

Technical Design Document

**Realistic Ocean Surface Rendering and Simulation**

Version 1.0

|  |  |  |
| --- | --- | --- |
| **Position Title** | **Name** | **Signature** |
| Programmer | *Spencer McKean* |  |
| Advisor | *Matt Butler* |  |

Table of Contents

[Table of Figures 3](#_Toc68640328)

[Introduction 4](#_Toc68640329)

[Overview 5](#_Toc68640330)

[Scope 5](#_Toc68640331)

[End Product 5](#_Toc68640332)

[Theory 6](#_Toc68640333)

[Navier-Stokes Equation 6](#_Toc68640334)

[Dispersion Relation 6](#_Toc68640335)

[Gerstner Waves 6](#_Toc68640336)

[Fourier Transform 7](#_Toc68640337)

[iWave 7](#_Toc68640338)

[Artifact 8](#_Toc68640339)

[Gerstner Waves 8](#_Toc68640340)

[Discrete Fourier Transform 8](#_Toc68640341)

[Fast Fourier Transform 8](#_Toc68640342)

[Reflections and Skybox 8](#_Toc68640343)

[iWave Implementation 8](#_Toc68640344)

[Implementation 8](#_Toc68640345)

[Architecture 8](#_Toc68640346)

[Results 9](#_Toc68640347)

[DFT versus FFT 9](#_Toc68640348)

[Not Threaded versus Threaded versus Compute Shader 9](#_Toc68640349)

[Adding iWave 9](#_Toc68640350)

[Profiling Results 9](#_Toc68640351)

[Data Analysis 9](#_Toc68640352)

[Bibliography 10](#_Toc68640353)

# Table of Figures

# Introduction

Inspired by the game Sea of Thieves, created by Rare, the idea of this project was to recreate the ocean surface simulation achieved in that game. Using the Fast Fourier Transform (FFT) the project achieves a realistic simulation of the ocean’s surface that runs at interactive speeds even on a large scale. This coupled with an additional simulation method called the iWave method allows objects to move across the surface creating ripples and wakes as well as react to variable ocean floor depths. To discover exactly what performance implications these formulas and methods have profiling data was collected for different wave settings.

# Overview

*In exploration of different wave generation methods used throughout the gaming industry, this thesis sets out to achieve the most realistic wave simulation possible for games. Using statistical models, this FFT method generates waves that behave realistically with finer detail then other methods. In the industry the simple solution for water surface simulation is Gerstner Waves. This methods aims to alleviate issues found within that method. In addition to vertex manipulation the appearance of the water’s surface is also important thus part of the exploration involves the pursuit of effective water rendering techniques.*

## Scope

*The goal of this project is to simulate the ocean’s surface as realistically possible while maintaining performance. This involves getting familiar with common practices and the easier forms of wave generation. The next step is to become more comfortable with the way in which to apply the Fourier Transform to the mesh vertices as effectively as possible. Once the mesh is generated the focus can shift to making the manipulation of the vertices performant, requiring the use of the FFT. After this the goal is to tackle the refraction and reflection and its effect on the visuals looking at and through the water’s surface. This leads to a need for terrain and back buffer manipulation. Once this is accomplished, the goal can then become to implement interactive waves using the iWave method.*

## End Product

*Gameplay*

* *Flying, no-clip, scene camera*
* *Wave Simulation and scene control hot key access*

*Game Objects*

* *Ocean Surface*
* *Terrain*
* *Interactive Water Objects*

*HUD & UI*

* *Runtime benchmark averages*
  + *Total Update time*
  + *FFT Time*
  + *FPS*
  + *Render Time*
* *Current Simulation Values*
  + *Wind Speed*
  + *Wind Direction*
  + *Global Wave Amplitude*
  + *Wave Suppression Value*
  + *Tiled Size*
  + *Choppiness Value*
  + *Surface Dimensions*
  + *Surface Vertex Dimensions*

# Theory

### Navier-Stokes Equation

The Navier-Stokes equations determines the motion of a fluid based on velocity u(x,t), pressure p(x,t), force F(x,t), and density [1]. Where x is the position of the fluid and t is time.

#### Bernoulli’s Equation

Bernoulli’s Equation takes the Navier-Stokes equation and simplifies the complex nature of the equation to only give the velocity one degree of freedom instead of three [2]. Where velocity u is converted into potential flow ɸ.

This allows the Navier-Stokes Equation to become a fully nonlinear equation. Where U(x,t) is some potential energy function [2].

#### Linearization

To achieve interactive speeds for the simulation Bernoulli’s equation needs to be further simplified to linearize the equations of motion and restrict the evaluation to only the points on the surface of the fluid.

### Dispersion Relation

The Navier-Stokes Equation, simplified and with approximated values can be expressed in a single equation imposing the constraint that the temporal frequency ω of surface height movements is connected to the spatial extent of the propagating wave k = |k|.

### Gerstner Waves

A simple implementation for modeling the surface of the ocean as a wave passes over a point. A good approximation of any point on the surface of an ocean reacting to a wave is in a circular motion [2]. This means that the equations boil down to simple sin and cos equations.

Where x is the point on the horizontal plane, x(x,y), and the height of the surface at that point is z. Where k is the magnitude of the wave, A is the amplitude of the wave, and the length of the wave is λ.

This equation for a single sin wave can be incorporated into a summation that allows for the modeling of a more complex wave pattern. Where ɸ is the wave’s phase.

### Fourier Transform

The Fourier Equation used on a wave vector field to determine the height of the waves at time, , for the point . For the summation of all wave vectors, , described in the form of . Where and . Where and are the dimensions of the water’s surface in the respective direction, and where and are the index of the vertex in the x and y direction respectively. To have sinusoidal behavior in the waves and are defined as such: and , where N is the total number of samples in the x direction and M is the total number of samples in the y direction. is the position of the point in space and defined .

#### Fourier Amplitudes at Time

To find the Fourier amplitudes at time, the equation above is used. is the precomputed Fourier amplitude for the vector . is the conjugate of that value and allows waves to propagate to the left and right. is the dispersion relation for the vector .

#### Fourier Amplitudes

The Fourier amplitudes at the vector can be described with this equation where is a random complex number. A good value for this number is a Gaussian distributed random number as waves tend to follow along with the numbers produced. However, other forms of random numbers can be used. is the Phillips spectrum for the vector .

#### Phillips Spectrum

The Phillips Spectrum is a good model for wind driven waves that are fairly large in an ocean like environment. This equation accounts for wind in the term and where is the wind’s speed and is the gravitational constant. is the direction of the wind. is a term used to represent the global wave amplitude. In order to suppress bad values below a certain threshold a multiplicative modifier can be added to this equation.

### Jacobian Matrix

### iWave

The iWave method is designed to simulate minor changes in an already simulated ocean surface. Its goal is to create interactive waves. This means that it accounts for objects in the water, shallow water, and creating ripple effects.

# Implementation

### Gerstner Waves

The Gerstner wave model has been used for a long time in games and is a simple approach to solving the problem of producing realistic waves. Waves follow a sinusoidal pattern and this method demonstrates that characteristic. However, there are a few draw backs to this methods, the first of which is that for every point, on update, needs to calculate each wave’s height at that time and add them together. The other drawback is that when values get too high, the waves start to fold in on themselves. Both of these in conjunction make it difficult to produce a complex wave surface.

Due to the simplicity of this model it is very easy to get working in code. To get a good grasp of what it takes to manipulate vertices on a mesh in relation to equations and to gain a better understanding what kind of effect waves have on the mesh, an implementation of this method is very beneficial.

### Discrete Fourier Transform

The Discrete Fourier Transform (DFT) is a naïve approach to solving the Fourier problem. To get the height of the water’s surface at any given point the Fourier Amplitude must be calculated for every wave vector and added to the total resulting in the height at that point. This happens for every point and results in a complexity of .

Taking this approach before trying the FFT allows for architecture to be put in place to calculate the values necessary as well as gain a better understanding of what the goal of the Fourier Transform is. Since the equations are the same for both FFT and DFT problems can be debugged easier with the straight forward approach in the DFT.

### Fast Fourier Transform

### Reflections and Refractions

### iWave

#### Objects

#### Environment

# Results

### DFT versus FFT

### Adding iWave

# Profiling Results

*Add sections under this heading as appropriate to describe any data, test results or conclusions.*

# Data Analysis

*Add sections under this heading as appropriate to describe any data, test results or conclusions.*

# Bibliography

|  |  |
| --- | --- |
| [1] | "Navier-Stokes Equations," Comsol, 22 February 2017. [Online]. Available: https://www.comsol.com/multiphysics/navier-stokes-equations. [Accessed 9 July 2020]. |
| [2] | J. Tessendorf, "Simulating Ocean Water," 2004. [Online]. Available: https://people.cs.clemson.edu/~jtessen/reports/papers\_files/coursenotes2004.pdf. |