FOSSEE, IIT Bombay
OpenFOAM Case Study Project
April 8, 2020

Gravity Driven Flow

Subham Kumar Das
Fourth Year B.Tech Student
National Institute of Technology Durgapur

Abstract

In this case-study project, the gravity-driven lock exchange flow between two fluids is simulated using the open-source toolbox, OpenFOAM. The currents generated in the case of mixing between the fluids of different densities is visualized through the phase-fraction contours of the fluids. Adaptive mesh refinement is utilized to capture the interface between the fluids clearly. We have simulated different cases with three pairs of fluids of widely varying density ratios, for partial and full lock release initial conditions given in Wood(2004)[1].

1 Introduction

The gravity driven lock exchange flow occurs when two fluids of different densities, initially at rest, are kept seperated by a vertical barrier in a tank and the barrier is removed to allow the two fluids to flow in opposite directions under the influence of gravity. The hydrostatic pressure causes the denser fluid to flow in one direction along the bottom boundary of the tank, while the lighter fluid flows in the opposite direction along the top boundary of the tank. This type of flow is quite common in nature, like the case of a lock gate with saline water on one side and fresh water on the other being opened. Such cases have been investigated by O'Brien and Cherno (1934)[2] and Keulegan [3]. Our study will numerically investigate a number of such cases in OpenFOAM(Open-source Field Operation and Manipulation), an open source CFD Toolbox. To capture the interface between two fluids more clearly, adaptive mesh refinement will be used, which will cause the cells containing the interface to get refined after a specified number of time steps.

2 Problem Statement

The primary objective of the proposed project will be to capture the the currents produced in a gravity driven lock exchange flow. In such a flow, fluids of different densities initially at rest are separated by a vertical barrier, like the lock-gate of a dam. The barrier is removed to allow the two fluids flow in opposite directions under the influence of gravity. We will visualize the phase fraction contours showing the movement of gravity currents and aim to comment on the physics of the gravity currents produced.

We will simulate two initial conditions listed in [1],as shown in Figure 1, a full-depth release when the depths of heavy and light fluid on both sides of the gate are equal as shown in (a) and a partial-depth release when the dense fluid occupies only a fraction of the full depth as shown in (b).

We will simulate the lock exchange flow for three pairs of liquids with a wide range of density ratios, $\gamma = \rho_1/\rho_2$ equal to 1.06 (negligable density difference), 100 (high density difference) and 13593 (very high density difference).

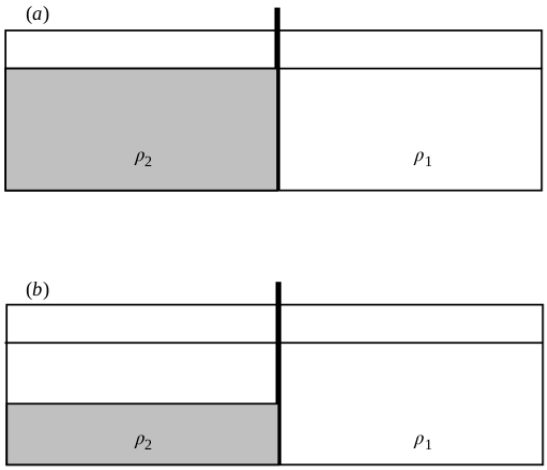

Figure 1: Diagram of the lock release initial conditions. The gate is vertically removed at t=0. The dense fluid with $\rho_1 > \rho_2$ occupies the total height of the geometry in a full-depth release (a) and a depth less than the total height (half the total height in our case) in a partial depth release (b)

3 Governing Equations

The governing equations solved by the interFoam solver in openFoam are :

$$\nabla.U = 0 \tag{1}$$

$$\rho(\frac{\partial U}{\partial t} + U.\nabla U) = -\nabla P + \nabla.\Sigma + \rho g + F_\sigma \tag{2}$$

In the momentum equations, F_σ denotes the surface tension forces.

$$\frac{\partial \alpha}{\partial t} + U.\nabla\alpha + \nabla.(\alpha(1-\alpha)u_r) = 0 \tag{3}$$

In the above equation, α is the fraction of the liquid phase in the cell for which the equations are solved and u_r refers to the velocity of the interface.[1] In the cells containing the interface, the fluid properties are modified as, $\rho = \alpha\rho_1 + (1-\alpha)\rho_2$ and $\mu = \alpha\mu_1 + (1-\alpha)\mu_2$

4 Simulation Procedure

OpenFoam needs three folders in the running directory, namely 0, constant and system. The `0` folder contains the initial and boundary conditions for the properties to be evaluated, eg. pressure, velocity etc. The `constant` folder contains the mesh information, transport properties eg. viscosity, density etc, turbulence models, if any, and other properties of a

[1]This is a modified version of the conventional Volume-of-Fluid(VoF) equation. InterFoam uses the third term on the LHS as a measure to reduce diffusion of the sharp interface, which is one of the major drawbacks of the VoF method. This term is called the interface compression term and becomes active only in the cells containing the interface. In other cells it is 0.

system which do not change. The system folder contains the files that control the simulation process, which can be altered to control time steps, domain decomposition constraints for running parallel simulations, setting custom fields in the domain eq. velocity, pressure, phase fraction etc.

4.1 Geometry and Mesh

The computational domain is a rectangular tank like system, which is essentially a 2D geometry, measuring 1 metre in x-direction, 0.5 metre in the y direction and 0.1 metre in the z direction. But it is assumed that the fluids extend infinitely in the positive and negative z-direction, and this condition is ensured by imposing suitable boundary conditions on the z-direction boundaries. Traditionally, in 2D cases, the governing equations are not solved in the direction which is not of concern to the study, but in openFoam, the adaptive mesh refinement utility refines the cells in all three directions, hence we had to keep the provision of calculation of variables in the z-direction, to allow refining in that direction too.

The domain consists of only one block, graded uniformly in all directions. It initially contains 80 cells in the x direction, 20 in the y direction and one cell in the z-direction, making it essentially a two dimensional case.

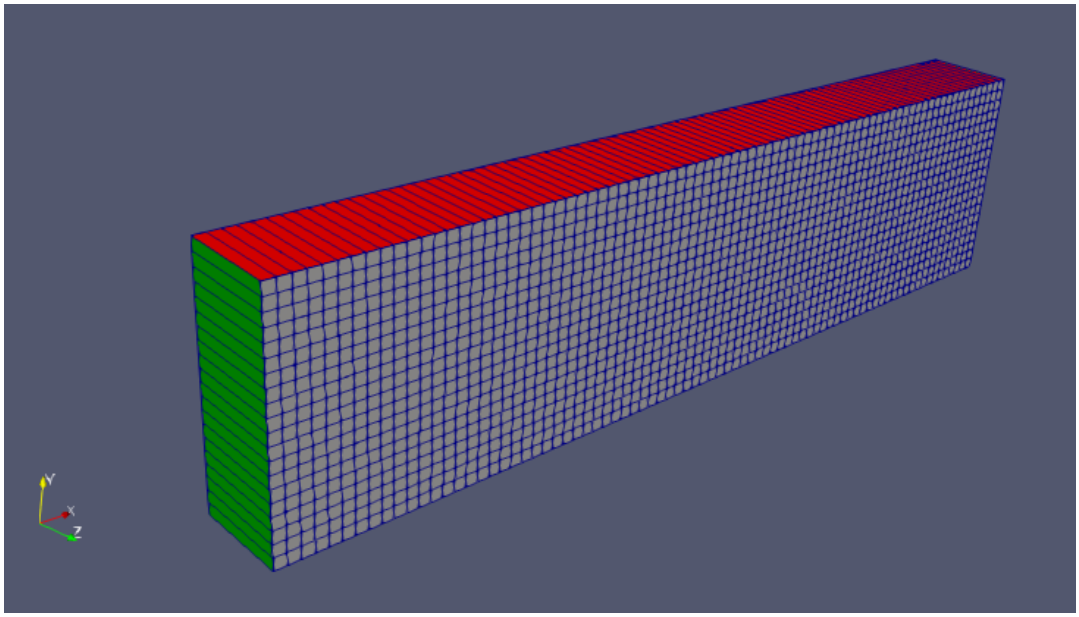

Figure 2: Three dimensional view of the computational domain

The mesh was generated using the **blockMesh** utility of openFoam. The parameters to generate the geometry was included in the **blockMeshDict** dictionary file included in the system directory. The mesh information is stored in the polyMesh directory of the constant directory.

The final mesh consists of 1600 volumes, each measuring 0.0125 x 0.0125 x 0.1 meters.

4.2 Initial and Boundary Conditions

The initial and boundary conditions for the simulation are set up by editing the alpha.water, p and U files in the 0 directory of the case folder.

- alpha.water : This file contains information about the volume fraction of one of the fluids in the cells, denoted by alpha. At the initial time, the internal region of the

domain is set as alpha.water = 0, which means the entire domain is filled with one of the fluids. We will use the setFields command to initialize the second fluid in half of the domain. This is done to to assign alpha values of 1 to the cells we desire in order to avoid the hassle of specifying the it in each cell manually in the alpha.water file. On the walls and floors of the domain, conditions are set as zeroGradient, which is a neumann boundary condition and it specifies the gradient of alpha on the walls equal to zero. Physically, it implies that a static contact angle of 90^o is set on the walls. On the front and back walls, a the symmetry boundary condition is used, to ensure the condition of the fluids extending in the z-direction.

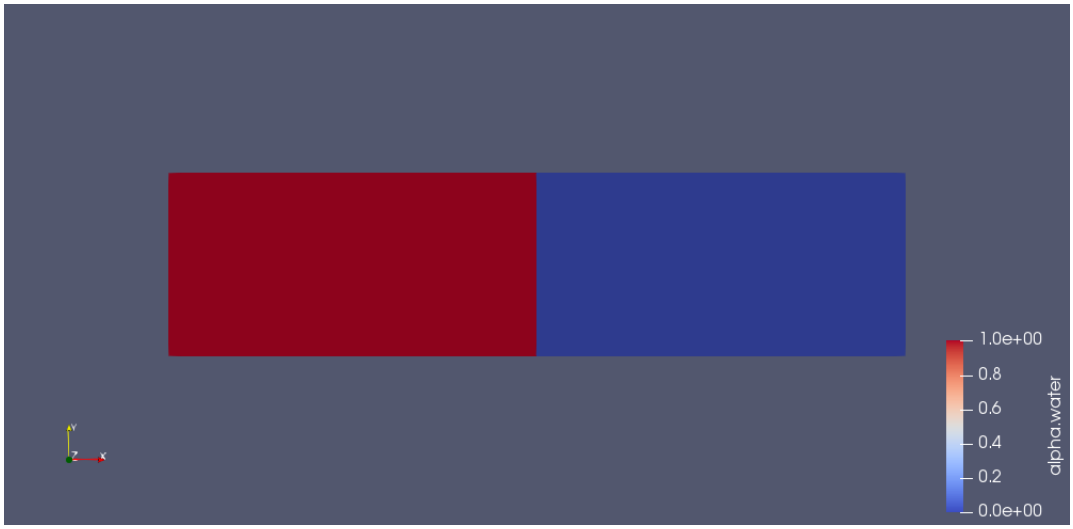

Figure 3: The two fluids at initial time for full depth lock release

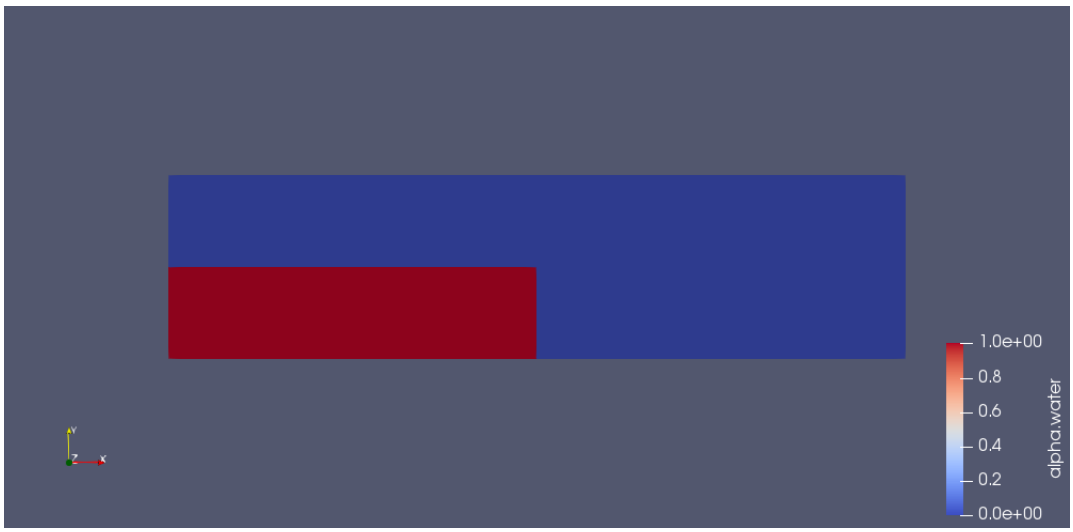

Figure 4: The two fluids at initial time for partial depth lock release

- p_rgh : This file contains information about the values of hydrostatic pressure in the cells, denoted by p_rgh. At the initial time, the internal region of the domain is set to zero. The side and bottom walls have a fixedFluxPressure condition used with value of zero which sets t he pressure gradient on the wall as the provided value. The top wall has a fixed pressure of zero. These boundary conditions for the walls were adopted from the damBreak tutorial case provided in openFoam, since our case and

the damBreak case have similar boundaries. The front and back planes in z-direction has a symmetry boundary condition, as used for alpha.water.

- U : This file contains information about the values of velocity in the cells, denoted by U. At the initial time, the velocity field in the internal region of the domain is set to zero. On the walls, on which the pressure is specified, pressureInletOutletVelocity with a value of zero is used. This condition sets the velocity gradient to zero for flow out of the domain, which is equivalent to a no slip boundary condition, and for flow into the domain, the normal component of the velocity in the cell adjacent to the boundary is used. On the other walls the noSlip condition is used which makes the velocity at the wall equal to zero. The front and back planes in z-direction has a symmetry boundary condition, as used for alpha.water.

4.3 Constant Properties of the system

The constant properties of the system ie. the mesh information, transport properties (ρ, ν etc.), gravitational acceleration, turbulence properties etc. will be specified by editing the transportProperties, turbulenceProperties and **g** files in the constant directory of the case folder.

- transportProperties : This file contains information about the transport properties of the system ie. the density(ρ), viscosity(ν) of air and water, and surface tension between the two phases. The Newtonian model is used for both fluids and the fluid properties at room temperature is used. For the first case, considering saline water(35g salt/kg water) and freshwater, the transport properties of saline water was collected from http://web.mit.edu/seawater/ and was originally taken from [4] and [5]. The interfacial tension between the two phases is considered negligible since they are very similar in composition.

  Rest of the transport properties and surface tension values have been collected from wikipedia.[2]

- g : This file contains information about the value and direction of gravitational acceleration. We have set this value to 9.81 m/s^2 in the negative y direction.

- turbulenceProperties : This file contains information about the turbulence properties of the system. We will not take into account the turbulence and will set the turbulence model to laminar.

- dynamicMeshDict : This file contains information about the dynamic mesh control. The dynamicRefineFvMesh type of dynamic mesh is used. The mesh is refined every ten time steps, based on the alpha.water field. The cells with a value of alpha.water between 0.01 and 0.99 is refined and other cells are unrefined. Maximum level of refinement is set to 1.

- polyMesh : the polyMesh folder contains the mesh information specified earlier. We do not need to alter it.

[2]For the second test case, the transport properties of one of the fluids is taken the same as water and other is an imaginary fluid, and the density of the imaginary fluid has been taken as $10Kg/m^3$, for the test purposes. The viscosity was taken the same as water and surface tension the same as a water air interface

4.4 Setting the runtime conditions and output control

The simulation controls like starting and end time, discretization, interpolation and solution schemes, and other parameters can be controlled by altering the controlDict, fvSchemes and fvSolution files in the system directory of the case folder.

- controlDict : This file contains the runtime conditions like starting and ending time, time step size, maximum courant number allowed, and is responsible for running the simulation. The running time is taken as 5 seconds with an initial time step of 0.01, so that the cell courant number[3] do not exceed the value specified by the user[4]. The adjustableRunTime is used so that in case the courant number exceeds the specified value, the time step is changed accordingly to bring the value below the maximum courant number. The format in which the data is written to the result files and precision with which it writes can also be controlled by altering this file. We will use the default inputs for these and will not change them.

- fvSchemes : This file can be altered to set the discretization and interpolation schemes for each of the terms in the governing equations, that the solver will use during the solution. We will use the eulerian scheme for time discretization and for discretization of the gradient and laplacian terms we will use the default settings set in the damBreak tutorial of openFoam.

- fvSolution : This file can be altered to set the solution schemes for the solver, tolerance and solution algorithms. For this case also we will use the default settings for the damBreak tutorial in openFoam.

- setFieldsDict : This dictionary file is altered to choose how the initial volume fraction field will be set. We will use the boxToCell command to initialize one of the fluids in half of the domain.

4.5 Solver

InterFoam is a transient solver that solves the Navier Stokes equations for two incompressible and immiscible fluids, with another marker function(taken as volume fraction here, α) equation, typical to the Volume-of-Fluid method utilized by interFoam to track the interface. We consider the fluids as newtonian with no phase change, under isothermal condition and surface tension is constant and uniform at the interface between the two fluids.

After setting up the dictionaries, boundary conditions and runtime conditions, the terminal was used to navigate to the case directory and create the mesh using *blockMesh* command, the two fluids at initial time were set using the *setFields* command and the final simulations were run using the *interFoam* command.

All the simulations were run on a DELL INSPIRON 5555, using OpenFOAM 7.0 in the Ubuntu 18.04 environment. The postprocessing was done in paraview 5.6.0, which was installed as a dependency with OpenFoam 7.0

[3]Courant number is defined as $\frac{\Delta t|U|}{\Delta x}$ where Δt is the time step, $|U|$ is the magnitude of velocity in the cell and Δx is the grid size

[4]Generally taking the courant number < 1 is a satisfactory condition for stability of the numerical methods.

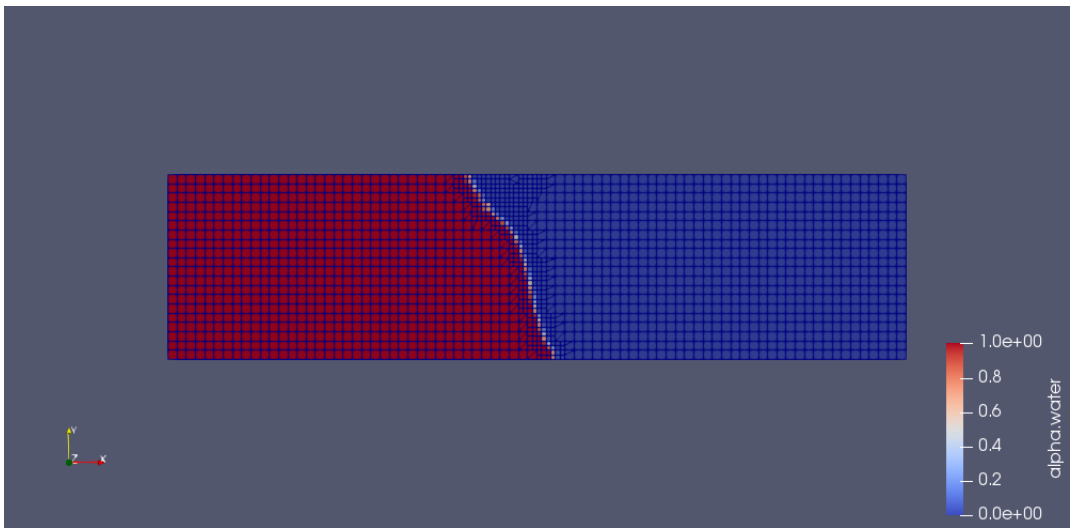

Figure 5: A plot of the phase fraction contours at time = 0.5 seconds and the selectively refined mesh in the interface region

5 Results and Discussions

5.1 Full lock release conditions

5.1.1 $\gamma = 1.06$

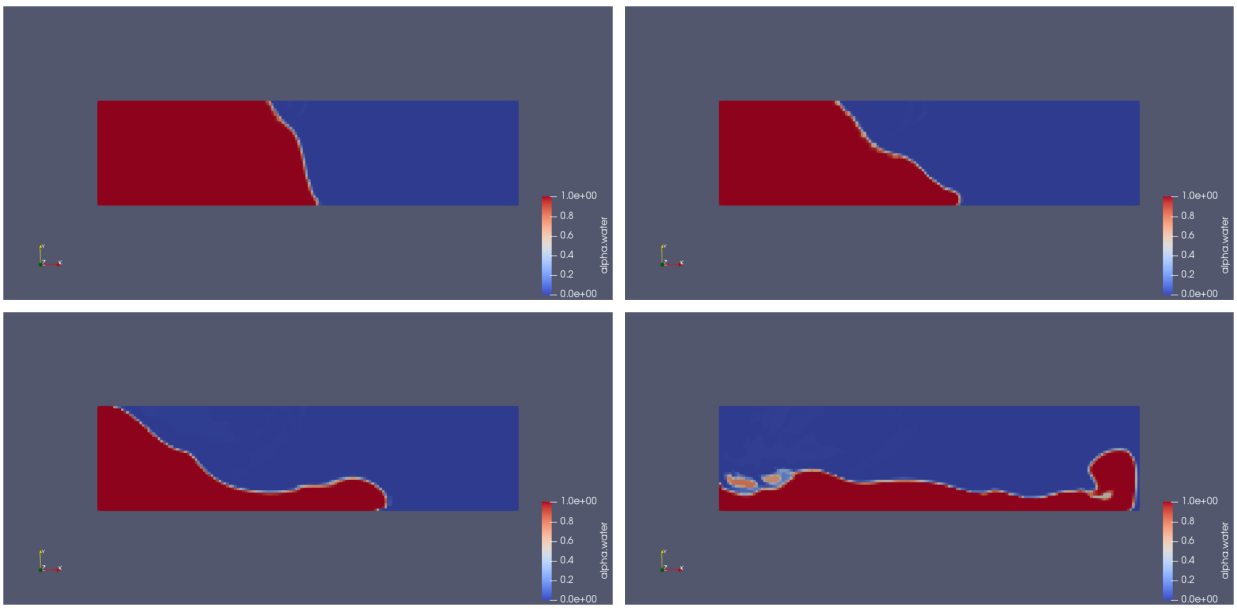

Figure 6: The volume fractions of the two fluids at time = 0.5, 1, 2, 5 seconds for full depth lock release

We can see that the denser fluid(Saline water) flows along the bottom of the tank and fresh water moves along the top. But the speed of the moving fronts are not equal, as we can see from the positions of the fronts at different time intervals. This can be attributed to the fact that the viscosities of the two liquids are different, hence one of them flow more easily than the other. The heavier fluid moves along the floor of the domain and strikes against the wall, and then the front moves throughout the domain hitting against the side walls, like an oscillation on a string, which is known as sloshing. Eventually the height of

the moving wavefront decreases and the sloshing diminishes at 99 seconds, as seen in figure 7. It can be assumed that steady state is reached after a very long time and the lighter fluid rests above the heavier one.[5]

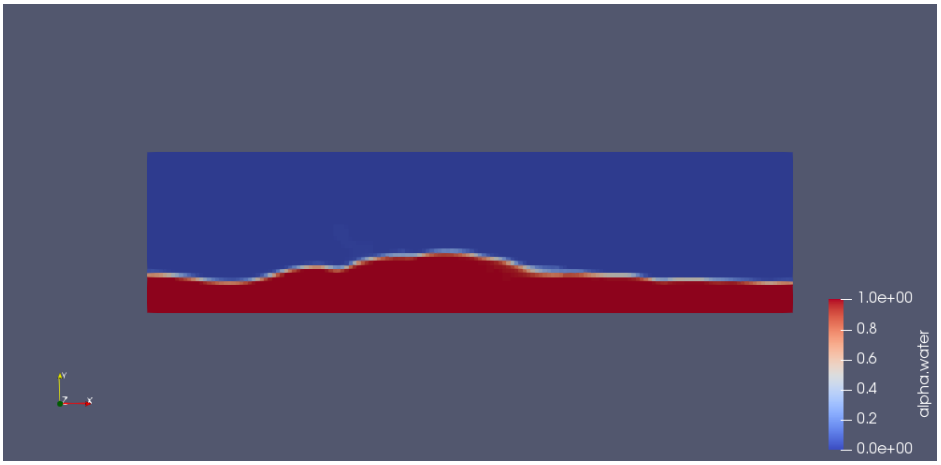

Figure 7: The volume fractions of the two fluids at time = 99 seconds

5.1.2 $\gamma = 100$

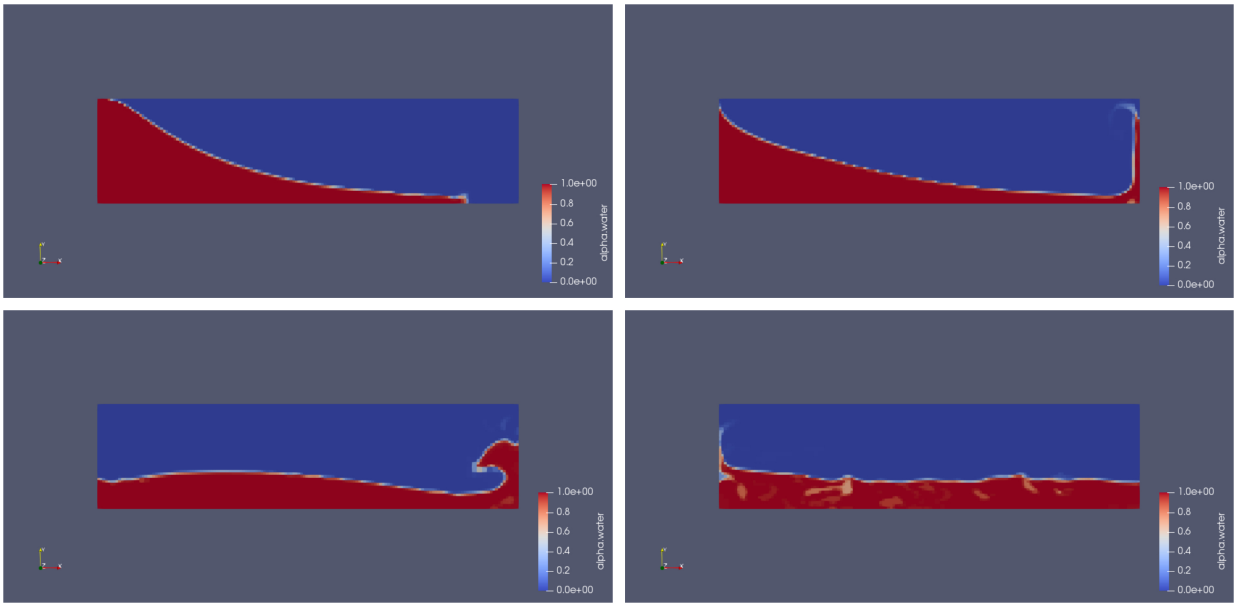

Figure 8: The volume fractions of the two fluids at time = 0.3, 0.5, 0.75, 4 seconds for full depth lock release

In figure 8, similar to the previous case, the denser fluid($\rho = 100$) flows along the bottom of the tank and the lighter fluid($\rho = 1$) moves along the top boundary of the tank. In this case the fronts move much faster than the previous case. This is because the difference in hydrostatic pressure between the two fluids is much higher in this case, hence the front velocities are also more. In this case, both the fronts move at almost similar speeds due to similar viscosities. The heavier fluid front moves much faster than the previous case and hits the wall at 0.5 seconds, while in the previous case, it took 5 seconds. The sloshing

[5]In a real experiment the two fluids would mix with each since they are miscible liquids, but our solver considers them immiscible so they form two seperate layers in the end and not a uniform solution

phenomena persists till 4 seconds and as visible in figure 9, at time = 7 seconds, sloshing almost subsides.

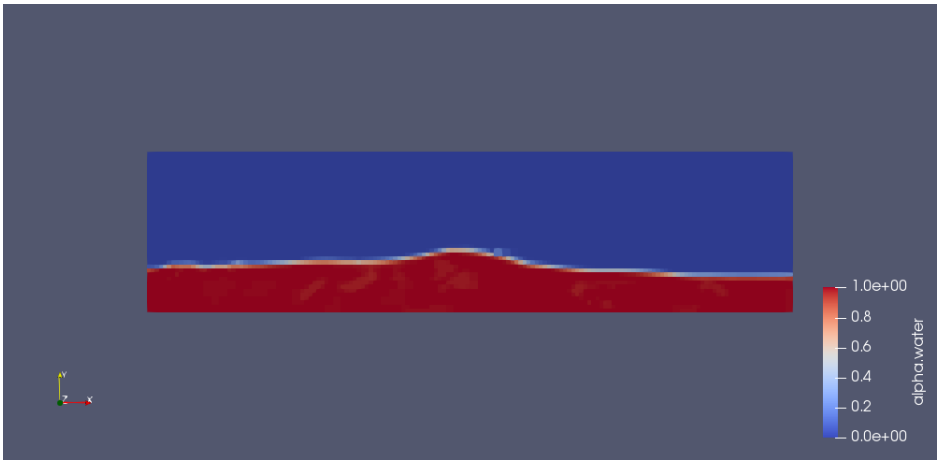

Figure 9: The volume fractions of the two fluids at time = 7 seconds

5.1.3 $\gamma = 13593$ (Air:Mercury)

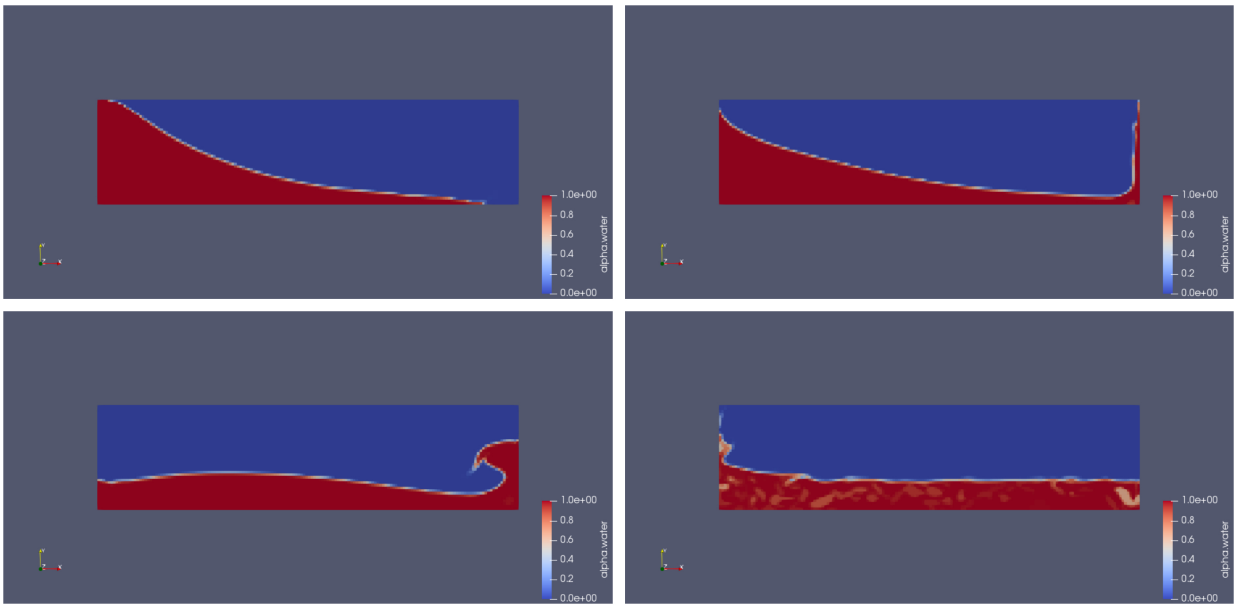

Figure 10: The volume fractions of the two fluids at time = 0.3, 0.5, 0.75, 4 seconds for full depth lock release

The volume fraction contours of this case is very similar to the previous case and the much larger density ratio of the fluids does not cause a very large change in the velocities of the two fluid fronts. In this case the fronts move slightly faster than the previous case. In this case, the fonts move at almost similar speeds due to similar viscosities. The heavier fluid front moves slightly faster than the previous case and hits the wall at 0.35 seconds, while in the previous case, it took 0.5 seconds. But in this case, the sloshing persists for a much longer time and as seen in figure 11, at time = 22 seconds, sloshing almost subsides.

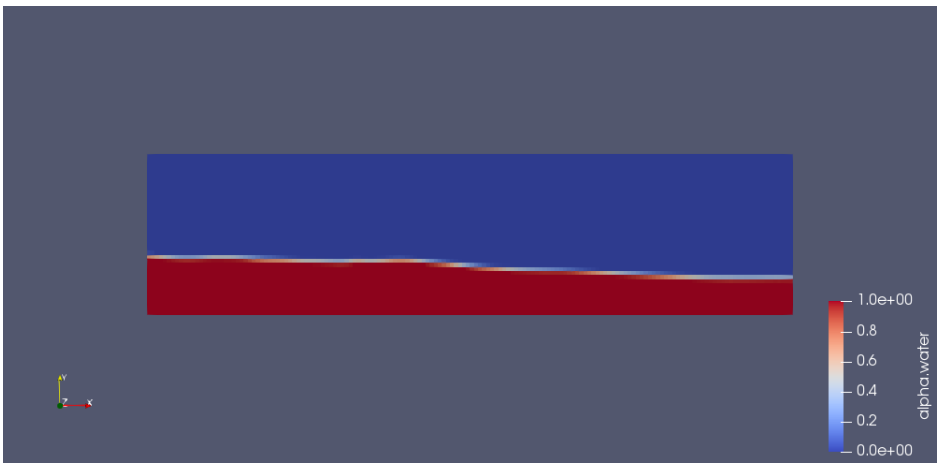

Figure 11: The volume fractions of the two fluids at time = 22 seconds

5.2 Partial lock release conditions

In case of the partial lock release conditions, the gravity currents follow trends similar to the full lock release conditions. We can see in figure 12 that the velocity of the denser fluid fronts follow a trend similar to the full depth lock release cases, with the saline-freshwater case with the least velocity and the air-mercury front having the maximum velocity. The nature of the gravity currents are similar to that of the full depth cases and their flow physics are similar to the previous cases.

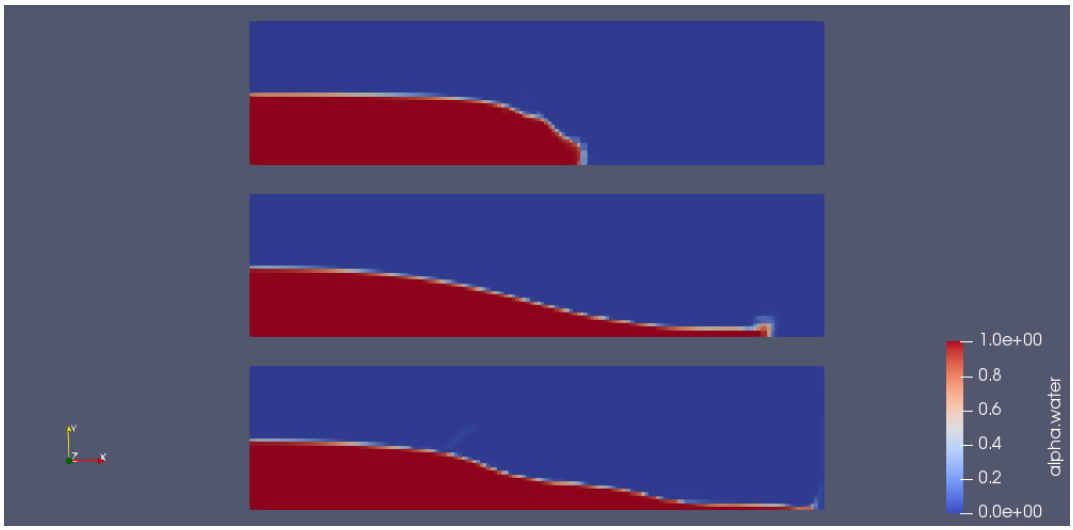

Figure 12: The volume fractions of saline-freshwater pair at time = 1 second and the pairs of fluids with $\gamma = 100$ and 13593 at time = 0.3 seconds for partial depth lock release

6 Conclusions

From the series of simulations run involving lock exchange gravity driven flows, it is seen that the flow physics is affected by the density ratio and viscosities of the fluids. For greater density ratios, the speed of the moving fluid fronts will be greater but the change in front velocities with density ratios is not uniform. For the least density ratio, flow is the slowest and sloshing persists for the maximum time. For 1:100 density ratio the front velocity is much higher and sloshing persists for much lesser time. For the maximum density ratio,

the front velocities are slightly higher than the previous case but the sloshing phenomena lasts longer, which can be caused by the greater density of mercury, causing it to have more mass than the same volume of another fluid with 100 times less density, and it takes more time to come to rest. In the first case, this effect is counterbalanced by the low density difference, causing the flow to occur much slowly, and the sloshing persists much longer, due to the slow movement of the front.

For future work, a different solver can be used for the first case which can work for miscible liquids like saline and fresh water, which can give more accurate results for the mixing phenomena occurring in case of an actual dam breaking or mixing of the two types of water at the mouth of estuaries.

References

[1] JO Shin, SB Dalziel, and PF Linden. "Gravity currents produced by lock exchange". In: *Journal of Fluid Mechanics* 521 (2004), pp. 1–34.

[2] M.P O'Brien and J Cherno. "Model law for motion of salt water through fresh". In: *Trans. Am. Soc. Civil Engrs* 99 (1934), pp. 576–94.

[3] Garbis Hovannes Keulegan. *Thirteenth progress report on model laws for density currents: an experimental study of the motion of saline water from locks into fresh water channels*. Vol. 5168. US Department of Commerce, National Bureau of Standards, 1957.

[4] M. H. Sharqawy, J. H. Lienhard V, and S. M. Zubair. "Thermophysical properties of seawater: A review of existing correlations and data". In: *Desalination and Water Treatment* 16.1-3 (Apr. 2010), pp. 354–380.

[5] K. G. Nayar et al. "Thermophysical properties of seawater: A review and new correlations that include pressure dependence". In: *Desalination* 387 (July 2016), pp. 1–24.