Scientific computing – Traffic investigation

How to best reduce Traffic flow: a study of Traffic flow using a simple cellular automata model

Introduction:

To investigate the formation of traffic a simple model was constructed. Starting with the model devised by Kai Nagel and Michael Schreckenberg an algorithm was produced which modelled traffic by using an array to represent a road and the elements in the array represented both the position and velocity of the cars that would make up the traffic.

The figures from their paper were reproduced to check the code was functioning properly and that the results agreed.

Chart, line chart

Description automatically generated

Figure 1: A plot of flow rate against density replicating the results from the reference paper. The line represents the flow rate averaged over 75 iterations of the code. There is a strong peak when the density is 0.1, in this case 10 cars on a road of length 100. Before the peak there are few cars on the road they can travel freely without interacting with each other however as there are only a few cars the flow will be low. As more cars are added the flow increases linearly until the road is full enough the cars start to interact with each other. Cars will be brake as they approach other cars this means the average speed of the cars will fall so the flow rate will fall, eventually to 0 when the density is 1 as when all the positions on the road are filled no cars will be able to move so the flow will be zero.

Background pattern

Description automatically generated with low confidence

Figure 2: This is a time space plot showing the position of each car (shown by the red crosses) for each step of the algorithm, you can see how traffic jams appear to move backwards, this is with 10 cars and a length of 0.1 ,however if we alter the density we get a nice visual of how it effects the traffic

Then the necessary parameters needed for the investigation were found. A convergence study was performed to find the optimal length of road, where the length of the road didn’t unduly affect the simulation

Chart, histogram

Description automatically generated

Graphical user interface, chart, histogram

Description automatically generated

Figure 3 convergence study for length of road, vary length of L from 10 to 1000 for constant density and see what happens to the flow rate, we will choose the smallest value of length as the code scales the most with L. -s from 10 to 1000. We can see the road length has to be a minimum of 25 when the density is 0.2,

Figure 3b density vs optimal length plot, we run this code for each density and plot the minimum length needed to get a result that converges, it can be seen for low densities larger distances are needed to get an answer that converges, or that there is more variation the lower the density. The convergence happens around s =50 for most the listed densities

Chart, scatter chart

Description automatically generated

Figure 4 This is a study into initial transient period, the flow rate for the same density with varying initial transient periods will be found and we can see how it depends as a function of tmax\* some fraction of s,0 to 1 of s

The first parameter to be investigated was the maximum speed limit, vmax, this will affect how the traffic forms, the gradient of the flow rate diagram depends on vmax.

Chart

Description automatically generated

Fig5 vmax investigation, find flow rate for several vmax’s plot them against each other to find max for L used. There is a speed which will maximise the flow.

Add more lanes to see if that will increase the flow rate and reduce the number of traffic jams, using vmax, vary the number of lanes to the number of cars with a maximum value of 4-5 as this is the realistic number of lanes.

Chart, line chart

Description automatically generated

Fig6 varying number of lanes: two lane traffic

Now we can introduce the idea of overtaking and see how that effects the flowrate.

Chart, line chart

Description automatically generated

Fig7 flowrate when overtaking is allowed, also varying the number of lanes.

Then errors in the code can be quantified by looking into the spread of data caused by the random elements used. Possibly include earlier on and instead look at random overtaking.

Chart, line chart

Description automatically generated

Fig8 errors-investigating speeding

Then compare to a real-world situation with actual traffic data.

--each timestep is roughly 2 seconds,1 = 10km per hour ,2 = 2km per hour ecct the length of one cell is 5.5m-

- cars don’t settle in equilibrium when cars overtake,cars overtaking can cause traffic jams as overtaking is a purely random process