



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ \_\_\_\_\_ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ \_\_\_\_\_

КАФЕДРА \_\_\_\_\_ ТЕОРЕТИЧЕСКАЯ ИНФОРМАТИКА И КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ \_\_\_\_\_

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ

## НА ТЕМУ:

Графическая визуализация поверхности воды и расчёты заданных  
параметров с применением программного интерфейса OpenGL

Студент \_\_\_\_\_ ИУ9-526 \_\_\_\_\_  
(Группа)

\_\_\_\_\_  
(Подпись, дата) М. Д. Соколовский  
(И.О.Фамилия)

Руководитель курсовой работы

\_\_\_\_\_  
(Подпись, дата) В. В. Соборова  
(И.О.Фамилия)

Консультант

\_\_\_\_\_  
(Подпись, дата) В. В. Соборова  
(И.О.Фамилия)

Москва  
2020

## СОДЕРЖАНИЕ

Введение.....	3
1 Модели расчёта геометрии поверхности .....	5
1.1 Приближённые методы расчёта.....	6
1.1.1 Шум Перлина.....	7
1.1.2 Сумма синусов.....	9
1.1.3 Закон Гука .....	11
1.2 Процедурные методы расчёта.....	12
1.2.1 Волны Герстнера .....	12
1.2.2 Статистический метод .....	15
2 Освещение и текстурирование.....	18
2.1 Модель Блинна-Фонга .....	18
Заключение .....	20
Список использованных источников .....	21

## ВВЕДЕНИЕ

Визуализация поверхности воды может использоваться для формирования фотореалистичных изображений в большом кругу сфер деятельности, таких как: средства массовой информации, различные дизайнерские решения, сфера развлечений (в частности – видеоигры и кинематограф), и т.д. Различные сферы представляют различные требования к визуализации, в частности к степени фотореалистичности и физической корректности, а также к производительности системы и минимальным требованиям к вычислительным мощностям оборудования.

Данная работа посвящена визуализации водной поверхности, учитывающей воздействие физических явлений при заданных параметрах. Система визуализации реализуется при помощи языка программирования C++ стандарта C++17 и предназначена для работы в операционной системе на базе Linux. Для работы с графической подсистемой компьютера используется программный интерфейс OpenGL версии не младше 3.3, с использованием языка программирования шейдеров GLSL.

Система визуализации состоит из двух частей. В первой части реализуются необходимые вычисления и обработка параметров. Вторая часть предоставляет демонстрационный графический интерфейс, отображающий результаты работы первой части, а также позволяющий динамически отображать и изменять параметры визуализации, такие как: свойства физических явлений, положение, направление, а также прочие свойства условной камеры и прочие настройки отображения, в том числе – связанные с корректным отображением визуализированной модели на экране монитора (например – гамма коррекция).

В данной работе будут рассмотрены и сравнены различные подходы к моделированию геометрии поверхности. После расчёта геометрии, реализуются оптические эффекты, связанные с освещением, а также преломлением и отражением лучей:

- Учёт освещения нескольких источников света. В частности, необходимо освещать как поверхность воды, так и расположенные под водой объекты.
- Корректное отображение подводных объектов с учётом преломления лучей в воде.
- Дополнительная обработка ситуации, когда дно находится слишком глубоко (визуализация океана).

Для расчётов геометрии и освещения в работе учитываются такие физические показатели, как:

- Глубина
- Скорость и направление ветра
- Показатель преломления воды
- Положение и интенсивность источника света

## 1 Модели расчёта геометрии поверхности

Существующие модели расчёта геометрии можно разделить на три категории [1]:

1. Приближённые методы, позволяющие получить лишь похожие на реальную поверхности. Они основаны на использовании функций шумов, и прочих подходов.
2. Процедурные методы, реализующие приближённую, однако более фотореалистичную, визуализацию.
3. Физически корректные, или близкие к этому методы.

Обычно наблюдается обратная зависимость между физической достоверностью и требованиями к мощности вычислительной системы. Однако, часто, для получения подходящего изображения, приближённые методы подходят лучше, нежели физически корректные.

Физически корректные методы используют подход, при котором жидкость рассматривается как множество элементарных объектов – частиц, разделённых по трёхмерной сетке. При этом рассчитывается их поведение на основе физических законов, таких как уравнение Навье-Стокса, и др. Однако для получения реалистичного изображения число таких частиц должно быть слишком большим, что приведёт к слишком сильному росту требований к вычислительным ресурсам. Поскольку целью данной работы является лишь визуализация поверхности воды, но не симуляция всего объёма жидкости, подобные методы в ней рассматриваться не будут.

Выбор метода выполняется исходя из поставленной задачи и доступных ресурсов. В большинстве случаев, в которых основным критерием качества является достаточная визуальная достоверность изображения, рационально использовать методы из второй категории.

Для большинства методов, описанных в данной работе, поверхность воды разбивается регулярной сеткой на треугольники, как показано на рисунке 1. Методы моделирования изменяют значение высоты (координата  $Y$ ) узлов сетки. Допустимо также введение горизонтального отклонения узла от исходного положения.

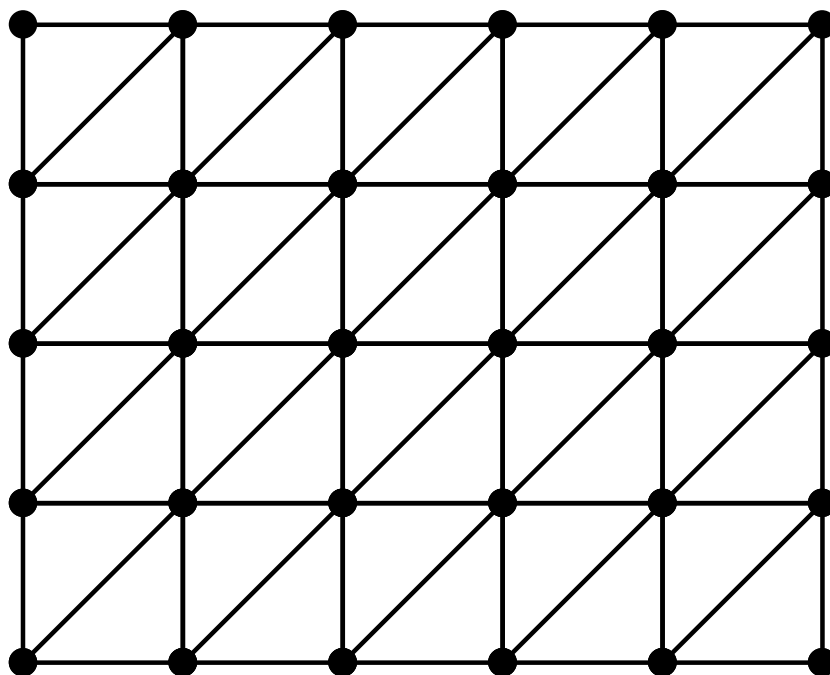


Рисунок 1: Регулярная сетка

### 1.1 Приближённые методы расчёта

Для визуализации воды в системах, не требовательных к физической достоверности, а также в системах с низкими требованиями к качеству моделей (например – низкополигональные системы) можно использовать приближённые методы расчёта. Дополнительно, они менее требовательны к вычислительным ресурсам системы, что позволяет увеличить частоту рендеринга кадров.

### 1.1.1 Шум Перлина

Шум – это функция, позволяющая получать случайные значения для заданной точки в пространстве заданной размерности. Для большей универсальности далее будем предполагать, что данная функция будет принимать значения в интервале от 0 до 1. Определённую таким образом функцию, домножением на константу, можно легко привести к требуемому интервалу значений. Также, далее будут рассматриваться только функции шума для пространства размерности 2.

В качестве примера шума можно рассмотреть функцию, которая каждой точке будет сопоставлять значение, независящее от остальных точек пространства, с равной вероятностью принимающее все значения из заданного диапазона. Шум, определённый таким образом называется белый шум. Однако такой шум не является плавным. Поэтому применить его для визуализации водной поверхности не получится ввиду резких перепадов высот, не характерных для жидкостей (необходимость плавности функции шума будет обоснована далее).

Для генерации непрерывного и более плавного шума может использоваться шум Перлина. Для его вычисления необходимо найти 4 точки с целочисленными координатами, наиболее близких к заданной точке. Так как точки с целочисленными координатами образуют сетку на плоскости, этими точками будут вершины квадрата, внутри которого находится заданная точка. Это показано на рисунке 2. Для каждой вершины случайный вектор единичной длины, называемый градиентом в данной точке (на рисунке – вектора  $a_i$ ). Далее, для каждой из 4 точек нужно скалярно умножить вектор градиента на вектор, направленный к заданной точке (на рисунке – вектора  $b_i$ ).

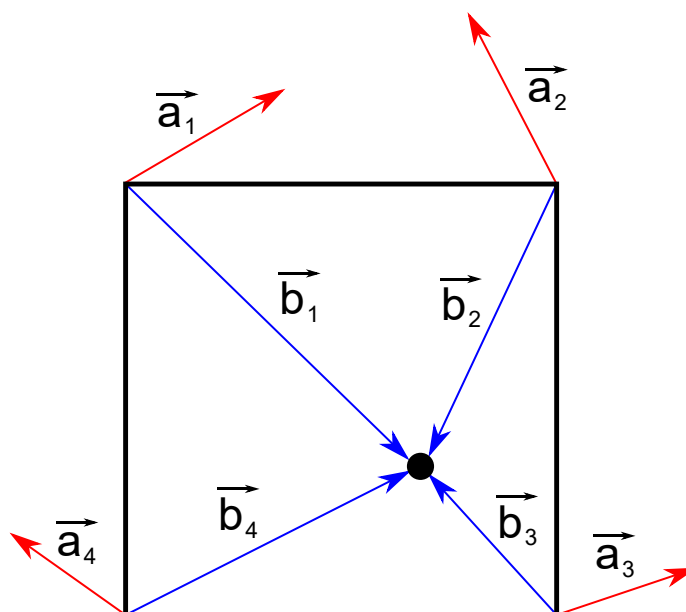


Рисунок 2: Вычисление шума Перлина

Полученные значения необходимо интерполировать, для получения результата. Однако, при использовании линейной интерполяции на границах ячеек сетки могут возникать неровности. В качестве интерполирующей функции может использоваться, например, стандартная функция языка GLSL smoothstep, либо другая, не приводящая к нарушению непрерывности. Сам процесс интерполяции в двумерном пространстве выполняется за 3 этапа: на первых двух этапах интерполируются значения между соседними вершинами, а на последнем — их результат.

Далее, мы можем уменьшить шаг ячеек сетки в два раза, и для такой сетки заново вычислить шум. Каждая следующая функция шума, полученная таким образом называется октавой. Для получения более достоверных результатов мы можем просуммировать несколько октав такого шума.

Таким образом, шум Перлина может быть гибко настроен для различных задач путём выбора таких параметров, как: функция выбора случайных градиентов, функция интерполяции и количество суммируемых октав.



Двумерный шум Перлина может быть использован для создания двумерной текстуры, определяющий изменение высоты узлов сетки. Подобный подход предлагается в статье [4]. Генерация такого шума обладает скромными требованиями к вычислительным ресурсам системы, позволяя за небольшие затраты получить плавную симуляцию водной поверхности.

Однако, использование этого метода не позволяет получать достоверное изображение волн. Это связано с тем, что в данной технике практически невозможно напрямую учитывать внешние физические параметры. Также, визуализация, полученная в результате использования данного метода, не удовлетворяет ни критериям физической достоверности, ни требованиям фотореалистичности. Несмотря на это, метод может быть использован в системах с критической нехваткой вычислительных ресурсов для получения удовлетворительного изображения низкого качества.

### 1.1.2 Сумма синусов

Данный подход заключается в использовании сумм тригонометрических функций для моделирования волн на поверхности. Рассмотрим одну из его реализаций, описанную в статье [2]. В ней будет визуализироваться несколько волн с постоянными параметрами: амплитудой, частотой волны и скоростью её изменения.

В данном методе вводится функция, позволяющая определять высоту каждого узла сетки поверхности в заданный момент времени. В начале рассмотрим случай визуализации одной волны. Для построения данной функции необходимо ввести промежуточное значение  $S$ . Для этого нужно скалярно перемножить вектор координат точки и вектор направления волны, после чего результат умножить на ряд коэффициентов, как показано в формуле (1).

$$S(\overrightarrow{pos}, t) = (\overrightarrow{pos} \cdot \overrightarrow{dir}) \cdot freq + velocity \cdot freq \cdot t \quad (1)$$

Где  $\overrightarrow{pos}$  – двумерный вектор координат узла сетки,  $\overrightarrow{dir}$  – двумерный вектор направления распространения волны,  $freq$  – частота волны,  $velocity$  – скорость изменения,  $t$  – параметр времени.

Далее вводится функция зависимости изменения высоты узла от времени (далее – функция волны) по формуле (2). На данном этапе возможно добавить отрицательное смещение после нормализации значения синуса для использования отрицательного изменения высоты узлов.

$$f_i(\overrightarrow{pos}, t) = A \cdot \left( \frac{\sin(S(\overrightarrow{pos}, t)) + 1}{2} \right)^{stepness} \quad (2)$$

Где  $\overrightarrow{pos}$  – вектор координат узла сетки,  $t$  – параметр времени,  $stepness$  – показатель крутизны волны,  $A$  – амплитуда волны.

Для введения дополнительных волн в модель, необходимо для каждой точки просуммировать значения функций волн по формуле (3).

$$height(\overrightarrow{pos}, t) = \sum f_i(\overrightarrow{pos}, t) \quad (3)$$

Данный метод также является простым в реализации и не требует затратных вычислений для моделирования, однако качество модели, полученной данным методом остаётся низким: она не является физически корректной, и обладает пониженной фотореалистичностью.

### 1.1.3 Закон Гука

В другом приближённом подходе, описанном в работе [3] используется следующий подход: предполагается, что узлы сетки являются материальными точками и связаны с соседними узлами абстрактной пружиной.

Сила, действующая на каждый узел  $p_i$  со стороны одной пружины  $S_k$ , связывающей этот узел с узлом  $p_j$ , вычисляется исходя из закона Гука по формуле (4).

$$\vec{f}_i^k(t) = c_k \cdot (\vec{x}_j - \vec{x}_i) \cdot \left(1 - L_k \frac{1}{l_k}\right) \quad (4)$$

Где  $c_k$  – коэффициент упругости пружины,  $\vec{x}_i$  и  $\vec{x}_j$  – вектора координат точек,  $L_k$  – длина пружины в состоянии покоя,  $l_k$  – текущая длина пружины,  $t$  – время.

Просуммировав значения таких сил для всех, связанных с заданным узлом, пружин, получаем суммарное значение сил, действующих на узел со стороны пружин. Обозначим его  $\vec{f}_i$ . Изначально система находится в состоянии покоя. Для визуализации волн необходимо вывести из состояния равновесия один или несколько узлов, приложив к нему внешнюю силу  $\vec{f}_i^{ext}$ . Уравнения движения получившейся системы представлены в системе (5).

$$\begin{cases} \vec{a}_i(t) = \frac{1}{m_i} \cdot (\vec{f}_i^{ext}(t) - \gamma_i \cdot \vec{v}_i(t) - \vec{f}_i(t)) \\ \vec{v}_i(t + \Delta t) = \vec{v}_i(t) + \vec{a}_i(t) \cdot \Delta t \\ \vec{x}_i(t + \Delta t) = \vec{x}_i(t) + \vec{v}_i(t + \Delta t) \cdot \Delta t \end{cases} \quad (5)$$

Где  $\vec{a}_i$  – ускорение узла,  $\vec{v}_i$  – скорость узла,  $\vec{x}_i$  – координаты узла,  $m_i$  – масса узла,  $\gamma_i$  – коэффициент демпфирования, характеризующий сопротивление среды, в которой движется узел.

В таблице 1 указаны рекомендуемые характеристики системы, предложенные в работе [3].

Таблица 1 – Характеристики системы пружин

$m_i$	$c_k$	$\gamma_i$	$\Delta t$
4.0	7.0	0.05	0.4

Для получения лучших результатов допустимо задавать возмущения в системе не только прикладыванием сил к узлам, но и изменением других его показателей: скорости и координаты. Также, можно запретить перемещение узлов в горизонтальной плоскости, разрешив движение только по вертикали.

Таким образом, данный метод позволяет гибко настраивать систему, для получения приемлемых результатов.

## 1.2 Процедурные методы расчёта

К процедурным методом расчёта геометрии водной поверхности относятся методы, также реализующие приближённые расчёты геометрии поверхности, однако, позволяющие добиться лучших результатов за счёт большей проработанности и физической обоснованности.

В отличие от рассмотренных ранее методов, в процедурных методах становится возможным учёт различных физических явлений, таких как влияние ветра, гравитации, поверхностного натяжения воды и т.д.

### 1.2.1 Волны Герстнера

Данный метод описан в работе Джерри Тессендорфа [5]. Как и в описанных ранее, в данной модели используется деление визуализируемой поверхности регулярной сеткой. Однако, в данной модели принципиально важно использование горизонтального смещения узлов сетки.

Для получения достоверных результатов узлы сетки должны совершать круговые движения в то время, когда она находится на пути прохождения волн. Зададим такое движение системой уравнений (6). Оно позволяет вычислить новые координаты для каждой точки регулярной сетки (подразумевается, что изначально все точки имеют нулевое значение высоты).

$$\begin{cases} \vec{v} = \vec{v}_0 - \frac{\vec{k}}{|\vec{k}|} \cdot A \cdot \sin(\vec{k} \cdot \vec{v}_0 - \omega t) \\ y = A \cdot \cos(\vec{k} \cdot \vec{v}_0 - \omega t) \end{cases} \quad (6)$$

Где  $\vec{v}_0$  – двумерный вектор, содержащий начальные координаты точки поверхности в плоскости  $XZ$ ,  $\vec{v}$  – двумерный вектор, содержащий новые координаты точки поверхности в плоскости  $XZ$ ,  $A$  – амплитуда рассматриваемой волны,  $\vec{k}$  – волновой вектор длины  $|\vec{k}| = \frac{2\pi}{\lambda}$  (далее – волновое число),  $\lambda$  – длина волны,  $\omega$  – фазовая скорость.

Значение фазовой скорости вычисляется на основе закона дисперсии по одному из следующих уравнений. Уравнение (7) применяется в случае, если расстояние до дна велико, а волны имеют длину более 1 сантиметра. Для визуализации воды в мелководье применяется уравнение (8). Для работы с волнами длиной менее 1 сантиметра применяется уравнение (9).

$$\omega(k) = \sqrt{gk} \quad (7)$$

$$\omega(k) = \sqrt{gk \cdot \tanh(kD)} \quad (8)$$

$$\omega(k) = \sqrt{gk(1 + k^2 L^2)} \quad (9)$$

Где  $g = 9.8 \text{ м/с}^2$  – ускорение свободного падения,  $k$  – волновое число,  $D$  – расстояние до дна,  $L$  – длина волн.

При необходимости получить периодическую модель, необходимо дополнительно преобразовать полученную функцию частоты по формуле (10). Это может быть использовано для повышения качества получаемой модели.

$$\omega(k) = \left\lfloor \frac{T \cdot \omega_0(k)}{2\pi} \right\rfloor \cdot \frac{2\pi}{T} \quad (10)$$

Где  $T$  – период модели,  $\omega_0(k)$  – одна из определённых выше функций вычисления частоты.

Также, от значения произведения  $kA$  зависит форма волны. Когда это значение равно нулю, волна вырождается в плоскость. Плавно увеличивая это значение до единицы, получаемая волна будет заостряться к вершинам. Продолжая увеличивать это значение, волна начнёт образовывать петлю на вершине, что также является вырожденным случаем. Данный эффект показан для сечения волны вертикальной плоскостью на рисунке 3. Цифрами указано значение  $kA$  для каждого графика. Таким образом, это значение должно принадлежать промежутку от 0 до 1.

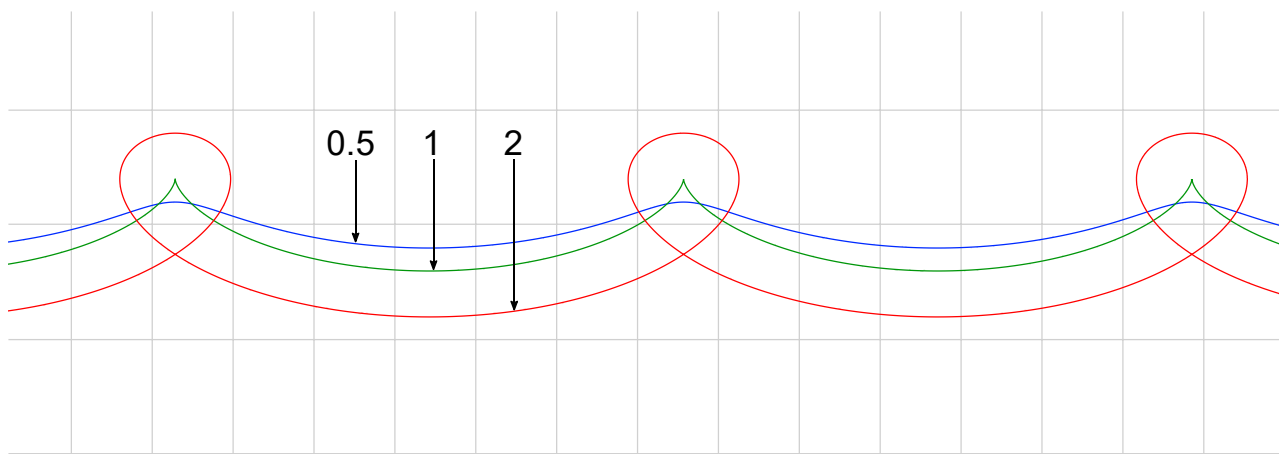


Рисунок 3: Сечения волн Герстнера

Описанный метод позволяет моделировать только одну волну на поверхности. Однако, аналогично методу суммы синусов, возможно просуммировать несколько волн. В таком случае, система будет описываться формулами (11).

$$\begin{cases} \vec{v} = \vec{v}_0 - \sum \frac{\vec{k}_i}{|\vec{k}_i|} \cdot A_i \cdot \sin(\vec{k}_i \cdot \vec{v}_0 - \omega_i t + \phi_i) \\ y = \sum A_i \cdot \cos(\vec{k}_i \cdot \vec{v}_0 - \omega_i t + \phi_i) \end{cases} \quad (11)$$

Где  $\phi_i$  – фазовый сдвиг волн, остальные параметры определяются аналогично формуле (6).

### 1.2.2 Статистический метод

Данный подход также описан в [5] и заключается в разложении функции высоты узлов на сумму синусоид с комплексной, зависящей от времени амплитудами, как показано на формуле (12).

$$y = \sum_k \tilde{h}(\vec{k}, t) \cdot \exp(i \cdot (\vec{k} \cdot \vec{v}_0)) \quad (12)$$

Где  $\tilde{h}$  – функция, определяющая поле высот водной поверхности,  $\vec{k}$  и  $\vec{v}_0$  – волновой вектор и двумерные координаты узла сетки, определённые аналогично формуле (6),  $t$  – параметр времени,  $i$  – мнимая единица.

Определение функции  $\tilde{h}$  является ключевым шагом, влияющим на свойства получаемой модели.

В статье [1] предлагается два варианта определения этой функции. Оба они построены по схеме, указанной на формуле (13), и различаются лишь выбором функции  $h_0$ . Вариант этой функции, предложенный Пирсоном и Москвицом представлен на формуле (14). Её улучшение, предложенное Хассельманом, показано на формуле (15).

$$\tilde{h}(\vec{k}, t) = h_0(\vec{k}) \cdot \exp(i\omega t) \quad (13)$$

$$h_0(\vec{k}) = h_{pm}(\vec{k}) = \frac{0.0081 g^2}{16 \pi^4 f^5} \exp\left(-\frac{5}{4} \left(\frac{f_m}{f}\right)^4\right), f_m = \frac{0.13g}{U_{10}} \quad (14)$$

$$h_0(\vec{k}) = h_{pm}(\vec{k}) \cdot \exp\left(\ln(3.3) \cdot \exp\left(\frac{-(f - f_m)^2}{2\sigma^2 f_m^2}\right)\right), \sigma = \begin{cases} 0.07, f \leq f_m \\ 0.09, f > f_m \end{cases} \quad (15)$$

Где  $\vec{k}$  – волновой вектор,  $\omega$  – фазовая скорость, определённая также, как и для волн Герстнера,  $t$  – параметр времени,  $f$  – частота волны,  $f_m$  – пик частоты,  $U_{10}$  – скорость ветра на высоте 10 метров от поверхности воды.

Другой подход, также описанный в [5], основан на использовании случайных чисел. В нём функция  $\tilde{h}$  определяется по формуле (16), в которой используется функция  $h_0$  (а также комплексно сопряжённая с ней  $h_0^*$ ), определённая по формуле (17). В последней формуле также используется спектр Филиппса  $P_h$ , определяемый формулой (18).

$$\tilde{h}(\vec{k}, t) = h_0(\vec{k}) \cdot \exp(i\omega t) + h_0^*(-\vec{k}) \cdot \exp(-i\omega t) \quad (16)$$

$$h_0(\vec{k}) = \frac{1}{\sqrt{2}} (\xi_r + i\xi_i) \sqrt{P_h(\vec{k})} \quad (17)$$

$$P_h(\vec{k}) = A \cdot \frac{\exp\left(-(|\vec{k}| \cdot L)^{-2}\right)}{|\vec{k}|^4} \cdot |\cos(\phi)|^2, L = \frac{V^2}{g} \quad (18)$$



Где  $\xi_r$  и  $\xi_i$  – случайные числа, полученные с математическим ожиданием 0 и стандартным отклонением 1,  $\phi$  – угол между волновым вектором и вектором скорости ветра,  $V$  – модуль скорости ветра.

Таким образом, данный метод может гибко изменяться путём выбора функций  $\tilde{h}$  и  $h_0$ , а также за счёт используемых параметров, таких как скорость ветра над поверхностью воды. Лучших результатов также можно достичь, используя горизонтальные смещения узлов сетки для получения эффекта заострённых волн.

Таким образом, данный алгоритм позволяет достичь лучших результатов, по сравнению с остальными алгоритмами, представленными в данной работе.

## 2 Освещение и текстурирование

Следующим за определением геометрии шагом является определение цвета треугольников регулярной сетки. Для этих целей необходимо определить модель освещения, рассчитать изображения объектов под водой и дна (при их наличии и видимости), а также – отражения надводных объектов и неба.

Сложность решения данной задачи определяется присутствием и характером посторонних объектов в модели. Наиболее простым случаем является модель достаточно глубокого океана (дно которого не видно) без посторонних объектов и одноцветным небом. Самым сложным – мелководье с большим количеством объектов как под водой, так и над водой.

### 2.1 Модель Блинна-Фонга

Модель Блинна-Фонга проста в реализации, в связи с чем она используется на первом этапе визуализации. Несмотря на то, что в ней напрямую не учитываются отражения и преломления света, она позволяет получить приемлемое изображение модели как для отладки моделей геометрии, так и для использования в задачах, не требовательных к качеству изображения.

Для работы с данной моделью необходимо определить характеристики источника (или источников) света, свойства материала полигонов. Они являются константными показателями. Также необходимо определить нормали к полигонам сетки. В целях оптимизации, нормали вычисляются только в узлах сетки, и интерполируются на полигоны. Нормали в узлах можно вычислить аналитически (если для этого достаточно данных), либо усреднив нормали ко всем прилегающим в данному узлу полигонам.

Интерполяция нормалей на полигон соответствует тонировке Фонга, однако, в случае критической нехватки вычислительных ресурсов, допустимо использовать тонировку Гуро – при которой цвет будет вычисляться лишь для узлов, и интерполироваться на полигоны. В этом случае, очевидно, дальнейшие действия по тонировке не могут выполняться, поэтому процесс визуализации будет закончен на этом шаге.

Освещение в данной модели складывается из 3 компонент – фонового, диффузного и зеркального освещения. Модель Блинна-Фонга является улучшением модели Фонга, и отличается от неё лишь методом вычисления третьей компоненты. Каждая компонента рассчитывается с учётом соответствующих коэффициентов как со стороны освещаемой поверхности, так и со стороны источника света. Далее упоминание данных коэффициентов будет опущено, для краткости.

Для расчёта фонового освещения необходимо определить лишь факт доступности данного объекта светом. Данная компонента учитывает только ненаправленный свет. Однако, обычно, её влияние не велико, в связи с чем, её вычисление не заслуживает отдельного внимания

Интенсивность диффузного освещения рассчитывается как скалярное произведение между вектором нормали к поверхности и вектором – направлением на источник света. Данная компонента даёт основной вклад в итоговый цвет полигона.

В модели освещения Фонга интенсивность зеркального освещения рассчитывается как скалярное произведение между вектором, направленным на точку наблюдателя (камеру), и вектором отражённого света. Однако данный подход ведёт себя некорректно в случаях, когда результат этого векторного произведения меньше нуля. Эта проблема была решена в модели Блинна-Фонга. В ней интенсивность равна векторному произведению между нормалью и медианным вектором – вектором единичной длины, находящимся посередине между направлением на камеру и на источник света.

## ЗАКЛЮЧЕНИЕ

Таким образом. Заключение.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Куров А.В., Завалин А.А. Визуализация водной поверхности на основе спектрального метода // Символ науки. 2017. Том №01-2. С. 71-77.
2. Adam T Lake Real-Time Deep Ocean Simulation on Multi-Threaded Architectures. URL:  
<https://software.intel.com/content/www/us/en/develop/articles/real-time-deep-ocean-simulation-on-multi-threaded-architectures.html> (Дата обращения: 04.10.2020).
3. Мальковский А. Визуализация природных явлений // Введение в компьютерную графику. URL:  
[https://www.graphicon.ru/oldgr/courses/cg02b/assigns/hw-5/hw5\\_wat.htm](https://www.graphicon.ru/oldgr/courses/cg02b/assigns/hw-5/hw5_wat.htm) (Дата обращения 04.10.2020).
4. Hua Li, Huamin Yang, Jianping Zhao Water Reflection, Refraction Simulation Based on Perlin Noise and Ray-tracing // International Journal of Signal Processing, Image Processing and Pattern Recognition. Vol. 10, No. 3 (2017), pp. 63-74.
5. Tessendorf J. Simulating ocean waters. ACM SIGGRAPH Computer Graphics. 2001.