Molecular dynamics coupled to a thermal lattice Boltzmann code in order to model particle agglomeration in flow

José Fernando Cuenca Jiménez

Department of Physics

Gdansk University of Technology

A thesis submitted for the degree of

MSc Mathematical Modelling in Engineering

September 2013

I would like to dedicate this thesis to all my family and friends,
because they make me every day a better person.

Acknowledgements

I am deeply grateful to Gizem Inci, because of her continuous help, and because without her advices, I would have never finished this work.

I would like to thank Axel Arnold, because he gave me the opportunity to work in ICP.

An special mention to my mathmods Stuttgart team (Bibek dai , Anand vai, Ganu Vai and Azahar), because the made my stay in Stuttgart easier and happier.

Abstract

Particle agglomeration can arise naturally (e.g. dust, salt) or as a result of industrial activities and/or combustion processes (e.g. spray drying, particle flame synthesis). The process itself and its mechanisms are important for many applications since the physical properties of the final structures are mainly determined by the composition, number, diameter and geometric arrangement of their constituent primary particles. Thus, knowing and controlling the extent of agglomeration meets a growing interest in environmental and industrial concerns.

The objective of the project is to develop a simulation model of particles suspended in a flowing fluid using MD simulations coupled to a Lattice Boltzmann (LB) solver. These simulations allowed to

- determine the agglomerate transport and deposition rates depending on flow conditions and agglomerate structure
- understand of the relationship between agglomerate characteristics (i.e. growth kinetics and morphology) and their behaviour in a flow field

2 systems of 2000 and 1000 particles were simulated at 300 K and 600 Kelvin both of them in a known fluid. Simulations using Langevin thermostat were also performed to compare with the LB thermostat ones. This allowed to quantify the influence of fluid flow on the agglomeration process and agglomerate properties. In further applications, this will help to a priori tailor the flow conditions to achieve a desired aggregate morphology.

As a result, reasonable values of aggregates morphology were achieved according to the literature. One of the main conclusions is that taking into account the fluid flow (LB solver) the agglomeration process of the particles suffers a notable acceleration in comparison to the Langevin simulations. One of the main implications of this work could be the possibility of using a known fluid to accelerate an agglomeration process given a suitable fluid and to find a desirable configuration of agglomerates. Another conclusion is that agglomeration process is sensible to the temperature variation and to conclude we will say that number of particles in the system influences the final configuration of agglomerates in LB simulations.

Contents

List of Figures

Nomenclature

Roman Symbols

a	primary particle mean radius
$\hat{A}_c$	normalization constant
$C(r)$	density correlation function
c_s	speed of sound
d	Euclidean dimension
D_f	fractal dimension
$f(\vec{x}, \vec{\epsilon}, t)$	particle distribution function
f_c	cutoff function
k_f	fractal prefactor
L	diameter of the largest obstacle
N	number of primary particles in the aggregate
n_j^{EQ}	local equilibrium value for particles in direction e_i
r_0	position of the center of mass of the cluster
R_g	radius of gyration
r_i	position of the ith spherule center

t time

$\vec{u}$ velocity

$\vec{u}_0$ inflow velocity

$\vec{x}$ particles location

Greek Symbols

δ_t time step

δ_x grid spacing in direction x

$\vec{\epsilon}$ particles velocity

ν kinematic viscosity

ω_α weighted factor

ρ density

τ relaxation time

Chapter 1

Introduction

Agglomeration of particles is an abounding phenomena in different technical applications of industry, appearing in form of either products or waste. In the last decades, environmental problems have gained a lot of importance and therefore, this complex-structured particles need to be handled, where transport, agglomerate growth, deposition and other complex processes have to be taken into account. Dietzel and Sommerfeld [2010] and Dietzel and Sommerfeld [2011].

In nanotechnology research field, the structure and structural properties of nanoparticles is crucial for the prediction of the particle behavior.

Due to the complexity of nano-particles and agglomerates, experimental techniques are limited to very elementary structures Cuendet [2008]. On the other hand, numerical simulations are able to provide an important alternative of experimental data, and can be an efficient way to analyze the structure and behavior of the particles and consequently to create new products faster and cheaper.

A precise calculation of the essential physical properties of nano-particles is needed for planning their technological applications and for improving the aspects related to their manufacturing. The properties of nanoparticles (seen as single particles), has been studied deeply in the last years, but the geometry of the aggregates (almost spherical particles sized between 1 nm and 100 nm that join together and remain fixed once they touch an initial point of contact), is described by the fractal geometry.

In this thesis, we use the Lattice Boltzmann Method (LBM) to be compared with a Langevin thermostat method in order to perform simulations of agglomerates

to analyze agglomerate transport and deposition depending on flow input and agglomerate structure. We try to present a comparative study of two computer simulation methods. Applying these methods to the same physical system will allow us to draw a detailed and quantitative comparison. This idea to compare this 2 kind of methods have been used in some studies like Pham et al. [2009]. LBM method has become more important in the last years because of his easy implementation in comparison to other methods that directly try to discretize the Navier-Stokes (NS) equations. Gao and Wang [2010].
In LBM method, the motion of particles and growth of agglomerates are simulated by molecular dynamics (MD) and it is coupled to a Lattice Boltzmann Code to take the fluid into account. This last part is what makes the difference between using a LB thermostat and a Langevin one.

1.1 Structure of this thesis

This thesis is divided into 7 main chapters. After this first introduction chapter, we will move to the second one, where the Lattice Boltzmann Method (LBM) is presented in details. It is important to understand the physics behind the theory. That way, we will be able to interpret the results from our simulations. Therefore, all the important equations and origins from LBM are presented in this chapter, ending with the treatment of the Boundary conditions for this method.
in Chapter 3, The Software ESPResSO is introduced and how to deal with the units in this program.
In Chapter 4, we present an introduction to particle agglomeration. It is presented how the "collision model" works, how it is treated the characterization of the agglomerates (i.e, fractal dimension, radius of gyration, longest distance, etc). We finish this chapter explaining how sliding is avoided in the agglomerates.
In chapter 5 we justify and give details about the whole simulation program, giving and explaining all the parameters used and giving the code used (Tcl language).
In chapter 6 we present all the results and discuss the complex agglomerates structures justifying the conclusions based in previous studies.
Finally, We close this Master Thesis with the conclusions in Chapter 7.

Chapter 2

The Lattice Boltzmann Method

In this part of the document, we will talk about the theoretical background of the mentioned method. All the formulas and equations used for this solver in ESPResSO, will be explained with details in this section. Due to the recent development of this method, many aspects from the understanding point of view have been recently discussed.

2.1 A little bit of Historical background

Normally when LBM is presented, it is very common to talk about the lattice gas cellular automata (LGCA).

The first LGCA, this means a special type of Cellular automata (CA) for fluid flow simulations, was introduced in 1973 by Hardy, Pomeau and de Pazzis. Hardy et al. [1973]. The HPP model (name derived from the initials of the authors) conserves momentum and mass, but it does not satisfy the Navier Stokes equation when we reach the macroscopic limit. Röhm [2011].

It was in 1986 when Frisch, Hasslacher and Pomeau brought back the interest in CA when they found that a CA over a lattice with symmetry (hexagonal), satisfies the Navier-Stokes equation in the macroscopic limit Asinari [2001]. The Lattice Boltzman Method (LMB) was proposed to solve an still remaining problem: Big oscillations that limited the applications. After some time, the LBM gained a lot of popularity and it has been one important research area in the last

years.

The basic idea which made it possible to formulate the Lattice Boltzman Methos (LBM) is: just replace the Boolean occupation numbers, involved in the previous LGCA with the corresponding ensemble-averaged populations. The fields were replaced by continuous distributions over the previous lattices. At first, Fermi-Dirac Distributions were used as equilibrium functions, instead of the Maxwellian distribution functions later introduced. Succi [2010].

McNamara and Zanetti published the method in 1988 Boorse and Motz [1966]. As we said, derived from the LGCA, which had many handicaps like noise because of the boolean fields for particle representation when used for CFD simulations.

After some years of development the Lattice Boltzmann Method (LBM) has proven to be an important tool to investigate dynamics in soft matter systems. In coarse grained simulations it allows to replace the solvent by a mesoscopic fluid. On large length and timescales the LBM leads to a hydrodynamic flow field that satisfies the Navier-Stokes equation. The relaxation of fluid degrees of freedom in liquid systems is much more faster than the transport of particles. This differentiation of timescales allows to neglect the microscopic details of the fluid.

Traditionally, CFD solvers get the macroscopic parameters , like velocity $\vec{u}$ and pressure p, by using the Navier-Stokes (NS) equations with finite elements (FE), volumes (FV) or differences (FD) Feichtinger et al. [2005]. On the other hand, in lattice Boltzman method (LBM), these macroscopic parameters come from propagation and collision of particles , that are represented by the particle distribution functions $f(\vec{x}, \vec{\epsilon}, t)$, being $\vec{x}$ the location,$\vec{\epsilon}$ the particles velocity and t time.

How do we get the particles interaction? Solving the microscopic kinetic equation for the particle distribution function $f(\vec{x}, \vec{\epsilon}, t)$ for each time step. It is important to notice that after the timing and spacing discretization, the fluid domain has been subdivided into lattice cells. These cells comprises values for the particle distribution $f_\alpha(\vec{x}, e_\alpha, t)$ with particles velocities e_α for each discrete lattice direction α.

The next important matter is to see how the discretization works. It is done in the following way: $\delta_x = e_\alpha \cdot \delta_t$ being δ_x the grid spacing in direction x and δ_t

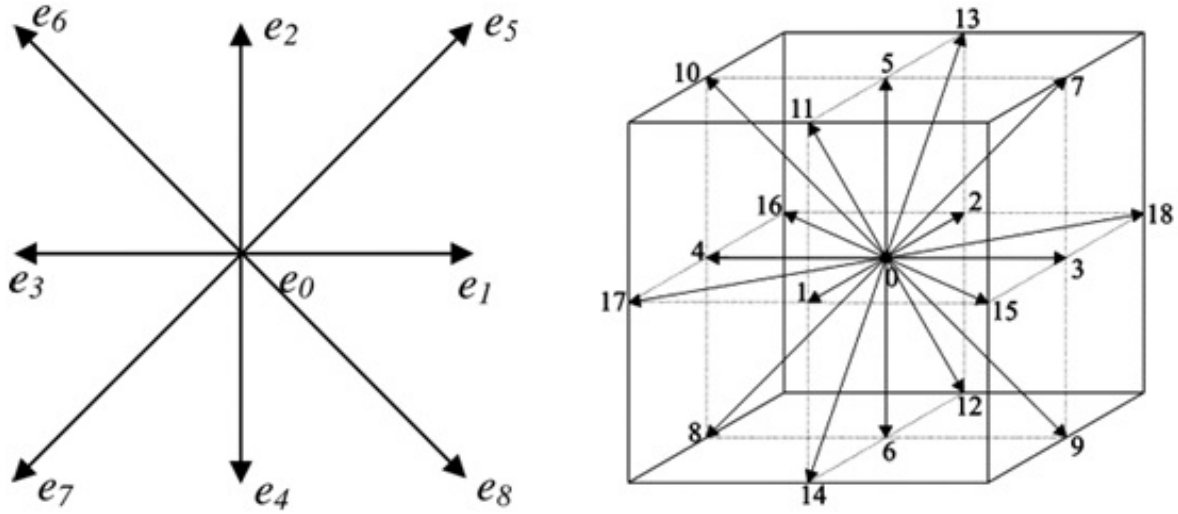

Figure 2.1: The left picture is D2Q9 model. Every cell contains 9 particle distribution particles. The right picture is D3Q19 model: 19 velocities

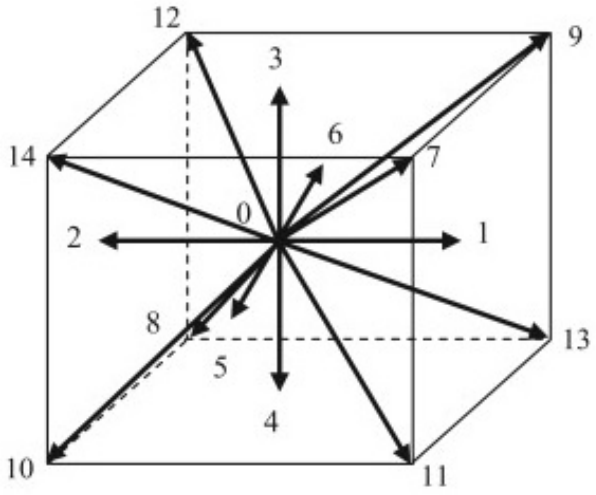

Figure 2.2: D3Q15 model

the time-step.

The meaning of this is that a particle moves from a cell to another in 1 time step. An exception are those particles who have no velocity that will remain in their original cells.

In every time step the particle distributions functions are collided and redirected to their neighboring cells according to their lattice direction, resulting a fluid behaviour equal to the Navier Stokes (NS) equations.

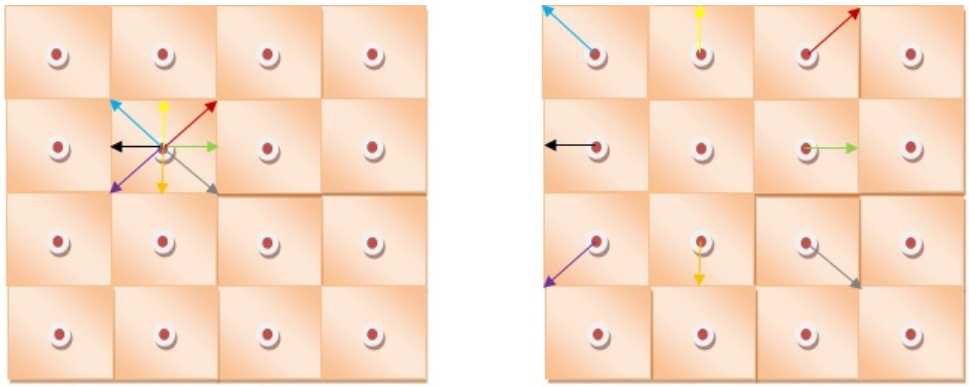

Figure 2.3: Stream step. Particles are redirected from the current cell to the neighboring ones. Left picture: before stream step. Right picture: after stream step

2.2 Relevant aspects of the Lattice Boltzmann Method

The main goal of the LBM is to calculate and get the particle distributions for the next time step. This can be performed using the Bhatnagar-Gross -Krook (BGK) model.

The Bhatnagar-Gross -Krook operator term refers to a collision operator used in LB equation and in the LBM and it is given by the following espression

$$\Omega_i = -\tau^{-1}(n_i - n_i^{EQ}), \tag{2.1}$$

where n_j^{EQ} is a local equilibrium value for the population of particles in the direction of link e_i .The term τ is a relaxation time, and related to the viscosity.

After the discretization of the BGK model in time and space we get the following equation,

$$f_\alpha(x_i + e_\alpha \delta t, t + \delta t) - f_\alpha(x_i, t) = -\frac{1}{\tau}\left[f_\alpha(x_i, t) - f_\alpha^{(eq)}\right], \tag{2.2}$$

and this equation can be divided into two steps: the stream step

$$f_\alpha(x_i + e_\alpha \delta t, t + \delta t) = \tilde{f}_\alpha(x_i, t + \delta t), \tag{2.3}$$

Representing the propagation of the particles. The particle distribution functions of every cell x_i are spread to the neighboring cells $x_i + e_\alpha \cdot \delta_t$ as we can see in the Figure (2.4).

Now we focus on the second part of the equation (2.1) (particle distribution functions after the second step: the collide step),

$$\tilde{f}_\alpha(x_i, t + \delta t) = f_\alpha(x_i, t) - \frac{1}{\tau}\left[f_\alpha(x_i, t) - f_\alpha^{(eq)}\right]. \tag{2.4}$$

The equation (2.3) is about the collision of the particles during their movement. During the collision step the particle distributions of the last time step $f_\alpha(\vec{x}_i, t)$ are adjusted by the weighted deviation from the local equilibrium distribution $f_\alpha^{(eq)}$ Feichtinger et al. [2005].

We try to explain graphically how particle distribution varies during the collide step (See Figure (2.4)). As it is explained in equation (2.3), the new particle distributions result from a weighting between the old distributions and the equilibrium distributions. In the case for example of $\tau = 2$, the results are an average of the old and the equilibrium functions.

The local equilibrium distribution functions $f_\alpha^{(eq)}$ are calculated in the following way:

$$f_\alpha^{(eq)} = \omega_\alpha \cdot \rho \left[1 + \frac{3}{c^2}(e_\alpha \cdot \vec{u}) + \frac{9}{2c^4}(e_\alpha \cdot \vec{u})^2 - \frac{3}{2c^2}\vec{u}^2\right], \tag{2.5}$$

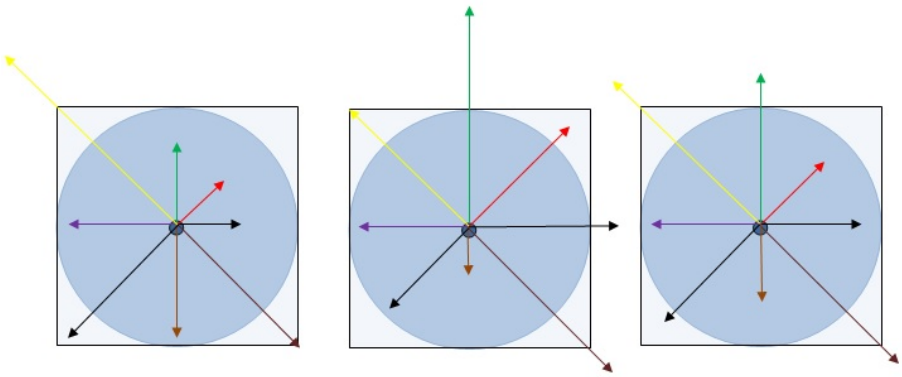

Figure 2.4: Change of the particle distribution during the collide step

where $c_s = \frac{c}{\sqrt{3}}$ with c_s the speed of sound.

This equation means that future collisions would not affect the new particle distributions. It is important to highlight that the collision step will not change the velocity $\vec{u}$ and density ρ.

Usually the parameter c is fixed to 1. Indeed, the weighted factor ω_α are also initial values for the particle distribution functions and their values change depending on the lattice model used.

After defining the main and more important equations of the LBM, nor the following question comes: How to finally obtain the macroscopic values? With the following three equations we get the pressure and the velocity (variables of interest) Feichtinger et al. [2005].

$$\rho = \sum_{\alpha=0}^{n} f_\alpha \tag{2.6}$$

$$p = \rho \cdot c_s^2 \tag{2.7}$$

$$\vec{u}\rho = \sum_{\alpha=0}^{n} e_i f_\alpha \tag{2.8}$$

It is important to know the dynamic behavior of our simulation fluid. This can be done using the definition of the Reynolds number (RE) , a well known parameter that measures the importance of the non-viscous forces versus the viscous ones. The smaller the RE, the more laminar is the fluid in the next equation

$$RE = \frac{|\vec{u}_0| \cdot L}{\nu} \tag{2.9}$$

Where L is the diameter of the largest obstacle , ν is the kinematic viscosity and $\vec{u}_0$ is the inflow velocity. In LBM and Espresso Software we use only dimensionless quantities, and therefore these units have to be changed as well to their corresponding non-dimensional value.

2.3 Boundary conditions for LBM in ESPResSO

The boundaries implemented in ESPResSO reproduce no-slip boundary conditions for the fluids and this is done by the bounce-back rule. The main idea of this rule, is that the link-based update-rule populations are reflected at the boundary nodes.

The bounce back is done to accomplish a vanishing flow at the boundary (velocity on the boundary has to be zero). This can get done by bouncing back as many particles in each direction as are streamed in the opposite direction. (See Figure (2.5)).

Indeed, this makes possible to solve the linear lattice Boltzmann equation with bounce-back boundary conditions analytically, too.

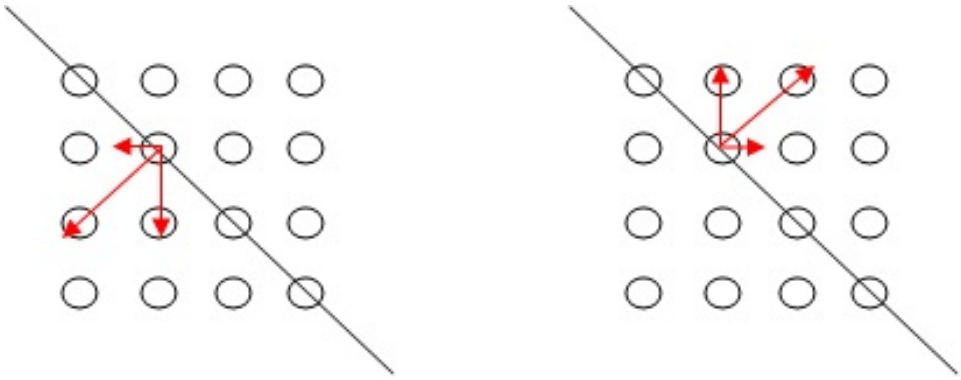

Figure 2.5: Bounce Back Boundary Conditions. Left picture: step t. Right picture: step $t + \Delta t$

Chapter 3

The Software ESPResSO

In this chapter I try to give an introduction to the Software ESPResSO, the software used in this thesis.
ESPResSO (See figure (3.1)) is a package for molecular dynamics (MD) simulations . ESPResSO is the acronym of Extensible Simulation Package for Research on Soft matter systems Ref [a].

Figure 3.1: Logo of ESPResSO. Source www.espressomd.org

A large amount of materials such as glasses, hydrogels, colloids, etc, are included in the term *soft matter* Dünweg and Ladd [2009]. In this kind of material, many properties are interesting from the molecular point of view rather than the atomic one Ref [b]. In conclusion, when one is doing an atomic simulation, to be able to notice the important effects, one should take in account millions of atoms,

what sometimes is very expensive and not possible.
Simulations need a lot of time steps to get some proper data, and therefore an efficient and fast software is still needed.
ESPResSO relies on other packages. In this sense is important to highlight that Tcl language is the interpreter for the simulation control. This means that every code should be written in a Tcl script.
These are some important characteristics are of ESPResSO Arnold et al. [2013]:

- Optimized for coarse-grained models: specially designed for these kind of models. In this context, the software gives the user the opportunity among others to create rigid bodies, to generate new bonds during simulations, or for example, and related to this thesis, to couple a Lattice Boltzmann fluid to the particle-agglomerates simulation. Moreover, the software treats the physical units in a flexible way (values can be given in any unit system). This deserves more attention and in the next part of the chapter I will explain deeply how the ESPResSO deals with units.

- Interaction between the user and the simulation engine via scripting language Tcl: the Tcl script language (Tool Command Language) determines all the simulation parameters such as length of the simulation box, number of particles, etc. Most of these values can be modified during the simulations . This is one of the advantages of using Tcl in simulations. Indeed, the Tcl interpreter contains some special commands as an extension to Tcl.

- The simulation engine is written in C.

- It has implemented all these possible simulation methods:

  - Ensembles

    1. NVE: The constant-energy, constant-volume ensemble (NVE), is obtained by solvingNewton's equationwithout any temperature and pressure control. Energy is conserved when this ensemble is generated. This ensemble is not recommended for equilibration because, without the energy flow facilitated by the temperature control methods, the desired temperature is not achieved.

2. NVT: This ensemble is obtained by controlling the temperature through direct temperature scaling during the initialization stage and by temperature-bath coupling during the data collection phase. The volume is kept constant throughout the run.
3. NPT: This ensemble allows control over both the temperature and pressure. The unit cell vectors are allowed to change, and the pressure is changed by adjusting the volume. This is the right ensemble when the correct pressure, volume, and densities are important in the simulation.

- Algorithms for charged systems:
  1. P3M for fully periodic systems
  2. ELC and MMM-family of algorithms for charged systems with non-periodic boundary conditions.
  3. MEMD (Maggs algorithm)
- Hydrodinamics
  1. DPD (as a thermostat)
  2. Lattice Boltzmann
- Non-equilibrium MD to simulate shear flow
- Parallel tempering
- Metadynamics
- Rigid bodies via virtual sites

3.1 Units on ESPResSO

Normally, in the majority of molecular dynamics (MD) software we define a set of units, this means that for example all times are measured in nano-seconds. On the contrary, in ESPResSO, mass (m) , length (σ) and energy (ϵ) are chosen by the user Inci [2012].
The rest of parameters such as time (t), pressure (p), density (ρ), viscosity (μ), gamma langevin constant, temperature (T), etc will be expressed in terms of

the three original units m^*, σ^*, ϵ^* by the dimensional analysis technique (see equations below).

$$\text{reference time: } t^* = \sigma^* \left(\frac{m^*}{\epsilon^*} \right)^{1/2} \tag{3.1}$$

$$\text{reference temperature: } T^* = \frac{\epsilon^*}{k_B} \tag{3.2}$$

$$\text{reference density: } \rho^* = \frac{m^*}{\sigma^{*3}} \tag{3.3}$$

$$\text{reference viscosity: } \mu^* = \frac{m^*}{\sigma^* t^*} \tag{3.4}$$

$$\text{reference gamma langevin constant:} \frac{1}{t^*} \tag{3.5}$$

m^* will we the mass of the particle, σ^* will be the diameter of the particle and ϵ^* will be the thermal energy of the system ($\epsilon^* = \frac{3}{2}KT$).
After defining the reference quantities, we can obtain the dimensionless parameters as follows:

$$\tilde{r} = \frac{r}{\sigma^*} \tag{3.6}$$

$$\tilde{E} = \frac{E}{\epsilon^*} \tag{3.7}$$

$$\tilde{T} = \frac{T}{T^*} \tag{3.8}$$

$$\tilde{t} = \frac{t}{t^*} \tag{3.9}$$

$$\tilde{\mu} = \frac{\mu}{\mu^*} \tag{3.10}$$

$$\tilde{\rho} = \frac{\rho}{\rho^*} \tag{3.11}$$

$$\tilde{\gamma} = \frac{\gamma}{\gamma^*} \tag{3.12}$$

To get proper simulations, the simulation time step must be smaller than the relaxation time of the particles.
In this thesis, Lattice Boltzmann Method (LBM) is used to simulate the fluid. The time step τ for the LBM must be equal or larger than the MD simulation time step.

Chapter 4

Particles and Agglomerates

In this chapter we explain how agglomerates behave: movement, particles attachment, characterization and how the agglomerates are implemented in ESPResSO. How to simulate the fluid by LBM is explained in chapter 2. The other part of the simulation (particles) needs to be presented deeply as well.

4.1 Agglomerates. A general view and formation

In this study, particles and its aggregations are simulated. Particles of same mass and shape (we suppose the nano-particles spherical) are connected to others (or groups of them) in one unique point contact. This process in time can lead to agglomerates of particles (see figure (4.1)).

We can approximate usually our particles as spheres because their behaviour in a flow normally reproduces the behaviour of spheres Iglberger [2005] and Iglberger et al. [2005].

The radius and mass of primary particles are fixed before the simulation starts (In our case we simulate soot particles). The particles can be seen as a point with the known mass in the center of mass of the spherical body. An agglomerate can also join another agglomerate.

There are simple agglomerates that have been used for model testing like cross-shape and linear shape but the reality is far away from these simple models (See

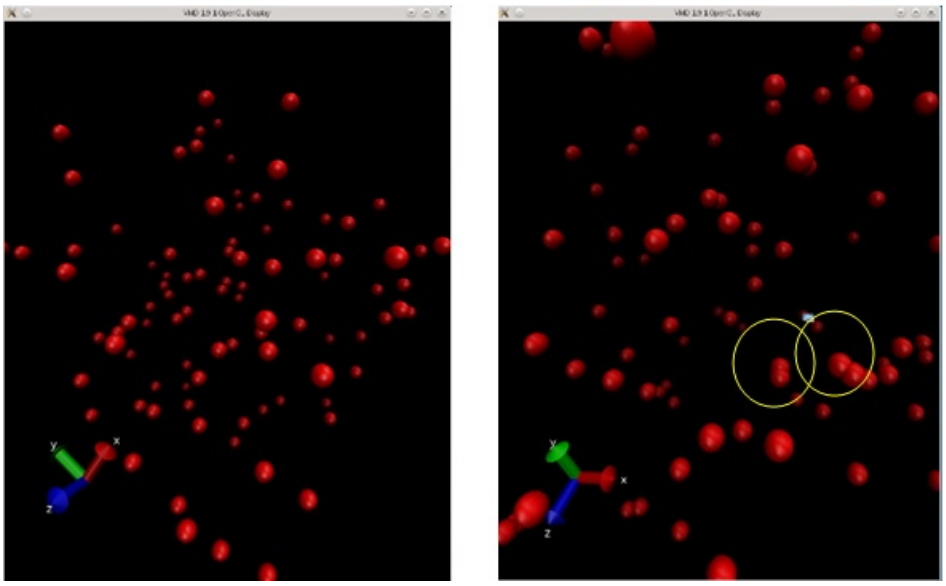

Figure 4.1: VMD view. On the left picture time step = 0. All particles are separated. On the right picture agglomeration process has started after some time step.

figure (4.2))

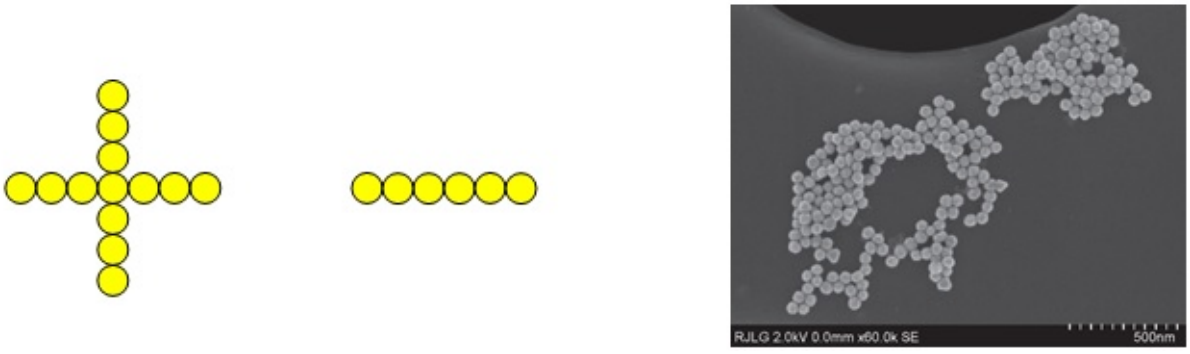

Figure 4.2: Left picture: Cross and linear-shape agglomerates. Right picture: Electron images from gold nano-particles. Real agglomerate. Source www.azonano.com

The next part is to explain how ESPResSO deals with the rigid bodies (agglomerates are formed of particles that move all together) and how are imple-

mented the forces between the particles that keep the agglomerate as a whole body (conexions). This will be treated in the next subsections.

4.2 Connection between particles: Bonding

When an agglomerate is created, some particles starts joining each other and sticking. In this section we try to explain what are the main physical and chemical processes that occur and how ESPResSO deals with this question.
To keep two particles together, there should be some adherence forces or chemical bonds that make this happen. When two particles meet, due to the induced-dipole forces, a short-range attraction appears and the consequence is an elastic deformation that can lead to make the contact between then stronger, or they can even create bridges among them.
During this kind of situations, the particles deform and thus they lose their spherical shape. Therefore sticking processes cannot be simulated properly. To solve this problem in Molecular Dynamics this can be done by introducing potentials or binding agents provided by the software.
The method used is "Angle bonded virtual particles model" (AnBV) Inci [2012]. In this method, two virtual particles are introduced in the contact point and the real particles pair is attached to them. How the virtual pair is connected to the real pair? By angle-dependent potential with angle $\Theta = \pi$ (see figure (4.3)).

4.3 Characterization of aggregates

4.3.1 Fractal dimension and Radius of gyration

Aggregates are structures formed by the summation of small and almost spherical particles. The complexity of this structures requires additional parameters to characterize the morphology and the basic properties of these aerosols Samson et al. [1987].
An irreversible aggregation of particles has as a result the creation of fractal aggregates, with a power-law relationship between the cluster radius of gyration Rg and the number of monomers N in a cluster. This relation was introduced by

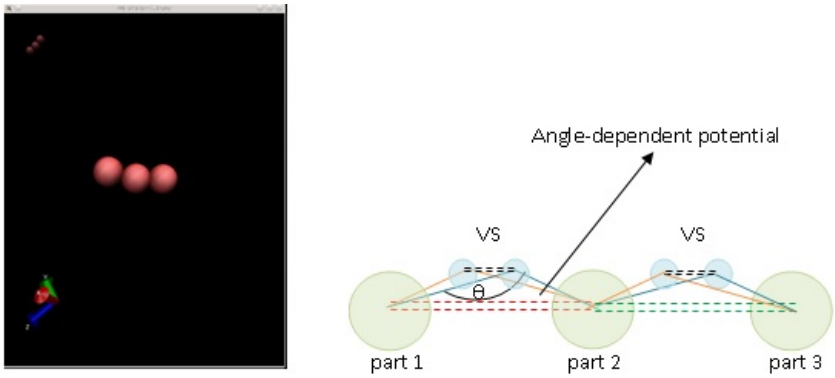

Figure 4.3: Agglomerate of 3 particles (two contacts). For every contact, a virtual pair is created at the collision point. The bond between the real pair or collided particles avoid motion and the collision point and the angular potential between real and virtual particles avoid particles from sliding among each other.

Forrest and Witten [1979] through the following equation:

$$N = k_f \left(\frac{R_g}{a} \right)^{D_f} \tag{4.1}$$

where a is the primary particle mean radius, k_f is the fractal prefactor and D_f is the fractal dimension. R_g is the radius of gyration and N is the number of particles in the aggregate.
The next step is to define what is the radius of gyration. There are different ways to define it but the normal way to do it is by the mean square of the distances between spherule centers and the center of mass of the aggregate Pierce et al. [2006].

$$R_g^2 = \frac{1}{N} \sum_{i=1}^{N} (r_i - r_0)^2 \tag{4.2}$$

$$r_0 = \frac{1}{N} \sum_{i=1}^{N} r_i \tag{4.3}$$

where r_i denote the position of the ith spherule center the and r_0 the center of mass of the cluster.
There is still a more accurate definition for the radius of gyration of each agglomerate. This is due to the fact that all transfer processes and the chemical reactions appear on the aggregate surface , and therefore we can express the radius of gyration as the mean square distance between r0 and all points in the spherule surfaces (see the following equation).

$$R_g^2 = \frac{1}{N}\sum_{i=1}^{N}(r_i - r_0)^2 + a^2 \tag{4.4}$$

The bigger the aggregate, more similar are both definitions defined in eq (4.6) and eq (4.8).
It exists another sense to characterize the morphology of an aggregate. This can be done by the two-point density -density correlation function Filippov et al. [2000] (see the following equation):

$$C(r) = \langle\langle \rho(r_0)\rho(r_0 + r)\rangle\rangle_{|r|=r} \tag{4.5}$$

having in the case of fractal-aggregates the following form:

$$C(r) = A_c r^{D_f - d} \tag{4.6}$$

where A_c is just a constant, d is the Euclidean dimension and D_f the fractal dimension.
This equation can be rewritten as follows (see equation below).

$$C(r) = \hat{A}_c \left(\frac{r}{R_g}\right)^{D_f - d} f_c\left(\frac{r}{R_g}\right) \tag{4.7}$$

where $\hat{A}_c$ is a normalization constant and f_c a cutoff function.

In our simulations the fractal dimension of the agglomerates are calculated from the slope of the log-log plot of the radius of gyration R_g versus the number of particles N.
Taking logarithms from the statistical law (4.5), we get the following equation

$$\log N = \log k_f + D_f \log R_g - D_f \log a \tag{4.8}$$

As an example we illustrate how to calculate the fractal dimension of an agglomerate got from an small simulation.
First we simulate a line-shape agglomerate of 6 particles and take a picture in time step 100 (see figure (4.4)) and calculate the fractal dimension (figure (4.5) and table).

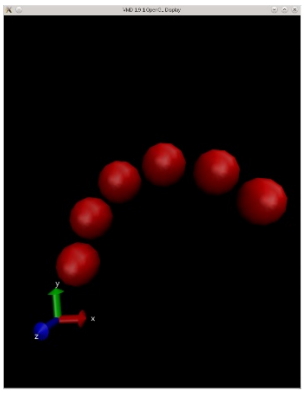

Figure 4.4: VDM view of the 6 particles agglomerate in time step 100

R_g	N	$\log R_g$	$\log N$
2	2	0.693	0.693
4	4	1.386	1.386
6	6	1.791	1.791
8	6	2.079	1.791
10	6	2.303	1.791

Table. Radius of gyration versus number of particles for D_f at $t = 10$

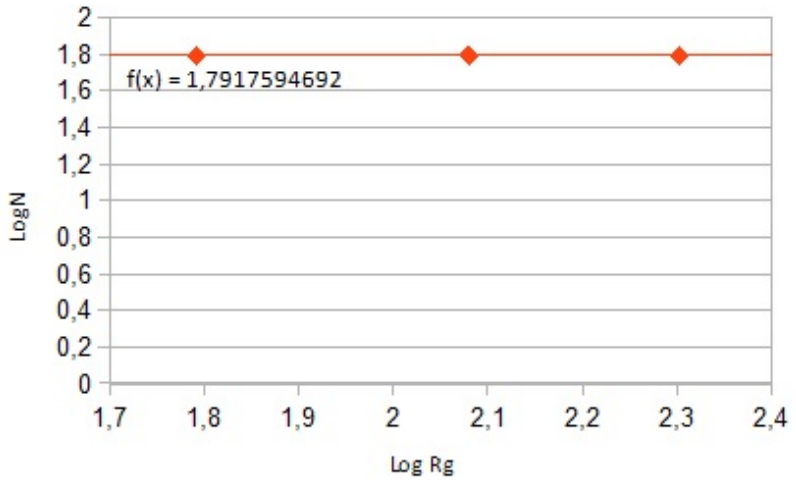

Figure 4.5: Fractal dimension is the slope of the line and has a value of 1.79

4.3.2 Longest distance between particles

It is calculated also to contribute in the characterization of agglomerates during the simulation process, and It is defined as the longest physical distance between all the possible pairs of monomers in the agglomerate.
We take the previous simulation example to illustrate a very simple calculation of a longest distance. See figure (4.6).

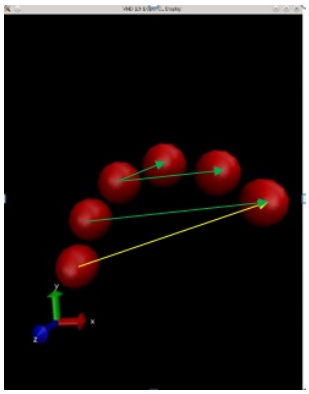

Figure 4.6: Longest distance: the yellow arrow represents the longest distance between monomers in the agglomerate.

Chapter 5

Simulation inputs and Code

In this chapter, we will introduce the parameters that need to be defined before starting the simulations. Moreover we will show with an example how to make the variables dimensionless to work with ESPResSO, and at the end of the chapter, the code (Tcl script) will be presented and explained in details.

5.1 Making the variables dimensionless

As I explained in Chapter 3, ESPResSO works with dimensionless variables that can be obtained dividing the real quantities over the reference ones. We choose mass (m) , length (σ) and energy (ϵ) in such a way that the dimensionless values are (equation below):

$$\text{length: } \tilde{r} = 1 \tag{5.1}$$

$$\text{mass: } \tilde{m} = 1 \tag{5.2}$$

$$\text{energy: } \tilde{\epsilon} = 1 \tag{5.3}$$

This means that $\sigma = \sigma^*$, $m = m^*$, $\epsilon = \epsilon^*$.
We try to clarify this with an example. Let us consider air as a fluid with temperature 300 K. Given the system characteristics (real values) like Fluid Temperature ($T = 300K$), particle diameter (σ = 100 nm), particle density (ρ = 1300 kg/m^3), fluid dynamic viscosity (μ= 1.9831e-5 kg/m*s), air pressure (P=101325 Pa), and the Boltzmann Constant K_B, we can compute the reference parameters and as well others values related to these 3 units like relaxation time, gamma langevin constant, etc (see equations below).

$$\epsilon^* = \frac{3}{2} K_B T = 6.2129 \cdot 10^{-21} \tag{5.4}$$

$$T^* = \frac{3}{2} T = 450 \text{ (K)} \tag{5.5}$$

$$m^* = \rho V^* = \frac{\rho \pi \sigma^3}{6} = 6.805 \cdot 10^{-19} \text{ (kg)} \tag{5.6}$$

$$t^* = \sigma^* \sqrt{\frac{m^*}{\epsilon^*}} = 1.0467 \cdot 10^{-6} \text{ (s)} \tag{5.7}$$

$$\gamma = \frac{1}{t^*} = 955381.61 \text{ (s}^{-1}\text{)} \tag{5.8}$$

$$\text{relaxation time } t'^* = \frac{\rho \sigma^2}{18 \mu} = 3.64 \cdot 10^{-8} \text{ (s)} \tag{5.9}$$

$$\mu^* = \frac{m}{\sigma^* t'^*} = 5.699562 \cdot 10^{-4} \text{ (kg/m} \cdot \text{s)} \tag{5.10}$$

After that, we are able to make dimensionless the variables (real value / asterisk one). So, we obtain from this examples the following values: $\tilde{r} = 1$, $\tilde{m} = 1$, $\tilde{\epsilon} = 1$, $\tilde{t} = 0.03479$, $\tilde{T} = 0.667$, $\tilde{\gamma} = 10.96$, $\tilde{\mu} = 0.106$. As a conclusion, we could state that these should be the values to be introduced in a simulation and not the real ones.

5.2 Inputs

Now in this section, we define all the parameters that must be specified in the Tcl script before running the simulation. Some values for the fluid, some for the particles and some for the system characteristics will have to be defined.

1. Particles, domain and system characteristics
    - number of particles
    - box length
    - boundary conditions (periodic, etc)
    - Initial position of particles
    - time step for the MD simulation
2. Thermostat, potentials and bonded potentials.
    - type of thermostat (Langevin, Boltzmann, etc)
    - type of potential used and parameters. (Lennard Jones, etc)
    - angle potential.
3. Fluid parameters (just for the Lattice Boltzmann thermostat case)
    - Temperature
    - density
    - viscosity
    - grid spacing
    - time step for the fluid simulation
    - friction coefficient

5.3 Simulation Code

In this part of the thesis, we present the Tcl script that was used to do the simulations. The agglomeration code was given by Gizem Inci, ITV. We analyze every part of the code and we explain the purpose they have in the simulation. As we said before, we are simulating nano-particles in a fluid (in this case air) using a Lattice Boltzmann thermostat or a Langevin one.
The modes that need to be activated in ESPResSO for this simulation are called by Algorithm 1

```
require_feature "COLLISION_DETECTION"
require_feature "VIRTUAL_SITES_RELATIVE"
require_feature "EXTERNAL_FORCES"
require_feature "LB" # (In LB thermostat case)
```

The total number of particles, box dimensions and initial random position of particles are defined in Algorithm 2. Indeed a vtf file is create to be able to watch the simulation in a viewer as VMD for instance.

```
#require_max_nodes_per_side {1 1 1}

set particle 1000
# (or 2000)
cellsystem domain_decomposition -no_verlet_list
set box_length 100
setmd box_l $box_length $box_length $box_length
set pidlist [list]
set fl [open LB1.vtf a]
set filei 0

for {set i 0} {$i<$particle} {incr i} {
set posx [expr $box_length*[t_random]]
set posy [expr $box_length*[t_random]]
set posz [expr $box_length*[t_random]]
```

```
part $i pos $posx $posy $posz type 0
}


writevsf $fl
```

In the next Algorithm 3, Lattice Boltzmann fluid parameters are introduced by the user. Moreover, potentials and boundary conditions are defined. Also the collision model is called. Finally some functions are presented that will be called later for the calculations of fractal dimension and longest distance.

```
#--------------------------------------------------------------------------#
# PART-2---> DEFINE THE MODELS AND CONSTANTS OF EQUATIONS
#--------------------------------------------------------------------------#
set min [analyze mindist 0 0]
puts "THE FIRST MINIMUM DISTANCE : $min"
setmd time_step 0.001
setmd skin 0.9999
#------------------LATTICE BOLTZMAN THERMOSTAT--------------------#
set temp 0.667
lbfluid dens 0.0017056513 visc 0.1061032954 agrid 2 tau 0.1 friction 15


thermostat lb $temp
#---------------lENNARD JONES POTENTIAL-------------------#
set eps 1.0
set sigma 1.0
set rcut 2.5
setmd min_global_cut 1

inter 0 0 lennard-jones $eps $sigma $rcut auto
#---------------BONDED POTENTIALS-------------------#
inter 1 angle 500 [PI]
inter 0 harmonic 500 1.0
```

```
#---------------BOUNDARY_CONDITIONS-----------------------#
#...Periodic boundary cond. Later, this may be changed....#
setmd periodic 1 1 1
#-----------------COLLISION-------------------------------#
on_collision bind_at_point_of_collision 1.0 0 1 1
#--------------------PARTICLE CONTROL---------------------#
set min 0
set cap 20
while { $min < 0.95 } {
    # set ljforcecap
    inter ljforcecap $cap
    # integrate a number of steps, e.g. 20
    integrate 40
    # check the status of the sytem
    set min [analyze mindist 0 0]
    # this is a shortcut for 'set cap [expr $cap+10]'
    incr cap 20
}
#puts "Warmup finished. Minimal distance now $min"
# turn off the ljforcecap, which is done by setting the
# force cap value to zero:
inter ljforcecap 0
for {set i 0} {$i <$particle} {incr i} {
    lappend pidlist $i
    vtfpid $i
}

#---------------Function----------------------------------#
proc distance_between_2_points {position1 position2 box} {
set ret 0.0
for {set j 0} {$j<3} {incr j} {
  set dist [expr abs([lindex $position1 $j]-[lindex $position2 $j])]
  if {$dist > [expr $box/2.0]} {
```

```
    set dist [expr abs($dist-$box)]
  }
  set ret [expr $ret+ [sqr $dist]]
}
return [expr sqrt($ret)]
}

proc real_distance_between_2_points {position1 position2 box} {
set ret 0.0
for {set j 0} {$j<3} {incr j} {
  set dist [expr abs([lindex $position1 $j]-[lindex $position2 $j])]

  set ret [expr $ret+ [sqr $dist]]
}
return [expr sqrt($ret)]
}

#----------------Fractal Dimension-----------------------#
set initial_radius 0
set incr_step $sigma
```

In the following algorithm the integration procedure starts, indicating the total number of steps and the period that we want for printing the results. In this process, agglomeration lists are printed in every time step indicating the number and identification of monomers included. (To see how Fractal dimension, center of mass and radius of gyration is calculated for every time step, see the rest of the code in Appendix 1).

```
#---------------PARTICLE MOTION--------------------------#

set n_cycle 10000

set n_steps 1000
```

```
set fileid 0
set i 0
while { $i<$n_cycle } {
     integrate $n_steps

     writevcf $fl pids $pidlist

if {$i==[expr $fileid*500]} {
set file [open "LB1_$fileid.txt" "a"]
```

Chapter 6

Results

Simulations have been carried out. 4 using a LBM thermostat and 4 were done using a Langevin thermostat. The purpose of this is to compare the 2 methods and see the influence on the agglomerates properties when we take in account the flow. The simulations were performed using 1000 particles, 2000 particles at 300K and 600 K. In total we have 4 simulations for each method.

6.1 The system

The system and the simulation had the following characteristics:

- $N_1 = 1000$.
- $N_2 = 2000$.
- Spherical Lennard Jones particles with a density of $\rho = 1300$ kg/m^3.
- Cubic box with periodic boundary conditions (PBC) in all directions. Each size of the box has a size of 50 in dimensionless units.
- Random initial position of particles inside the box.
- Particle diameter 100 nm.
- The temperature is **300 K or 600 K** (in dimensionless units 0.667).

- Time-step is 0.001
- The collided particles remain fixed at the initial point of contact.
- The Lennard Jones parameters are (in dimensionless units): epsilon 1 and sigma 1.
- The following features from ESPResSO are activated: "Collision detection", "Virtual_Sites_Relative", "External forces" and for the LBM simulations we also need "LB".
- Functions used to calculate fractal dimension, center of mass and longest distance in order to characterize the agglomerates that are created.
- Creation during the simulation of a vtf file in order to visualize the whole simulation in VMD.

6.2 First approach and final inputs for the simulations

The first idea was to consider the surrounding fluid is assumed to be air with viscosity of $\mu = 1.98310^5$kg/ms. Well, in fact, this system, using the Langevin Thermostat, had no problem, but when we tried to simulate our system considering air with the LBM, we could not simply perform the simulation.

The parameters needed to initiate the LB thermostat in ESPResSO are the following and considering air as the surrounding fluid had the following values (dimensionless):

- density of the fluid: 0.01
- viscosity of the fluid: 0.106
- agrid (grid spacing for LBM): 1
- tau (LB time-step): 0.01

Segmentation error occurred and this can be explained because of the low density fluctuations when Lattice Boltzmann method used Xiaoyi and Li-Shi [1997].

Density of the fluid was increased to reach values that have been tried in Lattice Boltzmann Simulations before Lenz et al..

- **density of the fluid: 1**
- **viscosity of the fluid: 0.8**
- **agrid (grid spacing for LBM): 1**
- **tau (LB time-step): 0.01**

6.3 Simulation results

Cluster size distribution in time, average fractal dimension and longest distance as a function of time and size of the agglomerates will be presented and discussed.

Results for each simulation are presented together (both methods) in order to be able to compare easily. VMD software will help to visualize and analyze the results. See figure (6.1).

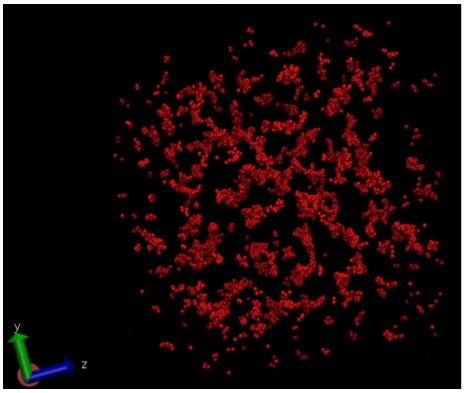

Figure 6.1: Orthographic view of the box simulation (300 K, 2000 particles and Langevin thermostat in time-step 7500. We can see how agglomeration process is going on.

6.3.1 Simulation 1: 300 K and 1000 particles

The agglomeration growth is assessed and shown in the figure (6.2) for LBM method and Langevin method. At the beginning of the simulations randomly placed 1000 particles collide and create clusters, and then with time cluster-cluster collisions form bigger clusters. To understand better the results, the study is divided between small clusters (< 10 monomers) and big clusters (> 10 monomers). Fractal dimension and longest distance will be calculated for clusters with 10 or more particles.

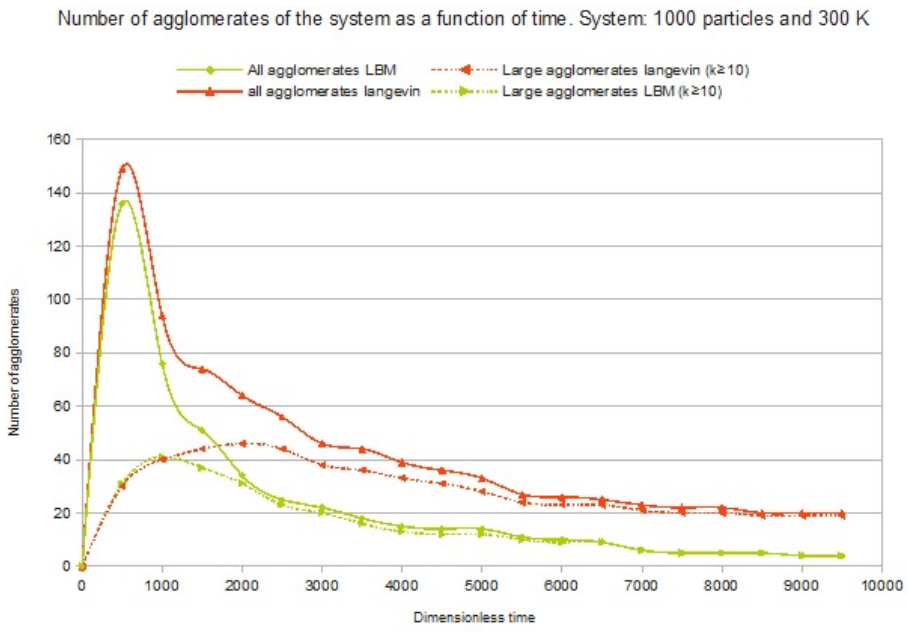

Figure 6.2: Number of agglomerates as a function of time. Every 500 time-steps number of agglomerates is printed out.

Now, in order to characterize the agglomerates, the average fractal dimension as a function on time is shown in the figure (6.3).

As we are moving on in the simulation process, values of the fractal dimension start to oscillate between 1.7 and 1.9, values in concordance with the studies of Filippov et al. [2000], for soot particles.

The next step is to do the same with the longest distance between particles (See figure (6.4)). The process is similar to the previous one. every 500 time steps,

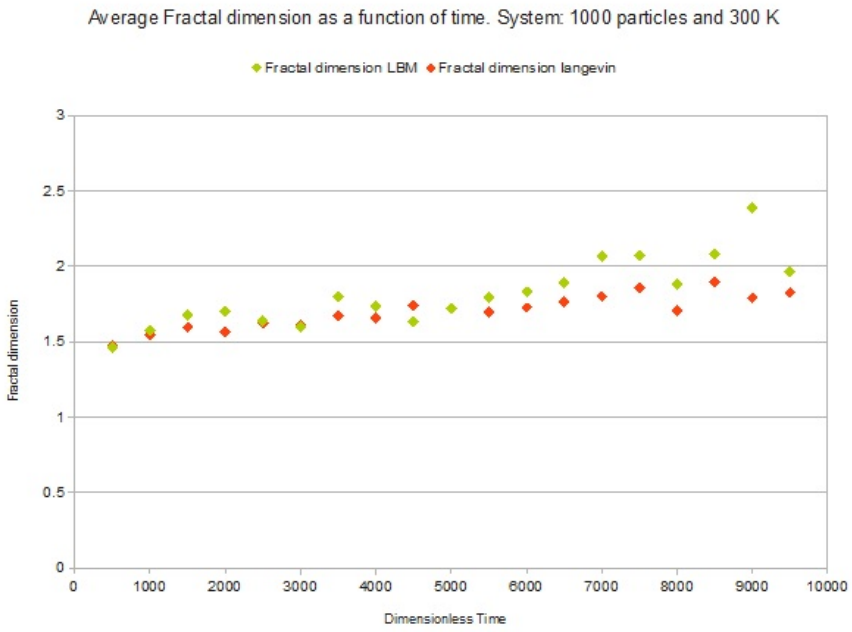

Figure 6.3: Average fractal dimension vs time.

we calculate the longest distance between particles within each agglomerate and then compute the average (clusters with 10 or more particles).

This result is what we were expecting. As the simulation moves on, agglomerates grow and therefore in average the longest distance between particles must grow as well, as shown in the graphic. The agglomerates in LBM method are bigger, therefore the longest distance between particles will be bigger also. We will discuss the reason of this further when we present all the results.

Another interesting way of characterize the agglomerates can be what was proposed in Isella and Drossinos [2010]. This is to represent the fractal dimension and longest distance between particles, as a function of cluster size. The process to do this is the following: we collect all the data, and for every size of agglomerate, represent all the values of fractal dimension and longest distance.

This is shown in the figures (6.5) and (6.6).

As we can see in figure (6.5), most of our agglomerates have a size between 20 and 90 monomers, and in that interval, our trend line gives values between 1.7 and 1.9 for the fractal dimension, again expected values according to Filippov

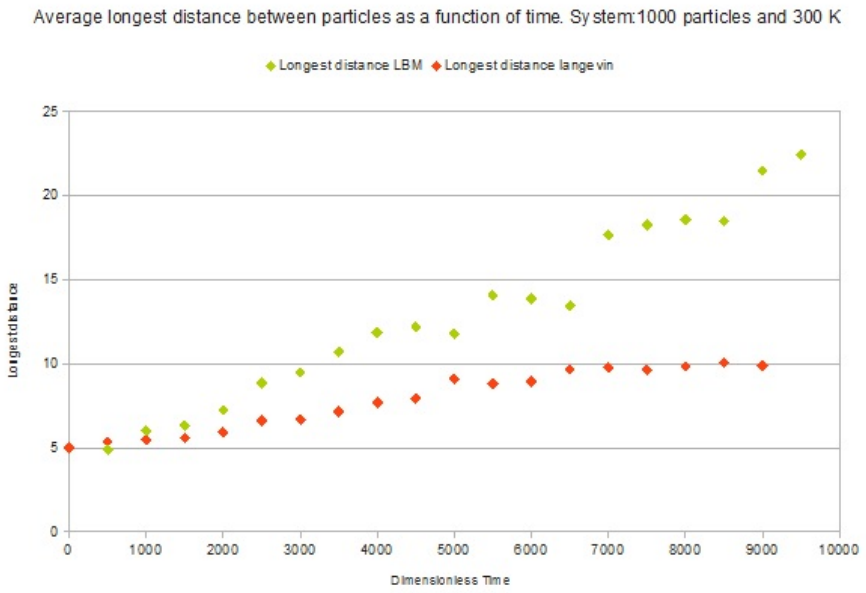

Figure 6.4: Average longest distance between particles vs time.

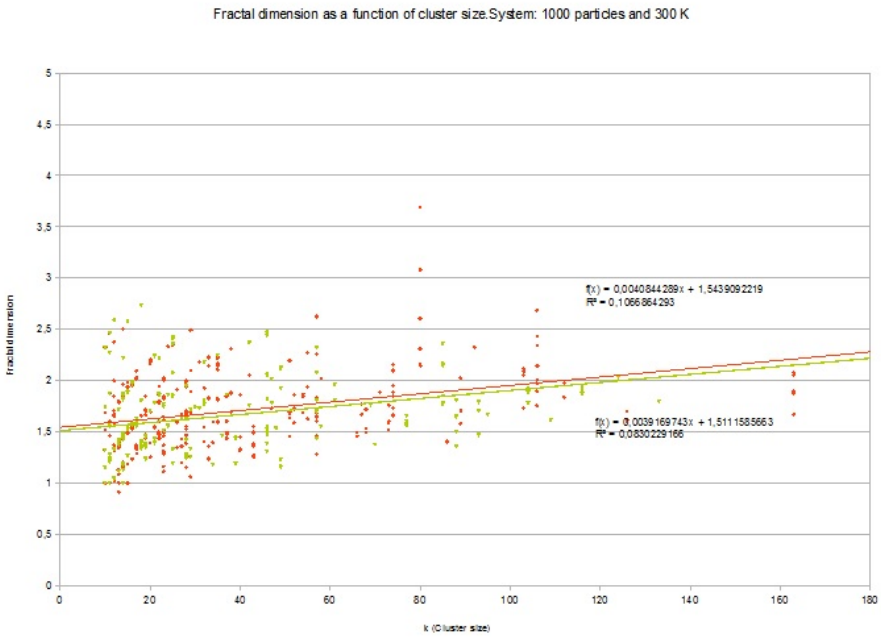

Figure 6.5: Fractal dimension vs cluster size. Green values correspond to LBM and red ones to Langevin.

et al. [2000] and Inci [2012]. Both methods produce a quite similar prediction.

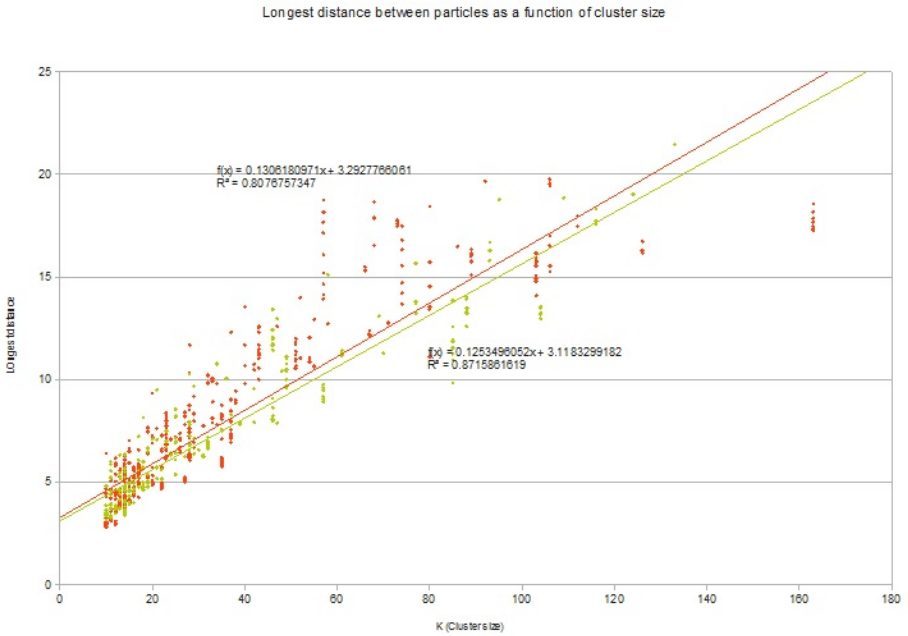

Figure 6.6: Longest distance vs cluster size. The trend line gives us an idea about values of longest distance between particles for every cluster size.

6.3.2 Simulation 2: 300 K and 2000 particles

We repeat the process and first of all, in figures (6.7), (6.8) and (6.9) we show the agglomeration growth, average fractal dimension and average longest distance.

As we saw in the simulation 1, agglomerates in the LBM simulation, show a little higher fractal dimension and longest distance. Taking a look into the figures (6.2) and (6.7), we can observe that cluster formation and cluster growth speed is much higher in LBM simulations than in langevin ones. This means that agglomeration process is faster in LBM method.

If we investigate the literature, we find the following statemens: *"Typically collisions are induced by Brownian motion, gravity, and velocity gradients in the uid carrying the particles since these phenomena bring about relative velocities between particles. Also particle inertia can be a source of collision"*, Derksen and Eskin [2011].

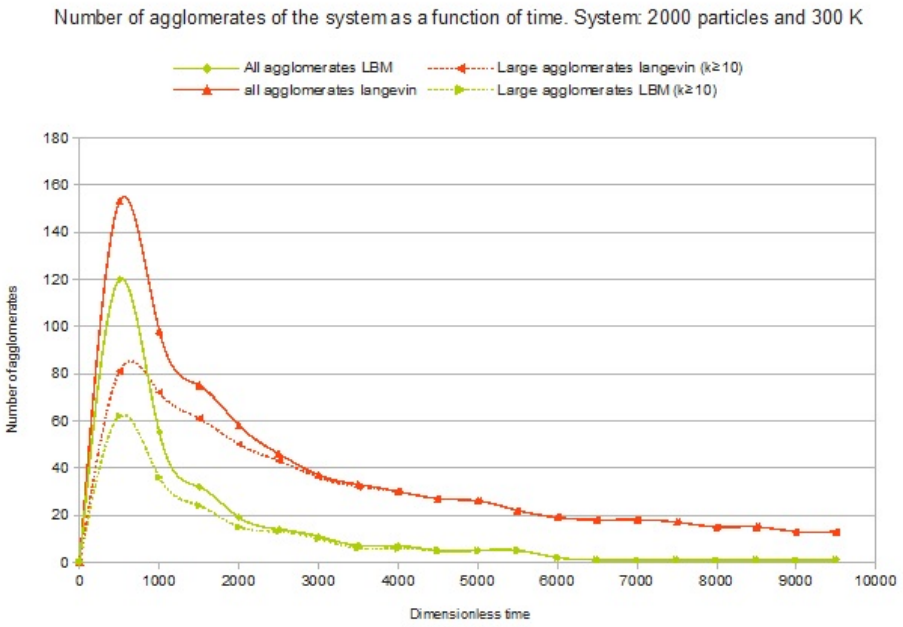

Figure 6.7: Number of agglomerates as a function of time. Every 500 time-steps number of agglomerates is printed out.

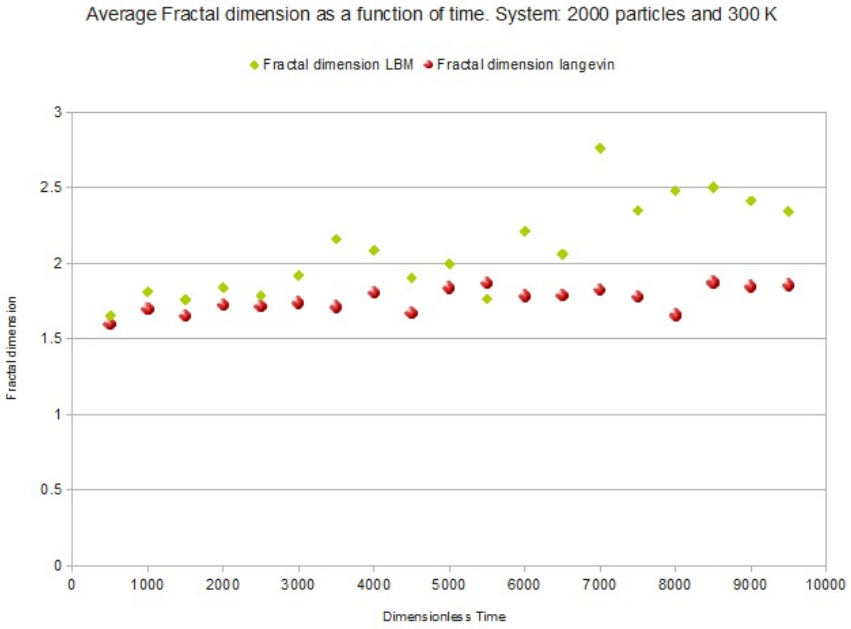

Figure 6.8: Average fractal dimension vs time.

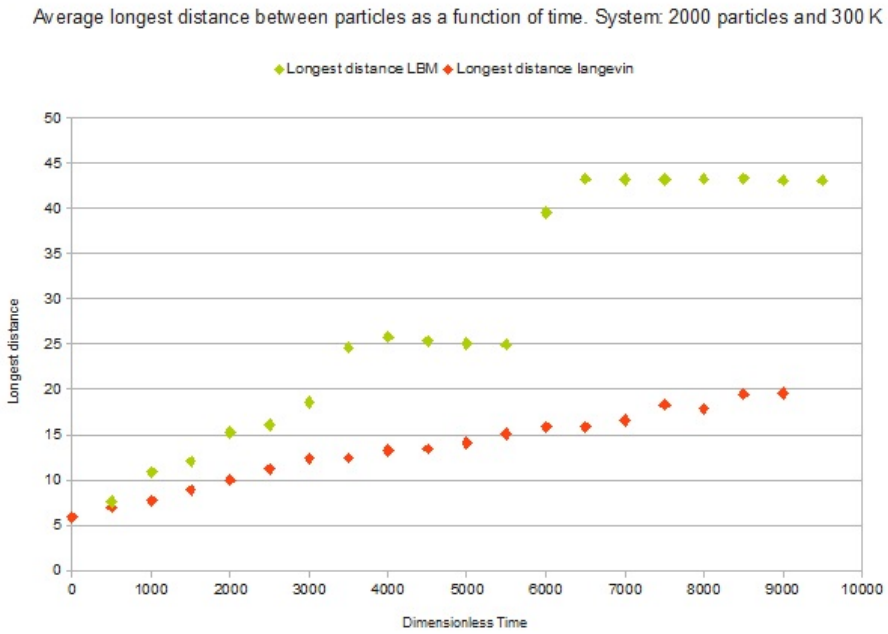

Figure 6.9: Average longest distance between particles vs time.

And also the observation that agglomeration of particles in the micrometer size range is largely affected by fluid, has first been formulated by Smoluchowski.

In Langevin simulations, hydrodynamics are not considered. Since in LBM method we are taking into account the hydrodynamics, and this also contributes in the agglomeration process according to the literature, It is reasonable to obtain faster agglomerations in LBM method than in the langevin ones. Because of the hydrodynamics, bonds could also be broken, but since we are avoiding this possibility in our model, just a positive effect in the agglomeration process is possible.

Another fact to highlight is that we obtain similar fractal dimension for both simulations 1 and 2, but the values for the LBM simulation are little bit higher at the end of the simulation with 2000 particles (order of 2) while in the 1000 particles simulation D_f between 1.5 and 2. See the figures (6.3) and (6.8).

In figures (6.10) and (6.11) one can see prediction for fractal dimension and longest distance as a function of cluster size.

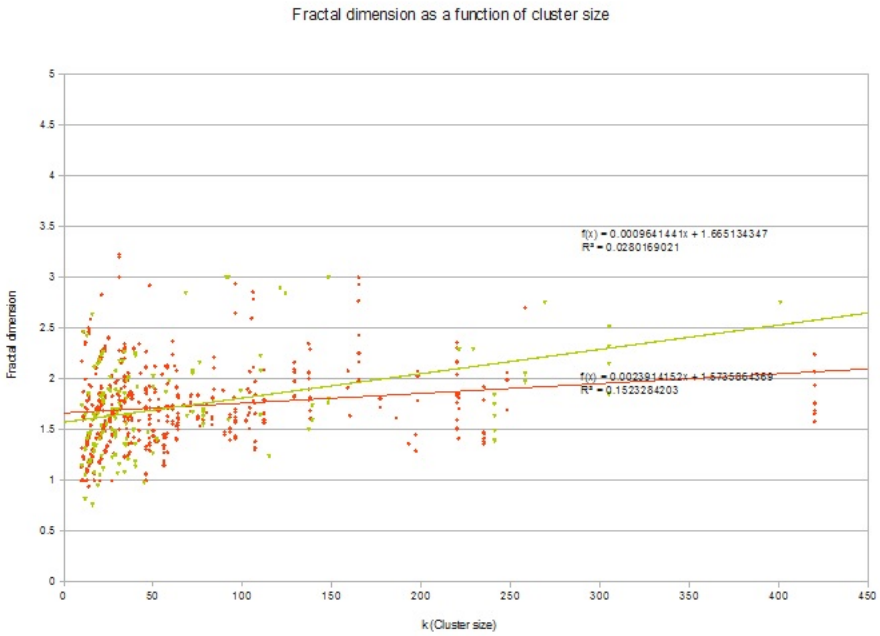

Figure 6.10: Fractal dimension vs cluster size. Green values correspond to LBM and red ones to Langevin.

6.3.3 Simulation 3 and 4: 600 K and 1000, 2000 particles

Since graphs will be similar, we try to summarize a little bit and present the results of the 2 last simulations together. The goal after checking these results will be try two answer these last 3 questions:

1. Changing the temperature, do we still obtain reasonable values according to the literature?

2. Why does the fractal dimension D_f change in LBM more compare to the Langevin Simulation?

3. Why do we get different results if we change the number of particles in the system?

Now we present the results for simulations 3 and 4.

As we can see in figures (6.14) and (6.15), we obtain slightly larger values for fractal dimension when we have 2000 particles (between 1.7 adn 2.2) while with

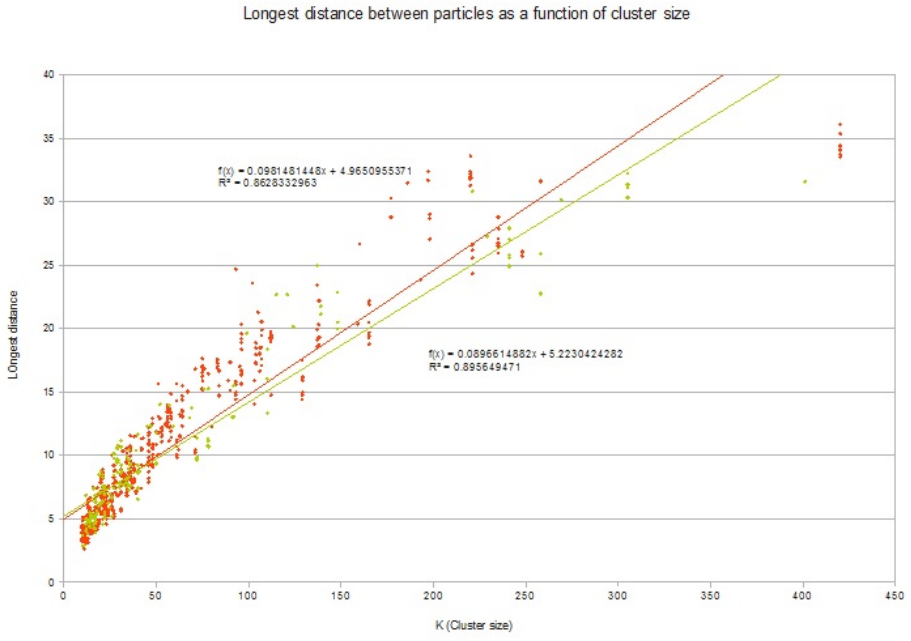

Figure 6.11: Longest distance vs cluster size. The trend line gives us an idea about values of longest distance between particles for every cluster size.

1000 particles our values vary from 1.5 to 2. Moreover we obtain larger longest distance between particles in simulation 4 than in 3. This is an expected result because the more particles we have, the bigger will become our agglomerates and therefore the larger the longest distance between monomers will be.

We take a look into the prediction for fractal dimension and longest distance. See figures (6.18)-(6.21).

Most of our agglomerates have less than 100 particles and for this range, our trend line predicts again for this temperature predicted values according to the literature Filippov et al. [2000] and Inci [2012].

Finally, we solve the questions left that are proposed at the beginning of this chapter:

1. Why does the fractal dimension D_f change in LBM more compare to the Langevin Simulation in almost all the cases?

2. Why do we get different results if we change the number of particles in the system?

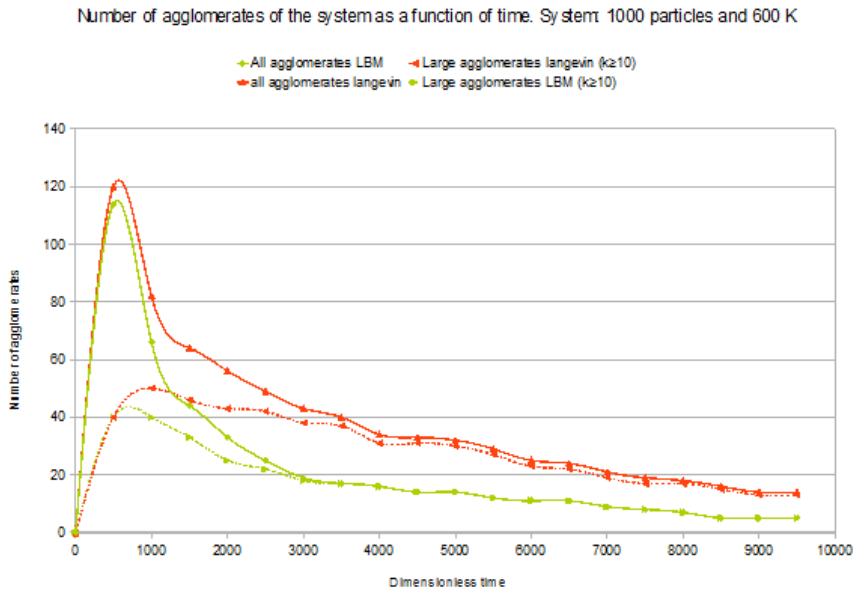

Figure 6.12: Number of agglomerates as a function of time. Every 500 time-steps number of agglomerates is printed out.

If we fix the temperature and vary the particle number (simulations 3 and 4), what do we get? The D_f values remain more or less in the same interval in the Langevin simulation. See figures (6.14) and (6.15). This can be due to the fact that Brownian motion is the same because the temperature remains constant, Soddy [1909] and Boorse and Motz [1966].

On the other hand, in LBM simulations, D_f has bigger values when we have 2000 particles than when we have 1000 particles. See figures (6.14) and (6.15). As before, the Brownian motion keeps being the same, but the effects of the surrounding fluid affects more particles and then contributes more to the agglomeration process Derksen and Eskin [2011]. This means the agglomeration procedure will be faster and we will have bigger agglomerates. In this simulations, the bigger agglomerates have bigger fractal dimension, therefore we can conclude:

- If we speak about langevin simulations, keeping the temperature constant, and increasing the number of particles, the average D_f will keep more or less the same.

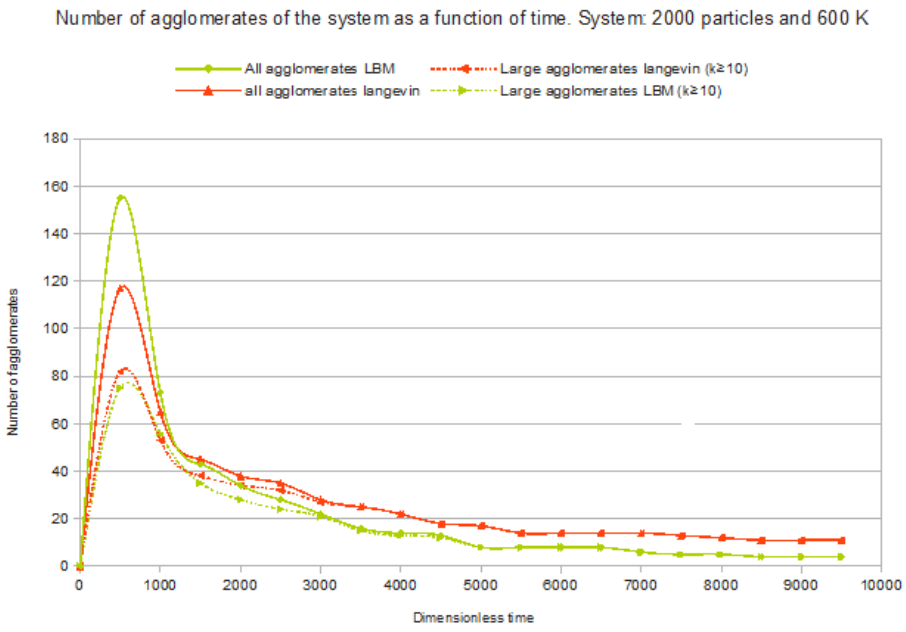

Figure 6.13: Number of agglomerates as a function of time. Every 500 time-steps number of agglomerates is printed out.

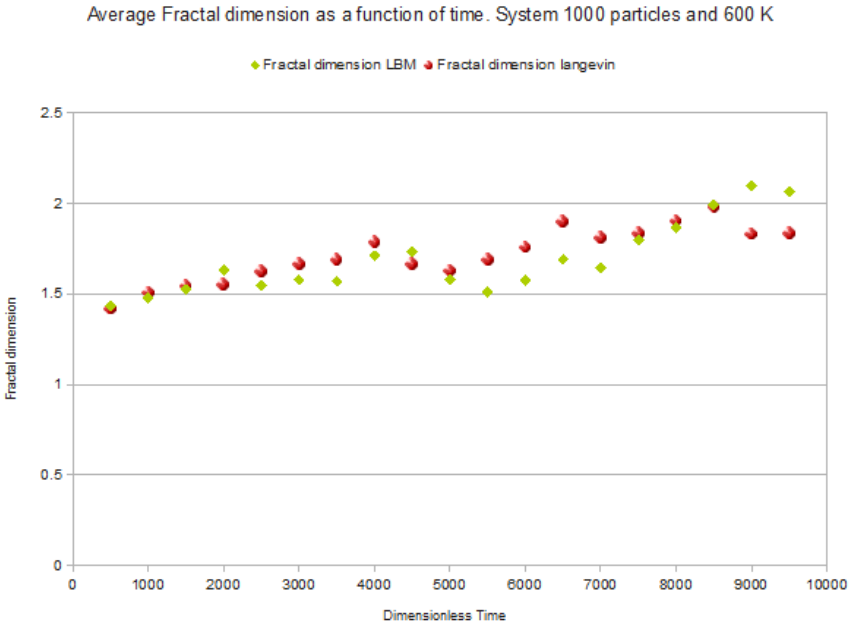

Figure 6.14: Average fractal dimension vs time.

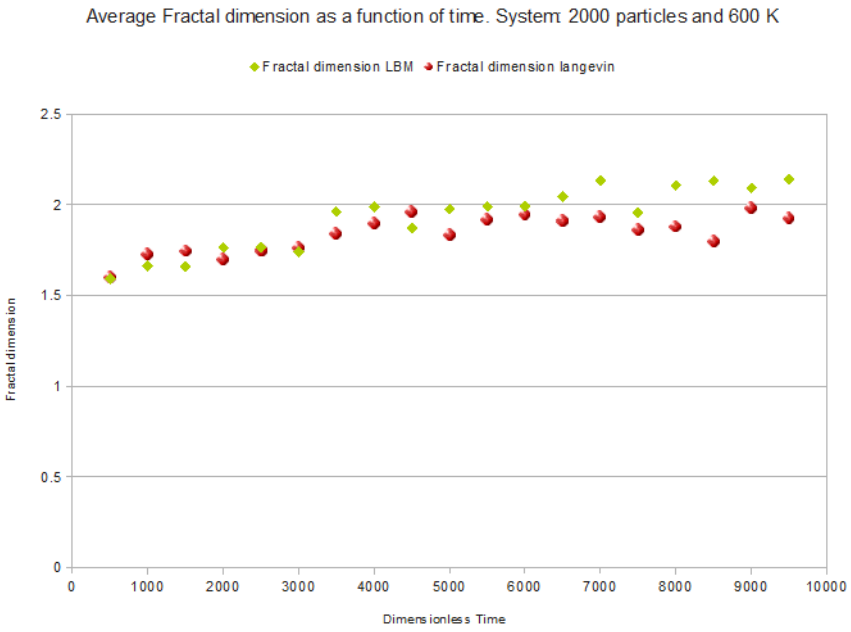

Figure 6.15: Average fractal dimension vs time.

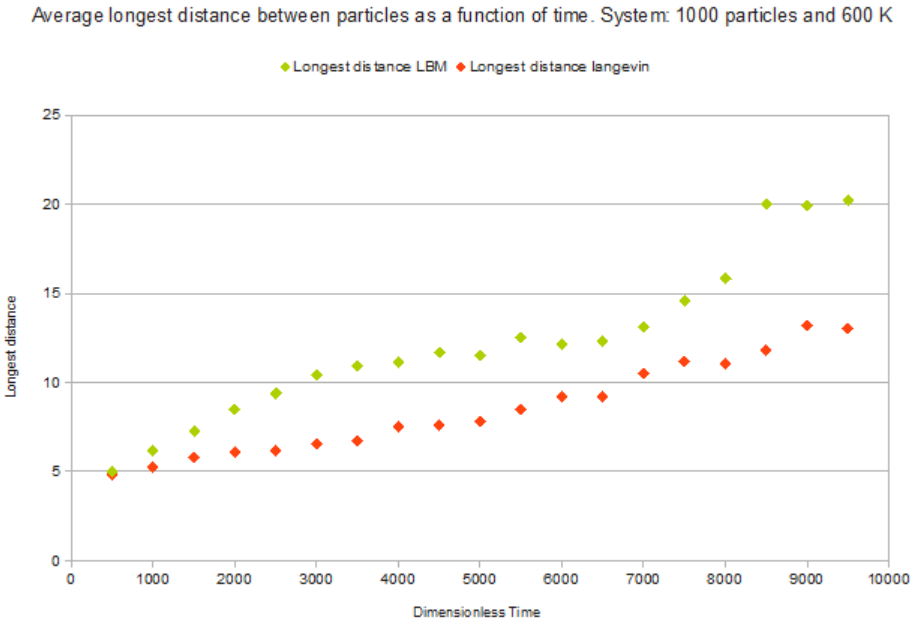

Figure 6.16: Average longest distance between particles vs time.

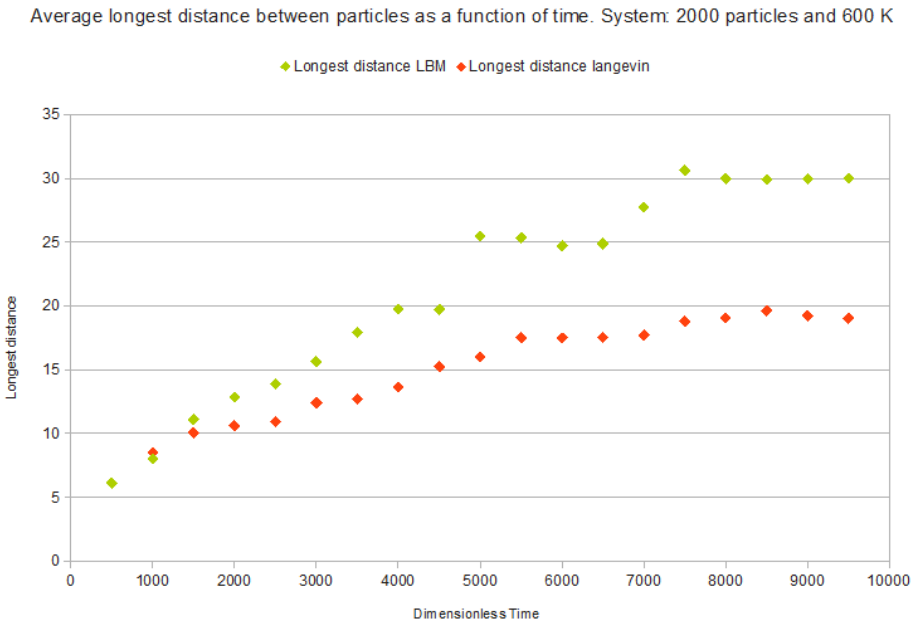

Figure 6.17: Average longest distance between particles vs time.

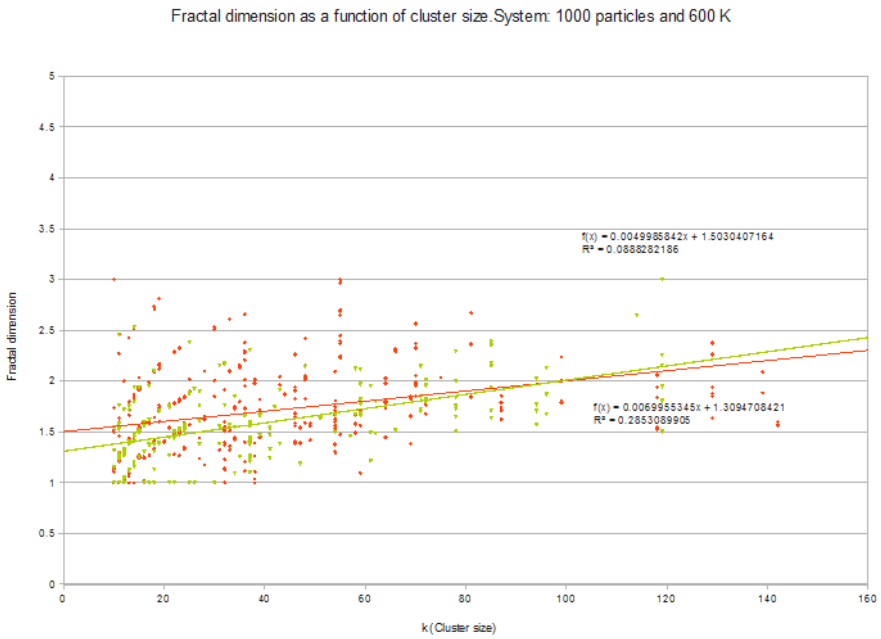

Figure 6.18: Fractal dimension vs cluster size. Green values correspond to LBM and red ones to Langevin.

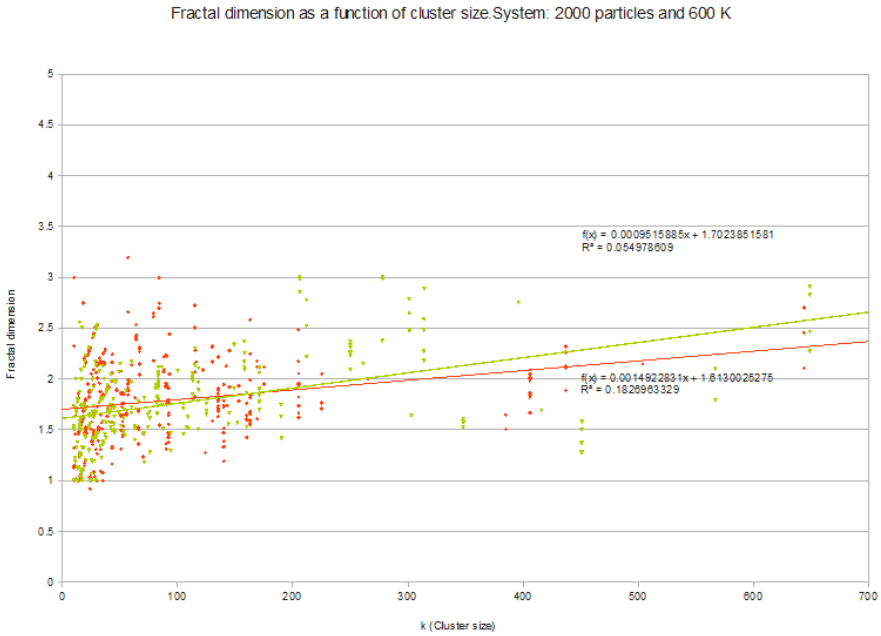

Figure 6.19: Fractal dimension vs cluster size. Green values correspond to LBM and red ones to Langevin.

- If we speak about LBM simulations, keeping the temperature constant, and increasing the number of particles, the average D_f increase due to the fluid contribution to the agglomeration process and it is higher than the D_f in the langevin simulations.

We need to ask ourselves another question. What happens if now we keep the number of particles constant and vary the temperature?. To analyze this case , see figures (6.8) and (6.15).

If we take a look on the figure (6.15), we have higher values for the LBM simulation than for the langevin simulation like we have observed in every simulation in this thesis.

Now, if we move to the figure (6.8), we can see that due to the descend of the temperature, the Brownian motion decreases as well, and therefore the contribution to create agglomerates (big ones) that are responsible for the high values of D_f. We also can notice that LBM D_f values are very sensible with the temperature increasing with less temperature (fluid contribution increases).

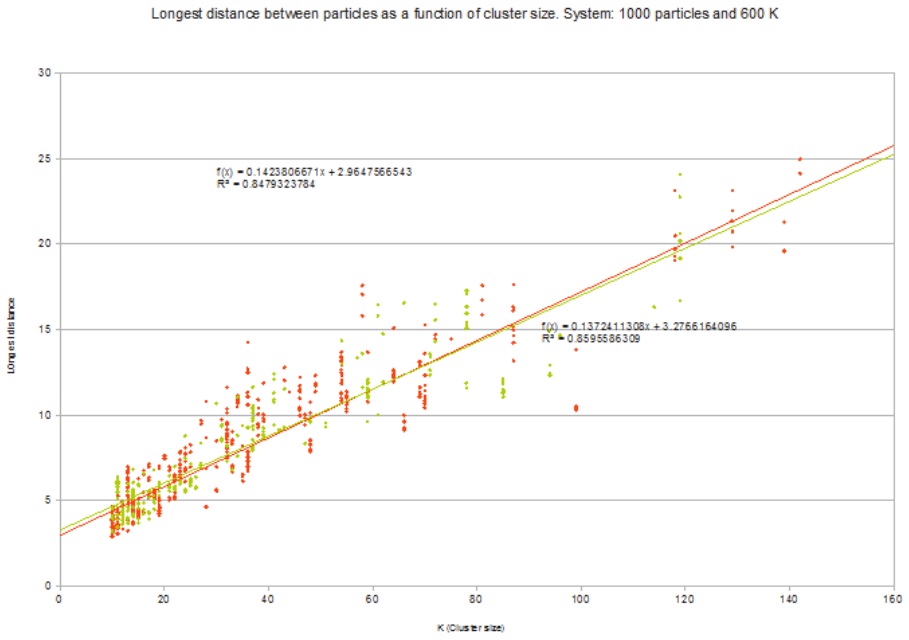

Figure 6.20: Longest distance vs cluster size. The trend line gives us an idea about values of longest distance between particles for every cluster size.

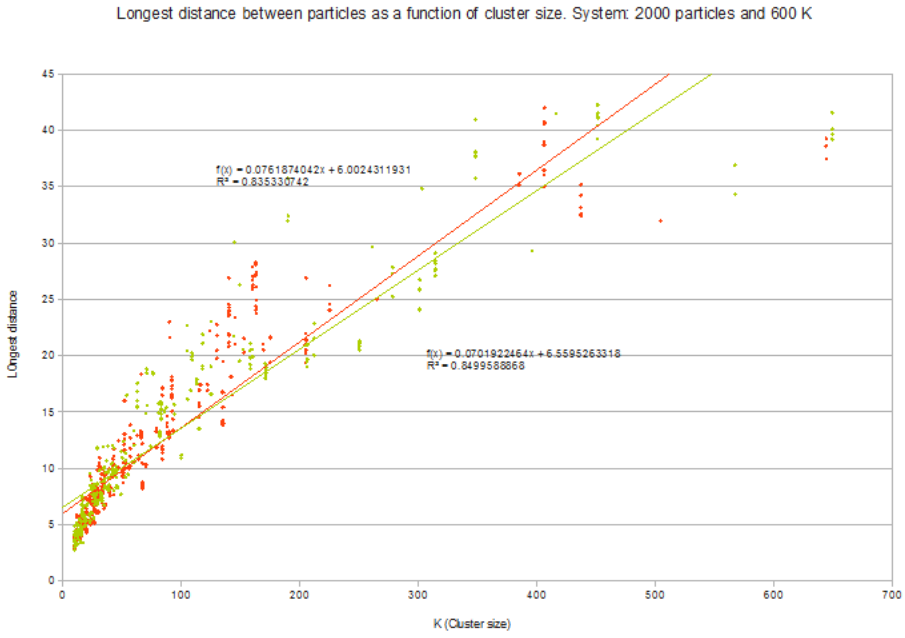

Figure 6.21: Longest distance vs cluster size. The trend line gives us an idea about values of longest distance between particles for every cluster size.

Chapter 7

Conclusions

In this study, we were able to performed simulations using 2 kind of thermostats: LB (particles suspended in a flowing fluid using MD simulations coupled to a LB solver.) and Langevin thermostat. The conclusions that we got from our study were the following:

1. From these simulations presented in details, it is found that the larger clusters have bigger fractal dimension values.

2. Taking into account the fluid flow (LB thermostat), the agglomeration process is notably accelerated in comparison to the Langevin thermostat simulations. In the same period of time, bigger clusters are formed in LB simulations.

3. If we keep constant the number of particles of our system, but we vary the temperature:

    - In the Langevin simulations: The Brownian motion will also vary and therefore the contribution to the agglomeration process (acceleration or deceleration).

4. If we keep constant the temperature and change the number of particles:

    - In LBM simulations: if we increase/decrease the number of particles, agglomeration rate increases/decreases because the surrounding fluid affects more/less particles (hydrodynamic shielding) and contributes more/less to the agglomeration process.

- In Langevin Simulations: the agglomeration process does not suffer any acceleration/deceleration.

5. If we talk about D_f as a function of cluster size, it is almost same for LBM and Langevin Simulations.

Future work can be oriented in the sense of testing different kind of surrounding fluids and see how agglomeration process and morphology of agglomerates are affected. The study could also be extended simulating more different temperatures, number of particles or type of particles.

It would be possible to take into account the possibility of breaking bonds. In this way we would be able to check if the hydrodynamics still accelerates the agglomeration process.

Appdx A

We show here the code that prints and calculates: bond lists, fractal dimension, radius of gyration and center of mass of each agglomerate.

```
# creates bond lists
     for {set b 0} {$b<$particle} {incr b} {
       set Bonds($b) [list]
     }

     for {set c 0} {$c<$particle} {incr c} {
         set bondi [part $c print bonds]
         set bondlist [lindex $bondi 0]
         set Nbonds [llength $bondlist]
         for {set d 0} {$d<$Nbonds} {incr d} {
             set bd [lindex $bondlist $d]
             set bdwith [lindex $bd 1]
             lappend Bonds($c) $bdwith
             lappend Bonds($bdwith) $c
             }
          }

#creates agglomeration lists
      set Nagg 0
      set Ag(0) [list]
      for {set f 0} {$f<$particle} {incr f} {
          set found 0
          for {set g 0} {$g<=$Nagg} {incr g} {
              if {[lsearch -exact $Ag($g) $f]!=-1} {
                 set found 1
                 }
          }
          if {$found==0} {
```

```
              set lst [list [concat $f $Bonds($f)]]
              set Ag($Nagg) [lindex $lst 0]
              set e 1
              set length [llength $Ag($Nagg)]
              while {$e<$length} {
                     set element [lindex $Ag($Nagg) $e]
                     foreach ii $Bonds($element) {
                        if {[lsearch $Ag($Nagg) $ii]==-1} {
                        lappend Ag($Nagg) $ii
                        }
                     }
                     set length [llength $Ag($Nagg)]
                     incr e
               }
              incr Nagg
              set Ag($Nagg) [list]
          }
        }

#eliminate 1-particle agglomerates
set Agg(0) [list]
set NAGG 0
for {set ii 0} {$ii<$Nagg} {incr ii} {
 set size [llength $Ag($ii)]
 if {$size>1} {
     set Agg($NAGG) $Ag($ii)
     incr NAGG
     set Agg($NAGG) [list]
     }
}

# list of sizes of agglomerates
for {set ff 0} {$ff<$NAGG} {incr ff} {
 set long($ff) [llength $Agg($ff)]
}


 #calculates the center of mass for each agglomerate

    for {set h 0} {$h<$NAGG} {incr h} {
         set center($h) [list]
```

```
         set part1 [lindex $Agg($h) 0]
         for {set jj 0} {$jj<3} {incr jj} {

           set centeri($jj) [lindex [part $part1 print folded_position] $jj]
           for {set xx 1} {$xx<$long($h)} {incr xx} {
              set partxx [lindex $Agg($h) $xx]
              set coor($jj) [lindex [part $partxx print folded_position] $jj]
              set temp_cent($jj) [expr $centeri($jj)/($xx)]
              if { [expr abs($temp_cent($jj)-$coor($jj))]<[expr $box_length/2]}
                 set centeri($jj) [expr $centeri($jj)+$coor($jj)]
                 } elseif {[expr $temp_cent($jj)-$coor($jj)]<0 &&
[expr abs($temp_cent($jj)-$coor($jj))]>[expr $box_length/2]} {
                       set centeri($jj) [expr $centeri($jj)+$coor($jj)-$box_leng
                          } elseif {[expr $temp_cent($jj)-$coor($jj)]>0 &&
[expr abs($temp_cent($jj)-$coor($jj))]>[expr $box_length/2]} {
                            set centeri($jj) [expr $centeri($jj)+$coor($jj)+$box
                          }

            }
           set centeri($jj) [expr $centeri($jj)/$long($h)]
           if {$centeri($jj)>$box_length} {
              set centeri($jj) [expr $centeri($jj)-$box_length]
              }
           if {$centeri($jj)<0} {
              set centeri($jj) [expr $centeri($jj)+$box_length]
              }
         }
         set center($h) "$centeri(0) $centeri(1) $centeri(2)"
    }



#calculates the longest distance between 2 particles in each aglomerate
  for { set p 0} {$p<$NAGG} {incr p} {
      set longest_distance($p) 0
      for {set o 0} {$o<$long($p)} {incr o } {
          for {set oo 0} {$oo<$long($p)} {incr oo} {
              set firstpt [lindex $Agg($p) $o]
              set scndpt [lindex $Agg($p) $oo]
              set distance [distance_between_2_points
[part $firstpt print folded_position] [part $scndpt print folded_position]
```

```
$box_length]
               if {$distance>$longest_distance($p)} {
                  set longest_distance($p) $distance
                  }
            }
        }
   }



#calculates the points for the Df of each agglomerate
   for {set m 0} {$m<$NAGG} {incr m} {
      set distance_list($m) [list]
      foreach k $Agg($m) {
         set dist_from_centre [distance_between_2_points $center($m)
[part $k print folded_position] $box_length]
         lappend distance_list($m) $dist_from_centre
         }

      set max_dist 0
      set l 0
      set lght [llength $distance_list($m)]
      set dist 0
      while {$l<$lght} {
          set dist [lindex $distance_list($m) $l]
          if {$dist>$max_dist} {
              set max_dist $dist
              }
          incr l

       }

      set Dflist($m) [list]
      set radius $initial_radius
      while { $radius <[expr $max_dist] } {
          set n 0
          set radius [expr $radius+$incr_step]
          foreach ll $distance_list($m) {
             if {$ll <= $radius} {
                 incr n
```

```
            }
          }
          lappend Dflist($m) "[expr $radius*2] $n"

       }
    }


         for {set kk 0} {$kk<$NAGG} {incr kk} {
              puts $file "Total n agg: $NAGG  #AgglomerateN=$kk of $long($kk)
partilces Longest distance: $longest_distance($kk)  "
              set Dflength [llength $Dflist($kk)]
              for {set mm 0} {$mm<$Dflength} {incr mm} {
                  puts $file "[lindex $Dflist($kk) $mm] "
                 }

             }
close $file

incr fileid
}

incr i
}


   # CLOSE MAIN LOOP


exit
```

References

www.espressomd.org. a. 11

www.softmatterworld.org. b. 11

A. Arnold, O. Lenz, S. Kesselheim, R. Weeber, F. Fahrenberger, D. Roehm, P. Kosovan, and C. Holm. Espresso 3.1: Molecular dynamics software for coarse-grained models. *Springer-Verlag Berlin Heidelberg*, 2013. 12

P. Asinari. *Multi-Scale Analysis of Heat and Mass Transfer in Mini/Micro-Structures*, 2001. 3

H. A. Boorse and L. Motz. *The World of the Atom*, 1, 1966. 4, 43

M. Cuendet. Molecular dynamics simulation. *EMBL*, 2008. 1

J. J. Derksen and E. Eskin. Flow-induced forces in agglomerates. *Fluid Dynamics and Materials Processing*, 7:341, 2011. 38, 43

M. Dietzel and M. Sommerfeld. Lbm simulations on agglomerate transport and deposition. *AIP. Conf. Proc*, page 796, 2010. 1

M. Dietzel and M. Martin Sommerfeld. Lattice boltzmann simulations for characterizing the behavior of agglomerates with different morphologies. *EUROMECH Colloquium*, 513, 2011. 1

B. Dünweg and A. J. C. Ladd. Lattice boltzmann simulations of soft matter systems. *Advanced in Polymer Science*, 221, 2009. 11

C. Feichtinger, N. Thürey, H. J. Schmidt, C. Binder, and K. Iglberger. *Drag Force Simulations of Particle Agglomerates with the Lattice-Boltzmann Method*, 2005. 4, 7, 8

A. V. Filippov, M. Zurita, and D. E. Rosner. Fractal-like aggregates: Relation between morphology and physical properties. *Journal of Colloid and Interace Science*, 229:261–273, 2000. 20, 35, 36, 42

S. R. Forrest and T. A. Witten. Long-range correlations in smoke-particle aggregates. *Journal Phys. A. Math. Gen.*, 12:109, 1979. 19

H. Gao and L.P. Wang. Lattice boltzmann simultaion of turbulent flow laden with finite-size particles. *7th International Conference on Multiphase Flow, ICMF Tampa*, 2010. 2

J. Hardy, Y. Pomeau, and O. Pazzis. Time evolution of a twodimensional model system. invariant states and time correlation functions. *Journal of Mathematical Physics*, 14:1746, 1973. 3

K. Iglberger. *Lattice Boltzmann Simulation of Flow around moving Particles*, 2005. 16

K. Iglberger, N. Thürey, H.J Schmid, and C. Feichtinger. Simulation of moving nano particles with the lattice boltzmann method in 3d. *Erlangen*, 2005. 16

G. Inci. *Modeling Agglomeration of Nano-Particles in Molecular Dynamics (MD) Software*, 2012. 13, 18, 38, 42

L. Isella and Y. Drossinos. Langevin agglomeration of nanoparticles interacting via a central potential. *Physical Review*, 82, 2010. 36

O. Lenz, S. Kesselheim, and G. Rempfer. Espresso 3: The lattice-boltzmann-method in espresso: Polymer diffusion. 34

T. Pham, U.D. Schiller, J. R. Prakash, and B. Dünweg. Implicit and explicit solvent models for the simulation of a single polymer chain in solution: Lattice boltzmann vs brownian dynamics. *Journal of Chemical Physics*, 131, 2009. 2

F. Pierce, C. M. Sorensen, and A. Chakrabarti. Computer simulation of diffusion-limited cluster-cluster aggregation with an epstein drag force. *Physical Review*, 74, 2006. 19

D. Röhm. *Lattice Boltzmann Simulations on GPUs*, 2011. 3

R. J. Samson, G. W. Mulholland, and J. W. Gentry. Structural analysis of soot agglomerates. *Langmuir*, 3:272–281, 1987. 18

Smoluchowski. Versuch einer mathematischen theorie der koagulationskinetik koloider lösungen. *Zeitschrift fuer Physikalisch Chemie*. 40

F. Soddy. *Annales de Chimie et de Physique*, 18:1–114, 1909. 43

S. Succi. The lattice boltzmann equation for fluid dynamics and beyond. *Oxford University Press*, 2010. 4

H. Xiaoyi and L. Li-Shi. Lattice boltzmann model for the incompressible navier-stokes equation. *Journal of Statistical Physics*, 88, 1997. 34