In-Class Work: Traffic Flow

At the beginning of class I will guide us through the program for the velocity update (5. of last class):

`~kvollmay/share.dir/inclass2025.dir/traffic5.py`

6. Update Positions

Now you are ready to add to the program the update of the positions. To do so use a new loop over all cars and update both the `carpos` array and the `road` array. For the update of `road` make sure to first copy the new velocity into a variable (e.g. `vnew`), then empty the old site and then put the car in road on its new site `xnew = xold + vnew` with the new velocity (the order of these commands matters). Remember to use the periodic boundary condition. After the update of all positions print the complete road and check if it is what you expected.

7. Finish Program

Now you are set to finish the program for our traffic flow model. Add to your program from 6. the time loop. Compare your result for the case of 15 timesteps (`MAXTIMESTEPS = 15`), `PCAR=0.3`, and similarly the other parameters the same as in `traffic6.py`. Compare your output with
`~kvollmay/share.dir/inclass2025.dir/traffic7_output.data`

8. Space-Time Diagrams

8a. Now we are ready to have a look at the flow of the cars. Next we will make a space-time diagram. This is an image of the road for successive time timesteps, i.e. on the x-axis is the road position and the y-axis is the time step increasing downwards. You find an example in Fig. 3 of the Chowdhury et al. traffic flow paper. This means that we want a picture similar to the DLA-fractal growth picture we made in class when we worked on the DLA model. You will need to first make a two-dimensional array which stores the space-time diagram data, for example named `spacetimearray`

Add to your program such an array and set the values of the `spacetimearray` during your time-loop.
8a. To be able to see the main patterns (and to get nice pictures) use a larger road `ROADLENGTH= 200` and `MAXTIMESTEPS=100`. Now you are ready to make nice picture of this space-time diagram. You may use at the end of your program the following commands:

```
plt.imshow(spacetimearray,interpolation='nearest',cmap='bwr')
plt.colorbar(orientation='horizontal')
pdffile = "spacetime8_PCAR"+str(PCAR)+".pdf"
plt.savefig(pdffile)
```

Look at the resulting space-time diagram and interpret it.

8b. To distinguish stopped cars easily from driving cars, let's indicate every empty site instead of with -1 now with -VMAX. Look at the space-time diagram.

8c. Vary PCAR. Interpret the resulting space-time diagrams. Once you have several space-time diagrams for different PCAR, get me so that we can discuss as a class your results.

9. Nagel-Schreckenberg Model

9a. Get me when you get to this part. We are now ready to finish the programming of the Nagel-Schreckenberg model. Add to your program of 8. the randomization of the velocity, so complete the Nagel-Schreckenberg Model. Use `VMAX = 5, PCAR = 0.3, PDEC = 0.25, ROADLENGTH = 200, MAXTSTEPS=100` and have a look at the resulting space-time diagram.

9b. Keep all parameters as in 9a but vary
(i) `PCAR` between 0.05 and 0.35
(ii) `PDEC` between 0.0 and 0.5
(iii) `VMAX` between 1 and 10.
What do you observe in each case? Please get me to discuss your observations and to share your results with the class.

10. Mean Velocity $v_{\rm av}(t)$ (if time)

Use your program from 9. with parameters `VMAX=5, PCAR=0.3, PDEC=0.25, ROADLENGTH=1000, and MAXTIMESTEPS=400` but after the time loop do not print out the space-time data and figure but instead print within the time loop one line with two numbers: time step `t` and $v_{\rm av}$ where $v_{\rm av}$ is the mean velocity:

$$v_{\rm av}(t) = \frac{1}{N}\sum_{i=0}^{N-1} v_i(t) \qquad (1)$$

N is the number of cars and v_i is the velocity of car i. Look at $v_{\rm av}(t)$ with:
`traffic10.py | xmgrace -pipe`
Get me so that we can discuss your result.