# Optimizing 3D Object Fabrication For Steadiness

Wenqing QU, Zhaoyang CHEN

**Abstract**

This is the course project for INF585-Computer Animation, in which we manage to implement the method presented in the paper *A Simple Model of Ocean Waves*. The project is implemented in Python.

## 1 Problem Statement and Motivation

Ocean waves are a complex natural phenomenon that has fascinated scientists, artists, and engineers for centuries. In this project, we implement a simple model of ocean waves based on the paper *A Simple Model of Ocean Waves* by Alain Fournier.

The paper presents a simple model for simulating ocean waves that is based on the Gerstner or Rankine model. The model is essentially a parametric surface with two spatial parameters and one time parameter, which allows for traditional rendering methods such as ray-tracing and adaptive subdivision. The model can produce a large range of wave shapes and phenomena, including the effects of depth such as refraction and surf, as well as the effects of the ocean floor on wave refraction and breaking on the shore. The foam generated by the breakers is modeled by particle systems whose direction, speed, and life expectancy are given by the surface model. The model can also determine the position, direction, and speed of breakers.

Overall, this simple yet effective model can be used for modeling and rendering most common waves caused by wind and gravity while taking into account various factors that affect wave behavior in real-life oceans. The objective of the project is to simulate and visualize the motion of ocean waves in a realistic and computationally efficient manner using Python programming language and Matplotlib library.

## 2 Implementation

Fluid motion is generally studied by two methods. In the Eulerian method, we consider a point $(x, y, z)$ and try to calculate the properties of the fluid at this point as a function of time, such as the speed:

$$U = f(x, y, z, t)$$

In the Lagrangian method, we follow the trajectory of a point $(x_0, y_0, z_0)$, given by a reference position. This can be seen as the trajectory of a particle of matter. For instance, we can calculate the speed at time t:

$$V \;\; = f_x(x_0, y_0, z_0, t)$$

$$Vy = f_y(x_0, y_0, z_0, t)$$

$$Vz = f_z(x_0, y_0, z_0, t)$$

The first method is favored in the study of waves, especially since the development of stochastic models for the analysis of the sea, The second method is more useful for graphics modeling. Therefore, it will be more convenient for us to treat the ocean surface as a graphical primitive and apply the Lagrangian approach.

We will consider that each particle on the ocean surface describes a circle around its rest position.We will also orient our world coordinates so that the XY plane is the plane of the sea at rest, and the Z axis is pointing up. For the time being, we will only consider the motion in the XZ plane. Then the equation of the motion of a particle is:

$$x = x_0 + r * sin(\kappa x_0 - \omega t)$$

$$z = z_0 - r * cos(\kappa x_0 - \omega t)$$

The equations above treat the surface as a trochoid, a generalization of a cycloid. It can be seen as the curve generated by a point P at distance r from the center of a circle of radius – rolling over a line at distance 1 under the X axis (Figure 3). The parametric equations are, for t=0 and z 0:

$$x = -\frac{\alpha}{\kappa}$$

$$z = -r * cos(\alpha) where \alpha = -\kappa x_0$$

So this is our basic model. To introduce the needed effects and variations, we will manipulate the parameters of the orbit equations (modifying the radius, the phase angle, turning the circle into an ellipse, etc.).

We define a set of variables related to the characteristics of the wave, such as its frequency, amplitude, wavelength, and steepness, as well as the dimensions and resolution of the domain. We then calculate the height of the waves at each point of the domain using the wave function, which takes into account the depth of the water, the wind speed, and the time.

The animation is generated using the matplotlib.animation module, which creates a sequence of frames that are displayed in succession, creating the illusion of motion. The update function is called at each frame, which recalculates the position of the waves at that time and updates the plot accordingly. The resulting animation shows a realistic simulation of ocean waves and is stored in a gif file.

# 3 Results

We present the results of our simulation in several figures. The figures show the states of the ocean surface at different points on the grid over time. We observe that the motion of waves is realistic and visually appealing, which demonstrates the effectiveness of the proposed algorithm.

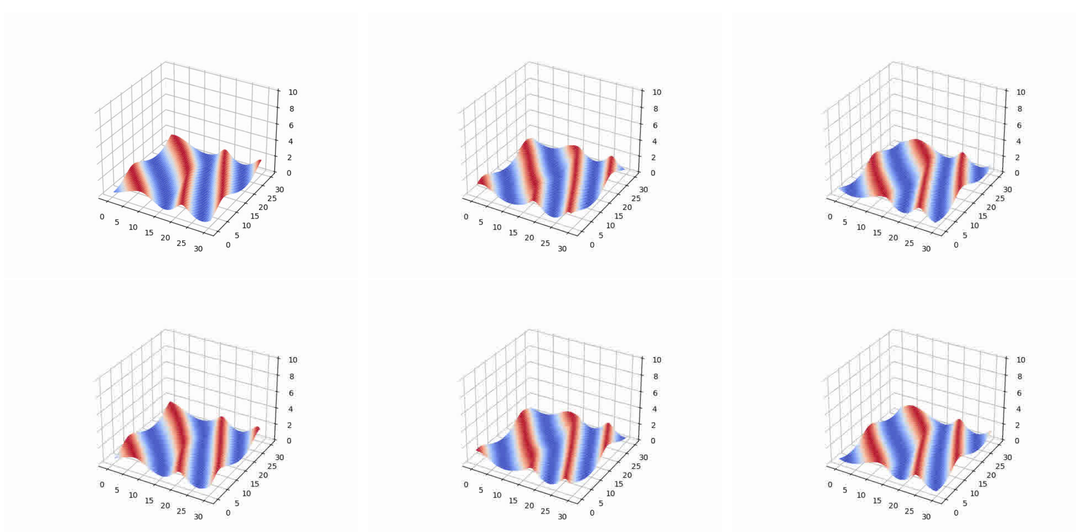

Figure 1: Implementation Results

## 4 Conclusion

In this project, we have implemented and visualized a simple model of ocean waves based on the paper *A Simple Model of Ocean Waves* by Alain Fournier. We have demonstrated the effectiveness of the proposed algorithm in generating realistic and visually appealing motion of waves. The implementation is done in Python programming language, using Numpy and Matplotlib libraries, which are commonly used tools in the scientific computing and data visualization community.