Josh Arrington, Jun Eshima, Zehua Liu

October 7, 2022

APC524 Final Project Outline

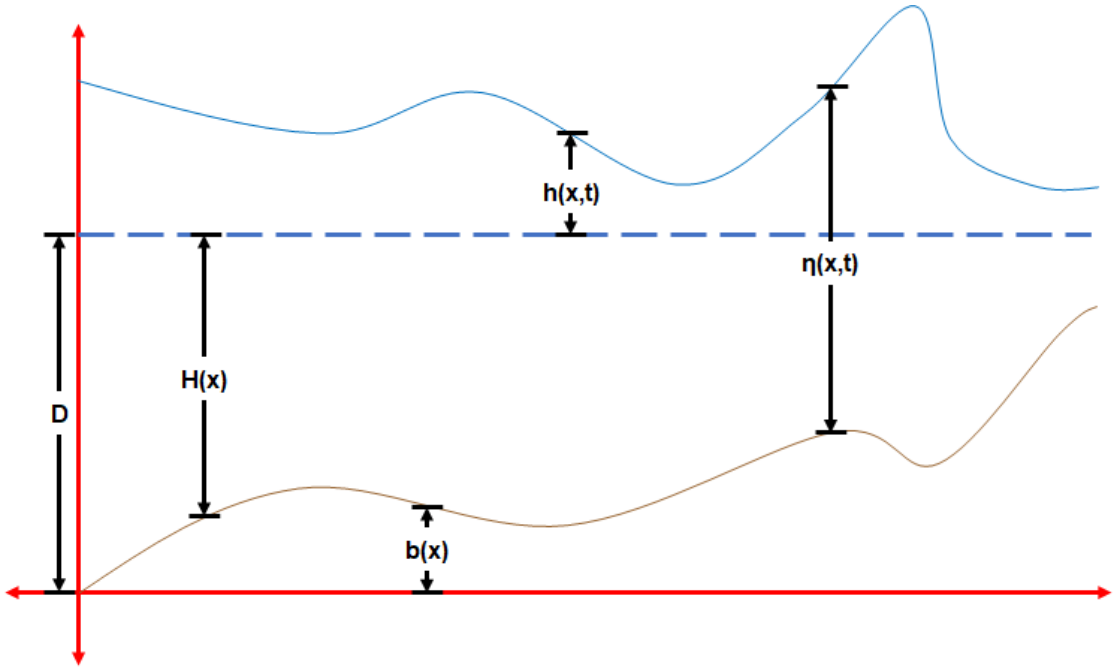

Figure 1: Set up of shallow water wave equations (taken from Wikipedia: `https://en.wikipedia.org/wiki/Shallow_water_equations#/media/File:Shallow_water_model_diagram.png`)

Summary

The set up will consist of a 1D/2D ocean-atmosphere interface. We will most likely start with 1D and see if we have enough time/computational power to do the 2D case. In order to make the equations computationally tractable, we will use the shallow water wave equations - from Wikipedia (2D equation):

$$\frac{\partial h}{\partial t} + \frac{\partial (H+h)u}{\partial x} + \frac{\partial (H+h)v}{\partial y} = 0 \tag{1}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -g\frac{\partial h}{\partial x} \tag{2}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -g\frac{\partial h}{\partial y} \tag{3}$$

Where u, v are the horizontal velocities, H is the height below sea level and h is the height above sea level.

Version control

Since this is a programming coursework, the base set up is deliberately simple. But in order to line up with the aim of working with git, version control etc, we propose that the main purpose of this program will be the ability to add in different aspects of physics.

For example, we could have the following modules to be added using version control:

- Gravity
- Wind
- Surface tension
- Different types of ocean floor topography
- Diffusion of contaminants/sediments in the water
- Wave-breaking structures

Testing

For testing, we will compare analytical solutions to numerical solutions. We will be using Pytest to test functions.

Implementation

In order to line up with the content learnt in APC524, we will be using Python. For the numerical analysis and solving, we will be using Numpy and its linear solvers/data arrays. Scipy also contains several numerical integrators which will be useful. Much of our plotting and visualization will be done using Matplotlib.

When the mesh grid becomes very refined, the computational load may become heavy, and so there may be room to implement parallel computing to accelerate the computation process. If we find the need to perform computations in parallel, the "multiproccessing" library will be helpful.