The Shallow Water Equations: Derivation and Numerical Simulation

Advanced Fluid Dynamics Course Project

Damyn Chipman

May 5th, 2021

Outline

Introduction

The Shallow Water Equations (SWEs) model the free surface flow of a shallow fluid. The term "shallow" refers to an explicit assumption made in the formulation of the model that assumes the horizontal length scale is much larger than the veritcal height of the fluid. The SWEs can be derived from the incompressible Navier-Stokes equations by making the aforementioned assumption and depth integrating over the vertical momentum equation. The SWEs have been used to model tsunami wave development as well as debris flow on land.

The SWEs form a hyperbolic system of partial differential equations (PDEs). They can also be expressed in a conservative form. As such, it is common to use a finite volume method to simulate the dynamics of the system. Finite volume methods can accurately model conservative, hyperbolic systems on complicated geometries.

In this project, we will study the shallow water equations by deriving them from the incompressible Navier-Stokes equations, and then simulate the system with a finite volume method on an unstructured mesh. We then present "flud dynamics in a box" simulations to verify both the numerical method and conservative nature of the SWEs.

From Incompressible Navier-Stokes to the Shallow Water Model

Throughout this course, we have explored various solutions to the Navier-Stokes equations by making various assumptions and then solving the resulting (simplified) equations. Following this procedure, to derive the shallow water equations, we will start with the incompressible Navier-Stokes, given below in their vector form

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = -\frac{1}{\rho}\nabla P + \nu\nabla^2\mathbf{u} + \mathbf{g}$$
$$\frac{\partial \rho}{\partial t} + \rho\nabla\mathbf{u} = 0$$

First, we assume that any viscous contributions are either negligible or we will incorporate them later as a source term. So for right now, $\nu = 0$.

Conservation of Momentum

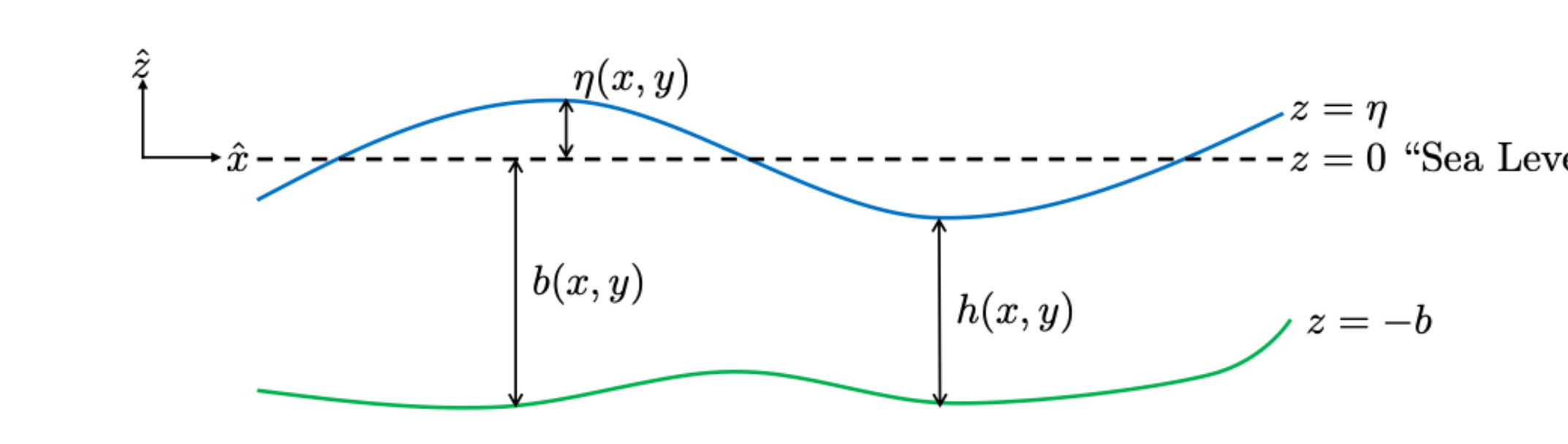
We start by looking at just the momentum equation. Before we look at the other assumptions, let's write out the full momentum equation into each component. We let $\mathbf{u} = (u, v, w)$ and $\mathbf{g} = (g_x, g_y, g_z) = (0, 0, g)$.

$$\text{x: } \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial x}$$
$$\text{y: } \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial y}$$
$$\text{z: } \frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial z} + g$$

Now, the next assumption we make is that of shallow water. This means we assume that $H/L << 1$, where H is a characteristic fluid depth and L is a characteristic horizontal scale. Another way of viewing this assumption is by saying that w is constant or zero. This reduces the z-component of momentum to just the hydrostatic equation

$$\frac{\partial P}{\partial z} = \rho g$$

We integrate from the bottom of the fluid column to the top, or from $z = -b$ to $z = h$, as shown in the figure below



The setup for this system is as follows. The horizontal variables are x and y (with their associated velocities) and the veritcal variable is z (with its associated velocity). The vertical distances are set from a reference of $x = 0$, which refers to sea level. Any height above sea level is given as η, and the bathymetry b is the topology of the sea floor. h is the height of the water column from the surface of the sea floor.

Integrating from the bottom of the fluid column to the top yields:

$$\begin{aligned} P = \int dP &= \int_{-b}^{h} \rho g dz \\ &= \rho g z\Big|_{-b}^{h} \\ &= \rho g h + \rho g b \\ P(x, y, z, t) &= \rho g h(x, y, t) + \rho g b(x, y) \end{aligned}$$

As the pressure gradient term appears in the x- and y-components of the momentum equation, we can differentiate

$$\frac{\partial P}{\partial x} = \rho g\frac{\partial h}{\partial x} + \rho g\frac{\partial b}{\partial x}$$
$$\frac{\partial P}{\partial y} = \rho g\frac{\partial h}{\partial y} + \rho g\frac{\partial b}{\partial y}$$

and substitute our expression for pressure into the x- and y-components of momentum:

$$\text{x: } \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -\left(g\frac{\partial h}{\partial x} + g\frac{\partial b}{\partial x}\right)$$
$$\text{y: } \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -\left(g\frac{\partial h}{\partial y} + g\frac{\partial b}{\partial y}\right).$$

The next assumption we make is that the horizontal velocities u and v are independent of vertical depth, i.e., $\frac{\partial u}{\partial z} = \frac{\partial v}{\partial z} = 0$. We eliminate these terms and simplify a little to get the momentum equations for the shallow water equations:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(u^2 + \frac{1}{2}gh\right) + \frac{\partial}{\partial y}\left(uv\right) = -g\frac{\partial b}{\partial x}$$
$$\frac{\partial v}{\partial t} + \frac{\partial}{\partial x}\left(uv\right) + \frac{\partial}{\partial y}\left(v^2 + \frac{1}{2}gh\right) = -g\frac{\partial b}{\partial y}$$

Conservation of Mass

Next, we look at how the conservation of mass changes under these assumptions. Conservation of mass states:

$$\frac{dm}{dt} = F_{net}.$$

The net flux into our out of a fluid element through advection is given as

$$\begin{aligned} F_{net} &= -\int_S \rho\mathbf{u}\cdot d\mathbf{S} \\ &= -\int_l \rho h\mathbf{u}\cdot\hat{n}dl \\ &= -\int_A \nabla\cdot(\rho h\mathbf{u})dA \end{aligned}$$

where $\mathbf{S}$ is the surface area of the fluid element, l is the perimeter of the fluid element, and A is the cross-sectional area of the fluid element.

The total mass inside a fluid element is given as $dm = \rho dV = \rho h dA \Rightarrow m = \int \rho h dA$. Plugging these expressions into the conservation of mass yields:

$$\frac{d}{dt}\int_A \rho h dA = -\int_A \nabla\cdot(\rho h\mathbf{u})dA$$
$$\Rightarrow \int_A \left(\frac{\partial h}{\partial t} + \nabla\cdot(h\mathbf{u})\right)dA = 0$$
$$\Rightarrow \frac{\partial h}{\partial t} + \nabla\cdot(h\mathbf{u}) = 0$$
$$\Rightarrow \frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) + \frac{\partial}{\partial y}(hv) = 0$$

Shallow Water Equations (Non-Conservative Form)

Summarizing the above gives the shallow water equations in their non-conservative form:

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) + \frac{\partial}{\partial y}(hv) = 0$$
$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(u^2 + \frac{1}{2}gh\right) + \frac{\partial}{\partial y}\left(uv\right) = -g\frac{\partial b}{\partial x}$$
$$\frac{\partial v}{\partial t} + \frac{\partial}{\partial x}\left(uv\right) + \frac{\partial}{\partial y}\left(v^2 + \frac{1}{2}gh\right) = -g\frac{\partial b}{\partial y}$$

Shallow Water Equations (Conservative Form)

We can express the shallow water equations in terms of conservative variables (h, hu, hv) by multiplying the momentum equation by h:

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) + \frac{\partial}{\partial y}(hv) = 0$$
$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) + \frac{\partial}{\partial y}\left(huv\right) = -gh\frac{\partial b}{\partial x}$$
$$\frac{\partial v}{\partial t} + \frac{\partial}{\partial x}\left(huv\right) + \frac{\partial}{\partial y}\left(hv^2 + \frac{1}{2}gh^2\right) = -gh\frac{\partial b}{\partial y}.$$

If we let $\mathbf{q}$ be our conservative variable, we can write the shallow water equations in the classic conservation form:

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla\cdot\mathbf{F}(\mathbf{q}) = \mathbf{s}$$

where

$$\mathbf{q} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad \mathbf{F}(\mathbf{q}) = \begin{bmatrix} hu & hv \\ hu^2 + \frac{1}{2}gh^2 & huv \\ huv & hv^2 + \frac{1}{2}gh^2 \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} 0 \\ -gh\frac{\partial b}{\partial x} \\ -gh\frac{\partial b}{\partial y} \end{bmatrix}$$

which is the form we will use when looking at the finite volume scheme.

An Unstructured Finite Volume Scheme

For the purposes of modeling the shallow water equations, we will derive a finite volume scheme to work on an unstructured mesh. This gives us the liberty of modeling on complicated geometries.

Finite Volume Scheme

We start with the general hyperbolic conservation law

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla\cdot\mathbf{F}(\mathbf{q}) = \mathbf{s}.$$

We define an arbitrary space-time control volume $\Omega_i \times [t^n, t^{n+1}]$ and integrate over it:

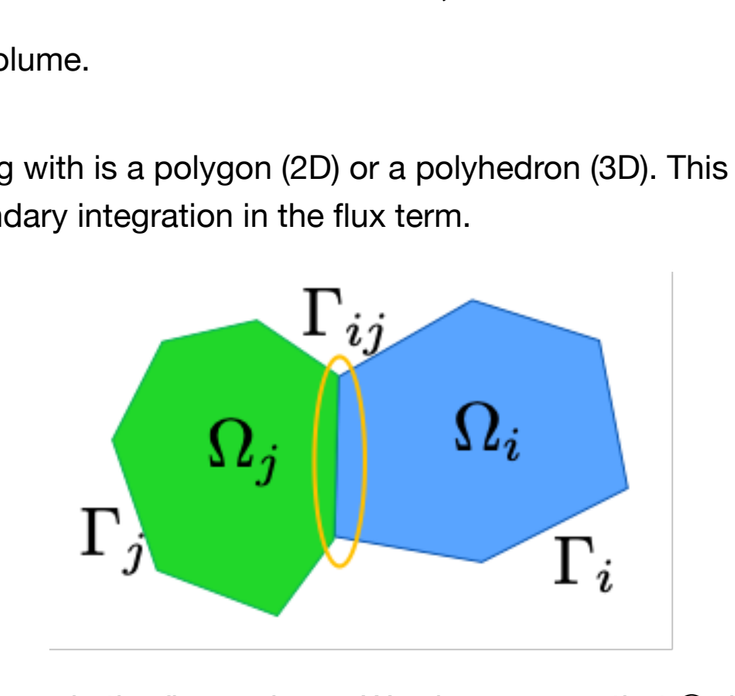
$$\int_{t^n}^{t^{n+1}}\int_{\Omega_i}\frac{\partial \mathbf{q}}{\partial t}d\Omega_i dt + \int_{t^n}^{t^{n+1}}\int_{\Omega_i}\nabla\cdot\mathbf{F}d\Omega_i dt = \int_{t^n}^{t^{n+1}}\int_{\Omega_i}\mathbf{s}d\Omega_i dt$$
$$\Rightarrow \int_{\Omega_i}\left(\mathbf{q}(\mathbf{x}, t^{n+1}) - \mathbf{q}(\mathbf{x}, t^n)\right)d\Omega_i + \int_{t^n}^{t^{n+1}}\int_{\Omega_i}\nabla\cdot\mathbf{F}d\Omega_i dt = \int_{t^n}^{t^{n+1}}\int_{\Omega_i}\mathbf{s}d\Omega_i dt$$

Now we apply the Divergence Theorem to the flux term:

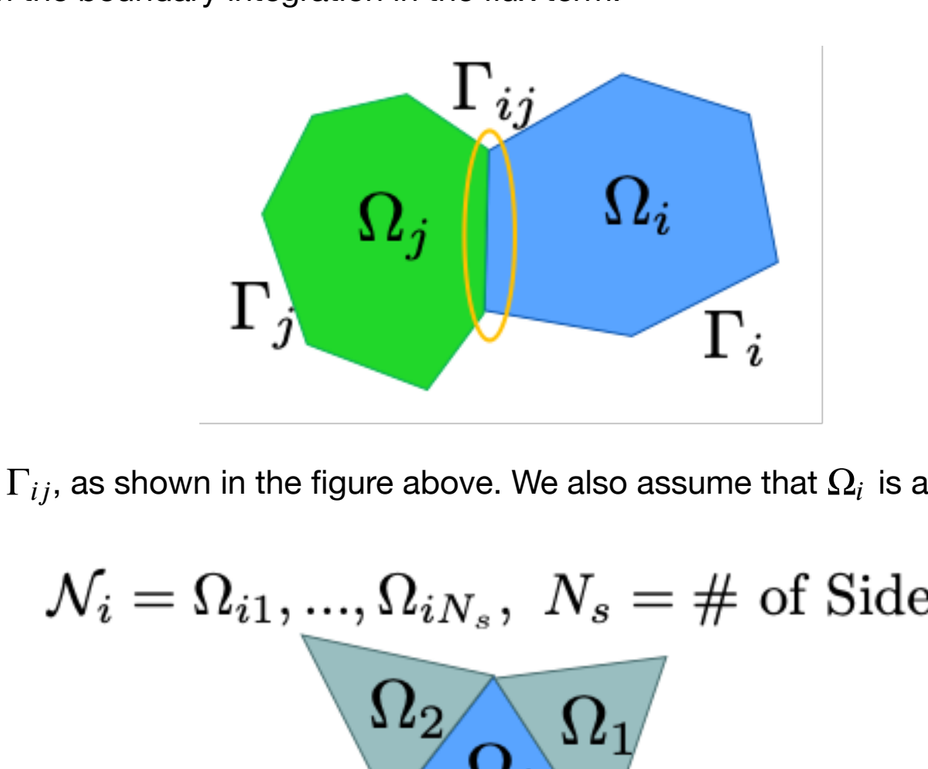
$$\int_{\Omega_i}\left(\mathbf{q}(\mathbf{x}, t^{n+1}) - \mathbf{q}(\mathbf{x}, t^n)\right)d\Omega_i + \int_{t^n}^{t^{n+1}}\int_{\Gamma_i}\mathbf{F}\cdot\hat{n}d\Gamma_i dt = \int_{t^n}^{t^{n+1}}\int_{\Omega_i}\mathbf{s}d\Omega_i dt$$

where $\Gamma_i = \partial\Omega_i$ is the boundary of the space control volume.

Next, we assume that the control volume we are working with is a polygon (2D) or a polyhedron (3D). This allows us to use straight sections for the boundary of Ω_i and sum over the line segments to perform the boundary integration in the flux term.



We define the boundary between two cells as Γ_{ij}, as shown in the figure above. We also assume that Ω_i is a triangle for use in our meshing scheme.



We define the set $\mathcal{N}_i$ as the set of cells that are neighbors to cell i, $\mathcal{N}_i = \Omega_{i1}, \ldots, \Omega_{iN_s}$, where N_s is the number of sides of the polygons we are using (in this case, 3).

Next, we introduce some definitions for the cell average, the ij-flux, and the source average. We define the cell average to be:

$$\mathbf{q}_i^n = \frac{1}{|\Omega_i|}\int_{\Omega_i}\mathbf{q}(\mathbf{x}_i, t^n)d\Omega_i$$

with

$$|\Omega_i| = \int_{\Omega_i}d\Omega_i$$

as the area or volume of the control volume Ω_i. The ij-flux is defined as

$$\mathbf{F}_{ij} = \frac{1}{\Delta t}\frac{1}{|\Gamma_{ij}|}\int_{t^n}^{t^{n+1}}\mathbf{F}(\mathbf{q}_i^n)\cdot\hat{n}_{ij}d\Gamma_{ij}dt$$

with

$$|\Gamma_{ij}| = \int_{\Gamma_{ij}}d\Gamma_{ij}$$

being the length or surface area of the ij-boundary, and $\Delta t = t^{n+1} - t^n$ being the time step. Lastly, we define the source average similar to the cell average:

$$\mathbf{s}_i^n = \frac{1}{|\Omega_i|}\int_{\Omega_i}\mathbf{s}(\mathbf{x}_i, t^n)d\Omega_i.$$

With these definitions, we can express the integral form of the hyperbolic conservation law in terms of cell and source averages, as well as a neighboring flux:

$$\int_{\Omega_i}\left(\mathbf{q}(\mathbf{x}, t^{n+1}) - \mathbf{q}(\mathbf{x}, t^n)\right)d\Omega_i + \int_{t^n}^{t^{n+1}}\int_{\Gamma_i}\mathbf{F}\cdot\hat{n}d\Gamma_i dt = \int_{t^n}^{t^{n+1}}\int_{\Omega_i}\mathbf{s}d\Omega_i dt$$
$$\Rightarrow |\Omega_i|(\mathbf{q}_i^{n+1} - \mathbf{q}_i^n) + \sum_{\Omega_j\in\mathcal{N}_i}\int_{t^n}^{t^{n+1}}\int_{\Gamma_{ij}}\mathbf{F}\cdot\hat{n}_{ij}d\Gamma_{ij}dt = |\Omega_i|\int_{t^n}^{t^{n+1}}\mathbf{s}_i^n dt$$
$$\Rightarrow |\Omega_i|(\mathbf{q}_i^{n+1} - \mathbf{q}_i^n) + \Delta t\sum_{\Omega_j\in\mathcal{N}_i}|\Gamma_{ij}|\mathbf{F}_{ij} = |\Omega_i|\int_{t^n}^{t^{n+1}}\mathbf{s}_i^n dt$$
$$\Rightarrow \mathbf{q}_i^{n+1} = \mathbf{q}_i^n - \frac{\Delta t}{|\Omega_i|}\sum_{\Omega_j\in\mathcal{N}_i}|\Gamma_{ij}|\mathbf{F}_{ij} + \int_{t^n}^{t^{n+1}}\mathbf{s}_i^n dt$$

The final expression above is our finite volume scheme for advancing the solution. In its current form, it is exact due to our integration over the space-time control volume. In order to get a numerical expression, we need to introduce a scheme for approximating the flux term $\mathbf{F}_{ij}$ and handle the source term (which is done differently for each model).

Numerical Flux

For the flux, we will use a 1st-order Rusanov flux

$$\mathbf{F}_{ij} = \frac{1}{2}\left(\mathbf{F}(\mathbf{q}_i^n) + \mathbf{F}(\mathbf{q}_j^n)\right)\cdot\hat{n}_{ij} - \frac{1}{2}s_{max}(\mathbf{q}_j^n - \mathbf{q}_i^n),$$

where

$$s_{max} = \max\left|\Lambda(\mathbf{q}_i^n, \hat{n}_{ij}), \Lambda(\mathbf{q}_j^n, \hat{n}_{ij})\right|.$$

Λ is a function that takes a state vector $\mathbf{q}$ and a normal vector and returns the eigenvalues of the Jacobian of the model. Higher order fluxes can also be derived and implemented, but for our purposes, we will just use the Rusanov flux. Higher order fluxes (WENO, MUSCL, etc.) are more difficult to use on unstructured meshes (though certainly possible).

Application to the Shallow Water Equations

Before we can simulate the shallow water equations, we need three things: 1) how to find the eigenvalues of the Jacboian of the model, 2) how to handle any source terms (bathymetry, ground friction, etc.), and 3) how to include boundary conditions. We start with the Λ function.

Eigenvalues

We can rewrite the hyperbolic conservation law into a quasi-linear form as follows:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{q})}{\partial x} + \frac{\partial \mathbf{g}(\mathbf{q})}{\partial y} = \mathbf{s}$$
$$\rightarrow \frac{\partial \mathbf{q}}{\partial t} + \mathbf{A}(\mathbf{q})\frac{\partial \mathbf{q}}{\partial x} + \mathbf{B}(\mathbf{q})\frac{\partial \mathbf{q}}{\partial y} = \mathbf{s}$$

where $\mathbf{A}$ and $\mathbf{B}$ are the Jacobians of $\mathbf{f}$ and $\mathbf{g}$, respectively.

In order to actually find the Jacobian and subsequent eigenvalues, we use Mathematica to do the symbolic calculations. The Mathematica notebook used to do this can be found with this submission as `SWE_eigensystem.nb`.

From it, we find the eigenvalues of the shallow water equations to be $(u - c, u, u + c)$ and $(v - c, v, v + c)$ for wave speed $c = \sqrt{gh}$. We can define a function `SWELambda(q, n)` to be:

```
function SWELambda(q, n):
    h = q_1
    u = q_2 / q_1
    v = q_3 / q_1
    c = sqrt(g*h)
    un = u*n_1 + v*n_2
    l = [un - c, un, un + c]
    return l
end function
```

Source Terms

The last piece we need to consider before we can simulate the SWEs is how to handle source term. There are various situations. If there is no bathymetry (i.e., the floor is level), then there is no source term and we ignore it. If we have a bathymetry field (time independent), we need to compute the gradient of the bathymetry field and then can use the source average to add the bathymetry contribution each time step at the cell center. Other source terms that could be included in a potential simulation are floor fricition contributions or time dependent additions to the field (i.e., water pouring into or coming out of the field, an impulse injection into the field, etc.). For any of these considerations, we will detail how to include the source term with the respective simulation.

Boundary Conditions

Imposing boundary conditions with this finite volume scheme is more difficult than using a finite difference method, for example. In order to impose boundary conditions, we need to introduce the correct flux to add or subtract from the cell that lives on the boundary.

Reflective boundaries are the most simple. With them, we just reverse the state going out of the domain back into the domain. For mass conservation, this is as simple as setting $q_i^{(1)} = q_j^{(1)}$, where q_i is the boundary cell, q_j refers to the 'neighbor'/boundary cell, and the superscript refers to the first variable in the state vector (in this case h).

Numerical Simulations

To implement this finite volume scheme and simulate the shallow water equations, we implement the method in Matlab.

Note: my original intention was to write a mesh reader, the finite volume scheme, and any visualization in C++. However, trying to install and use various software packages (VTK, VTK-m, libmesh, etc.) to read in a `.vtk` file took too long for me to finish for the purpose of this project. As such, I adjusted some Matlab code from a numerical methods in hyperbolic conservation law course. The code will be available with this submission, and I do intend to continue working on the C++ code for research and enthuiast purposes. I want to write a program that works on `.vtk` files that I can generate with a number of mesh/geometry generation tools (gmsh is a great, free example), constructs a mesh object, and employs the finite volume scheme described above to simulate the shallow water equations (or other hyperbolic conservation laws, i.e., gas dynamics, shock waves, etc.). I want to use `.vtk` files becuase they can be imported into ParaView for super cool visualizations.

For this project, we look at two simulations: A simple "fluid in a box" simulation to show mass conservation and the dynamics of the system, and a wave simulation to simulate an incoming wave.

The mesh used in these simulations is a triangular mesh built from a uniform point cloud. Matlab's `delaunay` triangulation function is used to generate the mesh structure. In theory, this would also work on an arbitrary point cloud, though for the purposes of this project, we just look at this uniform point cloud in a rectangular domain.

Simulation 1

In this simulation, we set the boundary conditions to be all reflective and initial conditions to a Gaussian spread

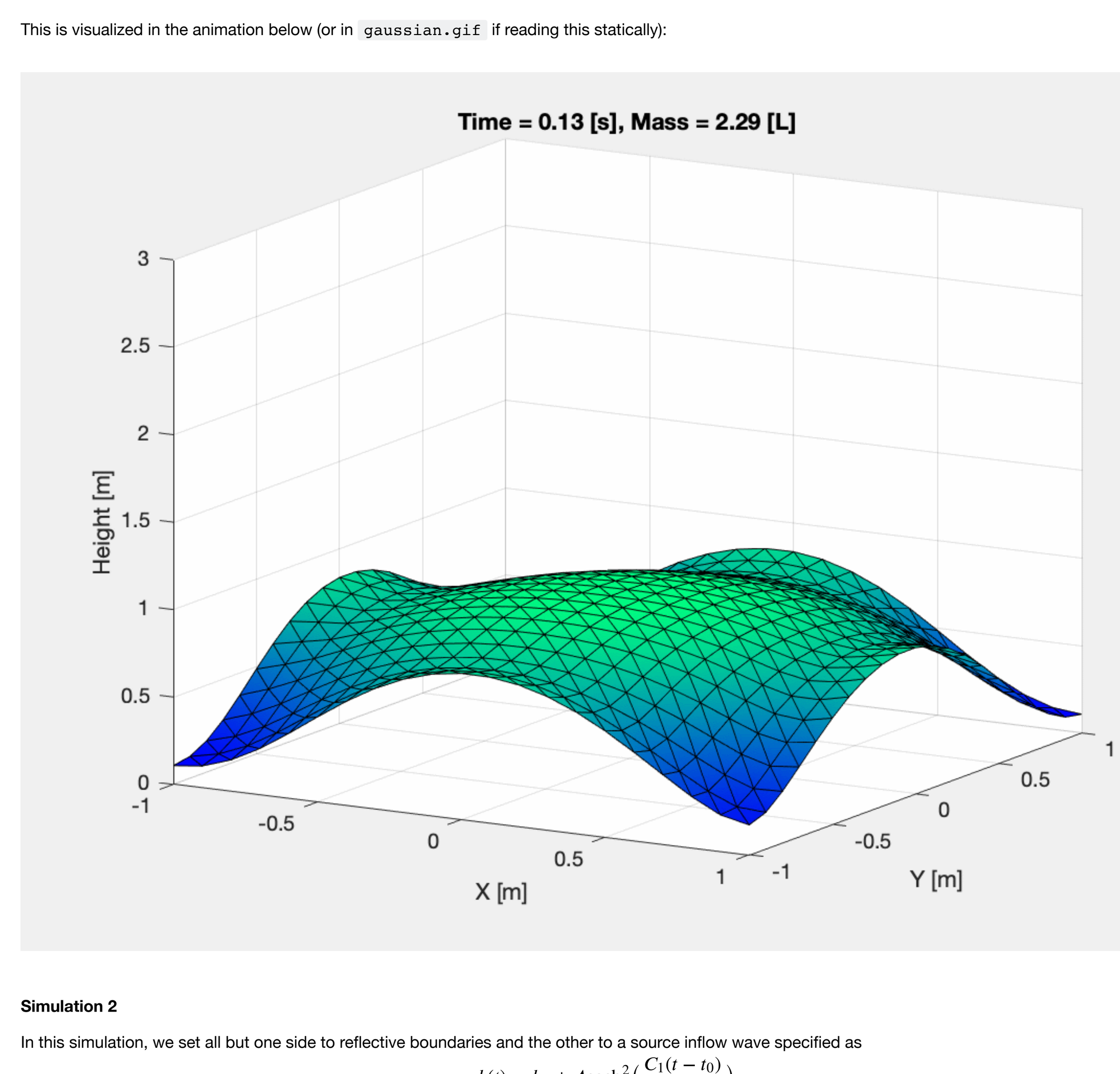
$$h(x, y, t = 0) = A\exp\left(-\left(\frac{(x - x_0)^2}{2\sigma_x^2} + \frac{(y - y_0)^2}{2\sigma_y^2}\right)\right)$$

To verify conservation, we compute the total mass in the domain as

$$\begin{aligned} m &= \int_{x_L}^{x_U}\int_{y_L}^{y_U}\rho h(x, y)dxdy \\ &= (x_U - x_L)(y_U - y_L)\rho\sum_{\Omega_i}h_i \end{aligned}$$

and check if it remains constant throughout the simulation.

This is visualized in the animation below (or in `gaussian.gif` if reading this statically):



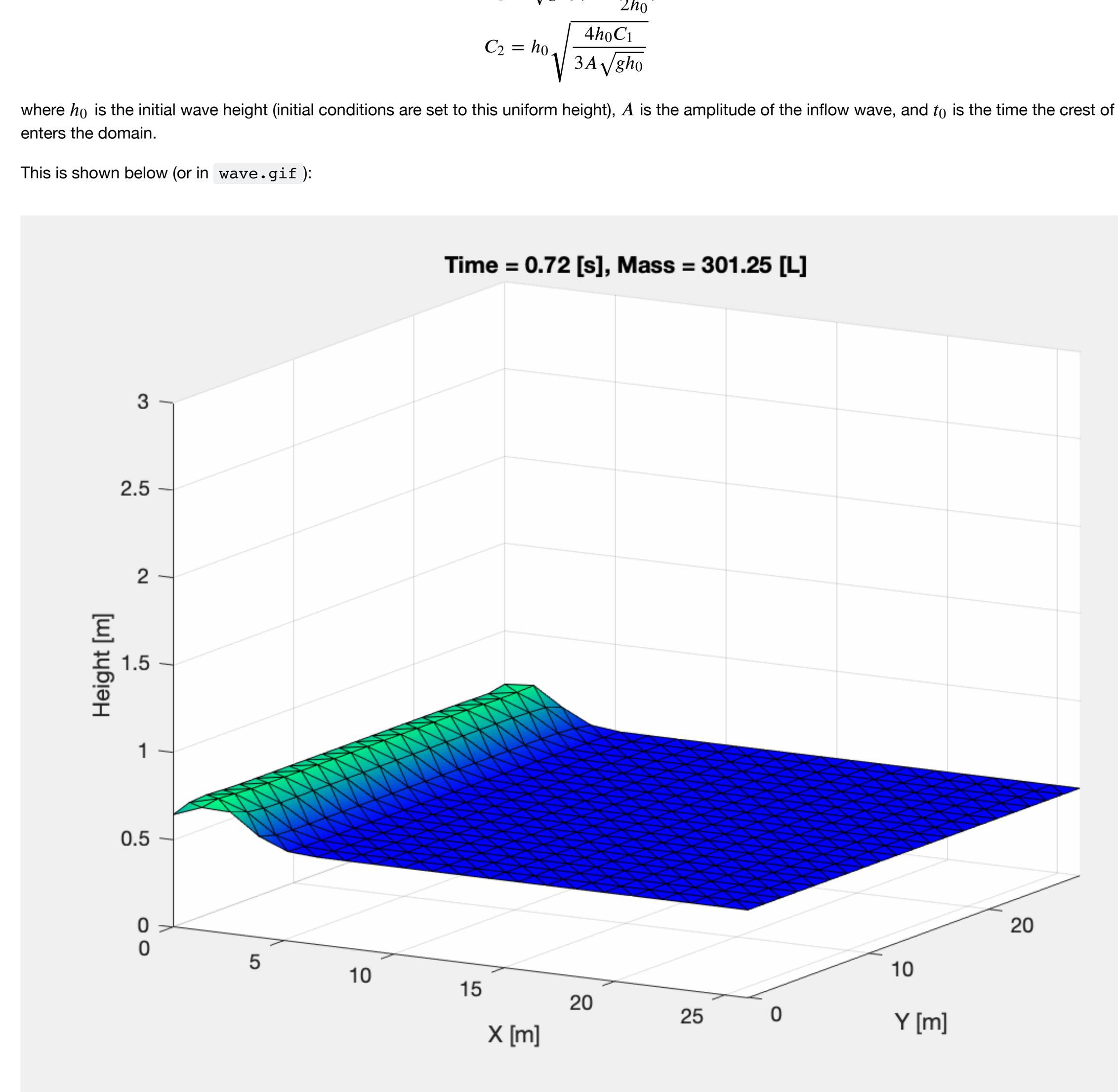
Simulation 2

In this simulation, we set all but one side to reflective boundaries and the other to a source inflow wave specified as

$$h(t) = h_0 + A\text{sech}^2\left(\frac{C_1(t - t_0)}{C_2}\right)$$
$$C_1 = \sqrt{gh_0}\left(1 + \frac{A}{2h_0}\right)$$
$$C_2 = h_0\sqrt{\frac{4h_0C_1}{3A\sqrt{gh_0}}}$$

where h_0 is the initial wave height (initial conditions are set to this uniform height), A is the amplitude of the inflow wave, and t_0 is the time the crest of the wave enters the domain.

This is shown below (or in `wave.gif`):



The idea behind this simulation is to use it as a verification study. Briggs, et al. perform an experiment with a wave pool that introduces a wave across one boundary, with a cone shaped island in the middle of the wave pool. Probes are set at different locations and the height of the water is monitored through the experiment. This simulation is the start of this validation; next steps are to introduce bathymetry and keep track of the water height at the locations of the probe. This work is ongoing.

Conclusion

In this project, we derived the shallow water equations from the incompressible Navier-Stokes equations, described a finite volume scheme for use on an unstructured mesh, and then simulated the shallow water equations with this scheme in Matlab. The shallow water equations are used to simulate incompressible fluids in domains where the horizontal length scale is much larger than the vertical fluid height. The SWEs are a free surface model. The finite volume scheme we implemented is a 1st order accurate Rusanov flux that works on a mesh of polygons. We use a triangular mesh for the simulations. Finally, we look at two simulations, one of an initial Gaussian distribution in a box and track the mass for verification, and one of a wave being introduced into the domain from the boundary. These simulations form the foundation for additional simulations. Next steps would be to introduce bathymetry, introduce floor friction, solve on more complex geometries, and add higher order physics to improve the model. Indeed, this is part of my current PhD research: to look at solvers for a shallow water model with an added dispersive term to the source term.

Sources

1) https://core.ac.uk/download/pdf/290001666.pdf

2) https://www.researchgate.net/profile/Costas-Synolakis/publication/227132664_Laboratory_Experiments_of_Tsunami_Runup_on_a_Circular_Island/links/02e7e51f57e19b7e1e000000/Laboratory-Experiments-of-Tsunami-Runup-on-a-Circular-Island.pdf

3) http://www.mathematik.tu-dortmund.de/lsiii/cms/papers/Kuehbacher2009.pdf

4) http://empslocal.ex.ac.uk/people/staff/gv219/ecmm719/ess-ecmm719.pdf

5) *Finite-Volume Methods for Hyperbolic Problems*, LeVeque.

6) *Advanced Numerical Methods for Hyperbolic Equations and Applications*, Dumbser.