A Traffic Jam Model

Monte Carlo Simulations in Physics

Noora Mäkelä

3. kesäkuuta 2018

Table of contents

1 Introduction

Here the traffic jam simulation is done with the Kinetic Monte Carlo method. The system in this model is a one dimensional road, where cars are pointed with the letter X and the vacansies on the road are pointed with space. In the starting point there are cars in every other point on the road, so the car density is 0,5. The Kinetic Monte Carlo method is iterative, hence in every iteration one car is moved. In the method the jump rates of the car are calculated and in that way the method can decide what car to move. This method suites well this task of simulating a traffic jam. This is because in the system there are discrete events for which the jump rates can be calculated.

2 The tasks

2.1 The traffic Jam simulation and the methods

The Kinetic Monte Carlo method is an efficient method for simulation a traffic jam model. The main principle in the method is to calculate rates of every event. Those rates are used as a probability when events are chosen at random. Now a little more in depth description of the method.

The Kinetic Monte Carlo method starts with putting time to zero. The next move is to find all of the events. Here the events are the places on the road where is a car. So we can go trough every place on the road one by one and check if there is a car. We have to also calculate the rates of the events, which means that we have to calculate how many empty blocks of roads every car has in front of them. Then we can calculate the cumulative function:

$$R_i = \sum_{j=1}^{i} r_j.$$

Here r is the rate of the event. We have to save the value at the last index of the cumulative function to a variable, $R = R_N$. N is the number of events.

Next we have to find a uniform random number between 0 and 1. I have in my simulation generated the random number with Mersenne Twister random number generator. With this random number we can find an event to carry out. The event must satisfy:

$$R_{i-1} < uR \leq R_i.$$

When we have found this event, we can carry out that event. Then we have to find another uniform random number between 0 and 1. Then we can update time:

$$t = t + -\frac{log(u)}{R}.$$

The time increase for every step is calculated, and the average increase over many MC steps is physically correct.

Now if we have not reached the end of our simulation, we can go back to forming a list of possible events and calculating their rates and then carry on with the new iteration.

In my simulation I print the state of the road every 100 time units. For this assigment I have created three programs, one for the basic traffic jam simulation (trafficJam.cpp), one with a speed camera (trafficJamSpeedCamera.cpp) and one with a speed limit (trafficJamSpeedLimit.cpp). An example output of these programs can be following:

```
The road:
X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
100.014
  XXXXXXXXXXX  X X   X   X X XX     XX X XXXX   X X X   XXX XX X  X  XXXX  X  X   X X  X  XX  X  XXXXX  X  X  X  XXX X XXX XX X XXX  X   XX X   XX X
200.105
 X X XXX XX X X  X XX X  XX X XX   X  X X  X X X XXXX X X   XX  XXXXX  X XXX  X  X XX  X  X X X XXX  X X X XX XX X XX X  XX    X    XX X XX   XX X  X
300.107
X X X X  X  X X X  X XXX  X X  XX  X   X XX  XXX XX  X XXX X   X  XX X   X X   X X X  XX  X X XX X XXX X XXX    XXXXXX X X   X XX X X X XX X X XX   XX
400.118
  X X   X XXX XX  X XX X  X  XXXXX XX XX X  XX   X  X   XX X   XX  XX X X X XXX   X XX X X   X XX  XXX X  X  X X   X X   XXXXX   XXX  XXX XXX  X X XX
500.147
 X XXXX  X  XXX    X X X X XX XX   XX XXXXX  X XXXXX X X  X X X XXXXX    X XXX  X  X XXX  X  X   XX  XX  X  X  X  X XXXXX  XX XXX  X  X X   X X  X
600.165
 X X X    X   XXX XX X  X  X X X  X  X X  XXXXX XXX X  XX    XX X XXXX X   XX   XX  XX   XX X X XX  X XX XXX X XX XX  X X  X XXX X X X X   X XXXX  X  X X
700.185
X  X   X   XX X   X X X X X XXX  XX  XX   X  X XXX X  XX   X X  XXX  XXX XXX  X  XX    XX  XXXX    X X X  X  X  XXX  XXX XXXX  X  X  XXXXXXX  X XXX X
```

This is done with the basic traffic jam simulation program (trafficJam.cpp). First the program prints the initial state of the road. Then the simulation starts and the program prints first the time and then the state of the road at that moment. In the simulation the queues move in the opposite direction than the cars. So the cars move to the right and the queues here move to the left. This is expected because the queues grow from the left when cars come from that side. On the right side of the queue cars exit the queue, so it looks like the queue are moving to the left. This is just because they grow from that side. The queues move like density waves.

I also calculated an estimate of the velocity of the queue propagation, measured in arbitrary time units per lenght units. Myt estimate is about 0.9585 time units per lenght units. This was calculated by following the evolution of a queue and then comparing it's placement in different points of time.

2.2 The effect of the car density

The effect of the car density on traffic jam can be clearly seen. If the car density is 0.5, i.e every other spot on the road has a car, there can be seen queues after few thousand steps.

```
2000.46
 X X  XX   XX X X X XXX X  X   X X   XX  X   X   XX X X  XX X  X  X X XXX   X XX X    XXX XXX X XX  X XXX XXX X X   X XX XXX X   X  X X XX XX XX X XXX
2100.49
XXX X  X X X XXX X XXX   XXX  X X  X XXX  X XX   XX XXX  XX XXX   X XX     X  XX X      XX   X  X  X X X  X X  X XX XXXX XXX  X XXX XXX X  X XX X   X
2200.52
X X X  XXX X  XX X XX XX  X X  XX X X X X  X  X XXX XXXXXX  X X  X X X  X XXX X  X X XXX  X X X X X  XX   X  X X XXX X X X XX   XX X X  X    X   X XX
2300.54
  X XX X X  X X X X XX  X X  XXX X  X X  X  X X X X  XXX  X X X X X X XXX X XXX X X  X X   XXX   XX XX XX X XXX X  XX XX  X X X    X X  X  XXX XX X X
2400.56
XX X X  X  X X X X  X X  XX XX X    XXX XX  X  XX X X X X X X X  X  X XX X X  X  X  XX X   XX X  XXXX  X   XXXXXX XX  X X XX   XX  X X XX X  X XXXXX
```

Here first comes the time and then the state of the road at that time. Here my program has printed the states of the road every 100 steps. Here we can se that there are queues formed, so 0.5 is not the threshold density below which no queues appears.

Then we can try 0.4 as the car density. This means that we have 60 cars on a 150 long road.

```
2000.46
 X X  XX   XX X X X XXX X  X   X X   XX  X   X   XX X X  XX X  X  X X XXX   X XX X    XXX XXX X XX  X XXX XXX X X   X XX XXX X   X  X X XX XX XX X XXX
2100.49
XXX X  X X X XXX X XXX   XXX  X X  X XXX  X XX   XX XXX  XX XXX   X XX     X  XX X      XX   X  X  X X X  X X  X XX XXXX XXX  X XXX XXX X  X XX X   X
2200.52
X X X  XXX X  XX X XX XX  X X  XX X X X X  X  X XXX XXXXXX  X X  X X X  X XXX X  X X XXX  X X X X X  XX   X  X X XXX X X X XX   XX X X  X    X   X XX
2300.54
  X XX X X  X X X X XX  X X  XXX X  X X  X  X X X X  XXX  X X X X X X XXX X XXX X X  X X   XXX   XX XX XX X XXX X  XX XX  X X X    X X  X  XXX XX X X
2400.56
XX X X  X  X X X X  X X  XX XX X    XXX XX  X  XX X X X X X X X  X  X XX X X  X  X  XX X   XX X  XXXX  X   XXXXXX XX  X X XX   XX  X X XX X  X XXXXX
```

Here is the situation around 2000 steps with car density 0.4. Again we can see that queues appear. The situation also looks very similar with the situation with car density 0.5.

Now we can try even smaller car density, $\frac{1}{3} \approx 0.33333....$ The situation arounc 2000 steps is the following:

```
2000.42
 X  X  X  X  X   X X  X    X X     X  X X    X X  X X      X      X  X X  XX X X X X  XX X    X    XX   X XXX   X   X  X  X XX  XX X X  X      X  X
2100.43
 X  X X  X  XX    X  X  X   X X   X   X  X X    X X X X   X X X  X   X  X    X    XX   X  X    X  X   XX   X  X     X    X XX X XX   X   X  X   X XX X
2200.45
X X XX   X    X  X   X X  X X  X X    X  X   X  X   X     X XX   X   X X  X  X X    X XXX X  X  X  X    X X   X  X  X X    X X   X X   X X   X  X  X
2300.47
  XX X X X   XX     X   X   X   X  X XX   X   X      XX  X  X X X X  X   X X  X   X X X  X  X  X  X   X X    X    X  X   X   X    X XX X  X X  XX  X
2400.47
    X X  X  X X    X  X  X    X   X XX X  X    X XX  X    X   X X  X  X   X   X X XX X  X   X   X  X X   XX  X  X   X X X    X   X  X XX X    X    X X
```

Here we can see that there are not so many queues than with the larger car densities. The longest queues are now 2 cars. But still there are queues, so this is not the threshold density.

Now we try 0.3 as the car density. Now we get:

```
2000.42
 X  X  X  X  X   X X  X    X X     X  X X    X X  X X      X      X  X X  XX X X X X  XX X    X    XX   X XXX   X   X  X  X XX  XX X X  X      X  X
2100.43
 X  X X  XX    X  X  X   X X   X   X  X X    X X X X   X X X  X   X  X    X    XX   X  X    X  X   XX   X  X     X    XX X XX   X   X  X   X XX X
2200.45
X X XX   X    X  X   X X  X X  X X    X  X   X  X   X     X XX   X   X X  X  X X    X XXX X  X  X  X    X X   X  X  X X    X X   X X   X X   X  X  X
2300.47
  XX X X X   XX     X   X   X   X  X XX   X   X      XX  X  X X X X  X   X X  X   X X X  X  X  X  X   X X    X    X  X   X   X    X XX X  X X  XX  X
2400.47
    X X  X  X X    X  X  X    X   X XX X  X    X XX  X    X   X X  X  X   X   X X XX X  X   X   X  X X   XX  X  X   X X X    X   X  X XX X    X    X X
```

This is still not the threshold density, because there are queues of two cars.

Now we try 0.2 as the car density. The results is:

```
2000.24
   X     X   X        X  X      X    X  X  X    X    X       X    X     X    X   X  X      X     X  X    X X     X   X   X      X  X     X    X    X
2100.25
   X   X      X   X     X    X       X    X      X   X   X      X    X   X    X  X   X   X      X    X      X     X         X  X  X  X X  X    X  X
2200.27
  X     X    X   X     X     X    X     X       X     X      X       X X      X    XX   X    X    X     X   X       X    X    X      X  X  X    X  X X
2300.28
    X     X   X X    X     X      X    X     X       X  X       X     X   X         X   X      X  X  XX   X     X    X   X  X      X  X  X      X    X
2400.31
 X       X   X   X         X    X X  X       X     X     X   X  X   X    X    X    X     X    X   X    X   X    X  X    X   X   X    X     X     X
```

Now we can see that there are very few queues. At some points there even are not queues.

When we try $\frac{1}{6} \approx 0.166667$ we get the following results:

```
2000.35
X  X X      X   X       X        X       X        X  X       X  X   X  X       X    X        X     X     X    X   X        X       X  X       X
2100.35
    X       X  X    X   X     X        X     X     X      X       X       X       X    X    X  X     XX    X        X    X        X      X  X       X
2200.38
  X     X     X  X      X        X  X    X         X      X   X    X    X            X  X       X      X    X      X X        X     X    X     X     X
2300.41
      X   X   X     X   X     X          X     X      X      X      X  X     X    X     X       X X  X       X X      X     X      X   X           X
2400.45
     X      X   X       X    X     X X   X     X         X     X  X   X X     X     X      X   X      X   X          X     X  X            X        X
```

Now we can see that there are only one queue and the system does not change much. So this could our threshold density below which no queues appears. The density is now $\frac{1}{6} \approx 0.166667$.

If we want even lower density on which no queues appears, we can try $\frac{2}{15} \approx 0.133333333$.

```
2000.2
   X    X       X    X      X       X      X       X    X      X   X   X    X  X      X        X       X          X       X           X     X     X
2100.21
 X           X      X         X     X X         X    X     X        X   X      X     X    X             X     X       X     X   X     X   X       X
2200.23
X   X     X     X      X  X  X         X       X          X      X       X       X       X      X         X        X       X   X      X   X      X
2300.24
     X        X     X       X     X   X        X       X   X     X      X      X       X     X X       X       X       X      X       X      X       X
2400.24
      X  X      X       X    X     X         X       X     X       X       X         X   X  X    X     X           X  X       X     X      X      X
```

With this car density there are no queues. So $\frac{2}{15} \approx 0.133333333$ is our threshold density below which no queues appears. And also the situation on the road does not change much. We can also see that with larger car densities the queues are longer and with smaller car densities the queues are also shorter.

2.3 The effect of a speed camera

For this tasks I have made a c++ program (trafficJamSpeedCamera.cpp) that works like my original traffic jam simulation, but there is a speed camera on the road at index five. Cars at that index can have their jump rates smaller or equal than three. The threshold density that I found was $\frac{2}{15} \approx 0.133333333$, so now I will choose a density lower than that. The density I am going to use is 0.1. Now we can take a look what the speed camera causes:

```
2000.41
                                                 X                     X                           X            X      X X   X   X  X X   X X   X  X X
2100.43
                       X                     X                             X          X         X       X     X   X   X     X       X   X  X    X   X
2200.44
     X                                                            X                    X                X        X      X X   X   X X  X   X   X X   X
2300.49
X                                          X                        X                    X       X        X      X    X X   X   X     X  X    X  X
2400.5
                                  X                       X                   X                X           X       X    X  X     X  X  X   X  X   X X
X
```

It appears that there are no queues with the threshold density. Although the situation on the road looks different than without the speed camera. The cars are packed to the other side of the road, though there are no queues. The speed camera definitely has an impact on the traffic. But with low densities the speed camera is not able to produce queues that otherwise (without the camera) would not produce queues in my simulation.

2.4 The effect of a speed limit

For this tasks I have created a c++ program (trafficJamSpeedLimit.cpp) that works like my original traffic jam simulation, but now there is a speed limit. So now every car's jump rate can be lower or equal than three. I will here use the same car density as in the previous tasks, that density is 0.1. This car density is about 0.033333 smaller than the threshold density below which no queues appear.

```
2000.87
X                                           X  X                 X X      XX           X  X       X      X             X                 X   X      X
2100.97
X    X  X      X X              X      X    X                X              X                      X                 X          X               X  X
2201
  X    X                             X           X   X    X               X    X   X                X  X X          X                           X X
2301.03
          X      X       X  X     X             X X                   X  X               X    X  X              X         X           X
2401.05
           X              X    X           X      X             X    X                          X  X         X  X         X         X    X       X
```

With the car density 0.1 there are no queues with the speed limit. The cars move evenly on the road and the situation looks good, so in my simulation the speed limit is not able to cause queues that would not appear without the speed limit.

When I run the simulation longer, occasionally there are queues of two cars. But they disappear by the next iteration.

When the car density is bigger, for example 0.5, there are queues and those queues are still moving the left and the cars move to the right.

3 Instructions

Here I have three different c++ programs, the first is trafficJam.cpp that is made for the basic traffic jam simulation. The next program is trafficJamSpeedCamera.cpp, that is for the traffic jam simulation with a speed camera. The third program is trafficJamSpeedLimit.cpp, that is for the traffic jam simulation with a speed limit.

The programs can be compiled with

```
g++ -O3 -std=c++11 trafficJam.cpp
g++ -O3 -std=c++11 trafficJamSpeedCamera.cpp
g++ -O3 -std=c++11 trafficJamSpeedLimit.cpp
```

All of the programs can be ran with ./a.out .

4 Conclusions

Now we have seen that Kinetic Monte Carlo method is an efficient method for simulating a traffic jam. We have also seen that with bigger car densities there are more queues and with smaller car densities there are fewer or none queues. We also found the threshold car density below which no queues appear. That density was $\frac{2}{15} \approx 0.133333333$.

We have also studied the impact of a speed camera or a speed limit. We saw that with a car density lower than the threshold density, the speed camera had an impact on the traffic but there were no queues. The traffic was just more packed on the right side of the road. Also the speed limit with a car density lower than the threshold density did not cause any queues. With the speed limit the traffic went very smoothly.

Kinetic Monte Carlo was a very good choice for this tasks. It made an efficient program, because we had discrete events and their jump rates could be calculated.