Computational Physics

Prof. Ulrich Kleinekathöfer
Fall term 2020
Project 4, due November 4, 2020 at 11:55 pm
to be uploaded to https://moodle.jacobs-university.de

JACOBS UNIVERSITY

4. Nagel-Schreckenberg model of traffic flow [10 points]

The Nagel-Schreckenberg model, is a prominent example of the cellular automaton model. It is used to microscopically simulate traffic flow:

- Therein, the street is composed of single segments, so called cells. The size of one cell is supposed to be the space required by one car within a traffic jam (7.5 m). Each cell is either occupied by one car or empty.
- The time is described discretely as turns i.e., the time per turn equals approximately the reaction time of a car driver (1 s).
- The velocity is also modelled discretely as cells/turn. The maximum velocity is defined as 5 cells/turn (equivalent to 150 km/h).
- For each car the following steps have to be executed:
    1. If the car does not yet drive with maximum velocity, the latter will be increased by one.
    2. If the distance to the next car is smaller than the velocity value, the velocity will be reduced to the gap length.
    3. The velocity will be reduced by one correponding to a probability of p (dawdle probability), as long as the velocity is non-zero.
    4. All cars are moved by their current velocities.

Here, the street is supposed to be ring-like with a length l. If a car drove to the position $s \geq l$, it would rather be placed on position $s - l$.

1. Create a program with the previously introduced cellular automaton.
   To implement the probability p, use the built-in functions of your programming language.
2. Program an output for a space-time-diagram.
3. Simulate the following scenarios of a ring-shaped street of 100 cells:
    - Simulate a traffic jam with 6 of 18 cars and $p = 0.0$. The rest of the cars are to distributed randomly with a random initial velocity.
    - Simulate the previous task with $p = 0.2$.
    - Double the number of cars in the model with equal p.
4. Plot the results in space time diagrams.

General remarks for all Projects

You will have to (i) analyze the problem, (ii) select an algorithm (if not specified), (iii) write a Python program, (iv) run the program, (v) visualize the data numerical data, and (vi) extract an answer to the physics question from the data.
Which checks did you perform to validate the code? State the results you got for these tests.
For each project you will submit a short report describing the physics problem, your way of attacking it, and the results you obtained. Provide the documented Python code in such a form that we can run the code. A Jupyter Notebook including the code and report is fine but not necessary.