In-Class Work: Traffic Flow

7. Finish Program (time-loop)

Copy the following program into your directory
`~kvollmay/share.dir/inclass.dir/traffic7.py`
(Or if you had gotten yesterday the time-loop successfully built in, you may use your program.

This program includes the time loop for the case of 100 timesteps (`MAXTIMESTEPS = 100`), `PCAR=0.3`, and otherwise the same parameters as before.

8. Space-Time Diagrams

8a. Now we are ready to have a look at the flow of the cars. Switch back to `PCAR=0.3`. 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 downwards and equal to the number of time steps. 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. To print the array at the end into a file, you may use the following lines:

```
#print spacetimearray into file
fileoutdat=open("spacetime8.data",mode='w')
for i in range(MAXTIMESTEPS):
  for j in range(ROADLENGTH):
     print(spacetimearray[i,j], end=" ",file=fileoutdat)
  print("\n", end=" ",file=fileoutdat)
fileoutdat.close()
```

To test your program compare the resulting `spacetime8.data` with your results in step 7.

8b. To be able to see the main patterns (and to get nice pictures) use a larger road `ROADLENGTH`= 200. Now you are ready to make nice picture of this space-time diagram. You find an example for the python syntax at the end of `~kvollmay/share.dir/inclass.dir/classfractal3.py`
To get also a colorbar, use before the command `plt.savefig` the following command:
`plt.colorbar(orientation='horizontal')`
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 = 0.2, 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 $v_{\mathbf{av}}(t)$

Use your program from 9. with parameters `VMAX=5, PCAR=0.2, PDEC=0.25, ROADLENGTH=1000,` and `MAXTIMESTEPS=200` 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_{av} where v_{av} is the mean velocity:

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

N is the number of cars and v_i is the velocity of car i. Compare your result with
`~kvollmay/share.dir/inclass.dir/traffic10out.data`
Look at $v_{\text{av}}(t)$ with:
`traffic10.py | xmgrace -pipe`
Get me so that we can discuss your result. Our interpretation will be necessary for next class.