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

Supervisor:
Miguel Jorge

April 10, 2015

1 Summary

Contents

2 Acknowledgements

3   Introduction

Molecular Dynamics (MD) and Monte Carlo (MC) have been powerful tools to simulate molecular interactions of surfactants in solvent systems, allowing a deeper understanding of self-assembly process of this sort of material (Need & Someone, 2025). Therefore, these structural conformations are specially useful to design bio-inspired silica materials (Patwardhan, 2011) since with the addition of silica they remain behaving 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), HR (Jin et al., 2008), MSU-V (Tanev, Liang, & Pinnavaia, 1997) and many others, in which the self-assembly structure depend on the type of surfactant and concentration of the substances involved. Moreover, 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), thus initial self-assembled conformations are predicted as a reflex of the final results and little is known about the mechanistic of this process. However, by using MD simulations it is possible to observe and analyse these initial steps of self-assembly and further predict, with more accuracy, properties and framework provided by surfactant (Tresset, 2009) to generate silica structures.

A major concern is that even though MD uses sophisticated software prepared to simulate enormous 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 hardly can achieve long time horizons and are commonly limited to a few μs depending on the size of the system. For this reason, several techniques have been develop to optimize the performance of the simulations such as coarse-grain models (Need & Someone, 2025). Briefly, the idea behind using this technique is to fit parameters of heavy atoms groups with similar properties in a bigger "bead", which includes atomic masses and electrostatic charges lumped in approximated values. For example, by merging carbon heavy atoms, that means carbon atom and its hydrogen atoms, at a lipid tail in groups of three or four since they have similar hydrophobic properties.

In order to provide topologies to GROMACS simulations several force-fields, for exemple MARTINI (Marrink, Risselada, Yefimov, Tieleman, & de Vries, 2007), use 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, Gomes, & Jorge, 2013) with this method recreated a model of surfactant in 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 focus of many studies (Pronk et al., 2013).

Different concepts has been applied to develop a suitable model for implicit solvents, studies conducted by Arnarez et al. (2015) developed the Dry MARTITNI 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 & Lyubartsev, 2013) is able to recreate a 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 thermodynamic changes on the system, originated from solvent interaction with amphiphilic molecules, are incorporated on the approximated potentials. Therefore, changes in parameters as concentration may not affect coarse-grained model performance on recreating self-assembly structures (Mirzoev & Lyubartsev, 2014).

The work presented on following experiments are an attempt to create a cross-link method to upscale silica crystal-liquid phase 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 MD simulation software, in this case GROMACS (Pronk et al., 2013), will provide a suitable approximation to self-assembly of amphiphilic molecules in presence of silica. Further explanations of the process are described on Experimental Methods section. For the scope of this project, a bolaamphiphilic molecule 1,12-diaminododecane (DMDD) has been chosen as surfactant because that as seen in previous research by Tanev et al. (1997) it self-assembly in multilamelar vesicles that in presence of a silica precursor is capable of generating a mesoporous structure with remarkable properties. In order to validate this structure formation and further framework formation for silica oligomers, a coarse-grain approximation is suitable option since amphiphilic molecules interaction with

solvents can be described efficiently with tabulated Lennard-Jones potentials interactions. As a final objective at the end of this project, a implicit water coarse-grain model for DMDD will be generated and properly validated based on MD simulations and thermodynamic properties,in order to provide a good approximation to interactions with silica oligomers in a mesoscale model.

4 Experimental Methods

4.1 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 of many systems, in this case thermodynamic equilibrium of molecular systems. Following sections will describe the mechanistic behind each technique and further explain how they are used in simulation software.

Molecular Dynamics simulation

Molecular Dynamics is a simulation method that originates from 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 follow differential equation:

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

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 derivatives 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 Eqn.(1) and Eqn.(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 on Verlet method, another approach can be made by accounting 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 preciser step cycle with more stability(Satoh, 2010). This set of equations are called "Velocity Verlet method" and they are 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 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 describe Lennard-Jones potentials in 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. Molecular dynamics technique enable 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.

Monte Carlo simulation

Monte Carlo is a simulation method based on statistical probability of existence of a system at thermodynamic equilibrium. This happens when 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 free Helmholtz energy of the 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 to give a simplified idea assume 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 using a probability distribution function (Satoh, 2010):

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

Whether the system configuration is generated with considerable number of microstates that satisfies this probability, the final configuration would have a real physical meaning. But, as soon as it is almost impossible to define an analytical solution to this equation if $\mathcal{N}$ 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 such things(Satoh, 2010). Finally, now that both molecular simulation methods had been properly 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.

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 and Uppsala University (Sweden). GROMACS is a widely used tool in the branch of computational chemistry because it is flexible and efficient. Its flexibly 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 system using supercomputing or clusters, but also it can run small simulations in any ordinary computers with great performance.

In order to interpret interatomic interactions, that means forces between each pair of atoms, GROMACS uses pre-defined parameters called Topologies in which the user can describe almost any molecule using force fields or tabled potentials. Data given in this description includes information regarding atoms' type, mass and charge. Furthermore, it includes detailed description of each 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 interactions potential, hence their applicability is suitable to almost any system from all-atoms to coarse-grain descriptions it is just necessary for the user to describe a realistic model.

When dealing of 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-atomic 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:

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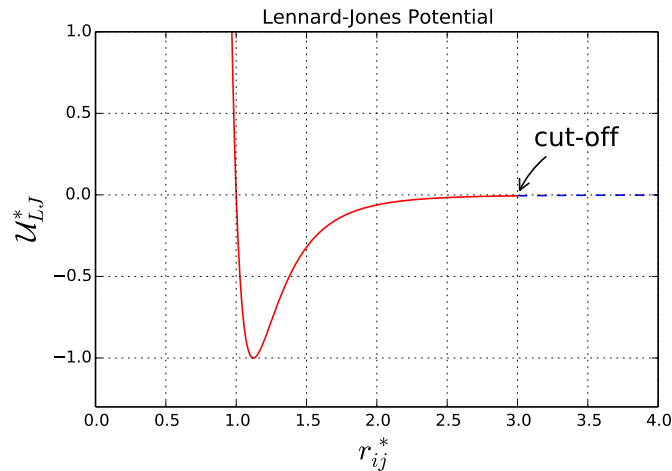

Figure 1: Example of Lennard-Jones Potential(with reduced units)

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

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

For every simulation time-step, GROMACS recalculates bonded potentials from Topology, and non-bonded potentials from the summation of Lennard-Jones and short-range Coulomb potentials. As shown in Fig.(1) whether $r_{ij} \to \infty \Rightarrow \mathcal{U}_{LJ} \to 0$ 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 recreates a continuous media environment for simulation boxes bigger than two times the cut-off radius, since particles can not interact with itself.

Although speed efficiency is a concern it is not only the unique preoccupation, it is also necessary to control environment variables such as temperature and pressure, in order to maintain the simulation within physical limits and that demands computational resources. GROMACS is capable of using three types of thermostat which are 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, and for this reason the aim of this project is to demonstrate the capacity of this software to simulate realistic multi-scale molecular models.

About MagiC

MagiC is 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 Fig.(2) method consists in 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 tabled potentials and a Topology for a coarse-grained model that uses these potentials to run Molecular Dynamics simulations using GROMACS.
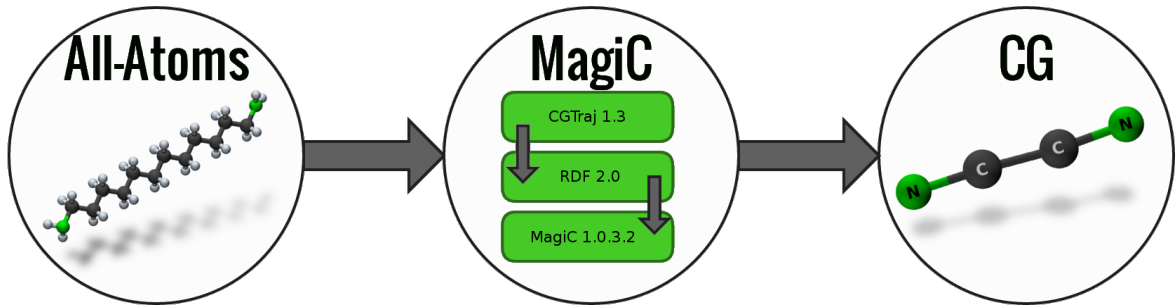

Figure 2: MagiC

The first step uses "CGtraj 1.3" (MagiC package utility written in Fortran 90) (?, ?), 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 assign 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) (?, ?), where several reference Radial Distributions Functions are created for each bond, angle and pairwise intermolecular interaction, as specified by 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) (?, ?), which is the key step of the process, where the final output might be a suitable coarse-grain model ready for GROMACS simulation.

In the kernel is where the actual inversion precesses occurs, as schematised in Fig.(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 on the next inversion step.

Figure 3: MagiC kernel work-flow scheme

The Iterative Boltzmann Inversion is a reasonable methodology to apply in the initials 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 regulation parameter to avoid excessive variations. For a system in a canonical ensemble, temperature will be constant, hence the 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 deviance 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 Inversion 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 Carlos 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 becomes useful for inverse Monte Carlo. Since this method deals with pairwise interactions the value 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.(21) 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:

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

Both of the previous methods described rely on a considerably large mount 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 his reason, MagiC has a built-in capacity of running parallel Monte Carlo simulation as it an be see on Fig.(3), which which enable the user to reach the order of billions of simulation steps in a feasible timespan. Moreover, there is not an unique solution for a given RDF (Moore et al., 2014) and eventually solution can possible diverge at a certain point. Therefore, a satisfactory result depend on analysis of all outputs from inverse steps in order to guarantee convergence. Further explanations regrading functionalities and parameters for MagiC and GROMACS will be provided the next section.

4.2   Methods Description

The experimental process of this project is divided in 3 main parts: Firstly, develop a efficient coarse-grain model and validate its use. Secondly, analyse the relationship between concentration of all-atoms simulation and concentration of coarse-grained simulation. Then finally, create and reproduce a Silica surfactant mesoscale system.

In order to maintain standard parameters for all GROMACS simulations, all-atoms simulations that were used for coarse-grained reference were run in NPT ensemble. All simulation boxes were created using genbox (GROMACS package utility), first by inserting the desired number of surfactant molecules and then adding the solvent at the adequate ratio. The molecules were designed based on OPLS-AA forcefield (Jorgensen et al., 1996) parameters, therefore the surfactant DMDD was created with aid of co-workers and water type was TIP4P (Jorgensen, Chandrasekhar, Madura, Impey, & Klein, 1983). Moreover, no initial configuration was determined, such as a pre self-assembled system, to ensure stochastic nature of the system.

Initially, all boxes received a energy minimization step in order to avoid system blow-up deal to possible extreme potential energy spots created during box generation, because due to random displacement of molecules some of them could overlap others and then cause a destabilization of the system. After, an equilibration step was necessary to reach desired initial temperature and pressure conditions for the MD simulation. During equilibration temperature coupling was kept as 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.

4.3   Experiment 1

–Objective

Figure 4: A) Schematic of M1 molecule split plan, Carbon atoms represented in black, Nitrogen in green and Hydrogen in gray. B)Final coarse-grain model for M1. This model aggregated the three most polar heavy atoms on the tips and four apolar heavy atoms at the center. In order to avoid electrostatic interactions on the IMC process and further simulation the heavy atoms groups (heavy atom + hydrogen) were kept thus final charge remain unchanged in each bead

–Model 1 vs Model 2
–gromacs Simulation Set-up
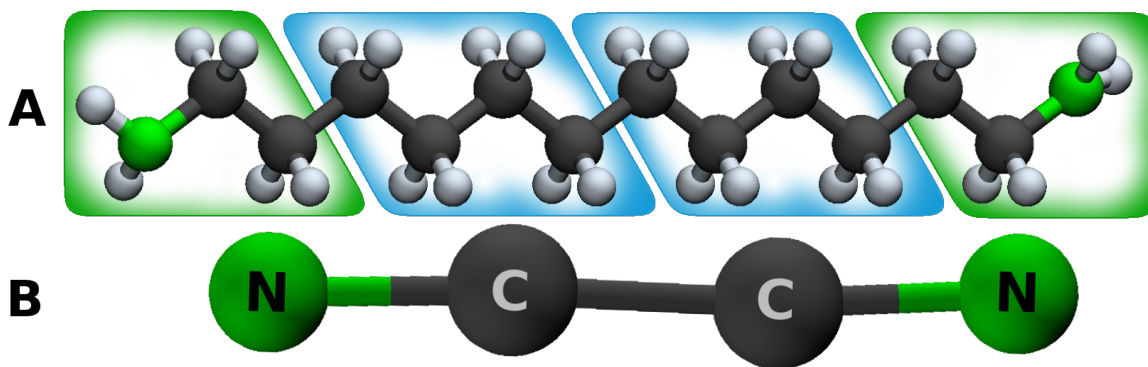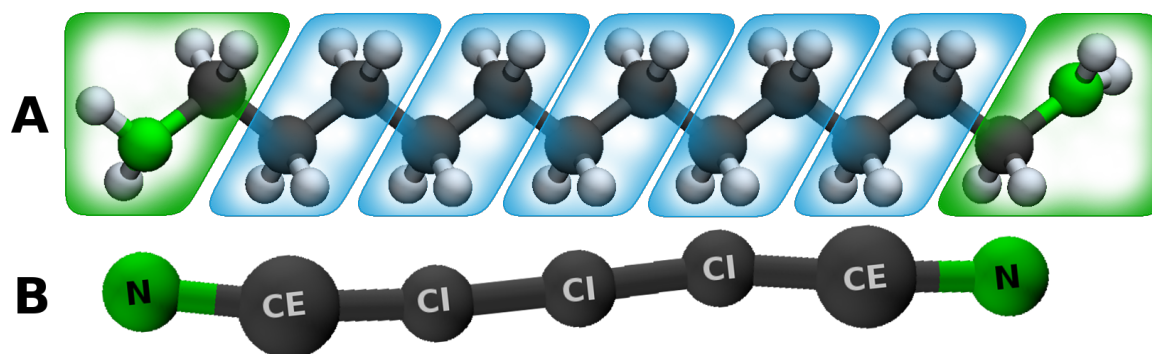–MagiC CG M1 and M2 topology generation+Ref RDFs

Figure 5: A) Schematic of M2 molecule split plan, Carbon atoms represented in black, Nitrogen in green and Hydrogen in gray. B)Final coarse-grain model for M2. Heavy atom were grouped pairwise in such a manner that amphiphilic parts were kept as distinct as possible. It is important to notice the necessity of two carbon bead types CE and CI

–IMC process
–gromacs Set-up Reproduction Tests

4.4   Experiment 2

–Objective
–gromacs Set-up Up-scaling
–gromacs Set-up Self-assembly Model X(best)
—-AA Low, CG Low
——1k(?) up-scale @ low
——1k(?) up-scale @ high
—-AA Ideal, CG ideal
——1k(?) up-scale @ low
——1k(?) up-scale @ ideal
——1k(?) up-scale @ high
—-AA High, CG High
——1k(?) up-scale @ high
——1k(?) up-scale @ high

4.5   Experiment 3

–Silica addition
—-MagiC CG silica topology
—-gromacs Set-up Self-assembly Model X + Si
——1000, 10000 (Maybe more?)

5   Results

Experiment 1
–Model 1 vs Model 2
–gromacs Simulation
–IMC process Convergence
–gromacs Reproduction Tests RDFs and Properties
Experiment 2
–Self-assembly vs concentration Model X
—-Ideal con validation: Low, Ideal, High con comparison
——Ideal @ Low -> validate
——Ideal @ Ideal -> evaluate
——Ideal @ High -> validate
—-Extremes evaluation:

——Low @ Low vs High @ Low -> compare
——High @ High vs Low @ High -> compare
Experiment 3
–Silica addition
–CG silica Properties
–Self-assembly Model X + Si
——1000, 10000 (Maybe more?)

6   Discussion

Experiment 1
–Which is the best Model?
–Why Model X is better than Model Y?
–To what extent Model X represent well the AA model?
–What are the limitations and advantages of Model X?
Experiment 2
–What is the relation between the entropy "error" and concentration?
–Is the concentration really an problem?
–Do you need to run an AA for each con or the Ideal can represent any?
–Even extreme cases can be approximated? In this case how much is the "error"?
Experiment 3
–Is The silica CG model suitable?
–Can you the proprieties validate it?
–The self-assembly behaviour change with silica addition?

7   Conclusion

What did you really see from the results?
There was any bad assumption that is contestable?
What will be the next step to your research?

8   Nomenclature

$\mathcal{H}$ Hamiltonian
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.

Jin, H., Qiu, H., Sakamoto, Y., Shu, P., Terasaki, O., & Che, S. (2008). Mesoporous silicas by self-assembly of lipid molecules: Ribbon, hollow sphere, and chiral materials. *Chemistry – A European Journal*, *14*(21), 6413–6420. Retrieved from `http://dx.doi.org/10.1002/chem.200701988` doi: 10.1002/chem.200701988

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

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). Systematic implicit solvent coarse graining of dimyristoylphosphatidylcholine lipids. *Journal of Computational Chemistry*, *35*(16), 1208–1218. Retrieved from `http://dx.doi.org/10.1002/jcc.23610` doi: 10.1002/jcc.23610

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

Need, A., & Someone, T. C. (2025). Mock article: You need to read more. *Journal of Void Theory*, *1*(1), 151-1510.

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].

10 Appendix

$$E_c = \frac{1}{2}mv^2 \tag{22}$$

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
131	1,00	2,50	10	441.735	654.499
132	1,00	2,50	15	460.925	681.645
133	1,00	2,50	20	469.115	693.700
211	1,25	1,50	10	374.784	557.029
212	1,25	1,50	15	399.053[b]	591.402
213	1,25	1,50	20	411.377	609.042
221	1,25	2,00	10	415.050	615.336
222	1,25	2,00	15	430.991	638.237
223	1,25	2,00	20	455.857	673.613
231	1,25	2,50	10	472.885	698.958
232	1,25	2,50	15	484.567	715.341
233	1,25	2,50	20	497.320	733.923
311	1,50	1,50	10	385.110	572.665
312	1,50	1,50	15	406.144	602.130
313	1,50	1,50	20	480.613	706.960
321	1,50	2,00	10	490.910	722.194
322	1,50	2,00	15	513.846	754.804
323	1,50	2,00	20	529.291	777.175
331	1,50	2,50	10	542.184	796.618
332	1,50	2,50	15	510.958	753.298
333	1,50	2,50	20	527.253	776.981
			Sum:	12.138.966	17.932.100

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

Figure 6: Work in Progress