UNIVERSITÀ DEGLI STUDI DI NAPOLI "FEDERICO II"

Dipartimento di Ingegneria Chimica

Dottorato di Ricerca in Ingegneria Chimica

XX Ciclo

Molecular Dynamics of Triglycerides: Atomistic and Coarse-Grained Approaches

Tesi di Dottorato

Comitato Scientifico

Ch.mo Prof. Silvestro Crescitelli (*relatore*)

Ch.mo Prof. Nino Grizzuti

Ch.mo Prof. Pier Luca Maffettone

Ch.mo Prof. Giuseppe Milano

Candidato

Ing. Antonio Brasiello

Coordinatore del Corso di Dottorato

Ch.mo Prof. Pier Luca Maffettone

Anno Accademico 2006–2007

I would in that sweet bosom be
(O sweet it is and fair it is!)
Where no rude wind might visit me.
Because of sad austerities
I would in that sweet bosom be.

I would be ever in that heart
(O soft I knock and soft entreat her!)
Where only peace might be my part.
Austerities were all the sweeter
So I were ever in that heart.

— James Joyce (*Chamber Music*)

Ad Ilaria mia.

Parlate pure: il vostro abito vi dà diritto
di dire tutto quello che volete.
Ho studiato matematica, signor Galilei.
Questo può tornarci utile se vi induce ad ammettere
che due e due possono anche fare quattro.
— Bertolt Brecht [72]

Ringraziamenti

Giunto alla fine di questo periodo di intenso studio desidero ringraziare brevemente alcune delle persone che mi hanno accompagnato ed aiutato in questi tre anni. Dovrò essere breve, mio malgrado, per mancanza di tempo.
Ringrazio il mio comitato scientifico per quanto fatto in questi anni.
Desidero ringraziare vivamente il Prof. Silvestro Crescitelli e la Prof. Marisa Di Matteo per la preziosa opportunità accordatami. Vorrei esprimere tutta la mia gratitudine al Prof. Giuseppe Milano per come ha saputo guidarmi nel mio lavoro e per il tempo che mi ha dedicato. Ringrazio inoltre Giuseppe Santangelo per i preziosi consigli; Erasmo, Lucia e Pietro per aver reso, con la loro compagnia, più piacevoli le giornate di duro lavoro.
Non finirò mai di ringraziare inoltre genitori, fratello, parenti ed amici tutti[1] che mi hanno sopportato e *continuano* a farlo.
Infine ringrazio Ilaria, a cui dedico questa tesi, per aver gioito con me nei momenti buoni ed avermi consolato in quelli meno buoni.

Grazie a tutti voi!

Seduto alla mia scrivania, come sempre in forte ritardo,
antonio brasiello

[1] sicuramente dimentico qualcuno ma non me ne voglia: ringrazio anche lui

Abstract

Molecular Dynamics of Triglycerides: Atomistic and Coarse-Grained Approaches

The objective of this thesis have been the development and the analysis of microscopic mathematical models to investigate the dependence of triglycerides conformations from environmental conditions. Triglycerides are important constituents of food products that show polymorphic solid transitions. Such behaviors influence greatly processes management involving fats mixtures. To investigate the relationship between macroscopic conditions and conformational induced properties, we dealt with microscopic mathematical models.

The first part of the thesis regards the building of a united-atoms model from which physical properties and structural distribution functions of a liquid phase were derived.

The second part regards the development of a coarse-grained model. The force field of such model was developed, by means of statistical tools, using suitable distribution functions derived from the atomistic one.

The coarse-grained model was used to perform numerical experiments to highlight the dependence of molecules conformations from experimental conditions.

The results of our simulations show a clear relationship between the conformational state of molecules and temperature annealing condition. Moreover, the improvement of order features through radial distribution functions was pointed out during several heating-cooling processes. The formation of small clusters of few planar molecules can be easily observed by means of visual inspection.

The effect of simple flow conditions imposed on the system was also investigated. Imposed flows increase molecules mobility causing a raise of planar molecules and a greater uniform distribution of inter-molecular orientational order vector.

Riassunto

Simulazioni Molecolari di Trigliceridi: Approccio Atomistico e Coarse-Grained

L'obiettivo di questa tesi è stato lo sviluppo e l'analisi i modelli matematici microscopici per lo studio della dipendenza della conformazione molecolare di trigliceridi dalle condizioni esterne imposte al sistema.

I trigliceridi sono dei costituenti importanti dei prodotti alimentari. Durante i processi di solidificazione, essi manifestano fenomeni di polimorfismo. Tali fenomeni influenzano molto la gestione dei processi industriali. Poichè si intendeva indagare la relazione tra condizioni macroscopiche imposte al sistema e proprietà conformazionali indotte, il nostro studio si incentrato sullo sviluppo di modelli matematici su scala microscopica.

La prima parte della tesi ha riguardato lo sviluppo di un modello matematico united-atoms da cui sono state ricavate proprietà fisiche e diverse funzioni di distribuzione relative a variabili strutturali di trigliceridi in fase liquida.

La seconda parte del lavoro di tesi ha riguardato la sviluppo di un modello matematico coarse-grained facendo uso di strumenti statistici di indagine. Per la costruzione del modello sono state prese in considerazione opportune funzioni di distribuzione derivate dal modello atomistico.

I risultati delle simulazioni mostrano una chiara relazione tra conformazione molecolare e condizioni di raffreddamento imposte. Inoltre è stata evidenziata, mediante opportune funzioni di distribuzione radiale, l'incremento delle proprietà di ordine del sistema nel corso di simulazioni cicliche di riscaldamento e raffreddamento. È stata osservata inoltre la formazione di piccoli *clusters* di poche molecole planari.

È stato investigato inoltre l'effetto sulle conformazioni molecolari di semplici condizioni di flusso. Il flusso imposto causa l'aumento della mobilità delle molecole producendo, conseguentemente, un incremento in media del numero di molecole planari ed una distribuzione più uniforme dei vettori d'ordine inter-molecolari.

MOLECULAR DYNAMICS OF TRIGLYCERIDES: ATOMISTIC AND COARSE-GRAINED APPROACHES

Antonio Brasiello

Contents

List of Figures

Chapter 1

Problem definition

1.1 Introduction

Fats are important constituents of many food products, in which often are present in a crystallized form. The crystallization behavior of fats is determined by their chemical compositions and processing conditions. An intriguing property of fats is that can solidify according to several crystal packing. It is the so called polymorphism[1]. Many experimental works have outlined the correlation between polymorphs and crystals shapes (see e.g. [3]).

Therefore polymorphism influences also the morphology of the crystals, a characteristic that belongs to a spatial scale of investigation higher than the atomistic one (concerning crystal packing). The polymorphic behavior of fat is related in most part to the high presence of triglycerides which are branch structured molecules.

Crystallization behavior of fats also influences the macroscopic properties and therefore the quality of the food products. For example, fat crystallization plays a critical role in determining sensorial properties and stability of confectionery coatings, dairy products such as butter and cream, vegetable spreads (e.g. margarine), and peanut butter [6].

A case in which the achievement of a suitable crystal phase is of primary importance is that of chocolate production process (as we will see in the following sections). One of the most important constituents of chocolate is, in fact, cocoa butter that can be considered as a mixture of triglycerides. High quality chocolate is a product characterized by cocoa butter with suitable crystal packing (one of the most stable). A wrong crystal phase influences negatively both the taste and the product shelf-life.

Crystallization of fats is important not only in the field of food process. A recent research [68] shows that an intriguing use of triglycerides in solid phases could be as a support in controlling proteins release materials. Ob-

[1] i.e. the existence of several crystalline phases of the same chemical composition that differ mutually in structure but yield identical liquid phases upon melting

viously, as the release mechanism depends on crystal forms, the analysis of triglycerides behavior is of fundamental importance in this field.

Despite of the lot of studies, the most of the mechanism of crystallization and polymorphic transitions is not still fully understood. From the experimental point of view this is due, substantially, to the complexity of the ramified molecular structures o triglycerides. Moreover, the analysis of crystal structures are complicated by the fact that, in all practical example, different types of triglycerides (differentiated each other by the length and the composition of their branches) are found together with mixtures of different kinds of fatty acids.

On the contrary literature in the field of numerical simulations is still meagre. This is due essentially to the fact that numerical techniques to the analysis of the phenomena regarding triglycerides molecules provide a very high computational effort.

Since in the last few years the computational resources were grown, a new route to the analysis of crystallization behavior of triglycerides was opened. The new route is represented by micro-scale modelling already applied in other fields of material science [43]. These technique in spite of high computational efforts seems to be promising to the theoretical comprehension of the dependence of nucleation mechanism from process conditions. Obviously, a primary condition is that with such methods someone should be able to perform the analysis of molecules conformations depending on environmental conditions as those strongly influence the nucleation process.

A wide literature on microscopic models exists. Regarding the spatial and temporal scales of description they can be grouped in several category going from ab initio calculations of few atoms to mean field models of long polymer chains. Such techniques, in some cases, have been successfully applied to analyse molecules conformations induced by environmental conditions. As an example we could cite the work of Marrink and Mark [38] in which a system constituted by phospholipid in water shows an hexagonal phase transitions varying temperature.

Surprisingly, literature about microscopic models of triglycerides is almost nonexistent. We could cite only the work of Chandrasekhar and van Gusteren [39] regarding atomistic simulations of triglycerides in a solid α phase. As we will widely discuss in the following, the microscopic model of Chandrasekhar and van Gusteren is not the most suitable to study triglycerides molecules conformations induced by process conditions because of the low temporal length scales reachable with such model.

The aim of this work is to investigate, through numerical simulations, conformational behaviors of triglycerides. For this purpose a microscopic coarse-grained model was developed. Such model, to our knowledge, will represent the first contribution in the comprehension of the mechanism of phase formations of such substances using microscopic approach. For this purpose, the model will be test analysing the behavior of a bulk volume of triglycerides

during temperature annealing. Moreover, the effect of simple flow conditions on molecules conformations will be conducted. Using this model, a greater increase of the amount of planar molecules , in the case of dynamic conditions, will be observed respect to the static ones. The planarity of molecules will be analyzed using order functions, suitably developed on our own, connected to precise molecule conformations.
In the next section (1.2) we focus our attention on chocolate quality development process as the conformational behavior of triglycerides molecules in this field plays a critical role in determining end product quality. However, the reported remarks about this topic have a wider significance, since they can be related to the most general topic of triglycerides crystallization.
Then, in section 1.3, we give an overview on triglycerides crystallization process and crystal growth dwelling upon polymorphism phenomenon.

1.2 A sample problem: Chocolate production process

1.2.1 Motivation

Chocolate is one of the most diffused products in the world, whose production constantly grow. It is a dispersion of small solid particles (cocoa powder, sugar and powder milk) in a continuous matter, a mixture of vegetable fats called cocoa butter [45]. The word *chocolate* has to respond to precise legal requirements.
Dark chocolate contains 30–35% of cocoa powder and 18% of cocoa butter. A high quality product contains 26% of cocoa butter.
The percentage of cocoa butter influences texture, gloss, color .
Common experience suggests us the existence of differences in quality of chocolate products. Several works have investigated about the reasons of such differences. We can cite as examples the papers of K. Frenke [46] and Bollinger et al.[47]. Bollinger et al. study the pre-crystallization of chocolate, comparing the difference caused by different flow conditions. The paper of Frenke on the other proposes a macroscopic model for the calculation of bulk crystallization, showing the possibility for optimizing the process with respect to expected surface and gloss hardness.
Chocolate production process influences dramatically the features of the end-product. As will be seen in the following sections, the complexity of production process reflect the complexity of the substance we deal with, which born from the presence of a continuous phase, the cocoa butter. Cocoa butter constituents (in great part triglycerides as will see), in fact, can solidify in several crystal structures called polymorphs, which influence greatly chocolate quality. There is not yet agreement about the number and the crystal packing concerning such polymorphs. This depends greatly from the fact

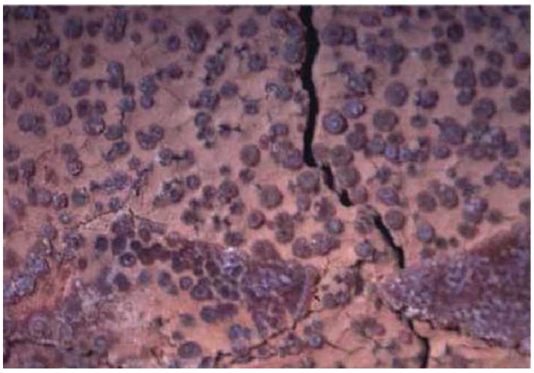


Figure 1.1: Fat bloom on a solid chocolate bar.

that the number of polymorphs and the type of crystal packing depends on the type and relative composition of the cocoa butter. Schenk and Peschar [48] for example, found at least five crystal structures.

The main problem that could be caused by a wrong crystallization process concerns the formation of *fat-bloom*. Fat-bloom is a physical defect which appears during chocolate storage and consists on a grayish or white appearance (i.e. large fat crystals formation, see figure 1.1) on the surface of the solid chocolate. Until now, the mechanism of formation is not clearly understood but even so it seems to be related to the polymorphism of cocoa butter.

From this, the interest of scientific community and chocolate confectionery on the comprehension of crystallization mechanism appear not surprising. The prospective is to improve product quality (e.g. product shelf-life as in [49] and [50]) and to optimize chocolate production process. This is the reason because we focus our attention on chocolate. In the following sections will emerge clearly the reason for which we decide to choose a microscopic modelling technique to analyze the solid phase formation of triglycerides.

The whole section, therefore, deal with the introduction of the topic of chocolate production. We have chosen this problem because it represents an technological example in which the importance of developing a theoretical model to study crystallization properties of triglycerides is of primary importance. After a brief history of cocoa and chocolate in subsection 1.2.2, useful to understand the evolution of production process, we describe a brief outline of modern process plants, in subsection 1.2.3. The aim is to focus the attention on specific unit operations in which is fundamental the use of a theoretical approach .

1.2.2 History

Let's start with some historical news.

Cacao trees were cultivated by Aztecs in Mexico long before arrival of Europeans. The beans were priced both for their use as a currency and for the production of a spiced drink called *chocolatl.* The beverage was considered to have aphrodisiac properties, a belief that still held as late as nowadays. The chocolate was prepared by roasting the cocoa bean in earthenware pots , before grinding them between stones. The mixture was added to cold water, often with other ingredients such as spice or honey, and whipped to a frothy consistency.

Cocoa beans were first brought to Europe by Columbus as a curiosity, but were later exploited commercially by Cortez as a new drink. Spaniards preferred their drink sweetened, and in this form its popularity spread to Europe. In 1606 chocolate drinking arrive in Italy. In 1615 reaches France. The introduction of milk into this chocolate drink was first recorded in UK in 1727 by Nicholas Sanders, although his reason for doing so are uncertain.

A mixture of ground cocoa beans and sugar does not by itself produce the solid chocolate so familiar to the modern consumer. Instead it gives a very hard substance that is not pleasant in the mouth. In order to enable it to melt easily, it is necessary to add extra fat. This can be obtained by pressing the cocoa beans and remove some of the fat content, known as cocoa butter. The ability to extract this fat was developed in 1828 by Van Houten of the Netherlands, and has a double advantage: the expressed fat was used to make solid chocolate bars, while the remaining lower-fat cocoa powder could stil incorporated into drink. This *drinking chocolate* was in fact usually preferred as it was less rich than the original high-fat mixture. Van Houten's development is even more remarkable is one considers that his factory and presses were ewntirely manually operated. In 1847, Fry, in Bristol, UK, used the recently developed steam engine to power the first factory to produce tablets of plain chocolate.

The solid form of milk chocolate is normally attributed to Daniel Peter of Vevey, Geneva, in 1876. In Switzerland, water-powered were able to operate for long periods at an economic rate. This enabled the extra water from milk to be driven out of the chocolate without incurring a large extra cost. Chocolates with moisture contents of above about 2% are normally unacceptable as they have poor keeping qualities as well as poor texture. Over the years many different flavors of both milk and plain (dark) chocolate have been developed. At other times flavor has been adjusted to complement the center of the sweet that is to be coated with chocolate. In 1988, world cocoa grindings exceed 2 million tonnes.

1.2.3 A brief outline of chocolate processing

Chocolate has to major distinguishing characteristics: its flavor and its texture. A primary feature of the texture is that it must be solid at a normal room temperature of 20–25 ^oC and then melt rapidly in the mouth at 37 ^oC, giving a liquid that feels smooth to the tongue. The processing of chocolate is related to obtain these two criteria and is therefore devoted either to develop the flavor of the product, since using a raw bean would produce a very unpleasant taste, or treat it so that it will flow properly and be largely free from gritty material.
Although many different methods of chocolate making exist, most traditional ones are based on the process briefly described below.

Preparation of cocoa nib: Flavor development

The cocoa tree produces pods containing a pulp and the raw beans. In the preparation of chocolate the outer pod is removed together with some of the pulp and the beans are fermented. This enables chemical compounds, which are the precursors of the flavor in the final chocolate, to develop inside the beans. Failure to carry put this stage properly cannot be rectified by processing at a later date. This is also true of the subsequent stage, when the fermented beans are dried. Poor control here can give rise to mould, which give a very unpleasantly flavored product even if the fermentation has been carried out correctly. In addition, correct transport conditions are required when the beans are moved from the country of growing to that of chocolate manufacture.

On arrival in the process factory, it is necessary to clean the beans to remove metal, stones and other extraneous material that might contaminate the product. Further flavor development is subsequently obtained by roasting the beans. This also loosens the shell around the outside of the bean and enable them to break more easily. The beans are then broken and the relatively lighter shell particles removed by a winnowing action . The presence of shell in the final chocolate is undesirable as it will impair the flavor as well as causing excessive wear to the subsequent grinding machine. It should also be noted that the shell content of chocolate is legally restricted in most markets.

Grinding: Particle size reduction

Up to this stage the cocoa is in discrete pieces, each several millimeters in diameter. Subsequent processing may take several forms, but all require the solid cocoa particles, sugar and any milk solids to be broken up so that they are small enough not to be detected on the tongue. The actual size depends on the type of chocolate and the market in which it is sold, but in general the vast majority of particles must be smaller than 40 μm. The most common method of grinding is by a five-roll refiner. In order to enable the chocolate ingredients to pass through the refiner, however, it is necessary to get them into a paste form. This may be done in a variety of ways. One of the most common is to grind the nib to form cocoa mass, which is a liquid at temperature above the melting point of cocoa butter, 35 ^oC. This usually involves hammer mills, disc mills, ball mills, three-roll refiners or a combination of all four. The sugar can then be added in a granulated or milled form, and the sugar and cocoa mass mixed with extra fat (and milk powder if milk chocolate is being manufactured). The mixing may include some grinding and traditionally melangeur pan was employed for the purpose. This machine has a rotating pan,often with a granite bed, on which two granite rollers rotate.

The modern requirement for continuous higher throughput methods has often lead to the mixing and grinding being carried out separately. The most widely used alternative method is to mill the solid ingredients (sugar,milk powder and or crumb) separately and then mix them with the liquid components (cocoa mass, cocoa butter and lecithin[2]) in the conche . This may result in different flavors from those obtained when all ingredients are processed together.

Conching: Flavor and texture development

Although the fermentation, drying and roasting processes are able to develop the precursors of chocolate flavor, there are also many undesirable chemical compounds present. these give rise to acidic and astringent tastes in the mouth. The object of conching is to remove the undesirable flavors while developing the pleasant ones. In addition, the previous grinding process will have created many new surfaces, which are not yet covered with fat. These uncoated surfaces prevent the chocolate flowing properly when the fat is in the liquid state. Because of this the chocolate cannot be used to make sweets and does not have the normal chocolate texture in mouth. The conching process coats these surfaces with fat and develops the flow properties,as well as modifying the flavor.

Conching is normally carried out by agitating the chocolate over an extended period in a large tank,known as conche. The mixing continuously changes the chocolate surface, and this, coupled with some heating and ventilation, enables the volatile compounds to escape and the flavor to be modified. Some manufacturers prefer to limit the conching time by restricting the conching process to primarily one of liquefying the chocolate. This is made possible by treating the cocoa mass at early stage, in order to remove some of the less desirable volatile chemicals.

[2] Lecithin is generally used as a surfactant

Tempering

Tempering is a further refinement process that lead to the obtainment of a best quality product. The tempering process consists of three parts:

- Proper de-seeding of chocolate.

- Precrystallization towards the correct seeds.

- Development or *maturation* of the seeds.

The aim to drive melted chocolate to the development of the desired properties, which can be divided in process properties (e.g. flow and viscous properties, end-consumer properties (e.g. texture, colour and gloss) and shelf-life properties(e.g. how to delay chocolate perishability). Traditionally, tempering process was conducted by hand, but by now this practice is almost disappeared. The industrial tempering technique are divided in *batch* or *continuous* processes. The last ones are the most commonly diffuse.
Tempering consists to impose suitable temperature changes and flow conditions, to produce the birth and growth of fat crystals of the desired type. It can be divided into four stages. First stage deal with the production of a melted chocolate completely free of crystal growth. This happens around 45^oC. The second stage is a gently cool, through a tempering machine that control cooling curve, to initiate the first stage of crystal growth. In this stage the runway solidification has to be prevented. The third stage is the reheat process through an heat exchanger. The aim of this stage is to allow the melting of the most unstable crystal seeds. The fourth stage, called the retention stage, the optimal periodic temperature oscillations and mechanical agitation produce the spread and the maturation of the most stable crystal seeds. The role of mechanical agitation is fundamental for the production of a high quality chocolate [42].

1.2.4 Cocoa butter

As can be seen from the brief outline on chocolate production processes given in the preceding sections, the main objective pursued during all the production phases is the development of the best chocolate quality, that consists in the development of a suitable solid phase formation both from the point of view of the workability and from the point of view of the organoleptic properties. A theoretical study of the solidification behavior of a complex matter like chocolate implies a system simplification. First of all we have to focus our attention on the continuous phase since only the molecules of such phase can form crystal packing. That's why in this section and in the following chapters, we focus our attention on cocoa butter or more precisely on triglycerides, the main component of it.

Cocoa butter is a fats mixture, most of them are triglycerides (triacylglycerol) (about $96\% - 98\%$, see [51]). Moreover, are also present free fatty acids, diacylglycerols, monoacylglycerols, phospholipids and other volatile compounds ([52] and [53]). In particular, free fatty acid (in most part oleic, stearic and palmitic ([54] and [55]) content is about $0.4\% - 2\%$. Diacylglycerol content is about 1.1%. Monoacyglycerols content is in pratical negligible and phospholipids content is about $0.1 - 0.8\%$ ([54] and [55]). Therefore, for our purpose, cocoa butter can be considered as a mixture of triglycerides.

Physical and chemical properties of cocoa butter depend greatly on chemical composition. The most important characteristic is, without any doubt, that it has a melting point at about $32 - 35^oC$, only few degrees under body temperature. It is important because allow food products, chocolate first of all, to be melted by the tongue, giving a soft taste. Obviously, melting point depends greatly also on crystal packing of triglycerides and therefore the next section is devoted to go into literature about triglycerides thoroughly. The attention will be focused on the polymorphic behavior of triglycerides and consequent polymorphic behavior of cocoa butter.

1.3 Solidification behavior of triglycerides

1.3.1 Crystallization: A general overview

Crystallization is a complex process not fully understood. It can be divided into several step, the main of which concerns with thermodynamic controlled nucleation process. Once nucleation processes happened, several growth processes take place on several length scales to form a crystal lattice[16].

Nucleation Nucleation is generally referred to as the formation of stable nuclei that are able to grow for a given set of external conditions.
Nucleation processes can be divided into two categories: primary and secondary nucleations. Homogeneous primary nucleation process happens when in a initial phase (e.g. a liquid phase) the presence of stochastic fluctuations in the local density and concentration promote changes in the free energy leading to the formation of atomic, or molecular, aggregates that are compatible with the structure of the original phase. If particles fluctuations lead to the formation of aggregates that can no longer be considered part of the original phase, because their organisation is that of the new phase, we have an heterogeneous primary nucleation. Often the heterogeneous nucleation

is catalysed by surfaces due to the presence of impurities[3]. Such heterogeneous nucleation is responsible of the birth of the so called embryos of the new phase. The embryos are formed as a result of atomic fluctuations which cause the continuous growth and shrinkage of embryos size. As an embryos reaches a critical size r_c, it become stable[4] and can be considered a *nucleus* of the new phase.

Theoretically, nucleation process is leaded by the difference in the Gibbs free energy, $G = H - TS$ between the initial and final states. At equilibrium between two phases Gibbs energy variation, dG, must be equal to zero. A transition occur when the difference in Gibbs energy between final and initial state, $\Delta G = G_f - G_i$, is less than zero.
The change in Gibbs free energy, ΔG_{hom}, caused by an embryos formation in the melt, is given by:

$$\Delta G_{hom} = -n\Delta\mu + \eta\sigma_{ls}n^{2/3} \tag{1.1}$$

in which $\Delta\mu = \mu_l - \mu_s$ is the difference between chemical potentials (or the free energy per atom) of the solid and the liquid bulk phases, η is a shape factor of the embryo, σ_{ls} is the interfacial free energy between solid and liquid and n is the number of particles condensed in the embryos. For a spherical embryos equation 1.1 become:

$$\Delta G_{hom}(r) = \frac{-4\pi r^3}{3v_s}\Delta\mu + 4\pi r^2\sigma_{ls} \tag{1.2}$$

in which v_s is the volume of the solid phase per atom. Increasing the radius of the embryos the equation 1.2 will show a maximum (that will correspond to the critical nucleus size). This maximum is responsible of the phenomenon of the *undercooling* of the system, that means that the system is able to maintain itself in a *metastable* liquid phase indefinitely at a temperature below freezing temperature. The critical size of the nucleus n_c and the corresponding critical value ΔG_c of ΔG can be obtained from the equation 1.1.

$$\Delta G_c = \frac{4\eta^3\sigma_{ls}^3}{27(\Delta\mu)^2} \tag{1.3}$$

$$n_c = \left(\frac{2\eta\sigma_{ls}}{3\Delta\mu}\right)^3 \tag{1.4}$$

The model described by the equation 1.1 take into account the homogeneous nucleation only. The heterogeneous nucleation happens when foreign particles are present in the system. Such particles stimulate nucleation that is

[3] This is the case of triglycerides in all industrial applications

[4] In literature a cluster whose size exceeds beyond r_c is normally referred to as a *critical nucleus*

therefore characterized by a smaller free energy of activation that in the homogeneous case. Indicating the liquid phase with a, the embryo solid phase with b (which is in contact with the surface S), the heterogeneous nucleation can be described through the following equation:

$$\Delta G_{het} = -\left(\frac{n_b r^3}{v_b}\right)\Delta\mu_{ab} + r^2\eta_{ab}\sigma_{ab} + r^2\eta_{aS}\left(\sigma_{bS} - \sigma_{aS}\right) \tag{1.5}$$

This equation can be specified for a segment of sphere that nucleates on a flat surface but this topic will not be treated here [5].

From the point of view of nucleation rate, theoretically the embryos are assumed to form through binary collisions. Therefore, the growth and the shrinkage of an embryos take place by adding or removing a single particle (i.e. atom or molecule). An embryo constituted by n atoms are usually named embryos of class n. According to hypotheses made, an embryo of class n can be formed by the growth of one of class $n-1$ or by the decay of one of class $n+1$. On the basis of the described mechanism can be shown that the number N_n of crystalline embryos of class n present in a liquid phase of N_0 particles follows the rule:

$$N_n = N_0 exp\left(-\frac{\Delta G_n^{hom}}{k_b T}\right) \tag{1.6}$$

where ΔG_n^{hom} is the Gibbs free energy difference corresponding to the formation of an embryo of class n and k_b is the Boltzmann constant.
The equation 1.6 means that there is always a number of embryos in a melt, although they could be not stable[6]. Obviously, the number of embryos depends on the value of ΔG_n^{hom}. The maximum in the value of ΔG_n^{hom}, as a function of n, for an under-cooled melt is called critical value ΔG_c^{hom}.
From the equation 1.6 the theoretically expression of the steady-state nucleation rate I_v can also be obtained. The resulting expression is reported in the following:

$$I_v = I_0 exp\left[-\frac{\left(\Delta G_c^{hom} + \Delta G_a\right)}{k_b T}\right] \tag{1.7}$$

in which ΔG_a is the activation energy required to the transfer of an atoms through the liquid/solid interface and the attachment to an embryo.The first term in exponential in equation 1.7 goes as T^{-2} and the second term as T^{-1}.
An increase in under-cooling produces a more numerous and smaller-size

[5] For a detailed treatment of this subject see reference [8]

[6] For further readings see [1] and [2]

critical nuclei which cause a further decrease of temperature and a decrease of the nucleation rate. The maximum value of the nucleation rate can be obtain at a suitable temperature below the melting point. For very high cooling rate there may be not enough time for nuclei to form, leading to the formation of a *glassy* amorphous solid[7].

Growth Once a nucleus is become stable it begin to growth forming various type of micro-structures. In general the factors that influence this process are the heat flow conditions in the liquid and in the solid and the thermal under-cooling. They contribute to determine the final shape of the grains [8]. If the temperature at the liquid/solid interface has a positive gradient then the latent heat of fusion generated at the liquid/solid interface is dissipated through the solid, otherwise in case of positive gradient, it is dissipated through the liquid. The former case is typical of *directional* (or *columnar*) solidification i.e. the growth is anisotropic and the preferential crystallographic direction is that of heat flow. The latter case instead produce the so called *inner equiaxied* solidification, producing grains of symmetric morphology.

According to process condition, a system can also show a transition from columnar to inner equiaxied solidification mode. This is due to the further growth of the broken arms of dendritic structures formed through columnar mode. The columnar-equiaxied transition depends on the degree of convection in the liquid.

The kinetic of grains growth is also influenced by the attachment kinetic of the embryos to the grains and the roughness of the grains surfaces. For further proofs see the book [5].

Order of phase transition Solidification is a phenomenon that involves changes of several length scales. If these changes are characterized by abrupt variations in latent heat and volume in addition to variations in thermodynamics properties such as specific heat and expansion coefficient, the phase transition is called *first-order* transition. An example of first order transition is the liquid-solid transition of water. On the other hand, if the changes are characterized by continuous variations in volume and latent heat and in discontinuous variations in specific heat, the phase transition is called *second-order* transition.

A distinction between first and second order transitions can be made through considerations involving Gibbs free energy. The volume and latent heat

[7] Further details about this topic can be found in [8] and [5]

[8] For binary mixtures important factors are also the composition and diffusion efficiency of mixture components.

changes in the first order transitions implies that

$$V_i = \left(\frac{\partial G_i}{\partial P}\right)_T \neq \left(\frac{\partial G_f}{\partial P}\right)_T = V_f \tag{1.8}$$

$$-S_i = \left(\frac{\partial G_i}{\partial T}\right)_P \neq \left(\frac{\partial G_f}{\partial T}\right)_P = -S_f \tag{1.9}$$

where G_i and G_f are the Gibbs free energy of the initial and final phases, respectively.
In the second order transition the first order derivatives, $-S$ and V are continuous, while the second order ones are discontinuous:

$$-V_i k_T = \left(\frac{\partial^2 G_i}{\partial P^2}\right)_T \neq \left(\frac{\partial^2 G_f}{\partial P^2}\right)_T = -V_f k_T \tag{1.10}$$

$$-\frac{C_{p,i}}{T} = \left(\frac{\partial^2 G_i}{\partial T^2}\right)_P \neq \left(\frac{\partial^2 G_f}{\partial T^2}\right)_P = -C_{p,f} \tag{1.11}$$

in which k_T is the isothermal bulk modulus and C_p is the specific heat at constant pressure.

1.3.2 Triglycerides

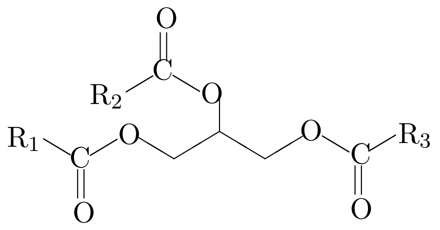


Table 1.1: A triglyceride structure formula.

Triglycerides are simple molecules consisting of three acyl chains attached to a glycerol group by an ester linkage (see figure 1.1). In literature (e.g. [56] and [57])are indicated with abbreviations specifying glycerol substitutes. For example the abbreviation *POS* indicates a triglyceride molecule with substitutes deriving from palmitic, oleic and stearic acids, respectively. Cocoa butter is constituted by *POP*, *POS* and *SOS* triglycerides types as can be seen in figure 1.2, taken from [52], showing triglyceride compositions of typical cocoa butter, deriving from several countries. The most interesting

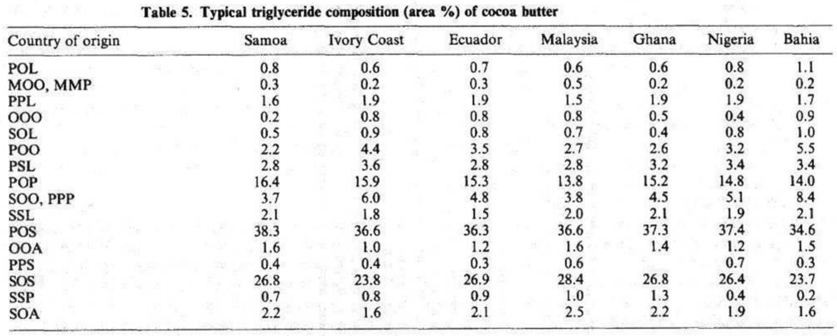


Table 5. Typical triglyceride composition (area %) of cocoa butter

Country of origin	Samoa	Ivory Coast	Ecuador	Malaysia	Ghana	Nigeria	Bahia
POL	0.8	0.6	0.7	0.6	0.6	0.8	1.1
MOO, MMP	0.3	0.2	0.3	0.5	0.2	0.2	0.2
PPL	1.6	1.9	1.9	1.5	1.9	1.9	1.7
OOO	0.2	0.8	0.8	0.8	0.5	0.4	0.9
SOL	0.5	0.9	0.8	0.7	0.4	0.8	1.0
POO	2.2	4.4	3.5	2.7	2.6	3.2	5.5
PSL	2.8	3.6	2.8	2.8	3.2	3.4	3.4
POP	16.4	15.9	15.3	13.8	15.2	14.8	14.0
SOO, PPP	3.7	6.0	4.8	3.8	4.5	5.1	8.4
SSL	2.1	1.8	1.5	2.0	2.1	1.9	2.1
POS	38.3	36.6	36.3	36.6	37.3	37.4	34.6
OOA	1.6	1.0	1.2	1.6	1.4	1.2	1.5
PPS	0.4	0.4	0.3	0.6		0.7	0.3
SOS	26.8	23.8	26.9	28.4	26.8	26.4	23.7
SSP	0.7	0.8	0.9	1.0	1.3	0.4	0.2
SOA	2.2	1.6	2.1	2.5	2.2	1.9	1.6

Figure 1.2: Triglycerides composition of cocoa butter.

feature of cocoa butter deriving from the presence of triglycerides molecules is its polymorphism. This means that it is able to crystallize in a number of different crystal packing configurations. The process technique of tempering, already treated, is just a technique to control crystallization of cocoa butter (often called pre-crystallization [44]) inside melted chocolate to induce the most stable form before full solidification occur.

Polymorphism Literature about polymorphism of triglycerides and of cocoa butter is a quite controversial. There is no agreement, in fact, about the number and crystal structure of triglycerides. This is probably due to the fact that the number and crystal packing of polymorphs depends on several elements e.g. molecules type, composition, presence of unsaturated branches and so on.

The first work on cocoa butter polymorphs is that of Wille and Lutton [61]. They identify six polymorphs of a commercial cocoa butter by means of X-ray diffraction. The polymorphs are labeled with roman numbers I to VI, according to growing stability and melting point. Polymorph I is obtained crystallizing cocoa butter at 0^oC or less. this is the most unstable form and easily turns in a more stable polymorph of type II. The polymorph III is obtained through a crystallization at about $5-10^oC$ or through a solid transition from the polymorph of type II. Polymorph IV can be obtained through a solidification process at $16-21^oC$. It is not easy to obtain polymorph V from liquid phase. One way is to keep the matter at a constant temperature for about one week. An alternative method that speed up greatly the polymorph formations is that analysed by Stapley et al. [64] and concerning the use of a shear flow during tempering process. Polymorph V is the crystal form that give rise to the best chocolate quality and therefore a lot of studies are addressed to the obtainment of such form. Polymorph VI, the most stable form, is obtained by Wille and Lutton only through a solid transition from polymorph V. The transition from V to VI polymorph is believed to

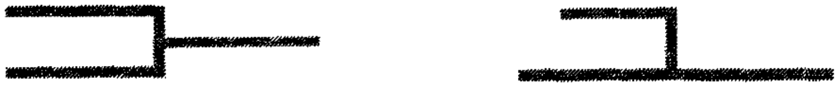


Figure 1.3: Possible molecular conformations of triglycerides.

be connected with fat bloom phenomena.
Nowadays polymorphism of triglycerides can be simply classified as α,β' and β, according to three main lateral (i.e. short) crystal packing. α form is characterized by an hexagonal cell packing, β' form has an orthorhombic cell packing, β form has triclinic cell packing. Moreover the polymorphs differs from the molecules conformations inside cell. In particular, the α form has a tuning fork conformation while the β' and β forms have a chair conformation in which the chain are tilted (see figure 1.3). The short crystal packing was studied in a quantitative manner through X-Ray diffraction experiments, showing that:

- α form shows a quite symmetric shape curve(depending also on the chains type) with a single strong line at 4.15 $\overset{o}{A}$.

- β' form shows an asymmetric shape curve with two quite strong lines at 3.8 and 4.2 $\overset{o}{A}$

- β form shows an asymmetric shape curve with a single strong line at 4.6 $\overset{o}{A}$

The asymmetric shape curves are evidence of asymmetric molecules conformations (i.e. chair), while the symmetric shape curve is evidence of a symmetric conformation (i.e. tuning-fork).
Triglycerides may also pack in different longitudinal arrangements (see figure 1.4): Double and triple hydrocarbon chain packing. The longitudinal arrangements depends on the nature of triglyceride molecules i.e on the chains length and degree of saturation and on the angle of orientation [62].

Despite of the great amount of experimental studies about structural arrangements of triglycerides, little structural information, deriving from X-ray diffraction analyses is available [63]. Moreover, experimental methods can be applied also to obtain the structural parameters of stable polymorphs and greatly depends on the characteristics of the studied triglycerides molecules (see e.g. [59] and [60]).

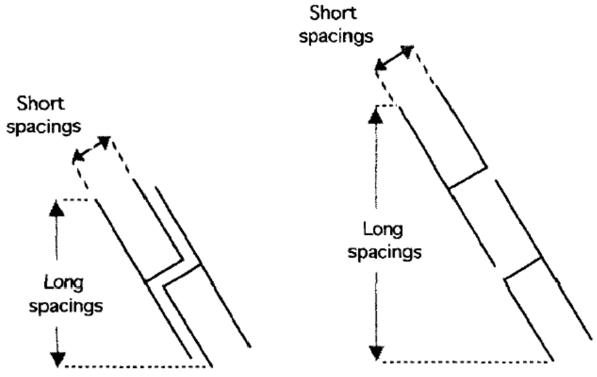


Figure 1.4: Possible longitudinal arrangements of triglycerides.

In particular it is worthwhile to cite the work of Yano and Sato [58] because states clearly a relationship between molecular spatial conformations and crystal structures in solid. From a theoretical point of view, the mathematical models used to describe solidification process occurring in triglycerides concern the nucleation process. We could cite the work of Phipps [4] in which a theoretical model based on equations 1.1–1.7 is used to described nucleation phenomenon. Recently (see [6]) crystallization kinetics of triglycerides were modeled using a phenomenological model in which the phase transition is treated as a chemical reaction and so characterized by a reaction kinetics equation whose parameters have to be determined according to experimental data. Such model is know in literature as Johnson-Mehl-Avrami-Kolmogorov model (e.g. see [1] and [2]).
The mathematical model used to describe phase transitions of triglycerides are therefore purely empirical model, through which a deeper comprehension of the transition phenomena cannot be reached.
As we have just said in section 1.3.1, the phase transition processes involve phenomena at several length scales, therefore fully describing models of such phenomena should be a multi-scale model, constituted by several parts interacting each other; each one describing a suitable length scale. That's why in the following sections we developed a coarse-grained model of triglycerides. The coarse-grained model will be used to the analysis of molecule conformations varying system conditions. This work represents, in our opinion, the first step towards the development of a multi-scale model for a deep comprehension of transition behavior of triglycerides.

Chapter 2

Mathematical modelling

In this chapter the mathematical model developed to characterize the behavior of triglycerides varying environmental conditions will be described.
In order to let the model to emerge naturally from the discussion, in section 2.1 we will start describing briefly the two main kind of modelling approaches: the macroscopic approach and the microscopic one. In section 2.2 we analyse thoroughly the model belonging to these two categories. This section represents also an historical background of mathematical modelling. We will emphasize differences and application fields of the different approaches. In section 2.3 we dwell upon the physical problem we deal with in order to choose the appropriate model to describe bulk behavior of triglycerides. Then, in section 2.4, we will treat on microscopic modelling techniques. The numerical algorithms for equations integration are discussed in section 2.5. In section 2.7 we deal with the tools to analyse a result of a numeric integration.

2.1 Introduction

The laws of physics have the unpleasant feature that they are expressed in terms of equations we cannot solve exactly, except in very few special cases. If we wish to study the motion of more than two interacting bodies, even the relatively simple laws of Newtonian mechanics become essentially unsolvable. That is to say, they cannot be solved analytically, using only pencil and paper. Nevertheless, mathematical modelling represents a primary support to the comprehension of the complex physical phenomena staying behind chemical processes and industrial plants design.
For example, we can briefly treat qualitatively modelling techniques of fluid flows, e.g. methane, water, polymer blends, toothpaste or honey, for industrial purposes.
Moreover, such models can be also applied in the field of food products plants design. This is a not trivial task since food processes involve almost

ever very complex materials.

Fluids are made of a very high number of microscopic particles interacting with themselves. For process plant design purposes there's not interest in modelling the detailed structure of matter. The model developed in such cases are therefore called *macroscopic model* since they deal with the macroscopic evolution of densities of extensive variables such as mass, energy and momentum. Mathematical models describe temporal (and spatial) evolutions of the density of these quantities . Therefore, in this case,a system is not modelled by a set of particles but rather by properties (mass, momentum and energy) of a continuum. Equations describing the time evolution of the continuum properties are represented by systems of ordinary or partial differential equations (e.g. Navier-Stokes equations) [7](usually such equations are solved on a spatial grid using finite difference or finite elements methods for discretizing the continuum equations).[11]

In other cases, however, someone can be interested in develop much detailed models.

In the field of materials science, for example, often we wish to compute the properties of materials. Few decades ago, there was only one way to predict the properties of a molecular substance, namely by making use of a theory that provided an approximate description of that material. Such approximations are inevitable precisely because there are very few systems for which the equilibrium properties can be computed exactly (examples are the ideal gas, the harmonic crystal, and the two-dimensional Ising model for ferro magnets) [10].

A more o less detailed mathematical model describing the evolutions of systems of many atoms or molecules (*many* almost always means very much more than two) can be built. That's why it's called *microscopic model.* Since the birth and development of computer science, we are able to solve numerically the model with the necessary accuracy. This model is quite strictly correlated to macroscopic model we are talking before, since the particles motion describes, in the sense we specify in the following chapters, the interactions representing the correct physical properties that figure as parameters in the continuum equations. So the parameter of the macroscopical model can be derived from the microscopic one.

2.2 Mathematical model

Several techniques were developed during the last seventy years. In figure 2.1 we summarize schematically modelling techniques showing time and temporal scales of application. In the upper right corner macroscopic models are shown. In particular we find two blocks in the scheme concerning macroscopic modelling scale. The block named *Engineering Design* concern with

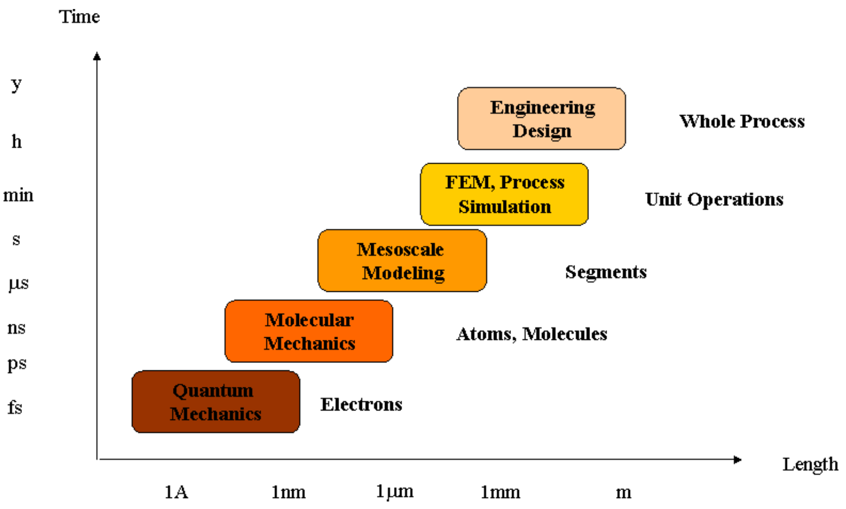


Figure 2.1: Temporal and spatial scale of mathematical modelling techniques

models of whole process plants and are based on balances of mass, energy and momentum according to conservation laws. *FEM*[1] *Process Simulation* deals with models of unit operations. Such models join balances equations, physical (e.g. transport phenomena) and chemical (e.g. reaction mechanisms) laws. We want to emphasize that someone could consider the division between those blocks so much artificial since they are almost ever used both for process plant design.

Going to the lower left, we found what, as engineers, we call microscopic models, in this scheme divided in three blocks according to the detail level. *Meso-scale modelling* is a wide block containing the less detailed modelling techniques. In this block can be found lattice methods [12], Langevin dynamics [13], coarse-graining models [14] and Fokker-Planck Equations [15]. The block *Molecular dynamics* deals with the evolution of quite large systems of atoms or molecules. An atom can be modelled according to the particular technique as single bead or can be considered as a part of a group of atoms constituting a bead (e.g. united atoms [16]).
The blocks we are talking about up to now are based on newtonian mechanics.
The last block,represents models with the higher detail level. Since it can describe spatio-temporal evolutions of electrons, these models are based on the laws of *Quantum mechanics*, as the name of the block tell us. Several techniques that differ each other by the level of approximations are available. Although, theoretically, a particles system can be fully described by

[1] i.e. Finite Element Method

the rules of (relativistic) quantum mechanics, in the most practical cases this is not possible because of the calculation limits and more *coarse* methods should be used.

This brief digression shows the available routes to model physical processes of industrial interest. Until now, the macroscopic methods are those most commonly used. Often in many practical cases the macroscopic model contain terms whose parameters are evaluated experimentally. The mean drawback in such cases is that we obtain a specific model describing a precise process. Changing a model parameter can cause the model to be inadequate to describe the new conditions.

Microscopic models instead have the mean advantage of a more detailed description of the phenomena but the serious drawback is that concerning the high dimensional order of the system we deal with and therefore in the past were not widely used (excluding numerical calculations regarding quantum mechanics). In the last few years the microscopic methods have become very widely used, because are able to explain the behavior of complex materials. Many examples can be found in literature, for the sake of brevity we cite only [65].

2.3 Crystallization of triglycerides

In the preceding chapter (chapter 1) clearly appears that to study phase transition and the influence of flow condition in a quite fundamental manner we have to deal with a mathematical model able to correlate macroscopic properties (first of all density) to microscopic ones, such as molecules conformations and in general microscopical orders. Therefore, the use of a microscopic model seems to be an unavoidable choice to do. Since several methods are available (see section 2.2), the right method has to be chosen on the basis of temporal and spatial scales of the phenomena we have to deal with. Obviously, this choice influences greatly the predictive power of the model.

Macroscopic physical properties ca be divided in two groups: static (or equilibrium properties) and dynamical (or non equilibrium) properties. An equilibrium property is, for example, the mean potential energy of the system; Non-equilibrium properties are, for example, viscosity and reaction kinetics. The analyses of equilibrium properties require in general more detailed mathematical models than in the case of non-equilibrium properties analyses. The last in fact may depend on the motion of molecules aggregates evolving on a greater temporal and spatial scale-lengths.

Molecular dynamics seems to be the right choice for our purposes, since we have to analysis non-equilibrium phenomena (e.g. flow induced molecular conformations).

Several methods owning to molecular mechanics field are available. Since the time scale of a solidification phenomena could be also one week, we should use the most coarse-grained model (which keeps the detail of molecules structures) available. Unfortunately, such model does not exist and therefore have to be developed by us. However, an existing very detailed model (i.e. atomistic), derived for alkane molecules, was recently used to the description of triglyceride molecules in a particular (α) solid phase. As can be seen in chapter 3 this model will be used to simulate a liquid phase giving quite good results in bulk properties prediction. Therefore it will be used in chapter 4 as a test model to the development of a new coarse-grained model, able to perform simulations on longer time and spatial scale-lengths.

In the following will be presented the general molecular mechanics concepts [36]regarding both atomistic simulations (chapter 3) and the coarse-grained model developed by us in the chapter 4.

2.4 Molecular mechanics

2.4.1 Introduction

Many of the problem of practical interest are unfortunately too large to be considered by *quantum mechanics.* Quantum mechanical simulations are time-consuming since they deal also with electrons of systems. Molecular mechanics (or molecular dynamics) instead ignores the electronic motions and calculate the energy of a system as a function of nuclear positions only. Thus, it is the best choice that can be done to perform calculations on systems containing significant numbers of atoms. However, it cannot of course provide properties that depend on the electronic distribution in a molecule. Molecular mechanics is based on the validity of several assumptions. The most important of these is the so called Born-Oppenheimer approximation. It consists to consider electrons to relax to equilibrium in a time negligible respect atomic motions, that is to neglect electronic motions respect to atomic motions. Therefore, energy can be regarded as a function of nuclear coordinates only.
Assumptions are made also regarding molecules interactions. These are based on rather simple models describing stretching of bonds, opening and closing angles and rotations about single bonds. These interaction laws joined with their parameters constitute the *force field* of the molecular model. The main characteristic of the force field is that it must be *transferable* in the sense that a force field developed and tested on a relatively small number of cases should be efficient to a much wider range of problems.

In this section we describe the molecular dynamics technique adopted to analysis the behavior of tryglicerides.

2.4.2 Molecular dynamics: basic principles

In molecular dynamics, successive configurations of the system are generated by integrating Newton's laws of motion 2.1.

$$\frac{d^2x_i}{dt^2} = \frac{F_{x_i}}{m_i} \tag{2.1}$$

This equation synthesize the motion of a particle (i.e. molecules, atoms etc.)of the system of mass m_i along one coordinate x_i with F_{x_i} being the force on the particle in that direction. Several techniques differ each other depending on F_{x_i} terms constituting the so called force field and will be briefly treated in the following paragraphs. As results of the integration of equations 2.1 we obtain trajectories that specifies positions and velocities of the system varying the time.

Theoretically, three different type of problem, to which Newton's laws of motion may be applied, can be distinguished. The simplest case is that in which no force acts on each particle between collisions. In this case positions of particles between two collision can be computed through $v_i * \delta t$ where v_i is the constant velocity and δt is the time between collisions. The second case, a constant force can be applied between collisions. An example of this type of motion is a charged particle moving in a uniform electric field. The third is the case in which the force acting on each particle depends on the position of the other particles. In this case the motion is almost impossible to describe, analytically, in every case of interest, due to strong coupling of particles' motions. Therefore a numerical integration have to be engaged in. The last case will be of interest for our purposes.

2.4.3 The Force Field

First step to be dealt with to start a molecular dynamics simulation is to define a force field i.e. the right-hand side of equation 2.1. In general expressed in terms of an interaction potential (see equation 2.2).

$$F_{x_i} = -\frac{\partial \mathscr{V}}{\partial x_i} \tag{2.2}$$

Interaction potential $\mathscr{V}$ can be broken up into terms involving interaction between pairs, triplets,…of particles 2.3.

$$\begin{aligned}\mathscr{V}(\mathbf{r}) \quad = \quad & \sum_i v_1(\mathbf{r}_i) + \sum_i \sum_{j>i} v_2(\mathbf{r}_i, \mathbf{r}_j) + \\ & + \sum_i \sum_{j>i} \sum_{k>j>i} v_3(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \dots \end{aligned} \tag{2.3}$$

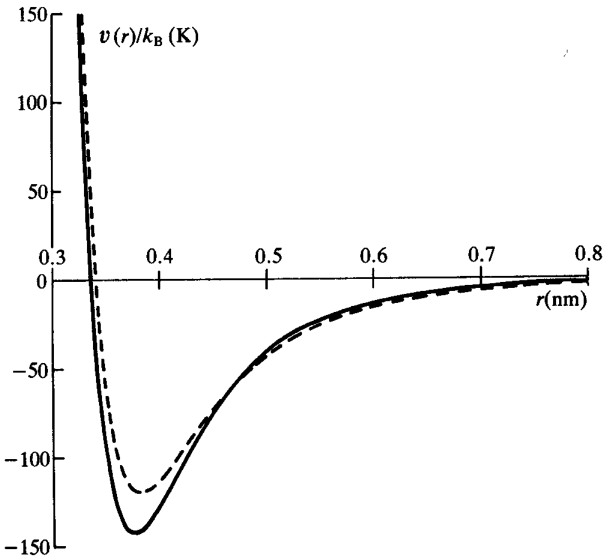


Figure 2.2: Experimental Argon pair potential and Lennard-Jones 12–6 effective pair potential (dashed line) for Argon.

The first term of the equation 2.3 represents the effect of an external field, including container walls effects on the system. The remaining terms represent particle interactions.

The second term, v_2, is the pair potential. Pair potential can be divided into bonded potential, that is the potential due to chemical bond inside molecules and non bonded potential. We deal with pair (i.e. two-bodies)and multi-bodies bonded potential and non-bonded potential in next paragraphs (2.4.3 and 2.4.3 respectively) Non-bonded potential shows the typical features of intermolecular interactions. In figure 2.2, for example, the experimental (non-bonded) pair potential of two Argon atoms is depicted [17]. As we can see, there is a steeply repulsive wall at short distances, due to non-bonded overlap between the electron clouds. Moreover, there is a negative well, responsible for cohesion in condensed phases. Finally, there is an attractive tail at large distance, essentially due to correlation between electron clouds surrounding atoms (Van der Waals interactions). Non-bonded pair potential, for charged particles, includes also Coulombic terms.

The term v_3 in equation 2.3 is the three bodies interaction potential. It can be divided, as the pair potential, into two contributions: bonded potential and non-bonded one. Bonded potential takes into account the interactions between particles belonging to two adjacent bonds and can be take into account in a quite easy manner.

Different address for non-bonded potential. It was found that it affects up to 10 percent of lattice energy of Argon [17]. The same order of magnitude can be conjecture to hold, in general, liquid phases. Despite the non-negligible size of non-bonded three-body term, there are only rarely included in numerical simulations. This is because the calculations involving sums over triplets of particles are very time-consuming in numerical simulations. Fortunately, we can overcome this drawback acting on pair potential. The pairwise approximation, in fact, gives a remarkably good approximation of liquid properties if the pair potential parameters were suitable correct to account for three-body interactions absence. Corrected pair potentials are often called *effective pair potentials.*

Groups rotations around a single bond are token into account by bonded four-body potential. Four and higher non-bonded interaction terms in equation 2.3 are expected to be small in comparison with the v_2 and v_3 non-bonded terms and are neglected in almost every applications.

Bonded potentials

Bonded potentials regard systems in which particles are organized in molecular structures. Systems constituted by atoms for example need non-bonded potential terms only. In principle the molecular systems could be treated as like as atomic systems. Chemical bonds are in fact simply interatomic potential energy terms. Obviously, to deal with chemical bonding and with chemical reactions means deal with quantum mechanics. The use of a classical approximation would in fact cause serious errors.
To overcome these problems, molecules can be treat as rigid or semi-rigid units with bond lengths, bond angles and torsion angles fixed or oscillating around equilibrium values. This hypothesis stand on the consideration that bond vibrations are of very high frequency but of low amplitude (therefore unimportant for liquid properties).

Two-bodies potentials In figure 2.3 the potential energy curve for a typical bond is shown. One way to model this curve is proposed by Morse [18]. The Morse potential has the following form (2.4):

$$v\left(l\right) \quad = \quad D_e\{1-exp\left[-a\left(l-l_0\right)\right]\}^2 \tag{2.4}$$

D_e is the depth of the potential energy minimum and $a = \omega\sqrt{\frac{\mu}{2D_e}}$ where μ is the reduced mass and ω is the frequency of the bond vibration. ω is related to the stretching constant of the bond, k, by $\omega = \sqrt{\frac{k}{\mu}}$. l_0 is the reference value of the bond. We refer to l_0 as *reference* value and not *equilibrium* because l_0 is the value assumed by the bond when the force field $v\left(l\right)$ in equation 2.4 is set

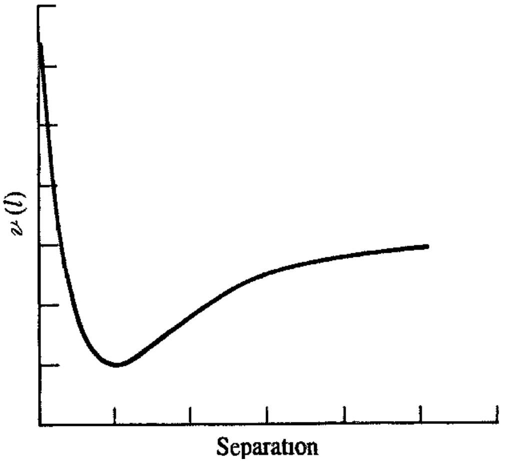


Figure 2.3: Variation in bond energy with interatomic separation.

to zero, not the equilibrium value assumed when the whole system reach a state of minimum energy, i.e. when all parts of force field contribute. On the other side we have to add the consideration that complex interplay between the various components of the force field causes the bonds to deviate slightly from their reference values in order to compensate for other contributions to the energy.
The morse potential is not usually used in molecular dynamics force fields. This is both because it is not particularly amenable to efficient computation as because it requires three parameters to be specified for each bond.

Considering the fact that in general, in molecular mechanics calculations, it is rare for bonds to deviate from their equilibrium values, simpler expressions are often used. The most elementary approach is to use a Hooke's law formula 2.5.

$$v(l) = \frac{k}{2}(l - l_0)^2 \tag{2.5}$$

k is a force constant and l_0 is the reference value. Hooke's law functional form is a reasonable approximation of the potential curve in figure 2.3 (and of the Morse potential curve[2]) at the bottom of the potential well (see figure 2.4), at a distance corresponding to ground-state bond. Theoretically, a better approximation of Morse potential can be obtained including cubic and higher terms in bond stretching potential as shown in the following equation 2.6.

$$v(l) = \frac{k}{2}(l - l_0)^2 \left[1 - k'(l - l_0) - k''(l - l_0)^2\right] - \dots \tag{2.6}$$

[2] as can be easily seen doing a Taylor expansion of the equation 2.4 around l_0

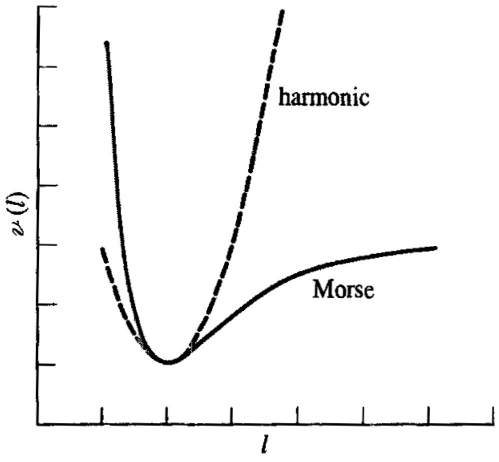


Figure 2.4: Comparison of the simple harmonic potential (Hooke's law) with Morse potential.

This expansion could cause undesirable effects such as for example catastrophic lengthening bonds due to a maximum generated by cubic term. Therefore is not commonly used.

Sometimes (e.g. GROMOS-96 forcefield [19]) the covalent bond potential is written for reasons of computational efficiency (because no square root has to be evaluated) as:

$$v(l) = \frac{k'}{4}\left(l^2 - l_0^2\right)^2 \tag{2.7}$$

The force constant of equation 2.7 is related to the force constant of the equation 2.5 as $k = 2k'l_0$ (this can be proved doing a Taylor expansion of the 2.7 around l_0).

In coarse-grained polymer simulation the particles are often connected by a FENE [3] potential [20]:

$$v(l) = -\frac{k}{2}l_0 \log\left(1 - \frac{l^2}{l_0^2}\right) \tag{2.8}$$

At short distance the potential asymptotically goes to harmonic potential with force constant k, while it diverges at distance l_0.

Three-bodies potentials The deviation of angle (three-body bonded interaction), formed by three contiguous atoms, from the reference value is frequently described using Hooke's law too:

$$v(\theta) = \frac{k_\theta}{2}(\theta - \theta_0)^2 \tag{2.9}$$

[3] Finitely Extendible Nonlinear Elastic

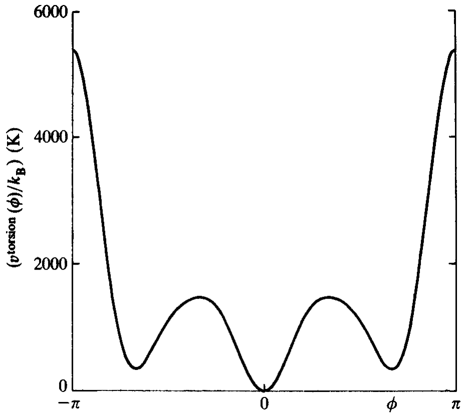


Figure 2.5: Torsional potential of butane.

k_θ is the force constant and θ_0 is the reference value. In general the force constant is smaller than that of bond stretching in equation 2.5. This means that less energy is required to distort an angle away from equilibrium than to stretch or compress a bond.

Other functional forms for three-body interactions are also possible. A simplified function used in GROMOS force field [19] to represent angle vibrations is the following:

$$v(\theta) = \frac{k'_\theta}{2} (\cos\theta - \cos\theta_0)^2 \qquad (2.10)$$

The force constant of equation 2.10 is related to that of harmonic potential (equation 2.9) by $k_\theta = k'_\theta \sin^2\theta_0$.
More complex angle potentials are available in literature (see for example [21]) but we don't care about them in this seat.

Four-bodies potentials The two-bodies (bonds) and three-bodies (angles) bonded interaction are often regarded as *hard* degrees of freedom because quite substantial energies are required to cause significant deformations from their reference values. Structural properties of molecules are instead due to complex interplays between torsional (i.e. four-bodies bonded potential) and non-bonded contributions.
The most common example of energy changes due to bond rotations, also called proper dihedral interactions, is represented by staggered and eclipsed structures of butane shown in figure 2.5.
Many functional forms are used to model torsional potential. Obviously, it is essential that these functions properly represent energy profiles connected to

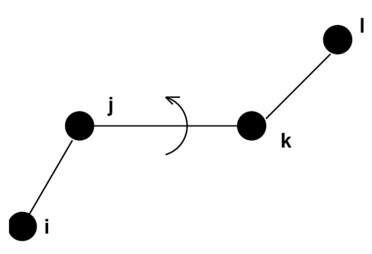


Figure 2.6: Proper dihedral angle (*trans* form).

bond rotations. Torsional potentials are almost always expressed as a cosine series expansion. One functional form is the following:

$$v\left(\phi\right) = \sum_{n} \frac{k_{\phi,n}}{2}\left(1+\cos\left(n\phi-\phi_0\right)\right) \tag{2.11}$$

$k_{\phi,n}$ influences the relative barriers to rotation. n is the *multiplicity* i.e. the number of minimum points in the function as the bond is rotated through 360^o. ϕ_0 is the phase factor. It determines where the torsion angle passes through the minimum value. ϕ is the *proper dihedral angle*. It is defined (see figure 2.6), according to IUPAC/IUB convention, as the angle between ijk and jkl planes, with *zero* corresponding to *cis* configuration (i and l on the same side). Often simplified form of equation 2.11 can be used:

$$v\left(\phi\right) = k_{\phi}\left(1+\cos\left(n\phi-\phi_0\right)\right) \tag{2.12}$$

An alternative expression but equivalent to equation 2.11 is the Ryckaert-Bellemans dihedral potential:

$$v\left(\phi\right) = \sum_{n} C_n\left(\cos\phi\right)^n \tag{2.13}$$

Another kind of four-bodies bonded potentials is represented by *out-of-plane* bending terms also called *improper dihedral* potentials. These potentials are meant to keep planar groups planar (e.g. aromatic rings) or to prevent molecules from flipping over to their mirror images (e.g. maintenance of stereochemistry at chiral centers). One of the most used functional form to take into account improper dihedral is the following:

$$v\left(\xi\right) = \frac{k_{\xi}}{2}\left(\xi-\xi_0\right)^2 \tag{2.14}$$

As we see equation 2.14 is a harmonic potential. Looking at the figure 2.7 ξ is the angle between ijk and jkl planes in each case.

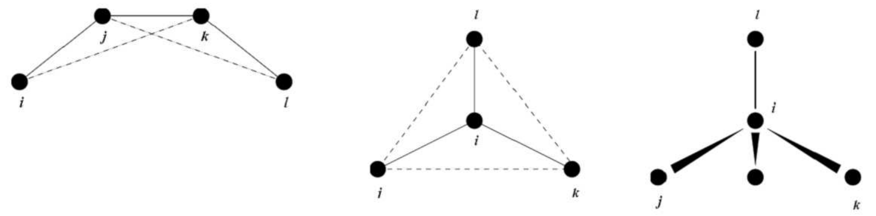


Figure 2.7: Improper dihedral angles. Out-of-plane bending for rings (left), planar structures (middle), out of tetrahedral (right)

Non-bonded potentials

Non-bonded interactions are *through-space* interactions, i.e. interactions between non-bonded particles (read atoms) inside the same molecule or owing to different molecules. The non-bonded terms are usually divided in two groups, one comprising electrostatic interactions and the other Van der Waals interactions.

Electrostatic interactions Electronegative elements attract electrons more than less electronegative elements, giving rise to an unequal distribution of charge in a molecule. This charge distribution can be represented in a number of ways, one common approach being an arrangement of fractional point charges throughout the molecule (*point charge model*). These charges are designed to reproduce the electrostatic properties of the molecule. If the charges are restricted to the nuclear centers they are often referred to as *partial atomic charges* or net atomic charges. The electrostatic interaction between two molecules (or between different parts of the same molecule) is then calculated as a sum of interactions between pairs of point charges, using Coulomb's law:

$$\mathscr{V} = \sum_{i=1}^{N_a}\sum_{j=1}^{N_b}\frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}} \tag{2.15}$$

N_a and N_b are the numbers of point charges in the two molecules.
If sufficient point charges are used then all of the electric moments can be reproduced and the multipole interaction energy is exactly equal to that calculated from the Coulomb summation (equation 2.15).
If an accurate representation of a molecule's electrostatic properties is needed, charges can be placed at locations other than at the atomic nuclei. An alternative to the point charge model is to assign dipoles to the bonds in the molecule. The electrostatic energy is then given as a sum of dipole-dipole interaction energies. This approach can be unwieldy for molecules that have a formal charge and which require charge-charge and charge-dipole terms to

be included in the energy expression. For charged species we are dealt with it is more naturally to use the point charge model.

Given the widespread use of the partial atomic charge model, it is important to consider, qualitatively at least, how the charges are obtained. For simple species the atomic charges required to reproduce the electric moments can be calculated exactly if the geometry is known. For example the tetrahedral arrangement of the hydrogens about the carbon in methane means that each hydrogen atom has an identical charge equal to one quarter the charge on the carbon.

In some cases the atomic charges are chosen to reproduce thermodynamic properties calculated using a molecular dynamics or a Monte Carlo simulation. A series of simulations is performed and the charge model is modified until satisfactory agreement with experiment is obtained. This approach can be quite powerful despite its apparent simplicity, but it is only really practical for small molecules or simple models.

The electrostatic properties of a molecule are a consequence of the distribution of the electrons and the nuclei and thus it is reasonable to assume that one should be able to obtain a set of partial atomic charges using quantum mechanics. Unfortunately, the partial atomic charge is not an experimentally observable quantity and cannot be unambiguously calculated from the wave-function. This explains why numerous ways to determine partial atomic charges have been proposed, and why there are still considerable debates as to the *best* method to derive them.

Indirect comparisons of the various methods are possible, usually by calculating appropriate quantities from the charge model and then comparing the results with either experiment or quantum mechanics. For example, in some cases one might examine how well the charge model reproduces the experimental or quantum mechanical multipole moments or the electrostatic potential around the molecule.

If molecular mechanics is used to model systems containing thousands of atoms such as polymers, charges cannot be derived routinely perform quantum mechanical calculations on a molecule with so many atoms. Therefore in this case molecules can usefully be broken into fragments of a suitable size. In some cases the fragments might appear relatively easy to define; for example, many polymeric systems are constructed by connecting together chemically defined monomeric units. The atomic charges for each monomer should be obtained from calculations on suitable fragments that recreate the immediate local environment of the fragment in the larger molecule. The charge sets obtained from electrostatic potential fitting can be highly dependent upon the basis set used to derive the wave-function. Most of the charge models place the charge on the nuclear centers. Atom-centered charges have many advantages. For example, the electrostatic forces due to charge-charge interactions then act directly on the nuclei. This is important if one wishes to calculate the forces on the nuclei as is required for energy minimization or a molecular dynamics simulation. Nuclear-centered charges do nevertheless suffer from some drawbacks. In particular, they assume that the charge density about each atom is spherically symmetrical. However, an atom's valence electrons are often distributed in a far from spherical manner, especially in molecules that contain features such as lone pairs and π electron clouds above aromatic ring systems.

One way to represent the anisotropy of a molecular charge distribution is to use distributed multipoles. In this model, point charges, dipoles, quadrupoles and higher multipoles are distributed throughout the molecule. These distributed multipoles can be determined in various ways such as the *distributed multipole analysis* or DMA model. We don't deal with DMA model in detail, we will say only that it is based on the calculation of the multipoles from a quantum mechanics wave-function.

Another aspect to be dealt with is the simulation of dielectric solvents. All of the formulae written for electrostatic energies, potentials and forces have included the permittivity of free space, ε_0. This is as one would expect for species acting in a vacuum. However, under some circumstances a different dielectric model is used in the equations for the electrostatic interactions. This is often done when it is desired to mimic solvent effects, without actually including any explicit solvent molecules. One effect of a solvent is to dampen the electrostatic interactions. A very simple way to model this damping effect is to increase the permittivity, most easily by using an appropriate value for the relative permittivity in the Coulomb's law equation (i.e. $\varepsilon = \varepsilon_0 \varepsilon_r$).

Van der Waals interactions Electrostatic interactions cannot account for all of the non-bonded interactions in a system. An obvious example is represented by the rare gas atoms in which all of the multipole moments are zero and so there can be no dipole-dipole or dipole-induced dipole interactions. But there clearly are interactions between the atoms to justify the existence of liquid and solid phases and the deviations from ideal gas behavior Deviations from ideal gas behavior were famously quantified by Van der Waals, thus the forces that give rise to such deviations are often referred to as Van der Waals forces.

Figure 2.8 shows the experimental interaction energy between two isolated argon atoms. The other rare gases show a similar behavior. Let's summarize the mean features of this curve. The interaction energy is zero at infinite distance (in fact could be negligible even at relatively short distances). As the separation is reduced, the energy decreases, passing through a minimum. The energy then rapidly increases as the separation decreases further.

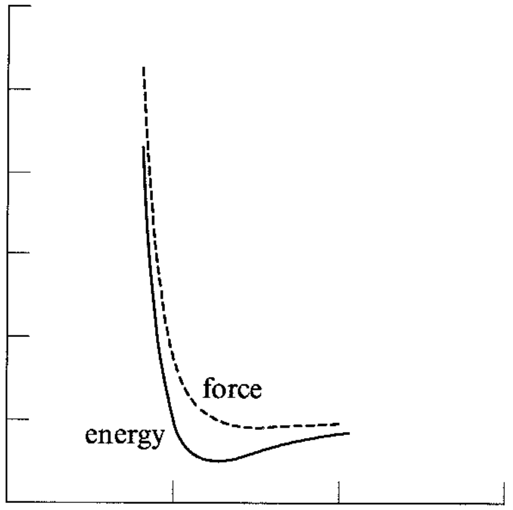


Figure 2.8: The Van der Waals interaction energy and the force between two Argon atoms.

In figure 2.8 the force between the atoms, which equals minus the first derivative of the potential energy with respect to distance, is also reported. A variety of experiments have been used to provide evidence for the nature of the van der Waals interactions.

The curve features in figure 2.8 is usually considered to arise from a balance between attractive and repulsive forces. The attractive forces are long-range, whereas the repulsive forces act at short distances. The attractive contribution is due to dispersive forces. London first showed how the dispersive forces could be explained using quantum mechanics [36] and so these interactions are sometimes referred to as *London forces.* The dispersive force is due to instantaneous dipoles which arise during the fluctuations in the electron clouds. An instantaneous dipole in a molecule can in turn induce a dipole in neighboring atoms, giving rise to an attractive inductive effect. Below a certain value, even a small decrease in the separation between a pair of argon atoms causes a large increase in the energy. This increase has a quantum mechanical origin and can be understood in terms of the Pauli principle, which formally prohibits any two electrons in a system from having the same set of quantum numbers. The interaction is due to electrons with the same spin, therefore the short-range repulsive forces are often referred to as *exchange forces.* They are also known as *overlap forces.* The effect of exchange is to reduce the electrostatic repulsion between pairs of electrons by forbidding them to occupy the same region of space (i.e. the internuclear region). The reduced electron density in the internuclear region

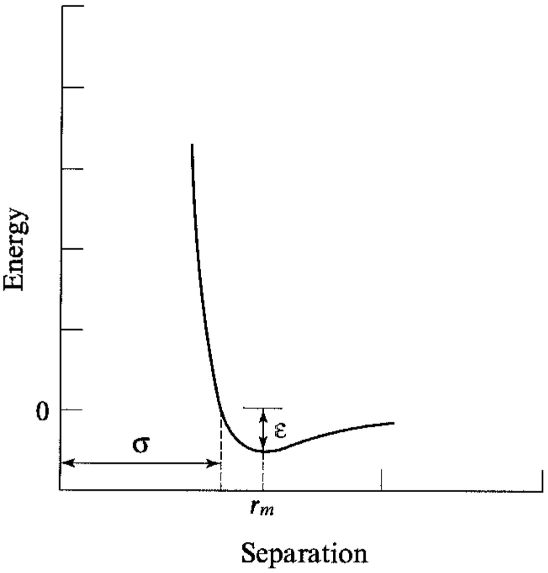


Figure 2.9: The Lennard–Jones 12 − 6 potential.

leads to repulsion between the incompletely shielded nuclei. At very short internuclear separations, the interaction energy varies as $1/r$ due to this nuclear repulsion, but at larger separations the energy decays exponentially, as $\exp\left(-\frac{2r}{a_0}\right)$, where a_0 is the Bohr radius.

The dispersive and exchange-repulsive interactions between atoms and molecules can be calculated using quantum mechanics, though such calculations are far from trivial.
For a force field we require a means to model the interatomic potential curve accurately (figure 2.8), using a simple empirical expression that can be rapidly calculated. The need for a function that can be rapidly evaluated is a consequence of the large number of Van der Waals interactions that must be determined in many of the systems that we would like to model.
The best known of the Van der Waals potential functions is the Lennard–Jones 12 − 6 function, which takes the following form for the interaction between two atoms:

$$v(r) \;=\; 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \qquad (2.16)$$

The Lennard-Jones 12−6 potential contains just two adjustable parameters: the collision diameter σ (the separation for which the energy is zero) and the well depth ε. These parameters are graphically illustrated in figure 2.9. The Lennard–Jones equation may also be expressed in terms of the separation at which the energy passes through a minimum, r_m. At this separation, the first derivative of the energy with respect to the internuclear distance is zero (i.e. $\frac{dv}{dr} = 0$), from which it can easily be shown that $r_m = 2^{\frac{1}{6}}\sigma$. We can thus also write the Lennard–Jones $12 - 6$ potential function as follows:

$$v(r) = \varepsilon\left[\left(\frac{r_m}{r}\right)^{12} - \left(\frac{r_m}{r}\right)^{6}\right] \tag{2.17}$$

or

$$v(r) = \frac{A}{r^{12}} - \frac{C}{r^{6}} \tag{2.18}$$

in which $A = 4\varepsilon\sigma^{12}$ and $C = 4\varepsilon\sigma^{6}$. The Lennard–Jones potential is characterised by an attractive part that varies as r^{-6} and a repulsive part that varies as r^{-12}. The power law r^{-6} can be derived from quantum mechanical theoretical treatment of the dispersion energy. On the contrary the repulsive power law r^{-12} cannot be theoretically justified with quantum mechanics calculations (that instead suggest an exponential term).
Although such power laws work quite well for rare gases, for other systems (in general for bigger molecules such as hydrocarbons) are too steep and therefore have to be decreased. For example values of 9 or 10 for repulsive term are used in some force fields. Even so, the Lennard–Jones $12 - 6$ potential is widely used for calculations on large systems because of the computational efficiency (r^{-12} can be calculated by squaring the r^{-6} term). Definitely, we can write Lennard–Jones potential in the follow most general form:

$$v(r) = k\varepsilon\left[\left(\frac{\sigma}{r}\right)^{n} - \left(\frac{\sigma}{r}\right)^{m}\right] \tag{2.19}$$

in which:

$$k = \frac{n}{n-m}\left(\frac{n}{m}\right)^{n/(n-m)} \tag{2.20}$$

For $n = 12$ and $m = 6$ the equation 2.19 gives Lennard–Jones $12 - 6$ potential (equation 2.16).

We want only, at this point, briefly cite one of the several choice to describe Van der Waals interactions that goes towards a more realistic exponential expression for the repulsion term: The Buckingham potential (equation 2.21.

$$v(r) = \varepsilon\left[\frac{6}{\alpha - 6}\exp\left[-\alpha\left(\frac{r}{r_m} - 1\right)\right] - \frac{\alpha}{\alpha - 6}\left(\frac{r_m}{r}\right)^{6}\right] \tag{2.21}$$

The Buckingham potential has three parameters ε, r_m and α. For α assuming a value between 14 and 15 this potential approximates the Lennard–Jones

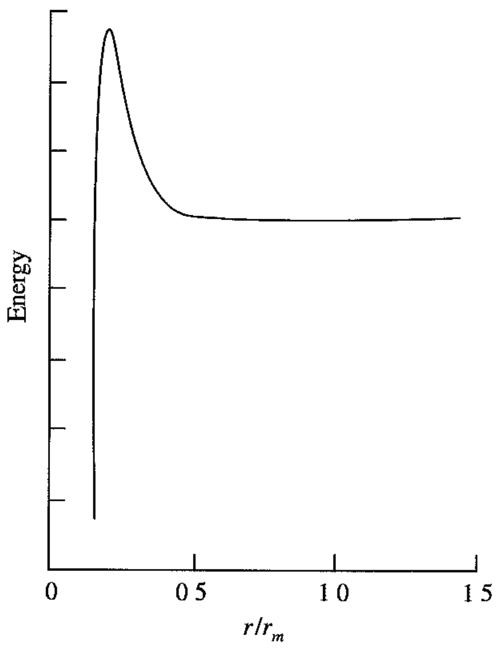


Figure 2.10: The Buckingham potential.

12−6 potential near the minimum-energy region. As shown in figure 2.10 the Buckingham potential becomes strongly attractive at very short distances. This could lead to nuclei being fused together during a calculation, and must be avoided.

Interactions in polyatomic systems Theoretically, the interaction energy between molecules depends also on their relative orientations and conformations. Even so, in all practical examples, Van der Waals interactions are determined as the sum of interaction between all pairs of particles (e.g. atoms) in which we have been considered the molecules to be divided. In the case of polyatomic systems, therefore, Van der Waals terms regard the interactions between different types of atoms. For example if we have to perform a simulation of a system constituted by carbon monoxide molecules using a two-site model we require Van der Waals parameters not only for carbon-carbon interactions and oxygen-oxygen interactions but also for the carbon-oxygen interactions. In general a system containing N different types of atom requires $N(N-1)/2$ sets of parameters for the interaction between unlike atoms. As the determination of Van der Waals parameters is difficult and time-consuming the cross interactions are usually obtained from the parameters of the pure atoms using mixing rules.

In the commonly used Lorentz-Berthelot mixing rules, the collision diameter

σ_{AB} for the $A - B$ interaction equals the arithmetic mean of the values for the two pure species,while the well depth ε_{AB} is equals the geometric mean of the values for the pure species (see equations 2.22 and 2.23).

$$\sigma_{AB} = \sigma_A + \sigma_B \tag{2.22}$$

$$\epsilon_{AB} = \epsilon_A \epsilon_B \tag{2.23}$$

The Lorentz-Berthelot combining rule method works well when applied to similar species. Moreover this method overestimates geometric mean rule of the well depth. We have to add that sometimes the collision diameter for mixed interactions is calculated as the geometric mean of the values for the two component atoms too (e.g OPLS force field [22]).

At this point we have to remark that the non-bonded interactions between atoms separated by three bonds (i.e. the $1,4$ atoms) are often scaled down by an empirical factor. This is done because the $1,4$ atoms are affected by torsional potential (four-bodies bonded potential see 2.4.3) too. Therefore non-bonded $1,4$ interactions are rectified to compensate the redistribution of the charge along the connecting bonds that aims to reduce the non-bonded interactions if two 1,4 atoms come close together. Another reason to scale down $1,4$ interactions is that they reduce the error associated with the use of an r_{-12} repulsion term (which is too steep compared with the more correct exponential term). This error is most significant for 1,4 atoms.

The parameters for the van der Waals interactions can be obtained in various ways. In the early force fields, such parameters were often determined from an analysis of crystal packing to reproduce as accurately as possible experimental geometries and thermodynamic properties such as the heat of sublimation. In subsequent force fields they were obtained using liquid simulations. In such case, parameters were optimized to reproduce a range of thermodynamic properties such as the density and enthalpy of vaporization for liquids in exam. Parameter optimization is performed also because in most cases we have to construct *effective* pair potentials, to take into account a significant proportion of the many-body effects. Doing so, pair potentials do not represent the *true* interaction energy between two isolated particles.

United atoms force fields

Molecular systems simulations can be performed naturally representing explicitly all atoms in the model. However, as the number of non-bonded interactions scales with the square of the number of interaction sites present, clear advantages can be obtained reducing the number of interaction sites.

The simplest way to do this is to include some of the atoms of a molecule into the atoms to which they are bonded. This can be easily gained in presence of hydrogen atoms. For example, a methyl group could be modelled as a single *pseudo-atom* or *united atom*. Thus, butane could be modelled as a four-particle model rather than one with twelve atoms saving high computational efforts [4]. In the united atom force field the Van der Waals center of a united atom is usually associated with the position of the heavy atom, that is non-hydrogen atom (in the case of the methyl group that means the carbon atom).

The Van der Waals and electrostatic parameters would be obviously modified to take account of the adjoining hydrogen atoms.

United atom models are useful to represent organic molecules such as hydrocarbons or proteins. In this case, not all of the hydrogen atoms in the molecules are included into their adjacent atoms, but just those that are bonded to carbon atoms. Similarly, in presence of hydrogen atoms bonded to electronegative atoms and therefore able to form hydrogen-bonds, explicit representation is needed.

One drawback of the united atom force fields is that regarding chiral centers that may be able to invert during a calculation. The inversion may be prevented through the use of an improper torsion term (see for example equation 2.14) to keep the center in the correct position.

In some force field models, even simpler representations are used than the united atom approach. Sometimes, entire groups of atoms are modelled as single interaction particles (e.g. a benzene ring). Even simpler models are possible but this model are obviously far from being good representations of microscopic systems. Nevertheless these models are useful because their simplicity enables more extensive calculations otherwise not possible[5].

An insight on force field development

As we have been now understood a force field, even if it is related to a small set of particles, contains a large number of parameters. The development of the sets of parameters of a force field from scratch requires significant efforts. Sometimes existing force fields can be modified in order to represent new kinds of molecules. Even this task is complicated and time-consuming. A general algorithm can be explained to a better force field development process.

First step is to single out which parameters affect the force field stronger. Usually, these parameters have to be singled out among the non-bonded and

[4] Van der Waals interaction between two butane molecules involves the calculation of 16 terms rather than 144

[5] A lot of examples can be found in polymer field; see for example [23]

torsional terms. The aim is to optimize these parameters first as the others (such as the bond-stretching and angle-bending terms) do not greatly affect the results.

The development of a force field depends on the studies are going to be carried out. If molecular dynamics will be used to determine structurally related properties, suitable data have to be select to guide the parametrization process and the force field parameters have to be chosen accordingly. The geometries and relative conformational energies of certain key molecules are usually included in the data set.
Molecular dynamics are often used to reproduce thermodynamic properties therefore parameters have to be determined or at any rate modified accordingly to this goal (e.g [22]). Quantum mechanics can be a powerful tool to provide data for parametrization of force fields to extend the range of chemical systems treated using the force field approach. Obviously parameters derived in such way should be validate against experimental data if possible. Ab initio (i.e. quantum mechanical) calculations have the drawback that are able to reproduce experimental results only for few small systems. That's why they are used only when experimental data may be non-existent or difficult to obtain for particular classes of molecules.

Two mean methods can be used to actually determine force field parameters: *parametrization by trial and error* and *parametrization by least-squares method.*
In parametrization by trial and error the parameters are gradually refined to reproduce better and better the data. A suitable strategy have to be adopted both because it is difficult to simultaneously modify a large number of parameters and because of the coupling between all the parameters. Therefore none of the parameters can truly be considered isolated. Even so, bond stretching and angle bending are often quite independent from other parameters and can be treated separately from the others or can often be transferred from one force field to another without modifications. This remarks cannot be addressed the soft degrees of freedom (i.e. non-bonded and torsional terms) because they are closely coupled and so highly influence each other.
The algorithm that could be used is to first set Van der Waals parameters and determine electrostatic model (e.g. using quantum mechanical calculation [24]). The torsional terms are last determined reproducing torsional barriers and energies of different conformations. To this aim, quantum mechanical calculations are also used as experimental information on torsional barriers is often sparse or non-existent. In any case, the torsional potential is fitted to reproduce the energy curve, in conjunction with the Van der Waals potential and partial charges. This process is obviously iterative because at any stage could be necessary to modified any of the parameters.

The alternative approach is parametrization by least-squares method (see [25]). The method is based on the concept of using least-squares fitting to determine the set of parameters that gives the best fit to the data. First of all , we have to choose a set of experimental or theoretical (i.e. quantum mechanics calculations derived) data (e.g. thermodynamical properties) to be reproduced by force field. The parameters are set in such way that the sum of squares of the differences between the observed and calculated values for the set of properties (also called *(squared) error*) are minimized. For simplicity we can assume that the squared root of the error is related to the force field parameters by a linear obtained through a Taylor series expansion truncated at first order term (equation 2.24).

$$\Delta \mathbf{y}\left(\mathbf{x}_0 + \delta \mathbf{x}\right) \quad = \quad \Delta \mathbf{y}\left(\mathbf{x}_0\right) + \mathbf{J}\delta \mathbf{x} + \ldots \tag{2.24}$$

In which $\Delta \mathbf{y}$ is the squared root of the error, $\mathbf{x}$ is the force field parameters vector and $\mathbf{J}$ is the Jacobian of the $\Delta \mathbf{y}\left(\mathbf{x}\right)$ function.
This method can be complicated introducing suitable weights in $\Delta \mathbf{y}\left(\mathbf{x}\right)^2$ term.

A concept that must be taken into account in force field development is the transferability of force field parameters. This is a crucial aspect as parametrization processes often require the calculation of so many parameters. Very good force fields can be constructed starting from force fields already available in literature. Common sense lead to choose force fields referring to molecules similar as much as possible to those being studied.
Among the force fields available, often a good compromise between a complex functional form and a large number of atom types must be faced.
Sometimes force fields can be derived from atomic properties starting from existent force fields designed to be used on a very wide range of elements and atom types, such as the Universal Force Field [26].
In practice it is important to try to ensure that the errors in the derived force field are balanced, or, in other words, to ensure that no parameters give rise to errors larger than the others.

The transferability's characteristics of existing force fields can be improved by using the same parameters for as wide a range of situations as possible. Doing so, a problem that often rise regard the non-bonded terms because in principle parameters for the non-bonded interactions between all possible pairs of atom types are necessary. Even so, usually, the same set of Van der Waals parameters can be used for most atoms of the same element, even if with different hybridizations. Moreover, further simplification is to make the torsional dependent solely upon the atom types of the two atoms that form the central bond, rather than on all four atoms that comprise the torsion angle [36].

2.4.4 Initial configurations

The first step to deal with in a molecular dynamics simulation is to *construct* the initial configuration of the system. This operation is not trivial because often could determine the success or the failure of a simulation. Although this point is strongly problem dependent, practical considerations can be mentioned. High-energy interactions have to be avoided as these cause instabilities. To this aim energy minimizations are usually performed, before starting the simulation itself. For some details see section 2.5. Moreover, for simulation of systems at equilibrium a very convenient choice could be to select an initial configuration that is close to the state we desire to simulate. The last consideration has especially weight in the case of simulation of crystal materials, in which molecules are organized in structured lattices. In case of homogeneous liquid simulation with a very high number of particles, a lattice initial configuration is often chosen, too. This is due, principally, to the easiness of construction of such lattice structures. Moreover, doing so, high-energy overlaps and instabilities could be avoided. If experiments are available (e.g. X-ray analyses) then the lattice structures are chosen accordingly. If no experiment is available the initial configuration can be chosen from one of the common lattice structures. A typical choice is a face-centred cubic lattice. The lattice parameters (i.e. lattice size) are chosen so that density of the initial configuration is (almost) equal to that of the real system.

For simulations of molecules, the orientation of each molecule are also needed. For small linear molecules a face-centred cubic lattice with the molecules oriented in a regular fashion along the four diagonals of the unit cell are usually chosen.
Alternatively, the orientations may be chosen completely at random or by making small random changes from the orientation in a regular lattice. To avoid high densities, due to overlaps between molecules (particularly if the molecules are large), a suitable, in case iterative, procedure can be adopted to reach step by step the initial condition of desired characteristics. According to the problem, to use an initial configuration that is close to the expected equilibrium distribution is however a good choice.

For simulations of inhomogeneous systems, for example systems with solute molecules immersed in a solvent, the starting conformation of the solute may be obtained from an experimental technique such as X-ray crystallography or Nuclear Magnetic Resonance (NMR), or may be generated by theoretical modelling. The coordinates of some solvent molecules may be known if the structure is obtained from X-ray crystallography, but it is usually necessary to add other solvent molecules to give the appropriate solvent density. A typical approach to construct the initial configuration in such case is to use the coordinates obtained from a previous simulation of the pure solvent and then add the solute molecules. Any solvent molecules that are too close to the solute are then discarded before the calculation proceeds.

2.4.5 Boundary conditions

In a molecular dynamics simulation great attention may be direct to the correct treatment of boundaries and boundary effects because as we want to calculate *macroscopic* properties from simulations using relatively small numbers of particles. The importance of boundary effects can be easily illustrated. In a real system with volume equal to one litre at room temperature molecules interacting with the walls of the container are typically one over 1.5 million. In general the number of particles in a molecular dynamics simulation is frequently less than 1000 or however very less than the Avogadro number. Therefore the number of molecules interacting with the walls of the boundary are at least comparable with the bulk molecules. Clearly, such simulation would be relevant to studies of liquid drops but it would not be appropriate to derive *bulk* properties without any treatment of boundary conditions. Suitable boundary conditions are therefore adopted to overcame this drawback. Boundary conditions can be divided in two groups: *Periodic* boundary conditions and *non-periodic* boundary conditions.

Periodic boundary conditions

Periodic boundary conditions are used in the most of applications to simulate systems in which the particles experience forces as if they were in bulk fluid. With periodic boundary conditions, *bulk* simulations using a relatively small number of particles can be performed.

To understand how periodic boundary conditions work, we can consider a cubic box of particles, replicated in all directions to give a periodic array as in figure 2.11. In this figure each box is surrounded by 8 neighbours as it is a two dimensional system; in three dimensions each box would have 26 nearest neighbours. Let's focus our attention on the central box (which is the system to simulate). If a particle leave the central box during the simulation then it is replaced by an image particle that enters from the opposite side, as illustrated in figure 2.11. The number of particles within the central box thus remains constant.

It's easy to compute the coordinates of the particles in the image boxes: They can be obtained from the coordinates of the particles of the central box simply by adding or subtracting integral multiples of the box sides.

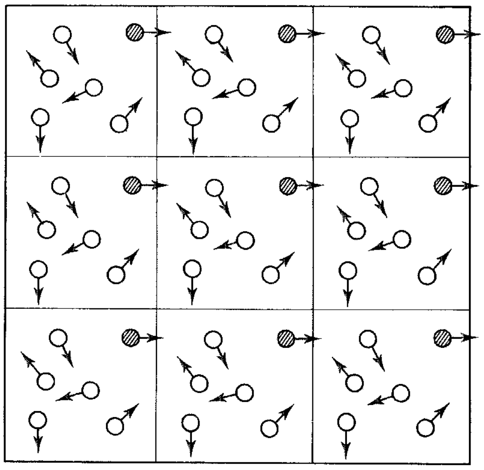


Figure 2.11: Scheme of periodic boundary conditions in two dimensions.

In general periodic boundary conditions are imposed on cubic cell systems (due also because of the simplicity of the cell to duplicate). However, cell with different shapes allow periodic boundary conditions. If, for example, we deal with a simulation that wants to analyse the behaviour of a single molecule inside a bulk of solvent molecules, we are theoretically only interested in a spherical volume surrounding the central molecule. Therefore, a different cell shape, which approximate a sphere better than a cubic cell, allow us to spend as little of the computer time as possible. This is because, given the number of particles, a simulation using a cell that approximate a spherical cell better will require fewer particles than a comparable simulation using a cubic cell. The cell shape must have the property of fill all of space by translation operations of the central box in three dimensions. According to this property the possible cell shapes are the following (see also figure 2.12: the cube (also the parallelepiped), the hexagonal prism, the truncated octahedron, the rhombic dodecahedron and the elongated dodecahedron [27].

The cell shape are often chosen according to the system's geometry, for example spherical shape for a spherical molecule conformation due to folding. In this case, a rhombic dodecahedron should be used as this shape can approximate a spherical one.

Among the five possible choice,in general, the cube/parallelepiped and the truncated octahedron (and sometimes hexagonal prism) have been most widely used. This is because the expressions for calculating the images are easier to implement in such cases.

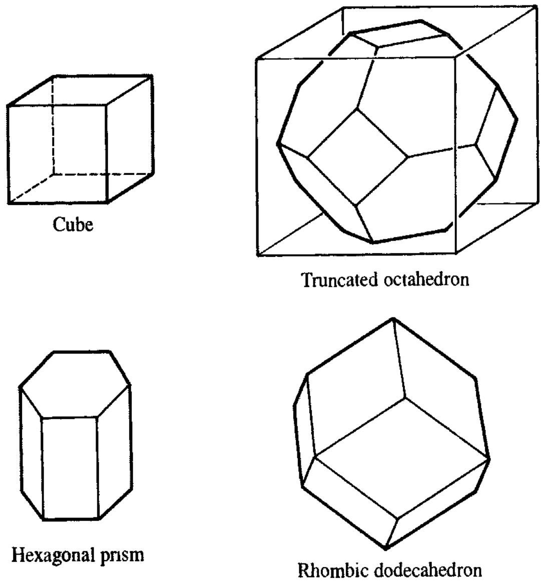


Figure 2.12: Possible cell shapes used with periodic boundary conditions.

Suitable (i.e. not standard) periodic boundary conditions have to be used according to the problem. If we want to study, for example, molecular adsorption on a surface layer, the surface is modelled as a true boundary. When a molecule strays out of the true boundary it is reflected back into the simulation cell. Usual periodic boundary conditions are applied to motions both parallel to the surface and through the opposite side of the surface layer.

Periodic boundary conditions have some drawbacks. The most important of which is the fact that it is not possible to achieve fluctuations that have a wavelength greater than the length of the cell.
Another important problem is the maximum interactions range allowed. If the range over which interactions act is short compared to the cell size we have no problem. Otherwise long-range interactions could influence negatively system's simulations because they introduce fictitious periodicity as a molecule interact with itself. The effects of imposing a periodic boundary condition can be evaluated empirically by comparing the results of simulations performed using a variety of cell shapes and sizes.

Non-periodic boundary conditions

In any cases, periodic boundary conditions are not appropriate in computer simulations. Such cases can be summarized as: Systems that inherently contain a boundary (e.g liquid droplets); inhomogeneous systems (e.g. solute-solvent systems) or systems that are not at equilibrium; study high order

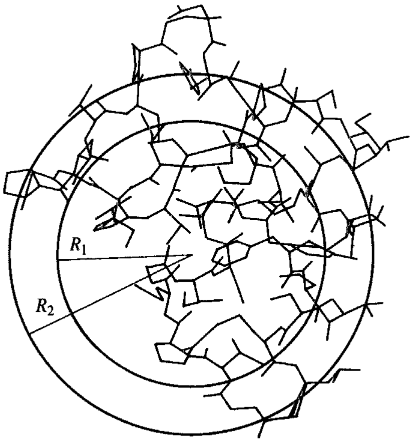


Figure 2.13: Reaction zone and reservoir zone.

systems (i.e. with high number of particles e.g. polymers). The first case is obviously motivated. The last two cases derive from considerations of computational difficulties and time-consuming and a suitable policy have to be adopted.
For example, in the case of systems in which there is the presence of a solvent, if not so many computational resources are available, the best thing to do is to simulate the systems ignoring solvent molecules. Vacuum calculations lead to significant problems. A vacuum boundary tends to minimise the surface area and so may distort the shape of the non-spherical systems. Moreover, small molecules may adopt more compact conformations when simulated in vacuo due to favourable intramolecular electrostatic and Van der Waals interactions, not dampened by the solvent molecules.
Although, neglect solvent molecules corresponds to a very unrealistic situation, the deviation of the predicted properties can be computed comparing such values with experimental data.
Another possibility can be performed if quite high computational resources are available: to incorporate explicitly some solvent molecules in the system in such a way that molecules are surrounded with a skin of solvent molecules. Boundary effects now act between the solvent-vacuum interface and not between the molecules-vacuum interface leading to a more realistic treatment of the solute.

Sometimes, if we have high dimensional solute solvent systems or if we are only interested in a specific part of the solute (e.g. active part of an enzyme), the region around a solute molecule (or part of it) can be divided in two regions (figure 2.13): *Reaction* region and *reservoir* region. The reaction region contains all atoms or groups within a given radius R_1. The atoms in the

reaction regions are subjected to the full simulation method. The reservoir region contains all atoms outside the reaction region and within a distance R_2 (see section 2.4.6). The atoms in the reservoir region can be kept fixed in their positions, or can be restrained or to stay within the shell defined by R_1 and R_2 or to oscillate around their initial positions using a harmonic potential. Any atoms whose distance from the molecule is greater than R_2 are discarded or can be kept fixed in their initial positions.
This method is sometimes called stochastic boundary conditions. A variety of schemes for performing simulations using such method have been proposed. It is important to say that using such boundary conditions can prevent natural evolution to occur thus leading to artificial behaviours.

2.4.6 The truncation of potential and the *Minimum Image Convention*

The calculation of the non-bonded terms requires the most part of calculation time in a molecular dynamics simulation. In fact, while the number of bonded terms are proportional to the number of atoms, the non-bonded ones are proportional to the square of the number of atoms (if only pair interactions are considered). Moreover, theoretically, the non-bonded interactions should be calculated between every pair of atoms in the system but in all practical applications this is not undertaken. The reasons are of two kinds. The first reason is that in most cases Lennard-Jones potential falls off very rapidly with distance. The second reason deal with the introduction of the boundary conditions.

In the calculations of non-bonded interactions the two techniques, applied to avoid the calculation of all the interactions, are called: *non-bonded cut-off* and *minimum image convention.*
When a cutoff is employed, the interactions between all pairs of atoms that are further apart than the cut-off value are set to zero. This can be done without cause any troubles because of the shape of Lennard-Jones potential as we have just said. In the minimum image convention, each atom *sees* at most just one image of every other atom in the system (which is repeated infinitely via the periodic boundary method). That means that only the closest atoms interact with a fixed atom, as illustrated in figure 2.14. The minimum image convection is a quite obvious way to fix a cut-off length: on the contrary each particles of the systems could sees its own image. Minimum image convection, on the other hand, could means to introduce fictitious periodicity in the system. These periodicity could not influence the correctness of properties derived as in the case of molecular dynamics simulations of well-ordered systems, but in the case of liquid systems or vitreous systems the periodicity could induce the formation of unreal ordered phases. Practically, to use minimum image convention means to cut-off the non-bonded terms no more than half the length of the shortest dimension.

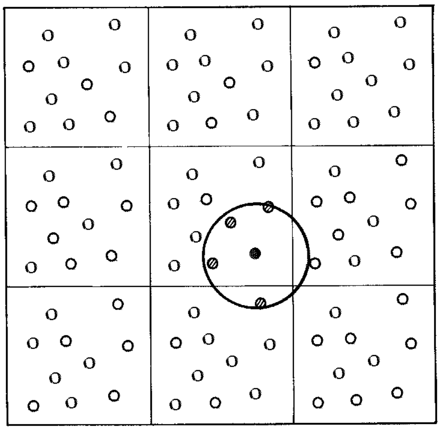


Figure 2.14: The cut-off and the minimum image convention.

For simulations of molecules the upper limit on the cutoff may also be affected by the size of the molecules. To have a dimensional order of cut-offs we can say that if we have to perform simulations where the Lennard-Jones potential is the only non-bonded interaction, a cutoff of 2.5σ gives rise to a relatively small error.
However, when we deal with simulations in which long-range electrostatic interactions are involved, practical tell us that the cutoff should be much greater and a cut-off value of at least 10σ have to be adopted, even if this could be insufficient. Therefore, more comprehensive methods have been developed for dealing with the electrostatic interactions.

2.4.7 Non-bonded neighbour lists

The use of a cut-off don't reduce the computational time required to calculate the non-bonded interactions. This is because the calculation of the distance between every pair of atoms in the system is still needed. When we deal with simulations of fluids, as an atom's neighbours (i.e. those atoms that are within the cut-off distance) do not change significantly, we can store the atoms to include in non-bonded calculations in an array. This allow us to calculate only the distances of the neighbours avoiding to calculate the distances of all the other atoms in the system. This technique is called *non-bonded neighbour list* and was first proposed by Verlet[28].
The Verlet neighbour list stores all atoms within the cutoff distance, together with all atoms that are slightly further away than the cut-off distance. The neighbour list is updated at regular intervals throughout the simulation. Between updates, the neighbour and pointer lists are used to directly identify the nearest neighbours of each atom. The distance used to calculate each atom's neighbours should be larger than the actual non- bonded cut-off dis-

tance so that no atom, initially outside the neighbour cut-off, approaches closer than the non-bonded cutoff distance before the neighbour list is updated again. The neighbour lists should be updated at the correct frequency. In fact, if the update frequency is too high the procedure is inefficient otherwise if it is too low, the energies may be calculated incorrectly because of the atoms could move moving inside the non-bonded cut-off region. There are no fixed rules that determine how much larger the neighbour list cut-off should be than the non-bonded cut-off. Clearly there will be a trade-off between the size of the cut-off. The following general rule can be taken into account: the larger is the difference between non-bonded cut-off a neighbour list cut-off the less frequently will the neighbour list have to be updated. Obviously if the neighbour list cut-off is too large we could have problems dealing with too large storage devices needed.

When significant electrostatic contributions are present, a different neighbour lists approach could be needed. This is because the electrostatic interaction has a much longer range. Therefore we have to use a longer cut-off but, doing so, we significantly increase the number of pairs that must be calculated.

A very efficient method is to use different treatments for electrostatic and for Lennard-Jones interactions. For example the *twin-range* method provides the statement of two cut-offs. The lower cut-off deal wiht Lennard-jones interactions. That means that all interactions below the lower cut-off are calculated as normal at each step. Interactions due to atoms between the lower and upper cut-offs deal with electrostatic interactions and are evaluated only when the neighbour list is updated and are kept constant between these updates. This method is based on the assumption that the contribution of the atoms that are further away will not vary much between updates.

When simulating large molecular systems, it is often advantageous to use a *group-based* cut-off (also called a *residue-based* cut-off). This technique is based on the partition of large molecules into *groups*, each of which contains a relatively small number of connected atoms. Each group should, if possible, be of zero charge. If, for example, the calculation involves small solvent molecules then each solvent molecule is also conveniently regarded as a single unconnected group.

2.4.8 An insight on problems regarding cut-off

Using a cut-off we introduce a discontinuity in the potential energy near the cut-off value. This creates problems of conservation of energy, required in each molecular dynamics simulation and therefore, if possible, should be avoided.

There are several ways to counteract this discontinuity. One approach is to use a *shifted potential*, in which a constant term is subtracted from the

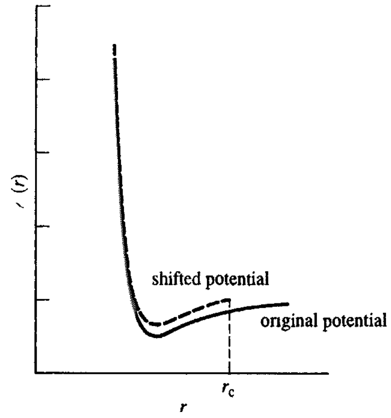


Figure 2.15: A shifted potential.

potential at all values (see figure 2.15):

$$v(r) = v(r) - v_c \qquad r \leq r_c \tag{2.25}$$

$$v(r) = 0 \qquad r > r_c \tag{2.26}$$

where r_c is the cut-off distance and v_c is equal to the value of the potential at the cutoff distance.
As the additional term is constant, it disappears when the potential is differentiated and so does not affect the force calculation. The use of the shifted potential does improve energy conservation2.25 2.26.
An additional problem is that there is a discontinuity in the force at cut-off r_c distance. In fact, at such distance, the force will have a finite value which drops suddenly to zero just beyond the cut-off. This can also give instabilities in simulations. To avoid this, a linear term can be introduced to the potential, making the derivative zero at the cutoff:

$$v(r) = v(r) - v_c - \left(\frac{dv(r)}{dr}\right)_{r=r_c} (r - r_c) \qquad r \leq r_c \tag{2.27}$$

$$v(r) = 0 \qquad r > r_c \tag{2.28}$$

The shifted potential is rarely used in real simulations as it produces the potential deviate from the true potential that implies the deviation of any calculated thermodynamic properties. However, the true values of this properties can be retrieved, in simple cases, using suitable techniques.

2.4.9 Long-range potentials

The charge-charge interactions, which decay as r, are particularly problematic in molecular simulations as their range is often greater than half the box length.

There is much evidence that it is important to properly model these long-range forces. A proper treatment of long-range potentials can also be important when calculating certain properties, such as the dielectric constant. One way to overcome the errors introduced by an inadequate treatment of long-range potentials is the use of a much larger simulation cell, but this is usually impractical if we don't have high computational resources.

Several methods have been developed to handle long-range interactions. We just cite the *Ewald summation*, the *reaction field* method and the *cell multiple* method.

The Ewald summation method was first devised by Ewald [17] to study the energetics of ionic crystals. In this method, a particle interacts with all the other particles in the simulation box and with all of their images in an infinite array of periodic cells. There is thus a contribution to the total energy from the interactions in the central box together with the interactions between the central box and all image boxes. The Ewald summation is computationally quite expensive to implement.

In the reaction field method, a sphere is constructed around the molecule with a radius equal to the cutoff distance. The interactions with molecules inside that sphere is calculated explicitly. The medium outside the sphere is modelled as it is homogeneous and with a suitable dielectric constant. This medium interacts with the sphere through an interaction energy term. which is added to the short-range molecule-molecule interaction. Problems with the reaction field method are related to discontinuities in the energy that arise when the number of molecules within the sphere of a molecule changes.

The cell multipole method (also called the *fast multipole* method) divides the simulation space in uniform cubic cells. The multipole moments (charge, dipole, quadrupole) of each cell are calculated through the sum over the atoms contained within the cell. The interaction between all of the atoms in the cell and another atom outside the cell (inside another cell) can then be calculated using an appropriate multipole expansion.

In the cell multipole method, the multipole expansion is used for interactions that are more than one cell distance away. For interactions that are within one cell distance the usual atomic pairwise interaction method is employed.

2.5 Numerical simulation methods

2.5.1 Energy minimization methods

Energy minimization is a crucial aspect of molecular dynamics simulations. The purpose of an energy minimization is double. First, during the box construction and setting, minimization is a powerful tool to remove the highest energy interactions due to possible particles overlapping. Moreover, energy minimizations are also carried out when we are interested in minimum points, i.e. the most stable states, on the energy surface of a system.
There may be a very large number of minima on the energy surface. Therefore energy minimization is not a trivial operation, specially when we are interested in the minimum with the very lowest energy (called *global energy minimum*).
To find a minimum point on the energy surface a suitable numerical algorithm is needed. There is a vast literature on such methods. Therefore we decide to focus our attention only to those most commonly used in molecular dynamics.

A minimization problem can be formally stated as follows: consider a function f which depends on independent variables called $x_1, x_2, \ldots$. We want to find the values of those variables where f has a minimum value.
At a minimum point the first derivative of the function with respect to each of the variables is zero (2.29) and the second derivatives are all positive (2.30).

$$\frac{\partial f}{\partial x_i} = 0 \tag{2.29}$$

$$\frac{\partial^2 f}{\partial x_i^2} > 0 \tag{2.30}$$

In our case the function f is the total energy of the system and the variables x_i are the Cartesian or the internal coordinates of the atoms. Therefore for a system of N particles f is a function of $3N$ variables. It is not rare to perform the minimum calculation using internal coordinates (as defined in the Z-matrix; see appendix A).
Minima are located using numerical methods, which gradually change the coordinates to produce configurations with lower and lower energies until the minimum is reached.

The choice of the most suitable method depends on many factors: the type of problem, the amount of computer memory available, the shape of the energy surface etc.
Until now, there doesn't exist algorithm that is the best for all molecular

dynamics problem, that's why the suitable for our purposes has to chosen carefully on the basis of the computational efficiency.
In general, a method that works well with quantum mechanics may not be the most suitable for use with molecular mechanics. Quantum mechanics in fact is usually used to model systems with fewer atoms than molecular mechanics. Therefore methods that work well for small systems, may be not suited for large systems. On the other side, very efficient algorithm in the field of molecular dynamics could be inappropriate for quantum mechanics. Algorithms for large systems are in fact characterized by a great number of simple step that make the minimization time longer for small systems.
The most of the minimization algorithms are local in the sense that allow only the calculation of a local minimum i.e. the nearest to the initial point. Therefore, to find the global minimum an iterative procedure have to be adopted. In general, this means to run a series of minimization processes starting from different initial conditions. Until now, no algorithm has yet proved capable of locating the global energy minimum from an arbitrary starting position.
Another important aspect influencing minima calculations is the shape of energy surface. First of all, the global shape (i.e. the shape of the surface quite far from the minima points) of the surface influence the efficiency of the algorithm. For example a minimum of a banana shape curve may be more difficult to find and a suitable minimization policy should be adopted. Moreover, the efficiency of the algorithm depends also on the local shape (i.e. the shape around each minimum). For example, a deep and narrow minimum may be less highly populated than a broad minimum that is higher in energy as the vibrational energy levels will be more widely spaced in the deeper minimum and so less accessible. For this reason the global energy minimum may not be the most highly populated minimum. In any case, it is easier (and the most populated) to find relative minima than the global minimum.
That's why, in general, if disposable, the starting point of a minimization procedure can be taken from experimental data, obtained for example from X-ray crystallography or NMR. In other cases, a theoretical methods or a combination of experimental and theoretical approaches can also be employed.

We can classify minimization algorithms into two groups: those which use derivatives of the energy with respect to the coordinates and those which do not. As the first ones are most commonly used, the non-derivatives methods are briefly treated only for completeness.

Non-Derivative methods

Among all the non-derivative methods, we treat the two most famous: the *simplex* method and the *sequential univariate.*
A simplex is a geometrical figure with $N + 1$ interconnected vertices, where N is the dimensionality of the energy function. For example a simplex of a function of two variables is triangle; a simplex of a function of three variables is a tetrahedron;a simplex of an energy function of $3N$ Cartesian coordinates will have $3N + 1$ vertices. Energy function have to be calculated for each vertex (corresponding to a set of coordinates).
The simplex algorithm locates a minimum by moving the vertices on the potential energy surface. The most simple movement is a reflection of the vertex with the highest value through the opposite face of the simplex, in an attempt to generate a new point that has a lower value. If this new point has the lowest energy value among the vertices than any a 'reflection and expansion' move may be applied. When the reflection fails to produce a better point the simplex contracts along one dimension from the highest point. If this fails to reduce the energy then the simplex tries to contract in all directions, pulling around the lowest point.

The sequential univariate method is a non-derivative method that is considered suitable for quantum calculations. This method is based on the generation from a starting coordinate point x_i of two new point i.e. $x_i + \delta x_i$ and $x_i + 2\delta x_i$. The energies of these two points are calculated and a parabola is then fitted through the three points (from the starting one and the two new points). The minimum point of this quadratic function is then determined and the starting point is then changed to the position of the minimum.
The sequential invariate method usually requires fewer function evaluations than the simplex method but it can be slow to converge if there is strong coupling between two or more of the coordinates or when the energy surface is a long narrow valley.

Derivative methods

Derivative methods are obviously used in molecular mechanics when derivatives of the energy with respect to the coordinates of the atoms are easily disposable. Therefore, these methods provide some information about the shape of energy surface. The most commonly minimisation algorithms used in molecular modelling are the method of *steepest descents* and the *conjugate gradient* method. They are iterative methods as they gradually change the coordinates of the atoms as they move the system closer and closer to the minimum point.
The direction of movement is defined by the first derivative of the energy

(i.e. the gradient) indicates the direction where the minimum lies, and the magnitude of the gradient indicates the steepness of the local slope.
The energy of the system is lowered by moving each atom in direction that depends on the opposite of the gradient.

The steepest descents method moves in the direction parallel through which the surface has the steepest slope. For a system of N particles this direction is most conveniently represented by the following $3N$-dimensional unit vector s_k.

$$\mathbf{s}_k = -\frac{\mathbf{g}_k}{\mid \mathbf{g}_k \mid} \tag{2.31}$$

The steepest descent method is a good method for relieving the highest-energy features in an initial configuration because the direction of the gradient is determined by the largest interatomic forces.
Starting from a coordinates point $\mathbf{x}_k$, we obtain the new coordinates point $\mathbf{x}_{k+1}$ by taking a step of length λ_k along the gradient unit vector s_k (see equation 2.32).

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k \mathbf{s}_k \tag{2.32}$$

The step of the algorithm have to be chosen using suitable policy as a wrong choice could generate critical errors. A large step, in fact, could cause the function value to pass through the minimum point without check it and then increase.
In most applications of the steepest descents algorithm in molecular modelling the step size initially has a predetermined default value. Until the iteration leads to a reduction in energy, the step size is increased by a multiplicative factor (e.g. 1.2). When a step produces an increase in energy, it is assumed that the algorithm has passed through the minimum and raised on the opposite face. The step size is therefore reduced by a multiplicative factor (e.g. 0.5).
The step size depends upon the nature of the energy surface; for a flat surface large step sizes would be appropriate but for a narrow one a much smaller step would be more adequate.
The steepest descent method is generally robust even if the starting point is far from a minimum. The mean drawback is that many small steps will be performed when proceeding down a long narrow energy surface.

The conjugate gradients method differs from steepest descent method because the first one produces a set of directions which does not show the problem of the last one in narrow surfaces. While in the steepest descents method both the gradients and the direction of successive steps are orthogonal. In conjugate gradients, the gradients at each point are orthogonal but

the directions are conjugate. The k^{th} step of conjugate gradients method has the direction $\mathbf{v}_k$, starting from point $\mathbf{x}_k$.
$\mathbf{v}_k$ is computed from the gradient $\mathbf{g}_k$ at the point $\mathbf{x}_k$ and the direction vector computed at $(k-1)^{\text{th}}$ step:

$$\mathbf{v}_k = -\mathbf{g}_k + \gamma_k \mathbf{v}_{k-1} \tag{2.33}$$

in which γ_k is scalar that can be computed as the follows:

$$\gamma_k = \frac{\mathbf{g}_k \cdot \mathbf{g}_k}{\mathbf{g}_k \cdot \mathbf{g}_k} \tag{2.34}$$

Obviously, the conjugate gradient method can only be used from the second step onwards and so for the first step the steepest descent method must be adopted.

2.5.2 Numerical methods for molecular dynamics simulations

Introduction

In a molecular dynamics simulation, theoretically, we have to consider that the force on each particle will change during time because of the particle positions changes. All the particles are under the influence of a continuous potential. As the motions of all the particles are coupled together, a molecular dynamics simulation means, from a practical viewpoint, to solve a many-body problem that cannot be solved analytically. Therefore, in general, the equations of motion are integrated using a finite difference method.

Finite difference techniques are the most commonly used methods to generate molecular dynamics trajectories[6] with continuous potential models. These techniques is based on the discretization of the motion equation. Molecular dynamics configurations are thus obtained at each δt. More particularly, from the total force on each particle at a time t, calculated as the vector sum of its interactions with other particles their acceleration at time t is determined. From the positions and velocities at the time t are then calculated the positions and velocities at a time $t + \delta t$. The process is then iterated to find next system configuration. During each time step the force is assumed to be constant.
The choice of the time interval δt is crucial to integrate the equation of motion with sufficient accuracy; it will depend somewhat on the method of solution. As a general rule we could say that δt have to be significantly smaller than the typical time taken for a molecule to travel its own length.

[6] A trajectory is the series of system configurations as the time varies

Among all the algorithms using finite difference methods,there are several that are suitable for molecular dynamics calculations. The reason deals with the characteristic should have a integration methods to be used in molecular mechanics and it will emerge in the next (see page 62).

The Verlet algorithm

All finite difference methods are based on the assumption than the estimated positions, velocities and accelerations of particles at time $t + \delta t$ may be obtained through a suitable Taylor expansion of those functions around t:

$$\mathbf{r}\left(t+\delta t\right) = \mathbf{r}\left(t\right)+\mathbf{v}\left(t\right)\delta t+\frac{1}{2}\mathbf{a}\left(t\right)\delta t^{2}+\frac{1}{6}\mathbf{b}\left(t\right)\delta t^{3}+\ldots \tag{2.35}$$

$$\mathbf{v}\left(t+\delta t\right) = \mathbf{v}\left(t\right)+\mathbf{a}\left(t\right)\delta t+\frac{1}{2}\mathbf{b}\left(t\right)\delta t^{2}+\ldots \tag{2.36}$$

$$\mathbf{a}\left(t+\delta t\right) = \mathbf{a}\left(t\right)+\mathbf{b}\left(t\right)\delta t+\ldots \tag{2.37}$$

$\ldots$and so on$\ldots$

in which **r** is the position vector, **v** is the velocity vector, **a** is the acceleration vector, **b** is the third derivative vector.

The *Verlet* algorithm [29] is the most widely used method for integrating the equations of motion in a molecular dynamics simulation.
The Verlet algorithm is a multi-step method that uses the positions at time t and $t - \delta t$ and accelerations at time t, to calculate the positions at $t + \delta t$, according to the following equation:

$$\mathbf{r}\left(t+\delta t\right) = 2\mathbf{r}\left(t\right)-\mathbf{r}\left(t-\delta t\right)+\mathbf{a}\left(t\right)\delta t^{2} \tag{2.38}$$

Obtained combining the Taylor expansions (shown in equations 2.39 and 2.40) of $\mathbf{r}\left(t+\delta t\right)$ and $\mathbf{r}\left(t-\delta t\right)$ around t.

$$\mathbf{r}\left(t+\delta t\right) = \mathbf{r}\left(t\right)+\mathbf{v}\left(t\right)\delta t+\frac{1}{2}\mathbf{a}\left(t\right)\delta t^{2}+\ldots \tag{2.39}$$

$$\mathbf{r}\left(t+\delta t\right) = \mathbf{r}\left(t\right)-\mathbf{v}\left(t\right)\delta t+\frac{1}{2}\mathbf{a}\left(t\right)\delta t^{2}-\ldots \tag{2.40}$$

As the velocities do not explicitly appear in the Verlet integration algorithm, they can be calculated, in the simplest way, dividing the difference in positions at times $t + \delta t$ and $t - \delta t$ by $2\delta t$:

$$\mathbf{v}\left(t\right) = \frac{\left[\mathbf{r}\left(t+\delta t\right)+\mathbf{r}\left(t-\delta t\right)\right]}{2\delta t} \tag{2.41}$$

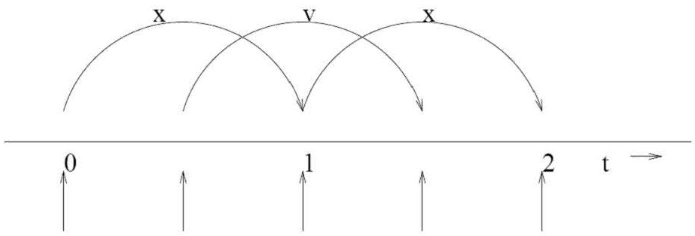


Figure 2.16: A scheme of Leap-frog algorithm.

The velocities can also be calculated at the half-step time:

$$\mathbf{v}\left(t\right) \quad = \quad \frac{\left[\mathbf{r}\left(t+\delta t\right)+\mathbf{r}\left(t\right)\right]}{\delta t} \tag{2.42}$$

The Verlet algorithm has the advantages of simplicity and modest storage requirements. The mean drawbacks of this algorithm are discuss in the following. The positions $r(t+\delta t)$ are obtained by adding a small term $\mathbf{a}\left(t\right)\delta t^2$ to the difference of $2\mathbf{r}\left(t\right)-\mathbf{r}\left(t-\delta t\right)$, that is much larger. This could cause integration errors. Moreover, the velocities not appear explicitly in the equation 2.38. This implies the difficulty to obtain the velocities before the next step have been computed. A minor drawback is the fact that as the Verlet algorithm is a multi-step method, a one-step method is necessary to let the Verlet algorithm to start.

The Leap-Frog algorithm

There are several variations on the Verlet algorithm. Among these, the *Leap-frog* algorithm [30] is the most used. The Leap-frog algorithm integrates the equation of motion according to the following equations:

$$\mathbf{r}\left(t+\delta t\right) \quad = \quad \mathbf{r}\left(t\right)+\mathbf{v}\left(t+\frac{1}{2}\delta t\right)\delta t \tag{2.43}$$

$$\mathbf{v}\left(t+\frac{1}{2}\delta t\right) \quad = \quad \mathbf{v}\left(t-\frac{1}{2}\delta t\right)+\mathbf{a}\left(t\right)\delta t \tag{2.44}$$

As we can see in equations 2.43 and2.44 the Leap-frog algorithm is a multi-step method that calculates the positions at $t+\delta t$ and the velocities at $t+\frac{1}{2}\delta t$. Thus the positions and the velocities mime the frog's leaps as shown in figure 2.16.

The velocities at time t can be computed as follows:

$$\mathbf{v}(t) \quad = \quad \frac{1}{2}\left[\mathbf{v}\left(t+\frac{1}{2}\delta t\right)+\mathbf{v}\left(t-\frac{1}{2}\delta t\right)\right] \qquad (2.45)$$

The Leap-frog algorithm overcome some drawbacks of the Verlet algorithm. In fact the velocity is explicitly included and the terms of the right-hand sides of equations 2.43 and 2.44 have the same order of magnitude. The mean drawback of this algorithm is that the positions and velocities are not synchronised. This implies that the total energy of the system cannot be precisely valued as it is not possible to calculate the kinetic energy (depending on velocity) at the same time as the potential energy (which depends on position).

We want to remark that several algorithms are available besides those treated. The general features that leads to the choice of an integration scheme is that it should be fast, require minimal memory and be easy to program. However, most important considerations have to be taken into account in molecular dynamics. A molecular dynamics integration scheme should conserve energy and momentum, be time-reversible, and should permit a long time step, δt, to be used.
The size of the time step is another crucial point as a simulation using a long time step will require fewer iterations to integrate the equations for a given time interval. In general, each algorithm has a precise relationship between time-step used and errors committed. Regarding this point, for example, Verlet algorithm allows a very large time-step. The right time-step is chosen on the basis of the consideration that it must be smaller than the minimum period of oscillation of the system. For example, when simulating an atomic fluid, the time-step should be smaller than the collisions mean-time. For flexible molecules the time-step should be in general 1/10 of the time of the shortest period of motion. This minimum period is due to bond stretches, vibrating with a period of the order of ten femto-seconds. This is clearly a severe restriction to molecular dynamics computational efficiency as these high-frequency motions are usually of relatively little interest and have a minimal effect on the overall behaviour of the system. To overcome this computational limitation the faster vibrations can be freezed out, constraining the bonds to their equilibrium values. This enables a longer time step to be used. Several methods are available. A brief statement is given in section 2.5.3.

2.5.3 Constraint algorithms

In molecular dynamics simulations of flexible molecules, the high frequency motions (e.g. bond vibrations) are usually of less interest than the lower frequency motions, corresponding, in general, to major conformational changes. Moreover, high frequency motions negatively influence the maximum time-step allowable. It would therefore be of considerable benefit to be able to

increase the time step without prejudicing the accuracy of the simulation. This can be done using the so called *constraint algorithms* which fix the faster degrees of freedom of the system during the simulation.
A constraint algorithm imposes the equations of motion to satisfy constraint conditions. The most commonly used constraint method is the *SHAKE* algorithm developed by Ryckaert et al. [31].

The equations of motion for a constrained system involve, besides the forces due to bonded and non-bonded interactions, also the forces due to the constraints. If we want to constraint each bond length to a constant value, the constraint conditions are as follows:

$$\sigma_{ij} = (\mathbf{r}_i - \mathbf{r}_j)^2 - d_{ij}^2 = 0 \tag{2.46}$$

From these conditions, constraint forces have to be generated. The constraint force will lie along each bond and will be at all times equal and opposite to the force acting on the two atoms joined by the bond. The overall effect is that the constraint forces do no work.
The constraint forces are obtained by differentiating the constraint conditions with respect to the coordinates of atoms joined by the bond and multiplying by an undetermined Lagrange multiplier λ (as we can see in equations 2.47 and 2.48).

$$\lambda \frac{\partial \sigma_{ij}}{\partial \mathbf{r}_i} = 2\lambda (\mathbf{r}_i - \mathbf{r}_j) \tag{2.47}$$

$$\lambda \frac{\partial \sigma_{ij}}{\partial \mathbf{r}_j} = -2\lambda (\mathbf{r}_i - \mathbf{r}_j) \tag{2.48}$$

This forces can be introduced, for example, in the Verlet algorithm (equation 2.38) that become:

$$\mathbf{r}_i (t + \delta t) = 2\mathbf{r}_i (t) - \mathbf{r}_i (t - \delta t) + \frac{\mathbf{F}_i (t)}{m_i} \delta t^2 + \sum_{bonds} \frac{\lambda_{ij} \mathbf{r}_{ij} (t)}{m_i} \delta t^2 \tag{2.49}$$

Obviously in equation 2.38 the term $a(t)$ is substituted by the term $\frac{\mathbf{F}_i(t)}{m_i}$ i.e. the total force acting on the atoms i.

The last thing to do is to determine the Lagrange multipliers λ_{ij} in such a way that all the constraints are satisfied simultaneously. This can be done algebraically.
Suppose that we want to find the Lagrange multipliers relative to a molecules constituted by two atoms, labelled 1 and 2. We can write the two equations 2.49 for such atoms. Moreover, we can write also a third equation derived from the equation 2.46 and that states that at time $t + \delta t$ the constraint conditions have to be fulfilled (equation 2.50).

$$(\mathbf{r}_i (t + \delta t) - \mathbf{r}_j (t + \delta t))^2 = (\mathbf{r}_i (t) - \mathbf{r}_j (t))^2 = d_{ij}^2 \tag{2.50}$$

Thus we have three equations in three unknowns $\mathbf{r}_i\left(t+\delta t\right)$, $\mathbf{r}_j\left(t+\delta t\right)$ and λ_{ij} that solved give the positions of the atoms i and j at time $t+\delta t$ and the Lagrange multiplier. Obviously these operations become harder and harder to solve as the number of constraints grows, as all the equations of motion must be solved simultaneously and are in general hardly coupled.
The SHAKE algorithm uses an alternative procedure to solved these equations: The constraints are considered in turn and solved as we have just seen. The decoupling of the equations of motion causes that satisfying one constraint means let another constraint to be violated. Therefore an iterative procedure is adopted by SHAKE algorithm until all the constraints are satisfied to within some tolerance.
The tolerance should be tight enough to ensure that the fluctuations due to the SHAKE algorithm are much smaller than the fluctuations due to other sources, such as the use of cut-offs.

An important condition that must be fulfilled to applied constraint algorithms without any troubles is that the constrained degrees of freedom should be only weakly coupled to the remaining degrees of freedom. This is because the motion of the molecules have to be almost not influenced by the application of the constraints.

2.6 Molecular dynamics at constant temperature and at constant pressure

The topics treated until now allow us to perform molecular dynamics simulation in constant NVE ensemble. This means that during the simulations the number of particles, the volume and the energy of the system are kept constant. The properties derived from such simulations can be transformed between ensembles in the limit of infinite system size, but it is not possible to do that in all practical applications. Therefore, it is desired to perform the simulations in different ensembles, such as the constant NVT and the constant NPT ensembles (for example in the case of liquid simulations [17]). In this section we will consider the simulation methods to perform molecular dynamics simulations in constant temperature 2.6.1 and/or constant pressure 2.6.2.

2.6.1 Molecular dynamics at constant temperature

The mean reason for which one could desire to perform simulations at constant temperature is to determine the properties of a system varying temperature, for example if we want to analyse glass transition or conformational properties.
The temperature of a system is a macroscopic property related to the time

average of the kinetic energy (equation

$$\langle \mathscr{K} \rangle_{NVT} \quad = \quad \frac{3}{2} N k_b T \tag{2.51}$$

in which N is the number of particles, k_b is the Boltzmann constant.
The simplest way to modify the temperature of a system, and therefore also to keep the temperature constant, is to scale the velocities [32]. This method is called *velocity scaling* method. The temperature change ΔT can be carried out multiplying the velocities by a factor λ. The associated temperature change can be computed as follows:

$$\Delta T \quad = \quad \frac{1}{2} \sum_{i=1}^{N} \frac{2}{3} \frac{m_i \left(\lambda v_i\right)^2}{N k_b} - \frac{1}{2} \sum_{i=1}^{N} \frac{2}{3} \frac{m_i v_i^2}{N k_b} \tag{2.52}$$

with obvious meaning of the symbols.
Equation 2.52 can be rewritten in a more compact form as:

$$\Delta T \quad = \quad \left(\lambda^2 - 1\right) T\left(t\right) \tag{2.53}$$

from which we can calculate λ:

$$\lambda \quad = \quad \sqrt{\frac{\Delta T - T\left(t\right)}{T\left(t\right)}} = \sqrt{\frac{T_{required}}{T_{current}}} \tag{2.54}$$

An alternative method, developed by Berendsen et al. [33], is to couple the system to an *external heat bath*, often called *Berendsen Thermostat.* The equation of the bath is the following:

$$\frac{dT}{dt} \quad = \quad \frac{1}{\tau} \left(T_{bath} - T\left(t\right)\right) \tag{2.55}$$

where τ is a coupling parameter. The bath acts as a source of thermal energy exchanging heat with the system to keep the temperature constant. The velocities are scaled at each step. The rate of change is proportional to the difference between the temperatures of the bath and of the system. From equation 2.55, the expression of temperature variation can be written (see equation 2.56).

$$\Delta T \quad = \quad \frac{\delta t}{\tau} \left(T_{bath} - T\left(t\right)\right) \tag{2.56}$$

and from the equation 2.52 [7](or equation 2.53 that is the same thing) the scaling factor λ can be computed:

$$\lambda^2 \quad = \quad 1 + \frac{\delta t}{\tau} \left(\frac{T_{bath}}{T\left(t\right)} - 1\right) \tag{2.57}$$

[7] Equation 2.52 is still valid as it derives from the general relationship 2.51, between velocities and temperature of a system

As the coupling parameter value is high as the coupling between the bath and the system is weak. When $\tau = \delta t$ the Berendsen thermostat is equivalent to the velocity scaling method. In general suitable value of the coupling parameter, using a time-step of 1 fs, is 0.5 ps.
The mean advantage of the Berendsen thermostat is that, differently from velocity scaling method, allows the temperature to fluctuate around the desired temperature value. The mean drawback of this method is that it do not allow the temperature difference between the components (e.g solute and solvent) of the system to be deleted. Even if a double Berendsen thermostat is applied (one for each components) such method do not allow an equal distribution of energy.

The method that overcome this drawback the method of *extended system* can be used. As it was developed by Nosè [34] and Hoover [35], this method is also called *Nosè-Hoover thermostat.*
In Nosè-Hoover method a heat reservoir is introduced as an additional degree of freedom, s. This degree of freedom represent a scaling factor of velocities. Therefore the velocities of the particles in the *extended* system, v_i, are related to the time derivatives of the positions, $\dot{r}$ (that now don't represent the velocities of the particles) by the following expression:

$$v_i = s\frac{dr}{dt} \tag{2.58}$$

The heat reservoir has the following potential energy: $\mathscr{V}_s = (f+1)\,k_bT\ln s$, in which k_b is the Boltzmann constant, f is the number of degree of freedom of the system and T is the desired temperature. Moreover, it has the following kinetic energy: $\mathscr{K}_s = (Q/2)\,\dot{s}^2$, in which Q acts as thermal inertial parameter (with dimensions of $(energy)\,(time)^2$) and controls the rate of energy fluctuations. The modified equations of motion of the extended system can be now obtained from the *Lagrangian* function of the system:

$$\mathscr{L}_s = \sum_N \frac{1}{2}m_i\dot{r}_i^2 s^2 + \frac{Q}{2}\dot{s}^2 - \mathscr{V} - (f+1)\,k_bT\ln s \tag{2.59}$$

From which the Newton's equations of motion for the extended system can be derived[8]:

$$\ddot{r}_i = \frac{\mathbf{F}}{ms^2} - \frac{2\dot{s}\dot{r}}{s} \tag{2.60}$$

$$Q\ddot{s} = \sum_i m_i\dot{r}_i^2 s - \frac{(f+1)\,k_bT}{s} \tag{2.61}$$

That have to be integrated through a standard numerical integration method. The adjusting parameter of the system is Q, that have to be chosen in an

[8] For widening see appendix B

appropriate way. Too high values of Q in fact result in slow energy flow between the system and the reservoir (for $Q \to +\infty$ we have a conventional molecular dynamics). On the contrary, if Q is too low, high oscillations of the energy occur, resulting in poor equilibration. In general Q is chosen by trial and error.

2.6.2 Molecular dynamics at constant pressure

Many experimental measurements are made under conditions of constant temperature and pressure, and so simulations in the isothermal-isobaric ensemble are most directly relevant to experimental data. Moreover, structural arrangements can be often more easily achieved in simulations at constant pressure than in simulations at constant volume. In many applications, for example, phase transitions are induced by pressure variation.

As the pressure is related to the *virial* of the system, $\sum r_i \frac{dV_i}{dr_i}$, it fluctuates much more than quantities such as the total energy in a constant NVE molecular dynamics simulation. The virial in fact, shows high variations respect to the internal energy.

To maintain a system at constant pressure (and/or at constant temperature) we have to change its volume. Volume fluctuations are related to the isothermal compressibility, k, that is defined by equation 2.62.

$$k = -\frac{1}{V}\left(\frac{dV}{dP}\right)_T \tag{2.62}$$

The isothermal compressibility is related to the mean square volume displacement by the equation 2.63.

$$k = \frac{1}{K_b T}\frac{\langle V^2\rangle - \langle V\rangle^2}{\langle V^2\rangle} \tag{2.63}$$

A large value of k means a more compressible substance and therefore, in a simulation at constant pressure, it cause a large volume fluctuation.

A volume change in an isobaric simulation can be achieved by changing the volume in all directions, or in just one direction.

The methods used for pressure control are similar to those for temperature control: the pressure can be controlled by scaling the volume or by coupling the system to a *pressure bath* as in [11].

The rate of change of pressure is given by:

$$\frac{dP}{dt} = \frac{1}{\tau_P}\left(P_{bath} - P\left(t\right)\right) \tag{2.64}$$

in which τ_P is the coupling constant, P_{bath} is the pressure of the bath, and $P\left(t\right)$ is the pressure at time t. From equation 2.64, the scaling factor, λ_P can be derived(see equation 2.65).

$$\lambda_P = 1 - k\frac{\delta t}{\tau_P}\left(P_{bath} - P\left(t\right)\right) \tag{2.65}$$

The system volume is thus scaled by a factor λ, which means that the atomic coordinates have to be scaled according to the following expression.

$$r_i' = \lambda^{\frac{1}{3}} r_i \tag{2.66}$$

The method described can be applied to scale the pressure of the system isotropically (which means that the volume of the system varies of the the same amount in all the directions) or anisotropically. In general, the anisotropic approach is the most widely used as it allows the volume to change independently in the three directions.

The aim of pressure control can be also reached through the extended pressure-coupling method introduced by Anderson, analogously to temperature coupling [36]. The method is based on the introduction of an extra degree of freedom, corresponding to the volume of the system. The kinetic energy and potential energy associated with this degree of freedom are introduced in a analogous way. Analogous procedures lead to the obtainment of the extended molecular dynamics equation of motion (see page 67 or see reference [17] for details).

2.7 Analysis methods

Molecular dynamics simulations generate information at the microscopic level (i.e. positions, velocities etc.) that have to be converted into macroscopic properties (e.g. pressure, internal energy). This can be derived using the concepts of statistical mechanics. In this section therefore a brief treatment of such topics will be carried out.

Radial distribution function Molecular dynamics can lead to structures formation or to changes in molecules conformations because of molecules interactions. Such changes are important because have a great influence on the macroscopic properties of materials. Since molecules are affected by continuous motions (e.g. Brownian motions) structural and conformational concepts acquire an average meaning.

Molecular conformational analyses can be carried out using the so called *radial distribution function* [36].

Radial distribution functions are a useful way to describe the structure of a system, as well as in the case of a liquid system. Consider a spherical shell of thickness δr at a distance r from a chosen atom (see figure 2.17). The volume of the shell is given by:

$$V = \frac{4}{3}\pi \left[(r+\delta r)^3 - r^3 \right] \simeq 4\pi r^2 \delta r \tag{2.67}$$

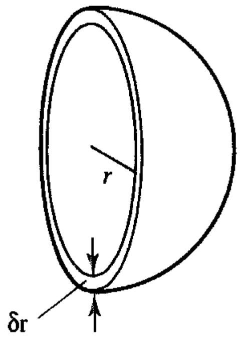


Figure 2.17: Radial distribution function.

If the number of particles per unit volume is ρ, then the total number of particles in the shell is $4\pi\rho r^2\delta$. The pair radial distribution function[9], $g\left(r\right)$, gives the probability of finding a particle at a distance r from another particle, compared to the ideal gas distribution. $g\left(r\right))$ is therefore given by the following equation (2.68):

$$g\left(r\right) \quad = \quad \frac{N_{\text{shell}}\left(r\right)}{N_{\text{ideal}}} \tag{2.68}$$

Radial distribution function is thus dimensionless.

In a crystal, the radial distribution function has an infinite number of sharp peaks whose separations and heights are characteristic of the lattice structure. The radial distribution function of a gas has a constant value, generally 1 because of the normalization, as gas molecules do not form structures. The radial distribution function of a liquid is intermediate between the solid and the gas, with a small number of peaks at short distances, superimposed on a steady decay to a constant value (equal to 1) at longer distances. As an example, the radial distribution function calculated from a molecular dynamics simulation of liquid argon, reported in figure 2.18, shows the typical liquid behavior.

The radial distribution function is a powerful tool to analyse structures formation in molecular dynamics simulations. A clear advantage is that it can be easily calculated from molecular positions data. In some cases, it can be computed experimentally using X-ray diffraction techniques [36]. Moreover, thermodynamic properties can be calculated using the radial distribution function, if pairwise additivity of the forces is assumed. These properties are usually given as an ideal gas part plus a real gas part. For example,

[9] Higher radial distribution functions (e.g. the triplet radial distribution function) can also be defined but are rarely calculated and so references to the *radial distribution function* are usually taken to mean the pairwise version.

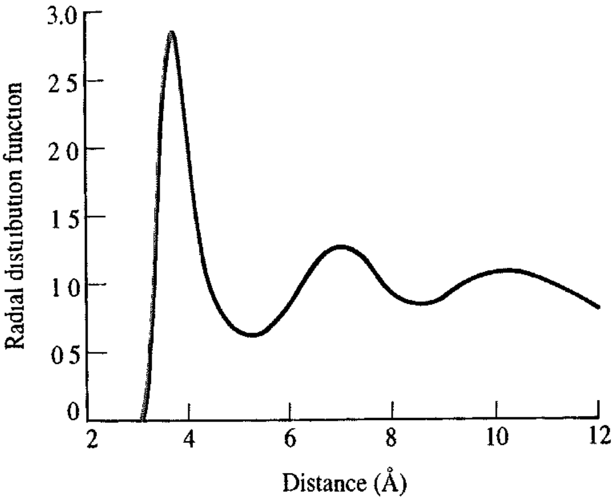


Figure 2.18: Radial distribution function of liquid argon.

to calculate the energy of a real gas, we consider the particles $4\pi\rho g\left(r\right)r^2\delta$ contained in a spherical shell of volume. If the pair potential at a distance r has a value $\mathscr{V}\left(r\right)$ then the energy of interaction between the particles in the shell and the central particle is $4\pi\rho\mathscr{V}\left(r\right)g\left(r\right)r^2\delta r$. The total potential energy of the real gas is obtained by integrating this between 0 and ∞ and multiplying the result by $N/2$. The total energy is thus:

$$E = \frac{3}{2}Nk_BT + 2\pi N\rho\int_0^\infty \mathscr{V}\left(r\right)g\left(r\right)r^2dr \tag{2.69}$$

In a similar way the following expression for the pressure can be derived:

$$PV = Nk_BT - \frac{2\pi N\rho}{3k_BT}\int_0^\infty r\frac{d\mathscr{V}\left(r\right)}{dr}r^2g\left(r\right)dr \tag{2.70}$$

This tool can be thus used to evaluate structures formations in a triglyceride box, due to temperature annealing and imposed flow, in the following.

Mean squared displacement In a molecular dynamics simulation, the molecules are affected by random motions and do not stay in fixed positions. The mean squared displacement (MSD) [66], describes on average the displacement of the molecules from their initial position (see equation 2.71)

$$MSD\left(t\right) = \langle\Delta r_i\left(t\right)^2\rangle = \langle\left(r_i\left(t\right)-r_i\left(0\right)\right)^2\rangle \tag{2.71}$$

If each molecule do not encounter other molecules, during a time interval, the mean squared displacement will be proportional to such interval and will grow quadratically with time. On the contrary, in the case of dense phases, the mean squared displacement will have an initial quadratic behavior (the duration is the mean time of collisions) and finally will assume a linear shape, characteristic of Brownian motions. In the case of simulations of solid systems the mean squared displacement will show a plateau [66]. The mean squared displacement allows the calculation of self-diffusion through the following formula (also called Einstein formula):

$$\lim_{t\to+\infty} \frac{d\langle \Delta r_i (t)^2}{dt} = 2d\mathscr{D} \tag{2.72}$$

in which d is the space dimension and $\mathscr{D}$ is the self-diffusion.

Correlation functions Given two quantity x and y deriving from a molecular dynamics simulation (e.g. positions or velocities), the correlation function between these two quantities can be evaluate in the most general form [36] as:

$$C_{xy} = \frac{1}{N}\sum_{i=1}^{N} x_i y_i \equiv \langle x_i y_i \rangle \tag{2.73}$$

The correlation function very often is normalized to have values belonging the interval $[0;1]$. High values of C_{xy} indicates high correlation while a zero value means completely uncorrelated quantities. If $x \equiv y$ the function is called autocorrelation function. The autocorrelation functions are useful because through them some transport properties can be determined such as viscosity, thermal conductivity. Viscosity, for example, can be calculated from the following formula:

$$\eta = \frac{V}{k_B T}\int_0^{\infty} \langle P_{\alpha\beta}(t)\, P_{\alpha\beta}(0)\rangle dt \tag{2.74}$$

Regarding viscosity computations, the mean drawback in using the formula 2.74 is that this method converge very slowly and depends greatly on pressure fluctuations. To overcome this disadvantage in the following paragraph a non-equilibrium method is described.

Viscosity calculation The non-equilibrium method for determining viscosity is based on the fact that energy, which is introduced into the system by external forces, is dissipated through viscous friction.The generated heat is removed by coupling to a heat bath. For a Newtonian liquid adding a small force will result in an acceleration gradient according to the following equation:

$$a_x(z) + \frac{\eta}{\rho}\frac{\partial^2 v_x(z)}{\partial z^2} = 0 \tag{2.75}$$

In equation 2.75 an acceleration $a_x(z)$ have been applied in x-direction. This acceleration depends on z-coordinate. A typical acceleration function (that used in the following chapters) is:

$$a_x(z) = A\cos\left(\frac{2\pi z}{l_z}\right) \tag{2.76}$$

in which l_z is the dimension of the box along z axis.

Some of this tools will be used in the following chapters to analyse structural characteristics of the molecular dynamics simulations. In the last chapter 4 new tools and related software for the analysis of molecules conformations of triglycerides will be developed.

Chapter 3

An atomistic model of triglycerides

3.1 Introduction

Although many experimental works have shown that solid tryglicerides are organized in a lamellar arrangements attained by glycerolipids, they are clearly not amphiphilic in nature. Moreover, in many practical applications triglycerides are present in mixture with diacylglycerol and monoacylglycerol [61]. Therefore we are interested to build an atomistic model of a bulk phase of triglycerides in which water is not present. This means to eliminate various source of complexities typical of the polar lipids such a head-group dependence, dependence on degree of hydration, adequate representation of polar solvent and ionic strength. In the next sections the atomistic model is described. The model is then validated using several properties described in section 2.7. Experimental physical properties available in literature will be compared with those numerically derived. As will be seen, the atomistic model is able to describe with adequate accuracy the behavior of a liquid phase varying system conditions. In the next chapter (4) suitable distribution functions will be used to derive from the beginning a coarse-grained model reproducing structural properties of the atomistic one.

3.2 Mathematical model

The first example of molecular dynamics simulation of triglycerides is the work of Chandrasekhar and van Gusteren in 2002 [39]. In their paper, they present constant pressure molecular dynamics simulations of a double-layer of α phase of trioctanoin triglyceride molecules, using several united-atoms parameters sets. Since the objective of the work is to test parameters for simulation of a solid alpha phase, they state the initial conditions of the molecular dynamics simulations in such a way that they reproduce as much as possible the molecular orientation of the alpha phase.

As we want to have a molecular dynamics model able to reproduce phases structure of triglycerides, the best choice to do is to use Chandrasekhar parameters to build a united-atoms model of a bulk phase of triglycerides. The model has to be used in the following chapter (chapter 4) as a numerical experiment through which develop a new coarse-grained model.

3.2.1 United-atoms

The united-atoms used by Chandrasekhar derive by GROMOS96 [19] model. The GROMOS96 is a very popular force field for simulation of alkanes and the parameters set are therefore further refined during time [41]. Moreover, GROMOS96 parameters have been tested widely through experimental data and *ab initio* calculation and thus this is a further motivation of why appears to represent in our opinion the best choice to do.
The groups derived from alkanes (i.e. CH_i groups) are treated by Chandrasekhar as united-atoms, whose interactions are described through a suitable modified GROMOS96 force field, as will be seen in detail in next sections (i.e. 3.2.3 and 3.2.3). The ester groups are modeled considering each single atom belonging to such group and a standard GROMOS96 force field is used in this case. A schematic tri-dimensional united-atoms representation of a triglyceride molecule is reported in figure 3.1. Red particles represent single oxygen atoms linked to a single carbon atom (which can be considered a united-atom of CH_0 type).

3.2.2 System definition

The system, we want to simulate, is constituted by 216 triglyceride molecules of tridecanoin (also called tricaprin $C_{33}H_{62}O_6$). Each of them constituted by 39 united atoms, hence the system is made of 8424 particles interacting each other. The simulation box is constructed, using the GROMACS software function *genconf* [11], which draws a cubic box of randomly rotated molecules from the coordinates of a single molecule. The intial dimension of the cubic box is set to 6.785 which gives an initial density value of $1200 Kg/m^3$. Thus, long equilibration time is avoided.
Initial velocities are imposed to the particles by a Maxwellian distribution random number generator at temperature of $446K$. We choose this particular temperature because we want to be sure to simulate triglyceride box in liquid conditions.
The box needs several minimizations to avoid particle overlapping. To this aim, the *steepest descendant* method is adopted. The time step have been fixed to $1.0nm$, while the energy minimum is reached with a tolerance of $10\frac{kJ}{mol \cdot nm}$.

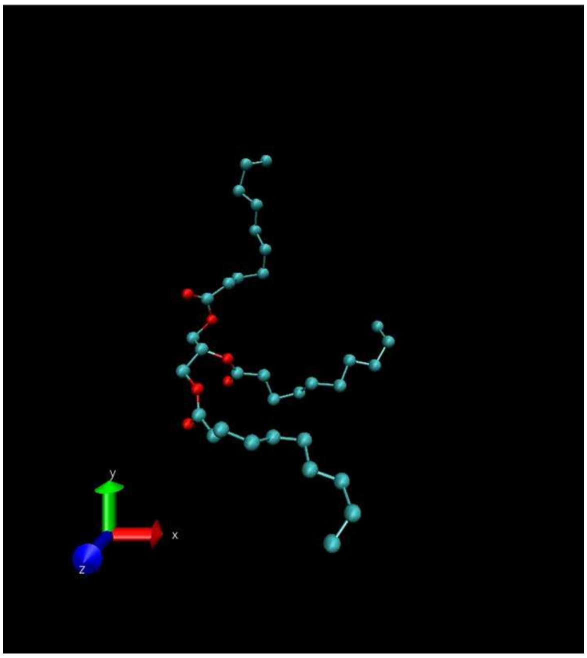


Figure 3.1: A united-atoms schematization of a molecule of triglyceride.

The equilibrated cubic box length, at a pressure of $1bar$ and at temperature of $446K$, is $6.142nm$. This will be important in the following for the definition of cut-off lengths.

3.2.3 Force field

Non-bonded interactions

GROMOS96 force field uses Lennard-Jones $12-6$ potentials for the treatment of non-bonded interactions. In table 3.1 such parameters are shown. The electrostatic interactions are computed through Coulomb potentials 2.15. The partial charge of each united-atom is shown in table 3.1. The CH_2 united-atoms present partial charges only when related to the glycerol group (i.e. no charge is applied for CH_2 related to alchilic chains).
The Lorentz-Berthelot mixing rule are used to calculate cross Lennard-Jones parameters.
Atoms within the triglyceride molecule that are close in the chain until three bonds are automatically excluded from non-bonded calculations. Until the third neighbor[1], in fact, the Lennard-Jones repulsions are still too strong. This means that if applied Lennard-Jones potentials to these atoms, the

[1] An $n-th$ neighbor is an atom that is far from a consecutive atom n bond

Particle	Mass (uma)	Charge (C)	σ (nm)	ε (kJ/mol)
CH_3	15.0350	0.00	$3.7480000 \cdot 10^{-1}$	$8.67200 \cdot 10^{-1}$
CH_2	14.0270	0.200	$4.0700000 \cdot 10^{-1}$	$4.10500 \cdot 10^{-1}$
CH_1	13.0190	0.200	$5.0190000 \cdot 10^{-1}$	$0.94890 \cdot 10^{-1}$
C	12.0110	0.540	$3.3610000 \cdot 10^{-1}$	$4.05900 \cdot 10^{-1}$
$O=$	15.994	-0.380	$2.6260000 \cdot 10^{-1}$	$17.2504 \cdot 10^{-1}$
$O-$	15.994	-0.360	$2.9550000 \cdot 10^{-1}$	$8.49600 \cdot 10^{-1}$

Table 3.1: Table of Chandrasekhar and van Gusteren Lennard-Jones parameters for united-atoms model.

Bond	l_0 (nm)	k' $(kJmol^{-1}nm^{-4})$
$CH_3 - CH_2$	0.153	$0.715 \cdot 10^7$
$CH_2 - CH_2$	0.153	$0.715 \cdot 10^7$
$CH_1 - CH_2$	0.153	$0.715 \cdot 10^7$
$C - CH_2$	0.148	$0.764 \cdot 10^7$
$CH_2 - O$	0.147	$0.871 \cdot 10^7$
$C - O$	0.133	$1.180 \cdot 10^7$
$C = O$	0.123	$1.660 \cdot 10^7$

Table 3.2: Table of Chandrasekhar and van Gusteren GROMOS96 pairwise potential parameters for united-atoms model.

molecule would deform and, at the end, break due to non-bonded repulsions.

The treatment of boundary conditions is performed through the introduction of periodic boundary conditions in the three directions. Hence, a cut-off have to be introduced to avoid an atom to interact with its own image.
The treatment of all non-bonded interaction are carried out using twin-range cut-off scheme (see page 52). The short-range cut-off radius is $0.8nm$, while the long-range cut-off radius is $1.4nm$. Within short-range cut-off radius the neighbor list is determined every time step, while outside the neighbor list is updated every five time step.

Bonded interactions

The treatment of two-bodies bonded interactions are obtain through GROMOS96 pairwise potential (equation 2.7), whose parameters values are shown in table 3.2. The three bonded potentials are taken into account, using equa-

Angle	θ_0 (deg)	k'_θ $(kJmol^{-1})$
$CH_2 - CH_1 - CH_2$	109.5	517.0
$CH_1 - CH_2 - O$	111.0	527.0
$CH_2 - CH_1 - O$	109.5	517.0
$CH_2 - O - C$	117.0	635.0
$CH_1 - O - C$	117.0	635.0
$O - C = O$	124.0	730.0
$O - C - CH_2$	115.0	610.0
$C - CH_2 - CH_2$	120.0	780.0
$CH_2 - CH_2 - CH_2$	111.0	527.0
$CH_3 - CH_2 - CH_2$	111.0	527.0
$O = C - CH_2$	121.0	685.0

Table 3.3: Table of Chandrasekhar and van Gusteren GROMOS96 three-bodies potential parameters for united-atoms model.

tion 2.10. In table 3.3 the relative parameters are reported. We could point out that as the ester group lies on a plane the sum of the angles, having the carbon atom C as a vertex, is equal to 360^o.
The torsional degrees of freedom of the system is described through the proper four-bodies bonded potential given by the equation 2.12. The relative parameters are reported in table 3.4. Because the model does not consider the presence of hydrogen atoms, further four-bodies potentials have to be introduce (see equation 2.14). This potentials, called improper dihedrals, permit the planar ester groups to remain planar and also prevent CH_2 particles and O particle linked to CH_1 united-atoms of the glycerol groups from flipping over to their mirror images. In table 3.5 are summarized the parameters for improper dihedrals calculations.

Simulations

For temperature control purpose, the box is coupled to a Berendsen thermostat with a relaxation time of $0.01ps$. An anisotropic Berendsen barostat is also used to fix pressure at $1atm$. The relaxation time used in this case is equal to $0.05ps$. Isothermal compressibility, $k_T = 0.01 \cdot 10^{-5}$ bar^{-1} is taken from literature (e.g. see [40]).
After the equilibration of the system, bond lengths were constrained to the ideal values through SHAKE algorithm with a geometric tolerance of 10^{-4}. The system is integrated by means of a leap-frog algorithm in GROMACS software package [21], whose time step is set to 0.002 ps.

Prop. Dihedral	θ_0 (deg)	k_φ $(kJmol^{-1})$	n
$CH_2 - CH_1 - CH_2 - O$	0.0	5.85	3
$CH_1 - CH_2 - O - C$	0.0	3.76	3
$CH_2 - CH_1 - O - C$	0.0	3.77	3
$CH_2 - O - C = O$	180.0	16.74	2
$CH_1 - O - C = O$	180.0	16.74	2
$O - C - CH_2 - CH_2$	0.0	0.42	6
$C - CH_2 - CH_2 - CH_2$	0.0	5.86	3
$CH_2 - CH_2 - CH_2 - CH_2$	0.0	5.86	3
$CH_3 - CH_2 - CH_2 - CH_2$	0.0	5.86	3
$CH_2 - O - C - CH_2$	180.0	16.7	2
$O = C - CH_2 - CH_2$	180.0	2.9	2
$O - CH_1 - CH2 - O$	180.0	2.09	2

Table 3.4: Table of Chandrasekhar and van Gusteren GROMOS96 four-bodies potential parameters for united-atoms model.

Imp. Dihedral	ξ_0 (deg)	k_ξ $(kJmol^{-1}rad^{-2})$
$C \quad O- \quad CH_2 \quad = O$	0.0	167.42
$CH_1 \quad -O \quad CH_2 \quad CH_2$	35.264	335.00

Table 3.5: Table of Chandrasekhar and van Gusteren GROMOS96 improper four-bodies potential parameters for united-atoms model.

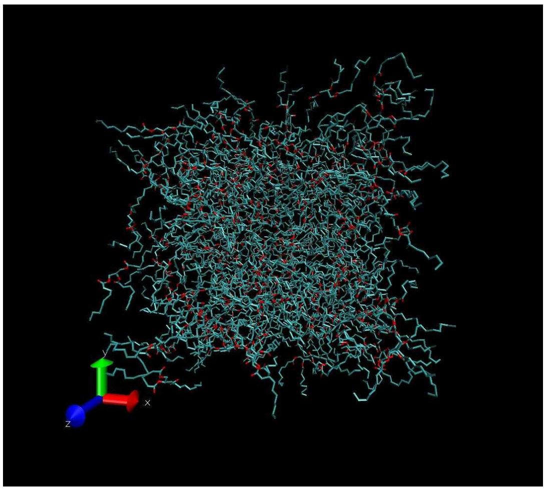


Figure 3.2: A box volume of tridecanoin representing a bulk of the liquid phase.

3.3 Integration of the model and validation

The united-atoms model of a bulk volume (e.g that relative to tridecanoin molecules shown in figure 3.2) of triglycerides represents the higher detailed microscopic model usable for the aim of analysis the behavior of triglycerides. However, this model is not adequate to stress the relationship between flow condition and molecules conformation because of the very high computational efforts. More precisely, we need a day to simulate the evolution of the system in a temporal interval of $3ns$. Therefore, go towards the time scales in which flow phenomena have a perceptible influence on molecules conformations (i.e. $200ns$ or more), means to perform a simulation that lasts 70 days. That's why this model will be overcome in the following chapter by the new coarse-grained model. Even so, the atomistic model can be useful to calculated some simple (i.e. that do not require long time interval simulations) macroscopic properties. In this field, as could be foreseen (see chapter 2 on page 2.1), the atomistic model gives better results compared with the coarse-grained model. In this section some properties of the volume of triglycerides owing to the bulk of the liquid phase are derived for the purpose of model validation.

3.3.1 Density Determination

The density of the system is one of most easy property to calculate. It is easy in fact to obtain the total mass of the system and the box dimension. Even so, it is one of the most important to state the validity of a microscopic model, since it is related to whole interaction between the particles. The experimental data are taken from the paper of Goodrum and Eiteman [67]. In figure 3.3 a diagram of the calculated and experimental density, taken

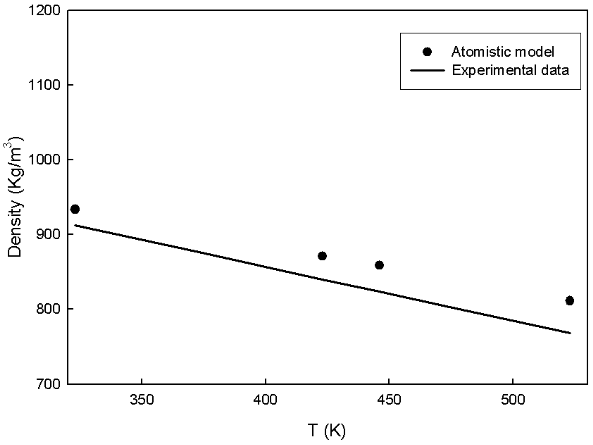


Figure 3.3: Experimental and atomistic molecular dynamics derived density for liquid tridecanoin.

from Goodrum and Eiteman paper, is shown. The density of the system at the initial condition (i.e. at temperature of $446K$ is $858.6kg/m^3$. Taking into account the level of detail of the atomistic model, it can be considered, to all practical purposes, equal to the experimental value of $823.9kg/m^3$ reported by Goodrum and Eiteman.

The diagram shows good agreement between data at several temperatures. This is the most evident proof of the validity of the atomistic model. However, several tests were carried out to better characterize the model.

3.3.2 Radial distribution functions

Radial distribution functions can be used to characterize the behavior of the simulated box regarding non-bonded interactions. Radial distribution functions (defined on page 70) deal with the distributions of molecules in the simulation box and therefore change significantly when phase changes (e.g. from liquid to solid) occur. The radial distribution functions of suitable non-bonded centers of mass are reported in figure 3.4. This means that only the relationship between centers of mass owing to different molecules are taken into account). The C symbols stand for suitable (as specified on page 95) non-bonded CH_i elements on the three triglycerides chains, while the N symbols stand for the centers of mass of the ester groups linked to the glycerol backbone. The diagram shows curves shapes typical of a liquid phase as expected. In particular the diagram shows the presence of several peaks in $C - C$ radial distribution function, not shown in the $N - N$ and shown

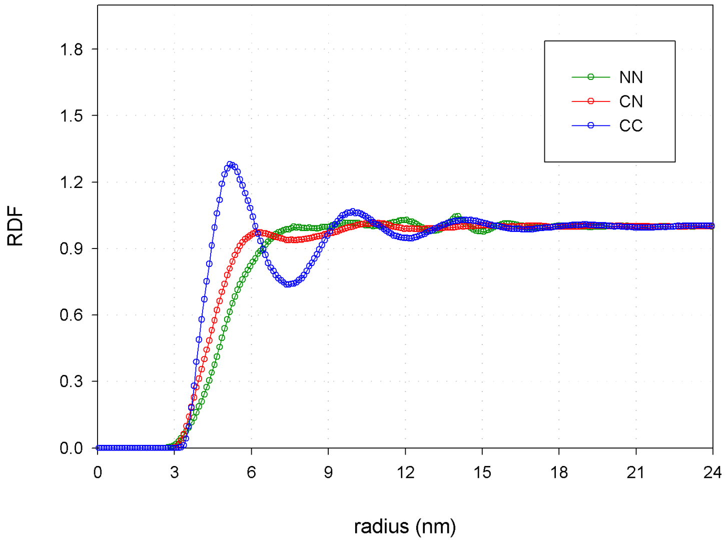


Figure 3.4: Atomistic radial distribution functions for the tridecanoin box.

with lower maxima in $N - C$ radial distribution. This is due to the high mobility of triglycerides chain in the liquid phase that cause the molecules of triglycerides to be placed side by side. Obviously the ester groups as linked to glycerol backbone are not influenced by triglycerides structures. Therefore the $N - N$ radial distribution function is similar to that of a liquid consisting on atoms or very small molecules.

3.3.3 Mean squared displacement

A further proof of physical state of the simulated system is given by the mean squared displacement[2] (defined on page 71) reported in figure 3.5. Through the mean squared displacement the self-diffusivity of the system can be computed (see equation 2.71). Equation 2.71 can be graphically interpreted as the slope of the linear part of the MSD curve. The computed self-diffusivity is equal to $0.89 \cdot 10^{-5} cm^2/s$. Unfortunately, we do not have experimental values for such property. Even so, it is of the same order of magnitude of liquid water and therefore can be considered an acceptable value.

[2] also known as MSD

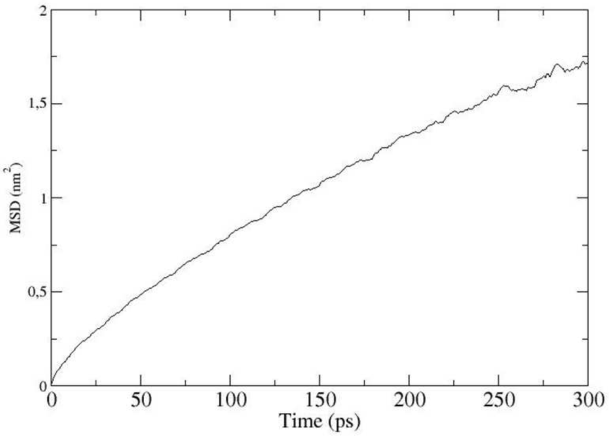


Figure 3.5: Mean squared displacement of the tridecanoin box.

3.3.4 Viscosity

Finally, we investigate the reliability of the atomistic model concerning with viscosities forecasts. The viscosity of the system varying temperature is reported in figure 3.6. It have been calculated using the non-equilibrium

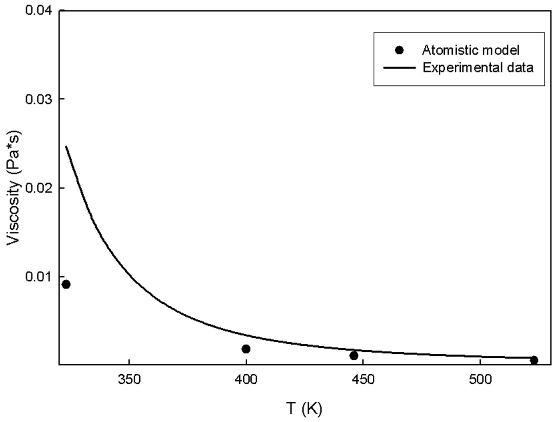


Figure 3.6: Mean squared displacement of the tridecanoin box.

method described on page 72. Moreover, the calculated values are also compared in the same figure with experimental data taken from the paper of Goodrum and Eiteman[67]. Calculated viscosity is in good agreement with experimental data for higher temperatures. For lower temperature it seems to move away from the experimental curve. This is probably due to the fact that as the temperature is reduced the system density grow (in the limit will have glass or crystal transition), therefore the hypothesis of Newtonian liquid[3] cannot be supported. Such hypothesis is at the basis of the used non-equilibrium method. This conjecture is confirmed by the fact that at $323K$

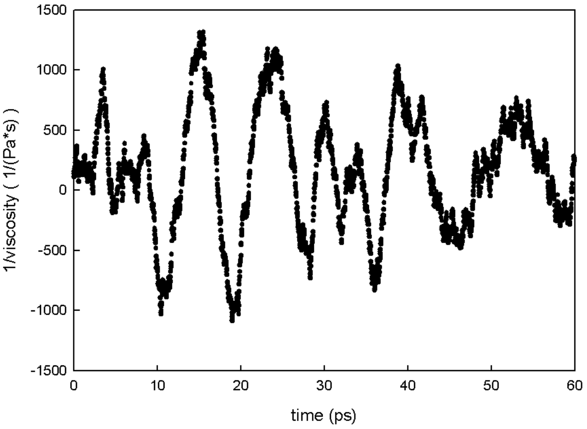


Figure 3.7: The instantaneous viscosity calculation varying simulation time.

the diagram of instantaneous viscosity varying simulation time 3.7 shows high oscillations. Therefore the mean value in this case cannot be taken as an acceptable value for the viscosity of the system.

3.4 Conclusions

In this chapter, the construction of the model of Chandrasekhar and van Gusteren [39] was described. It consists in a united-atom model, developed to analyse triglycerides boxes in a solid alpha phase. Here, we use the same model to the description of a box of 216 triglycerides (tridecanoin) molecules in liquid phase. The model is validated using statistical methods. First of all, radial distribution functions were used to analyse the non-bonded interactions between molecules. They show a quite random behavior of the glycerol backbones of the molecules, as expected in the case of a liquid phase. Moreover, the system shows density values close to experimental data. The agreement of densities data was demonstrated at several system temperatures. The mean squared displacement shows typical liquid diagram and the derived self-diffusivity shows a typical value. Finally, the viscosity was calculated. The viscosity dependence with temperature is shown to be in good

[3] The fact that liquid triglycerides behave as a Newtonian liquid is confirmed by experiments e.g. see [69]

agreement with the experimental data, only at higher temperature. A motivation of this behavior was given in the relative section. Since the atomistic model behavior is in quite good agreement with experimental data, in the chapter 4 it will be used to develop a new coarse-grained model.
The coarse-grained model will be then validated on the basis of a comparison with several statistical distributions belonging to the atomistic model. Moreover, in the same way, the predictive power of the new model will be checked.

Chapter 4

A coarse-grained model for triglycerides: Force field development and structures analysis

4.1 Introduction

Using the atomistic model treated in the chapter 3 we are able, theoretically, to calculate all the physical properties of a bulk of triglycerides e.g. density[1]. On the contrary, on a practical point of view, this model doesn't allow to determine properties that should be obtained with simulations evolving on long temporal scales. For example, flow conditions induce molecules conformations on time-scales longer than those reachable with atomistic models. That's why, if we want to study conformational behavior of triglycerides varying temperature and flow conditions, we need a microscopic model capable to investigate larger time-scale. In this chapter, we deal with a new coarse-grained model suitably developed for this aim. As a starting point, as we will see in most details in the following sections, the atomistic model described in chapter 3 is used. The derivation of the parameters of the corse-grained model to those of atomistic one could allow, in case, an easier matching of the two models in the outlook of a multi-scale model development.

In the next section (4.2) we start with the definition of the system, with the algorithm that lead to the definition of the basic units (i.e. *coarse-grained particles*) we will deal with. In section 4.3 we deal with the development of the force field by means of iterative procedures. A complete description

[1] This property, in particular, derives from intermolecular and intramolecular interactions and, as we will see, will be used as a test for the correctness microscopic models

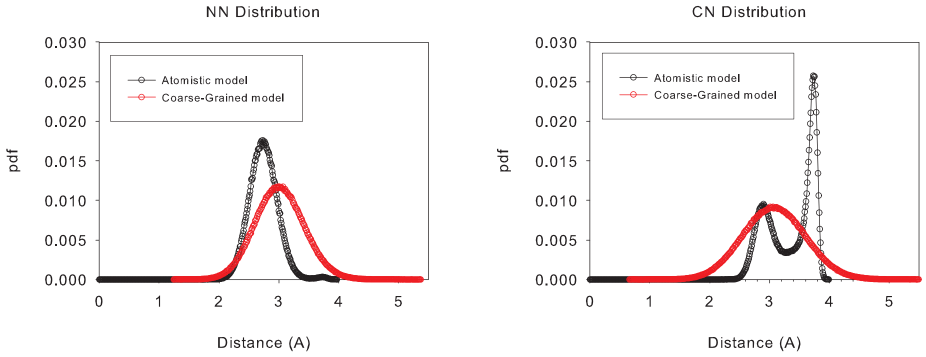
of the molecular dynamics model is described in section 4.4, in which are also reported temperature and pressure control characteristics and numerical integration algorithm parameters. Moreover, some details on equilibration technique and molecules box construction is given is section 4.3.2.
The analysis of the developed model is then carried out. In particular, in section 4.4.1, the model will be analysed in order to explain its prediction capabilities. The coarse-grained model was in fact developed to study molecules conformations and therefore, from the point of view of physical properties predictions, it may be considered a semi-quantitative model. The conformation analyses is then carried out in section 4.5 and in section 4.6, in the cases of no imposed flow and of imposed flow, respectively. To this aim suitable conformation parameters and related software have been developed as can be seen in section 4.4.2.

4.2 System definition: basic units

A coarse-grained model has to represent the *real* system particles (i.e. atoms) by means of *coarse-grained* particles. These particles represent suitable groups of atoms.
The coarse-grained particles can be create after that the description's level was chosen. To this aim, the following general rule can be adopted: The more the model is detailed (the limit goes towards quantum molecular dynamics) the more the simulations are able to reproduce experimental data, but the less the computation efforts are low.
According to this rule, as it's necessary to have, as much as possible, high detailed molecules simulations, the coarse-grained particles can derived grouping together united-atoms particles of the atomistic model analysed in the preceding chapter (3). That atomistic model simulations has, in fact, shown good agreement with experimental data[67].

The literature about coarse-grained model of triglycerides, to our knowledge is nonexistent. Even so, a good starting paper was written, few years ago, by Marrink et al. [37]. In their work, they developed a corse-grained model for semi-quantitative lipid (i.e. phospholipid) simulations from atomistic models. With the coarse-grained model they are able to reproduce densities of liquid alkanes with 5% error and solubilities of alkanes in water. Moreover, the model provides good results in simulations of phospholipids in water.
The coarse-grained model of Marrink is characterized by a reduction of degrees of freedom of the systems respect to the related atomistic models and by the use of short-range potentials, improving efficiency of the simulations. Moreover, the model provides the definition of only a small number of coarse-grained particles types.

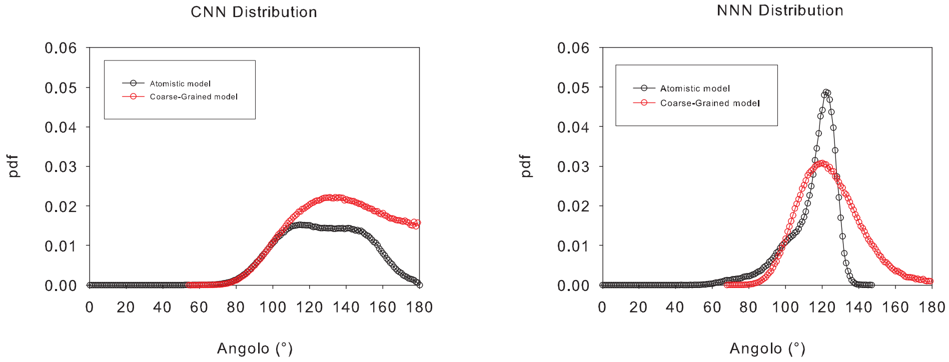
The Marrink model is not able to represent particles distribution of our atomistic model and therefore, it cannot be used to our aims. This can be easily shown comparing some significative probability distribution functions of the bonds and angles of the marrink model (red coloured) and of the atomistic model (black coloured)[2] as can be seen in figures 4.1 and 4.2.



(a) NN bond distributions

(b) CN bond distributions

Figure 4.1: Distribution functions of bond lengths: Comparison between Marrink and atomistic model.



(a) CNN angle distributions

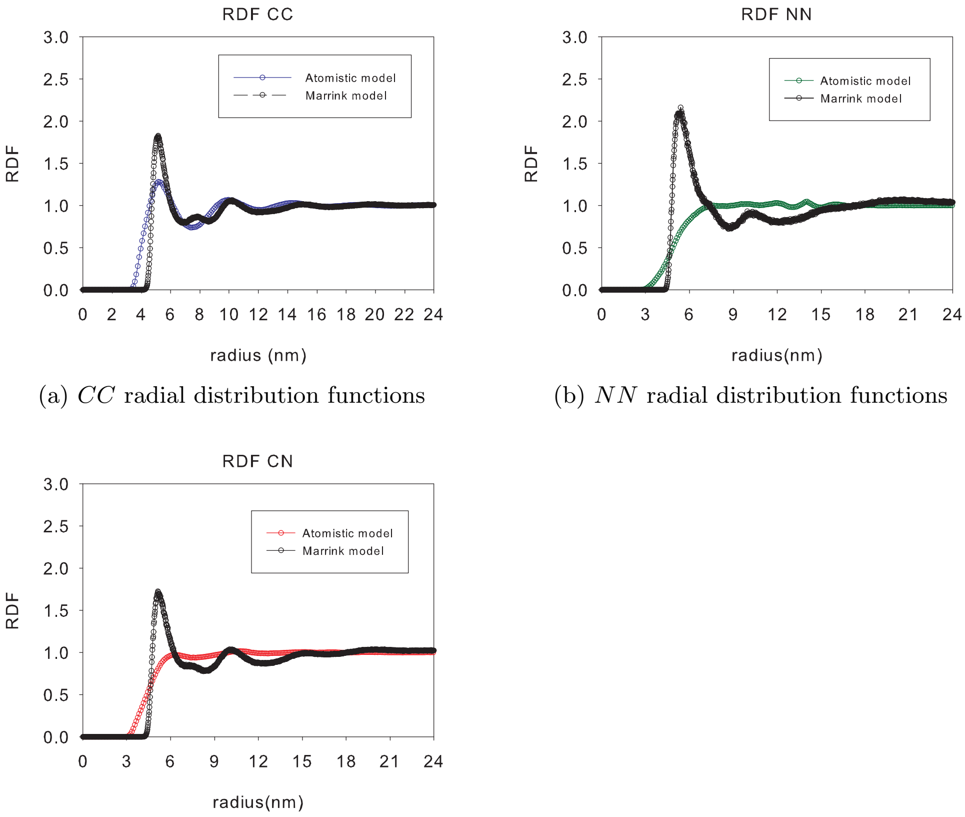
(b) NNN angle distributions

Figure 4.2: Distribution functions of angles: Comparison between Marrink and atomistic model.

In particular, the bond lengths (see figure 4.1) defined by Marrink et al. are less stiff and have wrong equilibrium values respect to that obtained through the atomistic simulations. Angles distributions derived by Marrink model are less stiff too (see figure 4.2). Moreover, the comparison between radial distribution functions of the Marrink model with that relative to the

[2] in the sense specified on page 95

atomistic model[3], show higher non-bonded interactions in the case of Marrink model as shown in figure



(a) CC radial distribution functions

(b) NN radial distribution functions

(c) CN radial distribution functions

Figure 4.3: Radial distribution functions of C and N particles: Comparison between Marrink and atomistic model

Even so, the main rules that governs Marrink model are chosen to be the guide lines to the development of our model for triglycerides, due to their simplicity.
Our model will be compared in section 4.3 to the Marrink model to point out the difference between the two models.

In figure 4.4 a schematic definition of the corse-grained particles in a simple case is reported. Triglyceride molecule is a quite simple molecule if seen from the point of view of the atomic groups. The three long arms can be represented by a set of coarse-grained particles of the same type, joint each other. Each particle are obtained merging together several CH_i groups (i.e. united-atoms). This kind of particle (drawn in cyan in figure

[3] derived in the way specified on page 95

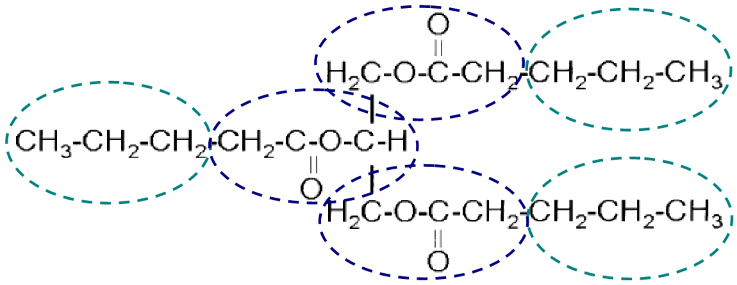


Figure 4.4: Coarse-grained particles for a triglyceride molecule.

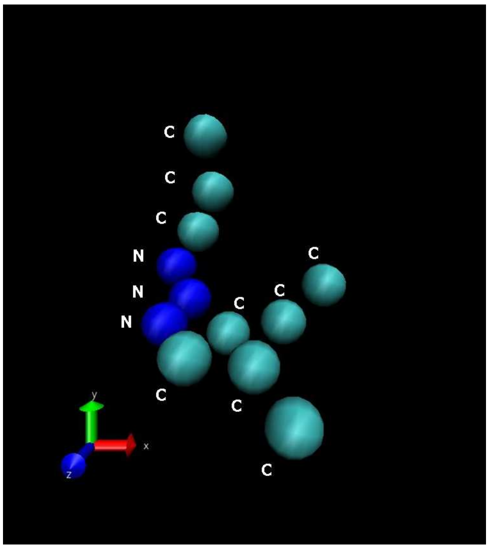


Figure 4.5: Coarse-grained representation of a triglycerides molecules.

4.4) will be named C, from now on. The other coarse-grained particles type is obtained merging a carbon of the carboxylic group with its substituents and with the related CH_i group deriving from glycerol (drawn in blue in figure 4.4). We will call that particle type as particle N. The united-atoms are merging together to form the coarse-grained particles in such a way that the net charge of the C and N particles (derived from the partial charges of the atomistic particles) is zero. The tri-dimensional coarse-grained representation of a triglycerides molecules is shown in figure 4.5.

In the operations described in the following, we refer to a system constituted by 216 triglyceride molecules of tridecanoin, as in the case of the atomistic model. Each of them constituted by 12 coarse-grained particles, hence the system is made of 2592 particles interacting each other. The simulation box is constructed, extracting the coarse-grained coordinate positions from the atomistic trajectories using a FORTRAN software suitable developed by us and reported in C. We wish to emphasize that the following calculations are carried out the first time in the case of a box of tributyrin molecules. As they lead to the same results, the treatment will be reported in the case of tridecanoin molecules only.

4.3 Force field development and model validation

4.3.1 Bonded and non-bonded interactions

The next step in the development of the force field is the definition of particles interactions.

As we have just seen in the chapter 2 several choice are possible for bonded and non-bonded potentials. For reason of computational efficiency an harmonic potential (see equation 2.5 on page 31) is chosen for the treatment of two-bodies bonded potentials. This choice allows the definition of two parameters k and l_0 instead of the three used in the case of Morse potential. The three-bodies bonded potentials are taken into account through the equation 2.10 on page 33. Also in this case is necessary the determination of two parameters k'_θ and θ_0 (from which derives $\cos\theta_0$).

No four-bonded bodies potential are defined. We can justify this approximation observing that the dihedral distribution of the centers of mass of the groups representing coarse-grained particles in a atomistic simulation are completely random (see appendix C for the relative fortran code).

Among the non-bonded interactions only the Van der Waals interactions have to be taken into account. In a coarse-grained model in fact only net charged groups, which are not present in triglycerides molecules, are considered, as can be seen in [37]. For the treatment of Van der Waals interactions (i.e. two-bodies non-bonded interactions), Lennard-Jones $12-6$ potentials are used (see equation 2.16 on page 39), as they are the most commonly used in literature to this aim. Therefore, this means to determine two parameters σ and ε. In the following the determination of such parameters will be carried out only for the interaction of coarse-grained particles of the same type. The Lennard-Jones parameters related to particles of different type (i.e. C–N interactions) are calculated through the Lorentz–Berthelot mixing rules shown in equations 2.22 and 2.23 on page 42.

4.3.2 Simulation conditions

Initial conditions As the development of the coarse-grained force field provides a trial and error procedure requiring several sets of simulations, the initial conditions have to be stated. That means that a simulation box volume have to be constructed. In this case the box volume is constructed from the equilibrated box of an atomistic simulation (see chapter 3), at certain temperature and pressure conditions. More precisely, in the atomistic model, the centers of mass of the groups from which the coarse-grained particles came from, can be derived by means of a Fortran code suitably developed by us (see appendix C). Each center of mass is labelled by the Fortran code as a precise coarse-grained particle.

Boundary conditions Before the system is minimized, we have to deal with the treatment of boundary conditions. Periodic boundary conditions along the three directions and then cut-off length are thus established. Cut-off length are fixed at a value of $1.5nm$ (the same used for atomistic simulations, according to Marrink rule of short range potentials. This cause also reduce highly computational efforts. Moreover, non-bonded neighbor list technique is adopted. For this purpose, neighbour list is updated every 10 integration steps.

Temperature and Pressure coupling For temperature and pressure coupling, Berendsen algorithms (are adopted see equations 2.55 on page 66 and 2.64 on 68, respectively). The coupling constant of the *Berendsen thermostat* is fixed at $\tau = 0.02ps$. The anisotropic pressure coupling algorithm has a coupling constant fixed to $\tau_P = 0.5ps$. Both values of coupling constants are typical and are used also in our atomistic simulations. The compressibility factors, taken from literature (see [39]), are equal to $1^{-5}bar^{-1}$. The reference pressure is $1bar$ for all the simulations performed, except where specified.

Constraint algorithm No constraint algorithms were applied in such simulations.

Equilibration of the system The simulation box have to be minimized until the bonded and non-bonded potentials parameters reach their *equilibrium* values. This is performed through the *steepest descendent* method (see page 52. The minimization algorithm uses a minimization step of $1nm$ and an energy tolerance of $10\frac{kJ}{mol\quad nm}$.

4.3.3 Force field non-bonded: parameters determination

The coarse-grained particles are characterized by several parameters: mass, charge and two Lennard-Jones parameters σ and ε). Mass is easily calculated summing atomic mass of all the atoms belonging to each coarse-grained particle. Charge can be computed in an analogous way but, as we have

Particle	Mass (uma)	Charge (C)	σ (nm)	ε (kJ/mol)
N	58.0368	0.00	$4.980000 \cdot 10^{-1}$	2.59000
$N2$	57.0288	0.00	$4.980000 \cdot 10^{-1}$	2.59000
C	43.0890	0.00	$4.190000 \cdot 10^{-1}$	1.29000

Table 4.1: Table of calculated Lennard-Jones coarse-grained parameters.

just said, the net charge for the defined particles is zero. σ and ε are the Lennard-Jones parameters related to two particle of the same type. The σ parameter are related to the dimension of the coarse-grained particle and of the whole system (i.e. volume or density) as a consequence. The strength of the interactions between two particle depends on ε.

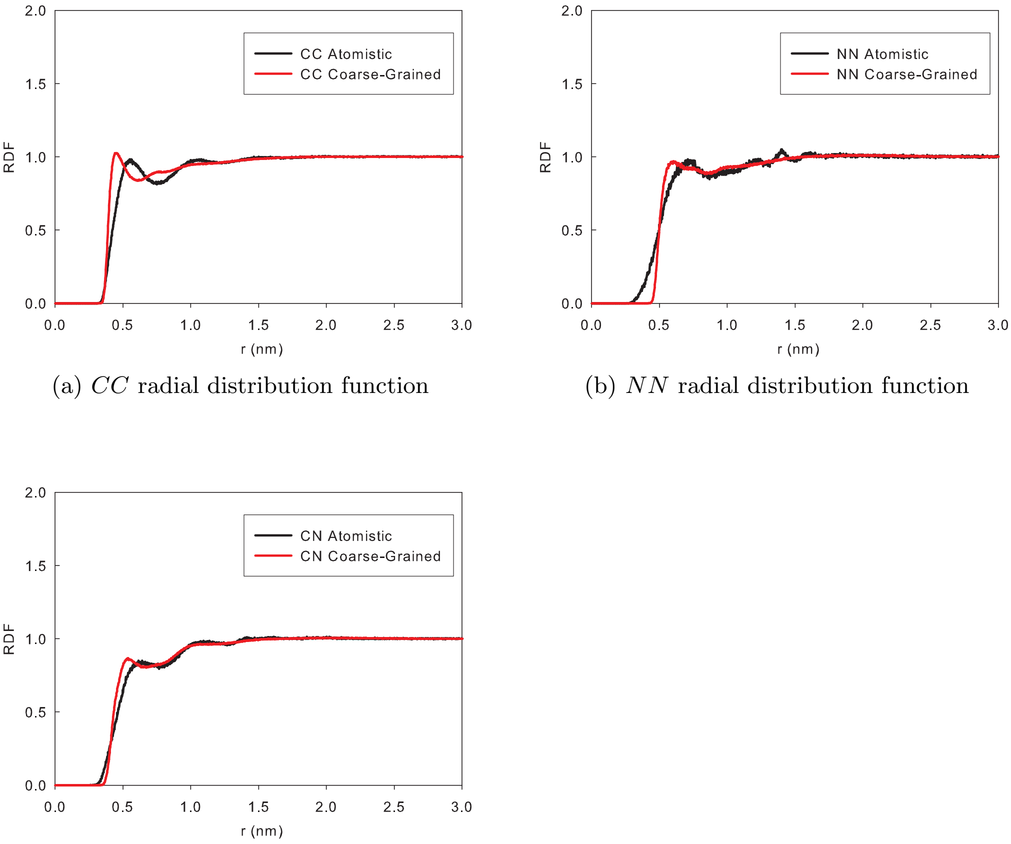
σ and ε parameters values, are determined using a trial and error procedure with the help of statical tools that provide the check of whole system density and of the radial distribution functions of C and N particles at each step. The dependence of the density from such parameters was just underlined. The radial distribution functions depends on the non-bonded interactions between every couple of particles types and therefore are strongly influenced by both the two Lennard-Jones parameters.

The Lennard-Jones parameters are determined in such a way that at the end the density of the coarse-grained system is equal to the density of the atomistic one (chapter 3). Moreover, the radial distribution functions of the coarse-grained model, related to $C-C$, $C-N$ and $N-N$ non-bonded interactions, matches those of the atomistic model. Obviously, in order to calculate these parameters, we have to fix the bonded parameters to plausible values. Moreover, we need also to fix the initial values for Lennard-Jones parameters. The values of Marrink model ([37]) are used as initial values for bonded and non-bonded parameters.

The determined values of Lennard-Jones parameters are summarized in table 4.1.

In table 4.1, the mass and charge values are also reported. The coarse-grained particle $N2$ is an N particle type that differs only because of one unit mass. It represent the particle of type N connecting the other two N particle in each molecule. It was introduced because we had needed to distinguish the central N particle from the other two, for configuration analysis purposes. It can be in fact considered a quite good approximation of the center of mass of a triglycerides molecule, which have three chains of same type, when it assume a symmetric spatial configuration.

In figure 4.6 are also reported (red coloured) the radial distribution functions of the particles CC (figure 4.6a), NN (figure 4.6b) and CN (figure 4.6c), related to the determined Lennard-Jones parameters. In each figure



(a) CC radial distribution function

(b) NN radial distribution function

(c) CN radial distribution function

Figure 4.6: Radial distribution functions of C and N particles.

the radial distribution functions are compares to the distribution function of the centers of mass of the coarse-grained particles in the atomistic model (black coloured), showing very good agreement.

It is useful to notice that the radial distribution functions relative to coarse-grained particles show a steeper growth than those of the atomistic particles. This is due to the fact that as molecules description have become more *coarse-grained* more soft non-bonded potentials could be used (see for example page 40). However, we choose to maintain the Lennard-Jones $12-6$ potentials, since they are the most widely used. This make our developed model less specific and therefore more flexible for future applications and improvements.

In table 4.2 the Lennard-Jones cross parameters values are also reported. We have to notice that the values of $C-N$ (and therefore $C-$

a_i	a_j	$\sigma\ (nm)$	$\varepsilon(kJ/mol)$
N	$N2$	$4.980000 \cdot 10^{-1}$	2.59000
C	$N2$	$4.200000 \cdot 10^{-1}$	1.83000
C	N	$4.200000 \cdot 10^{-1}$	1.83000

Table 4.2: Table of calculated Lennard-Jones coarse-grained cross parameters.

$N2$ Lennard-Jones interactions differs slightly from those calculated using Lorentz–Berthelot mixing rules. They are in fact corrected to give rise to a better overlapping of radial distribution functions.

4.3.4 Force field bonded: parameters determination

The bonded parameters for two-bodies interactions (equation 2.5) and for three-bodies interactions (equation 2.10) are determined by trial and error (with the help of statical tools) in an independent way, at least in the first rough calculation, according to the following goal: The parameters are determined in such a way that lengths and angles probability distribution functions overlap, respectively, those obtained from the atomistic simulations considering the centers of mass of the groups from which the coarse-grained particles were derived. The parameters representing the equilibrium values of lengths l_0 and angles θ_0 influence the first order momentum (i.e. *mean* in case of Gaussian distributions) of the distribution, while the elastic constants k and k'_θ influence the second order momentum (i.e. *variance* in case of Gaussian distributions). The determined values of the two-bodies and of the three-bodies interactions parameters are shown in table 4.3 and in table 4.4 respectively. Equilibrium angles values reflect the chemical structure of the triglycerides molecules. $N-N-N$ interaction, in fact, has a lower equilibrium value and a higher elastic constant than the other. This is probably due to the fact that N particles are related to the glycerol group particles of the atomistic model, on which an improper dihedral potential was applied. The improper dihedral thus reduces coarse-grained particle mobility.

a_i	a_j	$l_0\ (nm)$	$k\ (kJmol^{-1}nm^{-2})$
N	N	0.269	5000.0
C	N	0.337	2800.0
C	C	0.354	3000.0

Table 4.3: Table of calculated two-bodies bonded potential parameters.

a_i	a_j	a_k	θ_0 (degree)	k'_θ $(kJmol^{-1})$
N	N	N	114.5	120.0
C	N	N	127.0	110.0
C	C	N	142.7	19.0
C	C	C	143.0	34.0

Table 4.4: Table of calculated three-bodies bonded potential parameters.

In figure 4.7 the probability distribution functions of the bonds of the coarse-grained model (red coloured) and of the atomistic one (black coloured)[4] are compared. The probability distributions from the atomistic model reported in figures 4.7a and 4.7c show bimodal trends. This is due to the fact that, in the coarse-grained model, the distance between two consecutive $C-C$ or $C-N$ correspond to several harmonic bonds (proportional in number to the distance between the centers of mass of coarse-grained particles) in the atomistic one. For the opposite reason, this shortcoming is not present in the distribution of $N-N$. Moreover, in figure 4.8 the probability distribution functions of the angles are reported. Also in this case bimodal trends are present (see figures 4.8a and figure 4.8c). The bimodal trends cannot be reproduce by coarse-grained model as the both bonded interactions functional forms. A possible way to improve the overlapping between the distributions (creating bimodal coarse-grained distribution) is to use a more complex two-bodies and three-bodies bonded interaction function. For reasons of computational efficiency the choice of a most simple functional form prevails.
Even so, as can be clearly seen in figures 4.7 and 4.8, there is a quite good agreement among the distributions.

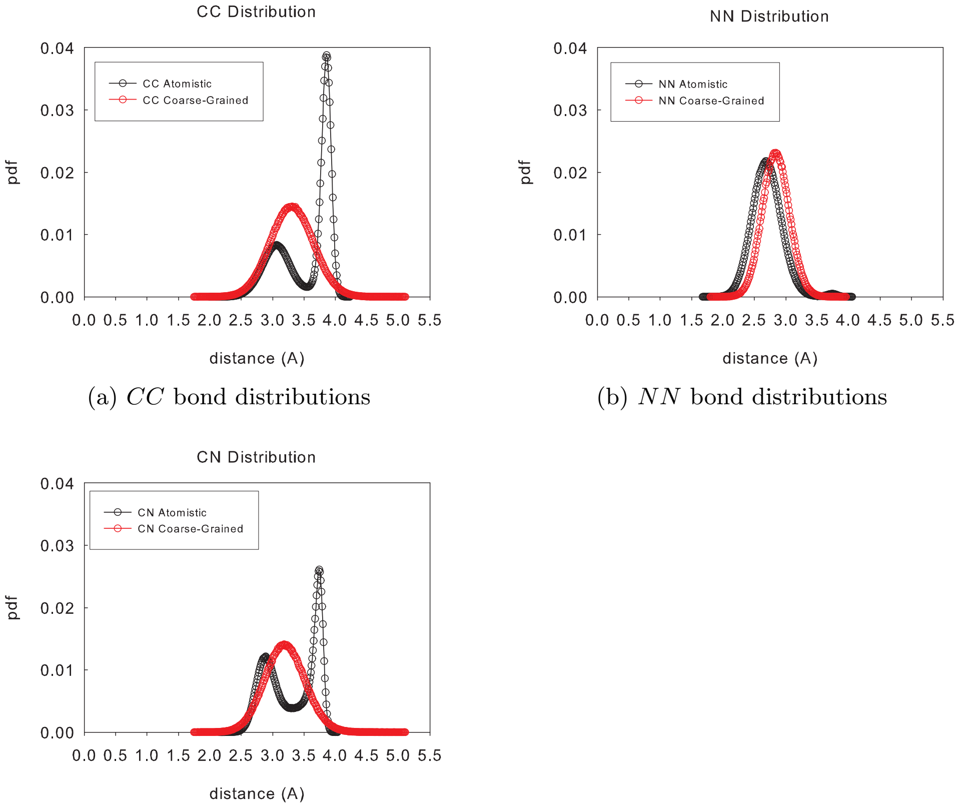
Bonded and non-bonded parameters are then refined iteratively. Therefore the main steps of the whole force field development can be summarized as follows:

- Fix the initial values of bonded and non-bonded interaction parameters.

- Determine non-bonded interaction parameters.

- Determine two-bodies and three-bodies bonded interaction parameters.

[4] in the sense specified on page 95



(a) CC bond distributions

(b) NN bond distributions

(c) CN bond distributions

Figure 4.7: Distribution functions of bond lengths.

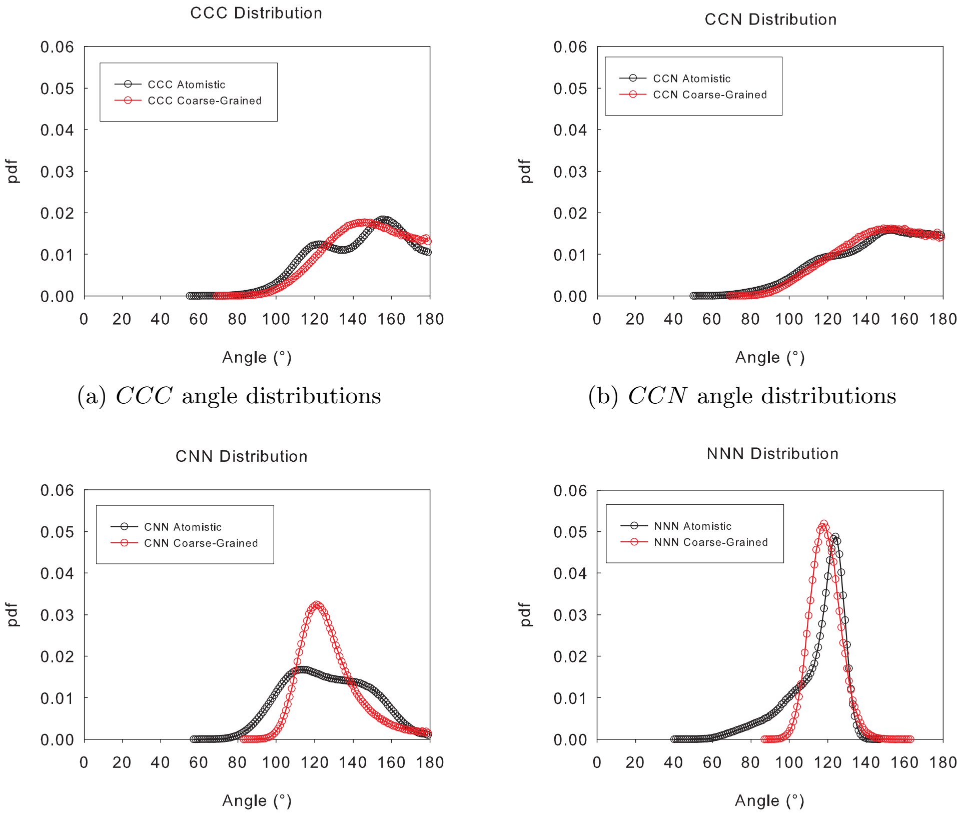
- Fix the determined bonded and non-bonded interactions as the new initial values.

This iterative procedure was repeated until convergence.

4.4 Molecular dynamics simulations of triglycerides

The procedure of parameters determination needs the execution of several molecular dynamics simulation (i.e several numerical integration of the equations of motion). The equation are integrated by means of GROMACS software package [11] using a leap-frog algorithm (see page 62). The time step of the leap-frog algorithm is fixed to $0.03ps$, an order of magnitude higher of that used in atomistic simulations.

The developed coarse-grained model is more *smooth* than the atomistic model. This means that each molecule requires less particles to be described and that less interactions have to be computed. A higher integration time-



(a) CCC angle distributions

(b) CCN angle distributions

(c) CNN angle distributions

(d) NNN angle distributions

Figure 4.8: Distribution functions of angles.

step can be then used. This cause a higher computational efficiency respect to the related atomistic model. We are able, in fact, to perform a simulation of 300ns of a box of tridecanoin (corresponding to that simulated in the chapter 3) in a day. This means that keeping an equal number of molecule in the box, our corse-grained model speed up 100 times the calculations respect to the atomistic one. Moreover, as we will see, in section 4.4.1 the increasing in computational efficiency is higher because of the higher molecules mobility.

Such efficiency will be used in the following sections both to speed up long time simulations and to simulate a larger molecules box.

4.4.1 Calculation of physical properties: comparison between atomistic model and experimental data

The calculation of physical properties is an important aspect of the force field development. Since we have deal with a coarse-grained model, i.e. with a less detailed model, we expect that will be a higher shifting between calculated and experimental properties respect to that found for the atomistic model in chapter 3. In the following paragraphs therefore we will quantify such shifting.

Density Density of the system depends on the whole the force-field and it have been optimized for a specific temperature $446K$, comparing the corse-grained density values with those obtained from atomistic model. At this temperature we have a coarse-grained model density of $856.0kg/m^3$, in perfect agreement between the respective atomistic model density of $858.6kg/m^3$.

In figure 4.9 the density values of the two models at various temperatures are compared. Moreover, in figure 4.9, experimental data are also reported in solid line. The coarse-grained model shows a steepest linear curve than that of the atomistic one which is in a good agreement with experimental data. Linear regressions in the two cases can be carried out. The results of the linear regression $y = y_0 + ax$ over the two sets of data are reported in table 4.5. The shifting between the two model can be quantified through the ratio of the direction coefficients that give a value of 2.20.

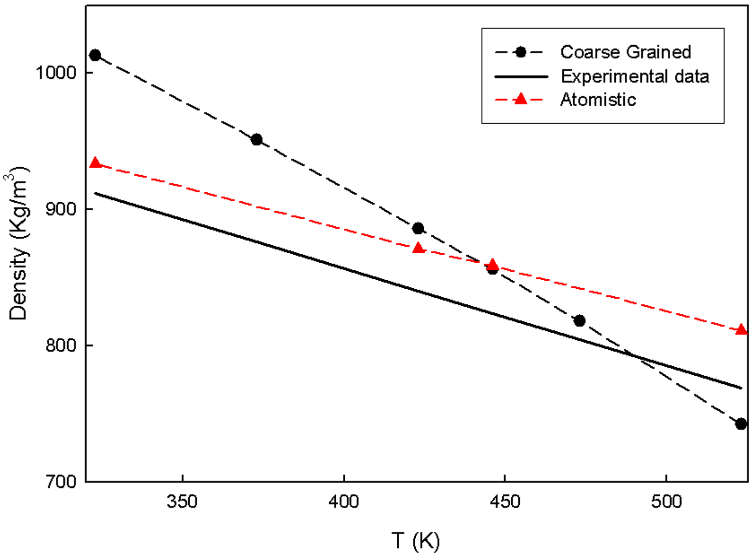


Figure 4.9: Comparison between density of the coarse-grained model and of the atomistic one varying temperature. Comparison with experimental data.

Mean squared displacement The mean squared displacement curve at a fixed temperature (i.e. $446K$) can be constructed as explained in section

Model	y_0 kg/m^3	a $kg/(m^3 \cdot K)$
Atomistic	1130.7630	-0.6116
Coarse-grained	1452.2677	-1.3462

Table 4.5: Parameters of the linear models describing the temperature dependence of density for the models.

2.7 on page 69.
In figure 4.10 such the mean squared displacement diagram versus simulation time is reported (black line) for a temperature of $446K$, together with the diagram relative to the atomistic model (red line). Although, they are similar in shape, at a fixed time, the coarse grained model shows a higher displacement than the atomistic one. This was to be expected[5], since the coarse-grained model has less interaction sites per molecule than the atomistic one. A part from a reduced computational efforts, the number of interaction sites influence positively also the tendency of each molecule to shift through the bulk constituted by the other molecules. For this purpose, the self-diffusivity of the system is computed by means of equation 2.71, giving the value $2.67 \cdot 10^{-5}$ cm^2/s. In the same conditions, for the atomistic model, we have obtained the lower value of $0.89 \cdot 10^{-5}$ cm^2/s. The ratio between the two value i equal to 3 and give us the *scaling factor* between the two models. In other words, the *real* simulation time for the developed coarse-grained model is obtained multiplying the simulation time by the scaling factor. This means that, for example, a simulation time for the coarse-grained system of 300 ns[6] corresponds to a simulation time of $3 \cdot 300 = 900$ ns the corresponding atomistic model.

The results reported until now show that, whereas on the one hand the developed coarse-grained model is able to simulate temporal scales close to typical nucleation times on the other hand it shows a properties shifting (e.g. density shifting or self-diffusivity shifting) which could increase the calculation time necessary to the appearance of such behaviors related to molecules rearrangements (such as for example the appearance of nuclei of solid phases).
Even so, the following sections will shown that the coarse-grained model is able to show the birth of nuclei constituted by planar molecules in suitable conformation, caused by temperature annealing. It will shown that simple shear flow conditions will increase molecules disposition to stay in a planar conformation. Moreover, the fact that the possible planar conformations are

[5] This result is also well known in literature; see for example [37]

[6] Such simulation time is performed in 1 day in the case of the coarse-grained simulation and in 100 days in the case of the atomistic simulation; see on page 80.

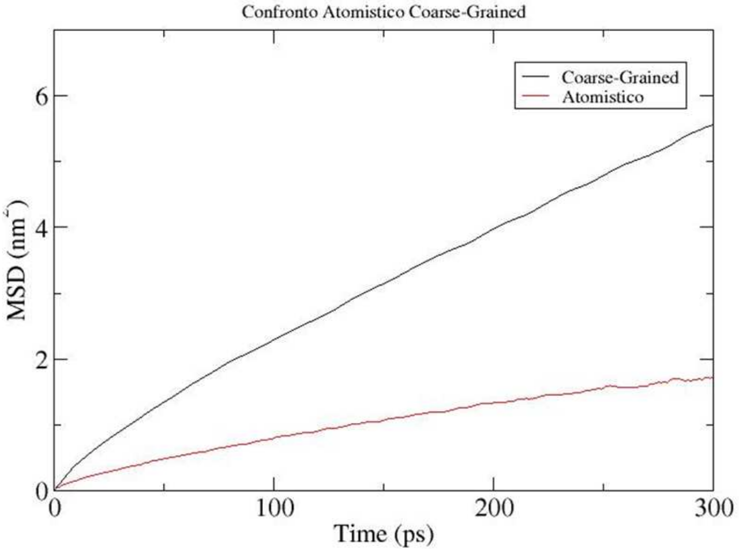


Figure 4.10: Mean squared displacement of a coarse-grained simulation at $446K$. Comparison between coarse-grained and atomistic models.

not equivalent will be shown.

4.4.2 Conformation analysis: rules development

This section deal with the developing of *order* parameters to the description of the conformational order of triglycerides molecules. Since triglyceride molecule has a characteristic structure, we have to develop suitable order parameters which take into account such structure.

Planarity

Crystal structures formed by triglycerides, apart from the specific polymorph, have a common feature: In crystal packing, triglyceride molecules adopt a planar conformation (e.g. [59]). Then the first order parameter concerns the identification of planar structures. In figure 4.11 is shown a simple scheme of a triglyceride molecule. The *planarity* is defined considering the triangles formed by two N particles of the glycerol backbone and the end C particle of each chain, as can be seen in figure 4.11. Looking at the figure, we define to angle: that formed by the orange planes is called θ_{12} and that formed by the green planes is called θ_{23}. The *planarity* parameter $\mathscr{P}$ is then define according to the following equation:

$$\mathscr{P} = \frac{\cos\theta'_{12} + \cos\theta'_{23}}{2} \tag{4.1}$$

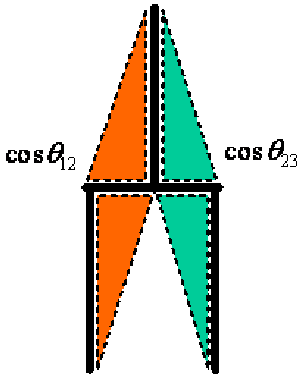


Figure 4.11: Simple scheme of a triglyceride molecule showing the plane used to define the orders parameters.

in which:

$$\theta'_{ij} = \begin{cases} \theta_{ij} & \text{se } \theta_{ij} \leq \frac{\pi}{2} \\ \theta_{ij} - \pi & \text{se } \theta_{ij} > \frac{\pi}{2} \end{cases}$$

with ij equal to 1 2 or 2 3 respectively.
The planarity $\mathscr{P}$ is a parameter belonging to $[0, 1]$ interval. If $\mathscr{P}$ assumes a value close to 1, that means that the triglyceride molecule has a planar conformation. On the contrary a $\mathscr{P}$ value close to zero means non planar conformation. Obviously, this rule is strictly valid only in the case of straight rigid branches[7]. Even so, this cannot be considered a serious problem because of the characteristic structure of the molecule. Triglycerides in fact have typically chain 20 atoms long. This means about 6 or 7 coarse-grained C particles, too short to have folded conformations. Moreover, even if the folded conformation appends this case will be not relevant as we will use the parameters in a statistical sense.

In practical, as we deal with numerical simulations, we will not found molecules with $\mathscr{P} = 1$ but molecules close to 1 with a certain tolerance i.e. $\mathscr{P} \in [1 - tol, 1]$. The same thing for $\mathscr{P} = 0$. The quantity tol will be suitable chosen.

Molecules conformations

We have to define also a second order parameter through which a planar molecule can be classified according to the relative position of the branches. We know in fact (see section1.3 on page 15) that triglycerides assume, in

[7] The parameter $\mathscr{P}$ define the planarity of the end particles of the three branches, that is possible even if the molecule is not globally planar

crystal packing, two kind of conformations defined, qualitatively, as *tuning-fork* and *chair*. Theoretically, they can assume also a third conformation with all branches at the same side of the glycerol backbone (we will call them *comb* conformations in the following). However, this conformation do not lead to a crystal packing.
The *conformation* parameter $\mathscr{C}$ is defined in not different way respect to the planarity, considering the same angles θ_{12} and θ_{23} in figure 4.11. For $\mathscr{P} = 1$, we define:

$$\mathscr{C} \quad = \quad \frac{\cos\theta_{12} + \cos\theta_{23}}{2} \ \textbf{se} \ \mathscr{P} \simeq 1 \tag{4.2}$$

The condition $\mathscr{P} \simeq 1$ have to be satisfied because otherwise $\mathscr{C}$ lose its meaning.
The *conformation* $\mathscr{C}$ belonging to $[-1, 1]$ interval. Tuning-fork planar molecules has a conformation parameter $\mathscr{C} = -1$, chair planar molecules has $\mathscr{C} = 0$, while comb planar molecules has $\mathscr{C} = 1$. Also in this case, since don't exist perfectly planar molecules, the preceding conditions will be modified in $\mathscr{C} \in [1 - tol, 1]$, $\mathscr{C} \in [-tol, tol]$ and $\mathscr{C} \in [-1, -1 + tol]$, respectively.

The order parameters defined in this section will be calculated with a FORTRAN software of our own and reported in appendix C. The defined parameters will be used in the following section to analyse the behavior of the system when a temperature annealing and/or flow conditions are imposed. We will demonstrate that simple shear flow induce an increase in the number of planar molecules.

4.4.3 Orientational analysis

Since we want to analysis also the orientation of planar molecules respect to the Cartesian coordinate system, we need to define an order vector. Among

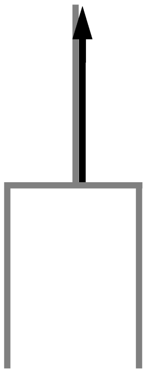


Figure 4.12: Simple scheme of the chosen orientational vector.

several possibility, the simplest seems to be that shown in figure 4.12. Exploiting the fact that we are interested to molecules bound to lie on a plane we define the orientational vector as the distance vector between the last C particle of the central arm and the central N particle. Such vector have the advantage of being quite independent from intra-molecular conformations. The space orientational distribution of system's molecules can be obtained through the calculation of the spherical coordinates related to this vector. In appendix C.6 the FORTRAN program developed for this purpose is reported.

4.5 Temperature annealing

A series of simulations were carried out to investigate the behavior of the developed model during temperature annealing. Temperature annealing is performed by integration software package GROMACS through Berendsen method described in chapter 2 on page 65.

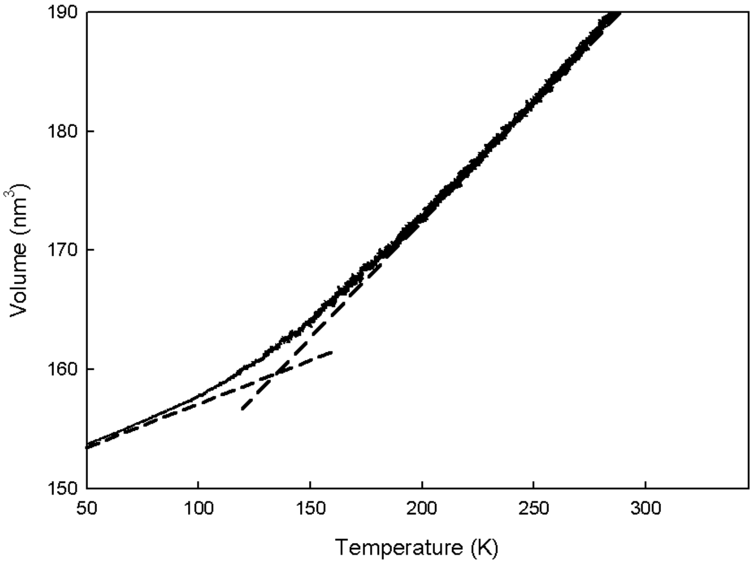


Figure 4.13: Temperature annealing of 216 tridecanoin molecules from $446K$ to $50K$ in static conditions.

In figure 4.13 a diagram relative to a temperature annealing from $446K$ to $50K$ with an annealing rate of $0.44 degree/ns$ was shown. The diagram show a glass transition at a temperature of $130K$. The fact is confirmed by the mean squared displacement diagram shown in figure 4.14. The figure

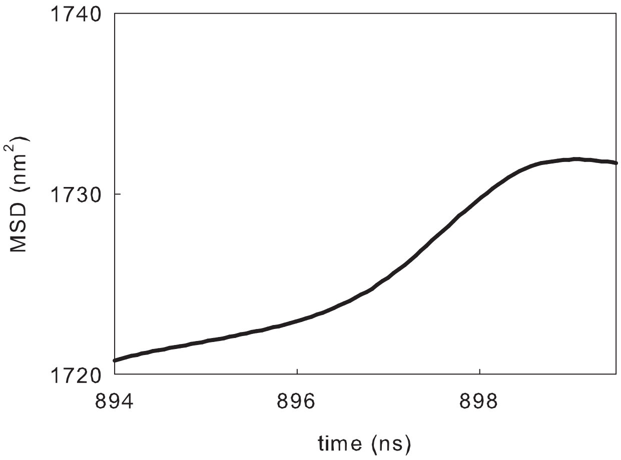


Figure 4.14: Mean Squared Displacement of the simulation of 216 tridecanoin molecules from $446K$ to $50K$ in static conditions.

Temperature (K)	Combs (%)	Chairs (%)	Forks (%)	Non-planar (%)
446	1.38	6.94	3.25	88.43
50	4.16	6.95	3.71	85.18

Table 4.6: Table percentages of planar and non-planar molecules.

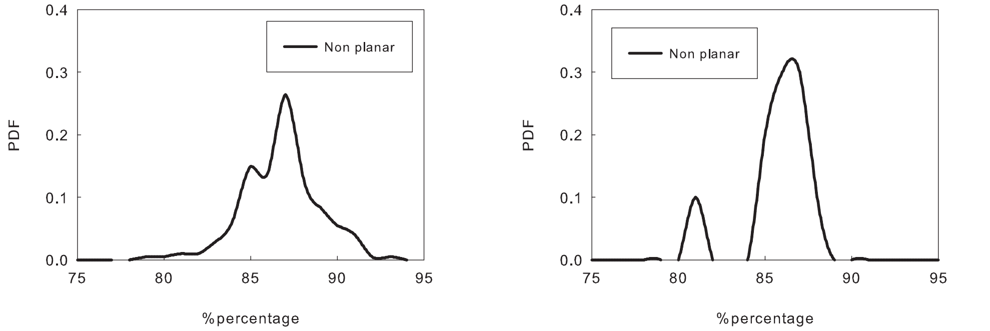
refers to the last part of the numerical simulation in which the presence of an horizontal plateau demonstrates the liquid-solid transition.
The transition temperature value is a quite acceptable quantitative value if compared with that of $175K$ available in literature [69] at a very lower annealing rate of $20degree/min$. Similar values of annealing rates cannot be obtained by us with the present computational resources. Even so, it is important to point out that the developed coarse-grained is able to describe qualitatively the liquid-glass transition of a bulk volume of triglycerides.

The conformational analysis of the initial and final conditions are summarized in table 4.6. As can be seen, there is no difference between the two temperature. This is due to the high annealing rate that have been *frozen* the molecules of the system. However, the percentage distribution of non-planar molecules at $50K$ shows a lower spread shape (see figure 4.15b) comparing with the same distribution at $446K$ (see figure 4.15a). Moreover, a shifting of the curves towards lower values of percentage of non-planar molecules is

Temperature (K)	Combs (%)	Chairs (%)	Forks (%)	Non-planar (%)
446	1.38	6.94	3.25	88.43
115	4.16	8.80	5.10	81.94

Table 4.7: Table percentages of planar and non-planar molecules.

also observed.



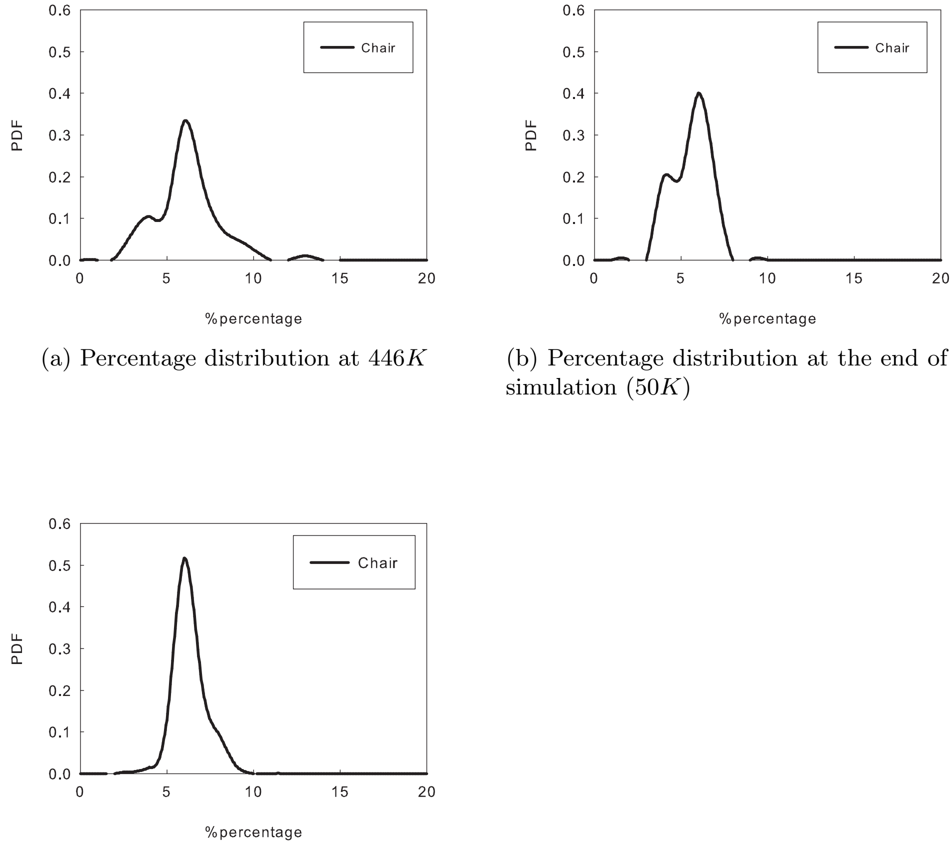
(a) Percentage distribution at $446K$

(b) Percentage distribution at the end of the simulation ($50K$)

Figure 4.15: Percentage distributions of non-planar molecules.

A quite intriguing observation is that chairs molecules number is approximately double than that of forks molecules. This is probably due to the fact that it is possible to form chair conformations in a double way respect to comb or tuning-fork cases. Seeing figure 4.11, we notice, in fact, that chairs can be from when $\theta_{12} = 0$ and $\theta_{23} = 1$ or viceversa, while comb can be formed only when $\theta_{12} = 0$ and $\theta_{23} = 0$ and forks only when $\theta_{12} = 1$ and $\theta_{23} = 1$.

An slight increase in the number of planar molecules is obtained in next numerical experiment. In this experiment a well equilibrated box of molecules at temperature of $446K$ is frozen very rapidly i.e. in $1ps$ at temperature of $115K$ which is a temperature probably close to temperature of glass transition. The system is taken at this temperature for $6ns$. The results are shown in table 4.7. A slight overall increase of planar molecules can be observed. The analysis of percentage distributions of planar and non-planar molecule allows to ascribe the cause to the fact that the distribution functions are less spread than the corresponding distribution in the preceding cases (i.e. at $446K$ and $50k$). As an example the distribution related to planar chair conformations are reported in figure 4.16.



(a) Percentage distribution at $446K$

(b) Percentage distribution at the end of the simulation ($50K$)

(c) Percentage distribution at the end of the simulation ($50K$)

Figure 4.16: Percentage distributions at $115K$.

It is interesting to observe the growth of the number of tuning-fork conformations respect to chairs conformations. This can be considered an circumstantial evidence of a system tendency to go towards more ordered structures performing simulations on a longer time scale.
In figure 4.17 a tri-dimensional representation of the chair planar molecules at the end of the simulation of table 4.7 at $115K$ is reported. As can be seen, the planar molecules are quite dispersed, with some little clusters made of two molecules. Moreover, molecules don't have a preferential orientation. Moreover, a further temperature annealing up to $105K$ has been performed without any substantial changes in conformations.

To point out the system tendency to form clusters we need to perform

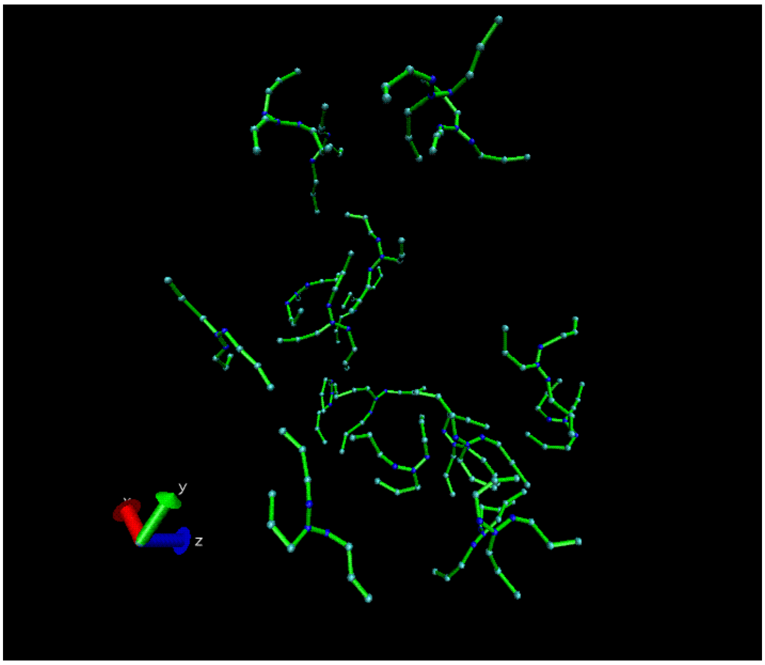
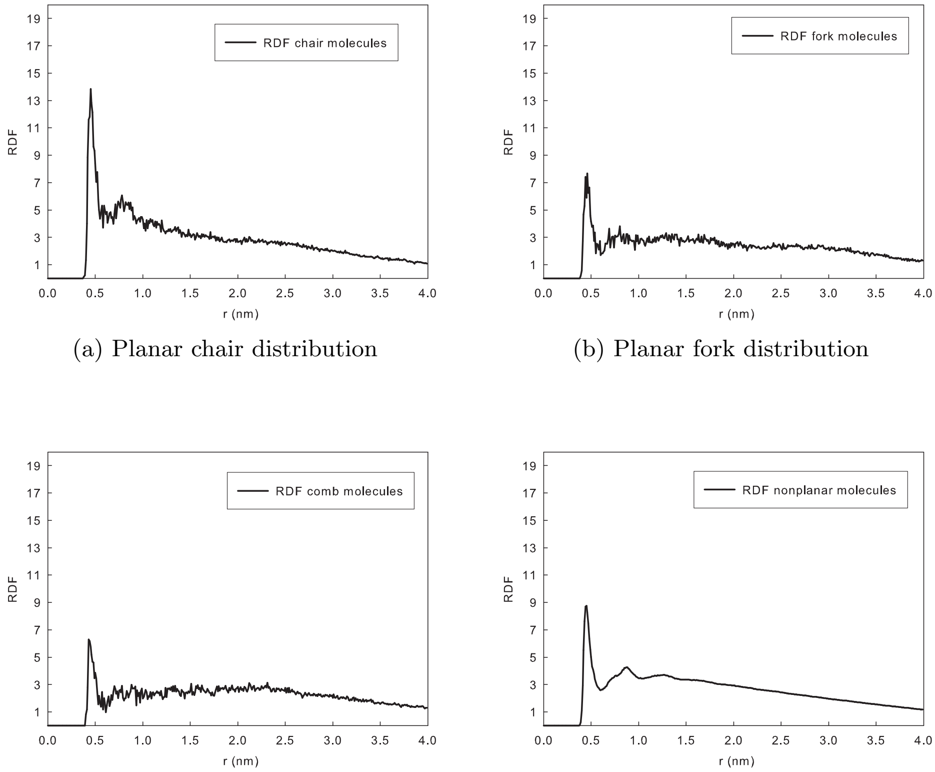


Figure 4.17: Chair planar molecules of the simulation box at $115K$.

simulations during which system is subject to cyclic stages of heating and cooling. So that molecules with less stable conformations have the required energy to rearrange themselves in more stable ones. Obviously the cooling processes should be as slow as the molecules should be able to change their conformation and to place themselves closest to each other. Unfortunately this would mean a very long simulation time, unreachable with our computational resources. Even so, the following numerical experiments will show clearly an increase of molecules arrangements.

The first numerical experiment consists of a series of four heating-cooling processes. The cycle process is the following: Starting from a temperature of $115K$ the system is heated until a temperature of $150K$ is reached and after is cooled to the final temperature of $115K$. The heating process last $25ns$ while the cooling process last $125ns$. At the end of each cycles a series of analysis is conducted on system trajectories.

In figure 4.18 the radial distribution functions of tridecanoin molecules after an isothermal simulation at $115k$ is reported. This can be considered for our purposes the initial state with which compares the distribution derived after the cyclic processes. Each radial distribution function shows a peak at low distance corresponding to the distance at which statistically we find a higher number of molecules with the same conformation. The presence of peak is due to the fact that the system in such temperature value is nearly in



(a) Planar chair distribution

(b) Planar fork distribution

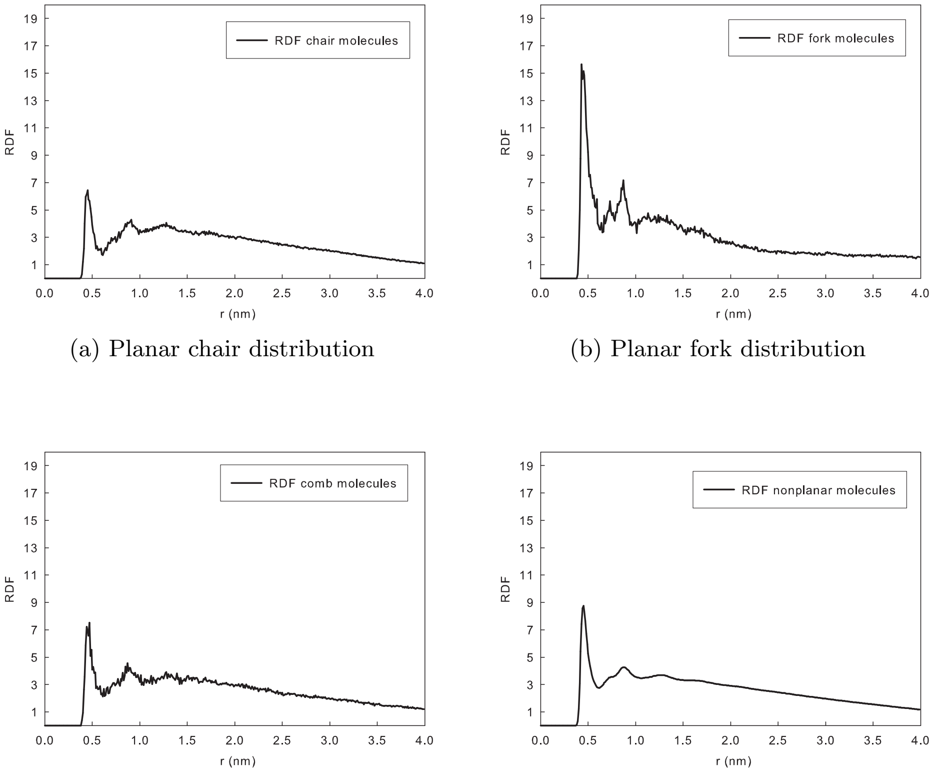
(c) Planar comb distribution

(d) Non-Planar molecules distribution

Figure 4.18: Radial distribution functions of tridecanoin molecules at $115K$.

a glassy solid state. However, a higher peak can be observed for planar chair molecules respect to the other planar molecules. Mixed radial distribution functions are also built, showing a very slight interrelation between fork and comb molecules.

In figure 4.19 the radial distribution functions at the end of four cyclic processes are shown. The reduction of the chair peak is balanced by an increase in the level of the peak related to the fork conformation. Moreover, it can be observed the appearance of small peaks corresponding to greater values of distance in all diagrams related to planar conformations. In the case of fork conformation the phenomenon is characterized by a higher maximum. This means that, through such cyclic processes, the system is tending to assume a more ordered planar fork conformation. As the process is very time consuming, it is not possible to observe the formation of a full developed crystal phase. Even so, these numerical experiments show that the developed model could be fruitfully used to study phase transition of triglycerides.



(a) Planar chair distribution

(b) Planar fork distribution

(c) Planar comb distribution

(d) NonPlanar molecules distribution

Figure 4.19: Radial distribution functions of tridecanoin molecules at $115K$ after four cycles.

4.6 Temperature annealing with velocity profile

In this section the role of shear flow will be investigated and the birth of more ordered cluster structures will be highlighted through radial distribution functions. For this purpose the simple acceleration profile defined by the equation 2.76 on page 73 was adopted. A typical velocity profile obtained with such imposition is shown in figure 4.20 from which an order of magnitude of $10^9 s^{-1}$ shear rate is obtained. This is a usual value (e.g. see [65]) that, though far from values used in all practical applications, allows to examine phenomena otherwise not observable.

As the imposed accelerations cause high deformations of the system volume a larger box have to be used. This lead to a heavy loss of computational efficiency that don't allow to perform simulations for a long time. Even so, as we will see briefly, the beginning effect of temperature annealing and shear flow on the molecules conformations can be appreciate.

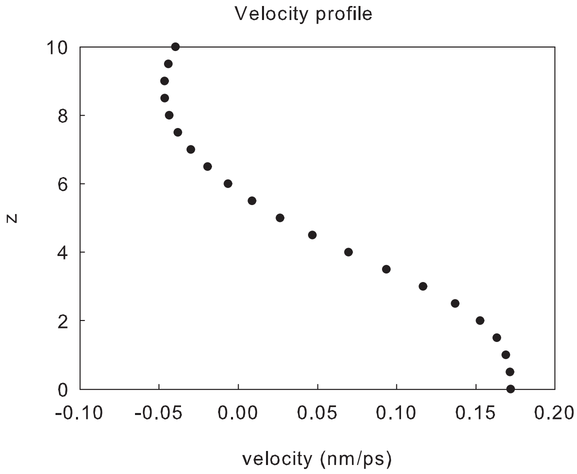


Figure 4.20: Typical velocity profile induced by the imposed acceleration profile.

Temperature (K)	Combs (%)	Chairs (%)	Forks (%)	Non-planar (%)
115	2.83	7.07	3.59	86.51
105	5.67	9.02	7.18	78.13

Table 4.8: Table percentages of planar and non-planar molecules.

The simulation box of the preceding section was duplicated in all the three directions giving an eight times larger system than the preceding cases. Such system, constituted by 1728 molecules of tridecanoin molecules, is equilibrated at a temperature of $446K$.

In the first numerical experiment the temperature is reduced from $115K$ to $105K$ in a time of $3ns$. A maximum acceleration value [8] of 0.04 $nm \cdot ps^{-2}$ was adopted. This is a typical value as can be seen in reference [70]. In figure 4.21 is shown the temporal evolution of the molecules conformations of the box. The diagram exhibit an instantaneous increase in the number of planar molecules. The comparison with the simulations presented in section 4.5 is enough to establish a clear connection between shear and increasing planarity. As a further proof in table 4.8 the percentages of planar and non-planar molecules at the initial and final temperatures are reported. A slightly increase in the number of planar molecules can be observed. We have to notice that the ratio between chair shape molecules and fork shape molecule at $105K$ is 1.26, quite different from 2. This means that in this

[8] See equation 2.76

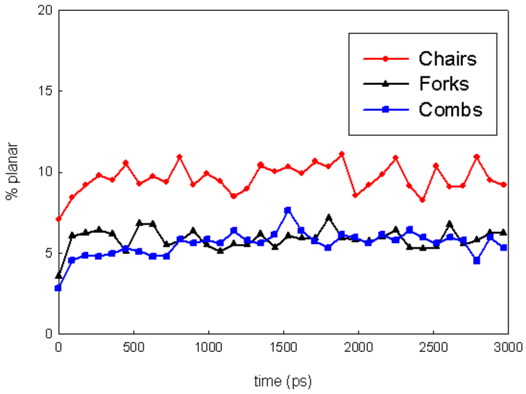


Figure 4.21: Molecular conformations percentage during a temperature annealing of 1728 tridecanoin molecules from $115K$ to $105K$ in dynamic conditions.

operation condition the fork shape molecules are slightly preferred to the chair shape molecules.

In figures 4.22 and 4.23 are shown the planar fork and the planar chair molecules extracted from the last trajectory (i.e. that at $105K$) of the simulation. It is evident that shear flow, as it raise molecules mobility, helps the system in inducing the orientation of molecules and increasing the number of planar molecules. Moreover, cluster formation phenomena can be appreciate.

The size of cluster formation phenomena can be quantitatively measured by means of radial distribution functions according to each planar shape. To this aim a suitable FORTRAN software was constructed and is reported in appendix C.

The diagrams in figure 4.24 show the highest peaks at low distance values that correspond to the distribution of the coarse-grained particles belonging to a same molecule. Moreover, straight after these peaks, other high peaks are present. These peaks correspond to groups of molecules (which are at short distance from each other) which therefore manifest quite ordered structures (see also the figures 4.22and 4.23). At a distance more than 2 nm the curves reach a plateau. That means that at such distance molecules are uncorrelated.

The effect of cyclic operations can also be highlighted. For this purpose, a series of four heating-cooling cycles have been carried out. The features of the simulations are those adopted in the static case on page 4.18. Moreover,

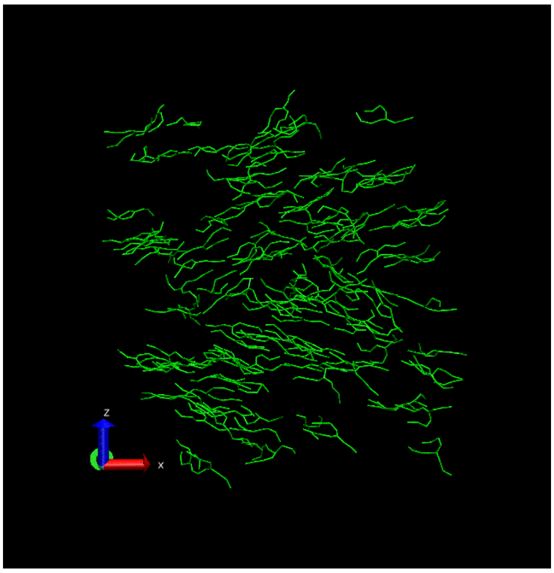


Figure 4.22: Fork planar molecules induced by shear flow.

the simple acceleration profile defined by the equation 2.76 on page 73 was adopted with a maximum acceleration value of 0.01 $nm \cdot ps^{-2}$.
Figures 4.25 shows higher peak level respect to the those related to initial simulation conditions (see figure 4.18). This confirms the trend of the system to assume more ordered configurations.

The effect of dynamical conditions can be pointed out comparing radial distribution functions in static and dynamical conditions after four heating-cooling cycles on figures 4.19 and figures 4.25, respectively. Radial distribution functions relative to dynamical conditions are quite similar in shape to each other. This means that, in dynamical case, cyclic operations influence the growth of each diagram in the same way, unlike the static case in which cyclic operations influence greatly the growth of the planar fork radial distribution function. This confirms the role of flow in raising molecules mobility.

4.7 Conclusion remarks

The objective of this thesis have been the development and the analysis of microscopic mathematical models to investigate the dependence of triglycerides conformations from environmental conditions.
Triglycerides are important constituents of food products. In literature is highlighted that they can solidify in several crystal packing (such phenomenon is called polymorphism). Moreover, it's widely held that the poly-

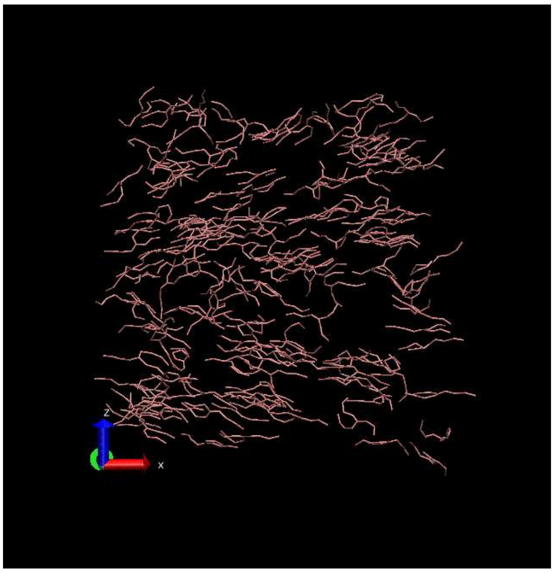
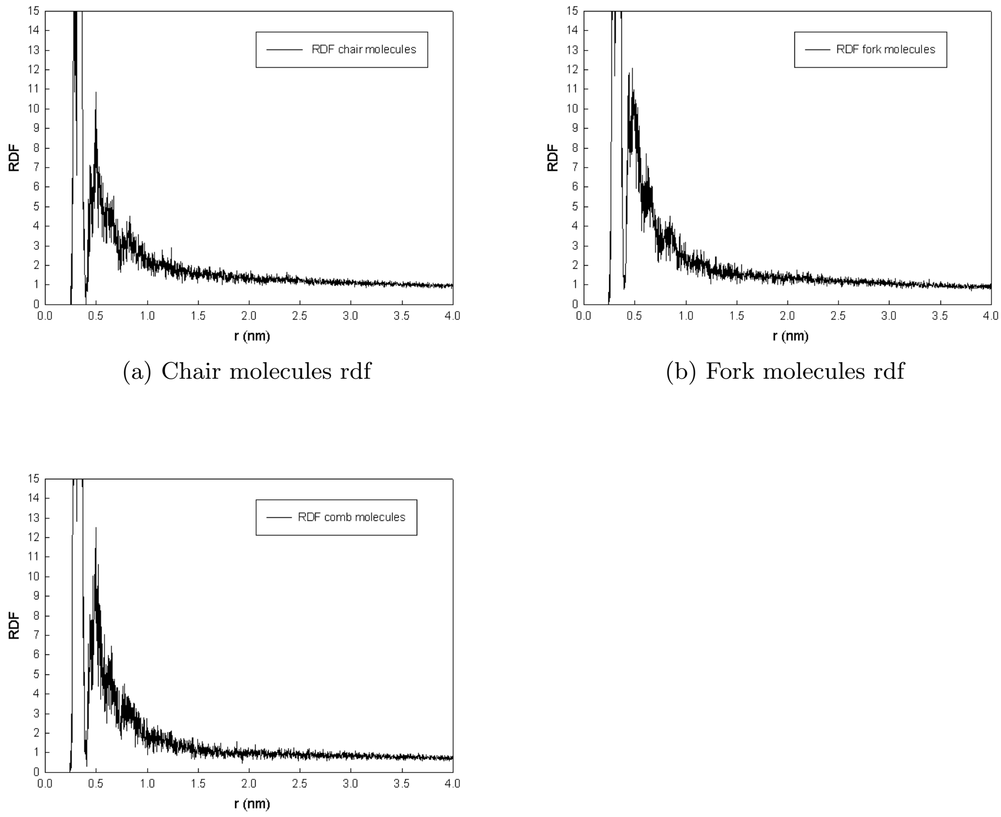


Figure 4.23: Chair planar molecules induced by shear flow.

morphic behavior of triglycerides influences greatly processes management involving fats mixtures. As an example chocolate production process have been discussed. However, the reported remarks about this topic have a wider significance, since they can be related to the most general topic of fats crystallization.

As we were interested to investigate the relationship between macroscopic conditions and microscopic induced properties, we steered our investigations towards microscopic mathematical models. Even though the triglycerides have been widely investigated from experimental point of view, the literature on the mathematical modelling of processes in which triglycerides are involved are strangely meagre. Moreover, regarding microscopical model of triglycerides there is one paper but deal with the simulation of a so called α solid phase [39]. Therefore, for our purposes, the microscopic mathematical model had to be developed from the beginning.

Starting from the atomistic force field of the one paper available, we built a united-atoms model. Comparison between numerically and experimentally derived physical properties together with distributions analyses showed that such model was able to describe with adequate accuracy the behavior of a liquid phase. Hence, such atomistic model was taken as a point of reference to the development of a new coarse-grained model. The force field of such model was developed, by means of statistical tools, using suitable distribution functions derived from the atomistic one. The aim was to obtain a coarse-grained model able to reproduce molecules fluctuations of the atomistic one.



(a) Chair molecules rdf

(b) Fork molecules rdf

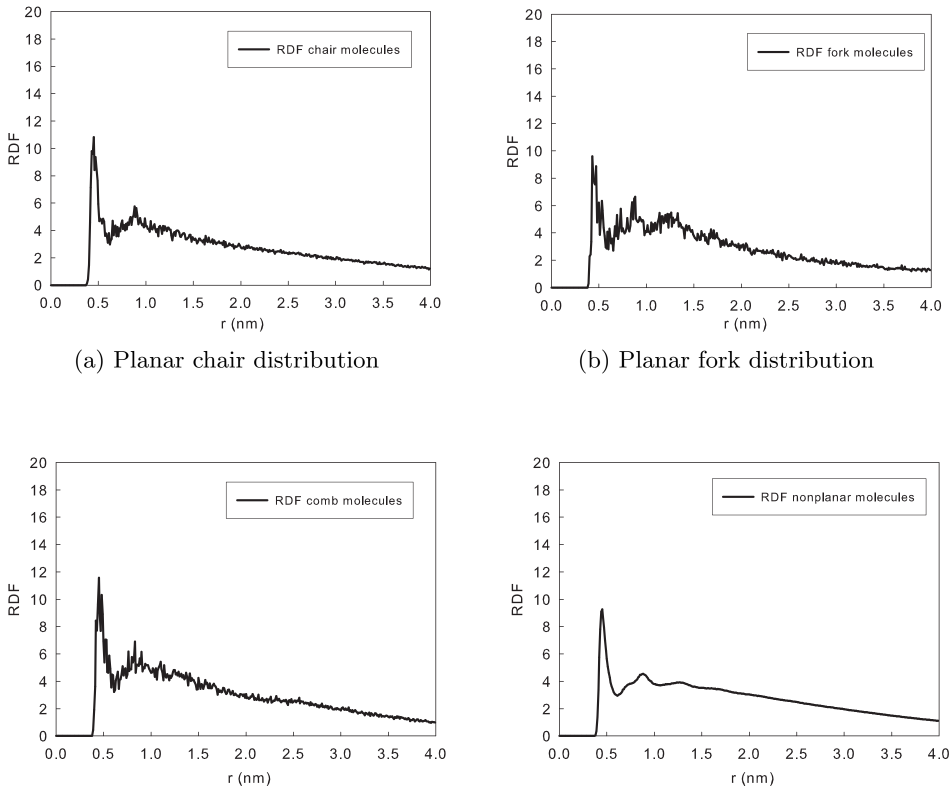
(c) Comb molecules

Figure 4.24: Radial distribution functions of tricanoin box at $115K$: Analysis of shear induced conformations.

The coarse-grained model was used to perform numerical experiments to highlight the dependence of molecules conformations from experimental conditions. For this purpose suitable intra-molecular order parameters were invented.

The results of our simulations show a clear relationship between conformational states of molecules and temperature annealing condition. In particular, an increase of the number of planar (chair, fork or comb) conformations during cooling processes was observed. The cooling process also influences greatly the shape of the distribution functions of the percentage of planar molecules.

Cyclic heating–cooling processes were also applied. Even though, for reasons related to the need of high computation resources, we could not perform so a high number of cyclic process that a completely ordering system box was obtained, the improvement of order features through radial distribution



(a) Planar chair distribution

(b) Planar fork distribution

(c) Planar comb distribution

(d) Non-Planar molecules distribution

Figure 4.25: Radial distribution functions of tridecanoin molecules at $115K$ after four cycles with shear imposed.

functions was pointed out. This means that the formation of small clusters of few planar molecules can be easily observed by visual inspection.

The effect of simple flow conditions imposed on the system was also investigated. A further increase of the number of planar conformation respect to the static cases was observed. To better assess the role of flow conditions in inducing molecules conformations, the radial distribution functions of planar and non–planar conformations during heating-cooling processes have been calculated. They show high similarity in shape. This could be a sign of a quite uniform molecules spreading due to higher mobility produced by the flow. This hypothesis was confirmed by the investigation of the inter-molecular order parameters distribution functions (i.e. the distribution functions of spherical coordinates related to an inter-molecular order vector) showing in dynamical cases more dispersed shapes than those observed in static cases.

Appendix A

Internal coordinates: The Z-matrix method

There are two main ways of specifying the molecular geometry(e.g. to the aim of construct a initial configuration of a system of molecules): cartesian coordinates and internal (or Z-matrix) coordinates. The former are more compact, but they cannot readily set up by hand. The latter is used to specify a molecule geometry in a text file in an easy way.
A Z-matrix describes a molecule geometry by means of bond distances, bond angles and dihedral angles. A general scheme of a Z-matrix text file is shown in figure A.1. Each line specifies the location of an atom in terms of its position with respect to its molecular neighbors. The overall orientation of the molecule obviously does not matter.

atom1						
atom2	1	r12				
atom3	2	r23	1	a123		
atom4	3	r34	2	a234	1	d1234
atom5	4	r45	3	a345	2	d2345
...						

Table A.1: A Z-matrix coordinates scheme.

The first atom (atom1 in figure A.1 constitutes the origin of the internal coordinate system and hence no parameters need to be specified. The position of the second item (atom2 in figure A.1) is completely defined providing the distance from the first atom. In particular, in the second row, the first column represents the name (atom2) of the atom, the second represents an index indicating the row (i.e. 1) where to find the atom from which the distance $r12$, written in the third column, is computed. The third row defines the position of the third atom (first column) through the distance $r23$ from the atom at row specified by the index in the second column (i.e.2) and through the angle $a123$. The angle is that formed with the atom specified by the index (i.e. 1) in the second column and in forth column (i.e. 2). The following atoms are defined by means of four parameters. In the case of atom4, $r34$ represents the distance from atom3, $a234$ represents the angle between atom2-atom3-atom4 and $d2345$ represent the dihedral angle between the plane formed by atom2-atom3-atom4 and atom3-atom4-atom5. The text file format can, obviously, vary because of the used package conventions. For example in the following the Z-matrix text file built for a tridecanoin molecule (united-atoms) is shown. As can be seen, the bond, angle and dihedral parameters are separated by zero symbols. Moreover, the indexes to which the parameters refer, are reported in the last three columns.

```
CH2  0.000000 0    0.000000 0    0.000000 0      0     0    0
CH1  0.153000 0    0.000000 0    0.000000 0      1     0    0
CH2  0.153000 0  110.000000 0    0.000000 0      2     1    0
 OA  0.114700 0  111.000000 0    0.000000 0      1     2    3
 OA  0.114700 0  110.000000 0  110.000000 0      2     1    3
 OA  0.114700 0  111.000000 0    0.000000 0      3     2    1
  C  0.113300 0  117.000000 0  180.000000 0      4     1    2
  C  0.113300 0  117.000000 0  180.000000 0      5     2    3
  C  0.113300 0  117.000000 0  180.000000 0      6     3    2
  O  0.123000 0  124.000000 0    0.000000 0      7     4    1
  O  0.123000 0  124.000000 0    0.000000 0      8     5    2
  O  0.123000 0  124.000000 0    0.000000 0      9     6    3
CH2  0.148000 0  115.000000 0  180.000000 0      7     4   10
CH2  0.148000 0  115.000000 0  180.000000 0      8     5   11
CH2  0.148000 0  115.000000 0  180.000000 0      9     6   12
CH2  0.153000 0  120.000000 0  180.000000 0     13     7    4
CH2  0.153000 0  120.000000 0  180.000000 0     14     8    5
CH2  0.153000 0  120.000000 0  180.000000 0     15     9    6
CH2  0.153000 0  120.000000 0  180.000000 0     16    13    7
CH2  0.153000 0  120.000000 0  180.000000 0     17    14    8
CH2  0.153000 0  120.000000 0  180.000000 0     18    15    9
CH2  0.153000 0  120.000000 0  180.000000 0     19    16   13
CH2  0.153000 0  120.000000 0  180.000000 0     20    17   14
CH2  0.153000 0  120.000000 0  180.000000 0     21    18   15
CH2  0.153000 0  120.000000 0  180.000000 0     22    19   16
CH2  0.153000 0  120.000000 0  180.000000 0     23    20   17
CH2  0.153000 0  120.000000 0  180.000000 0     24    21   18
CH2  0.153000 0  120.000000 0  180.000000 0     25    22   19
CH2  0.153000 0  120.000000 0  180.000000 0     26    23   20
CH2  0.153000 0  120.000000 0  180.000000 0     27    24   21
CH2  0.153000 0  120.000000 0  180.000000 0     28    25   22
CH2  0.153000 0  120.000000 0  180.000000 0     29    26   23
CH2  0.153000 0  120.000000 0  180.000000 0     30    27   24
CH2  0.153000 0  120.000000 0  180.000000 0     31    28   25
CH2  0.153000 0  120.000000 0  180.000000 0     32    29   26
CH2  0.153000 0  120.000000 0  180.000000 0     33    30   27
CH3  0.153000 0  111.000000 0  180.000000 0     34    31   28
CH3  0.153000 0  111.000000 0  180.000000 0     35    32   29
CH3  0.153000 0  111.000000 0  180.000000 0     36    33   30
```

Appendix B

Derivation of Newton's equation of motion from Lagrangian for Nosè–Hoover thermostat

The interactions between particles of a system can be equivalently described through the *Lagrangian* function of the system [71]. In the case of n interacting particles, the Lagrangian, in the most general form, is:

$$\mathscr{L} = \sum \frac{1}{2} m_a v_a^2 - U\left(r_1, r_2, \ldots\right) \tag{B.1}$$

in which the function U depends on the nature of interaction and r_a is the position vector of the ath particle. The Lagrangian is the sum of the total kinetic energy and of the potential energy of the system. Knowing the Lagrangian it is easy to derive the Newton's equations of motion:

$$\frac{d}{dt} \frac{\partial \mathscr{L}}{\partial v_a} = \frac{\partial \mathscr{L}}{\partial r_a} \tag{B.2}$$

Introducing a Nosè–Hoover thermostat the Lagrangian of the system becomes the equation 2.59 on page 67. Lets derive the Newton's equations of motion applying equation B.2 to that equation.
We have:

$$\frac{d}{dt}\left(m_a \dot{r}_a s^2\right) = -\frac{\partial \mathscr{V}}{\partial r_a} \tag{B.3}$$

$$\frac{d}{dt}\left(Q \dot{s}\right) = \sum_a m_a \dot{r}_a^2 s - \frac{\left(f+1\right) k_b T}{s} \tag{B.4}$$

From which:

$$m_a \ddot{r}_a s^2 + 2 m_a s \dot{s} \dot{r}_a = -\frac{\partial \mathscr{V}}{\partial r_a} \tag{B.5}$$

$$Q\ddot{s} = \sum_a m_a \dot{r}_a^2 s - \frac{(f+1)\, k_b T}{s} \tag{B.6}$$

and rearranging we have equations 2.60 and 2.61.

Appendix C

Developed Fortran codes

C.1 Dihedral calculation code

```
Program diedri
implicit none
Integer :: i,j,k,j0
character (len=29) frase, frase2
Integer totatm
Doubleprecision temp
Integer totatoms, newatoms
Parameter (totatoms=2592)
Parameter (newatoms=216)
TYPE rigo
    integer primo
    character (len=5) secon
    character (len=5) terzo
    integer quarto
    doubleprecision quinto
    doubleprecision sesto
    doubleprecision settimo
!   doubleprecision ottavo
!   doubleprecision nono
!   doubleprecision decimo
ENDTYPE
type (rigo), dimension(totatoms) :: matr
Doubleprecision, dimension(3) :: dim
!
Doubleprecision :: dist1, dist2, dist3
Doubleprecision, dimension(3) :: point11,point12,point13,point14
Doubleprecision, dimension(3) :: point21,point22,point23,point24
Doubleprecision :: a11,b11,c11,d11,a12,b12,c12,d12
Doubleprecision :: a21,b21,c21,d21,a22,b22,c22,d22
Doubleprecision :: cos12,cos23
Doubleprecision :: angle12, angle23
Doubleprecision :: angle12mod, angle23mod, plan, meanplan, conf
Doubleprecision, dimension(216) :: planvett, confvett
Doubleprecision :: pigreco, a
Integer :: countPP, countSS, countFF, countBB
!
Doubleprecision :: tol
Integer :: varcontrol

Open ( UNIT=1, FILE= 'traj1.gro', STATUS='OLD')
Open ( UNIT=2, FILE= 'diedri32.txt', STATUS='UNKNOWN')
Open ( UNIT=3, FILE= 'meanplanar.txt', STATUS='UNKNOWN')
Open ( UNIT=4, FILE= 'trajplanar.gro', STATUS='UNKNOWN')
Open ( UNIT=5, FILE= 'dddboxplanar.gro', STATUS='UNKNOWN')
Open ( UNIT=6, FILE= 'count.txt', STATUS='UNKNOWN')

varcontrol=1
tol=0.1
    write(6,*) '       tempo    Comb Chair Fork Non-planar'

DO WHILE (.NOT. EOF(1))
!********************************
!   LETTURA DELLA CONFIGURAZIONE
```

```
!********************************
    read(1,101) frase, temp
    read(1,102) totatm
    i=1
    DO WHILE (i.LE.totatoms)
        read (1,100) matr(i)
        i=i+1
    END DO
    read(1,104) dim
!********************************************************
!   CALCOLO DEI PARAMETRI DI LUNGHEZZA DELLE CATENE
!********************************************************
    i=1
    j=i-1
    k=0
    DO WHILE (i.LE.2592)
    j=i-1
!       dist1=SQRT((matr(j+1)%quinto-matr(j+10)%quinto)**2+ &
!           (matr(j+1)%sesto-matr(j+10)%sesto)**2+ &
!           (matr(j+1)%settimo-matr(j+10)%settimo)**2)
!       dist2=SQRT((matr(j+2)%quinto-matr(j+11)%quinto)**2+ &
!           (matr(j+2)%sesto-matr(j+11)%sesto)**2+ &
!       dist3=SQRT((matr(j+3)%quinto-matr(j+12)%quinto)**2+ &
!           (matr(j+3)%sesto-matr(j+12)%sesto)**2+ &
!           (matr(j+3)%settimo-matr(j+12)%settimo)**2)

!****************************
!       CALCOLO DEI DIEDRI
!****************************
        point11=(/matr(j+6)%quinto,matr(j+6)%sesto,matr(j+6)%settimo/)
        point12=(/matr(j+9)%quinto,matr(j+9)%sesto,matr(j+9)%settimo/)
        point13=(/matr(j+3)%quinto,matr(j+3)%sesto,matr(j+3)%settimo/)
        point14=(/matr(j+12)%quinto,matr(j+12)%sesto,matr(j+12)%settimo/)

        point21=(/matr(j+5)%quinto,matr(j+5)%sesto,matr(j+5)%settimo/)
        point22=(/matr(j+8)%quinto,matr(j+8)%sesto,matr(j+8)%settimo/)
        point23=(/matr(j+2)%quinto,matr(j+2)%sesto,matr(j+2)%settimo/)
        point24=(/matr(j+11)%quinto,matr(j+11)%sesto,matr(j+11)%settimo/)
!***********************************************
!       Coefficienti delle equazioni dei piani
!***********************************************
        a11=(point13(2)-point12(2))*point11(3)+(point11(2)-point13(2))*point12(3)+(point12(2)-point11(2))*point13(3)
        b11=(point12(1)-point13(1))*point11(3)+(point13(1)-point11(1))*point12(3)+(point11(1)-point12(1))*point13(3)
        c11=(point13(1)-point12(1))*point11(2)+(point11(1)-point13(1))*point12(2)+(point12(1)-point11(1))*point13(2)
!       d11=(point13(1)*point12(2)-point12(1)*point13(2))*point11(3)+(point11(1)*point13(2)- &
!        point13(1)*point11(2))*point12(3)+(point12(1)*point11(2)-point11(1)*point12(2))*point13(3)

        a12=(point14(2)-point12(2))*point11(3)+(point11(2)-point14(2))*point12(3)+(point12(2)-point11(2))*point14(3)
        b12=(point12(1)-point14(1))*point11(3)+(point14(1)-point11(1))*point12(3)+(point11(1)-point12(1))*point14(3)
        c12=(point14(1)-point12(1))*point11(2)+(point11(1)-point14(1))*point12(2)+(point12(1)-point11(1))*point14(2)
!       d12=(point14(1)*point12(2)-point12(1)*point14(2))*point11(3)+(point11(1)*point14(2)- &
!        point14(1)*point11(2))*point12(3)+(point12(1)*point11(2)-point11(1)*point12(2))*point14(3)

        a21=(point23(2)-point22(2))*point21(3)+(point21(2)-point23(2))*point22(3)+(point22(2)-point21(2))*point23(3)
        b21=(point22(1)-point23(1))*point21(3)+(point23(1)-point21(1))*point22(3)+(point21(1)-point22(1))*point23(3)
        c21=(point23(1)-point22(1))*point21(2)+(point21(1)-point23(1))*point22(2)+(point22(1)-point21(1))*point23(2)
!       d21=(point23(1)*point22(2)-point22(1)*point23(2))*point21(3)+(point21(1)*point23(2)- &
!        point23(1)*point21(2))*point22(3)+(point22(1)*point21(2)-point21(1)*point22(2))*point23(3)

        a22=(point24(2)-point22(2))*point21(3)+(point21(2)-point24(2))*point22(3)+(point22(2)-point21(2))*point24(3)
        b22=(point22(1)-point24(1))*point21(3)+(point24(1)-point21(1))*point22(3)+(point21(1)-point22(1))*point24(3)
        c22=(point24(1)-point22(1))*point21(2)+(point21(1)-point24(1))*point22(2)+(point22(1)-point21(1))*point24(2)
!       d12=(point24(1)*point22(2)-point22(1)*point24(2))*point21(3)+(point21(1)*point24(2)- &
!        point24(1)*point21(2))*point22(3)+(point22(1)*point21(2)-point21(1)*point22(2))*point24(3)
!**************************
!       Calcolo dei coseni
!**************************
        cos12= (a11*a12+b11*b12+c11*c12)/(SQRT(a11**2+b11**2+c11**2)*SQRT(a12**2+b12**2+c12**2))
        cos23= (a21*a22+b21*b22+c21*c22)/(SQRT(a21**2+b21**2+c21**2)*SQRT(a22**2+b22**2+c22**2))
!********************************
!       CALCOLO DELLA PLANARITA'
!********************************
        a=-1
        pigreco=ACOS(a)

        angle12=ACOS(cos12)
        angle23=ACOS(cos23)
!       angle12mod=angle12
!       angle23mod=angle23
!       IF (angle12mod.GT.pigreco/2) THEN
!           angle12mod=angle12mod-pigreco
!       ENDIF
!       IF (angle23mod.GT.pigreco/2) THEN
!           angle23mod=angle23mod-pigreco
!       ENDIF
!       plan=(COS(angle12mod)+COS(angle23mod))/2
!       conf=(COS(angle12)+COS(angle23))/2
!       planvett((j+12)/12)=plan
!       confvett((j+12)/12)=conf
!****************************
!       Scrittura su file
!****************************
!write(*,*) plan, conf
!write(*,103) (j+12)/12, temp, dist1, dist2, dist3, angle12*180/pigreco, angle23*180/pigreco, plan
!pause
        write(2,103) (j+12)/12, temp, angle12*180/pigreco, angle23*180/pigreco
        i=i+12
    END DO
!****************************************** ! Scrittura su file
della planarit\'{a} media
!****************************************** !
meanplan=SUM(planvett)/216 !   write(3,105) temp, meanplan
!*************************************************************************************************
!********************************************FINE PRIMA
PARTE**************************************
!*************************************************************************************************


!*************************************************************************************************
!************RICOSTRUZIONE DEL FILE DI TRAIETTORIE PER L'ANALISI DELLA PLANARITA'*****************
!*************************************************************************************************
    i=1
    j=i-1
    k=1
!   tol=0.06
countBB=0
countFF=0
countSS=0
countPP=0
    frase2='ICE                          '
    write(4,101) frase, temp
    write(4,102) newatoms
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,101) frase2
                                                    write(5,102) newatoms
                                                END IF
    DO WHILE (i.LE.totatoms)
        j=i-1
        IF (planvett(k).GT.1-tol) THEN
            IF (confvett(k).GT.1-tol) THEN
                countPP=countPP+1
!               matr(j+2)%secon='PDDD '
                matr(j+2)%terzo='   PP'
                matr(j+2)%quarto=k
                write(4,100) matr(j+2)
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,100) matr(j+2)
                                                END IF
            ELSE IF (confvett(k).LT.-1+tol) THEN
                countFF=countFF+1
!               matr(j+2)%secon='FDDD '
                matr(j+2)%terzo='   FF'
                matr(j+2)%quarto=k
                write(4,100) matr(j+2)
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,100) matr(j+2)
                                                END IF
            ELSE IF (confvett(k).LT.tol.AND.confvett(k).GT.-tol) THEN
                countSS=countSS+1
!               matr(j+2)%secon='SDDD '
                matr(j+2)%terzo='   SS'
                matr(j+2)%quarto=k
                write(4,100) matr(j+2)
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,100) matr(j+2)
                                                END IF
            ELSE
                write(*,*) 'ERRORE!!!!!'
            END IF
        ELSE
            countBB=countBB+1
            matr(j+2)%terzo='   BB'
            matr(j+2)%quarto=k
            write(4,100) matr(j+2)
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,100) matr(j+2)
                                                END IF
        END IF
            i=i+12
            k=k+1
    END DO
    write(4,104) dim
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,104) dim
                                                END IF

    write(6,106) temp, countPP, countSS, countFF, countBB

varcontrol=2

ENDDO

CLOSE(2, STATUS='KEEP')
CLOSE(3, STATUS='KEEP')
CLOSE(4, STATUS='KEEP')
CLOSE(5, STATUS='KEEP')

100 format (i5,2a5,i5,3f8.3) !,3f8.4)
101 format (1a29,1f15.5)
102 format (i5)
103 format (i4,1f15.5,2f10.5)
104 format (3f10.5)
105 format (2f15.5)
106 format (1f15.5,4i5)
end program
```

C.2 Trajectory extraction

This file extracts coarse-grained trajectory from a united-atoms atomistic trajectory.

```
Program estrtraj
implicit none
Integer :: i,j,k
character (len=29) frase
Integer totatm
Doubleprecision temp
Integer totatoms,newatoms
Parameter (totatoms=8424)
Parameter (newatoms=2592)
TYPE rigo
    integer primo
    character (len=5) secon
    character (len=5) terzo
    integer quarto
    doubleprecision quinto
    doubleprecision sesto
    doubleprecision settimo
!   doubleprecision ottavo
!   doubleprecision nono
!   doubleprecision decimo
ENDTYPE
type (rigo), dimension(totatoms) :: matr
Doubleprecision, dimension(3) :: dim

Open ( UNIT=1, FILE= 'traj.gro', STATUS='OLD')
Open ( UNIT=2, FILE= 'trajcg.gro', STATUS='UNKNOWN')


DO WHILE (.NOT. EOF(1))
    read(1,101) frase, temp
    read(1,102) totatm
    i=1
    DO WHILE (i.LE.totatoms)
        read (1,100) matr(i)
!       write (*,100) matr(i)
        i=i+1
    END DO
    read(1,104) dim
    i=1
    j=i-1
    k=0
    write(2,101) frase, temp
    write(2,102) newatoms
    DO WHILE (i.LE.totatoms)
    j=i-1
        matr(j+4)%secon='DDDm'
        matr(j+4)%terzo='   N1'
        matr(j+4)%quarto=k+1
        write(2,100) matr(j+4)
        matr(j+5)%secon='DDDm'
        matr(j+5)%terzo='   N2'
        matr(j+5)%quarto=k+2
        write(2,100) matr(j+5)
        matr(j+6)%secon='DDDm'
        matr(j+6)%terzo='   N1'
        matr(j+6)%quarto=k+3
        write(2,100) matr(j+6)
        matr(j+16)%secon='DDDm'
        matr(j+16)%terzo='   C1'
        matr(j+16)%quarto=k+4
        write(2,100) matr(j+16)
        matr(j+17)%secon='DDDm'
        matr(j+17)%terzo='   C1'
        matr(j+17)%quarto=k+5
        write(2,100) matr(j+17)
        matr(j+18)%secon='DDDm'
        matr(j+18)%terzo='   C1'
        matr(j+18)%quarto=k+6
        write(2,100) matr(j+18)
        matr(j+25)%secon='DDDm'
        matr(j+25)%terzo='   C1'
        matr(j+25)%quarto=k+7
        write(2,100) matr(j+25)
        matr(j+26)%secon='DDDm'
        matr(j+26)%terzo='   C1'
        matr(j+26)%quarto=k+8
        write(2,100) matr(j+26)
        matr(j+27)%secon='DDDm'
        matr(j+27)%terzo='   C1'
        matr(j+27)%quarto=k+9
        write(2,100) matr(j+27)
        matr(j+34)%secon='DDDm'
        matr(j+34)%terzo='   C1'
        matr(j+34)%quarto=k+10
        write(2,100) matr(j+34)
        matr(j+35)%secon='DDDm'
        matr(j+35)%terzo='   C1'
        matr(j+35)%quarto=k+11
        write(2,100) matr(j+35)
        matr(j+36)%secon='DDDm'
        matr(j+36)%terzo='   C1'
        matr(j+36)%quarto=k+12
        write(2,100) matr(j+36)
        i=i+39
        k=k+12
    END DO
    write(2,104) dim
ENDDO


!i=1
!DO WHILE (i.LE.totmoli)
!   write(2,102) i
!   i=i+1
!END DO

CLOSE(2, STATUS='KEEP')

100 format (i5,2a5,i5,3f8.3) !,3f8.4)
101 format (1a29,1f10.5)
102 format (i5)
103 format (1f10.5)
104 format (3f10.5)
end program
```

C.3 Planarity calculations

```
Program planarindex
```

```
implicit none
Integer :: i,j,k,j0
character (len=29) frase, frase2
Integer totatm
Doubleprecision temp
Integer totatoms, newatoms
!Parameter (totatoms=2592)
!Parameter (newatoms=216)
Parameter (totatoms=20736)
Parameter (newatoms=1728)
TYPE rigo
    integer primo
    character (len=5) secon
    character (len=5) terzo
    integer quarto
    doubleprecision quinto
    doubleprecision sesto
    doubleprecision settimo
!   doubleprecision ottavo
!   doubleprecision nono
!   doubleprecision decimo
ENDTYPE
type (rigo), dimension(totatoms) :: matr
Doubleprecision, dimension(3) :: dim
!
Doubleprecision :: dist1, dist2, dist3
Doubleprecision, dimension(3) :: point11,point12,point13,point14
Doubleprecision, dimension(3) :: point21,point22,point23,point24
Doubleprecision :: a11,b11,c11,d11,a12,b12,c12,d12
Doubleprecision :: a21,b21,c21,d21,a22,b22,c22,d22
Doubleprecision :: cos12,cos23
Doubleprecision :: angle12, angle23
Doubleprecision :: angle12mod, angle23mod, plan, meanplan, conf
Doubleprecision, dimension(newatoms) :: planvett, confvett
Doubleprecision :: pigreco, a
Integer :: countPP, countSS, countFF, countBB
!
Doubleprecision :: tol
Integer :: varcontrol
Integer ::jbis

!Open ( UNIT=1, FILE= 'traj.gro', STATUS='OLD')
Open ( UNIT=1, FILE= 'dddboxcg110shearmax22eq_2.gro', STATUS='OLD')
Open ( UNIT=2, FILE= 'planar.txt', STATUS='UNKNOWN')
Open ( UNIT=3, FILE= 'meanplanar.txt', STATUS='UNKNOWN')
Open ( UNIT=4, FILE= 'trajplanar.gro', STATUS='UNKNOWN')
Open ( UNIT=5, FILE= 'dddboxplanar.gro', STATUS='UNKNOWN')
Open ( UNIT=6, FILE= 'count.txt', STATUS='UNKNOWN')
!Open ( UNIT=7, FILE= 'molSS.ndx', STATUS='UNKNOWN')

varcontrol=1
tol=0.1
    write(6,*) '     tempo    Comb Chair Fork Non-planar'
    write(6,*) tol

DO WHILE (.NOT. EOF(1))
!*********************************
!   LETTURA DELLA CONFIGURAZIONE
!*********************************
    read(1,101) frase, temp
    read(1,102) totatm
    i=1
    DO WHILE (i.LE.totatoms)
        read (1,100) matr(i)
        i=i+1
    END DO
    read(1,104) dim
!********************************************************
!   CALCOLO DEI PARAMETRI DI LUNGHEZZA DELLE CATENE
!********************************************************
    i=1
    j=i-1
    k=0
    DO WHILE (i.LE.totatoms)
    j=i-1
        dist1=SQRT((matr(j+1)%quinto-matr(j+10)%quinto)**2+ &
            (matr(j+1)%sesto-matr(j+10)%sesto)**2+ &
            (matr(j+1)%settimo-matr(j+10)%settimo)**2)
        dist2=SQRT((matr(j+2)%quinto-matr(j+11)%quinto)**2+ &
            (matr(j+2)%sesto-matr(j+11)%sesto)**2+ &
            (matr(j+2)%settimo-matr(j+11)%settimo)**2)
        dist3=SQRT((matr(j+3)%quinto-matr(j+12)%quinto)**2+ &
            (matr(j+3)%sesto-matr(j+12)%sesto)**2+ &
            (matr(j+3)%settimo-matr(j+12)%settimo)**2)
```

```
!****************************
!       CALCOLO DEI DIEDRI
!****************************
        point11=(/matr(j+1)%quinto,matr(j+1)%sesto,matr(j+1)%settimo/)
        point12=(/matr(j+2)%quinto,matr(j+2)%sesto,matr(j+2)%settimo/)
        point13=(/matr(j+11)%quinto,matr(j+11)%sesto,matr(j+11)%settimo/)
        point14=(/matr(j+10)%quinto,matr(j+10)%sesto,matr(j+10)%settimo/)

        point21=(/matr(j+2)%quinto,matr(j+2)%sesto,matr(j+2)%settimo/)
        point22=(/matr(j+3)%quinto,matr(j+3)%sesto,matr(j+3)%settimo/)
        point23=(/matr(j+11)%quinto,matr(j+11)%sesto,matr(j+11)%settimo/)
        point24=(/matr(j+12)%quinto,matr(j+12)%sesto,matr(j+12)%settimo/)
!**********************************************
!       Coefficienti delle equazioni dei piani
!**********************************************
        a11=(point13(2)-point12(2))*point11(3)+(point11(2)-point13(2))*point12(3)+(point12(2)-point11(2))*point13(3)
        b11=(point12(1)-point13(1))*point11(3)+(point13(1)-point11(1))*point12(3)+(point11(1)-point12(1))*point13(3)
        c11=(point13(1)-point12(1))*point11(2)+(point11(1)-point13(1))*point12(2)+(point12(1)-point11(1))*point13(2)
!       d11=(point13(1)*point12(2)-point12(1)*point13(2))*point11(3)+(point11(1)*point13(2)- &
!        point13(1)*point11(2))*point12(3)+(point12(1)*point11(2)-point11(1)*point12(2))*point13(3)

        a12=(point14(2)-point12(2))*point11(3)+(point11(2)-point14(2))*point12(3)+(point12(2)-point11(2))*point14(3)
        b12=(point12(1)-point14(1))*point11(3)+(point14(1)-point11(1))*point12(3)+(point11(1)-point12(1))*point14(3)
        c12=(point14(1)-point12(1))*point11(2)+(point11(1)-point14(1))*point12(2)+(point12(1)-point11(1))*point14(2)
!       d12=(point14(1)*point12(2)-point12(1)*point14(2))*point11(3)+(point11(1)*point14(2)- &
!        point14(1)*point11(2))*point12(3)+(point12(1)*point11(2)-point11(1)*point12(2))*point14(3)

        a21=(point23(2)-point22(2))*point21(3)+(point21(2)-point23(2))*point22(3)+(point22(2)-point21(2))*point23(3)
        b21=(point22(1)-point23(1))*point21(3)+(point23(1)-point21(1))*point22(3)+(point21(1)-point22(1))*point23(3)
        c21=(point23(1)-point22(1))*point21(2)+(point21(1)-point23(1))*point22(2)+(point22(1)-point21(1))*point23(2)
!       d21=(point23(1)*point22(2)-point22(1)*point23(2))*point21(3)+(point21(1)*point23(2)- &
!        point23(1)*point21(2))*point22(3)+(point22(1)*point21(2)-point21(1)*point22(2))*point23(3)

        a22=(point24(2)-point22(2))*point21(3)+(point21(2)-point24(2))*point22(3)+(point22(2)-point21(2))*point24(3)
        b22=(point22(1)-point24(1))*point21(3)+(point24(1)-point21(1))*point22(3)+(point21(1)-point22(1))*point24(3)
        c22=(point24(1)-point22(1))*point21(2)+(point21(1)-point24(1))*point22(2)+(point22(1)-point21(1))*point24(2)
!       d12=(point24(1)*point22(2)-point22(1)*point24(2))*point21(3)+(point21(1)*point24(2)- &
!        point24(1)*point21(2))*point22(3)+(point22(1)*point21(2)-point21(1)*point22(2))*point24(3)
!***************************
!       Calcolo dei coseni
!***************************
        cos12= (a11*a12+b11*b12+c11*c12)/(SQRT(a11**2+b11**2+c11**2)*SQRT(a12**2+b12**2+c12**2))
        cos23= (a21*a22+b21*b22+c21*c22)/(SQRT(a21**2+b21**2+c21**2)*SQRT(a22**2+b22**2+c22**2))
!*********************************
!       CALCOLO DELLA PLANARITA'
!*********************************
        a=-1
        pigreco=ACOS(a)

        angle12=ACOS(cos12)
        angle23=ACOS(cos23)
        angle12mod=angle12
        angle23mod=angle23
        IF (angle12mod.GT.pigreco/2) THEN
            angle12mod=angle12mod-pigreco
        ENDIF
        IF (angle23mod.GT.pigreco/2) THEN
            angle23mod=angle23mod-pigreco
        ENDIF
        plan=(COS(angle12mod)+COS(angle23mod))/2
        conf=(COS(angle12)+COS(angle23))/2
        planvett((j+12)/12)=plan
        confvett((j+12)/12)=conf
!****************************
!       Scrittura su file
!****************************
!write(*,*) plan, conf
!write(*,103) (j+12)/12, temp, dist1, dist2, dist3, angle12*180/pigreco, angle23*180/pigreco, plan
!pause
        write(2,103) (j+12)/12, temp, dist1, dist2, dist3, angle12*180/pigreco, angle23*180/pigreco, plan
        i=i+12
    END DO
!****************************************** ! Scrittura su file
della planarit\'{a} media
!******************************************
    meanplan=SUM(planvett)/216
    write(3,105) temp, meanplan
!*************************************************************************************************
!*******************************************FINE PRIMA PARTE**************************************
!*************************************************************************************************


!*************************************************************************************************
!************RICOSTRUZIONE DEL FILE DI TRAIETTORIE PER L'ANALISI DELLA PLANARITA'*****************
```

```
!***************************************************************************************************
    i=1
    j=i-1
    k=1
    jbis=1
!   tol=0.06
countBB=0
countFF=0
countSS=0
countPP=0
    frase2='ICE                          '
    write(4,101) frase, temp
    write(4,102) totatoms
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,101) frase2
                                                    write(5,102) newatoms
                                                END IF
    DO WHILE (i.LE.totatoms)
        j=i-1
        jbis=1
        IF (planvett(k).GT.1-tol) THEN
            IF (confvett(k).GT.1-tol) THEN
                countPP=countPP+1
!               matr(j+2)%secon='PDDD '
                DO while (jbis.LE.12)
                matr(j+jbis)%secon='PPP  '

                write(4,100) matr(j+jbis)
                jbis=jbis+1
                ENDDO
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,100) matr(j+2)
                                                END IF
            ELSE IF (confvett(k).LT.-1+tol) THEN
                countFF=countFF+1
!               matr(j+2)%secon='FDDD '
                DO while (jbis.LE.12)
                matr(j+jbis)%secon='FFF  '

                write(4,100) matr(j+jbis)
                jbis=jbis+1
                ENDDO
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,100) matr(j+2)
                                                END IF
            ELSE IF (confvett(k).LT.tol.AND.confvett(k).GT.-tol) THEN
                countSS=countSS+1
!               matr(j+2)%secon='SDDD '
            DO while (jbis.LE.12)
                matr(j+jbis)%secon='SSS  '

                write(4,100) matr(j+jbis)
                jbis=jbis+1
                ENDDO
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,100) matr(j+2)
                                                END IF
            ELSE
                write(*,*) 'ERRORE!!!!!'
            END IF
        ELSE
            countBB=countBB+1
                DO while (jbis.LE.12)
                matr(j+jbis)%secon='BBB  '

                write(4,100) matr(j+jbis)
                jbis=jbis+1
                ENDDO
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,100) matr(j+2)
                                                END IF

        END IF
            i=i+12
            k=k+1
            jbis=1
    END DO
    write(4,104) dim
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,104) dim
                                                END IF

    write(6,106) temp, countPP, countSS, countFF, countBB
```

```
varcontrol=2


ENDDO

CLOSE(2, STATUS='KEEP')
CLOSE(3, STATUS='KEEP')
CLOSE(4, STATUS='KEEP')
CLOSE(5, STATUS='KEEP')


100 format (i5,2a5,i5,3f8.3) !,3f8.4)
101 format (1a29,1f15.5)
102 format (i5)
103 format (i4,1f15.5,6f10.5)
104 format (3f10.5)
105 format (2f15.5)
106 format (1f15.5,4i5)
end program
```

C.4 Calculations of radial distribution functions

C.4.1 Index calculation software

```
Program GDRindex
implicit none
Integer :: i,j,k
character (len=29) frase, frase2
Integer totatm
Doubleprecision temp
Integer totatoms, newatoms
!Parameter (totatoms=2592)
!Parameter (newatoms=216)
Parameter (totatoms=20736)
Parameter (newatoms=1728)
TYPE rigo
    integer primo
    character (len=5) secon
    character (len=5) terzo
    integer quarto
    doubleprecision quinto
    doubleprecision sesto
    doubleprecision settimo
!   doubleprecision ottavo
!   doubleprecision nono
!   doubleprecision decimo
ENDTYPE
type (rigo), dimension(totatoms) :: matr
Doubleprecision, dimension(3) :: dim
!
Doubleprecision :: dist1, dist2, dist3
Doubleprecision, dimension(3) :: point11,point12,point13,point14
Doubleprecision, dimension(3) :: point21,point22,point23,point24
Doubleprecision :: a11,b11,c11,d11,a12,b12,c12,d12
Doubleprecision :: a21,b21,c21,d21,a22,b22,c22,d22
Doubleprecision :: cos12,cos23
Doubleprecision :: angle12, angle23
Doubleprecision :: angle12mod, angle23mod, plan, meanplan, conf
Doubleprecision, dimension(newatoms) :: planvett, confvett
Doubleprecision :: pigreco, a
Integer :: countPP, countSS, countFF, countBB
!
Doubleprecision :: tol
Integer :: varcontrol
Integer ::jbis

!Open ( UNIT=1, FILE= 'trajcg.gro', STATUS='OLD')
Open ( UNIT=1, FILE= 'dddboxcgshear105max22_005.gro', STATUS='OLD')
Open ( UNIT=2, FILE= 'planar.txt', STATUS='UNKNOWN')
Open ( UNIT=3, FILE= 'meanplanar.txt', STATUS='UNKNOWN')
Open ( UNIT=4, FILE= 'trajplanar.gro', STATUS='UNKNOWN')
Open ( UNIT=5, FILE= 'dddboxplanar.gro', STATUS='UNKNOWN')
Open ( UNIT=6, FILE= 'count.txt', STATUS='UNKNOWN')
Open ( UNIT=7, FILE= 'BBB.gro', STATUS='UNKNOWN')
Open ( UNIT=8, FILE= 'SSS.gro', STATUS='UNKNOWN')
Open ( UNIT=9, FILE= 'PPP.gro', STATUS='UNKNOWN')
Open ( UNIT=10, FILE= 'FFF.gro', STATUS='UNKNOWN')
Open ( UNIT=11, FILE= 'BBBnum.txt', STATUS='UNKNOWN')
Open ( UNIT=12, FILE= 'SSSnum.txt', STATUS='UNKNOWN')
Open ( UNIT=13, FILE= 'PPPnum.txt', STATUS='UNKNOWN')
Open ( UNIT=14, FILE= 'FFFnum.txt', STATUS='UNKNOWN')

varcontrol=1
tol=0.1
    write(6,*) '      tempo    Comb Chair Fork Non-planar'
    write(6,*) tol

DO WHILE (.NOT. EOF(1))
!********************************
!   LETTURA DELLA CONFIGURAZIONE
!********************************
    read(1,101) frase, temp
    read(1,102) totatm
    i=1
    DO WHILE (i.LE.totatoms)
        read (1,100) matr(i)
        i=i+1
    END DO
    read(1,104) dim
!******************************************************
!   CALCOLO DEI PARAMETRI DI LUNGHEZZA DELLE CATENE
!******************************************************
    i=1
    j=i-1
    k=0
    DO WHILE (i.LE.totatoms)
    j=i-1
        dist1=SQRT((matr(j+1)%quinto-matr(j+10)%quinto)**2+ &
            (matr(j+1)%sesto-matr(j+10)%sesto)**2+ &
            (matr(j+1)%settimo-matr(j+10)%settimo)**2)
        dist2=SQRT((matr(j+2)%quinto-matr(j+11)%quinto)**2+ &
            (matr(j+2)%sesto-matr(j+11)%sesto)**2+ &
            (matr(j+2)%settimo-matr(j+11)%settimo)**2)
        dist3=SQRT((matr(j+3)%quinto-matr(j+12)%quinto)**2+ &
            (matr(j+3)%sesto-matr(j+12)%sesto)**2+ &
            (matr(j+3)%settimo-matr(j+12)%settimo)**2)
!***************************
!       CALCOLO DEI DIEDRI
!***************************
        point11=(/matr(j+1)%quinto,matr(j+1)%sesto,matr(j+1)%settimo/)
        point12=(/matr(j+2)%quinto,matr(j+2)%sesto,matr(j+2)%settimo/)
        point13=(/matr(j+11)%quinto,matr(j+11)%sesto,matr(j+11)%settimo/)
        point14=(/matr(j+10)%quinto,matr(j+10)%sesto,matr(j+10)%settimo/)

        point21=(/matr(j+2)%quinto,matr(j+2)%sesto,matr(j+2)%settimo/)
        point22=(/matr(j+3)%quinto,matr(j+3)%sesto,matr(j+3)%settimo/)
        point23=(/matr(j+11)%quinto,matr(j+11)%sesto,matr(j+11)%settimo/)
        point24=(/matr(j+12)%quinto,matr(j+12)%sesto,matr(j+12)%settimo/)
!*********************************************
!       Coefficienti delle equazioni dei piani
!*********************************************
        a11=(point13(2)-point12(2))*point11(3)+(point11(2)-point13(2))*point12(3)+(point12(2)-point11(2))*point13(3)
        b11=(point12(1)-point13(1))*point11(3)+(point13(1)-point11(1))*point12(3)+(point11(1)-point12(1))*point13(3)
        c11=(point13(1)-point12(1))*point11(2)+(point11(1)-point13(1))*point12(2)+(point12(1)-point11(1))*point13(2)
!       d11=(point13(1)*point12(2)-point12(1)*point13(2))*point11(3)+(point11(1)*point13(2)- &
!        point13(1)*point11(2))*point12(3)+(point12(1)*point11(2)-point11(1)*point12(2))*point13(3)

        a12=(point14(2)-point12(2))*point11(3)+(point11(2)-point14(2))*point12(3)+(point12(2)-point11(2))*point14(3)
        b12=(point12(1)-point14(1))*point11(3)+(point14(1)-point11(1))*point12(3)+(point11(1)-point12(1))*point14(3)
        c12=(point14(1)-point12(1))*point11(2)+(point11(1)-point14(1))*point12(2)+(point12(1)-point11(1))*point14(2)
!       d12=(point14(1)*point12(2)-point12(1)*point14(2))*point11(3)+(point11(1)*point14(2)- &
!        point14(1)*point11(2))*point12(3)+(point12(1)*point11(2)-point11(1)*point12(2))*point14(3)

        a21=(point23(2)-point22(2))*point21(3)+(point21(2)-point23(2))*point22(3)+(point22(2)-point21(2))*point23(3)
        b21=(point22(1)-point23(1))*point21(3)+(point23(1)-point21(1))*point22(3)+(point21(1)-point22(1))*point23(3)
        c21=(point23(1)-point22(1))*point21(2)+(point21(1)-point23(1))*point22(2)+(point22(1)-point21(1))*point23(2)
!       d21=(point23(1)*point22(2)-point22(1)*point23(2))*point21(3)+(point21(1)*point23(2)- &
!        point23(1)*point21(2))*point22(3)+(point22(1)*point21(2)-point21(1)*point22(2))*point23(3)

        a22=(point24(2)-point22(2))*point21(3)+(point21(2)-point24(2))*point22(3)+(point22(2)-point21(2))*point24(3)
        b22=(point22(1)-point24(1))*point21(3)+(point24(1)-point21(1))*point22(3)+(point21(1)-point22(1))*point24(3)
        c22=(point24(1)-point22(1))*point21(2)+(point21(1)-point24(1))*point22(2)+(point22(1)-point21(1))*point24(2)
!       d12=(point24(1)*point22(2)-point22(1)*point24(2))*point21(3)+(point21(1)*point24(2)- &
!        point24(1)*point21(2))*point22(3)+(point22(1)*point21(2)-point21(1)*point22(2))*point24(3)
!**************************
!       Calcolo dei coseni
!**************************
        cos12= (a11*a12+b11*b12+c11*c12)/(SQRT(a11**2+b11**2+c11**2)*SQRT(a12**2+b12**2+c12**2))
        cos23= (a21*a22+b21*b22+c21*c22)/(SQRT(a21**2+b21**2+c21**2)*SQRT(a22**2+b22**2+c22**2))
```

```
!*********************************
!       CALCOLO DELLA PLANARITA'
!*********************************
        a=-1
        pigreco=ACOS(a)

        angle12=ACOS(cos12)
        angle23=ACOS(cos23)
        angle12mod=angle12
        angle23mod=angle23
        IF (angle12mod.GT.pigreco/2) THEN
            angle12mod=angle12mod-pigreco
        ENDIF
        IF (angle23mod.GT.pigreco/2) THEN
            angle23mod=angle23mod-pigreco
        ENDIF
        plan=(COS(angle12mod)+COS(angle23mod))/2
        conf=(COS(angle12)+COS(angle23))/2
        planvett((j+12)/12)=plan
        confvett((j+12)/12)=conf
!****************************
!       Scrittura su file
!****************************
!write(*,*) plan, conf
!write(*,103) (j+12)/12, temp, dist1, dist2, dist3, angle12*180/pigreco, angle23*180/pigreco, plan
!pause
        write(2,103) (j+12)/12, temp, dist1, dist2, dist3, angle12*180/pigreco, angle23*180/pigreco, plan
        i=i+12
    END DO
!****************************************** ! Scrittura su file
della planarit\`{a} media
!******************************************
    meanplan=SUM(planvett)/216
    write(3,105) temp, meanplan
!*************************************************************************************************
!******************************************FINE PRIMA PARTE***************************************
!*************************************************************************************************


!*************************************************************************************************
!************RICOSTRUZIONE DEL FILE DI TRAIETTORIE PER L'ANALISI DELLA PLANARITA'*****************
!*************************************************************************************************
    i=1
    j=i-1
    k=1
    jbis=1
!   tol=0.06
countBB=0
countFF=0
countSS=0
countPP=0
    frase2='ICE                          '
    write(4,101) frase, temp
    write(7,101) frase, temp    !file BBB.gro
    write(8,101) frase, temp    !file SSS.gro
    write(9,101) frase, temp    !file PPP.gro
    write(10,101) frase, temp   !file FFF.gro
    write(4,102) totatoms
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,101) frase2
                                                    write(5,102) newatoms
                                                END IF
    DO WHILE (i.LE.totatoms)
        j=i-1
        jbis=1
        IF (planvett(k).GT.1-tol) THEN
            IF (confvett(k).GT.1-tol) THEN
                countPP=countPP+1
!               matr(j+2)%secon='PDDD '
                DO while (jbis.LE.12)
                matr(j+jbis)%secon='PPP  '

                write(4,100) matr(j+jbis)
                write(9,100) matr(j+jbis)
                jbis=jbis+1
                ENDDO
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,100) matr(j+2)
                                                END IF
            ELSE IF (confvett(k).LT.-1+tol) THEN
                countFF=countFF+1
!               matr(j+2)%secon='FDDD '
                DO while (jbis.LE.12)
                matr(j+jbis)%secon='FFF  '
                write(4,100) matr(j+jbis)
                write(10,100) matr(j+jbis)
                jbis=jbis+1
                ENDDO
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,100) matr(j+2)
                                                END IF
            ELSE IF (confvett(k).LT.tol.AND.confvett(k).GT.-tol) THEN
                countSS=countSS+1
!               matr(j+2)%secon='SDDD '
            DO while (jbis.LE.12)
                matr(j+jbis)%secon='SSS  '

                write(4,100) matr(j+jbis)
                write(8,100) matr(j+jbis)
                jbis=jbis+1
                ENDDO
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,100) matr(j+2)
                                                END IF
            ELSE
                write(*,*) 'ERRORE!!!!!'
            END IF
        ELSE
            countBB=countBB+1
                DO while (jbis.LE.12)
                matr(j+jbis)%secon='BBB  '

                write(4,100) matr(j+jbis)
                write(7,100) matr(j+jbis)
                jbis=jbis+1
                ENDDO
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,100) matr(j+2)
                                                END IF

        END IF
            i=i+12
            k=k+1
            jbis=1
    END DO
    write(4,104) dim
                                                IF (varcontrol.EQ.1) THEN
                                                    write(5,104) dim
                                                END IF

    write(6,106) temp, countPP, countSS, countFF, countBB
    write(11,102) countBB
    write(12,102) countSS
    write(13,102) countPP
    write(14,102) countFF
varcontrol=2


ENDDO

CLOSE(2, STATUS='KEEP')
CLOSE(3, STATUS='KEEP')
CLOSE(4, STATUS='KEEP')
CLOSE(5, STATUS='KEEP')
CLOSE(7, STATUS='KEEP')
CLOSE(8, STATUS='KEEP')
CLOSE(9, STATUS='KEEP')
CLOSE(10, STATUS='KEEP')


100 format (i5,2a5,i5,3f8.3) !,3f8.4)
101 format (1a29,1f15.5)
102 format (i5)
103 format (i4,1f15.5,6f10.5)
104 format (3f10.5)
105 format (2f15.5)
106 format (1f15.5,4i5)
end program
```

C.4.2 Radial distribution functions

```
Program GDRcomput
implicit none
Doubleprecision minl,l,norml,deltaint, distanza
Integer :: frame,i,j,k,nframe,INT,noperation
Doubleprecision temp, a, pigreco,density, dimensione
character (len=29) frase
Integer ninter, molatoms,dimframe, minimo, bin, maxbin, natoms
parameter (ninter=2000)
Parameter (molatoms=12)
Parameter (dimframe=30000)
Parameter (minimo=0)
!Parameter (newatoms=216)
TYPE rigo
    integer primo
    character (len=5) secon
    character (len=5) terzo
    integer quarto
    doubleprecision quinto
    doubleprecision sesto
    doubleprecision settimo
!   doubleprecision ottavo
!   doubleprecision nono
!   doubleprecision decimo
ENDTYPE
type (rigo), dimension(dimframe) :: matrframe
doubleprecision, dimension(ninter) :: gdrvett

Open ( UNIT=1, FILE= 'BBB.gro', STATUS='OLD')
Open ( UNIT=2, FILE= 'SSS.gro', STATUS='OLD')
Open ( UNIT=3, FILE= 'PPP.gro', STATUS='OLD')
Open ( UNIT=4, FILE= 'FFF.gro', STATUS='OLD')
Open ( UNIT=5, FILE= 'BBBnum.txt', STATUS='OLD')
Open ( UNIT=6, FILE= 'SSSnum.txt', STATUS='OLD')
Open ( UNIT=7, FILE= 'PPPnum.txt', STATUS='OLD')
Open ( UNIT=8, FILE= 'FFFnum.txt', STATUS='OLD')
Open ( UNIT=9, FILE= 'gdrBBB.txt', STATUS='UNKNOWN')
Open ( UNIT=10, FILE= 'gdrSSS.txt', STATUS='UNKNOWN')
Open ( UNIT=11, FILE= 'gdrPPP.txt', STATUS='UNKNOWN')
Open ( UNIT=12, FILE= 'gdrFFF.txt', STATUS='UNKNOWN')

a=-1
pigreco=ACOS(a)
density=1

!*********************************************************
! SSS.gro
!*********************************************************


i=1
minl=minimo
l=4
norml=(l-minl)
deltaint=norml/ninter
nframe=0
noperation=0
maxbin=0
DO WHILE(.NOT.EOF(2))
!*********************************
!   LETTURA DEL FRAME
!*********************************
    read(2,101) frase, temp
    read(6,102) frame

    !write(*,101) frase, temp
    !write(*,102) frame
    !pause
    i=1
    DO WHILE (i.LE.molatoms*frame)
        read (2,100) matrframe(i)
        i=i+1
    END DO
!*************************************************
    i=1
    DO WHILE (i.LE.molatoms*frame)
        j=i+1
        DO WHILE (j.LE.molatoms*frame)
            IF (matrframe(i)%primo.GE.0) THEN
            !NE.matrframe(j)%primo) THEN
                distanza=1*SQRT((matrframe(i)%quinto-matrframe(j)%quinto)**2+ &
                (matrframe(i)%sesto-matrframe(j)%sesto)**2+ &
                (matrframe(i)%settimo-matrframe(j)%settimo)**2)
                noperation=noperation+1
                bin=INT(distanza/deltaint)+1
                IF (bin.LE.ninter) then
                    gdrvett(bin)=gdrvett(bin)+2
                    IF (bin.GT.maxbin) then
                    maxbin=bin
                        dimensione=minl+deltaint*maxbin
                    ENDIF
            !   else
            !       write(*,*) ’fermati’
                ENDIF
            ENDIF
            j=j+1
        END DO
        i=i+1
    ENDDO
natoms=i-1
!***************************************************
    nframe=nframe+1
    write(*,*) nframe, natoms
    pause
ENDDO
write(*,*) noperation

!*******************
!Calcolo di density*
!*******************
!   i=ninter
!   DO WHILE (i.GE.1)
!       IF (gdrvett(i).NE.0) THEN
!           dimensione=minl+deltaint*i
!           GOTO 10
!       ENDIF
!       i=ninter-1
!   ENDDO

density=(natoms)/((4/3)*pigreco*(dimensione**3))

!   write(*,*) gdrvett
    write(*,*) nframe, density, natoms/2, dimensione
    pause

!*************
!fine density*
!*************
!density=16.16
density=1
i=1
minl=minimo
DO WHILE (i.LE.ninter)
    gdrvett(i)=(gdrvett(i)/natoms/nframe)/((4/3)*pigreco*density* &
    (((minl+deltaint*i)**3)-((minl+deltaint*(i-1))**3)))
!   write(*,*) gdrvett(i)/natoms, ((4/3)*pigreco*density* &
!   (((minl+deltaint*i)**3)-((minl+deltaint*(i-1))**3)))
!   pause
    i=i+1
END DO
!write(*,*) gdrvett

i=1
minl=minimo
DO WHILE (i.LE.ninter)
    write (10,105) minl, gdrvett(i)
    !gdrvett(ninter)
    minl=minl+deltaint
    i=i+1
END DO
write(*,*) ’FATTO SSS!!!!’

!*****************************************************
! FINE SSS.gro
!*****************************************************

!*********************************************************
! BBB.gro
!*********************************************************


i=1

nframe=0
noperation=0
maxbin=0
DO WHILE(.NOT.EOF(1))
!*********************************
!   LETTURA DEL FRAME
```

```
!*********************************
    read(1,101) frase, temp
    read(5,102) frame

    !write(*,101) frase, temp
    !write(*,102) frame
    !pause
    i=1
    DO WHILE (i.LE.molatoms*frame)
        read (1,100) matrframe(i)
        i=i+1
    END DO
!*************************************************
    i=1
    DO WHILE (i.LE.molatoms*frame)
        j=i+1
        DO WHILE (j.LE.molatoms*frame)
            IF (matrframe(i)%primo.GE.0) THEN
            !NE.matrframe(j)%primo) THEN
                distanza=1*SQRT((matrframe(i)%quinto-matrframe(j)%quinto)**2+ &
                (matrframe(i)%sesto-matrframe(j)%sesto)**2+ &
                (matrframe(i)%settimo-matrframe(j)%settimo)**2)
                noperation=noperation+1
                bin=INT(distanza/deltaint)+1
                IF (bin.LE.ninter) then
                    gdrvett(bin)=gdrvett(bin)+2
                    IF (bin.GT.maxbin) then
                    maxbin=bin
                        dimensione=minl+deltaint*maxbin
                    ENDIF
            !   else
            !       write(*,*) 'fermati'
                ENDIF
            ENDIF
            j=j+1
        END DO
        i=i+1
    ENDDO
natoms=i-1
!***************************************************
    nframe=nframe+1
    write(*,*) nframe, natoms
    pause
ENDDO
write(*,*) noperation

!*******************
!Calcolo di density*
!*******************
!   i=ninter
!   DO WHILE (i.GE.1)
!       IF (gdrvett(i).NE.0) THEN
!           dimensione=minl+deltaint*i
!           GOTO 10
!       ENDIF
!       i=ninter-1
!   ENDDO

density=(natoms)/((4/3)*pigreco*(dimensione**3))

!   write(*,*) gdrvett
    write(*,*) nframe, density, natoms/2, dimensione
    pause

!*************
!fine density*
!*************
density=1
i=1
minl=minimo
DO WHILE (i.LE.ninter)
    gdrvett(i)=(gdrvett(i)/natoms/nframe)/((4/3)*pigreco*density* &
    (((minl+deltaint*i)**3)-((minl+deltaint*(i-1))**3)))
!   write(*,*) gdrvett(i)/natoms, ((4/3)*pigreco*density* &
!   (((minl+deltaint*i)**3)-((minl+deltaint*(i-1))**3)))
!   pause
    i=i+1
END DO
!write(*,*) gdrvett

i=1
minl=minimo
DO WHILE (i.LE.ninter)
    write (9,105) minl, gdrvett(i)
    !gdrvett(ninter)
    minl=minl+deltaint
    i=i+1
END DO
write(*,*) 'FATTO BBB!!!!'

!*****************************************************
! FINE BBB.gro
!*****************************************************

!*********************************************************
! FFF.gro
!*********************************************************


i=1

nframe=0
noperation=0
maxbin=0
DO WHILE(.NOT.EOF(4))
!*********************************
!   LETTURA DEL FRAME
!*********************************
    read(4,101) frase, temp
    read(8,102) frame

    !write(*,101) frase, temp
    !write(*,102) frame
    !pause
    i=1
    DO WHILE (i.LE.molatoms*frame)
        read (4,100) matrframe(i)
        i=i+1
    END DO
!*************************************************
    i=1
    DO WHILE (i.LE.molatoms*frame)
        j=i+1
        DO WHILE (j.LE.molatoms*frame)
            IF (matrframe(i)%primo.GE.0) THEN
            !NE.matrframe(j)%primo) THEN
                distanza=1*SQRT((matrframe(i)%quinto-matrframe(j)%quinto)**2+ &
                (matrframe(i)%sesto-matrframe(j)%sesto)**2+ &
                (matrframe(i)%settimo-matrframe(j)%settimo)**2)
                noperation=noperation+1
                bin=INT(distanza/deltaint)+1
                IF (bin.LE.ninter) then
                    gdrvett(bin)=gdrvett(bin)+2
                    IF (bin.GT.maxbin) then
                    maxbin=bin
                        dimensione=minl+deltaint*maxbin
                    ENDIF
            !   else
            !       write(*,*) 'fermati'
                ENDIF
            ENDIF
            j=j+1
        END DO
        i=i+1
    ENDDO
natoms=i-1
!***************************************************
    nframe=nframe+1
    write(*,*) nframe, natoms
    pause
ENDDO
write(*,*) noperation

!*******************
!Calcolo di density*
!*******************
!   i=ninter
!   DO WHILE (i.GE.1)
!       IF (gdrvett(i).NE.0) THEN
!           dimensione=minl+deltaint*i
!           GOTO 10
!       ENDIF
!       i=ninter-1
!   ENDDO

density=(natoms)/((4/3)*pigreco*(dimensione**3))

!   write(*,*) gdrvett
```

```
    write(*,*) nframe, density, natoms/2, dimensione
    pause

!*************
!fine density*
!*************
density=1
i=1
minl=minimo
DO WHILE (i.LE.ninter)
    gdrvett(i)=(gdrvett(i)/natoms/nframe)/((4/3)*pigreco*density* &
    (((minl+deltaint*i)**3)-((minl+deltaint*(i-1))**3)))
!   write(*,*) gdrvett(i)/natoms, ((4/3)*pigreco*density* &
!   (((minl+deltaint*i)**3)-((minl+deltaint*(i-1))**3)))
!   pause
    i=i+1
END DO
!write(*,*) gdrvett

i=1
minl=minimo
DO WHILE (i.LE.ninter)
    write (12,105) minl, gdrvett(i)
    !gdrvett(ninter)
    minl=minl+deltaint
    i=i+1
END DO
write(*,*) 'FATTO FFF!!!!'

!*****************************************************
! FINE FFF.gro
!*****************************************************




!*********************************************************
! PPP.gro
!*********************************************************


i=1

nframe=0
noperation=0
maxbin=0
DO WHILE(.NOT.EOF(3))
!*********************************
!   LETTURA DEL FRAME
!*********************************
    read(3,101) frase, temp
    read(7,102) frame

    !write(*,101) frase, temp
    !write(*,102) frame
    !pause
    i=1
    DO WHILE (i.LE.molatoms*frame)
        read (3,100) matrframe(i)
        i=i+1
    END DO
!*************************************************
    i=1
    DO WHILE (i.LE.molatoms*frame)
        j=i+1
        DO WHILE (j.LE.molatoms*frame)
            IF (matrframe(i)%primo.GE.0) THEN
            !NE.matrframe(j)%primo) THEN
                distanza=1*SQRT((matrframe(i)%quinto-matrframe(j)%quinto)**2+ &
                (matrframe(i)%sesto-matrframe(j)%sesto)**2+ &
                (matrframe(i)%settimo-matrframe(j)%settimo)**2)
                noperation=noperation+1
                bin=INT(distanza/deltaint)+1
                IF (bin.LE.ninter) then
                    gdrvett(bin)=gdrvett(bin)+2
                    IF (bin.GT.maxbin) then
                    maxbin=bin
                        dimensione=minl+deltaint*maxbin
                    ENDIF
            !   else
            !       write(*,*) 'fermati'
                ENDIF
            ENDIF
            j=j+1
        END DO
        i=i+1
    ENDDO
natoms=i-1
!***************************************************
    nframe=nframe+1
    write(*,*) nframe, natoms
    pause
ENDDO
write(*,*) noperation

!*******************
!Calcolo di density*
!*******************
!   i=ninter
!   DO WHILE (i.GE.1)
!       IF (gdrvett(i).NE.0) THEN
!           dimensione=minl+deltaint*i
!           GOTO 10
!       ENDIF
!       i=ninter-1
!   ENDDO

density=(natoms)/((4/3)*pigreco*(dimensione**3))

!   write(*,*) gdrvett
    write(*,*) nframe, density, natoms/2, dimensione
    pause

!*************
!fine density*
!*************
density=1
i=1
minl=minimo
DO WHILE (i.LE.ninter)
    gdrvett(i)=(gdrvett(i)/natoms/nframe)/((4/3)*pigreco*density* &
    (((minl+deltaint*i)**3)-((minl+deltaint*(i-1))**3)))
!   write(*,*) gdrvett(i)/natoms, ((4/3)*pigreco*density* &
!   (((minl+deltaint*i)**3)-((minl+deltaint*(i-1))**3)))
!   pause
    i=i+1
END DO
!write(*,*) gdrvett

i=1
minl=minimo
DO WHILE (i.LE.ninter)
    write (11,105) minl, gdrvett(i)
    !gdrvett(ninter)
    minl=minl+deltaint
    i=i+1
END DO
write(*,*) 'FATTO PPP!!!!'

!*****************************************************
! FINE PPP.gro
!*****************************************************

CLOSE(1, STATUS='KEEP')
CLOSE(2, STATUS='KEEP')
CLOSE(3, STATUS='KEEP')
CLOSE(4, STATUS='KEEP')
CLOSE(5, STATUS='KEEP')
CLOSE(7, STATUS='KEEP')
CLOSE(8, STATUS='KEEP')
CLOSE(9, STATUS='KEEP')
CLOSE(10, STATUS='KEEP')


100 format (i5,2a5,i5,3f8.3) !,3f8.4)
101 format (1a29,1f15.5)
102 format (i5)
103 format (i4,1f15.5,6f10.5)
104 format (3f10.5)
105 format (2f15.5)
106 format (1f15.5,4i5)
end program
```

C.5 Distribution function of planar molecules

```
Program distrconform
implicit none
Integer :: i
character (len=50) frase
Doubleprecision minl,minimo,l,norml, deltaint, ntot
Integer inter, bin
Integer newatoms, righi
Doubleprecision tol, comb,chair,fork,nonplanar,bincomb,binchair,binfork,binnonplanar
Parameter (righi=1000)
Parameter (newatoms=216)
!Parameter (newatoms=1728)
parameter (inter=100)
Parameter (minimo=0)
Parameter (l=100)
TYPE rigo
    doubleprecision tempo
    integer comb
    integer chair
    integer fork
    integer nonplanar
ENDTYPE
type (rigo), dimension(righi) :: matr
doubleprecision, dimension(inter) :: distcomb, distchair, distfork, distnonplanar

Open ( UNIT=1, FILE= 'countminflow.txt', STATUS='OLD')
Open ( UNIT=2, FILE= 'distrcount.txt', STATUS='UNKNOWN')

i=1
minl=minimo
norml=(l-minl)
deltaint=norml/(inter)
ntot=0



!**************************************************************
!********    Calcolo distribuzione conformazioni     **********
!**************************************************************
read(1,111) frase
    read(1,112) tol
DO WHILE(.NOT.EOF(1))
    read (1,100) matr(i)
    comb=(matr(i)%comb/2.16)
    fork=(matr(i)%fork/2.16)
    chair=(matr(i)%chair/2.16)
    nonplanar=(matr(i)%nonplanar/2.16)
write(*,*) comb
!pause
    bincomb=INT(comb/deltaint)+1
    binfork=INT(fork/deltaint)+1
    binchair=INT(chair/deltaint)+1
    binnonplanar=INT(nonplanar/deltaint)+1
    IF (bincomb.LE.inter) then
        distcomb(bincomb)=distcomb(bincomb)+1
        ntot=ntot+1
    ENDIF
    IF (binchair.LE.inter) then
        distchair(binchair)=distchair(binchair)+1
    ENDIF
    IF (binfork.LE.inter) then
        distfork(binfork)=distfork(binfork)+1
    ENDIF
    IF (binnonplanar.LE.inter) then
        distnonplanar(binnonplanar)=distnonplanar(binnonplanar)+1
    ENDIF
        i=i+1
END DO
write(*,*) 'qui'

!***************************************************
i=1
minl=minimo

write(2,111) frase
DO WHILE (i.LE.inter)
    write (2,105) minl, distcomb(i)/ntot, distchair(i)/ntot, distfork(i)/ntot, distnonplanar(i)/ntot
    minl=minl+deltaint
    i=i+1
END DO
```

```
write(*,*) ’FATTO!!!!’


!*****************************************************
!
!*****************************************************

CLOSE(2, STATUS=’KEEP’)

100 format (1f15.5,4i5)
111 format (1a50)
112 format (1f9.7)
104 format (9f10.5)
105 format (5f13.5)
end program
```

C.6 Distribution function of inter-molecular orientation

```
Program ordinedistr
implicit none
Integer :: i,j,frame,k, noperation, binphi, bintheta
Doubleprecision ::temp
Doubleprecision :: a, pigreco
character (len=29) frase
Integer molatoms,dimframe, ninter
Doubleprecision :: minimo, minl, lphi, ltheta
Doubleprecision :: normlphi, normltheta, deltaintphi, deltainttheta
Parameter (ninter=100)
Parameter (molatoms=12)
Parameter (dimframe=30000)
Parameter (minimo=0)
TYPE rigo
    integer primo
    character (len=5) secon
    character (len=5) terzo
    integer quarto
    doubleprecision quinto
    doubleprecision sesto
    doubleprecision settimo
!   doubleprecision ottavo
!   doubleprecision nono
!   doubleprecision decimo
ENDTYPE
type (rigo), dimension(dimframe) :: matrframe
doubleprecision, dimension(ninter) :: vettphi,vetttheta
doubleprecision, dimension(dimframe) :: u, v, w
doubleprecision :: modulo
Doubleprecision :: tanthetaz, tanphixy, theta, phi, anglephi, angletheta

Open ( UNIT=1, FILE= ’SSS.gro’, STATUS=’OLD’)
Open ( UNIT=6, FILE= ’SSSnum.txt’, STATUS=’OLD’)
Open ( UNIT=10, FILE= ’coneSSS.txt’, STATUS=’UNKNOWN’)
Open ( UNIT=11, FILE= ’orientSSS.txt’, STATUS=’UNKNOWN’)

!Open ( UNIT=1, FILE= ’FFF.gro’, STATUS=’OLD’)
!Open ( UNIT=6, FILE= ’FFFnum.txt’, STATUS=’OLD’)
!Open ( UNIT=10, FILE= ’coneFFF.txt’, STATUS=’UNKNOWN’)
!Open ( UNIT=11, FILE= ’orientFFF.txt’, STATUS=’UNKNOWN’)

!Open ( UNIT=1, FILE= ’PPP.gro’, STATUS=’OLD’)
!Open ( UNIT=6, FILE= ’PPPnum.txt’, STATUS=’OLD’)
!Open ( UNIT=10, FILE= ’conePPP.txt’, STATUS=’UNKNOWN’)
!Open ( UNIT=11, FILE= ’orientPPP.txt’, STATUS=’UNKNOWN’)

!Open ( UNIT=1, FILE= ’BBB.gro’, STATUS=’OLD’)
!Open ( UNIT=6, FILE= ’BBBnum.txt’, STATUS=’OLD’)
!Open ( UNIT=10, FILE= ’coneBBB.txt’, STATUS=’UNKNOWN’)
!Open ( UNIT=11, FILE= ’orientBBB.txt’, STATUS=’UNKNOWN’)


a=-1
pigreco=ACOS(a)


!*********************************************************
```

```
!
!*********************************************************


i=1
minl=0
lphi=2*pigreco
ltheta=pigreco
normlphi=(lphi-minl)
normltheta=(ltheta-minl)
deltaintphi=normlphi/ninter
deltainttheta=normltheta/ninter
noperation=0
!natoms=0
DO WHILE(.NOT.EOF(1))
!*********************************
!   LETTURA DEL FRAME
!*********************************
    read(1,101) frase, temp
    read(6,102) frame

    !write(*,101) frase, temp
    !write(*,102) frame
    !pause
    i=1
    DO WHILE (i.LE.molatoms*frame)
        read (1,100) matrframe(i)
        i=i+1
    END DO
!*************************************************
    i=1
    k=0
    write(10,101) frase, temp
    DO WHILE (i.LE.molatoms*frame)
        IF (matrframe(i)%terzo.EQ.’   N2’) THEN
            k=k+1
            j=i+10
            u(k)=(matrframe(j)%quinto-matrframe(i)%quinto)
            v(k)=(matrframe(j)%sesto-matrframe(i)%sesto)
            w(k)=(matrframe(j)%settimo-matrframe(i)%settimo)
            write(10,110) matrframe(i)%quinto, matrframe(i)%sesto, matrframe(i)%settimo, u(k), v(k), w(k)
            noperation=noperation+1
            modulo=SQRT((u(k)**2)+(v(k)**2)+(w(k)**2))
            tanphixy=v(k)/u(k)
            tanthetaz=(SQRT(u(k)**2+v(k)**2)/w(k))

            IF (w(k).GE.0) then
                theta=ATAN(tanthetaz)
            ELSE IF(w(k).LT.0)  then
                theta=ATAN(tanthetaz)+pigreco
            ENDIF
            IF (u(k).GE.0.AND.v(k).GE.0) then
                phi=ATAN(tanphixy)
            ELSE IF (u(k).LT.0.AND.v(k).GE.0) then
                phi=ATAN(tanphixy)+pigreco
            ELSE IF (u(k).LT.0.AND.v(k).LT.0) then
                phi=ATAN(tanphixy)+pigreco
            ELSE IF (u(k).GE.0.AND.v(k).LT.0) then
                phi=ATAN(tanphixy)+2*pigreco
            ENDIF
            binphi=INT(phi/deltaintphi)+1
            IF (binphi.LE.ninter) then
                vettphi(binphi)=vettphi(binphi)+1
            ENDIF
            bintheta=INT(theta/deltainttheta)+1
            IF (bintheta.LE.ninter) then
                vetttheta(bintheta)=vetttheta(bintheta)+1
            ENDIF
        ENDIF
            i=i+1
    END DO
ENDDO

i=1
anglephi=minimo
angletheta=minimo
DO WHILE (i.LE.ninter)
    write (11,105) anglephi, vettphi(i)/noperation, angletheta, vetttheta(i)/noperation
    !gdrvett(ninter)
    anglephi=anglephi+deltaintphi
    angletheta=angletheta+deltainttheta
    i=i+1
END DO
write(*,*) ’FATTO !!!!’
```

```
CLOSE(1, STATUS='KEEP')
CLOSE(6, STATUS='KEEP')
CLOSE(10, STATUS='KEEP')


100 format (i5,2a5,i5,3f8.3) !,3f8.4)
101 format (1a29,1f15.5)
102 format (i5)
103 format (i4,1f15.5,6f10.5)
104 format (3f10.5)
105 format (4f15.5)
106 format (1f15.5,4i5)
110 format (6f8.3)
end program
```

C.7 Calculation of velocity profile inside box volume

```
Program vprofile
implicit none
Integer :: i,j,k
character (len=29) frase
Integer particles
Doubleprecision minl,minimo,l,norml, deltaint, ntot
Integer seminter, molatoms, bin, maxbin, minbin
Doubleprecision absvelocity
Doubleprecision temp
Integer totatoms,newatoms
!Parameter (totatoms=2592)
!Parameter (newatoms=216)
Parameter (totatoms=20736)
Parameter (newatoms=1728)

parameter (seminter=150)
Parameter (molatoms=12)
Parameter (minimo=-0.6)
Parameter (l=0.6)
TYPE rigo
    integer primo
    character (len=5) secon
    character (len=5) terzo
    integer quarto
    doubleprecision quinto
    doubleprecision sesto
    doubleprecision settimo
    doubleprecision ottavo
    doubleprecision nono
    doubleprecision decimo
ENDTYPE
type (rigo), dimension(totatoms) :: matr
Doubleprecision, dimension(9) :: dim
Doubleprecision, dimension(seminter) :: veldist1, veldist2
Doubleprecision :: disthigh, distlow, deltadist, vel, pdf, meanvel
Doubleprecision, dimension(300) :: pdfvel

Open ( UNIT=3, FILE= 'profile.txt', STATUS='UNKNOWN')

i=1
minl=minimo
norml=(l-minl)
deltaint=norml/(2*seminter)
ntot=0
maxbin=0
minbin=0
distlow=0

deltadist=0.5
disthigh=distlow+deltadist
!**************************************************************
!******** Calcolo distribuzione velocit             **********
!**************************************************************
DO WHILE(disthigh.LE.15)
```

```
    Open ( UNIT=1, FILE= 'trajshearmax3.gro', STATUS='OLD')
    Open ( UNIT=2, FILE= 'gammapunto.txt', POSITION='REWIND')
    DO WHILE(.NOT.EOF(1))
        read(1,101) frase, temp
        read(1,102) particles
        i=1
        DO WHILE (i.LE.particles)
            read (1,100) matr(i)
            !IF(matr(i)%terzo .EQ. '   N2') then
            !IF(matr(i)%settimo.LT.3.245.AND.matr(i)%settimo.GT.3.145) then
            IF(matr(i)%settimo.LT.disthigh.AND.matr(i)%settimo.GT.distlow) then
        !       absvelocity=SQRT(matr(i)%ottavo**2+matr(i)%nono**2+matr(i)%decimo**2)
                absvelocity=matr(i)%ottavo
                IF(absvelocity.GE.0) then
                    bin=INT(absvelocity/deltaint)+1
                    IF (bin.LE.seminter) then
                        veldist1(bin)=veldist1(bin)+1
                        ntot=ntot+1
                        IF (bin.GT.maxbin) then
                            maxbin=bin
                        ENDIF
                    ENDIF
                ENDIF
                IF(absvelocity.LT.0) then
                    bin=INT(-absvelocity/deltaint)+1
                    IF (bin.LE.seminter) then
                        veldist2(bin)=veldist2(bin)+1
                        ntot=ntot+1
                        IF (bin.lT.minbin) then
                                minbin=bin
                        ENDIF
                    ENDIF
                ENDIF
            ENDIF
            !ENDIF
            i=i+1
        END DO
        read(1,104) dim
    END DO
!   write(*,*) 'qui'

!***************************************************
    i=0
    minl=minimo
    IF (minimo.LT.0) then
        DO WHILE (i.LT.seminter)
            write (2,105) minl, veldist2(seminter-i)/ntot
            minl=minl+deltaint
            i=i+1
        END DO
        i=1
        DO WHILE (i.LE.seminter)
            write (2,105) minl, veldist1(i)/ntot
            minl=minl+deltaint
            i=i+1
        END DO
    ENDIF
!   write(*,*) 'qui2',minimo
    i=1
    IF (minimo.GE.0) then
        DO WHILE (i.LE.2*seminter)
            write (2,105) minl, veldist1(i)/ntot
            minl=minl+deltaint
            i=i+1
        END DO
    ENDIF
    CLOSE(2, STATUS='KEEP')
    Open ( UNIT=2, FILE= 'gammapunto.txt', STATUS='OLD')
!   pause
    i=1
    DO WHILE (.NOT.EOF(2))
        read(2,105) vel, pdf
        pdfvel(i)=vel*pdf
        i=i+1
    END DO
    meanvel=Sum(pdfvel)
    write(3,106) distlow, disthigh, meanvel
    write(*,*) distlow, disthigh
    distlow=disthigh
    disthigh=distlow+deltadist
    CLOSE(1, STATUS='KEEP')
    CLOSE(2, STATUS='KEEP')
END DO
```

```
write(*,*) 'FATTO!!!!'


!*****************************************************
!
!*****************************************************


100 format (i5,2a5,i5,3f8.3,3f8.4)
101 format (1a29,1f15.5)
102 format (i5)
104 format (9f10.5)
105 format (2f13.5)
106 format (3f13.5)
end program
```

Bibliography

[1] Avrami M., 1939, Kinetics of phase change 1: general theory, *J. Chem. Phys.*, 7, 1103-1112.

[2] Avrami M., 1940, Kinetics of phase change 2: transformation-time relations for random distribution of nuclei, *J. Chem. Phys.*, 8, 212-224.

[3] Narine S.S., A.G. Marangoni, 1999, Relating structure of fat crystal networks to mechanical properties: a review, *Food Res. Int.*, 32, 227–248.

[4] Phipps L.W., 1964, Heterogeneous and Homogeneous Nucleation in Supercooled Triglycerides and n-Paraffins,*Trans. Faraday Soc.*, 60, 1873–1883.

[5] Sethna J.P., 2006, Statistical Mechanics:Entropy, Order Parameters and Complexity, Oxford University Press.

[6] Foubert I., P.A. Vanrolleghem, B. Vanhoutte and K. Dewettinck, 2002, Dynamic mathematical model of the crystallization kinetics of fats, *Food Res. Int.*, 35, 945–956.

[7] D.J. Evans, G.P. Morriss, 1990, Statistical mechanics of non-equilibrium liquids, Academic press London.

[8] Rafii-Tabar H. and A. Chirazi, 2002, Multi-scale computational modelling of solidification phenomena, *Physics Reports*, 365, 145–249.

[9] A.A.V.V., 2002, Engineering and food for the 21th century, CRC press LLC.

[10] D. Frenkel, B. Smit, 2002, Understanding molecular simulations from algorithms to applications, Academic press.

[11] Berendsen H. J.C., 2004, Lectures notes on molecular dynamics, University of Groningen, Biophysical Chemestry, The Netherlands Groningen Institute for Biosciences and Biotechnology.

[12] Verdier P.H., D.E. Kranbuehl, 1987, Simulation of Polymer Chain Dynamics by lattice models with excluded volume: Lattice dependence, *Macromol.*, 20, 1362–1368.

[13] Liu T. W., 1989,Flexible polymer chain dynamics and rheological properties in steady flows, *J. Chem. Phys.*, 90, 10.

[14] Meyer H. and F. Müller-Plathe, 2002, Formation of chain-folded structures in supercooled polymer melts examined by MD simulations, *Macromol.*, 35, 1241–1252.

[15] M. Doi, S.F. Edwards, 1994, The theory of polymer dynamics, Clarendon press Oxford.

[16] Rapaport D.C., 1998, The Art of molecular Dynamics Simulation, Cambridge University Press.

[17] Allen, Tildesley,1987, Computer Simulation of Liquids, Clarendon Press – Oxford

[18] P. M. Morse, 1929, Diatomic molecules according to the wave mechanics. II. Vibrational levels. *Phys. Rev.*, 34, 57–64.

[19] van Gunsteren, W.F., S.R. Billeter, A.A. Eising, P.H. Hünenberger, P. Krüger, A.E. Mark, W.R.P. Scott,I. G. Tironi, 1996, Biomolecular Simulation: The GROMOS96 manual and user guide. Zürich, Switzerland: Hochschulverlag AG an der ETH Zürich.

[20] Neelov I.M., D.B. Adolf, 2003, Brownian Dynamics Simulations of Dendrimers under Elongational Flow: Bead-Rod Model with Hydrodynamic Interactions, *Macromol.*, 36, 6914–6924.

[21] D. van der Spoel, E. Lindahl, B. Hess, A. R. van Buuren, E. Apol, P. J. Meulenhoff, D. P. Tieleman, A. L. T. M. Sijbers, K. A. Feenstra, R. van Drunen and H. J. C. Berendsen, 2005, Gromacs User Manual version 3.3, www.gromacs.org.

[22] Jorgensen W L and J Tirado-Rives, 1988, The OPLS Potential Functions for Proteins - Energy Minimiza–Minimizations for Crystals of Cyclic-Peptides and Crambin, *Journal of the American Chemical Society*, 110, 1666–1671.

[23] De Loof H., S.C. Harvey, J.P. Segret, R.W. Pastor, Mean Field Stochastic Boundary Molecular Dynamics Simulation of a Phospholipid in a Membrane, *Biochem.*, 30, 2099–2113.

[24] Cox S R and D E Williams, 1981, Representation of the Molecular Electrostatic Potential by a New Atomic Charge Model, *J. of Comput. Chem.*, 2, 304–323.

[25] Lifson S. and A. Warshel, 1968, Consistent Force Field for Calculations of Conformations, Vibrational Spectra and Enthalpies of Cycloalkane and M-Alkane Molecules, *J. of Chem. Phys.*, 49, 5116–5129.

[26] Rappe A.K., C.J. Casewit, K.S. Colwell, W.A. Goddard III and W.M. Skiff, 1992, UFF, a Full Periodic Table Force Field for Molecular Mechanics and Molecular Dynamics Simulations, emphJ. Am. Chem. Soc., 114, 10024–10035.

[27] Adams D.J., 1983, Alternatives to the Periodic Cube in Computer Simulation. CCP5 Quarterly, 10,30–36.

[28] Verlet L., 1967, Computer Experiments on Classical Fluids. II. Equilibrium Correlation Functions, *Phys. Rev.*, 165, 201–204.

[29] Verlet L., 1967, Computer Experiments on Classical Fluids. I Thermodynamical Properties of Lennard–Jones Molecules*Physical Rev.*, 159, 98–103.

[30] Hockney R.W., 1970, The Potential Calculation and Some Applications, *Meth. Comp. Phys.*, 9, 136–211.

[31] Ryckaert J.P., G. Cicotti and H.J.C. Berensden, 1977, Numerical Integration of the Cartesian Equations of Motion of a System with Constraints Molecular Dynamics of n-Alkanes, *J. of Comp. Phys.*, 23, 327–341.

[32] Woodcock L.V., 1971, Isothermal Molecular Dynamics Calculations for Liquid Salts, *Chem. Phys. Lett.*, 10, 257–261.

[33] Berendsen H.J.C., J.P.M. Postma, W.F. van Gunsteren, A. Di Nola and J.R. Hddk, 1984, Molecular Dynamics with Coupling to an External Bath, *J. Chem. Phys.*, 81, 3684–3690.

[34] Nosè S., 1984, A Molecular Dynamics Method for Simulations in the Canonical Ensemble., *Molecular Physics*, 53, 255–268.

[35] Hoover W. G., 1985, Canonical Dynamics: equilibrium phase-space distributions, *Phys.Rev. A*, 31, 1695–1697.

[36] Leach A.R., 2001, Molecular Modelling Principle and Applications, Prentice Hall

[37] Marrink S.J.,A.H. de Vries and A.E. Mark, 2004, Coarse Grained Model for Semiquantitative Lipid Simulations, *J. Phys. Chem. B*, 108, 750–760.

[38] Marrink S.J. and A.E. Mark, 2004, Molecular View of Hexagonal Phase Formation in Phospholipid Membranes, *Biophys. J.*, 87, 3894–3900.

[39] Chandrasekhar I. and W.F. van Gusteren, 2002, A comparison of the potential energy parameters of aliphatic alkanes: Molecular dynamics simulations of triacylglycerols in the alpha phase, *Eur. Biophys. J.*, 31, 89–101.

[40] Chandrasekhar I., M. Kastenholts, R.D. Lins, C. Oostenbrink, L.D. Schuler, D.P. Tieleman, W.F. van Gunsteren, A consistent potential energy parameter set for lipids: dipalmitoylphosphatidylcholine as a benchmark of the GROMOS96 45A3 force field,*Eur. Biophys. J.*, 30, 67–77.

[41] Schuler L.D., P. Walde, P.L. Luisi, W.F. van Gusteren, 2001, Molecular dynamics simualtion of n-dodecyl phosphate aggregate structures, *Eur. Biophys. J.*, 30, 330-343.

[42] Marangoni A.G., T.C. Aurand, S. Martini and M. Ollivon, 2006, A Probabilistic Approach to Model the Nonisothermal Nucleation of Triacylglycerol Melts, *Cryst. Growth & Design*, 6, 5,1199-1205.

[43] Marrink S.J. and A.E. Mark, Molecular View of Hexagonal Phase Formation in Phospholipid Membranes, *Biophys. J.*, 87, 3894–3900.

[44] Beckett S.T., 1999, Industrial Chocolate Manufacture and Use, Blackwell Science.

[45] Loisel C., G. Lecq, G. Ponchel, G. Keller, M. Ollivon, 1997, Fat bloom and Chocolate Structure studied by Mercury Porosimetry, J. Food Science,62, 4, 781–788.

[46] Franke K., 1998, Modelling of Cooling Kinetics of Chocolate Coatings with Respect of Final Product Quality, *J. Food Eng.*, 36, 371–384.

[47] Bollinger S., B. Breitschuh, M. Stranzinger, T. Wagner, E. J. Windhab, 1998, Comparison of Precrystallization of Chocolate, 35, 281–297.

[48] Schenk H., R. Peschar, 2004, Understanding the structure of chocolate, *Radiation physics and chemistry*, 71, 829–835.

[49] Lonchampt P., R.W. Hartel, 2004, Fat bloom in chocolate and compound coatings, *Eur. J. Lipid Sci. Technol*, 106, 241–274.

[50] Hodge S.M., D. Rousseau, 2002, Fat bloom formation and characterizationin milk chocolate observed by atomic force microscopy, *J. Am. Oil Chemists' Society*, 79, 1115–1121.

[51] Tietz R.A. and R.W. Hartel, 2000, Effect of minor lipids on crystallization of milk fat–cocoa butter blends and bloom formation in chocolate, *J. Am. Oil Chemists' Society*, 77, 7, 763–771.

[52] Lipp M. and E. Anklam, 1998, Review of cocoa butter and alternative fats for use in chocolate–Part A. Compositional data, *Food Chemistry*, 62, 1, 73–97.

[53] Lipp M. and E. Anklam, 1998, Review of cocoa butter and alternative fats for use in chocolate–Part B. Analytical approaches for identification and determination, *Food Chemistry*, 62, 1, 99–108.

[54] Chaiseri S. and P. S. Dimick, 1995, Dynamic Crystallization of Cocoa Butter. I. Characterization of Simple Lipids in Rapid– and Slow–Nucleating Cocoa Butters and their Seed Crystals, *J. Am. Oil Chemists' Society*, 72, 1491–1496.

[55] Chaiseri S. and P. S. Dimick, 1995, Dynamic Crystallization of Cocoa Butter. II. Morphological, Thermaland Chemical Charactiristics during Crystal Growth, *J. Am. Oil Chemists' Society*, 72, 1497–1504.

[56] van Malssen K., R. Peschar and H. Schenk, 1996, Real–Time X–Ray Powder Diffraction Investigations on Cocoa Butter. II. The Relationship Between Melting Behavior and Composition of β-Cocoa Butter, *J. Am. Oil Chemists' Society*, 73, 10, 1217–1223.

[57] Sato K., 1993, Polymorphic Transformations in Crystal Growth, *J. Phys. D: Appl. Phys.*, 26, 77–84.

[58] Yano J. and K. Sato, 1999, FT–IR Studies on Polymorphism of Fats: Molecular Structures and Interactions, *Food Res. Int.*, 32, 249–259.

[59] Sato K., T. Arishima, Z. H. Wang, K. Ojima, N. Sagi and H. Mori, 1989, Polymorphism of POP and SOS. I. Occurrence and Polymorphic Transformation, *J. Am. Oil Chemists' Society*, 66, 5, 664–674.

[60] Koyano T., I. Hachiya and K. Sato, 1992, Phase Behavior of Mixed Systems of SOS and OSO, *J. Phys. Chem.*, 96, 10514–10520.

[61] Wille R. L. and E. S. Lutton, 1966, Polymorphism of Cocoa Butter,*J. Am. Oil Chemists' Society*, 43, 491–496.

[62] Hartel R.W., 2001, Crystallization in foods, Aspen Publishers, Inc.

[63] Culot C., F. Durant, D.H. Mosley, J.M. Andre, D.P. Vercauteren, 1994, Conformation Analyses of the polymorphism of triglycerides, *Int. J. Of Quantum Chem.: Quantum Biology Symposium*, 21, 57–77.

[64] Stapley A. G. F., H. Tewkesbury and P. J. Fryer, 1999, The Effect of Shear and Temperature History on the Crystallization of Chocolate, *J. Am. Oil Chemists' Society*, 76, 6, 677–685.

[65] Chen X., P. Carbone, W.L. Cavalcanti, G. Milano and F. Müller–Plathe, 2007, Viscosity and Structural Alteration of a Coarse-Grained Model of Polystyrene under Steady Shear Flow Studied by Reverse Nonequilibrium Molecular Dynamics, *Macromol.*, 40, 22, 8087–8095.

[66] Ercolessi F., 1997, A molecular dynamics primer, Università di Udine, Notes of the Spring College in Computational Physics, ICTP, Trieste. www.fisica.uniud.it/ ercolessi/.

[67] Goodrum, Eiteman, 1996, Physical properties of low molecular weight triglycerides for the development of bio-diesel fuel models, *Biores. Tech.*, 56, 55–60.

[68] Herrmann S., G. Winter, S. Mohl, F. Siepmann and J. Siepmann, 2007, Mechanism controlling protein release from lipidic implants: Effects of PEG addition, *J. Controlled Release*, 118, 161–168.

[69] Kishore K. and H.K. Shobha, 1992, Thermodynamics of flow and Vaporization processes in long-chain liquids, *J. Phys. Chem.*, 96, 8161–8168.

[70] B.Hess, Determining the shear viscosity of model liquids from molecular dynamics simulations, accepted for publication in *J. Chem. Phys.*

[71] Landau L.D. and E.M. Lifšits, 2000, Mechanics, third edition, Butterworth Heinenann.

[72] Bertolt Brecht, 1963, Vita di Galileo, Giulio Einaudi Editore.

Informazioni personali

Cognome/i nome/i	**Antonio Brasiello**
Indirizzo/i	Via G. Limone, 44 81030 Sant'Arpino (CE)
Telefono/i	tel +39–0815013739 mobile **+39–3283090967**
Email	abrasiel@unina.it
Nazionalità	Italiana
Data di nascita	18 Marzo 1977

Istruzione

2004	Università degli Studi di Napoli Federico II Laurea (quinquennale) in Ingegneria Chimica.Votazione: 110/110. Titolo della tesi: Dinamica e controllo di reattori chimici a flusso invertito: comportamento ibrido.(Relatore Prof. S. Crescitelli)
1996	Liceo Scientifico Statale Filippo Silvestri Portici (NA) Diploma di maturità scientifica. Votazione: 53/60. Corso di studi sperimentale di informatica.

Corsi e scuole

Corsi e scuole certificati

2007	Iscritto al terzo anno del corso di Dottorato di Ricerca in Ingegneria Chimica- XX ciclo, Università degli Studi di Napoli Federico II.
2007	Corso (36 ore) *Meccanica Quantistica*, Prof. U. Esposito, Scuola di Dottorato in Ingegneria Industriale, Facoltà di Ingegneria, Università di Napoli Federico II.
2006	Scuola Estiva *Ottimizzazione di Processo nell'Ingegneria Chimica*, 18-24 Giugno 2006, Hotel Alpe, Alba di Canazei, Trento.
2006	European Society of Rheology Short Course *Multi-scale Modelling Methodologies*, 25-26 Aprile 2006, Hersonissos, Creta, Grecia.
2006	Corso (18 ore) *Dinamiche non lineari*, Proff. S. Crescitelli, M. di Bernardo, C. Serpico, Scuola di Dottorato in Ingegneria Industriale, Facoltà di Ingegneria, Università di Napoli Federico II.

2006 Corso *Thermomechanics of Soft Matter*, Prof. G. Marrucci, Scuola di Dottorato in Ingegneria Industriale, Facoltà di Ingegneria, Università di Napoli Federico II.

2006 Workshop Internazionale *Matematica nei Materiali*, Istituto Nazionale di Alta Matematica, 3-7 Aprile 2006, Città Universitaria, Roma.

2006 Corso di *Reologia: elementi di base*, Proff. Nino Grizzuti e Pier Luca Maffettone, tenuto presso il Dipartimento di Ingegneria Chimica, Università degli Studi di Napoli Federico II.

2006 Corso *Tecnologie Grid e Loro Utilizzo in Ambito Scientifico*, 10 Gennaio - 9 Marzo 2006, Istituto Nazionale di Fisica Nucleare, Complesso Universitario Monte Sant'Angelo, Napoli.

2005 Scuola Estiva *Controllo nell'Industria di Processo*,16-26 Giugno, Is Morus, Santa Margherita di Pula, Cagliari.

2005 Scuola GRICU *Metodi Matematici per l'Ingegneria Chimica*, 4-8 Aprile, Facoltà di Ingegneria Università di Bologna, Bologna.

2004 Workshop on Bifurcations in Nonsmooth and Hybrid Dynamical Systems: Analysis, Control and Applications. Società Italiana Caos e Complessità in cooperazione con la Fondazione Eni Enrico Mattei (FEEM), 21-22 Ottobre, Palazzo delle Stelline, Milano.

2004 *ADP2004-Strumenti e Metodi dell'Analisi Dinamica dei Processi*, XIV Scuola di Chimica Computazionale, Società Chimica Italiana, 27 Giugno - 3 Luglio, Hotel Continental Terme, Ischia (NA).

Corsi e scuole non certificati

2006 Corso di Analisi Funzionale: complementi del Prof. Renato Fiorenza.

2005 Corso di Analisi Funzionale: fondamenti del Prof. Renato Fiorenza.

Partecipazione a convegni su invito

2007 Mandelbrot e la geometria frattale quaranta anni dopo, Università degli Studi di Messina, Messina, 21-24 Marzo.

Partecipazione a congressi

2006 3rd Annual European Rheology Conference - AERC 2006 Hersonissos, Crete, Greece, 17-19 Aprile.

2005 ICHEAP–7 The seventh Italian Conference on Chemical & Process Engineering, Giardini Naxos, Italia, 15-18 Maggio.

2004 International Workshop on Bifurcations in Nonsmooth and Hybrid Dynamical Systems, Milan, Italy, 21-22 Ottobre.

Conoscenze informatiche

Sistemi operativi: MsDos, Windows 95-98-NT,Windows XP, Linux Redhat, Unix IRIX 6.4 SGI, Unix TRU 64.

Pacchetti applicativi: Mathematica, Matlab, Mathcad, Office (Word, Excel, etc.), LaTeX, Sigma Plot e Autocad.

Software avanzati per lo studio della dinamica dei sistemi: AUTO86, AUTO97, TISEAN.

Linguaggi di programmazione: FORTRAN77, FORTRAN90, C++.

Software per lo studio della dinamica molecolare: GROMACS, VMD.

Associazioni

2004–2008 Membro della Società Italiana Caos e Complessità (SICC) c/o DEI-Politecnico di Milano Via Ponzio 34/5 20133 Milano.

Impiego attuale

Luglio 2007–Ottobre 2007 Contratto di collaborazione ad attività di ricerca finalizzata a: *Analisi e studio di materiali sensibili all'umidità e alla pioggia per applicazioni nel campo del monitoraggio del traffico stradale.*, presso l'Università degli Studi di Salerno. Responsabile scientifico: Prof. Antonio Pietrosanto.

Esperienze professionali

Luglio 2007 Vincitore del concorso *IMAST scarl* per titoli, test e colloquio nell'ambito del progetto di alta formazione e work on the job finanziato dalla Regione Campania, programma Operativo Regionale 2000–2006, finalizzato al rafforzamento ed alla qualificazione del capitale umano nel settore *Ricercatore nell'ingegneria dei materiali polimerici e compositi.*

.

Marzo 2006–Giugno 2007 Contratto di collaborazione coordinata e continuativa: *Previsione della shelf-life di Noci*, presso l'Università degli Studi di Salerno. Responsabile scientifico: Prof. Marisa Di Matteo.

Dicembre 2006–Gennaio 2007 Contratto di collaborazione scientifica: *Messa a punto di tecniche agronomiche e sistemi colturali innovativi basati sull'impiego di contenitori biodegradabili e/o compostabili. Modellazione dell'attacco fungino*, presso l'Università degli Studi di Salerno. Responsabile scientifico: Prof. Marisa Di Matteo.

Settembre 2005–Novembre 2006

Borsa di studio *Formazione di esperti nella gestione di sistemi colturali innovativi e nell'utilizzo di materiali biodegradabili: Esperti in chimica applicata ai biomateriali* , P.O.N. 2000-2006, presso l'Università degli Studi di Salerno.

Dicembre 2004–Febbraio 2005

Contratto di collaborazione scientifica per il supporto all'attività di ricerca in processi innovativi di conservazione e di previsione della shelf-life, presso l'Università degli Studi di Salerno. Responsabile scientifico: Prof.ssa Marisa Di Matteo.

Luglio 2004–Settembre 2004

Contratto di collaborazione scientifica *Previsione delle variazioni dei principali indici di qualità durante la conservazione delle pesche* presso il Dipartimento d'Ingegneria Chimica e Alimentare dell'Università degli Studi di Salerno. Responsabile scientifico: Prof.ssa Marisa Di Matteo.

Maggio 2004–Luglio 2004

Contratto di collaborazione scientifica *Messa a punto di un programma per lo studio della dinamica dei sistemi ibridi* presso il Dipartimento d'Ingegneria Chimica dell'Università degli Studi di Napoli Federico II. Responsabile scientifico: Prof. Silvestro Crescitelli.

Attività didattica

Settembre 2007

Lezioni tenute nell'ambito della XVII Scuola di Chimica Computazionale, Società Chimica Italiana: *Strumenti e Metodi dell'Analisi Dinamica dei Processi 2007*, Complesso di Sant'Agostino, Università del Sannio, Benevento:
MATLAB/PPlane: orbite e diagrammi di fase di modelli di processi chimici/biologici (colonne di assorbimento, modelli di frammentazione, bilanci di popolazione, dinamica di crescita di linee cellulari) (2 ore);
MATCONT: diagrammi delle soluzioni di regime di processi chimici (1 ora);
MATCONT: studio delle biforcazioni di processi chimici (3 ore).

Settembre 2005–Gennaio 2006

Esercitazioni del corso di *Metodi di Analisi dei Dati Sperimentali*, Corso di Laurea in Ingegneria Chimica, Università degli Studi di Napoli Federico II, A.A. 2005/2006.

Marzo 2005–Giugno 2005

Esercitazioni del corso di *Analisi e Simulazione dei Processi Chimici*, Corso di Laurea in Ingegneria Chimica, Università degli Studi di Napoli Federico II, A.A. 2004/2005.

Settembre 2004–Gennaio 2005

Esercitazioni del corso di *Teoria dello Sviluppo dei Processi Chimici*, Corso di Laurea in Ingegneria Chimica, Università degli Studi di Napoli Federico II, A.A. 2004/2005.

Giugno 2004–Luglio 2004 Esercitazioni su *Applicazioni dei metodi di continuazione per lo studio della dinamica di reattori chimici* tenute nell'ambito della XIV Scuola di Chimica Computazionale, Società Chimica Italiana: *Strumenti e Metodi dell'Analisi Dinamica dei Processi 2004*, Hotel Continental Terme, Ischia (NA).

Attività volontarie

2001–2003 Rappresentante degli studenti al Consiglio di Corso di Laurea in Ingegneria Chimica e membro del Consiglio degli Studenti di facoltà.

2001–2003 Membro della commissione paritetica docenti-studenti per il rinnovo del piano di studi del Corso di Laurea in Ingegneria Chimica D.M. 509/99.

Pubblicazioni

P. internazionali giudicate da commissioni internazionali di tre membri

2007 E. Mancusi, L. Russo, A. Brasiello, M. di Bernardo and S. Crescitelli, *Hybrid Modelling and Dynamics of a Controlled Reverse Flow Reactor*, AIChE Journal, 53, (8), 2084-2096.

2007 D. Albanese, L. Russo, L. Cinquanta, A. Brasiello, and M. Di Matteo, *Physical and Chemical Changes in Minimally Processed Green Asparagus during Cold-Storage*, Food Chemestry, 101, (1), 274–280.

2005 A. Brasiello, L. Russo, E. Mancusi, M. di Bernardo and S. Crescitelli, *Nonlinear Dynamics of a Controlled Reverse Flow Reactor*, paper number 04925,Proceedings of 16th IFAC World Congress Volume 16 part 1, Elsevier, ISBN: 0-08-045108-X.

P. su atti di congressi internazionali con referaggio

2007 Albanese D., Russo L., Brasiello A., Crescitelli S., Di Matteo M., *Model of Bacterial Growth for the Prediction of Seabream (Sparus Aurata) Shelf-Life in Active Packaging*, Proceedings of CIGR 3rd International Symposium, Napoli, 24-26 settembre.

2006 D. Albanese, L. Cinquanta, L. Russo, S. Crescitelli, M. Farina, A. Brasiello and M. Di Matteo, *Modelling Convective and Microwave Drying of Potatoes Slices*, Proceedings of IUFoST 13th World Congress on Food Science and Technology, Nantes, France,17-21 Settembre. DOI:10.1051/IUFoST:20060901

2005 A. Brasiello, L. Russo, E. Mancusi, M.di Bernardo and S. Crescitelli, *Bifurcations of Periodic Regimes in a Controlled Reverse Flow Reactor*, Proceedings of ICHEAP–7 The seventh Italian Conference on Chemical & Process Engineering,Giardini Naxos, Italia, 15-18 Maggio. ISBN:88–900775–7–3

Comunicazioni a convegni

2007 S. Crescitelli, A. Brasiello, L.Russo, *Regimi strani e struttura frattale nell'industria di processo*,Mandelbrot e la geometria frattale quaranta anni dopo, Università degli Studi di Messina, Messina, 21-24 Marzo.

2004 A. Brasiello, E.Mancusi, L. Russo and S. Crescitelli, *Complex Dynamic of a Zeno Hybrid Model of Controlled Reverse Flow Reactors*, Proceedings of International Conference and Summer School of the Centre for Research and Applications of Nonlinear Systems: Complexity in Science and Society, Ancient Olympia, Grecia, 14-26 Luglio.

2004 A. Brasiello, L. Russo, E. Mancusi and S. Crescitelli, *Hybrid system approach to study the dynamics of a controlled reverse flow reactor II. Continuation of limit cycles and bifurcations*, 2nd International workshop of the Bristol Centre for Applied Nonlinear Mathematics, Piecewise Smooth Dynamical Systems: analysis, numerics and applications, Bristol, UK, 13-16 Settembre.

Altre pubblicazioni

Tesi di laurea: Dinamica e controllo di reattori chimici a flusso invertito: comportamento ibrido.

Contributi vari

2004 Citato nei ringraziamenti in E. Mancusi, L. Russo, M. di Bernardo, S. Crescitelli, *Zeno Trajectories in a Non-Smooth Model of a Reverse Flow Reactor*, Proceedings of ESCAPE 14: European Symposium on Computer Aided Process Engineering Lisbona, Portogallo, Maggio 16-19.

Lingue straniere

Ottima conoscenza della lingua inglese

Servizio militare

Dispensato per meriti scientifici (articolo 9, comma 2-bis, lettera b della legge 230/98).