TP 8 : HPP model

Cours de modélisation numérique

28 avril 2023

We are interested in a network gas model proposed by Hardy, Pomeau and de Pazzis from 1973.

The HPP model

We consider fictitious particles which move in a regular size network $n_x \times n_y$. These particles travel at a constant speed in one of the four directions, as shown in the figure 1.

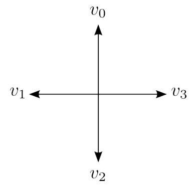

FIGURE 1 – The four possible directions of the HPP model

To better understand how the HPP algorithm works, we subdivide an iteration into two parts :
— *Collision :* The particles in the same space site collide. In some cases, this has the effect of changing the directions of the particles trajectories.
— *Propagation :* The particles move freely towards the neighboring site.

In the HPP model only two of the possible collisions affect the direction of the particles involved. They are shown in the figure 2. All other collisions leave the system unchanged. As for the propagation, we schematize it on the figure 3.

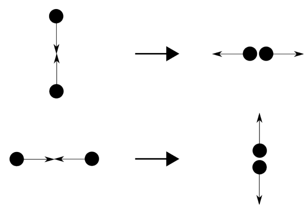

FIGURE 2 – HPP model collisions

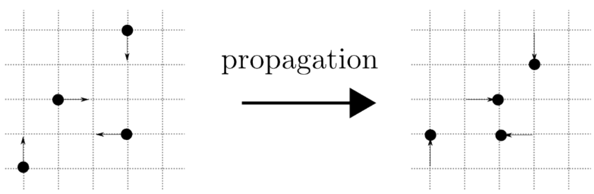

FIGURE 3 – Streaming (propagation)

It is also possible to add solid and immobile bodies (*e.g* of the walls) in our simulation. To do this, we assign an additional bit b_{ij} at site n_{ij}. If $b_{ij} = 0$ we apply the algorithm described above. On the other hand, for the case $b_{ij} = 1$ it is necessary to apply a simpler algorithm called *bounce-back*, thay reverses the speed of the particles which arrive on the site. See figure 4.

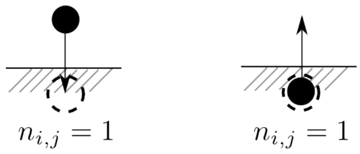

FIGURE 4 – Collision with a solid site.

Work to do

Implement the HPP model. You can start from the template file given on Moodle.

— As in the previous exercises, for performance and readibility reasons, you should avoid using space loops as much as possible. The rule can be implemented very effectively, since there are only two important cases. In order to encode the particles traveling in the different directions, we have the choice to use four different binary matrices or a single matrix in which it is necessary to use numbers coded on 4 bits.

— First, test your code with an initial random configuration on a periodic domain. Check that the collision and propagation steps are correctly implemented. Prove that the mass remains constant during the simulation,*i.e.* count the number of particles in the system and verifying that this quantity remains constant.

— Place a set of particles in a circular domain, and then let them evolve. Describe this evolution (eg with plots). What kind of phenomenon does it describe ? How does it compare with your perception of real world ?

— Then add the possibility of having edges. Implement the case shown by the figure 5. In this situation there are two connected compartments, one of which is empty while the other contains a large amount of particles. How does this system evolve over time ?

— Finally, reverse all particle speeds and apply the algorithm again for the same number of iterations. This should bring us back to the original situation. What happens if a particle is perturbed before the speeds are inverted ?

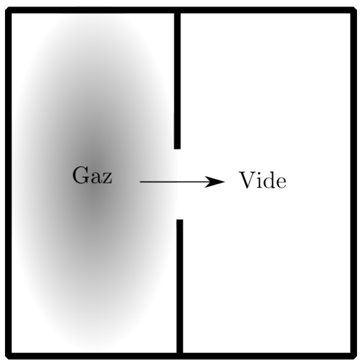

FIGURE 5 – *Gaz*