University of Strathclyde

Department of Chemical & Process Engineering

M.Eng Chemical & Process Engineering 18530

Computer-aided Design of Bio-inspired Nanoporous Silica Materials

---

Author:
André Crescenzo

Supervisors:
Miguel Jorge
Alessia Centi
Carlos F. Rangel

August 3, 2015

1 Summary

...

Contents

2 Acknowledgements

...

3   Introduction

Molecular Dynamics (MD) and Monte Carlo (MC) are powerful tools to simulate molecular interactions of surfactants in solvent systems, allowing a deeper understanding of their self-assembly process (Pérez-Sánchez, Gomes, & Jorge, 2013). This process results in several types of surfactant mesoscale conformations that are specially useful to design bio-inspired silica materials (Patwardhan, 2011). With the addition of silica, these structures behave as scaffolds to mesoporous or nanoporous structures that are maintained even after surfactant removal, silica oligomer polymerization and calcination (Tolbert, Firouzi, Stucky, & Chmelka, 1997). A vast range of silica materials are examples of this phenomena, such as MCM-41 as reported by Kresge and Roth (2013), SBA-15 (?, ?), MSU-V (Tanev, Liang, & Pinnavaia, 1997) and many others, in which the self-assembled structure depends on the type of surfactant, concentration of the substances involved and synthesis conditions such as temperature and pH. It should be noted that most of the experimental methods used to obtain data are based on observation and interpretation of final silica structure by using X-ray diffraction (XRD) and transmission electron microscopy (TEM). It follows that initial self-assembled conformations are predicted as a reflex of the final results and little is known about the mechanistic of this process. However, with MD simulations it is possible to observe and analyse these initial steps of self-assembly and predict, with more accuracy, properties and frameworks provided by surfactants (Tresset, 2009).

On the other hand, a major concern is that even though MD uses sophisticated software prepared to simulate systems with thousands of atoms, using all capacities of hardware available, such as high-speed multi-core processors in conjunction with GPUs designed specifically to process data from arrays (Pronk et al., 2013), they can hardly achieve long time horizons and are commonly limited to a few microseconds depending on the size of the system. For this reason, several techniques have been developed to optimize the performance of the simulations such as coarse-grain methods (Pérez-Sánchez et al., 2013). The basic idea of this technique is to fit parameters of atom groups with similar properties in a bigger "bead", which includes atomic masses and electrostatic charges lumped in approximated values. For example, given a simulation of an arbitrary surfactant with a long hydrophobic tail, it is possible to merge three or four carbon atoms (and its hydrogen atoms) in groups since they have similar hydrophobic properties, by this means reducing the number of particles in the system and speeding up the simulation.

In order to provide molecular topologies to simulations software packages several force-fields, for example MARTINI (Marrink, Risselada, Yefimov, Tieleman, & de Vries, 2007), uses Lennard-Jones potentials fitted to a range of pre-defined bead types to describe coarse-grain models. Furthermore, not only intramolecular beads are possible, but also intermolecular beads can be specified, such as multiple solvent molecules merged in a single bead or ions surround by water molecules. Previous works conducted by (Pérez-Sánchez et al., 2013) with this method recreated a model of surfactant in the presence of silica with explicit water that was successful in describing rod-like self-assembly structures detected on MCM-41 materials, demonstrating the capacities of up-scaling this type of systems. Nevertheless, solvent presence demands most of the computational resources, hence implicit solvent scheme has been the focus of many studies (Pronk et al., 2013).

Different concepts have been applied to develop a suitable model for implicit solvents; For example, Arnarez et al. (2015) developed the Dry MARTINI force-field by modifying parameters of its predecessor, in such manner that solvent interactions became incorporated in these values and then solvent beads are no longer necessary. Another method, described by Mirzoev and Lyubartsev (2013) is able to recreate an implicit solvent system from interaction potentials generated from a bottom-up approach, that means by using an all-atoms simulation to generate parameters for the coarse-grain model. It is supposed that these approximated potentials can be refined iteratively by this systematic method based on averages calculated from Monte Carlo simulations. During this process thermodynamic changes originated from solvent interaction with amphiphilic molecules or even electrostatic interactions can be incorporated in the potentials derived, therefore providing a flexible methodology to up-scale systems allowing the simulation of mesoscale self-assembly structures using coarse-graining.

The experiments presented in this work are an attempt to develop a method to upscale silica-surfactant interactions (Tolbert et al., 1997) from the atomistic model to a mesoscale model with the advent of this later coarse-grain technique. The methodology applied to reach the desired model is based on the MagiC software package (Mirzoev & Lyubartsev, 2013) that in conjunction with a MD simulation software, in this case GROMACS (Pronk et al., 2013), will provide a suitable approximation to self-assembly of amphiphilic molecules. Further explanations of the process are described in the Experimental Methods section. For the scope of this project, a bolaamphiphilic molecule called 1,12-diaminododecane (DMDD) has been chosen as surfactant because, as seen in previous research by Tanev et al. (1997), it self-assembles in multilamelar vesicles that in the presence of a silica precursor

are capable of generating a mesoporous structure with remarkable properties. In order to validate this structure formation and framework formation for silica oligomers, a coarse-grain approximation is a suitable option since amphiphilic molecules interaction with solvents can be described efficiently with potentials derived from the chosen methodology (?, ?). As a final objective at the end of this project, an implicit water coarse-grain model for DMDD and Silica will be generated and properly validated based on MD simulations and thermodynamic properties, in order to provide a satisfactory approximation of self-assembled silica-surfactant structures in mesoscale.

4   Theoretical Basis

This work is based on two molecular simulation techniques. The former, called Molecular Dynamics, accounts for integration of Newton forces in order to describe positions and velocities of particles in the system. The latter, called Monte Carlo method, is a powerful statistical tool based on stochastic inputs to measure properties from many systems, in this case molecular systems in thermodynamic equilibrium. In the following sections the mechanism behind each technique will be described, including additional explanations on how they are applied in simulation software.

4.1   Simulation Methods

4.1.1   Molecular Dynamics simulation

Molecular Dynamics is a simulation method that originates from the dynamic nature of atomic interactions. Generally, a system of atoms can be treated as a multi-particle system ruled by Newton's Law (Frenkel & Smit, 2001). Considering a system with $\mathcal{N}$ atoms, for each ith particle of the system the following differential equation will determine its dynamic behaviour:

$$m_i \frac{d^2 \vec{r}_i(t)}{dt^2} = \vec{F}_i(t) \tag{1}$$

Where $\vec{r} = (x, y, z)$ is the position vector and $F = (F_x, F_y, F_z)$ are the force components. Therefore, for this $\mathcal{N}$ particle system, it is necessary to solve $3\mathcal{N}$ differential equations in order to fully describe it analytically at any t. Since any molecular system involves an enormous number of molecules, solving these equations analytically is impracticable and simulation methods are necessary. The derivative terms need to be numerically calculated and as soon as simulation efficiency is a major concern, a suitable approximation to the second derivative of position is the Taylor's expansion, taking the x coordinate as an example, for a given Δt is:

$$x(t + \Delta t) = x(t) + \Delta t \frac{dx(t)}{dt} + \frac{1}{2!} \Delta t^2 \frac{d^2 x(t)}{dt^2} + \frac{1}{3!} \Delta t^3 \frac{d^3 x(t)}{dt^3} + \mathcal{O}(\Delta t^4) \tag{2}$$

$$x(t - \Delta t) = x(t) - \Delta t \frac{dx(t)}{dt} + \frac{1}{2!} \Delta t^2 \frac{d^2 x(t)}{dt^2} - \frac{1}{3!} \Delta t^3 \frac{d^3 x(t)}{dt^3} + \mathcal{O}(\Delta t^4) \tag{3}$$

$$\frac{d^2 x(t)}{dt^2} = \frac{x(t + \Delta t) - 2x(t) + x(t - \Delta t)}{\Delta t^2} + \mathcal{O}(\Delta t^2) \tag{4}$$

Hence, by neglecting the error of order $\mathcal{O}(\Delta t^2)$, it is possible to use equations (1) and (4) to derive the called "Verlet method" where position and velocity vectors for each ith atom on next step $(t + \Delta t)$ are calculated using previous and current step:

$$\vec{r}_i(t + \Delta t) = 2\vec{r}_i(t) - \vec{r}_i(t - \Delta t) + \frac{\Delta t^2}{m_i} \vec{F}_i(t) \tag{5}$$

$$\vec{v}_i(t) = \frac{\vec{r}_i(t + \Delta t) - \vec{r}_i(t - \Delta t)}{2\Delta t} \tag{6}$$

In order to improve accuracy of the Verlet method, another approach can be made by accounting for a new force term in the velocity. This term is calculated from the updated position vector, and this improved velocity term is used to calculate the new position in the next step, creating a more precise step cycle with more stability (Satoh, 2010). This set of equations is called "Velocity Verlet method" and it is described as:

$$\vec{r}_i(t + \Delta t) = \vec{r}_i(t) + \vec{v}_i(t)\Delta t + \frac{\vec{F}_i(t)}{2m_i} \Delta t^2 \tag{7}$$

$$\vec{v}_i(t + \Delta t) = \vec{v}_i(t) + \frac{\vec{F}_i(t + \Delta t) + \vec{F}_i(t)}{2m_i} \Delta t \tag{8}$$

As a final variation for Verlet method one can derive the "Leap frog method", in which one considers half-step when calculating the velocity term. Even though the use of this technique leads to a more stable and accurate behaviour when compared to Verlet method, it is noticeable that velocity and position are not in the same time steps, therefore it is not possible to calculate the total energy at a given t, just kinetic or potential energy separately (Frenkel & Smit, 2001). By applying first-order derivatives with half-steps, one can obtain the following equations:

$$\vec{r}_i(t + \Delta t) = \vec{r}_i(t) + \vec{v}_i(t + {}^{\Delta t}/_2)\Delta t \tag{9}$$

$$\vec{v}_i(t + {}^{\Delta t}/_2) = \vec{v}_i(t - {}^{\Delta t}/_2) + \frac{\vec{F}_i(t)}{m_i} \Delta t \tag{10}$$

The force term for each step is a key factor to define whether Molecular Dynamics are realistic or not. They are strictly related to the force-field adopted to describe molecular interactions, since they provide parameters to specify intra and intermolecular potentials from which forces can be calculated at any system configuration. Eventually, as seen in innovative methods such as the used on this project, the potential values can also be provided by tables generated specifically for each molecular interaction, explanations about this technique will be given on further sections (See Section 4.2.2). Molecular dynamics technique enables the use of thermostats and barostats to control dynamic behaviour of temperature and pressure of the system, recreating different types of thermodynamic ensemble. Hence, at equilibrated states, system properties can be obtained as a temporal average of instantaneous values and those averages can be compared to real experimental data in order to validate simulations.

4.1.2 Monte Carlo simulation

Monte Carlo is a simulation method based on the statistical probability of existence of a system at thermodynamic equilibrium. The equilibrium condition happens when the free energy reaches a minimum by calculating the energy for each particle's microscopic state. Given a system of $\mathcal{N}$ particles, temperature T and volume V (thus it can be denoted as a canonical ensemble), it is possible to assume that the Helmholtz free energy of this system is:

$$A = U - TS \tag{11}$$

where S is the entropy and U is the internal energy, meaning that not only a minimum in free energy can arise from a minimum in total energy, but also an increase in entropy of the system can contribute to this energy minimization. A system can be described by a group of coordinates, in this case for clarity, a group of distance vectors $\lambda = (\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_\mathcal{N})$ from system origin. For an arbitrary λ the probability of a single particle of the system to statistically occupy that position is described by a probability distribution function (Satoh, 2010):

$$\rho(\lambda) = \frac{\exp\left(-\dfrac{U(\lambda)}{kT}\right)}{\displaystyle\int_V \cdots \int_V \exp\left(-\dfrac{U(\lambda)}{kT}\right) d\vec{r}_1 \, d\vec{r}_2 \ldots d\vec{r}_\mathcal{N}} \tag{12}$$

If the system configuration is generated with a considerable number of microstates that satisfies this probability, then the final configuration will have a real physical meaning. But, as soon as it is almost impossible to define an analytical solution to this equation when $\mathcal{N}$ is too large, another method became necessary to use Monte Carlo simulations in molecular dynamics. The Metropolis method (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953) allowed the use of Monte Carlo technique by introducing the following concept: given 2 different microstates, the probability to change from state 1 to 2 is defined by:

$$P_{1 \mapsto 2} = \begin{cases} 1 & for \ \frac{\rho(\lambda_2)}{\rho(\lambda_1)} \geqslant 1 \\ \dfrac{\rho(\lambda_2)}{\rho(\lambda_1)} & for \ \frac{\rho(\lambda_2)}{\rho(\lambda_1)} < 1 \end{cases} \tag{13}$$

Therefore, the integral term in eq.(12) vanishes and it is clear to observe that when $U(\lambda_1) \geqslant U(\lambda_2)$ the system will certainly change to this new microstate since it has lower energy, however in the case of $U(\lambda_1) < U(\lambda_2)$ the system has a certain probability of changing to this new microstate indicating

an increase of entropy in the system. So, by applying stochastic inputs for particle displacements and acceptance values for eq.(13), with a large number of Monte Carlo steps the system will eventually reach a minimum in free energy. Moreover, when this system reaches equilibrium it is possible to calculate accurately microstate dependent properties via ensemble averages. That means with n samples of Monte Carlo steps, the average of a ξ property can be obtained by:

$$\langle \xi \rangle = \sum_{i=1}^{n} \frac{\xi_i}{n} \tag{14}$$

It is important to notice that Monte Carlo simulations do not consider dynamic properties of the system such as kinetic energy, and therefore only Molecular Dynamics can account for such things (Satoh, 2010). Finally, now that both molecular simulation methods have been introduced it is possible to proceed to an overview of both simulation software used on this project. Together they can unite the benefits of both simulation methods, in order to recreate a suitable coarse-grain model for Molecular Dynamics.

4.2 Software Description

4.2.1 About GROMACS

GROMACS (**Gro**ningen **Ma**chine for **C**hemical **S**imulations) (van der Spoel et al., 2013) is a Molecular Dynamics and energy minimization software created at the University of Groningen (the Netherlands), and currently has been developed and updated by Royal Institute of Technology (Sweden) and Uppsala University (Sweden). GROMACS is a widely used tool in the branch of computational chemistry because it is flexible and efficient. Its flexibility comes from the capacity of simulating multiscale molecular models, from atomistic to mesoscale depending on topology described by the user. The efficiency originates from the computational optimisation, by this meaning not only that GROMACS can achieve impressive simulation speed in large systems using supercomputing or clusters, but also it can run small simulations in any ordinary computer with great performance.

In order to interpret interatomic interactions, that means forces between each pair of atoms, GROMACS uses input files called Topologies in which the user can describe almost any molecule using force fields or tabulated potentials. Data given in this description includes information regarding atoms' type, mass and charge. Moreover, it includes a detailed description of each bond, angle and dihedral for the molecule. There are several Force-fields available for usage depending on the desired objective of the simulation, their role is to be a database that describes interaction potentials. Hence their applicability is suitable to almost any system from all-atoms to coarse-grain descriptions, but it is entitled to the user to choose the most appropriate to obtain a realistic model.

When dealing with simulations of complex systems, realism means more computational power expended trying to achieve it (Satoh, 2010), therefore a major concern is how to achieve longer time horizons without loosing precision. A good example is the Lennard-Jones potential for Van der Waals forces, used by OPLS-AA Force-field (Jorgensen, Maxwell, & Tirado-Rives, 1996), which is the one chosen for all-atom simulations in this project. This type of potential approximation for a given atom i in relation to atom j is defined simply by two parameters ϵ and σ as it can be seen in Eq.15:

$$\mathcal{U}_{LJ}(r_{ij}) = 4\epsilon_{ij} \left(\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right) \tag{15}$$

Where the forces used in Molecular Dynamics equations come from the gradient of this potential for each pair of particles:

$$\vec{F}_{ij} = \nabla \mathcal{U}_{LJ}(r_{ij}) = \left(\frac{\partial \mathcal{U}_{LJ}(r_{ij})}{\partial x} \hat{x} + \frac{\partial \mathcal{U}_{LJ}(r_{ij})}{\partial y} \hat{y} + \frac{\partial \mathcal{U}_{LJ}(r_{ij})}{\partial z} \hat{z} \right) \tag{16}$$

For every simulation time-step, GROMACS recalculates bonded potentials from topology, non-bonded potentials from the summation of Lennard-Jones potentials and Coulomb potentials. As shown in Figure (1) if $r_{ij} \to \infty \Rightarrow \mathcal{U}_{LJ} \to 0$ and a similar behaviour occurs for Coulomb forces. Hence, GROMACS generates a neighbour list of atoms determined by a cut-off radius for each particle, by this means avoiding having to spend a lot of computational power calculating numbers with negligible value. Moreover, this feature allows the use of periodic boundary conditions that recreate a continuous media environment for simulation boxes bigger than two times the cut-off radius, since each particle cannot interact with itself.
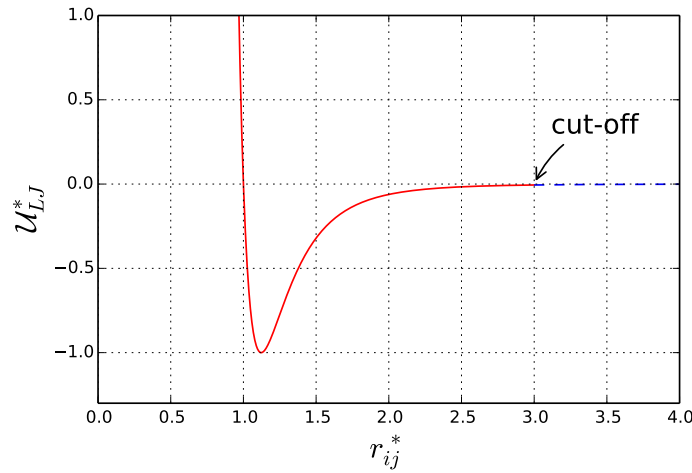
Figure 1: Example of Lennard-Jones Potential (with reduced units). For this plot $\epsilon = 1$ and $\sigma = 1$

Although speed efficiency is a concern, it is not the only preoccupation. In fact, it is also necessary to control environment variables such as temperature and pressure, in order to maintain the simulation within physical limits, which demands computational resources. GROMACS is capable of using three types of thermostat: Berendsen, v-rescaling and Nosé-Hoover and three types of barostat: Berendsen, Parrinello-Rahman and MTTK (van der Spoel et al., 2013). Furthermore, other computational expensive technique is the use of Particle-mesh Ewald (PME) summation method (?, ?) to deal with long-range electrostatic interactions, which is indispensable in ionic environments. All these tools makes GROMACS not only fast but also reliable for Molecular Dynamics simulation, moreover this project will as well demonstrate the capacity of this software to simulate realistic multi-scale molecular models.

4.2.2   About MagiC

MagiC is a software package developed by Mirzoev and Lyubartsev (2013), which is a systematic method to generate coarse-grain models from all-atomic simulations using Iterative Boltzmann Inversion (IBI) and Inverse Monte Carlo (IMC). As described in Figure (2), this method consists of a three step process where the input is the trajectory file of an equilibrated all-atoms simulation from GROMACS, and outputs are a set of tabulated potentials and a Topology for a coarse-grained model that uses these potentials to run Molecular Dynamics simulations using GROMACS.
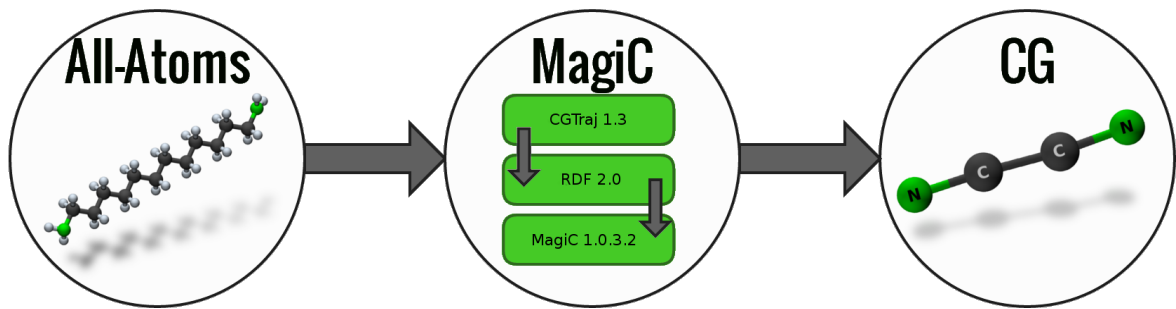

Figure 2: Graphical representation of Magic's systematic process. The central circle represent the process itself while the other two are input and output. The green boxes indicates each step inside MagiC process.

The first step uses "CGtraj 1.3" (MagiC package utility written in Fortran 90) (Mirzoev & Lyubartsev, 2014), where the trajectory of the all-atoms simulation is recalculated based on a bead mapping described by the user. Basically, it analyses the group of atoms in each bead and assigns the position of this bead as the center of mass, charge as summation of all charges and mass as

summation of atomic masses, thus giving as output a representation of the input system as if it was an "ideal" coarse-grain model. Next, this rewritten trajectory is used as input to rdf-2.0 (MagiC package utility written in Python) (Mirzoev & Lyubartsev, 2014), where several reference Radial Distributions Functions are created for each bond, angle and pairwise intermolecular interaction, as specified by the user. These reference RDFs in conjunction with the coarse-grain topology are the inputs to the MagiC kernel (MagiC package utility written in Fortran 90) (Mirzoev & Lyubartsev, 2014), which is the key step of the process, where the final output might be a suitable coarse-grain model ready for GROMACS simulation.

The kernel is where the actual Inversion process occurs, as schematised in Figure (3). Whether using IMC or IBI, a couple of millions of Metropolis Monte Carlo steps are simulated using a set of trial potentials to create several samples of averaged thermodynamics properties. Then, these averages are applied in the equations of the chosen inversion method in order to refine the trial potentials that will be used in the next inversion step.
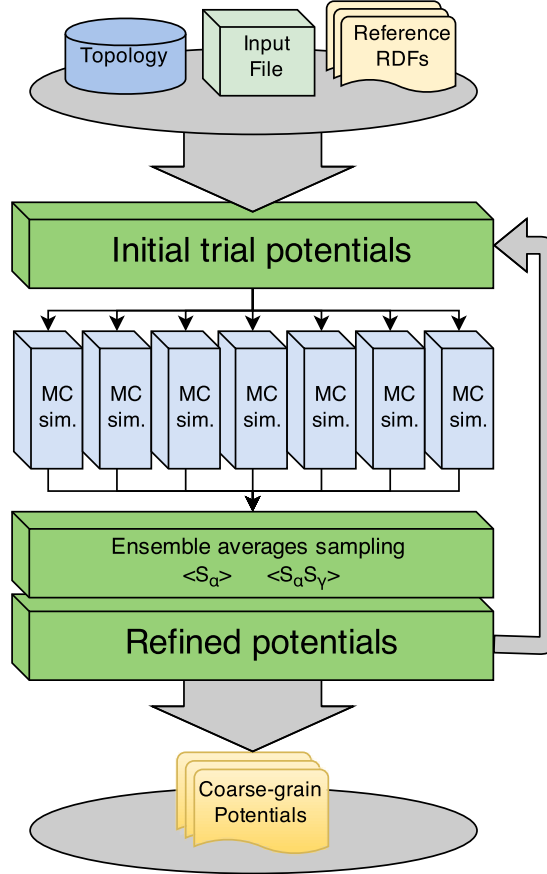

Figure 3: MagiC kernel work-flow scheme

The Iterative Boltzmann Inversion is a reasonable methodology to apply in the initial iterations, since it has fast convergence. The process derives from probability distribution function (Eq.(12)), thus for a pairwise particle interaction, the relationship between the reference RDF and Potential Mean Force (PMF) enables the use of an iterative method shown in Eq.(17) (Moore, Iacovella, & McCabe, 2014):

$$\mathcal{U}^{(i+1)}(r) = \mathcal{U}^{(i)}(r) - \eta k_B T \ln\left(\frac{\rho^{(i)}(r)}{\rho_{ref}(r)}\right) \tag{17}$$

Where η is a regularization parameter to avoid excessive variations. For a system in a canonical ensemble, temperature will be constant, hence the potential is corrected for each iteration using the $\rho_{ref}(r)$ calculated from reference RDF in comparison to $\rho^{(i)}(r)$ from RDF generated by Metropolis Monte Carlo simulation. Although this method is efficient to achieve small deviations from references, it does not guarantee that the generated potential represents one that will reproduce a behaviour similar to atomistic model because IBI does not consider cross-correlation terms between bonds and angles (Mirzoev & Lyubartsev, 2013).

Regarding the Inverse Monte Carlo process, it follows the methodology described by Lyubartsev and Laaksonen (1995). Given a Hamiltonian, that means the total energy of the system:

$$\mathcal{H}(q) = \int \mathcal{U}_\alpha \mathcal{S}_\alpha(q) \, d\alpha \tag{18}$$

For pair interactions, the degree of freedom q in Eq.(18) becomes the distance r between a pair of atoms. Therefore, for a real system the Hamiltonian would be the summation of an infinite set of $\mathcal{U}_\alpha$. However, a good approximation for numerical methods is to assume a cut-off radius r_{cut}, where potentials no longer have relevant value after this point. One can define a $\Delta r = \dfrac{r_{\mathsf{cut}}}{\mathcal{M}}$, obtaining $\mathcal{M}$ subdivisions and thus a finite set of $\mathcal{U}_\alpha$ in which $\mathcal{S}_\alpha$ is the number of pairs within interval $[r_\alpha, (r_\alpha + \Delta r)]$. Since $\mathcal{S}_\alpha$ is function of $\mathcal{U}$, one can derive the following Taylor expansion:

$$\Delta \langle \mathcal{S}_\alpha \rangle = \sum_{\phi=\alpha}^{\mathcal{M}} \frac{\partial \langle \mathcal{S}_\alpha \rangle}{\partial \mathcal{U}_\phi} \Delta \mathcal{U}_\phi + \mathcal{O}(\Delta \mathcal{U}_\phi{}^2) \tag{19}$$

$$\frac{\partial \langle \mathcal{S}_\alpha \rangle}{\partial \mathcal{U}_\phi} = \frac{\langle \mathcal{S}_\alpha \rangle \langle \mathcal{S}_\phi \rangle - \langle \mathcal{S}_\alpha \mathcal{S}_\phi \rangle}{k_B T} \tag{20}$$

For each inversion step k, Monte Carlo simulations are run using a set of potentials $\mathcal{U}_\alpha^{(k)}$, then as explained previously it is possible to calculate ensemble averages of the cross-correlation terms $\langle \mathcal{S}_\alpha \mathcal{S}_\phi \rangle^{(k)}$ and averages $\langle \mathcal{S}_\alpha \rangle^{(k)}$, $\langle \mathcal{S}_\phi \rangle^{(k)}$. At this point, the reference RDFs become useful for inverse Monte Carlo. Since this method deals with pairwise interactions the values of $\mathcal{S}_\alpha$ and $\rho(r)$ are directly proportional following the rule $\mathcal{S}_\alpha^* = \rho(r_\alpha)\Delta r$ (Mirzoev & Lyubartsev, 2013) because both measure the quantity of one type of coarse-grain bead in relation to another. Hence, by using $\Delta \langle \mathcal{S}_\alpha \rangle = \langle \mathcal{S}_\alpha \rangle^{(k)} - \mathcal{S}_\alpha^*$ in conjunction with Eq.(19) and Eq.(20) and neglecting the error $\mathcal{O}(\Delta \mathcal{U}^2)$ the potential correction for next step $\Delta \mathcal{U}_\alpha^{(k)}$ can be obtained and finally potentials are updated using the following equation:

$$\mathcal{U}_\alpha^{(k+1)} = \mathcal{U}_\alpha^{(k)} + \Delta \mathcal{U}_\alpha^{(k)} \tag{21}$$

All of the previous methods described rely on a considerably large amount of Monte Carlo simulation steps. That is due to the fact that the closer the set of potentials gets to solution, the harder it becomes to distinguish between statistical noise and a suitable potential correction. For this reason, as it can be seen in Figure (3), MagiC has a built-in capacity of running parallel Monte Carlo simulation, which enables the user to reach the order of billions of simulation steps in a feasible timespan. Moreover, there is not a unique solution for a given RDF (Moore et al., 2014) and eventually the solution can possible diverge at a certain point. Therefore, a satisfactory result depends on analysis of all outputs from inversion steps in order to guarantee convergence. Further explanations regarding functionalities and parameters for MagiC and GROMACS will be provided in the next section.

5   Methods Description

The main objectives of the experiments of this project can be characterized in three stages: the first is to develop and simulate a suitable model for the silica oligomers and the surfactant in an atomistic scale considering a high pH system. Secondly, the aim is to use MagiC software package to develop potentials for the coarse-grain model proposed using a variety of techniques to improve efficiency of the process. Lastly, the focus will be the evaluation and validation of the self-assembly structures simulated using the coarse-grain models that were generated from the previous experiments.

5.1   Atomistic Molecular Model

A initial concern for the atomistic simulations was the design of the water, surfactant and silica molecules. Beginning with the water molecule, the model used was the TIP4P water (Figure (4a)) created by (Jorgensen, Chandrasekhar, Madura, Impey, & Klein, 1983) which is a widely used water model that uses OPLS-AA forcefield (Jorgensen et al., 1996) parameters. For the DMDD surfactant, a model was created with the aid of co-workers in OPLS-AA forcefield which increases the reliability of the atomic interactions with the water models, since they use the same forcefield. This surfactant model includes 3 variations as a result of the high pH environment, because as described by (?, ?) the hydrophilic Nitrogen head receives an extra hydrogen thus in a solution it may be found the Neutral DMDD molecule (DMDDn), the Single charged DMDD molecule (DMDDs) and the Double charged

DMDD molecule (DMDDd). Since atomistic simulations need to be preferably set in a neutral charge environment, an inert anionic Bromide counter-ion was selected to equilibrate the charges.
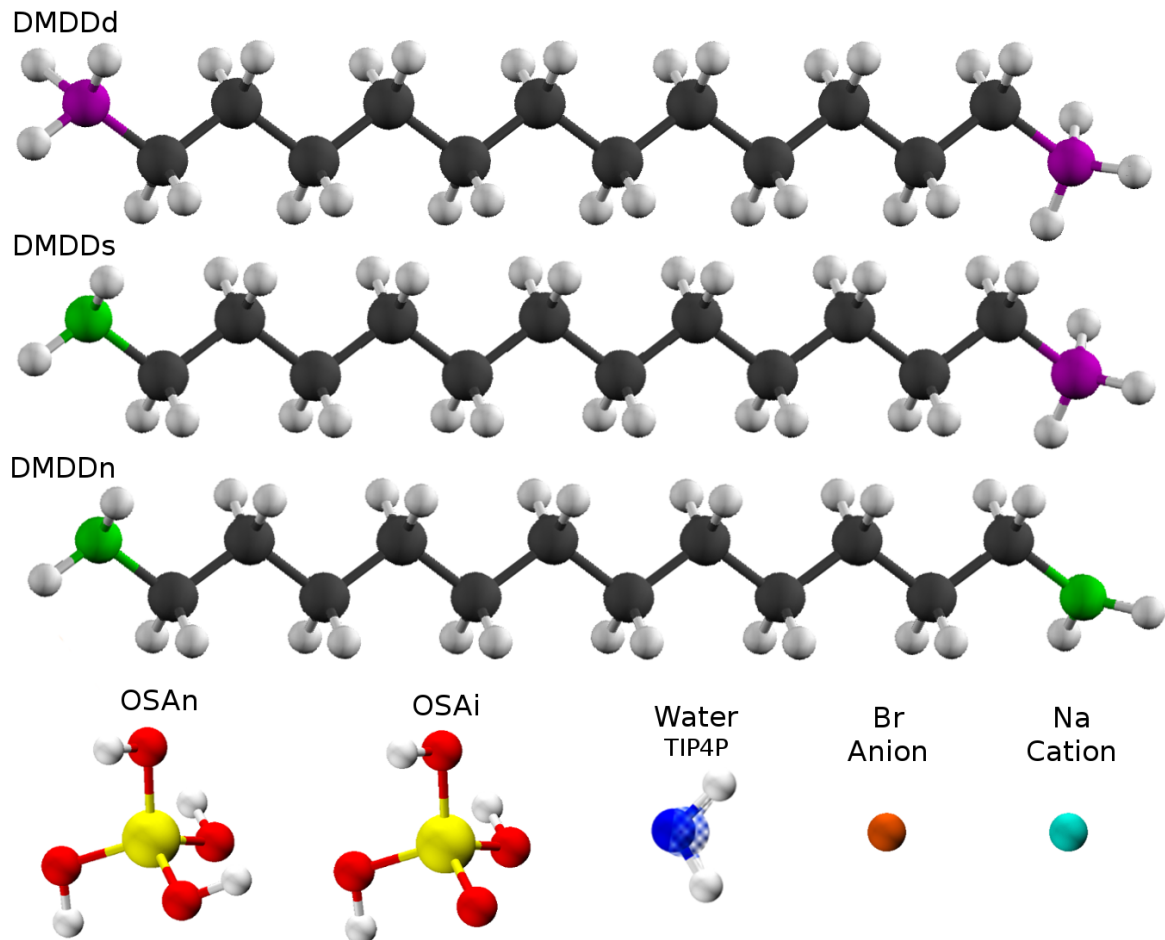

Figure 4: Graphical representation of the Atomistic models.

When it comes to the description of the silica model, it is important to notice that the in solution substances should be treated differently from pure liquid silica precursor. As described in (Tanev et al., 1997) the silica precursor tetraethoxysilane (TEOS = $Si(OCH_2CH_3)_4$) is mixed in solution in a ratio of 1:2 in relation to the DMDD surfactant, that means in an atomistic level, one silica to each surfactant head. However, due to high pH condition a substitution process that might occur in TEOS where it is forced to replace its Ethanol ramifications by hydrogen atoms forming an temporary unstable substance, the orthosilic acid (OSA). As stated by (Pérez-Sánchez et al., 2013) this unstable silica oligomer is the major representative of the silica portion in the system at early moments in solution state, which in later stages may interact with surfactant structures and polymerise around them in a variety of configurations, turning into a nanoporous and highly organized silica structure. For the scope of this project only the orthosilic acid model will be considered, since the simulations will take place during the formation of the surfactant self-assembly structure which occurs in the early moments of the solution stabilization. Therefore, as it can be seen in Figure (4c) it is possible to consider two principal representations of the orthosilic acid: the neutral molecule (OSAn) and the ionized molecule (OSAi). Similarly to Bromide, cationic Sodium counter-ion will be used to equilibrate the system charge.

5.2   Atomistic Simulation

Regarding the composition of each simulation box, the main idea was to work upon the pH value of the solution. The fact of having multiple molecular models allows the creation of simulation boxes with a varying pH, in which each one represent a different stage of deprotonation, that means the

higher the number of molecules with missing hydrogen the higher the pH that is represented on the system. It is difficult to determine exactly what is the number of molecules of each type to reach an specific pH, since the molecular types which are deprotonated should vary dynamically (Wallace & Shen, 2012). This means that even when the ratio of deprotonated and protonated molecules is in an adequate value to represent a certain pH, it might not be possible to guarantee that an specific molecule will remain at the same level of deprotonation at all times. If a molecule stays at the same state during the whole simulation it will be not representing the desired pH in the system, instead it could create an ambiguity where different regions of the box may represent a different pH value.

There are several techniques such as represented in work done by (Lee, Salsbury, & Brooks, 2004) which are capable to recreate the dynamic behaviour of the pH in solution, which probably can aid to obtain a more accurate results in when it comes to the self-assembled structures in an atomistic level. However, the adoption of such methods goes far beyond the scope of this project's main objectives, and instead in order to maintain the system with a less complex structure only high pH values were represented in simulation boxes. The types of boxes that were chosen used a single state of the molecule to represent a pseudo-high pH solution, because most of them are extreme cases that may not exist in reality but they are good approximations of real systems.

In previous works, the main subject of study was the DMDDn model, because even though it represent the most extreme case in which all polar heads are deprotonated, indicating a very high pH, it still a feasible approach to real high pH solution and furthermore provides the advantage of not presenting coulomb forces in the coarse-grain model. However, when the silica precursor is introduced in the system it might be unrealistic to assume that the OSA model will ionize all its hydrogens, since it only polymerizes in further stages after the interaction with the surfactant. Hence, the adopted model was the OSAi molecular type that when used together with DMDDn still a system with elevated pH but slightly reduced when compared to pure surfactant solution, since it has non-ionized hydrogens. For this kind of systems two approaches were adopted in order to recreate the simulation boxes. The first method was to randomly insert all molecules inside the box at adequate ratios. The second was to use simulation boxes that were generated based in DMDDn simulations that already had been equilibrated into a layer just by removing the water and putting the right amount of OSAi and then resolvating the box. The specific number of molecules of each type is described in Table (1), furthermore the implications concerning the use of both methods of box creation will be discussed in Section (**??**).

The second molecular model chosen to represent the surfactant was DMDDs. It was chosen because not only it represents a mid-state between protonation and deprotonation in the polar heads, but also it presents differentiated behaviour when compared to the previous model due to coulombic interactions within the assembled layer. Moreover, the DMDDs simulation boxes can be considered as slightly elevated pH systems when used in conjunction with ionic silica model, since it has just only one ionized head extra per surfactant. In the case of DMDDs, two types of systems were studied: one containing only DMDDs surfactant with Bromide counter-ions to equilibrate the charges, and another with DMDDs and OSAi using Sodium as counter-ion. All boxes were generated using by random placement of the molecules, to ensure whether it occurs layer self-assembly of surfactant or not in both cases, and as the same information about the composition is shown in Table (1).

Table 1: Table generated by Excel2LaTeX from sheet 'Sheet1'

Model	Depth (mm)	Width (mm)	Angle (°)	Factor 1	Factor 2
111	1,00	1,50	10[a]	357.921	532.289
112	1,00	1,50	15	382.379	567.234
113	1,00	1,50	20	383.863	569.600
121	1,00	2,00	10	398.199	590.473
122	1,00	2,00	15	486.306	710.483
123	1,00	2,00	20	430.330	636.471
			Sum:	12.138.966	17.932.100

[a] This should be cited with a Mirzoev and Lyubartsev (2013)
[b] And this is another note

It is noticeable that most atomistic boxes have a different concentration. This fact derives from the idea of providing flexibility to the coarse grained model, because they will be used as references to the coarse-grain model. The analysis of the results from previous experiences (?, ?), it is possible to observe that the coarse-grain model can represent and adapt to different apparent concentrations,

this fact also means that the coarse-grain potentials obtained from these atomistic simulation boxes contains a certain degree of similarity between them, at least when it refers to the position of the peaks and valleys. Therefore, having multiple similar concentrations within a short range can only benefit the final result obtained since flexibility is a desirable property. Another important point highlight is the box size. For all cases we have two ranges of size: the first is a small box with side bigger than the DMDD molecule (about 16 Å) used to obtain initial short range potentials; and a big size box with side at least over 40 Å to obtain long range potentials. A more detailed explanation about the difference of long range and short range potentials will be given in Section (**??**)

The simulation boxes generated for all experiments were created using "genbox" (GROMACS package utility) following a standard procedure in order to ensure the consistency of the experiments, moreover all the atomistic simulations that were used as reference for coarse-graining process were run in NPT ensemble. In order to reach this NPT condition, all boxes received an energy minimization step in order to avoid system blow-up due to possible extreme potential energy spots created during box generation. That is because during random displacement of molecules some of them could overlap others and then cause a destabilization of the system. Afterwards, an equilibration step was necessary to reach the desired initial temperature and pressure conditions for the MD simulation. During equilibration, temperature coupling was kept at 298 K using a v-rescaling thermostat (?, ?) with time constant of 0.01 ps and pressure coupling was kept at 1 bar using a Berendsen barostat (?, ?) with time constant of 0.5 ps. Every equilibration simulation was run for 200 ps with time step of 0.5 fs using leap-frog algorithm.

When it comes to the final simulation run, even though it have different structural parameters for each box content, all of them followed the same ensemble with temperature kept at 298 K by a Nosé-Hoover thermostat (?, ?) (?, ?) and pressure kept at 1 bar by a Parrinello-Rahman barostat (?, ?). Moreover, simulation time was also standardized. Every simulation lasted 150 ns with a time step of 1 fs using leap-frog algorithm. But for the sake of experiments using MagiC, the first 50 ns of the simulations were disregarded since during this time the system was still in equilibration, and therefore that would affect reference RDF values.

5.3   Coarse-grained molecular Model

6   Results and Discussion

7   Conclusion

8   Nomenclature

k_B Boltzmann's Constant
C_n Molar concentration of species n (mM)

9   References

Arnarez, C., Uusitalo, J. J., Masman, M. F., Ingólfsson, H. I., de Jong, D. H., Melo, M. N., . . . Marrink, S. J. (2015). Dry martini, a coarse-grained force field for lipid membrane simulations with implicit solvent. *Journal of Chemical Theory and Computation*, *11*(1), 260-275. Retrieved from `http://dx.doi.org/10.1021/ct500477k` doi: 10.1021/ct500477k

Frenkel, D., & Smit, B. (2001). *Understanding molecular simulatio: From algorithms to applications.* Elsevier Science.

Jorgensen, W. L., Chandrasekhar, J., Madura, J. D., Impey, R. W., & Klein, M. L. (1983). Comparison of simple potential functions for simulating liquid water. *The Journal of Chemical Physics*, *79*(2), 926-935. Retrieved from `http://scitation.aip.org/content/aip/journal/jcp/79/2/10.1063/1.445869` doi: http://dx.doi.org/10.1063/1.445869

Jorgensen, W. L., Maxwell, D. S., & Tirado-Rives, J. (1996). Development and testing of the opls all-atom force field on conformational energetics and properties of organic liquids. *Journal of the American Chemical Society*, *118*(45), 11225-11236.

Kresge, C. T., & Roth, W. J. (2013). The discovery of mesoporous molecular sieves from the twenty year perspective. *Chem. Soc. Rev.*, *42*, 3663-3670. Retrieved from `http://dx.doi.org/10.1039/C3CS60016E` doi: 10.1039/C3CS60016E

Lee, M. S., Salsbury, F. R., & Brooks, C. L. (2004). Constant-ph molecular dynamics using continuous titration coordinates. *Proteins: Structure, Function, and Bioinformatics*, *56*(4), 738–752. Retrieved from `http://dx.doi.org/10.1002/prot.20128` doi: 10.1002/prot.20128

Lyubartsev, A. P., & Laaksonen, A. (1995, Oct). Calculation of effective interaction potentials from radial distribution functions: A reverse monte carlo approach. *Phys. Rev. E*, *52*, 3730–3737. Retrieved from `http://link.aps.org/doi/10.1103/PhysRevE.52.3730` doi: 10.1103/PhysRevE.52.3730

Marrink, S. J., Risselada, H. J., Yefimov, S., Tieleman, D. P., & de Vries, A. H. (2007). The martini force field: Coarse grained model for biomolecular simulations. *The Journal of Physical Chemistry B*, *111*(27), 7812-7824. Retrieved from `http://dx.doi.org/10.1021/jp071097f` (PMID: 17569554) doi: 10.1021/jp071097f

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, *21*(6), 1087-1092. Retrieved from `http://scitation.aip.org/content/aip/journal/jcp/21/6/10.1063/1.1699114` doi: http://dx.doi.org/10.1063/1.1699114

Mirzoev, A., & Lyubartsev, A. P. (2013). Magic: Software package for multiscale modeling. *Journal of Chemical Theory and Computation*, *9*(3), 1512-1520. Retrieved from `http://dx.doi.org/10.1021/ct301019v` doi: 10.1021/ct301019v

Mirzoev, A., & Lyubartsev, A. P. (2014). Magic: User manual. version 1.0.3 [Computer software manual].

Moore, T. C., Iacovella, C. R., & McCabe, C. (2014). Derivation of coarse-grained potentials via multistate iterative boltzmann inversion. *The Journal of Chemical Physics*, *140*(22), -. Retrieved from `http://scitation.aip.org/content/aip/journal/jcp/140/22/10.1063/1.4880555` doi: http://dx.doi.org/10.1063/1.4880555

Patwardhan, S. V. (2011). Biomimetic and bioinspired silica: recent developments and applications. *Chem. Commun.*, *47*, 7567-7582. Retrieved from `http://dx.doi.org/10.1039/C0CC05648K` doi: 10.1039/C0CC05648K

Pronk, S., Páll, S., Schulz, R., Larsson, P., Bjelkmar, P., Apostolov, R., ... Lindahl, E. (2013). Gromacs 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics*, *29*(7), 845-854. Retrieved from `http://bioinformatics.oxfordjournals.org/content/29/7/845.abstract` doi: 10.1093/bioinformatics/btt055

Pérez-Sánchez, G., Gomes, J. R. B., & Jorge, M. (2013). Modeling self-assembly of silica/surfactant mesostructures in the templated synthesis of nanoporous solids. *Langmuir*, *29*(7), 2387-2396. Retrieved from `http://dx.doi.org/10.1021/la3046274` (PMID: 23343439) doi: 10.1021/la3046274

Satoh, A. (2010). *Introduction to practice of molecular simulation: Molecular dynamics, monte carlo, brownian dynamics, lattice boltzmann and dissipative particle dynamics*. Elsevier Science.

Tanev, P. T., Liang, Y., & Pinnavaia, T. J. (1997). Assembly of mesoporous lamellar silicas with hierarchical particle architectures. *Journal of the American Chemical Society*, *119*(37), 8616-8624. Retrieved from `http://dx.doi.org/10.1021/ja970228v` doi: 10.1021/ja970228v

Tolbert, S. H., Firouzi, A., Stucky, G. D., & Chmelka, B. F. (1997). Magnetic field alignment of ordered silicate-surfactant composites and mesoporous silica. *Science*, *278*(5336), 264-268. Retrieved from `http://www.jstor.org/stable/2894713` doi: 10.2307/2894713

Tresset, G. (2009). The multiple faces of self-assembled lipidic systems. *PMC Biophysics*, *2*(1). Retrieved from `http://dx.doi.org/10.1186/1757-5036-2-3` doi: 10.1186/1757-5036-2-3

van der Spoel, D., Lindahl, E., Hess, B., van Buuren, A. R., Apol, E., Meulenhoff, P. J., ... Berendsen, H. J. C. (2013). Gromacs user manual version 4.6.5 [Computer software manual].

Wallace, J. A., & Shen, J. K. (2012). Charge-leveling and proper treatment of long-range electrostatics in all-atom molecular dynamics at constant ph. *The Journal of Chemical Physics*, *137*(18), -. Retrieved from `http://scitation.aip.org/content/aip/journal/jcp/137/18/10.1063/1.4766352` doi: http://dx.doi.org/10.1063/1.4766352

10 Appendix