# Patterns and shapes, by simple mathematics rules

Paolo Poli
No associated University
Italy

September 2, 2020

## Abstract

In this paper, simulations carried out with a program developed by myself (MISSP) concerning correlated agents dynamics, are shown through some examples, where the results have shown interesting patterns and shapes.

## The mathematics behind the program simulation

The dynamics of the agents, is leaded by a specific mathematics. Any agent interacts with each other, in function of some specific set functions, written into the program. Any agent chooses, a random number of points around it, with a maximum and minimum step set into the script. For each point chosen, there is a sampling rule that depends from other connected agents (represented by set $A$), and the type of algorithm chosen in the simulation. An example can make clearer, the statement did before. Taking in consideration a single agent, it samples the sorrounding space with a fixed number of points represented by the set $S = \vec{x_1} ... \vec{x_n}$. The choice of the point where the agent moves at the next step, depends on the three possible conditions that you can set with MISSP. These conditions are the following:

$$x_n : \max \sum_{i \in A} f_i(\vec{x_j}) \text{with} x_n \text{and} x_j \in S \tag{1}$$

$$x_n : \min \sum_{i \in A} f_i(\vec{x_j}) \text{with} x_n \text{and} x_j \in S$$

(2)

$$x_n : p(\sum_{i \in A} f_i(\vec{x_j})) \text{with} x_n \text{and} x_j \in S$$

(3)

The last condition means that each sampling point has an associated probability that agents can choose, and the probability is correlated at other agents. Only some kind of functions, are set into the source code, but all wanted functions can be added, into the code.

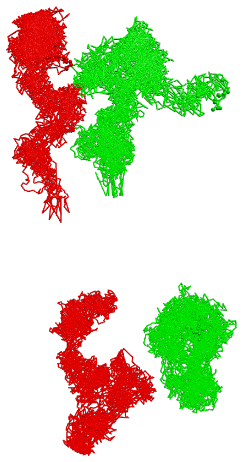

Figure 1: The top figure shows in red, the pin-up drawn by the bees swarm. The bottom figure shows in red the grandmother picture.

The working way of the calculation is shown in the next sections through examples.

## The Bee swarms

The first example, about how MISSP works and its results, are two simulations of two bee swarms, where the simple bees are connected with their own queen. Let's see the mathematics rules that connect the bees and the queen. The function correlates bees and queen is only one:

$$\exp(k_1(x_i - x_q)^2 + k_2(y_i - y_q)^2) \text{with} k < 0 \tag{4}$$

The queens are laid, in $x = -2$, $y = 0$ and $x = 2$, $y = 0$ positions and any queen has 20 worker bees around them, with an initial random position, with a distance from the queen of a range value of $\Delta x = + - 0.5$ and $\Delta y = + - 0.5$. The worker bees has the own queen and they don't betray her. The time simulation is fixed at a specific value. The final figures 1, represents, the path carried out by the two swarms. The queen moves in a $2d$ random walk, whereas the bees follow her, by the correlation imposed by the formula 4, with $i$ the coordinate of the points around the considered worker bee, whereas $q$ is the coordinate of the queen position.

How is possible to see in figure were represented two results where in both cases the swarm drew some-

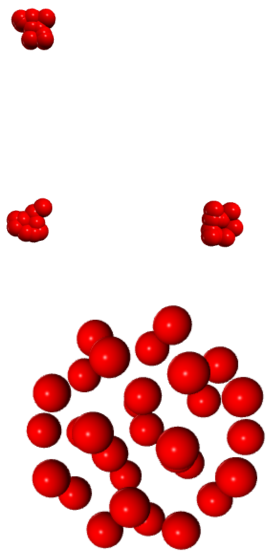
Figure 2: Upside, is shown disassembled agents at time $t = 0$, whereas in the bottom, the formed cell is represented.

thing of interesting. In the upper figure is possible to see the red swarm drew a pin-up with behind to her, something want to eat her bottom drawn by the green swarm. In the second figure, is possible to recognize a grandmother (red swarm), whereas in the green swarm, is not recognizable any geometry.

## The cell formation

The second example of MISSP simulation, is the unique cell of agents formation starting from separated agents. The mathematics rule is very simple, any agent starts from a separated position and they connect themself through the following correlation function for each agent,

$$\sum_{a\in A}\frac{k_0\sqrt{k_1(x_i-x_a)^2+k_2(y_i-y_a)^2+k_3(z_i-z_a)^2}}{k_1(x_i-x_a)^2+k_2(y_i-y_a)^2+k_3(z_i-z_a)^2+1}\text{with}k>0 \tag{5}$$

with A the set of all other agents and 1 the algorithm chosen. Three separated set of agents (figure 2) were positioned in three different points of the space, after the interaction by the function 5, a cell of agents was formed, shown in the lower side of the figure 2.

## The bees and the flower crown

In this example, The bees have to pick up, flowers pollen and a simulation was carried out. Two simulations were carried out of the same system but with slightly different conditions. The environment is formed by 20 fixed agents representing the flower crown and 20 free agents representing the bees, started from a single point in the bottom of the flowers crown. The Bees are not correlated with each others, whereas they are correlated with the crown flowers with the function 6 (in this case we have a $3d$ space environment), in both simulations. The only difference between the two simulations, is that in the second simulation, any bee can stay around a flower for a maximum number of 30 times, with the around distance settled to 0.25. This means that after 30 times that the agent is at a distance less or equal than 0.25 it doesn't consider the correlation anymore. This condition allowed the bees to visit all flowers, differently by the first simulation where only a part of crown was covered by bees (figure 3).

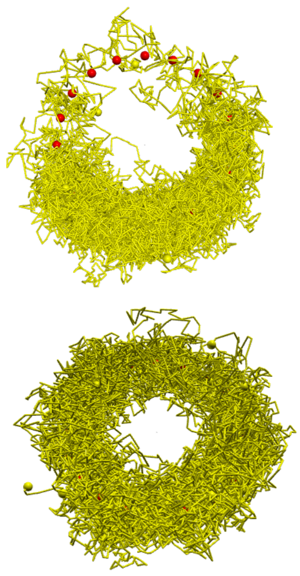
Figure 3: The upper figure, shows the bees not subjected at the condition of maximum touch with flowers. The bottom figure, shows the bees subjected at the maximum touch condition, with each flower.

$$\sum_{f\in A}\exp(k_1(x_i-x_f)^2+k_2(y_i-y_f)^2+k_3(z_i-z_f)^2)$$
$$\text{with}k<0 and\ A\ the\ set\ of\ flowers \tag{6}$$

## Twister pattern before a linear path

In this example I simulate, four agents, with two different kind of interactions. Four agents were put in a $2d$ space, on the same line, with coordinates $x = 1, y = 0$, $x = 3, y = 0$, for the first kind of agents and, $x = 2, y = 0$ and $x = 4, y = 0$ for the second kind of agents. The maximum step used was of 0.02, whereas the minimum one was of 0.01. The functions used were the 7 and 8. The first two agents, correlate all agents with the function 7, whereas the other two agents correlate all agents with the function 8. The results are shown in figure 4, where is possible to see how initially the two pair of the agents carry on, to turn around at a center point (upper figure), until they go away separately (lower figure 4).

$$\sum_{a\in A}\exp(k_1(x_i-x_a)^2+k_2(y_i-y_a)^2)\text{with}k<0 \tag{7}$$

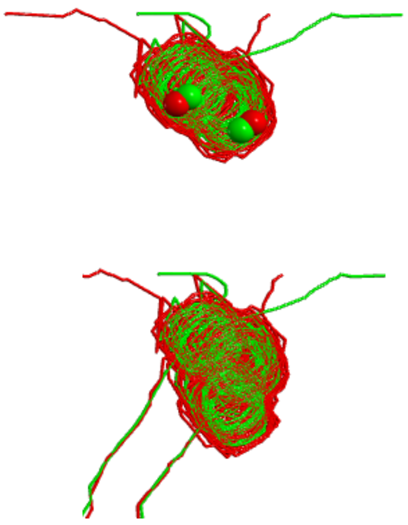

Figure 4: In the upper figure, the rotation around a point of the agents pair, is shown. In the lower figure, is shown the moment when the agents separate from themself.

$$\sum_{a\in A} exp(-k_1(x_i-x_a)^2-k_2(y_i-y_a)^2)(k_1(x_i-x_a)^2 +k_2(y_i-y_a)^2) \text{with} k>0 \text{and A set of all agents} \quad (8)$$

## The formed rotating agent systems

The last example proposed, is the one where the agents form two different structures.The structers were named the helicopters. Any, helicopter has an own core, with four agents around it.The agents around the core follow, the next agent in clockwise way through the function 9, whereas the core keep any agent at a fixed distance using the function 10 and the agents interact with its core with the function 9 summed on all agents around the core, with the agents positioned in $(1,0)$, $(0,-1)$, $(-1,0)$, $(0,1)$, with the coordinates respect the core of the single structure (the single helicopter considered). The cores are correlated by the function 11. The results are shown in the figure 5, and it shows two helicopter structers turning around at the associated core and initially move away from each other.

$$\frac{k_0\sqrt{k_1(x_i-x_a)^2+k_2(y_i-y_a)^2+k_3(z_i-z_a)^2}}{k_1(x_i-x_a)^2+k_2(y_i-y_a)^2+k_3(z_i-z_a)^2+1} \quad (9)$$
$$\text{with} k>0$$

$$\exp(-k_1(x_i-x_c)^2-k_2(y_i-y_c)^2-k_3(z_i-z_c)^2) (k_1(x_i-x_c)^2+k_2(y_i-y_c)^2+k_3(z_i-z_c)^2)\text{with} k>0 \quad (10)$$

$$\exp(-k_1(x_i-x_c)^2-k_2(y_i-y_c)^2-k_3(z_i-z_c)^2)(k_1(x_i-x_c)^2 +k_2(y_i-y_c)^2+k_3(z_i-z_c)^2)^2\text{with} k>0 \quad (11)$$

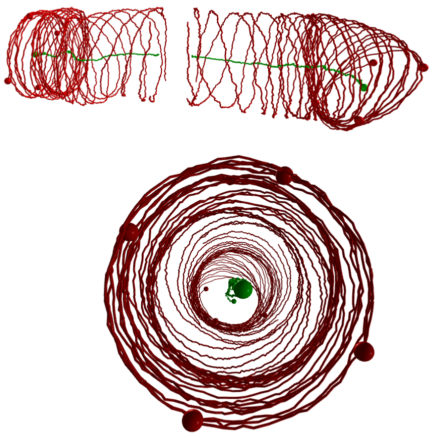

Figure 5: In the upper figure is shown, the trace of the two helicopter structures, whereas the front-side is shown, in the lower figure.

## Methods

The MISSP, was programmed in C, and Python was used for the graphic visualization, through VPYTHON library. A script named "main.aut" allows, at the user to configure the simulation, whereas the initial positions of the object are initialized through "pos.aut" file. The correlation function, are set through the file "potmem.aut", where the first number (the negative one) corresponds at the function put in "calculate.c", and other numbers are the parameters "k", with the last number the radius, of the action potential. The positions of the dynamics are in the file "posres.aut". To have more detailed information, please to write an email to "paolopoli1980@gmail.com". The simulations, were carried out with a laptop, with an I7 intel core, and 4 Gb of ram. .

## Conclusions

In this paper, MISSP was presented, through some simulations. We saw how is possible to create an own code, to simulate correlated agents, using an own approach and a simple laptop. The examples shown, could be observed in real world with structures such as cells, bee swarms, but they can be also abstracted dynamics. This was only a theoretical simulation, but it could be interesting to connect real agents using the MISSP method, to have real agents around us moving how the simulations showed in this paper.

## References

[1] https://github.com/paolopoli1980/missp