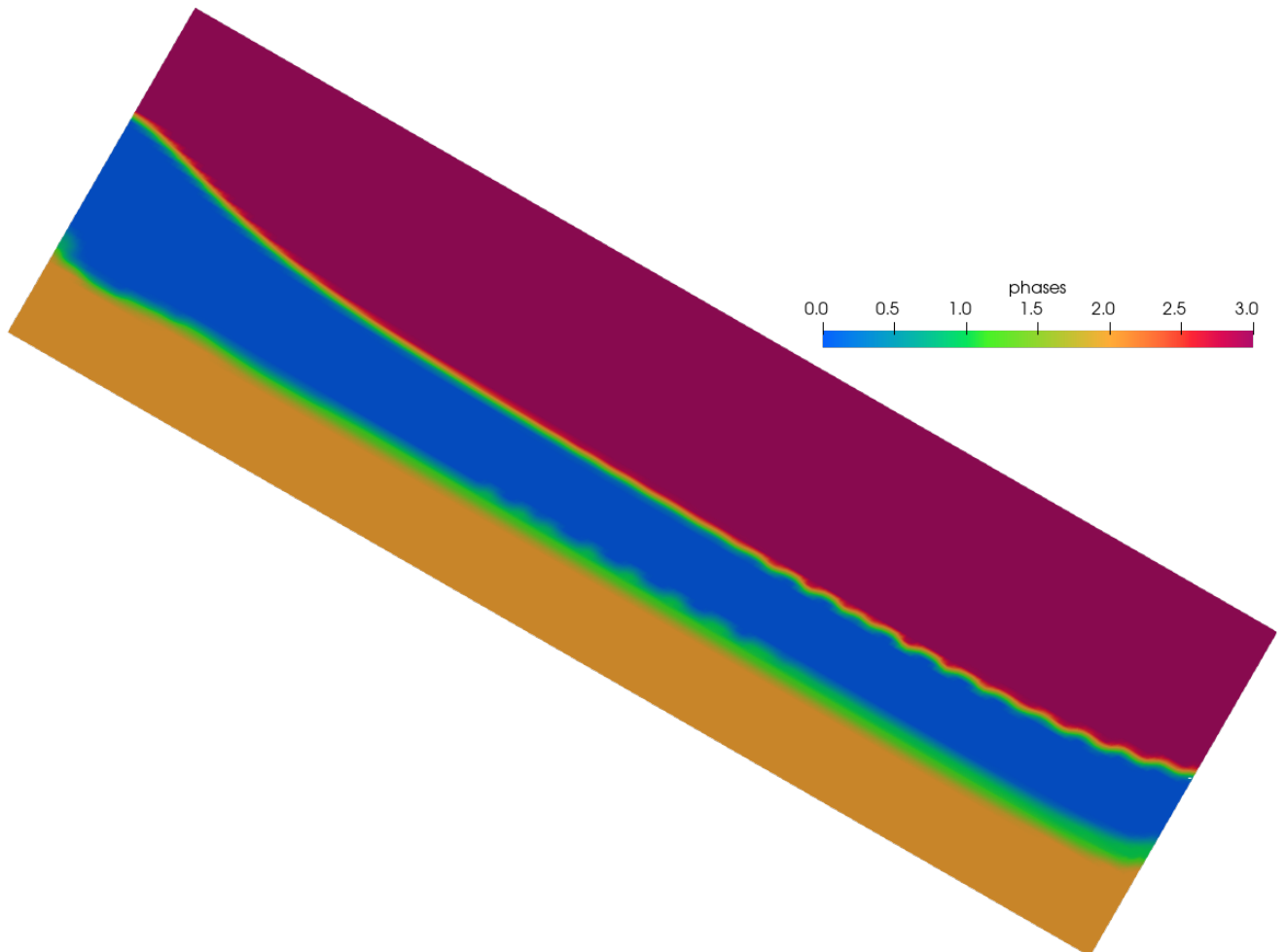


Рисунок 4.6 — Расчётная область численного эксперимента базального захвата спустя 10 с после начала расчёта.

Процесс захвата происходит таким образом, что материал потока разрушает материал склона, после чего разрушенный материал переходит в новую

фазу — «захваченного» материала. Захваченный материал в настоящем расчёте имеет реологические свойства движущегося материала потока.

На рисунке 4.6 можно видеть, что слой захваченного материала утолщается слева направо, что объясняется тем, что разрушение склона происходит по всей границе между материалом склона и материалом потока, и после разрушения этот материал уносится потоком вправо. Таким образом толщина слоя «захваченного» материала должна увеличиваться справа.

Так же на рисунке видно образование волн, что связано с геометрией задачи. Для фиксации области подачи жидкости необходимо было установить стенку в нижней части левой границы расчётной области. Из-за этой стенки образуется горка из покоящегося снега в начале лотка, скатываясь на которой поток отпрыгивает и образуются волны на поверхности разрушаемого материала а в связи с ними и волны на поверхности потока.

На рисунке 4.7 показан поток через 11 с после начала расчёта. Проведено исследование состояния границ склон–захваченный материал, захваченный материал–поток, поток–воздух в расчёте на равномерной сетке  $200 \times 60$  ячеек. Можем видеть, что граница чёткая, располагается в пределах одного слоя ячеек. Общая картина показывает все фазы одновременно, вставки в центре показывают красным лишь одну фазу (воздух, поток, захваченный материал, материал склона).

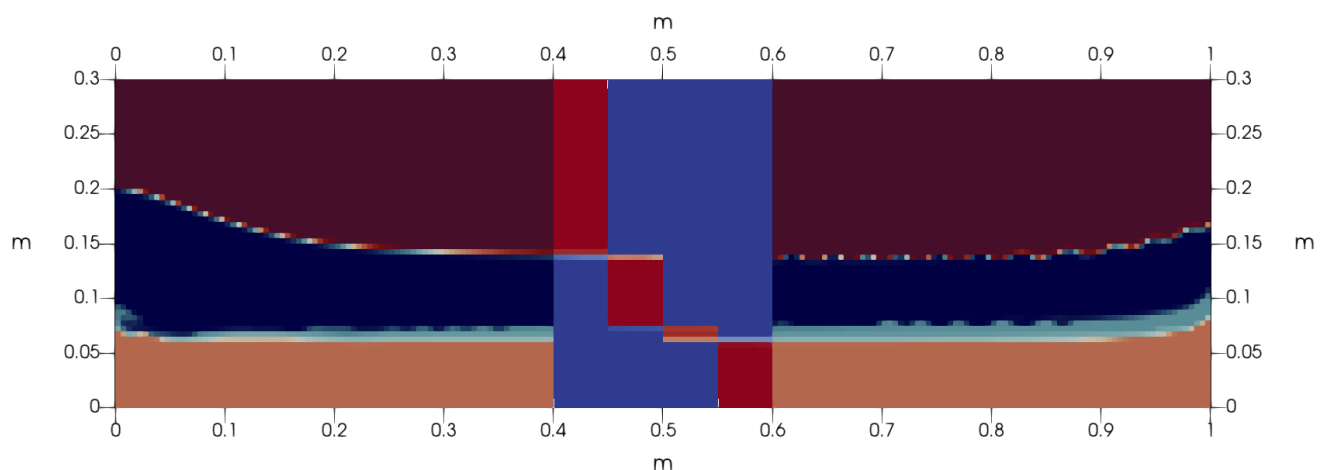


Рисунок 4.7 — Исследование размытия границ склон–захваченный материал, захваченный материал–поток, поток–воздух.

Было проведено исследование сеточной сходимости в задаче захвата подстилающего материала потоком на модельном склоне постоянного угла наклона при постоянном расходе потока.

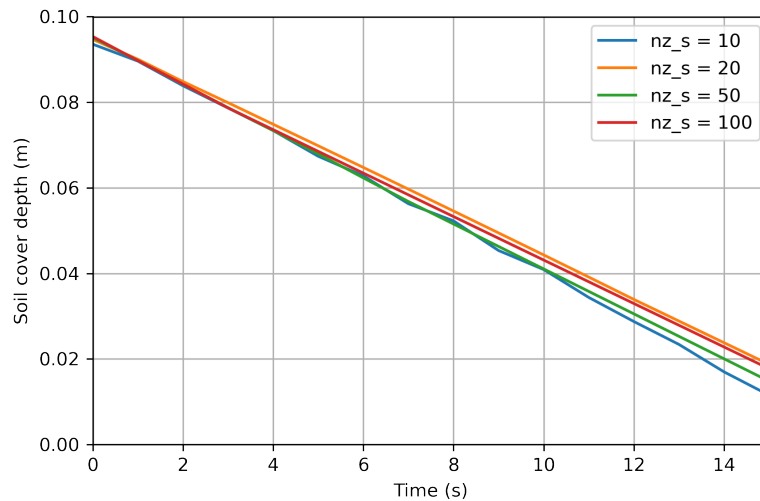


Рисунок 4.8 — Исследование сеточной сходимости при различном количестве слоёв ячеек по глубине  $nz_s$  в материале склона.

Результаты расчёта базального захвата с различным количеством слоёв ячеек по глубине  $nz_s$  в материале склона: 10 слоёв ячеек на слой подстилающего материала, 20 слоёв ячеек, 50 и 100. Можно видеть, что при слишком крупном разбиении (10) захват происходит неравномерно, а при более мелком разбиении захват равномерный. При всех масштабах разбиения разрушение подстилающего материала происходит приблизительно одинаково. Решение сходится по сетке.

### Влияние предела прочности материала склона на процесс захвата

Было проведено исследование влияния предела прочности материала склона, на процесс захвата. Предел прочности в настоящей модели задается пределом скорости сдвига. Проведено сравнение 4.9 результатов расчёта базального захвата с различными значениями предела прочности (breaking point)  $BP = 0.5, 1.0, 2.0, 50.0$  1/с.

На рисунке 4.9 можем видеть, что чем меньше предел прочности материала склона тем быстрее происходит разрушение слоя и унос материала. Скорость разрушения слоя постоянна, что совпадает с результатами, полученными Эглит (на бесконечности скорость захвата материала постоянная) [51].

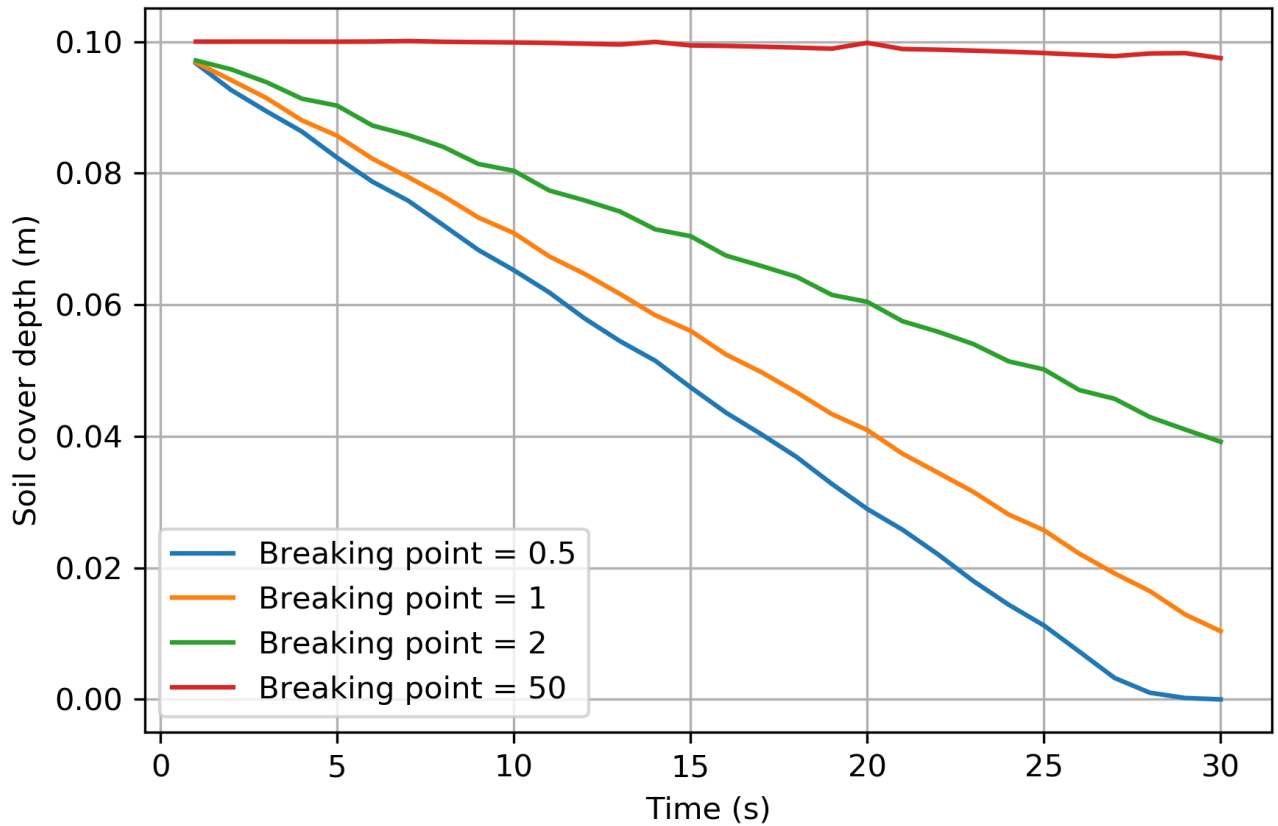


Рисунок 4.9 — Изменение глубины слоя подстилающего материала со временем.

Для значений  $BP = 0.5, 1.0, 2.0$  1/с происходит разрушение подстилающего слоя тем быстрее, чем меньше предел прочности.

Для значения  $BP = 50.0$  1/с также есть незначительное размывание склона, однако оно уже связано с переносом фазы материала склона, а не её разрушением. Данный перенос возникает вследствие того, что условие неподвижности материала склона задано не строго. Так как расчётная ячейка может быть заполнена не только материалом склона, но и другими фазами потока, чья скорость не обнуляется. В силу вычислительных особенностей в ячейке всегда присутствует как минимум очень маленькая доля всех фаз, поэтому было задано условие, что скорость в ячейке зануляется пропорционально объёмной доле материала склона в ней. В связи с этим происходит некоторое размывание материала склона. Влияние данного размывания на исследуемый класс задач незначителен и им можно пренебречь.

## Влияние реологических свойств потока на процесс захвата

Для среды Хершеля-Балкли проведено исследование влияния коэффициента консистенции на процесс захвата материала склона [4.10](#)

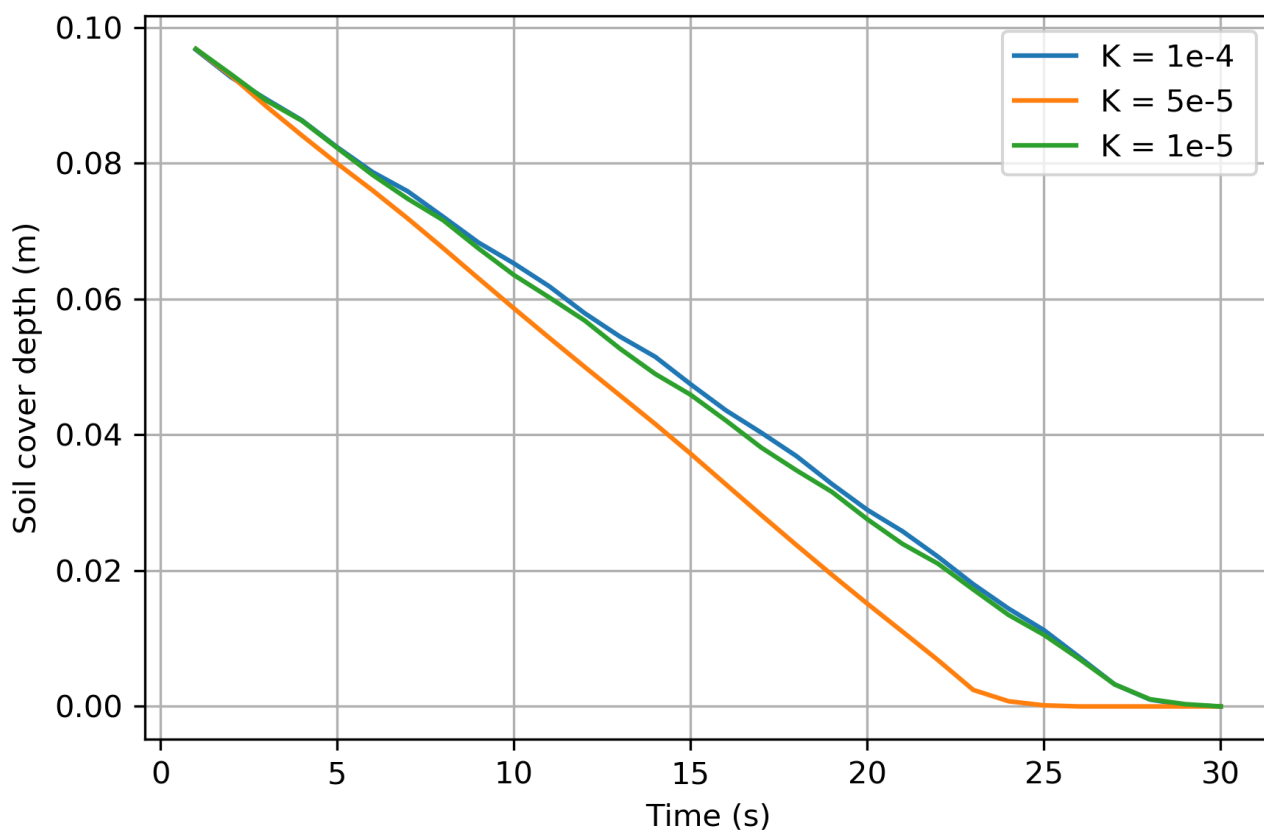


Рисунок 4.10 — Изменение глубины слоя подстилающего материала со временем для потоков жидкостей с различным коэффициентом консистенции.

На рисунке [4.10](#) можем видеть нелинейное изменение скорости разрушения подстилающего материала в зависимости от коэффициента консистенции.

Для среды Хершеля-Балкли проведено исследование влияния степенного индекса на процесс захвата материала склона [4.11](#)

На рисунке [4.11](#) можем видеть, что поток со степенным индексом 1 менее всех разрушает подстилающий материал. При увеличении степенного индекса скорость разрушения подстилающего материала увеличивается. При значении степенного индекса менее 1 скорость разрушения подстилающего материала увеличивается до некоторого предела и далее уже не изменяется.

Для среды Хершеля-Балкли проведено исследование влияния предела текучести на процесс захвата материала склона [4.12](#)

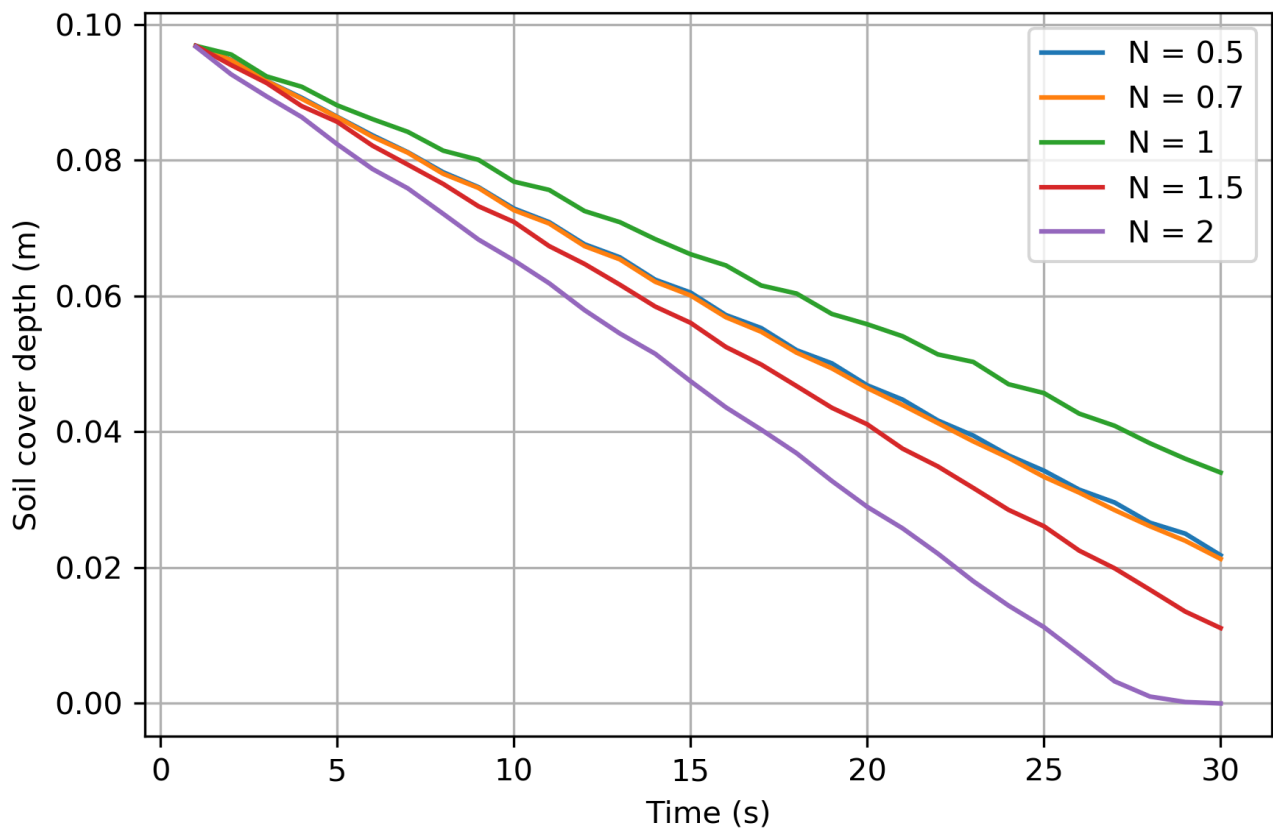


Рисунок 4.11 — Изменение глубины слоя подстилающего материала со временем для потоков жидкостей с различным степенным индексом.

На рисунке 4.12 можно видеть, что предел текучести менее  $0.01 \text{ м}^2/\text{с}^2$  не влияет на скорость разрушения подстилающего материала. При значении предела текучести выше  $0.01 \text{ м}^2/\text{с}^2$ , он линейно влияет на скорость разрушения подстилающего материала.



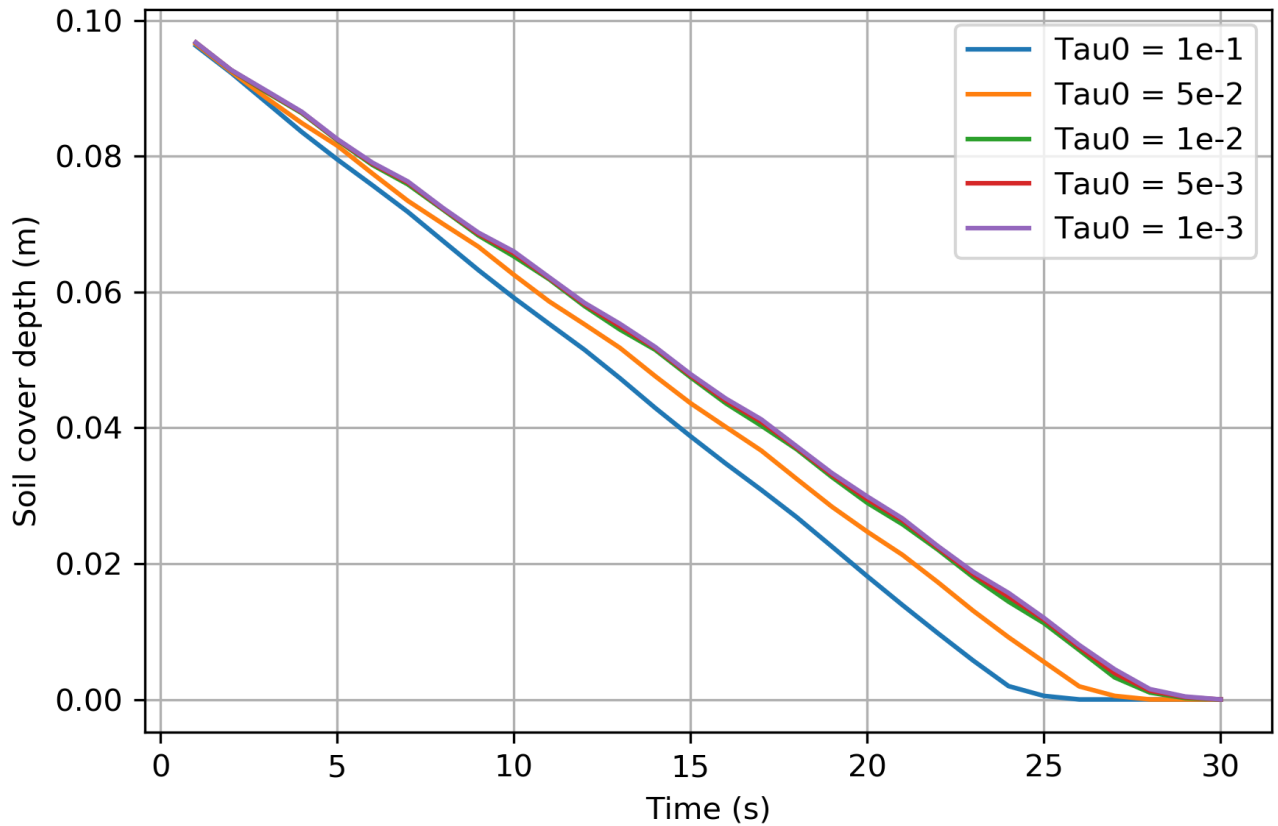


Рисунок 4.12 — Изменение глубины слоя подстилающего материала со временем для потоков жидкостей с различным пределом текучести.

#### 4.7.2 Прорыв ледникового озера Малый Азау

Одним из исследуемых объектов является прорыв ледникового озера Малый Азау на горе Эльбрус (Центральный Кавказ) [131]. Группа озёр, расположенная около ледника Малый Азау на южном склоне Эльбруса, широко известна и ежегодно посещается туристами и альпинистами. В период 1957–2021 гг. зафиксировано два случая прорывов озёр: озеро западное в 1978 г. и озеро восточное в 2011 г. Первый прорыв связан с процессами оползневых деформаций моренного массива, слагающего часть озёрной котловины, второй — с деградацией ледяной плотины озера и переливом воды поверх неё.

На рисунке 4.13 показаны карты разных годов. На рисунках 4.14, 4.15, 4.16 показаны топографические карты разных масштабов, а также снимок со спутника. На рисунках зеленым цветом схематически обозначена граница исследуемой области.

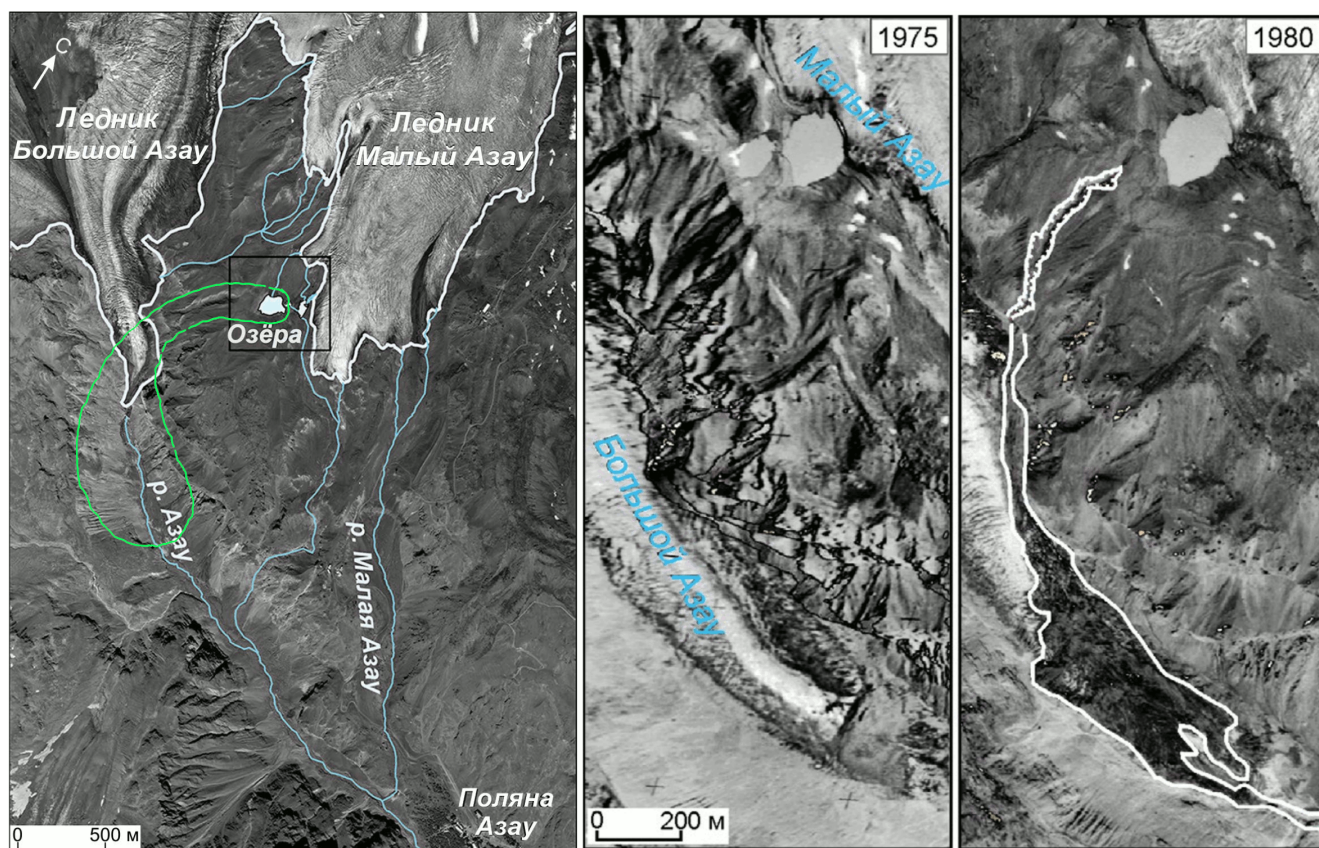


Рисунок 4.13 — Карта рельефа исследуемой местности 2016 года с схематически обозначенным зелёным цветом контуром расчётной области (слева); снимок исследуемых озёр 1975 года (центр); следы прорыва западного озера, обозначенные белым цветом (справа) [131].

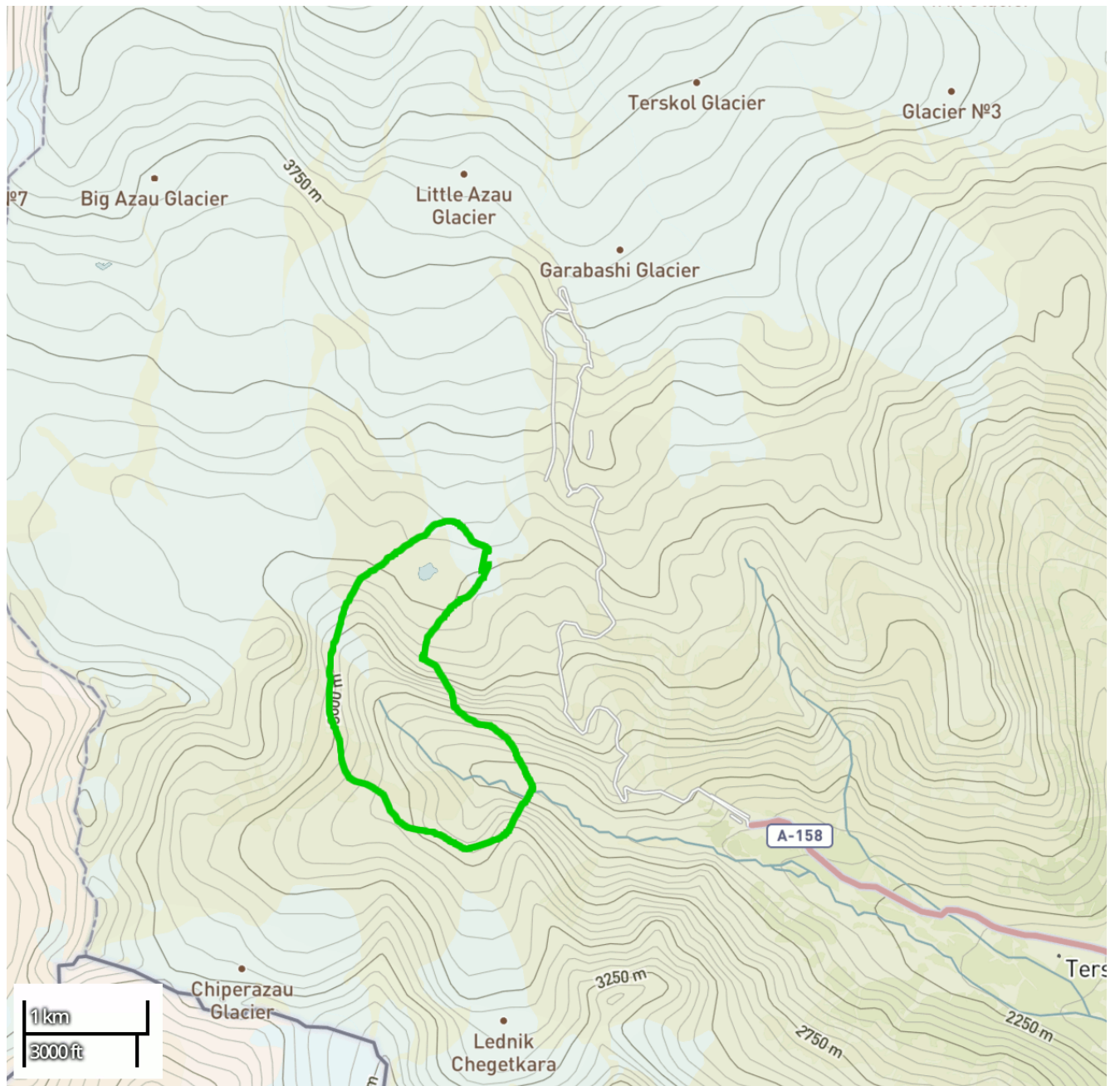


Рисунок 4.14 — Карта рельефа исследуемой местности с схематически обозначенным зелёным цветом контуром расчётной области [132].



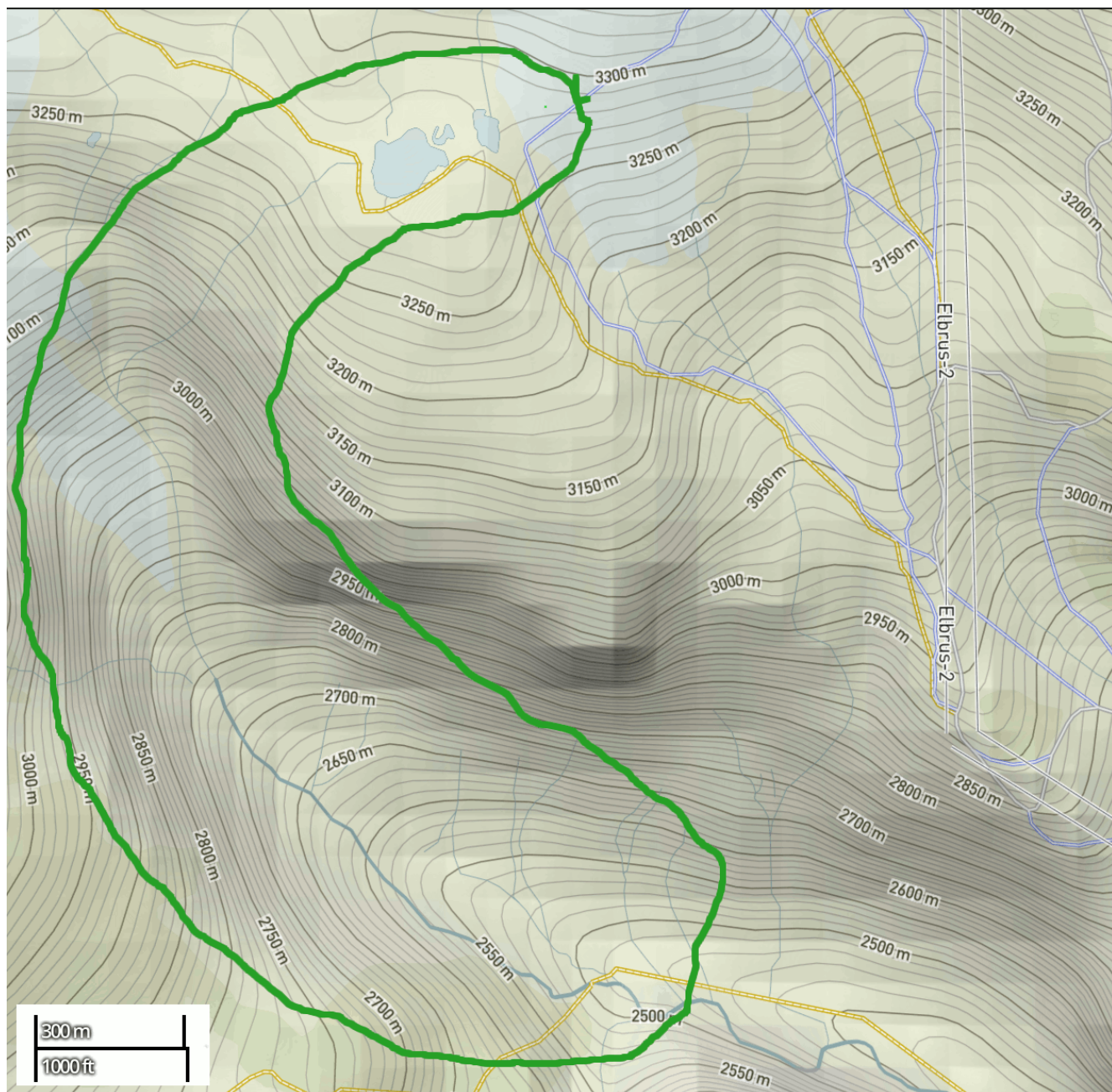


Рисунок 4.15 — Карта рельефа исследуемой местности с схематически обозначенным зелёным цветом контуром расчётной области [132].



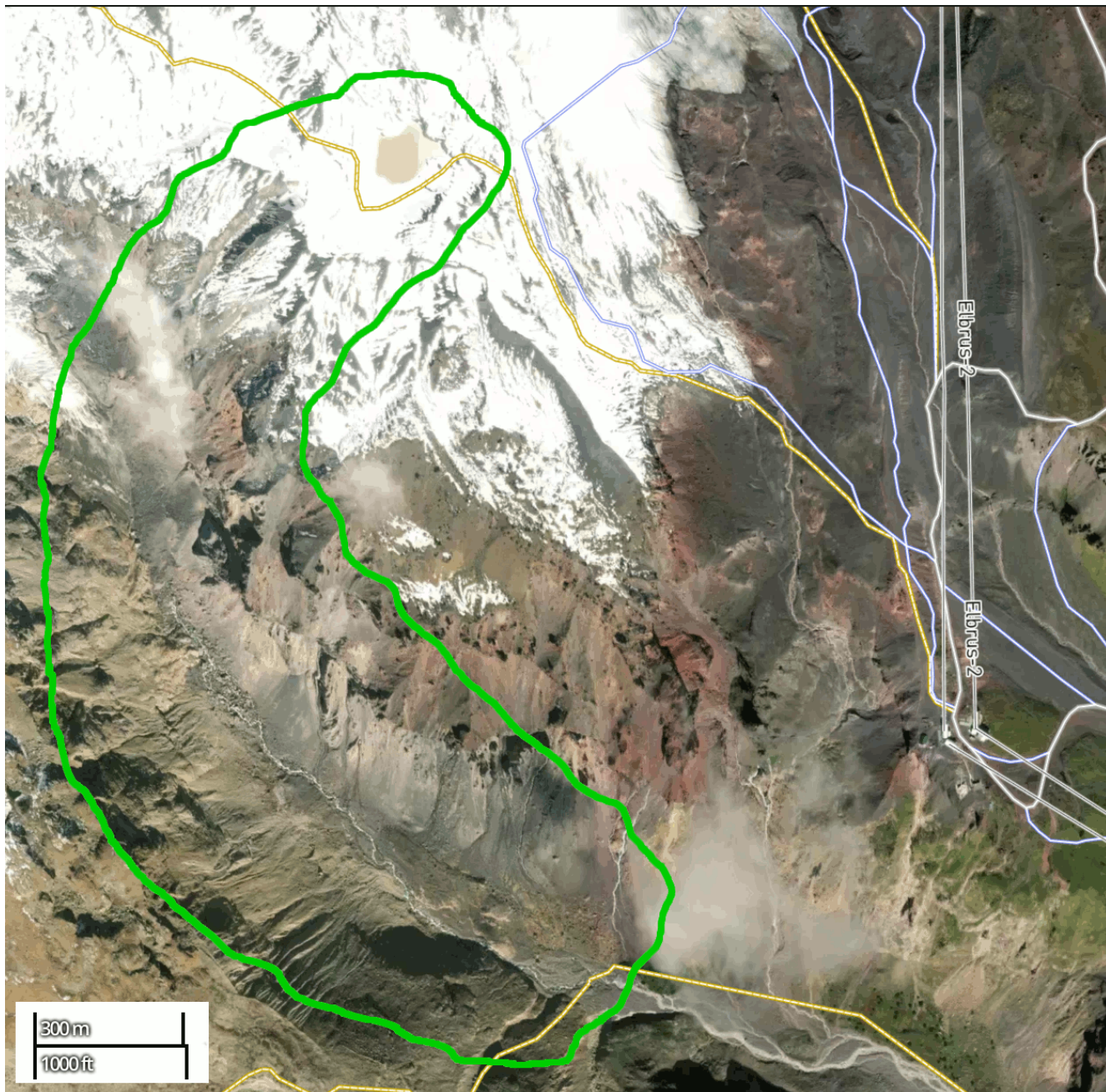


Рисунок 4.16 — Карта рельефа исследуемой местности с схематически обозначенным зелёным цветом контуром расчётной области [132].

## Параметры течения

Рассматривался прорыв озера в период поздней весны, когда происходит активное таяние снежных покровов, и снег ещё присутствует на склоне. Материал склона (stationary) — мокрый снег. Реологические параметры снега, а также воды, воздуха и захваченного материала представлены в таблице 6.

Таблица 6 — Реологические параметры сред в задаче расчета водоснежного потока при прорыве ледникового озера Малый Азау.

air	flow	stationary	entrained
$\rho_a = 1.3 \text{ кг/м}^3$	$\rho_f = 1000 \text{ кг/м}^3$	$\rho_s = 800 \text{ кг/м}^3$	$\rho_e = 800 \text{ кг/м}^3$
Ньютоновская среда		Среда Хершеля-Балкли	
$\mu_a = 1.7 \cdot 10^{-5} \text{ Па}\cdot\text{с}$	$\mu_f = 1.8 \cdot 10^{-3} \text{ Па}\cdot\text{с}$	$\tau_{s_0} = 10 \cdot 10^4 \text{ Па}$ $\kappa_s = 0.1 \text{ Па}\cdot\text{с}^n$ $n_s = 2$	$\tau_{e_0} = 10 \cdot 10^3 \text{ Па}$ $\kappa_e = 0.1 \text{ Па}\cdot\text{с}^n$ $n_e = 2$

На склоне лежит слой снега глубиной 1 метр. Объем воды в озере составляет 66.7 тыс. м<sup>3</sup> [131].

## Результаты расчёта

По результатам моделирования удалось построить карту эрозии слоя снега 4.17. Красным цветом на рисунке обозначено разрушение слоя снега на глубину 1 м. Во второй половине пути потока можно видеть отклонение потока от русла реки в сторону линии наибольшего спуска. Следовательно, можно сделать вывод, что при прорыве ледникового озера Малый Азау изменится русло реки Азау и будет разрушен левый берег.



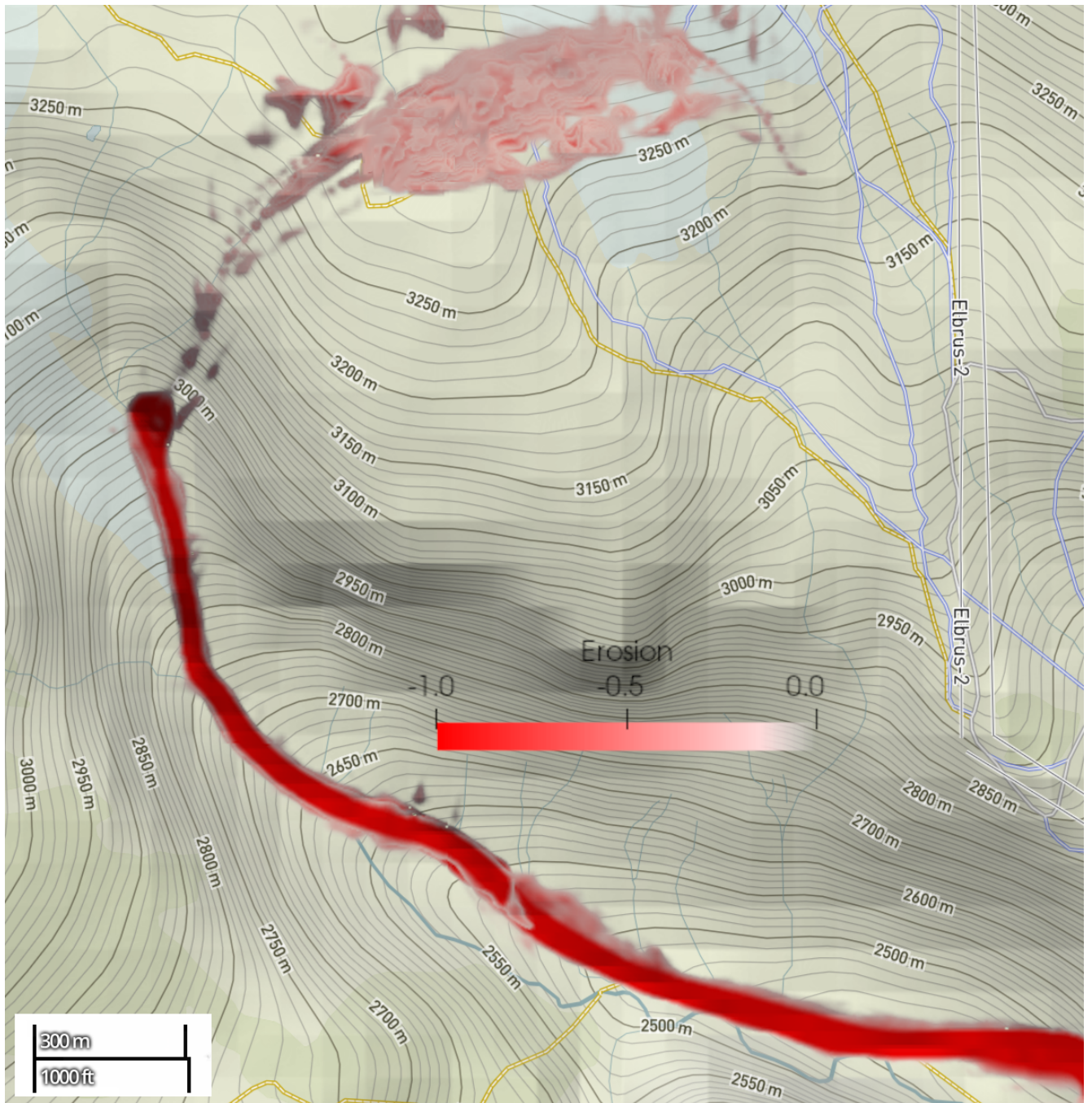


Рисунок 4.17 — Карта эрозии слоя снега водоснежным потоком при прорыве ледникового озера Малый Азау.

#### 4.8 Выводы

В настоящей работе была разработана модель захвата подстилающего материала, которая позволяет учесть все сложные аспекты моделирования потоков на склонах гор. Модель учитывает

- сложный рельеф местности,
- трёхмерную структуру,

- взаимодействие с препятствиями,
- турбулентный режим течения,
- многофазность,
- нелинейную реологию,
- разрушение и захват материала склона.

Разработанный решатель avalancheFoam был использован для расчёта базального захвата в лотке постоянного угла наклона. В данном эксперименте решатель продемонстрировал постоянную скорость разрушения склона. Этот результат соответствует данным, полученным в [51; 80], при использовании гипотезы превышения порогового значения напряжениями сдвига в материале склона.

С использованием разработанного решателя avalancheFoam был произведен расчёт возможного прорыва ледникового озера (ПЛО) вблизи ледника Малый Азау. По результатам расчёта была построена карта эрозии склона. В результате расчёта было предсказано разрушение левого берега реки Азау и изменение русла реки с отклонением влево на величину до 50 метров.

При расчёте ПЛО Малый Азау «захваченный» материал отличался реологическими свойствами от материала склона и от материала потока. Материалом склона был рассмотрен весенний мокрый снег. При вовлечении в движение в снежной толще разрушаются кристаллические связи, изменяется плотность и вязкость снежной массы. В связи с этим можно видеть границу резкого увеличения разрушения материала склона. Это связано с накоплением в потоке массы «захваченного» материала и значительного увеличения эффективной вязкости потока. Это увеличило напряжения сдвига, в результате чего скорость эрозии склона увеличилась.

Все вышеперечисленные результаты говорят о высоком качестве разработанного ПО, которое удовлетворяет всем заявленным особенностям.

В работе было проведено моделирование потока в результате прорыва ледникового озера Малый Азау (гора Эльбрус, Центральный Кавказ) в период поздней весны (при наличии слоя снега на склоне) с учетом реального рельефа и разрушения подстилающего материала. За последние годы было два реальных прорыва озера, но, к сожалению, замеры пути и параметров течения не проводились. Поскольку окрестности озера очень популярны у туристов и альпинистов, важна информация о параметрах возможных прорывных паводков. Результаты в этом разделе предоставляют некоторую информацию.



Результаты расчёта показали, что

- в верхней части течения, где угол наклона склона большой, захват снега прорывным паводком происходит незначительно;
- на высоте порядка 3000 метров происходит уменьшение уклона и поток резко врежется в склон, вследствие чего сдирает слой снега на всю глубину;
- в некоторых местах сбоку от потока можно видеть участки, где снег потерял устойчивость и произошло обрушение;
- в процессе движения поток отклоняется от русла реки влево и полностью сносит слой снега, что говорит о возможном разрушении материала склона и изменении русла реки в случае ПЛО Малый Азау.

## Глава 5. Оптимизация коэффициентов турбулентных моделей с использованием методов машинного обучения<sup>1</sup>

Как было показано в разделе 3.5.2 результаты моделирования не очень хорошо повторяют данные эксперимента, одной из причин этого может быть неподходящие значения коэффициентов турбулентной модели. Поток, рассматриваемый поток в лотке Университета Исландии турбулентный, число Рейнольдса для него составляет  $Re = 10^5$ . Также стоит отметить, что из-за сложной формы подстилающей поверхности (составной желоб нескольких углов наклона) граница потока имеет сложную форму со множеством самопересечений (в потоке большое количество брызг). Из-за сложной формы границы потока используется метод объёмной доли жидкости лоя моделирования динамики потока. Однако, при осреднении по Рейнольдсу предложенной в разделе 2 математической модели помимо напряжений Рейнольдса появляются дополнительные члены, связанные с неконстантными значениями плотности и вязкости потока (они являются функциями объёмной доли фазы). В случае, когда граница потока простая (например прямая линия) количество ячеек содержащих смесь фаз не велико и тогда этими дополнительными членами, возникающими при осреднении, можно пренебречь. В ячейках заполненных полностью одной фазой, эти члены равны нулю. Однако в потоке, с большим количеством брызг вышеупомянутыми членами пренебрегать нельзя, так как большое количество ячеек содержит смесь фаз.

Произведём обзор работ, посвященных калибровке турбулентных моделей.

Турбулентные потоки встречаются в мире повсеместно и разработано уже большое количество различных турбулентных моделей с различным количеством параметров. Если некоторые из моделей хорошо подходят для довольно узкого класса течений, то другие, наоборот, претендуют на универсальность. Такие модели турбулентности, как  $k - \varepsilon$ ,  $k - \omega$ ,  $k - \omega SST$  имеют наиболее широкое применение, и чаще всего используются в расчётах. Однако и они имеют ограничения в области применимости, в частности, это связано с калибровкой

<sup>1</sup>При подготовке данной главы диссертации использовались следующие публикации автора, в которых, согласно «Положению о присуждении ученых степеней в Московском государственном университете имени М.В.Ломоносова», отражены основные результаты, положения и выводы исследования: [4; 5; 10].

констант, содержащихся в данных моделях. Так, например, константы  $k - \varepsilon$  турбулентной модели калибровались на следующих течениях [93; 133]: плоская струя в движущемся потоке, обтекание пристенной струёй конусов различного угла раскрытия, потоки в трубах, пограничный слой на лопатках турбин, плёночное охлаждение, коаксиальные струи, течение в каверне, течение в трубе со вставкой в виде скрученной ленты, потоки в воздуховоде квадратного сечения. Калибровка  $k - \omega$  [110; 134] и  $k - \omega SST$  [106; 107; 135] моделей в основном проводилась для таких задач, как обтекание потоком воздуха различных профилей, в том числе, с учётом сжимаемости потока.

В более поздних работах всё чаще константы турбулентных моделей заменяются на функции и используемые зависимости всё более обобщаются. Так в работе [136] при моделировании обтекания потоком со свободной поверхностью профиля константа  $C_\mu$  представлена функцией средних тензора скоростей деформаций, тензора завихренности и других параметров потока, однако, форма этой зависимости в статье не приводится.

Исследование применимости и точности моделирования потоков со свободной поверхностью с помощью различных турбулентных моделей пакета OpenFOAM проделано в работе [137]. В работе результаты численного моделирования сравнивались с результатами эксперимента двойного разрыва дамбы (исследовалось смешение двух разных жидкостей).

Сравнение турбулентных моделей пакета OpenFOAM также проводилось для обтекания потоком жидкости куба, вмонтированного в стену [138].

В работе [139] к  $k - \omega$  турбулентной модели добавлен один новый замыкающий член, значительно улучшающий точность моделирования для свободных сдвиговых течений.

Улучшение  $k - \omega SST$  модели турбулентности проводится в работе [140] для лучшего воспроизведения сдвигового потока при обтекании тонкой пластины.

Калибровка  $k - \omega SST$  модели проводится в работе [141; 142] для малогабаритных ветровых турбин.

Разработка улучшенных  $k - \omega$  и  $k - \omega SST$  моделей в пакете OpenFOAM для моделирования задачи разрушения волны проводилось в работе [143]. Также модифицированная  $k - \omega SST$  модель была применена для расчёта волны при набегании потока на препятствие [144].

Исследование точности предсказания сдвигового потока в водосбросе различными турбулентными моделями с помощью пакетов OpenFOAM и FLOW-3D представлено в работе [145].

### 5.1 Экспериментальные данные полученные в НИИ Механики МГУ

Автором и инициативной группой, состоящей из сотрудников Института механики МГУ (Олег Иванов, Владимир Трифонов), студентов географического факультета МГУ (Ника Гинзбург, Дарья Коровина), выпускника физического факультета МГУ (Борис Гинзбург), был поставлен эксперимент в НИИ Механики МГУ для калибровки коэффициентов турбулентных моделей. В процессе эксперимента в лотке постоянного уклона (эксперимент проведён для нескольких разных углов наклона) производился спуск турбулентного потока жидкости. Использовался лоток следующей геометрии: длина — 1 м, ширина — 12 см, высота бортиков — 10 см. В расчётах использовалась область лотка между двумя точками замера скорости и глубины, находящимися на расстоянии 23 см и 82 см от верхнего края лотка. Схема эксперимента показана на рисунке 5.1. Для проведения эксперимента используется водопроводная вода, как в эксперименте Университета Исландии. Лоток выполнен полностью из акрилового стекла толщиной 4 мм. Вода подаётся из резервуара с возможностью регулировки высоты столба жидкости при подаче. При входе в исследуемый участок лотка установлен успокоитель потока. Эксперимент проводится в стационарном режиме с замкнутым жидкостным контуром. Стационарность обеспечивает насос перекачки жидкости из нижнего водосборника в верхний резервуар. Стационарность обеспечивал погружной насос производительностью 200 л в минуту, который перекачивал жидкость из нижнего водосборника в верхний резервуар. Измерение скорости проводилось с помощью трубки Пито, соединённой с датчиком давления, работающим с погрешностью 0,1 %. Точки измерения располагались через каждые 0.5 мм. Для каждого измерения использовалось среднее значение за 10 с. Температура воды в эксперименте составляла 20 градусов Цельсия (комнатная температура). Все коэффициенты модели, связанные с температурой, были выбраны на основе этого значения.

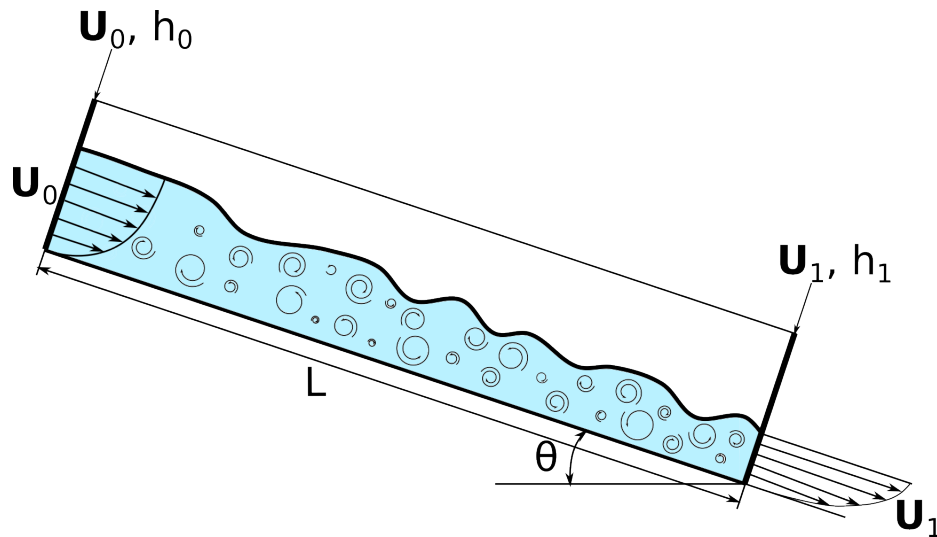


Рисунок 5.1 — Схема экспериментального лотка.

На рисунке 5.1 показана область между двумя точками замера. На первой точке замера снимается глубина потока  $h_0$  и профиль скорости потока вдоль нормали ко дну  $U_0$ . На второй точке снимаются аналогичные величины  $h_1$  и  $U_1$ .

Было проведено 3 серии экспериментов, в которых менялся угол наклона склона, начальная глубина потока, скорость потока, как показано в таблице 7

Таблица 7 — Параметры экспериментов

$\tilde{U}_0$ среднее по глубине, м/с	$h_0$ , мм	$\theta$
1.63	4.20	25°
2.00	4.95	28°
1.78	3.45	33°

## 5.2 Алгоритм оптимизации коэффициентов моделей турбулентности

$k$ - $\varepsilon$  модель турбулентности содержит ряд констант  $C_\mu$ ,  $C_1$ ,  $C_2$ ,  $C_3$ ,  $\sigma_k$ ,  $\sigma_\varepsilon$ .  $k$ - $\omega$  SST модель турбулентности содержит ряд констант:  $\alpha_{k1}$ ,  $\alpha_{k2}$ ,  $\alpha_{\omega1}$ ,  $\alpha_{\omega2}$ ,  $\beta_1$ ,  $\beta_2$ . Использование данных турбулентных моделей для расчёта потоков неньютоновских жидкостей на склонах требует калибровки коэффициентов модели.

Для организации процесса калибровки используются алгоритмы машинного обучения.

Калибровка турбулентных моделей будет проводиться с использованием следующего алгоритма, показанного на рисунке 5.2 для  $k$ - $\epsilon$  модели турбулентности.

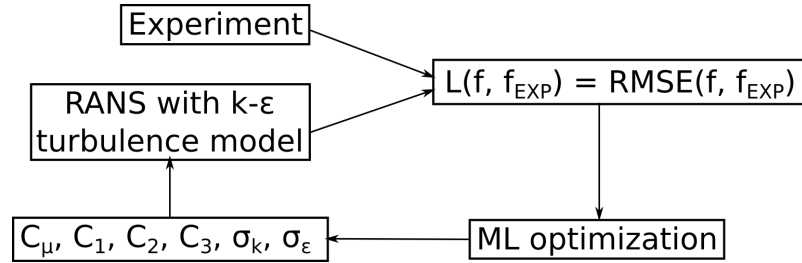


Рисунок 5.2 — Архитектура предлагаемого алгоритма для калибровки коэффициентов  $k$ - $\epsilon$  модели турбулентности

На рисунке 5.3 представлен алгоритм для калибровки коэффициентов  $k$ - $\omega$   $SST$  модели турбулентности.

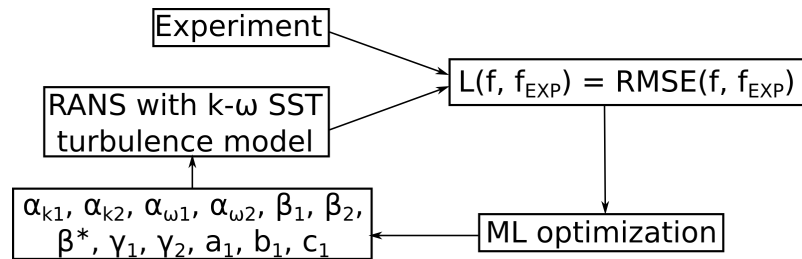


Рисунок 5.3 — Архитектура предлагаемого алгоритма для калибровки коэффициентов  $k$ - $\omega$   $SST$  модели турбулентности

Процедура оптимизации включает в себя следующие шаги:

1. Обучение на базе ряда расчётов проведённых с использованием RANS модели с  $k$ - $\epsilon$  или  $k$ - $\omega$   $SST$  моделями турбулентности с различными значениями констант;
2. Получение новых значений коэффициентов турбулентной модели посредством алгоритма оптимизации;
3. Расчёт гидродинамики потока с использованием турбулентной модели с коэффициентами, полученными с помощью алгоритма оптимизации;
4. Дообучение алгоритма с использованием полученных данных расчёта гидродинамики потока.

При оптимизации коэффициентов турбулентной модели минимизируется корень из среднеквадратического отклонения вычисленного профиля скорости потока на выходной плоскости от экспериментального профиля (5.1):

$$\mathbf{L}_{RMSE} = \sqrt{\frac{\int_{z=0}^{h_1} (|\mathbf{U}_{exp}|(z) - |\mathbf{U}|_{calc}(z))^2}{h_1}}, \quad (5.1)$$

где  $h_1$  — глубина потока на выходной плоскости.

Для минимизации используется метод Нелдера–Мида (метод безусловной оптимизации функции от нескольких переменных, не использующий градиентов), реализованный в библиотеке SciPy — основанной на Python экосистеме программного обеспечения с открытым исходным кодом.

### 5.3 Разработка модуля автоматической оптимизации MLTFoamOptimization

Для реализации совместной автоматической работы метода оптимизации и вычислительного гидродинамического моделирования на языке Python был разработан модуль **MLTFoamOptimization** с использованием библиотек Pandas, NumPy, SciPy, Multiprocessing, Subprocess, PyFoam и др. Разрабатываемый модуль принимает на вход обновленные значения коэффициентов турбулентной модели, сформированной в процессе оптимизации, записывает файл с настройками для формирования нового расчета и запускает утилиту предобработки, которая подготавливает варианты расчета. Затем параллельно запускаются расчеты кейсов OpenFOAM. По результатам расчетов вычисляется значение функции потерь и передается оптимизатору.

Модуль приведен в приложении и доступен на GitHub [117] вместе с другими утилитами, необходимыми для расчета потоков на горных склонах, таких как утилиты для построения расчётной области на основе цифровой модели рельефа и т.д.

## 5.4 Оптимизация коэффициентов $k$ - $\epsilon$ и $k$ - $\omega$ SST моделей турбулентности

Для оптимизации коэффициентов  $k - \epsilon$  модели турбулентности было проведено 180 расчётов калибровочного эксперимента (рисунок 5.1), что заняло 213 часов работы на 24 ядрах.

Начальные значения коэффициентов были заданы следующими (5.2):

$$C_\mu = 0.09, \quad C_1 = 1.44, \quad C_2 = 1.92, \quad C_3 = 0.0, \quad \sigma_k = 1.0, \quad \sigma_\epsilon = 1.3. \quad (5.2)$$

После калибровки значения коэффициентов стали следующими (5.3):

$$\begin{aligned} C_\mu = 0.0902, \quad C_1 = 1.5419, \quad C_2 = 1.7245, \quad C_3 = 0.0001, \\ \sigma_k = 1.0215, \quad \sigma_\epsilon = 1.3334. \end{aligned} \quad (5.3)$$

Были получены следующие профили скорости на выходной плоскости для различных углов наклона лотка, как показано на рисунке 5.4.

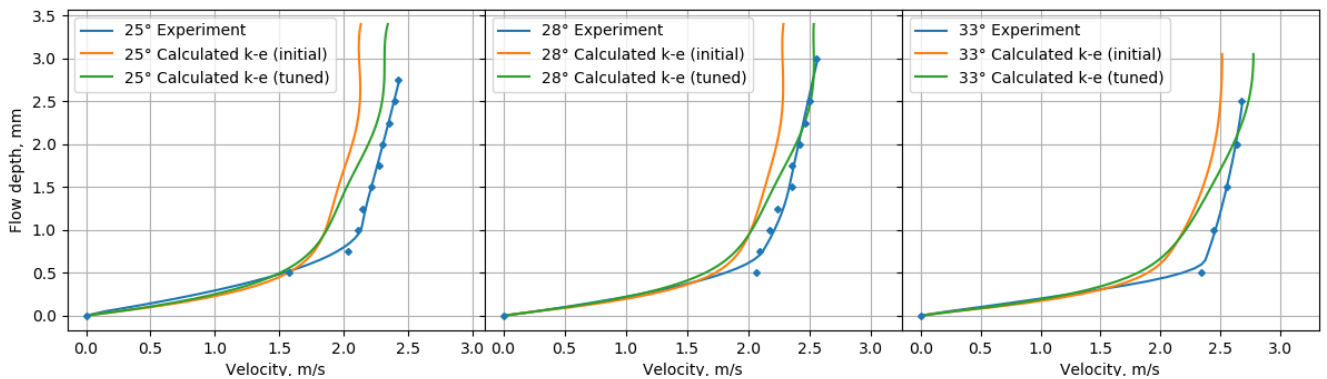


Рисунок 5.4 — Сравнение экспериментального профиля скорости с расчётным при использовании стандартных значений коэффициентов  $k - \epsilon$  модели турбулентности и откалиброванных значений коэффициентов в лотках различного угла наклона к горизонту

Калибровка привела к следующей минимизации функции потерь (5.1), показанной в таблице 8.

Для оптимизации коэффициентов  $k - \omega$  SST модели турбулентности было проведено 924 расчёта калибровочного эксперимента (рисунок 5.1), что заняло 125 часов работы на 24 ядрах.

Начальные значения коэффициентов были заданы следующими (5.4):



Таблица 8 — Минимизация функции потерь

Угол наклона лотка	Начальное значение функции потерь	Минимизированное значение функции потерь
25°	0.242	0.218
28°	0.377	0.389
33°	0.236	0.168

$$\alpha_{k1} = 0.85; \quad \alpha_{k2} = 1.0; \quad \alpha_{\omega1} = 0.5; \quad \alpha_{\omega2} = 0.856; \quad \beta_1 = 0.075; \quad \beta_2 = 0.0828; \\ \beta^* = 0.09; \quad \gamma_1 = 0.5556; \quad \gamma_2 = 0.44; \quad a_1 = 0.31; \quad b_1 = 1.0; \quad c_1 = 10.0. \quad (5.4)$$

После калибровки значения коэффициентов стали следующими (5.5):

$$\alpha_{k1} = 0.822; \quad \alpha_{k2} = 1.1304; \quad \alpha_{\omega1} = 0.466; \quad \alpha_{\omega2} = 0.7796; \\ \beta_1 = 0.0852; \quad \beta_2 = 0.0749; \quad \beta^* = 0.1131; \quad (5.5) \\ \gamma_1 = 0.5156; \quad \gamma_2 = 0.4066; \quad a_1 = 0.3338; \quad b_1 = 0.9508; \quad c_1 = 9.8864.$$

Были получены следующие профили скорости на выходной плоскости для различных углов наклона лотка, как показано на рисунке 5.5.

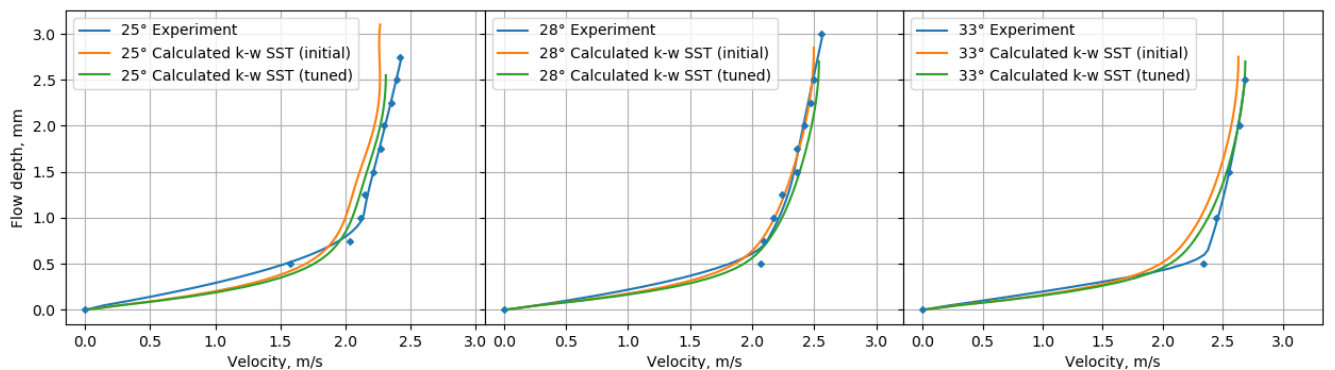


Рисунок 5.5 — Сравнение экспериментального профиля скорости с расчётным при использовании стандартных значений коэффициентов  $k$ - $\omega$  SST модели турбулентности и откалиброванных значений коэффициентов в лотках различного угла наклона к горизонту

Калибровка привела к следующей минимизации функции потерь (5.1), показанной в таблице 9.

Таблица 9 — Минимизация функции потерь

Угол наклона лотка	Начальное значение функции потерь	Минимизированное значение функции потерь
25°	0.165	0.154
28°	0.085	0.082
33°	0.150	0.124

### 5.5 Валидация полученных коэффициентов $k$ - $\epsilon$ и $k$ - $\omega$ $SST$ моделей турбулентности

Был произведён расчёт эксперимента УИ с дамбами (рис. 3.8) и без (рис. 3.9) с использованием откалиброванных коэффициентов  $k$ - $\epsilon$  и  $k$ - $\omega$   $SST$  моделей турбулентности. Были получены следующие результаты, показанные на рисунке 5.6 и в таблице 10.

На рис. 5.6 показано, что осредненная по глубине скорость потока лучше совпадает с экспериментальными данными после калибровки коэффициентов как для  $k$ - $\epsilon$ , так и для  $k$ - $\omega$   $SST$  турбулентной модели. Для параметров, перечисленных в таблице 10 можем видеть уменьшение расхождения для высоты первого всплеска на основной дамбе для модели  $k$ - $\epsilon$  с 26% до 15% и с 50% до 15% для  $k$ - $\omega$   $SST$  модели. для времени взаимодействия потока с основной дамбой ошибка сократилась с 4% до 0.8% для  $k$ - $\omega$   $SST$  модели и не изменилась для  $k$ - $\epsilon$  модели. Для объёма, удержанного дамбой ошибка сократилась для  $k$ - $\epsilon$  модели — с 6% до 2%, для  $k$ - $\omega$   $SST$  — с 1.8% до 0.3%. Из вышесказанного можно сделать вывод, что калибровка коэффициентов турбулентной модели для исследуемого класса задач позволяет увеличить точность расчётов.

Таблица 10 — Сравнение измеренных и рассчитанных параметров потока в эксперименте с дамбами.

	Высота первого всплеска на основной дамбе	Время с начала взаимодействия потока с основной дамбой, до окончания переливания потока через дамбу	Объём, удержанный дамбой из 2.7 м <sup>3</sup>
Эксперимент	1.3 м	1.25 с	2.684 м <sup>3</sup>
<i>k-<math>\omega</math></i> SST модель турбулентности с исходными коэффициентами	1.95 м	1.3 с	2.635 м <sup>3</sup>
<i>k-<math>\omega</math></i> SST модель турбулентности с откалиброванными коэффициентами	1.5 м	1.26 с	2.676 м <sup>3</sup>
<i>k-<math>\epsilon</math></i> модель турбулентности с исходными коэффициентами	1.64 м	1.3 с	2.528 м <sup>3</sup>
<i>k-<math>\epsilon</math></i> модель турбулентности с откалиброванными коэффициентами	1.5 м	1.3 с	2.622 м <sup>3</sup>

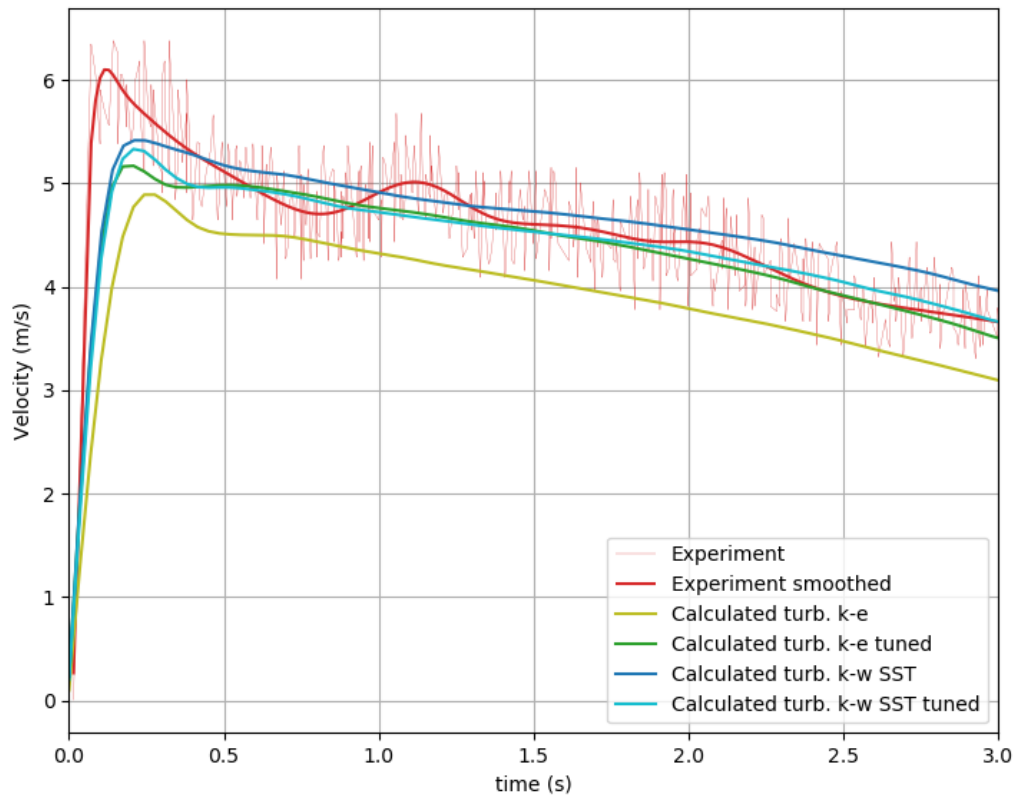


Рисунок 5.6 — Сравнение средней по глубине потока скорости в экспериментальном лотке Университета Исландии при отсутствии дамб, в сечении, находящемся на расстоянии 11.1 м от начала желоба. «turb» — трёхмерный подход с использованием  $k$ - $\varepsilon$  и  $k$ - $\omega$   $SST$  моделей турбулентности, в том числе и с оптимизированными коэффициентами.

## 5.6 Выводы

Данная глава посвящена вопросу определения констант турбулентных моделей при расчёте склоновых потоков. В частности, целью было выяснить, можно ли оптимизировать коэффициенты турбулентных моделей для моделирования таких течений. С этой целью были рассмотрены два экспериментальных исследования течений желобах. Одно из этих исследований выполнено в Университете Исландии, другое — в Институте механики МГУ имени М.В. Ломоносова на экспериментальной установке, созданной автором и инициативной группой.

Была проделана следующая работа. Во-первых, потоки, измеренные в этих двух исследованиях, моделировались с использованием стандартных турбулентных моделей  $k$ - $\varepsilon$  и  $k$ - $\omega$   $SST$  с использованием бесплатного пакета

программ OpenFOAM, и было обнаружено определенное расхождение между некоторыми измеренными и моделированными параметрами потока.

Во-вторых, были разработаны дополнительные модули для пакета OpenFOAM, необходимые для организации оптимизации коэффициентов моделей, в том числе с применением машинного обучения. Эти дополнения к стандартному пакету были размещены в Интернете, чтобы сделать их доступными для всех.

В-третьих, в Институте механики МГУ им. М.В. Ломоносова была изготовлена специальная экспериментальная установка и измерены параметры течения, необходимые для калибровки модели турбулентности.

Затем коэффициенты моделей турбулентности были откалиброваны с использованием экспериментальных результатов, полученных автором и инициативной группой на экспериментальном желобе МГУ. Были использованы три серии экспериментов, что позволило избежать переобучения алгоритма. В результате были скорректированы некоторые значения коэффициентов моделей.

Наконец, моделирование экспериментов УИ, включая взаимодействие с плотинами, было выполнено с использованием скорректированных коэффициентов модели. Результаты лучше согласуются с экспериментальными данными; поэтому сделан вывод о положительном влиянии оптимизации коэффициентов.

## Заключение

Основные результаты работы заключаются в следующем.

1. В результате работы были разработаны трёхмерная математическая и численная модели плотных геофизических потоков на склонах гор, учитывающие сложные реологические свойства материала потока, турбулентный режим движения, сложный реальный рельеф склона, а также возможное вовлечение потоком в движение донного материала.
2. Для чего потребовалось сформулировать обобщенный на трехмерный случай критерий разрушения донного материала и вовлечения его в движение.
3. Математическая постановка задачи была переформулирована в виде, удобном для создания численной модели.
4. Разработан обработчик и конвертер цифровой модели рельефа (библиотека `avalancheUtils`) для получения расчётной области для проведения компьютерного моделирования потоков на склонах реального рельефа.
5. Разработан решатель `avalancheFoam` для расчёта склоновых потоков с использованием модели, учитывающей все сложные свойства потоков на склонах гор. Модель учитывает
  - сложный рельеф местности,
  - трёхмерную структуру,
  - взаимодействие с препятствиями,
  - турбулентный режим течения,
  - нелинейную реологию,
  - разрушение и захват материала склона.
6. Произведена валидация модели и разработанного решателя на модельной задаче потока на склоне постоянного уклона.
7. На основе расчетов потоков в лотках, экспериментально исследованных учеными в университете Исландии, а также потоков на созданной с участием автора экспериментальной установке в НИИ механики МГУ имени М.В. Ломоносова проведено сравнение результатов, полученных при использовании  $k-\varepsilon$  и  $k-\omega SST$  моделей турбулентности при моделировании склоновых потоков разработанным решателем `avalancheFoam`.

8. Проведено трёхмерное моделирование снежной лавины как потока неньютоновской среды в 22-ом лавинном очаге горы Юкспор (Хибины), показавшее хорошее совпадение с натурными данными.
9. Проведено трёхмерное моделирование потока, вызванного прорывом ледникового озера Малый Азау (Центральный Кавказ), продемонстрировавшее предсказание эрозии донного материала потоком.
10. Разработан алгоритм для оптимизации коэффициентов турбулентной модели по результатам эксперимента на базе методов машинного обучения, показавший в среднем улучшение точности расчётов на 5%.
11. В дальнейшем планируется проводить дальнейшую разработку модели и внедрение в инженерные отрасли.

## Список сокращений и условных обозначений

$(P_{n_{fs}})_g$	вектор напряжений в газе, см. (2.18), стр. 34
$(P_{n_{fs}})_l$	вектор напряжений в жидкости, действующий на площадке с нормалью $n_{fs}$ , см. (2.18), стр. 34
$(P_{n_s})_l$	вектор напряжений в потоке, действующий на площадке с нормалью $n_s$ , см. (2.19), стр. 34
$(P_{n_s})_s$	вектор напряжений в склоновом материале, см. (2.19), стр. 34
$\alpha$	объёмная доля жидкой среды, см. (3.0), стр. 37
$\alpha_\omega$	коэффициент турбулентной модели
$\alpha_i$	объёмная доля $i$ -ой фазы, см. (4.1), стр. 62
$\alpha_k$	коэффициент турбулентной модели
$\alpha_{\omega 1}$	коэффициент турбулентной модели
$\alpha_{\omega 2}$	коэффициент турбулентной модели
$\alpha_{k1}$	коэффициент турбулентной модели
$\alpha_{k2}$	коэффициент турбулентной модели
$\bar{p}$	осреднённое по Рейнольдсу давление, см. (2.1), стр. 25
$\beta$	коэффициент турбулентной модели
$\beta^*$	коэффициент турбулентной модели
$\beta_1$	коэффициент турбулентной модели
$\beta_2$	коэффициент турбулентной модели
$\bar{\tau}$	осреднённый по Рейнольдсу тензор напряжений, см. (2.1), стр. 25
$\bar{\tau}_m$	осреднённый по Рейнольдсу тензор молекулярных напряжений, см. (2.2), стр. 25
$\bar{\tau}_t$	тензор турбулентных напряжений, см. (2.2), стр. 25
$\bar{U}$	осреднённая по Рейнольдсу скорость, см. (2.1), стр. 25
$\nabla$	оператор набла
$g$	ускорение свободного падения, см. (2.1), стр. 25
$I$	единичный тензор, см. (2.10), стр. 31
$n_{fs}$	внешняя по отношению к потоку нормаль к свободной поверхности, см. (2.18), стр. 34
$s$	осреднённый тензор скоростей деформаций, см. (2.10), стр. 31



$U$	скорость потока, см. (2.18), стр. 34
$U$	скорость смеси, см. (4.1), стр. 62
$U$	скорость, см. (3.0), стр. 37
$U_0$	профиль скорости потока вдоль нормали ко дну в первой точке замера в эксперименте МГУ, см. (5.0), стр. 89
$U_1$	профиль скорости потока вдоль нормали ко дну во второй точке замера в эксперименте МГУ, см. (5.0), стр. 89
$U_{entr}$	скорость перемещения границы поток-склон, см. (2.20), стр. 35
$U_{fs}$	скорость движения свободной поверхности, см. (2.18), стр. 34
$\cdot$	скалярное произведение
$\chi$	глубина захвата, см. (4.5), стр. 63
$\Delta t$	единица времени, см. (4.5), стр. 63
$\delta_{s/(f+e)}$	дельта функция, определяющая границу разрушаемого материала склона и движущегося потока ( $\delta_{s/e} = 0$ всюду, кроме границы между подстилающим материалом и движущимся потоком, который состоит из материала потока и захваченного материала), см. (4.6), стр. 64
$\dot{m}_{fs}$	поток массы через свободную поверхность, см. (2.18), стр. 34
$\gamma$	коэффициент турбулентной модели
$\gamma_1$	коэффициент турбулентной модели
$\gamma_2$	коэффициент турбулентной модели
$\kappa$	коэффициент консистенции, см. (1.3), стр. 19
$\mu_0$	коэффициент «недеформируемой» вязкости (очень большое число), см. (2.7), стр. 28
$\mu_g$	вязкость газа, см. (3.2), стр. 37
$\mu_h$	коэффициент «сухого» трения, см. (1.1), стр. 15
$\mu_l$	вязкость жидкости, см. (3.2), стр. 37
$\mu_m$	коэффициент эффективной молекулярной вязкости, см. (2.5), стр. 27
$\mu_t$	коэффициент турбулентной вязкости, см. (2.10), стр. 31
$\omega$	специальная скорость диссипации турбулентной кинетической энергии, см. (2.15), стр. 32
$\omega$	специальная скорость диссипации турбулентной кинетической энергии, см. (5.0), стр. 89

$\partial$	частная производная
$\rho$	плотность потока, см. (2.1), стр. 25
$\rho_g$	плотность газа, см. (3.2), стр. 37
$\rho_l$	плотность жидкости, см. (3.2), стр. 37
$\sigma_\varepsilon$	коэффициент турбулентной модели, см. (2.12), стр. 31
$\sigma_k$	коэффициент турбулентной модели, см. (2.12), стр. 31
$\tau$	напряжения сдвига в одномерном сдвиговом потоке, см. (1.3), стр. 19
$\tau_0$	предел текучести, см. (1.3), стр. 19
$Re_{\text{крит}}$	критическое число Рейнольдса, см. (2.9), стр. 31
$\theta$	угол наклона склона к горизонту в одномерном потоке, см. (1.1), стр. 15
$\varepsilon$	диссипация турбулентной кинетической энергии, см. (2.11), стр. 31
$a_1$	коэффициент турбулентной модели
$b_1$	коэффициент турбулентной модели
$BP$	breaking point (предел прочности) см. $s_{s0}$ , см. (4.12), стр. 73
$c_1$	коэффициент турбулентной модели
$C_\mu$	коэффициент турбулентной модели, см. (2.11), стр. 31
$C_{\varepsilon 1}$	коэффициент турбулентной модели, см. (2.12), стр. 31
$C_{\varepsilon 2}$	коэффициент турбулентной модели, см. (2.12), стр. 31
$C_{er}$	коэффициент, характеризующий скорость захвата, см. (4.3), стр. 62
$C_{great}$	большое число, см. (4.12), стр. 65
$F_1$	первая функция перемешивания, см. (2.15), стр. 32
$F_2$	вторая функция перемешивания, см. (2.15), стр. 32
$h_0$	глубина потока в первой точке замера в эксперименте МГУ, см. (5.0), стр. 89
$h_1$	глубина потока во второй точке замера в эксперименте МГУ, см. (5.0), стр. 89
$He$	число Хёдстрёма, см. (2.9), стр. 31
$k$	коэффициент «гидравлического» трения, см. (1.1), стр. 15
$k$	турбулентная кинетическая энергия, см. (2.11), стр. 31
$n$	степенной индекс, см. (1.3), стр. 19

$P_k$	скорость производства турбулентной кинетической энергии средним течением, см. (2.12), стр. 31
$q_{s \rightarrow e}$	скорость изменения объемной доли материала склона за единицу времени за счет его разрушения и перехода в захваченный материал на границе склона и потока, см. (4.1), стр. 62
$S$	единица площади свободной поверхности,, см. (4.5), стр. 63
$s$	скорость деформации в одномерном сдвиговом потоке, см. (1.3), стр. 19
$s_0$	параметр регуляризации реологической модели Хершеля-Балкли, см. (2.7), стр. 28
$s_{er}$	предел скорости сдвига материала склона, который задаёт его предел прочности, см. (4.3), стр. 62
$S_{exp_i}$	явный источниковый член, см. (4.7), стр. 65
$S_{imp_i}$	неявный источниковый член, см. (4.7), стр. 65
$t$	время
$V$	элементарный объём,, см. (4.5), стр. 63
$P_k$	ограничитель нарастания турбулентности, используемый в режимах стагнации, см. (2.15), стр. 32

## Словарь терминов

**OpenFOAM** : Вычислительный пакет для решения задач механики сплошной среды

**nonNewtonianMultiphaseEulerFoam** : Решатель для многофазного многоскоростного потока неньютоновской среды в пакете OpenFOAM

**avalancheFoam** : Решатель для многофазного многоскоростного потока неньютоновской среды с фазовым переходом для разрушения и захвата материала горным потоком в пакете OpenFOAM

**MULES** : Multidimensional Universal Limiter for Explicit Solution

## Список литературы

1. «Домашняя» снежная лавина [Электронный ресурс]. — Accessed: 2023-01-19. [http://www.geogr.msu.ru/science/aero/center/int\\_sem5/5\\_2\\_laviny.htm](http://www.geogr.msu.ru/science/aero/center/int_sem5/5_2_laviny.htm).
2. Лахар [Электронный ресурс]. — Accessed: 2023-01-19. <https://ru.wikipedia.org/wiki/%D0%9B%D0%B0%D1%85%D0%B0%D1%80>.
3. *Докукин М. Д., Черноморец С. С., Севернюк Е. А., Запорожченко Э. В., Бобов Р. А., Пирмамадов У. Р.* Барсемская селевая катастрофа на Памире в 2015 году и ее аналоги на Центральном Кавказе // Геориск. — 2019. — т. 13, № 1. — с. 26—36.
4. *Romanova D., Ivanov O., Trifonov V., Ginzburg N., Korovina D., Ginzburg B., Koltunov N., Eglit M., Strijhak S.* Calibration of the k- $\omega$  SST Turbulence Model for Free Surface Flows on Mountain Slopes Using an Experiment // *Fluids*. — 2022. — т. 7, № 3. — DOI: [10.3390/fluids7030111](https://doi.org/10.3390/fluids7030111). — SJR: 0,41.
5. *Barkalov K., Lebedev I., Usova M., Romanova D., Ryazanov D., Strijhak S.* Optimization of Turbulence Model Parameters Using the Global Search Method Combined with Machine Learning // *Mathematics*. — 2022. — т. 10, № 15. — DOI: [10.3390/math10152708](https://doi.org/10.3390/math10152708). — SJR: 0,54.
6. *Romanova D., Eglit M.* Bottom material entrainment in 3D simulations of flows on mountain slopes in open-source solver avalancheFoam // 2022 Ivannikov Ispras Open Conference (ISPRAS). — 2023. — с. 70—73. — DOI: [10.1109/ISPRAS57371.2022.10076266](https://doi.org/10.1109/ISPRAS57371.2022.10076266).
7. *Романова Д. И.* Трёхмерное моделирование схода лавинных потоков средствами пакета OpenFOAM // Труды Института системного программирования РАН. — 2017. — т. 29, № 1. — с. 85—100. — DOI: [10.15514/ISPRAS-2017-29\(1\)-6](https://doi.org/10.15514/ISPRAS-2017-29(1)-6).
8. *Romanova D.* Comparison of Single-Velocity and Multi-Velocity Multiphase Models for Slope Flow Simulations // 2020 Ivannikov Ispras Open Conference (ISPRAS). — 2020. — с. 170—174. — DOI: [10.1109/ISPRAS51486.2020.00033](https://doi.org/10.1109/ISPRAS51486.2020.00033).

9. *Романова Д. И.* Архитектура программного средства с открытым исходным кодом для численного моделирования потоков на горных склонах // Труды Института системного программирования РАН (электронный журнал). — 2020. — т. 32, № 6. — с. 14. — DOI: [10.15514/ISPRAS-2020-32\(6\)-14](https://doi.org/10.15514/ISPRAS-2020-32(6)-14). — РИНЦ: 0,19.
10. *Романова Д. И.* Калибровка  $k-\epsilon$  модели турбулентности в пакете OpenFOAM с помощью методов машинного обучения для моделирования потоков на склонах гор на основе эксперимента // Труды Института системного программирования РАН. — 2021. — т. 33, № 4. — с. 227–240. — DOI: [10.15514/ispras-2021-33\(4\)-16](https://doi.org/10.15514/ispras-2021-33(4)-16). — РИНЦ: 0,19.
11. Лавинная катастрофа в г. Кировске 5 декабря 1935 года [Электронный ресурс]. — Accessed: 2023-01-20. <https://mvc-apatit.ru/news/49>.
12. *Гофф А. Г., Оттен Г. Ф.* Экспериментальное определение силы удара снежных облаков // Извещения академии наук СССР. — 1939. — т. Серия географическая, № 23.
13. *Козик С. М.* Расчет движения снежных лавин. — Л.: Гидрометиздат, 1962.
14. *Salm B.* Contribution to avalanche dynamics // International Association of Scientific Hydrology Pulzlication. — 1966. — т. Symposium at Davos 1965 —Scientific Aspects aISnow and Ice Avalanches, № 69. — с. 199–214.
15. *Григорян С. С., Эглит М. Э., Якимов Ю. Л.* Новая математическая постановка задачи о движении лавины и решение этой задачи // Труды Высокогорного геофизического института. — 1967. — № 12. — с. 104–113.
16. *Миронова Е.* Математическое моделирование движения водных потоков, снежных лавин и селей : дис. . . . канд. / Миронова Е. — Москва, Россия : МГУ, 1987.
17. *Куликовский А. Г., Эглит М. Э.* Двумерная задача о движении снежной лавины по склону с плавно меняющимися свойствами // ПММ. — 1973. — т. 37, № 5. — с. 837–848.
18. *Данилова Е. М., Эглит М. Э.* Движение лотковых лавин // Материалы гляциологических исследований. — 1977. — № 31. — с. 65–74.

19. *Григорян С. С.* Новый закон трения и механизм крупномасштабных горных обвалов и оползней // Доклады Академии наук. — М., 1979. — т. 244, № 4. — с. 846—846.
20. *Eglit M., Demidov K.* Mathematical modeling of snow entrainment in avalanche motion // Cold Regions Science and Technology. — 2005. — т. 43, № 1. — с. 10—23. — DOI: [10.1016/j.coldregions.2005.03.005](https://doi.org/10.1016/j.coldregions.2005.03.005). — Snow and Avalanches.
21. *Eglit M.* Some Mathematical Models of Snow Avalanches // Advances in the Mechanics and the Flow of Granular Materials. — 1983. — т. 2. — с. 557—588.
22. *Rauter M., Kofler A., Huber A., Fellin W.* faSavageHutterFOAM 1.0: depth-integrated simulation of dense snow avalanches on natural terrain with OpenFOAM // Geoscientific Model Development. — 2018. — т. 11, № 7. — с. 2923—2939. — DOI: [10.5194/gmd-11-2923-2018](https://doi.org/10.5194/gmd-11-2923-2018).
23. *Rauter M., Tukovic Z.* A finite area scheme for shallow granular flows on three-dimensional surfaces // Computers & Fluids. — 2018. — т. 166. — с. 184—199. — DOI: [10.1016/j.compfluid.2018.02.017](https://doi.org/10.1016/j.compfluid.2018.02.017).
24. *Rauter M., Heerema K., Issler D., Talling P.* Applications of the Finite Area Method on a Geographic Scale: From Dense Snow Avalanches to Turbidity Currents //. — 10.2019. — с. 1—10.
25. *Беликов В., Алексюк А., Васильева Е.* Численное моделирование волн прорыва. — Москва : РАН, 2023.
26. *Yamaguchi Y., Takase S., Moriguchi S., Terada K., Oda K., Kamiishi I.* Three-dimensional nonstructural finite element analysis of snow avalanche using non-Newtonian fluid model // Transactions of the Japan Society for Computational Engineering and Science. — 2017. — т. 2017. — DOI: [10.11421/jsces.2017.20170011](https://doi.org/10.11421/jsces.2017.20170011).
27. *Li X., Sovilla B., Ligneau C., Jiang C., Gaume J.* Different erosion and entrainment mechanisms in snow avalanches // Mechanics Research Communications. — 2022. — т. 124. — с. 103914. — DOI: [10.1016/j.mechrescom.2022.103914](https://doi.org/10.1016/j.mechrescom.2022.103914).



28. *Franci A., Cremonesi M., Perego U., Crosta G., Onate E.* 3D simulation of Vajont disaster. Part 1: Numerical formulation and validation // *Engineering Geology*. — 2020. — т. 279. — с. 105854. — DOI: [10.1016/j.enggeo.2020.105854](https://doi.org/10.1016/j.enggeo.2020.105854).
29. *Peng C., Li S., Wu W., An H., Chen X., Ouyang C., Tang H.* On three-dimensional SPH modelling of large-scale landslides // *Canadian Geotechnical Journal*. — 2022. — т. 59, № 1. — с. 24–39. — DOI: [10.1139/cgj-2020-0774](https://doi.org/10.1139/cgj-2020-0774).
30. *Петров А., Потапов И.* Избранные разделы русловой динамики. — Москва : URSS, 2019.
31. *Эглит М. Э.* Неустановившиеся движения в руслах и на склонах. — Издательство Московского университета Москва, 1986. — 96 с.
32. *Hungr O.* A model for the runout analysis of rapid flow slides, debris flows, and avalanches // *Canadian Geotechnical Journal*. — 1995. — авг. — т. 32. — с. 610–623. — DOI: [10.1139/t95-063](https://doi.org/10.1139/t95-063).
33. *Sampl P., Granig M.* Avalanche Simulation with SAMOS-AT // *International Snow Science Workshop*. — 2009. — с. 519–523.
34. *Sampl P., Zwinger T.* Avalanche simulation with SAMOS // *Annals of Glaciology*. — 2004. — т. 38. — с. 393–398. — DOI: [10.3189/172756404781814780](https://doi.org/10.3189/172756404781814780).
35. *Medina V., Hurlimann M., Bateman A.* Application of FLATModel, a 2D finite volume code, to debris flows in the northeastern part of the Iberian Peninsula // *Landslides*. — 2007. — февр. — т. 5. — с. 127–142. — DOI: [10.1007/s10346-007-0102-3](https://doi.org/10.1007/s10346-007-0102-3).
36. *Christen M., Kowalski J., Bartelt P.* RAMMS: Numerical simulation of dense snow avalanches in three-dimensional terrain // *Cold Regions Science and Technology*. — 2010. — т. 63, № 1. — с. 1–14. — DOI: [10.1016/j.coldregions.2010.04.005](https://doi.org/10.1016/j.coldregions.2010.04.005).
37. *Pitman E. B., Nichita C. C., Patra A., Bauer A., Sheridan M., Bursik M.* Computing granular avalanches and landslides // *Physics of Fluids*. — 2003. — т. 15, № 12. — с. 3638–3646. — DOI: [10.1063/1.1614253](https://doi.org/10.1063/1.1614253).

38. *Patra A., Bauer A., Nichita C., Pitman E., Sheridan M., Bursik M., Rupp B., Webber A., Stinton A., Namikawa L., Renschler C.* Parallel adaptive numerical simulation of dry avalanches over natural terrain // *Journal of Volcanology and Geothermal Research*. — 2005. — т. 139, № 1. — с. 1—21. — DOI: [10.1016/j.jvolgeores.2004.06.014](https://doi.org/10.1016/j.jvolgeores.2004.06.014). — Modeling and Simulation of Geophysical Mass Flows.
39. *Mergili M., Schratz K., Ostermann A., Fellin W.* Physically-based modelling of granular flows with Open Source GIS // *Natural Hazards and Earth System Sciences*. — 2012. — т. 12, № 1. — с. 187—200. — DOI: [10.5194/nhess-12-187-2012](https://doi.org/10.5194/nhess-12-187-2012).
40. *Mergili M., Fischer J.-T., Krenn J., Pudasaini S. P.* r.avaflow v1, an advanced open-source computational framework for the propagation and interaction of two-phase mass flows // *Geoscientific Model Development*. — 2017. — т. 10, № 2. — с. 553—569. — DOI: [10.5194/gmd-10-553-2017](https://doi.org/10.5194/gmd-10-553-2017).
41. *Hergarten S., Robl J.* Modelling rapid mass movements using the shallow water equations in Cartesian coordinates // *Natural Hazards and Earth System Sciences*. — 2015. — т. 15, № 3. — с. 671—685. — DOI: [10.5194/nhess-15-671-2015](https://doi.org/10.5194/nhess-15-671-2015).
42. *Oda K., Moriguchi S., Kamiishi I., Yashima A., Sawada K., Sato A.* Simulation of a snow avalanche model test using computational fluid dynamics // *Annals of Glaciology*. — 2011. — т. 52, № 58. — с. 57—64. — DOI: [10.3189/172756411797252284](https://doi.org/10.3189/172756411797252284).
43. *Clement Rastello M.* Etude de la dynamique des avalanches de neige en aerosol : дис. ... канд. / Clement Rastello Marie. — 2002. — 1 vol. (171 p.) — URL: <http://www.theses.fr/2002GRE10099/document> ; 2002GRE10099.
44. *Coussot P., Laigle D., Arattano M., Deganutti A. M., Marchi L.* Direct Determination of Rheological Characteristics of Debris Flow // *Journal of Hydraulic Engineering*. — 2000. — т. 124. — с. 865—868.
45. *Kern M., Tiefenbacher F., McElwaine J.* The rheology of snow in large chute flows // *Cold Regions Science and Technology*. — 2004. — т. 39, № 2/3. — с. 181—192. — DOI: [10.1016/j.coldregions.2004.03.006](https://doi.org/10.1016/j.coldregions.2004.03.006).

46. *Rognon P., Chevoir F., Bellot H., Ousset F., Naaim M., Coussot P.* Rheology of dense snow flows: inferences from steady state chute-flow experiments // *Journal of Rheology*. — 2008. — т. 52, № 3. — с. 729—748. — DOI: [10.1122/1.2897609](https://doi.org/10.1122/1.2897609).
47. *Dent J., Burrell K., Schmidt D. S., Louge M., Adams E., Jazbutis T.* Density, velocity and friction measurements in a dry-snow avalanche // *Annals of Glaciology*. — 1998. — т. 26. — с. 247—252. — DOI: [10.3189/1998AoG26-1-247-252](https://doi.org/10.3189/1998AoG26-1-247-252).
48. *Peng C., Guo X., Wu W., Wang Y.* Unified modelling of granular media with Smoothed Particle Hydrodynamics // *Acta Geotechnica*. — 2016. — дек. — т. 11, № 6. — с. 1231—1247. — DOI: [10.1007/s11440-016-0496-y](https://doi.org/10.1007/s11440-016-0496-y).
49. *Зайко Ю.* Режимы течения и устойчивость потоков на склонах : 01.02.05 / Зайко Ю.С. — Москва, Россия : МГУ имени М.В. Ломоносова, 2020.
50. *Balmforth N., Craster R., Sassi R.* Dynamics of cooling viscoplastic domes // *Journal of Fluid Mechanics*. — 2004. — янв. — т. 499. — с. 149—182. — DOI: [10.1017/S0022112003006840](https://doi.org/10.1017/S0022112003006840).
51. *Eglit M., Yakubenko A.* Numerical modeling of slope flows entraining bottom material // *Cold Regions Science and Technology*. — 2014. — т. 108. — с. 139—148. — DOI: [10.1016/j.coldregions.2014.07.002](https://doi.org/10.1016/j.coldregions.2014.07.002).
52. *Major J., Pierson T.* Debris Flow Rheology: Experimental Analysis of Fine-Grained Slurries // *Water Resources Research - WATER RESOUR RES*. — 1992. — март. — т. 28. — с. 841—857. — DOI: [10.1029/91WR02834](https://doi.org/10.1029/91WR02834).
53. *Rzadkiewicz S. A., Mariotti C., Heinrich P.* Numerical Simulation of Submarine Landslides and Their Hydraulic Effects // *Journal of Waterway, Port, Coastal, and Ocean Engineering*. — 1997. — т. 123, № 4. — с. 149—157. — DOI: [10.1061/\(ASCE\)0733-950X\(1997\)123:4\(149\)](https://doi.org/10.1061/(ASCE)0733-950X(1997)123:4(149)).
54. *O'Brien J. S., Julien P. Y.* Laboratory Analysis of Mudflow Properties // *Journal of Hydraulic Engineering*. — 1988. — т. 114, № 8. — с. 877—887. — DOI: [10.1061/\(ASCE\)0733-9429\(1988\)114:8\(877\)](https://doi.org/10.1061/(ASCE)0733-9429(1988)114:8(877)).
55. *Mellor M., Research C. R., (U.S.) E. L.* A Review of Basic Snow Mechanics. — U.S. Army Cold Regions Research, Engineering Laboratory, 1974.

56. *Dent J., Lang T.* Modeling of snow flow // *Glaciology*. — 1980. — т. 26. — с. 131—140.
57. *Григорян С. С., Остроумов А. В.* Математическая модель склоновых процессов лавинного типа // *НИИ Механики МГУ. Научный отчёт*. — 1977. — № 1955.
58. *Куликовский А. Г., Свешникова Е. И.* Модель для расчета движения пылевой снежной лавины // *Материалы гляциологических исследований*. — 1977. — т. 31. — с. 74—80.
59. *Briukhanov A., Grigorian S., Miagkov S., Plam M., Shurova I., Eglit M., Yakimov Y.* On some new approaches to the dynamics of snow avalanches // т. 1. — 1966. — с. 1223—1241.
60. *Eglit M.* Theoretical approaches to the calculation of the motion of snow avalanches // *Itogi Nauki*. — 1968. — с. 60—97.
61. *Sovilla B.* Field Experiments and Numerical Modelling of Mass Entrainment and Deposition Processes in Snow Avalanches, PhD thesis / *Sovilla Betty*. — 01.2004.
62. *Бахвалов Н. С., Эглит М. Э.* Исследование решений уравнений снежных лавин // *Материалы гляциологических исследований*. — 1970. — т. 16. — с. 7—14.
63. *Eglit M.* Calculation of the parameters of avalanches in the runout zone // *Mater. Glyatsiologicheskikh Issled. (Data Glaciol. Stud.)* — 1982. — т. 53. — с. 35—39.
64. *Eglit M.* Unsteady Motions in Channels and on Slopes // *MSU Press*. — 1986.
65. *Blagovechshenskiy V., Eglit M.* Mathematical modelling of the effect of the parameters of avalanche sources and physical properties of the snow on the dynamics of avalanches // *Mater. Glyatsiologicheskikh Issled. (Data Glaciol. Stud.)* — 1985. — т. 53. — с. 108—112.
66. *Blagovechshenskiy V., Mironova E., Eglit M.* Calculation of avalanche parameters in little-studied mountain areas // *Mater. Glyatsiologicheskikh Issled. (Data Glaciol. Stud.)* — 1995. — т. 79. — с. 36—40.
67. *Blagovechshenskiy V.* Determination of Avalanche Loads // *Gylym: Almaty, Kazakhstan*. — 1995.

68. *Потанов Д. И., Потанов И. И.* Развитие берегового откоса в русле трапециевидного канала // Компьютерные исследования и моделирование. — 2022. — т. 14, № 3. — с. 581—592. — DOI: [10.20537/2076-7633-2022-14-3-581-592](https://doi.org/10.20537/2076-7633-2022-14-3-581-592).
69. *Бахвалов Н. С., Эглит М. Э.* Исследование одномерного движения снежной лавины по плоскому склону // Известия Академии наук СССР. Механика жидкости и газа. — 1973. — № 5. — с. 7—14.
70. *Issler D.* Dynamically consistent entrainment laws for depth-averaged avalanche models // Journal of Fluid Mechanics. — 2014. — т. 759. — с. 701—738. — DOI: [10.1017/jfm.2014.584](https://doi.org/10.1017/jfm.2014.584).
71. *Grigorian S. S., Ostroumov A. V.* On a Continuum Model for Avalanche Flow and Its Simplified Variants // Geosciences. — 2020. — т. 10, № 1. — DOI: [10.3390/geosciences10010035](https://doi.org/10.3390/geosciences10010035).
72. *Sovilla B., Burlando P., Bartelt P.* Field experiments and numerical modeling of mass entrainment in snow avalanches // Journal of Geophysical Research: Earth Surface. — 2006. — т. 111, F3. — 16 стр. — DOI: [doi.org/10.1029/2005JF000391](https://doi.org/10.1029/2005JF000391).
73. *Григорян С., Остроумов А.* Математическая модель склоновых процессов лавинного типа // Научный отчёт №1955; НИИ Механики МГУ, Москва, Россия. — 1977. — т. 55. — с. 70—84.
74. *Эглит М. Э., Вельтищев Н. Н.* Исследование математических моделей снежно-пылевой лавины // Материалы гляциологических исследований. — 1985. — т. 53. — с. 116—120.
75. *Nazarov A. N.* Mathematical modeling of a snow-powder avalanche in the framework of the equations of two-layer shallow water // Fluid Dynamics. — 1991. — т. 26. — с. 70—75. — DOI: [10.1007/BF01050115](https://doi.org/10.1007/BF01050115).
76. *Назаров А. Н.* Опыт применения двухслойной модели для расчёта движения пылевых лавин // Материалы гляциологических исследований. — 1992. — с. 73—79.
77. *Назаров А. Н.* Математическое моделирование нестационарного движения снежно-пылевых лавин : дис. ... канд. / Назаров А. Н. — Москва, Россия : МГУ имени М.В. Ломоносова, 1993.

78. *Эглит М. Э.* Математическое и физическое моделирование снежно-пылевых лавин // Материалы гляциологических исследований. — 1998. — т. 84. — с. 76—79.
79. *Landau L., Lifshitz E.* Fluid Mechanics: Volume 6 of Course of Theoretical Physics. — Elsevier Science, 1987.
80. *Issler D., Pastor Perez M.* Interplay of entrainment and rheology in snow avalanches: a numerical study // Annals of Glaciology. — 2011. — т. 52, № 58. — с. 143—147. — DOI: [10.3189/172756411797252031](https://doi.org/10.3189/172756411797252031).
81. *Fischer J.-T., Kofler A., Fellin W., Granig M., Kleemayr K.* Multivariate parameter optimization for computational snow avalanche simulation // Journal of Glaciology. — 2015. — т. 61, № 229. — с. 875—888. — DOI: [10.3189/2015JoG14J168](https://doi.org/10.3189/2015JoG14J168).
82. *Nishimura K., Barpi F., Issler D.* Perspectives on Snow Avalanche Dynamics Research // Geosciences. — 2021. — т. 11, № 2. — DOI: [10.3390/geosciences11020057](https://doi.org/10.3390/geosciences11020057).
83. *Issler D.* Comments on "On a Continuum Model for Avalanche Flow and Its Simplified Variants" by S. S. Grigorian and A. V. Ostroumov // Geosciences. — 2020. — т. 10, № 3. — DOI: [10.3390/geosciences10030096](https://doi.org/10.3390/geosciences10030096).
84. *Grigorian S. S., Eglit M. E., Iakimov Y. L.* A new formulation and solution of the problem of snow avalanche motion // Trudy Vycokogornogo Geofizicheskogo Inst. — 1967. — т. 12. — с. 104—113.
85. *Issler D., Jóhannesson T.* Dynamically consistent entrainment and deposition rates in depth-averaged gravity mass flow models // NGI Technical Note 20110112-01-TN; Norwegian Geotechnical Institute: Oslo, Norway. — 2011. — DOI: [10.13140/RG.2.2.31327.71840](https://doi.org/10.13140/RG.2.2.31327.71840).
86. *Rauter M., Köhler A.* Constraints on Entrainment and Deposition Models in Avalanche Simulations from High-Resolution Radar Data // Geosciences. — 2020. — т. 10, № 1. — DOI: [10.3390/geosciences10010009](https://doi.org/10.3390/geosciences10010009).
87. *Sovilla B., Bartelt P.* Observations and modelling of snow avalanche entrainment // Natural Hazards and Earth System Science. — 2002. — дек. — т. 2. — DOI: [10.5194/nhess-2-169-2002](https://doi.org/10.5194/nhess-2-169-2002).

88. *Gavrilov A. A., Rudyak V. Y.* Reynolds-averaged modeling of turbulent flows of power-law fluids // Journal of Non-Newtonian Fluid Mechanics. — 2016. — т. 227. — с. 45–55. — DOI: [10.1016/j.jnnfm.2015.11.006](https://doi.org/10.1016/j.jnnfm.2015.11.006).
89. *Lovato S., Keetels G., Toxopeus S., Settels J.* An eddy-viscosity model for turbulent flows of Herschel–Bulkley fluids // Journal of Non-Newtonian Fluid Mechanics. — 2022. — т. 301. — с. 104729. — DOI: <https://doi.org/10.1016/j.jnnfm.2021.104729>.
90. *Burgos G. R., Alexandrou A. N., Entov V.* On the determination of yield surfaces in Herschel-Bulkley fluids // Journal of Rheology. — 1999. — т. 43, № 3. — с. 463–483. — DOI: [10.1122/1.550992](https://doi.org/10.1122/1.550992).
91. *Greenshields C., Weller H.* Notes on Computational Fluid Dynamics: General Principles. — Reading, UK : CFD Direct Ltd, 2022.
92. *Tahry S. H. E.* k-epsilon equation for compressible reciprocating engine flows // Journal of Energy. — 1983. — т. 7, № 4. — с. 345–353. — DOI: [10.2514/3.48086](https://doi.org/10.2514/3.48086).
93. *Launder B., Spalding D.* The numerical computation of turbulent flows // Comput. Methods Appl. Mech. Eng. — 1974. — янв. — т. 103. — с. 456–460.
94. *Launder B., Morse A., Rodi W., Spalding D.* Spalding, The prediction of free shear flows - A comparison of the performance of six turbulence models // Proceedings of NASA Conference on Free Shear Flows. — 1972.
95. *Xenakis A.* Modelling Multi-phase Non-Newtonian Flows using Incompressible SPH : PhD thesis / Xenakis Antonios. — University of Manchester, School of Mechanical, Aerospace, Civil Engineering, 2016.
96. *Handler R., Swearingen J., Swaan T., Leighton R.* Length Scales of Turbulence Near a Free Surface // AIAA. — 1991. — июнь. — т. 91–01775. — с. 16. — DOI: [10.2514/6.1991-1775](https://doi.org/10.2514/6.1991-1775).
97. *Leighton R., Swaan T., Handler R., Swearingen J.* Interaction of Vorticity with a Free Surface in Turbulent Open Channel Flow. — 1991. — янв. — DOI: [10.2514/6.1991-236](https://doi.org/10.2514/6.1991-236).
98. *Swaan T., Leighton R., Handler R., Swearingen J.* Turbulence Modeling Near the Free Surface in an Open Channel Flow // AIAA. — 1991. — янв. — т. 91–0613. — с. 11. — DOI: [10.2514/6.1991-613](https://doi.org/10.2514/6.1991-613).



99. Храбрый А. И., Смирнов Е. М., Зайцев Д. К. Влияние модели турбулентности на результаты расчета обтекания препятствия потоком воды после обрушения дамбы // Научно-технические ведомости СПбГПУ. Физико-математические науки. — 2013. — т. 165, № 1. — с. 182—187.
100. Храбрый А. И. Численное моделирование нестационарных турбулентных течений жидкости со свободной поверхностью : дис. ... канд. / Храбрый Александр Иосифович. — ФГАОУ ВО «Санкт-Петербургский государственный политехнический университет», 2015.
101. Wilcox D. Turbulence Modeling for CFD (Third Edition). — 01.2006. — с. 536.
102. Malin M. Turbulent pipe flow of Herschel-Bulkley fluids // International Communications in Heat and Mass Transfer. — 1998. — т. 25, № 3. — с. 321—330. — DOI: [10.1016/S0735-1933\(98\)00019-0](https://doi.org/10.1016/S0735-1933(98)00019-0).
103. Malin M. Turbulent pipe flow of power-law fluids // International Communications in Heat and Mass Transfer. — 1997. — т. 24, № 7. — с. 977—988. — DOI: [10.1016/S0735-1933\(97\)00083-3](https://doi.org/10.1016/S0735-1933(97)00083-3).
104. Haldenwang R. Flow of non-newtonian fluids in open channels : дис. ... канд. / Haldenwang Rainer. — Department of Civil Engineering Cape Technikon, South Afrika, 2003.
105. OpenFOAM: User Guide [Электронный ресурс]. — Accessed: 2023-05-25. <https://www.openfoam.com/documentation/guides/v2112/doc/index.html>.
106. Menter F. Zonal Two Equation k-w Turbulence Models For Aerodynamic Flows // 23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference. — 1993. — DOI: [10.2514/6.1993-2906](https://doi.org/10.2514/6.1993-2906).
107. Menter F., Kuntz M., Langtry R. Ten years of industrial experience with the SST turbulence model // Heat and Mass Transfer. — 2003. — янв. — т. 4.
108. Задание параметров турбулентности в начальный момент и на границе [Электронный ресурс]. — Accessed: 2023-05-08. [https://www.cfd-online.com/Wiki/Turbulence\\_free-stream\\_boundary\\_conditions](https://www.cfd-online.com/Wiki/Turbulence_free-stream_boundary_conditions).
109. Седов Л. И. Механика сплошной среды. В 2 томах. — М.: Наука, 1970. — с. 492.

110. *Wilcox D. C.* Reassessment of the scale-determining equation for advanced turbulence models // *AIAA Journal*. — 1988. — т. 26, № 11. — с. 1299–1310. — DOI: [10.2514/3.10041](https://doi.org/10.2514/3.10041).
111. *Liu S., Ong M. C., Obhrai C., Gatin I., Vukcevic V.* Influences of free surface jump conditions and different  $k-\omega$  SST turbulence models on breaking wave modelling // *Ocean Engineering*. — 2020. — т. 217. — с. 107746. — DOI: [10.1016/j.oceaneng.2020.107746](https://doi.org/10.1016/j.oceaneng.2020.107746).
112. *Harlow F. H., Welch J. E.* Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface // *The Physics of Fluids*. — 1965. — т. 8, № 12. — с. 2182–2189. — DOI: [10.1063/1.1761178](https://doi.org/10.1063/1.1761178).
113. *Hirt C., Nichols B.* Volume of fluid (VOF) method for the dynamics of free boundaries // *Journal of Computational Physics*. — 1981. — т. 39, № 1. — с. 201–225. — DOI: [10.1016/0021-9991\(81\)90145-5](https://doi.org/10.1016/0021-9991(81)90145-5).
114. *Ferziger J., Peric M.* Computational Methods for Fluid Dynamics. т. 3. — Springer, 01.2002. — DOI: [10.1007/978-3-642-56026-2](https://doi.org/10.1007/978-3-642-56026-2).
115. *Rusche H.* Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions : дис. ... канд. / Rusche Henrik. — 01.2002.
116. *Tryggvason G., Bunner B., Che J., Ebrat O., Esmaeeli A., Han J., Homma S., Juric D., Mortazavi S., Nas S., Nobari M. R. H., Tauber W.* Computations of Multiphase Flows by a Finite Difference/Front Tracking Method – I. Multi-Fluid Flows; II. Applications; III. Variable Surface Tension and Phase Change. — In 29th Comp. Fluid Dyn. Lecture Ser. Von Karman Institute for Fluid Dynamics, 1998.
117. *Romanova D.* avalancheUtils [Электронный ресурс]. — 2023. — URL: <https://github.com/RomanovaDI/avalancheUtils> ; Last accessed 20 May 2023.
118. *Romanova D.* avalancheFoam [Электронный ресурс]. — 2023. — URL: <https://github.com/RomanovaDI/avalancheFoam> ; Last accessed 20 May 2023.
119. *Holzmann T.* Mathematics, Numerics, Derivations and OpenFOAM. — 11.2019.

120. *Barton I. E.* Comparison of SIMPLE- and PISO-type algorithms for transient flows // International Journal for Numerical Methods in Fluids. — 1998. — т. 26, № 4. — с. 459—483. — DOI: [10.1002/\(SICI\)1097-0363\(19980228\)26:4<459::AID-FLD645>3.0.CO;2-U](https://doi.org/10.1002/(SICI)1097-0363(19980228)26:4<459::AID-FLD645>3.0.CO;2-U).
121. *Issa R.* Solution of the implicitly discretised fluid flow equations by operator-splitting // Journal of Computational Physics. — 1986. — т. 62, № 1. — с. 40—65. — DOI: [10.1016/0021-9991\(86\)90099-9](https://doi.org/10.1016/0021-9991(86)90099-9).
122. *Patankar S., Spalding D.* A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows // International Journal of Heat and Mass Transfer. — 1972. — т. 15, № 10. — с. 1787—1806. — DOI: [10.1016/0017-9310\(72\)90054-3](https://doi.org/10.1016/0017-9310(72)90054-3).
123. ASTER Global Digital Elevation Map Announcement [Электронный ресурс]. — Accessed: 2023-05-09. <https://asterweb.jpl.nasa.gov/gdem.asp>.
124. USGS TopoView [Электронный ресурс]. — Accessed: 2023-05-09. <https://ngmdb.usgs.gov/topoview/viewer/#1/66/89>.
125. ALOS (AW3D) OpenTopography High-Resolution Topography Data and Tools [Электронный ресурс]. — Accessed: 2023-05-09. <https://portal.opentopography.org/login>.
126. *Agustsdottir K. H.* The design of slushflow barriers: Laboratory experiments : дис. ... канд. / Agustsdottir Katrin Helga. — Haskolaprent, Falkagata 2, 107 Reykjavik, Iceland : Faculty of Industrial Eng., Mechanical Eng., Computer Science, University of Iceland, 05.2019.
127. *Jones R. A.* The Design of Slushflow Barriers:CFD Simulations : дис. ... канд. / Jones Rebecca Anne. — Haskolaprent, Falkagata 2, 107 Reykjavik, Iceland : Faculty of Industrial Eng., Mechanical Eng., Computer Science, University of Iceland, 10.2019.
128. *Sovilla B., Margreth S., Bartelt P.* On snow entrainment in avalanche dynamics calculations // Cold Regions Science and Technology. — 2007. — янв. — т. 47. — с. 69—79. — DOI: [10.1016/j.coldregions.2006.08.012](https://doi.org/10.1016/j.coldregions.2006.08.012).
129. *Eglit M., Yakubenko A., Zayko J.* A Review of Russian Snow Avalanche Models—From Analytical Solutions to Novel 3D Models // Geosciences. — 2020. — февр. — т. 10. — с. 77. — DOI: [10.3390/geosciences10020077](https://doi.org/10.3390/geosciences10020077).

130. *Oruganti V. K.* Implementation of cavitation models into the multiphaseEulerFoam solver. — 2017. — сент.
131. *Dokukin M., Khatkutov A.* Lakes near the glacier Maliy Azau on the Elbrus (Central Caucasus): dynamics and outbursts // Ice and Snow. — 2016. — дек. — т. 56. — с. 472—479. — DOI: [10.15356/2076-6734-2016-4-472-479](https://doi.org/10.15356/2076-6734-2016-4-472-479).
132. topoView map [Электронный ресурс]. — Accessed: 2023-05-25. <https://ngmdb.usgs.gov/topoview/viewer>.
133. *Chen Y.-S., Kim S.-W.* Computation of turbulent flows using an extended k-epsilon turbulence closure model. — 1987.
134. *Wilcox D. C., Rubesin M. W.* Progress in turbulence modeling for complex flow fields including effects of compressibility //. — 1980.
135. *Menter F. R.* Two-equation eddy-viscosity turbulence models for engineering applications // AIAA Journal. — 1994. — т. 32, № 8. — с. 1598—1605. — DOI: [10.2514/3.12149](https://doi.org/10.2514/3.12149).
136. *Rhee S. H., Makarov B. P., Krishinan H., Ivanov V.* Assessment of the volume of fluid method for free-surface wave flow // Journal of marine science and technology. — 2005. — т. 10, № 4. — с. 173—180.
137. *Rodriguez-Ocampo P., Ring M., Hernandez-Fontes J., Alcerreca-Huerta J., Mendoza E., Gallegos-Diez-Barroso G., Silva R.* A 2D Image-Based Approach for CFD Validation of Liquid Mixing in a Free-Surface Condition // Journal of Applied Fluid Mechanics. — 2020. — т. 13, № 5.
138. *Doe J.* Comparison of turbulence models in predicting the flow around a surface mounted cube. —
139. *Wilcox D. C.* Formulation of the kw turbulence model revisited // AIAA journal. — 2008. — т. 46, № 11. — с. 2823—2838.
140. *Kalitzin G., Medic G., Xia G.* Improvements to SST turbulence model for free shear layers, turbulent separation and stagnation point anomaly // 54th AIAA Aerospace Sciences Meeting. — 2016. — с. 1601.
141. *Rocha P. C., Rocha H. B., Carneiro F. M., Silva M. V. da, Andrade C. F. de.* A case study on the calibration of the k- $\omega$  SST (shear stress transport) turbulence model for small scale wind turbines designed with cambered and symmetrical airfoils // Energy. — 2016. — т. 97. — с. 144—150.

142. *Rocha P., Rocha H., Carneiro F., Silva M., Bueno A.* K- $\omega$  SST (shear stress transport) turbulence model calibration: A case study on a small scale horizontal axis wind turbine // *Energy*. — 2013. — янв. — т. 65. — DOI: [10.1016/j.energy.2013.11.050](https://doi.org/10.1016/j.energy.2013.11.050).
143. *Devolder B., Troch P., Rauwoens P.* Performance of a buoyancy-modified k- $\omega$  and k- $\omega$  SST turbulence model for simulating wave breaking under regular waves using OpenFOAM® // *Coastal Engineering*. — 2018. — т. 138. — с. 49—65.
144. *Devolder B., Rauwoens P., Troch P.* Application of a buoyancy-modified k- $\omega$  SST turbulence model to simulate wave run-up around a monopile subjected to regular waves using OpenFOAM // *Coastal Engineering*. — 2017. — т. 125. — с. 81—94. — DOI: [10.1016/j.coastaleng.2017.04.004](https://doi.org/10.1016/j.coastaleng.2017.04.004).
145. *Bayon A., Toro J. P., Bombardelli F. A., Matos J., López-Jiménez P. A.* Influence of VOF technique, turbulence model and discretization scheme on the numerical simulation of the non-aerated, skimming flow in stepped spillways // *Journal of hydro-environment research*. — 2018. — т. 19. — с. 137—149.

## Список рисунков

1	Примеры склоновых потоков. . . . .	5
2	Участок Барсемской селевой катастрофы на космических снимках [3]: a, d — снимки со спутника WorldView-2 от 20.09.2012; b, e — снимки со спутника Канопус-В No 1 от 06.10.2015; c, f — снимки со спутника Sentinel-2A от 07.08.2018. . . . .	7
3	Противолавинные защитные сооружения в Шамони. . . . .	7
1.1	Неньютоновские эффекты потоков на склонах гор. . . . .	18
1.2	Различные типы захвата донного материала. . . . .	20
2.1	Схема склонового потока. . . . .	25
2.2	Зависимость напряжения сдвига от скорости сдвига в простом сдвиговом потоке для различных реологических моделей. . . . .	27
2.3	Регуляризация реологической модели Хершеля-Балкли. . . . .	29
2.4	Параметр «недеформируемой» вязкости $\mu_0$ . . . . .	30
3.1	Определение свободной поверхности с помощью метода VoF. . . . .	38
3.2	Генерация расчётной сетки. . . . .	44
3.3	22-ой лавинный очаг горы Юкспор (Хибины). . . . .	45
3.4	Фото последствий схода снежной лавины из 22-ого лавинного очага горы Юкспор (Хибины). . . . .	46
3.5	Цифровая модель рельефа 22 лавинного очага с зонами зарождения лавины (зелёным) и лавинных отложений (голубой). . . . .	47
3.6	Расчёт динамики снежной лавины в 22-ом лавинном очаге горы Юкспор (Хибины). . . . .	50
3.7	Расчёт динамики снежной лавины в 22-ом лавинном очаге горы Юкспор (Хибины). . . . .	51
3.8	Схема экспериментального лотка УИ с дамбами. . . . .	52
3.9	Схема экспериментального лотка УИ без дамб. . . . .	53
3.10	Область расчета и начальные условия для моделирования эксперимента УИ с дамбами. . . . .	53
3.11	Область расчета и начальные условия для моделирования эксперимента УИ без дамб. Жёлтая линия показывает расположение сечения замера скорости потока. . . . .	54
3.12	Границы расчётной области. . . . .	55

3.13	Взаимодействие потока с тремя дамбами (результат численного моделирования, проведенного в этой работе). На плотинах наблюдаются всплески и образование струй. Цвета связаны со значениями $\alpha$ (т. е. относительной части ячейки, занятой жидкостью): синий — 1, черный — 0. Белые линии — плотины. . . . .	56
3.14	Сравнение средней по глубине потока скорости в экспериментальном лотке Университета Исландии при отсутствии дамб, в сечении, находящемся на расстоянии 11.1 м от начала желоба.	57
4.1	Разрушение и захват донного материала. . . . .	60
4.2	Вид коэффициента $\delta_{er}$ . . . . .	66
4.3	Блок-схема архитектуры решателя <code>avalancheFoam</code> . . . . .	67
4.4	Поток в лотке постоянного уклона. . . . .	68
4.5	Расчётная область численного эксперимента базального захвата в начальный момент времени. . . . .	69
4.6	Расчётная область численного эксперимента базального захвата спустя 10 с после начала расчёта. . . . .	71
4.7	Исследование размытия границ склон–захваченный материал, захваченный материал–поток, поток–воздух. . . . .	72
4.8	Исследование сеточной сходимости при различном количестве слоёв ячеек по глубине $nz_s$ в материале склона. . . . .	73
4.9	Изменение глубины слоя подстилающего материала со временем. . . . .	74
4.10	Изменение глубины слоя подстилающего материала со временем для потоков жидкостей с различным коэффициентом консистенции. . . . .	75
4.11	Изменение глубины слоя подстилающего материала со временем для потоков жидкостей с различным степенным индексом. . . . .	76
4.12	Изменение глубины слоя подстилающего материала со временем для потоков жидкостей с различным пределом текучести. . . . .	77
4.13	Карта рельефа исследуемой местности 2016 года с схематически обозначенным зелёным цветом контуром расчётной области (слева); снимок исследуемых озёр 1975 года (центр); следы прорыва западного озера, обозначенные белым цветом (справа) [131]. . . . .	78
4.14	Карта рельефа исследуемой местности с схематически обозначенным зелёным цветом контуром расчётной области [132]. . . . .	79



4.15	Карта рельефа исследуемой местности с схематически обозначенным зелёным цветом контуром расчётной области [132]. . . . .	80
4.16	Карта рельефа исследуемой местности с схематически обозначенным зелёным цветом контуром расчётной области [132]. . . . .	81
4.17	Карта эрозии слоя снега водоснежным потоком при прорыве ледникового озера Малый Азау. . . . .	83
5.1	Схема экспериментального лотка. . . . .	89
5.2	Архитектура предлагаемого алгоритма для калибровки коэффициентов $k$ - $\epsilon$ модели турбулентности . . . . .	90
5.3	Архитектура предлагаемого алгоритма для калибровки коэффициентов $k$ - $\omega$ $SST$ модели турбулентности . . . . .	90
5.4	Сравнение экспериментального профиля скорости с расчётным при использовании стандартных значений коэффициентов $k - \epsilon$ модели турбулентности и откалиброванных значений коэффициентов в лотках различного угла наклона к горизонту . . . . .	92
5.5	Сравнение экспериментального профиля скорости с расчётным при использовании стандартных значений коэффициентов $k$ - $\omega$ $SST$ модели турбулентности и откалиброванных значений коэффициентов в лотках различного угла наклона к горизонту . . . . .	93
5.6	Сравнение средней по глубине потока скорости в экспериментальном лотке Университета Исландии при отсутствии дамб, в сечении, находящемся на расстоянии 11.1 м от начала желоба. «turb» — трёхмерный подход с использованием $k$ - $\epsilon$ и $k$ - $\omega$ $SST$ моделей турбулентности, в том числе и с оптимизированными коэффициентами. . . . .	96

## Список таблиц

1	Плотность и реологические свойства жидкости и газа . . . . .	38
2	Условия на границах расчетной области в задаче расчёта снежной лавины на горе Юкспор (Хибины) . . . . .	48
3	Граничные условия в задаче расчёта снежной лавины на горе Юкспор (Хибины) . . . . .	55
4	Сравнение измеренных и рассчитанных параметров потока. . . . .	56
5	Реологические свойства в задаче захвата материала потоком в лотке постоянного уклона . . . . .	70
6	Реологические параметры сред в задаче расчета водоснежного потока при прорыве ледникового озера Малый Азау. . . . .	82
7	Параметры экспериментов . . . . .	89
8	Минимизация функции потерь . . . . .	93
9	Минимизация функции потерь . . . . .	94
10	Сравнение измеренных и рассчитанных параметров потока в эксперименте с дамбами. . . . .	95

## Приложение А

### Библиотека `avalancheUtils` для генерации расчётной сетки, начальных и граничных условий

Библиотека `avalancheUtils` включает в себя несколько python-скриптов, среди которых

- `asc2bmdict.py` — управляющий скрипт,
- `readAsc.py` — скрипт обработки ЦМР,
- `blockMeshDict.py` — скрипт создания расчётной сетки и граничных условий,
- `setFieldsDict.py` — скрипт задания начальных условий.

Управляющий скрипт представлен в листинге [А.1](#).

Листинг А.1: `asc2bmdict.py`

```
import sys
#sys.path.append(r'/home/romanovadi/cases/avalanche/preAndPostProcessing/
    readAsc')
#sys.path.append("~/cases/avalanche/preAndPostProcessing/setFieldsDict.py")
#sys.path.append("~/cases/avalanche/preAndPostProcessing/blockMeshDict.py")
5 import numpy as np
from scipy import interpolate
from operator import add
import math
import readAsc as ra
10 import setFieldsDict as sfd
import blockMeshDict as bmd

def main(argv):
    mapfile, regionfile, cellsize, dz, flowdepth, soildepth, areaheight,
        simpleGrading = ra.readFileNames(argv)
15 slope = ra.asc(mapfile, regionfile)
    slope.am, slope.rg = ra.interpolateMap(slope.am, slope.rg, cellsize)
    bmd.createBlockMeshDictInclined(slope.am, height=areaheight, dz=dz,
        simpleGrading=simpleGrading)
    sfd.createSetFieldsFourPhasesSoilRotated(slope.am, slope.rg, height =
        flowdepth, height_soil = soildepth)
    ## Dat file reading
20 #slope = ra.dat(mapfile, cellsize)
    #bmd.createBlockMeshDictInclined(slope.am, height=areaheight, dz=dz,
        simpleGrading=simpleGrading)
    #sfd.createSetFieldsFourPhasesSoilRotated(slope.am, slope.rg, height =
        flowdepth, height_soil = soildepth)
```

```

25 | if __name__ == "__main__":
    |     main(sys.argv)

```

Скрипт чтения и предобработки ЦМР представлен в листинге [A.2](#).

#### Листинг A.2: readAsc.py

```

import numpy as np
from scipy import interpolate
from operator import add
import math
5 | import sys, getopt
  | from scipy.interpolate import LinearNDInterpolator

def readFileNames(argv):
    mapfile = 'relief_22.asc'
10 | regionfile = 'region_22.asc'
    cellsize = 0
    dz = 0
    flowdepth = 5
    soildepth = 5
15 | areaheight = 20
    simpleGrading = 1
    try:
        opts, args = getopt.getopt(argv[1:], 'hm:r:c:d:f:s:z:g:', ['help', '
            mapfile=', 'regionfile=', 'cellsize=', 'dz=', 'flowdepth=', 'soildepth=',
            'areaheight=', 'simplegrading='])
    except getopt.GetoptError:
20 | print(argv[0] + ' -m <map file> -r <region file> -c <cellsize> -d <dz>
        -f <flowdepth> -s <soildepth> -z <areaheight> -g <gradingratio>')
        sys.exit(2)
    for opt, arg in opts:
        if opt == '-h':
            print(argv[0] + ' -m <map file> -r <region file> -c <cellsize> -d <dz>
                -f <flowdepth> -s <soildepth> -z <areaheight> -g <gradingratio>')
25 |         sys.exit()
        elif opt in ("-m", "--mapfile"):
            mapfile = arg
        elif opt in ("-r", "--regionfile"):
            regionfile = arg
30 |         elif opt in ("-c", "--cellsize"):
            cellsize = float(arg)
        elif opt in ("-d", "--dz"):
            dz = float(arg)
        elif opt in ("-f", "--flowdepth"):
35 |         flowdepth = float(arg)
        elif opt in ("-s", "--soildepth"):
            soildepth = float(arg)
        elif opt in ("-z", "--areaheight"):
            areaheight = float(arg)
40 |         elif opt in ("-g", "--simplegrading"):
            simpleGrading = float(arg)

```

```

print('Map file is \'' + mapfile + '\')
print('Region file is \'' + regionfile + '\')
print('Cellsize is ' + str(cellsize) + ' meters')
45 print('dz is ' + str(dz) + ' meters')
print('Depth of flow is ' + str(flowdepth) + ' meters')
print('Depth of soil cover is ' + str(soildepth) + ' meters')
print('Depth of calculation area is ' + str(areaheight) + ' meters')
print('Grading ratio is' + str(simpleGrading))
50 return mapfile, regionfile, cellsize, dz, flowdepth, soildepth,
    areaheight, simpleGrading

class altMap:
    def __init__(self, altitude, nx, ny, dx, NODATA_value):
        self.nx = nx
        55 self.ny = ny
        self.dx = dx
        self.altitude = altitude
        self.NODATA_value = NODATA_value
        self.alt_max = np.amax(self.altitude[self.altitude != self.NODATA_value
60 ])
        self.alt_min = np.amin(self.altitude[self.altitude != self.NODATA_value
        ])

class regMap:
    def __init__(self, region, nx, ny, dx, NODATA_value):
        self.nx = nx
        65 self.ny = ny
        self.dx = dx
        self.region = region
        self.NODATA_value = NODATA_value

70 def interpolateMap(mapIn, regIn, dx = -1):
    mapOut = altMap(mapIn.altitude, mapIn.nx, mapIn.ny, mapIn.dx, mapIn.
        NODATA_value)
    regOut = regMap(regIn.region, regIn.nx, regIn.ny, regIn.dx, regIn.
        NODATA_value)
    if dx == -1:
        print("Write new cell size:")
        75 mapOut.dx = input()
        if mapOut.dx == "":
            return;
        else:
            mapOut.dx = regOut.dx = float(mapOut.dx)
80 elif dx == 0:
        return mapIn, regIn
    else:
        mapOut.dx = regOut.dx = dx

85 altitude_mask = np.copy(mapIn.altitude)
f = lambda a: 0 if a == mapIn.NODATA_value else 1
fv = np.vectorize(f)

```

```

altitude_mask = fv(altitude_mask)

90  x = np.arange(0, mapIn.dx * mapIn.ny, mapIn.dx)
    y = np.arange(0, mapIn.dx * mapIn.nx, mapIn.dx)
    xnew = np.arange(0, mapIn.dx * mapIn.ny, mapOut.dx)
    ynew = np.arange(0, mapIn.dx * mapIn.nx, mapOut.dx)
    f = interpolate.interp2d(x, y, mapIn.altitude, kind='linear')
95  altitude_interpolation = f(xnew, ynew)
    f = interpolate.interp2d(x, y, altitude_mask, kind='linear')
    altitude_interpolation_mask = f(xnew, ynew)
    f = lambda a: 0 if a < 0.9999 else 1
    fv = np.vectorize(f)
100 altitude_interpolation_mask = fv(altitude_interpolation_mask)
    altitude_interpolation = altitude_interpolation *
        altitude_interpolation_mask
    f = lambda a: mapIn.NODATA_value if a == 0 else a
    fv = np.vectorize(f)
    altitude_interpolation = fv(altitude_interpolation)
105 mapOut.ny = xnew.shape[0]
    mapOut.nx = ynew.shape[0]
    mapOut.altitude = altitude_interpolation

    region_mask = np.copy(regIn.region)
110 f = lambda a: 0 if a == regIn.NODATA_value else 1
    fv = np.vectorize(f)
    region_mask = fv(region_mask)
    f = lambda a: -1 if a == 0 else a
    fv = np.vectorize(f)
115 regIn.region = fv(regIn.region)

    x = np.arange(0, regIn.dx * regIn.ny, regIn.dx)
    y = np.arange(0, regIn.dx * regIn.nx, regIn.dx)
    xnew = np.arange(0, regIn.dx * regIn.ny, regOut.dx)
120 ynew = np.arange(0, regIn.dx * regIn.nx, regOut.dx)
    f = interpolate.interp2d(x, y, regIn.region, kind='linear')
    region_interpolation = f(xnew, ynew)
    f = interpolate.interp2d(x, y, region_mask, kind='linear')
    region_interpolation_mask = f(xnew, ynew)
125 f = lambda a: 0 if a < 0.9999 else 1
    fv = np.vectorize(f)
    region_interpolation_mask = fv(region_interpolation_mask)
    region_interpolation = region_interpolation * region_interpolation_mask
# f = lambda a: 1 if a > 0 else 0 if a < 0 else regIn.NODATA_value
130 f = lambda a: regIn.NODATA_value if a == 0 else a
    fv = np.vectorize(f)
    region_interpolation = fv(region_interpolation)
    regOut.ny = xnew.shape[0]
    regOut.nx = ynew.shape[0]
135 regOut.region = region_interpolation

    return mapOut, regOut

```

```

class asc:
140 def __init__(self, map_name='', region_map_name=''):
    self.map_name = map_name
    self.region_map_name = region_map_name
    self.readMapFile()
    self.readRegionFile()
145 self.checkPair()

def readMapFile(self):
    print("Opening file: \" + self.map_name + "\")
    file_map = open(self.map_name, "r")
150
    line = file_map.readline()
    line_list = line.split()
    if line_list[0] != 'ncols':
        raise ValueError('No tag \"ncols\" in first line')
155 self.ncols = int(line_list[1])

    line = file_map.readline()
    line_list = line.split()
    if line_list[0] != 'nrows':
160 raise ValueError('No tag \"nrows\" in second line')
    self.nrows = int(line_list[1])

    line = file_map.readline()
    line_list = line.split()
165 if line_list[0] != 'xllcorner':
        raise ValueError('No tag \"xllcorner\" in third line')
    self.xllcorner = float(line_list[1].replace(",", "."))

    line = file_map.readline()
    line_list = line.split()
170 if line_list[0] != 'yllcorner':
        raise ValueError('No tag \"yllcorner\" in fourth line')
    self.yllcorner = float(line_list[1].replace(",", "."))

    line = file_map.readline()
    line_list = line.split()
175 if line_list[0] != 'cellsize':
        raise ValueError('No tag \"cellsize\" in fifth line')
    self.cellsize = float(line_list[1].replace(",", "."))
180

    line = file_map.readline()
    line_list = line.split()
    if line_list[0] != 'NODATA_value':
185 raise ValueError('No tag \"NODATA_value\" in sixth line')
    self.NODATA_value = float(line_list[1].replace(",", "."))

    altitude = np.loadtxt(file_map, dtype=np.str)
    file_map.close()

```



```

altitude = np.char.replace(altitude, ',','.'').astype(np.float32)
190 self.am = altMap(altitude, self.nrows, self.ncols, self.cellsize, self.
NODATA_value)

def readRegionFile(self):
    print("Opening file: \" + self.region_map_name + "\"")
    region_file_map = open(self.region_map_name, "r")
195
    line = region_file_map.readline()
    line_list = line.split()
    if line_list[0] != 'ncols':
        raise ValueError('No tag \"ncols\" in first line')
200 self.region_ncols = int(line_list[1])

    line = region_file_map.readline()
    line_list = line.split()
    if line_list[0] != 'nrows':
205     raise ValueError('No tag \"nrows\" in second line')
    self.region_nrows = int(line_list[1])

    line = region_file_map.readline()
    line_list = line.split()
210 if line_list[0] != 'xllcorner':
        raise ValueError('No tag \"xllcorner\" in third line')
    self.region_xllcorner = float(line_list[1].replace(",","."))

    line = region_file_map.readline()
    line_list = line.split()
215 if line_list[0] != 'yllcorner':
        raise ValueError('No tag \"yllcorner\" in fourth line')
    self.region_yllcorner = float(line_list[1].replace(",","."))

    line = region_file_map.readline()
    line_list = line.split()
220 if line_list[0] != 'cellsize':
        raise ValueError('No tag \"cellsize\" in fifth line')
    self.region_cellsize = float(line_list[1].replace(",","."))
225

    line = region_file_map.readline()
    line_list = line.split()
    if line_list[0] != 'NODATA_value':
230     raise ValueError('No tag \"NODATA_value\" in sixth line')
    self.region_NODATA_value = float(line_list[1].replace(",","."))

    region_pre = np.loadtxt(region_file_map, dtype=np.str)
    region_file_map.close()
    region_pre = np.char.replace(region_pre, ',','.'').astype(np.float32)
235 x_margin = self.nrows - self.region_nrows - int((self.region_yllcorner
- self.yllcorner) / self.cellsize)
    y_margin = int((self.region_xllcorner - self.xllcorner) / self.cellsize
)

```

```

region = np.full(self.am.altitude.shape, self.am.NODATA_value)
with np.nditer(region, flags=['multi_index'], op_flags=['writeonly'])
as it:
    while not it.finished:
240         if it.multi_index[0] >= x_margin and\
            it.multi_index[1] >= y_margin and\
            it.multi_index[0] - x_margin < self.region_nrows and\
            it.multi_index[1] - y_margin < self.region_ncols and\
            region_pre[it.multi_index[0]-x_margin, it.multi_index[1]-y_margin
] != self.region_NODATA_value:
245             it[0] = region_pre[it.multi_index[0]-x_margin, it.multi_index
[1]-y_margin]
                it.iternext()
            self.rg = regMap(region, self.nrows, self.ncols, self.cellsize, self.
NODATA_value)

def checkPair(self):
250     if not self.xllcorner <= self.region_xllcorner <= self.xllcorner + self
.ncols * self.cellsize:
        raise ValueError('Error pair of map and region map')
        if not self.yllcorner <= self.region_yllcorner <= self.yllcorner + self
.nrows * self.cellsize:
            raise ValueError('Error pair of map and region map')

255 class dat:
    def __init__(self, map_name='', cellsize = 0):
        self.map_name = map_name
        self.cellsize = cellsize
        self.readMapFile()

260     def readMapFile(self):
        data = np.loadtxt(self.map_name, usecols=(0, 1, 2), dtype=float)
        x = data.T[0]
        y = data.T[1]
265         z = data.T[2]
        self.NODATA_value = -9999
        self.ncols = int((max(x) - min(x)) / self.cellsize)
        self.nrows = int((max(y) - min(y)) / self.cellsize)
        nx = int(14.9 / self.cellsize)
270         X = np.linspace(min(x), max(x), num = self.ncols)
        Y = np.linspace(min(y), max(y), num = self.nrows)
        X, Y = np.meshgrid(X, Y) # 2D grid for interpolation
        interp = LinearNDInterpolator(list(zip(x, y)), z)
        Z = interp(X, Y)
275         Z = np.nan_to_num(Z, nan=0.0)
        np.set_printoptions(threshold=sys.maxsize)
        Zr = np.copy(Z)
        Zr[:,nx:] = self.NODATA_value
        self.am = altMap(Z, self.nrows, self.ncols, self.cellsize, self.
NODATA_value)

```

```
280 |     self.rg = regMap(Zr, self.nrows, self.ncols, self.cellsize, self.
      |     NODATA_value)
```

Скрипт создания расчётной сетки, представленный в листинге [A.3](#), позволяет построить два типа сеток:

- `createBlockMeshDictInclined` — параллельная склону сетка, когда верхняя и нижняя грани ячеек параллельны склону, а боковые грани вертикальны (ячейки в этом случае представляют из себя параллелепипеды),
- `createBlockMeshDict` — ступенчатая сетка.

Листинг A.3: `blockMeshDict.py`

```
import numpy as np
from scipy import interpolate
from operator import add
import math
5 import readAsc as ra
import sys

class BMDarr:
    def __init__(self, vertices, blocks, nx, ny, nz, dx):
10     self.vertices = vertices
        self.blocks = blocks
        self.nx = nx
        self.ny = ny
        self.nz = nz
15     self.dx = dx

def prepareSlopeData(am, height):
    f = lambda a: math.floor(a / am.dx) * am.dx if a != am.NODATA_value else
        am.NODATA_value
    fv = np.vectorize(f)
20     altitude = np.copy(am.altitude)
        altitude = fv(altitude)
        height = math.floor(height / am.dx) * am.dx
        nz = int((am.alt_max - am.alt_min + height) / am.dx)
        vertices = np.full((am.nx + 1, am.ny + 1, nz + 1), -1, dtype=np.int32)
25     blocks = np.zeros((am.nx, am.ny, nz), dtype=np.float32)
        with np.nditer(altitude, flags=['multi_index'], op_flags=['readonly']) as
            it:
                while not it.finished:
                    if it[0] != am.NODATA_value:
                        for z in range(int((it[0] - am.alt_min) / am.dx), int((it[0] - am.
30     alt_min + height) / am.dx) + 1):
                            vertices[it.multi_index[0], it.multi_index[1], z] = 1
                            vertices[it.multi_index[0] + 1, it.multi_index[1], z] = 1
                            vertices[it.multi_index[0], it.multi_index[1] + 1, z] = 1
                            vertices[it.multi_index[0] + 1, it.multi_index[1] + 1, z] = 1
```

```

        for z in range(int((it[0] - am.alt_min) / am.dx), int((it[0] - am.
alt_min + height) / am.dx)):
35         blocks[it.multi_index[0], it.multi_index[1], z] = 1
            blocks[it.multi_index[0], it.multi_index[1], int((it[0] - am.
alt_min) / am.dx)] = 2
            it.iternext()
ind = 0
for it in np.nditer(vertices, op_flags=['readwrite']):
40     if it == 1:
        it[...] = ind
        ind += 1
return BMDarr(vertices, blocks, am.nx, am.ny, nz, am.dx)

45 def createBlockMeshDict(am, height = 16):
    ba = prepareSlopeData(am, height) # blockMeshDict array
    print("Creating blockMeshDict file")
    blockMeshDictFileName = "blockMeshDict"
    file_blockMeshDict = open(blockMeshDictFileName, "w")
50     file_blockMeshDict.write("FoamFile\n{\n\tversion\t2.0;\n\tformat\tascii;\n
\tclass\tdictionary;\n\tobject\tblockMeshDict;\n}\n\nconvertToMeters
1.0;\n\n")
    file_blockMeshDict.write("vertices\n(\n\n")
    with np.nditer(ba.vertices, flags=['multi_index'], op_flags=["readonly"])
        as it:
        while not it.finished:
            if it[0] != -1:
55                 file_blockMeshDict.write("\t(%f\t%f\t%f)\n" % (it.multi_index[0] *
ba.dx, it.multi_index[1] * ba.dx, it.multi_index[2] * ba.dx))
                it.iternext()
    file_blockMeshDict.write(");\n\nblocks\n(\n\n")
    with np.nditer(ba.blocks, flags=['multi_index'], op_flags=["readonly"])
        as it:
        while not it.finished:
60             if it[0]:
                vert0 = it.multi_index
                vert1 = tuple(map(add, it.multi_index, (1, 0, 0)))
                vert2 = tuple(map(add, it.multi_index, (1, 1, 0)))
                vert3 = tuple(map(add, it.multi_index, (0, 1, 0)))
65                 vert4 = tuple(map(add, it.multi_index, (0, 0, 1)))
                vert5 = tuple(map(add, it.multi_index, (1, 0, 1)))
                vert6 = tuple(map(add, it.multi_index, (1, 1, 1)))
                vert7 = tuple(map(add, it.multi_index, (0, 1, 1)))
                file_blockMeshDict.write("\thex (%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d)\t
(1 1 1) simpleGrading (1 1 1)\n" % \
70                 (ba.vertices[vert0], ba.vertices[vert1], ba.vertices[vert2], ba.
vertices[vert3], ba.vertices[vert4], \
ba.vertices[vert5], ba.vertices[vert6], ba.vertices[vert7]))
                it.iternext()
    file_blockMeshDict.write(");\n\nedges\n(\n\n);\n\nboundary\n(\n\n\slope\n\t
{\n\t\ttype wall;\n\t\tfaces\n\t\t(\n\n")

```

```

with np.nditer(ba.blocks, flags=['multi_index'], op_flags=["readonly"])
  as it:
75   while not it.finished:
      if it[0] == 2:
          vert0 = it.multi_index
          vert1 = tuple(map(add, it.multi_index, (1, 0, 0)))
          vert2 = tuple(map(add, it.multi_index, (1, 1, 0)))
80   vert3 = tuple(map(add, it.multi_index, (0, 1, 0)))
          vert4 = tuple(map(add, it.multi_index, (0, 0, 1)))
          vert5 = tuple(map(add, it.multi_index, (1, 0, 1)))
          vert6 = tuple(map(add, it.multi_index, (1, 1, 1)))
          vert7 = tuple(map(add, it.multi_index, (0, 1, 1)))
85   file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" % (ba.vertices[
vert3], ba.vertices[vert2], ba.vertices[vert1], ba.vertices[vert0]))
          neighbour_ind = tuple(map(add, it.multi_index, (-1, 0, 0)))
          if neighbour_ind[0] < 0 or ba.blocks[neighbour_ind] == 0:
              file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" % (ba.vertices[
vert0], ba.vertices[vert4], ba.vertices[vert7], ba.vertices[vert3]))
              neighbour_ind = tuple(map(add, it.multi_index, (1, 0, 0)))
90   if neighbour_ind[0] >= ba.nx or ba.blocks[neighbour_ind] == 0:
              file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" % (ba.vertices[
vert1], ba.vertices[vert2], ba.vertices[vert6], ba.vertices[vert5]))
              neighbour_ind = tuple(map(add, it.multi_index, (0, -1, 0)))
              if neighbour_ind[1] < 0 or ba.blocks[neighbour_ind] == 0:
                  file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" % (ba.vertices[
vert0], ba.vertices[vert1], ba.vertices[vert5], ba.vertices[vert4]))
95   neighbour_ind = tuple(map(add, it.multi_index, (0, 1, 0)))
              if neighbour_ind[1] >= ba.ny or ba.blocks[neighbour_ind] == 0:
                  file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" % (ba.vertices[
vert2], ba.vertices[vert3], ba.vertices[vert7], ba.vertices[vert6]))
                  neighbour_ind = tuple(map(add, it.multi_index, (0, 0, 1)))
                  if neighbour_ind[2] >= ba.nz or ba.blocks[neighbour_ind] == 0:
100   file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" % (ba.vertices[
vert4], ba.vertices[vert5], ba.vertices[vert6], ba.vertices[vert7]))
              it.iternext()
file_blockMeshDict.write("\t\t); \n\t} \n\tatmosphere \n\t{ \n\t\ttype patch
; \n\t\tfaces \n\t\t(\n")
with np.nditer(ba.blocks, flags=['multi_index'], op_flags=["readonly"])
  as it:
105   while not it.finished:
      if it[0] == 1:
          vert0 = it.multi_index
          vert1 = tuple(map(add, it.multi_index, (1, 0, 0)))
          vert2 = tuple(map(add, it.multi_index, (1, 1, 0)))
          vert3 = tuple(map(add, it.multi_index, (0, 1, 0)))
110   vert4 = tuple(map(add, it.multi_index, (0, 0, 1)))
          vert5 = tuple(map(add, it.multi_index, (1, 0, 1)))
          vert6 = tuple(map(add, it.multi_index, (1, 1, 1)))
          vert7 = tuple(map(add, it.multi_index, (0, 1, 1)))
          neighbour_ind = tuple(map(add, it.multi_index, (-1, 0, 0)))
115   if neighbour_ind[0] < 0 or ba.blocks[neighbour_ind] == 0:

```

```

        file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" % (ba.vertices[
vert0], ba.vertices[vert4], ba.vertices[vert7], ba.vertices[vert3]))
        neighbour_ind = tuple(map(add, it.multi_index, (1, 0, 0)))
        if neighbour_ind[0] >= ba.nx or ba.blocks[neighbour_ind] == 0:
            file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" % (ba.vertices[
vert1], ba.vertices[vert2], ba.vertices[vert6], ba.vertices[vert5]))
120         neighbour_ind = tuple(map(add, it.multi_index, (0, -1, 0)))
            if neighbour_ind[1] < 0 or ba.blocks[neighbour_ind] == 0:
                file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" % (ba.vertices[
vert0], ba.vertices[vert1], ba.vertices[vert5], ba.vertices[vert4]))
                neighbour_ind = tuple(map(add, it.multi_index, (0, 1, 0)))
                if neighbour_ind[1] >= ba.ny or ba.blocks[neighbour_ind] == 0:
125                 file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" % (ba.vertices[
vert2], ba.vertices[vert3], ba.vertices[vert7], ba.vertices[vert6]))
                neighbour_ind = tuple(map(add, it.multi_index, (0, 0, 1)))
                if neighbour_ind[2] >= ba.nz or ba.blocks[neighbour_ind] == 0:
                    file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" % (ba.vertices[
vert4], ba.vertices[vert5], ba.vertices[vert6], ba.vertices[vert7]))
                    it.iternext()
130 file_blockMeshDict.write("\t\t);\n\t}\n);\n\nmergePatchPairs\n(\n);\n")
file_blockMeshDict.close()
print("blockMeshDict file is ready")

def createBlockMeshDictInclined(am, height = 20, dz = 0, simpleGrading = 1)
:
135 if dz == 0:
        dz = am.dx
        print("Creating blockMeshDict file")
        blockMeshDictFileName = "blockMeshDict"
        file_blockMeshDict = open(blockMeshDictFileName, "w")
140 alt_ind = np.zeros_like(am.altitude)
        i = 0
        with np.nditer(am.altitude, flags=['multi_index'], op_flags=['readonly'])
            as it:
                while not it.finished:
                    if it[0] != am.NODATA_value and\
145                     it.multi_index[0]+1 < am.nx and\
                        it.multi_index[1]+1 < am.ny and\
                            am.altitude[it.multi_index[0]+1,it.multi_index[1]] != am.
NODATA_value and\
                                am.altitude[it.multi_index[0],it.multi_index[1]+1] != am.
NODATA_value and\
                                    am.altitude[it.multi_index[0]+1,it.multi_index[1]+1] != am.
NODATA_value:
150                     alt_ind[it.multi_index] = alt_ind[it.multi_index[0]+1,it.
multi_index[1]] =\
                            alt_ind[it.multi_index[0],it.multi_index[1]+1] = alt_ind[it.
multi_index[0]+1,it.multi_index[1]+1] = 1
                            it.iternext()
n_vertices = 0
for it in np.nditer(alt_ind, op_flags=['readwrite']):

```

```

155     if it == 0:
        it[...] = -1
    else:
        it[...] = n_vertices
        n_vertices += 1
160 file_blockMeshDict.write("FoamFile\n{\n\tversion\t2.0;\n\tformat\tascii;\n\tclass\tdictionary;\n\tobject\tblockMeshDict;\n}\n\nconvertToMeters\n1.0;\n\n")
file_blockMeshDict.write("vertices\n(\n\n")
with np.nditer(alt_ind, flags=['multi_index'], op_flags=['readonly']) as it:
    while not it.finished:
        if it[0] > -1:
165             file_blockMeshDict.write("\t(%f\t%f\t%f)\n" % (am.dx * it.
multi_index[0], am.dx * it.multi_index[1],\
                am.altitude[it.multi_index] - am.alt_min))
            file_blockMeshDict.write("\t(%f\t%f\t%f)\n" % (am.dx * it.
multi_index[0], am.dx * it.multi_index[1],\
                am.altitude[it.multi_index] - am.alt_min + height))
            it.iternext()
170 file_blockMeshDict.write(");\n\nblocks\n(\n\n")
n_layers = 2
l1 = 0
l2 = 1
nz = math.ceil(height / dz)
175 with np.nditer(alt_ind, flags=['multi_index'], op_flags=['readonly']) as it:
    while not it.finished:
        vert0 = it.multi_index
        vert1 = tuple(map(add, it.multi_index, (1, 0)))
        vert2 = tuple(map(add, it.multi_index, (1, 1)))
180         vert3 = tuple(map(add, it.multi_index, (0, 1)))
        if it[0] > -1 and\
            it.multi_index[0]+1 < am.nx and\
            it.multi_index[1]+1 < am.ny and\
            alt_ind[vert1] != -1 and\
185             alt_ind[vert2] != -1 and\
            alt_ind[vert3] != -1:
            file_blockMeshDict.write("\thex (%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d)\n\t(1 1 %d) simpleGrading (1 1 %f)\n" % \
                (alt_ind[vert0] * n_layers+l1, alt_ind[vert1] * n_layers+l1,
alt_ind[vert2] * n_layers+l1, alt_ind[vert3] * n_layers+l1,\
                alt_ind[vert0] * n_layers + l2, alt_ind[vert1] * n_layers + l2,
alt_ind[vert2] * n_layers+l2, alt_ind[vert3] * n_layers+l2,\
190                 nz, simpleGrading))
            it.iternext()
file_blockMeshDict.write(");\n\nedges\n(\n\n);\n\nboundary\n(\n\n\tslope\n\t{\n\t\ttype wall;\n\t\tfaces\n\t\t\t(\n\n")
with np.nditer(alt_ind, flags=['multi_index'], op_flags=['readonly']) as it:
    while not it.finished:

```



```

195     vert0 = it.multi_index
        vert1 = tuple(map(add, it.multi_index, (1, 0)))
        vert2 = tuple(map(add, it.multi_index, (1, 1)))
        vert3 = tuple(map(add, it.multi_index, (0, 1)))
        if it[0] > -1 and\
200         it.multi_index[0]+1 < am.nx and\
            it.multi_index[1]+1 < am.ny and\
                alt_ind[vert1] != -1 and\
                alt_ind[vert2] != -1 and\
                alt_ind[vert3] != -1:
205             file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" %\
                (alt_ind[vert3] * n_layers+11, alt_ind[vert2] * n_layers+11,
                alt_ind[vert1] * n_layers+11, alt_ind[vert0] * n_layers+11))
            it.iternext()
file_blockMeshDict.write("\t\t);\n\t}\n\tatmosphere\n\t{\n\t\ttype patch
;\n\t\t\tfaces\n\t\t\t(\n")
with np.nditer(alt_ind, flags=['multi_index'], op_flags=['readonly']) as
it:
210     while not it.finished:
            vert0 = it.multi_index
            vert1 = tuple(map(add, it.multi_index, (1, 0)))
            vert2 = tuple(map(add, it.multi_index, (1, 1)))
            vert3 = tuple(map(add, it.multi_index, (0, 1)))
215         if it[0] > -1 and\
            it.multi_index[0]+1 < am.nx and\
                it.multi_index[1]+1 < am.ny and\
                    alt_ind[vert1] != -1 and\
                    alt_ind[vert2] != -1 and\
220                    alt_ind[vert3] != -1:
                file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" %\
                (alt_ind[vert0] * n_layers+12, alt_ind[vert1] * n_layers+12,
                alt_ind[vert2] * n_layers+12, alt_ind[vert3] * n_layers+12))
                it.iternext()
file_blockMeshDict.write("\t\t);\n\t}\n\t\sides\n\t{\n\t\t\ttype wall;\n\t\t\t
\t\t\t\tfaces\n\t\t\t\t(\n")
225 with np.nditer(alt_ind, flags=['multi_index'], op_flags=['readonly']) as
it:
        while not it.finished:
            vert0 = it.multi_index
            vert1 = tuple(map(add, it.multi_index, (1, 0)))
            vert2 = tuple(map(add, it.multi_index, (1, 1)))
230            vert3 = tuple(map(add, it.multi_index, (0, 1)))
            if it[0] > -1 and\
                it.multi_index[0]+1 < am.nx and\
                    it.multi_index[1]+1 < am.ny and\
                        alt_ind[vert1] != -1 and\
                        alt_ind[vert2] != -1 and\
235                        alt_ind[vert3] != -1:
                            if it.multi_index[1]-1 < 0 or\
                                alt_ind[it.multi_index[0],it.multi_index[1]-1] == -1 or\
                                    alt_ind[it.multi_index[0]+1,it.multi_index[1]-1] == -1:

```

```

240         file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" %\
            (alt_ind[vert0] * n_layers+11, alt_ind[vert1] * n_layers+11
, alt_ind[vert1] * n_layers+12, alt_ind[vert0] * n_layers+12))
            if it.multi_index[0]-1 < 0 or\
                alt_ind[it.multi_index[0]-1,it.multi_index[1]] == -1 or\
                alt_ind[it.multi_index[0]-1,it.multi_index[1]+1] == -1:
245         file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" %\
            (alt_ind[vert0] * n_layers+11, alt_ind[vert0] * n_layers+12
, alt_ind[vert3] * n_layers+12, alt_ind[vert3] * n_layers+11))
            if it.multi_index[1]+2 >= am.ny or\
                alt_ind[it.multi_index[0],it.multi_index[1]+2] == -1 or\
                alt_ind[it.multi_index[0]+1,it.multi_index[1]+2] == -1:
250         file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" %\
            (alt_ind[vert3] * n_layers+11, alt_ind[vert3] * n_layers+12
, alt_ind[vert2] * n_layers+12, alt_ind[vert2] * n_layers+11))
            if it.multi_index[0]+2 >= am.nx or\
                alt_ind[it.multi_index[0]+2,it.multi_index[1]] == -1 or\
                alt_ind[it.multi_index[0]+2,it.multi_index[1]+1] == -1:
255         file_blockMeshDict.write("\t\t\t(%d %d %d %d)\n" %\
            (alt_ind[vert1] * n_layers+11, alt_ind[vert2] * n_layers+11
, alt_ind[vert2] * n_layers+12, alt_ind[vert1] * n_layers+12))
            it.iternext()
file_blockMeshDict.write("\t\t);\n\t}\n);\n\nmergePatchPairs\n(\n);\n")
file_blockMeshDict.close()
260 print("blockMeshDict file is ready")

```

Скрипт задания начальных данных, представленный в листинге [A.4](#), позволяет задать следующие типы начальных данных для соответствующих задач:

- `createSetFields` — задаёт слой снега (или массы потока) заданной толщины в зоне зарождения в задаче, не учитывающей захват (сетка ступенчатая) с точностью до разрешения сетки;
- `createSetFieldsRotated` — задаёт слой снега (или массы потока) заданной толщины в зоне зарождения в задаче, не учитывающей захват (сетка параллельна склону) с точностью до разрешения сетки;
- `createSetFieldsRotatedStartFinish` — задаёт слой снега (или массы потока) заданной толщины в зоне зарождения в задаче, не учитывающей захват (сетка параллельна склону) с точностью до разрешения сетки, на сетке также отображается зона зарождения и зона лавинных отложений;
- `createSetFieldsRotatedHeight` — задаёт слой снега (или массы потока) заданной толщины в зоне зарождения в задаче, не учитывающей захват (сетка параллельна склону) независимо от разрешения сетки;



```

35 print("Creating setFieldsDict file.")
file = open("setFieldsDict", "w")
file.write('FoamFile\n{\n\tversion\t2.0;\n\tformat\tascii;\n\tclass\
tdictionary;\n\tlocation\t"system";\n\tobject\tsetFieldsDict;\n}\n')
file.write('defaultFieldValues\n(\n\tvolScalarFieldValue alpha.water 0\n)
;\nregions\n(\n')
indent = 0
40 with np.nditer(rg.region, flags=['multi_index'], op_flags=['readonly'])
as it:
while not it.finished:
vert0 = it.multi_index
vert1 = tuple(map(add, it.multi_index, (1, 0)))
vert2 = tuple(map(add, it.multi_index, (1, 1)))
45 vert3 = tuple(map(add, it.multi_index, (0, 1)))
if it[0] != am.NODATA_value and\
it.multi_index[0]+1 < am.nx and\
it.multi_index[1]+1 < am.ny and\
rg.region[vert1] != am.NODATA_value and\
50 rg.region[vert2] != am.NODATA_value and\
rg.region[vert3] != am.NODATA_value:
file.write('\trotatedBoxToCell\n\t{\n\t\ttorigin (%f %f %f);\n\t\t
ti (%f %f %f);\n\t\t\tj (%f %f %f);\n\t\t\tk (%f %f %f);\n\t\t\tfieldValues\n\t
\t(\n\t\t\t\t\tvolScalarFieldValue alpha.water 1\n\t\t\t\t);\n\t\t}\n' % (\
it.multi_index[0] * am.dx, it.multi_index[1] * am.dx, am.
altitude[vert0] - am.alt_min + indent,\
am.dx, 0, am.altitude[vert1] - am.altitude[vert0],\
55 0, am.dx, am.altitude[vert3] - am.altitude[vert0],\
0, 0, height))
it.iternext()
file.write(');')
file.close()
60 print("setFieldsDict file is ready")

def createSetFieldsRotatedHeight(am, rg): #rg - region map, height - height
of snow cover
print("Creating setFieldsDict file.")
file = open("setFieldsDict", "w")
65 file.write('FoamFile\n{\n\tversion\t2.0;\n\tformat\tascii;\n\tclass\
tdictionary;\n\tlocation\t"system";\n\tobject\tsetFieldsDict;\n}\n')
file.write('defaultFieldValues\n(\n\tvolScalarFieldValue alpha.water 0\n)
;\nregions\n(\n')
indent = 0
height = 0
with np.nditer(rg.region, flags=['multi_index'], op_flags=['readonly'])
as it:
70 while not it.finished:
vert0 = it.multi_index
vert1 = tuple(map(add, it.multi_index, (1, 0)))
vert2 = tuple(map(add, it.multi_index, (1, 1)))
vert3 = tuple(map(add, it.multi_index, (0, 1)))
75 if it[0] != am.NODATA_value and\

```



```

alpha.soil 0\n\t\t\tvolScalarFieldValue alpha.entrained 0\n\t\t\tvolScalarFi
alpha.gas 0\n\t\t\t);\n\t}\n' % (\
    it.multi_index[0] * am.dx, it.multi_index[1] * am.dx, am.
altitude[vert0] - am.alt_min + indent + soil_indent,\
    am.dx, 0, am.altitude[vert1] - am.altitude[vert0],\
    0, am.dx, am.altitude[vert3] - am.altitude[vert0],\
120     0, 0, height))
    file.write('\trotatedBoxToCell\n\t\t{\n\t\t\torigin (%f %f %f);\n\t\t\
ti (%f %f %f);\n\t\t\tj (%f %f %f);\n\t\t\tk (%f %f %f);\n\t\t\tfieldValues\n\t\t\t\t(\n\t\t\t\t\tvolScalarFieldValue alpha.liquid 0\n\t\t\t\t\tvolScalarFieldValue
alpha.soil 1\n\t\t\t\t\tvolScalarFieldValue alpha.entrained 0\n\t\t\t\t\tvolScalarFi
alpha.gas 0\n\t\t\t\t\t);\n\t\t}\n' % (\
    it.multi_index[0] * am.dx, it.multi_index[1] * am.dx, am.
altitude[vert0] - am.alt_min + indent,\
    am.dx, 0, am.altitude[vert1] - am.altitude[vert0],\
    0, am.dx, am.altitude[vert3] - am.altitude[vert0],\
125     0, 0, soil_indent))
    it.iternext()
file.write(');')
file.close()
print("setFieldsDict file is ready")
130
def createSetFieldsRotatedStartFinish(am, rg, height = 2): #rg - region map
    , height - height of snow cover
    print("Creating setFieldsDict file.")
    file = open("setFieldsDict", "w")
    file.write('FoamFile\n{\n\tversion\t2.0;\n\tformat\tascii;\n\tclass\
tdictionary;\n\tlocation\t"system";\n\tobject\tsetFieldsDict;\n}\n')
135 file.write('defaultFieldValues\n(\n\tvolScalarFieldValue alpha.water 0\n\t\t\t\t\tvolScalarFieldValue region 0\n);\nregions\n(\n\t\t\t\t\t)')
    indent = 0
    with np.nditer(rg.region, flags=['multi_index'], op_flags=['readonly'])
    as it:
        while not it.finished:
            vert0 = it.multi_index
            vert1 = tuple(map(add, it.multi_index, (1, 0)))
            vert2 = tuple(map(add, it.multi_index, (1, 1)))
            vert3 = tuple(map(add, it.multi_index, (0, 1)))
            if it[0] == 0 and\
            it.multi_index[0]+1 < am.nx and\
            it.multi_index[1]+1 < am.ny and\
            rg.region[vert1] == 0 and\
            rg.region[vert2] == 0 and\
            rg.region[vert3] == 0:
                file.write('\trotatedBoxToCell\n\t\t{\n\t\t\torigin (%f %f %f);\n\t\t\
ti (%f %f %f);\n\t\t\tj (%f %f %f);\n\t\t\tk (%f %f %f);\n\t\t\tfieldValues\n\t\t\t\t(\n\t\t\t\t\tvolScalarFieldValue alpha.water 1\n\t\t\t\t\t);\n\t\t}\n' % (\
150     it.multi_index[0] * am.dx, it.multi_index[1] * am.dx, am.
altitude[vert0] - am.alt_min + indent,\
    am.dx, 0, am.altitude[vert1] - am.altitude[vert0],\
    0, am.dx, am.altitude[vert3] - am.altitude[vert0],\

```

```

        0, 0, height))
if it[0] == 1 and\
155     it.multi_index[0]+1 < am.nx and\
        it.multi_index[1]+1 < am.ny and\
        rg.region[vert1] == 1 and\
        rg.region[vert2] == 1 and\
        rg.region[vert3] == 1:
160     file.write('\trrotatedBoxToCell\n\t{\n\t\t\torigin (%f %f %f);\n\t\t\
ti (%f %f %f);\n\t\t\tj (%f %f %f);\n\t\t\tk (%f %f %f);\n\t\t\tffieldValues\n\t\t\t(\n\t\t\t\t\
tvolScalarFieldValue region 1\n\t\t\t);\n\t\t}\n' % (\
        it.multi_index[0] * am.dx, it.multi_index[1] * am.dx, am.
altitude[vert0] - am.alt_min + indent,\
        am.dx, 0, am.altitude[vert1] - am.altitude[vert0],\
        0, am.dx, am.altitude[vert3] - am.altitude[vert0],\
        0, 0, height))
165     it.iternext()
file.write(');')
file.close()
print("setFieldsDict file is ready")

```

## Приложение Б

### Код решателя `avalancheFoam` для многофазного потока неньютоновской среды с учетом захвата донного материала

Для создания решателя `avalancheFoam` на базе решателя `multiphaseInterFoam` скопируем директорию с исходным решателем в отдельный каталог, например, домашний.

```
cp -r $FOAM_INSTALL_DIR/applications/solvers/multiphase/
multiphaseInterFoam ~/avalancheFoam
```

Далее копируем директорию с алгоритмами для расчёта интерфейса в новый решатель:

```
cp -r $FOAM_INSTALL_DIR/applications/solvers/multiphase/VoF ~/
avalancheFoam/
```

Скопируем в директорию нового решателя файлы, импортируемые из решателя `interFoam`

```
cp -r $FOAM_INSTALL_DIR/applications/solvers/multiphase/interFoam/
correctPhi.H ~/avalancheFoam/
cp -r $FOAM_INSTALL_DIR/applications/solvers/multiphase/interFoam/
initCorrectPhi.H ~/avalancheFoam/
cp -r $FOAM_INSTALL_DIR/applications/solvers/multiphase/interFoam/pEqn.
H ~/avalancheFoam/
cp -r $FOAM_INSTALL_DIR/applications/solvers/multiphase/interFoam/UEqn.
H ~/avalancheFoam/
```

Далее создаем в директории решателя `multiphaseMixture/` подкаталог `massTransferModels/`, состоящий в свою очередь из двух подкаталогов `massTransferModel/` и `entrainment/`. В подкаталог `massTransferModel/` помещаются файлы `massTransferModel.H` и `massTransferModel.C` представленные в листингах [Б.1](#) и [Б.2](#). В подкаталог `entrainment/` помещаются файлы `entrainment.H` и `entrainment.C` представленные в листингах [Б.3](#) и [Б.4](#).

Листинг Б.1: `massTransferModel.H`

```
#ifndef massTransferModel_H
#define massTransferModel_H
// * * * * *
* //
#include "dictionary.H"
5 #include "phase.H"
#include "runTimeSelectionTables.H"
```



```

namespace Foam
{
10 /*
-----*|
                                     Class massTransferModel Declaration
\*-----
    */
class massTransferModel
{
15 protected:
    // Protected Data
    const dictionary& interfaceDict_;
    const phase& phase1_; //this phase affect
    const phase& phase2_; //this phase is the object of affection
20 dimensionedScalar residualPhaseFraction_;
public:
    //- Runtime type information
    TypeName("massTransferModel");
    // Declare runtime construction
25 declareRunTimeSelectionTable
    (
        autoPtr,
        massTransferModel,
        dictionary,
30     (
        const dictionary& interfaceDict,
        const phase& phase1,
        const phase& phase2
        ),
35     (interfaceDict, phase1, phase2)
    );
    // Constructors
    massTransferModel
    (
40     const dictionary& interfaceDict,
        const phase& phase1,
        const phase& phase2
    );
    //- Destructor
45 virtual ~massTransferModel() = default;
    // Selectors
    static autoPtr<massTransferModel> New
    (
50     const dictionary& intefaceDict,
        const phase& phase1,
        const phase& phase2
    );
    // Member Functions
    const phase& phase1() const

```

```

55     {
        return phase1_;
    }

    const phase& phase2() const
60     {
        return phase2_;
    }

    const dimensionedScalar& residualPhaseFraction() const
65     {
        return residualPhaseFraction_;
    }

    virtual tmp<volScalarField> K() const = 0;
70 };
} // End namespace Foam
// * * * * *
* //
#endif

```

### ЛИСТИНГ Б.2: massTransferModel.C

```

#include "massTransferModel.H"

// * * * * * Static Data Members * * * * *
* //
namespace Foam
5 {
    defineTypeNameAndDebug(massTransferModel, 0);
    defineRunTimeSelectionTable(massTransferModel, dictionary);
}

10 // * * * * * Constructors * * * * *
* //
Foam::massTransferModel::massTransferModel
(
    const dictionary& interfaceDict,
    const phase& phase1,
15     const phase& phase2
)
:
    interfaceDict_(interfaceDict),
    phase1_(phase1),
20     phase2_(phase2),
    residualPhaseFraction_("residualPhaseFraction", dimless, interfaceDict)
{}

// * * * * * Selectors * * * * *
* //
25 Foam::autoPtr<Foam::massTransferModel> Foam::massTransferModel::New

```

```

(
  const dictionary& interfaceDict,
  const phase& phase1,
  const phase& phase2
30 )
{
  const word modelType(interfaceDict.get<word>("type"));

  Info<< "Selecting massTransferModel for phase "
35   << phase1.name()
   << ": "
   << modelType << endl;

  auto* ctorPtr = dictionaryConstructorTable(modelType);

40
  if (!ctorPtr)
  {
    FatalIOErrorInLookup
    (
45     interfaceDict,
     "massTransferModel",
     modelType,
     *dictionaryConstructorTablePtr_
    ) << exit(FatalIOError);
50 }

  return ctorPtr
  (
    interfaceDict,
55   phase1,
    phase2
  );
}
//
// *****
//

```

Листинг Б.3: entrainment.H

```

#ifndef entrainment_H
#define entrainment_H

#include "massTransferModel.H"
5
// * * * * *
// * //

namespace Foam
{
10 namespace massTransferModels
{

```

```

/*
-----*|
                                     Class entrainment Declaration
15 |*-----
    */

class entrainment
:
    public massTransferModel
20 {
    // Private data
    const dictionary& dict_;
    dimensionedScalar entrCoeff_;
    dimensionedScalar breakingPoint_;
25 public:
    //- Runtime type information
    TypeName("entrainment");

    // Constructors
30 //- Construct from components
    entrainment
    (
        const dictionary& interfaceDict,
        const phase& phase1,
35         const phase& phase2
    );

    //- Destructor
40 virtual ~entrainment() = default;

    // Member Functions
    tmp<volScalarField> K() const;
};
} // End namespace massTransferModels
45 } // End namespace Foam

// * * * * *
* //
#endif

```

Листинг Б.4: entrainment.C

```

#include "entrainment.H"
#include "addToRunTimeSelectionTable.H"
// * * * * * Static Data Members * * * * *
* //
namespace Foam
5 {
namespace massTransferModels

```

```

{
    defineTypeNameAndDebug(entrainment, 0);
10    addToRunTimeSelectionTable
        (
            massTransferModel,
            entrainment,
            dictionary
15        );
}
}
// * * * * * Constructors * * * * *
* //
Foam::massTransferModels::entrainment::entrainment
20 (
    const dictionary& interfaceDict,
    const phase& phase1, //soil
    const phase& phase2 //entrained
)
25 :
    massTransferModel(interfaceDict, phase1, phase2),
    dict_(interfaceDict.optionalSubDict("entrainmentCoeffs")),
    entrCoeff_("entrCoeff", dimless, dict_),
    breakingPoint_("breakingPoint", dimless, dict_)
30 {
    Info << "entrCoeff = " << entrCoeff_ << endl;
    Info << "breakingPoint = " << breakingPoint_ << endl;
}
// * * * * * Member Functions * * * * *
* //
35 //phase1 - from, phase2 - to
Foam::tmp<Foam::volScalarField> Foam::massTransferModels::entrainment::K()
    const
{
    tmp<volScalarField> limitedSpecificStrainRate = phase1_.
        specificStrainRate();
    tmp<volScalarField> alpha1 = phase1_;
40 tmp<volScalarField> erDepth = phase1_.magU() / (dimensionedScalar(dimless
    /dimTime, SMALL) + limitedSpecificStrainRate);
    limitedSpecificStrainRate = \
        Foam::min(\
            Foam::max(\
                (limitedSpecificStrainRate - breakingPoint_*dimensionedScalar(
                    dimless/dimTime, 1.0)) * GREAT,\
45                dimensionedScalar(dimless/dimTime, 0.0)\
            ),\
            dimensionedScalar(dimless/dimTime, 1.0)\
        );
    tmp<volScalarField> Kvalue(
50        erDepth * limitedSpecificStrainRate * entrCoeff_ *
        dimensionedScalar(dimless/dimLength, 1.0)

```

```

    );
    return Kvalue;
}
//
// *****
/

```

В файл `avalancheFoam/multiphaseMixture/Make/files` добавляются новые файлы для компиляции, как показано в листинге [Б.5](#).

Листинг Б.5: `filesMultiphaseMixture`

```

phase/phase.C
alphaContactAngle/alphaContactAngleFvPatchScalarField.C
multiphaseMixture.C
5 massTransfer = massTransferModels
$(massTransfer)/massTransferModel/massTransferModel.C
$(massTransfer)/entrainment/entrainment.C
LIB = $(FOAM_USER_LIBBIN)/libavalancheMultiphaseMixture

```

Вносим изменения в класс, описывающий фазу. Изменения связаны с добавлением поля второго инварианта тензора скоростей деформаций и поля `gFlag`, задающего неподвижность выбранной фазы. Конечный вид файла `phase.H` представлены в листинге [Б.6](#). Конечный вид файла `phase.C` представлены в листинге [Б.7](#).

Листинг Б.6: `phase.H`

```

#ifndef phase_H
#define phase_H

#include "volFields.H"
5 #include "dictionaryEntry.H"
#include "incompressible/viscosityModels/viscosityModel/viscosityModel.H"

namespace Foam
{
10 class phase
:
    public volScalarField
{
    // Private data
15     word name_;
    dictionary phaseDict_;
    autoPtr<viscosityModel> nuModel_;
    dimensionedScalar rho_;
    dimensionedScalar gFlag_;
20     const volVectorField& U_;

```

```

    volScalarField magU_;
    volScalarField magGradAlpha_;
    volScalarField magGradStrainRate_;
    volScalarField specificStrainRate_;
25 public:
    // Constructors
    // - Construct from components
    phase
    (
30     const word& name,
        const dictionary& phaseDict,
        const volVectorField& U,
        const surfaceScalarField& phi
    );
    // - Return clone
    autoPtr<phase> clone() const;
    // - Return a pointer to a new phase created on freestore
    // from Istream
    class iNew
40     {
        const volVectorField& U_;
        const surfaceScalarField& phi_;
    public:
        iNew
45     (
            const volVectorField& U,
            const surfaceScalarField& phi
        )
        :
50     U_(U),
        phi_(phi)
    {}
    autoPtr<phase> operator()(Istream& is) const
    {
55     dictionaryEntry ent(dictionary::null, is);
        return autoPtr<phase>(new phase(ent.keyword(), ent, U_,
phi_));
    }
};
// Member Functions
60     const word& name() const
    {
        return name_;
    }
    const word& keyword() const
65     {
        return name();
    }
    const viscosityModel& nuModel() const
    {
70     return *nuModel_;
    }

```

```
    }
    tmp<volScalarField> nu() const
    {
        return nuModel_->nu();
75    }
    tmp<scalarField> nu(const label patchi) const
    {
        return nuModel_->nu(patchi);
    }
    const dimensionedScalar& rho() const
    {
        return rho_;
    }
    const dimensionedScalar& gFlag() const
85    {
        return gFlag_;
    }
    const volScalarField& magU() const
    {
90    return magU_;
    }
    volScalarField& magU()
    {
        return magU_;
    }
95    const volScalarField& magGradAlpha() const
    {
        return magGradAlpha_;
    }
100    volScalarField& magGradAlpha()
    {
        return magGradAlpha_;
    }
    const volScalarField& magGradStrainRate() const
105    {
        return magGradStrainRate_;
    }
    volScalarField& magGradStrainRate()
    {
110    return magGradStrainRate_;
    }
    const volScalarField& specificStrainRate() const
    {
        return specificStrainRate_;
115    }
    volScalarField& specificStrainRate()
    {
        return specificStrainRate_;
    }
120    void correct();
    void calcMagU();
```



```

        void calcMagGradAlpha();
        void calcSpecificStrainRate();
        using volScalarField::read;
125     bool read(const dictionary& phaseDict);
    };
} // End namespace Foam
#endif

```

## Листинг Б.7: phase.C

```

#include "phase.H"
#include "fvcGrad.H"

// * * * * * Constructors * * * * *
* //
5 Foam::phase::phase
(
    const word& phaseName,
    const dictionary& phaseDict,
    const volVectorField& U,
10    const surfaceScalarField& phi
)
:
    volScalarField
    (
15        IObject
        (
            IObject::groupName("alpha", phaseName),
            U.mesh().time().timeName(),
            U.mesh(),
20            IObject::MUST_READ,
            IObject::AUTO_WRITE
        ),
        U.mesh()
    ),
    name_(phaseName),
    phaseDict_(phaseDict),
    nuModel_
    (
30        viscosityModel::New
        (
            IObject::groupName("nu", phaseName),
            phaseDict_,
            U,
            phi
35        )
    ),
    rho_("rho", dimDensity, phaseDict_),
    gFlag_("gFlag", dimless, phaseDict_),
    U_(U),
40    magU_

```

```

(
  IOobject
  (
    IOobject::groupName("magU", phaseName),
45    U.mesh().time().timeName(),
    U.mesh(),
    IOobject::NO_READ,
    IOobject::AUTO_WRITE
  ),
50  U.mesh(),
  dimLength/dimTime
),
magGradAlpha_
(
55  IOobject
  (
    IOobject::groupName("magGradAlpha", phaseName),
    U.mesh().time().timeName(),
    U.mesh(),
60    IOobject::NO_READ,
    IOobject::AUTO_WRITE
  ),
  U.mesh(),
  dimless
65 ),
magGradStrainRate_
(
  IOobject
  (
70    IOobject::groupName("magGradStrainRate", phaseName),
    U.mesh().time().timeName(),
    U.mesh(),
    IOobject::NO_READ,
    IOobject::AUTO_WRITE
75  ),
  U.mesh(),
  dimless/(dimTime*dimLength)
),
specificStrainRate_
80 (
  IOobject
  (
    IOobject::groupName("specificStrainRate", phaseName),
    U.mesh().time().timeName(),
85    U.mesh(),
    IOobject::NO_READ,
    IOobject::AUTO_WRITE
  ),
  U.mesh(),
90  dimless/dimTime
)

```

```

{}
// * * * * * Member Functions * * * * *
* //
Foam::autoPtr<Foam::phase> Foam::phase::clone() const
95 {
    NotImplemented;
    return nullptr;
}
void Foam::phase::calcMagU()
100 {
    magU_ = mag(U_);
}
void Foam::phase::calcMagGradAlpha()
{
105     volScalarField& alpha = *this;
    magGradAlpha_ = mag(fvc::grad(alpha))*
        dimensionedScalar(dimensionSet(0,1,0,0,0),1);
}
void Foam::phase::calcSpecificStrainRate()
110 {
    calcMagU();
    calcMagGradAlpha();
    specificStrainRate_ = nuModel_->strainRate();
    magGradStrainRate_ = mag(fvc::grad(specificStrainRate_));
115 }
void Foam::phase::correct()
{
    nuModel_->correct();
}
120 bool Foam::phase::read(const dictionary& phaseDict)
{
    phaseDict_ = phaseDict;
    if (nuModel_->read(phaseDict_))
    {
125         phaseDict_.readEntry("rho", rho_);
        phaseDict_.readEntry("gFlag", gFlag_);
        return true;
    }
    return false;
130 }

```

Также необходимо внести изменения в класс `multiphaseMixture`. Изменения внесенные в данный класс описывают фазовый переход. Изменения, внесенные в файл `multiphaseMixture.H` показаны в листинге [Б.8](#). Изменения, внесенные в файл `multiphaseMixture.C` показаны в листинге [Б.9](#).

#### Листинг Б.8: `multiphaseMixture.H`

```

--- multiphaseMixtureOrig.H 2022-10-06 21:31:58.364555019 +0300
+++ multiphaseMixture.H 2023-05-21 16:27:01.017182932 +0300

```

```

@@ -51,7 +51,8 @@
#include "PtrDictionary.H"
5 #include "volFields.H"
#include "surfaceFields.H"
+#include "massTransferModel.H"
// * * * * *
// * //

10 namespace Foam
@@ -121,6 +122,11 @@
    }

    };

15 +   typedef HashPtrTable<massTransferModel, interfacePair, interfacePair::
hash>
+       massTransferModelTable;
+
+   typedef HashPtrTable<volScalarField, interfacePair, interfacePair::
hash>
+       massTransferCoeffFields;

20 private:

@@ -136,6 +142,7 @@
    surfaceScalarField rhoPhi_;
25     volScalarField alphas_;

+     volScalarField cg_;
+     volScalarField nu_;

30     typedef HashTable<scalar, interfacePair, interfacePair::hash>
@@ -144,6 +151,11 @@
    sigmaTable sigmas_;
    dimensionSet dimSigma_;

35 +     typedef HashTable<dictionary, interfacePair, interfacePair::hash>
+         interfaceDictTable;
+
+     massTransferModelTable massTransferModels_;
+
40     //- Stabilisation for normalisation of the interface normal
    const dimensionedScalar deltaN_;

@@ -226,6 +238,9 @@
    return rhoPhi_;
45     }

+     //- Return the mixture gravity correction coefficient
+     tmp<volScalarField> correctG();
+
50     //- Return the mixture density

```

```

        tmp<volScalarField> rho() const;

@@ -259,6 +274,37 @@
        //- Solve for the mixture phase-fractions
55         void solve();

        + //- Calculate strain rate tensors invariants
        + void calcSpecificStrainRate();
        +
        60 + //- Return the table of mass transfer models
        + const massTransferModelTable& massTransferModels() const
        + {
        +     return massTransferModels_;
        + }
        65 +
        + //- Return the drag coefficients for all of the interfaces
        + autoPtr<massTransferCoeffFields> massTransferCoeffs() const;
        +
        + //- define the Su and Sp terms for alphaEqn
70 + tmp<volScalarField> mtmSp
        + (
        +     const phase& phase ,
        +     const massTransferCoeffFields& massTransferCoeffs
        + ) const;
75 +
        + tmp<volScalarField> mtmSu
        + (
        +     const phase& phase ,
        +     const massTransferCoeffFields& massTransferCoeffs
80 + ) const;
        +
        + //- Correct the mixture properties
        + void correct();

```

### ЛИСТИНГ Б.9: multiphaseMixture.C

```

--- multiphaseMixtureOrig.C 2022-10-06 21:31:58.360555105 +0300
+++ multiphaseMixture.C 2023-05-21 16:28:50.024886050 +0300
@@ -107,6 +107,20 @@
        dimensionedScalar(dimless, Zero)
5         ),
        + cg_
        + (
        +     IOobject
10 +     (
        +         "cg",
        +         mesh_.time().timeName(),
        +         mesh_ ,
        +         IOobject::NO_READ,
15 +         IOobject::AUTO_WRITE

```

```

+         ),
+         mesh_,
+         dimensionedScalar(dimless, Zero)
+     ),
20 +
+     nu_
+     (
+         IObject
@@ -128,6 +142,22 @@
25 {
+     rhoPhi_.setOriented();
+
+     interfaceDictTable massTransferModelsDict(lookup("massTransfer"));
+
30 +     forAllConstIters(massTransferModelsDict, iter)
+     {
+         massTransferModels_.set
+         (
+             iter.key(),
35 +             massTransferModel::New
+             (
+                 iter(),
+                 *phases_.lookup(iter.key().first()),
+                 *phases_.lookup(iter.key().second())
40 +             ).ptr()
+         );
+     }
+
+     calcAlphas();
45 +     alphas_.write();
+ }
@@ -136,6 +166,27 @@
// * * * * * Member Functions * * * * *
* //
50 Foam::tmp<Foam::volScalarField>
+Foam::multiphaseMixture::correctG()
+{
+     auto iter = phases_.cbegin();
+
55 +     cg_ *= 0.0;
+
+     for (;iter != phases_.cend(); ++iter)
+     {
+         cg_ += iter()*iter().gFlag();
60 +     }
+
+     return cg_;
+}
+
65 +

```

```

+Foam::tmp<Foam::volScalarField>
Foam::multiphaseMixture::rho() const
{
    auto iter = phases_.cbegin();
70 @@ -626,6 +668,10 @@
        );
    }

+ // Adding Sp and Su terms to MULES limit
75 + const volScalarField Sp(mtmSp(alpha, massTransferCoeffs()));
+ const volScalarField Su(mtmSu(alpha, massTransferCoeffs()));
+
    MULES::limit
    (
80     1.0/mesh_.time().deltaT().value(),
@@ -633,8 +679,8 @@
        alpha,
        phi_,
        alphaPhiCorr,
85 - zeroField(),
- zeroField(),
+ Sp(), //zeroField(),
+ Su(), //zeroField(),
        oneField(),
90     zeroField(),
        true
@@ -659,6 +705,30 @@
        dimensionedScalar(dimless, Zero)
    );
95
+ volScalarField sumSp
+ (
+     IObject
+     (
100 +         "sumSp",
+         mesh_.time().timeName(),
+         mesh_
+     ),
+     mesh_,
105 +     dimensionedScalar("sumSp", dimless/dimTime, 0)
+ );
+
+ volScalarField sumSu
+ (
110 +     IObject
+     (
+         "sumSu",
+         mesh_.time().timeName(),
+         mesh_
115 +     ),
+     mesh_,

```

```

+         dimensionedScalar("sumSu", dimless/dimTime, 0)
+     );
+
120     phasei = 0;

        for (phase& alpha : phases_)
@@ -666,11 +736,17 @@
            surfaceScalarField& alphaPhi = alphaPhiCorrs[phasei];
125            alphaPhi += upwind<scalar>(mesh_, phi_).flux(alpha);

+         // Adding Sp and Su terms to MULES explicit solve
+         const volScalarField Sp(mtmSp(alpha, massTransferCoeffs()));
+         const volScalarField Su(mtmSu(alpha, massTransferCoeffs()));
130 +
            MULES::explicitSolve
            (
                geometricOneField(),
                alpha,
135 +         alphaPhi,
+         Sp(),
+         Su()
            );

140         rhoPhi_ += alphaPhi*alpha.rho();
@@ -681,7 +757,21 @@
            << ' ' << max(alpha).value()
            << endl;

145 +         Info<< alpha.name() << " Sp volume fraction, min, max = "
+         << Sp.weightedAverage(mesh_.V()).value()
+         << ' ' << min(Sp).value()
+         << ' ' << max(Sp).value()
+         << endl;
150 +
            Info<< alpha.name() << " Su volume fraction, min, max = "
+         << Su.weightedAverage(mesh_.V()).value()
+         << ' ' << min(Su).value()
+         << ' ' << max(Su).value()
155 +         << endl;
+
            sumAlpha += alpha;
+         sumSp += Sp;
+         sumSu += Su;
160
            ++phasei;
        }
@@ -692,6 +782,18 @@
            << ' ' << max(sumAlpha).value()
165            << endl;

+         Info<< "Sp-sum volume fraction, min, max = "

```



```

+         << sumSp.weightedAverage(mesh_.V()).value()
+         << ' ' << min(sumSp).value()
170 +         << ' ' << max(sumSp).value()
+         << endl;
+
+     Info<< "Su-sum volume fraction, min, max = "
+         << sumSu.weightedAverage(mesh_.V()).value()
175 +         << ' ' << min(sumSu).value()
+         << ' ' << max(sumSu).value()
+         << endl;
+
+     // Correct the sum of the phase-fractions to avoid 'drift'
180     volScalarField sumCorr(1.0 - sumAlpha);
        for (phase& alpha : phases_)
@@ -703,6 +805,132 @@
    }

185
+void Foam::multiphaseMixture::calcSpecificStrainRate()
+{
+    for (phase& phase : phases_)
+    {
190 +        phase.calcSpecificStrainRate();
+    }
+}
+
+
195 +Foam::autoPtr<Foam::multiphaseMixture::massTransferCoeffFields>
+Foam::multiphaseMixture::massTransferCoeffs() const
+{
+    autoPtr<massTransferCoeffFields> massTransferCoeffsPtr(new
massTransferCoeffFields);
+
+    forAllConstIterators(massTransferModels_, iter)
+    {
+        const massTransferModel& mtm = *iter();
+        const tmp<volScalarField> K = mtm.K();
+
205 +        volScalarField* Kptr = (mtm.K()).ptr();
+
+        massTransferCoeffsPtr().set(iter.key(), Kptr);
+    }
+
210 +    return massTransferCoeffsPtr;
+}
+
+
+// Calculation of the source terms in alphaEqn
215 +Foam::tmp<Foam::volScalarField>Foam::multiphaseMixture::mtmSp
+(
+    const phase& phasei,

```

```

+ const massTransferCoeffFields& massTransferCoeffField
+) const
220 +{
+ tmp<volScalarField> tmtmSp
+ (
+   new volScalarField
+   (
225 +     IOobject
+     (
+       "mtmSp" ,
+       mesh_.time().timeName(),
+       mesh_
230 +     ),
+     mesh_ ,
+     dimensionedScalar
+     (
+       "mtmSp",
235 +     dimless/dimTime ,
+     0
+     )
+   )
+ );
240 + massTransferModelTable::const_iterator mtmIter = massTransferModels_.
+   begin();
+ massTransferCoeffFields::const_iterator cIter = massTransferCoeffField.
+   begin();
+ for
+ (
+   ;
245 +   mtmIter.good() && cIter.good();
+   ++mtmIter, ++cIter
+ )
+ {
+   if (&phasei == &mtmIter()->phase1())
250 +   {
+     const phase *phasePtr = &mtmIter()->phase1();
+     const volScalarField alpha = *phasePtr;
+     const volScalarField Sp = *cIter();
+     tmtmSp.ref() = Sp;
255 +     tmtmSp.ref().primitiveFieldRef() /= Foam::pow(mesh_.V(), 0.3333);
+     tmtmSp.ref() = Foam::min(tmtmSp.ref(), alpha / dimensionedScalar(
+       dimTime, 1.0));
+     tmtmSp.ref() = - tmtmSp.ref() / (SMALL + alpha);
+   }
+ }
260 + return tmtmSp;
+}
+
+
+Foam::tmp<Foam::volScalarField>Foam::multiphaseMixture::mtmSu
265 +(

```

```

+ const phase& phasei,
+ const massTransferCoeffFields& massTransferCoeffField
+) const
+{
270 + tmp<volScalarField> tmtmSu
+ (
+   new volScalarField
+   (
+     IOobject
275 +     (
+       "mtmSu" ,
+       mesh_.time().timeName(),
+       mesh_
+     ),
280 +     mesh_,
+     dimensionedScalar
+     (
+       "mtmSu",
+       dimless/dimTime,
285 +       0
+     )
+   )
+ );
+ massTransferModelTable::const_iterator mtmIter = massTransferModels_.
  begin();
290 + massTransferCoeffFields::const_iterator cIter = massTransferCoeffField.
  begin();
+ for
+ (
+   ;
+   mtmIter.good() && cIter.good();
295 +   ++mtmIter, ++cIter
+ )
+ {
+   if (&phasei == &mtmIter()->phase2())
+   {
300 +     const phase *phasePtr = &mtmIter()->phase1();
+     const volScalarField alpha = *phasePtr;
+     const volScalarField Su = *cIter();
+     tmtmSu.ref() = Su;
+     tmtmSu.ref().primitiveFieldRef() /= Foam::pow(mesh_.V(), 0.3333);
305 +     tmtmSu.ref() = Foam::min(tmtmSu.ref(), alpha / dimensionedScalar(
      dimTime, 1.0));
+   }
+ }
+ return tmtmSu;
+}
310 +
+
bool Foam::multiphaseMixture::read()
{

```

```
|   if (transportModel::read())
```

## Приложение В

### Модуль автоматической оптимизации коэффициентов турбулентных моделей MLTFoamOptimization

Ниже приведен код модуля автоматической оптимизации коэффициентов  $k$ - $\omega$  SST турбулентной модели MLTFoamOptimization. Для оптимизации коэффициентов других моделей необходимо изменить имена коэффициентов и их начальные значения.

Листинг В.1: MLTFoamOptimization.py

```

import subprocess as sp
import pandas as pd
import numpy as np
from scipy.optimize import minimize
5 import math
from multiprocessing import Pool
from functools import partial
from datetime import datetime
import os
10
def readLoss(fileName):
    loss = float(open(fileName, "r").read())
    return loss

15 def runOF(caseDir, coeffs, coeffsKeys):
    try:
        os.remove(caseDir+'default.parameters')
    except OSError:
        pass
20 for key, value in list(zip(coeffsKeys, coeffs)):
    open(caseDir+'default.parameters', 'a').write(key+' '+str(value)+';\n')
    sp.call("cd "+caseDir+"\
        sbatch -n8 -W -o log.sbatch runStep.sh;\
        cd ../", shell=True)
25
def calcLoss(coeffs, caseDirs, coeffsKeys):
    open("log.MLTransportCoeffsCorrection", "a").write(str(datetime.now())+'\
        n')
    with Pool() as pool:
        pool.map(partial(runOF, coeffs=coeffs, coeffsKeys=coeffsKeys), caseDirs
        )
30 open("log.MLTransportCoeffsCorrection", "a").write("coeffs: "+str(coeffs)
        +"\n")

```

```

lossList = list(map(readLoss, list(map(lambda e : e + "data/loss",
    caseDirs))))
loss = sum(lossList)
open("log.MLTransportCoeffsCorrection", "a").write("lossList = "+str(
    lossList)+"\n")
open("log.MLTransportCoeffsCorrection", "a").write("loss = "+str(loss)+"\
    n")
35 return loss

def minimizeSciPy(initCoeffs, bounds, calcLoss, caseDirs):
    res = minimize(calcLoss, list(initCoeffs.values()), args=(caseDirs, list(
        initCoeffs.keys())), method='Nelder-Mead', tol=1e-6)#, bounds=bounds)
    print(res.x)
40

def main():
    bounds = [[0.5, 1.0], [0.5, 2.5], [0.2, 0.8], [0.7, 1.0], [0.0, 0.15],
        [0.0, 0.15], [0.0, 1.0], [0.0, 1.0], [0.0, 1.0], [0.5, 1.5], [5.0,
        15.0]]
    coeffs = {}
    coeffs.update(alphaK1=0.85)
45 coeffs.update(alphaK2=1.0)
    coeffs.update(alphaOmega1=0.5)
    coeffs.update(alphaOmega2=0.856)
    coeffs.update(beta1=0.075)
    coeffs.update(beta2=0.0828)
50 coeffs.update(betaStar=0.09)
    coeffs.update(gamma1=0.5555556)
    coeffs.update(gamma2=0.44)
    coeffs.update(a1=0.31)
    coeffs.update(b1=1.0)
55 coeffs.update(c1=10.0)
    caseDirs = ["constantAngleSlopeTurbKWUProfileInlet180321/",
        "constantAngleSlopeTurbKWUProfileInlet230421/",
        "constantAngleSlopeTurbKWUProfileInlet100621/"]
    print(list(coeffs.values()))
60 minimizeSciPy(coeffs, bounds, calcLoss, caseDirs)

if __name__ == "__main__":
    main()

```