AMath 574
Homework 2
Due by 11:00pm on January 26, 2023

For submission instructions, see:

`http://faculty.washington.edu/rjl/classes/am574w2023/homework2.html`

Problem #3.4 in the book:

Consider the acoustics equations,

$$\begin{bmatrix} p \\ u \end{bmatrix}_t + \begin{bmatrix} 0 & K_0 \\ 1/\rho_0 & 0 \end{bmatrix} \begin{bmatrix} p \\ u \end{bmatrix}_x = 0,$$

with

$$A = \begin{bmatrix} 0 & K_0 \\ 1/\rho_0 & 0 \end{bmatrix}, \quad \mathring{p}(x) = \begin{cases} 1 & \text{if } 1 \le x \le 2, \\ 0 & \text{otherwise,} \end{cases} \quad \mathring{u}(x) \equiv 0.$$

Find the solution for $t > 0$. This might model a popping balloon, for example (in one dimension).

Also sketch the solution in the x–t plane, labeling the states that appear in the solution. Then show where these states are in the p–u phase plane and how they are connected by eigenvectors.

Note that the script `problem_3_5.py` was used to generate the figure for the next problem and could be simplified for this problem if desired.

Solution:

First we calculate the eigenvalues λ_1 and λ_2 of A :

$$(\lambda_1, \lambda_2) = (-\sqrt{K_0/\rho_0}, \sqrt{K_0/\rho_0}) = (-c_0, c_0).$$

The eigenvalues are given by

$$r_1 = \begin{bmatrix} -\rho_0 c_0 \\ 1 \end{bmatrix} = \begin{bmatrix} -Z_0 \\ 1 \end{bmatrix}, \quad r_2 = \begin{bmatrix} \rho_0 c_0 \\ 1 \end{bmatrix} = \begin{bmatrix} Z_0 \\ 1 \end{bmatrix}.$$

Skipping calculations, and reading it from the text book: the solution is thus given by

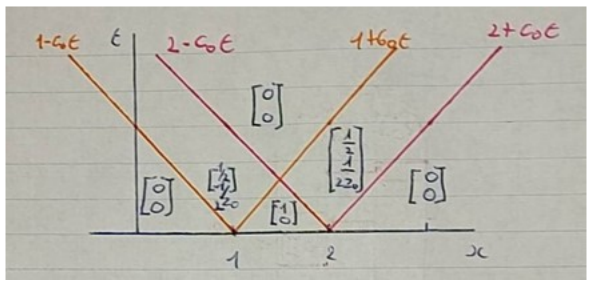
$$\begin{aligned} p(x,t) &= \frac{1}{2}[\mathring{p}(x + c_0 t) + \mathring{p}(x - c_0 t)] - \frac{Z_0}{2}[\mathring{u}(x + c_0 t) - \mathring{u}(x - c_0 t)], \\ u(x,t) &= -\frac{1}{2Z_0}[\mathring{p}(x + c_0 t) - \mathring{p}(x - c_0 t)] + \frac{1}{2}[\mathring{u}(x + c_0 t) + \mathring{u}(x - c_0 t)]. \end{aligned}$$

Therefore, assuming $c_0 \geq 0$

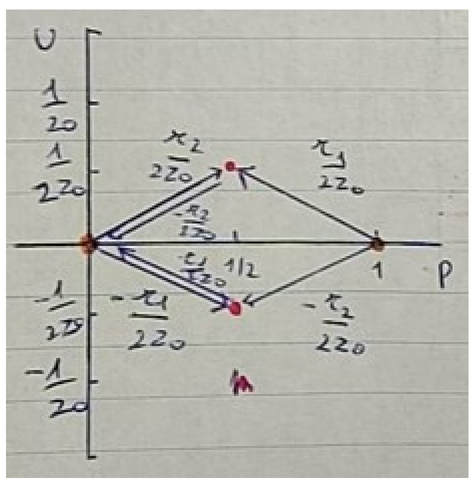
$$p(x,t) = \begin{cases} 1 & 1 + c_0 t \leq x \leq 2 - c_0 t \\ \frac{1}{2} & 1 - c_0 t \leq x < 1 + c_0 t | 2 - c_0 t < x \leq 2 + c_0 t \\ 0 & \text{elsewhere} \end{cases}$$

$$u(x,t) = \begin{cases} 0 & 1 + c_0 t \leq x \leq 2 - c_0 t \\ \frac{-1}{2Z_0} & 1 - c_0 t \leq x < 1 + c_0 t \\ \frac{1}{2Z_0} & 2 - c_0 t < x \leq 2 + c_0 t \\ 0 & \text{elsewhere} \end{cases}.$$

A sketch of the x–t plane is drawn as:



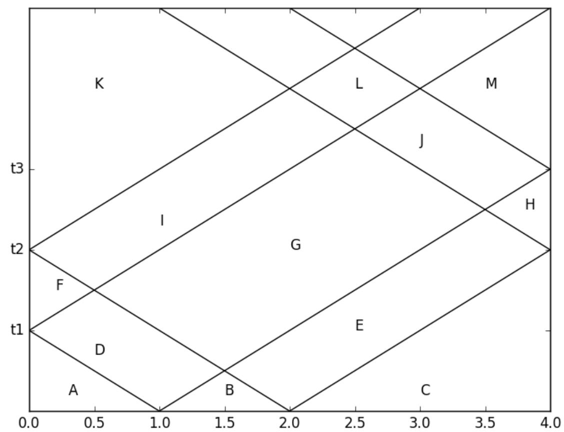
The p–u plane is given as:



Problem #3.5 in the book:

Solve the IBVP for the acoustics equations from Exercise 3.4 on the finite domain $0 \leq x \leq 4$ with boundary conditions $u(0,t) = u(4,t) = 0$ (solid walls). Sketch the solution (u and p as functions of x) at time $t = 0, 0.5, 1, 1.5, 2, 3$.

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



To solve this problem, determine the states $A,\ B,\ \ldots,\ M$ 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,

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

Hint: Another way to think about what happens when waves reflect off the boundary is suggested in Section 7.3.3.

Solution: Because of the boundary conditions, we already know what value u will hold in A, C, F, H, K and M. The bouncing off of the wall only comes into effect after t_1, therefore we can just use the results from last problem. I assume "using the acoustics equations from Exercise 3.4 on the finite domain $0 \le x \le 4$" also meant that we use the same initial value. Define $Z_0 = \sqrt{\rho_0 K_0}$, we get that

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

These are clear from the initial conditions. For D, G and E we have

$$D = \begin{bmatrix} \frac{1}{2} \\ \frac{-1}{2Z_0} \end{bmatrix}, \quad E = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2Z_0} \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Let's now look at the solution at time t_1, which is when $1 - c_0 t_1 = 0 \Rightarrow t_1 = \frac{1}{c_0}$. We can clearly see that the solution is then

$$\begin{bmatrix} p(x,t_1) \\ u(x,t_1) \end{bmatrix} = \begin{cases} D & 0 \le x \le 1 \\ G & 1 < x < 2 \\ E & 2 \le x \le 3 \\ C & 3 < x \le 4 \end{cases}$$

We now try to find F, or at least the p value. I will look at it from a physical standpoint. Here we can look at it as if the wave bounces back. Thus p stays the same, and u becomes zero for a while. Two parts of the wave are overlapping but with opposite direction, mathematically this is confirmed with the boundary value. Eventually the wave starts moving forward (once it caught up with its tail), thus velocity u is positive and takes on the same worth as before. This can also be explained in a more mathematical sense and this is what I got from Section 7.3.3. Thus we have that

$$F = \begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix}.$$

In I we have that the pressure p stays the same, but now the (shock)wave caught up with the other end and thus we notice a velocity which is positive. There is nothing to dissipate this velocity, thus it stays the same value as before. Therefore we have

$$I = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2Z_0} \end{bmatrix}.$$

The same kind of thing happens at H, the wave just bounces back and it seems like the velocity is zero for a while and then once the tail end caught up, we get a velocity u. However here the pressure moves backward, and thus u will be negative. Therefore,

$$H = \begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix} \text{ and } J = \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{2Z_0} \end{bmatrix}.$$

Note, like in A, C and G, that nothing really happens in K and M until the wave comes, thus we know that the pressure and velocity are zero:

$$K = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ and } M = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Eventually the waves combine again in L, they both came from different directions with the same velocity, that is why u will cancel out, thus being zero. The pressure will however combine and be 1. Therefore

$$L = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

In t_2, we know that $2 - c_0 t_2 = 0$, thus $t_2 = 2/c_0$. We can thus look at the image and conclude that

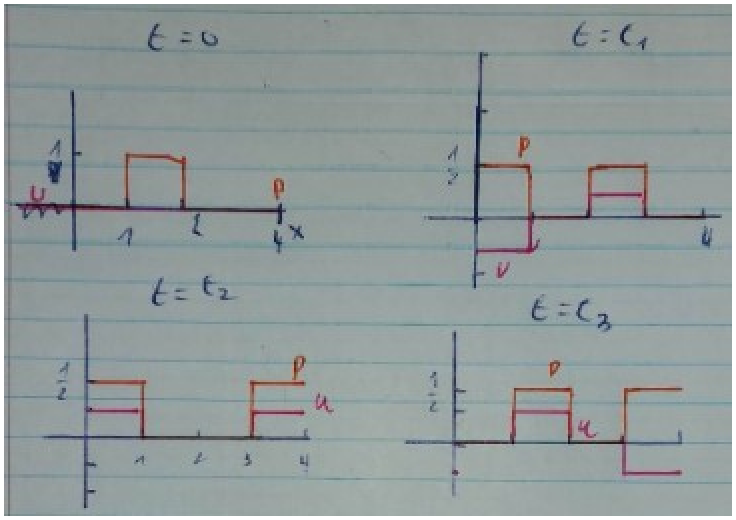
$$\begin{bmatrix} p(x,t_2) \\ u(x,t_2) \end{bmatrix} = \begin{cases} I & 0 \le x \le 1 \\ G & 1 < x < 3 \\ E & 3 \le x \le 4 \end{cases}.$$

The same can be done for $t_3 = 3/c_0$ $(1 + c_0 t_3 = 4)$:

$$\begin{bmatrix} p(x,t_3) \\ u(x,t_3) \end{bmatrix} = \begin{cases} K & 0 \le x < 1 \\ I & 1 \le x \le 2 \\ G & 2 < x < 3 \\ J & 3 \le x \le 4 \end{cases}.$$

Note that$A = C = G = K = M,\ B = L,\ F = H,\ E = I$ and $D = J$. This is all reasonable if we look at it from a physical perspective, plus we could go on to infinity like this.

Finally, a sketch of the functions at time $t = 0, t_1, t_2$ and t_3 can be found here:



Problem #3.8 in the book:

Consider a general hyperbolic system $q_t + Aq_x = 0$ in which $\lambda = 0$ is a simple or multiple eigenvalue. How many boundary conditions do we need to impose at each boundary in this case? As a specific example consider the system (3.33) in the case $u_0 = 0$.

Solution:

Let's say we have that $q \in V$, with $V \subset \mathbb{R}^d$. Thus $A \in M$ with $M \subset \mathbb{R}^{d\times d}$. If $\lambda = 0$ is a simple or multiple eigennvalue, we know that $\det(A) = 0$, it is thus singular. This will mean that either one or an infinite amount of solutions might exist. We know from the last homework set, that a unique solution can be found for a case with an eigenvalues which was zero, without boundary conditions. I believe that, if we have n eigenvalues which are zero, and have n unique linear independent eigenvectors to go along with this, we will have a unique solution without the need of boundary conditions. However, say that we do not have $n \neq d$ unique linear independent eigenvectors. For example the case:

$$A = \begin{bmatrix} 0 & K_0 & 0 & 0 \\ 1/\rho_0 & 0 & 0 & 0 \\ 0 & K_0 & 0 & 0 \\ 1/\rho_0 & 0 & 0 & 0 \end{bmatrix}.$$

Then an infinite amount of solutions exist. Making the intermediate states depends on the choice of the eigenvectors, and these are not unique, thus an infinite amount of solutions can exist. We can turn this into a unique solution by using boundary values. Say that we have k unique eigenvectors. I am unsure if we can even have unique eigenvectors in this case, if not, take $k = 0$. Nevertheless, then we will probably need $n - k$ or $2(n - k)$ boundary conditions to make up for this lack of uniqueness. The question now is "where?" and is it multiplied by 2?. From the perspective of solving a linear system of equation, we would guess that we just would need to limit $n-k$ function. By limiting, we mean putting boundary conditions on them. I believe that, depending on how many non-zero eigenvalues with different signs we have. if we have that they all have the same sign, the waves of discontinuouty would move in one direction, therefore we would only need boundary conditions on one side (the one the waves are moving towards). If they had different signs, we would however need boundary conditions on both sides. This is ofcourse assuming we cannot look into the past $t<0$. If we could, one would be enough per function that is going to be restricted.

Say that we have d eigenvalues which are zero, the jump will them happen only on 1 line: $x=0$. What happens on this line is not noticeable, thus we will act as if this solution is unique. No boundaries are needed. (example in homework 1)

Say that we have n eigenvalues of value 0, but only $k<n$ eigenvectors. This would mean that we would not be able to construct a basis for $\mathbb{R}^d$ and thus not be able to solve the problem for all initial conditions. However, if we would, the same as the last 2 paragraphs would hold. However, we should not be thinking of this as we are dealing with a hyperbolic system.

Problem #4.2 in the book:

If we apply the upwind method,

$$Q_i^{n+1} = Q_i^n - \frac{\bar{u}\Delta t}{\Delta x}(Q_i^n - Q_{i-1}^n),$$

to the advection equation $q_t + \bar{u}q_x = 0$ with $\bar{u} < 0$, and choose the time step so that $\bar{u}\Delta t = \Delta x$, then the method reduces to

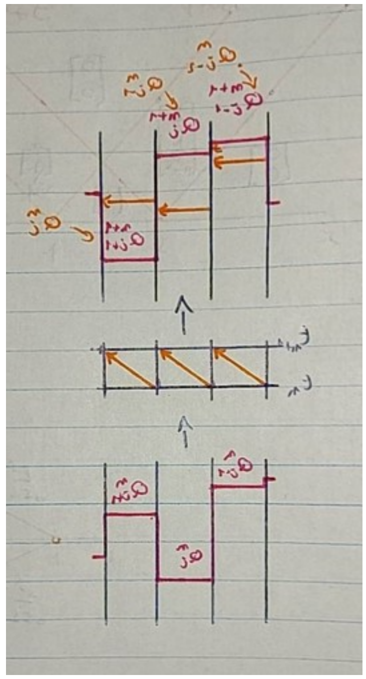
$$Q_i^{n+1} = Q_{i-1}^n.$$

The initial data simply shifts one grid cell each time step and the exact solution is obtained up to accuracy of the initial data. (If the data Q_i^0 is the exact cell average of $\mathring{q}(x)$, the n-the numerical solution will be the exact cell average for every step.) This is a nice property for a numerical method to have and is sometimes called the unit CFL condition.

1. Sketch figures analogous to Figure 4.5(a) for this case to illustrate the wave-propagation interpretation of this result.
2. Does the Lax-Friedrichs method (4.20) satisfy the unit CFL condition? Does the two-step Lax-Wendroff method of section 4.7?
3. Show that the exact solution (in the same sense as above) is also obtained for the constant-coefficient acoustics equations (2.50) with $u_0 = 0$ if we choose the time step so that $c_0\Delta t = \Delta x$ and apply Godunov's method. Determine the formulas for p_i^{n+1} and u_i^{n+1} that result in this case, and show how they are related to the solution obtained from characteristic theory.
4. Is it possible to obtain a similar exact result by a suitable choice of Δt in the case where $u_0 \neq 0$ in acoustics?

Solution:

1. The figure is the following:



   Excuse the flipped figure, I noticed last at the last moment that $\bar{u} < 0$

2. Noting that in the definition we have that $f(Q) = \bar{u}Q$, the flux. The Lax-Friedrichs method is given by,

$$Q_i^{n+1} = \frac{1}{2}(Q_{i-1}^n + Q_{i+1}^n) - \frac{\bar{u}\Delta t}{2\Delta x}[Q_{i+1}^n - Q_{i-1}^n].$$

   Taking $\bar{u}\Delta t = \Delta x$ clearly gives us

$$Q_i^{n+1} = Q_{i-1}^n.$$

   Thus, the Lax-Friedrichs method satisfies the unit CFL condition.

   The two-step Lax-Wendroff method is given by,

$$Q_{i-1/2}^{n+1/2} = \frac{1}{2}(Q_{i-1}^n + Q_{i+1}^n) - \frac{\bar{u}\Delta t}{2\Delta x}[Q_{i+1}^n - Q_{i-1}^n].$$

   Taking $\bar{u}\Delta t = \Delta x$ clearly gives us

$$Q_{i-1/2}^{n+1/2} = Q_{i-1}^n \Rightarrow Q_i^{n+1} = Q_{i-1/2}^{n+1/2} = Q_{i-1}^n.$$

   Thus, the Lax-Friedrichs method satisfies the unit CFL condition, it just takes two steps.

3. The constant coefficient acoustics equations with $u_0 = 0$ are given by

$$\begin{bmatrix} p \\ u \end{bmatrix}_t + \begin{bmatrix} 0 & K_0 \\ 1/\rho_0 & 0 \end{bmatrix} \begin{bmatrix} p \\ u \end{bmatrix}_x = 0.$$

We already know that the eigenvalues and eigenvectors again given by

$$\left(-c_0, \begin{bmatrix} -Z_0 \\ 1 \end{bmatrix}\right) \text{ and } \left(c_0, \begin{bmatrix} Z_0 \\ 1 \end{bmatrix}\right).$$

We now ave that

$$\begin{aligned}
A^+ &= \begin{bmatrix} -Z_0 & Z_0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & c_0 \end{bmatrix} \begin{bmatrix} -Z_0 & Z_0 \\ 1 & 1 \end{bmatrix}^{-1} = \frac{-1}{2Z_0} \begin{bmatrix} -Z_0 & Z_0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & c_0 \end{bmatrix} \begin{bmatrix} 1 & -Z_0 \\ -1 & -Z_0 \end{bmatrix} \\
&= \frac{1}{2Z_0} \begin{bmatrix} 0 & c_0 Z_0 \\ 0 & c_0 \end{bmatrix} \begin{bmatrix} -1 & Z_0 \\ 1 & Z_0 \end{bmatrix} = \frac{1}{2Z_0} \begin{bmatrix} c_0 Z_0 & c_0 Z_0^2 \\ c_0 & c_0 Z_0 \end{bmatrix} \\
&= \frac{c_0}{2} \begin{bmatrix} 1 & Z_0 \\ \frac{1}{Z_0} & 1 \end{bmatrix}
\end{aligned}$$

and

$$\begin{aligned}
A^- &= \begin{bmatrix} -Z_0 & Z_0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -c_0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -Z_0 & Z_0 \\ 1 & 1 \end{bmatrix}^{-1} = \frac{-1}{2Z_0} \begin{bmatrix} -Z_0 & Z_0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -c_0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & -Z_0 \\ -1 & -Z_0 \end{bmatrix} \\
&= \frac{1}{2Z_0} \begin{bmatrix} c_0 Z_0 & 0 \\ -c_0 & 0 \end{bmatrix} \begin{bmatrix} -1 & Z_0 \\ 1 & Z_0 \end{bmatrix} = \frac{1}{2Z_0} \begin{bmatrix} -c_0 Z_0 & c_0 Z_0^2 \\ c_0 & -c_0 Z_0 \end{bmatrix} \\
&= \frac{c_0}{2} \begin{bmatrix} -1 & Z_0 \\ \frac{1}{Z_0} & -1 \end{bmatrix}
\end{aligned}$$

Note that this is correct since,

$$A^+ + A^- = \begin{bmatrix} 0 & c_0 Z_0 \\ c_0/Z_0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & K_0 \\ 1/\rho_0 & 0 \end{bmatrix} = A.$$

The Godunov's method is now

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}[A^+ \Delta Q_{i-1/2} + A^- \Delta Q_{i+1/2}].$$

Filling in $\frac{\Delta t}{x} = \frac{1}{c_0}$, gives us

$$Q_i^{n+1} = Q_i^n - \frac{1}{c_0}[A^+ \Delta Q_{i-1/2} + A^- \Delta Q_{i+1/2}].$$

Thus,

$$\begin{aligned}
\begin{bmatrix} p_i^{n+1} \\ u_i^{n+1} \end{bmatrix} &= \begin{bmatrix} p_i^n \\ u_i^n \end{bmatrix} - \frac{1}{2}\left(\begin{bmatrix} 1 & Z_0 \\ 1/Z_0 & 1 \end{bmatrix} \begin{bmatrix} p_i^n - p_{i-1}^n \\ u_i^n - u_{i-1}^n \end{bmatrix} + \begin{bmatrix} -1 & Z_0 \\ 1/Z_0 & -1 \end{bmatrix} \begin{bmatrix} p_{i+1}^n - p_i^n \\ u_{i+1}^n - u_i^n \end{bmatrix} \right) \\
&= \begin{bmatrix} p_i^n - \frac{1}{2}\left(p_i^n - p_{i-1}^n + Z_0(u_i^n - u_{i-1}^n) - (p_{i+1}^n - p_i^n) + Z_0(u_{i+1}^n - u_i^n)\right) \\ u_i^n - \frac{1}{2}\left((p_i^n - p_{i-1}^n)/Z_0 + u_i^n - u_{i-1}^n + (p_{i+1}^n - p_i^n)/Z_0 - (u_{i+1}^n - u_i^n)\right) \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{2}\left(p_{i-1}^n + p_{i+1}^n + Z_0(u_{i-1}^n - u_{i+1}^n)\right) \\ \frac{1}{2}\left((p_{i-1}^n - p_{i+1}^n)/Z_0 + u_{i-1}^n + u_{i+1}^n\right) \end{bmatrix} \\
&= \frac{1}{2} \begin{bmatrix} p_{i-1}^n + p_{i+1}^n + Z_0(u_{i-1}^n - u_{i+1}^n) \\ (p_{i-1}^n - p_{i+1}^n)/Z_0 + u_{i-1}^n + u_{i+1}^n \end{bmatrix}
\end{aligned}$$

Remind yourself than the exact solution for p and u, given initial data, is given by

$$p(x,t) = \frac{1}{2}[\mathring{p}(x+c_0t) + \mathring{p}(x-c_0t)] - \frac{Z_0}{2}[\mathring{u}(x+c_0t) - \mathring{u}(x-c_0t)],$$
$$u(x,t) = -\frac{1}{2Z_0}[\mathring{p}(x+c_0t) - \mathring{p}(x-c_0t)] + \frac{1}{2}[\mathring{u}(x+c_0t) + \mathring{u}(x-c_0t)].$$

Since $c_0\Delta t = \Delta x$, (if x moves at c_0 time) we have that our approximation is exactly encompassing the exact solution at certain discreet steps.

4. if $u_0 \neq 0$, we have eigenvalues and eigenvectors

$$\left(u_0 - c_0, \begin{bmatrix} -Z_0 \\ 1 \end{bmatrix}\right) \text{ and } \left(u_0 + c_0, \begin{bmatrix} Z_0 \\ 1 \end{bmatrix}\right).$$

We could thus just substitute the c_0 in the case where $u_0 = 0$ with $u_0 + c_0$ in A^+:

$$A^+ = \frac{u_0 + c_0}{2}\begin{bmatrix} 1 & Z_0 \\ 1/Z_0 & 1 \end{bmatrix}.$$

A^- is a bit spicier, we calculate it as:

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

Note now that the reason (3) worked, was because we could divide away the c_0, here it is a bit more complicated than that since we have that we want to divide either $u_0 + c_0$ or $u_0 - c_0$. Doing one results in the other staying. There is thus no easy way to divide out the c_0 and u_0, which we don't have in the actual solution and must thus go away to get snippets of the actual solution. Note that we could only divide or multiply it out since we can only set $\frac{\Delta t}{\Delta x} = a$. It is thus not possible through this easy route. In conclusion: Not that I know of.

Problem #4.3 in the book:

Consider the following method for the advection equation with $\bar{u} > 0$:

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

1. Show that this method results from a wave-propagation algorithm of the sort illustrated in Figure 4.5(a) in the case where $\Delta x \leq \bar{u}\Delta t \leq 2\Delta x$, so that each wave propagates all the way through the adjacent cell and part way through the next.

2. Give an interpretation of this method based on linear interpolation, similar to what is illustrated in Figure 4.4(a).
3. Show that this method is exact if $\bar{u}\Delta t/\Delta x = 1$ or $\bar{u}\Delta t/\Delta x = 2$.
4. For what range of Courant numbers is the CFL condition satisfied for this method? (See also Exercise 8.6.)
5. Determine a method of this same type that works if each wave propagates through more than two cells but less than three, i.e., if $2\Delta x \leq \bar{u}\Delta t \leq 3\Delta x$.

Large-time-step methods of this type can also be applied, with limited succes, to nonlinear problems, e.g., ...

Solution:

1. Another way of writing the method is

$$Q_i^{n+1} = Q_i^n - (Q_i^n - Q_{i-2}^n) - \frac{\bar{u}\Delta t}{\Delta x}(Q_{i-1}^n - Q_{i-2}^n).$$

First it fully goes through the first two adjacent cells, but then it goes back a bit between the cells where Q_{i-1}^n and Q_{i-2}^n live. This comes down to what is asked in the question: it goes fully through the first adjacent cell and then partly through the second. I found it easier to look at it in this way.

2. Let's think of an Q_i^n as a value of a function $q(x_i - \bar{u}t_n)$ over a grid point and not the average over an element. We could then possibly interpolate the function q in order to find Q_i^n or Q_i^{n+1}. We could thus approximation Q_i^{n+1}, or the right hand side, by using Taylor/simple interpolation:

$$\begin{aligned} Q_{i-1}^n - \left(\frac{\bar{u}\Delta t}{\Delta x} - 1\right)(Q_{i-1}^n - Q_{i-2}^n) &\approx Q_i^n - \Delta x Q_i^{n\prime} - \left(\frac{\bar{u}\Delta t}{\Delta x} - 1\right)(Q_i^n - \Delta x Q_i^{n\prime} - Q_i^n + 2\Delta x Q_i^{n\prime}) \\ &= Q_i^n - \bar{u}\Delta t Q_i^{n\prime} \\ &\approx Q_i^{n+1}. \end{aligned}$$

In conclusion: this method holds when looking at it from an interpolation standpoint.

3. If $\bar{u}\Delta t/\Delta x = 1$, then

$$Q_i^{n+1} = Q_{i-1}^n,$$

which is clearly the solution since, if we go one time step further, we then have the Q of the previous spatial grid section.
If $\bar{u}\Delta t/\Delta x = 2$, then

$$Q_i^{n+1} = Q_{i-2}^n,$$

which is clearly the solution since, if we go one time step further, we then have that $\Delta x = 2\bar{u}$, which would mean that we should have the Q of 2 spatial elements ago. This is exactly what happens.

4. Since we get the previous results for $\Delta x \leq \bar{u}\Delta t \leq 2\Delta x$, we know that

$$1 \leq \frac{\bar{u}\Delta t}{\Delta x} \leq 2.$$

With these restrictions, don't overshoot the element where we want to be in. In conclusion: it holds for the CFL condition:

$$1 \leq \frac{\bar{u}\Delta t}{\Delta x} \leq 2.$$

5.

$$\begin{aligned} Q_i^{n+1} &= Q_i^n - (Q_i^n - Q_{i-1}^n) - (Q_{i-1}^n - Q_{i-2}^n) - \left(\frac{\bar{u}\Delta t - 2\Delta x}{\Delta x}\right)(Q_{i-2}^n - Q_{i-3}^n) \\ &= Q_{i-2}^n - \left(\frac{\bar{u}\Delta t}{\Delta x} - 2\right)(Q_{i-2}^n - Q_{i-3}^n). \end{aligned}$$

The 2 is merely there because of the choice of the domain of $\bar{u}\Delta t$. Filling in $\Delta x = 2\bar{u}\Delta t$ gives

$$Q_i^{n+1} = Q_{i-2}^n,$$

and filling in $\Delta x = 3\bar{u}\Delta t$ gives

$$Q_i^{n+1} = Q_{i-3}^n,$$

which is what we want.

Programming Problem.

The notebook in the class repository at `$AM574/notebooks/acoustics_godunov.ipynb` has been updated since class on 1/20 to make it a bit easier to modify for this problem. Please make sure you "git pull" to get the latest version. And please contact me if you are having problems with Jupyter of Python! We will do more with this same notebook in future homeworks.

(a) Implement a new function `LxF_step` similar to `Godunov_step` that takes a single step with the Lax-Friedrichs methods (for the same acoustics system):

$$Q_i^{n+1} = \frac{1}{2}(Q_{i-1}^n + Q_{i+1}^n) - \frac{\Delta t}{2\Delta x}A(Q_{i+1}^n - Q_{i-1}^n)$$

Since `Pn` and `Un` are separate arrays, you will have to implement the matrix-vector products implied in Lax-Friedrichs componentwise.

You might want to test with Courant number 1 first when debugging.

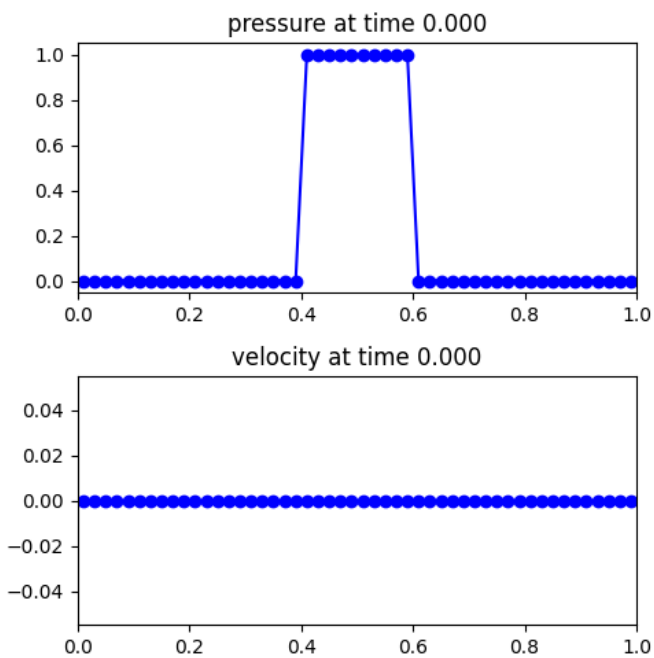
Submit your notebook that contains the implementation and also does at least the experiment described in part (b) below. You can include other tests too if you want.

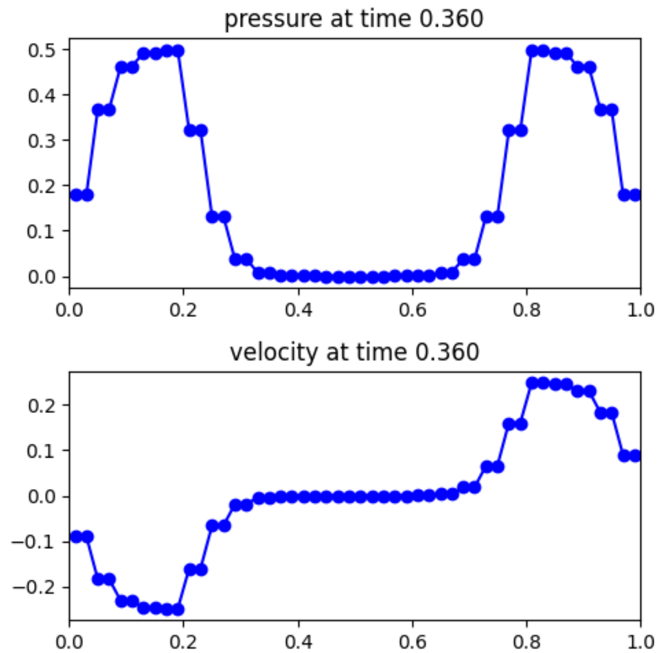
(b) When you do the following experiment:

In [55]:
```
num_cells = 50
dx = (xupper - xlower)/num_cells
x = arange(xlower-dx/2, xupper+dx, dx)
print('including 2 ghost cells, the grid has %i cells' % len(x))

P0 = where(logical_and(x>0.4,x<0.6), 1., 0.)
U0 = zeros(x.shape)

t0 = 0.
fig = figure(figsize=(5,5))
plotQ(P0,U0,t0)
savefig('plot1.png', bbox_inches='tight')

dt = 0.018
cfl = c0*dt/dx
nsteps = 20
tn = t0 + nsteps*dt
print('Using dt = %.4f, cfl = %.2f, taking %i steps to time %.3f' \
      % (dt,cfl,nsteps,tn))

Pn,Un = time_stepper(t0, P0, U0, dt, nsteps, LxF_step)
fig = figure(figsize=(5,5))
plotQ(Pn,Un,tn)
savefig('plot2.png', bbox_inches='tight')
```

```
including 2 ghost cells, the grid has 52 cells
Using dt = 0.0180, cfl = 0.90, taking 20 steps to time 0.360
```

you should get plots like these:





Explain why there are always two adjacent points with exactly the same value, for this particular choice of initial data.

Note that that I inserted `savefig` commands in the input to create the png files used here.

See `https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.savefig.html` for more about this command.

Solution:

(a) Attached with this pdf are the files required for this part.

(b) The code was ran with CFL condition being 1 at first, afterwards with it being .9 and then again with it being .8 (and eventually with .4). At first we did not see this grouping, but after changing the CFL condition the grouping happened and it stayed. Afterwards I ran the code for a smooth and two discontinuous intial conditions and this mostly showed no such grouping. The discontinuous initial conditions showed grouping if such pairs could be made per continuous piece. A continuous interval had an even amount of grid points. Thus I am pretty sure that it is because of the combination between CFL number, initial condition and gridpoints. The more difficult question however is "why?". My main hypothesis is that it has to do with the link between the CFL condition and the numerical domain of dependence. Two grid points, who form a couple, are just being updated based on the same range of points that they can access. They have the same numerical domain of dependence, therefore they are identical. In order to not make this homework giant, the examples can be found in the code.