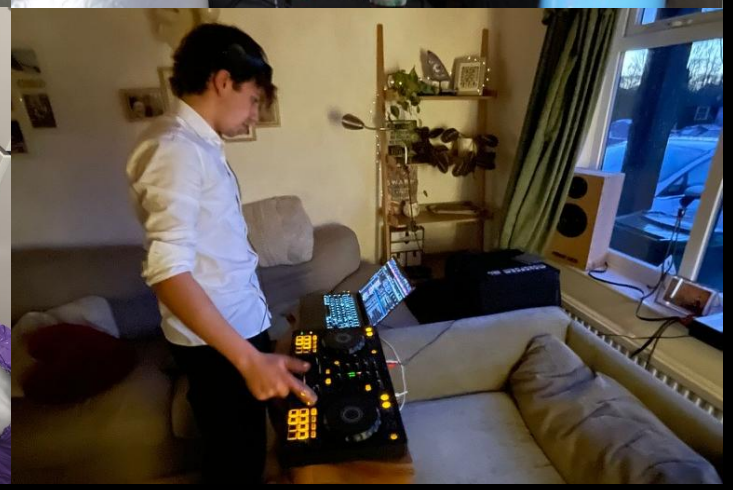
WaveRadar Visualisation

KIRAN GOPEE

About Me

Kiran



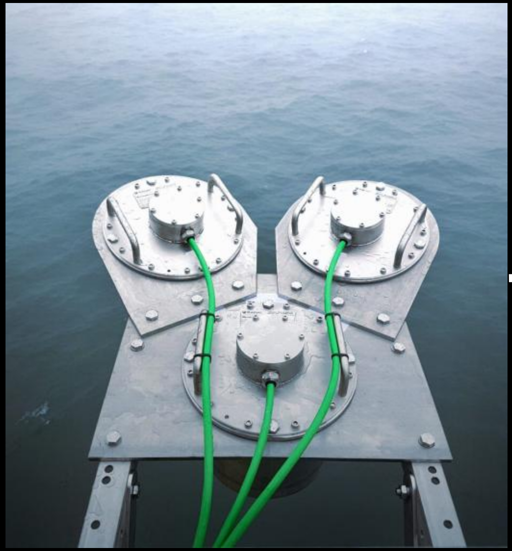
How the project started

EEEGR Internship

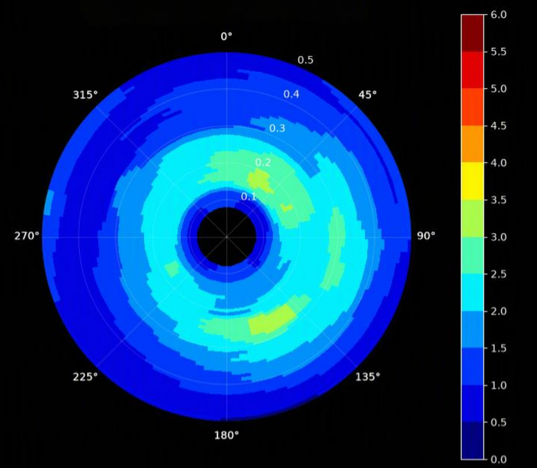
The project

Overview

The goal of the project is to take the data measured by the WaveRadars and turn it into interactive simulations.



time=1785743400000;sensor=direction;Dmem_51x91=NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,
NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,
NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,
NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,
NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,
NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,
NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN,18.941874,20.0
3929,20.493038,20.22796,19.325506,17.980513,16.417057,14.820981,13.313957,11.958946,10.777799,9.768357,8.916756,8.204878,7.6143
72,7.128536,6.733047,6.416077,6.168165,5.9819846,5.852113,5.7748394,5.74802,5.7709966,5.8445835,5.9711204,6.154591,6.4008303,6.7
17796,7.115945,7.6086526,8.21267,8.948474,9.840255,10.915003,12.199701,13.714849,15.461821,17.401466,19.424664,21.325743,22.807
047,23.54978,23.347504,22.219858,20.406158,18.242607,16.02798,13.958056,12.127617,10.561653,9.2469845,8.154104,7.249492,6.50149
8,5.882617,5.369933,4.9448113,4.592281,4.3004146,4.059752,3.862827,3.7037737,3.578036,3.4821277,3.41346,3.370216,3.3512568,3.356
0731,3.3847535,3.4379878,3.5170968,3.6240923,3.7617605,3.933795,4.1449614,4.401313,4.71046,5.081898,5.527372,6.0612583,6.700863
4,7.4664607,8.380697,9.466689,10.743713,12.218892,13.8733015,15.642654,17.398075,18.941874,10.680643,10.263942,9.743701,9.12256
6,8.420672,7.6790223,6.9443936,6.254466,5.6318593,5.0855618,4.615268,4.215568,3.8789167,3.597405,3.363669,3.1712759,3.0148337,2.
8899577,2.7931871,2.7218919,2.6741843,2.6488597,2.6453521,2.663725,2.704688,2.7696517,2.8608172,2.9813173,3.1354172,3.3287945,3
.5689168,3.8655496,4.231419,4.6830597,5.241833,5.9350405,6.796806,7.8679085,9.192362,10.805033,12.696405,14.731235,16.536396,17.
559208,17.52901,16.780766,15.829809,14.893642,13.898296,12.718328,11.355195,9.933456,8.593367,7.4191213,6.4350615,5.6298723,4.9
782114,4.452752,4.029167,3.6874921,3.4119685,3.1903975,3.0134363,2.8739886,2.766714,2.6876638,2.6340098,2.6038527,2.596097,2.61
0376,2.6470218,2.707083,2.792382,2.9056206,3.050546,3.2321815,3.457144,3.7340424,4.0739655,4.49095,5.0021725,5.627118,6.383994,7
.279978,8.290852,9.331608,10.243725,10.850262,11.072198,10.975323,10.680643,15.230608,12.288147,10.203645,8.689709,7.5658045,6.7
158046,6.0633326,5.5571136,5.162069,4.853793,4.6150765,4.433677,4.30087,4.2104983,4.158342,4.141692,4.159083,4.210106,4.2953086,
4.416132,4.574889,4.774753,5.0197506,5.3147297,5.6652737,6.077516,6.557841,7.112426,7.7467346,8.465231,9.271969,10.17337,11.1851
8,12.346171,13.741032,15.532858,17.996243,21.497015,26.201412,31.098976,33.307587,31.294403,27.293991,23.273388,19.523169,16.10
979,13.236306,10.978251,9.262804,7.969368,6.9884624,6.2366743,5.6543036,5.199405,4.842491,4.5626717,4.3450108,4.1787457,4.05610
47,3.9715054,3.9210114,3.9019637,3.9127288,3.9525228,4.021295,4.1196475,4.248781,4.4104595,4.606986,4.8411913,5.116439,5.436681
7,5.8066225,6.2320843,6.720839,7.284234,7.9402876,8.719303,9.673804,10.896023,12.548956,14.921471,18.516922,24.116575,32.299706,
40.683487,41.67189,34.19287,25.733162,19.492329,15.230608,11.82719,9.860545,8.659666,7.9231625,7.478091,7.2210627,7.086109,7.027
232,7.0088673,7.001069,6.9779735,6.9186244,6.809059,6.6441627,6.4279838,6.171987,5.8918667,5.604125,5.3234906,5.061586,4.826706
4,4.624321,4.457867,4.329608,4.241421,4.195502,4.1950383,4.2449474,4.352798,4.5301247,4.794449,5.1725254,5.7057157,6.4590282,7.5
36124,9.102396,11.41022,14.775487,19.290024,23.857609,25.535118,22.84045,18.300919,14.388852,11.668105,9.899911,8.768413,8.0435
91,7.577523,7.2782364,7.0892487,6.97585,6.915649,6.8918686,6.888846,6.889741,6.8764668,6.8316026,6.7415304,6.5994344,6.406806,6.
172725,5.911284,5.6383333,5.368724,5.1146736,4.8852773,4.686821,4.5234957,4.3982463,4.313596,4.272422,4.27874,4.338594,4.461239
3,4.6608953,4.9595594,5.391776,6.013069,6.9153733,8.256022,10.311946,13.570713,18.803328,26.60493,34.664406,35.52574,28.26883,20
.425438,15.083925,11.82719,6.792093,6.5649385,6.349601,6.1431093,5.9437246,5.7508125,5.5647373,5.386695,5.218505,5.062361,4.9205
894,4.7954693,4.6891174,4.603468,4.5403137,4.5013995,4.488557,4.5038457,4.549728,4.629244,4.7462044,4.9053984,5.112803,5.375758
6,5.7030554,6.1047535,6.591463,7.172561,7.8525844,8.624938,9.462862,10.310227,11.079791,11.669775,12.001477,12.0584755,11.89591
6,11.611171,11.301091,11.034014,10.843117,10.732357,10.68568,10.675274,10.668297,10.632607,10.541744,10.378698,10.137807,9.8244
36,9.452657,9.041767,8.612608,8.184561,7.7736707,7.3919387,7.0474844,6.7452297,6.4877644,6.276184,6.1108017,5.991721,5.919262,5.
8942966,5.9185143,5.9946337,6.1265726,6.319544,6.579994,6.915237,7.3324504,7.836567,8.42635,9.08809,9.787373,10.462222,11.02515
2,11.382735,11.471071,11.286629,10.887269,10.362379,9.796633,9.249175,8.751009,8.312028,7.929716,7.5958815,7.30081,7.035412,6.79
2093,25.613016,29.863352,26.68061,20.202568,15.229708,12.230053,10.556421,9.714813,9.432497,9.565213,10.020579,10.695131,11.418
352,11.93489,11.986446,11.474844,10.534118,9.4148035,8.326187,7.377502,6.59982,5.984294,5.508158,5.1473413,4.881155,4.693382,4.5
719934,4.5085073,4.497353,4.535383,4.6215363,4.7566576,4.943478,5.1867576,5.493651,5.8744125,6.3436675,6.9227295,7.643881,8.558
46,9.752197,11.3735695,13.6796875,17.069632,21.866121,27.046597,28.72295,25.170307,20.264683,16.593958,14.296293,12.972088,12.3
2563,12.203516,12.53763,13.281439,14.327574,15.403997,16.0427,15.803688,14.653675,12.990614,11.273249,9.764931,8.542117,7.58716
1,6.8534565,6.2940097,5.870158,5.552442,5.3191285,5.154404,5.046849,4.9882836,4.9729548,4.996986,5.058055,5.1552725,5.2892814,5.
4626517,5.680699,5.952995,6.296042,6.7380114,7.3273225,8.1486025,9.3530655,11.215189,14.219451,19.048054,25.613016,34.343742,36
.94183,32.111366,24.153109,17.89669,13.773392,11.076403,9.248558,7.958432,7.0145617,6.3033686,5.754842,5.3241706,4.981703,4.7072
61,4.486801,4.310387,4.1709166,4.0633082,3.9839745,3.930471,3.9012747,3.8956423,3.9135432,3.955637,4.0233035,4.118732,4.2450585,
4.406595,4.609155,4.860532,5.171196,5.5553236,6.032305,6.629019,7.383251,8.34883,9.603212,11.25798,13.470966,16.45202,20.436157,
25.56813,31.661228,37.94246,42.595757,42.165398,36.060303,28.946415,23.642925,20.215197,18.092726,16.79515,15.994872,15.471254,
15.073233,14.699244,14.287346,13.80866,13.260256,12.656691,12.021397,11.379599,10.753674,10.160987,9.613517,9.118562,8.679857,8.
298741,7.9751697,7.708549,7.4983864,7.344822,7.2491083,7.2140803,7.24472,7.34886,7.538128,7.8292356,8.245696,8.819998,9.595986,1
0.630478,11.991417,13.7469635,15.93836,18.542892,21.48854,24.84577,29.096886,34.343742,21.288984,19.368114,17.549269,15.998262,
14.752129,13.787194,13.062429,12.538858,12.186045,11.983309,11.919352,11.991862,12.207782,12.584579,13.153018.941874,20.03929,2
0.493038,20.22796,19.325506,17.980513,16.417057,14.820981,13.313957,11.958946,10.777799,9.768357,8.916756,8.204878,7.614372,7.12
8536,6.733047,6.416077,6.168165,5.9819846,5.852113,5.7748394,5.74802,5.7709966,5.8445835,5.9711204,6.154591,6.4008303,6.717796,7
.115945,7.6086526,8.21267,8



How we got there

Approaches

- A line view from the side using:
  - Hm0 and Fp
  - Czz10 spectrum (table of waves moving in a single direction)

- A polar plot of the oceanographic spectrum using the Directional Maximum Entropy Method (directional wave energy spectrum)

- A 3D visualisation using the DMEM spectrum and Unity

2D Line View

Using Hm0 and Tp

Understanding the Data

Hm0 and Tp

From the WaveRadar:

Hm0

Significant wave height

Derived from M0

Fp

Wave frequency with the highest energy

Calculated:

Tp

Peak period from Fp

Creating waves

How do we represent them?

Creating waves

Mathematics

Wave Equation

$$y(x,t) = A\ \sin(kx - \omega t + \phi)$$

$$y(\mathrm{x},\mathrm{t}) = A\ \sin(kx\ -\ \omega\mathrm{t} + \phi)$$

y – height at a point on the wave

A – amplitude of the wave

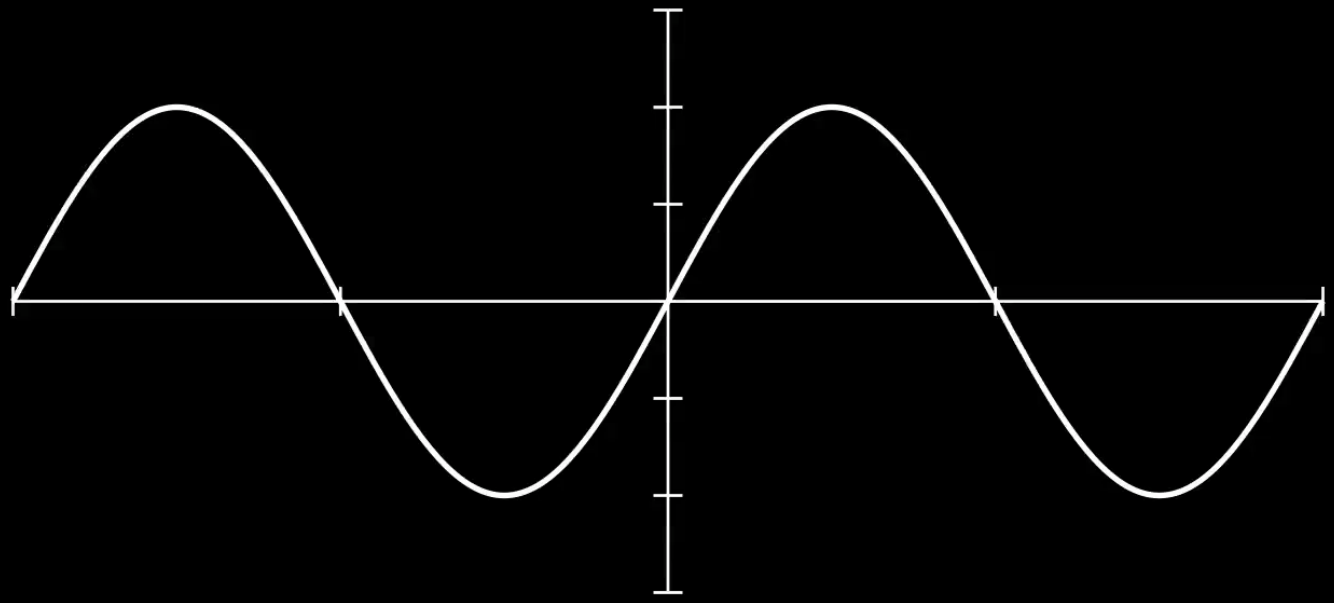
k – wave number

x – x-coordinate

ω – angular frequency

t – time

ϕ – the phase



Creating waves

Summary

We now know:

- Waves can be represented using an equation
- We can manipulate the wave by changing the parameters in its equation

2D Visualisation

Hm0 and Tp

Parameters

Hm0 = 1.2m Fp = 0.125Hz

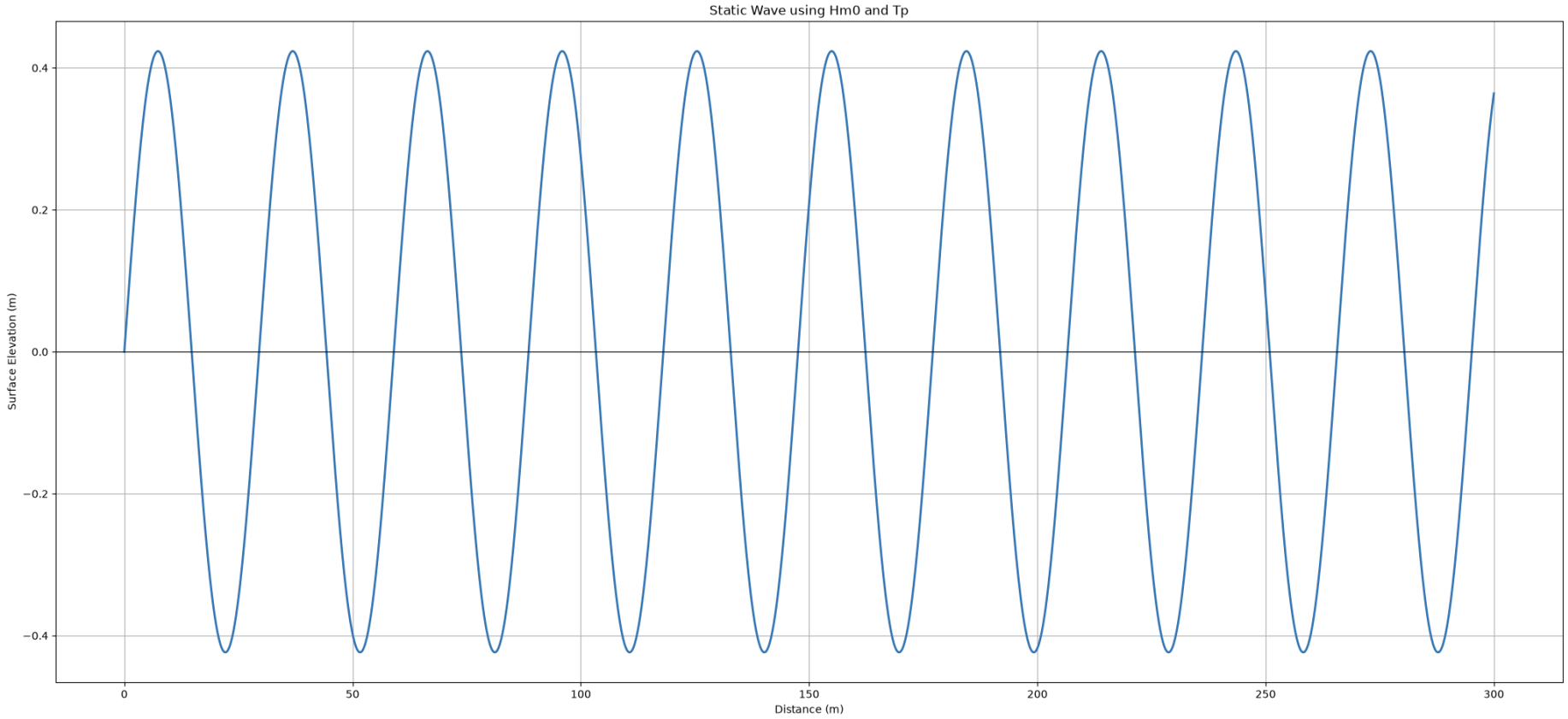
$$A = \frac{Hm0}{2}$$

$$Tp = \frac{1}{Fp}$$

$$\omega = \frac{2\pi}{Tp}$$

2D Visualisation

Hm0 and Tp

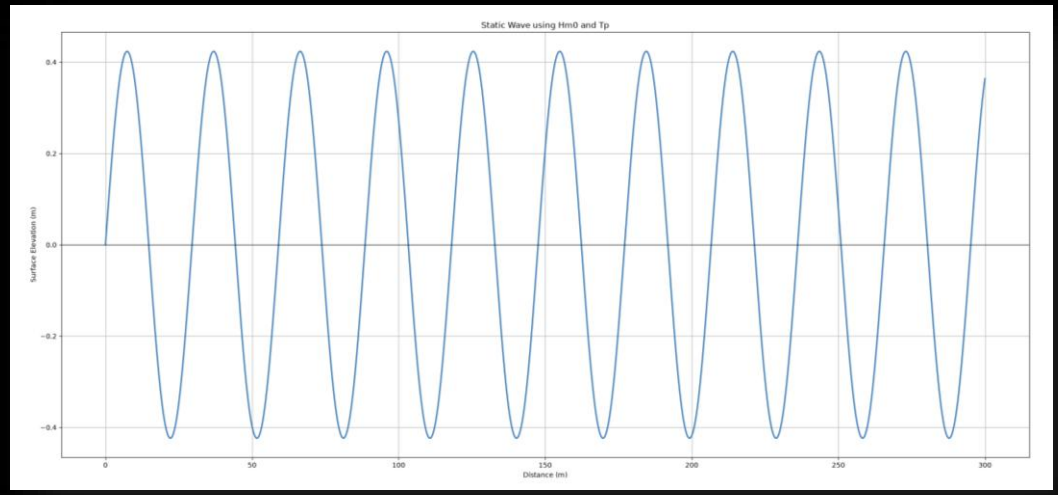


2D Visualisation

Summary

Issues

- Only the significant wave
- Only considers the prominent direction
- Static wave



We now know:

- The Hm0 and Fp parameters from the WaveRadar can be processed
- We take that processed data and use it to modify our wave equation
- It produces a single wave that represents the most significant wave in the sea state

2D Line View

Using the Czz10 Spectrum

Improvements

Compared to Hm0 and Tp

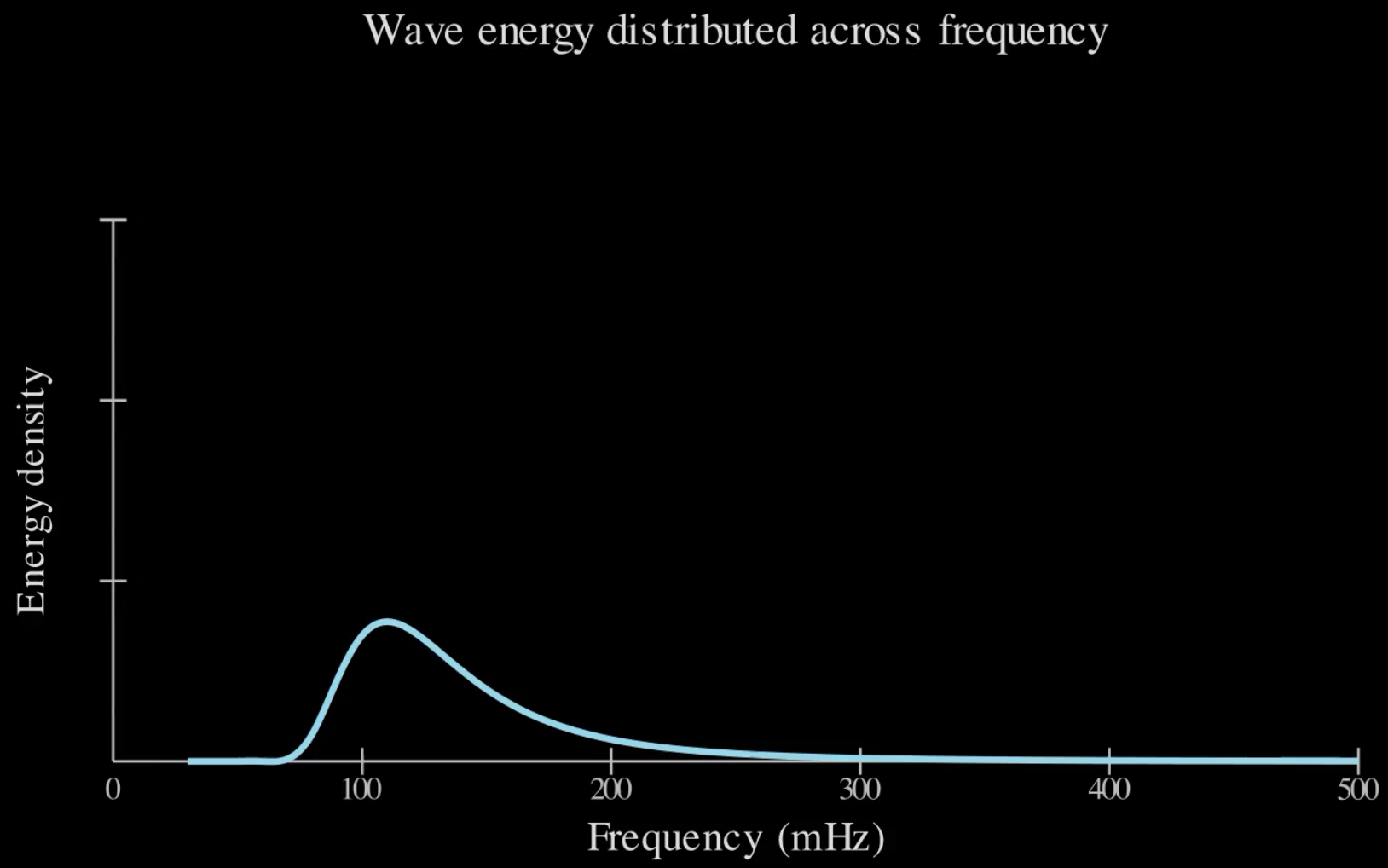
A real sea state is made of many waves, not just one.

We can use data that contains more information that we can split up to create many wave components.

The Czz10 spectrum is ideal for this.

Understanding the Data

Czz10 Spectrum



Understanding the Data

Czz10 Spectrum

Each dot is one discrete data value in the spectrum.

Understanding the Data

Czz10 Spectrum

Each frequency bin in the spectrum becomes one sine wave component.

For each bin:

- Energy density determines amplitude
- Frequency determines wavelength

Then add together all of the sine wave components.

$$\eta(x,t) = \sum_i A_i \sin(k_i x - \omega_i t + \phi_i)$$

Understanding the Data

Czz10 Spectrum

We now have a new term:

$$\eta(x,t) = \sum_i A_i \sin(k_i x - \omega_i t + \phi_i)$$

Phase ϕ_i

Phase describes how far through the cycle a wave is.

Understanding Phase

Mathematics

Understanding Phase

Why use a random phase?

The WaveRadar can’t measure phase.

It has no way of knowing how far through a cycle a wave is, only whether or not an entire cycle has been completed.

This means we have to manually assign phase and it must be random.

Understanding Phase

Random Phase

Understanding the Data

Mathematics

The Czz10 spectrum gives us energy at a range of frequencies, but energy can’t be directly used in the wave equation.

We must convert it into the parameters the wave equation uses.

Understanding the Data

Czz10 Spectrum

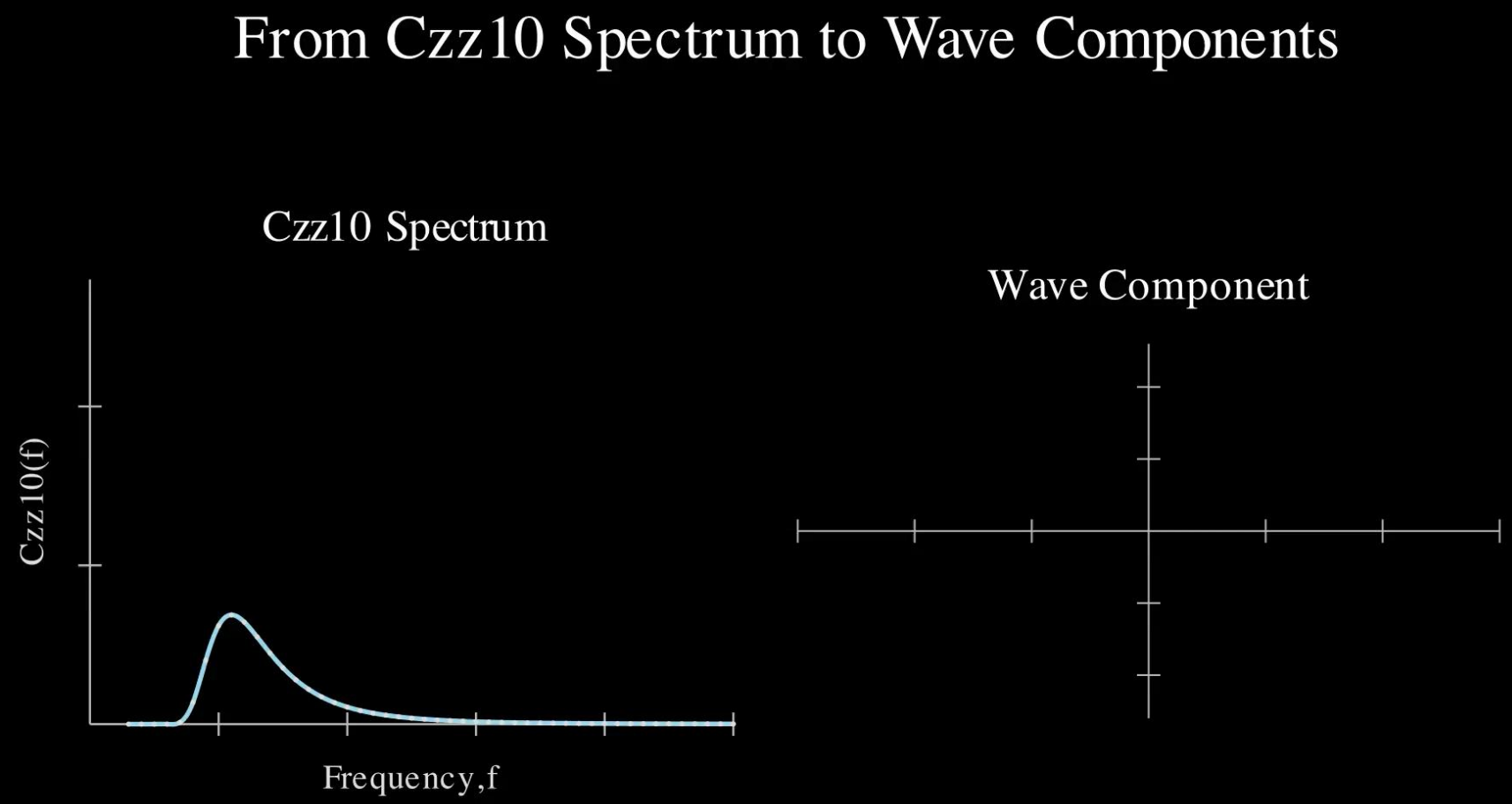
Parameters

$$A = \frac{\sqrt{2S_i\Delta f}}{100}$$

$$\omega_i = 2\pi f_i$$

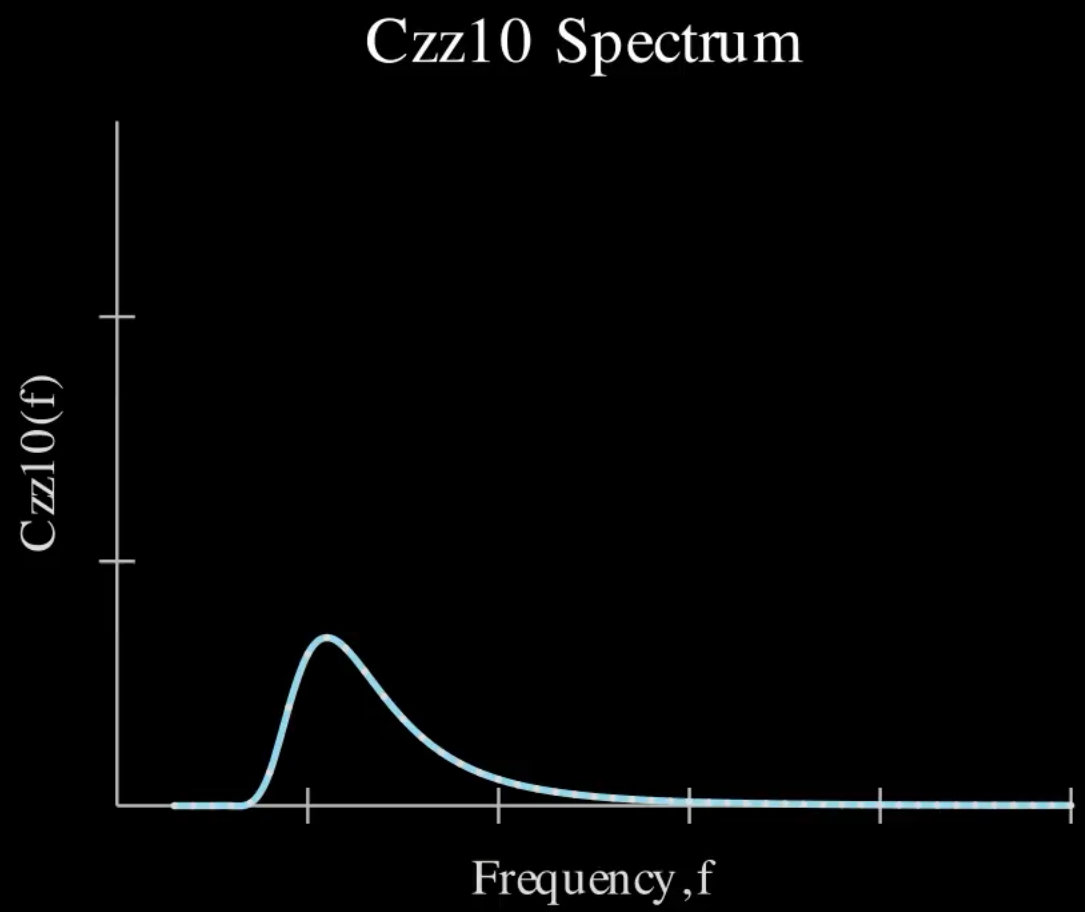
$$k_i = \frac{\omega_i^2}{g}$$

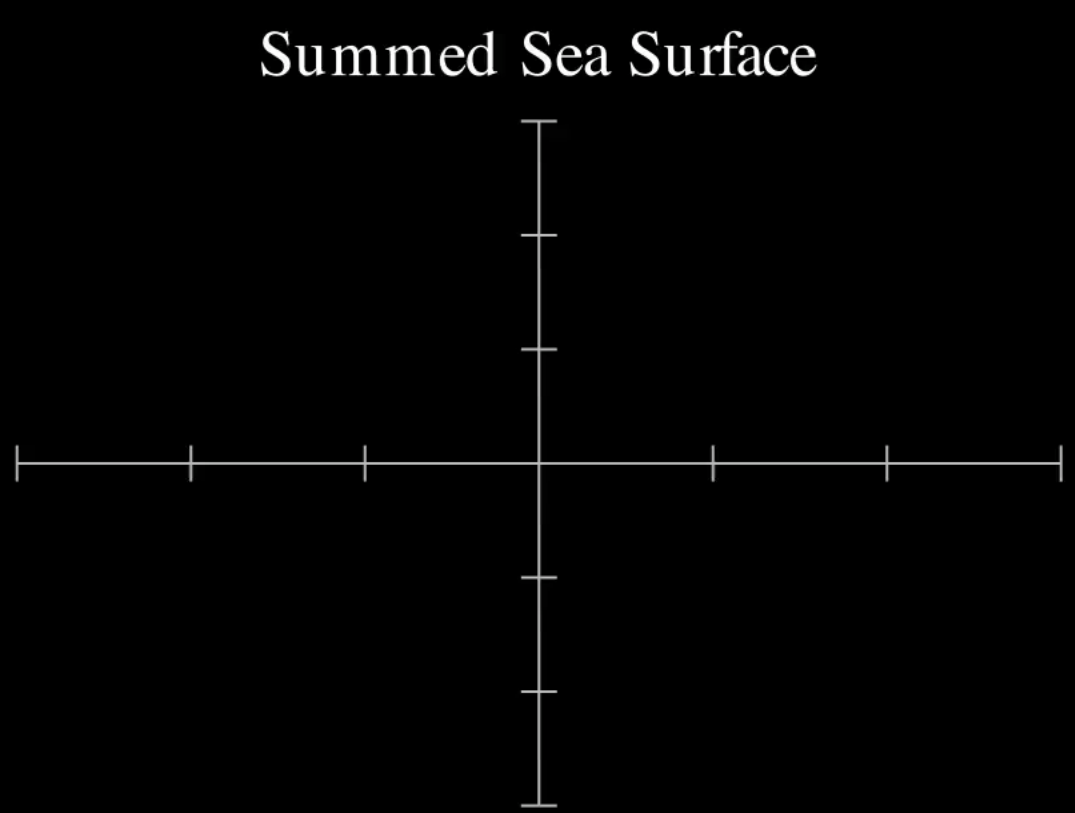
$$\lambda_i = \frac{2\pi}{k_i} = \frac{g}{2\pi f_i^2}$$



2D Visualisation

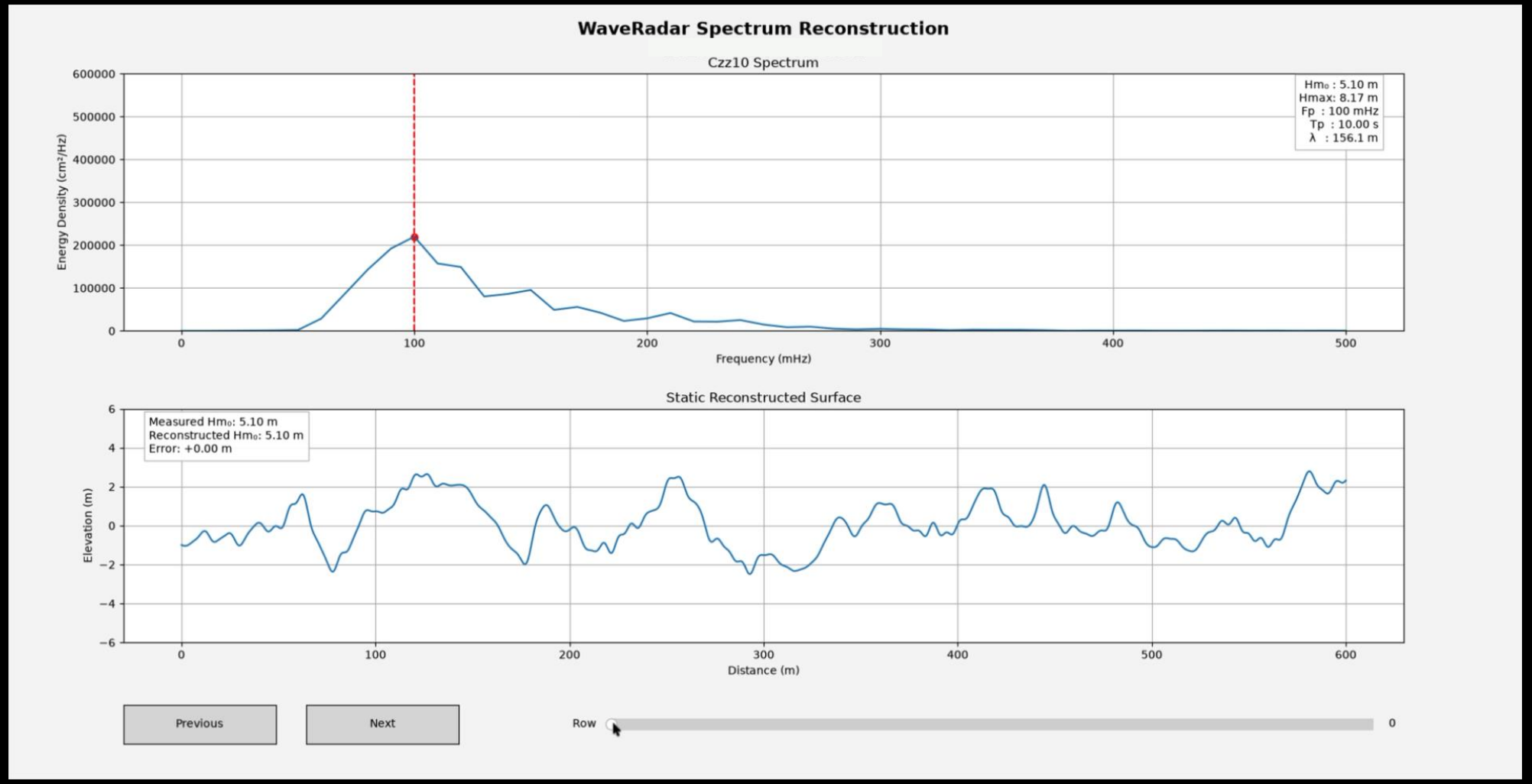
Summing the waves





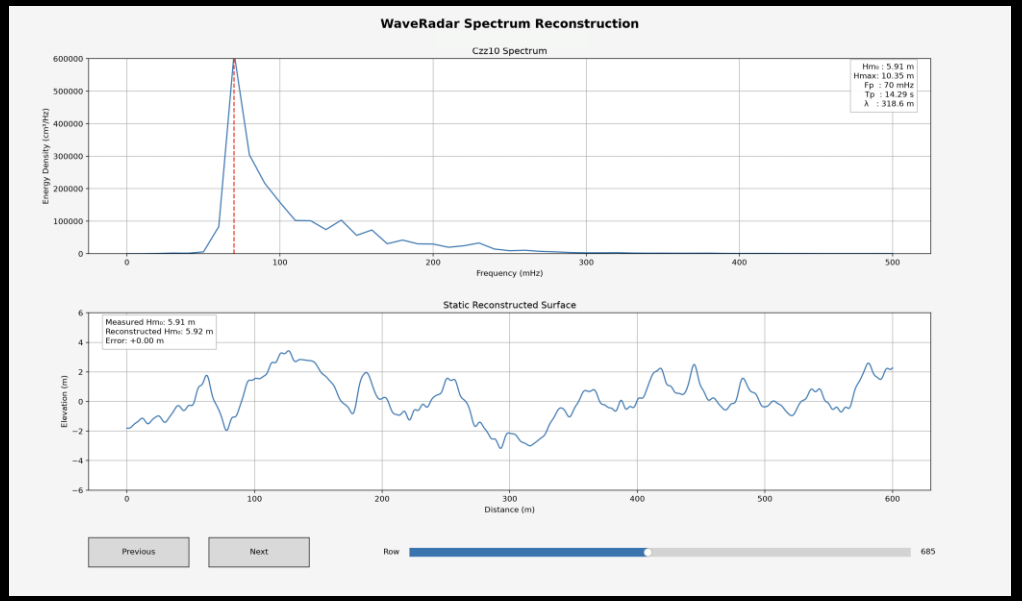
2D Visualisation

Czz10 Spectrum



2D Visualisation

Czz10



Issues

- Only considers one direction
- A side on view of the sea isn’t very descriptive

We now know:

- The Czz10 spectrum tells use how much energy waves have in different frequency ranges
- Processing each value for energy gives us parameters the wave equation can use
- Adding all of these waves together produces a more accurate representation of the sea

DMEM

Polar plot of the oceanographic spectrum

Understanding the Data

DMEM Spectrum

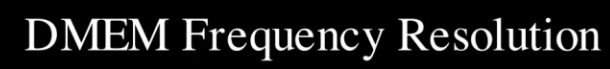
The DMEM describes the sea state in a table.

	0°	4°	8°	…	356°	360°
0mHz	NaN	NaN	NaN	…	NaN	NaN
10mHz	NaN	NaN	NaN	…	NaN	NaN
20mHz	18.941874	20.03929	20.22796	…	13.313957	10.561653
…	…	…	…	…	…	…
490mHz	21.707626	20.20916	17.482151	…	8.40764	9.098132
500mHz	13.489159	11.957963	10.809789	…	8.417857	8.594357

Understanding the Data

DMEM Spectrum

Each bin represents a small range of wave frequencies in a small range of directions and contains the amount of energy associated with that range.

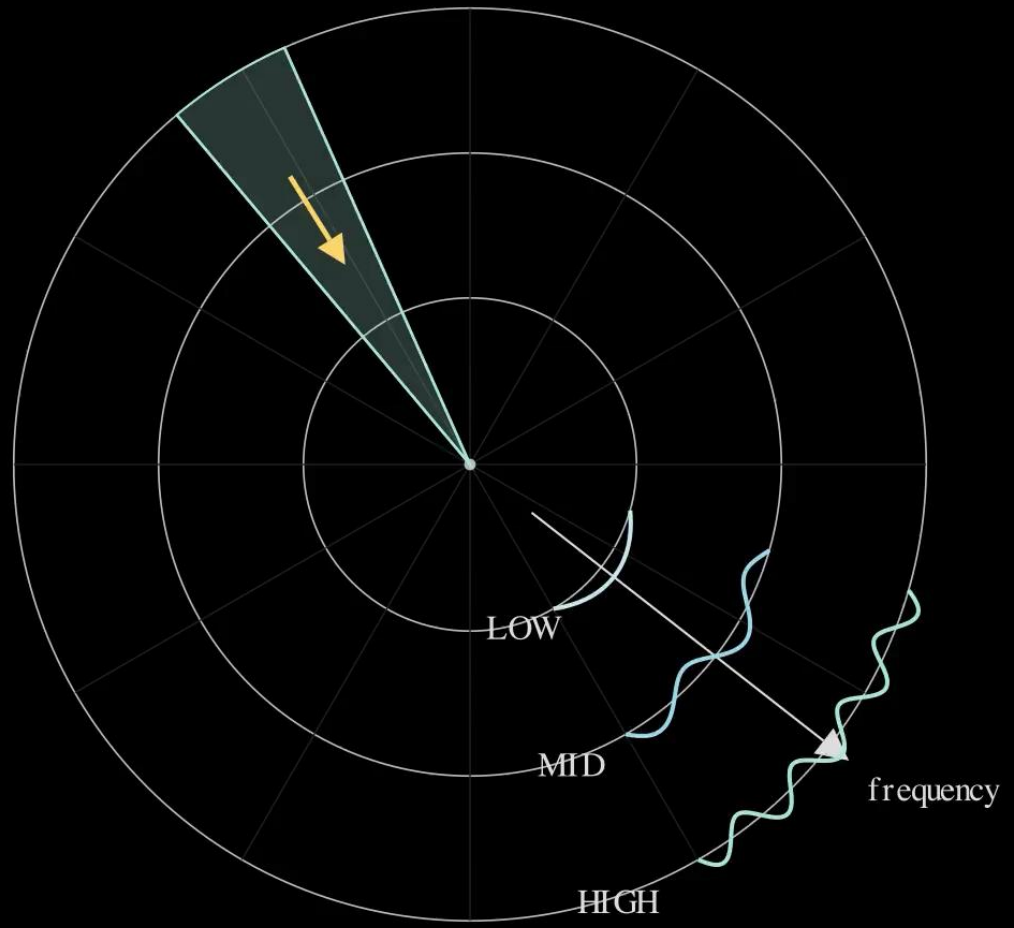


Representing the Data

Polar Plot

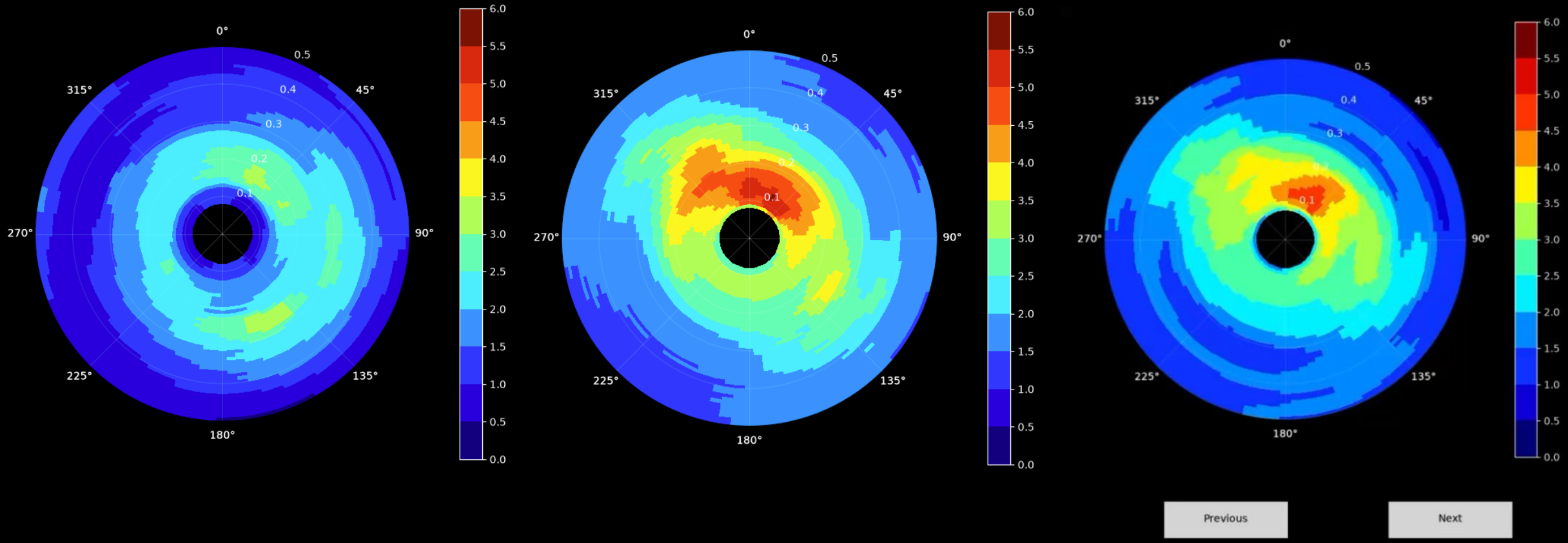
Instead of having an energy density graph for each measured direction, we can use a polar plot to show the energy associated both direction and frequency.

By plotting frequency along the radius of the circle and direction around it, we can see which wave frequencies have the most energy based on their colour and clearly see from which direction they are coming from.



Representing the Data

DMEM Spectrum



Representing the Data

DMEM Spectrum

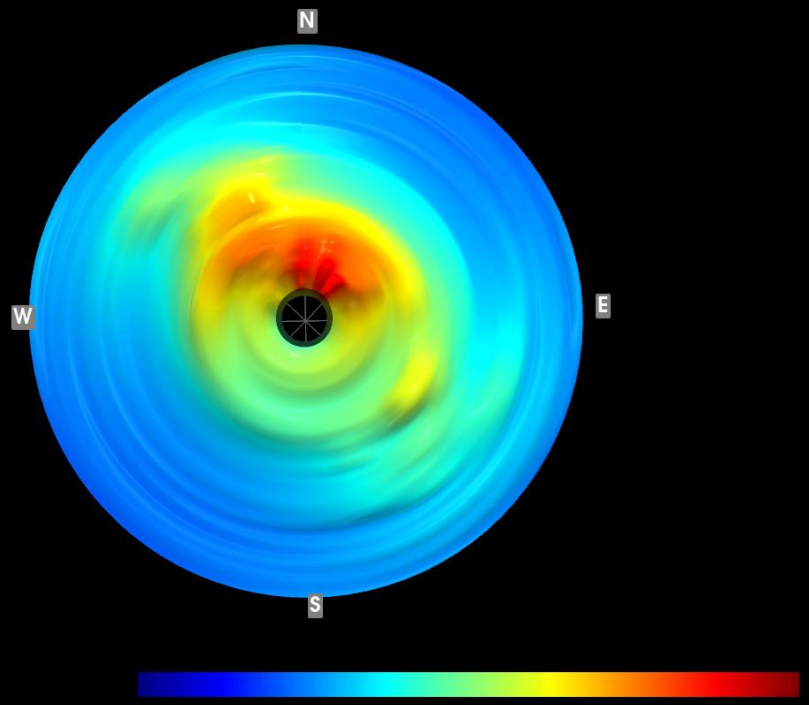
These plots are useful if you understand what they describe.

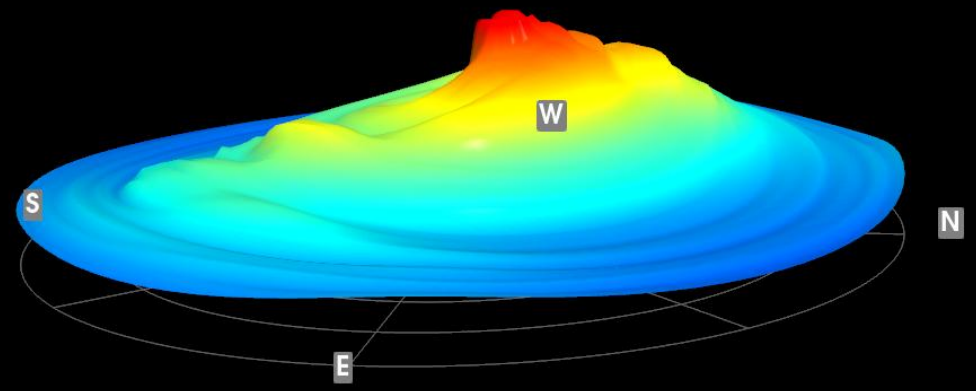
The challenge is to understand that this plot doesn’t tell you how large a wave is, it just tells you which waves have the most **energy.**

To make it clear when a sea contains lots of energy is rough, one approach was to make the plot 3D.

Representing the Data

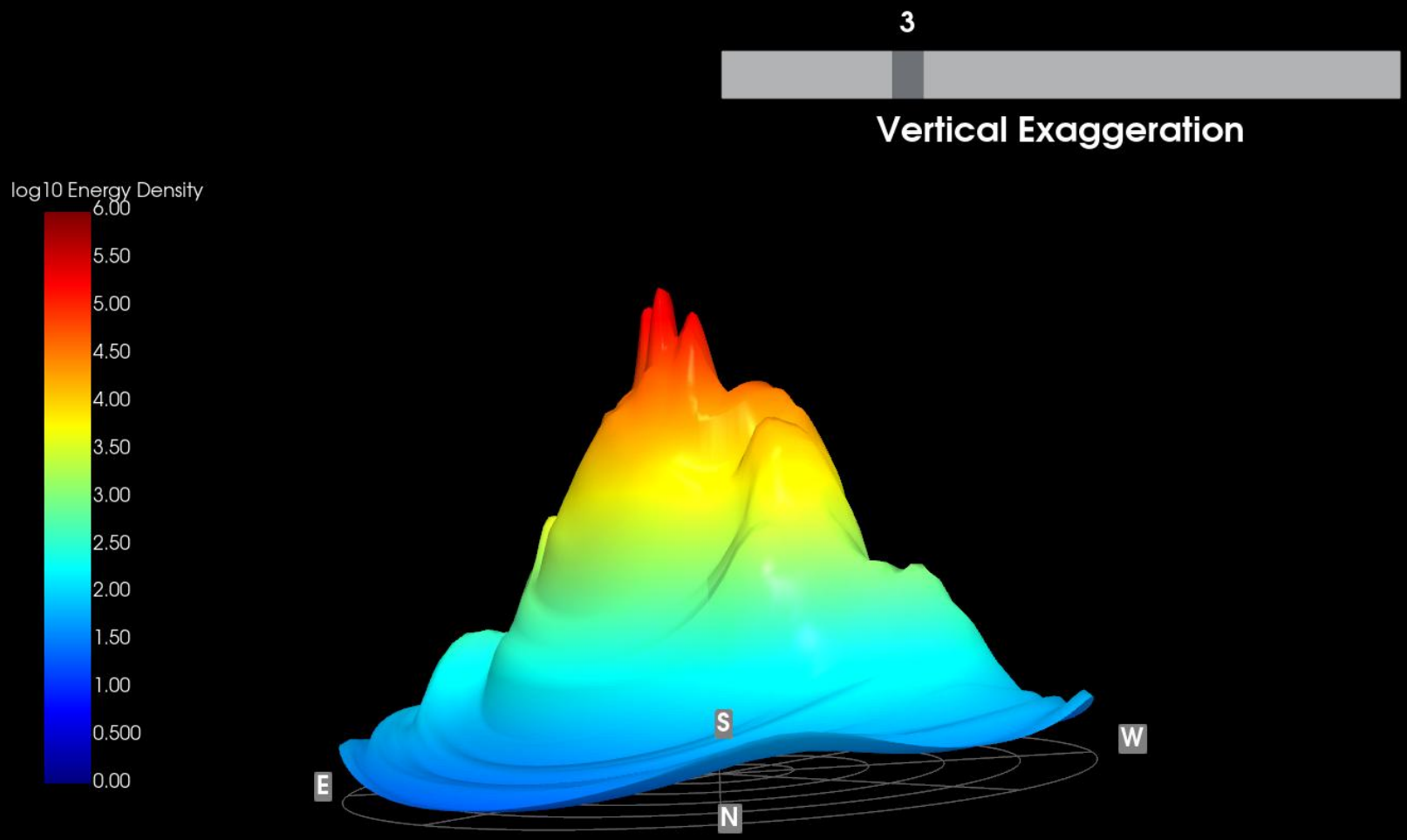
DMEM Spectrum





Representing the Data

DMEM Spectrum



3D

DMEM

Understanding the Data

DMEM Spectrum

Now that we understand energy spectrums (Czz10) and directional energy spectrums (DMEM), it is possible to turn the data they provide into a 3D representation of the sea state.

The Czz10 proved the idea that frequency bins can be converted into waves.

Now with DMEM, we can create those waves and send them in the directions they were travelling past the WaveRadar.

Understanding the Data

DMEM Spectrum

One DMEM Bin Becomes One Wave Component

Parameters

$$A = \frac{\sqrt{2S_i \Delta f \Delta\theta}}{100}$$

$$\omega_i = 2\pi f_i$$

$$k_i = \frac{\omega_i^2}{g}$$

$$\lambda_i = \frac{2\pi}{k_i} = \frac{g}{2\pi f_i^2}$$

Unity

What is it?

A real-time physics and rendering engine, typically used for games and interactive 3D applications.

What it does

- Takes data describing a 3D scene
- Uses programmed behaviour to update that scene
- Renders the result as an image in real time

What the engine needs

- Geometry: vertices and meshes describing objects
- Data: positions, wave heights, displacement, etc.
- Code: methods describing how the simulation behaves
- Materials and shaders: instructions for how objects are rendered

Unity

How do we make a simulation?

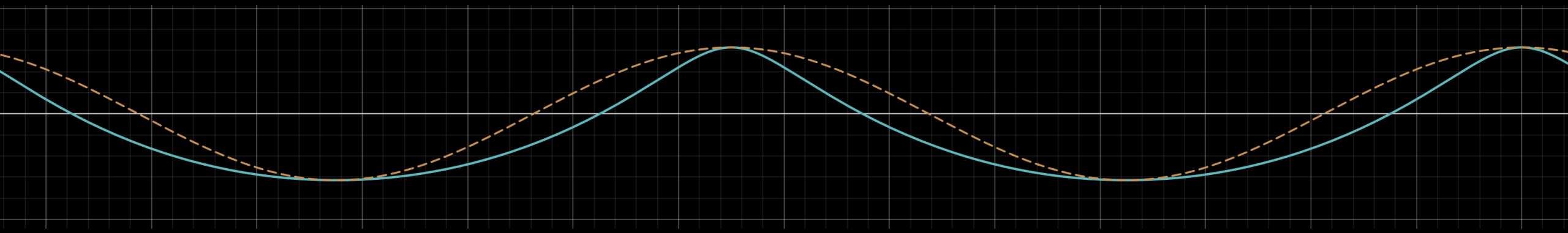
Limitations of sine waves

Simulation Accuracy

Water waves don’t just move up and down. A simple sine function does not model horizontal movement of water particles.

= sine wave

= Gerstner/Troichoidal wave

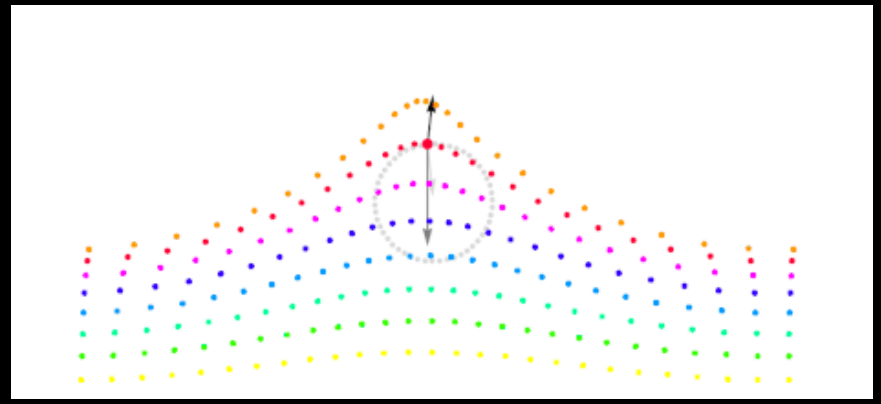


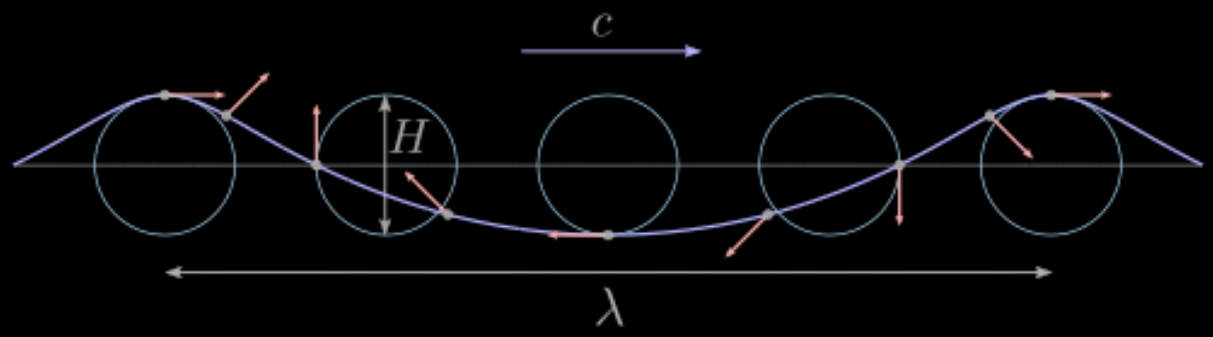
Gerstner/Troichoidal Waves

Simulation Accuracy

Real water particles in waves don’t just move up and down but rather follow a circular path.

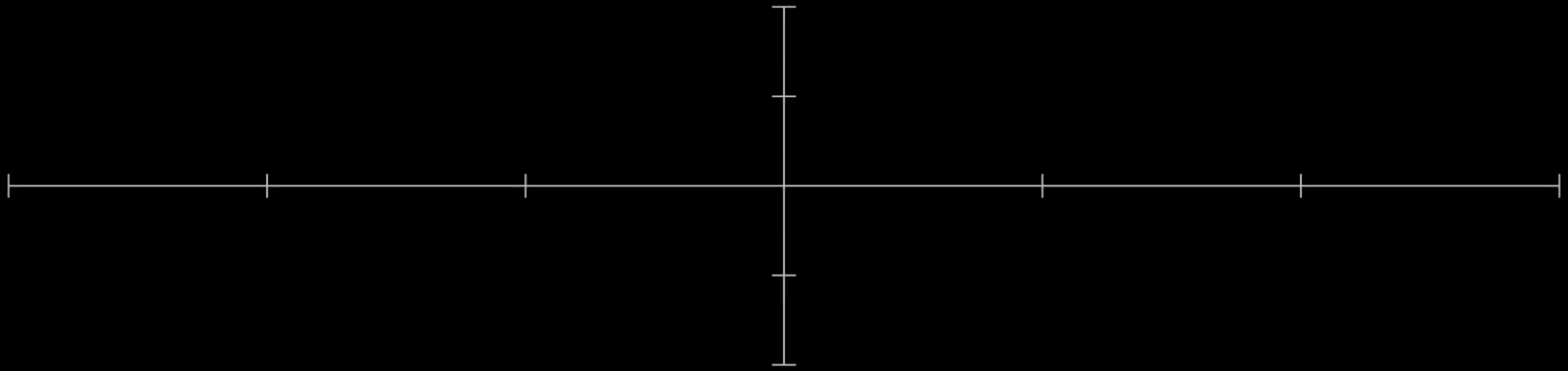
Gerstner waves make the ocean more realistic by introducing horizontal movement of particles on the wave surface.





Linear Sine Wave vs Gerstner Wave

Sine wave



Gerstner / Troichoidal Waves

Mathematics

Phase

$$\theta = \mathrm{k} \cdot \mathrm{x} - \omega \mathrm{t} + \phi$$

Vertical displacement

$$\eta = A \sin \theta$$

Horizontal displacement

$$D_{xy} = -Q\mathrm{A}\hat{\mathrm{k}}\cos\theta$$

where:

- A = wave amplitude
- Q = steepness/choppiness factor
- $\hat{\mathrm{k}}$ = wave direction vector
- θ = wave phase

The full surface position becomes:

$$\mathbf{x}' = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} -QAk_x \cos(\theta) \\ A \sin(\theta) \\ -QAk_z \cos(\theta) \end{bmatrix}$$

Recap

Simulation components

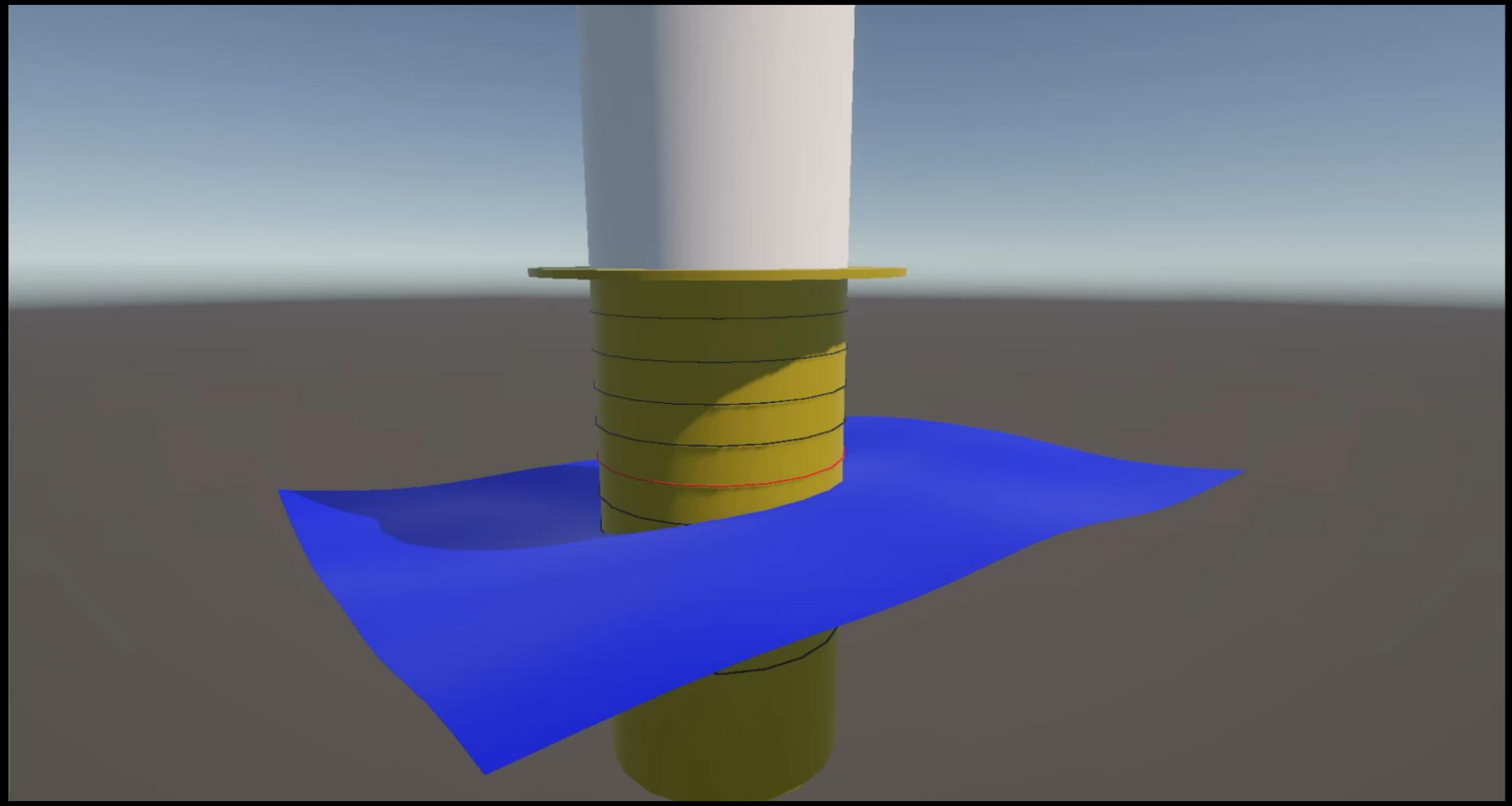
✓ Mesh made of vertices

✓ Vertical displacement

✓ Horizontal displacement using Gerstner displacement

Basic Version

CPU – Strongest wave components



Turbine:

7m diameter

Platform:

5m above sea level

Red line:

Sea level

Black lines:

Spaced 1m apart

Performance

Currently low

This version has poor performance.

This is because for each coordinate, the code calculates the height of each coordinate by sequentially adding sine components:

- $y = \sin(\theta_1) + \sin(\theta_2) + \sin(\theta_3) + \ldots$
- $y = \sin(\theta_1) + \sin(\theta_2) + \sin(\theta_3) + \ldots$
- $y = \sin(\theta_1) + \sin(\theta_2) + \sin(\theta_3) + \ldots$
- $y = \sin(\theta_1) + \sin(\theta_2) + \sin(\theta_3) + \ldots$
- $y = \sin(\theta_1) + \sin(\theta_2) + \sin(\theta_3) + \ldots$
- …

Performance

Current Issues

The two ways to improve the performance is by:

- Improving the maths to reduce the volume and effort required for each position in the wave field
    - We can do this using the Fourier Transform
- Moving the calculations to a more parallel processor to calculate more positions at once
    - The GPU is designed specifically for tasks like this

Improving the maths

Fourier Transform

Discrete Fourier Transform

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi \frac{k}{N} n}$$

Taking spatial domain information and converting it to frequency domain information.

Inverse Fourier Transform

$$x_n = \frac{1}{N} \sum_{\mathrm{k}=0}^{N-1} X_k \cdot e^{i2\pi \frac{k}{N} n}$$

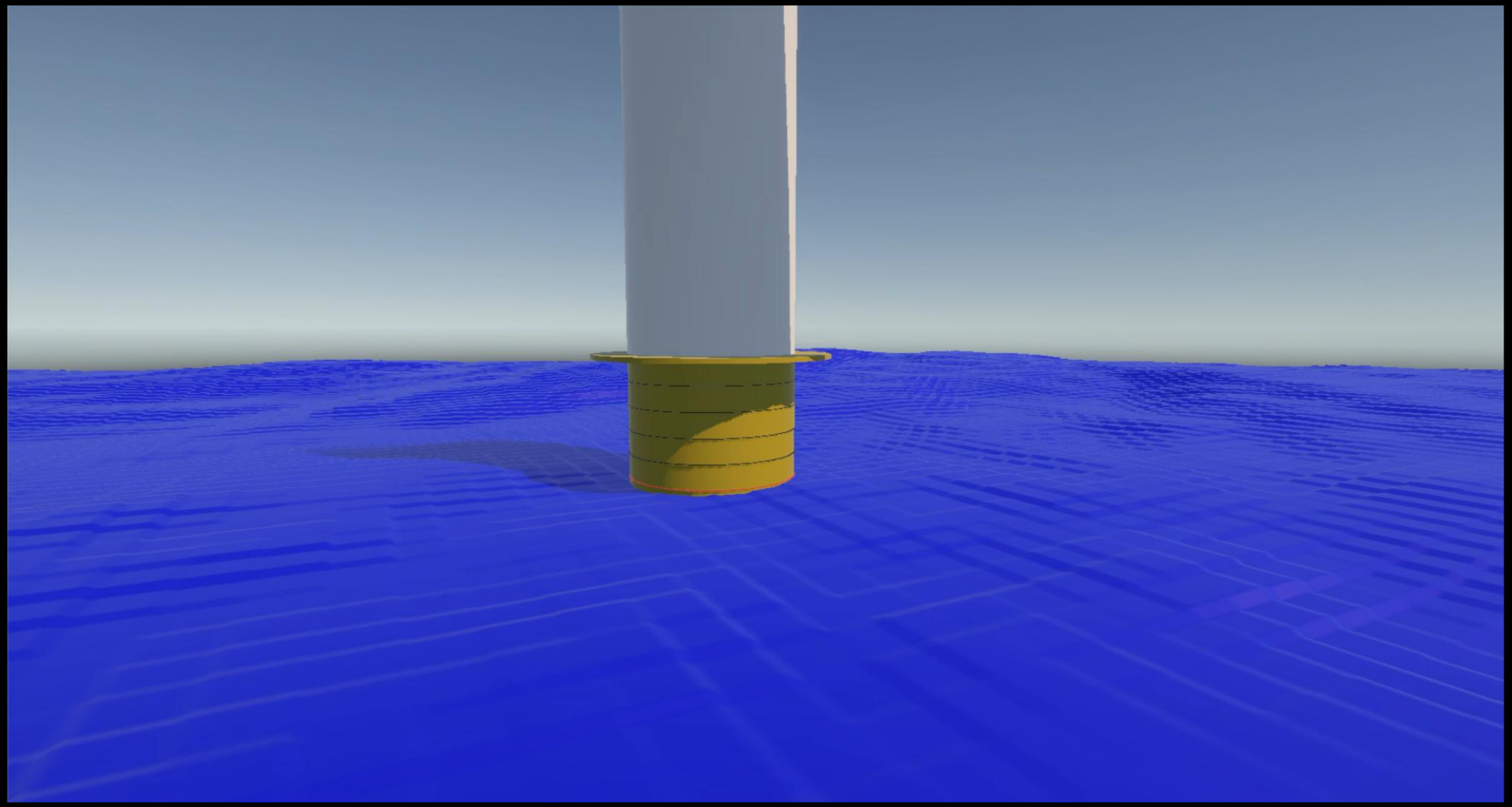
Taking frequency domain information and converting it to spatial domain information.

Improving the maths

Inverse Fourier Transform

Improved ocean

CPU IFFT



Turbine:

7m diameter

Platform:

5m above sea level

Red line:

Sea level

Black lines:

Spaced 1m apart

Performance

Still not enough…

The simulation does the same thing to each point on the mesh.

Tasks like this are known as “embarrassingly parallel” tasks.

This means the GPU is well suited for these operations. Moving these calculations to the GPU massively improves performance.

The benefit of this is that we can now simulate more waves gathered from the spectrum, across a large grid, providing more detail.

So why is the GPU the better choice?

Performance

Still not enough.

Computers use two main types of processors:

- The CPU (Central Processing Unit)
- The GPU (Graphics Processing Unit)

Performance

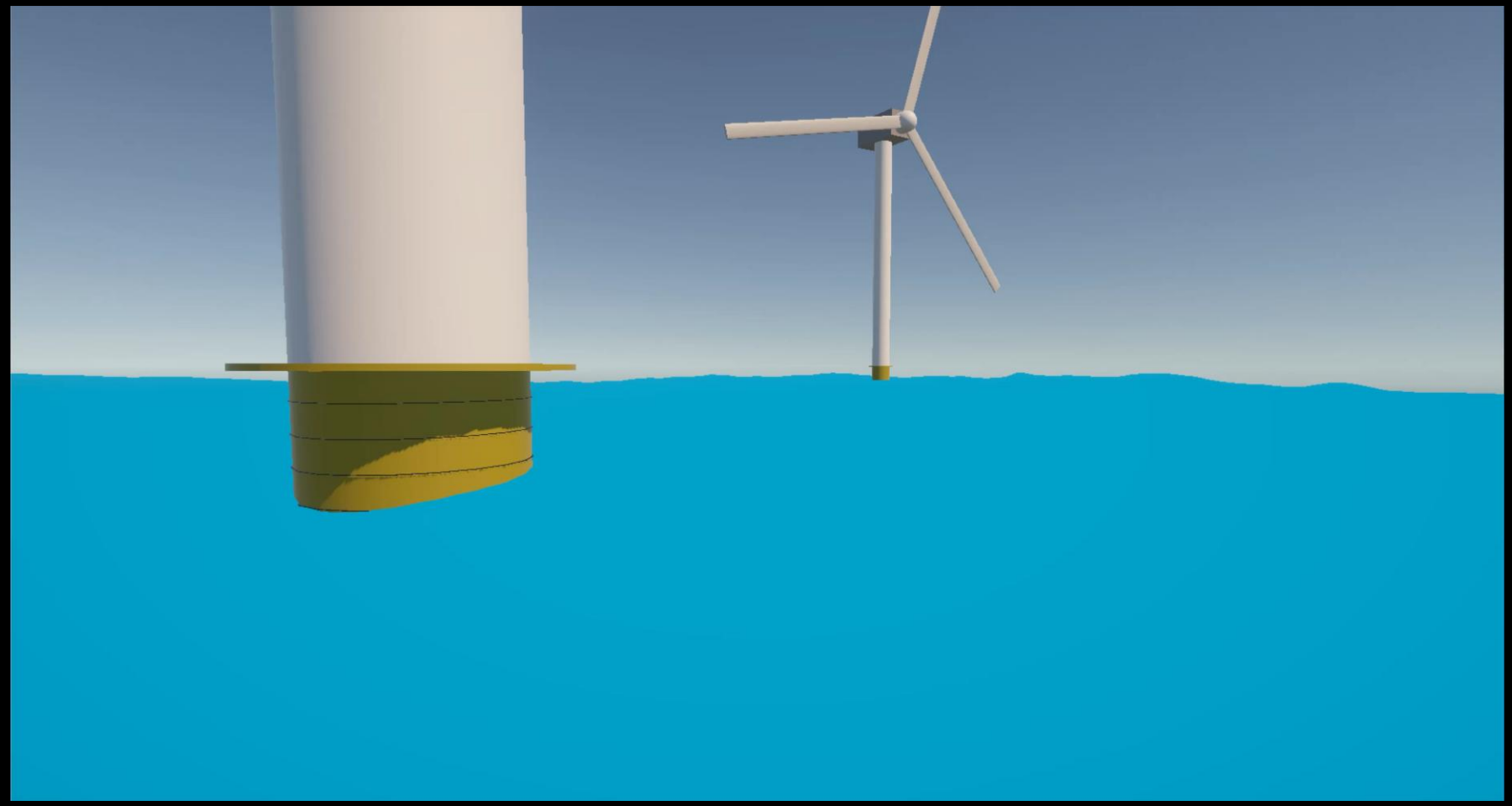
Architecture Choices

Performance

How the GPU is working

High speed rendering

GPU IFFT



Displacement calculated on the GPU and mesh being updated

This is working

But it doesn’t look right?

The simulation is missing some final processing to make it look correct.

It still needs:

- Lighting
- Reflections
- Foam

Lighting using normals

Making the surface show the correct colour

For Unity to know what to show on screen, it needs to know what colour each pixel on the screen needs to be.

And to work out what colour a pixel should be, it must know what the scene looks like.

We use the light in the scene and ‘normals’ to work out how much light is being bounced towards at the camera.

Lighting

Why normals?

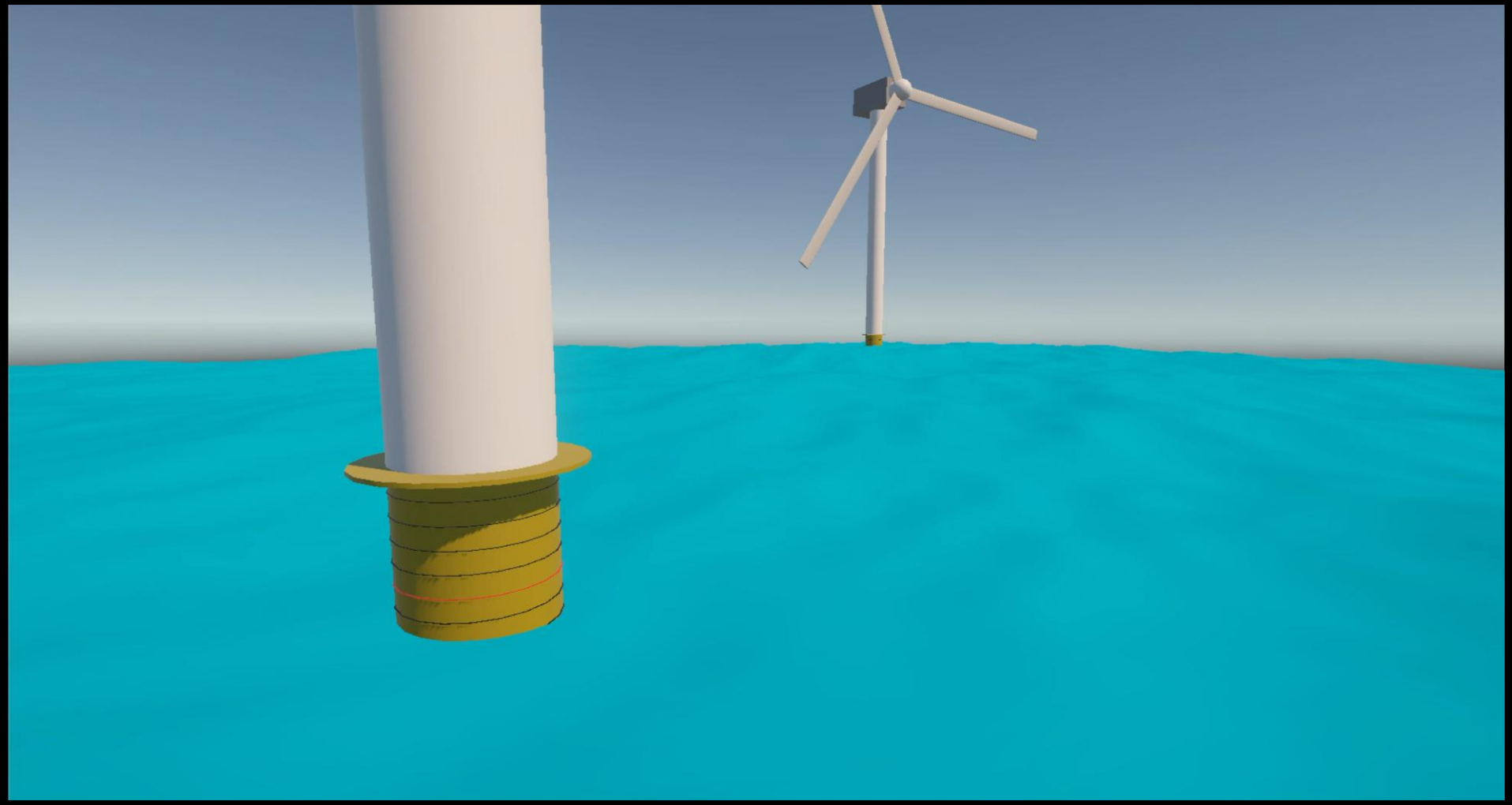
Lighting

How do we render the simulation?

Surface vertices

Lit Version

GPU IFFT and normals



Turbine:

7m diameter

Platform:

5m above sea level

Red line:

Sea level

Black lines:

Spaced 1m apart

Reflections

Visual fidelity improvements

A real ocean surface isn’t opaque and completely smooth.

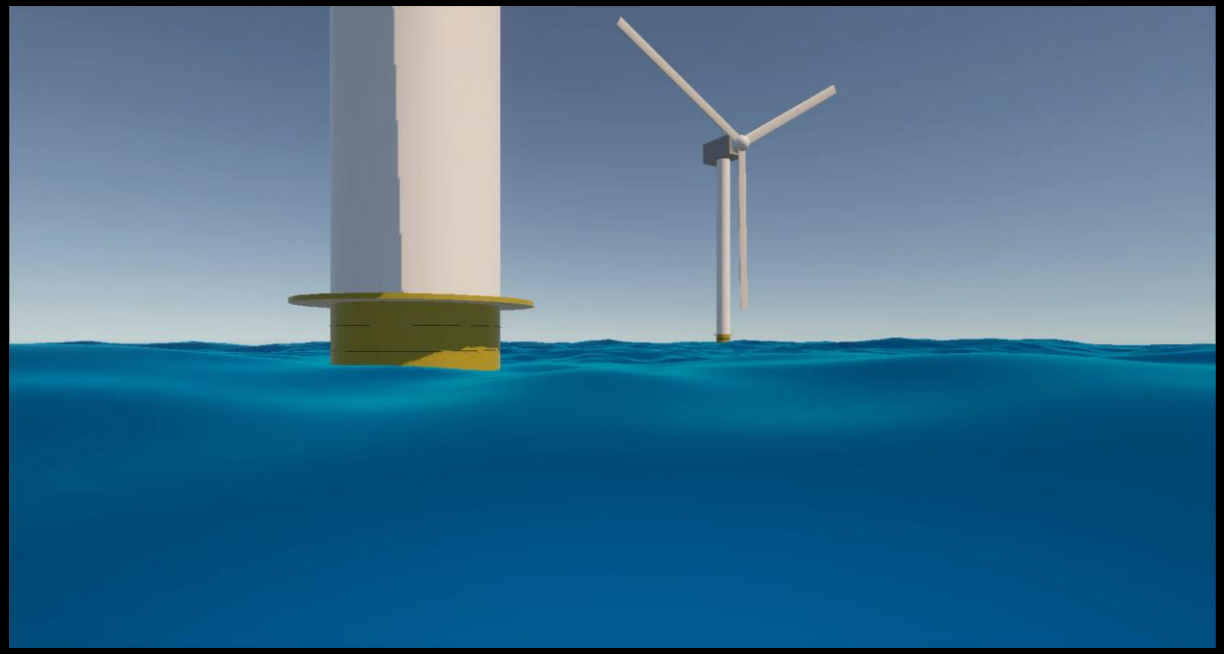
Light reflects at angles off the surface.

Reflections

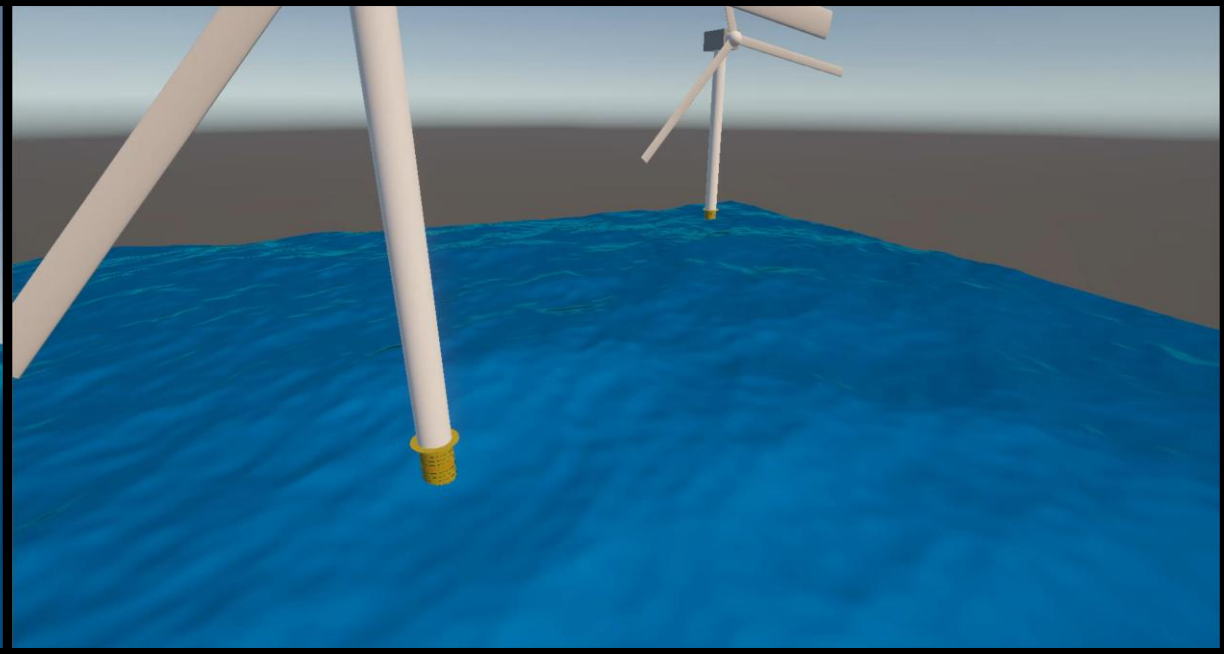
Reflecting light and Frensel effect

Visual improvements

GPU IFFT with normals and Frensel lighting



Lower viewing angle shows greater reflections due to the Frensel effect



Higher viewing angles show less reflection

Foam/White caps

Clearer ocean conditions

In choppy conditions, waves break. This causes air to get trapped under the breaking crest of the wave and foam forms on the surface of the water.

Seeing where foam is forming visually demonstrates the conditions of the sea more than simply showing displacement.

Foam

Where does it go?

Foam

Jacobian

$$J = \begin{bmatrix} 1 + \frac{\partial D_x}{\partial x} & \frac{\partial D_x}{\partial z} \\ \frac{\partial D_z}{\partial x} & 1 + \frac{\partial D_z}{\partial z} \end{bmatrix}$$

Describes how the horizontal displacement field changes in space.

$$\det(J) = \left(1 + \frac{\partial D_x}{\partial x}\right)\left(1 + \frac{\partial D_z}{\partial z}\right) - \left(\frac{\partial D_x}{\partial z}\right)\left(\frac{\partial D_z}{\partial x}\right)$$

Represents the change in local surface area

Foam

Jacobian

$$J = \begin{bmatrix} 1 + \frac{\partial D_x}{\partial x} & \frac{\partial D_x}{\partial z} \\ \frac{\partial D_z}{\partial x} & 1 + \frac{\partial D_z}{\partial z} \end{bmatrix}$$

$$\det(J) = \left(1 + \frac{\partial D_x}{\partial x}\right)\left(1 + \frac{\partial D_z}{\partial z}\right) - \left(\frac{\partial D_x}{\partial z}\right)\left(\frac{\partial D_z}{\partial x}\right)$$

How is this used?

- $\det(J) \approx 1$ → No compression

  No foam

- $\det(J) < 1$ → Compression

  The points are being pushed together. This happens at wave crests where water is bunching up

- $\det(J) \to 0$ → Extreme case

  The surface is approaching a fold. Physically, this represents a breaking wave.

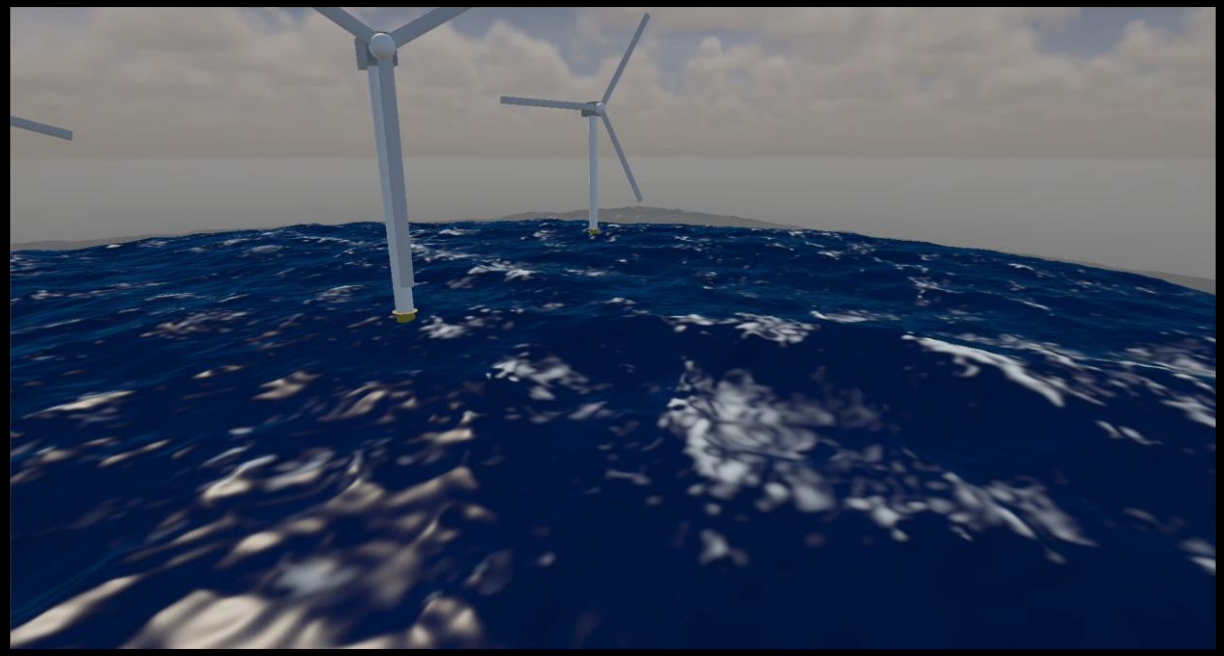
Foam

Jacobian

Convincing ocean

Foam

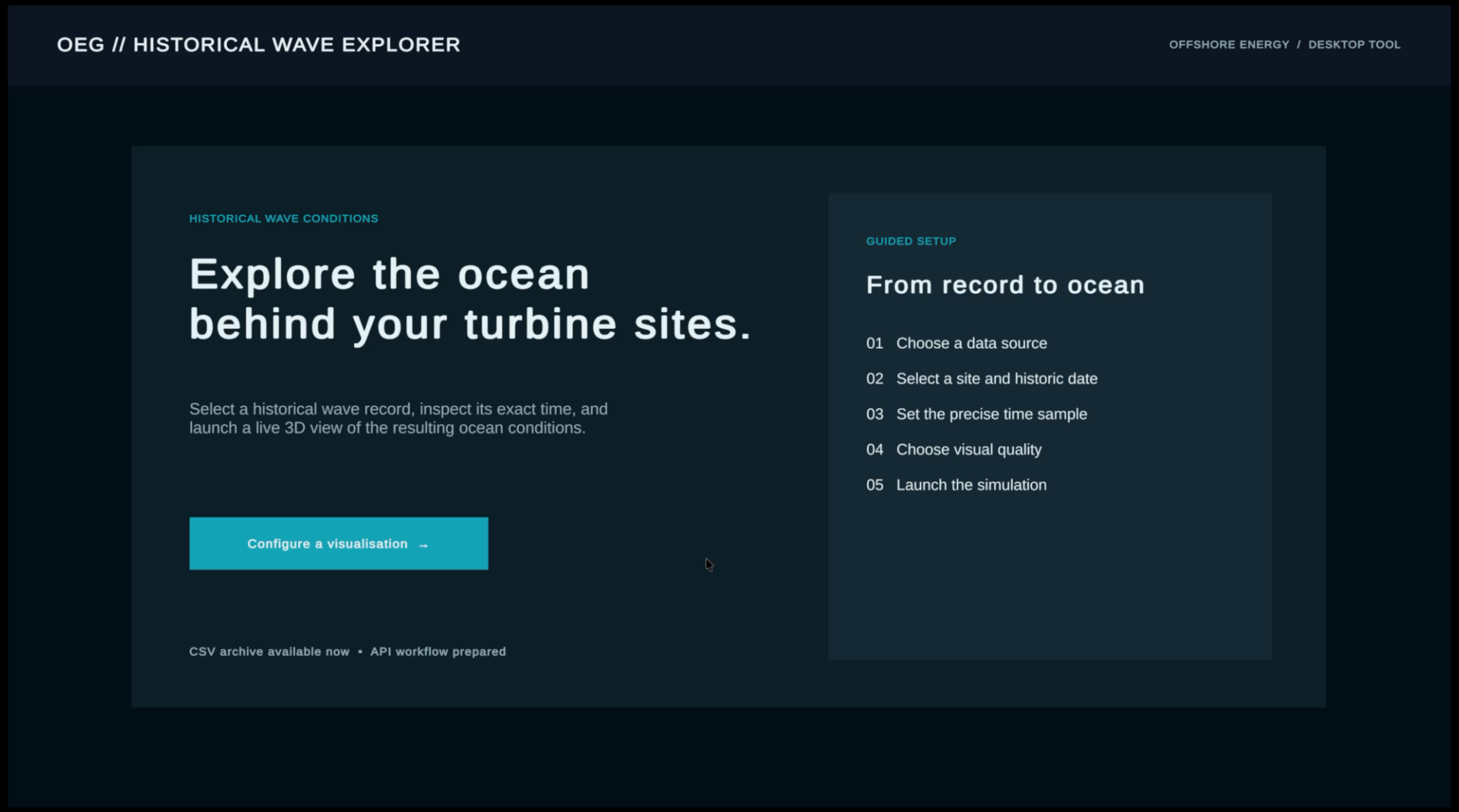
Displacement calculated on the GPU but mesh not updated

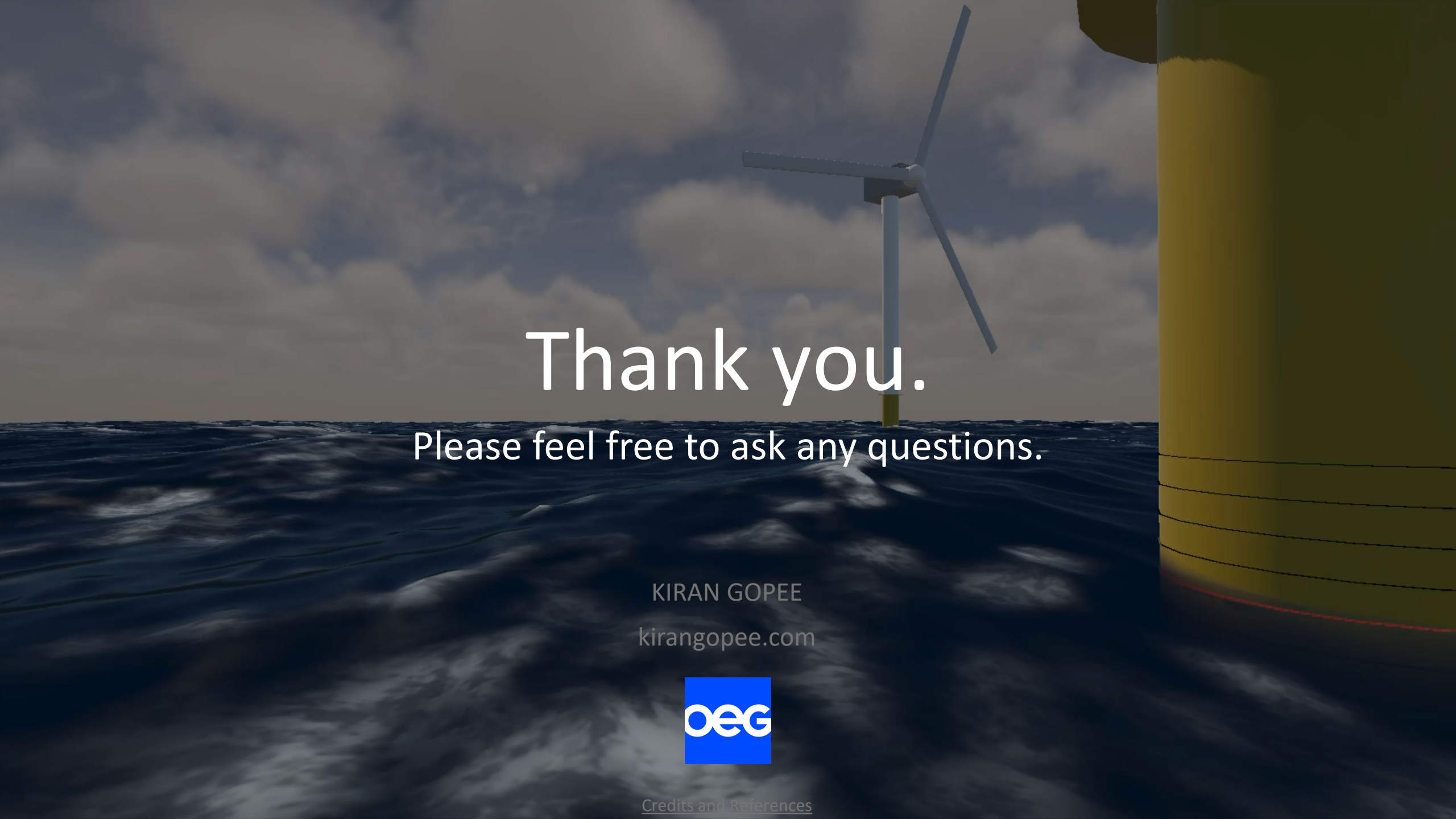


Displacement calculated on the GPU and mesh being updated

Final product

Completed simulation





Thank you.

Please feel free to ask any questions.

KIRAN GOPEE

kirangopee.com

Credits and References

Questions

Links

Definitions

Hm0 and Tp

Why triangles?

End

Wave Equation

Fast FT

Race Conditions

Superposition

IDFT vs IFFT

Wave Equation

← Back

$$y(x,t) = A\ \sin(kx\ -\ \omega t + \phi)$$

$$y(x,t) = A \sin(kx - \omega t + \phi)$$

y – height at a point on the wave

A – amplitude of the wave

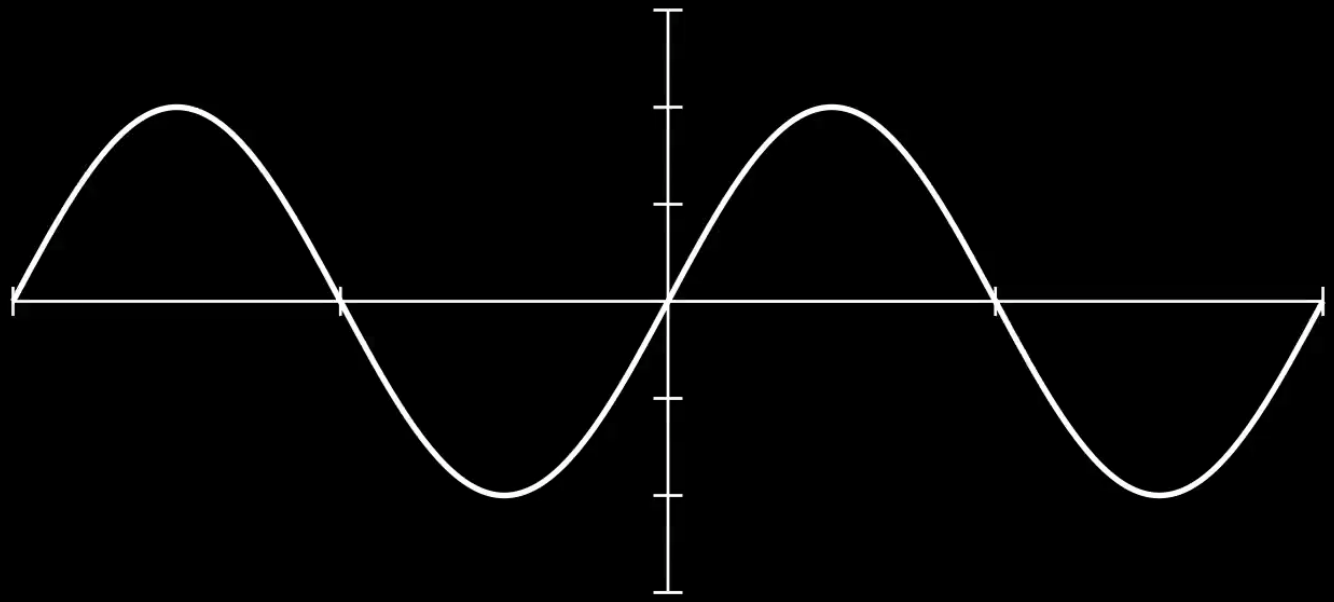
k – wave number

x – x-coordinate

ω – angular frequency

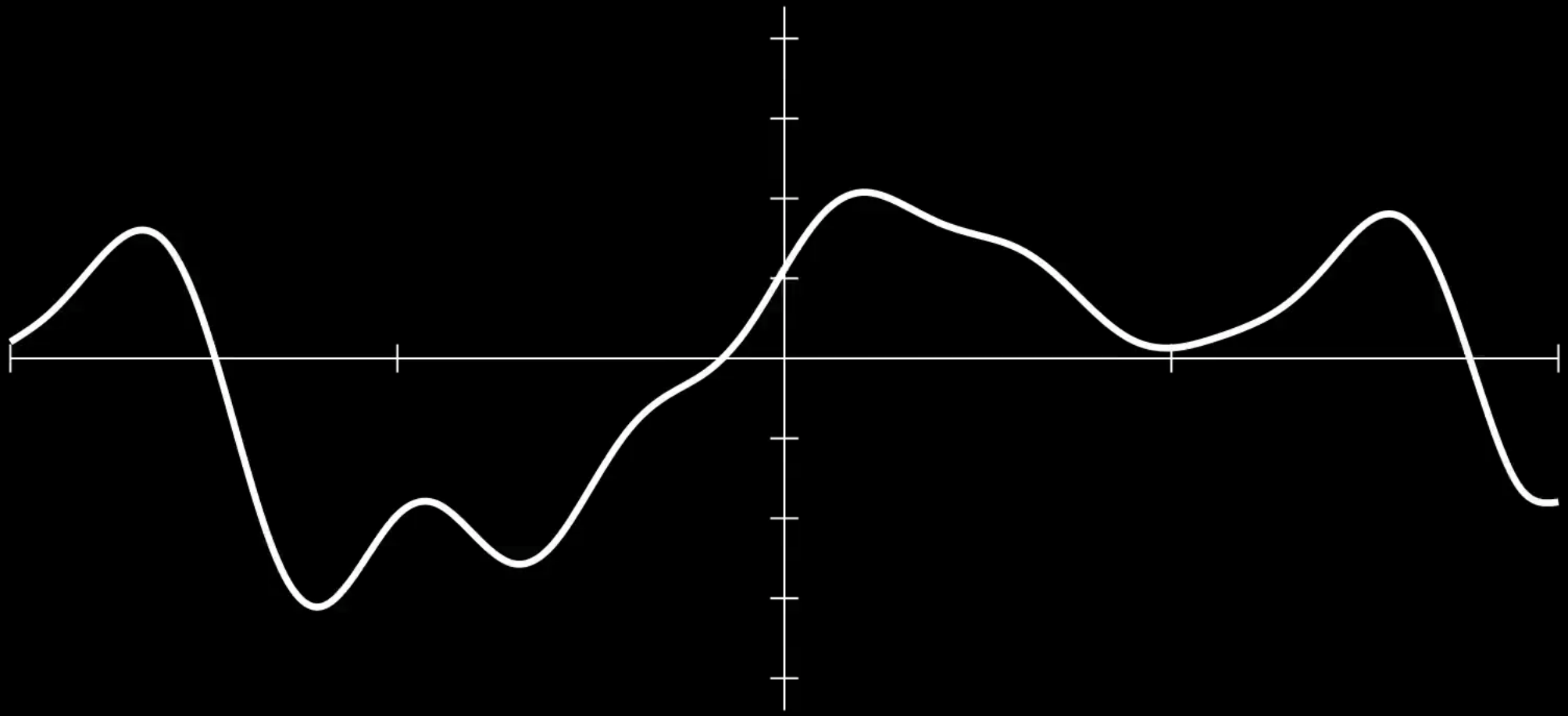
t – time

ϕ – the phase



Superposition

← Back



Definitions

← Back

Amplitude

Vertical displacement

Wavelength

Peak to peak distance

Frequency

How many waves arrive per second

Phase

How a wave

Period

Time between two peaks crossing a point

Angular Frequency

How quickly a wave completes one cycle

Wave Number

How rapidly the wave changes with distance

Hm0 and Tp

← Back

$$\mathrm{Hm0} = 4\sqrt{\mathrm{M0}}$$

$$\mathrm{Tp} = \frac{1}{\mathrm{Fp}}$$

Fast FT

← Back

The Inverse Discrete Fourier Transform can reconstruct the spatial surface directly, but it is computationally expensive because each output point depends on every frequency component.

The Inverse Fast Fourier Transform produces the same result but reorganises the calculation to avoid repeating work.

$$\text{Inverse DFT: } O(N^2) \rightarrow O(2048^2) = 4{,}194{,}304$$

$$\text{IFFT: } O(N \log_2 N) \rightarrow O(2048 \log_2 2048) = (2048 \times 11) = 22{,}528$$

$$\frac{4{,}194{,}304}{22{,}528} = 186.18\ldots \text{ times less complexity}$$

IDFT vs IFFT

← Back

Why triangles?

Race conditions

GPU considerations

Credits and References

Resources

Animation software

Manim Community Edition Python library

WaveRadar Visualisation

TEMPLATES

TEMPLATE

Subtitle

-D Visualisation

DATA

Issues

- 1

TITLE