Simulation of Aggregation of Surfactants in Hydrocarbon-Water Interfaces Utilizing Diffusion Limited Cluster Aggregation (DLCA) Model (Part 2)

DIEGO ROSENBERG[1,*] AND RICARDO PAREDES[2]

[1] Student of Engineering Physics, Universidad Iberoamericana, Student ID: 203766-2
[2] Departamento de Física y Matemátiacs, Universidad Iberoamericana, Prolongación Paseo de la Reforma, 880, Lomas de Santa Fe, C.P. 01219 Mexico City, Mexico
[*] Corresponding author: diegorosenberg@gmail.com

Compiled April 24, 2021

The purpose of this paper is to study the aggregation of sodium hexadecane-benzene sulfonate (8C16) in a hydrocarbon-water interface utilizing reversible Diffusion Limited Cluster Aggregation (Reversible DLCA), and observing the stabilization time as well as the structure of these aggregates. When observing the equilibration times it is important to consider the effects of density, as well as the effects of the pre-established probability of separation to better understand the long equilibration times of surfactant aggregation systems. Furthermore when observing the structural characteristics of these clusters it is important to study the fractal dimension, coordination number, and even in some cases the radius of gyration (since it can provide another way of calculating the fractal dimension). It has been demonstrated that when these clusters percolate the interface they cause a great deal of instability in the interface[1] which results in some of the surfactants being expelled from the interface. For these reasons it is of great importance to study the conditions in which the system goes from the floculattion regime to the percolative regime [2], since at this characteristic density the system will take even longer to find equilibrium. To study all of these qualities we have created a Reversible DLCA simulator in C that is capable of simulating clusters with different probabilities, sizes and particle numbers; with the aim of being able to create enough statistical data so that we can confidently analyze and obtain results for the aforementioned properties. From these results we observed that the aggregation of surfactants can be replicated by utilizing Reversible DLCA since the fractal dimension, as well as the coordination numbers are extremely close to the expected values. And have further reinforced the idea that the equilibration times of surfactant aggregation systems are incredibly long, as proposed by Ricardo Paredes (2018)[3]. Additionally, simulations in which particles/clusters could only aggregate to other particles/clusters if their coordination number was less than two, *i.e.* only connected to one particle initially, were run as a comparison method to observe if they more closely simulate the system of surfactant aggregation. However, this was discarded due to the fact that they require extremely low probability of separation values which practically makes them a normal DLCA simulation with an added restriction.

1. INTRODUCTION

Surfactants are a unique type of molecule that has amphillic properties, meaning that it has both hydrophillic and lipophilic properties, this makes it an incredibly useful chemical compound in multiple industries. Surfactants, in the industry, are normally placed in a system with two liquids that cannot mix, these then migrate to the interface between these two liquids and once they are there they begin to reduce the interfacial tension and decrease the interfacial energy, making it possible for the two liquids to mix. [4] This is due to the fact that it modifies the intefacial properties of the system by varying, in multiple orders of magnitude, the intefacial tension. One of the practical uses of surfactants is that it can reduce, in multiple orders of magnitude, the intefacial tension of a substance as well as modify the interfacial properties of the system.[3] Which is extremely useful in multiple industries such as paint, food, and even the oil industry[5, 6]. The process fro the surfactants to migrate to This is relatively long process, taking multiple miliseconds to arrive, once the surfactants arrive at the interface they diffuse within it. This diffusion process is also incredibly slow, in the order of multiple microseconds, since most surfactants begin aggregating inside the hydrocarbon-water interface [3]. To detect the time the surfactant system takes to reach equilibrium the evolution of the interfacial energy and tension of the system were observed and it was shown that this continues for multiple micro seconds until the interfacial parameters reach a stable point where they fluctuate between a certain mean value [3]. Independent of the clusterization of the surfactants, it was observed that when increasing the concentration of surfactants the interfacial tension decreases until it hits a critical density in which it falls drastically multiple orders of magnitude [3]. This point is known as the saturation concentration, this recieves this name since at this characteristic concentration any new surfactant introduced into the interface will cause the system to become unstable. For this causes the clusters to fold, generating a great deal of instability in the interface, causing multiple individual sodium hexadecane-benzene sulfonate (8C16) surfactants to be pushed out of the interface. For this reason it is imperative to observe the conditions with which the final clusters of the system percolate so that this instability point can be quantified. [1] The clusters that are formed during the aggregation process have been shown to have fractal properties. [7] It is important to quantify these qualities since the area covered by the molecules is inversely correlated to the interfacial tension. For all of these reasons it is of great importance to study the structural properties of these agglomerates as well as some of the microscopic attributes, including: fractal dimension, the correlation function, coordination number, as well as observe the densities at which these molecules begin to percolate the system.

The process of equilibration in these systems has been heavily disputed however, it has been previously shown, in a study performed by Ricardo Paredes (2018), that the equilibration time of these systems longer than previously studied, which was believed to be in the order of magnitude of a few nanoseconds, are actually in the order of magnitude of microseconds. This deeply affects previous studies on these systems, since the simulations did not reach the real equilibrium state of the system. [3] Knowing that the system has a much longer equilibration time than previously thought, and the necessity to study the fractal properties and the percolative properties of the system one must propose an alternative method to Molecular Dynamics (MD) to achieve a correct approximations of the experimental results for the aggregation of these surfactant molecules. For this we have chosen Reversible Diffusion Limited Cluster Aggregation (DLCA) as possible method to recreate the aggregation of surfactants in a hydrocarbon-water interface.[8–10] Irreversible DLCA simulations are usually applied to the aggregation of colloidal particles [11], the idea is to oversimplify the simulations seen in molecular dynamics to where every surfactant is estimated to be a single particle, and in the fluids it traverses are continuous. However, this oversimplification does not lead to exact results and therefore the reversible quality has to be added to achieve a better estimation of the aggregations of surfactants. The main reason reversibility is required to accurately describe the problem is due to the fact that most surfactants, including 8C16, do not have a potential well than $k_b T$. This small potential, and the thermic fluctuations in the real system, cause the 8C16 surfactants to break apart form already formed clusters and create smaller ones, to simulate this aspect reversibility is an extremely important part of accurately attempting to show the properties of the system.

In this paper we establish the Reversible DLCA model which was utilized to approximate the agglomeration of sodium hexadecane-benzene sulfonate surfactants in a hydrocarbon-water interface. For that reason the paper is divided into the following parts: section 2 of the paper tackles the theoretical framework required to understand the properties of the analyzed system, section 3 describes the computational complexities of the created program, section 4 shows a summarized version of the results from the irreversible DLCA and comparing to [7] results, section 5 shows the results obtained from the reversible simulations, and finally in section 6 conclusions are drawn.

2. THEORETICAL FRAMEWORK

A. Surfactants

Surfactants are a type of molecule that has a chemical structure with two different functional groups with different affinity within the same molecule. [12] To achieve this surfactans have an alkyl chain with 8-22 carbon atoms, which is the part of the molecule that shows hydrophobic properties, and an head which can have multiple elements comprising it. Within this there are two types of surfactants, ionic surfactants and non ionic surfactants. For the sodium hexadecane-benzene sulfonate, studied in this paper, it is known that this is an ionic surfactant which can be seen in figure. 1

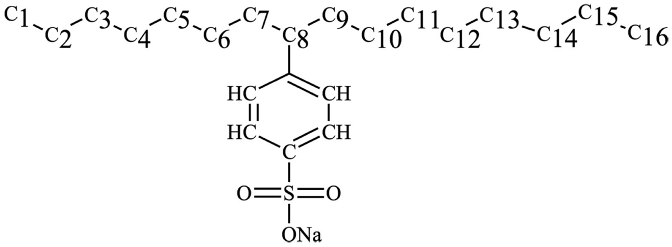


Fig. 1. Chemical structure of sodium hexadecane-benzene sulfonate.[3]

These surfactants are utilized to keep immiscible materials in liquid evenly mixed, or to help mixtures penetrate other materials. They are incredibly useful for lowering the intefacial tension between two liquids and acting as solubilizing agents, emulsifiers, antifogging agents, defoaming agents, etc. These makes them an extremely versatile type of material that is heavily utilized in multiple industries. Some examples of this are the soap industry in which surfactants are enlisted to help penetrate skin and hair so they can deeply clean and give better results, they are utilized in the makeup and pain industry to act as emulsifiers and solubizing agents for their numerous products [4, 5, 12], and surfactants are even utilized in the oil industry to better extract oil from materials due to their penetration ability.[6]

B. Choosing a Simulation Method

The most natural simulation model for these types of systems is Molecular Dynamics (MD). MD simulations calculate the forces acting on every particle and move all of the particles at once, utilizing Newtonian dynamics, for a given time differential. This precision causes MD to be extremely precise and simulate diffusion and aggregation of chemical molecules as if they were a real life experiment. However, since MD simulations have to calculate and move all of the particles/clusters at the same time, the computational requirements for large systems is extremely high. The computational time complexity is of $O(N * log(N))$, while this is not such a large number of operations for simulations with a small amount of particles it quickly climbs up when adding more and more particles resulting in ridiculously long times to complete a single simulation. With these limitations, a basic MD algorithm will not be able to approximate real life scale systems. Moreover, knowing that the aggregation of surfactans has an incredibly high stabilization time, in the order of microseconds [3], does not improve the prospects of MD for simulating the aggregation of surfactants, since MD has a characteristic simulation time in the order of picoseconds. Knowing this, it was important to choose a different simulation type that can accurately approximate the aggregation of sodium hexadecane-benzene sulfonate (8C16) surfactants in hydrocarbon-water interfaces, that can computationally handle extremely large systems with thousands of particles, and handle the large stabilization times seen in these systems. The primary methods that can substitute an MD simulation are the following: the Ballistic Model, the Reaction Limited Cluster Aggregation Model(RLCA), Irreversible Diffusion Limited Cluster Aggregation (DLCA), and Reversible Diffusion Limited Cluster Aggregation (DLCA Reversible)[8, 10].

B.1. Ballistic Model

The Ballistic model consists of a randomly generated set of particles or clusters placed in a grid. Of these, two clusters are chosen at random and utilizing a function called a kernel these two are either joined and moved together or another set of clusters is chosen. This model of extreme utility to study the kinetics of the system, *i.e.* the properties of the system during the evolution of the simulation, and has certain limitations when it comes to how different clusters are made.

B.2. RLCA

This process starts out similar to the ballistic model in which N particles are placed in a grid with certain specifications. From here particles partake in a random walk, following Brownian motion, as the move around the system. This leads them to eventually colliding, when one of these collisions happens a probability is computed which decides if the collision will result in the formation of a permanent cluster or if the particles will "bounce" off of each other and continue their random walks. This probability restriction causes simulations to take a slightly different form but eventually the system tends to one single cluster with all particles attached to it.

B.3. Irreversible DLCA

Diffusion Limited Cluster Aggregation (DLCA) is a simulation method in which N particles are placed in a 2D or 3D space and by utilizing Brownian motion these individual particles will move and connect with other particles. This type of simulation is appropriate for aggregation in a fluid with an attractive particle-particle interaction and potential wells that are much larger than $k_b T$ on short length scales. This type of simulation is extremely useful for approximating colloidal aggregations, like the ones found in aerogels and other extremely porous materials [11]. These types of simulation will always achieve a single final cluster as their final result and have a fractal dimension, in 2D at low densities, of around $d_f = 1.4$ [7].

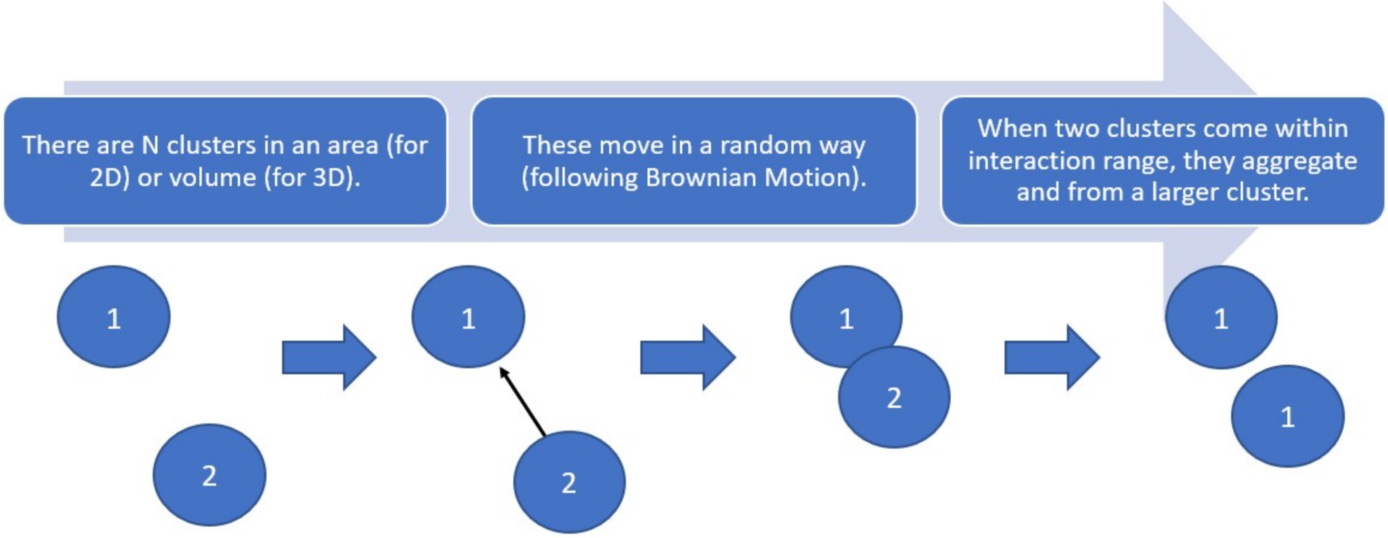

There are N clusters in an area (for 2D) or volume (for 3D).
These move in a random way (following Brownian Motion).
When two clusters come within interaction range, they aggregate and from a larger cluster.
1
2
1
2
1
2
1
1

Fig. 2. Flowchart of Irreversible DLCA simulation.

B.4. Reversible DLCA

Reversible DLCA follows the same base principals as the Irreversible DLCA however, this type of simulation is appropriate for aggregation of particles in a fluid with attractive particle-particle interaction with potential wells smaller than $k_b T$ in small distances. This small potential well results in the possibility of already formed clusters breaking into smaller clusters. This instability present in formed clusters is what has been observed in previous simulations in the aggregation of sodium hexadecane-benzene sulfonate surfactants in hydrocarbon-water interfaces [3], and will the main method of simulation to observe the behaviour of these aggregation processes.

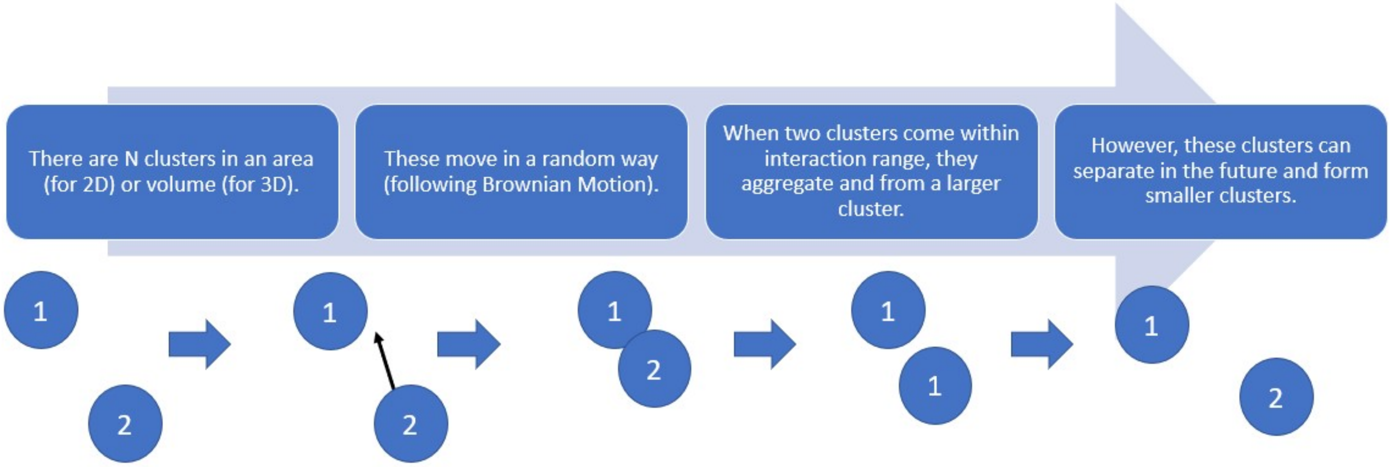

There are N clusters in an area (for 2D) or volume (for 3D).
These move in a random way (following Brownian Motion).
When two clusters come within interaction range, they aggregate and from a larger cluster.
However, these clusters can separate in the future and form smaller clusters.
1
2
1
2
1
2
1
1
1
2

Fig. 3. Flowchart of Reversible DLCA simulation.

B.5. Choosing a Model

To choose a model one must consider multiple factors, simulation times, flexibility to approach the real conditions as well as possible and the end result. Starting with the Ballistic model, there are kernel functions that emulate colloidal aggregation [8], however the strength of the Ballistic model lies in the study of the kinetics, and the evolution of the clusters during the simulation. This adds extra unnecessary complexities in the code which is not required for this specific study. Next is RLCA, this model has very good characteristics but the surfactant aggregation does not have the probability aggregation properties that RLCA proposes, and therefore will have different results. Next are the DLCA models, irreversible and reversible. Starting with the irreversible model, this model can be adjusted easily to include mass dependant diffusion, and has a small amount of intermediate complexities which results in smaller computation times. However, the surfactant aggregation does not have the k_bT potentials and therefore there is an extra step needed. Finally, observing the reversible DLCA model, this simulation type has all of the advantages of normal DLCA simulations, and has the added property of taking into account the smaller potential wells since clusters can disaggregate. For these reasons the best way to approximate the sodium hexadecane-benzene sulfonate aggregation in a hydrocarbon-water interface.

C. Fractal Geometry

Fractal geometry studies the geometric scaling and the symmetries associated with fractal objects. These characteristics are scale invariant which means that at any size of the fractal object they can be studied and are of value. Understanding the fractal geometry of these objects lays a pivotal role in understanding how the resulting fractals interact with fields, as well as their quantum properties, spectroscopy, crystallography, etc. In the case of DLCA the final formed clusters are known as a self-affine clusters, which is not identical in any given area but still has fractal properties. One of the characteristics of self-affine clusters is the fractal dimension, this characteristic is measured in experiments utilizing the correlation between the correlation function, found utilizing x-ray diffraction, and the fractal dimension. This can be reproduced in simulations through multiple methods, including the box counting method, utilizing the evolution of the average radius of gyration of all clusters, utilizing the correlation function (which closely approximates the real method of measurement), as well as other methods. The fractal dimension can give an idea of how the structure interacts with wave-particles, fields and can give us other physical properties of the structure.[8]

D. Percolation Theory

Percolation describes the conditions in which the system connects to itself, a common way to describe percolation is with the following thought experiment. if one was to pour a liquid on the top of a porous material, if the liquid is able to reach the bottom of the material then the water successfully percolates the system, since it can go from one end to the other end. [13] This is different from the percolation studied in this article but the thought experiment can help visualize the properties that are being looked at. For the simulations in this article what can be thought as the water in the previous experiment are actually the final clusters in the system, therefore if a cluster spans the entire system, either in the x or y direction, this cluster can be seen as percolating the system. In the study of surfactant aggregation this phenomena is incredibly important due to the fact that clusters that are said to percolate the system locally create a great amount of instability in the interface. [1] Observing figure 4 (b) we can clearly see that MD simulation of leaded to a very large cluster that not only percolates the system but bends the interface creating a great deal of instability. This instability results in the interface reacting by pushing the percolating cluster out of the interface (thereby hindering the stabilization process of the surfactants) and forcing them to once again travel to the interface and causing the system to always fluctuate out of equilibrium.

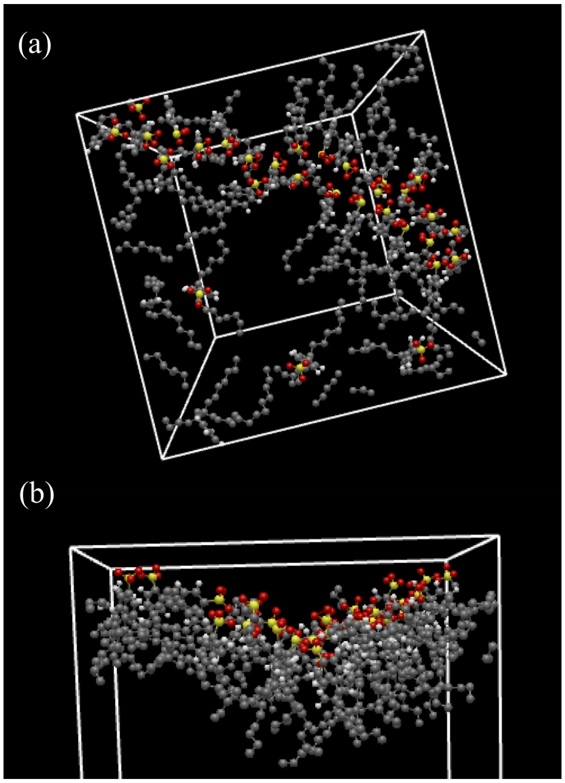


Fig. 4. $8C16^{-}$ molecules at 4032 ns on one of the interfaces. (a) from one of the aqueous phase the cluster of size 27 and three free modules are observed. The huge cluster percolates the system. (b) From one side we note that the huge cluster tends to bend the interface generating instability.[1]

With the previous fact demonstrated, it is of great importance to study these properties in the system. It has been shown that at different densities of particles in a space two percolative states can be obtained [2]. First is the flocculation state, in which the percolating cluster can or cannot percolate the system but particles have to move a great deal of space, relatively speaking, to find other particles and aggregate. On the other hand there is the percolation state or the gelation state, in which particles have do not have to move much to find other particles and from clusters. The difference between the two states can be easily appreciated while comparing the fractal dimension of both states, theoretically there is a characteristic density for every different system in which the fractal dimension goes from being 1.4 for the flocculation state to around 1.89 (for 2D systems) which is called the gelation point. [2] This point is of great importance due to the fact that all densities that are greater than the density at the gelation point will most certainly result in a percolating cluster, and therefore in a great deal of instability in the interface.

E. Periodic Boundary Conditions (PBC)

Periodic Boundary Conditions (PBC) are a common characteristic placed on simulations on a local piece of a much larger system. This condition is placed on the system to attempt to reach the thermodynamic limits of the real system, which means that the original 2D square lattice can "expand" infinitely in all directions. The PBC cause the original 2D square lattice system to act as a continuous torus which implies that when particles cross the western boarder of the system they will appear on the eastern boarder of the system, when particles cross over the northern boarder they will appear on the southern one, and vice versa. This can be better exemplified in figure 5 in which a particle in the cluster is pushed beyond the western boarder and appears on the eastern one.

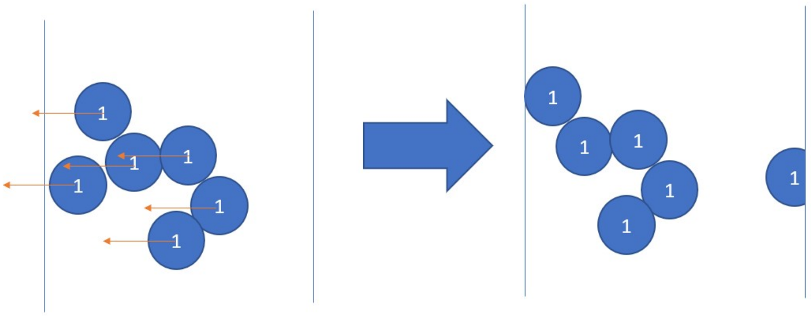


Fig. 5. Drawn example of Periodic Boundary Conditions and how they affect the system.

3. SIMULATION DETAILS

A. Cluster Generation

The simulation was created with the main idea of Reversible DLCA in mind however, other elements such as diffusion coefficients, probability of separation, and cluster stability were varied to attempt to get closer to the results seen in the MD simulations presented by Ricardo Paredes (2018). For these simulation the original surfactants have been approximated to be particles with a diameter of 4 nm, this is then translated into lattice units (LU), which gives us the equivalence of $4\ nm \approx 1\ LU$. From this we follow the standard DLCA protocol established previously, in which we place N particles in the system without any of them touching in a square system of sides L. Following this particles, clusters are then semi-randomly selected to move depending on their mass (to simulate the diffusion coefficient). Once clusters are within range to aggregate they are first removed of any overlap, and then established as one connected cluster. Once clusters exist there is a random probability that one of the particles with coordination number equal to 2 will be chosen to separate from the cluster it belongs to. Finally this process of aggregation and separation continues until a certain number of steps is hit, the cluster oscillates over a certain number of times over having one cluster, or if the probability of separation is 0, then when all particles belong to only one cluster the simulation stops. To quantify the density at which the simulation was run we can utilize the size of each particle, the total amount of particles and the size of the system, a relative density formula can be established where the relative density is equal to the percentage of area covered by the particles in the system. This results in formula 1.

$$\Phi_{off-lattice} = \frac{N \times A}{L^2} = \frac{N_{off-lattice}(\frac{\pi d^2}{4})}{L^2} \tag{1}$$

Where N is the number of particles in the system A is the area occupied by each sphere, and L is the length of the total system. Utilizing these parameters multiple simulations were conducted for systems of different sizes, particle counts and separation probabilities; these can be seen in further detail in the table 1.

Table 1. Main Simulation Sizes

Lattice Size	Particles	Probabilities
128	524	0.01, 0.015, 0.02, 0.03, 0.04, 0.05, 0.1
256	2098	0.01, 0.015, 0.02, 0.03, 0.04, 0.05, 0.1
512	8384 10486 12582 14680 15728	0.01, 0.015, 0.02, 0.03

B. Simulation of Diffusion Coefficient

Diffusion of clusters is known to be inversely proportional to the diameter of the cluster that is moving throughout the liquid $D_{ci} \propto d_{ci}^{-1}$. This means that larger clusters will move at smaller speeds inside of the interface. Correlating the diameter of the cluster to the mass of each cluster, a probabilistic method can be proposed to select a cluster depending on their mass and move them respectively. Therefore the proposed probability distribution to fit this model is as follows:

$$\sum_{i=1}^{n} \frac{A}{m_{ci}} = 1 \tag{2}$$

Where n is the number of clusters m_{ci} is the mass of a given cluster, and$A$ is a dynamic normalization constant, which is calculated every time two clusters aggregate by isolating A from 2in the following way:

$$A = \frac{1}{\sum_{i=1}^{n} \frac{1}{m_{ci}}} \tag{3}$$

Where n and m_{ci} are the same as previously stated. Knowing A one can now calculate the probability of a cluster moving utilizing the following equation:

$$P(m_{ci}) = (Num.\ Clusters\ Mass m_{ci}) \times \frac{A}{m_{ci}} \tag{4}$$

This means that all clusters of hte same mass will have the same probability of moving in a given time.

C. Checking for Aggregation (Cell Algorithm)

Now that particles can move freely around the system a method of revising the cluster that moved for new adjacent or connecting particles is required so that clusters can aggregate. If one were to brute force this task and simply ask all particles in the system, the time complexity of this would be of $O(n^2)$, this is one of the worst time complexities and will increase the time that the program requires to run exponentially as it becomes larger. For this an algorithm, very commonly utilized in MD, called the Cell Algorithm is applied here to improve this time complexity. In this algorithm all spaces in the square lattice are separated into $(L/d)^2$ smaller cells, where L is the lattice size and d is the diameter of the surfactants. This results in $(L/d)^2$ squares of size L/d. Knowing this squares are enumerated horizontally so that every adjacent square to the left or right so that these are a difference of 1 number away, while squares above and bellow are a distance of L numbers away. These values are named k and follow the following formula:

$$k = (y//d) \times L + (x//d) \tag{5}$$

Where x and y are the position of a given particle on the 2D lattice, // simbolizes integer division (*i.e.* $5/2 = 2$) and L is the lattice size of the system. Applying this to the entire systems results in an enumeration which can be exemplified in table 2:

Table 2. Example of the end result of enumerating a 5x5 lattice

20	**21**	**22**	**23**	**24**
15	**16**	**17**	**18**	**19**
10	**11**	**12**	**13**	**14**
5	**6**	**7**	**8**	**9**
0	**1**	**2**	**3**	**4**

D. Removing Overlapping

Once particles overlap, since they all steps performed by the clusters are of the same size, there is a great deal of overlapping in between particles. This is not a desired feature, since overlapping particles will cause the relative density described in equation 1 to decrease as clusters aggregate. To remove this overlap the following mathematical procedure seen in equation 6 is done in reference to the vectors and properties from figure 6.

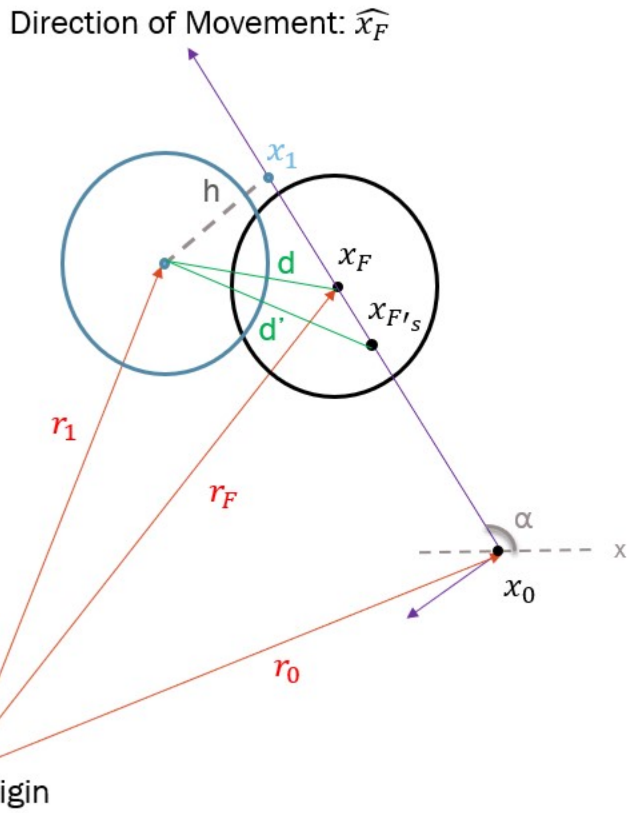


Fig. 6. Figure of utilized vectors for removal of overlap, where x_0 to x_F is equal to 1 LU (or one step).

Utilizing the vectors found in figure 6 the following quantities are established: x_1 is the projection of particle 1 on the direction of movement, r_1 is the position of particle 1 relative to the origin. x_0 is the projection of particle 0 on the direction of movement (treated as origin), r_0 is the initial position of particle 0. r_F is the position of particle 0 after a step and before undoing overlap, x_F is the projection of the position of particle 0 in the direction of movement after performing a step. $x_{F'_s}$ projection of the position of particle 0 in the direction of movement, and is the final position of particle 0 after undoing overlap. And d is the distance from x_F to x_1 and d' is the distance from $x_{F'_s}$ to x_1. Finally h is perpendicular to the direction of movement and measures the distance from x_1 to the position of particle 1. Knowing this, first we define 2 square triangles utilizing d′ and d as the hypotenuse and the distance in the direction of movement from the point F′ or F to the projection of the particle 1 on the direction of movement, named x1.

$$\begin{gathered}
d^2 = (x_1 - x_F)^2 + h^2 \text{ Triangle with len } x_F \\
d'^2 = (x_1 - x_{F'_s})^2 + h^2 \text{ Triangle with len } x_{F'_s} \\
\text{Isolating the "lengths" of the triangles} \\
(x_1 - x_F)^2 = d^2 - h^2 \\
(x_1 - x_{F'_s})^2 = d'^2 - h^2 \\
\text{and subtracting the first from the second triangle we get:} \\
(x_1 - x_{F'_s})^2 - (x_1 - x_F)^2 = d'^2 - d^2 \\
x_1 - x_{F'_s} = \sqrt{d'^2 - d^2 + (x_1 - x_F)^2} \\
\therefore x_{F'_s} = \sqrt{d'^2 - d^2 + (x_1 - x_F)^2} - x_1
\end{gathered} \tag{6}$$

By definition it is known that the projection of a vector on another is the dot product of both of these, therefore:

$$x_1 = (\vec{r_1} - \vec{r_0}) \cdot (\cos(\alpha) i + \sin(\alpha) j) \tag{7}$$

Substituting the value of x_1 from equation 7 into 6 we can arrive at the following expression in equation 8 for the distance from x_0 (*i.e.* the origin of the direction of movement) to $x_{F'_s}$.

$$\boxed{x_{F'_s} = \sqrt{d'^2 - d^2 + ((\vec{r_1} - \vec{r_0}) \cdot (\cos(\alpha) i + \sin(\alpha) j) - x_F)^2} - [(\vec{r_1} - \vec{r_0}) \cdot (\cos(\alpha) i + \sin(\alpha) j)]} \tag{8}$$

When applying this formula and taking the step size minus the $x_{F'_s}$ we can obtain the distance that the entire cluster has to be pushed back in order to have no overlap at any point.

E. Dynamic Calculation of Center of Mass

Previous algorithms utilized a formula which took in a composite list of all positions of both clusters to then calculate the center of mass every time two clusters joined together [7]. This procedure is extremely inneficient and results in a great loss of time during computation since a concatenation of both lists has to be created and iterated over. To reduce the computational time of this a new method that only requires the previous center of masses as well as the previous masses of both connecting clusters is required to calculate the final center of mass, taking into consideration the PBC conditions.

Let us assume that clusters indexed 1 and 2 have joined together, where cluster 2 is aggregating to cluster 1 and overlap has been removed, as can be seen in figure 7:

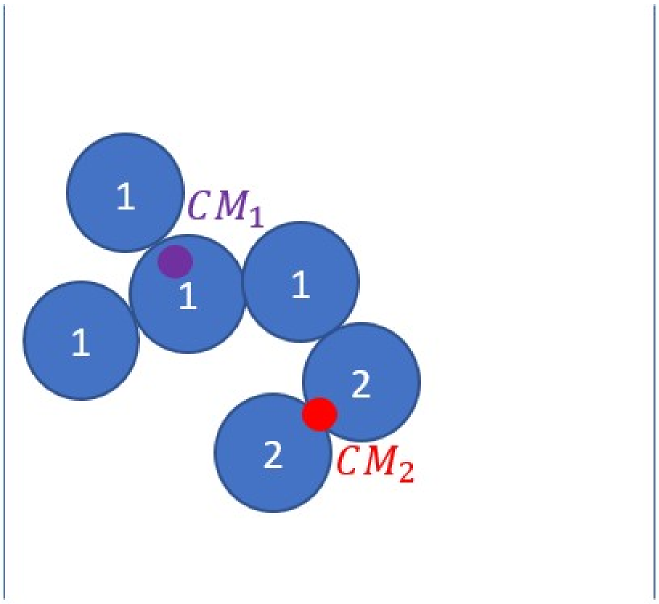


Fig. 7. Image of two test clusters aggregating.

With this figure we can calculate the distance from once center of mass to another and utilize properties of center of mass to calculate the final result. First, to take into consideration PBC, the Cartesian distance has to be divided into its x and y components dx and dy respectively. If either dx or dy is greater or less than than half of the lattice size of the system (*i.e.* $dx < -lattice\ size/2$ or $dx > lattice\ size/2$) then the Cartesian distance is taking the longest available path to reach the other and the shortest distance is crossing the PBC, and therefore the distance will be $dx = dx \pm lattice\ size$. As an example take two particles on opposing sides of a 10x10 lattice, one at x = 1 and the other at x = 9. When measuring dx we obtain $dx = 8$, however the shortest distance in the x axis would actually be to cross the periodic boundary conditions, causing the actual distance between both particles to be $dx = 2$, or the complement of the Cartesian dx. Knowing this we can calculate the center of mass utilizing the following formula:

$$c\vec{m}_{new} = c\vec{m}_1 + \frac{m_2}{m_T} d\vec{x} \tag{9}$$

Where cm_1 is the center of mass of particle 1, m_2 is the mass of particle 2, m_T is the mass of cluster 1 plus cluster 2, and $d\vec{x}$ is the vector between cluster 1 and cluster 2. This formula is extremely useful since it takes into consideration the PBC conditions of the system.

F. Dynamic Calculation of Radius of Gyration

Calculating the radius of gyration proves to be a similar task to calculating the center of mass. We will take the existing radii of gyration of the two aggregating clusters and create a type of weighted average to decide the new radius of gyration. Utilizing the values in image 8:

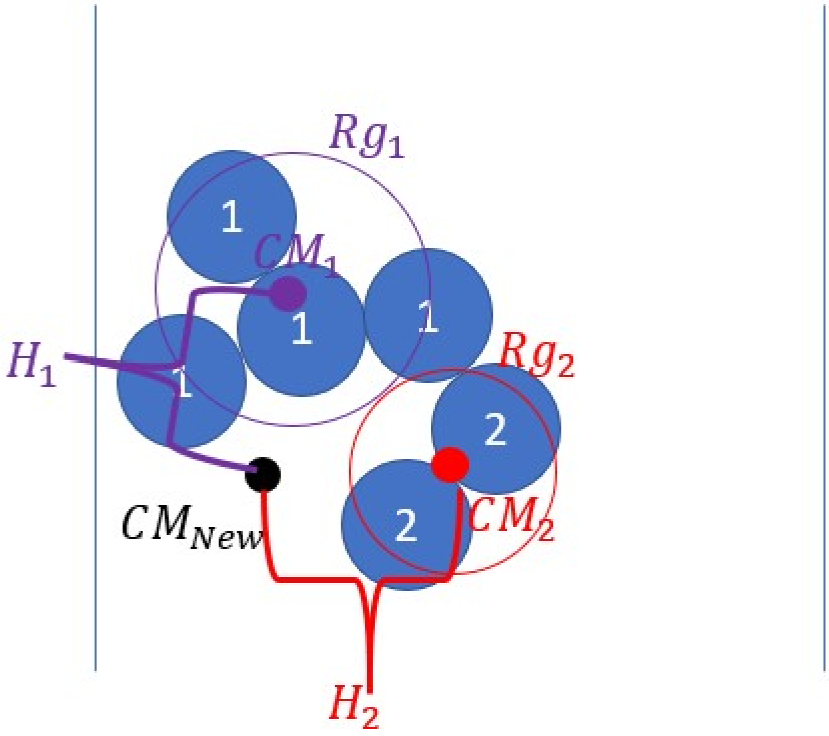


Fig. 8. Values utilized to calculate Radius of gyration dynamically.

Knowing that Rg_1 and CM_1 are the radius of gyration and center of mass of cluster 1, Rg_2 and CM_2 are the radius of gyration and center of mass of cluster 2, CM_{New} is the center of mass of clusters 1 and 2 aggregated, H_1 is the distance between CM_1 and CM_{New}, and H_2 is the distance between CM_2 and CM_{New}. Utilizing the fact that the radius of gyration can be calculated as:

$$Rg^2 = \frac{\sum_i^N m_i(\vec{r'_i} + \vec{cm})^2}{\sum_i^N m_i}$$
$$Rg^2 = \frac{\sum_i^N m_i \vec{r'_i}^2 + \sum_i^N m_i \vec{cm}^2 + \sum_i^N m_i \vec{r'_i} \cdot \vec{cm}}{\sum_i^N m_i} \tag{10}$$

Where $\vec{r'_i}$ is a vector from the center of mass to the center of another particle, and can be expressed as $\vec{r'_i} = \vec{r} - \vec{cm}$. Knowing this the first term of the equation 10 is relate to the ith radius of gyration squared the system, the second term is the distance from the ith center of mass to another point (in this case the new center of mass) which will be relabeled H_i, and the third term is the position of the center ith center of mass with respect to itself, which is always 0. This gives the following expresion:

$$Rg^2 = \frac{\sum_i^N m_i Rg_i^2 + \sum_i^N m_i \vec{H_i}^2 + \cancel{\sum_i^N m_i \vec{r'_i} \cdot \vec{cm}}}{\sum_i^N m_i}$$
$$Rg^2 = \frac{\sum_i^N m_i (Rg_i^2 + \vec{H_i}^2)}{\sum_i^N m_i} \tag{11}$$

Simplifying this to only two clusters we obtain the formula:

$$\boxed{Rg^2 = \frac{m_1(Rg_1^2 + \vec{H_1}^2) + m_2(Rg_2^2 + \vec{H_2}^2)}{m_1 + m_2}} \tag{12}$$

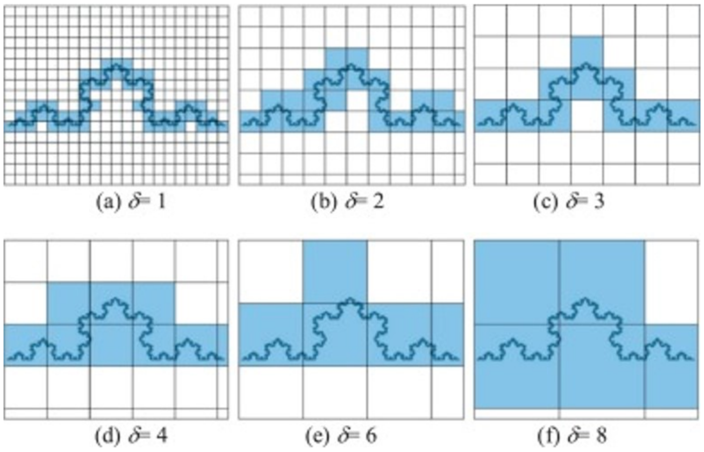
G. Reversibility

To add the reversible aspect we reached a simplification in which only clusters of coordination number equal 1 one can separate from larger clusters, this is justified with the fact that surfactants with 2 or more bonds to other surfactants are more stable than the ones on the edges. Utilizing this we can create a system in which all particles of z=1 are in a dynamic list, from this list one of the particles is chosen and given an initial probability of separation (between 0 and 1) these particles will either separate in the given step or the algorithm will choose another cluster to move and continue the process normally. If a particle is chosen to be separated from the cluster, the appropriate measures and calculations will take place to remove it from the linked list of its cluster, recalculate the center of mass and the radius of gyration for both clusters, and then push the particle in the opposite direction of its most adjacent particle. This step in the opposite direction is done to ensure that the particle fully separates from the cluster and removes the possibility of the particle becoming completely overlapped with its neighbour.

H. Calculations after Completed Simulation

H.1. Fractal Dimension (d_f)

To calculate the fractal dimension one of many methods can be utilized, including the box counting method, the average particles per box method, counting particles for different radii, utilizing the radius of gyration over time, and even utilizing the correlation function. However, for these simulations the box counting method and the average particles per box method were utilized. They were chosen due to the fact that they are not complicated in their development and require no extra information apart from the final positions of particles, as well as the cluster indexes for these particles (in the case that one final cluster was not reached). The box counting method consists on dividing the lattice into equally sized squares, from this the amount of boxes that contain the fractal object are counted and averaged over several simulations, after counting the size of the boxes is reduced and the process starts over again, this procedure is shown in theory for the Koch curve in figure 9. This results in an exponential relationship between the size of the boxes and the number of boxes that contain the fractal object, *i.e.* $size\ boxes \propto Exp(-a \times num.\ boxes)$. For which the coefficient a represents the fractal dimension d_f of the system.


(a) $\delta = 1$ (b) $\delta = 2$ (c) $\delta = 3$

(d) $\delta = 4$ (e) $\delta = 6$ (f) $\delta = 8$

Fig. 9. Box counting method example for a Koch curve.

The second method utilized in this project was the average particles per box method, this follows the same logic as the previous method however, instead of counting the number of boxes the studied value is the amount of particles present per box for the different sizes. This once again results in an exponential relationship between the size of the boxes and the average number of particles per boxes, $size\ boxes \propto Exp(a \times avg.\ particles\ in\ boxes)$, where once again the coefficient a represents the d_f. This can be applied, similarly to [7], however in this case, since not all simulations end in one cluster, only the largest cluster per simulation that has a relative size of 40% of the particles in the system will be considered for the fractal

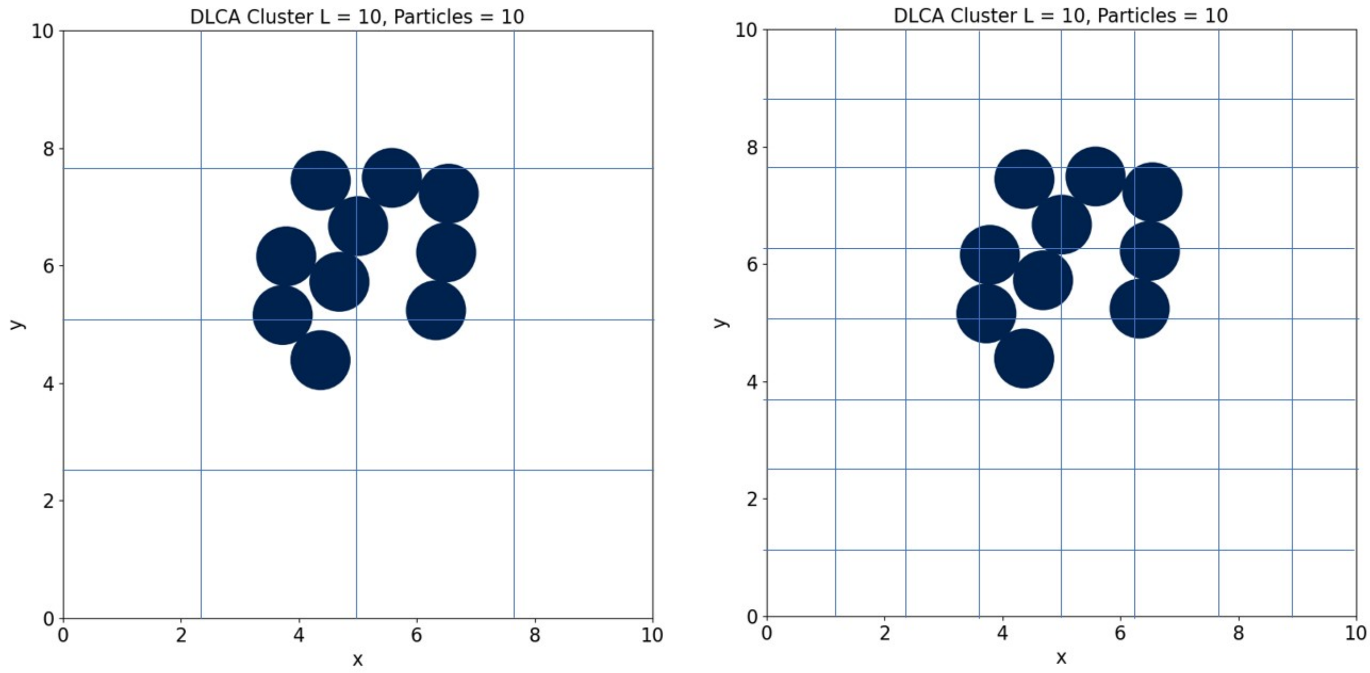


Fig. 10. Progression of smaller boxes in box counting methods to find fractal dimension.

dimension calculations. This is to avoid calculating fractal dimension on a object that is too small, since this would cause size effects to appear and skew the data towards an unrealistic value.

This can be applied to our simulations in a very similar way as the one described in figure 9. This can be observed in more detail in figure 10 After counting, applying logarithm to the results and adjusting a linear model to the system we can arrive at a value of a fractal dimension, as specified previously.

H.2. Cluster Percolation

To calculate cluster percolation in the system, one must view the largest remaining cluster in the system after the cluster arrives at its equilibrium point. Once this occurs the lattice grid is split into n^2 blocks of size LU^2, which can be seen on figure 11. Once this has occurred a revision is done column by column and row by row to see if at least one particle is present either in each column or each row of the final largest cluster. If one of these two conditions is met then it is said that the cluster percolates either in the x direction or the y direction (this percolation direction does not affect the physical properties of the problem). If both conditions are disproved, *i.e.* there was at least one empty row and one empty column in the system then the cluster does not percolate the system.

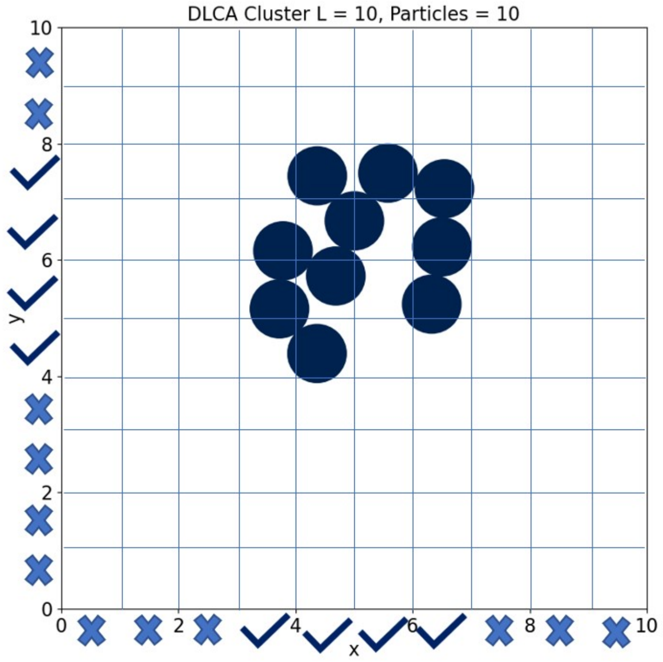


Fig. 11. Example of percolation algorithm.

4. IRREVERSIBLE DLCA RESULTS AND DISCUSSION

A. Simulation Results

As discussed in [7] the DLCA irreversible system cannot be utilized to analyze the aggregation of surfactants, including 8C16, due to the fact that it is unrealistic to assume that surfactants aggregate and have a potential well larger than $k_b T$, however there is value to comparing previous DLCA simulations to the new version passed to C. For this section we will briefly analyze run-times, fractal dimension results, coordination numbers, and percolation results, to ensure that both have the same base results.

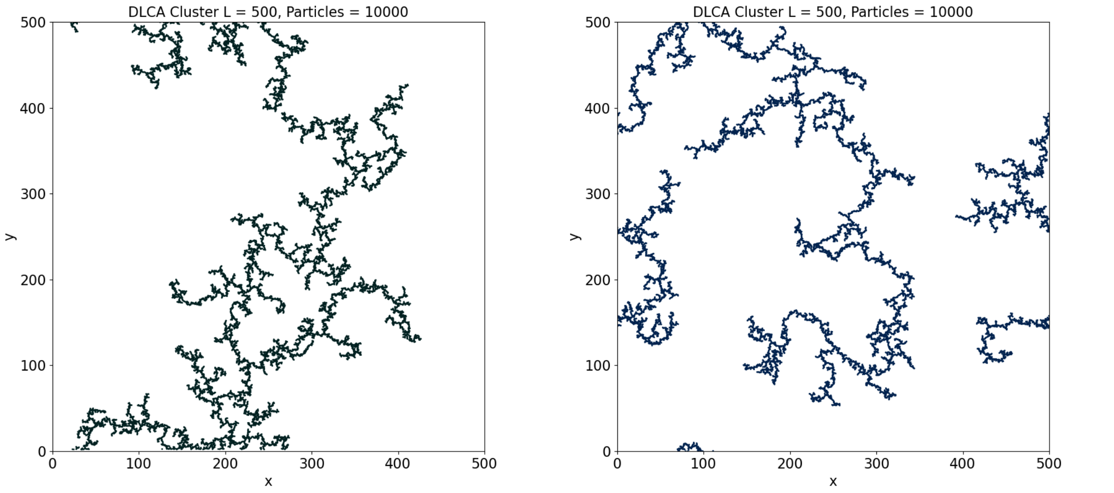
A.1. Run-Times

Table 3. Run-Time Difference Between Irreversible DLCA On and Off Lattice Simulations in Python and Off-Lattice Simulations in C

Lattice Size	Density	Average Simulation Time Off-Lattice Python [s]	Average Simulation Time On-Lattice Python [s]	Average Simulation Time C [s]
500	0.003141593	5350.97	1812.46	24.19
500	0.006283185	6250.96	2510.41	26.81
500	0.012566371	6623.29	2962.05	27.08
500	0.025132741	5384.67	2727.83	27.21
500	0.031415927	5053.36	2616.78	33.51
500	0.037699112	4218.3	2470.65	30.83
500	0.043982297	3757.07	2302.85	25.27
500	0.04712389	4028.9	2124.44	21.67

From table 3 it is clear why it was necessary to change the simulation code to a much faster language like C. This drastic drop in computation time will allow reversible systems to be something attainable, since it is known that these systems increment in multiple orders of magnitude the time required to finish a simulation.

A.2. Final Systems and Fractal Dimensions

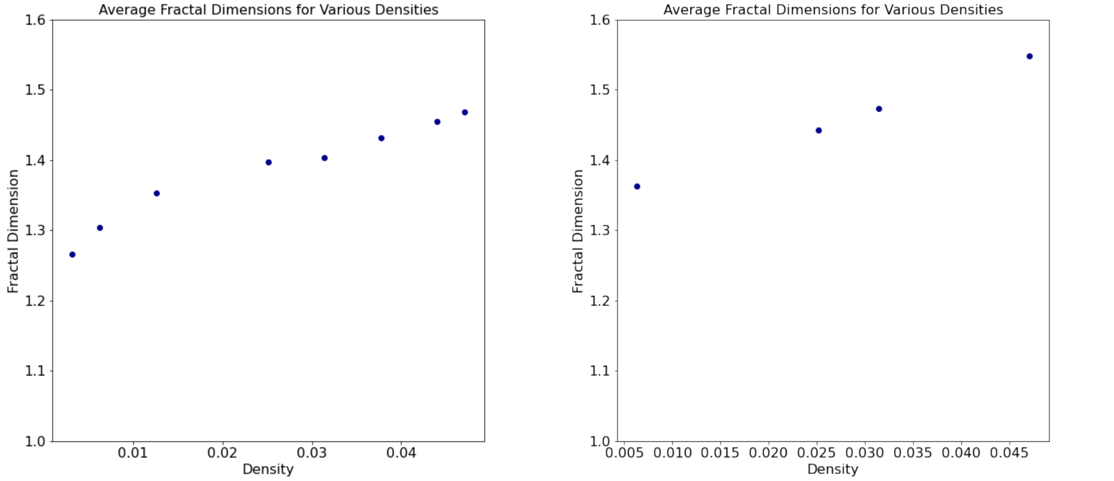


(a) Resulting L = 500 with 10,000 particles for Python cluster.

(b) Resulting L = 500 with 10,000 particles for C cluster.

Fig. 12. Resulting Clusters for both simulation types.

These clusters seem to have a very similar structure, however to ensure this one can compare the fractal dimension and their evolution across multiple densities.



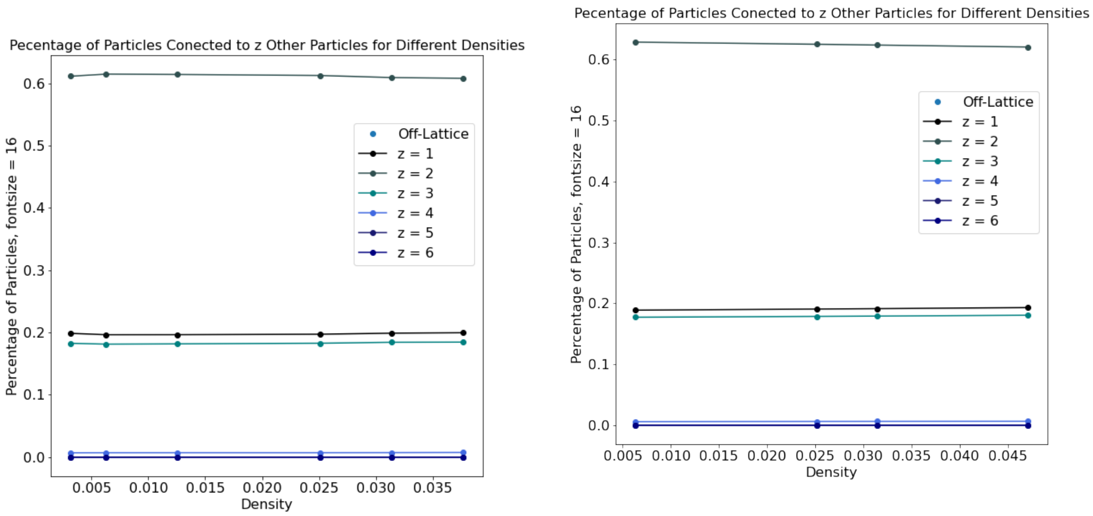
(a) Average fractal dimension for simulations in [7], for systems of lattice size 500.

(b) Average fractal dimension for simulations utilizing C, for systems of lattice size 1024.

Fig. 13. Resulting fractal dimensions for both simulation types.

Utilizing figures 13a and 13b it is very clear that both fractal dimensions increment int he same way. The only key difference between these two figures is that, since figure 13b is simulated at a larger size, the size effects experienced by the original system have been mitigated and therefore a more constant value of fractal dimension is observed, before starting to enter the percolative regime.

A.3. Coordination Numbers



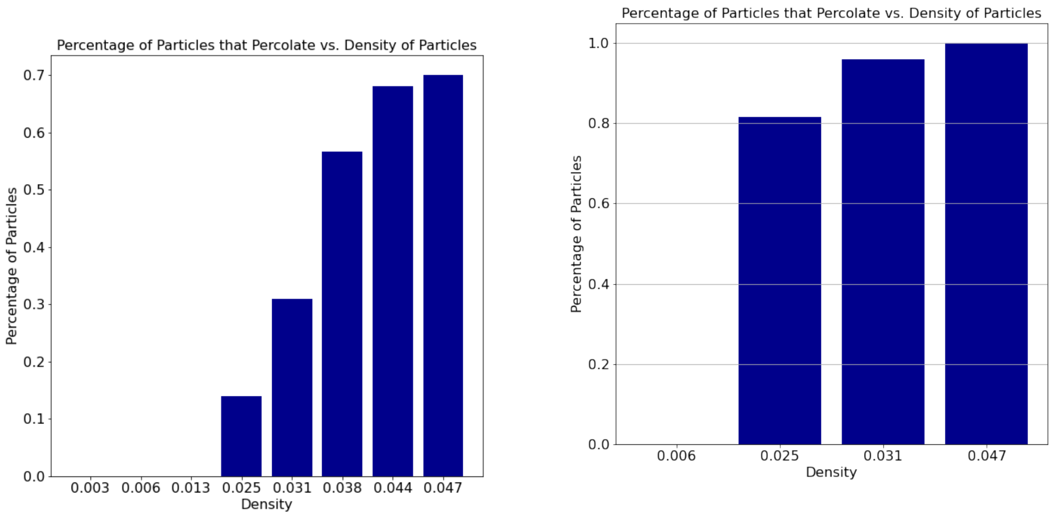
(a) Normalized coordination numbers for simulations in [7], for systems of lattice size 500.

(b) Normalized coordination numbers for simulations utilizing C, for systems of lattice size 1024.

Fig. 14. Resulting average coordination numbers for both simulation types.

Figures 14a and 14b once again cement the fact that both systems are identical in construction, since not only do their fractal dimensions align but their coordination numbers are almost identical one to the other.

A.4. Percolation Results



(a) Percentage of percolated clusters for simulations in [7], for systems of lattice size 500.

(b) Percentage of percolated clusters for simulations utilizing C, for systems of lattice size 1024.

Fig. 15. Percolation percentage for multiple systems for both simulation types.

Finally observing the percolation results one can clearly observe that the graphs seen in figures 15a and 15b are not the same. However, the percolative results from 15b better align themselves with the theoretical cross from the floculattion regime to the percolative regime, since this point is theorized to be discrete [2].

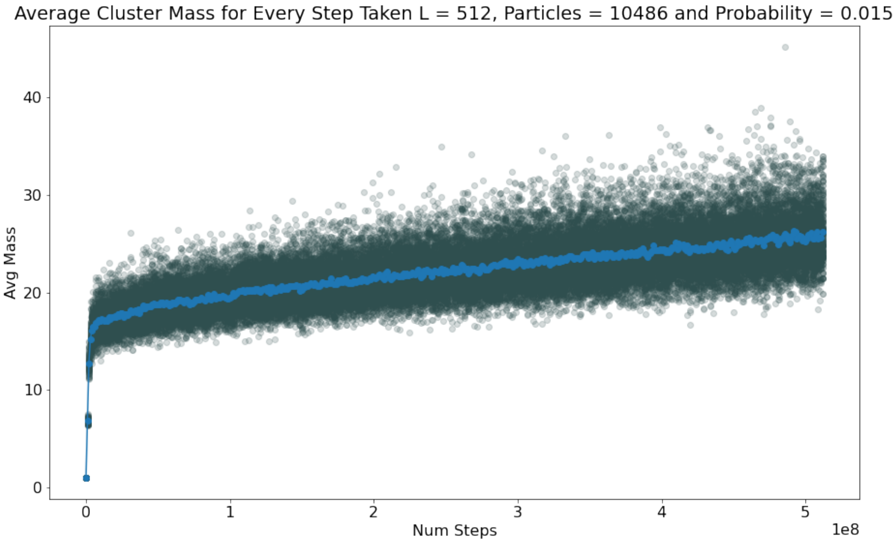
All of the previous information demonstrates that the C simulations completely outperform its Python counterpart, both in speed and accuracy of results. This increase in accuracy comes from the fact that there is less time required to run each simulation and therefore one can have a larger amount of results, and one can even increment the size of the system to avoid size errors which can occur when the systems are too small (even though systems with lattice size 500 are not small at all).

5. REVERSIBLE RESULTS AND DISCUSSION

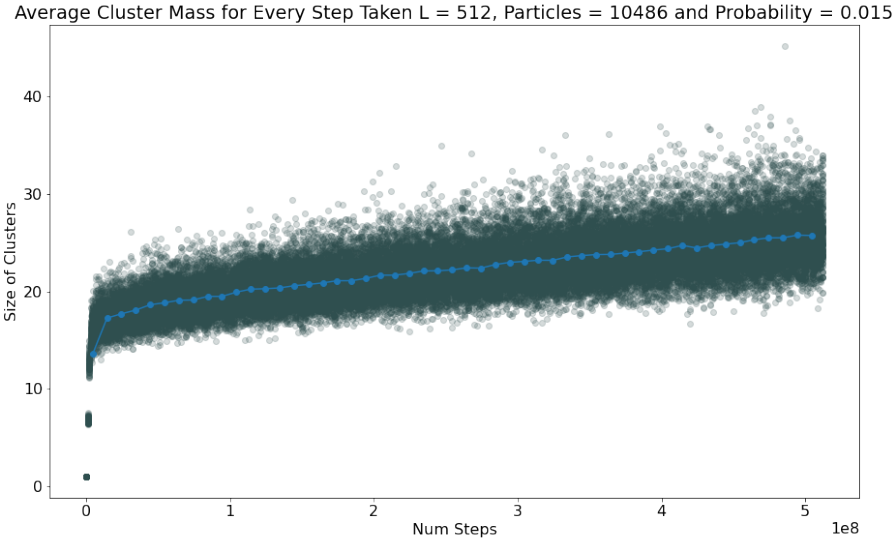
For this section the simulated systems where the ones represented in table 1, however the more accurate results are obtained from larger systems, and therefore reported results for this section will mostly include those of lattice size 512 and the combination of particles and probabilities. Observing the 1 table there is a difference in the probabilities that were studied in the 128 and 256 lattice size systems vs. the 512. This reduction in probabilities was due to the fact that the probabilities of 0.05 and 0.1 did not give interesting results since clusters where never able to form and sustain themselves for too long, which gives results of dilution without aggregation, which is not the objective for this article.

A. Stabilization Times

Stabilization times were studied in two different ways, this was due to the fact that some graphs were inconsistent due to the number of steps the system took to arrive at a stable position. The two types of graphs were the Average cluster mass graph, as well as the largest cluster graph. These graphs were a compilation of all step numbers for the simulations. Once the dispersion of all of the data is achieved, a mean value for every step number is calculated to arrive at a more concise line plot. However since there are thousands of revisions placed on the system, a moving average graph is proposed to soften the brute graphs and arrive at a more sensible line plot. This procedure can is shown in figures 16a, and figure 16b for the average mass clusters, as well as figures 17b and 17b for the largest cluster per step.

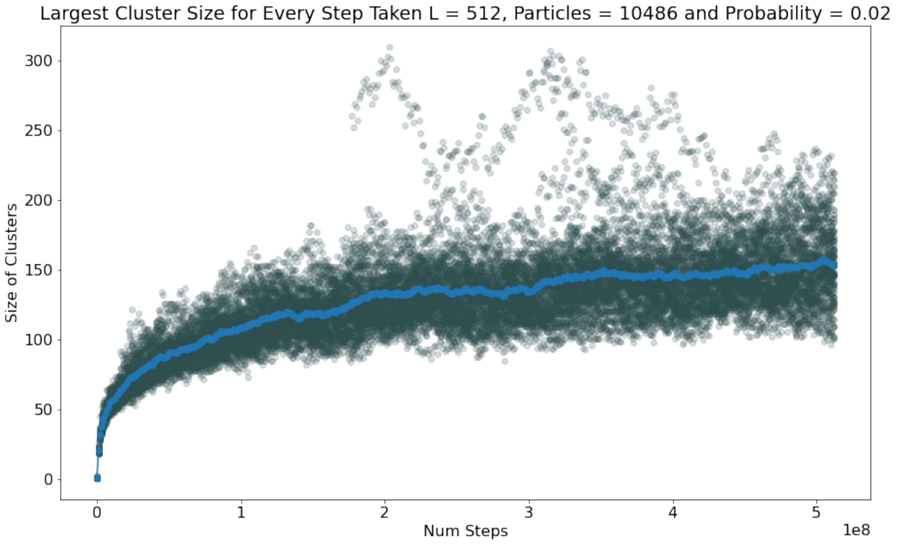


(a) Average masses per step brute graph with dispersion and average for every step.

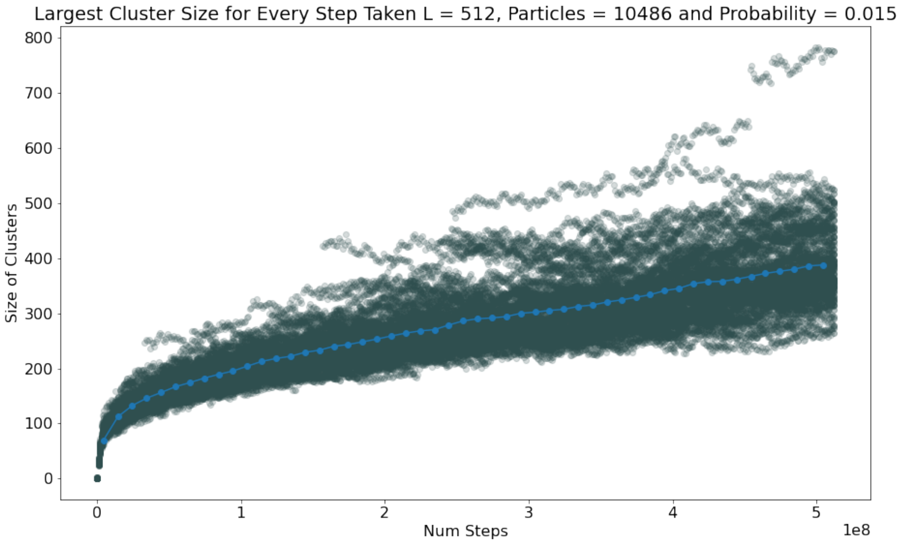


(b) Average masses per step softened graph with dispersion and average for every step.

Fig. 16. Example plots for brute and softened plots of average mass size for every step taken.



(a) Largest Cluster per step brute graph with dispersion and average for every step.

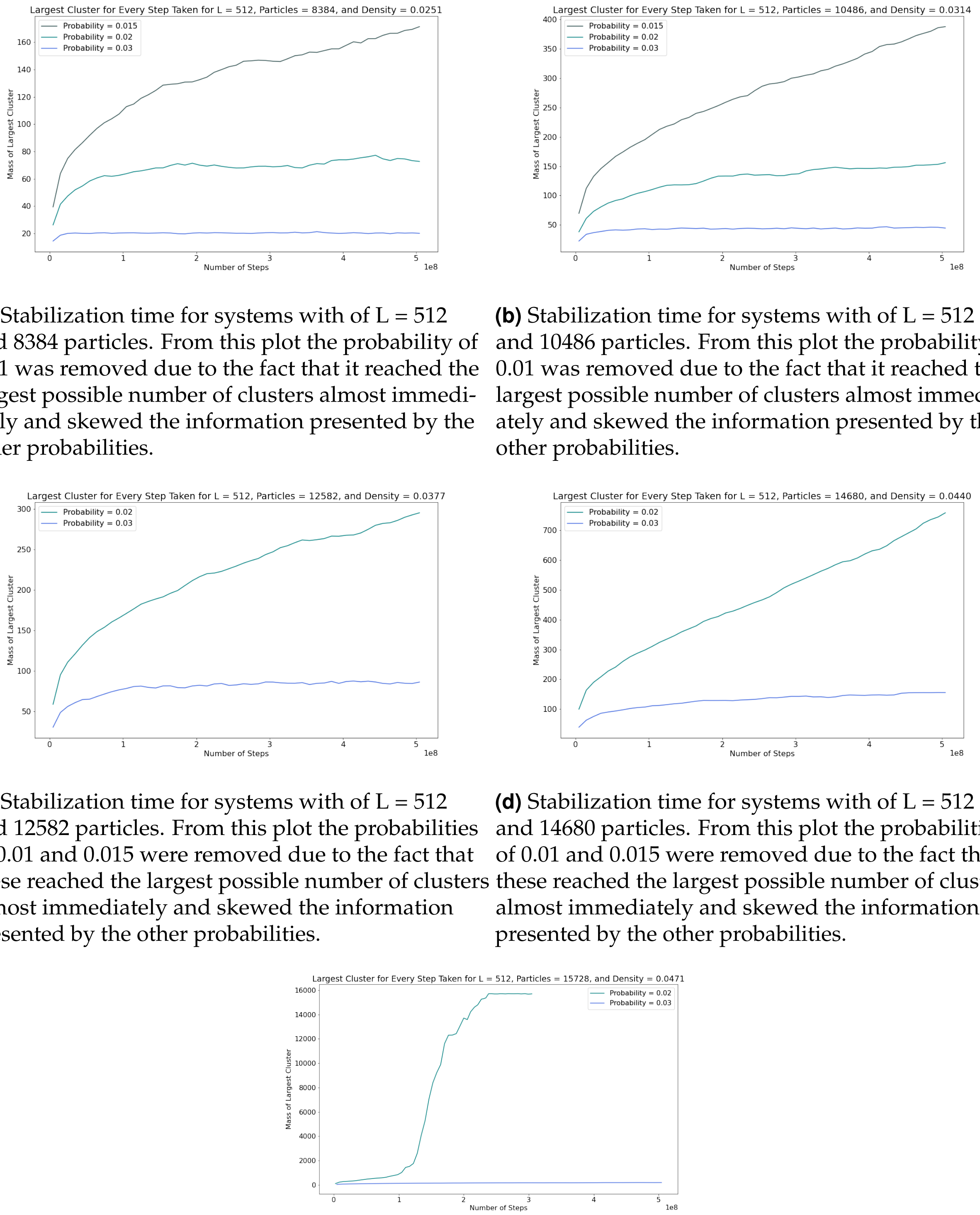


(b) Largest Cluster per step softened graph with dispersion and average for every step.

Fig. 17. Example plots for brute and softened plots of largest cluster size for every step taken.

To better observe the effects of probability and density on the stabilization time we compiled the values obtained for lattice size 512 and all combinations of particles and probabilities. From this we can then observe how the changes in probability affect the equilibration of they system by leaving the density constant, as seen in fig 18. And we can vary the density and keep the probabilities the same to observe how different densities affect stabilization time, to obtain the graphs from figure 19.

Observing figures in figure 18 it is clear that the probability has a sever impact on the stabilization on the system. Analyzing figure 18a we can observe how even though the 0.02 and 0.03 probability clusters show very little variation after 100 million steps, the 0.015 cluster continues to climb towards the stable conditions where it oscillates between a certain value of maximum mass. This is similar to what happens in figure 18b where probabilities of 0.02 and 0.03 have stabilized but 0.015 continues to climb at a steep pace, which means it



(a) Stabilization time for systems with of L = 512 and 8384 particles. From this plot the probability of 0.01 was removed due to the fact that it reached the largest possible number of clusters almost immediately and skewed the information presented by the other probabilities.

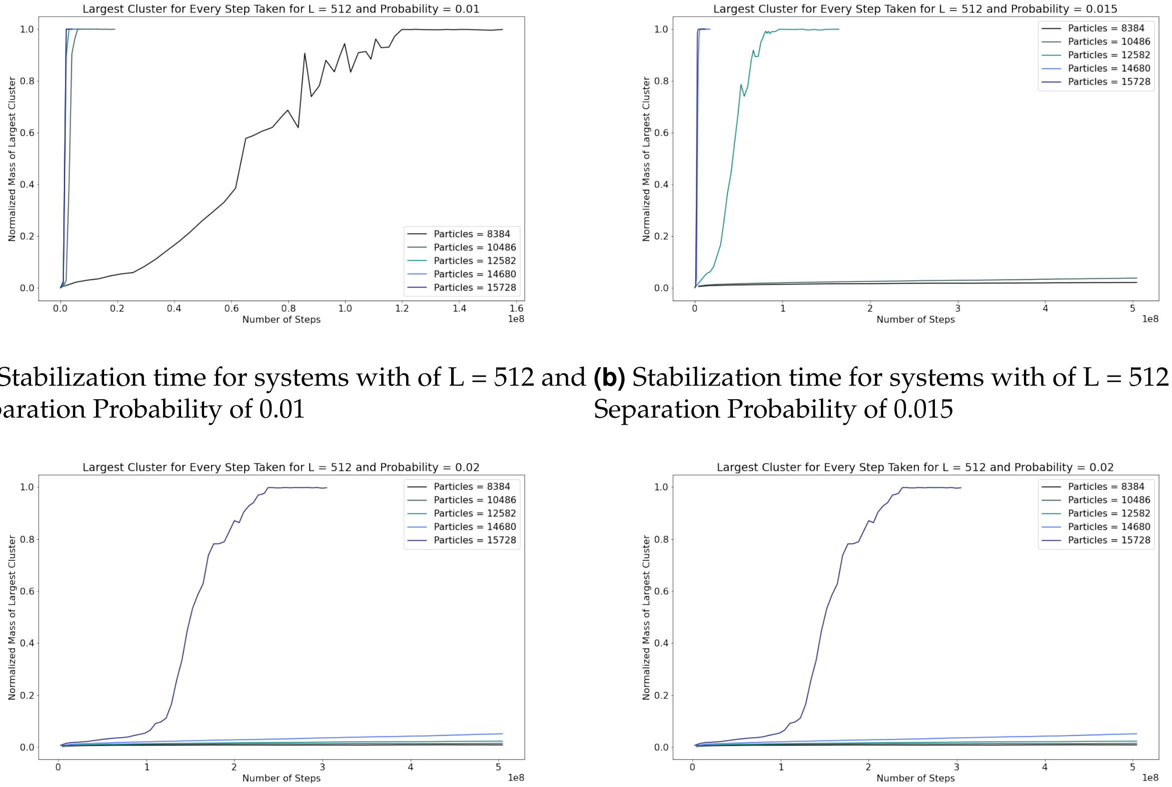
(b) Stabilization time for systems with of L = 512 and 10486 particles. From this plot the probability of 0.01 was removed due to the fact that it reached the largest possible number of clusters almost immediately and skewed the information presented by the other probabilities.

(c) Stabilization time for systems with of L = 512 and 12582 particles. From this plot the probabilities of 0.01 and 0.015 were removed due to the fact that these reached the largest possible number of clusters almost immediately and skewed the information presented by the other probabilities.

(d) Stabilization time for systems with of L = 512 and 14680 particles. From this plot the probabilities of 0.01 and 0.015 were removed due to the fact that these reached the largest possible number of clusters almost immediately and skewed the information presented by the other probabilities.

(e) Stabilization time for systems with of L = 512 and 15728 particles. From this plot the probabilities of 0.01 and 0.015 were removed due to the fact that these reached the largest possible number of clusters almost immediately and skewed the information presented by the other probabilities.

Fig. 18. Stabilization times while varying the probability within one single density. From this plot the skewing of data is clear, since the simulations with probability of 0.02 show that these clusters stabilize at around 250 million steps but there is still variability in the 0.03 cluster.



(a) Stabilization time for systems with of L = 512 and Separation Probability of 0.01

(b) Stabilization time for systems with of L = 512 and Separation Probability of 0.015

(c) Stabilization time for systems with of L = 512 and Separation Probability of 0.02

(d) Stabilization time for systems with of L = 512 and Separation Probability of 0.03.

Fig. 19. Stabilization times while varying the density within one probability value.

is much further away from finding equilibrium, even after 500 million steps. This trend continues with the larger density systems in which one or two probabilities find equilibrium extremely quickly but there is always one probability configuration that struggles to find equilibrium. This speaks to the difficulty of arriving at the equilibrium point for surfactant systems, since it is unlikely that these systems have similar properties to the ones seen with the 0.01 probability of separation configurations, and are more likely to behave like the systems that do not find their equilibrium point that easily, as seen in the 0.015 systems at low densities (around 0.0314) and 0.02 at higher densities (at least 0.0377).

Moving forward to analyze the effects of densities on the system we can look at the figure in figure 19. Here it is evident the effect of density on the equilibration time of the system. In all figures the largest cluster density always arrives at a period of stability before its less dense counterparts. This is due to the fact that the increase in density has the effect of more particle collisions for every step, this increase in the number of collisions (and thereby the probability of a collision occurring) acts as a counter measure to the probability of separation for the clusters in the system, and therefore allows more dense systems to achieve the same largest mass cluster at much higher densities than the less dense systems. This can be clearly shown between figures 19c and 19a, here the cluster with density 0.0471 at the probability of 0.02 is able to reach the same equilibration position as the system of density 0.0251. This idea of offsetting the probability of separation is most clear in figure 19d in which the cluster with the highest density is able to make clusters of much greater size than the ones with a much lesser number of particles. And even though this results in quicker stabilization times, the drawback of these high densities in real systems is that they would most certainly percolate, causing them to cross the saturation concentration point and cause greater instability to the system.

B. Fractal Dimension

Observing the fractal dimensions for systems that have reached their equilibration point we can obtain the following graphs seen in figure 20 and 21

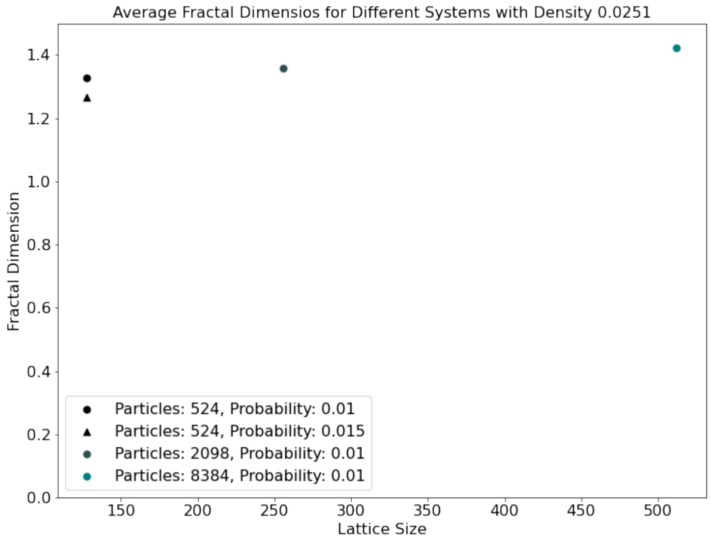


Fig. 20. Fractal dimension by varying the lattice size at density 0.0251.

Figure 20 shows the size errors that occur when systems are too small, since the fractal dimension increases, with no reason, from 1.35 to 1.42. This gives further proof to the reason for utilizing systems with lattice size of at least 512 LU.

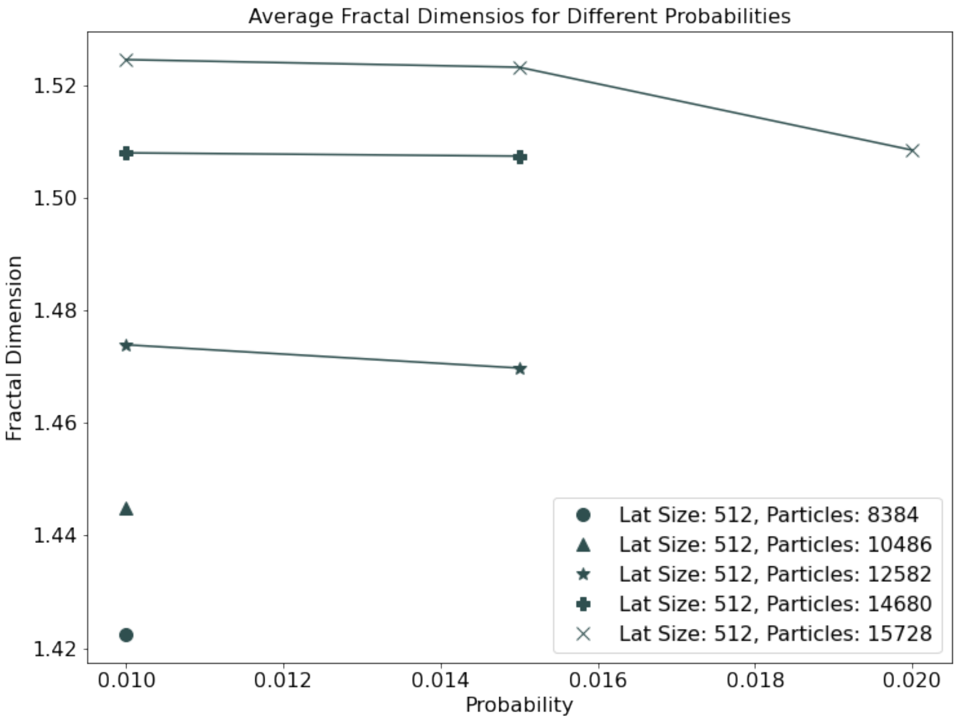


Fig. 21. Fractal dimension by varying the probability.

Observing figure 21 it can clearly be observed that when increasing the probability of separation, the fractal dimension goes slightly down. This follow previous results seen in the literature [14] where the fractal dimension can decrease from the original 1.45 observed by [8, 9] to around 1.39. This exact result was not obtained in these simulations due to the fact that not being able to calculate the fractal dimensions of the systems of low density (due to size errors) at probabilities greater than 0.015 removes the opportunity to view these results in full. However this is slightly apparent with the simulations with 15728 particles since those reduce from just above 1.52 to just above 1.50. This decrease in fractal dimension is due to the fact that the system while separating and re aggregating is becoming less ramified and has an increased number of particles with coordination number 2 instead of 3 or 4.

C. Coordination Numbers

Plotting the coordination numbers for simulation with lattice size 512 we can obtain the figure 22.

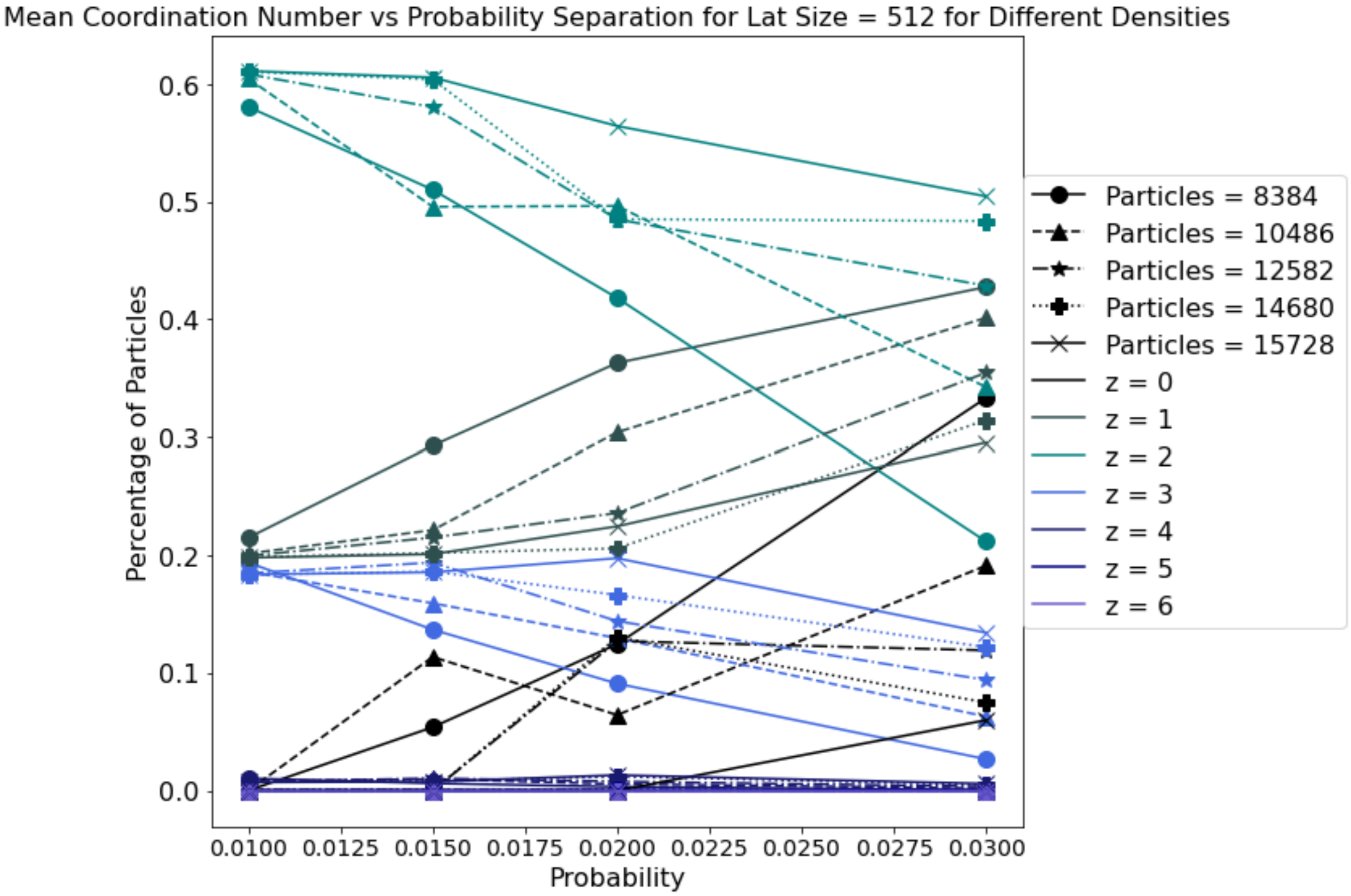


Fig. 22. Coordination values normalized to number of particles, at the position of equilibrium or last step.

Observing fig 22 the evolution of the coordination values of clusters at probability of separation 0.01 all begin at around the same point, which is comparably the same to the results seen in 14b and 14a. However, once this value increments the values go into two discernible groups, the low density systems (particles = 8384 and 10486) reduce in their amount of particles connected to two other particles and drastically increase the amount of individual particles (z=0), this means that the edges of larges clusters have broken apart and now exist as individual particles. On the other hand the high density clusters maintain the similar values of mean coordination numbers, which implies that these clusters have not changed in their formation (following the example set by figure 19b). Additionally to this, in consequential higher probabilities of separation the density of the system becomes unable to continue stabilizing itself and tends towards the same mean coordination numbers as the low density systems, in which most particles are either connected to one or two particles but they are in the minority, while the number of free particles increments greatly.

D. Percolation Results

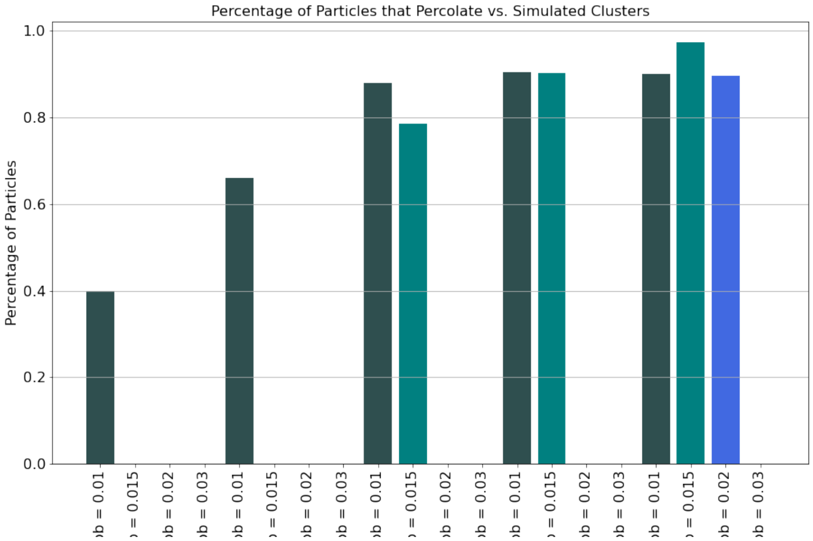
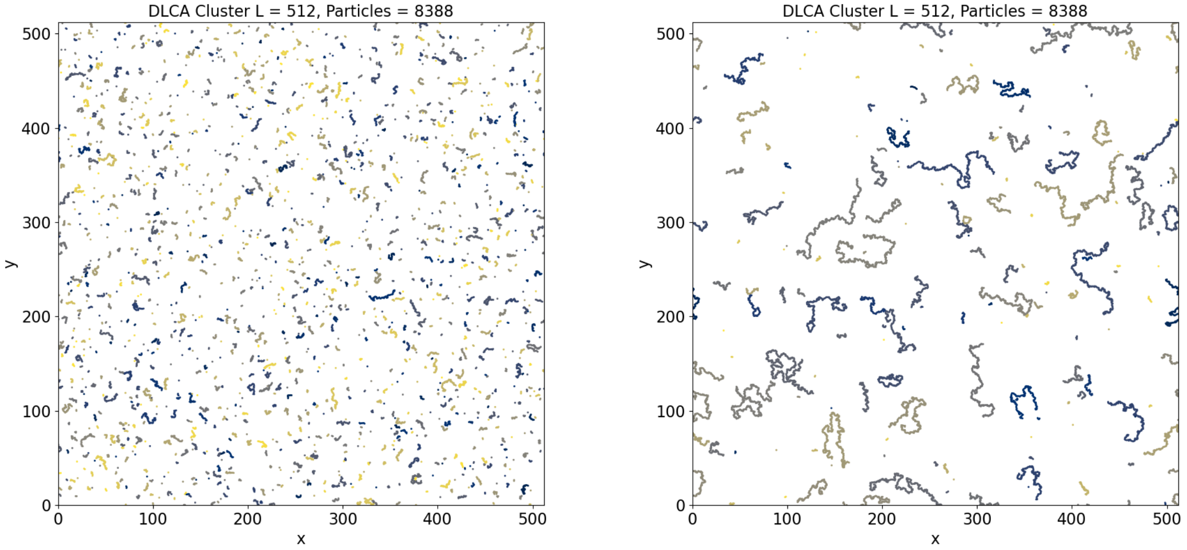


Fig. 23. Percentage of at least one percolated cluster per simulation for different combinations of simulations at L = 512.

For the results in 23 the percolative percentages seen in the simulations with probability of 0.01 are extremely similar to the ones seen previously in 15b and 15a. However, the results for higher probability of separation simulations show a much more abrupt change in the percolation percentage of clusters. This result is extremely important since the saturation concentration is theoretically an almost immediate drop and this abrupt change from no percolation to an incredibly high amount of percolation supports the fact that these simulations are approximating the aggregation of surfactants incredibly well.

Fig. 25. Fractal dimension for edge aggregation system.

E. Edge Aggregation Restriction



(a) Plot of Edge Cluster Simulation with separation probability of 0.01 after 250 million steps.

(b) Plot of Edge Cluster Simulation with separation probability of 0.001 after 250 million steps.

Fig. 24. Sample of Edge Restriction simulations.

Observing fig 25 we can see that the only clusters who where large enough to be studied where the ones with a probability of separation of 0.001. This incredibly small probability is required to see an actual growth in the size of the edge clusters due to the extremely low probability that they will find another particle to aggregate. Even though this structure is similar to the real aggregation of surfactants seen in the literature [3], the incredibly low probability of aggregation and the extremely small clusters that are able to be formed at normal probabilities are cause to dismiss these types of simulations. Since they not only do not provide much information, they are unrealistic in their construction.

6. CONCLUSIONS AND FINAL DISCUSSION

In this paper we developed a computer computer simulation that successfully emulates the aggregation and deaggregation of clusters, along with the corresponding routines to calculate the fractal dimension, coordination number, the probability of percolation, the evolution and eventual stabilization of the system as well as the center of mass and the radius of gyration for all clusters. The results obtained show that these reversible DLCA simulations are a good approximation to the aggregation seen in the 8C16 surfactant system, due to the fact that the long periods of stabilization of the system is clearly observed, the fractal objects approximate the connectivity of the real surfactant system and that the saturation concentration point of the interface can be clearly observed in the percentage of clusters percolated. Furthermore it is clear that, even though the edge clusters approximate the connectivity of the real 8C16 system better, the extremely low probability of separation as well as the drastically reduced fractal dimensions indicate that these types of simulations are not the best suited to attempt to reach the real 8C16 aggregation.

The results from the group of figures 18 and 19 show the extremely high amount of time for the systems to reach equilibrium (and some even don't reach it by that time), aligning with the results seen previously

[3]. The fractal dimension values observed in fig 21 follow the estimated results for the decreasing fractal dimension in reversible DLCA simulations [14]. This correct result is further supported by the evolution of the coordination number seen in figure 22 which show that when the probability of separation increases the number of particles connected to two other particles is greater, relatively, to the amount of particles with 3 other particles connected to them. Lastly the sharp increase in the percentage of percolating clusters observed in 23 are a directly related to the saturation concentration which is seen in the literature [1], which is considered a sharp drop in the interfacial tension once the maximum number of surfactans are present and these are about to percolate the system.

REFERENCES

1. R. Paredes, A. I. Fariñas-Sánchez, B. Medina-Rodrıguez, S. Samaniego, Y. Aray, and L. J. Álvarez, “Dynamics of surfactant clustering at alkane-water interfaces (preprint),” Langmuir (2018).
2. M. Rottereau, J. Gimel, T. Nicolai, and D. Durand, “Monte carlo simulation of particle aggregation and gelation: I. growth, structure and size distribution of the clusters,” The Eur. physical journal. E, Soft matter **15**, 133—140 (2004).
3. R. Paredes, A. I. Fariñas-Sánchez, B. Medina-Rodrıguez, S. Samaniego, Y. Aray, and L. J. Álvarez, “Dynamics of surfactant clustering at interfaces and its influence on the interfacial tension: Atomistic simulation of a sodium hexadecane–benzene sulfonate–tetradecane–water system,” Langmuir **34**, 3146–3157 (2018). PMID: 29411980.
4. M. J. Rosen and J. T. Kunjappu, *Surfactants and Interfacial Phenomena* (John Wiley amp; Sons, 2012).
5. J. M. Campos, T. L. Stamford, R. D. Rufino, J. M. Luna, T. C. M. Stamford, and L. A. Sarubbo, “Formulation of mayonnaise with the addition of a bioemulsifier isolated from candida utilis,” Toxicol. Reports **2**, 1164 – 1170 (2015).
6. J. J. Sheng, *Modern Chemical Enhanced Oil Recovery Theory and Practice* (Gulf Professional Publishing, 2011).
7. D. Rosenberg de Angoitia and R. Paredes, “Simulation of aggregation of surfactants in hydrocarbon-water interfaces utilizing diffusion limited cluster aggregation (dlca) model (part 1),” (2021).
8. P. Meakin, “Fractal aggregates,” Adv. Colloid Interface Sci. **28**, 249 – 331 (1987).
9. R. Jullien, R. Botet, and P. M. Mors, “Computer simulations of cluster–cluster aggregation,” Faraday Discuss. Chem. Soc. **83**, 125–137 (1987).
10. P. Meakin, “The effects of random bond breaking on diffusion limited cluster–cluster aggregation.” (1985).
11. K. N. Christopher S. A. Musgrave, Shuntaro Shoji, “A review of low density porous materials used in laser plasma experiments,” **25** (2018).
12. Y. Nakama, *Surfactants* (2017), pp. 231–244.
13. V. K. Shante and S. Kirkpatrick, “An introduction to percolation theory,” Adv. Phys. **20**, 325–357 (1971).
14. J.-M. Jin, K. Parbhakar, L. H. Dao, and K. H. Lee, “Gel formation by reversible cluster-cluster aggregation,” Phys. Rev. E **54**, 997–1000 (1996).
15. S. Jungblut, J.-O. Joswig, and A. Eychmüller, “Diffusion-limited cluster aggregation: Impact of rotational diffusion,” The J. Phys. Chem. C **123**, 950–954 (2019).
16. J. C. Gimel, D. Durand, and T. Nicolai, “Transition between flocculation and percolation of a diffusion-limited cluster-cluster aggregation process using three-dimensional monte carlo simulation,” Phys. Rev. B **51**, 11348–11357 (1995).
17. P. Meakin, “Formation of fractal clusters and networks by irreversible diffusion-limited aggregation,” Phys. Rev. Lett. **51**, 1119–1122 (1983).
18. M. Kolb, R. Botet, and R. Jullien, “Scaling of kinetically growing clusters,” Phys. Rev. Lett. **51**, 1123–1126 (1983).
19. M. Rottereau, J. Gimel, T. Nicolai, and D. Durand, “Monte carlo simulation of particle aggregation and gelation: Ii. pair correlation function and structure factor,” The Eur. physical journal. E, Soft matter **15**, 141—148 (2004).
20. S. van der Burgh, A. de Keizer, and M. A. Cohen Stuart, “Complex coacervation core micelles. colloidal stability and aggregation mechanism,” Langmuir **20**, 1073–1084 (2004). PMID: 15803680.

7. ASSOCIATED CONTENT

Original code for off-lattice simulations available in github: https://github.com/DiegoRos/DLCA-OffLattice