PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

FACULTAD DE FISICA

An Agent-Based Model governed by physical and biological principles to model Multicellularity

by

Sebastián Urrejola Barrios

Tesis presentada a la Facultad de Física de la
Pontificia Universidad Católica de Chile, para
optar al grado académico de licenciado en física.

Supervisor	:	Dr. Timothy James Rudge (PUC Chile)
Co-supervisor	:	Dr. Edgardo Dörner (PUC Chile)
Comision	:	Dr. Ignacio Espinoza (PUC Chile)
		Dr. Juan Keymer (PUC Chile)

June, 2019
Santiago, Chile

"What I cannot create, I do not understand."

Richard Feynman

On his blackboard at the time of death, February 1988

Acknowledgments

We are grateful to the people of 'el cardumen' for the scientific conversations and contributions they made to us. Also to Tim Rudge who gave us the opportunity of making this work.

Abstract

There is a variety of scenarios where multicellularity evolved. The main idea of this investigation is to study how does simple rules of interaction between agents can lead to a multicellular-like collective behaviour, in an Agent-Base Model (ABM). Here we generated an ABM with agents that interact with each other via repulsive or attractive forces. With these forces and with a measurements of their local neighbourhood, the agents are able to differentiate and/or divide, generating a complex system in which a multicellular-like behavior can be found. Our main results are the frequency of clusters sizes distributions and that is possible to obtain simple rules for the interaction between agents in a ABM leading to a collective multicellular-like behaviour.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Multicellularity:

Looking forward to generate a model about multicellularity, we would like to start talking about the ideas and abstractions of biological systems that we want to study. First, multicellularity refers to organisms with more than one cell, more precisely, more than one type of cell (Becker et al, Wayne M. 2009) and is an event that has occurred independently several times in the evolutionary history of life (Grosberg, RK; Strathmann, RR 2007). It could come from and individual cell that divide via mitosis until a point where the cells start differentiating forming the multicellular organism, or from different independently cells that come together to then start working together, eventually differentiating and dividing labours among the cells in order to form the multicellular organism. By differentiating and dividing their labours, individual cells manage to altruistically work together and depend on each other to survive. With this, multicellular organisms can simultaneously partition complementary tasks among different cells, providing a larger pool of labourers comparing it with unicellular organisms(Grosberg, RK; Strathmann, RR 2007).

Considering a group of different specialised of cells that work together and that can go through internal processes more efficiently than non-specialised cells. The group of cells that work together will be in a favourable competitive position, comparing it with a group of just clustered cells that are near each other but where each one still work independently. The group with cells that work together will be able to reach ecological

niches than individual cell couldn't reach, giving us the idea that at the small spatial scales where individual cells live, multicellularity is not necessarily the most efficient lifestyle, but when we consider bigger spacial scales than those that an individual cell can reach, it is indeed more efficient(Bonner, John Tyler 1998).

In multicellular organisms, when everything is working as it should, the cells of the organism behave in an altruistic way, working in order to keep the whole organism alive. Also we can see that individual cell are not so important for the whole organism, if one dies, the organism could keep living, showing that this kind of organisms are robust to perturbation on their local agents i.e. the individual cell (Stelling, Jörg; Sauer, Uwe; Szallasi, Zoltan; Doyle, Francis J., Doyle, John (2004)).

1.2 Agent-Based Model and Particle-Life model:

In Agent-Based Models we have individual agents that interact with each other allowing us to study the dynamics between the agents in the system, looking to get an idea of their collective behaviour (Hiroki Sayama 2015). In our system the agents will be represented by particles and dynamics for the agents will be based on biological and physical ideas. With this our aim is to study the clustering of the agents measuring proximity between them.

We base our work on a model called 'Particle - Life' (HackerPoet 2018), the source code is uploaded to github (also there is a link to a youtube video that explains how does the system works). Giving an insight on how does the model works, Particle-Life is an agent based model (it could be considered as a particle-game of life), where the agents are self-propelled particles that observe their neighbourhood in order to updates its information, unlike Cornway's game of life, in the Particle-Life model the agents are placed in a continuous two dimensional finite space and the updated information of the agents are the speed and movement direction (the velocity of the agent). Continuing, in the Particle-Life model we have different types of agents where each type is represented with a particular colour. To update its information, each particle will observe its neighbourhood and then interact with the other particles on it. These interactions could be attractive

or repulsive forces (depending on the colours of agents and radius between the interacting agents), resulting in a change on the velocity of the observing agent. Also, the interactions between all types of particles are randomly picked at the beginning of the simulation.

Here we can present two frames from the Particle-Life model,

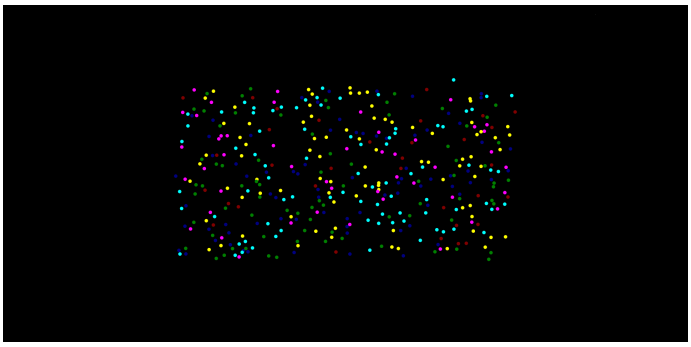

Figure 1.1: "Initial state of the Simulation". First frame, of a Particle-Life run.

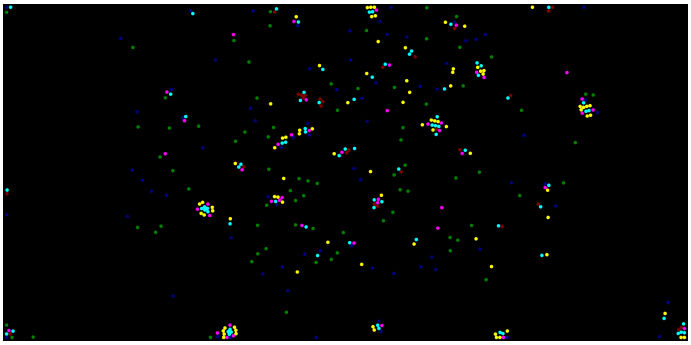

Figure 1.2: "Evolution of the Simulation". Second frame, a few steps later from the same run of the first frame.

Because of the randomness of how are picked the interaction between the particle types, these particles do not conserve the energy of the system. For example the force that a Blue particles feels in proximity of a Red particle will be different to the force that the Red particle feels in proximity to the Blue one, leading to different changes in the particles velocities, in this case the momentum will not be conserved and therefore the energy is not conserved. In other words, by picking random interaction forces between the different types of agents, we are going to have a non reciprocal way of interacting between different types of agents, this could be understood as a system that does not follow the third law of

Newton, Action and Reaction.

To fix this, is added a viscous drag to the system. By doing this the particles will feel a force proportional to its velocities and in the opposite direction of their movement, loosing energy when they move.

Because we are interested on multicellularity, we added some features to the agents of the Particle-Life model in order to represent a more cellular-like behaviour. In our model the agents can divide, like in cellular division, and also they can change their type, like in cellular differentiation or a mutation, finally we added a random walk for the agents that could be understood as a noise parameter in the motion of the agents.

As a summary, in our model we have different types of self-propelled particles. These particles feel attractive or repulsive forces to other particles, leading to interesting self-organised dynamics. Finishing, in our model the particles are able to divide introducing new particles to the system, and also they are able to change their type, changing its own possible interaction with the other particles, both of these events controlled with a parameter that is used to measure a minimum time between these functions in a particle.

All the codes were wrote in Python3.7.2 and are uploaded to github. There will be a link for the animations of some of the simulations done for this work (S.Urrejola 2019).

1.3 Hypothesis:

Considering the model and concepts describe above, our hypothesis is that in an agent-based model, with rules of interaction for the agents based in physical and biological ideas, we will be able to find configurations of the controllable parameters of the code, where the system replicates a multicellular-like behaviour.

1.4 Objectives:

With all this said, our objective are:

- Generate an agent-based model with interaction rules for its agents based on physical and biological implications, aiming to obtain a platform to study multicellularity.

- Analyse the clustering characteristics of the agents, searching for the information of how long the cluster could keep alive and how the frequency of the clusters sizes is distributed.
- Study how does the control parameters of the code, change the collective behaviour of the agents in order to obtain a lists of parameters that lead to multicellular-like behaviour.

Chapter 2

Methods

Here we present the methods used in the code to compute the dynamics of the particles and how do we obtained the clustering results.

The interactions between agents in the system are considered as forces,therefore, in order to compute the particles motion we need to solve differential equations. To do this we are going to use the forward Euler method (B.A. Stickler; E. Schachinger 2014). This method is used to compute first order differential equations, with it we will be able to compute the change in the velocities of the particles from the forces of the system by solving $\vec{F} = m\vec{a}$. After we obtain the velocity of the particle for the next simulation step, we calculate the change in the position of the particle using the new velocity.

For the clustering measurements, we are going to use an algorithm based on iterative deepening depth-first search (Korf, Richard 1985). This algorithm is used to search on graphs or search-trees by going through the connections of a node and then going deeper in the connections of the nodes obtained in the step before, and so on. This method is going to be used to reach all the particles connected in a cluster.

2.1 Forward Euler Method

This method consists on taking a first order differential equation and then discretise the variable in which we are differentiating. With this we approximate the differential

equation to a series of finite differences. Consider the following differential equation,

$$\frac{dy}{dt} = F(y(t)) \tag{2.1}$$

, under the discretisacion $t_n = t_0 + nk$, with the integer $k \geq 0$. Each step in this grid will be of size n, now considering the notation $y(t_k) = y_k$ and $F(y_k) = F_k$ we obtain,

$$\frac{\Delta y_k}{\Delta t} = F_k \tag{2.2}$$

, taking the Δ's as the difference between two consecutive points in the grid, and then solving for we y_{k+1} we obtain,

$$y_{k+1} = y_k + F_k * n \tag{2.3}$$

With this equation to obtain the value of y_{k+1} is just needed y_k and F_k, showing as that this method is an explicit one ,i.e., in order to obtain the information of the next step, is just needed the information of the previous step, permitting us to loop this algorithm eventually obtaining all the possibles y_k's from what we had at the beginning, $y(0) = y_0$ and $F(y(t))$. Is important to recall that in this method every time we loop the algorithm to obtain the info of the next step, we are going to have an error in the approximation proportional to n^2, where n is the size of the step.

2.2 Iterative Deepening Depth-First Search

In a few words, for this method of search we pick a node of the graph, then we go one step deeper into the connected nodes to the originally picked node, after this we then go another step deeper in the graph, getting the information of the nodes connected to the nodes connected to our originally picked node. Looping the action of going one step deeper,

we will eventually have visited all the nodes that can be reachable from our original node.

As an example, consider the following graph

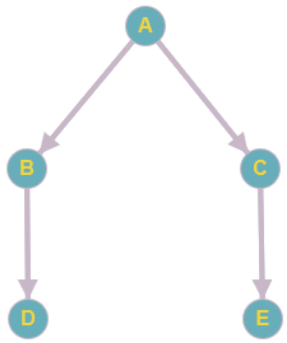

Figure 2.1: "Diagram of a graph". The circles are the nodes and the arrows between them, are the vertices or connections between the nodes.

Starting the search in the node **A** we are going to obtain just the information of this particular node. Then we go one step deeper, obtaining the information of the nodes **A**, **B** and **C**. Going one step deeper we are going to obtain the information of the nodes **A**, **B**, **C**, **D** and **E**.

To obtain the clustering data with this algorithm, we are going to check the neighbours of all agents and then iterate over the observation of the neighbours of the neighbours until we get all the connected agents at all possibles depths. With this we are going to obtain a list of objects with the information of the clustered particles, these objects will be called *cluster*. Is important to keep in mind that these objects (clusters) will not share particles between them, being closed groups of particles obtained by reaching all the neighbours of the neighbours and so on, of each agent.

Chapter 3

Results

3.1 The Model

In this section, we will explain the main objects and functions of the model, with the aim of giving an idea of the dynamics of the systems.

3.1.1 Particles:

The agents of our model are going to be circular particles. Because we are always going to have a countable amount of particles, we use an index to identify them, then the $agent_j$ would be the j-th agent who entered the system. All of them are going to have the same mass m and radius R.

Each particular agent will have its own pair of (x, y) spacial coordinates, its own pair of (v_x, v_y), speeds in each direction, and its own colour. They will be randomly placed in the system with also a random movement direction and speed. In presence of other particles they will feel attractive or repulsive forces, this interaction will depend on the distance between the particles and the colours of the interacting particles. Also all the agents will have the same radius of action to define their own neighbourhood. If an agent finds another particle within a distance smaller than the *action radio*, then the agent will interact (feel the force) with the other particle, changing its own velocity.

Finally, the agents have a "Life counter", parameter that increases each time a step in

the simulation finishes. We use this parameter to control the division and the differentiation functions of the agents.

3.1.2 Clusters:

We are going to consider that a particle is clustered when it has been interacting to at least one other particle for a given interval of steps, without changing the types of both particles.

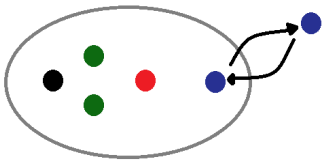

Figure 3.1: "Diagram of a cluster". The blue particles could be changed keeping the essence of the cluster thus maintaining the cluster 'alive'.

To measure this phenomenon we are going to take a measurement of the neighbourhoods of the particles at two different moments, so we can compare the neighbours that each particle could see in both instants. The first measurement will be taken on an step, lets say s_0, saving the information of the neighbours of each agent in the system. Then at the step $s_1 = s_0 + \Delta s$, we are going to take the second measurement of the agents neighbourhood. With these measurements we can compare the local neighbourhood of each particle obtaining the information of which particles have not change their colours and have kept interacting together in the given Δs interval of simulation steps. Locally, the particles just need to observe and interact with the same types of particles in a Δs to be considered as a clustered particle.

Finally, we are going to let the cluster won and lose particles in order to track the information of the particles changes in the clusters. We are going to consider that a cluster will stay alive if it loses less than half of the particles it had, and the cluster will also stay alive if it won less particle than the amount of particles it had before.

3.1.3 Particle-Life Force:

As explained in the introduction, the forces could be attractive or repulsive forces. The kind of force (attractive or repulsive) and its amplitude (how strong the force is) are randomly picked at the beginning of each simulation. All the forces will have this kind form:

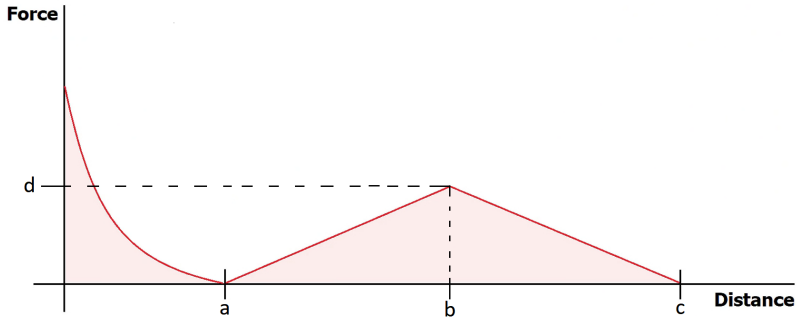

Figure 3.2: "Force diagram". In this figure we show an example of a repulsive interaction force between the agents.

, to generate this kind of forces we just need the three parameters a, b and d. With them we manage to intersect a parabola with only one solution at the point $r = a$, with an straight line with slope $m =$, and then this line is intersected with another straight line at the point $r = b$, but now with slope $-m$- Finishing, this last part of the force will be a line that will eventually intersect the X-axis at the point $r = c$. With this, the equation for our force is as follows,

$$F(r) = \begin{cases} \alpha * ((r-a)^2) & \text{for } 0 \leq r \leq a \\ m * r + n_1 & \text{for } a \leq r \leq b \\ -m * r + n_2 & \text{for } b \leq r \leq c \\ 0 & \text{for } c \leq r \end{cases} \tag{3.1}$$

, the α term is there just to control the size of the first part of this function, and the parameters c, n_1, n_2 and m are obtained from the trio (a, b, d) using the following relations,

- $c = -a + 2 * b$
- $c_1 = \frac{d*a}{a-b}$
- $m = \frac{d}{b-a}$
- $c_2 = \frac{d*c}{c-b}$

From the trio (a, b, d) the values of b and d, also the sign of the last one, will be randomly picked for all the pairs of colours that we can make given an amount of possible colours. To obtain the value of b we are going randomly pick a number from the interval $[a, ActionRadio]$, where *Action Radio* is the maximum radio for an interaction between agents. Then for d we are going to pick a random number from the interval $[0, a^2]$, and for the sign of d we are going to use a probability gate. First by picking a random number from the interval $[0, 1]$, then if this random number is smaller than a given threshold γ, d will be negative, and if the number is bigger than γ, d will be positive, leading to and attractive or a repulsive force respectively. Before going on some implications about the randomness of the forces, we are going to explain how they act on the particles.

The first part of the force, in the interval where $0 \leq r \leq a$, the force will be repulsive representing the radius of the particle. This section of the force is to avoid the overlapping of the agents. Because all the particles have the same radius, for all the interaction forces we are going to have the same value for a. The following point where the force will be stronger is at the point $r = b$, the type of the force depends on the sign of d. Therefore, if d is positive, the force will be repulsive and for a negative d the force will be attractive.

For all forces we are going to have a minimum in the amplitude in two points, at the point $r = a$ and also for $r \geq c$. In the case of $r \geq c$ the force will no longer act on the particle, but in the case of $r = a$ we have an equilibrium point. For the attractive forces, this point is stable but for the repulsive forces it will be unstable.

Is important to notice that for a system with k amount of particles we need a k^2 amount of trios (a, b, d) to have all the possible interactions. Because of the randomness of how we build the forces, between different particles, unless they are of the same type, they are going to feel different forces leading to different changes on their own velocities. In other words, the interaction between particles of different colours won't be symmetrical, because the forces are picked randomly at the beginning of each simulation.For example, lets say

that we have two particles. In this first case they will be of the same type, lets say both of them are Blue particles. If the random force between blue-blue particles is attractive, then the particles will get near each other. In the other hand, if the random force between blue-blue particles is repulsive, then the particles will distance themselves from each other.

Now, thinking in a scenario with two particles of different types, lets say a Red one and a Blue one, so the force that each particle is going to feel will be different. If both forces are attractive, then the particles with get near each other, and if both forces are repulsive, then the particles will repel each other until they leave each others neighbourhood, but in the case where one force is attractive and the other one is repulsive we have three possible outcomes. Lets say that the force that the Red particles feels in presence of a Blue particle, denoted by $F_{r,b}(r)$ where r is the distance between the particles, is attractive and that the force that the Blue particle feels in presence of a Red particle, $F_{b,r}(r)$, is repulsive.

In the first case, if we have that $|F_{r,b}(r)| > |F_{b,r}(r)|$, i.e. the attractive force is stronger than the repulsive force, then the attractive force will dominate resulting in that both particle getting near each other.

In the other case, where we have $|F_{r,b}(r)| < |F_{b,r}(r)|$, i.e. the repulsive force is stronger than the attractive force, we have two possible situations. If $|F_{r,b}(r)| << |F_{b,r}(r)|$, then the Blue particle will feel a very strong repulsive force making it get away from the Red particle. The other possible situation is that, if we have $|F_{r,b}(r)| < |F_{b,r}(r)|$ but with $|F_{r,b}(r)| \sim |F_{b,r}(r)|$, then the Blue particle will try to move away from the Red particle, while the Red particle will try to get near the Blue one, resulting in pursuit.

Here we present an example of the interaction between agents,

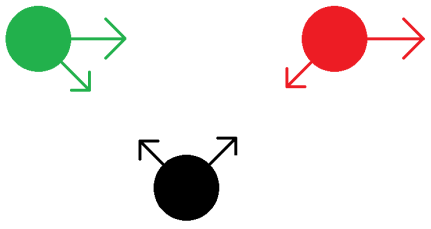

Figure 3.3: "Diagram of a Interaction ". Three particles of different types starting to interact.

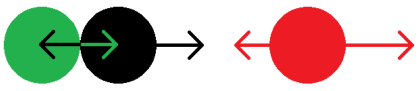

Figure 3.4: "Diagram of the outcome of the Interaction".The same 3 particles after letting them interact.

The arrows coming from the center of each particle represent the forces that each particles feels because of the presence of the other ones. In this case, the Green particles are attracted to the Red and the Black particles, the Black particles are attracted to the Red and the Green ones, and finally the Red particles are attracted to Black particles and repelled from Green particles.

The Black and Green particles will get near each other. With the movement of the group of particles. To get the situation of the second figure 3.4, the attractive forces that the Black particle feels to the Red particle have to be stronger than the attractive force that the Black one feels to the Green particle. Also the repulsive force that the Red particle feels to the Green one, can not be too strong so that the red does not go and dissolve the group.

3.1.4 The Random Walk:

Like a heat reservoir for a particles system, from where the particle can gain or lose energy. The agents of our model have a random walk function that will consist on adding a random change in the velocity of the particles, proportional to a 'noise' parameter.

3.1.5 Cellular Division:

In order to obtain a system with agents capable cellular-like behaviour we added a division function based on mitosis or clonal growth. Per step we are going to randomly pick one particle to go though this event. After picking the random agent, we are going to go through a probability gate to see if we start the division process. This probability gate consist on picking a random number from 0 to 1, if this random number is smaller than a given threshold then the gate is opened and the event can occur. If this gate is passed we check if the Life counter of the agent passes a given threshold. If it passes, a new agent will be added to the system next to the old agent, also the Life counter of the old agent goes to zero. The new agent in most cases will be of the same type (colour) of the original one, but in a few cases (0.01% of the time) it will be of a random type. This last idea was added thinking in the idea of mutations events in the process of mitotic division (clonal growth).

3.1.6 Cellular Differentiation:

The purpose of this function is to have agents capable of mutating and changing their possible interactions. The idea is that in biology we can find systems where individual and independent cells get added together forming multicellular organisms with cells that work together and that they even depend on each other to survive. To achieve this point, the original cells need to change their way of working. This event of a change in the original agents is what we are trying to emulate. These mutations are going to be taken as a change in the colour of the agent.

Like in the division function, we are going to use a probability gate to see if the agents can differentiate, but now we are not going to pick just one particle, we are going to go through all the particles checking if they can differentiate.

In order to differentiate our particles will need to be clustered for a while. The idea is that agents that are together for a few simulation steps will be able to mutate, changing their colour and also its possible interactions with its neighbours. To check if some particles where together for a few steps, lets say Δs, we need to save the information of the neighbours of the particle at a step, lets say s_0, and at the step $s_0 + \Delta s$, comparing these two measurements the agent will be able to know if it saw the same types of particles in both measurements. With this and because of what we are considering as a cluster, the agent will be able to know if it is locally clustered. If it is indeed clustered, and the Life counter of the agent is bigger than a given threshold, the agent will be able to change its own colour, mutating into an other type of particles with different interaction with its neighbours.

The change of type of the agent will be determined with a set of mutation rules randomly picked at the beginning of the simulation, and with the types of the agent going though differentiation and with the colours that the agent could saw in the cluster. Like in the forces, here each combination of pairs of types will result on a random type, but fixed for the simulation. With this we are breaking again an observational symmetry, for example if we have two particles of different colours we have different outcomes depending on which particle is the one that is observing and which one is the observed. To work out the mutations rules we use a notation of tuples where we put the type of the observer

agent in the first position and the type of the observed agent in the second position, equal to the new colour for the observer agent (the observer agent is the one going through differentiation):

$$(colour_{observer}, colour_{observed}) = NewColour \tag{3.2}$$

If both particles are of the same colour, then the interaction will be symmetrical, but if we have two different colours, lets say Blue and Red, the mutations rules for this situation will be

$$(Blue, Red) = c_1 \text{ and } (Red, Blue) = c_2 \tag{3.3}$$

,where c_1 is the colour to which a Blue particle will mutate in presence of a Red particle and c_2 is the colour to which a Red particle will mutate in presence of a Blue particle. Because c_1 and c_2 are randomly picked at the beginning of the simulation, it is probable that $c_1 \neq c_1$, in these cases the outcome of this interaction will depend on which particle is the observing one.

We made this function with a restriction related to the amount of particles in the system. In the system, for this function to occur, we put that there must be at leas the half of the maximum amount of particles that the system will have. With this we give the system a time to evolve until the agents starts changing their type.

This differentiation event does not have more restrictions, so there is the possibility on a neutral mutation leading to possible clusters that are not affected by this event. With 'neutral mutation' we mean that sometimes, this random new colour for the agent could be the same colour the agent already had. For example if we have, using the mutation rules explained above, that $c_1 = Blue$, then the Blue particles will mutate to the colour that they already have in presence of Re particles, finishing the differentiation process with the same pair of types of agents that we had at the beginning. This neutral mutations sometimes leads to clusters that will not have particles that change their colour (for example with the rule "$(Blue, Blue) = Blue$"), also this kind of mutation lead to clusters that are more resistant to perturbations due to mutations leading to more stable clusters that manage to stay alive, without major changes for longer periods of time.

3.1.7 Particles Dynamics:

Finally, returning to the particles dynamics. Our model has discrete time, letting us compute our differential equations using an Euler explicit method. With this we are able to compute the changes in position and velocities of the particles. In each simulation step we will calculate the changes in the agents velocity from the interaction with other agents, from the random walk and finally from the viscous drag.

At the beginning of each step for a particle we will have its position r_t and a velocity v_t, then we will compute the the velocity that the particle will have at the beginning of the next step, v_{t+1}. Finally, with this we calculate r_{t+1}.

Agent-Agent Interaction Forces

From now on, we are going to work out the equations that affects the dynamics of the j-th agent at a moment t of the simulation. This agent will interact with each i-th particle on its local neighbourhood N_j. Then the change in velocity from to this interaction will be given by,

$$\begin{aligned} m_j * \vec{a}_{t,j} &= \vec{F}_{N_j t,j} \\ &= \sum_{i}^{N_j} \vec{F}_{c_j,c_i}(|\vec{r}_{t,j} - \vec{r}_{t,i}|) \end{aligned} \tag{3.4}$$

, where the index i denotes the identity index of the particles in the neighbourhood N_j of the j-th particle, and c_j, c_i are the colours of the j-th and i-th particle respectively. Although all the particles have equal mass, we wrote m_j to emphasise that the observing agent is the j-th one. Using the Euler method the change in velocity from the agents interaction forces $\vec{v}_{F,j}$ is going to be computed as follows,

$$m_j * \frac{(\vec{v}_{F,j} - \vec{v}_{t,j})}{\Delta t} = \sum_{i}^{N_j} \vec{F}_{c_j,c_i}(|\vec{r}_{t,j} - \vec{r}_{t,i}|) \tag{3.5}$$

, rearranging this solving for $\vec{v}_{F,j}$ we obtain that the change in velocity of the j-th agent at a step t from the interaction forces will be,

$$\vec{v}_{F,j} = \frac{\Delta t}{m_j} \sum_{i}^{N_j} \vec{F}_{c_j,c_i}(|\vec{r}_{t,j} - \vec{r}_{t,i}|) + \vec{v}_{t,j} \tag{3.6}$$

Random Walk

To calculate the random walk we will use two random numbers, ξ_1 and ξ_2. This operation could be considered as a thermal force due to the random interaction between the particles and its environment. The value of ξ_1 will be randomly picked from the interval $[0, noise]$ where *noise* is our control parameter and it represents a speed. Then ξ_2 is randomly picked from the interval $[0, 2\pi]$ and it represents the angle of this random movement. Once we have picked the random numbers, the change in velocity due to the random walk will be given by,

$$\vec{\xi}_{t,j} = \begin{bmatrix} \xi_1 \cos \xi_2 \\ \xi_1 \sin \xi_2 \end{bmatrix} \tag{3.7}$$

Viscous Drag

Finally, we have to compute the change in velocity from the viscous drag, for spherical particles the viscous drag will be give by,

$$\vec{F}_D = -6\pi R \eta \vec{v} \tag{3.8}$$

, where η is the viscous coefficient and R is the radius of the particle. Using the Euler

method we obtain that the change in velocity from this factor will be,

$$\begin{aligned}\vec{v}_{D,j} &= \vec{v}_{t,j} - \frac{\Delta t}{m_j} * 6\pi R * \vec{v}_{t,j} \\ &= \vec{v}_{t,j}\Big(1 - \frac{\Delta t}{m_j} * 6\pi R\Big)\end{aligned} \tag{3.9}$$

, where $\vec{v}_{D,j}$ is the final velocity of the j-th particle after calculating the viscus drag. In contrast to the force obtained by G.G. Stokes, we are going to absorb the constants $(\Delta t/m_j)6\pi R$ into a new parameter μ. Using these parameters is easier to control how much speed the particles loses when the move.

Final Motion Equation

After this, to put together all the changes in velocities, first is computed the $\vec{v}_{F,j}$, then is added the random walk term $\vec{\xi}$, and finally is computed the viscous drag. This way of working lead us to the following equation for the particles velocity,

$$\vec{v}_{t+1,j} = (1-\mu)\left[\frac{\Delta t}{m_j}\sum_{i}^{N_j}\vec{F}_{c_j,c_i}(|\vec{r}_{t,j} - \vec{r}_{t,i}|) + \vec{\xi}_{t,j} + \vec{v}_{t,j}\right] \tag{3.10}$$

Finally, we compute the new position for the particles using the Euler method on the equation,

$$\vec{v}_{t+1,j} = \frac{(\vec{r}_{t+1,j} - \vec{r}_{t,j})}{\Delta t} \tag{3.11}$$

, solving it for $\vec{r}_{t+1,j}$ we obtain

$$\vec{r}_{t+1,j} = \vec{r}_{t,j} + \vec{v}_{t+1,j} * \Delta t \tag{3.12}$$

3.2 Parameters list:

With all this said, in this section we are going to present the list of parameters that control the systems dynamics, giving an insight on how does the code works with them, and how are we going to obtain the results from the simulations.

In order to get a better understanding of the parameters, we are going to group the parameters in two lists. For the first one we are going to show the parameters that will be constant through all the simulations. On the other hand, the parameters that are going to change between, or inside, a simulation will be placed in the second list.

3.2.1 Fixed Parameters

- Colours $\rightarrow$ Number of possible colours for the particles.
- Steps $\rightarrow$ Amount of steps of the simulation.
- $\gamma \rightarrow$ Probability threshold to pick the type of the forces. (Attractive or Repulsive)
- Boundary $\rightarrow$ Size of the system.
- Mass $\rightarrow$ Particles mass.
- $\mu \rightarrow$ Effective Viscous Drag coefficient.
- $\Delta t \rightarrow$ Time discretisation for the internal dynamics of the system.
- $\Delta s \rightarrow$ Amount of steps between the clustering measurements.
- Action Radio $\rightarrow$ Maximum radius of interaction between agents.
- Clustering Radius $\rightarrow$ Minimum distance between agents to consider them as a cluster.

The model works with different types of agents, each one represented by its own particular colour, the parameters ***Colours*** is an integer number that represent the amount of different possible colours for the particles. As noted in the list, the ***steps*** is the amount of steps of the simulation, we noticed that the computational time scales linearly with

this parameter. The ***boundary*** parameters the number that we use to define the limits of the space for the particles, for the case of the Y-axis, we are going to have the interval $\big[0, boundary\big]$, but in the case of the X-axis we are going to have the interval $\big[0, 2 * boundary\big]$.

The parameter γ is the probability threshold that we are going to use to pick the type of the forces. The ***mass*** parameter, apart from been the mass for the particles, it could be taken also as a parameter to modulate the intensity of the forces that the particles feel, with this we could relate this parameter to the particles colour, leading to a system with different types of particles, each type with its own particular mass, and therefore each type with different reactions to the interaction forces. As said in the section 3.1.7, the parameter μ is our effective viscous constant, by absorbing the drag force constants in this way, we have a parameter that goes from 0 to 1 permitting us use it to control the percentage of the particles velocities is lost when they move.

The Δt is the parameter we use for the time discretisation of the Euler method and Δs is the interval of steps between the two neighbourhoods measurements for the clustering information and the differentiation event, also, we are going to use this parameter as the interval of steps between the frames of the animations of the system.

For the interactions between agents, each particle observes into a circle of radius equal to ***Action radio*** centred in the position of the observing particle, this parameter is also directly involved in the computational time each simulation takes, because it modulates how much interactions an agent could have. In the case of the ***Clustering radius***, this parameter is the minimum distance between agents to take them as clustered, considering that restriction and the rule *'the neighbour of my neighbour is my neighbour'*, we obtain all the connections between the agents of a cluster. The *Clustering radius* is smaller than the *Action radio* because when particles interact, if they cluster they will stay at a distance smaller than the *Action radio*, so using a *Clustering radius* smaller than *Action radio* we are going to be able to get the particles that are indeed clustered.

3.2.2 Variable Parameters

- $N \rightarrow$ Number of particles in the system.

- Initial Colours → Initial number of possible colours for the particles in a simulation.
- Noise → Random Walk parameter. ('Heat')
- θ_{div} → Division life counter Threshold.
- θ_{dif} → Differentiation life counter Threshold.

The parameter $\boldsymbol{N}$ indicates the amount of particles in the system, because we have the division function, in the simulations where this function is turned on the system will have a variable number of particles. In those systems, N go between the limits of the interval $\left[N_0, N_{max}\right]$, where N_0 is the initial amount of particles of the system and N_{max} is the maximum amount of particles for the simulation. In the other scenarios where the division function is turned off, $\boldsymbol{N}$ is constant and also we will have $\boldsymbol{N} = N_0 = N_{max}$.

The parameters ***Initial Colours*** is an integer number that will indicates the amount of different colours of particles in the initial conditions of the system. We put it as a variable parameter because with the differentiation function, a system starting just from one type of agents, could evolve to a system with all the possible colours, with this if the differentiation function is turned on, the amount of colours of the system will change until it reaches the parameter ***Colours***, changing the spectrum of possible forces in the simulation.

As explained in 3.1.7, the ***Noise*** parameters is the number that modulates the maximum random speed change that a particle could get from the random walk function. Is important to notices that if this term if bigger than the maximum of the interaction forces, $\alpha * a^2$ then the particles interactions will be 'less dominant' in contrast to the random walk, then the agents motion will be dominated by the random walk, leading to an ideal gas like behaviour. Finally the parameters θ_{div} and θ_{dif} are the threshold for division and differentiation functions respectively. In both of these functions have gates to check if the agent starts passing through the process. By comparing these parameters to the agents life counters, if the life counter is bigger than the threshold, then the process starts and the life counter of the agent goes to zero. Because the life counter all of agents increases in +1 on each step, these thresholds represent a relaxation time for the particles, in other

words, these parameters set the amount of steps the agents have to wait to be able to go though a process.

3.3 Results:

Now we are going to present the results of our simulations. We want to study how does our differentiation, cellular division and the random walk functions affects the clustering behaviour of the particles in the Particle-Life model, so we are going to start obtaining the clustering info of our model with the parameters tuned to represent the Particle-Life model. We are going to obtain the cluster size frequency distribution, the average life spam of a cluster and the mean cluster size for each simulation scenario.

To test the functions of random walk, differentiation and division we are going to work in 4 different simulation scenarios. The first one is going to be our code representing the Particle-Life model, then in the second scenario we are going to add the random walk function to the Particle-Life model. For the third scenario we are going to have the random walk and the differentiation functions working, and as it could be expected, for the forth scenario we are going to have the random walk, differentiation and division functions turned on.

Before we go to the results of the simulations, here is the list of the fixed parameters with their respective values. Because of the long computational time each simulation took, for each scenario we ran 5 simulations and then we took our average values of interest.

- Colours = 7
- Steps = 20000 (20K)
- $\gamma = 0.5$
- Boundary = 20
- Mass = 1.
- $\mu = 0.9$

- $\Delta t = 0.01$
- $\Delta s = 50$
- Action Radio = 1.5
- Clustering Radius = 0.7

After obtaining the results of each scenario, we are going to compare the clusters size frequency distribution, the average life spam of a cluster and the mean cluster size results from the Particle-Life scenario with all the other scenarios.

3.3.1 Particle-Life Mode Results:

To have the Particle-Life model with our code, we have to turn off the functions for random walk, differentiation and cellular division. In this simulation scenario, our system is going to have a constant amount of particles with just interactions between them via the interaction forces explained above.

Here we present the total plot of cluster size frequency vs size of the clusters, of the Particle-Life mode in our code.

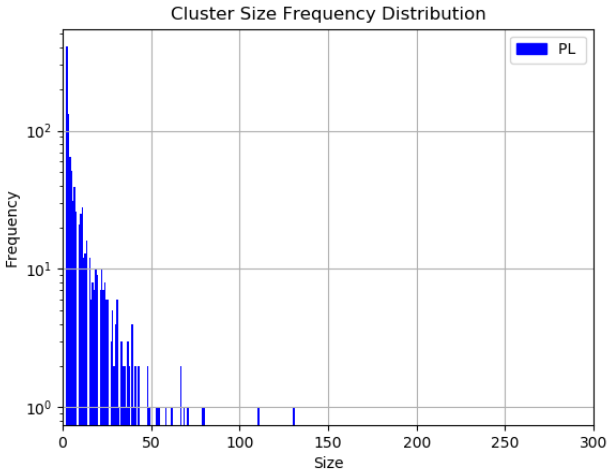

Figure 3.5: "PL Size Frequency plot". Histogram showing the amount of cluster of each size that the Particle-Life scenario had. Logarithm of the amount of clusters of each size in the Y-axis, and the sizes in the X-axis.

In the Y-axis of the last plot we have the logarithm of the amount of clusters that the scenario had of each size, and the X-axis are the sizes that the clusters could have, where the minimum size of a cluster is of 2 particles and the maximum is a cluster of 300 particles. We are going to have these configurations for the axes for all the clusters size frequency plots.

For this scenario we obtained a mean size for the clusters of 8.06 particles with and standard deviation of 11.56 particles. In the case of the average life spam of a cluster for this group of simulations we obtained a mean value of 59.05 frames with an standard deviation of 100.66 frames, and the variable parameters of the simulations of this scenario where

- N = N_0 = N_{max} = 300
- Initial Colours = 7
- Noise = 0
- $\theta_{div} >$ steps
- $\theta_{dif} >$ steps

With this, each of this simulations had a constant amount of particles (300) and the simulations started with all the possible colours. By adjusting the noise parameter to zero there will be no random walk, also, with θ_{div} and θ_{dif} being bigger than the amount of steps of the simulations, the particles won't be able to pass these thresholds so there will be no differentiation or division in this scenario.

Now we present 4 frames from one of the simulations made for the Particle-Life scenario.

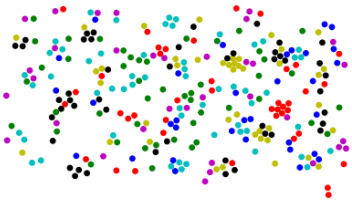

Figure 3.6: "Frame = 1, Particle-Life scenario". First frame of the simulations, here we have the initial conditions of the system.

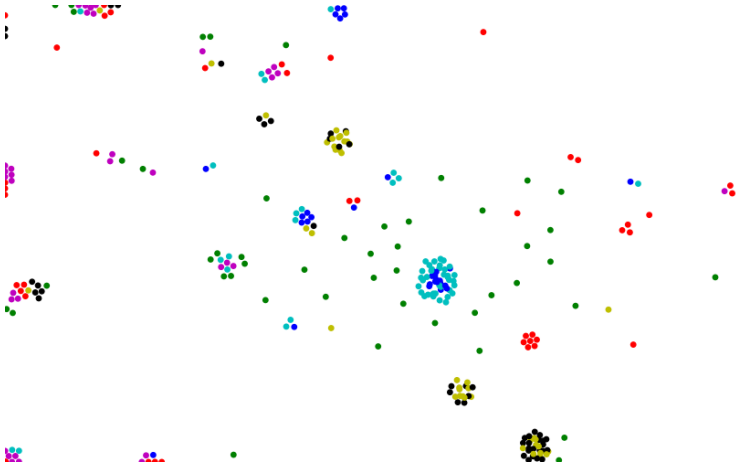

Figure 3.7: "Frame = 100, Particle-Life scenario". This frame is from the first quarter of the simulation.

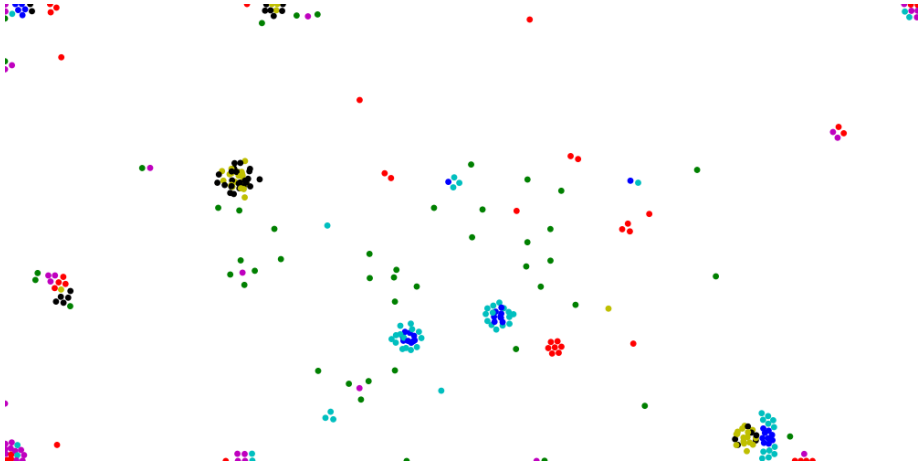

Figure 3.8: "Frame = 200, Particle-Life scenario". This frame is from the second quarter of the simulation.

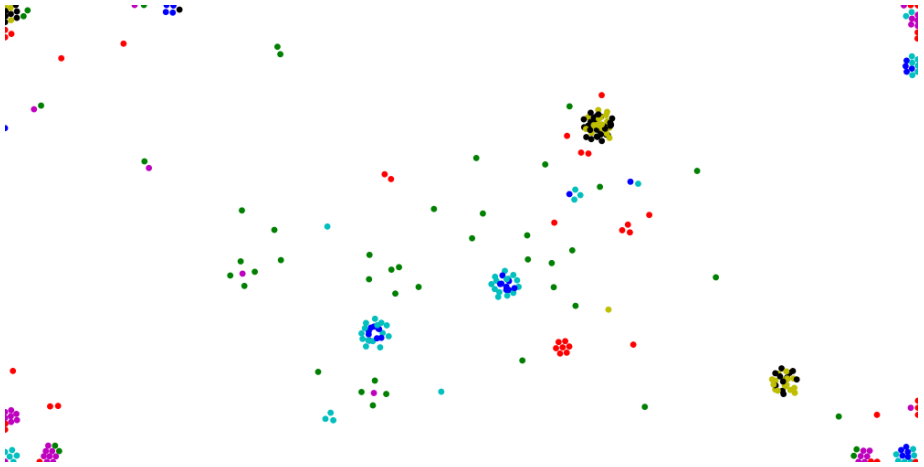

Figure 3.9: "Frame = 400, Particle-Life scenario". This is the last frame from the simulation.

3.3.2 Particle-Life + Random Walk Results:

In this scenario we added the random walk to Particle-Life model.

For all the simulations done here to obtain the clustering information with the random walk function, the noise parameter was tuned to 25. This leads to systems where the interaction forces were still the dominant influence over the motion of the agents, but with this random motion each agent moved in a more *cellular-way* to the directions that the forces leads them to go.

Here we present the plot of cluster size frequency vs size of the clusters, of the Particle-Life with random walk scenario,

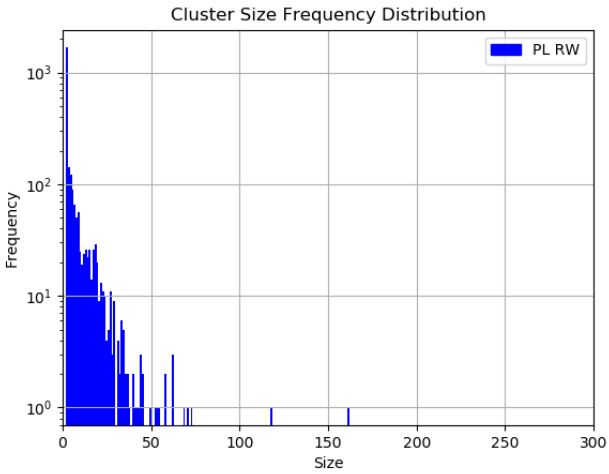

Figure 3.10: "PL + RW Size Frequency plot". Histogram showing the amount of cluster of each size that the Particle-Life with the Random Walk function scenario had. Logarithm of the amount of clusters of each size in the Y-axis, and the sizes in the X-axis.

With the Random walk turned on, we obtained a mean size for the clusters of 5.56 particles with an standard deviation of 8.33 particles. The average life spam for a cluster in this scenario was 20.29 frames with an standard deviation of 53.43 frames, and the list of variable parameters with their respective value was

- N = N_0 = N_{max} = 300
- Initial Colours = 7
- Noise = 25
- $\theta_{div} >$ steps
- $\theta_{dif} >$ steps

Here the division and differentiation functions are still turned off, so the number of particles will be constant and also the simulations will start will all the different possible colours.

Now we present 4 frames from one of the simulations made for the Particle-Life with Random Walk scenario.

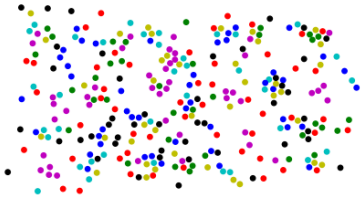

Figure 3.11: "Frame = 1, Particle-Life with Random Walk scenario". First frame of the simulations, here we have the initial conditions of the system.

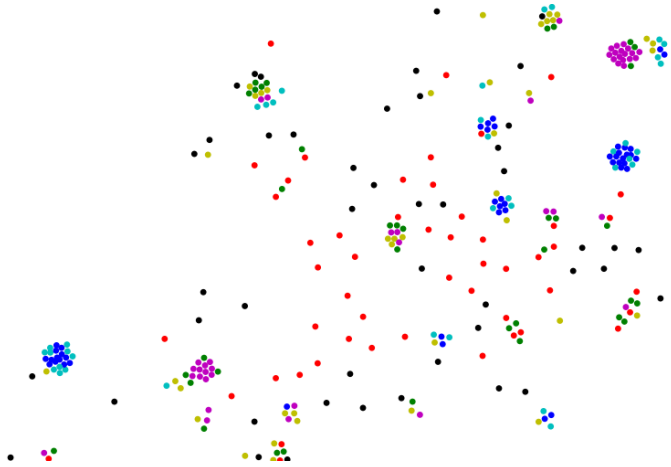

Figure 3.12: "Frame = 100, Particle-Life with Random Walk scenario". This frame is from the first quarter of the simulation.

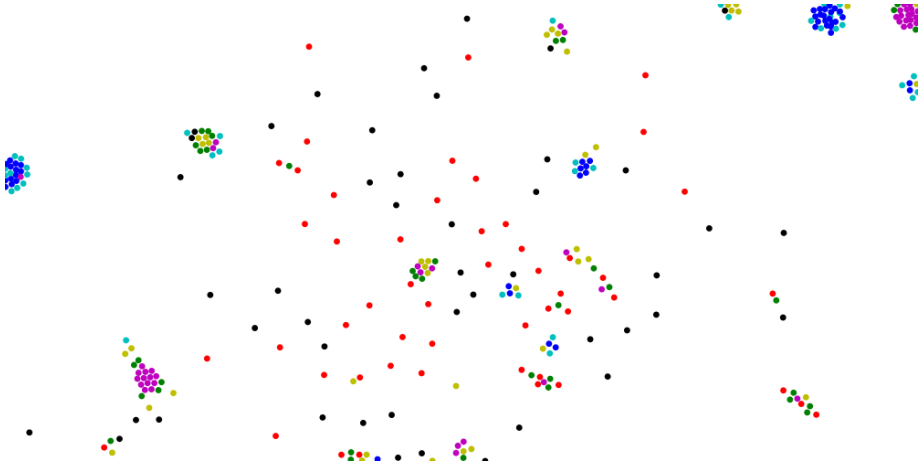

Figure 3.13: "Frame = 200, Particle-Life with Random Walk scenario". This frame is from the second quarter of the simulation.

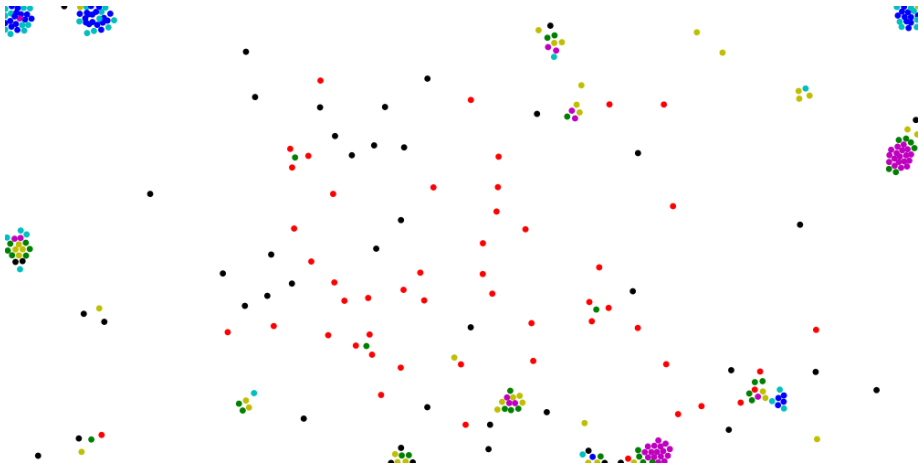

Figure 3.14: "Frame = 400, Particle-Life with Random Walk scenario". This is the last frame from the simulation.

3.3.3 Particle-Life + Random Walk + Differentiation Results:

In this scenario we added the random walk and the differentiation functions to the Particle-Life model. Here we could tested the system with the parameter *Initial Colours* = 1, but because we wanted to be sure that, we were going to have attractive and repulsive forces from the beginning of the simulations, we put the *Initial Colours* to be equal to 7, the maximum amount of colours for the system.

In this scenario the differentiation event will be triggered when a group of particles

get together forming a cluster for at least Δs amount of steps and if the probability gate and the life counter threshold are passed by the agent. With this, every time a particle mutates, the cluster will have a change in the internal forces that have build it up. This change in the internal mechanics of the cluster could easily lead to breaking the clusters, because of the instability of the agents types in the cluster.

We obtained the following graph for the clusters size frequency distribution,

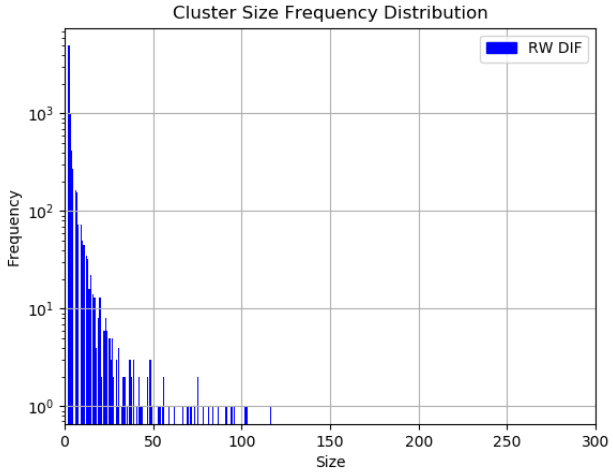

Figure 3.15: "RW + DIF Size Frequency plot". Histogram showing the amount of cluster of each size that the Particle-Life with the Random Walk and Differentiation functions scenario had. Logarithm of the amount of clusters of each size in the Y-axis, and the sizes in the X-axis.

In this scenario we obtained a mean cluster size of 3.56 particles with an standard deviation of 5.64 particles. For the average life spam of the clusters, we obtained 12.88 frames with a standard deviation of 28.79 frames, and the variable parameters used for these simulations were,

- N = N_0 = N_{max} = 300
- Initial Colours = 7
- Noise = 25
- θ_{div} > steps
- θ_{dif} = 500

Here the division function was still turned off, but the random walk and the differentiation were on. For the random walk, like before, our noise parameter had a value of 25, and the differentiation threshold was tuned to 500, meaning that if a particles goes through differentiation, that same particle will have to wait at least 500 steps to go again through this process. In this scenario there is no division, so the amount of particles is constant in the system.

Now we present 4 frames from one of the simulations made for the Particle-Life with Random Walk and Differentiation scenario.

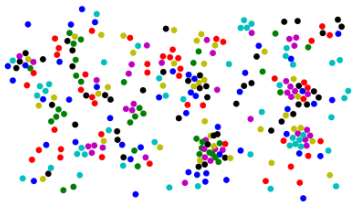

Figure 3.16: "Frame = 1, Particle-Life with Random Walk and Differentiation scenario". First frame of the simulations, here we have the initial conditions of the system.

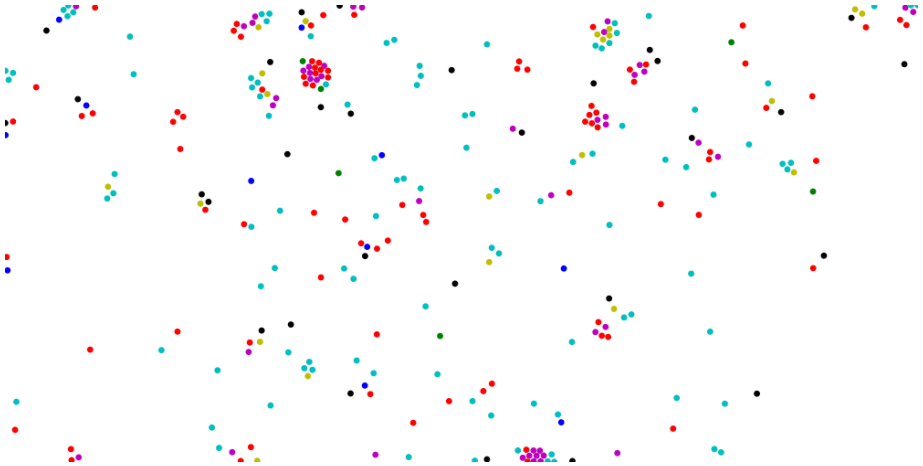

Figure 3.17: "Frame = 100, Particle-Life with Random Walk and Differentiation scenario". This frame is from the first quarter of the simulation.

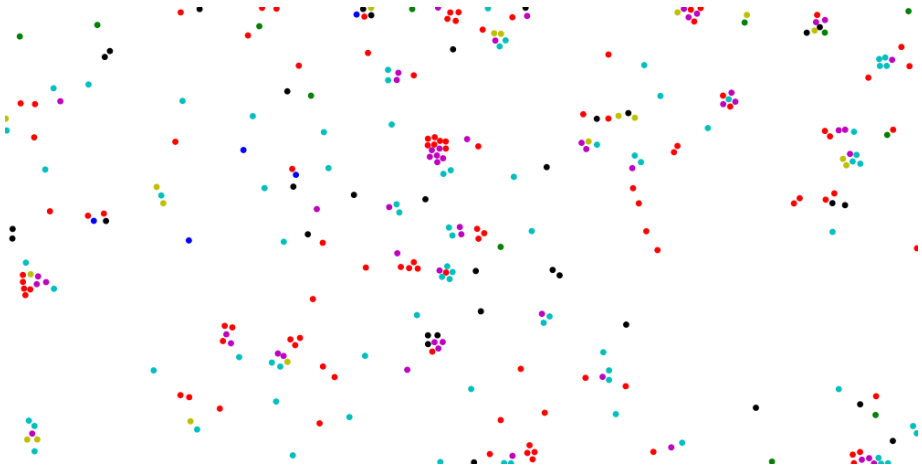

Figure 3.18: "Frame = 200, Particle-Life with Random Walk and Differentiation scenario". This frame is from the second quarter of the simulation.

Figure 3.19: "Frame = 400, Particle-Life with Random Walk and Differentiation". This is the last frame from the simulation.

3.3.4 Particle-Life + Random Walk + Differentiation + Division Results:

For this particular scenario, we are going to use three different settings. Because here the division and differentiation functions are turned on, we are going to have both of the threshold in action and both of them act on the same counter (the agents life counter). Because of this we are going to test the scenarios where we have $\theta_{div} = \theta_{dif}$, $\theta_{div} > \theta_{dif}$ and finally $\theta_{div} < \theta_{dif}$.

Each of these thresholds θ_{div} and θ_{dif} could be thought as *life counter costs* for the division and differentiation processes respectively. When an agent goes through either of these events, the agent will lose its life counters restarting them to zero, therefore the scenarios mentioned above, could be thought as, a scenario where the division and the differentiation processes have the same cost for the agents ($\theta_{div} = \theta_{dif}$), other scenario where differentiation cost less than the division ($\theta_{div} > \theta_{dif}$), and a scenario where the division cost less than differentiation ($\theta_{div} < \theta_{dif}$) leading to systems where, in terms of life counters, it would be easier to divide or to differentiate changing the behaviour of the agents.

It is important to remember that for the differentiation event to occur, the amount of particles in the system N, must be at least half of the maximum number of particles in the system.

Because all the simulations in these scenarios started with 1 particle, we are not going to show the first frame, it didn't make sense to show plots with only 1 dot.

Equal Cost for Division and Differentiation.

In this case the differentiation and the division functions have the same life counter cost for the agents.

In this scenario, the clusters size frequency distribution was,

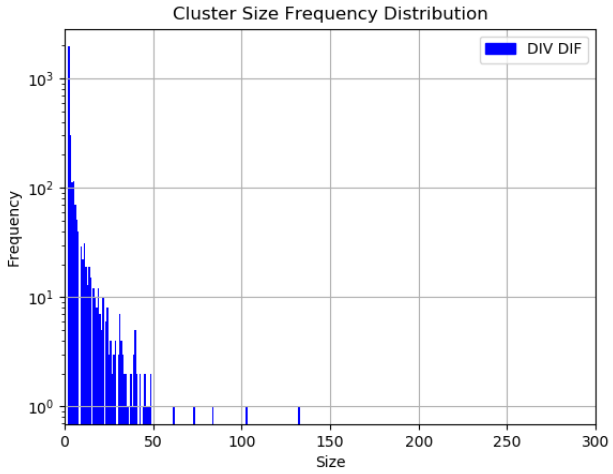

Figure 3.20: "DIV + DIF Size Frequency plot". Histogram showing the amount of cluster of each size that the Particle-Life with the Random Walk, Differentiation and Division functions scenario had. Logarithm of the amount of clusters of each size in the Y-axis, and the sizes in the X-axis. Here for the threshold we had $\theta_{dif} = \theta_{div}$.

The mean cluster size for this scenario was 4.33 particles with and standard deviation of 6.86 particles. For the average life spam of the clusters we obtained 9.50 frames with a standard deviation of 23.33 frames. The variable parameters used for these simulations were,

- $N_0 = 1$, $N_{max} = 300$
- Initial Colours = 1
- Noise = 25
- $\theta_{div} = 500$
- $\theta_{dif} = 500$

Finally with all the functions turned on. These systems started with just 1 particle and therefore 1 colour. We kept the noise parameter on a value of 25, and in this case, the costs of division and differentiation were the same and equal to 500, meaning that the agents will have at least 500 steps to go through division or differentiation after going through one of these events.

Now we present 4 frames from one of the simulations made for the Particle-Life with Random Walk, Differentiation and Division scenario, $\theta_{div} > \theta_{dif}$.

Figure 3.21: "Frame = 100, Particle-Life with Random Walk , Differentiation and Division scenario.". This frame is from the first quarter of the simulation. $\theta_{div} = \theta_{dif}$.

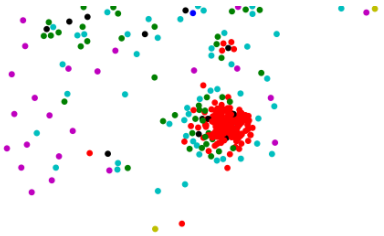

Figure 3.22: "Frame = 200, Particle-Life with Random Walk, Differentiation and Division scenario". This frame is from the second quarter of the simulation. $\theta_{div} = \theta_{dif}$.

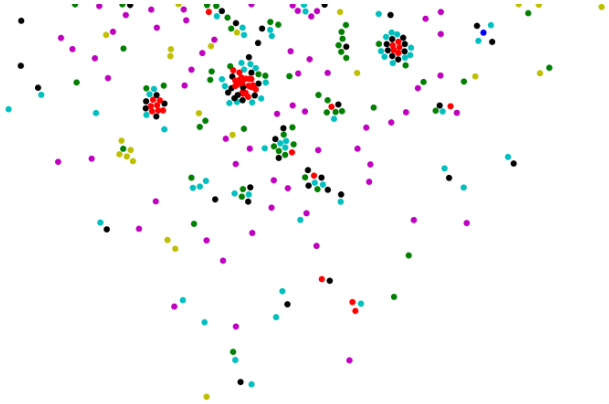

Figure 3.23: "Frame = 400, Particle-Life with Random Walk, Differentiation and Division scenario". This is the last frame from the simulation. $\theta_{div} = \theta_{dif}$.

Differentiation costing more.

In this case the differentiation event had a higher cost than the division event for the agents, meaning that after the amount of particles in the system reaches the half of the maximum amount of particles (in this case 150 particles) the division event will "easier" to execute for an agent.

In this scenario, the clusters size frequency distribution was,

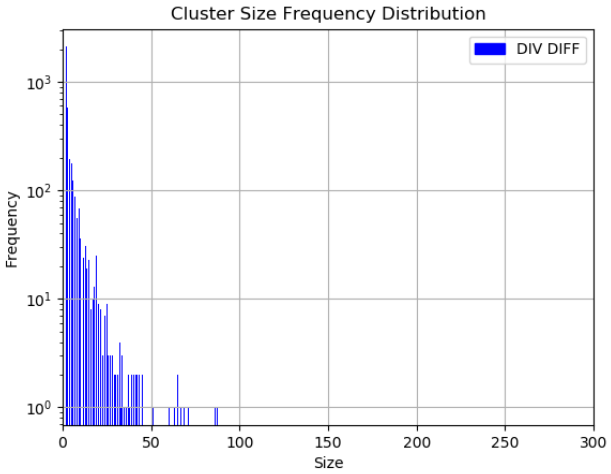

Figure 3.24: "DIV + DIFF Size Frequency plot". Histogram showing the amount of cluster of each size that the Particle-Life with the Random Walk, Differentiation and Division functions scenario had. Logarithm of the amount of clusters of each size in the Y-axis, and the sizes in the X-axis. Here for the threshold we had $\theta_{dif} > \theta_{div}$.

For these simulations we obtained a mean cluster size of 4.42 particles with an standard deviation of 6.26 particles. The average life spam of the clusters obtained was 10.87 frames with a standard deviation of 27.23 frames, and the variable parameters used for these simulations were,

- $N_0 = 1$, $N_{max} = 300$
- Initial Colours = 1
- Noise = 25
- $\theta_{div} = 500$
- $\theta_{dif} = 750$

Like before, here we have all the functions turned on. These systems started with just 1 particle and therefore 1 colour. We kept the noise parameter on a value of 25, and in this case, the costs of differentiation was 750 and the cost for division was 500 life counters.

Now we present 4 frames from one of the simulations made for the Particle-Life with Random Walk, Differentiation and Division scenario, $\theta_{div} < \theta_{dif}$.

Figure 3.25: "Frame = 100, Particle-Life with Random Walk , Differentiation and Division scenario.". This frame is from the first quarter of the simulation. $\theta_{div} < \theta_{dif}$.

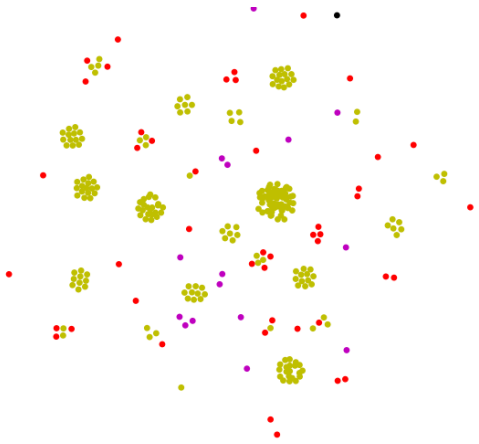

Figure 3.26: "Frame = 200, Particle-Life with Random Walk, Differentiation and Division scenario". This frame is from the second quarter of the simulation. $\theta_{div} < \theta_{dif}$.

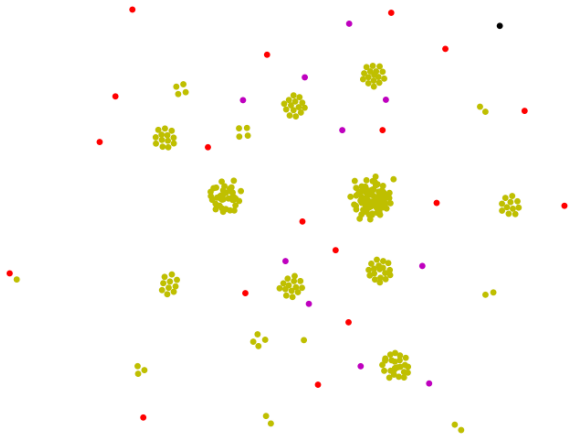

Figure 3.27: "Frame = 400, Particle-Life with Random Walk, Differentiation and Division scenario". This is the last frame from the simulation. $\theta_{div} < \theta_{dif}$.

Division costing more.

In this case the division event had a higher cost than the differentiation event for the agents. This leads to a system where increasing the size of a cluster is "harder" for the particles, compared to changing the particles that the cluster has.

In this scenario, the clusters size frequency distribution was,

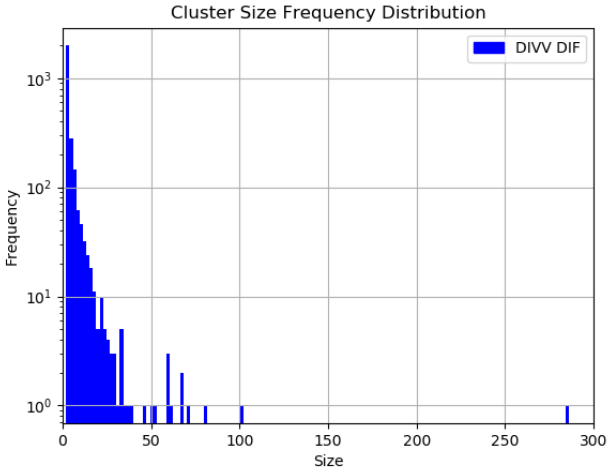

Figure 3.28: "DIVV + DIF Size Frequency plot". Histogram showing the amount of cluster of each size that the Particle-Life with the Random Walk, Differentiation and Division functions scenario had. Logarithm of the amount of clusters of each size in the Y-axis, and the sizes in the X-axis. Here for the threshold we had $\theta_{dif} < \theta_{div}$.

These systems had a cluster size mean value of 4.07 particles with and standard deviation of 7.94 particles,and an average life spam of the clusters equal to 12.74 frame with a standard deviation of 25.26 frames, and the variable parameters used for these simulations were,

- $N_0 = 1$, $N_{max} = 300$
- Initial Colours = 1
- Noise = 25
- $\theta_{div} = 750$
- $\theta_{dif} = 500$

Like before, here we have all the functions turned on. These systems started with just 1 particle and therefore 1 colour. We kept the noise parameter on a value of 25, and in this case, the costs of differentiation was 500 and the cost for division was 750 life counters.

Now we present 4 frames from one of the simulations made for the Particle-Life with Random Walk, Differentiation and Division scenario, $\theta_{div} > \theta_{dif}$.

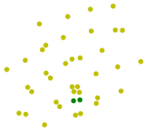

Figure 3.29: "Frame = 100, Particle-Life with Random Walk , Differentiation and Division scenario.". This frame is from the first quarter of the simulation. $\theta_{div} > \theta_{dif}$.

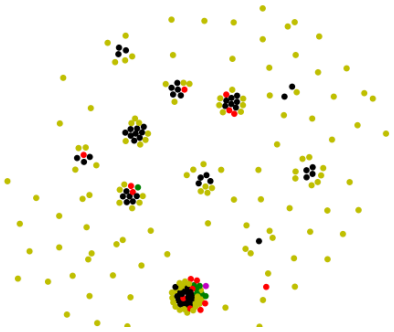

Figure 3.30: "Frame = 200, Particle-Life with Random Walk, Differentiation and Division scenario". This frame is from the second quarter of the simulation. $\theta_{div} > \theta_{dif}$.

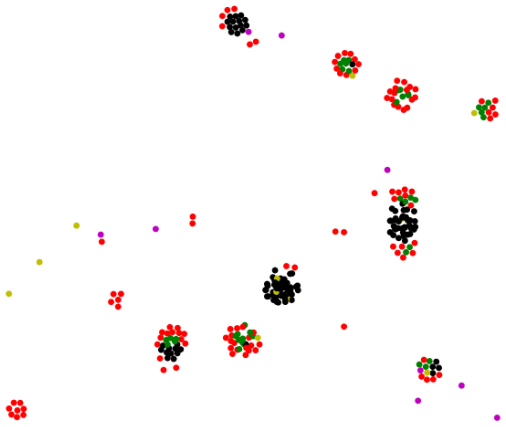

Figure 3.31: "Frame = 400, Particle-Life with Random Walk, Differentiation and Division scenario". This is the last frame from the simulation. $\theta_{div} > \theta_{dif}$.

3.4 Results Analysis and Discussion:

As we said before, we want to compare the clusters average life spams and the mean cluster size of the different scenarios to the results of the Particle-Life mode.

Here we summarise the results of the average life spam of a cluster and the mean cluster size per scenario in the following table,

Scenario	Average Life Spam (frames)	Mean Size (# of particles)
PL	59.05 ± 100.66	8.06 ± 11.56
PL + RW	20.29 ± 53.43	5.56 ± 8.33
RW +DIF	12.88 ± 28.79	3.56 ± 5.64
DIV + DIF	9.50 ± 23.33	4.33 ± 6.86
DIV + DIFF	10.87 ± 27.23	4.42 ± 6.26
DIVV + DIF	12.74 ± 25.26	4.07 ± 7.94

Table 3.1: Table of average clusters life spam and the mean cluster size in each scenario.

We have named the scenarios with the functions that each scenario is testing. PL refers to the Particle-Life mode, all the scenarios are based in that so we omit the 'PL' in some of them. RW refers to the Random Walk function, DIF to the Differentiation functions and DIV to the division function. In the case of the three scenarios with the differentiation and division functions working, but each with different threshold values for the functions,

the scenario with equal thresholds is DIV + DIF, the one with the differentiation costing more is DIV + DIFF and the one with the division costing more is DIVV + DIF.

To show the comparisons in a clear way, we are going to use the following table,

Scenario	Average Life Spam	Mean Size
PL + RW	34.36 %	68.96%
RW + DIF	21.81 %	44.26%
DIV + DIF	16.08 %	53.80%
DIV + DIFF	18.40 %	54.83%
DIVV + DIF	21.57 %	50.52%

Table 3.2: Table showing the percentage difference of the average cluster life spam and the mean cluster size of each scenario considering that the results of the PL scenario where the 100%.

With this we can say that by maintaining the Particle-Life parameters but also adding our functions, the average life spam of the clusters decreases, same for the cluster mean size.

Is also important to notice that in each scenario, the standard deviation of the average cluster life spams where bigger than the actual average value. This comes from the fact that in most of the cases, the simulation started with a cluster that lasted for almost the whole simulation, with this, the average life spam of a cluster per simulation had a big standard deviation compared to the average life spam obtained.

Finally, by comparing the different graphs obtained for frequency of clusters sizes vs the clusters size of each scenario, we can say that they look very similar, giving us the idea that independently from the amount of frames that a cluster could stay alive, the distribution of clusters sizes is maintained between scenarios, but further tests have to be done to be sure about this specific idea.

Is important to notice that the Particle-Life simulation scenario were the one with the bigger average cluster life spam and mean cluster size compared to the other scenarios. We think that this is because of the functions we added to the Particle-Life model. With the **Random Walk**, we added a new way of introducing energy to the system, with this is expected to have clusters that are more unstable, because the particles have more energy to broke them up. The **Differentiation** function has a direct influence in the clustering

events. Because we haven't controlled how much times an agent could differentiate, they could change their type indefinitely, leading to less stable clusters, as it could be thought from the results presented in all the scenarios with differentiation function in the percentages table 3.4. This is because when a particle in the cluster mutates, it will change the internal interactions of the cluster, and in most cases,it will dissolve the cluster into little clusters or juts un-clustered particles. Finally, the **Division** function could influence the clustering events, but it depends on the forces between the dividing agent an the new agent in the system, if they have attractive forces, they will get clustered, maintaining the cluster alive, but if they have repulsive forces they will distance themselves.

An unexpected result for us was that in the Particle-Life scenario we had by far less clusters formed in the 5 simulations (one order of magnitude just comparing the most abundant clusters, the ones with just 2 particles), compared to all other scenarios. This is because the particles that managed to cluster in the Particle-Life scenario, stuck clustered for a longer amount of steps and in bigger clusters than in all other scenarios, obtaining a bigger average life spam and mean size. This let us count more clusters, and in particular more smaller clusters, in all other scenarios.

Clusters:

In the different simulations, we found different clusters structures that repeated between different systems. We have categorise the clusters into 3 groups.

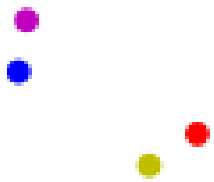

Figure 3.32: "Pursuit". Two pairs of particles where one of the particles is attracted to the other one, and the other particle is repelled by the other one.

In the first group we are going to have clusters of two particles making a *pursuit*, so the particles have to be of different types. We call these clusters the *Pursuit Clusters*.

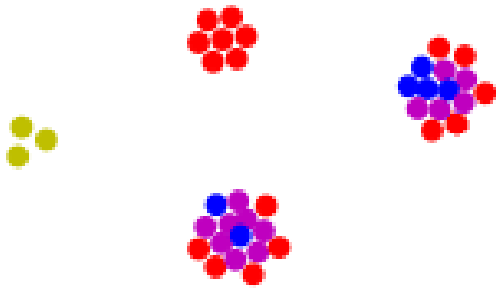

Figure 3.33: "Ring Clusters". Clusters of particles with only attractive forces making a cluster with a regular shape.

In this second group we have the *Ring Clusters*. These clusters are build up of particles that have mainly attractive forces between them, this leads to clusters that do not move. When particles of the same type have attractive forces between them, they normally form

these kind of clusters. Is notable that these clusters tend to regular and symmetrical shapes like the red or yellow clusters in the last figure.

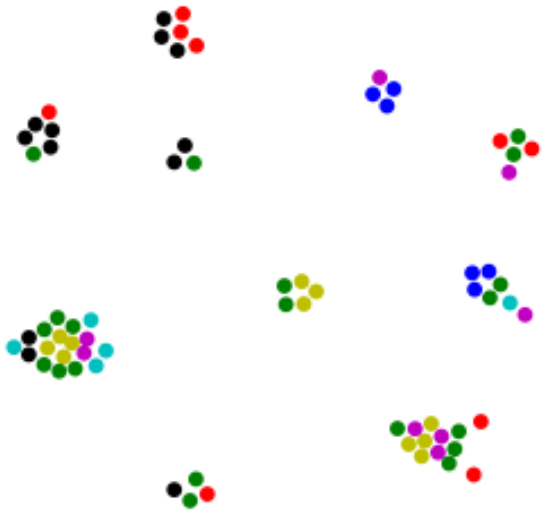

Figure 3.34: "Gliders". 10 different *gliders clusters* from different simulations. These clusters have different particles in combination of attractive and repulsive forces.

Finally we have the *gliders clusters*, as it can be seen in the figure above, these clusters are build up of different type of particles, they also have an elongated triangular-like shape. The gliders are the combinations of the other two types of clusters we just mentioned, so they are groups of particles attracted to each other but with some of them been also repelled, from this repulsion the pursuits that build up these kind of clusters let them move.

Computation Time:

Just by observing how long the simulations took, we made and approximation the computational time of a simulation with relation to the number of particles and the amount of steps,

$$ECT = \frac{Steps}{1000} * \frac{N^2}{100} \tag{3.13}$$

, where ECT is our Estimated Computational Time, in minutes, for a simulation with N particles and Steps amount of steps.

With this, each simulation we made ($N_{max} = 300$ and Steps=20K) took approximately 3 hours to compute, the simulations starting from $N_0 = 1$, took 2 hours to compute. This is without counting the time it took to run the codes to obtain the data from the simulations.

Chapter 4

Conclusions

Considering the objectives we had for this work, we can conclude that we managed to made our own version of the Particle-Life model and also add some features to change the behaviour of its agents, and some functions to obtain the data of the simulations, obtaining finally 4 codes that work together to make and obtain the data of the simulations. With this we could say that we generated a model where we can find cellular-like behaviour that it could be used to study multicellularity.

From the results is clear that the Particle-Life mode was the one with the bigger average life spam of the clusters and bigger mean size of the clusters, but as it can be seen from the different graphs of cluster size frequency, we can say that while the Particle-Life had bigger clusters that lasted for longer, in all scenarios we had a very similar distribution of clusters sizes. With this we think that the measure of the average cluster life spam doesn't says a lot about the behaviour of the system, it could be used as a measure of stability of the clusters. These two measurements could be related but is needed further testing.

For the parameters, with our results we can say that the functions we added to the Particle-Life mode decreased the stability of the clusters but we had clusters anyway. With this we conclude that while we have the fixed parameters with values like the ones used in here, we should see clusters. In particular, to obtain clusters from the systems starting from 1 particle, it would be preferable to have an attractive force between the particles of the colours of the first particle in the system, with this the simulation should start

with a big cluster until the differentiation event starts going on, changing the internal interactions of the cluster. It is notable that each frame of the simulations were 50 steps in the simulation, with this, by taking the mean value of the results of average cluster life spam, we can say that the clusters survived in average for 1044 simulation steps. Considering that all simulations had $20K$ steps, this result tell us that from simple rules and a set of fixed parameters, we could expect to have clusters surviving for at least a 5.22% of the simulation total time in our model.

Chapter 5

Summary and future work

As a summary, we made a model to study and represent multicellular-like behaviour. In the model presented here we have different types of particles that interact with each other via attractive or repulsive forces like in the Particle-Life model. In our case, these particles have functions for a random walk motion, for cellular division and a function for the differentiation of the particles. As it could be thought, from the random walk function the particles are able to move in random directions with random velocities, in the case of the cellular division function, the particles add new ones to the system, and finally with the differentiation function the particles can mutate by changing their own type, with this the possible interactions that the mutating agent could have with the other agents are changed. With all of this we are facing a complex system with interesting dynamics where we can find cluster of the agents and a cellular-like behaviour..

We have a lot considered for the future work on this model. In this work we couldn't test all the parameters of the system. For example, we know but we didn't measure that the γ parameter (the threshold used to have a proportion of attractive or repulsive forces) has direct influences on the clustering events, if we only have attractive forces is expected to have less but bigger clusters. Also the ratio between the total area of the system ($2 * boundary^2$) and the maximum amount of particles in the system (N_{max}), understandable as a density of particles in the system, is an important parameter for the clustering events that we didn't tested, with more particles been near each other, is

expected to have more interactions leading to the clusters eventually.

Also, is crucial to optimise the codes functions and run the code on a better computer. With this we could run bigger simulations in the same amount of time or more simulations of the size used in the simulated scenarios (size refers to the parameters N_{max} and $Steps$ of the simulation).

The rules from differentiation for now are just randomly picked, but to represent a 'real' differentiation event we should build these rules based on biology. For the probability gates associated to the differentiation event, we just picked a number while we were writing the code. The probability of a differentiation event should be related to the surrounding of the agent, not just the amount of steps that the agent haven't used its life counters. Also we think that the agents shouldn't go indefinitely through this event, there should be an amount of times each agent could mutate and/or terminal types of agents that do not change anymore.

For the division function we think we should change the way we are picking the agent to divide in each step, for now we are just picking randomly one agent from the system to see if it can go through division in each step until we get the maximum amount of particles for the simulation. We think that like in the differentiation event, here we should check if all the agents could divide in each step, leading to a faster and exponential growth of the system. Also for the probability gate of the division event, we think we should search in the literature for a biological base to get a probability for the division event.

We would like to add two more things to the system. First, we would also like to see if the model we made could be useful to study the evolution of multicellular systems. So we would like to give the particles an interaction with their environment, like make them search for food or search for other particles to get clustered, with this, the second thing we would like to add is a death function to the agents. We could get results for systems with apoptosis, a feature that a multicellular systems needs (Grosberg, RK; Strathmann, RR 2007), also the death of the agents related to environmental interaction could lead to a natural selection-like process, leading to a system where the population of particles evolve.

Because we know the interaction inside the system, we would like to get an statistical mechanics analysis over the system dynamics. We could also compute the energy of the

system in each step of the simulation using the momentum of the particles. Analysing the dynamics of the system from this perspective could help us get a better understanding of how are the local characteristics of a cluster, like the average radius between particles that are interacting in an stable way given a set of forces between the agents. By studying the mechanics of the system we could get a better idea of how to measure clusters and also what to measure from the clusters.

Finally, we would like to inspect which type of distribution follows the frequency of the size of the clusters. We thought that it could be a power-law, but when we see the histogram of the frequency of size of a cluster distribution of a simulation, for example lets see the plot of the scenario where we had the Particle-Life with Random Walk and Differentiation,

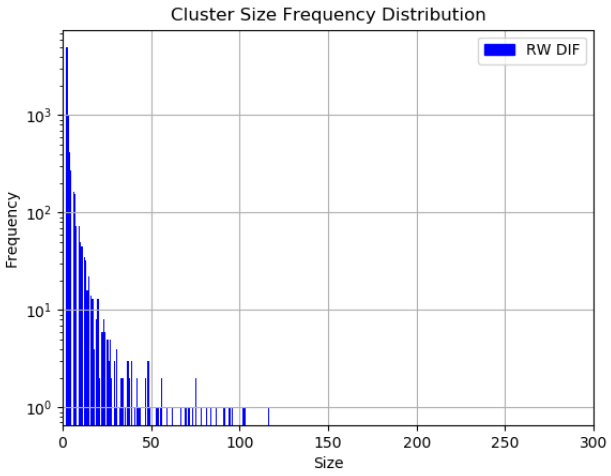

Figure 5.1: "RW + DIF Size Frequency plot". Histogram showing the cluster size frequency distribution of the Particle-Life with Random Walk and Differentiation scenario. Logarithm of the amount of clusters of each size in the Y-axis, and the sizes in the X-axis.

, this plot is the same one presented in 3.15. As it could be seen, this graph doesn't look like an straight line, giving us the idea that maybe it isn't a power law distribution, but we need further testing to know which type of distribution. Also we need bigger simulations to test this, if we keep working in these scenarios, our bigger cluster could be of just 300 particles and the maximum amount of clusters of this size is one. With bigger simulations we could gather the information of bigger clusters and more clusters of bigger sizes, with

more *Steps* for the simulations we could have clusters that last for longer. By making a bigger simulation we could also see if this distribution doesn't changes with the scale of the system, meaning that this measurement could be *scale-invariant*, important requirement for a power-law distribution.

Bibliography

Becker et al, Wayne M. (2009). The world of the cell. Pearson Benjamin Cummings.

Grosberg, RK; Strathmann, RR (2007). The evolution of multicellularity: A minor major transition?

Bonner, John Tyler (1998). The Origins of Multicellularity

Stelling, Jörg; Sauer, Uwe; Szallasi, Zoltan; Doyle, Francis J.; Doyle, John (2004). Robustness of Cellular Functions.

Hiroki Sayama (2015). Introduction to the Modeling of Complex Systems

B.A. Stickler; E. Schachinger (2014). Basic Concepts in Computational Physics.

Korf, Richard (1985). Depth-first Iterative-Deepening: An Optimal Admissible Tree Search

HackerPoet, Particle-Life, (2018),
GitHub repository, https://github.com/HackerPoet/Particle-Life

S. Urrejola, Thesis-Project, (2019),
GitHub repository, https://github.com/seburrejola/Thesis-Project-2019