Multi-Agents Model

by Martino Ferrari

1 Implementation

In this TP we will implement a simple multi-agent model of a bacteria. The bacteria at each iteration can continue on its direction or randomly change direction.
The probability of each behaviors depends from the density of nutriment on its position.

To model this agent I choose to use a simple class:

```
class Agent:
    """Bacterial agent."""
    def __init__(self, pos):
        self.pos = np.array(pos)
        self.speed = 2 # 20 um/s
        self.direction = __randir__()
        self.density = 0
        self.alpha = 0.01

    def move(self, dt):
        """Move bacteria."""
        self.pos += self.speed * dt * self.direction

    def update(self, density):
        """Update direction."""
        p = 0.9 if density > self.density else 0.5
        self.density = density
        if rnd.rand() > p:
            self.direction = __randir__()
```

2 Results

In this section I will comment the results of different experiment I did.

2.1 Continuous density

First to be able to update the bacteria we need to define the nutriment density in function of the position. In this first part I will consider the following:

$$\rho(\bar{x}) = \frac{1}{1 + |\bar{x} - \bar{c}|}$$

where $\bar{x}$ is the current position and $\bar{c}$ is the center of the simulation.

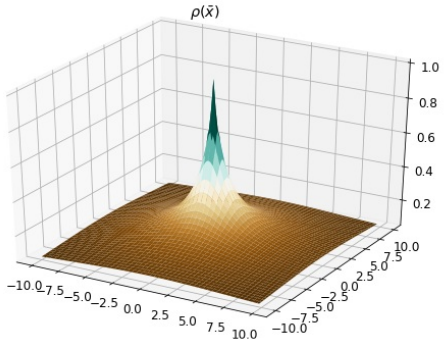

Figure 1. Nutrition density

2.2 1 Bacteria Itinerary

The first experience we were asked to do is to draw the itinerary of a single bacteria in our simulation, after 30 iterations the results is the following:

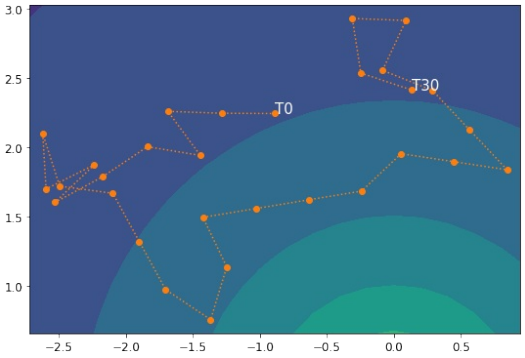

Figure 2. Single bacteria intinerary

It's interesting to see how the bacteria in our simulation moves at a constant speed, and how it's somehow attracted by the nutriment (as the probability of direction change is smaller).

2.3 100 Bacteria

In this experiment we were asked to analyze a simulation with 100 bacteria, the first experimentation I did was to see how the population evolve during the time (100 iterations, 20s):

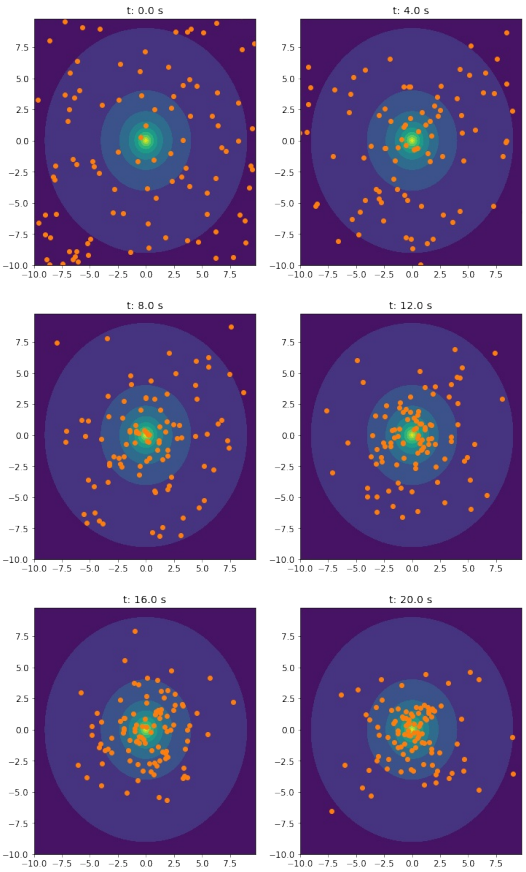

Figure 3. Evolution of 100 bacteria

Then I tried a more statistical approach, I simulated 20 times the same system for 1000 iterations and I counted the numbers of bacteria within 15 μm from the center:

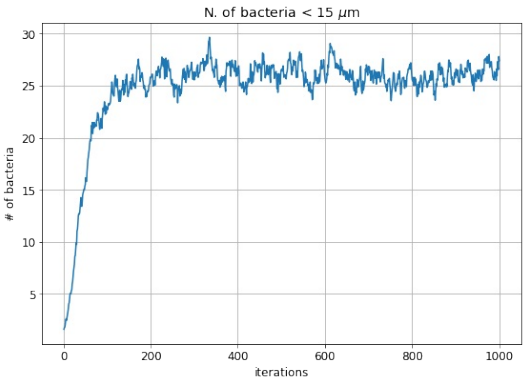

Figure 4. Evolution of the number of bacteria in time

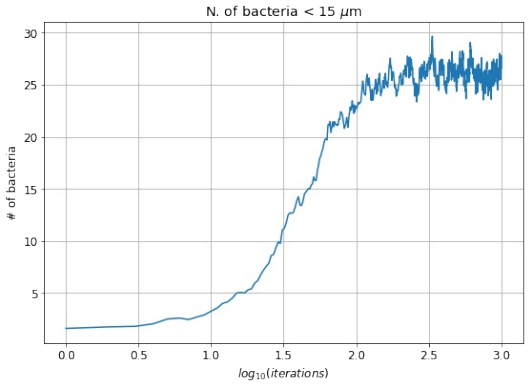

Figure 5. Evolution of the number of bacteria in logarithmic time scale

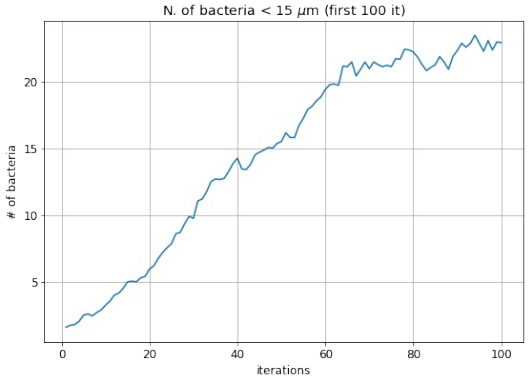

Figure 6. Details of the firt 100 iterations

The result is very interesting, as the norm of the speed is constant the bacteria can not stabilize at the center and start to "orbit" around it. In this way the number of bacteria close to the center grow linearly until saturate around 25 (as visible in both logarithmic scale and in the first 100 iterations plot).

2.4 Non-continuous density

I also experimented with a non-continuous density in the form of:

$$\rho(\bar{(}x)) = \begin{cases} 1 & |\bar{x} - \bar{c}| < 15\,\mu\,m \\ 0 & |\bar{x} - \bar{c}| \geq 15\,\mu\,m \end{cases}$$

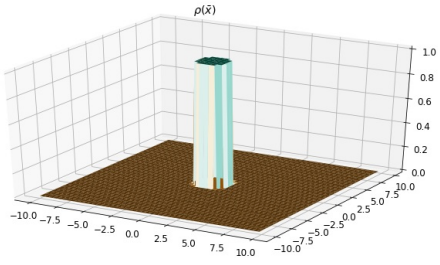

Figure 7. Nutrition density

2.5 100 Bacteria and non-continuous density

I then re-did the same experiment of before (population of 100 bacteria) with this new density distribution:

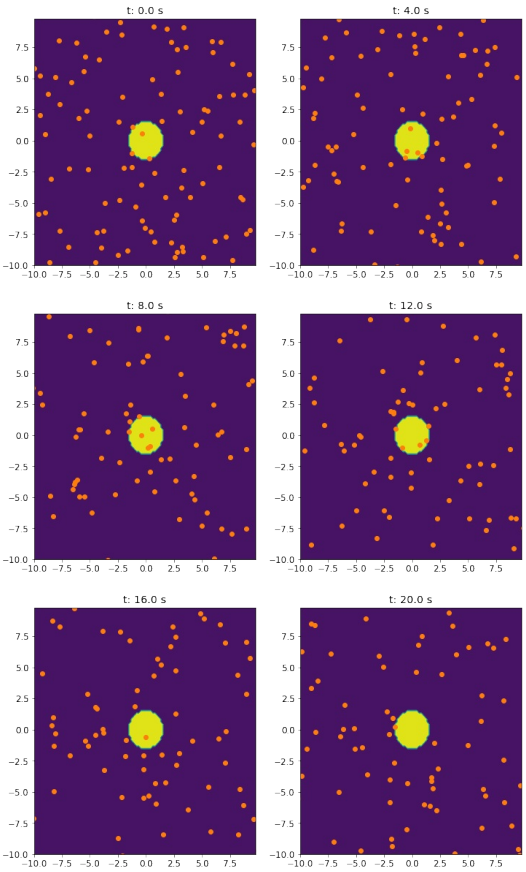

Figure 8. Evolution of 100 bacteria in time

At first view it seems that as the bacteria have constant speed and the nutrition density is distributed uniformly in the 2 areas of our simulation the bacteria are not able to be attracted by the source of food. The statistical analysis of the center of the simulation confirmed my hypothesis:

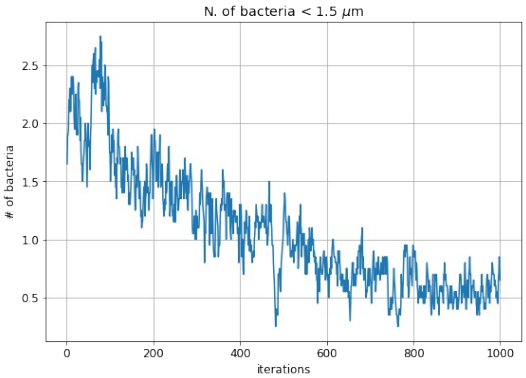

Figure 9. Evolution of the number of bacteria in time

As the attraction to the food is not efficient in this case in the long run the number of bacteria at the center tend to 0.

2.6 Actracted Bacteria

Then we were asked to implement a new movement function to the bacteria in a way that they would be attracted not only by the nutrients but as well by the other bacteria (only if the distance is $\leq 10\,\mu m$).

After implementing this new move function, I chose to redo the 100 bacteria experiment with continuous density distribution and the new move function:

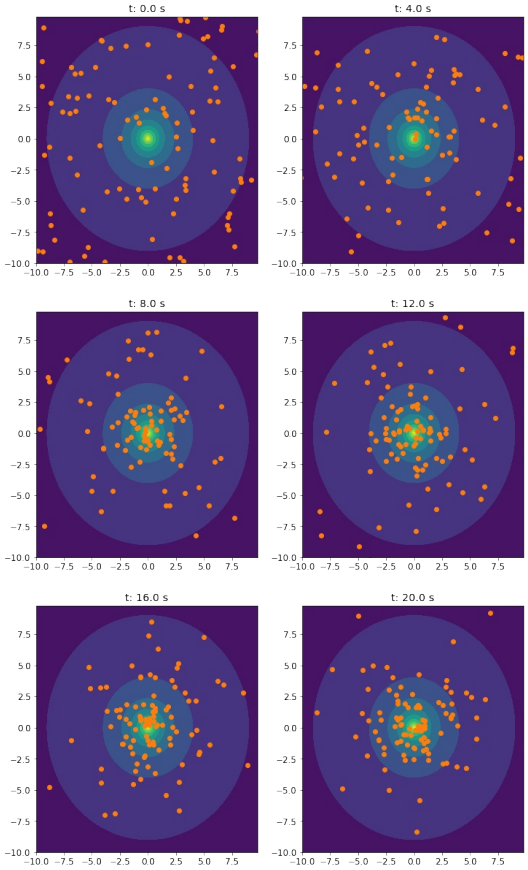

Figure 10. Evolution of 100 bacteria in time

As the α parameter (that control the attraction) chosen is small at first view there is no big difference. However the statistics (and longer simulations) show clearly the impact of this attraction:

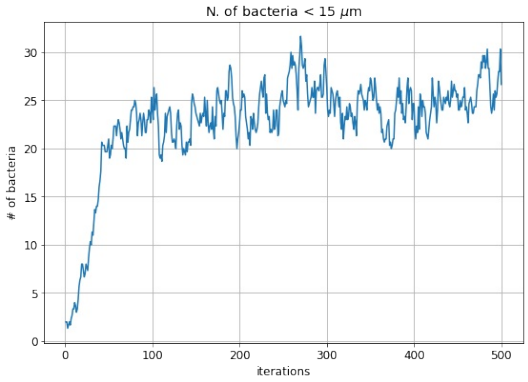

Figure 11. Evolution of the number of bacteria in time

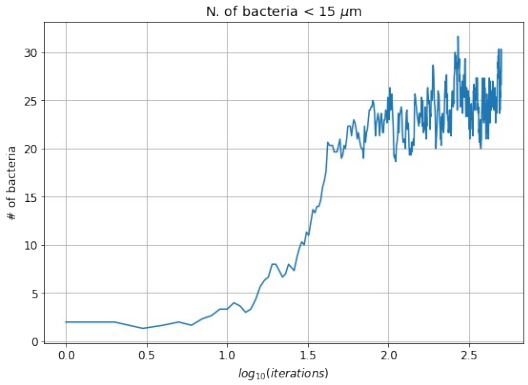

Figure 12. Evolution of the number of bacteria in logarithmic time

In the longer run it appears that the saturation limit is higher as the attraction tent to concentrate the bacteria and to counterbalance the constant speed problem, as well is possible to see that the concentration at the center is as well faster.

2.7 Parallelization

As last theoretical question we were asked a what could be a possible way to implement a parallel version of this multi-agent simulation.

Without attraction the answer is very simple, as there is no interaction between different bacteria, the agents could be divided in N groups (one per processor) and the simulation of this N groups could be done independently.

However whit the attraction the problem is more complex. With a big number of bacteria some sort of parallel Barnes-Hut algorithm should be the optimal solution, in order to minimize the inter-processors messages the closest bacteria are simulated together and the use of the tree structure should permit to minimize the number of messages.

With smaller number of bacteria a simple division of the space in grid should be enough (and parallel simulation of each area).