

Problem #2.7 in the book

Solution:

$$v_t - u_x = 0$$

$$u_t - p(v)_x = 0$$

This is the same as

$$u_t - p'(v)v_x = 0$$

$$\begin{bmatrix} v \\ u \end{bmatrix}_t + \begin{bmatrix} 0 & 1 \\ -p'(v) & 0 \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix}_x = 0$$

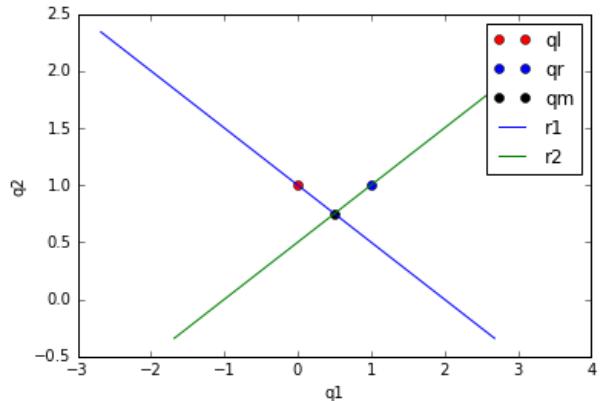
$$\lambda^2 = -p'(v)$$

For the eigenvalues to be real, $p'(v)$ has to be negative. Thus, for the matrix to be diagonalizable and the system of equations to be hyperbolic, $p'(v)$ must be less than 0.

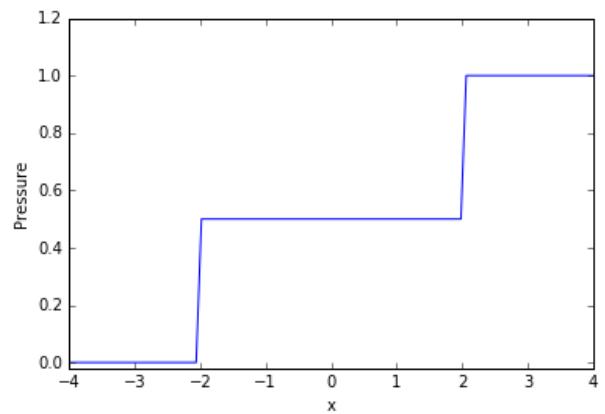
Problem #3.1 in the book You might want to do Problem 3.2 first.

Solution:

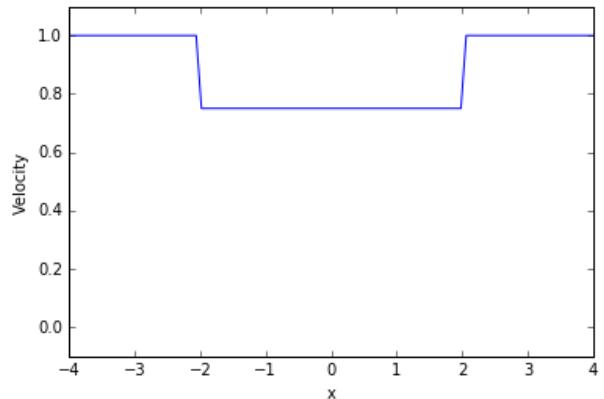
- (a) • The phase plot of q



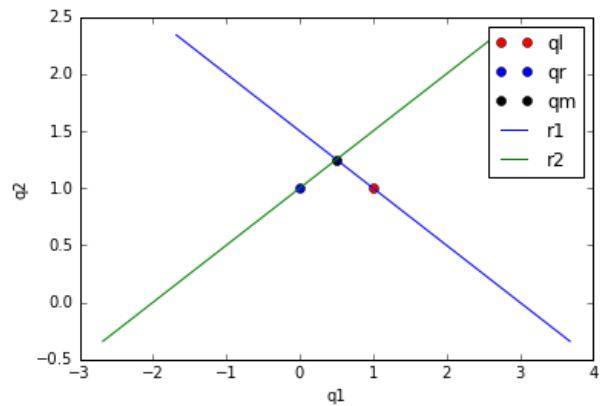
- Pressure at time =1



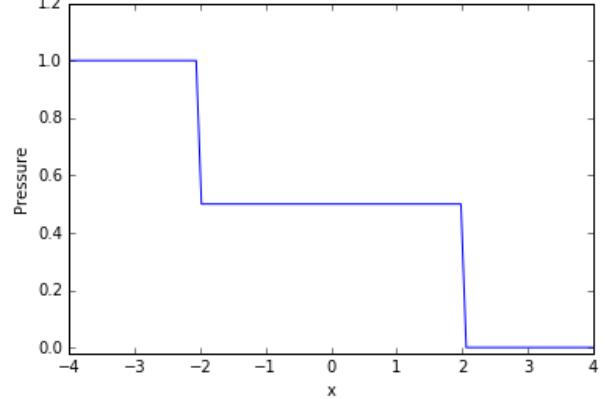
- Velocity at time =1



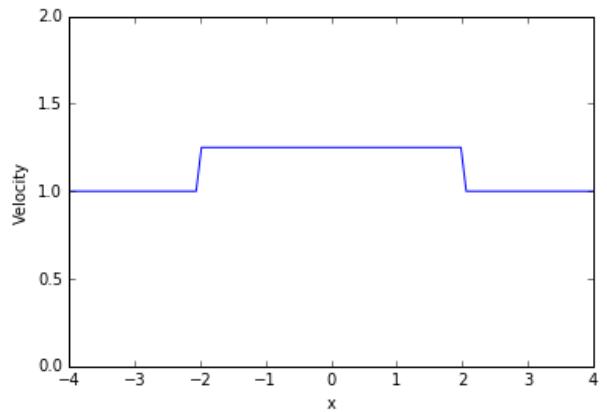
- (b) • The phase plot of q



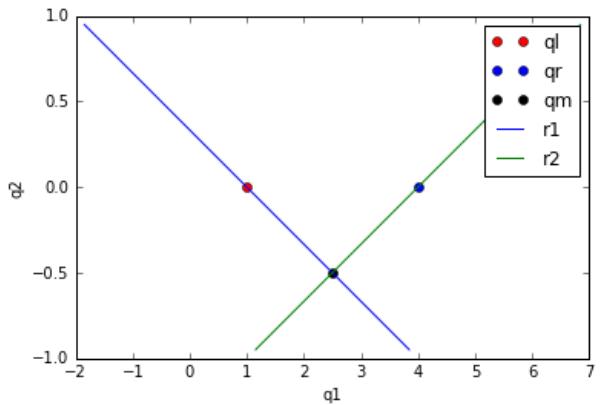
- Pressure at time =1



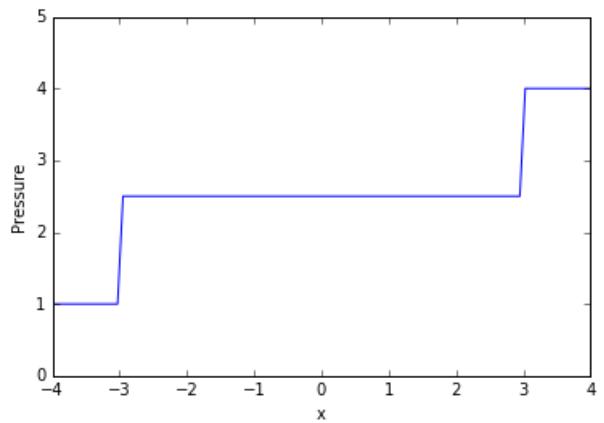
- Velocity at time =1



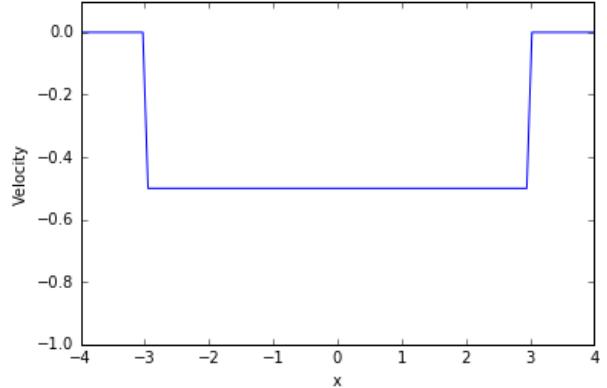
- (c) • The phase plot of q



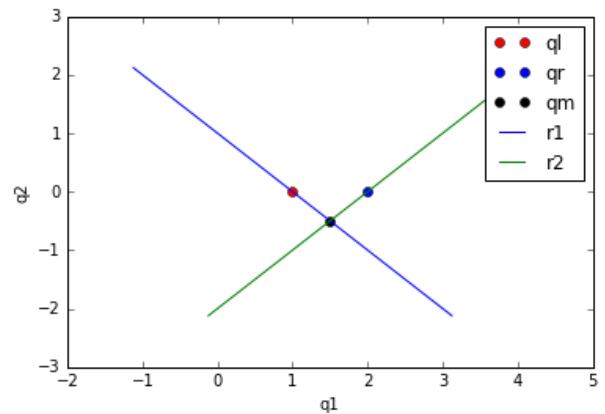
- Pressure at time =1



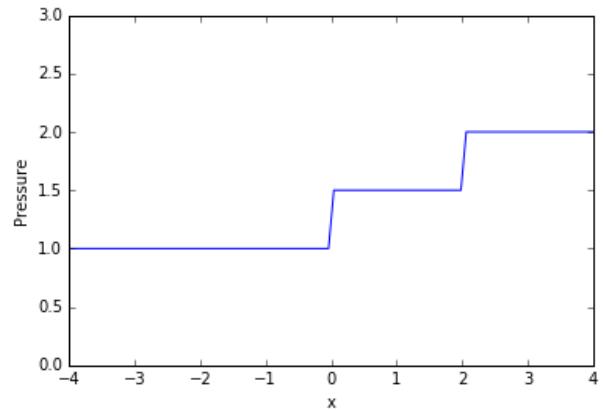
- Velocity at time =1



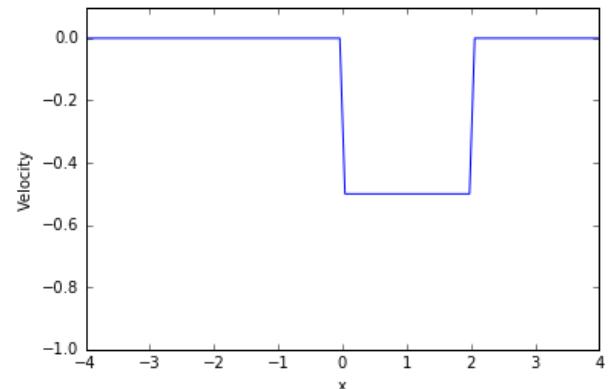
- (d) • The phase plot of q



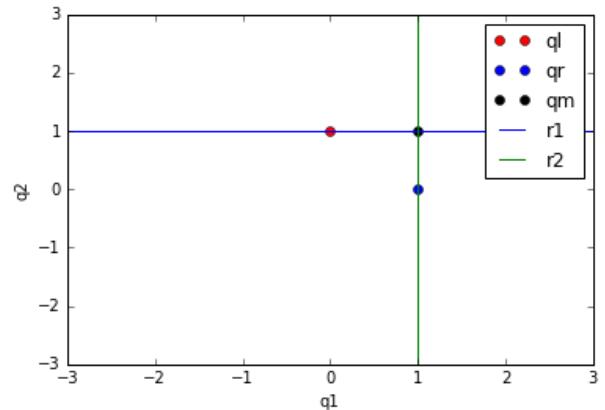
- Pressure at time =1



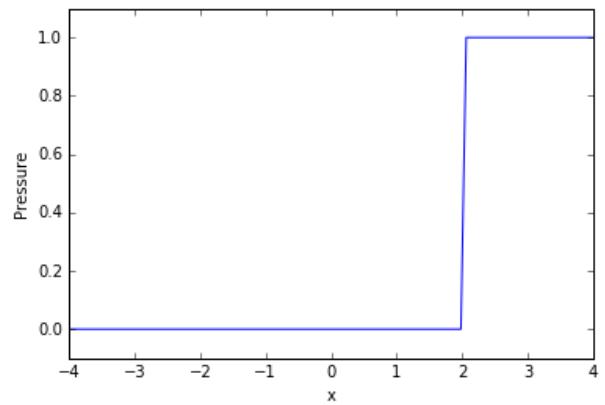
- Velocity at time =1



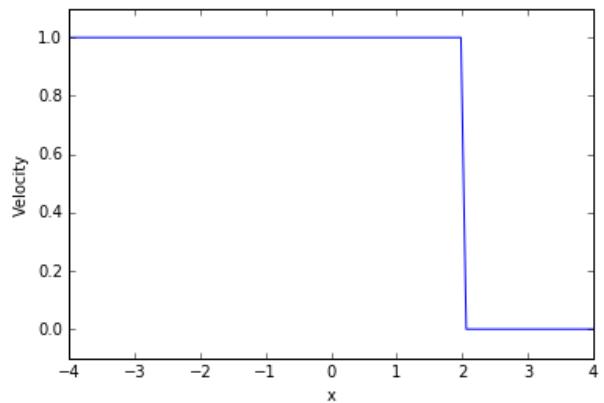
- (e) • The phase plot of q



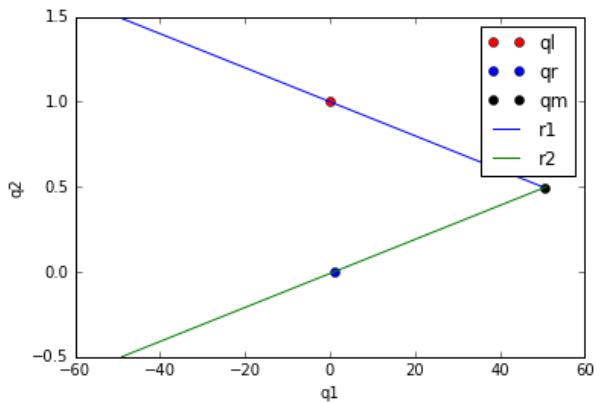
- Pressure at time =1



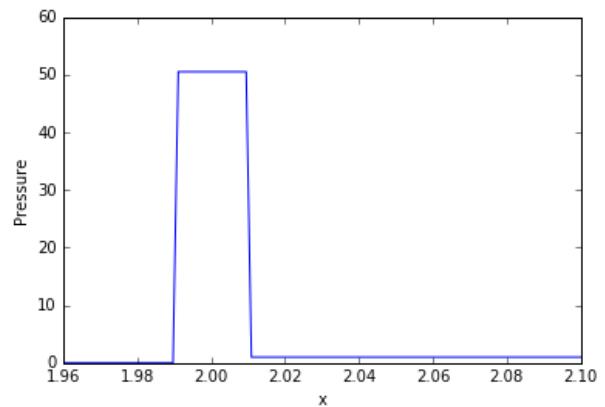
- Velocity at time =1



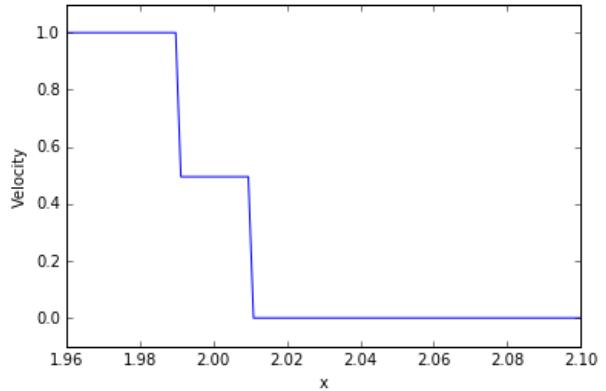
- (f) • The phase plot of q



- Pressure at time =1



- Velocity at time =1



Problem #3.2 in the book

You can use Matlab for this one, but I suggest you try writing the program in Python. A Jupyter notebook will be provided to help you get started.

Note that the module `numpy.linalg` contains an `eig` function similar to Matlab.

Solution:

We start with the initial conditions.

```
import numpy as np
import pylab as pl
A = np.array([[2., 2], [0, -3]])
ql = np.array([1, 1])
qr = np.array([0, 3])
```

Make some useful functions

```
#Get Eigen-values/vectors
def Reisolve(A,ql,qr):
    lam,R = np.linalg.eig(A)
    T= qr-ql
    Rinv = np.linalg.inv(R)
    alpha =np.dot(Rinv,T)
    return alpha, lam ,R

#Phaseplots
def phaseplot(ql,qr,R,qm):
    k1= -3
    k2= 3
    r1 = R[:,0]
    r2 = R[:,1]
    u1 = ql+k1*r1
    u2 = ql + k2*r1
    u = np.vstack((u1,u2))
    v1 = qr+k1*r2
    v2 = qr + k2*r2
```

```

v = np.vstack((v1,v2))
pl.plot(ql[0],ql[1], 'ro') #ql
pl.plot(qr[0],qr[1], 'bo') #qr
pl.plot(qm[0],qm[1], 'ko') #qm
pl.plot(u[:,0],u[:,1]) #First eigenvector
pl.plot(v[:,0],v[:,1]) #Second Eigenvector
pl.xlabel("q1")
pl.ylabel("q2")

#Solve for various x at particular time
def qsolve(xcol,t,lambda,R,alpha):
    i=0
    q=np.zeros((2,len(xcol)))
    for x in xcol:
        if x/t < lambda[0]: #Naturally, there is a bug at t=0
            q[:,i]=ql
        elif x/t < lambda[1]:
            q[:,i] = ql + alpha[0]*R[:,0]
        else:
            q[:,i]=qr
        i+=1
    return q

#Solve for middle states
def qmsolve(R,alpha,ql):
    i=0
    qm=np.zeros((2,1))
    qm[:,i] = ql + alpha[0]*R[:,0]
    return qm

```

Solve and plot.

```

al, lambda, R1= Reisolve(A,ql,qr)
ind = np.argsort(lambda)
lambda=lambda[ind]
R= R1[:,ind]
al= al[ind]
x=np.linspace(-10,10,100)
t=1
qm=qmsolve(x,t,lambda,R,al)
pl.figure(3)
phaseplot(ql,qr,R,qm)
pl.figure(1)
pl.plot(x,q[0,:])
pl.xlabel("x")
pl.ylabel("q1")

pl.figure(2)
pl.plot(x,q[1,:])
pl.xlabel("x")
pl.ylabel("q2")

```

Problem #3.3 in the book

Following the sort of thing done in script `problem_3_5.py` might be useful if you want to insert a figure in your solution, or you can draw with another programming language, or sketch the solution by hand and scan.

Solution:

The code to plot the figure is given below.

```
from pylab import *

clf() # clear figure

plot([-4,4],[0,0], 'k') # x-axis 'k' means black line
plot([-4,-4],[0,6], 'k') # t-axis
plot([4,4],[0,6], 'k') # right boundary

def plot_rightgoing(x1,t1,s):
"""
plot right-going wave of slope s starting at (x1,t1) to right boundary
"""
    t2 = t1 + s*(4-x1)
    plot([x1,4], [t1,t2], 'k')

def plot_leftgoing(x1,t1,s):
"""
plot left-going wave of slope s starting at (x1,t1) to left boundary
"""
    t2 = t1 + s*(-4-x1)
    plot([x1,-4], [t1,t2], 'k')

plot_rightgoing(0,0,1)
plot_rightgoing(0,0,1)
def Reisolve(A,ql,qr):
    lam,R = linalg.eig(A)
    T= qr-ql
    Rinv = linalg.inv(R)
    alpha =dot(Rinv,T)
    return alpha, lam ,R
def qsolve(xcol,t,lambda,R,alpha):
    i=0
    q=np.zeros((2,len(xcol)))
    for x in xcol:
        if x/t <lambda[0]: #Naturally, there is a bug at t=0
            q[:,i]=ql
        elif x/t <lambda[1]:
            q[:,i] = ql + alpha[0]*R[:,0]
        else:
            q[:,i]=qr
        i+=1
    return q
A = np.array([[0,0,4],[0,1,0],[1,0,0]])
```

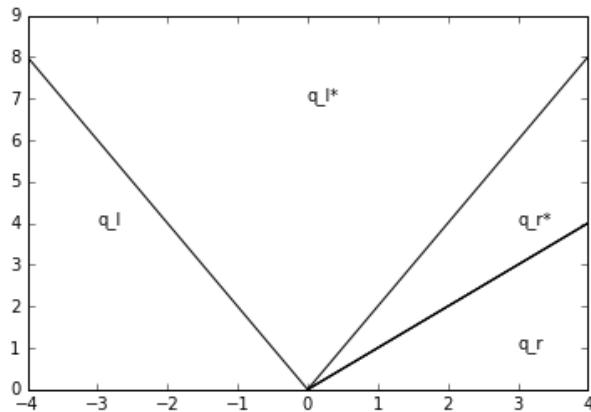
```

ql = np.array([1, 2, 0])
qr = np.array([1, 5, 1])
al, lamda, R1= Reisolve(A,ql,qr)
ind = np.argsort(lamda)
lamda=lamda[ind]
R= R1[:,ind]

#Plotting the regions
for l in lamda:
    if l>0:
        plot_rightgoing(0,0,l)
    else:
        plot_leftgoing(0,0,l)
text(-3, 4, 'q_l')
text(0, 7, 'q_l*')
text(3.0, 4, 'q_r*')
text(3, 1, 'q_r')

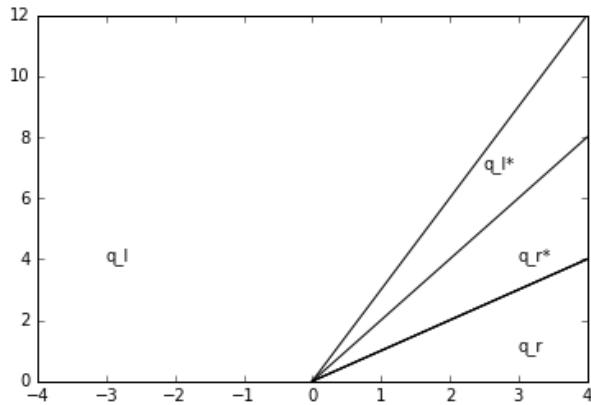
```

(a)



$$q_l^* = \begin{bmatrix} 0 \\ 2 \\ 0.5 \end{bmatrix}, q_r^* = \begin{bmatrix} 0 \\ 5 \\ 0.5 \end{bmatrix}$$

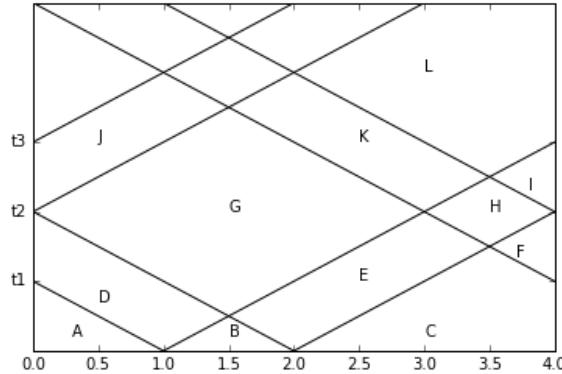
(b)



$$q_l^* = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, q_r^* = \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}$$

Problem #3.5 in the book

The script `problem_3_5.py` was used to generate this figure:



To solve this problem, determine the states A, B, \dots, M and also the times t_1, t_2, t_3 . The times can be written in terms of the parameters ρ_0 and K_0 , which were not stated in the problem.

For example,

$$A = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad , B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad , C = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \dots$$

Solution:

From the initial conditions, we have

$$A = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad , B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad , C = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The eigenvalues are $\sqrt{K_0/\rho_0}$ and $-\sqrt{K_0/\rho_0}$.

$$t_1 = \frac{\Delta x}{-\lambda} = \sqrt{\frac{\rho_0}{K_0}}$$

$$t_2 = 2\sqrt{\frac{\rho_0}{K_0}}$$

$$t_3 = 3\sqrt{\frac{\rho_0}{K_0}}$$

The eigenvector matrix and it's inverse are

$$R = \begin{bmatrix} -Z_0 & Z_0 \\ 1 & 1 \end{bmatrix}, \quad , R^{-1} = \frac{1}{2Z_0} \begin{bmatrix} -1 & Z_0 \\ 1 & Z_0 \end{bmatrix} \quad ,$$

where

$$Z_0 = \sqrt{K_0\rho_0}$$

D:

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = R^{-1}(B - A) = \frac{1}{2Z_0} \begin{bmatrix} -1 & Z_0 \\ 1 & Z_0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{2Z_0} \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$W_1 = a_1 R_1 = \frac{1}{2Z_0} \begin{bmatrix} Z_0 \\ -1 \end{bmatrix}$$

$$W_2 = a_2 R_2 = \frac{1}{2Z_0} \begin{bmatrix} Z_0 \\ 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + a_1 R_1 = \begin{bmatrix} \frac{1}{2} \\ \frac{-1}{2Z_0} \end{bmatrix}$$

E:

$$\begin{bmatrix} a_3 \\ a_4 \end{bmatrix} = R^{-1}(C - B) = \frac{1}{2Z_0} \begin{bmatrix} -1 & Z_0 \\ 1 & Z_0 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \end{bmatrix} = \frac{1}{2Z_0} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$E = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + a_1 R_1 = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2Z_0} \end{bmatrix}$$

$$W_3 = a_3 R_1 = \frac{1}{2Z_0} \begin{bmatrix} -Z_0 \\ 1 \end{bmatrix}$$

$$W_4 = a_4 R_2 = \frac{1}{2Z_0} \begin{bmatrix} -Z_0 \\ -1 \end{bmatrix}$$

F: At the boundary, the left going wave W_1 reflects and we have the new wave W'_1

$$W'_1 = -W_1 = \frac{1}{2Z_0} \begin{bmatrix} -Z_0 \\ 1 \end{bmatrix}$$

$$F = D - W'_1 = D + W_1 = \begin{bmatrix} 1 \\ \frac{-1}{Z_0} \end{bmatrix}$$

G:

$$G = D + W'_1 = D - W_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

H:

$$H = E + W_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

I:

$$I = F + W_3 = \frac{1}{2Z_0} \begin{bmatrix} Z_0 \\ -1 \end{bmatrix}$$

J:

$$J = G + W'_4 = G - W_4 = \frac{1}{2Z_0} \begin{bmatrix} Z_0 \\ 1 \end{bmatrix}$$

K:

$$K = I - W'_3 = I + W_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

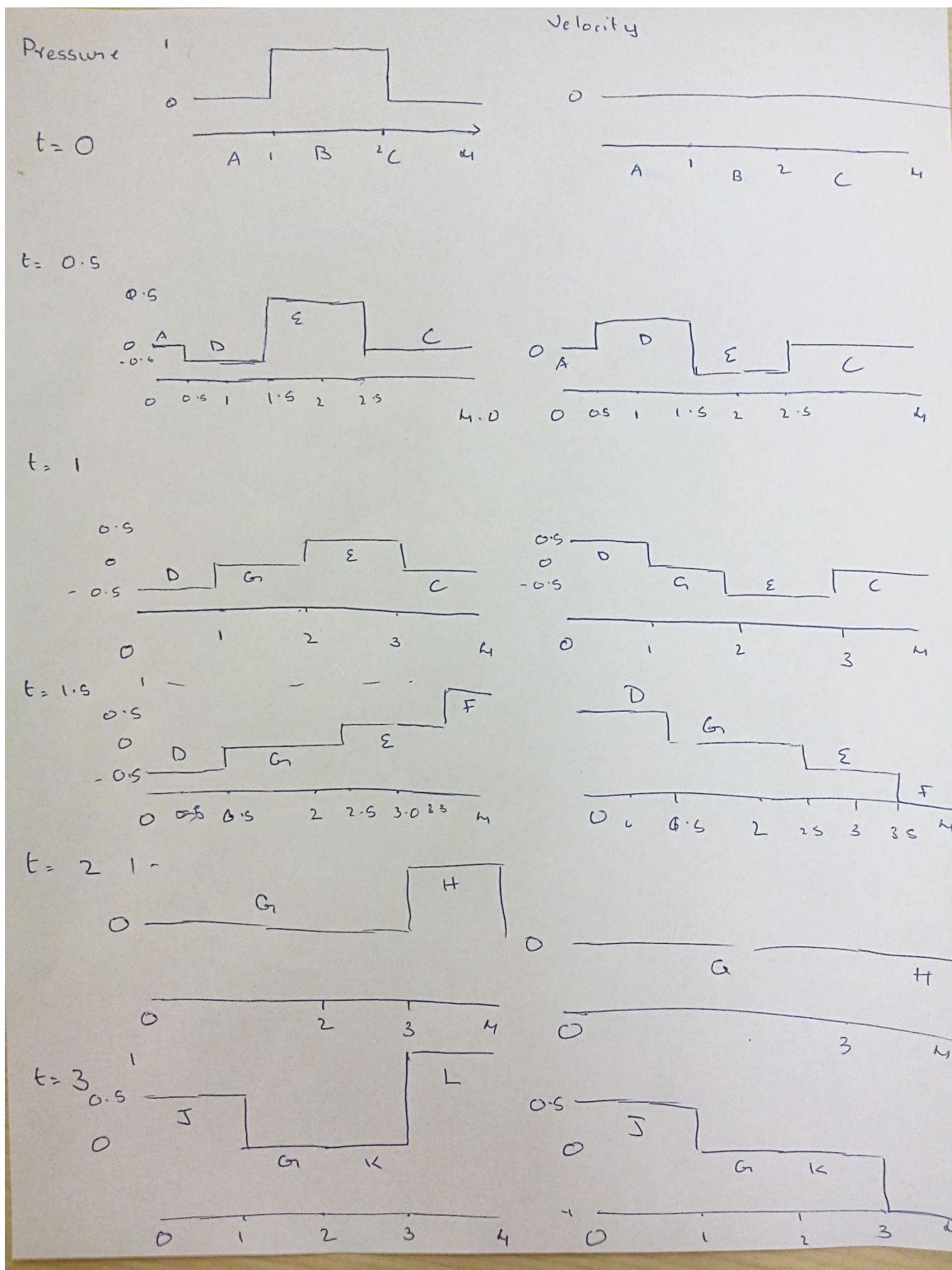
L:

$$L = I + W'_4 = I - W_4 = \begin{bmatrix} 1 \\ \frac{-1}{Z_0} \end{bmatrix}$$

M:

$$M = J + W'_2 = J - W_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Taking Z_0 as 1, the following graphs are sketched.



Problem #3.5A Solve #3.5 with *periodic* boundary conditions instead of reflecting walls. Sketch the solution in the $x-t$ plane up to at least time t_3 (as in #3.5 the time the right-going wave from $x_0 = 1$ hits the right boundary) and indicate the state in each section. You might

want to modify the script `problem_3_5.py` to make the plot.

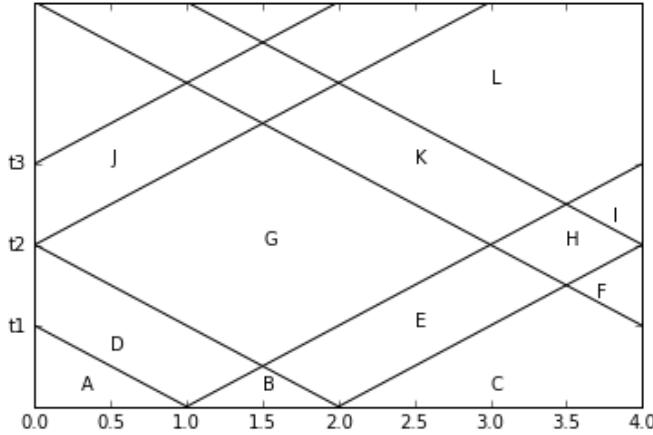
Solution:

With periodic boundary conditions, we get

$$W_2(0, t) = W_1(4, t)$$

$$W_1(4, t) = W_2(0, t)$$

And this following image is obtained.



A, B, C, D and E remain the same.

$$A = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad , B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad , C = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The eigenvalues are $\sqrt{K_0/\rho_0}$ and $-\sqrt{K_0/\rho_0}$.

$$t_1 = \frac{\Delta x}{-\lambda} = \sqrt{\frac{\rho_0}{K_0}}$$

$$t_2 = 2\sqrt{\frac{\rho_0}{K_0}}$$

$$t_3 = 3\sqrt{\frac{\rho_0}{K_0}}$$

The eigenvector matrix and it's inverse are

$$R = \begin{bmatrix} -Z_0 & Z_0 \\ 1 & 1 \end{bmatrix}, \quad , R^{-1} = \frac{1}{2Z_0} \begin{bmatrix} -1 & Z_0 \\ 1 & Z_0 \end{bmatrix} \quad ,$$

where

$$Z_0 = \sqrt{K_0\rho_0}$$

D:

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = R^{-1}(B - A) = \frac{1}{2Z_0} \begin{bmatrix} -1 & Z_0 \\ 1 & Z_0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{2Z_0} \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$W_1 = a_1 R_1 = \frac{1}{2Z_0} \begin{bmatrix} Z_0 \\ -1 \end{bmatrix}$$

$$W_2 = a_2 R_2 = \frac{1}{2Z_0} \begin{bmatrix} Z_0 \\ 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + a_1 R_1 = \begin{bmatrix} \frac{1}{2} \\ \frac{-1}{2Z_0} \end{bmatrix}$$

E:

$$\begin{bmatrix} a_3 \\ a_4 \end{bmatrix} = R^{-1}(C - B) = \frac{1}{2Z_0} \begin{bmatrix} -1 & Z_0 \\ 1 & Z_0 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \end{bmatrix} = \frac{1}{2Z_0} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$E = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + a_1 R_1 = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2Z_0} \end{bmatrix}$$

$$W_3 = a_3 R_1 = \frac{1}{2Z_0} \begin{bmatrix} -Z_0 \\ 1 \end{bmatrix}$$

$$W_4 = a_4 R_2 = \frac{1}{2Z_0} \begin{bmatrix} -Z_0 \\ -1 \end{bmatrix}$$

F:

$$F = C + W_1 = \frac{1}{2Z_0} \begin{bmatrix} Z_0 \\ -1 \end{bmatrix}$$

G:

$$G = D + W_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

H:

$$H = E + W_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

I:

$$I = H + W_3 = \frac{1}{2Z_0} \begin{bmatrix} Z_0 \\ 1 \end{bmatrix}$$

J:

$$J = G - W_4 = \frac{1}{2Z_0} \begin{bmatrix} Z_0 \\ 1 \end{bmatrix}$$

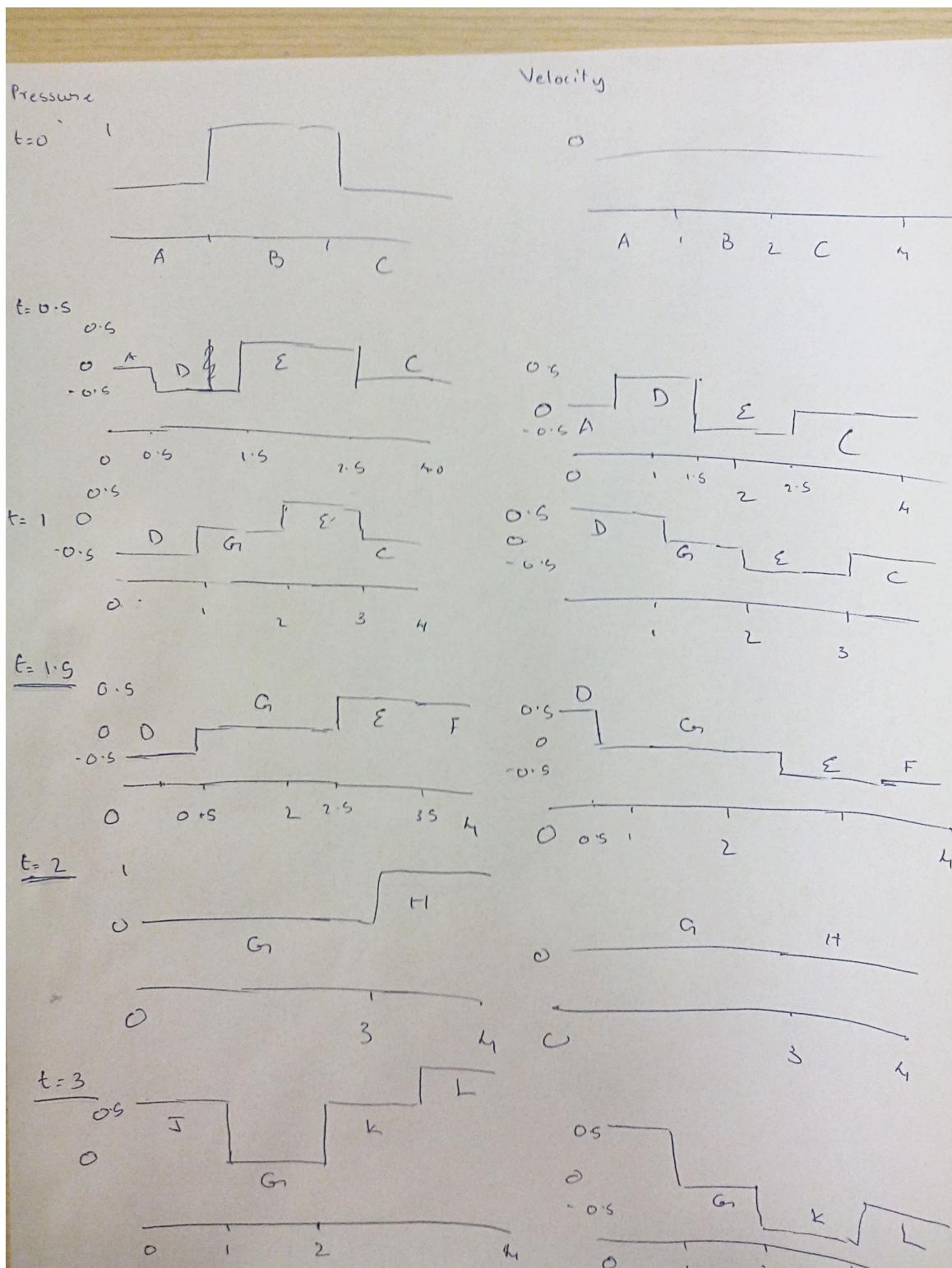
K:

$$K = G + W_1 = \frac{1}{2Z_0} \begin{bmatrix} Z_0 \\ -1 \end{bmatrix}$$

L:

$$L = K + W_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Taking Z=1, the following graphs are sketched. At t=0,0.5 and 1, the graphs are the same as before.



Problem #4.1 in the book

Solution:

$$\lambda_1 = u_0 - c_0$$

$$\lambda_2 = u_0 + c_0$$

- If the flow is subsonic, there is one positive eigenvalue and one negative one.

$$A^+ = R\lambda_2 R^{-1} = \begin{bmatrix} -Z_0 & Z_0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & (u_0 + c_0) \end{bmatrix} \frac{1}{2Z_0} \begin{bmatrix} -1 & Z_0 \\ 1 & Z_0 \end{bmatrix} = \frac{1}{2Z_0} \begin{bmatrix} Z_0\lambda_2 & Z_0^2\lambda_2 \\ \lambda_2 & Z_0\lambda_2 \end{bmatrix}$$

$$A^- = R\lambda_1 R^{-1} = \begin{bmatrix} -Z_0 & Z_0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} (u_0 - c_0) & 0 \\ 0 & 0 \end{bmatrix} \frac{1}{2Z_0} \begin{bmatrix} -1 & Z_0 \\ 1 & Z_0 \end{bmatrix} = \frac{1}{2Z_0} \begin{bmatrix} Z_0\lambda_1 & Z_0^2\lambda_1 \\ \lambda_1 & Z_0\lambda_1 \end{bmatrix}$$

If Q_i and Q_{i-1} are given, let

$$\Delta Q_{i-1/2} = Q_i - Q_{i-1}$$

$$W_{i-1/2}^1 = \frac{1}{\lambda_1} A^- \Delta Q_{i-1/2}$$

$$W_{i-1/2}^2 = \frac{1}{\lambda_2} A^+ \Delta Q_{i-1/2}$$

- If $u_0 > c_0$, both eigenvalues are positive,

$$A^+ = R\lambda R^{-1} = A$$

$$A^- = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

- If $u_0 < -c_0$,

$$A^+ = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$A^- = R\lambda R^{-1} = A$$

For the last two cases,

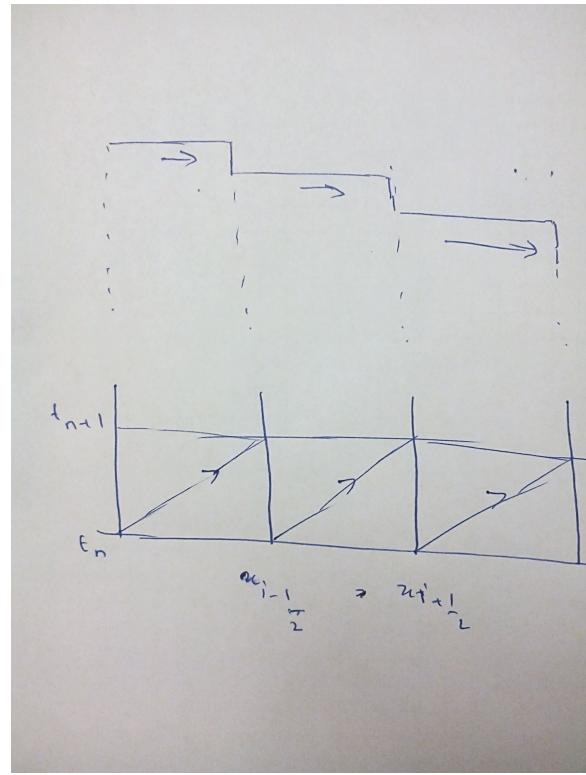
$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = R^{-1} \Delta Q_{i-1/2}$$

$$W_{i-1/2}^1 = a_1 R_1$$

$$W_{i-1/2}^2 = a_2 R_2$$

Problem #4.2 in the book

Solution:



(a)

K4. 2

b)
$$\begin{aligned} Q_i^{n+1} &= \frac{1}{2} (Q_{i-1}^n + Q_{i+1}^n) - \frac{\Delta t}{2\Delta x} (u Q_{i+1}^n - u Q_{i-1}^n) \\ &= \frac{1}{2} (Q_{i-1}^n + Q_{i+1}^n) - \frac{1}{2} (Q_{i+1}^n - Q_{i-1}^n) \\ &= Q_{i-1}^n \end{aligned}$$

∴ The Lax-Friedrich satisfies it.

Lax-Wendroff

$$\begin{aligned} Q_{i-\frac{1}{2}}^{n+\frac{1}{2}} &= \frac{1}{2} (Q_{i-1}^n + Q_i^n) - \frac{\Delta t}{2\Delta x} (u Q_i^n - u Q_{i-1}^n) \\ &= Q_{i-1}^n \\ \therefore Q_i^{n+1} &= Q_i^n - \frac{\Delta t}{\Delta x} (F_{i+\frac{1}{2}}^n - F_{i-\frac{1}{2}}^n) \\ &= Q_i^n - \frac{\Delta t}{\Delta x} (u Q_i^n - u Q_{i-1}^n) \\ &= Q_{i-1}^n \end{aligned}$$

∴ LW also satisfies it

c)

$$q_{i-\frac{1}{2}} = \frac{1}{2} \left[p_i + p_{i-1} \right]$$

Solving the
Riemann Problem

$$f_{i-\frac{1}{2}} = f(q_{i-\frac{1}{2}})$$

$$= \frac{c}{2} \left[p_i + p_{i-1} - (p_i - p_{i-1}) z_0 \right]$$

$$Q_i^{n+1} = Q_i^n - \frac{c \Delta t}{2 \Delta x} \left[p_{i+1} + p_i - p_i - p_{i-1} - (p_{i+1} - p_{i-1}) z_0 + (p_i - p_{i-1}) z_0 \right]$$

$$\begin{bmatrix} p_i^{n+1} \\ u_i^{n+1} \end{bmatrix} = \begin{bmatrix} p_i^n \\ 0 \end{bmatrix} - \frac{1}{2} \left[p_{i+1} - p_{i-1} - (p_{i+1} + p_{i-1}) z_0 + 2 p_i z_0 \right]$$

$$p_i^{n+1} = p_i^n - \frac{1}{2} (p_{i+1} - p_{i-1})$$

$$u_i^{n+1} = 0 - \frac{z_0}{2} (2 p_i - (p_{i+1} + p_{i-1}))$$

d) Possibly