

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

Chanyang Judah Ryoo, Kyle Mandli, Marsha Berger

February 21, 2020

1 Problem Setup

Here (see Fig.1) we have a problem of solving the 1D shallow water equations (SWE) with a zero-width barrier of height ω that is placed within a grid cell. This creates two small cells, denoted as u_{-1} and u_0 in the diagram below. In the case when the wall is placed on a grid edge, 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. Li has also implemented this method. However, this LTS method involves complicated and repeated calculation of time substeps and cell averages. In this work, we show an h-box approach to circumvent the small cell problem and the cumbersome LTS method, which reduces to updating “virtual” 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.

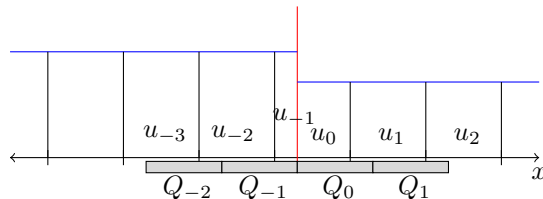


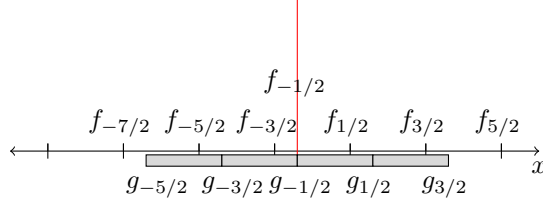
Figure 1: Zero-width barrier in an arbitrary position

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.

2.1 Fluxes of h-boxes

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:

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

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

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

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 $n+1$ given by $Q_0^{n+1} + Q_1^{n+1}$ should be given by:

$$Q_0^{n+1} + Q_1^{n+1} = Q_0^n + Q_1^n - \frac{\Delta t}{\Delta x} [g_{\frac{3}{2}} - g_{-\frac{1}{2}}] \quad (4)$$

or,

$$\begin{aligned} (1-\alpha)u_0^{n+1} + \alpha u_1^{n+1} + (1-\alpha)u_1^{n+1} + \alpha u_2^{n+1} = \\ (1-\alpha)u_0^n + \alpha u_1^n + (1-\alpha)u_1^n + \alpha u_2^n - \frac{\Delta t}{\Delta x} [g_{\frac{3}{2}} - g_{-\frac{1}{2}}]. \end{aligned} \quad (5)$$

Using equations (1)-(3) (because values of Q_0, Q_1 depend on them) and $g_{-\frac{1}{2}} = f_{-\frac{1}{2}}$ gives us further cancellations:

$$-\frac{\Delta t}{\Delta x}[f_{\frac{1}{2}} - f_{-\frac{1}{2}}] - \frac{\Delta t}{\Delta x}[f_{\frac{3}{2}} - f_{\frac{1}{2}}] - \alpha \frac{\Delta t}{\Delta x}[f_{\frac{5}{2}} - f_{\frac{3}{2}}] = -\frac{\Delta t}{\Delta x}[g_{\frac{3}{2}} - f_{-\frac{1}{2}}] \quad (6)$$

This gives us:

$$g_{\frac{3}{2}} = \alpha f_{\frac{5}{2}} + (1 - \alpha)f_{\frac{3}{2}} \quad (7)$$

Doing similar calculation on the other side of the wall gives us:

$$g_{-\frac{5}{2}} = \alpha f_{-\frac{5}{2}} + (1 - \alpha)f_{-\frac{7}{2}} \quad (8)$$

2.2 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

$$\mathcal{A}^- \Delta Q_{i-1/2} = g_{i-1/2} - F(Q_{i-1}) \quad (9)$$

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

and

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

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

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

2.3 The flux $f_{-1/2}$ at barrier: Redistribution

To come up with the positive direction waves and negative direction waves at the infinitely thin barrier, we must introduce a ghost cell, denoted q^w , and solve two Riemann problems (henceforth abbreviated as $\text{RP}(q_i, q_j)$), which will be $\text{RP}(q_l, q^w)$ and $\text{RP}(q^w, q_r)$, and redistribute the waves arising from these problems (Ph.D. Thesis, Jiao Li). We use the augmented solver in GeoClaw, which uses three waves to solve a RP and redistribute them.

2.3.1 Setting the ghost state

First it must be noted that there are certain cases where we do not need to introduce a ghost state and thus solve two RPs:

- The zero-width wall is has zero height:

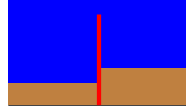


- The zero-width wall is shorter than the bathymetry of either side:

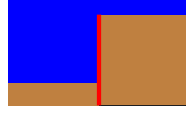


The first case is obvious as to why no ghost cell is necessary. In the second case, it is unnecessary because effectively the thin wall does not affect the flow of water in any way; it is “hidden” in one of the bathymetry. In this case, the RP with the barrier is solved as the RP without the barrier.

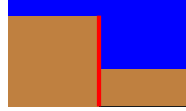
Now in every other case we need to have a ghost state and introduce two RPs. To set the ghost state, we reason as follows. In the case of a steady state, with bathymetric variation, also known as “lake at rest” problem, we need to set the ghost state such that no waves result in any of the RPs near the barrier. Consider the diagram below:



To achieve this steady state, we note that from the perspective of the left cell, the barrier appears as the following diagram:



From the perspective of the right cell, the barrier appears as the following diagram:



Therefore, in this lake-at-rest case, we set the ghost state bathymetry, height and momentum to be

$$\begin{aligned} b^* &= \min(b_L, b_R) + \omega, \\ h^* &= h_R - (b^* - b_R) = h_L - (b^* - b_L), \\ (hu)^* &= 0 \text{ (since at rest)} \end{aligned} \tag{13}$$

See figure 5. In the case where the heights are both above the wall and equal, but the momentum differs on either side of the zero-width barrier, we choose

$$(hu)^* = \min((hu)_L, (hu)_R). \tag{14}$$

Now in the case when h_L and h_R are different, we “upwind” the ghost state, i.e. choose

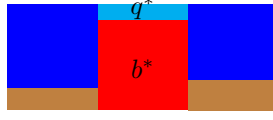


Figure 2: Ghost state for lake at rest

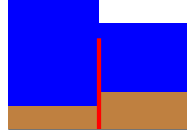
either q_L or q_R to determine the height and momentum, with some modification. Without loss of generality, if the left state has higher water, we set

$$h^* = h_R - (b^* - b_R) \quad (15)$$

with b^* being the same as (13). We call this upwind ghost state because in general when the water is higher on one side, the wave is moving towards the other side and we are basing the evaluation of the ghost cell by q_R (WLOG). The subtraction of $(b^* - b_R)$ is due to the “effective” height of the water that the left state sees being on top of the wall. Effectively the wave from the left “sees” only h^* and b^* . In other words, our ghost height is

$$h^* = \min(h_L - (b^* - b_L), h_R - (b^* - b_R)). \quad (16)$$

In pictures,



becomes as figure 3. Now for our ghost momentum, we choose as (14):

$$(hu)^* = \min((hu)_L, (hu)_R).$$

We choose the minimum of the momentum as this reduces oscillation in the numerical solution to subcritical flow problems. For example, in the case of differing momentum on either side of and with equal heights above the single-cell width wall (see Fig. 19), the actual solution has a dip (i.e. actual oscillation) at the wall. By setting the ghost state momentum to have the minimum momentum, we reduce the instability in the numerical solution of the zero-width wall case when using the more robust augmented RP solver, which will be explained later. There is instability because examples of subcritical flow cannot be modeled with zero-width wall problems by their inherent nature of not being able to locate water on top of the zero-width wall. In these cases, we must resort to a simpler solver where we effectively ignore such dip. See section **2.3.3**.

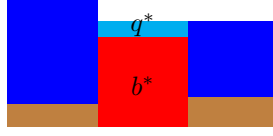


Figure 3: Ghost state for general case of waters higher than barrier

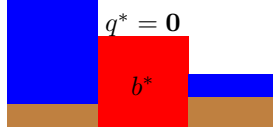


Figure 4: Ghost state for general case of barrier higher than water on either side

In other cases, which are cases where water only flows from one side to the other or not at all, we introduce a *dry* ghost state and must solve two RPs that generally look like figure 4. This is again based on the fact that if there are not water above the barrier on both sides, either side “sees” no water in the other side.

2.3.2 Redistributing waves

Using the 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 with the setup explained above.

Now we must choose three vectors to which the fluxes arising from the two RPs will be redistributed. We select three speeds: (1) \bar{s}_1 , the average of the two 1-wave speeds, (2) \bar{s}_2 , the average of two 2-wave speeds, and (3) \bar{s}_3 , the average of the two 3-wave 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

$$\begin{bmatrix} 1 & 1 & 1 \\ \bar{s}_1 & \bar{s}_2 & \bar{s}_3 \\ \bar{s}_1^2 & \bar{s}_2^2 & \bar{s}_3^2 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^2 \Delta h \\ \sum_{j=1}^2 \Delta(hu) \\ \sum_{j=1}^2 \Delta\phi \end{bmatrix} \quad (17)$$

when water overtops from both sides, or

$$\begin{bmatrix} 1 & 0 & 1 \\ \bar{s}_1 & 0 & \bar{s}_3 \\ \bar{s}_1^2 & 1 & \bar{s}_3^2 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^2 \Delta h \\ \sum_{j=1}^2 \Delta(hu) \\ \sum_{j=1}^2 \Delta\phi \end{bmatrix} \quad (18)$$

when water overtops only from one side (or does not at all). 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.

2.3.3 Subcritical flow

For subcritical flows, i.e. where there will be a “dip” on the wall in the single cell-width problem and the wall is $\max_i b_i$, we note that the inherent nature of the setup of the zero-width wall problem makes it incompatible to capture the variation. This is because the wall has zero width and cannot hold water on top of it, which means the dip has to appear on either side of the wall. However, we have not further explored this as it is more convoluted than our goal of simplifying methods to solve small-cell problems. There is also instability when using the augmented solver because it uses three waves, one of which captures this dip. The only way to address this issue is to ‘erase’ the dip. That is, in the redistribution of the waves from the ghost RPs, we use only the waves propagating outward from the wall ghost state. Doing this reduces to using two waves solver for the redistribution. In this way, the state on the left and the right of the barrier will only reflect the left and right state of the dip and thus look more or less flat. An example in the numerical results will show this.

3 Numerical Results

We have five simulations to show. In four cases, the red dots are results of GeoClaw solver with the barrier being represented by jump in bathymetry ($b_j = b^*$), 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 (thus 400 in total). The differences in results

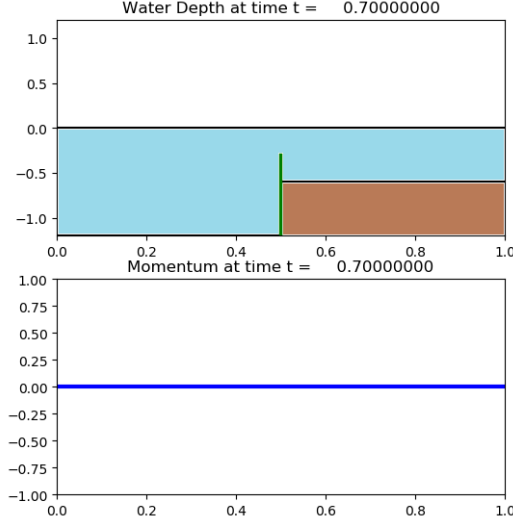


Figure 5: Lake at rest at time $t = 0.7$, with $\Delta x_{\text{red}} = 0.0025$, $\Delta x_{\text{black}} = 0.002506$ and $\alpha = 0.1$ for all plots. Wall BC on both sides.

are due to the flux approximation error inherent in the h-box method, the discrepancy in the two grids' alignment and the fact that the infinitely thin barrier is compared to finite length barrier (0.0025).

The first simulation is the classical “lake at rest” problem, which tests the numerical conservation of the scheme in the presence of a step in bathymetry fig. 5. We also provide the difference in mass from initial time to show numerical conservation in fig. 6. The second 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 step bathymetry jump and other a non-zero velocity steady state problem. See figures 5 - 23.

4 Convergence

Although we compared our results with standard Clawpack simulations with single-cell width walls, we must note that these simulations are not the “true” solutions nor approximations of true solutions. This is because the grid spacings are not aligned—the standard Clawpack simulations being uniformly spaced and the zero-width wall simulation being nonuniformly spaced with two small cells. Also, there is no known true solution to the SWE problem with a zero-width wall. This leads us to study “auto-convergence”, where we will assume that the zero-width wall simulation with 16000 grid cells is the “true” solution and see how our simulations with fewer grid cells converge to this many-grid cells solution.

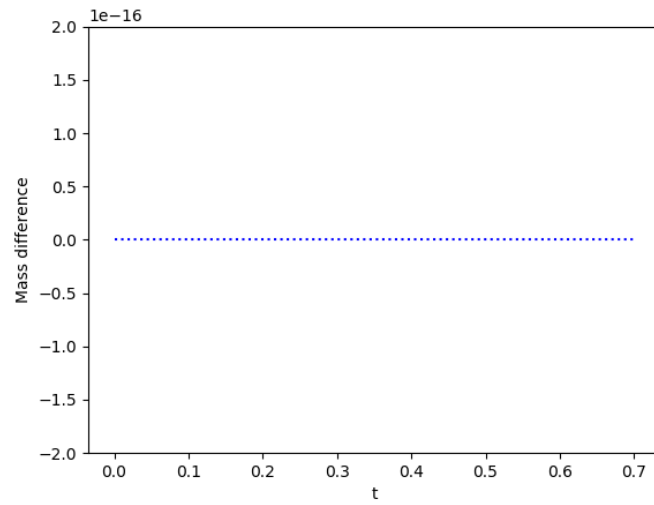


Figure 6: Mass change for lake at rest problem

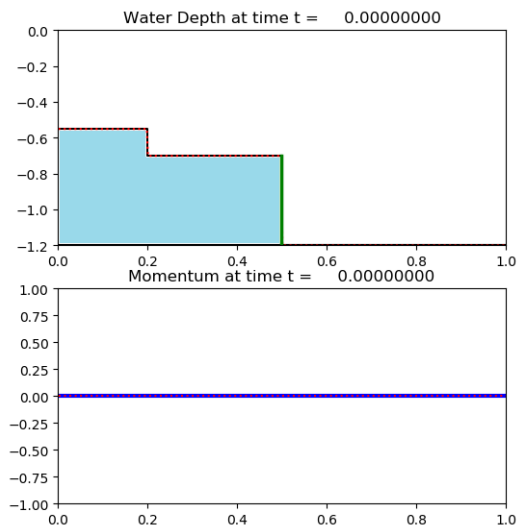


Figure 7: Flat bathymetry inundation at time $t = 0.0$ with wall BC on both sides.

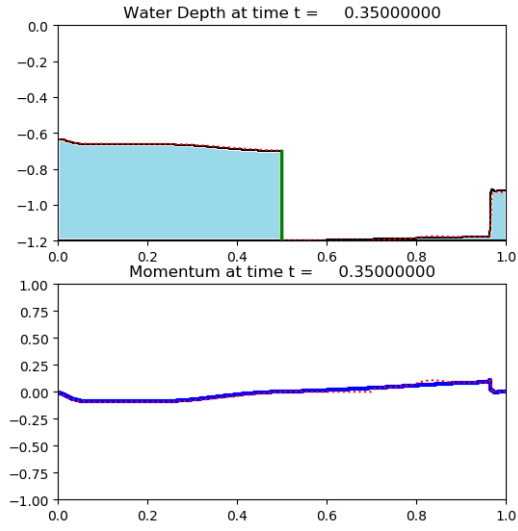


Figure 8: Flat bathymetry inundation at time $t = 0.35$

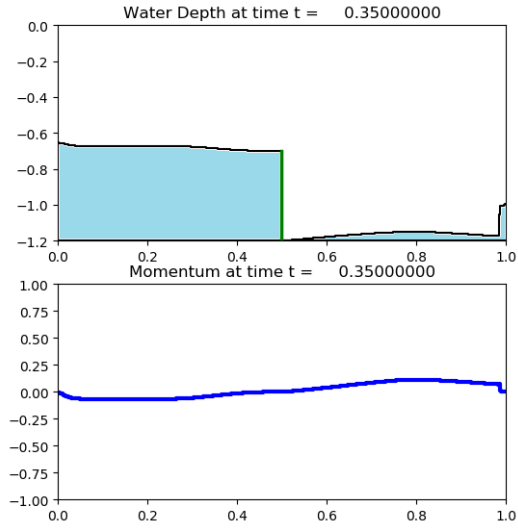


Figure 9: Flat bathymetry inundation at time $t = 0.35$, with old solver

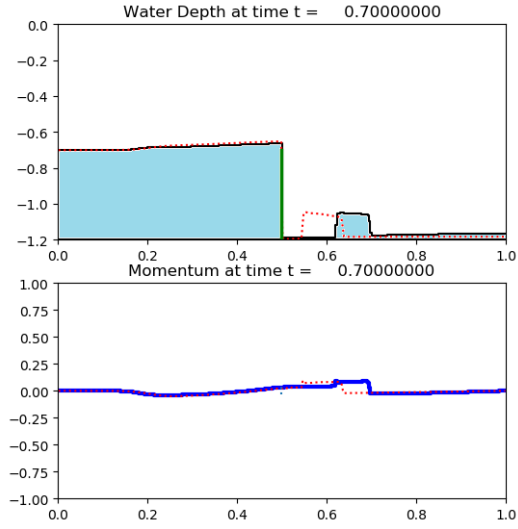


Figure 10: Flat bathymetry inundation at time $t = 0.7$. Notice the gap in the distance between the wave on the RHS of wall. This gap reduces as we refine the grid.

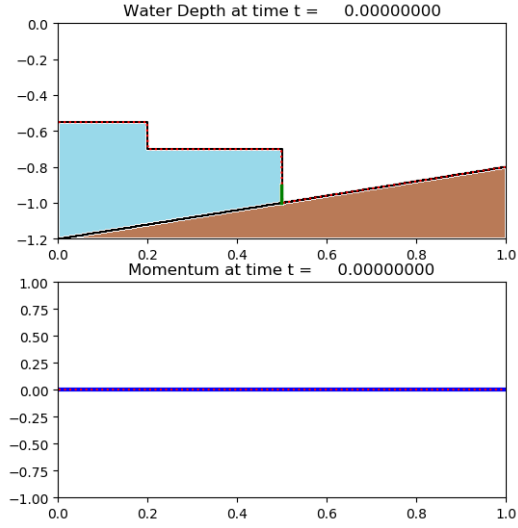


Figure 11: Sloping bathymetry inundation at time $t = 0.0$, with short wall and wall BC on both sides

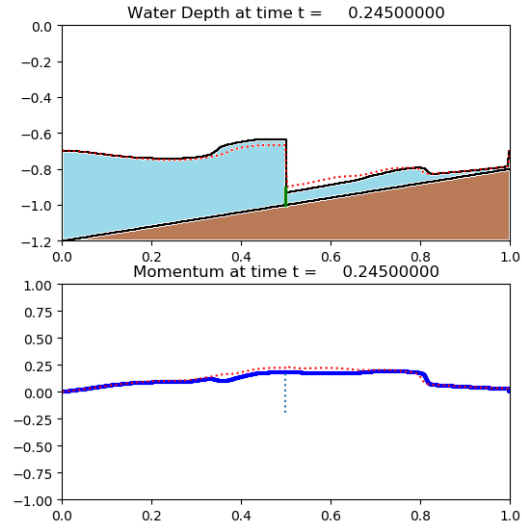


Figure 12: Sloping bathymetry inundation at time $t = 0.245$

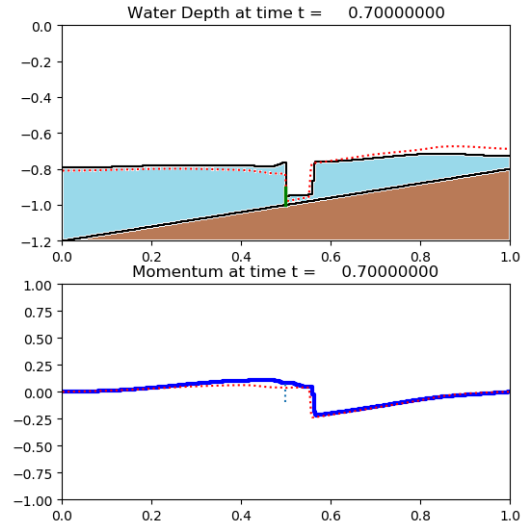


Figure 13: Sloping bathymetry inundation at time $t = 0.7$

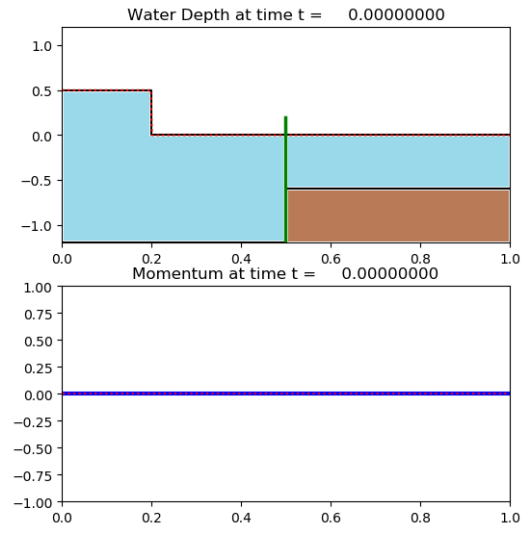


Figure 14: Step bathymetry with right-moving wave at time $t = 0.0$ and outflowing right BC

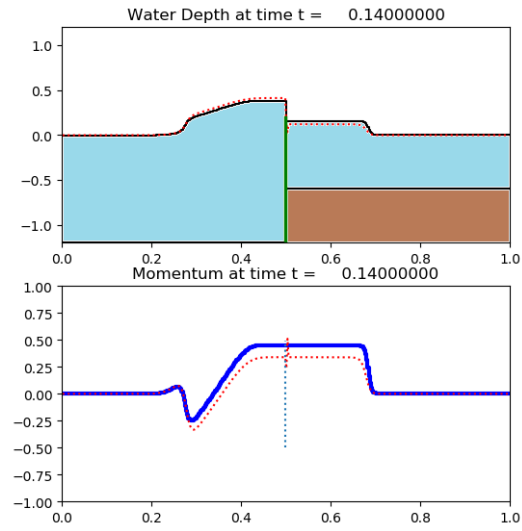


Figure 15: Step bathymetry with right moving wave at time $t = 0.14$

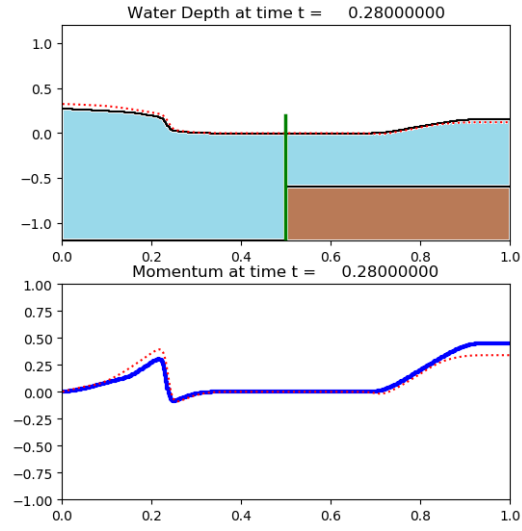


Figure 16: Step bathymetry with right moving wave at time $t = 0.28$

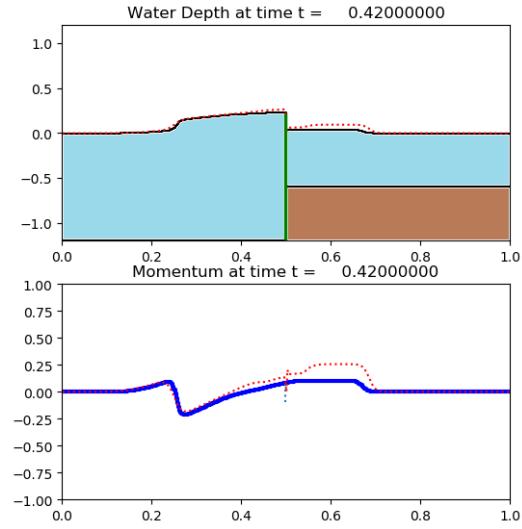


Figure 17: Step bathymetry with right moving wave at time $t = 0.42$

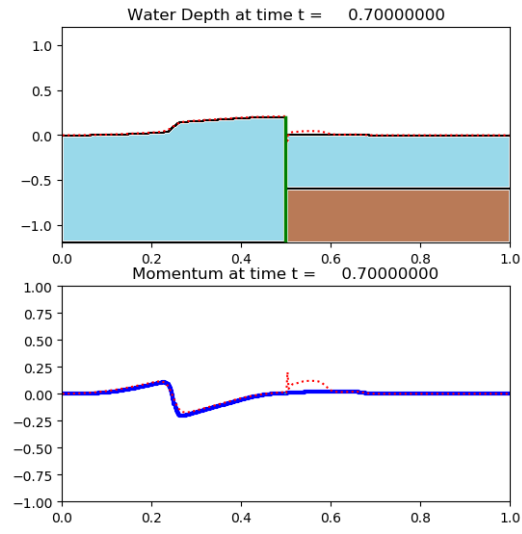


Figure 18: Step bathymetry with right moving wave at time $t = 0.7$

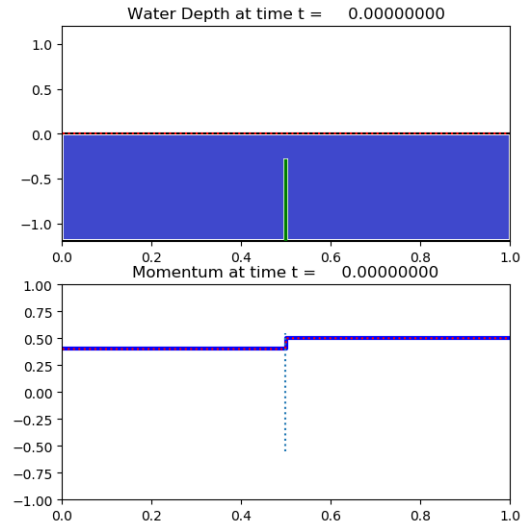


Figure 19: Subcritical flow with constant velocity 0.4 at left BC and outflowing right BC at time $t = 0.0$

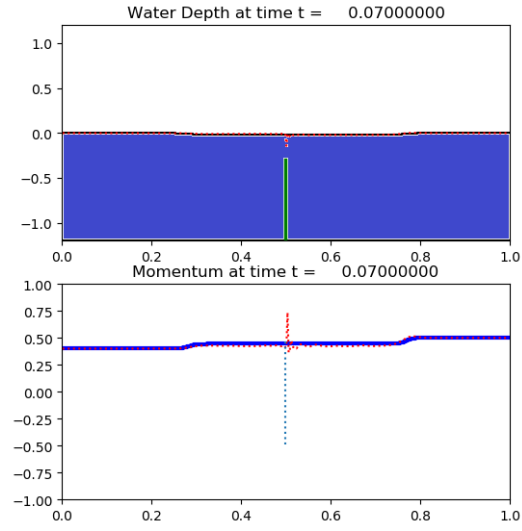


Figure 20: Subcritical flow at time $t = 0.07$. Note the absence of dip in black line.

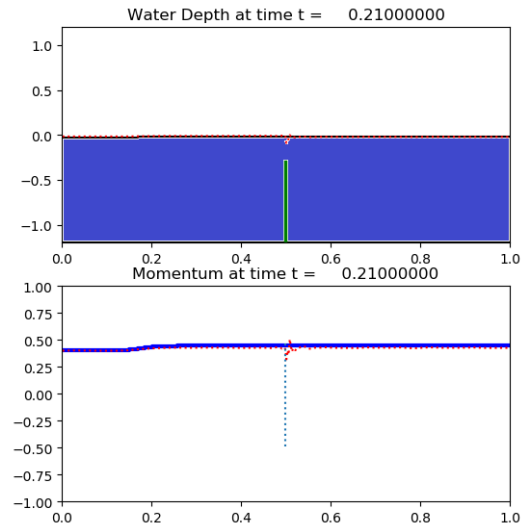


Figure 21: Subcritical flow at time $t = 0.21$

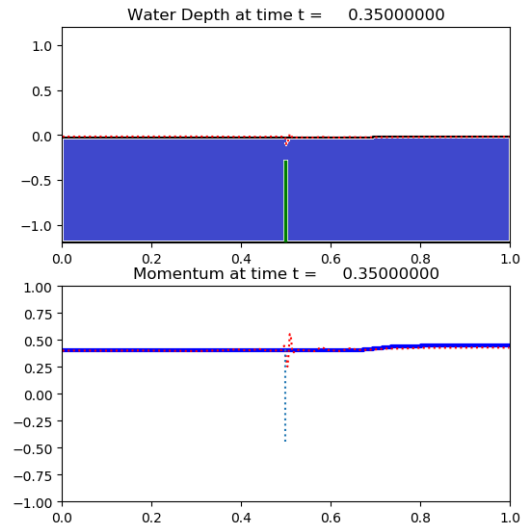


Figure 22: Subcritical flow at time $t = 0.35$.

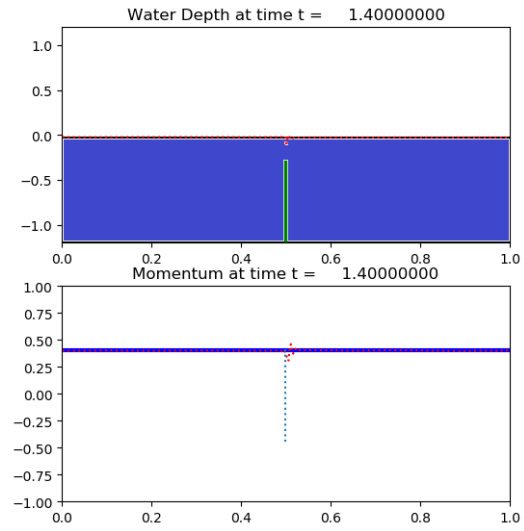


Figure 23: Subcritical flow at time $t = 1.4$.

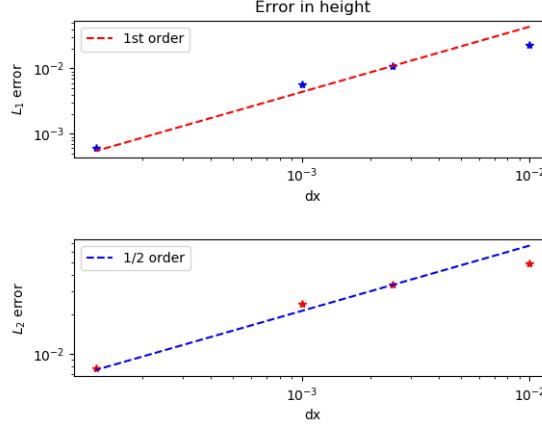


Figure 24: Height error compared to 16000 grid cell result for inundation with flat bottom example.

We choose the first flat bathymetry inundation case to do our convergence study, as we wanted to observe whether the phase gap in the right hand side of the wall would close. As can be seen in figures 24 and 25, there is first order convergence in L_1 norm, as expected since no flux limiters were used and linear interpolation was applied in our updates. In L_2 norm, we see the errors amplified and observe a 1/2 order convergence.

5 Conclusion

5.1 Benefits compared to LTS or using single-cell width wall

The benefits of using the proposed double h-box method are four-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 the numerical conservation of mass that is easily kept in

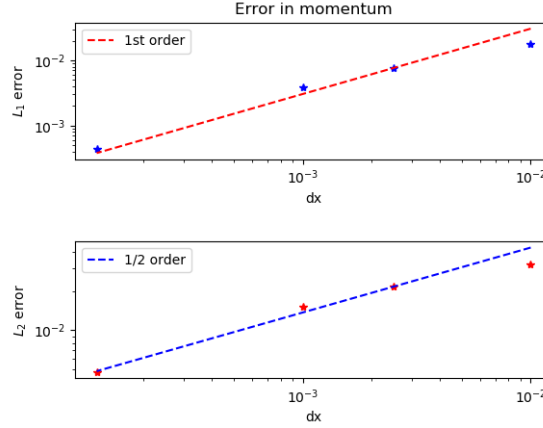


Figure 25: Momentum error compared to 16000 grid cell result for inundation with flat bottom example.

the double h-box method. Although the LTS method is theoretically mass-conserving, it is often hard to show numerically that is the case. However, in our double h-box method, mass is conserved to machine-precision.

The last benefit is the method's better handling of dry-state problems due to the implementation of the augmented solver [D. George, 2008]. Although this solver is more costly, figure 8 and figure 9 show the large difference in the results.

5.2 Summary

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. We observe first order convergence in L_1 norm and 1/2 order convergence in L_2 norm. Various cases have been checked, with flat bathymetry and varying bathymetry and with dry states and wet states, and with nonzero steady states. The only limitation in the 1D method and also expected in the 2D method is the inability to capture subcritical flow over the wall, due to the inherent nature of the problem i.e. how there can be no water on top of the infinitely thin wall.