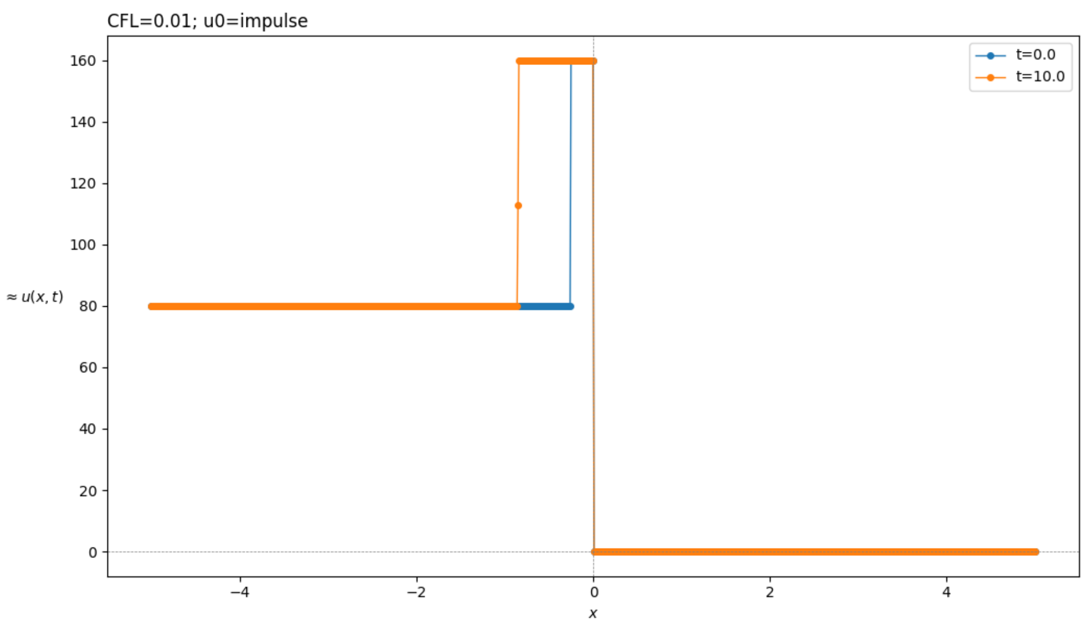
Question 1.1

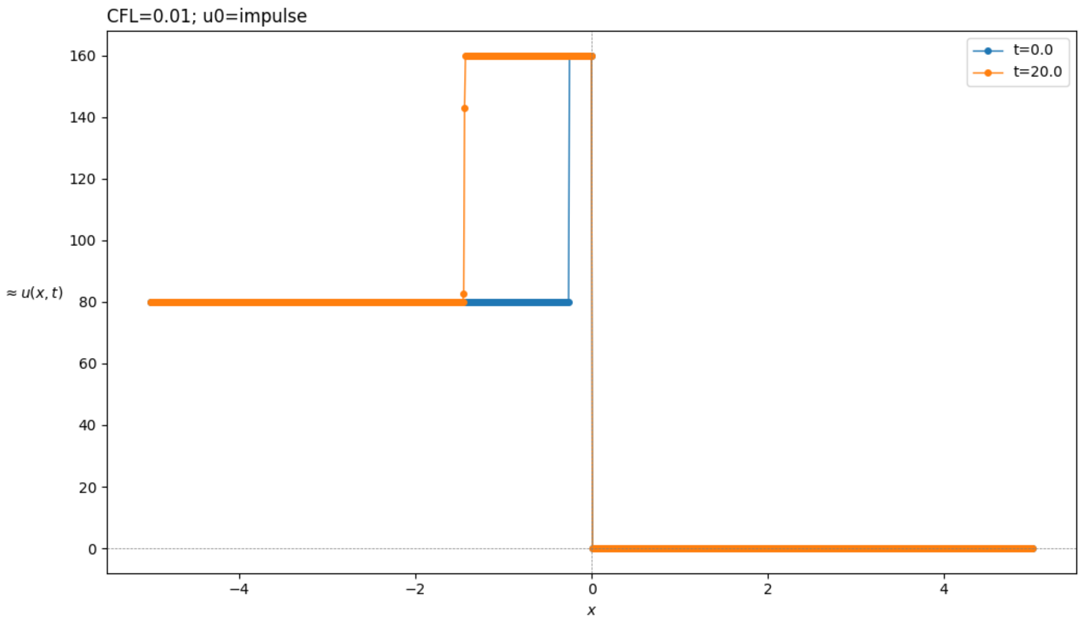
1. An upwind scheme works by evaluating the densities based on one direction of movement. In the Bungartz et al. equation, the formula focuses on the next time step (i+1) and grows the current point toward it. Anything before the time step (i-n) isn't calculated, which is why this is an upwind scheme. My implementation utilizes the code from the demo's upwind scheme and edits the items in the parenthesis, making it so that it is a difference between fractions of the two densities instead of just the difference of the two densities.

2.

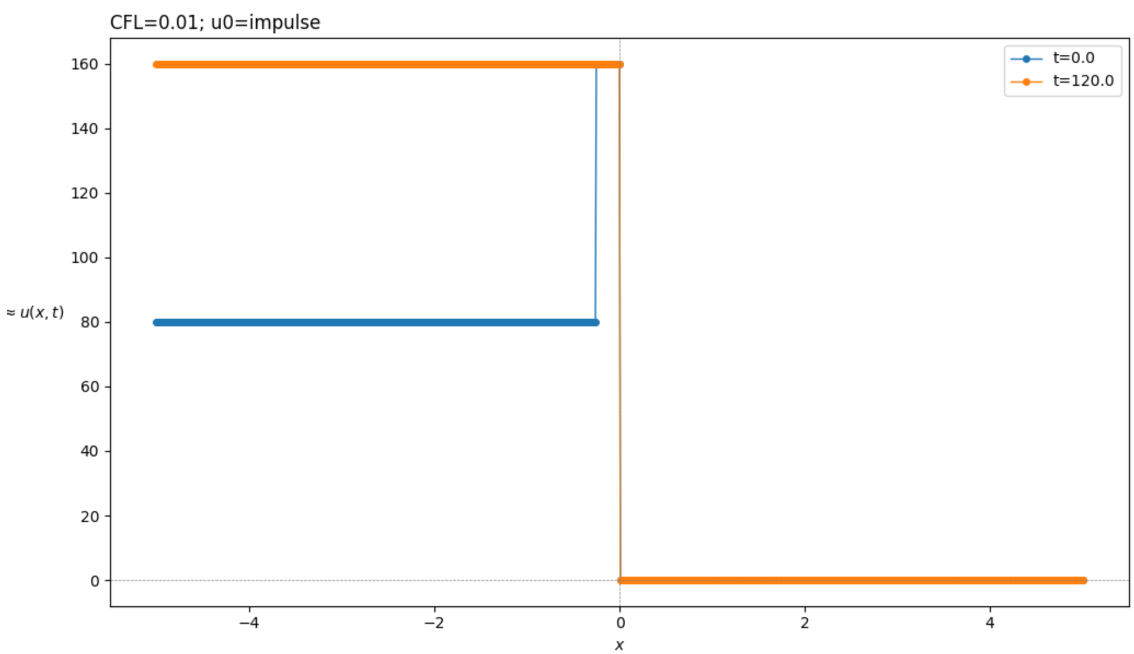
At time = 10s,



At time = 20s,



At time = 120s,



3.

From the formula, the Bungartz et al. formula basically looks at the next spatial step in the densities and adds a portion of it to the previous one (by subtracting the negative result of the current and next spatial steps). This result is negative because the second product inside the parentheses ends up being larger than the first (due to the 1-p_i,j/p_max decimal being larger for the smaller density).

The results for this formula aren't too reasonable for most traffic scenarios. In this scenario, there is a section of max density that then extends over a larger distance left over some time, with no changes to density on the right side. In most traffic scenarios (other than a blockage of all lanes at once), it is expected that the traffic at the front of the crowd moves forward, alleviating the max density at the front of the crowd. As such, it is expected that the density at x>0 starts increasing while the initial max density drops down which the upwind scheme does not show.

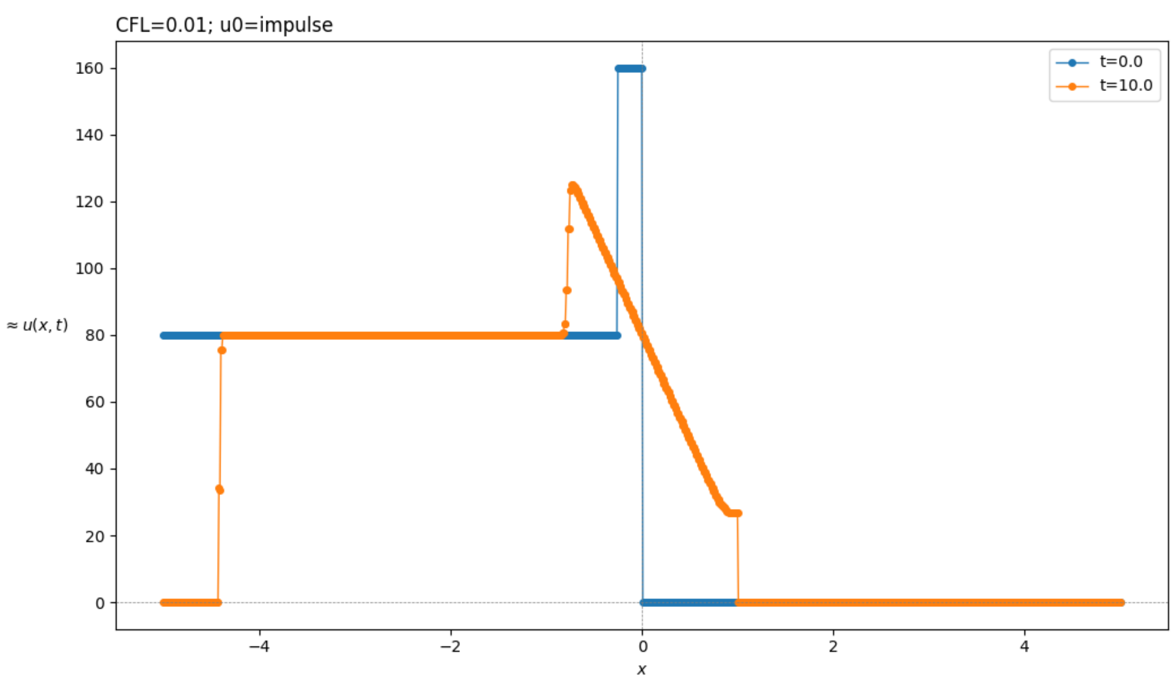
Question 1.2

1. The Lax-Freidrichs update formula basically computes an average of the surrounding densities to update the middle density. It focuses on the next and previous time steps and gets the average density and subtracts it by a fraction of the average flow between the points. My implementation (again) uses the demo's Lax-Friedrichs formula as a reference. The first fraction is from the demo's formula (same as in the instructions), and the second one replaces the flow functions with their formula based on max speed and density:

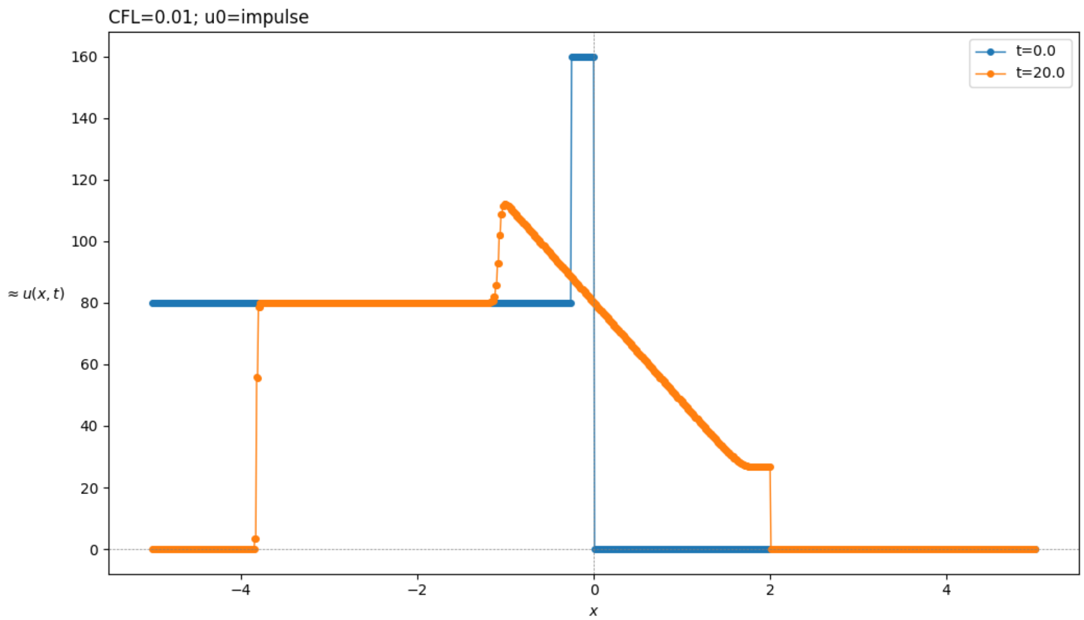
$$f(\rho) = v_{\max}\rho\left(1 - \frac{\rho}{\rho_{\max}}\right).$$

2.

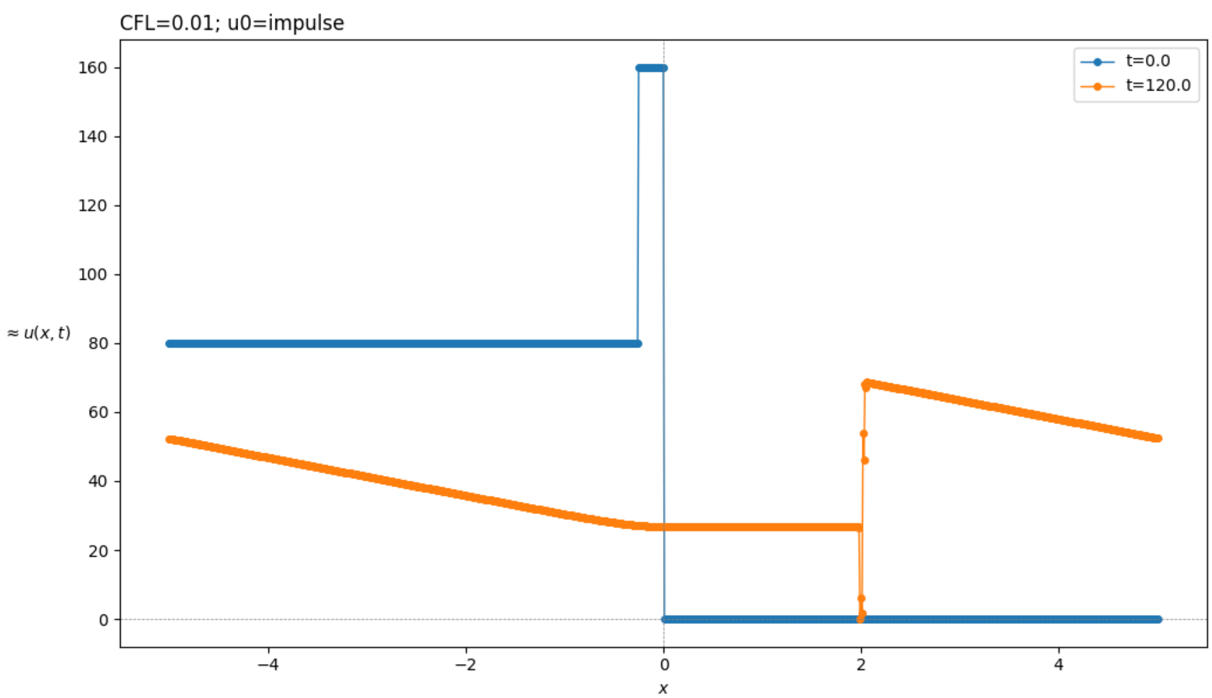
At time = 10s,



At time = 20s,



At time = 120s,



3.

Overall, these results seemed more reasonable for most traffic situations (ones that don't involve a blockage over all lanes). Since the Lax-Friedrichs scheme utilized averages for density updates, the algorithm looked at both sides around a point before updating it. This allowed the density spike to move out to both ends, increasing the low end and decreasing the spike. In addition, since the boundaries are circular, this means that the 0 density on the right spreads to the left end (which, if assuming the number of cars is fixed, means that those fixed cars moved on from the left area to the right). At the end of the simulation, it seemed like the initial densities spread out across the whole of the spatial domain, and if the simulation continued, the densities would even out after some time.

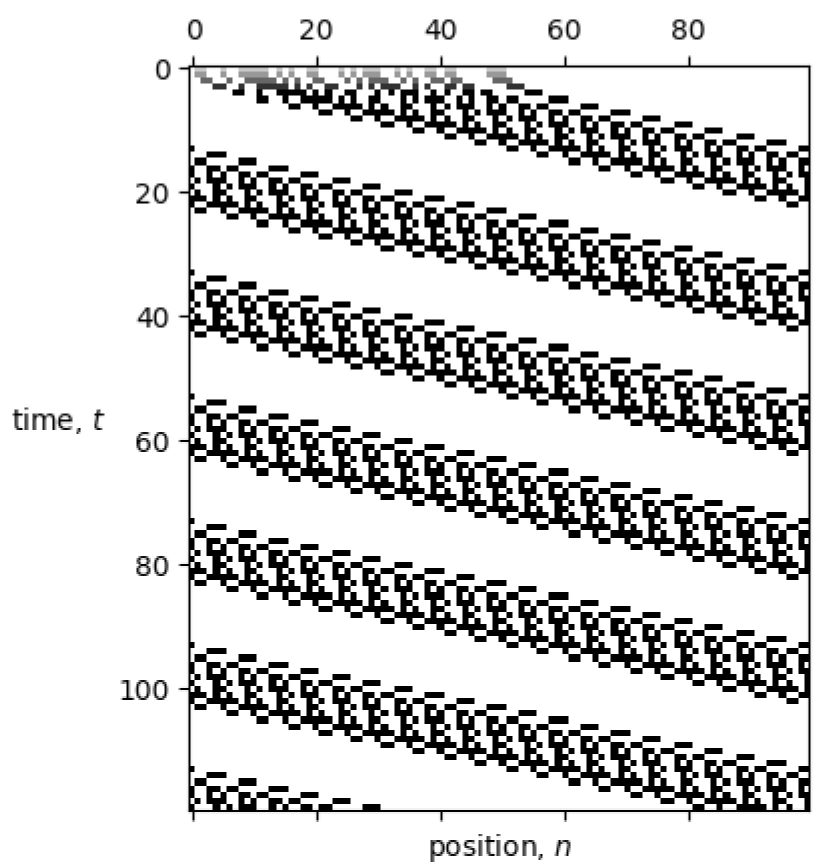
Question 2.1

1.

My implementation consists of a 100x120 grid, in which there are 100 cells for the position(each one being .1km to total 10km) and 120 cells for the time (1s time steps for 120s total time). At initialization, a certain grid cells are initiated to match the density shock in part 1 (to be exact, half of the positions up to -0.3 km (47 cells) are filled to show the density of 80, then all from -0.3 to 0km (3 cells) to show the density of 160, and none afterwards (50 cells) to show the 0 density). Unfortunately, I was unable to figure out proper acceleration/ deceleration and random starting speeds, and as such all vehicles start at the same speed and move as a single unit (showing the trend in the image below).

However, if I was able to figure out proper movement, the graph would change in that the cars would begin to cause jams at the vehicles with low initialized speeds. As the vehicles speed up, the front will begin to spread out while the back stays in the jam. Depending on the size of the jam, the front vehicles will loop around and join the back, however, over a long period of time, the massive jam will break up into constant movement with tiny slowdowns.

2.



3.

My results don't seem too reasonable to a realistic situation. My results depend on homogenous movement of vehicles (same, constant speed/ speed changes). In a real life situation, drivers have different preferences to speed and as such the vehicles will not move as a single object, making my results unreasonable (although, this may change in some futuristic world when everyone has an autonomous vehicle connected to a hive mind). Compared to the results in question 1, a density graph at a time t (like the ones in question 1) will end up looking like the original time, just translated a certain x-distance. It will not change its values because all vehicles move together, so the densities/units stay the same and only move side to side.

With a proper implementation, the results of the density graph would look more like the Lax-Friedrichs scheme in question 1.2.