CX 4230 - Mini Project 2

Eduardo Amorim

February 2023

Disclaimer: Part of the implementations were discussed with the students Reza Reimoo and Andrej Vrtanoski.

1 Introduction

The goal of this mini-project is to model the behavior of freeway traffic.

As stated in Bungartz et al. (2014), On the one hand, we want as much of it as possible. We want to be mobile and get as quickly and comfortably from the apartment to the university or to work, from the hometown to the vacation place or from shopping to the leisure activity. For this, we desire well-developed roads and parking places and convenient and immediately available public transportation.

On the other hand, we also want as little of it as possible. Road traffic should not bother us. We do not want to be in commuter traffic nor in the kilometer long traffic jams at the beginning and ending of holidays. We wish to be without the noise and exhaust emissions and prefer green corridors over asphalt streets.

Therefore, we will try to simulate the behavior of the world from different perspectives that will give insights on how traffic jams develop over time.

2 PDE Based Model

Our model follows the fundamental diagrams provided on the assignment:

$$\frac{\partial \rho}{\partial t} = -\frac{\partial f}{\partial x}$$

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

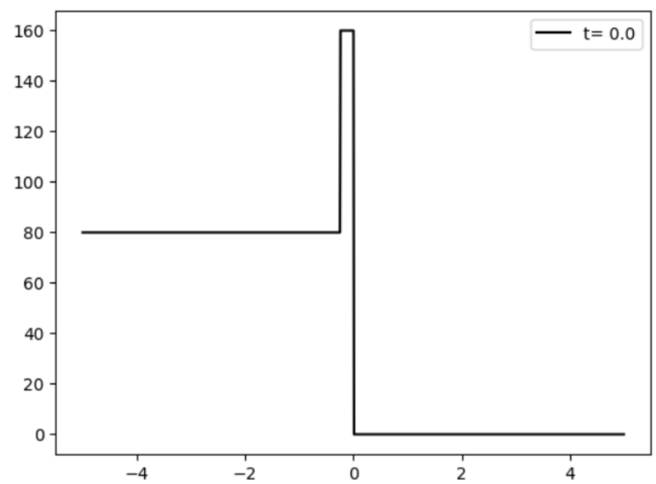
From these equations we derive:

$$\frac{\partial \rho}{\partial t} = -v_{max}\left(1 - 2\frac{\rho}{\rho_{max}}\right)\frac{\partial \rho}{\partial x}$$

2.1 Upwind scheme for an (artificial) shock

Our objective is to use an Upwind Scheme to numerically solve this equations. The validation of our model comes from substituting our function following the implementation described on the Chapter 7.4.1 of Bungartz et al. 2014 substituting the boundary values. Below is the update formula for the Upwind Scheme:

$$\rho_{i,j+1} = \rho_{i,j} - \frac{\delta t}{h} v_{max}((1 - \frac{\rho_{i+1,j}}{\rho_{max}})\rho_{i+1,j} - ((1 - \frac{\rho_{i,j}}{\rho_{max}})\rho_{i,j})$$



Now, we test our periodic bounds to validate our model. Note:

$$\frac{\delta t}{h} = 0.002778$$

$$v_{max} = 120$$

1. $\rho_{i,j} = 80$ and $\rho_{i+1,j} = 160$ Then,

$$\rho_{i,j+1} = 93.334$$

2. $\rho_{i,j} = 160$ and $\rho_{i+1,j} = 0$ Then,

$$\rho_{i,j+1} = 160$$

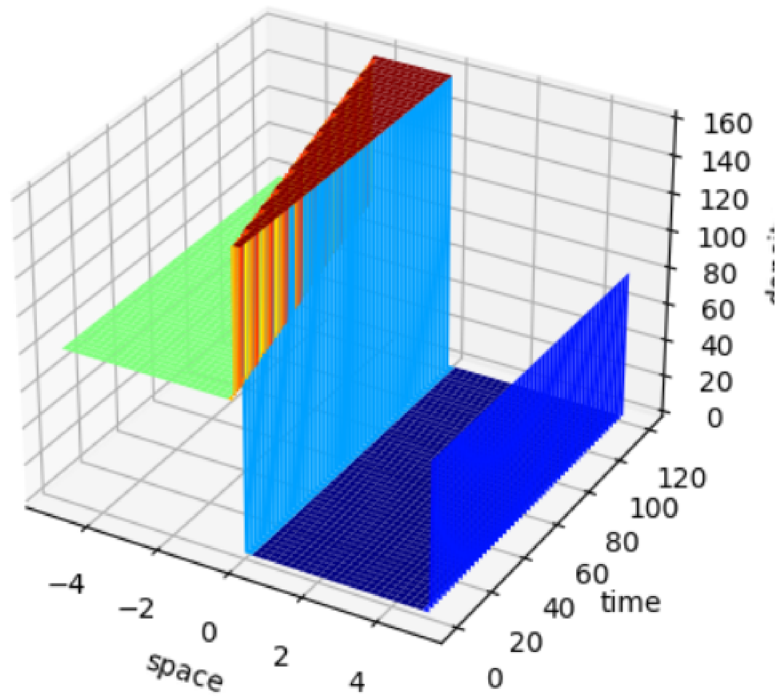
3. Now, recall that due to the periodic boundary condition asked in the problem makes our last value in the spatial iteration equal to the first position. Hence, $\rho_{n,j} = \rho_{0,j}, \ \forall j$. Therefore, let us consider the case $\rho_{i,j} = 0$ and $\rho_{I+1,j} = 80$. Then,

$$\rho_{i,j+1} = -13.334$$

By definition, a negative density must be 0. Hence,

$$\rho_{i,j+1} = 0$$

Generalizing to gather more data points we have:



These values along with the 3-D map are sufficient evidence of the validity of the model. It does not, however, provide evidence that it is an accurate representation of our system. We expect the system to dissipate the 'shock' gradually to both sides, and we can easily see from the plot that this is not the case for the upwind scheme. Therefore, we can conclude that it is not an accurate model representation of the traffic jam system.

2.2 Lax-Friederichs Scheme

With this scheme, we discretize our space and time to give approximations for the flow and density governed then by the following update formula:

$$\rho_{i,j+1} = \frac{\rho_{i+1,j} + \rho_{i-1,j}}{2} - \frac{h}{s}\frac{f_{i+1,j} - f_{i-1,j}}{2}$$

The process to validate the method is the same as in the Upwind Scheme, that is, test the boundaries where there are drastic changes on the system. Once again, note:

$$\frac{h}{s} = 0.002778$$

1. $rho_{i,j} = 80$ and $\rho_{i+1,j} = 160$ Then, applying the update formula we get:

$$\rho_{i,j+1} = 126.667$$

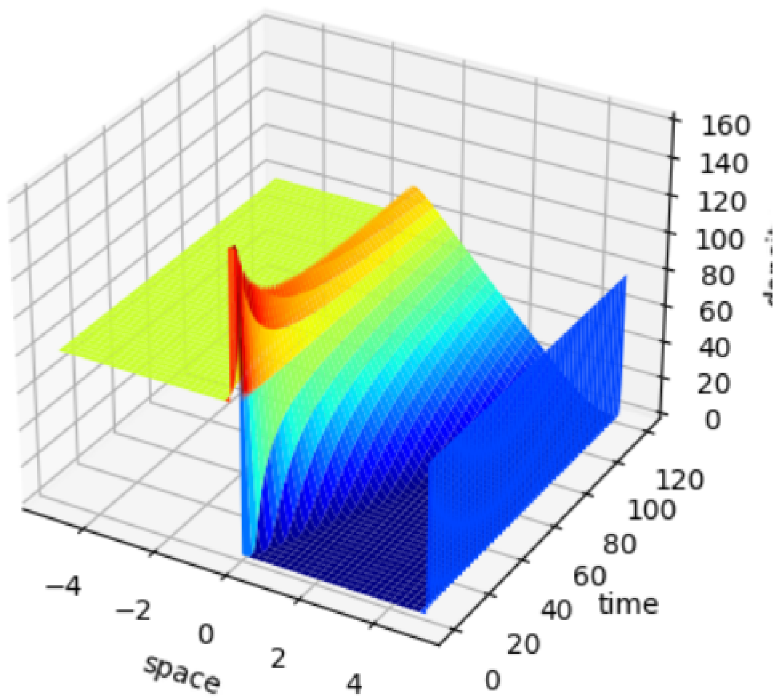
2. $\rho_{i,j} = 160$ and $\rho_{i+1,j} = 0$ Then, applying the update formula we get:

$$\rho_{i,j+1} = 80$$

3. Again, we will use the same periodic boundary scheme as the Upwind Scheme, and evaluate at $p_{i,j} = 0$ and $\rho_{i+1,j} = 80$ Then, applying the update formula we get:

$$\rho_{i,j+1} = 33.334$$

Now, the numerical solution given by the code is denoted on the following map:



We can see that this model dissipates the jam over time and differently from the upwind scheme, can be more closely related to a real world traffic jam. It has a better handle of the initial 'shock' by smoothing the evolution of the system through time.

3 Cellular Automata Based Model

3.1 Task 2.1

The implementation of the CA for this project can be labeled as an extension of Wolfram's 1-D nearest neighbor models for Cellular Automata. The biggest difference is on what is considered the neighborhood and how it affects the development of the highway section.
The implementation of this model is governed by the following rules:

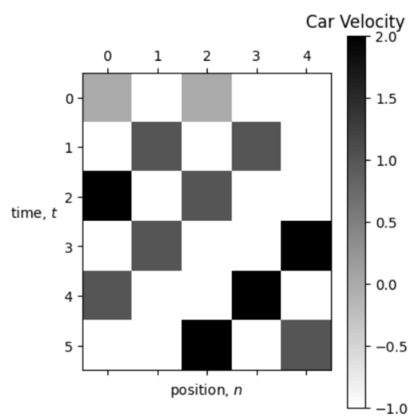
- The world is represented as a 1-Dimensional grid;
- Each car occupies a single 'cell' each time step;
- The system assumes that the world is collision and overtake free;
- We are simulating a circular road where there are no new cars entering the system nor leaving the system.

Now, let us describe the algorithm used for the evolution of the system:

Update for vehicle i :

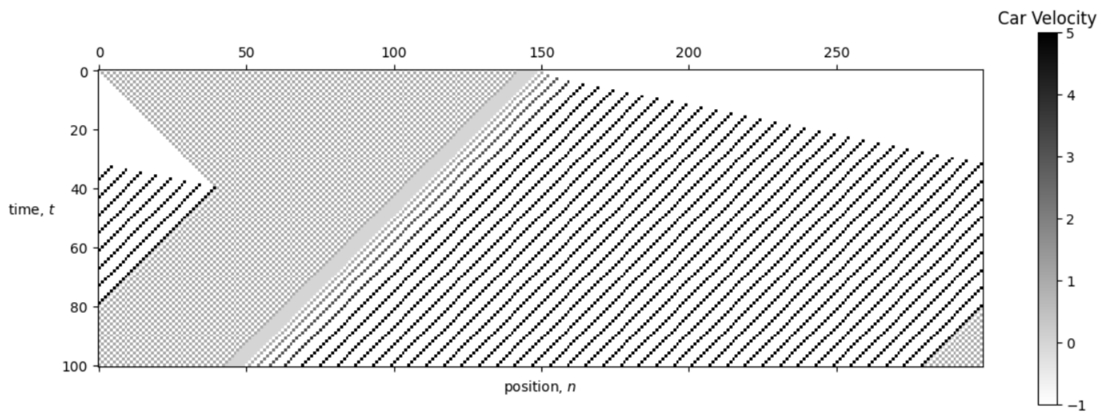
1. **Accelerate:** $v_i := \min\{v_i + 1, v_{max}\}$
2. **Decelerate:** $v_i := d(i, i+1)$, if $v_i > d(i, i+1)$
3. **Move:** vehicle i moves v_i cells forward

To validate the implementation, let's consider a smaller version of the world. Suppose we have two cars in a five-tile size grid. Then, we have:



Note how the implementation works since the evolution of the world is correct. If we follow the first car, his velocity starts at zero then since there is no car on the front, it accelerates using the first update for vehicle rule. Similarly, the

second car evolves according to the same rule.
Then, for the second iteration, we see that the first car cannot accelerate since it breaks the second update rule. Note that the same rule does not apply to the second car, since it does not have anything in front of it. Then, it accelerates and travels two tiles.
For the third iteration, we see that the roles are reversed since the second car has finished a lap. Then, the first car now has free road to accelerate using the first rule. The second car has do decelerate following the second rule.
This proves that our model works as expected and it is valid given the assumptions that we made above.
Now, let us generalize to compare with our PDE Model using relatively close parameters:



For a better visualization, refer to the notebook with the implementation.

The results follow the expectation given our constraints and model assumptions. Note the jam propagating to the left of its starting point, and that is reasonable, since we have the cars approaching and stopping at the back of the jam while the front has free way and is starting to accelerate. Also note that the propagation as a wave reflects closely our PDE model that had the shock. It is accurate given what we assumed it to be but it does not look very realistic. The behavior of drivers is not constant and we have to take into account the stochastic aspect of the real world. Hence, our next section.

3.2 Extra: Stochastic Extension

As stated in Bungartz et al. (2014): 'For this, the crucial idea in the NaSch-model was to introduce a randomization parameter ("dally factor") p and to introduce a further step in the transition function. This leads to an extension of the cellular automaton to a stochastic cellular automaton (SCA)':

Update for vehicle i :

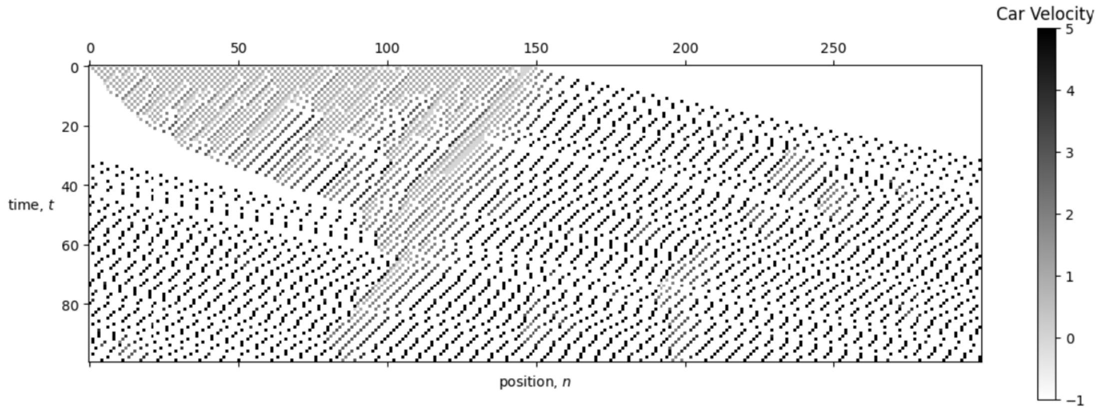
1. **Accelerate**: $v_i \coloneqq \min\{v_i + 1,\ v_{max}\}$
2. **Decelerate**: $v_i \coloneqq d(i, i+1)$, if $v_i > d(i, i+1)$
3. **Randomize**: $v_i \coloneqq \max\{v_i - 1,\ 0\}$ with probability $p < 1$
4. **Move**: vehicle i moves forward v_i cells

Also, following closely the model given by Bungartz et al. (2014):

1. Delay when accelerating: A vehicle that does not drive with maximum velocity v_{max} and has an open road, i.e., one that would theoretically accelerate, does not do so as soon as possible but only with some time delay to a later step. One notes that the velocity is not reduced in the acceleration phase. The acceleration just drags on a little longer. The vehicle will reach v_{max} eventually despite dallying. For example, $0 \to 1 \to 2 \to 3 \to 4 \to 5$ can become $0 \to 1 \to 1 \to 2 \to 2 \to 2 \to 3 \to 4 \to 4 \to 5$.
2. Dallying on an open road: Drivers who drive v_{max} for a longer period of time on an open road tend to not keep their velocity constant. Again the model excludes that a vehicle suddenly decelerates completely in several steps.
3. Overreaction when decelerating: A vehicle is hindered by a slower vehicle in front and must adjust its velocity accordingly. For example, drivers tend to hit the brakes too hard since they misjudge the distance or speed or because they drive too carefully with respect to what is deemed an optimal traffic behavior. The condition describing an underreaction that leads to rear end collisions is excluded in the model.

3.2.1 Using the same initial conditions as Task 2.1

Below is the CA for the stochastic extension of the model written above.



For a better visualization, visit the notebook with the implementation.

We can apply the fundamental diagram provided on the assignment to also validate our model.
Recall:

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

To determine the flow, one measures the number of vehicles that pass a determined section of the highway during a certain interval.

$$f = \frac{N}{\Delta T}$$

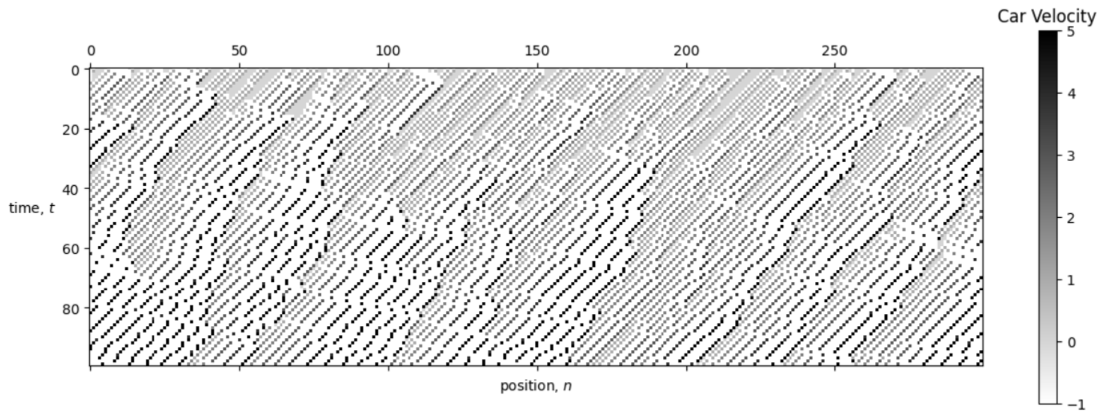
We then check our work by calculating the density using the following formula:

$$\rho = \frac{1}{m}\sum_{i=1}^{m}\frac{N_i}{L}$$

We can see from calculations on end points that this model reflects more accurately a real world representation of a traffic system and gives a dissipation of the jam overtime for a single cell but gives a more global perspective on the creation of new jams along the highway which could not be seen on the PDE model.

3.2.2 Random Initial Condition

Now, let us look at a random initial condition evolution of the world.



For a better visualization, visit the notebook with the implementation.