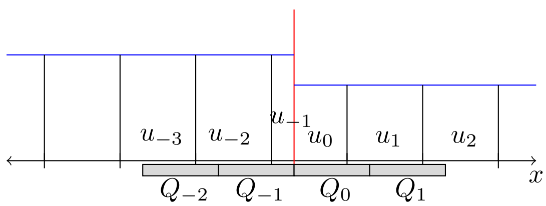
Non-LTS H-box Barrier Solver for 1D Shallow Water Equations

Chanyang Judah Ryoo, Kyle Mandli, Marsha Berger

January 20, 2020

1 Problem Setup

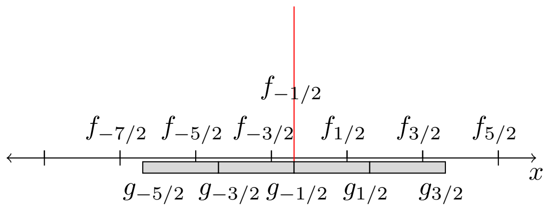


Here we have a problem of solving the 1D shallow water equations (SWE) with a zero-width barrier that is placed within a grid cell. This creates two small cells, denoted as u_{-1} and u_0 in the diagram above. In the case when the wall is placed on a grid edge, Jiao Li has come up with a method of redistributing the fluxes which directly updates the left and right cells of the wall. In the case when the wall is located inside the cell, the two small cells impose a limit on the time step because of the CFL condition. One way to circumvent taking many small time steps is to use large time stepping (LTS) method, which violates the strict CFL condition but tracks the waves from the Riemann problems throughout the larger-than-allowed time step and updates the small and neighboring cells appropriately. However, this LTS method involves complicated and repeated calculation of time substeps and cell averages. In this work, we show a h-box approach to circumventing the small cell problem and the cumbersome LTS method, which reduces to updating regular sized grid cells and using these values to update the small and neighboring cells. In the diagram the shaded grid cells are what are called h-boxes, which are mixed averages of the small cell with the neighboring cell.

2 The H-box Method

The method is divided into two parts. The first part is to find a consistent and conservative way of updating the h-box cells, which are simply the cell averages of their covering physical cells. The second part is to update the physical cells proportionally according to the updated h-box cells.

To find consistent and conservative updates of the h-boxes, we find the fluxes at the h-box edges based on the fluxes at the physical grid edges in a conservative manner. Note that all the fluxes at the physical grid edges are all well-defined except at the wall. We will discuss how to define the flux at the barrier later.



We must first somehow prescribe two of them $(g_{-1/2}, g_{1/2})$ to solve for the last third $(g_{3/2})$. We set $g_{-1/2} := f_{-1/2}$ because they align with each other, and we set $g_{1/2} := f_{1/2}$ by proximity. To find what $g_{3/2}$ should be, we impose conservation.

To maintain conservation, we must have total mass from the h-box updates to be the same as the total mass from the standard Godunov updates. That is, we use the following Godunov updates:

$$u_0^{n+1} = u_0^n - \frac{\Delta t}{(1-\alpha)\Delta x}(f_{1/2} - f_{-1/2}) \tag{1}$$

$$u_1^{n+1} = u_1^n - \frac{\Delta t}{\Delta x}(f_{3/2} - f_{1/2}) \tag{2}$$

$$u_2^{n+1} = u_2^n - \frac{\Delta t}{\Delta x}(f_{5/2} - f_{3/2}), \tag{3}$$

which are the right half cells of the barrier for simplicity. The other half cells are treated in similar manner. The total mass at step $n+1$ is given by $u_0^{n+1} + u_1^{n+1} + u_2^{n+1}$. This sum must be equal to the sum of h-box based updates:

$$
\begin{aligned}
u_0^{n+1} &= Q_0^{n+1} \\
&= Q_0^n - \frac{\Delta t}{\Delta x}(g_{1/2} - g_{-1/2}) \\
&= \alpha u_1^n + (1-\alpha)u_0^n - \frac{\Delta t}{\Delta x}(f_{1/2} - f_{-1/2}) && (4) \\
u_1^{n+1} &= \alpha Q_0^{n+1} + (1-\alpha)Q_1^{n+1} \\
&= \alpha(\alpha u_1^n + (1-\alpha)u_0^n - \frac{\Delta t}{\Delta x}(f_{1/2} - f_{-1/2})) + (1-\alpha)(Q_1^n - \frac{\Delta t}{\Delta x}(g_{3/2} - g_{1/2})) \\
&= (\alpha^2 u_1^n + \alpha(1-\alpha)u_0^n - \frac{\alpha\Delta t}{\Delta x}(f_{1/2} - f_{-1/2})) \\
&+ (1-\alpha)(\alpha u_2^n + (1-\alpha)u_1^n - \frac{\Delta t}{\Delta x}(g_{3/2} - f_{1/2})) && (5) \\
u_2^{n+1} &= \alpha Q_1^{n+1} + (1-\alpha)u_2^{n+1} \\
&= \alpha(\alpha u_2^n + (1-\alpha)u_1^n - \frac{\Delta t}{\Delta x}(g_{3/2} - f_{1/2})) + (1-\alpha)(u_2^n - \frac{\Delta t}{\Delta x}(f_{5/2} - f_{3/2})) && (6)
\end{aligned}
$$

Summing the three equations above and setting it equal to the sum of (1)-(3), we obtain

$$
g_{3/2} = \alpha f_{5/2} + (1-\alpha)f_{3/2} + \frac{\alpha^2}{1-\alpha}(f_{1/2} - f_{-1/2}) + \frac{\Delta x}{\Delta t}\alpha^2(u_1^n - u_0^n). \tag{7}
$$

Doing the same for the left half of the barrier, where now $g_{-3/2} := f_{-3/2}$, we obtain

$$
g_{-5/2} = \alpha f_{-5/2} + (1-\alpha)f_{-7/2} + \frac{(1-\alpha)^2}{\alpha}(f_{-3/2} - f_{-1/2}) + \frac{\Delta x}{\Delta t}(1-\alpha)^2(u_{-1}^n - u_{-2}^n) \tag{8}
$$

2.1 Wave propagation form

The updates based on differences of fluxes at the edges can be put in terms of updates in wave propagation form as follows. We use the relation

$$
\mathcal{A}^-\Delta Q_{i-1/2} = g_{i-1/2} - F(Q_{i-1}) \tag{9}
$$
$$
\mathcal{A}^+\Delta Q_{i-1/2} = F(Q_i) - g_{i-1/2}. \tag{10}
$$

and

$$
\mathcal{A}^-\Delta u_{i-1/2} = f_{i-1/2} - F(u_{i-1}) \tag{11}
$$
$$
\mathcal{A}^+\Delta u_{i-1/2} = F(u_i) - f_{i-1/2} \tag{12}
$$

where $F(q) = F([h, hu]) = [hu, hu^2 + \frac{1}{2}gh^2]$.

2.2 Flux at barrier

To compute the flux at the barrier, $f_{-1/2}$ we make two observations. First is that the flux at the barrier must not add or subtract mass, which then gives a relation between $f_{-1/2}$ and neighboring fluxes. Second is that for actual calculation purposes we must use some redistribution of fluxes to obtain the right going waves and left going waves at the barrier.

2.2.1 Mass balancing definition of flux

First, we impose the condition of no mass being destroyed or created at the barrier. This is equivalent to summing up the following:

$$u_0^{n+1} = u_0^n - \frac{\Delta t}{(1-\alpha)\Delta x}(f_{1/2} - f_{-1/2}) \tag{13}$$

$$u_1^{n+1} = u_1^n - \frac{\Delta t}{\Delta x}(f_{3/2} - f_{1/2}) \tag{14}$$

$$u_2^{n+1} = u_2^n - \frac{\Delta t}{\Delta x}(f_{5/2} - f_{3/2}) \tag{15}$$

$$u_{-1}^{n+1} = u_{-1}^n - \frac{\Delta t}{\alpha\Delta x}(f_{-1/2} - f_{-3/2}) \tag{16}$$

$$u_{-2}^{n+1} = u_{-2}^n - \frac{\Delta t}{\Delta x}(f_{-3/2} - f_{-5/2}) \tag{17}$$

$$u_{-3}^{n+1} = u_{-3}^n - \frac{\Delta t}{\Delta x}(f_{-5/2} - f_{-7/2}) \tag{18}$$

and setting it equal to

$$\sum_{i=-3}^{2} u_i^{n+1} = \sum_{i=-3}^{2} u_i^n - \frac{\Delta t}{\Delta x}(f_{5/2} - f_{-7/2}) \tag{19}$$

Matching and solving for $f_{-1/2}$ then gives us:

$$f_{-1/2} = \frac{(1-\alpha)^2}{1-2\alpha} f_{-3/2} - \frac{\alpha^2}{1-2\alpha} f_{1/2}. \tag{20}$$

We can use this relation in the definition of $g_{3/2}$ and $g_{-5/2}$ derived above. This way of defining $f_{-1/2}$ seems to be more stable when solving a problem with water on both sides of the wall. But for problems with dry state on either side of the wall, this way of defining flux introduces unstable oscillations, and we simply resort to equations (11)-(12) to define $f_{-1/2}$ along with the waves which will be discussed next.

2.2.2 Redistribution at the barrier and Setting up ghost state

To actually come up with the positive direction waves and negative direction waves at the barrier, we must introduce a ghost cell, denoted q^w, and solve two Riemann problems (henceforth abbreviated as $\mathrm{RP}(q_i, q_j)$), which will be $\mathrm{RP}(q_l, q^w)$ and $\mathrm{RP}(q^w, q_r)$, and redistribute the waves arising from these problems (Ph.D. Thesis, Jiao Li). For the choice of physical cells $q_{l,r}$, the left state q_l could be the small cell u_{-1} or the h-box cell Q_{-1} and $q_r = u_0$ or Q_0. However, when there is a dry state on one side of the wall (i.e. $q_i = 0$, $i = l, r$), Li's redistribution method does not capture the water movement onto the dry bed accurately. To mitigate this problem, we use the augmented solver in GeoClaw, which uses three waves to solve one RP and redistribute them.

The ghost state

Introducing this augmented solver complicates the algebra of redistributing fluxes, as now there are in total 6 variables to work with (three for each problem) in order to solve the problem:

$$[r_1, r_2, r_3, r_4, r_5, r_6]\beta = [\rho_1, 0, 0, 0, \rho_2, \rho_3]\tilde{\beta}$$

where the six r_i are the 6 eigenvectors arising from the two RPs, and the three ρ_i are the 3 chosen eigenvectors to which all the fluxes will be redistributed. Theoretically one must solve for $\tilde{\beta}$ exactly in order to solve for the ghost state. However, we can still obtain good results by (1) setting the ghost state to have bathymetry equal to the height of the wall as expected, and (2) to set the water height to be the average of the left and right "star heights" minus the wall height (or one of them minus the wall height, if the other side is dry) and (3) the water momentum to average of the "star momentum" (or one of them, if the other side is dry). The star heights and momentum are the middle states that result from running the water up against the wall from both left and right sides, i.e. solving $\mathrm{RP}(q_l, q_l^t)$ and $\mathrm{RP}(q_r^t, q_r)$, where $q_k^t = [h_k, -(hu)_k]$, $k = l, r$. If the water has enough momentum to get over the wall from either side, then we set the ghost cell with these values. If not, it is dry. In symbols, we have:

$$h^w = \begin{cases} 0.5(h_l^* + h_r^*) - \omega, & \text{if } h_l \neq 0 \neq h_r \\ h_j^* - \omega, & \text{if } h_j = 0 \text{ for either } j = l \text{ or } r \end{cases}$$

$$(hu)^w = \begin{cases} \big(0.5(h_l^* + h_r^*) - \omega\big)\big(0.5(u_l^* + u_r^*)\big), & \text{if } h_l \neq 0 \neq h_r \\ \big(h_j^* - \omega\big)\big(u_j^*\big), & \text{if } h_j = 0 \text{ for either } j = l \text{ or } r, \end{cases}$$

where ω is the wall height.

Now we must choose three vectors to which the fluxes arising from the two RPs will be redistributed. To choose three vectors, we take a similar approach as Li. We select (1) the maximum speed arising from the two RPs, (2) the minimum speed, and (3) the average of these two as our three speeds. This averaging of the two speeds is recommended also for various reasons (Ph.D. Thesis, David George, 84-85). The speed then determines our three eigenvectors since the eigenvector is of the form $[1, s, s^2]$. Now in using the augmented solver, there is another linearly degenerate eigenvector of the form $[0, 0, 1]$. We use the form of $[1, s, s^2]$ for *all* the speeds *if* there are water overtopping the wall from *both* sides, since this will help us to accurately capture large rarefactions, if any, and also ensure positivity (George). Otherwise, we only use the maximum and minimum speeds as our eigenvectors, and one linearly degenerate wave in the augmented solver $[0, 0, 1]$ as our third eigenvector. In matrix equation form, the redistribution of waves solves

$$\begin{bmatrix} 1 & 1 & 1 \\ \min(s) & \frac{1}{2}(\min(s)+\max(s)) & \max(s) \\ \min(s)^2 & \frac{1}{4}(\min(s)+\max(s))^2 & \max(s)^2 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{2} \Delta h \\ \sum_{j=1}^{2} \Delta(hu) \\ \sum_{j=1}^{2} \Delta\phi \end{bmatrix} \tag{21}$$

when water overtops from both sides, or

$$\begin{bmatrix} 1 & 0 & 1 \\ \min(s) & 0 & \max(s) \\ \min(s)^2 & 1 & \max(s)^2 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{2} \Delta h \\ \sum_{j=1}^{2} \Delta(hu) \\ \sum_{j=1}^{2} \Delta\phi \end{bmatrix} \tag{22}$$

when water overtops only from one side. Here Δh is the height difference between the physical and ghost states, $\Delta(hu)$ the momentum difference, and $\Delta\phi$ the momentum flux difference. The second matrix reduces to Jiao Li's redistribution using two waves. The left going or right going wave is then determined by the signs of the speeds.

3 Numerical Results

We have five simulations to show. In all cases, the red dots are results of GeoClaw solver with the barrier being represented by jump in bathymetry ($b_j = \omega$), and the black line is the h-box method result with zero-width barrier. There are 400 uniform grid cells from $x = 0$ to 1 for the simulation in red ($\Delta x = 0.0025$), and there are 399 uniform grid cells ($\Delta x = 0.002506$) with one being split into two small cells with $\alpha = 0.1$ and $1 - \alpha = 0.9$ by the barrier for the simulation in black. The differences in results are most likely due to this small discrepancy in the two grids and the fact that the infinitely thin barrier is compared to finite length barrier (0.0025).

The first two simulations are dry state problems (inundation), one with flat bottom and other sloping bottom. The other two are cases with all wet states, one with flat and other with sloping bottom. The final example has waves coming from both sides on a flat bottom, to check overtopping of barrier from both sides of the barrier. See Figures 1-15 below.

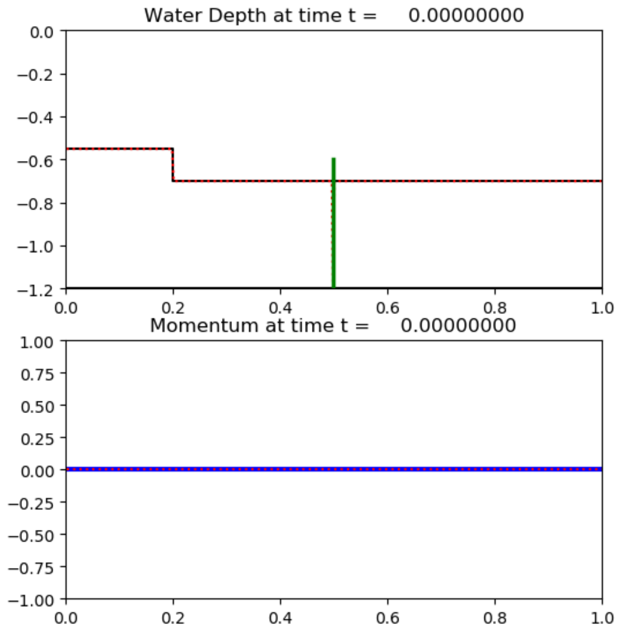


Figure 1: Flat bottom with all wet states at time $t = 0$, with $\Delta x_{\text{red}} = 0.0025$, $\Delta x_{\text{black}} = 0.002506$ and $\alpha = 0.1$ for all plots

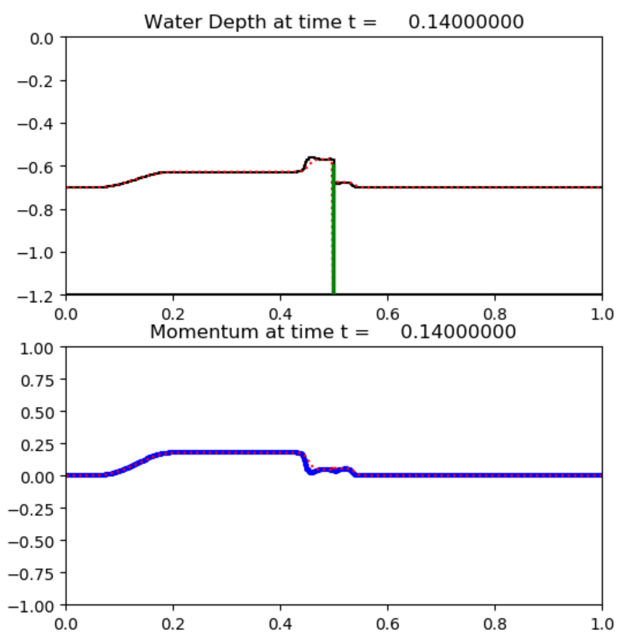


Figure 2: Flat bathymetry with all wet states, overtopping wave, at time $t = 0.14$

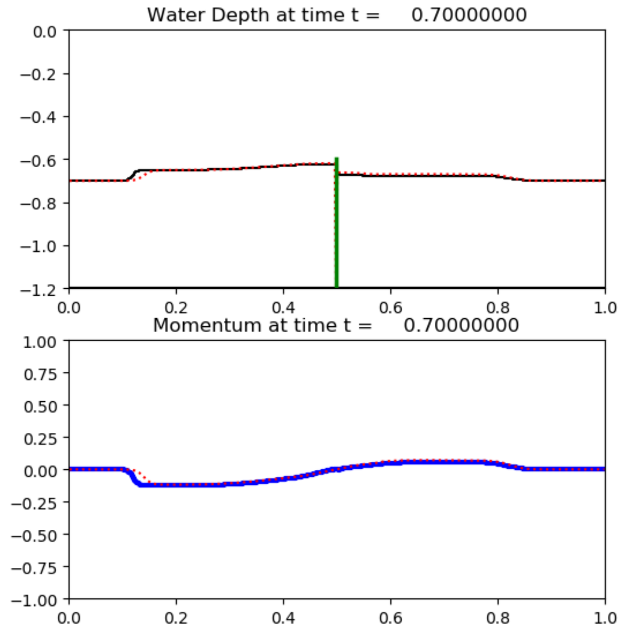


Figure 3: Flat bathymetry with all wet states at time $t = 0.7$

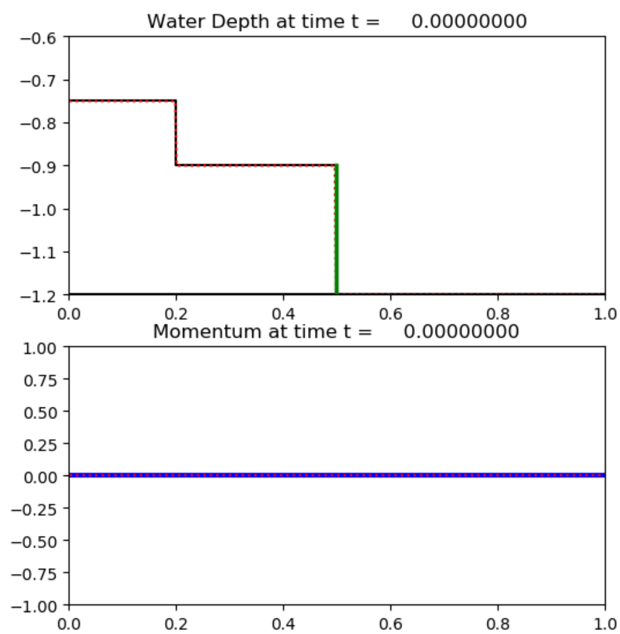


Figure 4: Flat bathymetry with dry state at time $t = 0$

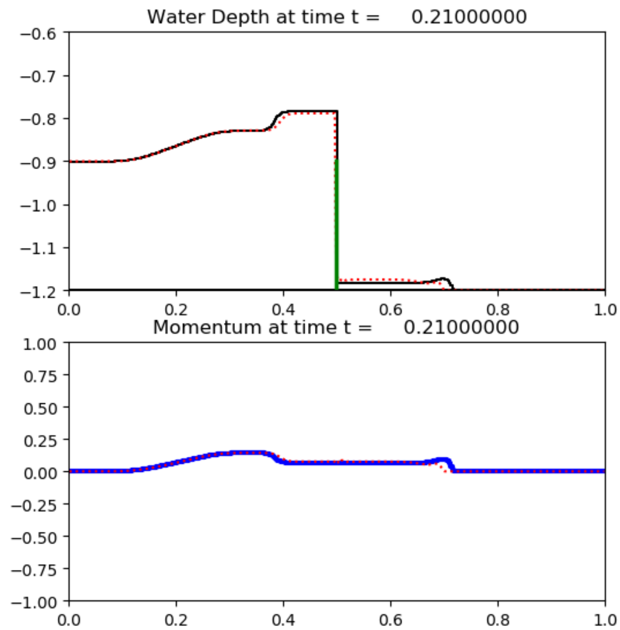


Figure 5: Flat bathymetry with dry state, overtopping wave, at time $t = 0.21$

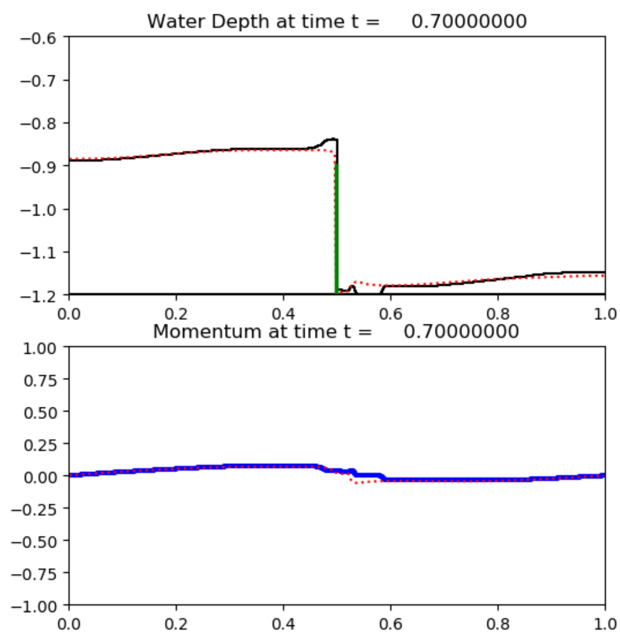


Figure 6: Flat bathymetry with dry state at time $t = 0.7$

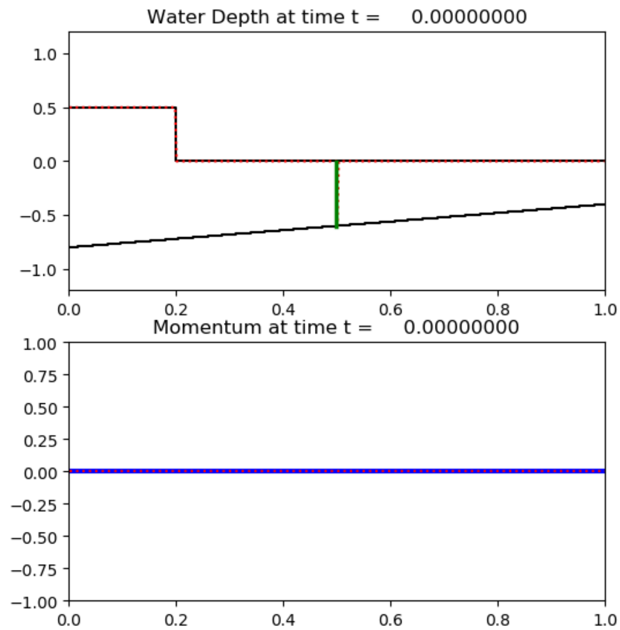


Figure 7: Sloping bathymetry with all wet states at time $t = 0$

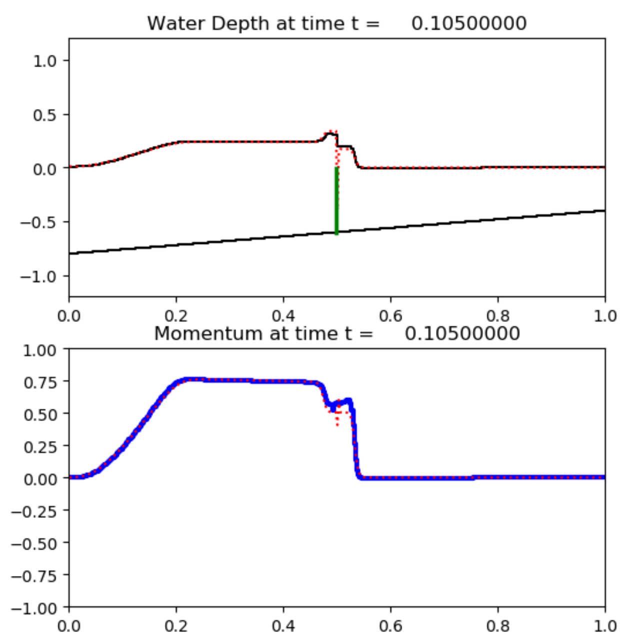


Figure 8: Sloping bathymetry with all wet states, overtopping barrier, at time $t = 0.105$

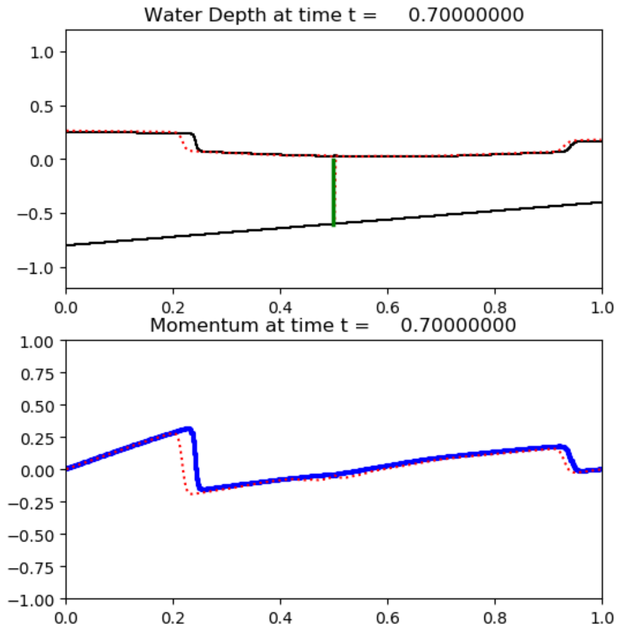


Figure 9: Sloping bathymetry with all wet states at time $t = 0.7$

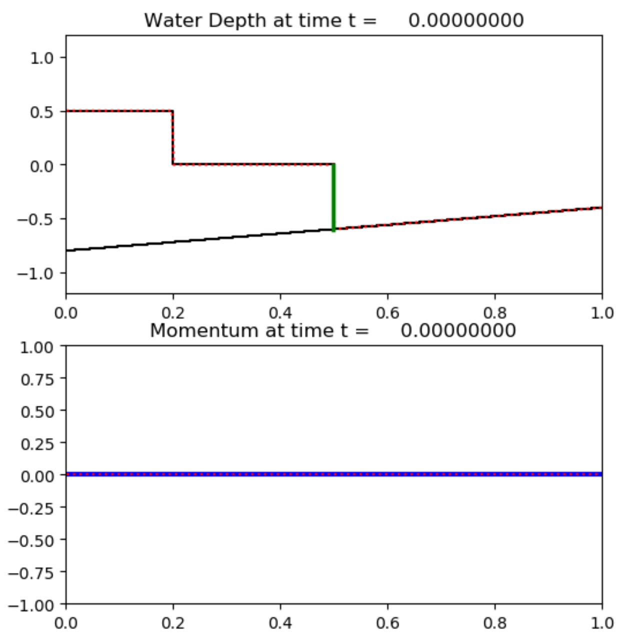


Figure 10: Sloping bathymetry with dry state at time $t = 0.0$

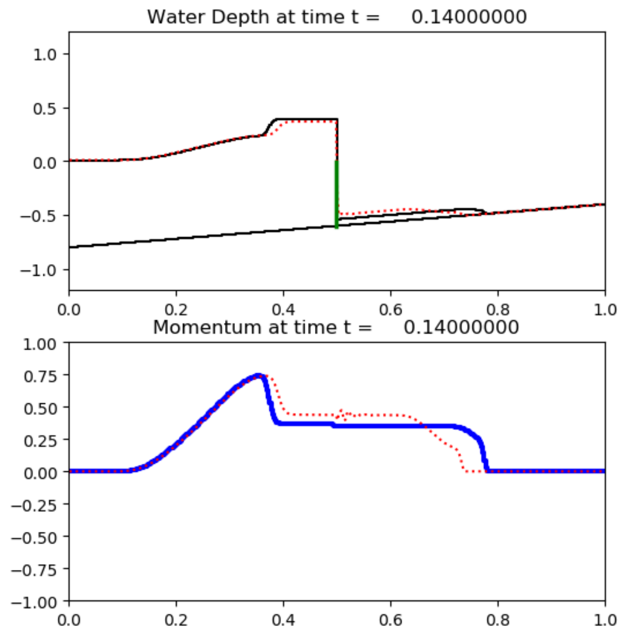


Figure 11: Sloping bathymetry with dry state, overtopping barrier at time $t = 0.14$

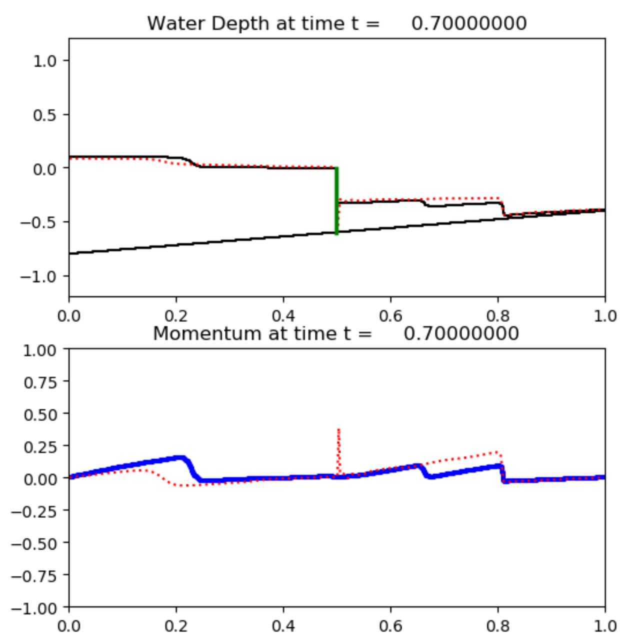


Figure 12: Sloping bathymetry with dry state at time $t = 0.70$

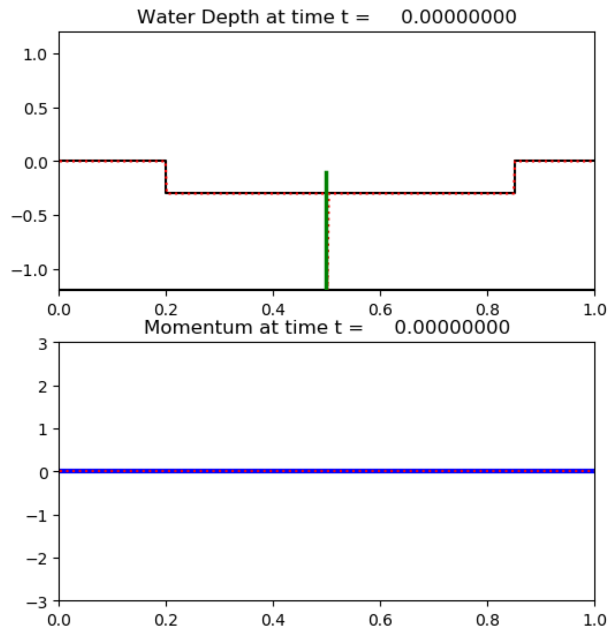


Figure 13: Flat bathymetry at time $t = 0$, to check overtopping from both sides

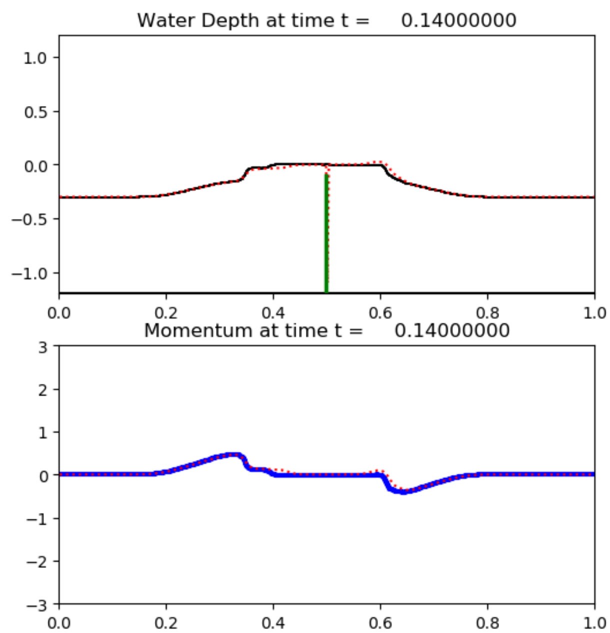


Figure 14: Flat bathymetry at time $t = 0.14$, overtopping barrier from both sides

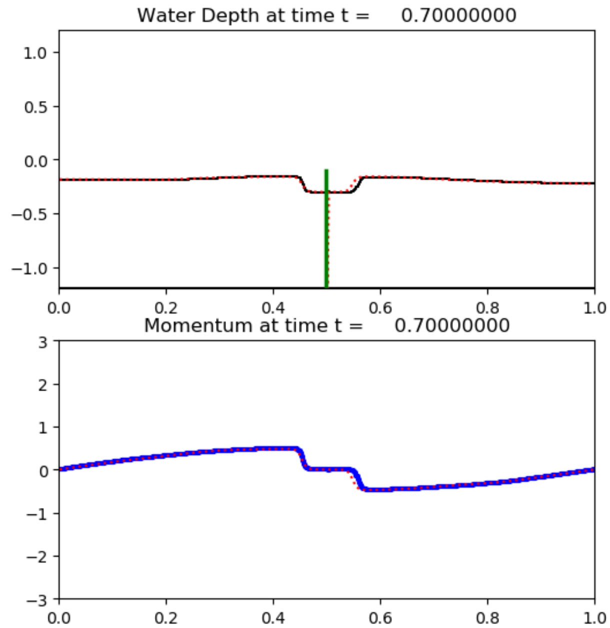


Figure 15: Flat bathymetry with all wet states at time $t = 0.70$

4 Benefits compared to LTS or using cell width wall

The benefits of using the proposed double h-box method are three-fold. First, it circumvents using the cumbersome and complicated large-time-step (LTS) method that needs to track waves crossing the small cells introduced by the barrier placed within a cell. Li proposed this method for the 1D shallow water equation with zero-width barrier in his thesis. Also, this method has potential of being scaled up to 2D, whereas LTS will become impractical in two dimensions.

The second benefit is that we do not need to refine the barrier to be a bathymetry jump with a single cell width. This is beneficial because one of the problems in storm simulation is to simulate storm-protection barriers or obstacles that are much smaller compared to scale of the landscape. Because of this high refinement must be applied around the barrier, which is costly. However, with a zero-width barrier, we do not need to refine any walls, because we are approximating physical barriers with a zero-width, immovable wall.

The third benefit is shown in problems with dry state. Li's LTS method does not accurately solve problems with dry state, but because this method uses robust GeoClaw augmented solver, it handles these well.

5 Conclusion

In conclusion, we have come up with an h-box method that uses two h-boxes around the zero-width barrier, that uses augmented SWE riemann solver of GeoClaw, that is simpler than the LTS method and that conserves mass. Various cases have been checked, with flat bathymetry and varying bathymetry and with dry states and wet states, and with waves overtopping from one side and both sides. Future work that remains is to scale this method up to two dimensions.