Lab 8 Report

PHY407 Week 7 Assignment

Landon Wang, Yishi Liu

November 15, 2022

Contribution: Landon Wang writes the entirety of question 1 and completes this document, Yinshi Liu writes the entirety of question 2.

Question 1

a)

To maintain the expendability and help to debug, the code is written in such a way that magnifies the dimension, not strictly define it.

To do that, we transform the dimension of the field to units of cm, which helps simplify the coding.

Figure 1.1 shows the configuration of the capacitor, compared to the figure 3 from the lab manual, it is absolutely identical!

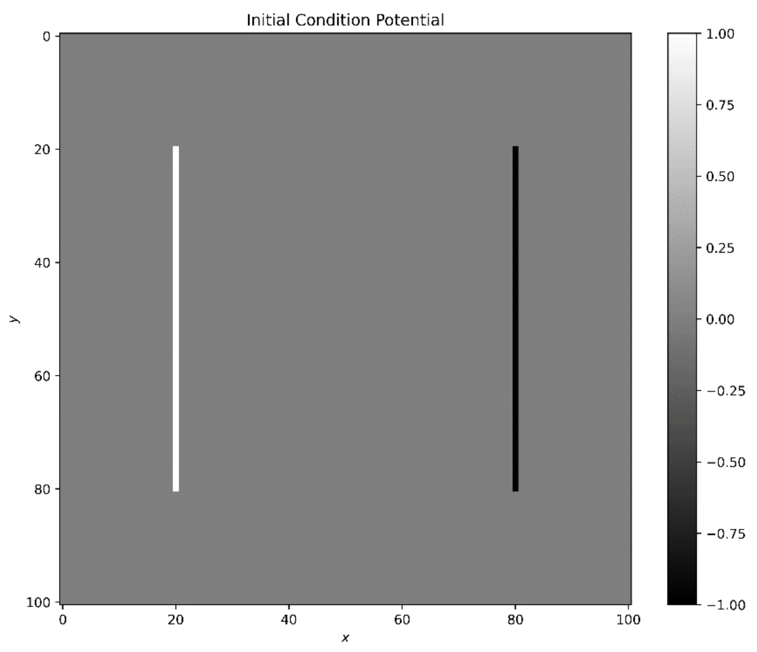

Figure 1.1 Initial Condition Potential Plot

We adopt the code from the laplace.py by Mark Newman, which uses a Jacobi relaxation method to calculate the solution. Since we already have that in hand and its is a good reference to understand the speed increase of the other method, we also run the code using Jacobi method.

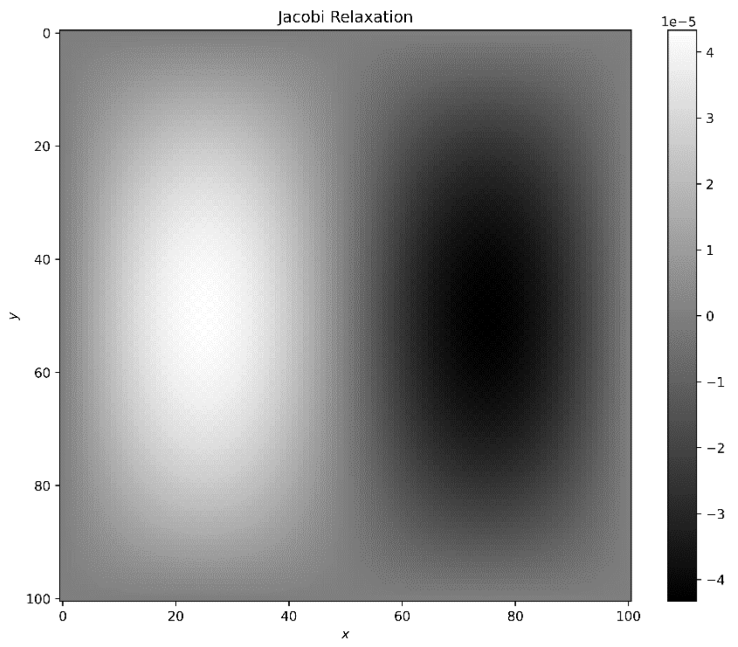

Figure 1.2 Jacobi Relaxation

The time taken for the Jacobi method is 61.19s. This is a result with newly started python kernel. The use of np.empty seems increases the speed a little bit when the code is running the second time.

The Gauss-Seidel method uses two newly calculated value and two previously calculated value to compute the solution. Since it uses slightly newer data, it is more advantageous to complete the required accuracy. Another advantage of this method is the fact that only one matrix is used to store the data which supposably increased its speed.
However, for some reason, the normal assignment of the memory got some error that when we assign another parameter a value from a element of the matrix, the value changes as the element of the matrix manipulated. We suspect this is a problem with numpy array, but we are not sure about it. The direct result of this problem is that we always get the delta been zero because the previous number have been covered with the new number at the same memory address, therefore the previous number is lost and cannot be called normally.
To solve this issue, we used the deepcopy function from the copy package that is come with the basic python. Figure 1.3 shows the result of using the Gauss-Seidel, the run time is 59.25s slightly better than Jacobi which is expected. The result is apparently slightly different, but it seems very silier from the plot.

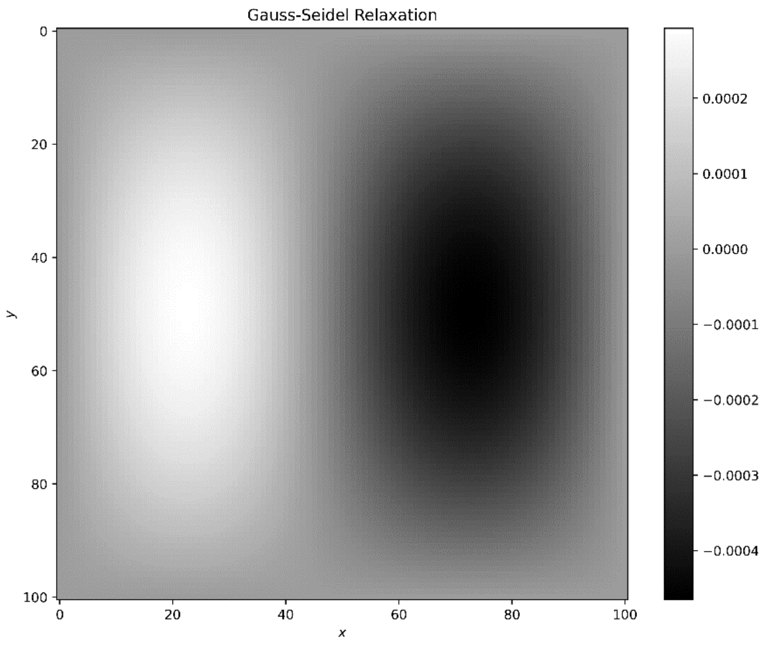

Figure 1.3 Gauss-Seidel Relaxation

b)

The over relaxation is a time saver for a long computation like this example. However it is known to be unstable as we change the ω.

Figure 1.4 shows the Gauss-Seidel with Over-relaxation ω = 0.1, the run time is 56.27s, not much better form the normal Gauss-Seidel.

Figure 1.5 shows the Gauss-Seidel with Over-relaxation ω = 0.5, it is much faster at 30.62s, almost half the normal Gauss-Seidel but the result is apparently way off from the other solutions.

Obviously we saw a mass increase in run time for the two over relaxation solutions. With ω = 0.1, we have a very similar result as we have with Jacobi and normal Gauss-Seidel. But the run time difference are not substantial, at least for our case, we can certainly withstand a few more second of computation. To increased the runtime, we used the ω = 0.5 which does give us a great speed, but the solution is obviously far from the 'correct' way. For some reasons, we lost every positive result and they simply goes to zero. This does not happen with every other case, this leads us to believe that ω = 0.5 is not a good way to go. Even than, we are happy to say the solution is at least stable, not making the computer crash!

Code out puts:

```
Jacobi Time taken  61.197545766830444

Gauss-Seidel Time taken  59.254398822784424

Gauss-Seidel Over-relaxation ω = 0.10 Time taken  56.276207447052

Gauss-Seidel Over-relaxation ω = 0.50 Time taken  30.6275053024292
```

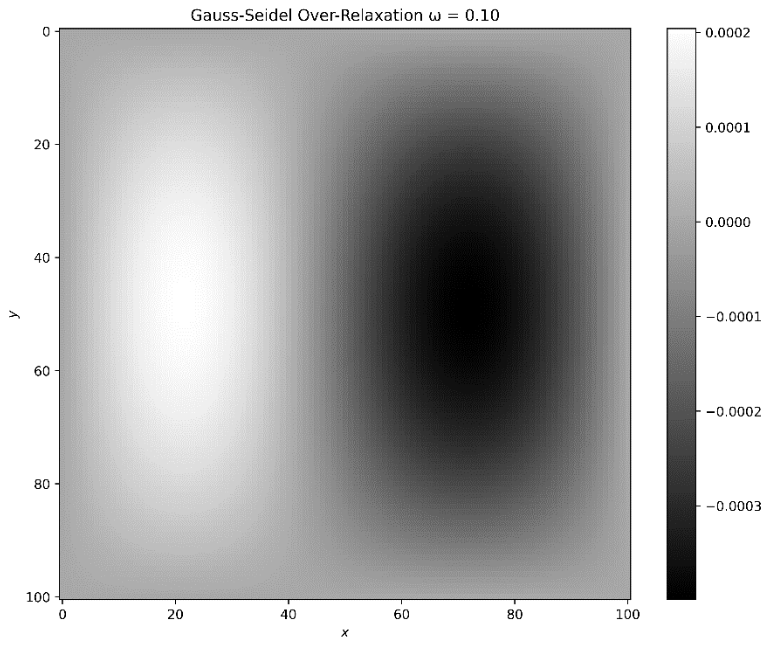

Figure 1.4 Gauss-Seidel Over-relaxation ω = 0.1

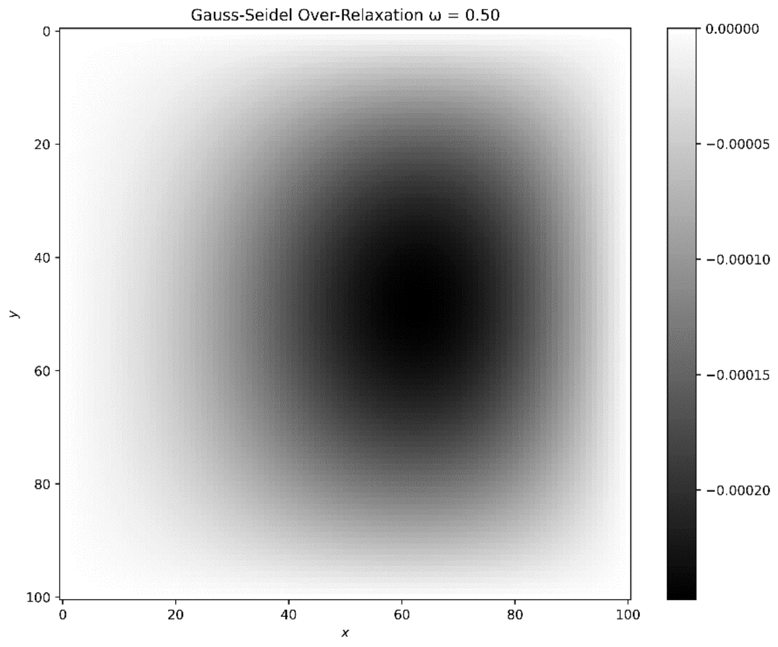

Figure 1.5 Gauss-Seidel Over-relaxation ω = 0.5

Question 2

a)

Rearrange equation (6) into:

$$-\frac{\partial u}{\partial t} = u\frac{\partial u}{\partial x} + g\frac{\partial \eta}{\partial x} \tag{2.1}$$

$$-\frac{\partial \eta}{\partial t} = \frac{\partial (uh)}{\partial x} \tag{2.2}$$

Convert equations into flux conservative form using

$-\frac{\partial \vec{F}(\vec{u})}{\partial x} = \frac{\partial \vec{u}}{\partial t}$, with $\vec{u} = (u, \eta)$:

$$\frac{\partial F(u)}{\partial x} = u\frac{\partial u}{\partial x} + g\frac{\partial \eta}{\partial x} \tag{2.3}$$

$$\frac{\partial F(\eta)}{\partial x} = \frac{\partial (uh)}{\partial x} \tag{2.4}$$

Integrate both sides, get:

$$\int \frac{\partial F(u)}{\partial x} dx = \int u\frac{\partial u}{\partial x} dx + \int g\frac{\partial \eta}{\partial x} dx$$

$$\Rightarrow F(u) = \frac{1}{2}u^2 + g\eta \tag{2.5}$$

$$\int \frac{\partial F(\eta)}{\partial x} dx = \int \frac{\partial (uh)}{\partial x}$$

$$\int \frac{\partial F(\eta)}{\partial x} dx = \int \frac{\partial [u(\eta - \eta_b)]}{\partial x} dx$$

$$\Rightarrow F(\eta) = u(\eta - \eta_b) \tag{2.6}$$

Combine eqn. (2.5), (2.6) to obtain the final result:

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

as required.

Discretize the function using equation (5):

$$\begin{aligned} u_j^{n+1} &= u_j^n - \frac{\Delta t}{2\Delta x}\left[\left(F_{j+1}^n\right) - \left(F_{j-1}^n\right)\right]u_j^{n+1} \\ &= u_j^n - \frac{\Delta t}{2\Delta x}\left[\left(\frac{1}{2}\left(u_{j+1}^n\right)^2 + g\eta_{j+1}^n\right) - \left(\frac{1}{2}\left(u_{j-1}^n\right)^2 + g\eta_{j-1}^n\right)\right] \end{aligned} \tag{2.8}$$

$$\begin{aligned} \eta_j^{n+1} &= \eta_j^n - \frac{\Delta t}{2\Delta x}\left(F_{j+1}^n - F_{j-1}^n\right)\eta_j^{n+1} \\ &= \eta_j^n - \frac{\Delta t}{2\Delta x}\left[\left(\eta_{j+1}^n - \eta_{b,j+1}^n\right)u_{j+1}^n - \left(\eta_{j-1}^n - \eta_{b,j-1}^n\right)u_{j-1}^n\right] \end{aligned} \tag{2.9}$$

for j = 0 and j = 50:

$$u_0^{n+1} = u_0^n - \frac{\Delta t}{\Delta x}(F_1^n - F_0^n) \tag{2.10}$$

$$u_J^{n+1} = u_J^n - \frac{\Delta t}{\Delta x}(F_J^n - F_{J-1}^n) \tag{2.11}$$

$$\eta_0^{n+1} = \eta_0^n - \frac{\Delta t}{\Delta x}(F_1^n - F_0^n) \tag{2.12}$$

$$\eta_J^{n+1} = u_J^n - \frac{\Delta t}{\Delta x}(F_J^n - F_{J-1}^n) \tag{2.13}$$

Because the derivative $\frac{\partial F}{\partial x}$ has to be approximated to:

$$\left.\frac{\partial F}{\partial x}\right|_0^n = \frac{1}{\Delta x}(F_1^n - F_0^n)$$
$$\left.\frac{\partial F}{\partial x}\right|_J^n = \frac{1}{\Delta x}(F_J^n - F_{J-1}^n)$$

b)
Code Output:

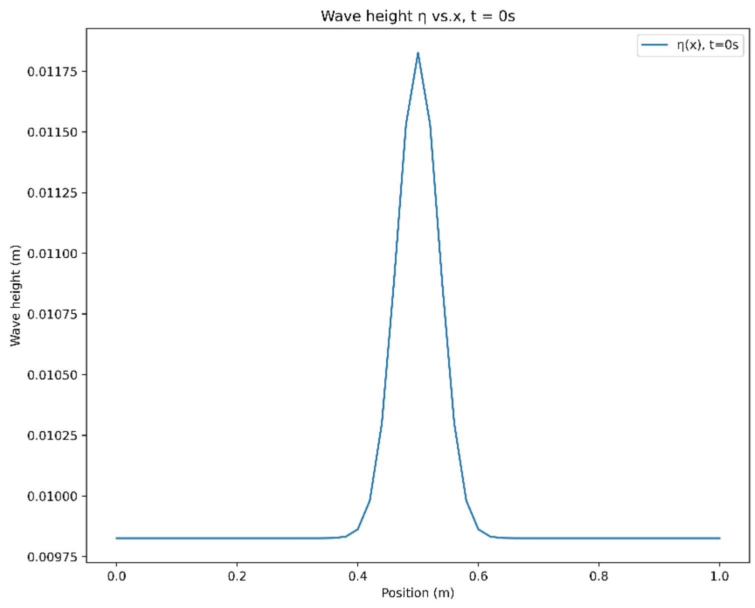

Figure 2.1 Wave height at t = 0s

At t = 0, the wave shape follows the initial condition

$$\eta(x,0) = H + Ae^{-\frac{(x-\mu)^2}{\sigma^2}} - \langle Ae^{-\frac{(x-\mu)^2}{\sigma^2}} \rangle \tag{2.14}$$

which does indeed resemble a Gaussian distribution.

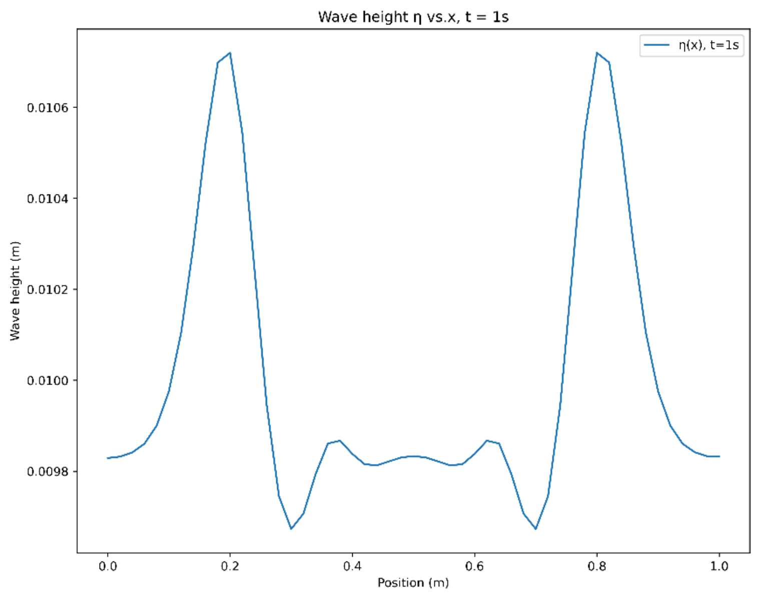

Figure 2.2 Wave height at t = 1s

At t = 1 second, the wave appears to evolve normally, with no noticeable problems in this figure.

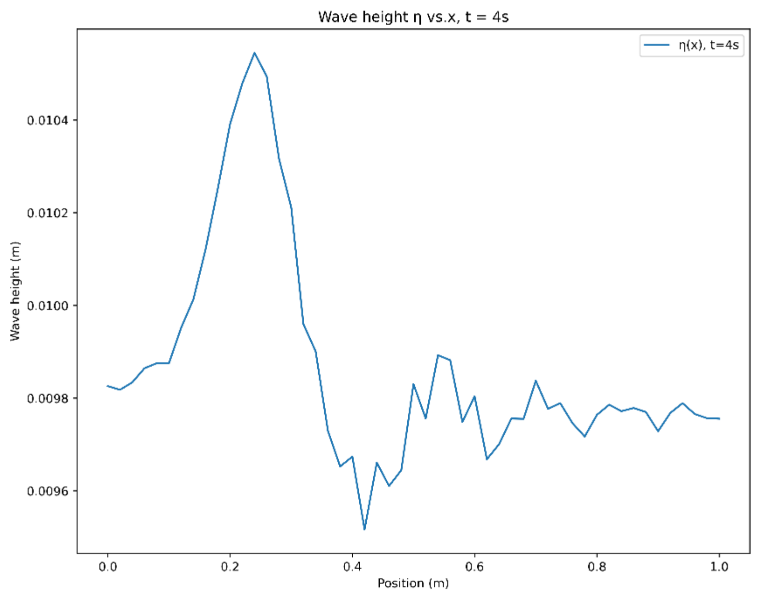

Figure 2.3 Wave height at t = 4s

At t = 4, there are significant problems with the solution. This solution cannot be used to determine the evolution of the wave. This behaviour corresponds with an unstable system very well, which is expected for this system.

c)

Substitute initial value u = 0, $\eta = H$ into equation 6 and expand, get:

$$\frac{\partial u}{\partial t} = -g\frac{\partial \eta}{\partial x} - (0)\frac{\partial u}{\partial x}$$

$$= -g\frac{\partial \eta}{\partial x} \tag{2.15}$$

For η:

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

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

$$= -(H - 0)\frac{\partial u}{\partial x} + (0)u$$

$$= -H\frac{\partial u}{\partial x} \tag{2.16}$$

as required.

Discretize equation (2.15), (2.16):

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -g\frac{\eta_{j+1}^n - \eta_j^n}{2\Delta x} \tag{2.17}$$

$$\frac{\eta_j^{n+1} - \eta_j^n}{\Delta t} = -H\frac{u_{j+1}^n - u_j^n}{2\Delta x} \tag{2.18}$$

Expand $u(x,t), \eta(x,t)$ into a Fourier Series, since $u(x,t), \eta(x,t) = c_k(t)e^{ikx}$:

$$u(x, t + \Delta t) = u(x,t) - g\frac{\Delta t}{2\Delta x}\left(\eta_{j+1}^n - \eta_j^n\right)$$

$$u(x, t + \Delta t) = c_{ku}(t)e^{ikx} - c_{k\eta}(t)g\frac{\Delta t}{2\Delta x}\left(e^{ik(x+\Delta x)} - e^{ik(x-\Delta x)}\right)$$

$$u(x, t + \Delta t) = c_{ku}(t)e^{ikx} - c_{k\eta}(t)e^{ikx}g\frac{\Delta t}{\Delta x}\sin(k\Delta x)$$

$$\Rightarrow c_{ku}(t + \Delta t) = c_{ku}(t) - c_{k\eta}(t)e^{ikx}g\frac{\Delta t}{\Delta x}\sin(k\Delta x)$$

$$c_{ku}(t + \Delta t) = c_{ku}(t) - c_{k\eta}(t)e^{ikx}g\frac{\Delta t}{\Delta x}\sin(k\Delta x) \tag{2.19}$$

$$\eta(x, t + \Delta t) = \eta(x,t) - H\frac{\Delta t}{2\Delta x}\left(\eta_{j+1}^n - \eta_j^n\right)$$

$$\eta(x, t + \Delta t) = c_{k\eta}(t)e^{ikx} - c_{ku}(t)H\frac{\Delta t}{2\Delta x}\left(e^{ik(x+\Delta x)} - e^{ik(x-\Delta x)}\right)$$

$$\eta(x, t + \Delta t) = c_{k\eta}(t)e^{ikx} - c_{k\eta}(t)H\frac{\Delta t}{\Delta x}\sin(k\Delta x)$$

$$\Rightarrow c_{k\eta}(t + \Delta t) = -c_{ku}(t)H\frac{\Delta t}{\Delta x}\sin(k\Delta x) + c_{k\eta}(t) \tag{2.20}$$

Write $u(x,t), \eta(x,t)$ in vector form:

$$\begin{pmatrix} u(x,t) \\ \eta(x,t) \end{pmatrix} = \begin{pmatrix} c_{ku}(t) \\ c_{k\eta}(t) \end{pmatrix} e^{ikx} \tag{2.21}$$

$$c_k(t+h) = \begin{pmatrix} 1 & -g\dfrac{\Delta t}{\Delta x}\sin(k\Delta x) \\ -H\dfrac{\Delta t}{\Delta x}\sin(k\Delta x) & 1 \end{pmatrix} c_k(t) \tag{2.22}$$

Find eigenvalue λ of the matrix:

$$A = \begin{pmatrix} 1-\lambda & -g\dfrac{\Delta t}{\Delta x}\sin(k\Delta x) \\ -H\dfrac{\Delta t}{\Delta x}\sin(k\Delta x) & 1-\lambda \end{pmatrix} \tag{2.23}$$

Find the characteristic equation:

$$(1-\lambda)^2 - gH\left(\frac{\Delta t}{\Delta x}\right)^2 \sin(k\Delta x)^2 = 0$$

$$\lambda^2 - 2\lambda - gH\left(\frac{\Delta t}{\Delta x}\right)^2 \sin(k\Delta x)^2 + 1 = 0 \tag{2.24}$$

$$\Rightarrow \lambda = 2 \pm \frac{\sqrt{4 - 4\left(gH\left(\frac{\Delta t}{\Delta x}\right)^2 (\sin(k\Delta x))^2 + 1\right)}}{2}$$

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

$$= 1 \pm i\sqrt{\left(gH\left(\frac{\Delta t}{\Delta x}\right)^2 (\sin(k\Delta x))^2\right)} \tag{2.25}$$

Find the magnitude of eigenvalue

$$|\lambda| = \sqrt{1 + \left(\frac{\Delta t}{\Delta x}\right)^2 gH\sin^2(k\Delta x)} \tag{2.26}$$

as required.

Since the magnitude of the eigenvalue is always greater than unity, this system is numerically unstable. This agree with the behaviour of the system in part b, where numerical error gradually propagates and eventually dominated the solution, which is the expected result.