PHY407 (Lab 7: Partial Differential equations, Pt. I)

Utkarsh Mali[1] and Aslesha Pokhrel[1,2]

[1]Department of Physics, University of Toronto
[2]Department of Computer Science, University of Toronto

November 6, 2020

Contents

Work Allocation:
Utkarsh: Question 2
Aslesha: Question 1, Question 3

Question 1: Electrostatics and Laplace's equation

(a) In this section we calculate the electrostatic potential for the given capacitor using the Gauss-Seidel method without overrelaxation and plot the contour plot of the potential and stream plot of the electric field lines.

The contour plot of the potential is given below:

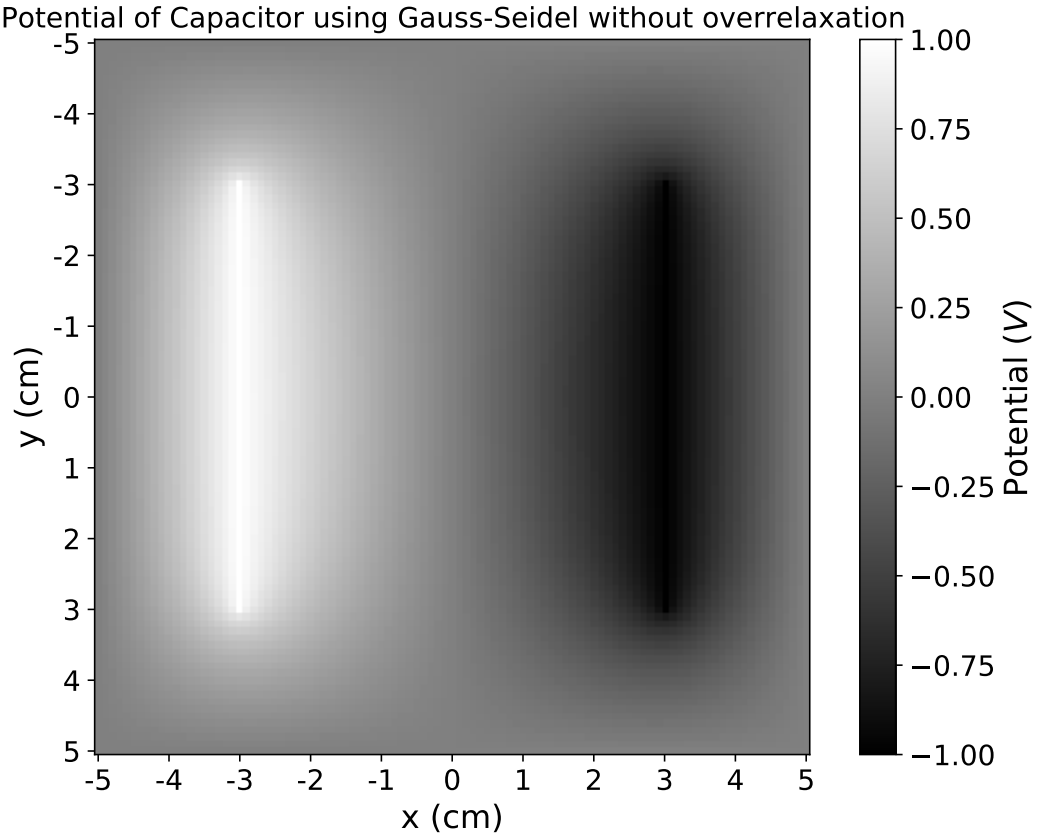

Figure 1: The contour plot of the electrostatic potential using the Gauss-Seidel method without overrelaxation.

The stream plot of the electric field lines is given below:

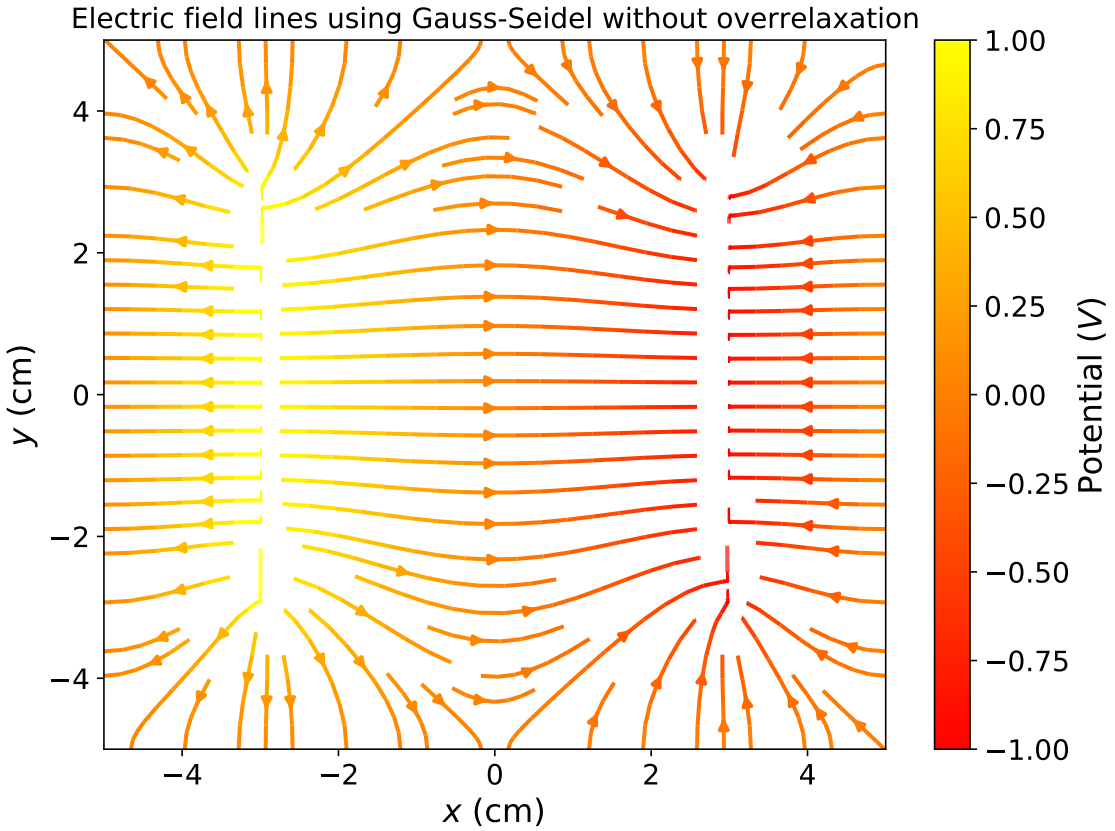

Figure 2: The stream plot of the electric field lines using the Gauss-Seidel method without overrelaxation, colour-coded by the value of the electric potential.

(b) Now, we implement the same method with overrelaxation using $\omega = 0.1$ and $\omega = 0.5$. There was no visible difference in the contour plot of the potential (compare fig. 1, fig. 3 and fig. 4) as well as the stream plot of the electric field lines (compare fig. 2, fig. 5 and fig. 6, however, we noticed the change in run-time of the algorithm. The Gauss-Seidel with overrelaxation with $\omega = 0.5$ had the smallest run-time while the Gauss-Seidel without overrelaxation had the largest run-time as expected. The output of the code showing the run-time of the algorithm in each case is as follows:

```
OUTPUT:

The time taken to calculate the electrostatic potential using Gauss-Seidel without
    overrelaxation 37.08s
The time taken to calculate the electrostatic potential using Gauss-Seidel with
    overrelaxation with omega=0.1 is 30.86s
The time taken to calculate the electrostatic potential using Gauss-Seidel with
    overrelaxation with omega=0.5 is 16.98s
```

The contour plots of the electrostatic potential using the Gauss-Seidel method with overrelaxation are visualized below:

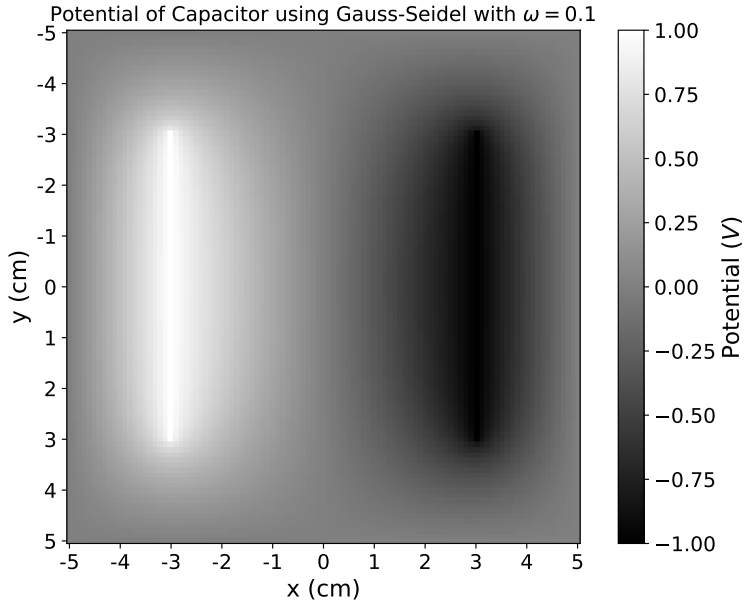

Figure 3: The contour plot of the electrostatic potential using the Gauss-Seidel method with overrelaxation with $\omega = 0.1$.

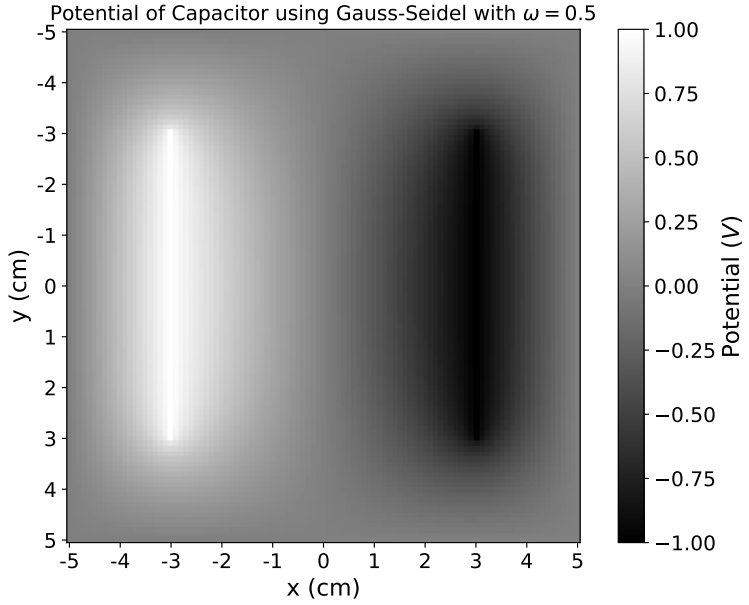

Figure 4: The contour plot of the electrostatic potential using the Gauss-Seidel method with overrelaxation with $\omega = 0.5$.

The stream plots of the electric field lines using the Gauss-Seidel method with overrelaxation are visualized below:

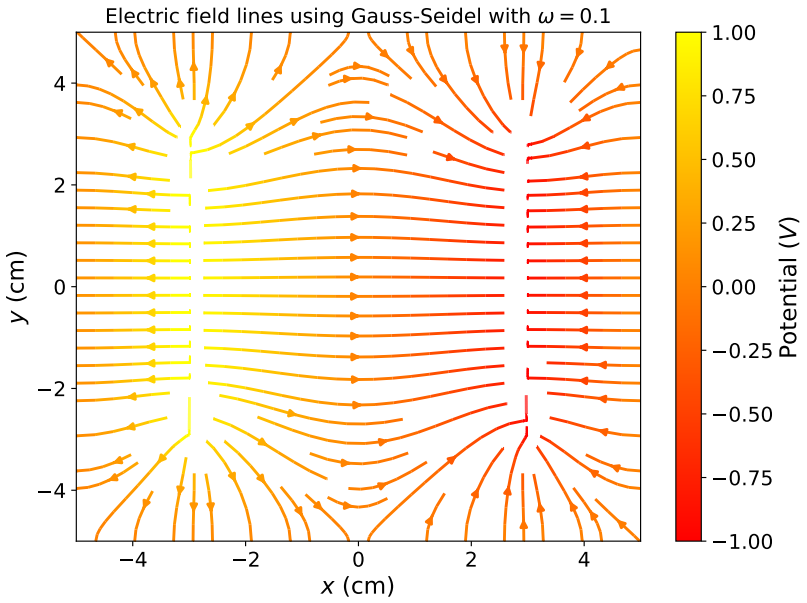

Figure 5: The stream plot of the electric field lines using the Gauss-Seidel method with overrelaxation with $\omega = 0.1$.

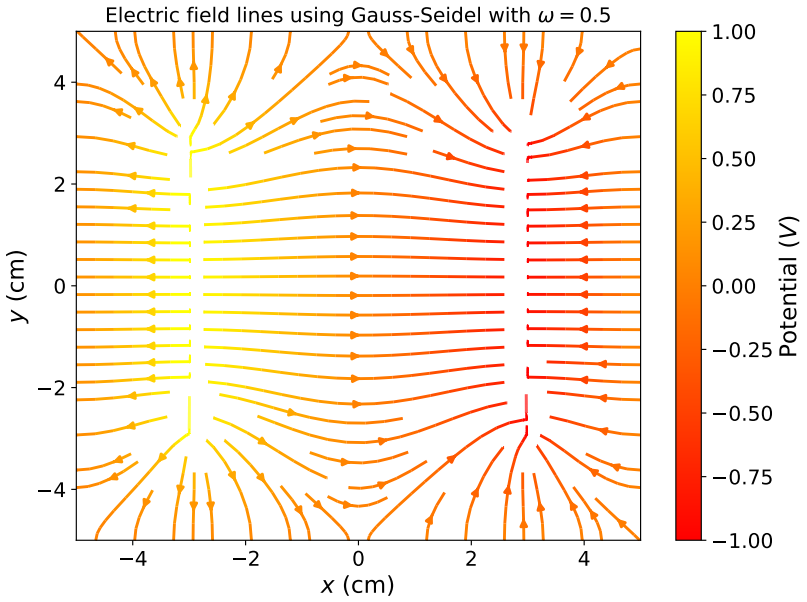

Figure 6: The stream plot of the electric field lines using the Gauss-Seidel method with overrelaxation with $\omega = 0.5$.

Question 2: Simulating the shallow water system, Part I

(a) Write the derivation of equation (7) in the lab handout. We have our initial given equations

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -g\frac{\partial \eta}{\partial x}$$

This can be rewritten as:

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x}\left(\frac{1}{2}u^2 + g\eta\right)$$

Now, our other equation:

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}(uh) = 0$$

This is equivalently:

$$\frac{\partial \eta}{\partial t} = -\frac{\partial}{\partial x}(uh) = -\frac{\partial}{\partial x}u(\eta - \eta_b)$$

Now we choose:

$$\vec{u} = \begin{bmatrix} u \\ \eta \end{bmatrix}$$

Then as a result:

$$\vec{F} = \begin{bmatrix} \frac{1}{2}u^2 + g\eta \\ u(\eta - \eta_b) \end{bmatrix}$$

As needed.
Now we continue to show the second part of the formula. We know that:

$$\vec{F}^n = \left(\frac{1}{2}(u^2)^n + g\eta^n, u^n(\eta^n - \eta_b^n)\right)$$

We will use this to find out formula's for $\vec{F}^n_{1,2,j\pm1}$

$$\vec{F}^n_{1,j+1} = \left(\frac{1}{2}(u^2_{j+1})^n + g\eta^n_{j+1}\right)$$

$$\vec{F}^n_{2,j+1} = \left(u^n_{j+1}(\eta^n_{j+1} - \eta^n_{j+1})\right)$$

$$\vec{F}^n_{1,j-1} = \left(\frac{1}{2}(u^2_{j-1})^n + g\eta^n_{j-1}\right)$$

$$\vec{F}^n_{2,j-1} = \left(u^n_{j-1}(\eta^n_{j-1} - \eta^n_{b\,j-1})\right)$$

We now use these given formula's in our equation for u^{n+1}_j and η^{n+1}_j. We use equation (5) in the lab handout.

$$u^{n+1}_j = u^n_j - \frac{\Delta t}{2\Delta x}\left(\frac{1}{2}u^{n^2}_{j+1} + g\eta^n_{j+1} - \frac{1}{2}u^{n^2}_{j-1} - g\eta^n_{j-1}\right)$$

$$\eta^{n+1}_j = \eta^n_j - \frac{\Delta t}{2\Delta x}\left(u^n_{j+1}(\eta^n_{j+1} - \eta^n_{b,j+1}) - u^n_{j-1}(\eta^n_{j-1} - \eta^n_{b\,j-1})\right)$$

These are the equations which we will use when coding our partial differential equation solver.

(b) Here we are asked to formulate the code computed in part a of the question and run it to plot the graph of the waves. We wrote the code as required and set the initial conditions as previously defined in the question.

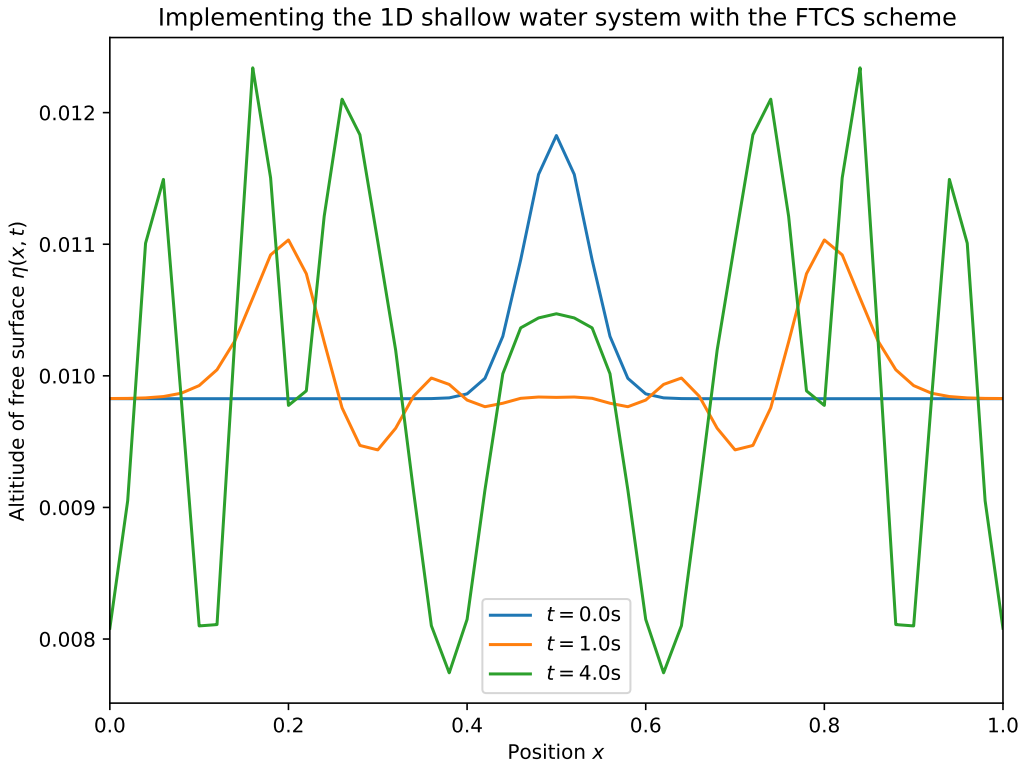

Figure 7: We can see the wave starting off at the center at $t = 0$, as we traverse through time we see the wave starting to disperse and hit the boundary. Once we have $t = 4$ we start to see the wave start to diverge.

(c) Submit your von Neumann stability analysis of the system. We begin with the system of equations that we will be using, analogous to Newman's example on pg429.

$$\begin{bmatrix} u(x,t) \\ \eta(x,t) \end{bmatrix} = \begin{bmatrix} c_u(t) \\ c_\eta(t) \end{bmatrix} e^{ikx}$$

We can add a dx element, analogous to a in Newman's system.

$$\begin{bmatrix} u(x+dx,t) \\ \eta(x+dx,t) \end{bmatrix} = \begin{bmatrix} c_u(t) \\ c_\eta(t) \end{bmatrix} e^{ik(x+dx)}$$

We now calculate our time steps.

$$c_u(t+dt)\cancel{e^{ikx}} = c_u(t)\cancel{e^{ikx}} + dtc_n\cancel{e^{ikx}}$$

$$c_u(t+dt) = c_u(t) + dtc_n$$

Similarly we compute the one. And simplify:

$$c_n(t+dt) = c_n(t)e^{ikx} + \frac{dtv^2}{(dx)^2}\left(c_u(t)e^{ik(x+dx)} + c_u(t)e^{ik(x-dx)} - 2c_u(t)e^{ikx}\right)$$

$$c_n(t+dt) = c_n(t) + \frac{dtv^2}{(dx)^2}\left(c_u(t)e^{ikdx} + c_u(t)e^{-ikdx)} - 2c_u(t)\right)$$

$$c_n(t+dt) = c_n(t) + \frac{dtv^2}{(dx)^2}\left(c_u(t)(e^{ikdx} + e^{-ikdx)} - 2)\right)$$

Notice that using euler's formula we can simplify further:

$$(e^{ikdx} + e^{-ikdx)} - 2) = \sin^2(kdx)$$

so

$$\vec{c}(t+dt) = \begin{bmatrix} c_u(t+dt) \\ c_n(t+dt) \end{bmatrix} = \begin{bmatrix} 1 & dt \\ \frac{dtv^2}{(dx)^2}\sin^2(kdx) & 1 \end{bmatrix} \begin{bmatrix} c_u(t) \\ c_n(t) \end{bmatrix}$$

We now take the determinant of this and find the eigenvalues.

$$\det\begin{bmatrix} 1-\lambda & dt \\ \frac{dtv^2}{(dx)^2}\sin^2(kdx) & 1-\lambda \end{bmatrix} = \frac{(dt)^2v^2}{(dx)^2}\sin^2(kdx) + (1-\lambda)^2 = 0$$

Let $dt \to \Delta t$ and $dx \to \Delta x$

$$\frac{\Delta t^2 v^2}{\Delta x^2}\sin^2(kdx) + (1-\lambda)^2 = 0$$

$$\frac{\Delta t^2 v^2}{\Delta x^2}\sin^2(k\Delta x) = (1-\lambda)(1+\lambda) = \lambda^2 - 1$$

$$\sqrt{\frac{\Delta t^2 v^2}{\Delta x^2}\sin^2(k\Delta x) + 1} = |\lambda|$$

Now notice from the lab handout that $v^2 = gH$.Then we have the equation we need:

$$\sqrt{\left(\frac{\Delta t}{\Delta x}\right)^2 gH\sin^2(k\Delta x) + 1} = |\lambda|$$

We do not expect the FTCS to be stable. We notice that when the eigenvalue is greater than one we expect the system to diverge. And we can see that the equation above is greater than one since $\left(\frac{\Delta t}{\Delta x}\right)^2 gH\sin^2(k\Delta x) > 1$. As a result of this the FTCS method will never be stable.

Question 3: Simulating the shallow water system, Part II

The code for this section is given in two files `lab08_Q3.py` and `lab08_Q3_broken_axes.py`. One of the files `lab08_Q3.py` will run as it is and produce fig. 8 and fig. 9 while `lab08_Q3_broken_axes.py` will produce fig. 8 and fig. 10. However, you must run **pip install brokenaxes** in the command prompt before running this file. We used brokenaxes module to produce fig. 10 because it was very cumbersome to produce a similar plot using matplotlib only and fig. 10 is not really a requirement as we also show fig. 9.

(a) Now, we implement the Two-Step Lax-Wendroff scheme and run the simulation using the same setup as in Q2(b). The simulation at $t = 0, 1, 4$ s is visualized in the plot below:

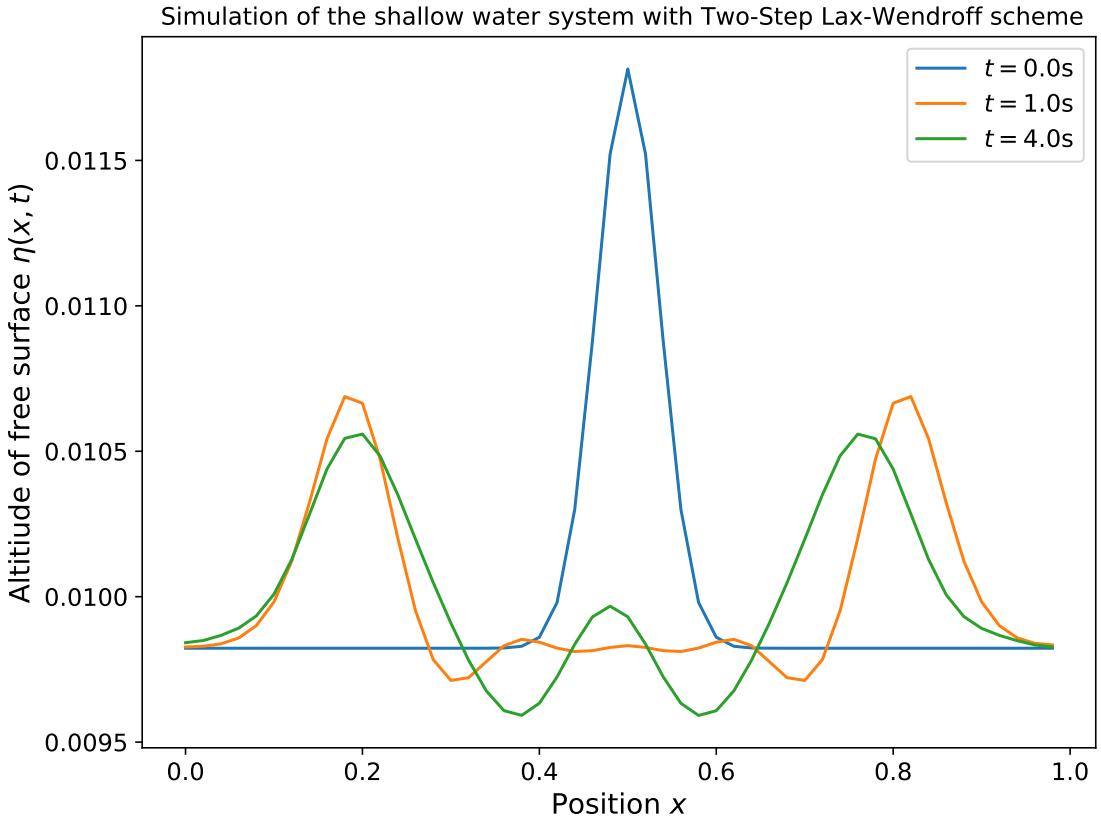

Figure 8: The simulation of the shallow water system with Two-Step Lax-Wendroff scheme at $t = 0, 1, 4$ s.

The run-time for this scheme was slightly higher compared to the FTCS scheme. However, as we can see from fig. 8, using Two-Step Lax-Wendroff scheme leads to a improved stability and accuracy compared to the FTCS scheme shown in fig. 7.

(b) Here, we allow for variable bottom topography and simulate a 1D tsunami using the setup given in the handout. The 1D tsunami simulation showing the altitude of the free surface is visualized in fig. 9.

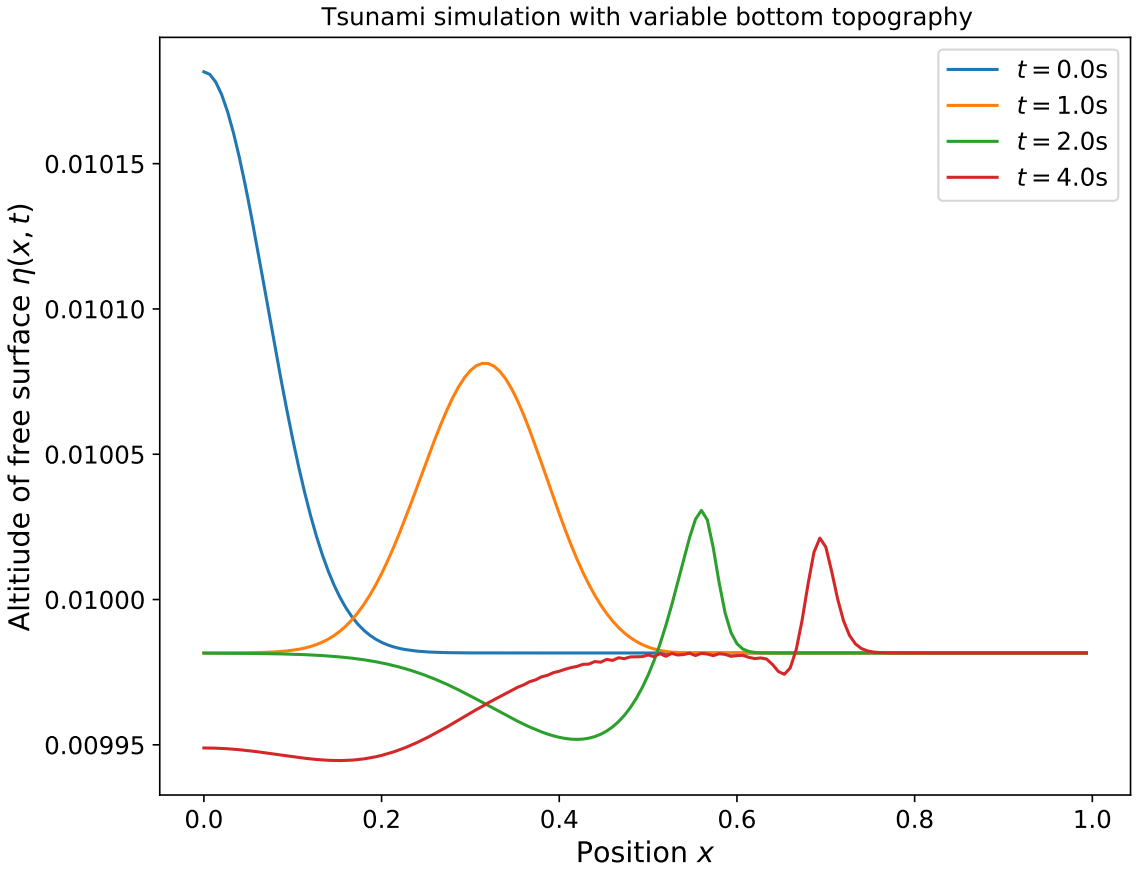

Figure 9: The plot showing the altitude of free surface for 1D tsunami simulation with variable topography.

The 1D tsunami simulation showing the altitude of the free surface along with the bottom topography on the same figure is visualized in fig. 10.

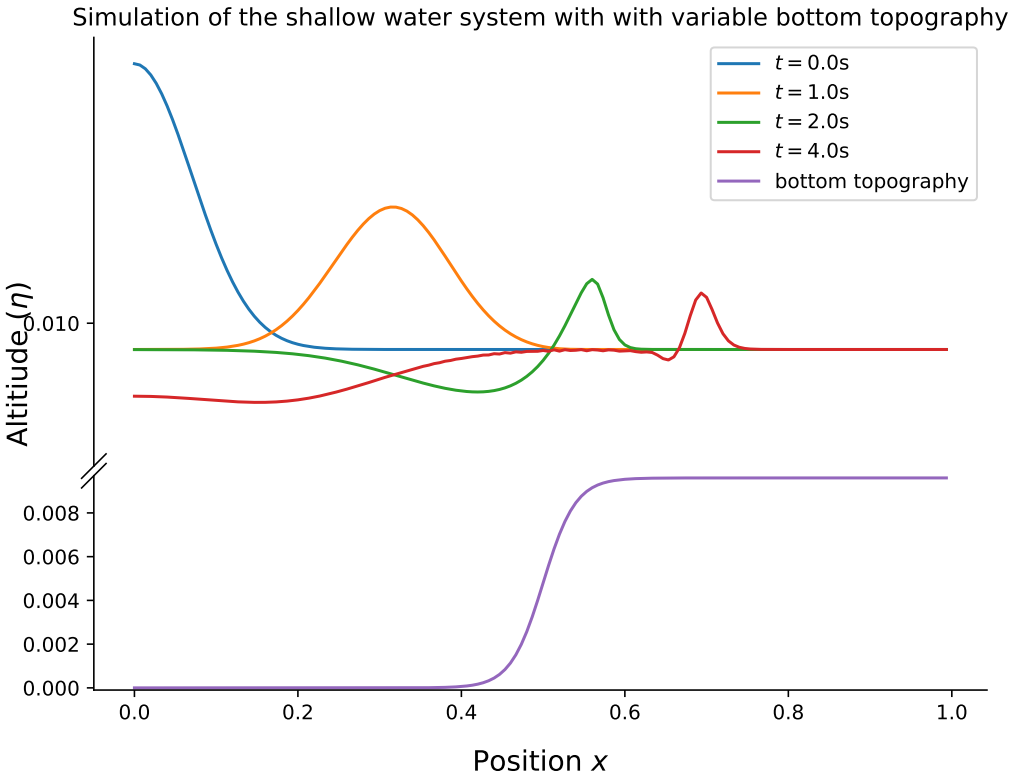

Figure 10: The plot with broken y-axis showing the altitude of free surface and variable topography for 1D tsunami simulation with variable topography.

As shown in fig. 9 and fig. 10, a shallow wide bump starts at the left of the domain at $t = 0s$ and moves to the right towards the continental shelf. As the waveform moves to the right, the height of the waveform gets smaller. Similarly, in the shallower water there seems to be a reflection or downward bump as seen for $t = 2$s and $t = 4$s which appears to be moving to the left. Likewise, the speed of the waveform decreases when it reaches shallower water as the speed is given by $\sqrt{gH}$ and H, the water column height, is smaller in shallower part,