

# Оглавление

<b>Введение</b>	<b>2</b>
<b>1 Аналитическая часть</b>	<b>3</b>
1.1 Водопад как природное явление . . . . .	3
1.2 Методы визуализации водопадов . . . . .	4
1.2.1 Метод, основанный на уравнении Навье-Стокса . . . . .	4
1.2.2 Метод, основанный на частицах . . . . .	5
1.2.3 Сеточные методы . . . . .	5
1.2.4 Комбинированный метод . . . . .	6
1.3 Формализация модели . . . . .	7
1.4 Выбор алгоритма удаления невидимых ребер и поверхностей . .	8
1.4.1 Алгоритм, использующий Z-буфер . . . . .	8
1.4.2 Алгоритм обратной трассировки лучей . . . . .	9
1.4.3 Алгоритм Робертса . . . . .	9
1.4.4 Алгоритм художника . . . . .	10
1.5 Выбор метода закрашивания областей . . . . .	10
1.5.1 Простая закрашка . . . . .	11
1.5.2 Закраска по Гуро . . . . .	11
1.5.3 Закраска по Фонгу . . . . .	11
1.6 Существующие программные обеспечения . . . . .	12
<b>Литература</b>	<b>15</b>

# Введение

В современном мире компьютерная графика вышла на совершенно новый уровень, что связано с её высокой востребованностью в области игр и фильмов [1]. Методы отрисовки все время развиваются, появляются новые способы. Главным требованием является, в первую очередь, является реалистичность изображения. Множество исследований физических явлений происходит постоянно, чтобы с максимальной точностью смоделировать тот или иной объект. Но чем выше точность, тем выше сложность разрабатываемых алгоритмов, что часто приводит к высоким затратам по времени и памяти.

Одной из самых сложных тем при моделировании являются жидкости [2]. Имеется огромная потребность в качественной и эффективной отрисовке океанов, рек, ручейков и даже луж. В данном курсовом проекте речь пойдет о моделировании водопада, с учетом аэрозольных облаков, которые возникают у подножия самого водопада при падении воды.

Цель работы - создать качественную симуляцию водопада с использованием современных методов и технологий

- проанализировать методы и алгоритмы, моделирующие водопады;
- определить алгоритм, который наиболее эффективно справляется с поставленной задачей;
- реализовать алгоритм;

# 1 Аналитическая часть

## 1.1 Водопад как природное явление

Водопад - падение воды в реке с уступа, который пересекает речное русло. Не всякий поток воды можно назвать водопадом. Для этого должны соблюдаться несколько важных условий - высота более 1 метра и уклон свыше 45 градусов.

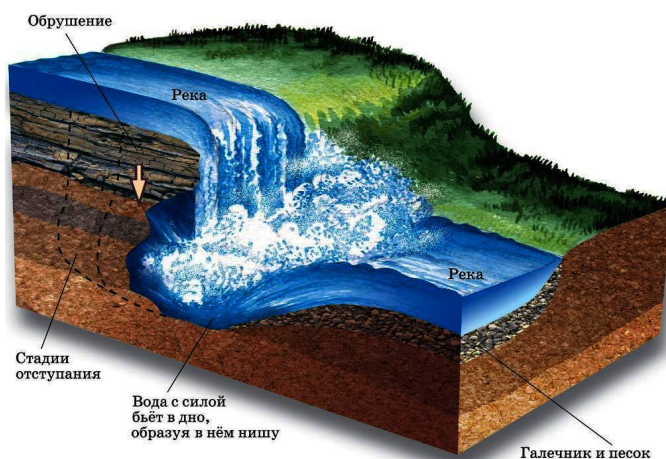


Рис. 1.1: Модель водопада

Чаще всего водопады возникают при перепаде высот, которые были там еще до появления реки. Также водопады появляются по причине эрозии - водяной поток размывает мягкие осадочные породы, тем самым происходит обрушение вышележащих слоев породы, вследствие чего на границе твердых и мягких пород образуется вертикальный уступ, с которого падает вода.

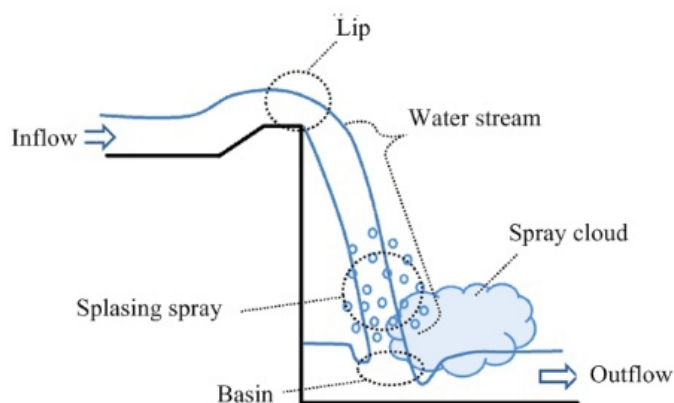


Рис. 1.2: Составные части водопада

При моделировании водопада чаще всего обращают внимание лишь на поток воды, который падает вниз с обрыва. Но для качественной и реалистичной картинки этого не достаточно. При построении модели водопада нужно учитывать три важные его составляющие - сам поток воды (water stream), брызги (splashing spray), которые от него исходят, а также аэрозольное облако (spray cloud), которое образуется при ударе потока об воду.

В процессе моделирования каждая часть будет рассмотрена отдельно, чтобы в конце совместить результаты и получить наиболее реалистичную картинку.

## 1.2 Методы визуализации водопадов

Водопад представляет собой водяной поток с частицами брызг. За все время моделирования текучей воды было разработано несколько основных методов, конкретнее о которых речь и пойдет дальше.

### 1.2.1 Метод, основанный на уравнении Навье-Стокса

Уравнение Навье-Стокса [3] является системой дифференциальных уравнений в частных производных, которое описывает движение вязких ньютоновских жидкостей, которое до сих пор не имеет решения в общем виде. Оно часто используется для математического моделирования сложных моделей природных явлений. При разработке алгоритмов используются частные решения.

Методы, которые строятся на основе этого уравнения, являются довольно трудно реализуемы за счет большого количества сложной математики, с которой приходится работать, а также множества вычислений, которые приходится делать компьютеру при обработке всех формул, что является крайне неэффективным. По этим причинам методы на основе уравнения Навье-Стокса рассматриваться подробно не будут. [4]

## 1.2.2 Метод, основанный на частицах

Метод частиц заключается в использовании трилинейной интерполяции для каждой частицы, чтобы определить ее скорость движения. И каждая частица перемещается в соответствии с определенным инерциальным физическим уравнением. Данный подход имеет довольно небольшие затраты по компьютерным ресурсам, а также большое количество частиц позволяет достичь максимальной точности изображения. Главным минусом является невозможность точно определить границы, в которых должен находиться водяной поток

Наиболее распространённые методы с частицами:

- метод на основе понятия о полу-Лагранже, который объединяется с новым подходом расчета жидкости вокруг объектов. Это позволяет эффективно решать уравнения движения жидкости, сохраняя при этом достаточно деталей, чтобы получить реалистичное изображение. Высококачественная поверхность получается из результирующего поля скоростей с использованием новой адаптивной техники для создания неявной поверхности [5];
- метод, использующий диаграммы Вороного. Ключевым компонентом этого алгоритма является аппроксимация геометрии пены путем обработки частиц пузырьков как участков взвешенной диаграммы Вороного. Информация о связности, предоставляемая диаграммой Вороного, позволяет нам точно моделировать различные эффекты взаимодействия между пузырьками. Используя ячейки Вороного и веса, мы также можем явно решить проблему потери объема при моделировании пены, что является общей проблемой во многих подходах. Может быть встроен в симуляторы жидкости, в частности, водопада [6].

## 1.2.3 Сеточные методы

Сетки часто используются при моделировании жидкостей. В свою очередь, сама сетка - регулярная кубическая, в каждой ячейке которой содержится информация о точке поверхности, которая задает саму картину. Данный метод

при большом количестве ячеек вызывает большое потребление памяти, хотя чем больше ячеек, тем выше качество получаемого изображения. Но также этот метод имеет и свои преимущества. Во-первых, визуализировать результат симуляции проще. Во-вторых, точно известны границы желаемой области, что приводит к возможности встроить данный водяной поток в нужное место.

Наиболее распространённые методы с сеткой:

- метод моделирования пузырька воздуха в воде. Этот метод представляет собой физически обоснованный метод расчета размеров пузырьков воздуха, основанный на скорости замерзания и давлении. В отличие от большинства методов, этот метод может представлять множество мелких пузырьков, которые не могут быть представлены с помощью решений имитационных сеток [7];
- метод, использующий уравнение Эйлера. Благодаря новому подходу можно моделировать крупномасштабные трехмерные жидкости. Здесь используется гибридное представление сетки, состоящее из правильных кубических ячеек поверх слоя высоких ячеек. Также благодаря оптимизации в представлении сетки была предложена реализация ГПУ решателя для жидкости [8].

#### 1.2.4 Комбинированный метод

Поскольку каждый метод по отдельности - на частицах и на сетке - имеет свои недостатки, то было предложено объединить эти два метода в один. Совместное их использование помогает одновременно с потоком воды моделировать брызги, пузырьки. Также подобные методы имеют высокую эффективность за счет несложных математических вычислений. Комбинированный метод частиц и сетки и ложится в основу алгоритма моделирования водопада, который будет рассматриваться в дальнейшем

Наиболее распространённые комбинированные методы:

- метод, при котором моделируются пузырьки воздуха и пена вместе. Обе фазы рассматриваются отдельно. Также в этом методе применяется такие подходы, как использование силы сопротивления (для того, чтобы

совместить оба этапа и при этом сильно не увеличить вычислительные нагрузки), функция насыщения (зависит от объема). При достижении поверхности пузырьки преобразуются в пену, а потом удаляются за указанное пользователем время [9];

- метод, который напрямую использует подходы на основе сеток и частиц. Области жидкости, которые не могут быть представлены полем высот (например, разбивающиеся волны и брызги из-за ударов воды), то они представляются частицами и превращаются в брызги и пену. Сами частицы рассматриваются как простые невзаимодействующие точечные массы, которые обмениваются массой и моментом с жидкостью поля высот [10].

## Вывод

В качестве приоритетного выбирается комбинированный метод визуализации водопада. Он имеет преимущества над остальными, которое заключается в том, что водопад можно разбить на несколько частей и каждую часть обрабатывать конкретным подходом - где-то частицами, где-то сетками. Подобная возможность повышает эффективность работы алгоритма за счет ступенчатого подхода к визуализации результата.

## 1.3 Формализация модели

Модель водопада будет задаваться такими характеристиками, как:

- высота - высота уступа, с которого будет падать вода. Число типа *float* в промежутке от 1 до 50 (в метрах);
- ширина - по сути, ширина уступа, с которого течет вода. Число типа *float* в промежутке от 1 до 10 (в метрах);
- угол падения - угол, который зависит от уступа, с которого падает водопад. Число типа *float* от 45 до 90 (в градусах).

Также частью сцены будет являться скала, которая будет задаваться геометрическим объектом, который будет постоянной формы. Пользователю будет предоставлена возможность изменить его высоту, ширину и угол наклона одной из его сторон.

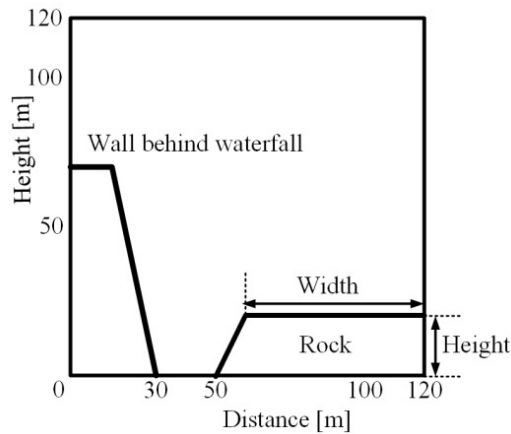


Рис. 1.3: Объекты сцены и их параметры

## 1.4 Выбор алгоритма удаления невидимых ребер и поверхностей

Для начала выделим несколько свойств, которыми должен обладать выбранный алгоритм, чтобы обеспечить оптимальную работу и реалистичное изображение.

Свойства:

- алгоритм может работать как в объектном пространстве, так и в пространстве изображений;
- алгоритм должен быть достаточно быстрым и использовать мало памяти.

### 1.4.1 Алгоритм, использующий Z-буфер

*Суть данного алгоритма* - это использование двух буферов: буфера кадра, в котором хранятся атрибуты каждого пикселя, и Z-буфера, в котором



хранятся информация о координате  $Z$  для каждого пикселя.

Изначально в  $Z$ -буфере находятся минимальные значения  $Z$ , а в буфере кадра - информация о пикселях, которые описывают фон. Глубина каждого нового пикселя при подсчете сравнивается со значением, которое уже есть в  $Z$ -буфере. В случае, если новый пиксель расположен ближе к наблюдателю, чем предыдущий, то информация о нем заносится в буфер кадра и происходит редактирование  $Z$ -буфера.

*Положительной стороной* данного алгоритма является простота реализации, экономия времени, так как нет сортировки.

*Недостатком* - большой объем требуемой памяти, а также сложная реализация эффектов прозрачности.

### 1.4.2 Алгоритм обратной трассировки лучей

*Суть данного алгоритма* состоит в том, что наблюдатель видит объект с помощью испускаемого света, который согласно законам оптики доходит до наблюдателя некоторым путем. Алгоритм называется “обратной трассировкой” из-за того, что эффективнее выходит отслеживать лучи, которые идут от наблюдателя к объекту.

*Положительным моментом* в этом алгоритме является возможность использования в параллельных вычислительных системах.

*Недостатком* же является то, что требуется большое количество необходимых вычислений.

### 1.4.3 Алгоритм Робертса

Данный алгоритм решает задачу удаления невидимых ребер и граней только с выпуклыми телами, работает только в объектном пространстве.

У алгоритма три этапа выполнения:

- этап 0 - подготовка данных;
- этап 1 - удаление ребер, экранируемых самим телом;

- этап 2 - удаление невидимых ребер, экранируемых другими телами сцены.

*Положительным моментом* в этом алгоритме является высокая точность вычислений.

*Недостатком* же является то, что все тела должны быть выпуклыми.

#### 1.4.4 Алгоритм художника

Данный алгоритм работает аналогично тому, как художник рисует картину - то есть сначала рисуются дальние объекты, а затем более близкие. Наиболее распространенная реализация алгоритма - сортировка по глубине, которая заключается в том, что произвольное множество граней сортируется по ближнему расстоянию от наблюдателя, а затем отсортированные грани выводятся на экран в порядке от самой дальней до самой ближней. Данный метод работает лучше для построения сцен, в которых отсутствуют пересекающиеся грани.

*Положительным моментом* является то, что данный алгоритм требует памяти меньше, чем, например, алгоритм Z-буфера.

*Недостатком* же является то, что у него недостаточно высока реалистичность изображения.

### Вывод

Для удаления невидимых линий выбран алгоритм Z-буфера. Данный алгоритм довольно прост в реализации, работает довольно быстро из-за отсутствия сортировок, а также позволяет добиться хорошей реалистичности, что очень важно при моделировании таких явлений природы, как водопад.

## 1.5 Выбор метода закрашивания областей

Существует несколько основных алгоритмов закрашки:

- простая закрашка;
- закрашка по Гуро;
- закрашка по Фонгу.

### 1.5.1 Простая закрашка

Вся грань закрашивается одним уровнем интенсивности, который вычисляется по *закону Ламберта*.

Этот метод достаточно прост в реализации и совершенно не требователен к ресурсам. Однако плохо подходит для тел вращения и плохо учитывает отраженный свет.

### 1.5.2 Закрашка по Гуро

*Основа закрашки по Гуро* – билинейная интерполяция интенсивностей, за счет которой устраняется дискретность изменения интенсивности и создается иллюзия гладкой криволинейной поверхности. Хорошо сочетается с диффузным отражением.

### 1.5.3 Закрашка по Фонгу

*Основа закрашки по Фонгу* – билинейная интерполяция векторов нормалей. Достигается лучшая локальная аппроксимация кривизны поверхности. Изображение выходит более реалистичным, зеркальные блики выглядят правдоподобнее, чем в методе закрашки по Гуро.

Однако по сравнению с методом Гуро, закрашка по Фонгу требует больших вычислительных затрат, так как интерполируются значения векторов нормалей, на основе которых потом вычисляется интенсивность.

## Вывод

Для решения поставленной задачи будет использован метод закраски по Гуро, потому что он отвечает всем требованиям для максимально реалистичной закраски сложных объектов сцены.

## 1.6 Существующие программные обеспечения

*Blender* - бесплатный, кроссплатформенный набор инструментов для работы с 3D с открытым кодом. Он обладает широким функционалом - подходит для моделирования, симуляции, рендеринга, монтажа, записи видео и создания игр.

Благодаря огромному количеству аддонов, которые создают сами пользователи, можно смоделировать практически все, что угодно. Это относится и к водопаду, который можно пустить по любой поверхности, при этом это будет выглядеть максимально реалистично.

На фотографиях ниже представлен пример такого водопада.

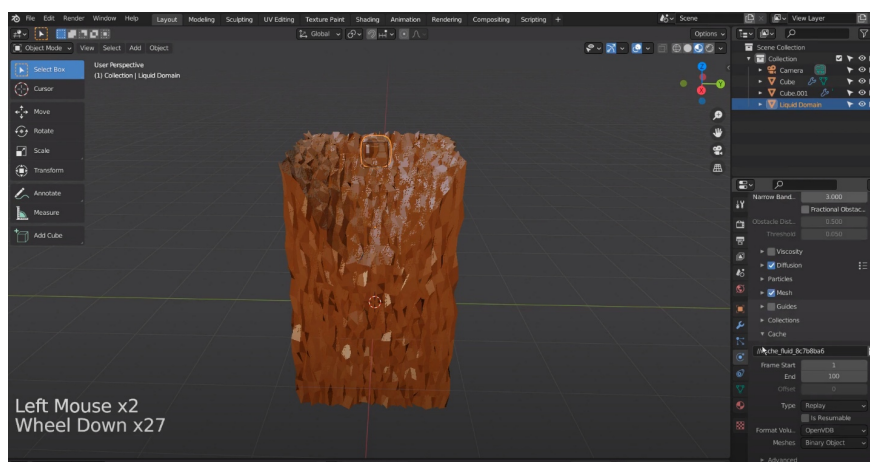


Рис. 1.4: Пример реализации водопада в программе Blender ч.1

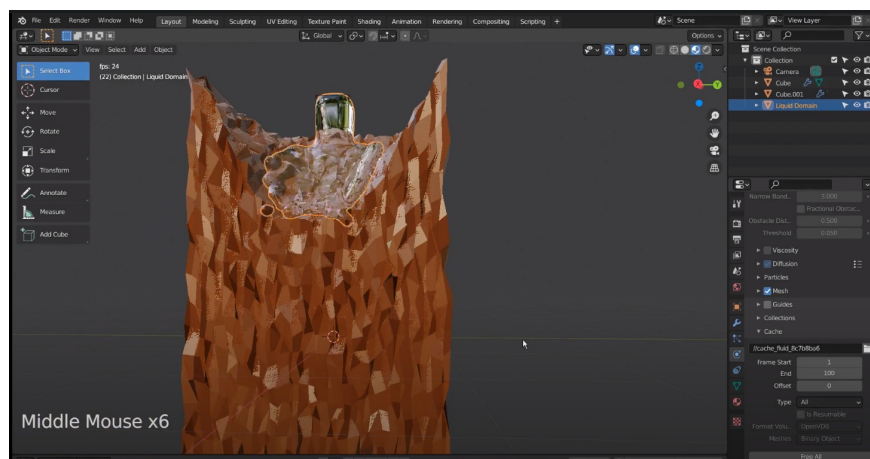


Рис. 1.5: Пример реализации водопада в программе Blender ч.2

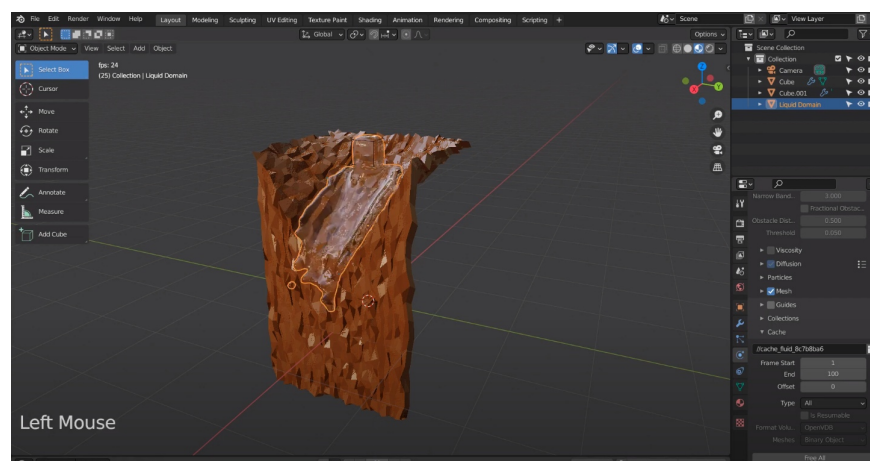


Рис. 1.6: Пример реализации водопада в программе Blender ч.3

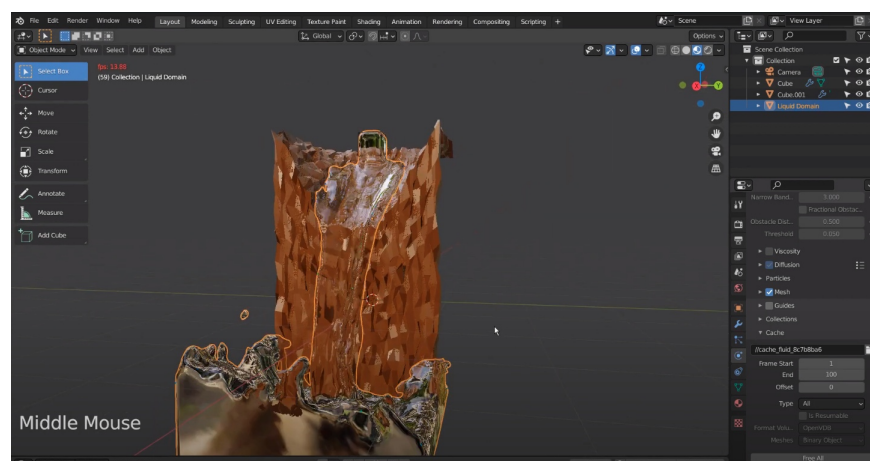


Рис. 1.7: Пример реализации водопада в программе Blender ч.4

## Вывод

В данном разделе были формально описаны все методы по визуализации текучей воды, с помощью которых можно получить реализацию водопада, а также методы удаления невидимых линий и поверхностей и закрашивания областей. В качестве алгоритма визуализации водопада предпочтение отдается методу, который реализует комбинированный подход. В качестве алгоритма удаления невидимых линий и областей - z-буфер, а закраски - по Гуро.

# Литература

- [1] Water Simulating in Computer Graphics [Электронный ресурс]. Режим доступа: <https://www.diva-portal.org/smash/get/diva2:205412/FULLTEXT01.pdf> (дата обращения: 17.09.2021).
- [2] Modeling Surface Water-Groundwater Interaction with MODFLOW: Some Considerations [Электронный ресурс]. Режим доступа: <https://core.ac.uk/download/pdf/20658268.pdf> (дата обращения: 17.09.2021).
- [3] The Navier-Stokes Equations [Электронный ресурс]. Режим доступа: [https://www.iit.edu/sites/default/files/2021-02/navier\\_stokes.pdf](https://www.iit.edu/sites/default/files/2021-02/navier_stokes.pdf) (дата обращения: 23.09.2021).
- [4] Solving 3D incompressible Navier-Stokes equations on hybrid CPU/GPU systems [Электронный ресурс]. Режим доступа: <https://www.lri.fr/~baboulin/hpc14.pdf> (дата обращения: 23.09.2021).
- [5] Practical Animation of Liquids [Электронный ресурс]. Режим доступа: <http://physbam.stanford.edu/~fedkiw/papers/stanford2001-02.pdf> (дата обращения: 17.09.2021).
- [6] Animating Bubble Interactions in a Liquid Foam [Электронный ресурс]. Режим доступа: <https://web.cse.ohio-state.edu/~dey.8/paper/foam/bubble.pdf> (дата обращения: 17.09.2021).
- [7] Visual Simulation of Freezing Ice with Air Bubbles [Электронный ресурс]. Режим доступа: [https://www.researchgate.net/profile/Kei-Iwasaki/publication/262171575\\_Visual\\_simulation\\_of\\_freezing\\_ice\\_with\\_air\\_bubbles/links/simulation-of-freezing-ice-with-air-bubbles.pdf](https://www.researchgate.net/profile/Kei-Iwasaki/publication/262171575_Visual_simulation_of_freezing_ice_with_air_bubbles/links/simulation-of-freezing-ice-with-air-bubbles.pdf) (дата обращения: 17.09.2021).
- [8] Real-Time Eulerian Water Simulation Using a Restricted Tall Cell Grid [Электронный ресурс]. Режим доступа: <https://matthias-research.github.io/pages/publications/tallCells.pdf> (дата обращения: 17.09.2021).
- [9] ANIMATION OF AIR BUBBLES WITH SPH [Электронный ресурс]. Режим доступа: <https://www.scitepress.org/Papers/2011/33229/33229.pdf> (дата обращения: 17.09.2021).

- [10] Real-time Simulation of Large Bodies of Water with Small Scale Details [Электронный ресурс]. Режим доступа: <https://matthias-research.github.io/pages/publications/hfFluid.pdf> (дата обращения: 17.09.2021).