NS model: replicate the model in section VI and VIII-D of the following paper.

Nagel, K., Wolf, D. E., Wagner, P., & Simon, P. (1998). Two-lane traffic rules for cellular automata: A systematic approach. *Physical Review E*, *58*(2), 1425–1437. http://doi.org/10.1103/PhysRevE.58.1425

Some main points

* To get the model in section VI: Set APPLY\_SYMMETRIC\_RULE = false, SLACK=0
* To get the model in section VIII-D: Set APPLY\_SYMMETRIC\_RULE = true, SLACK=3
* ROAD\_SIZE is the size of road segment in cells
* The default parameters’ values in the code are for NS model. If one changes these parameters, take into account the corresponding parameters such as MAX\_ACCELERATION (default is 1).
* getFlow() returned the travelled cells of all car at the moment of calling over the given measured time (numIterations)
* getFlow2() returned the number of cars passing the end point of road segment at the moment of calling over the given measured time (numIterations).
* numIterations should be long enough such that cars have time to cycle the road segment several times, in that case we could see state state

**Car generation**

The distance between a car A and the car in the front B is 2\*speed of the car A (following the rule 2-seconds). Depends on the density on the road, the speed of cars at the beginning will be constraint to satisfy the above condition.

Broken car if has will be counted in the number of slow cars and it will be generated after about ¼ of the slow cars have been generated.