PHY407 Lab 08

Sang Bum Yi, 1004597714
Jianbang Lin, 1004970720

The workload was distributed as followings:

- **Jianbang Lin did question 2 and 3**
- **Sang Bum Yi did question 1**

Question 1

Part a)

A simple model of an electronic capacitor consisting of two flat metal plates enclosed in a square metal box was investigated using Gauss-Seidel method without overrelaxation. The two flat metal plates had the potential of 1V and -1V each. The square metal box had a dimension of 10cm x 10cm and plotted on a grid of 100 x 100 points, whose walls were at zero voltage. The electric potential was calculated at each grid point with a target accuracy of 10^{-6} volts.

The main difference between Jacobi method and the Gauss-Seidel method is that the latter does not require an additional array to hold the values that are newly computed, whereas the former does. Instead, the Gauss-Seidel method stores the new values in the same array as the old values, using the equation 1.1 below. Note that 'a' refers to the spacing between grid points and therefore is 0.1 cm for this configuration. The left-pointing arrow is a symbol that represents an overwriting of the same array.

$$\phi(x,y) \leftarrow \frac{1}{4}[\phi(x+a,y) + \phi(x-a,y) + \phi(x,y+a) + \phi(x,y-a)] \qquad \text{[Eq 1.1]}$$

Using the computed values of the electric potential, the density plot and contour plot are plotted as shown in Figure 1.1 and Figure 1.2. Note that the electric potential on each metal plate remains the same as they are constant values, whereas the potential increases and decreases as it gets closer to the left metal plate with 1V potential and the right metal plate with -1V potential, respectively.

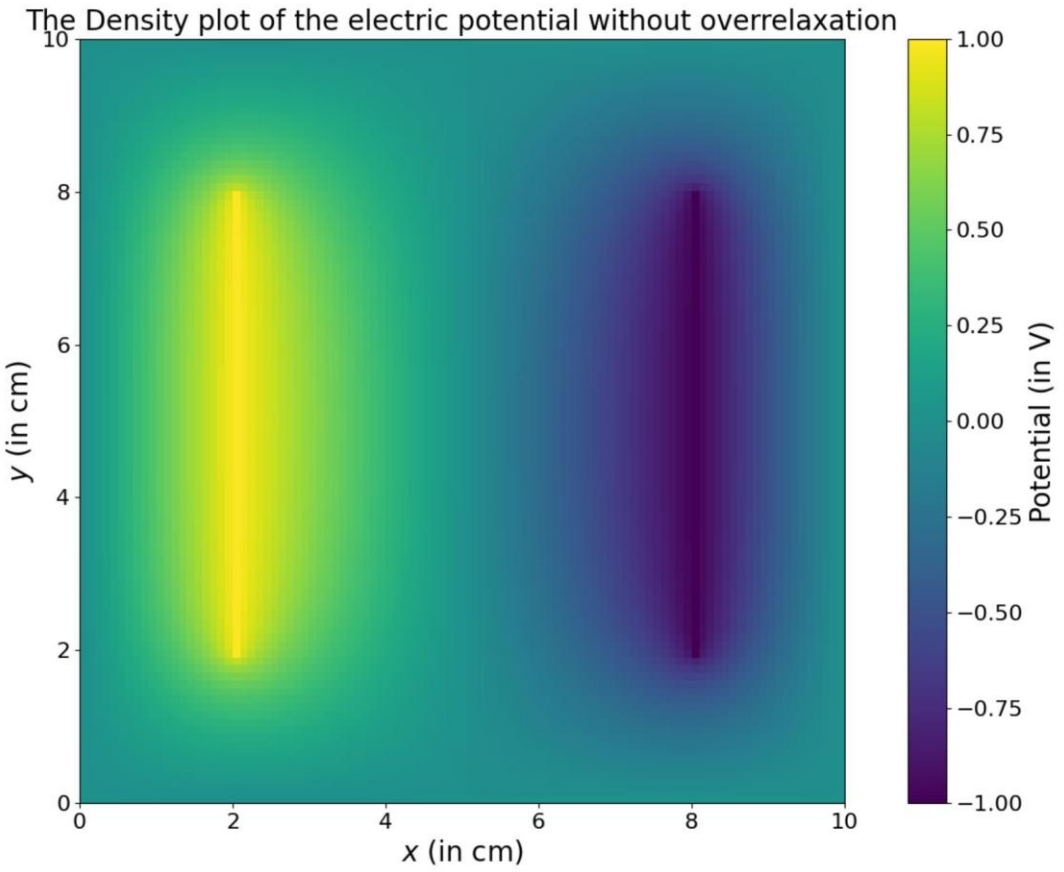


Figure 1.1: The Density plot of the electric potential on a grid of 100 x 100 points. The electric potential is the highest and lowest on the left metal plate and right metal plate respectively.

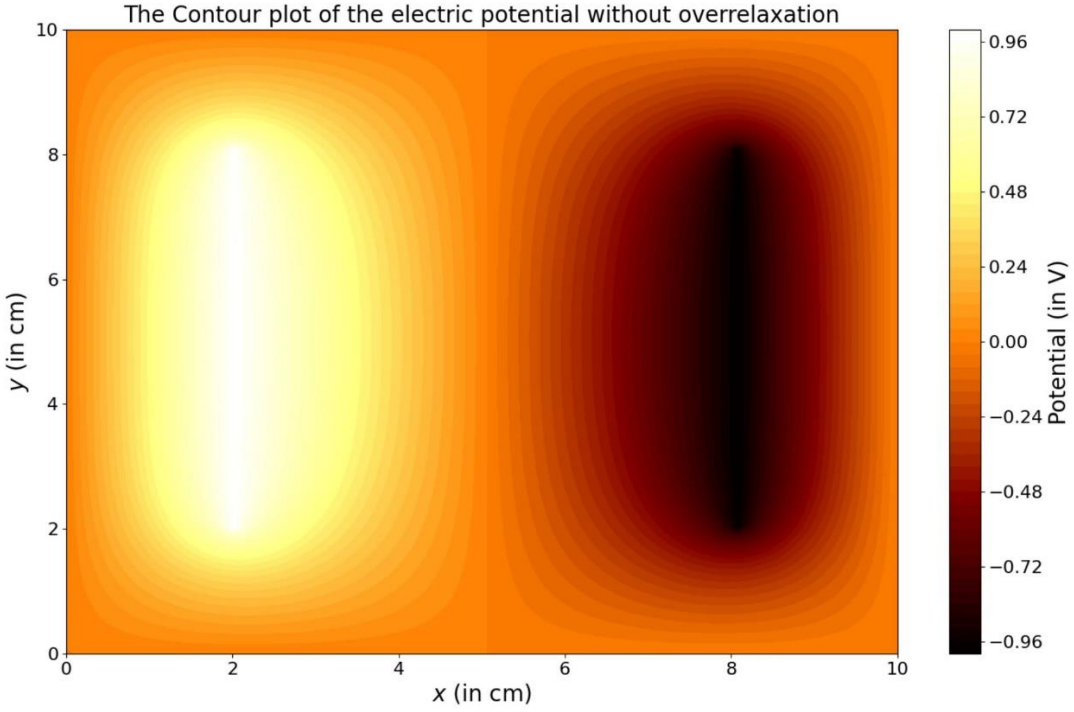


Figure 1.2: The Contour plot of the electric potential on a grid of 100 x 100 points.

Using the relation between the electric field and the potential, which is expressed in Equation 1.2, the electric field at each grid point has been calculated. Specifically, the electric fields in x- and y-directions were calculated using Equation 1.3 and 1.4 respectively.

$$E = -\nabla V \quad \text{[Eq 1.2]}$$

$$E_x = -\frac{\partial V}{\partial x} \quad \text{[Eq 1.3]}$$

$$E_y = -\frac{\partial V}{\partial y} \quad \text{[Eq 1.4]}$$

Using the computed values of the electric field, the stream-line plot of the electric field lines was plotted in Figure 1.3. Note that the electric field lines exit out of the left metal plate that has the highest potential at 1V and enter into the right metal plate that has the lowest potential at -1V, which agrees with the theoretical expectation.

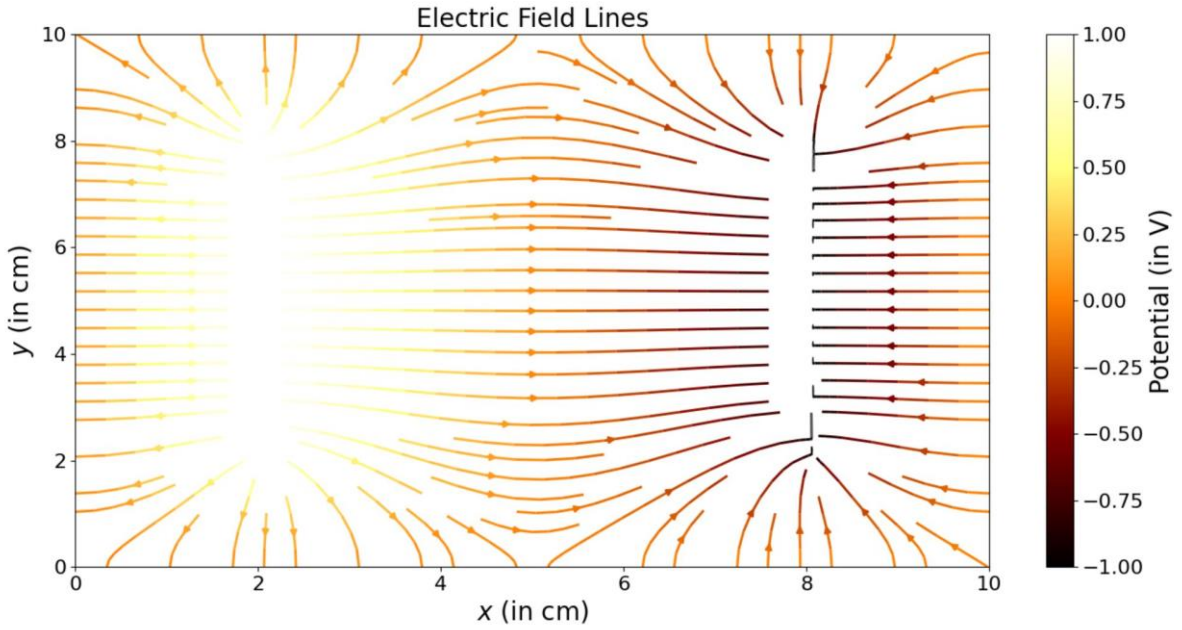


Figure 1.3: The stream line plot of the electric field lines. The line-width and colour-map used are 2 and "afmhot", respectively.

Part b)

The electric potential in part 'a' above was computed using Gauss-Seidel method without overrelaxation. In part 'b', however, the electric potential was calculated using the same method, but with overrelaxation this time. The overrelaxation introduces a new parameter ω, which

determines how much the computation "overestimates" the value at the next iteration. The Gauss-Seidel method with overrelaxation is expressed in Equation 1.5.

$$\phi(x,y) \leftarrow \frac{1+\omega}{4}[\phi(x+a,y)+\phi(x-a,y)+\phi(x,y+a)+\phi(x,y-a)] - \omega\phi(x,y) \qquad \text{[Eq 1.5]}$$

Using the same target accuracy of 10^{-6} volts, the execution time for four cases was compared. The first of which is the method used in part 'a', or Gauss-Seidel method without overrelaxation. The rest are Gauss-Seidel method with overrelaxation, with the parameter ω being 0.1, 0.5 and 0.9. The result is summarized in Table 1.1 below and the actual output of the python program can be found in Appendix A.

	Execution Time (1 decimal place)
No Overrelaxation	44.9 s
Overrelaxation (w = 0.1)	47.4 s
Overrelaxation (w = 0.5)	28.0 s
Overrelaxation (w = 0.9)	6.0 s

Table 1.1: Execution time of Gauss-Seidel method.

Note that the execution time of Gauss-Seidel method with no overrelaxation is 44.9 seconds, whereas that of Gauss-Seidel method with overrelaxation ($\omega = 0.9$) is only 6.0 seconds, which is about 7.5 times less. On the other hand, Gauss-Seidel method with overrelaxation ($\omega = 0.1$) shows an opposite behaviour by having an increase of execution time compared to no overrelaxation, from 44.9 seconds to 47.4 seconds. However, the overrelaxation ($\omega = 0.5$) exhibited the shorter execution time, 28.0 seconds,compared to both no overrelaxation and overrelaxation ($\omega = 0.1$).In other words, it has been shown that the overrelaxation can reduce the execution time significantly, but only when the appropriate value is chosen for the parameter ω. In this exercise, the parameter ω = 0.5 is a better choice than ω = 0.1 and ω = 0.9 is the best choice, which is actually anticipated on page 417 of the textbook that says "For the solution of Laplace's equation using a square grid of points, the best value, the value that gives the fastest solution, is somewhere in the vicinity of ω = 0.9" (Newman).

Question 2

Part a)

The shallow water equation in 1D version can be written as:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -g\frac{\partial \eta}{\partial x} \quad \text{[eq 2.1]}$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial (uh)}{\partial x} = 0 \quad \text{[eq 2.2]}$$

Where h=η-ηb, and equation 2.1 can be rearranged as:

$$\frac{\partial u}{\partial t} = -g\frac{\partial \eta}{\partial x} - \frac{\partial (u^2)}{2\partial x} \quad \text{[eq 2.3]}$$

The time derivative of $\vec{u} = (u, \eta)$ is:

$$\begin{aligned}\frac{\partial u}{\partial t} &= \left(\frac{\partial u}{\partial t}, \frac{\partial \eta}{\partial t}\right) \\ &= \left(-g\frac{\partial \eta}{\partial x} - \frac{\partial (u^2)}{2\partial x}, -\frac{\partial (uh)}{\partial x}\right) \\ &= -\left(\frac{\partial \left(g\eta + \frac{1}{2}u^2\right)}{\partial x}, \frac{\partial u(\eta - \eta_b)}{\partial x}\right)\end{aligned} \quad \text{[eq 2.4]}$$

In 1D, flux-conservative can be written as:

$$\frac{\partial \vec{u}}{\partial t} = -\frac{\partial \vec{F}(\vec{u})}{\partial x} \quad \text{[eq 2.5]}$$

Combining equation 2.4 and 2.5:

$$-\frac{\partial \vec{F}(u,\eta)}{\partial x} = \frac{\partial (u,\eta)}{\partial t} = -\frac{\partial}{\partial x}\left(\frac{1}{2}u^2 + g\eta,\ (\eta - \eta_b)u\right) \quad \text{[eq 2.6]}$$

Therefore:

$$\vec{F}(u,\eta) = \left(\frac{1}{2}u^2 + g\eta, (\eta - \eta_b)u\right) \quad \text{[eq 2.7]}$$

Equation 2.4 can also be expressed using approximation:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -\frac{1}{2\Delta x}\left(\frac{1}{2}\left(u_{j+1}^n\right)^2 + g\eta_{j+1}^n - \frac{1}{2}\left(u_{j-1}^n\right)^2 - g\eta_{j-1}^n\right) \quad \text{[eq 2.8]}$$

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{2\Delta x}\left(\frac{1}{2}\left(u_{j+1}^n\right)^2 + g\eta_{j+1}^n - \frac{1}{2}\left(u_{j-1}^n\right)^2 - g\eta_{j-1}^n\right) \quad \text{[eq 2.9]}$$

$$\frac{\eta_j^{n+1} - \eta_j^n}{\Delta t} = -\frac{1}{2\Delta x}\left(u_{j+1}^n\left(\eta_{j+1}^n - \eta_{b_{j+1}}^n\right) - u_{j-1}^n\left(\eta_{j-1}^n - \eta_{b_{j-1}}^n\right)\right) \quad \text{[eq 2.10]}$$

$$\eta_j^{n+1} = \eta_j^n - \frac{\Delta t}{2\Delta x}\left(u_{j+1}^n\left(\eta_{j+1}^n - \eta_{b_{j+1}}^n\right) - u_{j-1}^n\left(\eta_{j-1}^n - \eta_{b_{j-1}}^n\right)\right) \quad \text{[eq 2.11]}$$

The first point of u and η are calculated using forward difference method:

$$u_0^{n+1} = u_0^n - \frac{\Delta t}{\Delta x}\left(\frac{1}{2}\left(u_1^n\right)^2 + g\eta_1^n - \frac{1}{2}\left(u_0^n\right)^2 - g\eta_0^n\right) \quad \text{[eq 2.12]}$$

$$\eta_0^{n+1} = \eta_0^n - \frac{\Delta t}{\Delta x}\left(u_1^n\left(\eta_1^n - \eta_{b_1}^n\right) - u_0^n\left(\eta_0^n - \eta_{b_0}^n\right)\right) \quad \text{[eq 2.13]}$$

The last point of u and η are calculated using backward difference method:

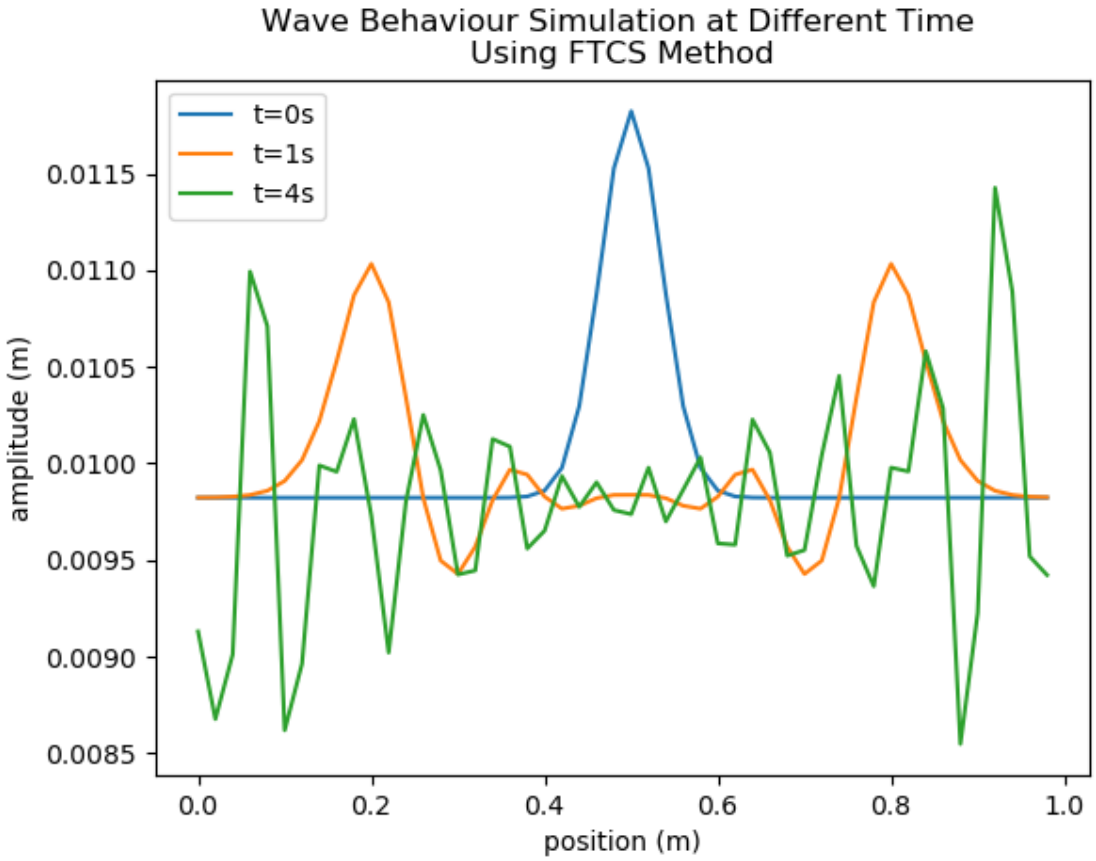
$$u_J^{n+1} = u_J^n - \frac{\Delta t}{\Delta x}\left(\frac{1}{2}\left(u_J^n\right)^2 + g\eta_J^n - \frac{1}{2}\left(u_{J-1}^n\right)^2 - g\eta_{J-1}^n\right) \quad \text{[eq 2.14]}$$

$$\eta_J^{n+1} = \eta_J^n - \frac{\Delta t}{\Delta x}\left(u_J^n\left(\eta_J^n - \eta_{b_J}^n\right) - u_{J-1}^n\left(\eta_{J-1}^n - \eta_{b_{J-1}}^n\right)\right) \quad \text{[eq 2.15]}$$

Part b)

For a simple simulation, 1D shallow water system is implemented on an even ground, where the position of ground is 0 everywhere, and the behaviour of wave at time 0s, 1s and 4s look as following:

Figure 2.1: Simulation of wave at time 0s, 1s and 4s using FTCS method on an even ground at 0m



This figure shows that the waves becomes more chaotic as time increases. However, this should not be the case, because the boundary points in this case are not rigid, meaning the waves are free to move past the boundary points and not bounce back. Therefore, the surface should be less chaotic as waves pass through the boundary. The error at 4s is caused by the inaccuracy of FTCS method, this method will eventually diverge when approximating waves, part c will give a stability analysis to confirm this. A more stable method is used in question 3.

Part c)

Von Neumann stability analysis will be implemented to determine whether the FTCS method is stable when dealing with this wave function. Since it is calculated at (u, η) = (0, 0), it means the wave is initially at rest, therefore h=H, where H is the water column height at rest. So, equation 2.1, 2.3 can be written as:

$$\frac{\partial u}{\partial t} = -g\frac{\partial \eta}{\partial x} \qquad \text{[eq 2.16]}$$

$$\frac{\partial \eta}{\partial t} = -\frac{\partial (uH)}{\partial x}$$
$$= -H\frac{\partial (u)}{\partial x} - u\frac{\partial (H)}{\partial x}$$
$$= -H\frac{\partial (u)}{\partial x} \qquad \text{[eq 2.17]}$$

Express u and η in the form of Fourier series:

$$u(x,t) = \sum_k c_{uk}(t)\, e^{ikx} \qquad \text{[eq 2.18]}$$
$$\eta(x,t) = \sum_k c_{\eta k}(t)\, e^{ikx} \qquad \text{[eq 2.19]}$$

Plugging in a single term from equations 2.18 and 2.19 into equations 2.16 and 2.17 and approximate the result using FTCS:

$$\frac{c_\eta(t+\Delta t) - c_\eta(t)}{\Delta t} e^{ikx} = -\frac{Hc_u(t)\left(e^{ik(x+\Delta x)} - e^{ik(x-\Delta x)}\right)}{2\Delta x} \qquad \text{[eq 2.20]}$$

$$c_\eta(t+\Delta t) = c_\eta(t) - \frac{Hc_u(t)\,\Delta t\left(e^{ik(\Delta x)} - e^{-ik(\Delta x)}\right)}{2\Delta x}$$
$$= c_\eta(t) - \frac{Hc_u(t)\,\Delta t}{\Delta x} i\sin(k\Delta x) \qquad \text{[eq 2.21]}$$

$$\frac{c_u(t+\Delta t) - c_u(t)}{\Delta t} e^{ikx} = -\frac{gc_\eta(t)\left(e^{ik(x+\Delta x)} - e^{ik(x-\Delta x)}\right)}{2\Delta x} \qquad \text{[eq 2.22]}$$

$$c_u(t+\Delta t) = c_u(t) - \frac{gc_\eta(t)\,\Delta t}{\Delta x} i\sin(k\Delta x) \qquad \text{[eq 2.23]}$$

Express above equations in a matrix form:

$$\begin{pmatrix} c_\eta(t+\Delta t) \\ c_u(t+\Delta t) \end{pmatrix} = \begin{pmatrix} 1 & -iH\frac{\Delta t}{\Delta x}\sin(k\Delta x) \\ -ig\frac{\Delta t}{\Delta x}\sin(k\Delta x) & 1 \end{pmatrix} \begin{pmatrix} c_\eta(t) \\ c_u(t) \end{pmatrix}$$
$$= A\begin{pmatrix} c_\eta(t) \\ c_u(t) \end{pmatrix} \qquad \text{[eq 2.24]}$$

Find the eigenvalue value of this matrix:

$$det(A-\lambda I)=0$$

$$det\begin{pmatrix} 1-\lambda & -iH\dfrac{\Delta t}{\Delta x}\sin(k\Delta x) \\ -ig\dfrac{\Delta t}{\Delta x}\sin(k\Delta x) & 1-\lambda \end{pmatrix}=0 \qquad \text{[eq 2.25]}$$

$$(1-\lambda)^2+gH\left(\frac{\Delta t}{\Delta x}\right)^2\sin^2(k\Delta x)=0 \qquad \text{[eq 2.26]}$$

$$\lambda=1\pm i\sqrt{gH\left(\frac{\Delta t}{\Delta x}\right)^2\sin^2(k\Delta x)} \qquad \text{[eq 2.27]}$$

$$|\lambda|=\sqrt{1+gH\left(\frac{\Delta t}{\Delta x}\right)^2\sin^2(k\Delta x)} \qquad \text{[eq 2.28]}$$

Since magnitude of ƛ is always greater than 1, so the solution will diverge eventually. Hence, FTCS method is never stable.

Question 3

Part a)

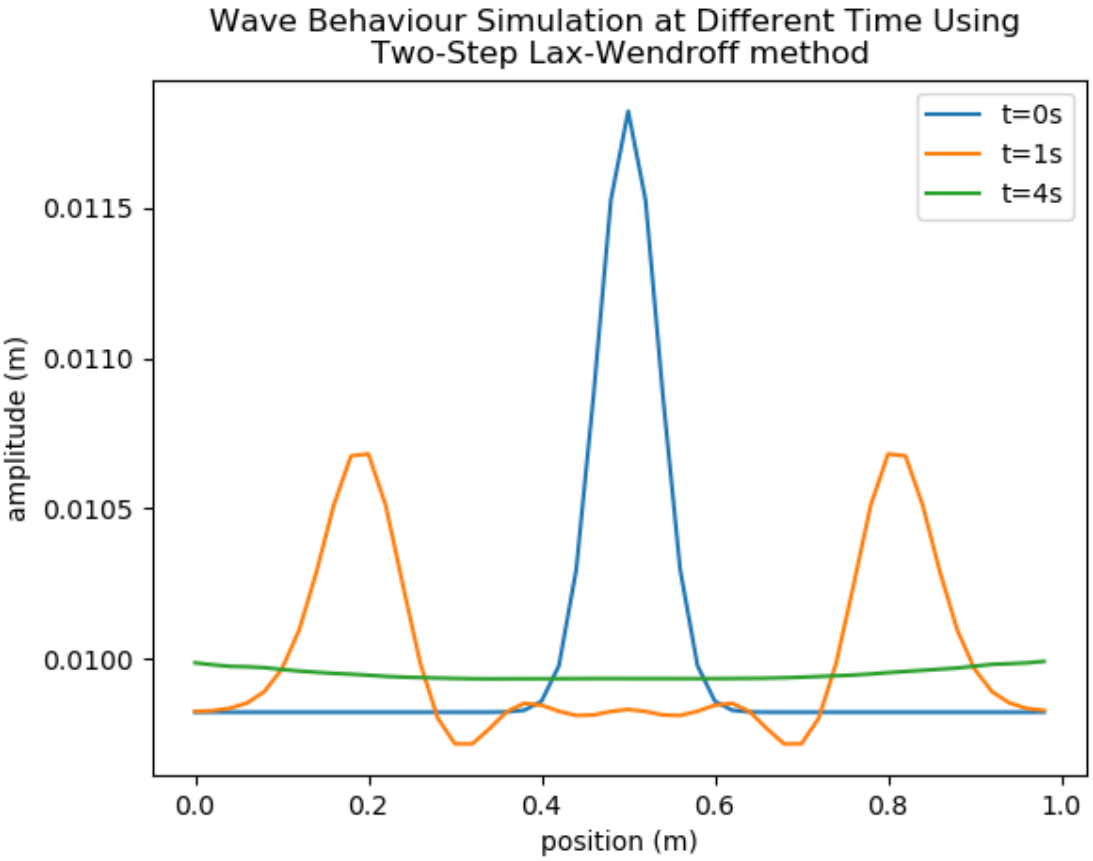
For a better simulation of the wave mentioned in question 2, Two-Step Lax-Wendroff scheme is used, the value of each step is calculated from value of half step:

$$u_{j+1/2}^{n+1/2}=\frac{1}{2}\left(u_{j+1}^{n}+u_{j}^{n}\right)-\frac{\Delta t}{2\Delta x}\left(F\left(u_{j+1}^{n}\right)-F\left(u_{j}^{n}\right)\right) \qquad \text{[eq 3.1]}$$

$$u_{j}^{n+1}=u_{j}^{n}-\frac{\Delta t}{\Delta x}\left(F\left(u_{j+1/2}^{n+1/2}\right)-F\left(u_{j-1/2}^{n+1/2}\right)\right) \qquad \text{[eq 3.2]}$$

The conditions are the same as question 2, and the behaviour of wave at time 0s, 1s and 4s are shown as following:

Figure 3.1: Simulation of wave at time 0s, 1s and 4s using Two-Step Lax-Wendroff method on an even ground at 0m



The waves at time 0s and 1s are similar to figure 2.1; the wave started at the center, and it spread out gradually. However, the wave at 4s looks different from figure 2.1, figure 3.1 shows a calm surface with no wave. It makes sense, because the boundary points are not rigid, so the waves are free to pass through. And there will be no wave left as they pass the boundary.

Part b)

The simulation for question 2 and 3a kept the ground constant at 0. However, if the ground is not even, but given by equation:

$$\eta_b(x) = \frac{\eta_{bs}}{2}\left(1 + \tanh\left(\left(x - x_0\right)\alpha\right)\right) \qquad \text{[eq 3.3]}$$

Where ηbs= H-0.0004m, α=1/8π m-1, and x0= 0.5 m. Also, the boundary points are kept rigid. The wave at 0s, 1s and 4s look as following:

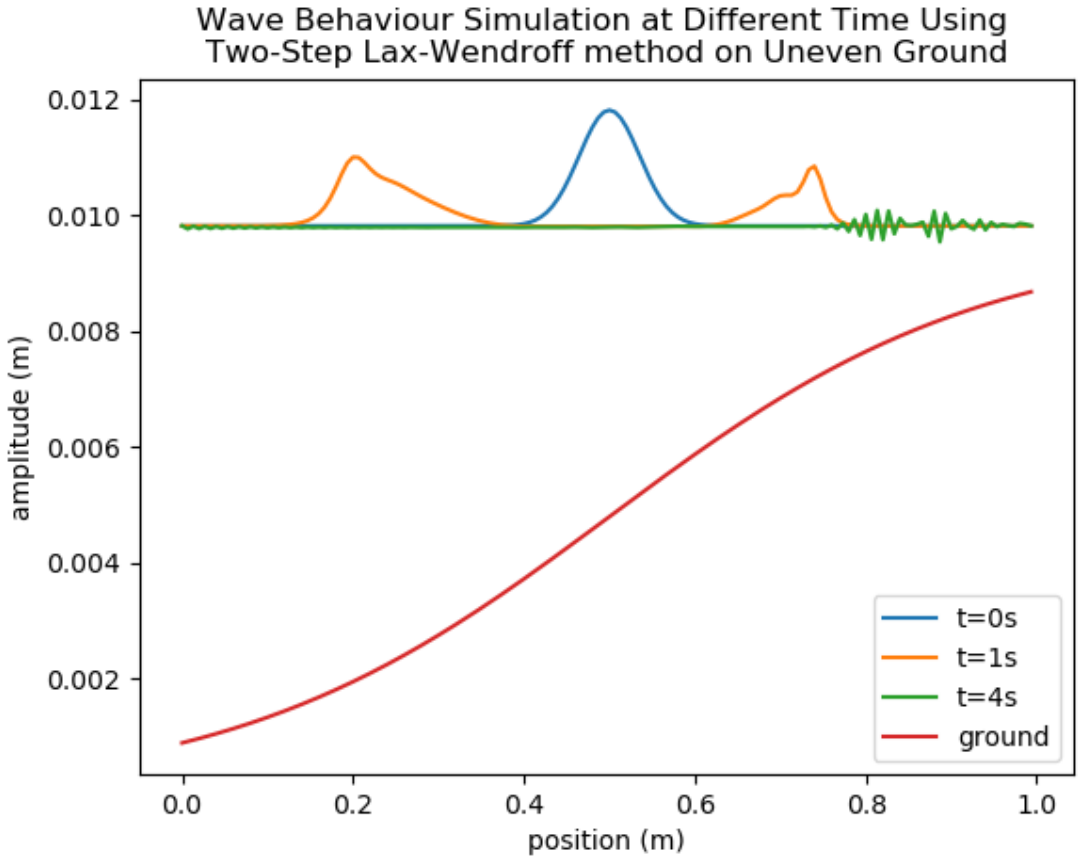


Figure 3.2: Simulation of wave at time 0s, 1s and 4s using Two-Step Lax-Wendroff method on an uneven ground

As the waves spread, the waves skew outward, and the wave going left is faster than the wave going right. Noticeably, the amplitude of the wave decreases as it moves left, and the amplitude of the wave moving right stays almost the same. Since the boundary is set to be rigid, the bounce back waves can be seen at time 4s.

Citation

Newman, Mark. Computational Physics. CreateSpace Independent Publishing Platform. 2012

Appendix A – Execution time of Gauss-Seidel methods

The execution time of Gauss-Seidel method without overrelaxation is 44.92389130592346 s

The execution time of Gauss-Seidel method with overrelaxation (w = 0.1) is 47.362279176712036 s

The execution time of Gauss-Seidel method with overrelaxation (w = 0.5) is 28.041381359100342 s

The execution time of Gauss-Seidel method with overrelaxation (w = 0.9) is 6.048362970352173 s