In-Class Work: Traffic Flow

4. Distance

Please copy the program
`~kvollmay/share.dir/inclass2021.dir/traffic4.py` into your working directory, and use this program as starting program for today's inclass work. I will explain what exactly this program does.

5. Update Velocities

5a. For the update of the velocities we will need a function which determines the smallest number of three integers. Python has a function which does this, for example `min(3,1,9)` gives 1 as result. Test this function with a few print commands

5b. Add to the program that it determines all new velocities (and updates them in the `road` array), and prints out the road with the new velocities. Check your program with print commands.

6. Update Positions

Now you are ready to add to your 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). After the update of all positions print the complete road and check if it is what you expected.

7. Finish Program (if time)

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 `traffic4.py`. Compare your output with
`~kvollmay/share.dir/inclass2021.dir/traffic7_output.data`

8. Space-Time Diagrams (IF TIME)

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.