

Nick Gray

4226166

The Effect of Variable Speed Limits on Traffic Flow

## A traffic flow simulation using the Nagel-Schreckenberg cellular automation model

# Abstract

A computer programme was created in MATLAB™ to investigate traffic flow using the Nagel-Schreckenberg cellular automation method. The effect of the density of cars on the road upon traffic flow and the average speed of cars was looked at, as well as the transition to jamming. Results of these investigations were then compared to analytical values at the deterministic limit and previously calculated values from peer reviewed journals. It was shown that using variable speed limits impedes both the average speed of cars and the flux of cars moving along the road, however they do reduce the amount of stationary traffic on the road thus reducing the amount of jamming on the road. It was possible to set the speed limits in such a way to reduce the amount jamming without having a significant effect on the average speed and through flow of cars.

# Introduction

With over four million journeys on British motorways every day, a billion tonnes of cargo transported annually and with 98% of UK manufacturers saying that motorways have a critical impact on their business there are obviously major economic benefits to studying and improving traffic flow. [1] The government is currently investing in an £11 billion programme to build over 400 miles of smart motorway across England before 2020. Smart motorways use several methods to control the flow of traffic such as hard shoulder running, ramp metering and variable speed limits. As a result, I decided to look at the effects traffic flow and how variable speed limits can be used to manage it using the Nagel-Schreckenberg (NaSch) model.

The NaSch model is a cellular automation model for traffic flow that works in the following way. A road of length is broken up into cells in which only one car can occupy, and each car has an associated speed, , that can take integer values up to .Four simple rules are then used to simulate the flow of traffic on a road.

1. Acceleration: if the speed of a car is less than the maximum speed limit of the road . Then its speed is increased by 1 []
2. Slowing Down: If the distance between one cars and the car in front, , is smaller than the current speed, then the speed of the car is reduced to []
3. Randomisation: With probability , the speed of a car is reduced by 1 []
4. Movement: The position of each car is advanced by

These four steps are carried out in order, for all cars on the road in parallel, these actions are then repeated to simulate the traffic flow. [2] In the model the road is a close loop such that when a car reaches the end of the road in reappears with the same velocity at the start of the road. This means that there are periodic boundary conditions and the is a constant total density as cars don’t enter or exit the road.

A deterministic limit is reached when , as at this limit, all the cars will flow at up until a critical density, , is reached [3]. The critical density can be calculated from as shown in equation (i)

|  |  |
| --- | --- |
|  | (i) |

Below this limit the average speed decays with the limit that when the density is 1 then the average speed is zero, as at this point no cars can move as there will always be a car in the cell ahead of it. The maximum flow rate, , is also found at this critical value [4] and is given by

|  |  |
| --- | --- |
|  | (ii) |

For non-zero values of , as step 3 requires that some cars will randomly slow down, then the average speed will always be less than , thus the critical density and the peak flow rate are lower.

The final element of traffic flow considered was the effect of jamming, there are numerous possible definitions of jamming, such as the number of cars that are stationary (), the number of cars travelling below the speed limit () and the number of cars with . [5] All of these methods are order parameters so they are equal to 0 at and equal to 1 at . In this investigation, the jamming parameter was defined as the fraction of cars with . At the limit , for all values of , the parameter will be zero for . As the flow is deterministic, there will always be a space of at least one cell 1 between all of the cars, therefore, there will be no stationary vehicles thus the jamming parameter will be equal to zero. At for non-zero values of , the minimum value of jamming will be equal to the probability of slowing as this fraction of cars will be randomly slowed and therefore be stationary.

# The Model

The code to run the traffic flow simulation was created using the *MATLAB™* programming language using build number *2016b*. The code made use of the *Parallel and GPU Computing Toolbox*.

Firstly, the conditions were encoded as variables. as *“maxSpeed”,* the density of cars on the road *as “density”*, the probability of each car slowing as *“prob”* and the road length as *“roadLength”.*

For the initial set-up of the cars on the road, the code calculates the number of cars () on the road from the density and the length of the road. It then creates an array of size , *“cars”*, in which the first column contains the position coordinate of the car and the second contains the speed of each cars. For the initial set up the cars are evenly spaced along the road and have speed equal to the max speed of the road. The cars array is arranged such that the car nearest the start of the road is the first car in the matrix. The programme then simulates the flow of traffic on the road.

The cars are passed into a function entitled *“carRules.m”,* in which the four NaSch rules are applied to each car. Firstly, the function splits the *cars* array into position and speed, it then calculated the distance between all the cars and uses to apply the rules. The code uses a loop to calculate the new position and velocity value for each car, this is different to

the original NaSch method which applies the rules in parallel for all cars, however the method used here is equivalent to parallel computation as each car is referencing the old position and speed, rather than any updated values. The function also contains an if statement to prevent cars obtaining a negative velocity, which occurs if there has been an error and two vehicles ended up with the same position coordinates, this is added in between steps three and four of NaSch. If the new position of the car has exceeded the road length then its new position is then given by subtracting the road length from the calculated position, this is done using an if statement at the end of the programme, this if statement also calculated the number of cars that have exited the loop. The function then recombines the position and speed of each car into a new *cars* array, before outputting.

The programme the sorts the outputted cars array so that the ordering of the cars is maintained. To improve efficiency over sorting the array after each iteration, the function *“shuffle.m”* was created, it takes an array and an integer () and then moves the last rows of the array to the start.

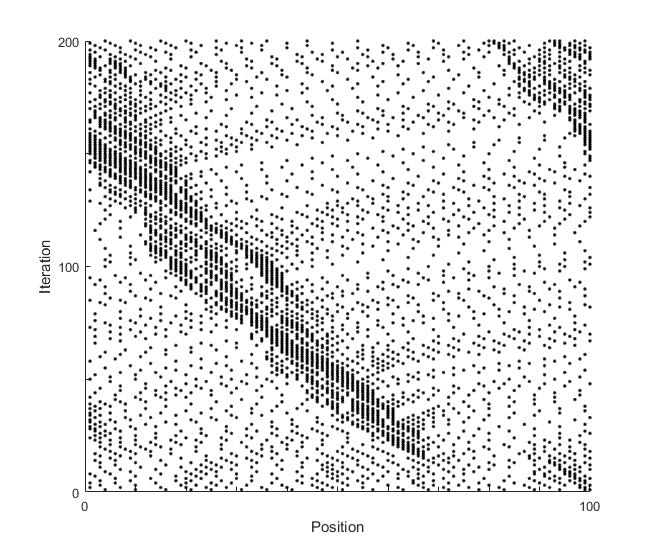


Figure 1: Visualisation of a simulation with 100 iterations. each black dot represents the position of a car. the parameters used are:

This process this then repeated for as many iterations, or time steps, as required using a for loop. Figure 1 shows a simulation for 100 iterations, jamming can be clearly seen and the effect of traffic jams moving backwards down the road can also been seen. This is version 1 of the code.

In order to look at the dynamics of traffic flow more closely, the code was modified to collect information about the flow, average speed of cars and jamming. As the flow is the number of cars passing a point in the road and information about cars passing the end of the road has already been coded this point was used. The code then worked out the average speed of the cars and did a search to find the number of cars with , and therefore worked out the jamming parameter accordingly. For efficiency, these data collection arrays were preallocated. The mean of these values was then calculated after the simulation loop was over. For loops were then added to the programme to calculate the traffic conditions for a range of densities, max speeds and slowing probabilities. As the density loop is the largest it was executed in parallel to reduce the total computational time, using a 4 core CPU it reduced the computation time by a factor of 4.

For the data collection run the following parameters were used. The road length was 10000, the densities chosen were between 0 and 1 in steps of 0.01, the values of were integers between 1 and 5 inclusive, and the probabilities were chosen were 0, 0.3, 0.7. The simulation was run for 105 iterations, this is an order of magnitude less than that in [2] but there was only a limited amount of computation time available this value was deemed to produce data of sufficient quality without taking too long to compute.

Graphs of density against average speed and density against flow rate were then plotted, with the critical density and compared to that of the analytical results in equations (i) and (ii) respectively. These results can see seen in Figure 2 and Figure 3. In both these figures, the analytical critical density and peak flow match the calculated results.

Having confirmed the results at the deterministic limit, the effect of having a non-zero probability was then explored. The values and were chosen as these provide insights for both low and high probabilistic behaviour. The results of this can be seen in Figure 4 and Figure 5. The expected result that the average speed and flow rate drops can be observed in these graphs. Both these figures agree with those in [4] [6] [7] .

In order to show the effect of jamming, plots were made of density against the fraction of cars that are stationary. Figure 6 shows that in the deterministic where the max speed has no effect on the number of stationary of cars. Figure 7 shows that for higher speed limits the transition to jamming occurs sooner for higher values of , apart from the case where , where the lowest fraction value is that of the slowing probability. These results agree with what was expected when thinking about it analytically and those in [6].

As all the above results match the analytical values and those found in previous research, we can be confident that the programme is accurate.

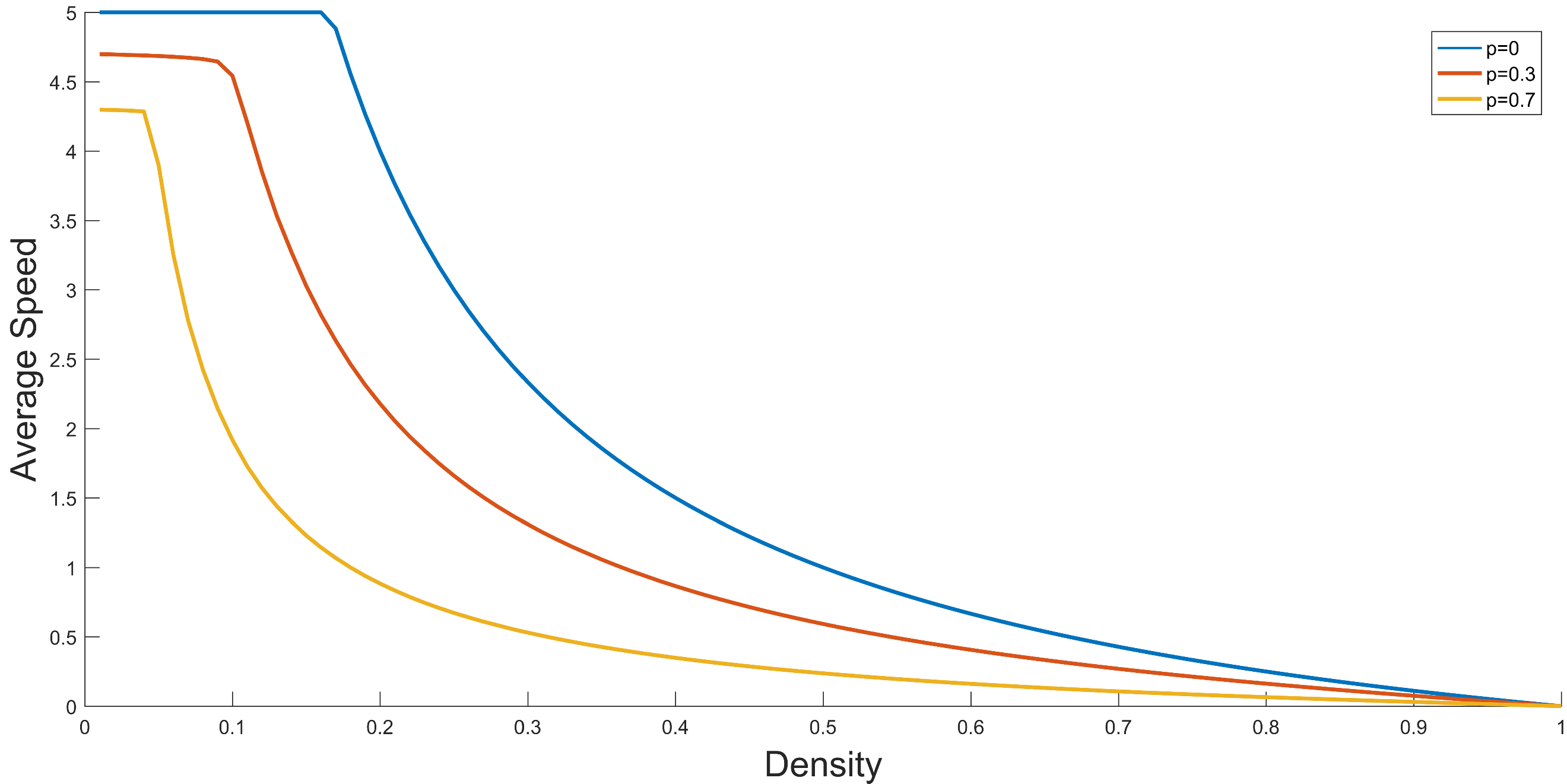


Figure 4: Plot of density v average speed for various .

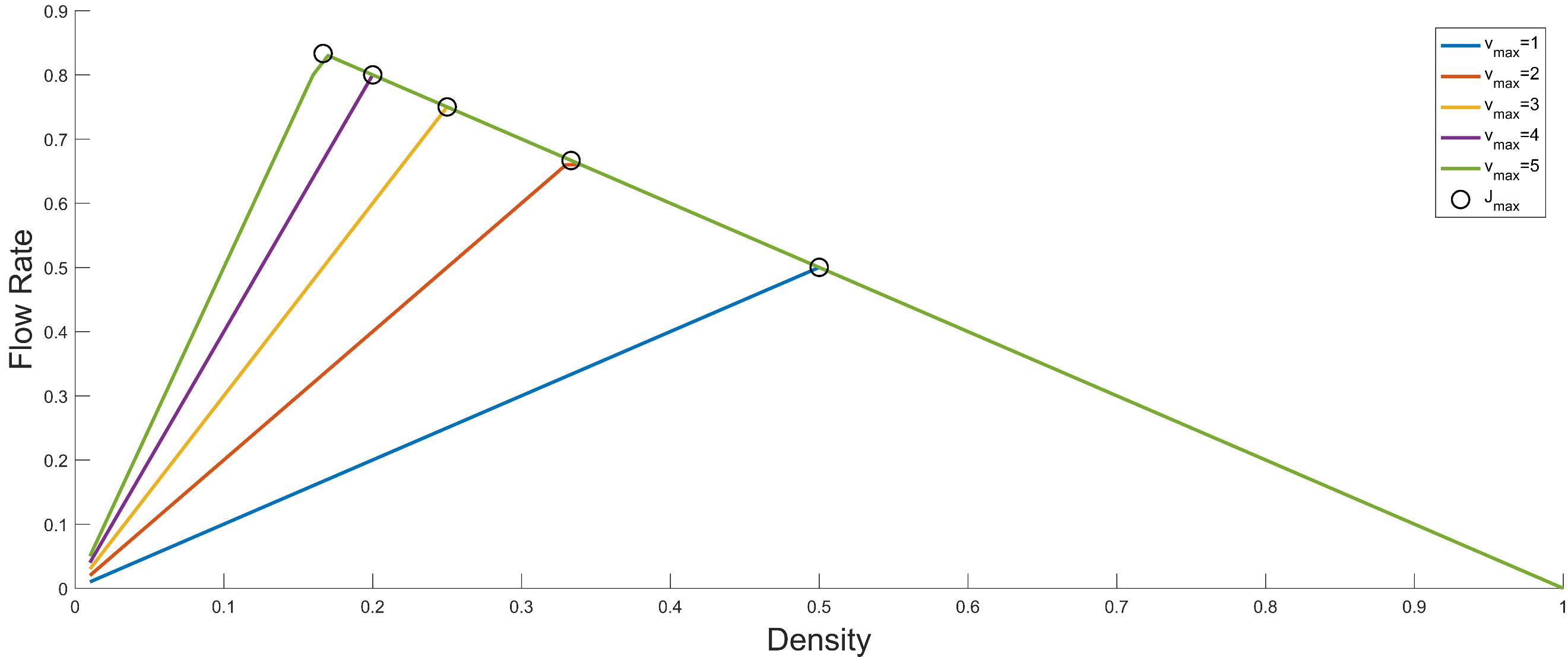


Figure 3: Plot of density v flow rate for various .

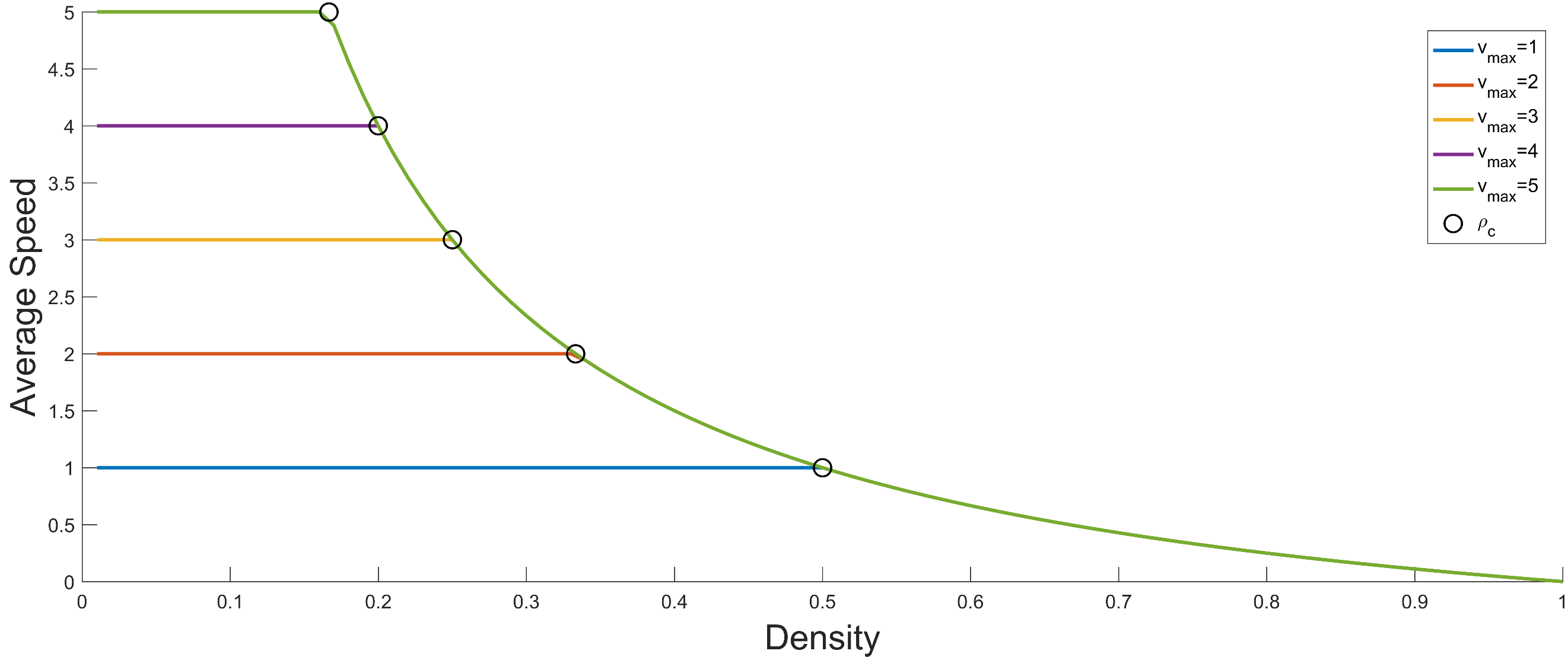


Figure 2: Plot of density v average speed for various .

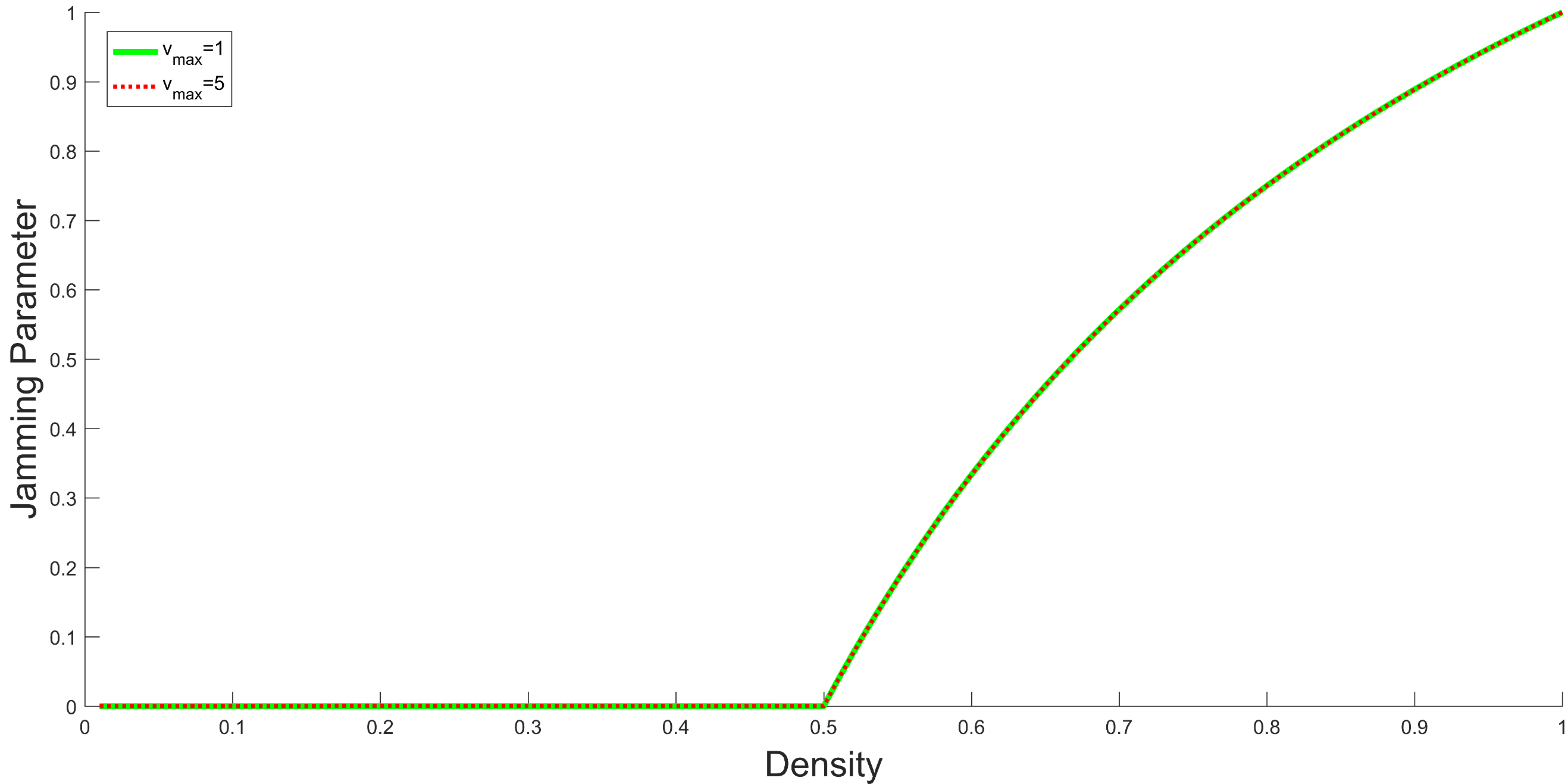


Figure 6: Plot of density v jamming at high and low for

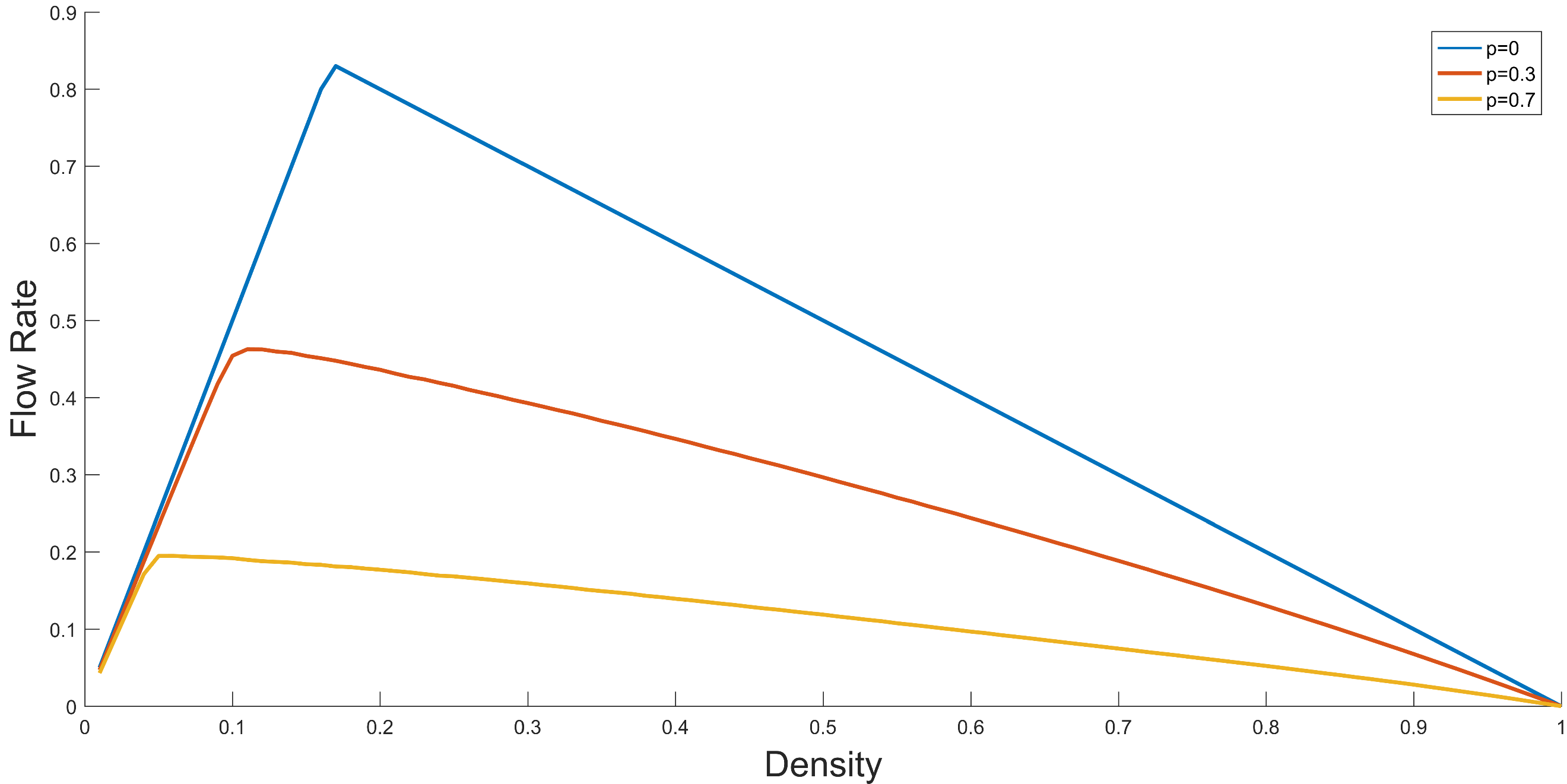


Figure 5: Plot of density v flow rate for various . .

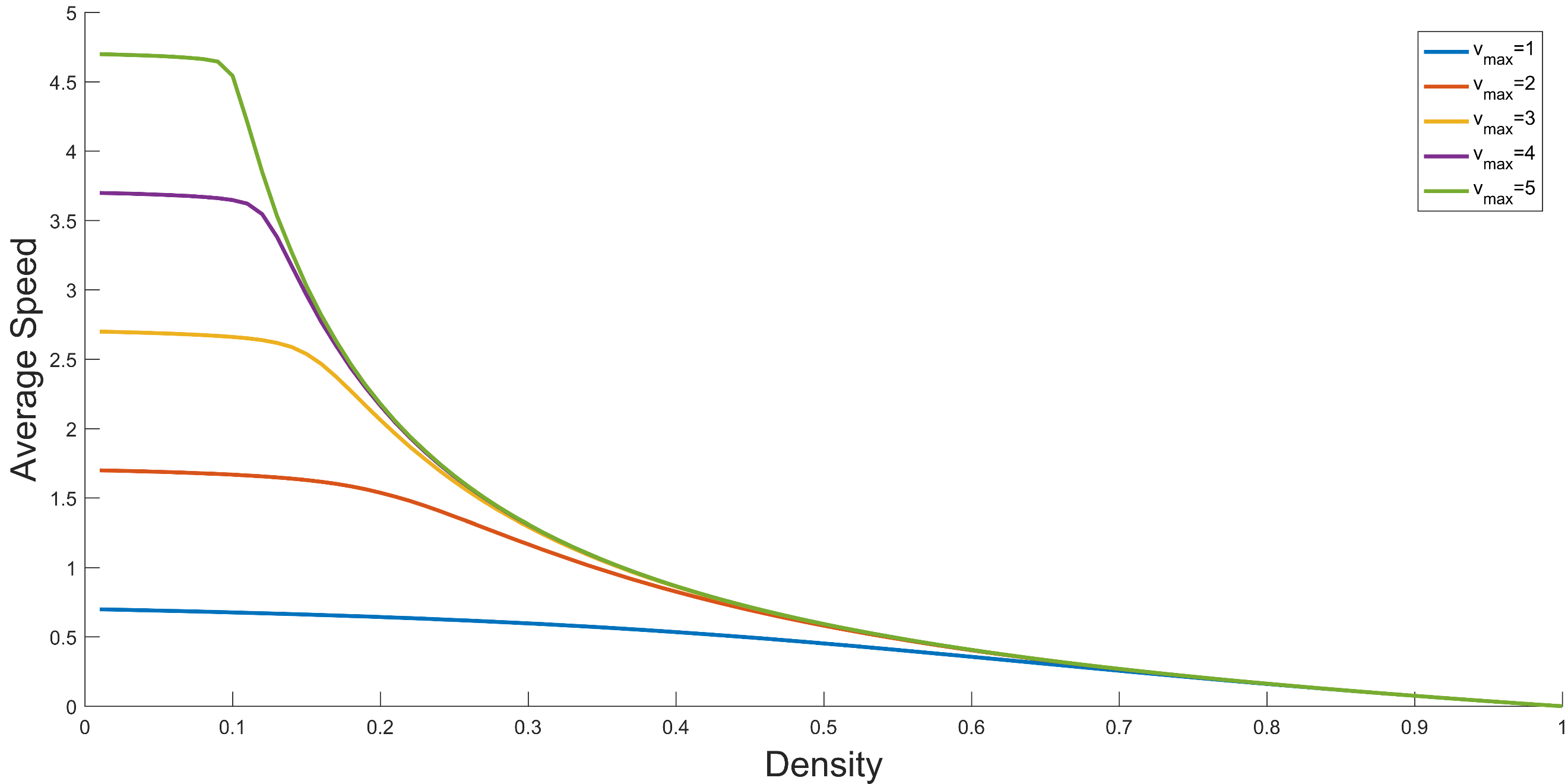


Figure 8: Plot of density v jamming parameter at for various .

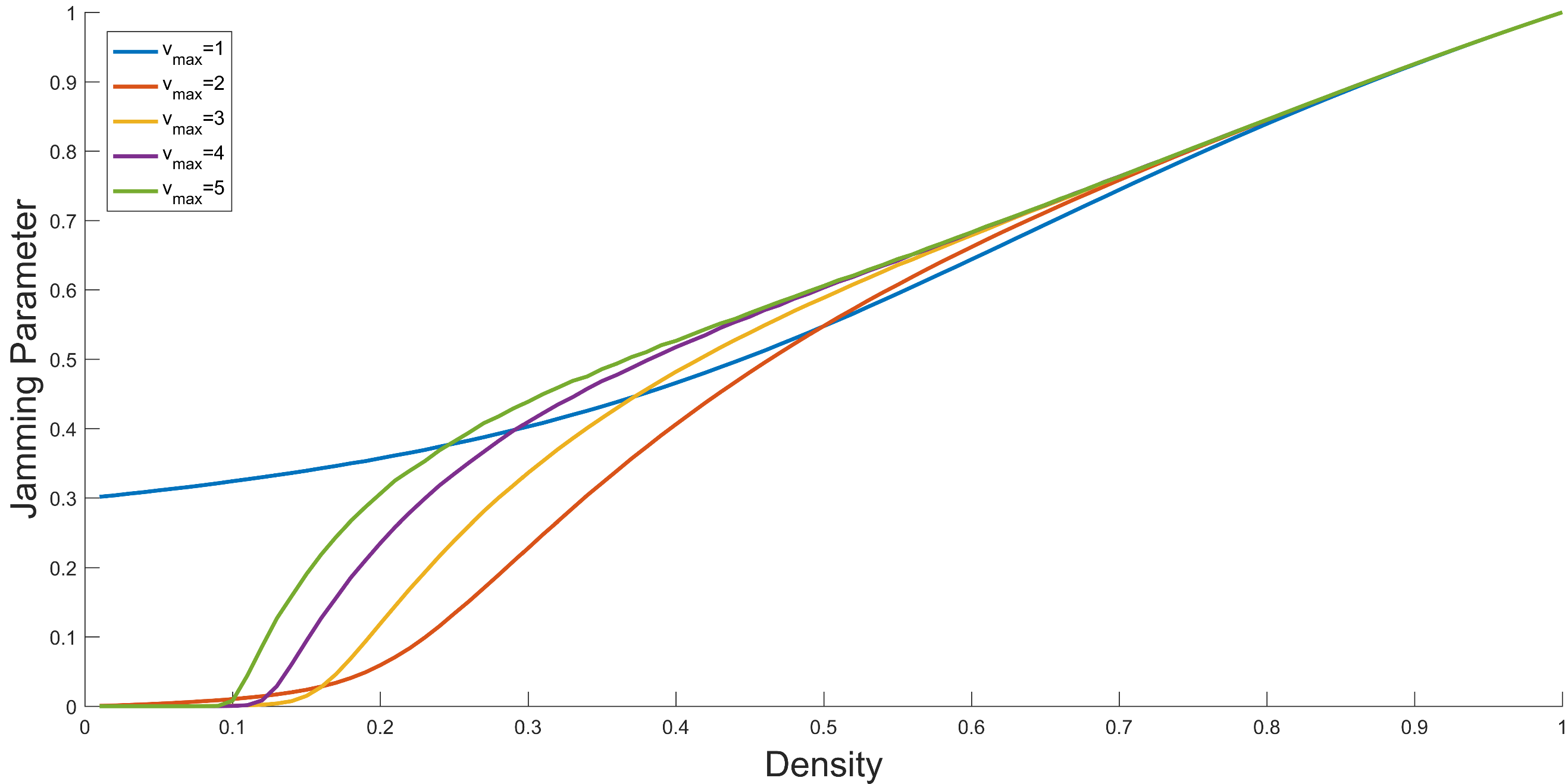


Figure 7: Plot of density v jamming parameter at for various .

# Investigation

To investigate the effect of having a variable speed limit on the road was broken up into a number of segments within which a new speed limit was to be set based upon the density within each of the segments.

To do this the following modifications to the code were made to simulate the effects. First a new variable, *“numberSegments”*, was added to the start of the code to control the number of segments with different speed limits. The length of the segments was then calculated and stored in the *“segmentLength”* array. The density values for the speed limit change were stored in the *“changeDensity”* array.

A new function, *“variableSpeedLimiter.m”*, was created to calculate the speed limits in each of the segments, it works in the following way. Firstly, the function preallocates the speed limit array for each segment such that all segments start with the maximum allowed speed. In a for loop it then calculates the density within each of the segments and then uses the density to set the speed limit within each of the segments. The density values at which the speed limit was to be changed was something to be explored. The function then ensures that the speed limits between adjacent cells are within two of each other in order to ensure that there are no sudden speed limit changes which could cause a bottle neck.

A couple of approaches to determine the values at which the density needed to be changed were explored. The value at which the change occurred can be seen in table 1. In approach 1 the change values are determined by taking the critical densities from the deterministic limit at calculated from equation (i), for example if the speed limit in the segment was then the speed limit would be reduced to once . In approach 2 the deterministic limit is used again but uses the theoretical maximum average speed to set the speed limit. Hence the density values chosen are after which the average speed drops below that that the speed limit is being changed to, this is the critical value for the speed being changed to. For example, once the density passes the critical value for then the speed limit is set to 4, this preserves the highest average speed possible. Approaches 3 and 4 follow from similar logic to approaches 1 and 2 but instead the density values chosen are those from the critical density and average speed graphs for the (Figure 8). The final approach (5) is to use the value at which jamming starts from figure 6, for the transition to the value for which was taken from the point at which the curve went above the curve. All of the values can be seen in table 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Approach |  | 4 | 3 | 2 | 1 |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |

Table 1: density values for which the speed limit is set.

This was then run with the same basic parameters as in the data run in the methods section, with the exception that this time the max speed was set to and the probability of slowing was . Graphs were then plotted to show the effect of the variable speed limit upon flow rate, average speed and jamming. These can be seen in figures 9, 10, 11. It is clear from the data that all the methods hinder both the speed and flow rate to some extent but they all do on average, reduce the number of stationary cars. As it is not possible to improve average speed and flow rate of traffic using variable speed limits but is possible to reduce the number of stationary cars on the road then. It is, therefore, logical to set the objective of using variable speed limits to minimise the number of stationary cars whilst having the least impact on speed and flow. Therefore, we can evaluate the effectiveness of the approaches to using a variable speed limit based on this information.

The methods used in approaches 1 and 3 is clearly very sensitive to the probability hence the large amount of noise in the data and for some values both increase the number of stationary cars, and therefore must be deemed unsuitable. Approach 5 reduces the number of stationary cars the most but has the biggest effect on flow and average speed, thus it must also be rules out. Approaches 2 and 4 reduce average speed and flow rate the least are therefore it must be concluded that it is best to set a variable speed limit based on the average speed of the traffic. Approach 2 has the least effect on average speed and flow rate but was also the worst at preventing jamming, when compared to approach 4. Approach 4 did however have a much larger effect on flow rate. To decide which was better would depend on the objectives required for the use of a variable speed limit, further testing would likely find a more optimal method.

To conclude it is possible to set a variable speed limit in order to reduce the number of stationary cars on a road, however there is a knock on effect on the through flow of cars and their average speeds. These results agree with those observed by studies of the A2 motorway in the Netherlands and the M42 motorway in the United Kingdom. [8] [9] Although in the M42 study the variable speed limits were part of a larger traffic management scheme that included ramp metering and running on hard shoulders. Reducing the number of stationary cars also helps to reduce the carbon emissions from cars as it reduces the amount of acceleration required which is more polluting that running at a constant speed [10].

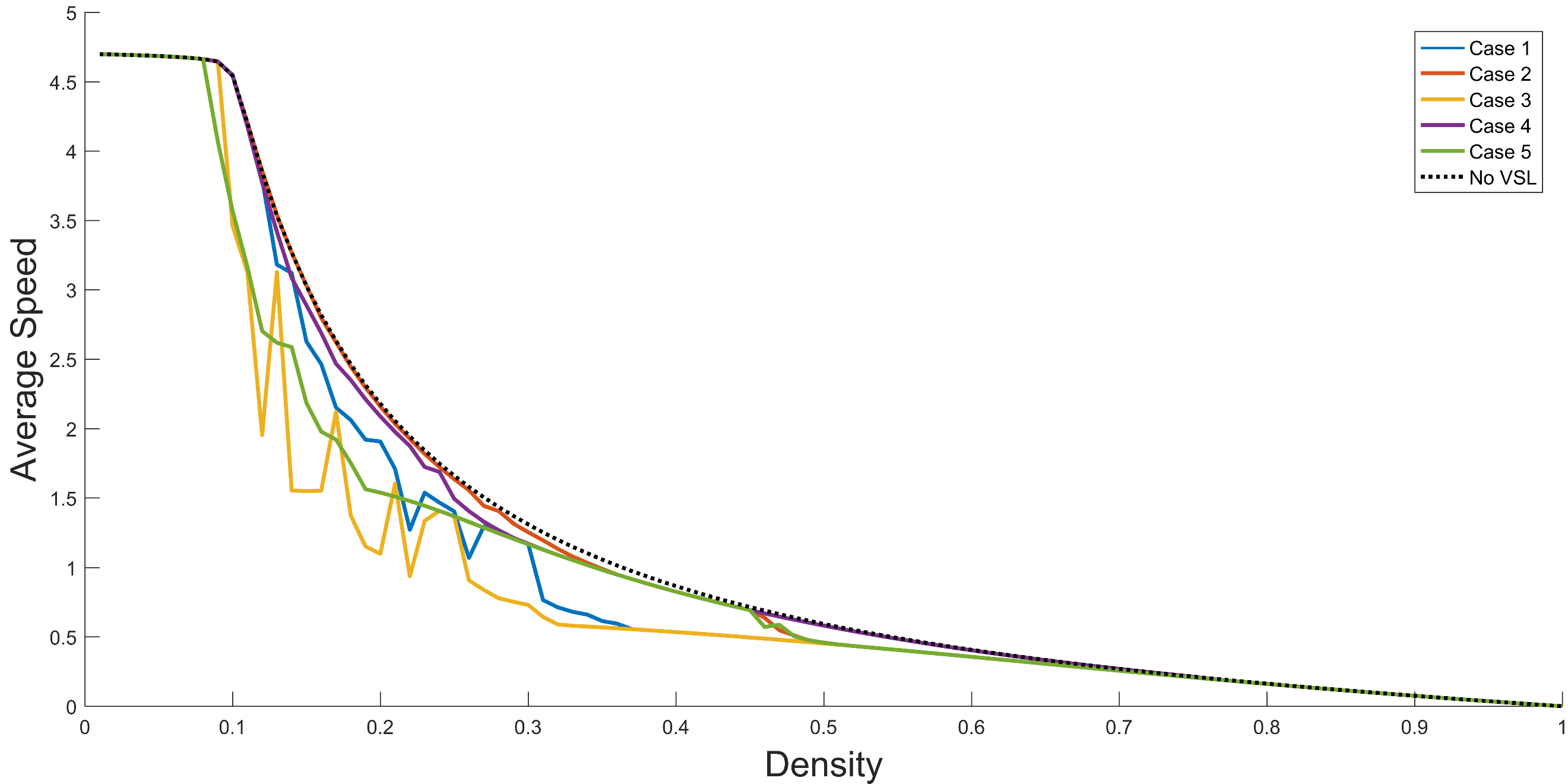


Figure 9: Plot of density v average speed using variable speed limits , .

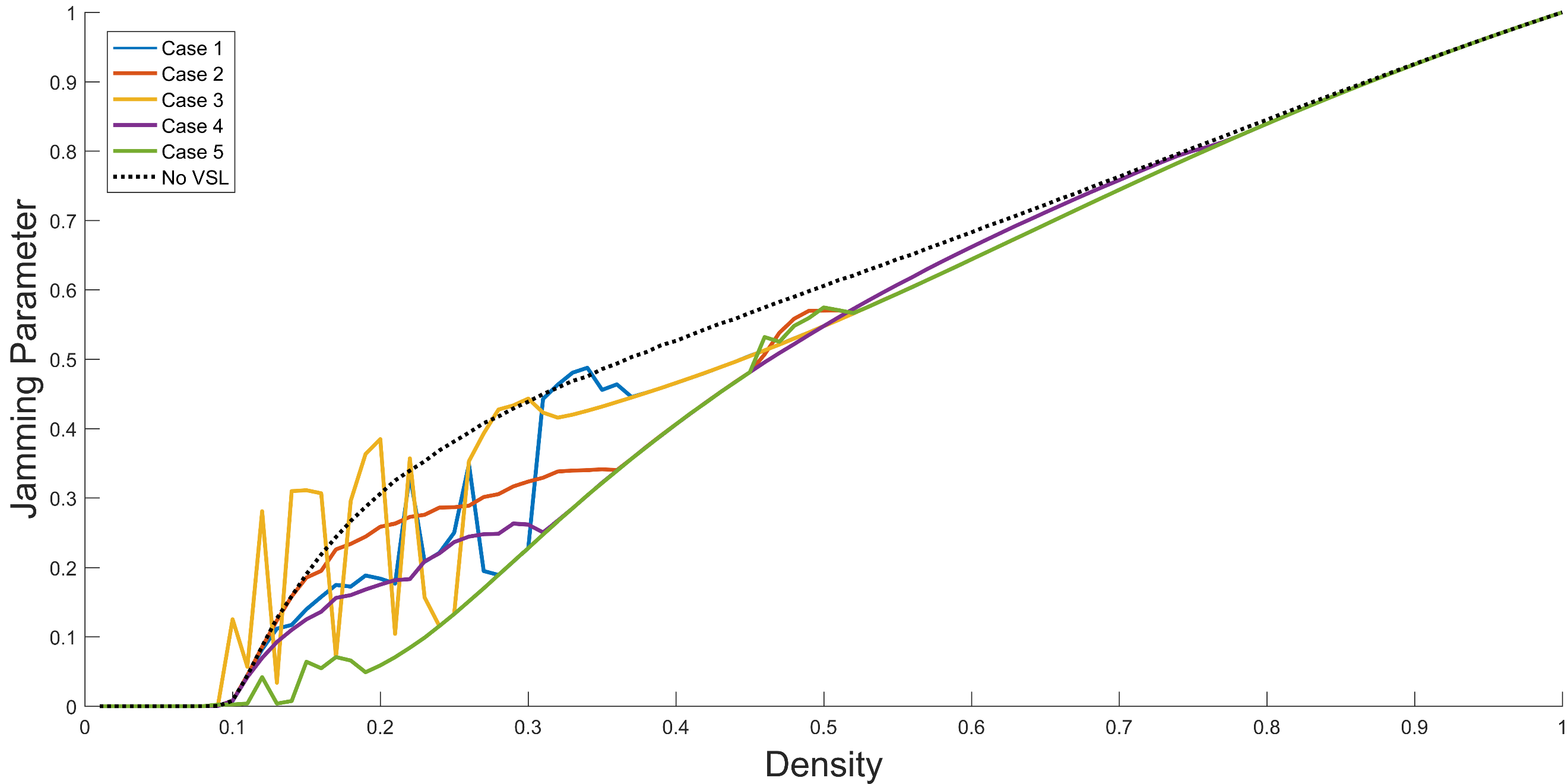


Figure 11: Plot of density v jamming parameter using variable speed limits , .

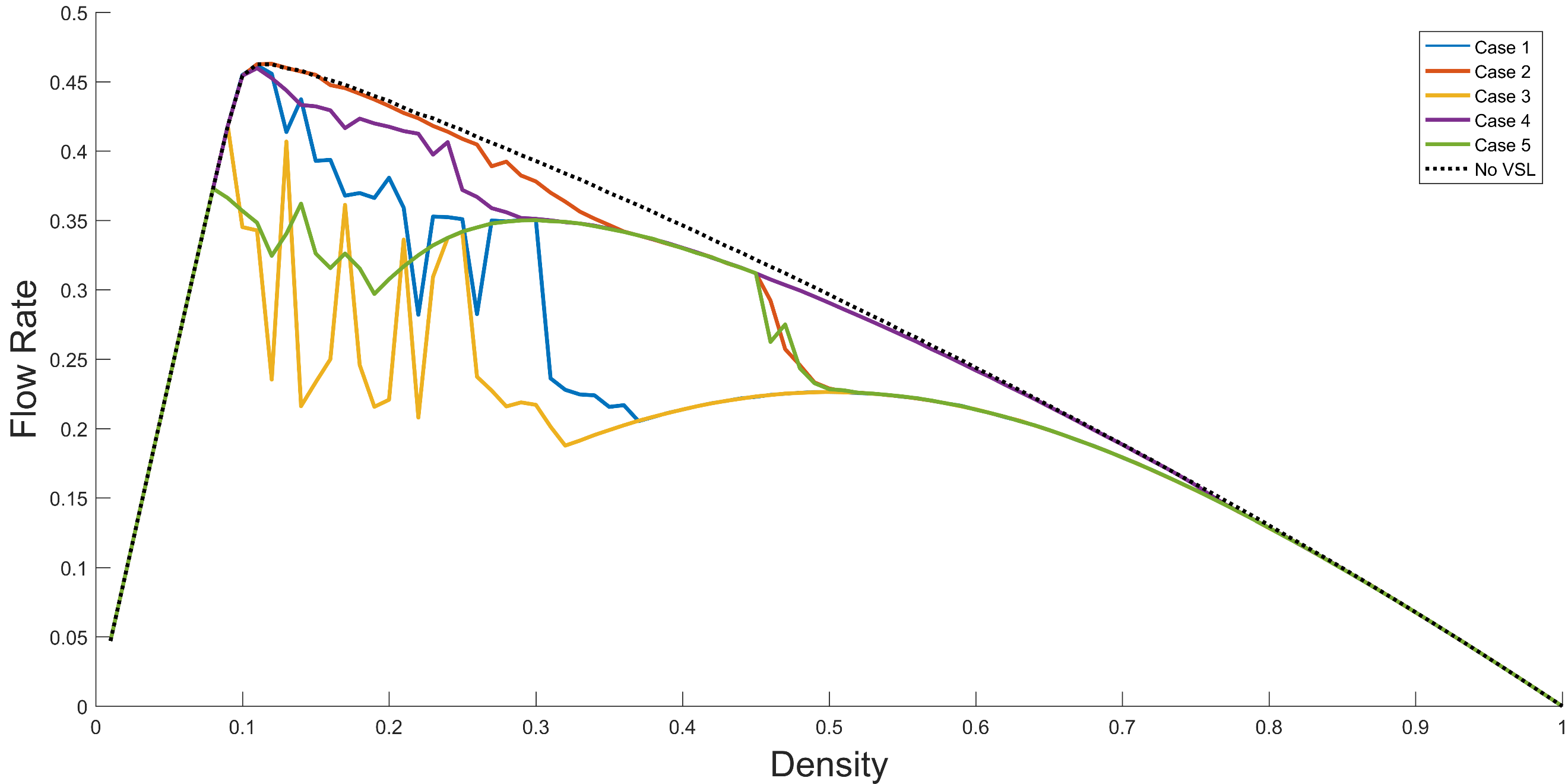


Figure 10: Plot of density v flow rate using variable speed limits , .

# Summary

The programme proved to be a successful way of simulating the traffic flow on the road. It showed how the density of cars on the road effects the rate of flow of traffic and the average speed of cars. It showed that a critical density exists at which there is a peak flow rate, cars travel at a maximum average speed up until this critical value, after which the average speed drops until it is equal to zero when the density of cars on the road is one.

The Nagel-Schreckenberg, whilst it is a useful way to simulate traffic flow, could be improved in a number of ways in order to more accurately represent real world traffic. Firstly, the model assumes that all cars travel at the speed limit wherever possible, whereas, in real world traffic, you get speeding cars as well as slow moving vehicles, such as busses or lorries which have their speed limited on roads. Step 2 of the model assumes cars always leave a safe gap between it and the car in front, but in reality, this does not occur as often cars travel to closely to each other, there is also the reverse problem of drivers being too cautious and leaving too much of a gap. In real world traffic, the probability that a car slows at random would be dependent on the speed that a car is travelling at not uniform for all speed as it is in this model. These are all areas which could be investigated in further studies.

The investigation into the use of variable speed limits found that it was not possible to increase either the traffic flow nor the average speed of cars, they could, however, be used to reduce the number of cars that were stationary on the road. This information would be useful to transport planner if they were thinking of using variable speed limits as a way to control traffic conditions. The model could also be extended to include the effects of ramp metering and or increasing the number of lanes, to simulated hard shoulder running, so that a more accurate simulation of traffic on smart motorways could be made.

# References

|  |  |
| --- | --- |
| [1] | Highways England, “Strategic Business Plan Overview 2015-2020,” [Online]. Available: https://www.gov.uk/government/uploads/system/uploads/attachment\_data/file/384974/141209\_Strategic\_Business\_Plan\_Overview\_Final.pdf. |
| [2] | K. Nagel and M. Schreckenberg, “A cellular automation model for freeway traffic,” 1992. |
| [3] | B. Eisenblätter, L. Santen, A. Schadschnider and M. Schreckenberg, “Jamming transition in a cellular automaton model for traffic flow,” 2008. |
| [4] | A. Schadschneider, D. Chowdhury and K. Nishinari, “Chapter Seven - Vehicular Traffic II: The Nagel–Schreckenberg Model,” in *Stochastic Transport in Complex Systems* , Philadelphia, Elsevier BV., 2010, pp. 243-279. |
| [5] | A. Balouchi and D. A. Browne, “Finite-Size effects in the Nagel-Schreckenberg traffic model,” *Physical Review E,* 2016. |
| [6] | J. Write, “Investigating Traffic Flow in The Nagel-Schreckenberg Model,” 2013. |
| [7] | N. Bain, T. Emig, F.-J. Ulm and M. Schreckenberg, “Velocity statistics of the Nagel-Schreckenberg model,” *Phys. Rev. E,* vol. 93, no. 2, p. 022305, 2016. |
| [8] | E. v. d. Hoogen and S. Smulders, “Control By Variable Speed Limit Signs: Results of the Dutch Experiment,” 1994. |
| [9] | Highways England, “M42 Active Traffic Management Results - First Six Months,” 2007. |
| [10] | L. C. Landman, “Effects of Acceleration Rate on Vehicle Exhaust Emissions and Fuel Economy,” National Service Center for Environmental Publications (NSCEP), 1982. |

Title page image © Highways England