On the nature of self-assembly in low-density condensed matter – A classical molecular dynamics approach

Alfredo Metere
On the nature of self-assembly in low-density condensed matter

A classical molecular dynamics approach

Alfredo Metere
Abstract

The study of the physics of self-assembly in low density condensed matter is an extremely interesting, still a mainly unexplored field of scientific research. The contribution reported in this thesis explains how this problem can be addressed using molecular dynamics simulation of systems composed by simple, identical particles, interacting via a spherically symmetric pair potential, that belongs to a class of Dzugutov potentials [A:1-2]. Such approach resulted in four, self-assembled archetypal structures, which are reported in the included papers I, II, III, IV. In order to produce the reported results, a major effort of software development has been done by the author, both in the simulation and the analysis programs used.

This thesis will start with a brief introduction to the field, highlighting the important aspects to have a more complete, general understanding of the reported scientific results. Some conclusions will be drawn, and possible future endeavors.

References

Physics is my religion, because I have faith in that it works even though I do not fully comprehend it.

(Albert Einstein)
List of Papers

The following papers, referred to in the text by their Roman numerals, are included in this thesis.

PAPER I: **Formation of the Smectic-B Crystal from a Simple Monatomic Liquid**
DOI: [10.1103/PhysRevE.88.062502](https://doi.org/10.1103/PhysRevE.88.062502)

PAPER II: **Formation of a new archetypal Metal-Organic Framework from a simple monatomic liquid**
DOI: [10.1063/1.4903925](https://doi.org/10.1063/1.4903925)

PAPER III: **Formation of a columnar liquid crystal in a simple one-component system of particles**

PAPER IV: **A new kind of soft-matter quasicrystal**

PAPER V: **A new computer program for topological, visual analysis of 3D particle configurations based on visual representation of radial distribution function peaks as bonds**
Reprints were made with permission from the publishers.
Author’s contribution

Paper I

Structure classification, characterization of the real space configuration, figures.

Paper II

Initial structure classification, characterization of the real space configuration, figures, writing of the sections: I, III-B, and III-C.

Paper III

Simulations (16384 particles and 131072 particles), structure classification, characterization of the real space configuration, figures. Supplementary material.

Paper IV

Initial structure classification, characterization of the real space configuration, figures, quasicrystal tiling.

Paper V

Writing of the article, figures, main idea, writing of the code of the program.
Contents

Abstract iv
List of Papers vii
Author's contribution ix
List of Figures xv
List of Tables xvii

1 Introduction 1

2 Molecular Dynamics Fundamentals 3
  2.1 The MD fundamental algorithm ......................... 3
  2.2 Setting the initial conditions ......................... 4
    2.2.1 System size ........................................ 4
    2.2.2 Density ............................................. 5
    2.2.3 Boundary conditions ............................... 5
    2.2.4 Interparticle potential ............................ 6
      2.2.4.1 Reduced Units ................................. 7
    2.2.5 Integrator .......................................... 7
    2.2.6 Time step .......................................... 9
    2.2.7 Length of Sampling ............................... 10
### 3 Self-assembly with molecular dynamics

- **3.1 The system** ........................................... 13
- **3.2 The potential** ........................................ 14
- **3.3 The procedure** ...................................... 15
  - **3.3.1 Simulation stage** .................................. 15
  - **3.3.2 Post-simulation analysis** ....................... 17
    - **3.3.2.1 Real space structure analysis** ............... 17
    - **3.3.2.2 PASYVAT** .................................... 20
    - **3.3.2.3 Reciprocal space structure analysis** ........ 20
    - **3.3.2.4 Thermodynamic analysis** .................. 23

### 4 GROMACS

- **4.1 Introduction** ........................................ 25
- **4.2 Parallelism in GROMACS** .......................... 29
  - **4.2.1 The GPU architecture** ......................... 30
- **4.3 Implementation of Dzugutov potential in GROMACS** .... 32
  - **4.3.1 The $j$-loop** .................................. 33
- **4.4 Performance Analysis** .............................. 35
- **4.5 Conclusions** ....................................... 37

### 5 Liquid Crystals

- **5.1 Introduction** ....................................... 39
- **5.2 Nematics** .......................................... 41
- **5.3 Smectics** .......................................... 43
- **5.4 Columnar Phases** .................................. 45
- **5.5 Conclusions** ....................................... 46
6 Smectic-B phases with \( \geq 3D \) periodicity

6.1 Smectic-B quasicrystal .......................... 47
6.2 Smectic-B crystal ................................. 49
6.3 Conclusions ........................................... 50

7 Metal Organic Frameworks

7.1 Coordination chemistry ............................. 52
7.2 Properties, classification and uses of MOFs .............. 53
7.3 Reticular Chemistry .................................. 55
7.4 MOF structure discovery ........................... 57
7.5 Molecular dynamics simulation ....................... 58
7.6 Conclusions ........................................... 59

8 Conclusions

8.1 Future perspective .................................... 61

9 Popular Science

10 Summary of Papers

10.1 Paper I. Formation of the Smectic-B Crystal from a Simple Monatomic Liquid ............................................ 65
10.2 Paper II. Formation of a new archetypal Metal-Organic Framework from a simple monatomic liquid ..................... 65
10.3 Paper III. Formation of a columnar liquid crystal in a simple one-component system of particles ......................... 66
10.4 Paper IV. A new kind of soft-matter quasicrystal ............... 67
10.5 Paper V. A new computer program for topological, visual analysis of 3D particle configurations based on visual representation of radial distribution function peaks as bonds ..................... 67

Acknowledgments lxix
10.6 Personal acknowledgments . . . . . . . . . . . . . . . . . . . . . . lxx

References lxxiii
List of Figures

2.1 Schematic representation of periodic boundary condition . . . 6
2.2 Leapfrog integration scheme . . . . . . . . . . . . . . . . . . . . 8
2.3 Numerical approximation of Einstein frequency $\Omega$ . . . . . 10
2.4 Conceptual representation of ergodic hypothesis . . . . . . . . 11

3.1 Example of self-assembled structure . . . . . . . . . . . . . . . 13
3.2 Plot of Dzugutov pair potential. . . . . . . . . . . . . . . . . . . . 14
3.3 Example of phase diagram . . . . . . . . . . . . . . . . . . . . . . 16
3.4 Screenshot of VMD . . . . . . . . . . . . . . . . . . . . . . . . . . 17
3.5 Radial Distribution Function calculation . . . . . . . . . . . . . 18
3.6 Radial distribution function plot of $alm$-MOF . . . . . . . . . . 19
3.7 Screenshot of PASYVAT . . . . . . . . . . . . . . . . . . . . . . . . 20
3.8 RDF vs. structure factor . . . . . . . . . . . . . . . . . . . . . . . . 21
3.9 Example of spherically averaged structure factor . . . . . . . . 22

4.1 NVIDIA Geforce GTX 780 Graphics Accelerator . . . . . . . . 25
4.2 NVIDIA DIGITS DEVBOX computer with four installed Geforce GTX Titan X GPUs . . . . . . . . . . . . . . . . . . . . . . . . . . 26
4.3 A heterogeneous supercomputer: Titan . . . . . . . . . . . . . . 27
4.4 Schematic representation of a heterogeneous CPU-GPU cluster 28
4.5 Schematic of Kepler CC 3.5 Streaming Multiprocessor Archi-
tecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
List of Tables

4.1 Performances for GROMACS runs reported in millions of time steps computed per day ............................................. 35

4.2 Different computational efficiency $P$ in function of system size $N$. $P$ is expressed in nanoseconds per time-step per particle. 37
1. Introduction

The work presented in this thesis concerns the use of a simulation method, molecular dynamics (MD), for the investigation of the fundamental physical principles driving the self-assembly process in low-density, condensed matter. Condensed matter physics studies the behavior of matter which is found in solid, liquid and all other possible states of matter in between. Soft matter is a branch of condensed matter specific for some phases known as mesomorphic phases (or mesophases), which occur somewhere in between liquids and solids, for example liquid crystals and colloids [1]. Hence, self-assembly can be defined as the spontaneous arrangement of atomic, molecular or supra-molecular building blocks into a well defined, thermodynamically stable, three-dimensional, condensed structure being either crystalline, quasicrystalline or mesophase.

Molecular dynamics (MD) is a computer simulation method used in physics to monitor the behavior of microscopic-scale physical systems in time. Because MD simulation uses the laws of classical mechanics to describe the system, it is conceptually analogous to the N-body simulation, used instead to simulate the behavior in time of massive objects, like planets and galaxies. This is possible because the laws of classical mechanics are in principle scale-invariant [2][3].

Another important point, explained with more detail in the next chapter, is that MD simulation will produce scientifically valid results by simulating systems of particles exhibiting negligible quantum mechanical behavior (the classical hypothesis [4]). Because of its compute-heavy nature, the adoption of MD simulation as a research methodology is bound to the available computational resources and technological advances in computer performance. The global trend of advances in computing technology, well described by Moore's law [5], is exponentially growing, letting scientists use MD to simulate larger physical systems and for longer time. Furthermore, breakthrough technologies, like General Purpose GPU computing, enable scientists to use graphical processors for numerical scientific calculations. This technology brought serious number crunching capacity even to common desktop computers, so impressive that it is equivalent to what is currently available in a ≈ 512 nodes Linux cluster. It follows that the adoption
of MD simulation as an investigation method is becoming more and more common in many different branches of physical sciences. Currently the largest user base of MD simulation is constituted by computational biophysicists, who mainly investigate nucleic acids or proteins for diverse scientific purposes. Another populated group of MD users are trying to reproduce numerous different properties of real materials, with focus on their chemical behavior. In condensed matter physics MD simulation is used to understand fundamental properties of matter, like self-assembly, phase transitions, and topological properties of solids. Depending on the wanted result, the models describing the matter and MD simulation conditions can be dramatically different. The study of the fundamental physical principles driving the self assembly processes in condensed matter is of certain scientific interest. When approaching the problem, the following preliminary questions should be addressed: does self-assembly depend on the chemical order? Could it depend on the anisotropic interactions of the constituent molecules? How simple should a model be to reproduce a non-trivial self-assembled structure?

Physical laws driving the universe are simple after all, although they are usually hidden by the enormous variability of all the possible combinations of the parameters applying to them. The main idea pursued in the studies reported in this thesis is to create a simple model comprising identical particles (one-component) for the self-assembly of complex structures, like metal-organic frameworks and liquid crystals. The simplicity of the chosen physical model has been key for the success of the reported research. However, for computational studies like the one presented in this thesis, the goodness of the physical model is a necessary, although not sufficient condition. Simulations are run in computers and, for large simulation runs, a very efficient and fast MD simulation program must be adopted. In the reported research, GROMACS, a very fast and efficient parallel program for MD simulation, has been used. GROMACS was not immediately suitable to work with the physical models needed in condensed matter physics, so it has been customized accordingly. To analyze the resulting particle configurations, PASYVAT, a visual analysis program, was developed from scratch. The importance of the creation of PASYVAT and expansion of GROMACS will be explained in the coming chapters.

Often happens that almost nobody will ever read a doctoral thesis. If you, dear reader, are one of the exceptions, congratulations for your excellent choice. Have a pleasant reading.

Stockholm, 2015-03-31. Alfredo Metere
2. Molecular Dynamics
Fundamentals

The classical molecular dynamics (MD) simulation is a numerical, predictive technique aimed to observe the evolution in time of a physical system described by classical mechanics, and composed by microscopic particles. Practically, MD simulation obtains the trajectory of the system by integrating numerically the time-dependent equations of motion for each particle. Such approach is possible, provided that the particles exhibit a behavior that can be accurately described by the laws of classical mechanics, hence having a negligible quantum mechanical behavior (classical hypothesis) [4].

The study of a physical system can be pursued either by calculating precisely its evolution in time, which is called the \textit{deterministic} approach, or alternatively by assuming that its possible physical states represent a random process, independent of time, that can be described with a statistical distribution function; for this reason the latter approach is referred as \textit{stochastic}. With the \textit{deterministic} approach the mechanical state of the system is exactly estimated for each instant of time "observed" (sampled). The \textit{stochastic} approach instead assumes that a particular state of the system fits to an unknown statistical distribution function resulting from a random distribution of physical processes, which are assumed \textit{a priori} to be equiprobable. Actually there are also some theoretical methods that blend the two approaches (deterministic and stochastic), but they will not be discussed in this thesis. Classical MD simulation is a fully \textit{deterministic} method.

2.1 The MD fundamental algorithm

The MD algorithm is constructed so that the time integration is performed by a loop. The generic MD simulation algorithm comprises the following steps:
1. Start simulation
2. Set initial conditions (time = 0)
3. Start time integration loop
4. Calculate particles forces
5. Calculate particles velocities
6. Calculate particles positions
7. Need to calculate more system trajectory?
8. If so, time = time + 1; Jump to step 2
9. If not, Write final particle positions and velocities
10. End simulation

2.2 Setting the initial conditions

The initial conditions are a group of parameters that need to be set to begin a molecular dynamics simulation. The minimum initial conditions needed are: system size, density, boundary conditions, form and eventual parameters of potential energy function, numerical integrator, time step width, and length of time sampling (measured in number of time steps).

2.2.1 System size

The system size is defined as the number $N$ of microscopic particles composing it. If the macroscopic properties of the simulated system depend on its system size, it means that the system size must be increased until this correlation disappears. It is clear that the greater the number of particles, the greater the computational resources required for the simulation. Hence, the choice of the system size should be considered as a trade-off, within certain limits, between accuracy of desired macroscopic properties and available computational resources. Taking into account the average, currently available computational resources, it is generally not advisable to setup a system size with order of magnitude greater than millions of particles ($\log_{10} N > 6$), although much larger system sizes have been used for MD simulation ($\log_{10} N \approx 12$) [6].
2.2.2 Density

The particles are enclosed in a box of an initial, arbitrarily chosen, volume $V$. This means that the initial number density $\rho = N/V$ is known. Despite $\rho$ being a derived quantity, it is the commonly referred property (instead of $V$). This is because it is easier to compare results obtained in systems of different size using $\rho$ instead of $V$.

2.2.3 Boundary conditions

The boundary conditions are a set of parameters which are provided to the calculation and kept fixed during the simulation, such as the macroscopic physical states (macrostates) and the type of interaction of the particles with the walls of the simulation box in which they are enclosed. The macroscopic states of the system that could be fixed are: the size $N$, the volume $V$, the internal energy $E$ (which is assumed to be the sum of kinetic and potential energy), the average temperature $T$, and the average pressure $P$. In case the system presents chemical order (the particles of the system represent an atomistic description of molecules), an additional parameter $\mu$, the chemical potential, can be fixed.

The Newtonian equations of motion are defined for a system with fixed $NV E$. However, the equations of motion can be rewritten to keep other macroscopic states constant instead, with combinations like $NVT$ and $NPT$. The equation to keep the system at average constant temperature is called thermostat, while the one to induce average constant pressure is called barostat.

Another important boundary condition is the type of interaction of the particles in the system with the walls of the simulation box in which they are enclosed. The boundary condition responsible for the interaction of the particles with the walls of the simulation box can be considered to be either fixed or periodic. With periodic boundary condition (PBC) the system is regarded as a periodically repeated cell along the 3D translational axes defined by the geometry of the box. The system itself is referred as the primary cell, while its replicas are defined as the image cells. A schematic, graphical representation of a system with PBC is shown in Fig. 2.1.

When PBC is applied, the particles of the primary cell, close to the cell walls, are free to continue their trajectory in the neighboring image cell, conserving their momentum. In this way, no collision of the particles with the cell walls can happen. The advantage of using PBC consists mostly in the possibility to reduce dramatically the system size, without sacrificing the
Figure 2.1: Schematic representation of periodic boundary condition - The particles inside the blue box in the center represent the primary cell. The surrounding grey particles are the virtual replicas, located in the image cells. The green borders are the cell borders.

accuracy of the desired average macroscopic states of the simulated system. This is the reason why in the research reported in the included papers of this thesis, PBC is always applied.

2.2.4 Interparticle potential

The interparticle potential is an energy function which describes the potential energy contribution of the single particles to the potential energy of the system. The interparticle potential assumes that each particle in the system is influenced by its neighboring particles. If the system presents chemical order, the interparticle potential could be referred as interatomic potential. If the interparticle potential varies only in function of unique pairwise interactions between the particles, it is defined as pair potential. If the result of the potential does not depend of the orientation in space of each particle in respect neighboring particles, the interparticle potential is found to be spherically symmetric. The simulations presented in this thesis are all based on a parametric, spherically symmetric pair potential, the Dzugutov
2.2.4.1 Reduced Units

Another important aspect of the interparticle potential is that it can be used as a reference for the measurement units of energy. When energy units are defined by the form of the interparticle potential, they are referred as reduced units (r.u.). Because the simulated system in molecular dynamics is only described mechanically, its full thermodynamic description can only be obtained through statistical mechanics. In statistical mechanics the macroscopic states of the system are all derived from either or both kinetic and potential energy contributions. For this reason, the reduced units also define time, length, temperature, pressure, etc. In the following chapter more rigorous details about the potential form used in the simulations reported in this thesis will be elucidated with greater detail.

2.2.5 Integrator

The MD simulation calculates the trajectory of the particles of the system by solving time-dependent Newton's equations of motion. Because these equations are ordinary differential equations, their numerical solution requires discrete integration with respect to time. The methods that can be used for numerical integration are called integration schemes or simply integrators. The process of numerical integration can be thought of a simple iterative computer algorithm comprising simple algebraic operations with finite quantities. The numerical solution of an integral requires both the choice of an arbitrary finite integration step and an arbitrary number of integration steps to compute. For integration respect to time, the integration step is often referred as the time step. The accuracy of all numerical integrators is affected by a certain, unavoidable round-off error due to the approximation of the integration step to a finite-precision value (discretization). Such error can be minimized by reducing the size of the discrete time step considered.

Some differential equations, like the equations of motion defined in classical mechanics, present a certain degree of interdependence between their terms. Newton's laws are constructed on the basis of an energy function describing the energetic configuration of the system of particles, referred as the Hamiltonian function (or simply Hamiltonian)\[4\].
For the kind of systems used in this thesis the Hamiltonian is a scalar function in two terms, the kinetic and potential energy, dependent respectively on particles’ momenta and positions. The momenta and positions of the particles are correlated: they form a special kind of vector field, known as Hamiltonian vector field, which is called symplectic (from Greek, intertwined). In practice, the intertwining of the two vectors is due to the fact that they describe two properties (position and momentum) of the same object (the particle). Because both of these properties are related by the mathematical form of the time-dependent equation of motion, the variation of the position and momentum vectors cannot happen independently.

If the integrator neglects the symplecticity of the position and momentum vectors in the equation of motion, the calculation of potential and kinetic energy for the system will start diverging considerably from the realistic (or expected) value in few time integration iterations. The degree of divergence between the realistic (or expected) value and the computed value is referred in mathematics as drift. Within the context of MD simulation, the drifting is mainly observed in the energy calculations, and it is referred as the energy drift. As mentioned above in the paragraph regarding the definition of reduced units, the full thermodynamic description of the simulated system is ultimately based on the accuracy of both kinetic and potential energies. Hence, if energy is not computed accurately, such error will directly affect the accuracy of the estimated thermodynamical states of the system.

The most stable and computationally efficient symplectic integrator is the leapfrog scheme. Other examples of integrators are the Euler scheme (non symplectic) and the Velocity Verlet scheme (symplectic). The leapfrog has numerous advantages, but, in the context of this thesis, the most relevant is
its *numerical stability* \cite{12,13}. This means that the leapfrog scheme allows the usage of time steps of larger width, while keeping the energy drift acceptably low. The leapfrog scheme is usually preferred to other symplectic integrators, like the Velocity Verlet scheme, for being computationally more efficient \cite{14-16}. A schematic of how the leapfrog calculates the velocities and the positions at interleaved time points is shown in Fig. 2.2. The basic integration formulas for the leap-frog algorithm are:

\begin{align}
\mathbf{x}_i &= \mathbf{x}_{i-1} + \mathbf{v}_{i-1/2} \Delta t, \\
\mathbf{a}_i &= \mathbf{F}(\mathbf{x}_i), \\
\mathbf{v}_{i+1/2} &= \mathbf{v}_{i-1/2} + \mathbf{a}_i \Delta t
\end{align}

(2.1a) \hspace{1cm} (2.1b) \hspace{1cm} (2.1c)

where $\Delta t$ is the size of each time step, $\mathbf{x}_i$, $\mathbf{v}_i$ and $\mathbf{a}_i$ are respectively the position, velocity and acceleration for each particle of the system at MD iteration step $i$.

GROMACS, the MD simulation program that has been used for the research works reported in this thesis, uses the leapfrog scheme as the default time integrator.

2.2.6 Time step

Another important initial condition required for MD simulation is the choice of the time step width. To maintain its numerical stability, the leapfrog integrator requires that the time step $\Delta t$ width is kept constant throughout the whole simulation \cite{17}. It should also be considered that the particles are constrained in their motion by their neighbors, which create a potential energy well, assumed *a priori* to be spherically shaped. The particle in the potential energy well is also assumed to follow a 3D trajectory described by an isotropic harmonic motion with frequency $\Omega$, which is known as *Einstein frequency*. For a time step width equal to $\Delta t < \Omega/10$ the sampling of such particle vibration will be affected by a negligible accuracy error, due to discretization. For example, if $x(t)$ describes the 1D component of the vibrational harmonic trajectory followed by a particle for one period ($t = \Omega^{-1} s$), with Einstein frequency $\Omega = 1 s^{-1}$ its equation can be simply written approximating the canonical harmonic equation to:

$$x(t) = \sin(2\pi \Omega t)$$

(2.2)

where $2\pi$ is the conversion factor of radians to length unit. It can be seen in Fig. 2.3 how a $\Delta t \leq \Omega/10$ can correctly calculate the trajectory described by Eq. 2.2.
Figure 2.3: Numerical approximation of Einstein frequency $\Omega$ - The blue and red curves in the graph are respectively the theoretical and the approximated harmonic function with discretization of the time axis in time steps of size $\Delta t = \Omega/10$.

2.2.7 Length of Sampling

One important question that must be addressed during the initial setup of a MD simulation is: for how many time iterations should the simulation run? There is not a correct, single answer to this question, but it rather depends by the case. What can be said is that the simulation must produce an equilibrium state corresponding to a certain region of motion in the phase space. The region of motion is formed by the phase space trajectory of the system. The length of sampling should be long enough for the system to lose its correlation with its initial conditions. This is a necessary condition for the result to be considered realistic and it is referred as ergodicity $[2, 4, 18, 19]$. A conceptual schematics of ergodicity is shown in Fig. 2.4.

With the purpose of using MD simulation to describe self-assembly phenomena, the simulations reported in the papers included in this thesis have been always run long enough to obtain a phase in ergodic equilibrium, at a fixed average temperature $T$. 
2.3 Statistical Mechanics in MD simulation

Molecular dynamics simulation uses statistical mechanics to obtain the thermodynamical description of the simulated system of microscopic particles at ergodic equilibrium. In practice this is done by substituting the classical thermodynamic definitions with statistical mechanical definitions of pressure, temperature, and energy in the thermodynamical equation of state to obtain what is called the virial equation:

\[ PV = Nk_B T + \frac{1}{3} \langle \mathcal{V}_{\text{int}} \rangle_t \]  

(2.3a)

where

\[ \mathcal{V}_{\text{int}} = \sum_{i=1}^{N} \vec{r}_i(t) \cdot \vec{\nabla}_i V_N [\vec{r}^N(t)] \]  

(2.3b)

where \( P \) is pressure, \( V \) is volume, \( N \) is the system size defined as the number of particles in the system, \( k_B \) is the Boltzmann's constant, \( T \) is temperature, and \( \langle \mathcal{V}_{\text{int}} \rangle_t \) is the time-averaged internal part of the total virial function which takes into account the potential energy contribution due to the inter-particle potential \[4\].
Due to the complexity of statistical mechanics itself, and due to the limited understanding of its foundations [18–23], a deeper discussion about statistical mechanical methods of investigation would exceed the boundaries of this thesis’ scope. Hence, in this text it will be assumed that, for the simulated systems reported in the included papers, statistical mechanics provides an accurate enough estimation of the system’s thermodynamic average states in function of its classical mechanical description. It can be summarized that, when the laws of statistical mechanics can be applied, MD simulation is a suitable model to predict physical processes involving systems for which their microscopic physical states determine a macroscopic observable state. The study of self-assembly in low-density condensed matter indeed belongs to such category of physical problems.
3. Self-assembly with molecular dynamics

After describing more generally the basics of MD simulation and why it has been chosen for the study of self-assembly, this chapter will be specific for the methods used to obtain the results reported in the papers included in this compilation thesis.

![Example of self-assembled structure - The alm-MOF](image)

**Figure 3.1: Example of self-assembled structure** - The *alm*-MOF [24]. This is a low density, porous, crystalline self-assembled structure.

3.1 The system

For all the MD simulations reported in the included papers, the systems were composed by identical particles (one-component), with no chemical order (particles do not represent specific atoms), and at different number densities. The boundary conditions have been always fixed *NVT* and PBC.
Multiple system sizes (respectively 16384, 131072, and 1048576 particles) have been used. The choice of omitting the chemical order in the system description relates with the following conceptual question: *how simple the description of a system can be to accurately reproduce the dynamic process of self-assembly?*

The simulations reported in the papers included in this thesis demonstrate that the chemical order is a redundant detail for the physical study of self-assembly, at least for some cases.

### 3.2 The potential

![Figure 3.2: Plot of Dzugutov pair potential.](image)

- In this graph the parameters for *alm-MOF* [24] were used.

The potential used in the reported papers has the same form, but different parameters in each paper. In literature such potential form is reported as *Dzugutov potential*, from the name of its creator Mikhail Dzugutov, who first introduced it in 1992 [7][8].

The Dzugutov potential is a non-bonded pair-potential, and has the following general form:

\[
V(r) = a_1(r^{-m} - a_0)H_1(r, b_1, c_1) + a_2 H_2(r, b_2, c_2) \tag{2.3aa}
\]

with

\[
H(r, b, c) = \begin{cases} 
\exp \left( \frac{b}{r-c} \right) & r < c \\
0 & r \geq c 
\end{cases} \tag{2.3ab}
\]

where \( r = |\vec{r}_i - \vec{r}_j | \) represents the pair-wise \( i, j \)-interparticle distance. The Dzugutov potential presents eight parameters which can be adjusted arbitrarily. One of the most important features in the potential is the minimum
at $r \approx 1$ followed by a repulsive hunch, which then goes to zero at $r \geq c_2$. The narrowness of the first potential well determines the number of particles allowed in the first neighbor shell, while the value of $c_2$ determines in principle the radius of the second neighbor shell. The graph in Fig. 3.2 shows the potential shape with the parameters for the $al m$-MOF [24].

3.3 The procedure

For all simulations presented in this thesis the procedure has been similar. Because of the definition of self-assembly reported at the beginning of this chapter, the common starting point is the isotropic liquid phase. This means that any system at an appropriately high temperature and pressure will be found in liquid phase. A liquid is referred as isotropic if there is no orientational order in the particles composing it. Once the Dzugutov potential parameters, the density and other initial and boundary conditions are set, the simulation can begin.

3.3.1 Simulation stage

The initial system is stepwise heated with short, fixed-$NV T$ MD simulation runs, until an isotropic liquid in equilibrium is formed. This is the starting point from which a stepwise cooling will begin. For each stepwise decrement in the fixed temperature value, the simulation runs for long enough to obtain an equilibrium phase. This process of stepwise cooling proceeds usually with differences of temperature in the order of $T = T_0 - 10^{-1} r.u.$, so that an eventual, sudden phase transition is more difficult to miss. A phase transition is observable as a sharp drop of the system’s potential energy drop. Often, this phenomenon comes with the appearance of a certain degree of local or global order in the 3D particle configuration. Upon phase transition, the simulation parameters and the trajectory are saved for further investigation. The stepwise cooling is usually protracted for two or more extra temperature steps after phase transition. To establish the phase transition order, observable in either of the $[u/T],[p/T]$ phase diagrams (Fig. 3.3), a stepwise heating from the coldest particle configuration is operated, until the system reverts to an equilibrium isotropic liquid phase.
Figure 3.3: Example of phase diagram - Isochoric phase transition diagram for the Dzugutov columnar mesophase [25]. a) and b) are respectively Temperature vs. Potential Energy and Temperature vs. Pressure. Dots and open triangles correspond to cooling and heating, respectively.
3.3.2 Post-simulation analysis

After each simulation, the particle configuration is visually inspected with programs like VMD \cite{26-28}. A screenshot of VMD running on Microsoft Windows can be seen in Fig. 3.4. The analysis of a resulting particle configuration can be divided in three major stages: real space structure analysis, reciprocal space structure analysis, and analysis of the thermodynamical properties.

3.3.2.1 Real space structure analysis

The first analysis stage concerns the inspection of the resulting 3D particle configuration in real space. With a program called VMD (it stands for Visual Molecular Dynamics) it is also possible to see the simulated trajectory of the system as a movie. If some degree of order in the particle configuration is noticed, the radial distribution function, is calculated. The common notation for the radial distribution function is $g(r)$, where $r$ represents the interparticle distance. It is also called pair correlation function, because it is a particular $n$-particle distribution function $g_N^{(n)}(r^N)$ with $n = 2$ \cite{4}. The radial distribution function describes the variation of the system's density in
function of the interparticle distances. The radial distribution function \( g(r) \) can be operationally defined as follows [3]:

\[
g(r) = 2 \frac{\langle N(r, \Delta r) \rangle}{N \rho V(r, \Delta r)} \tag{2.3ba}
\]

where

\[
N(r, \Delta r) = \sum_i \sum_{j < i} \delta (r - r_{ij}) V(r, \Delta r) \tag{2.3bb}
\]

with

\[
\int_{-\infty}^{\infty} \delta(x) dx = 1 \quad \text{Dirac’s } \delta \text{ improper function} \tag{2.3bc}
\]

and

\[
V(r, \Delta r) = \frac{4\pi}{3} (r^3 - \Delta r^3) \tag{2.3bd}
\]

\( N(r, \Delta r) \) is a function returning the particle count within a spherical shell of volume \( V(r, \Delta r) \). To summarize, Eq. 2.3ba defines \( g(r) \) as the average particle count in a spherical shell of radii \( \{r, \Delta r\} \), normalized respect to the volume of the spherical shell, the number of particles in the system and the number density of the system. A graphic, schematic representation of it is

**Figure 3.5: Radial Distribution Function calculation** - Highlighted in light blue the section of the spherical shell centered on the particle highlighted in red. Its volume \( V(r, \Delta r) \) overlaps with some particles, so that for the same values of \( r \) and \( \Delta r \), \( N(r, \Delta r) = 6 \). The particle is counted only if the spherical shell overlaps the particle center.
shown in Fig. 3.5. Intuitively it can be imagined how a crystalline solid will present strong particle pair-correlation, opposed to a gas, where the inter-particle distances will be randomly distributed. To establish the degree of order in the system it is possible to plot $g(r)$ for an arbitrary interparticle distances interval and observe the sharpness and narrowness of the peaks. The plot of $g(r)$ will show a sequence of sharp, narrow peaks for the crystalline solid, opposed to the wide, flat curve that would be resulting from the calculation of $g(r)$ for a gas.

It should also be remarked how some peaks in the $g(r)$ plot of a crystalline solid would correspond to the potential minima at approximately the same value of $r$ in the interparticle pair potential. Despite this phenomenon is quite expected, what it is unpredictable a-priori is the intensity and the narrowness of such peaks. The radial distribution function plot of the $alm$-MOF is shown in Fig. 3.6. Because $alm$-MOF is a highly ordered crystal, the peaks of $g(r)$ are very sharp [24].

MOF [24] is shown in Fig. 3.6. The peak positions at $r \approx 1.1$ and at $r \approx 2.5$ correspond to the minima of the potential shown in Fig. 3.2 p. 14.
3.3.2.2 PASYVAT

Additionally, it has been created from scratch a program able to perform visual, topological analysis of any 3D particle configuration from the radial distribution function (RDF) plot. This program, PASYVAT [32] (PArticle SYstem Visual Analysis Tool) enables the user to select multiple radius intervals from the RDF plot and draw bonds between the matching particles found at the selected interparticle distances (see Fig. 3.7). This way, the crystal net underlying the structure can be clearly identified. The unique feature of PASYVAT, consisting in the multiple interparticle distances bonding functionality, has been used to identify and classify topologically the crystal net of a self-assembled, archetypal, crystalline porous material named alm-MOF [24]. The program has also been used for the characterization of the layer stacking in the smectic-B crystal [33], the columnar mesophase [25] and the smectic-B quasicrystal [29].

3.3.2.3 Reciprocal space structure analysis

This method is described in solid state physics and applied to X-ray diffraction crystallography. Reciprocal space is also called $Q$-space or Fourier space.
The reason is that the reciprocal space coordinates are obtained as the result of a *Fourier transformation* of certain, corresponding real space coordinates. In X-ray crystallography the experimental data consists of X-ray diffraction patterns, which are reciprocal-space projections of the real-space atomic configuration, which is usually unknown. Crystallographers struggle to reconstruct the real space configuration of the atomic structure of a chemical compound, from the single crystal X-ray diffraction data. The X-ray diffraction pattern is affected by instrumental error, and for this reason the procedure of obtaining the real space configuration is very complex.

![Figure 3.8: RDF vs. structure factor](image)

**Figure 3.8: RDF vs. structure factor** - The radial distribution function and structure factor calculated for the alm-MOF [24].

The exact inverse process has been applied to analyze the 3D real space configurations of the simulated systems treated in this thesis. Being the *raw simulation data* a specific real-space particle configuration, the inter-
est is in the computation and analysis of its corresponding diffraction patterns defined in the reciprocal space. The obtained diffraction patterns can be thought as the reciprocal space description of density-density correlation functions operated on the real space. Practically, this operation is done by calculating the \textit{structure factor} (see Fig. 3.8), which we define as \( S(\vec{Q}) = \langle \rho(\vec{Q})\rho(-\vec{Q}) \rangle \), where \( \rho(\vec{Q}) \) is the Fourier-component of the number density \( \rho \) of a system of \( N \) particles:

\[
\rho(\vec{Q}) = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} \exp(-i\vec{Q}\vec{r}_j)
\]  

(2.3c)

where \( \vec{r}_j \) represents the position of the \( j \) particle and \( \langle \rangle \) being the notation for \textit{ensemble average}. According to this definition, \( S(\vec{Q}) \) represents a diffraction intensity which is qualitatively comparable to the one obtainable by diffraction experiments. Another important characterization method is the spherically averaged structure factor \( S(Q) \). This is related with the structure factor calculation, and has been employed in the presented research works. Considering \( S(\vec{Q}) \) in one dimension leads to an equation that can be thought as the reciprocal-space counterpart of the radial distribution function \( g(r) \):

\[
S(Q) = 1 + 4\pi\rho \int_0^\infty [g(r) - 1] \frac{\sin(Qr)}{Qr} r^2 dr
\]  

(2.3d)

\[\text{Figure 3.9: Example of spherically averaged structure factor - Reciprocal space analysis of the alm-MOF structure [24]. Isointensity plots of respectively: a) Diffraction intensity in the reciprocal-space sphere. - b) Reciprocal-space plane orthogonal to the axis, } Q_z = 0 \text{ The axially oriented plane, } Q_y = 0. Q_z \text{ denotes the axial dimension, and } Q_y \text{ corresponds to a translational symmetry vector, orthogonal to the axis [24].}\]
3.3.2.4 Thermodynamic analysis

The thermodynamic description of the simulated system is obtained by the statistical mechanical virial equation, as mentioned in Chap. 2. In the specific of the work presented in this thesis, only fixed-$NV T$ simulations with periodic boundary conditions have been conducted. To keep the temperature constant, the velocity calculation term in the equation of motion is substituted with a special equation, commonly referred as thermostat. The thermostat which has been used in the reported simulations is the simplest: velocity rescaling. As the name suggests, it rescales every $n$ time step iterations the velocities, so that the arbitrarily chosen temperature $T$ is kept constant in time average. Because the kinetic energy is calculated in function of the particles velocities, with the velocity rescaling thermostat the kinetic energy will also remain constant in time average. With constant kinetic energy the simulation will lose its time-reversibility and the internal energy fluctuations will depend exclusively on potential energy.
4. GROMACS

4.1 Introduction

The computational study of condensed matter physics with molecular dynamics (MD) simulation is increasing in popularity in relationship with the ever growing computing capacity available. Traditionally, High Performance Computing (HPC) problems, like MD simulation indeed, have been addressed by the usage of massive, parallel computers comprising of many thousands of Central Processing Units (CPUs), general purpose electronic processors, organized as exotic supercomputers or as a network consisting of thousands of common mortal man Personal Computers running mostly Linux (Beowulf clusters). However, one of the latest breakthrough innovations is the possibility of using Graphics Processing Units (GPUs) as general-purpose processors for computational, scientific problems. A GPU capable of general purpose computing produced by NVIDIA, one of the leaders in the GPU market, is shown in Fig. 4.1. The synergistic usage of CPU and GPU resources for general purpose computing led programmers to rediscover a
programming paradigm commonly referred as *heterogeneous computing*. If we compare peak, floating point operations per second (FLOPS), GPUs perform much faster than CPUs. However, in real world applications the performance is strongly bound to the computational problem type and size, so that squeezing all the FLOPS out of both CPUs and GPUs is a quite complex procedure, which involves plenty of expertise, study of the specific problem and patience. For the most cases the speed-up obtained by offloading the computationally demanding tasks to the GPUs is consistent enough to attract progressively more adepts to the cult of GPGPU computing. One of the most appreciated advantages of GPGPU computing is the possibility of running impressively large calculations on common gaming desktop computers (see Fig. 4.2), instead of using a remote Linux cluster: whoever ran simulation jobs in a cluster knows how long and frustrating the whole procedure can be.

![NVIDIA DIGITS DEVBOX computer with four installed Geforce GTX Titan X GPUs](image)

*Figure 4.2: NVIDIA DIGITS DEVBOX computer with four installed Geforce GTX Titan X GPUs* - This is an advanced single-node/ multi-GPU hardware setup. The GPUs in this desktop computer have a total theoretical peak performance of ≈ 28 TFLOPS (trillions of single precision floating point operations per second), which is theoretically ≈ 56 times faster than a single Intel Core i7 3770K @ 3.9 GHz CPU (0.5 TFLOPS peak)\[34,35\]. *Image courtesy of NVIDIA*
NVIDIA was the first to introduce a closed source, yet flexible and quite usable software development platform for general purpose GPU computing (GPGPU): the Compute Unified Device Architecture (CUDA). The CUDA platform comprises of GPGPU-specific compilers, libraries, development tools like debuggers, performance profilers, a virtual assembly language, PTX, and semantic modifications of the most common programming languages used in computational science, like C, C++, and Fortran. Among them, CUDA C is a semantic modification of C99 dialect [36,38] which remarkably facilitates the writing of GPGPU programs which could otherwise be written only using dedicated 3D graphical libraries as OpenGL and GLSL [37,39]. Because computing power is never enough, now the Beowulf clusters are incorporating nodes with one or more GPUs (multi-GPU is more common than single-GPU in this case), giving birth to what is known as heterogeneous HPC. This programming paradigm represents the high performance oriented variant of the previously mentioned heterogeneous computing, which means that is more focused in the performance that can be squeezed by a large Beowulf cluster (see Fig. 4.3), rather than a single multicore desktop computer. One of the most performant MD simulation programs that takes advantage of GPGPU computing is GROMACS [40-44]. The key features which make GROMACS much faster than the competitors are the presence of compute intensive CPU functions written in C using hardware-specific assembly language intrinsics, and the very efficient parallelism implementation both for shared and distributed memory model hardware architectures. One of the most computationally demanding oper-

**Figure 4.3: A heterogeneous supercomputer: Titan** - This computer is among the fastest on the planet and it has been installed in Oak Ridge National Labs, USA. Notice the woman next to the letter T to have an idea of the size of the supercomputer. The visible raw of cabinets is only one out of eight total. The remaining rows of cabinets are located behind the visible one. *Image courtesy of ORNL*
ations during MD simulation run time is the calculation of the force acting on each particle for every time integration step. A recent addition to GROMACS, available from its version 4.6.x, consists in the presence of efficient GPGPU force calculation functions (called in CUDA jargon *kernels*) [44]. GROMACS is strongly oriented to biomolecular simulation, which means that the interparticle potentials from which the force of each particle is calculated, and more in general the physical models implemented are the ones used in the field of computational biophysics. Consequently, the physical models implemented in the official version of GROMACS are poorly suitable for research in condensed matter physics. As mentioned before, the research presented in the included papers refers to MD simulation using the Dzugutov potential [7–9], which is a very suitable physical model for the study of self-assembly in condensed matter physics [24; 25; 29; 33]. The work presented in this chapter concerns the CUDA C implementation of the analytical Dzugutov potential in GROMACS. This implementation works efficiently in single-node/single-GPU, single-node/multi-GPU, multi-node/single-GPU, multi-node/multi-GPU (see Fig. 4.4) run time configurations. The implementation was written in CUDA C, and it takes advantage of some libraries commonly adopted in parallel programming: the OpenMP, the thread-MPI and the MPI libraries.

![Figure 4.4: Schematic representation of a heterogeneous CPU-GPU cluster - The thread-MPI and OpenMP both generate parallelism within a single node, either single or multi-GPU, while MPI can be used for all possible parallel execution configurations.](image-url)
4.2 Parallelism in GROMACS

The source code of GROMACS consists of approximately five millions lines of C instructions and its structure is extremely complex. Any modification in the source code is a complicated work which requires an experienced programmer to succeed. The structure of GROMACS makes it possible to implement a new physical model in CUDA C for force and energy calculation without the need to modify the CPU-side parallelism (OpenMP directives and MPI calls). The system to be simulated, when a parallel, MPI job is requested, is split into nearly equal sub-domains and each of these sub-domains is mapped into the computing resources in function of the MPI topology requested. This operation is called domain decomposition. Parallelism in CUDA force calculation is only implemented with MPI and not with OpenMP directives. The OpenMP directives are reserved for neighbor list recalculation and accessory run time functions. GROMACS can use two major types of MPI ranks: thread-MPI and traditional MPI. Opposed to traditional MPI, which works in any possible multi-processor/multi-node configuration, the thread-MPI works only on single-node multicore configurations and allows the usage of MPI symmetric multi-threading (SMT) to provide either multi-domain/single-GPU or multi-domain/multi-GPU calculations.

For performance reasons it is sometimes convenient to operate domain decomposition even when running on a single card to hide latency and improve utilization of all the execution units in the GPU. For each hardware setup there is a golden particles/GPU-cores ratio run time configuration that must be always taken into account. This is especially relevant when the problem size is not scalable with the amounts of cores available, like it often happens when simulating on cards with large amounts of cores, like Kepler architecture. In Fermi architectures there will be little or no advantage to use such run-time configuration. To summarize, it can be stated that such run-time configuration is aimed to provide an improved strong scaling.

The traditional MPI must be used for both multi-node/single-GPU and multi-node/multi-GPU calculations. With such run-time configuration it is only possible to use one MPI rank per GPU. GROMACS does not need exclusive resource utilization, but it is highly suggested. In a system with multi-GPU/multi-node hardware configuration, a perfect MPI topology is obtained by decomposing the domain so that one MPI rank is generated for each of the GPUs available in the cluster. In theory, it should be possible to use mixed multi-GPU nodes, where the number of GPUs per node changes. However, this configuration setup is not optimal because GROMACS will assign
the MPI ranks based on the node with the smallest number of GPUs installed. This means that MPI communication is needed when the neighbour list changes in some sub-domain. The neighbor list recalculation happens usually about \( \approx 100 \) times (this number is purely indicative, because in GROMACS it can be manually adjusted) less often than force calculation.

4.2.1 The GPU architecture

![Schematic of Kepler CC 3.5 Streaming Multiprocessor Architecture](image)

Figure 4.5: Schematic of Kepler CC 3.5 Streaming Multiprocessor Architecture - Kepler is a highly multithreaded architecture. Image courtesy of NVIDIA

GPUs are chips containing a massive amount of execution cores, subdivided in many, highly threaded units, called the streaming multiprocessors (SM). The organization and hierarchy of the SM depends on the hardware architecture of the GPU, reported as **CUDA Compute Capability** (CC) and a version number. For example, Fermi architectures are CC 2.0 or CC 2.1, while Kepler are CC 3.0 or CC 3.5, and Maxwell are CC 5.0 or CC 5.2. Regardless of the CC version, each SM includes a certain number of **streaming processors** (SP) which are **logically dumb**, as they cannot autonomously sched-
ule the execution of the instructions, and they work in single precision floating point (32 bit). Next to the SP are the **double precision streaming processors** (DP), which are streaming processors analogous to the SP, but they work exclusively with double precision floating point numbers (64 bit). Next to the SPs and the DPs are located the **special function units** (SFU). They are specialized execution cores for some complex mathematical and trigonometric operations, like square roots, logarithms, exponential functions and many others. The SPs, DPs and SFUs use **load/store units** (LD/ST) for fetching the input data and instructions from the **Register file** (RF), which is fed by the **dispatch units** (DU). Each dispatch unit is then managed by a **warp scheduler** (WS), which orchestrates the dispatch units feeding from the **Instruction Cache** [36–38; 48]. An interconnect network provides the bandwidth to put the cores in communication with the L1 Cache, also called the **shared memory** (ShMem) (the fastest memory accessible by all execution cores, SFUs, and Load/Store units). Slower than the shared memory and bigger in size is the **Uniform cache** (UC), which is a quite low latency memory space reserved for variables used frequently during run time, which do not change in size and are preloaded. The last, in order of speed and latency is the **texture cache** (TC), which preloads data on request from the **texture memory** (TM). All of these units are shown in Fig. 4.5 which reports the schematic of a single hardware streaming multiprocessor of a Kepler CC 3.5, called the SMX.

The description of the SM architecture is important because each SM works as a Single Instruction Multiple Data (SIMD) machine, while CUDA adopts a popular parallel programming style called the Single Program Multiple Data (SPMD) [37]. Programming SPMD means that the each SM of the GPU (which is a SIMD) will execute a different instruction, hence generating instruction parallelism (when possible). CUDA maps the execution of the kernel in the available SM, creating a so called **grid**. Each grid is divided in blocks and each block contains the execution threads. The important part of this subdivision is that each block addresses a portion of shared memory, accessible by all threads in the block, as well as register and local memory, which instead is accessible only by the single threads. Global, constant and texture memory are instead shared with all blocks in the grid (see Fig. 4.6 for a schematic representation). This way the CUDA program does not need to know (at least in principle), on which hardware configuration the device kernel will be run, resulting in a certain degree of code scalability.
4.3 Implementation of Dzugutov potential in GROMACS

All the force calculation CUDA kernels are located in (the indicated path is relative to git repository root directory): src/gromacs/mdlib/nbnxn_cuda. The file containing the force calculation kernels is nbnxn_cuda_kernel.cuh. A replica of this file has been created, nbnxn_cuda_kernel_dzugutov.cuh and then the potential has been separately implemented there. The file nbnxn_cuda_kernel.cuh is a template source used by the C preprocessor to generate compilable CUDA source files, one for each of the calculation kernels used by GROMACS for its functioning.

The file nbnxn_cuda_kernel_list.cuh contains the list of all the kernel prototype functions used by the preprocessor to generate the respective compilable sources. The selection of the appropriate calculation kernel is done in the file nbnxn_cuda.cu with a simple switch block.

In the first temporary implementation, a new kernel prototype which points to nbnxn_cuda_kernel_dzugutov.cuh has been added and the selection of Dzugutov potential kernel functions has been forced by hacking the switch block in this file. The time integration update of particles trajectories and velocities is implemented with two nested loops, namely the $i$-loop and the

![CUDA logical resource partition to launch a GPU kernel](image)

*Figure 4.6: CUDA logical resource partition to launch a GPU kernel - GPUs are hierarchical, massively multi-threaded architectures. Threads are organized in Blocks and blocks are organized in grids. Grids are then assigned to the available SMs. Image courtesy of NVIDIA*
The $j$-loop, as the outer and inner loops, respectively. The $i$-loop cycles through the time steps, while the $j$-loop cycles through the particles in the neighbor list. The neighbor list is calculated using a special algorithm which is called *cluster-based neighbor list calculation* \cite{49}. The $i$-loop just updates the particles trajectories, velocities and other non compute-intensive tasks. The $j$-loop is the compute intensive portion of the code.

There are four separate $j$-loop CUDA kernel types *per interparticle potential*: only force ($C_F$); force and energy ($C_{F,E}$); force and neighbor list recalculation ($C_{F,P}$); force, energy and neighbor list recalculation ($C_{F,E,P}$). The $C_F$ kernel is the most used in run time. As its name suggests, it only calculates the force, skipping other possible operations, like energy or neighbor list calculation. Analogous logic has been used to implement the remaining kernel types. To implement the Dzugutov potential these kernel types were to be considered in the modified preprocessor template. This means that in fact four distinct Dzugutov potential CUDA kernels are compiled and used in run time.

4.3.1 The $j$-loop

The per-atom data of each particle to be updated is usually copied from host memory to device texture memory as 1D array and preloaded in shared memory only on Kepler architectures. The hardware architecture selection for the preloading option is implemented with preprocessor directives. The data structure storing the per-atom information is quite complex as it takes into account all possible different potentials, position, velocity, mass, atomic radius, potential parameters and other related particle description values. Since the Dzugutov potential only considers the positions and not the identity of the particles, the standard per-atom data structure used in GROMACS is highly redundant. The implementation of a flexible, dynamic data structure is ongoing, so that it will take into account the Dzugutov potential and other similar potentials that are used in condensed matter physics. Before the particle state can be updated, the per-atom data structure of all particles involved in the calculation is copied from texture memory to shared memory, so that the shared memory stays filled with aligned data, possibly until the end of the calculation. A smaller per-atom data structure will likely lead to less shared-memory cycles needed, hence a better utilization of the multiprocessor unit, especially on Kepler architectures, where the control unit accepts SIMD instructions for 192 regular execution cores.
The interparticle calculations needed to update each particle state (position, velocity) are done using an opportunely tuned, shared memory based, tiling technique. Shared memory access is second in performance only to memory registers access and it can be used by threads as “cache” memory to feed registers with new data. Since we impose a fixed tile size optimizing register memory coalescence, the shared memory is declared and used as static.

As previously mentioned in the earlier chapter, the Dzugutov potential has eight tunable parameters, which in this implementation are hard-coded into the source code. This is simply a temporary solution to get simulation to work. There is an ongoing effort for the Dzugutov parameters to be read through the per-atom data structure, which is a non-trivial operation in such a massive, intertwined code. The force calculation is done solving numerically the first derivative of the potential (the force modulus) so that:

\[
\vec{F}_i = -\nabla_{\vec{r}_i} V_D(r_{ij}) = -\sum_{i<j} \frac{\vec{r}_i - \vec{r}_j}{r_{ij}} \frac{\partial}{\partial r_{ij}} V_D(r_{ij})
\]

(2.3a)

where \( r_{ij} = |\vec{r}_i - \vec{r}_j| \) and \( V_D(r_{ij}) \) is Dzugutov potential.

The calculation of \( r_{ij} \) can be sped up doing the following:

\[ r_{ij} = \vec{r}_i - \vec{r}_j \]

(2.3b)

and then using a CUDA special function unit (SFU) instruction to calculate \( r_{ij} = |\vec{r}_{ij}| \):

\[ \_\_d_e_v_i_c_e\_\_ \text{sqrtf(float } x) \]

The form of \(|\vec{F}_i|\) for the Dzugutov potential is the following:

\[
|\vec{F}_i| = -\sum_{i<j} \frac{\partial}{\partial r_{ij}} \left( a_1(r_{ij}^{-w} - d) H_1(r_{ij}, b_1, c_1) + a_2 H_2(r_{ij}, b_2, c_2) \right)
\]

(2.3c)

\[
\frac{\partial}{\partial r_{ij}} a_1(r_{ij}^{-w} - d) H_1(r_{ij}, b_1, c_1) = a_1 e^{\frac{b_1}{r_{ij}^{w+1}}} \left( \frac{w}{r_{ij}^{w+1}} + b_1 \frac{r_{ij}^{-w} - d}{(r_{ij} - c_1)^2} \right)
\]

(2.3d)

\[
\frac{\partial}{\partial r_{ij}} a_2 H_2(r_{ij}, b_2, c_2) = a_2 b_2 \frac{e^{\frac{b_2}{r_{ij} - c_2}}}{(r_{ij} - c_2)^2}
\]

(2.3e)
Table 4.1: Performances for GROMACS runs reported in millions of time steps computed per day

<table>
<thead>
<tr>
<th>#Particles</th>
<th>Node</th>
<th>GPU</th>
<th>#GPUs</th>
<th>Perf. ((10^6 \Delta t/day))</th>
</tr>
</thead>
<tbody>
<tr>
<td>16384</td>
<td>i7 3770K</td>
<td>GTX 780 Ti</td>
<td>1 GPU</td>
<td>240.695</td>
</tr>
<tr>
<td>131072</td>
<td>i7 3770K</td>
<td>GTX 780 Ti</td>
<td>1 GPU</td>
<td>32.767</td>
</tr>
<tr>
<td>1048576</td>
<td>i7 3770K</td>
<td>GTX 780 Ti</td>
<td>1 GPU</td>
<td>2.71</td>
</tr>
<tr>
<td>1048576</td>
<td>10 x Dual E5-2650</td>
<td>Tesla K20</td>
<td>20 GPUs</td>
<td>53.449</td>
</tr>
</tbody>
</table>

The parameters are: \(w, d, a_1, b_1, c_1, a_2, b_2, c_2\). The Eq. 2.3d is non-zero for \(r_{ij} < c_1\), while Eq. 2.3e is non-zero for \(r_{ij} < c_2\). It is possible to see that in the Eq. 2.3d the parameter \(w\) is present summed to a constant value \(w + 1\). Hence, another variable has been created, \(w_1 = w + 1.0\); to reduce the computational load of 1 FLOP per iteration. This produced a speed-up proportional to the simulated system size \(N\) (number of particles) and number of time integration iterations.

Another important consideration looking at the \(|\vec{F}_i|\) is about the presence of exponential functions. They are currently calculated using SFU instructions \(__device__ expf(float x)\). A possible performance gain could in principle be obtained by pre-computing a high-order polynomial which approximates the force function, so that no SFU instructions will be needed. This should speed up significantly the calculation, because the SFUs in the multiprocessor are significantly less than the standard execution cores. Hence, the multiprocessor utilization should increase dramatically when SFU instructions creating data races are avoided. Such implementation would also decrease the GPU workload by removing the conditional blocks for which the Eq. 2.3d and 2.3e are either computed or zeroed. This is just a quick-and-dirty (actually already very quick and not so dirty after all) solution that will soon be replaced by a release-grade implementation, which will support the reading of Dzugutov parameters from input files, without the need of recompilation and that will be fully integrated in GROMACS.

4.4 Performance Analysis

The potential has been tried with different problem sizes and run time configurations in two different computers. However, in table 4.1 are only reported the most relevant performance comparisons for different system sizes and run time configurations. It is interesting to compare the 1048576 parti-
cles system simulated in a desktop workstation with a GTX 780 Ti GPU and a 10 node job on Erik, a GPU cluster in Lunarc, the Data Center of Lund University, Sweden. The Geforce GTX 780 Ti is overclocked at 980 MHz, with 7 GHz memory clock and it features 2880 cores. The Tesla K20m is standard clocked at 706 MHz, memory clock at 5.2 GHz, with ECC activated, and it features 2496 cores. If we compare the strong scaling performance ratio (fixed problem size), a 10 nodes (20 GPUs) job is $19.72$ times faster than the single-node/single-GPU run. Such performance is quite interesting as the Geforce GTX 780 Ti card is faster than the Tesla K20m. It could be assumed that such performance trend is valid within a certain range of particles per GPU ratio. Assuming linear dependence between the system size and the measured performance, the GPU computational efficiency can be estimated by comparing the performance values of the single-node/single-GPU and normalizing them with respect to their system sizes, as shown in Fig. 4.7.

**Figure 4.7: GPU Performance Graph** - Each line represents the estimated performance of the GPU in other system sizes if the performance would scale linearly respect to the system size.

From the available data it is possible to establish that the GPU is most ef-
Table 4.2: Different computational efficiency $P$ in function of system size $N$. $P$ is expressed in nanoseconds per time-step per particle.

<table>
<thead>
<tr>
<th>$N$</th>
<th>16384</th>
<th>131072</th>
<th>1048576</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P[ns/\text{ts}/\text{pt}]$</td>
<td>21.91</td>
<td>20.11</td>
<td>30.41</td>
</tr>
</tbody>
</table>

Efficient with a problem size of 131072 atoms, but the absolute top performance is obtained with 16384 atoms. To better compare the GPU efficiency at different problem sizes, it is possible to express performance in time per timestep ($ts$) per particle ($pt$) so that:

$$P_{ns/ts/pt} = \frac{D_{ns}}{NP_{ts/day}(N)}$$

(2.3f)

where $D_{ns} = 24 \times 3.6 \times 10^{12}$ is one day expressed in nanoseconds (ns) and $N$ is the number of particles in the simulated system. The performance results are reported in Table 4.2. The system size with highest GPU efficiency is 131072 particles. This is expected for the strong-scaling limits of a massively parallel SPMD architecture, which is the commonly found for a CUDA GPU.

### 4.5 Conclusions

The CUDA implementation of Dzugutov potential in GROMACS has proven to be perform exceptionally well on single-node/single-GPU configuration. The most efficient $pt/\text{GPU}$ ratio is with no doubts 131072 $pt/\text{GPU}$. However, the absolute fastest simulation has been obtained with 16384 $pt/\text{GPU}$. The performance scalability trend between single-node/single-GPU setup with Geforce 780 Ti and multi-node/multi-GPU setup with Tesla K20m is almost linear, but this is probably a false positive. Indeed, the single Geforce GTX 780 Ti GPU, which is also factory overclocked, is much more powerful than the Tesla K20m, hence the performance in the multi-node/multi-GPU setup was expected to be lower than the actually observed. The only explanation currently available for the observed linear trend consists with the $pt/\text{GPU}$ ratio number. As stated above, the best $pt/\text{GPU}$ ratio has been found to be 131072 $pt/\text{GPU}$. This ratio is on the same order of magnitude of the effective ratio on the single Tesla K20m, which was $1048576 \, pt/20 \, \text{GPU} \approx 52429 \, pt/\text{GPU}$.
5. Liquid Crystals

Figure 5.1: Columnar hexagonal mesophase - Artistic representation of the first liquid crystal predicted with a system of identical particles interacting via a spherically symmetric pair potential [25]. Columns are liquid and hexagonally packed.

5.1 Introduction

It is commonly understood from basic education that matter can exist at equilibrium in three physical states (phases): solid, liquid and gas. However, this information is not completely correct, because for some materials the transition from liquid to solid is not straight, and other intermediate phases exist. With solid is meant the crystalline solid (or simply the crystal), and with liquid is meant the isotropic liquid. These intermediate phases are known in physics as mesomorphic phases (or simply mesophases). Mesophases are commonly known as liquid crystals [1], and the molecules capable of forming mesophases are called mesogens. Crystalline phases are formed by molecules or atoms arranged in a three-dimensional periodic lattice, while in the isotropic liquid phase the periodicity is not present. Liquid crystals can be then defined as 3D structures where the periodicity typical of crystals is confined to two or one dimension, while the remaining dimen-
sions present features typical of the liquid phase. Because of the dualism in the structural order, liquid crystals are anisotropic structures.

![Figure 5.2: Some rod-like mesogens](image)

Figure 5.2: Some rod-like mesogens - Mesogens with elongated shape tend to form nematic and smectic phases. In this figure are shown two nematogens: a) p-azoxyanisole (PAA). b) N-((p-methoxybenzylidene)-p-butyranilide (MMBA).

There is a high variability in the atomic structure of the mesogens (see Fig. 5.2), although their topology is reduced to either of these two possible morphologies: disc-shaped (or discotic) and rod-like. The rod-like mesogens present orientational order. Further cooling will eventually lead to the formation of a full 3D crystal.

![Figure 5.3: Thermotropic phases](image)

Figure 5.3: Thermotropic phases - Depending on the shape of the mesogens (rod-like or discotic) and other boundary conditions, not all of the phases will occur upon cooling of the same mesogenic system.

There are three fundamental categories of liquid crystals phases: nematic, smectic and columnar phases (examples are shown respectively in Figs. 5.4, 5.6, and 5.8). If the formation of different mesophases depends solely upon a change of temperature in the mesogenic systems, then both the mesogens and their related mesophase formations can be defined as thermotropic [1, 5]. A schematic of how thermotropic phases occur in order upon cooling is
shown in Fig. 5.3. Please note that the isotropic liquid, the smectic-B quasicrystal, the smectic-B crystal and the columnar crystal cannot be classified as mesophases. One-dimensional periodic order in liquid crystals can be obtained both as nematic mesophase, where the directors of each mesogen are oriented along a single directional axis, and as smectic-A mesophase, where the mesogens are arranged in liquid layers with periodic stacking. Nematic and smectic phases are formed commonly by rod-like mesogens. Some smectic mesophases, like the smectic-B mesophases, can present also 2D or 3D periodicity. Discotic mesogens form often 2D-periodic columnar phases, where the mesogens are stacked upon each other forming columns, which are hexagonally stacked. The hexagonal stacking order extends periodically along the plane orthogonal to the column elongation axis, which instead presents liquid character.

The Landau-Peierls instability theory [22; 51] demonstrates that for 1D-periodic mesophases the order cannot be maintained on the long range, while 2D-periodic mesophases and 3D crystals do exhibit long range, global order.

5.2 Nematics

![Figure 5.4: Nematic mesophase](image)

Nematic phases (from Greek, thread), are those mesophases which are closer in structural order to the isotropic liquid. Their distinctive characteristic is that the mesogens present a directional order consistent with their neighbors, so that a vector representing the such order, called the director can be traced (see Fig. 5.4). The director can assume any direction in
space, although its orientation is indistinguishable \[1\]. As mentioned in the introductory section of this chapter, nematics have been observed frequently, among elongated, rod-like molecules. It is however possible, under certain physical conditions, to observe discotic molecules arranged as a nematic mesophase. The missing correlation between the centers of gravity of the particles in the plane perpendicular to the director cause the nematics to lack long-range order. For the same reason, in the plane orthogonal to the director they flow like liquids. Despite most nematics are uniaxially anisotropic, biaxial nematics and more exotic symmetry (hexagonal, cubic, and icosahedral) have been observed, although they are very rare. By dissolving a chiral molecule in a nematic mesophase, the nematic structure will distort like a helix, forming a mesophase known as cholesteric (see Fig. 5.5). The nematic and cholesteric mesophases are usually classified as nematics, because locally the cholesteric phase behaves as a nematic

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{cholesteric.png}
\caption{Cholesteric mesophase - This cholesteric phase is undergoing thermomechanical coupling (also known as the Lehmann effect)\[52\]. This means that, under the effect of temperature gradient parallel to the cholesteric axis, the director keeps rotating at a constant rate \[53\].}
\end{figure}
mesophase. The preference of formation between nematic and cholesteric phases is related with the chirality (mirror plane symmetry) of the mesogens. Nematic will form only with achiral mesogens (molecules identical to their mirror images), while cholesteric will form only with chiral mesogens [1, 52, 53].

It is currently unknown whether the anisotropy of the mesogens is a requirement for the formation of nematic mesophases, and a study of these phases with the Dzugutov potential is deemed currently impossible, due to the spherical symmetry (isotropy) of the pair potential.

5.3 Smectics

![Smectic-A mesophase](image)

**Figure 5.6: Smectic-A mesophase** - The layers behave like a bidimensional liquid, while the interlayer spacing presents long range order.

Smectic phases (from Greek, *soap*) owe their current denomination to their mechanical properties apparently similar to surfactants. As mentioned already above, they are mainly formed by elongated, rod-like mesogens and they start forming at colder temperatures than nematics. The characteristic trait of smectic phases consists of the presence of a layered structure with distinct interlayer spacing, so that smectics can be considered more ordered with respect to nematics. There are many types of smectic phases, some of which are not mesophases [1, 29, 33]. Smectic phases can be seen in both chiral and achiral mesogens, producing different types of molecular arrangement. However, among the mesophases, the Smectic-A, Smectic-C and Smectic-B hexatic are the most recurrent. The first smectic phase that has been discovered is the Smectic-A, shown schematically in Fig. 5.6. In
this mesophase the layer stacking is periodically ordered, but the mesogens in each layer behave like a 2D liquid. Because of its nature, Smectic-A is not susceptible about the chirality of the mesogens.

Figure 5.7: Smectic-C mesophase - The green arrow indicates the direction of layer stacking. The red arrow represents the director of the mesogens, clearly tilted respect to the layer stacking direction. If mesogens are chiral, then the helical distorted Smectic-C* will appear.

The smectic-C (see Fig. 5.7) differs from smectic-A because the mesogens are tilted of about the same angle around their center of mass. In this case the chirality becomes important, because chiral molecules will distort the mesophase by correlating with the mesogens in the adjacent layers and forming helical structures, reminiscent of a cholesteric phase. Upon decreasing the temperature of the liquid crystal, the next phase found is the Smectic-B hexatic. The distinguished character of Smectic-B hexatic is that layers still behave as a two-dimensional liquid, but the mesogens present local hexagonal interlayer packing. Such local intralayer order can be observed both experimentally with X-ray crystallography and theoretically from MD simulation, respectively with the reciprocal space analysis of the X-ray diffraction pattern and the diffraction pattern of the simulated structure, as described in chapter 3. As for the previous smectic mesophases, the layers of the Smectic-B hexatic are stacked periodically and exhibit long range uniaxial order.
5.4 Columnar Phases

Columnar phases are commonly observed as thermotropic mesophase formations and the most common is the hexagonal columnar phase \[1\]. The hexagonal columnar phase is characteristic of molecules with discotic topology (see Fig. 5.9). The columnar phases are 2D periodic, as the discotic mesogens are hexagonally ordered in the plane orthogonal to the elongation axis of the columns. A schematic representing this molecular arrangement is shown in Fig. 5.8. The mesogens in each column however are free to move along the axis of the column (intracolumnar self-diffusion). Much...
more seldom some mesogens can diffuse to the adjacent columns, however with a negligible ratio, if compared to the intracolumnar self-diffusion.

Figure 5.10: Columnar hexagonal mesophase - This columnar phase has been observed in MD simulation of a 3D, one-component system of particles interacting via isotropic Dzugutov potential [25].

5.5 Conclusions

It is still commonly believed in the scientific community that smectic and columnar mesophases form through entropy-driven self-assembly. In other words, it is commonly believed that self-assembly is achieved because the anisometric topology of the mesogens is responsible for the anisotropic nature of the interactions between mesogens. The research presented in this thesis demonstrates how the anisometry of the mesogens is negligible for the anisotropic arrangement typically observed in columnar hexagonal mesophases [25]. Another interesting aspect highlighted by the included papers is that the columnar mesophase can in principle be formed in colloids.
6. Smectic-B phases with $\geq 3D$ periodicity

6.1 Smectic-B quasicrystal

![Smectic-B quasicrystal](image)

**Figure 6.1: Smectic-B quasicrystal** - A 3D representation of two adjacent layers of the smectic-B quasicrystal [29]. Note the $30^\circ$ rotation of each visible layer respect to its adjacent.

The formation of the smectic-B quasicrystal has been recently observed in a molecular dynamics simulation of a 3D system of identical particles, interacting via a spherically symmetric pair potential [29]. The formation of this quasicrystalline smectic phase is the most important result reported in this thesis, because this phase was never observed experimentally, nor has it been predicted theoretically. This exotic structure presents twelve-fold symmetry obtained through the $1:1$ periodic stacking ratio of ordered, flat layers, rotated respect to each other of $30^\circ$ (see Fig. 6.1), where in each
layer the particles are hexagonally ordered. This arrangement produces an incommensurate rotation along the axis parallel to the layer stacking direction. It must be remarked that the simulated structure is an *approximant*. This means that the simulated structure is indeed a crystal, but this is an artifact due to the usage of periodic boundary conditions. One of the most important operations in the analysis of a quasicrystal is the determination of the tiling. From the tiling (see Fig. 6.2) it is possible to understand if the phason flipping occurs in the structure [54,55]. The presence of phason disorder in the tiling to break the long range periodicity is fundamental to assess that the quasicrystal is stable. It should be remarked that this structure differs substantially from the only one smectic quasicrystalline structure known before: the twisted grain bound-

![Figure 6.2: Smectic-B quasicrystal tiling](image)

*Figure 6.2: Smectic-B quasicrystal tiling* - The tiling has been analyzed on two adjacent layers of the quasicrystal [29].
6.2 Smectic-B crystal

For many years the smectic-B crystal (see Fig. 6.3) has been considered as a mesophase; however, it is clear nowadays how such classification is not correct. As its denomination suggests, the smectic-B crystal is a true crystal, with 3D long range periodicity. What confused scientists in the beginning was that this crystalline phase can be obtained upon cooling of thermotropic mesogens [1].

More in detail, the result reported in the included paper [33] describes the observed formation of the smectic-B crystal in a MD simulation of a 3D system of identical particles, using a spherically symmetric pair potential. This is the first low-density self-assembled structure ever reported with such in-
vestigation method.

6.3 Conclusions

The biggest achievement of the reported research is with no doubt the prediction of a new state of matter: the smectic-B quasicrystal. The second important scientific achievement about the work done on the formation of the smectic-B crystal was to demonstrate that neither the anisometry of the mesogens, nor the anisotropy of the interparticle potentials are necessary to produce anisotropic smectic phases. These findings are indeed of fundamental importance for the advance of understanding self-assembly in low-density condensed matter. Because of the generic, simple nature of the physical models used, this research suggests that smectic phases and smectic quasicrystals can be found in colloids [29][33].
7. Metal Organic Frameworks

Metal Organic Frameworks (MOFs) are porous, crystalline, supramolecular structures formed by the coordination of organic ligands and metals (including metal oxide clusters). These materials are believed to form the supra-molecular structure by self-assembly, because their topologies are not directly related to the chemical nature of the molecular building blocks involved in the formation of the supramolecular self-assembled structure. This means that chemically unrelated MOFs can possess the exact same topology (isoreticular). One of the papers included in this thesis reports the formation of an archetypal MOF structure observed in a MD simulation of a 3D system of identical particles interacting via a spherically symmetric pair potential \[24\]. The resulting crystalline MOF has found to have a new topology net called the \textit{alm} net \[56\]. The process of self-assembly from the isotropic liquid phase is shown in Fig. 7.1.

![Figure 7.1: Self-assembly in MOFs - MD simulation snapshots of \textit{alm}-MOF \[24\]. a) Isotropic liquid - b) Intermediate unstable phase - c) \textit{alm}-MOF crystal.](image)
7.1 Coordination chemistry

![Mechanism of coordination according to Molecular Orbital theory](image)

**Figure 7.2: Mechanism of coordination according to Molecular Orbital theory** - *a*) Negative charges form a specific coordination geometry around the metal cation. *b*) The d-orbitals are the ones usually involved in coordination. For lanthanide and heavier elements the f-orbitals will be also involved. $e_g$ orbitals possess higher energy than $t_{2g}$ orbitals. *Image courtesy of internet*

The chemical nature of MOFs as coordination complexes is probably the main reason for the existence of isoreticular metal-organic, porous structures. Coordination complexes are those chemical species formed from a transition metal atom or ion, which acts as an electron acceptor (Lewis acid), and from one or more molecules acting as electron donors (Lewis bases), which are defined as ligands. Fig. [7.2] shows the distribution of charges around the metal cations according to Molecular Orbital theory [57]. The atoms of the ligands involved in the coordination will displace around the metal atom with a very specific geometry. Each ligand can have one or more
atoms coordinated to the same metallic center. The number of atoms of the ligand molecule involved in coordination determine its denticity. For example, a ligand with one electron donor is defined as monodentated, with two electron donors bidentated and with many electron donors polydentated. Each metallic center can bind a variable number of electron donor atoms of the ligands, depending on the level of saturation of the $d$-orbitals for the transition metals or of the $f$ orbitals for the lanthanide and heavier metals. The amount of atoms from which a metal can accept electrons is defined as coordination number. Changes in the oxidation state of the metal induces a change in its coordination orbitals saturation, so the same atom can accordingly exhibit a different coordination number. Because of the non-covalent nature of the coordination bonds, the coordination complexes are commonly defined as supramolecular self-assemblies. When a single group of ligands saturate their electron donor atoms by capping a finite number of metal atoms or inorganic clusters, the result is the formation of a finite, supramolecular complex. A ligand could eventually bridge two or more metal centers and this makes possible to obtain a supramolecular infinite network, defined as coordination polymer [58]. Depending on how many spatial dimensions the structure elongation occurs, coordination polymers exhibit frameworks with different dimensionality, ranging from linear to three-dimensional. For those materials possessing a three-dimensional framework, it often occurs that the structure of the coordination polymers exhibit long-range order, however this is not a necessary condition.

7.2 Properties, classification and uses of MOFs

Even if the study of coordination complexes dates back to more than 120 years ago, it is just as recently as few decades ago that two new branches of coordination chemistry appeared: Metal-Organic Frameworks (MOFs) and Supra-molecular Coordination Complexes (SCCs). They both comprise coordination complexes formed from metals and organic ligands and they can be summarized as Metal-Organic Materials (MOMs). SCCs are discrete systems in which well defined metal centers self-assemble with organic ligands oriented with specific angularity to produce finite supra-molecular complexes. MOFs are periodic networks of metal centers, or inorganic clusters, bridged by organic linkers via coordination bonding. Both the metal centers, or inorganic clusters, and the organic ligands are called Secondary Building Units (SBUs) [59] as an attempt to establish a structural analogy with zeolites. Some examples on how to extrapolate the SBUs from the molec-
ular building blocks are shown in Fig. 7.3. A core feature of most MOFs is that they are low density micro-porous (pore size up to 2 nm) crystal structures, where pores are permanent and with extremely narrow pore size distribution. MOFs are often used in the energy industry for their efficiency in the physical and chemical adsorption of small and large molecules, for example hydrogen, methane, carbon dioxide and others [60-64]. Because of the numerous applications in which adsorption is useful in the industry, porous crystalline materials are of great interest. The field of porous materials has been dominated for many decades by inorganic compounds known as zeolites. MOFs, for many aspects, are more versatile than zeolites. For this reason, trying to replace zeolites with MOFs exhibiting same functionality is a very popular research topic.

![Figure 7.3: Derivation of the SBUs - a) Molecular Building Blocks - b) SBUs. The orange spheres represent the vacancies inside the molecular structures of the MOFs. Image courtesy of KAUST](image)

Another interesting aspect of research concerns the rational design of MOFs, in order to tailor them for specific applications. MOFs have applications in different areas of the industry in function of the combination of their geometric properties, chemical properties or physical properties. MOFs geometric properties of interest are pore size, channel system and framework dimensionality. Chemical properties of MOFs are relative to the functionalization of their porous surfaces and selectivity and reversibility in chemical
adsorption. Physical properties are selectivity and reversibility in physical adsorption, thermal stability and mechanical strength. However, the first goal to achieve in rational design of MOFs is the synthesis of a real structure having previously chosen arbitrary geometric properties. Even if this ambitious goal has not yet been achieved completely, important advances have been made in the field \cite{59,65,66}. Topology can be defined as the study of those features that are preserved upon the deformation of a system. It has been experimentally observed that MOFs synthesized by completely uncorrelated chemical precursors can exhibit the same topology \cite{62,67}.

7.3 Reticular Chemistry

![Reticular chemistry](image)

**Figure 7.4: Reticular chemistry.** - Example topological nets. These are the chemical templates to design realistic, tailored structures \cite{68}.

In crystallography, the concept of describing real crystal structures as specific mathematical graphs, the nets, is well established. Mathematical graphs are geometric representations comprising points connected by lines. When the lines in a graph present a direction, the whole graph is defined as directed. Graphs can be infinitely periodic or finite. Periodic, directed or undirected graphs are defined as nets in crystallography. A net represents the topological structure of a real material. However, as for real crystal structures, nets present a unit cell. The unit cell is defined in crystallography as the smallest repeating unit of a certain configuration in space. The unit cell is obtained by applying certain symmetry operations to a set of vertices and connecting edges defined in a three-dimensional space. Such vertices and edges, and the smallest region of space able to contain them, are defined as asymmetric unit. The symmetry operations identify the space group of the
asymmetric unit. Each net can therefore be defined by its own symmetry group and asymmetric unit parameters. The study, classification and character of MOF nets are the foundation of reticular chemistry [69]. To refresh, MOFs are coordination polymers formed by metal SBUs, bridged by organic linkers SBUs. A MOF net arises from the periodic repetition along the 3D translational axes of a self-repeating pattern formed by a combination of those graphs corresponding to the topological nets of the metal SBUs and the organic linker SBUs [59]. The graphic notation of nets used in reticular chemistry is derived from crystallography. MOFs exhibiting the same net are called isoreticular (see Fig. 7.5) [62]. The classification of SBUs and topological nets became increasingly more demanding, because of the large number of newly discovered compounds and because of their topological complexity. For this reasons, Michael O’Keeffe and his collaborators have created a publicly available internet database called Reticular Chem-

Figure 7.5: Example of isoreticular MOF structures - Their chemical properties are uncorrelated, but their topology is the same. That is why they are called isoreticular. Image courtesy of Omar Yaghi Research Group
This web resource is of core importance for the design of the desired MOF topology, which is the very first step of MOF rational design. The RCSR database contains topological information about all the known, publicly available nets of full MOFs, both experimentally synthesized or hypothetical, SBUs and linkers. The importance of this database however is not simply limited to the first stage of MOF rational design and neither is just limited to the classification of MOFs. When a new MOF structure is synthesized experimentally or obtained in-silico, the RCSR is an indispensable tool for its topological characterization. Porous MOFs are low-density structures where the pores often result from the expansion in space of the unit cell. Low-density structures are characterized by having a small number of nearest neighbors for each atom.

7.4 MOF structure discovery

The MOF discovery process takes its foundation from drug discovery, a common practice in the pharmaceutical industry to obtain novel drugs able to act on a specific molecular target or biological mechanism, so to cure a specific illness. The MOF discovery process can be subdivided in different stages: net design, net decoration, real structure screening and real structure selection. For many years the net design has been performed by creating nets via analytical geometry, usually with the help of some visual crystallographic software. A net can be manually obtained by defining a space group, unit cell size and vertex positions in the asymmetric unit. Net decoration is performed by further decoration of such nets with the real atomic structures of SBUs and linkers. The result is a list, or library, of different realistic MOF structures. This library is then reduced by deleting the surely unwanted structures from the list. From this point and on, the synthesis of such MOFs is attempted. This procedure is known as screening. If the screening is operated on a large number of compounds, say between few thousands to millions compounds, it is defined as high-throughput screening (HTS) [60]. HTS is usually highly automated, with robots capable of starting few hundreds to few thousands syntheses per cycle. The products are then quickly characterized and either saved or discarded. The successful syntheses achieved in stage of screening will lead to a collection of real products, usually referred as product library. Each product is characterized with X-ray diffraction, thermogravimetric analysis (TGA) and gas adsorption. Further characterization might be necessary in order to select from the structure library those materials that satisfy the initially chosen requisites, for example, efficient carbon dioxide chemical adsorption. Because of the
strong combinatorial nature of the methods adopted in MOF structure discovery, automated, computer-aided net design and decoration software tools have been quickly created and adopted. One of the most popular software projects for topological net design is called GAVROG (Generation, Analysis and Visualization of Reticular Ornaments using Gavrog). GAVROG project includes two tools: SYSTRE and 3dt. SYSTRE stands for SYmmetry, STructure (REcognition) and REefinement. With SYSTRE it is in fact possible to generate or fetch from the RCSR a wanted net. Typical criteria useful when designing or querying for a net are its symmetry, coordination number of nodes, topology of SBUs and linkers. The output of SYSTRE is one or more files containing those nets presenting the specified features. The utility 3dt is a tiling visualization program, complementary to SYSTRE. Another important tool which is gaining more and more popularity in the design of nets is a software suite called TOPOS. TOPOS is a valid tool also for automated net structure decoration. This software package is able to take one or more nets as an input and return as an output a set of files (a library) containing atomic structures of isoreticular MOFs, respect to each of the input nets. The algorithms adopted are based on non-trivial, pure geometric calculations. Because of the very high costs of performing HTS in-vitro, it has been theorized the possibility of performing a virtual HTS (vHTS). The ultimate purpose of vHTS is to substitute entirely the traditional HTS made in-vitro and produce a library of physically sound, stable structure candidates, with known synthesis methodology. One of the most difficult challenges in vHTS research is the accurate prediction of the structural stability of the MOF material.

7.5 Molecular dynamics simulation

Atomistic, classical molecular dynamics simulation of MOFs is not a common investigation method because of the poor reproducibility of the quantum interactions between metal ion and its coordinated ligands (especially for heavy ions), which often results in non accurate interatomic potential parameters. The simulated systems are very simple MOF structures, like MOF-5 or CuBTC-MOF. The atomistic MD simulations usually try to predict the chemical behavior of MOFs with respect to their adsorption properties, rather than investigating the process of self-assembly. In this sense, the approach proposed in the included papers represents the first theoretical prediction of the self-assembly of a MOF structure using classical molecular dynamics. The total absence of chemical order in the simulated system represents a significant advantage in the characterization of the structure,
because of its remarkable similarity with its topological net (which was also found to be novel, see Fig. 7.6 [24][56].

![Figure 7.6: The alm net - Screenshot of RCSR database web page.](56)

7.6 Conclusions

The research work about MOFs reported in the papers included in this thesis represent a significant advance in the understanding of self-assembly mechanism in MOFs. Another important result was the prediction of a novel topological net. Respect to the traditional, analytical (geometric) method for producing novel topological nets, the method reported in this thesis represents the advantage of producing thermodynamically stable nets.
8. Conclusions

The presented scientific contribution should be seen merely as the beginning of a new research field, based on the study of self-assembly of low-density condensed matter using simple, one-component systems of particles and spherically symmetric pair potentials like the Dzugutov potential. Probably the most important work reported is the one describing the finding of a novel type of smectic phase, namely the smectic-B quasicrystal, which was not even thought to exist. Definitely relevant for the soft matter community has been the demonstration that for both the formation of smectic and columnar phases (either crystalline or quasicrystalline) the anisometry of the molecule does not influence the anisotropic self-assembly process. Another important addition to human knowledge has been to demonstrate how in principle supramolecular self-assembly and formation of colloidal structures obey the same, simple physical laws. The usage of a toy model for the mechanical description of the simulated systems, has been a key feature for producing such general, fundamental results.

8.1 Future perspective

One of the most wonderful features of the scientific discoveries is that they seem to be endless. A very clever person once said: "When you look at science as a beginner, science appears like a far, impenetrable wall made of solid bricks. The more that you learn, the more you approach that wall. However, as you come closer to it, the impenetrable look of the wall will progressively fade out, while revealing plenty of empty holes through which you can pass". The study of self-assembled, low density condensed matter is probably the last frontier of crystallography. This new branch of scientific research is young and mostly unexplored, so it offers clear possibilities for future investigations.
9. Popular Science

Computer simulation is a very important scientific investigation tool in different fields of science. It allows the exploration of physical conditions which are difficult to measure experimentally. When appropriately performed, computer simulation can predict the behavior of physical systems. For this reason it is commonly used in many different fields of science and engineering. For example, in the production process of cars, airplanes, trains, and vehicles more in general, the safety of the passenger is initially estimated with a simulated crash test. This allows a substantial decrease in the production costs, which directly reflects in a cheaper final product.

Similarly, weather forecast allows people to better prepare themselves against the atmospheric conditions, which in some parts of the world, including Sweden, can be extreme, is performed by computer simulations. Without weather forecast, the yearly incidence of death and damage to people and our beloved pets would substantially increased.

Computer simulations are fundamental in the process of drug discovery, which is performed within the pharmaceutical industry to discover and perfection the drugs we take when we are sick. Without computer simulation we could not find the cures to diseases and illnesses as fast as we do now. This would eventually result in increased population mortality.

Additional to these very practical applications, computer simulations can be applied in apparently more abstract, which are instead very concrete and important fields. All of the scientific findings used in technology to improve our everyday life derive from more abstract studies, commonly referred as basic scientific research. Basic research generates the building blocks used in technology to make our life nicer, longer and safer.

The interest of the author of this thesis concerns the basic scientific research in low-density condensed matter physics, which consists in the study of the physical laws, at a microscopic level, behind the formation of certain ordered structures in low-density matter either in the liquid state, the solid state or any of the liquid crystal states in between. Matter is composed by tiny elements called atoms, which arrange into molecules, which are always found in a certain physical state, called more technically phase. Some of these molecules interact with each other without changing their struc-
ture. In this case, they could arrange themselves to form certain structures, known as supramolecular structures. The driving force for such supramolecular structures does not seem to be induced by any external help. For this reason, this mechanism of formation is known in physics as *self-assembly*. Since self-assembly is common to many, very different types of molecules, the approach taken by the author to the study of this formation mechanism wanted to be as general as possible. Hence, only very simple models have been used, as essential as they could possibly be to describe the phenomenon. The system was a 3D set of points (particles) in space with initial position and a certain random initial velocity, which have been assumed to interact under the effect of a certain *potential energy* arising from the interaction of each particle with the other neighboring particles in the system. Only pair-wise, interparticle interactions have been considered, assuming the same behavior independently from the orientation of the particle pairs in space (spherically symmetric potential). Such research approach has turned out to be very successful and led to the publication of several scientific papers. The major scientific discovery achieved in this research work has been the theoretical prediction of a peculiar phase arranged in layers, so that the layers are stacked like a pile of sliced sandwich bread, where each slice is rotated with respect to the adjacent one of 30 degrees. Each layer is perfectly straight and composed of particles forming a honeycomb pattern (hexagonal packing). Such arrangement has been denominated Smectic-B quasicrystal, because the formed pattern is not exactly repeatable infinitely in space, opposed to what happens for a regular crystal (hence quasicrystal), and it is reminiscent of a layered structure common among liquid crystals known as Smectic-B. A smectic-B crystal has also been predicted successfully. Another important result has been the formation of a porous crystal classifiable as MOF (metal-organic framework), with a specific, new conformation never seen before. Finally, the discovery that a certain liquid crystal phase known as columnar hexagonal mesophase has been predicted. The results are common to molecules that have non-spherical, elongated, rod-like or disc-like shapes and the phases that they form were thought to be mainly due to the anisotropy (not spherically symmetric) of their interactions with each other. In conclusion, the whole research presented in this thesis suggests an even more important and general result, which is that self-assembly occurs independently of the shape of the molecules, at least for molecules capable of forming smectic and columnar liquid crystals, and also porous crystals of the same class of the one discovered. Such conclusion can be made because the same phases were obtained using a physically sound method (molecular dynamics), using 3D systems made of identical particles interacting via a spherically symmetric potential.
10. Summary of Papers

10.1 Paper I. Formation of the Smectic-B Crystal from a Simple Monatomic Liquid

This paper describes the formation of a Smectic-B crystal phase, commonly seen among anisotropic rod-like mesogens, has been reproduced in a MD simulation of a 3D system composed by identical particles and interacting via an isotropic pair potential, proving that the anisotropy of the mesogens is not a mandatory requirement for the mechanism of phase formation. From a low-density isotropic liquid phase it has been observed, upon cooling, a first-order phase transition which resulted in the smectic-B crystal formation. The aspect ratio, measured between the interlayer and the intralayer distances, has been estimated as 1.8. An analogy can be made between the obtained result and a hypothetical smectic-B crystal system of hard spherocylinders (a commonly used model for the simulation of smectic phases) with the same aspect ratio. The layer stacking sequence of the reported phase is "ABCA...", which has been demonstrated to be possible experimentally. This is the first result demonstrating that 3D systems of identical particles interacting via a spherically symmetric potential can reproduce smectic phases. The result suggests that smectic-B crystal phase and in principle smectic phases can be also observed in colloidal systems.

10.2 Paper II. Formation of a new archetypal Metal-Organic Framework from a simple monatomic liquid

In this paper the formation of an low density porous crystal is reported as a result of a MD simulation of a 3D system composed by identical particles interacting via a spherically symmetric pair potential. The crystal has been structurally characterized and it has been established that it belongs to the MOF class. The novel structure also exhibits a novel topological net, alm, which has been included in the Reticular Chemistry Resource (RCSR)
database. Because of the absence of chemical order in the structure the obtained crystal has been defined as archetypal structure. The formation of such complex porous structure is extremely relevant as it confirms theoretically one of the fundamental assumptions of reticular chemistry, which is the independence of the topological arrangement of a porous structure from the chemical nature of the molecular building blocks composing it. Additionally such result suggests that similar porous structures can be formed by colloidal systems of spherical particles with appropriate interparticle interaction.

10.3 Paper III. Formation of a columnar liquid crystal in a simple one-component system of particles

This paper reports and discusses the result of a molecular dynamics simulation of a 3D system made of identical particles and interacting via a spherically symmetric pair potential. This result demonstrates that even though columnar mesophases are observed in real world with discotic molecules, the mechanism of formation (self-assembly) is not related to the topology of the mesogens. The mesophase formation was observed upon isochoric cooling from a low-density isotropic liquid state. The phase transition is of first order and the result was the sudden formation of a liquid crystal phase composed by the particles arranged as liquid columns producing an hexagonal patter in the plane orthogonal to the column elongation axis. The liquid columns exhibited a remarkable intracolumn self-diffusion, as confirmed by the mean square displacement along the column elongation axis versus the mean square displacement in the plane orthogonal to the column elongation axis. The structure was then cooled further, resulting in another first order phase transition, so that, additionally to the periodic hexagonal stacking already observed in the mesophase, the liquid columns became periodically ordered, transforming the 2D-periodic mesophase in a full 3D crystal. This result is important for being the first liquid crystal phase ever observed in a 3D system of identical particles interacting via a spherically symmetric pair potential. The findings of this paper also suggest that the columnar hexagonal mesophase can be formed by mesoscopic soft-matter and colloidal systems with spherical particles, provided the usage of an appropriate interparticle potential.
10.4 Paper IV. A new kind of soft-matter quasicrystal

This paper reports the formation of a novel smectic-B quasicrystal phase, which has never been observed, nor theorized before. The result has been observed in a MD simulation of 3D identical particles interacting via a spherically symmetric pair potential. Before the reported finding, the only quasiperiodic structure ever observed in smectic phases has been the twist grain boundaries (TGB) structure, which consists in the commensurate helical rotation of smectic slabs around the axis parallel to the layer plane. The reported structure consists of layers of particles with intralayer hexagonal close packing, where the layers are rotated respect to each other, along the axis parallel to the layer stacking direction, of 30 degrees, with ABA... stacking. Because of the similarity of the potential parameters used to produce the quasicrystal with the ones used to reproduce the Smectic-B crystal, it is believed that in principle this quasiperiodic phase occurs in systems that can form smectic phases. The isotropic nature of the potential also suggests that the phase can be possibly observed in colloidal systems with spherical particles and opportune interparticle potential.

10.5 Paper V. A new computer program for topological, visual analysis of 3D particle configurations based on visual representation of radial distribution function peaks as bonds

This paper describes a new computer program which has been created for the visual analysis of systems of particles. The program, PASYVAT, has been created in response of the need of a new type of structural analysis which is absent in the previously existing software tools for the analysis of systems composed by 3D position vectors representing systems particles used for MD simulation. Its novel, main functionality consists of the possibility of selecting from a radial distribution function graph \( n \) distinct interparticle distance intervals, representing them as bonds in a 3D window showing the particle system. For each interparticle distance interval selected, a bond color is assigned, so that the resulting bond network will produce well defined geometries, easy to identify in the case of analyzing ordered structures (like crystals). Additionally, a structure slicing tool has been added, to let the user interactively dissection the particle system, highlighting eventual features of interest, like single layers of particles. The program is unique
for its radial distribution function based bonding tool, but it also presents a quite innovative, modular software structure, which allows people with basic programming knowledge to expand or modify the program to better fit their needs. The program has been indispensable in the understanding of the topological net underlying the alm-MOF [24].
Acknowledgments

I want to begin by acknowledging the supervisors of my doctoral studies, and close friends Professor Peter Oleynikov and Professor Mikhail Dzugutov for their indispensable, flawless support.

I would like to thank Professor Berk Hess and Professor Erik Lindahl for giving me the possibility to collaborate with them to the GROMACS software development. A special thanks also goes to Dr. Mark Abraham and Szilárd Páll for their extremely valuable support. Working with you guys has been and still is an exceptionally great experience!

Thanks to my co-authors, especially to Dr. Tomas Oppelstrup, who has been always very supportive, Professor Michael O’Keeffe, and Dr. Sten Sarman.

From my department, MMK, I would like to acknowledge in particular the prefekt Professor Gunnar Svensson and the director of the doctoral studies Professor Arnold Maliniak, who gave their best support to finalize my doctoral study career. Thanks also to the administrative officers of MMK for their work. My acknowledgments also must especially go to Professor Niklas Hedin, Professor Alexander Lyubartsev, and Professor Lennart Bergström for their illuminating suggestions and the interest that they expressed in my activities.

Thanks to my friend Professor Jonas Lindemann, and the sysadmin staff at Lunarc for their prompt support with the GPU cluster which I used the most during my last stage of doctoral studies.

My acknowledgments must also include Dr. Gert Svensson, Michael Schliephake, Dr. Jonathan Vincent and the staff at PDC for their exceptionally valuable support and very welcome free hardware donations.
Thanks to my elite friend and colleague Dr. Arto Ojuva and my wife Viviana Reverón for revising the language my thesis.

Thanks to my friend and valid colleague Lorenzo Agosta for useful physics discussions and tons of espressos.

10.6 Personal acknowledgments

First, I want to thank Viviana Beatriz Reverón Zozaya Coimbra de la Coronilla y Azevedo, my wife, for her priceless support and patience during this experience.

I would also like to thank my parents Antonio Metere and Paola Mura, my brothers Roberto Metere, Riccardo Metere, Irene Metere and Antonio Papapietro. A special, warm-as-Venezuela thanks to my wife's family, especially Jaime Reverón, Vilma Zozaya, Veronica Reverón, Antonio Reverón, Pedro Itriago and Miriana Morales.

A special thanks for the brotherhood to my closest friends, whom I consider as part of my family:

- Professor Gaetano Verani
- Dr. Mikhail Voznesenskiy
- Dr. Arto Ojuva
- Dr. Luca Deiana
- Dr. Iwan Zimmermann
- Dr. Ladislav Benda
- Dr. Jesper Lindkvist
- Nathalie Gonska
- Daniel Schlesinger
- Dan Wickström
- Christopher Whitman
- Matteo Trincas
- Tanja Laine
- Federico Conticchio
- Georg Ekblad
- Arduro Mazzabrutta Bjarnason
- Erik Beard Lindström
- Elizabeth Castaldo
- David Castaldo
- Pierpaolo Frau
- Mauro Leoni
- Aurelio Raspino
- Maurizio Demurtas
- Daniele Casu
- Emmi Cook
- Arjun Ray

I would like to thank more in general my friends for their awesomeness
and to make my life easier and nicer. I would like to mention you all and in detail, but apparently is not very formal, so I need to shrink the list to the ones with whom I shared and keep sharing the experiences in my life. Within this context, I would like to mention:

- Alberto Pibiri
- Ana Lucia De Bastos
- Andrea Zidio
- Anni Golz
- Antoinette Bagge
- Arnaud Mayence
- Barbara Schmauss
- Christina Vitale
- Dr. Axel Kohlmeyer
- Elisa Alonso Aller
- Filippo Bombaci
- Hector Da Silva Soares
- Jonas Wallenius
- Eidi Alvarado
- Juan Carlos Redondo
- Juan Francisco Alonso
- Juste Ula
- Katrin Schmidt
- Kess Marks
- Laura Santa Cruz
- Lisa Carmelita
- Marcin Skwark
- Agnieszka Skwark
- Marcin Kalek
- Marco Grongo
- Marco Salvatore
- Enrichetta Mileti
- Mariana León
- Lars Millevik
- Marian Grocky
- Marin van Regteren
- Massimiliano Vatiero
- Michelle Perry Cuchera
- Miguel Salas
- Minttu De Marothy
- Professor Giuseppe Saba
- Professor Jan Lang
- Professor Jim Haile
- Rodolfo Osmo
- Sara Massidda
- Shirin Rahmanian
- Simone Aresu
- Sophie Louise E. Olausson
- Stefan Fleischmann
- Vanessa Itriago
- Vanessa Quero Suárez
- Walter Basile
- Wilian Zurita
- Yayi Hernández

Stockholm, 2015-03-31. Alfredo Metere
References


[34] 2015. [link]


NVIDIA. Tuning CUDA Applications for Kepler v. 7.0. 2015.


NVIDIA. CUDA C Programming Guide v. 7.0. 2015.


The alm net in RCSR database.


