WP12: Theoretische Methoden zur Untersuchung kondensierter Systeme

Praktikumsteil AK Kirchner

WS 2015/2016

Contents

1 Introduction to Molecular Dynamics Simulation

The following introduction is a shortened version of Michael P. Allen's lecture script on Molecular Dynamics (MD) simulation,[1] which itself is an excerpt from one of the standard textbooks on MD simulation.[2] This document is intended to give you a working knowledge of MD simulation. You will learn more about the theory behind MD in the lecture that accompanies this practical exercise and you are welcome to delve further into the literature on your own.

1.1 The Aims of Molecular Dynamics

We carry out computer simulations in the hope of understanding the properties of assemblies of molecules in terms of their structure and the microscopic interactions between them. This serves as a complement to conventional experiments, enabling us to learn something new, something that cannot be found out in other ways. The two main families of simulation technique are molecular dynamics (MD) and Monte Carlo (MC); additionally, there is a whole range of hybrid techniques which combine features from both. In this lecture we shall concentrate on MD. The obvious advantage of MD over MC is that it gives a route to dynamical properties of the system: transport coefficients, time-dependent responses to perturbations, rheological properties and spectra.

Computer simulations act as a bridge (see Fig. 1.1 on the following page) between microscopic length and time scales and the macroscopic world of the laboratory: we provide a guess at the interactions between molecules, and obtain "exact" predictions of bulk properties. The predictions are "exact" in the sense that they can be made as accurate as we like, subject to the limitations imposed by our computer budget. At the same time, the hidden detail behind bulk measurements can be revealed. An example is the link between the diffusion coefficient and velocity autocorrelation function (the former easy to measure experimentally, the latter much harder). Simulations act as a bridge in another sense: between theory and experiment. We may test a theory by conducting a simulation using the same model. We may test the model by comparing with experimental results. We may also carry out simulations on the computer that are difficult or impossible in the laboratory (for example, working at extremes of temperature or pressure).

Ultimately we may want to make direct comparisons with experimental measurements made on specific materials, in which case a good model of molecular interactions is essential. The aim of so-called *ab initio* molecular dynamics is to reduce the amount of fitting and guesswork in this process to a minimum. On the other hand, we may be interested in phenomena of a rather generic nature, or we may simply want to discriminate between good and bad theories. When it comes to aims of this kind,

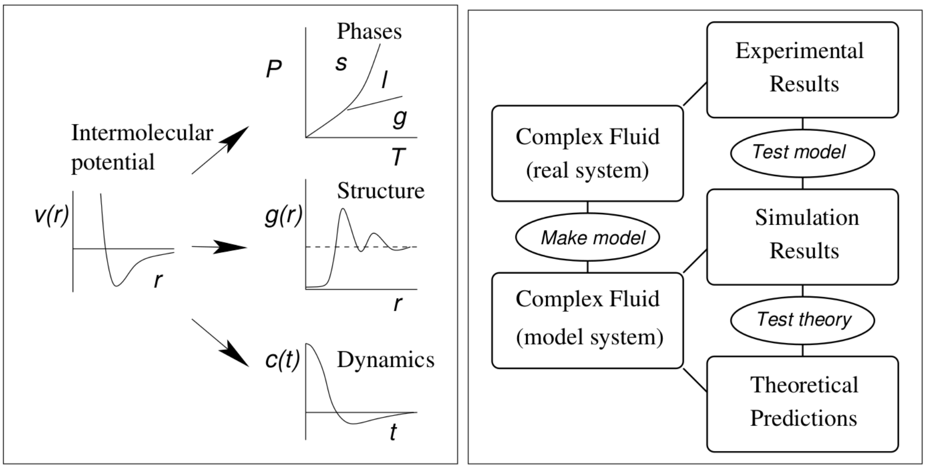


Figure 1.1: Simulations as a bridge between (a) microscopic and macroscopic; (b) theory and experiment.

it is not necessary to have a perfectly realistic molecular model; one that contains the essential physics may be quite suitable.

1.2 Molecular Interactions

Molecular dynamics simulation consists of the numerical, step-by-step, solution of the classical equations of motion, which for a simple atomic system may be written

$$m_i\ddot{\mathbf{r}}_i = \mathbf{f}_i \quad \mathbf{f}_i = \frac{\partial}{\partial \mathbf{r}_i} U \,. \tag{1.1}$$

For this purpose we need to be able to calculate the forces $\mathbf{f}_i$ acting on the atoms, and these are usually derived from a potential energy $U\left(\mathbf{r}^N\right)$, where $\mathbf{r}^N = \left(\mathbf{r}_1, \mathbf{r}_2, \ldots \mathbf{r}_N\right)$ represents the complete set of $3N$ atomic coordinates. In this section we focus on this function $U\left(\mathbf{r}^N\right)$, restricting ourselves to an atomic description for simplicity.

1.2.1 Non-bonded Interactions

The part of the potential energy $U_{\text{non-bonded}}$ representing non-bonded interactions between atoms is traditionally split into 1-body, 2-body, 3-body … terms:

$$U_{\text{non-bonded}}\left(\mathbf{r}^N\right) = \sum_i u\left(\mathbf{r}_i\right) + \sum_i \sum_{j>i} v\left(\mathbf{r}_i, \mathbf{r}_j\right) + \ldots \,. \tag{1.2}$$

The $u\left(\mathbf{r}\right)$ term represents an externally applied potential field or the effects of the container walls; it is usually dropped for fully periodic simulations of bulk systems. Also, it is usual to concentrate on the pair potential $v\left(\mathbf{r}_i, \mathbf{r}_j\right) = v\left(r_{ij}\right)$ and neglect three-body (and higher order) interactions.

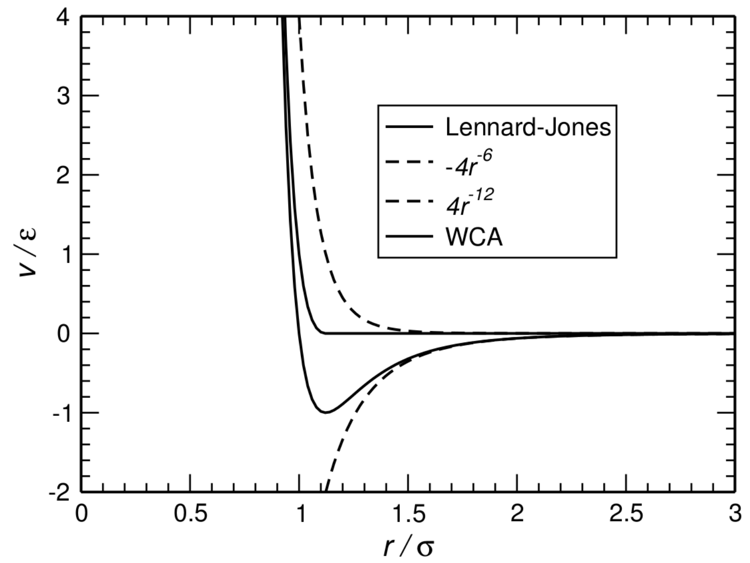


Figure 1.2: Lennard–Jones pair potential showing the r^{-12} and r^{-6} contributions. Also shown is the WCA shifted repulsive part of the potential.

In some simulations of complex fluids, it is sufficient to use the simplest models that faithfully represent the essential physics. In this chapter we shall concentrate on continuous, differentiable pair-potentials. The Lennard–Jones potential is the most commonly used form:

$$v^{\mathrm{LJ}}(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right], \tag{1.3}$$

with two parameters: σ, the diameter, and ε, the well depth. This potential was used, for instance, in the earliest studies of the properties of liquid argon and is illustrated in Fig. 1.2. If electrostatic charges are present, we add the appropriate Coulomb potentials

$$v^{\mathrm{Coulomb}}(r) = \frac{Q_1 Q_2}{4\pi\epsilon_0 r}, \tag{1.4}$$

where Q_1, Q_2 are the charges and ϵ_0 is the permittivity of free space.

1.2.2 Bonding Potentials

For molecular systems, we simply build the molecules out of site-site potentials of the form of Eq. 1.3 or similar. Typically, a single-molecule quantum-chemical calculation may be used to estimate the electron density throughout the molecule, which may then be modelled by a distribution of partial charges via Eq. 1.4, or more accurately by a distribution of electrostatic multipoles. For molecules we must also consider the intramolecular bonding interactions. The simplest molecular model will include terms

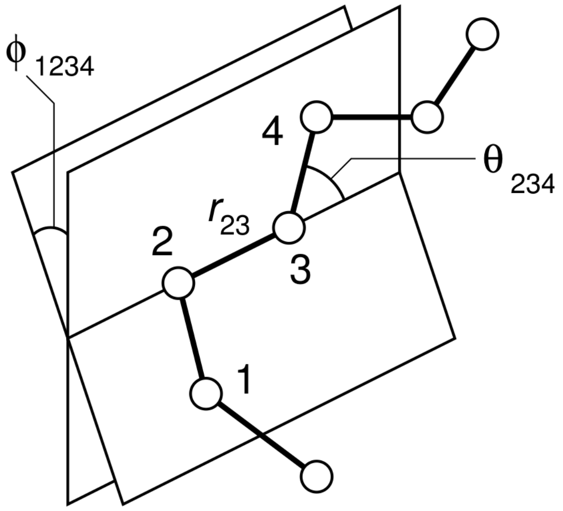


Figure 1.3: Geometry of a simple chain molecule, illustrating the definition of interatomic distance r_{23}, bend angle θ_{234}, and torsion angle ϕ_{1234}.

of the following kind:

$$U_{\text{intramolecular}} = \frac{1}{2}\sum_{\text{bonds}} k^r_{ij}\left(r_{ij} - r_{\text{eq}}\right)^2 \tag{1.5}$$

$$+\frac{1}{2}\sum_{\text{bend angles}} k^{\theta}_{ijk}\left(\theta_{ijk} - \theta_{\text{eq}}\right)^2 \tag{1.6}$$

$$+\frac{1}{2}\sum_{\text{torsion angles}}\sum_{m} k^{\phi,m}_{ijkl}\left(1+\cos\left(m\phi_{ijkl} - \gamma_m\right)\right) . \tag{1.7}$$

The geometry is illustrated in Fig. 1.3. The "bonds" will typically involve the separation $r_{ij} = \left|\mathbf{r}_i - \mathbf{r}_j\right|$ between adjacent pairs of atoms in a molecular framework, and we assume in Eq. 1.5 a harmonic form with specified equilibrium separation, although this is not the only possibility. The "bend angles" θ_{ijk} are between successive bond vectors such as $\mathbf{r}_i - \mathbf{r}_j$ and $\mathbf{r}_j - \mathbf{r}_k$, and therefore involve three atom coordinates:

$$\cos\theta_{ijk} = \hat{\mathbf{r}}_{ij}\cdot\hat{\mathbf{r}}_{jk} , \tag{1.8}$$

where $\hat{\mathbf{r}} = \mathbf{r}/r$. Usually this bending term is taken to be quadratic in the angular displacement from the equilibrium value, as in Eq. 1.6, although periodic functions are also used. The "torsion angles" ϕ_{ijkl} are defined in terms of three connected bonds, hence four atomic coordinates:

$$\cos\phi_{ijkl} = -\hat{\mathbf{n}}_{ijk}\cdot\hat{\mathbf{n}}_{jkl} , \tag{1.9}$$

where $\mathbf{n}_{ijk} = \mathbf{r}_{ij}\times\mathbf{r}_{jk}$, $\mathbf{n}_{jkl} = \mathbf{r}_{jk}\times\mathbf{r}_{kl}$, and $\hat{\mathbf{n}} = \mathbf{n}/n$, the unit normal to the plane defined by each pair of bonds. Usually the torsional potential involves an expansion in periodic functions of order $m = 1, 2, \ldots$, Eq. 1.7.

A force-field will specify the precise form of Eqns. 1.5-1.7, and the various strength parameters k and other constants therein. Actually, these equations are a considerable oversimplification. Molecular mechanics force-fields, aimed at accurately predicting structures and properties, will include many cross-terms (e.g. stretch-bend). Quantum mechanical calculations may give a guide to the "best" molecular force-field; also comparison of simulation results with thermophysical properties and vibration frequencies is invaluable in force-field development and refinement.

1.3 The MD Algorithm

Solving Newton's equations of motion does not immediately suggest activity at the cutting edge of research. The molecular dynamics algorithm in most common use today may even have been known to Newton. Nonetheless, the last decade has seen a rapid development in our understanding of numerical algorithms.

Continuing to discuss, for simplicity, a system composed of atoms with coordinates $\mathbf{r}^N = \left(\mathbf{r}_1, \mathbf{r}_2, \ldots \mathbf{r}^N\right)$ and potential energy $U\left(\mathbf{r}^N\right)$, we introduce the atomic momenta $\mathbf{p}^N = \left(\mathbf{p}_1, \mathbf{p}_2, \ldots \mathbf{p}^N\right)$, in terms of which the kinetic energy may be written $K\left(\mathbf{p}^N\right) = \sum_{i=1}^{N} |\mathbf{p}_i|^2 \Big/ 2m_i$. Then the energy, or Hamiltonian, may be written as a sum of kinetic and potential terms $H = K + U$. Write the classical equations of motion as

$$\dot{\mathbf{r}}_i = \mathbf{p}_i / m_i \quad \text{and} \quad \dot{\mathbf{p}}_i = \mathbf{f}_i \,. \tag{1.10}$$

This is a system of coupled ordinary differential equations. Many methods exist to perform step-by-step numerical integration of them. Characteristics of these equations are: (a) they are "stiff", i.e. there may be short and long timescales, and the algorithm must cope with both; (b) calculating the forces is expensive, typically involving a sum over pairs of atoms, and should be performed as infrequently as possible. Also we must bear in mind that the advancement of the coordinates fulfills two functions: (i) accurate calculation of dynamical properties, especially over times as long as typical correlation times τ_a of properties a of interest (we shall define this later); (ii) accurately staying on the constant-energy hypersurface, for much longer times $\tau_{\text{run}} \gg \tau_a$, in order to sample the correct ensemble.

To ensure rapid sampling of phase space, we wish to make the timestep as large as possible consistent with these requirements. For these reasons, simulation algorithms have tended to be of low order (i.e. they do not involve storing high derivatives of positions, velocities etc.): this allows the timestep to be increased as much as possible without jeopardizing energy conservation. It is unrealistic to expect the numerical method to accurately follow the true trajectory for very long times τ_{run}. The "ergodic" and "mixing" properties of classical trajectories, i.e. the fact that nearby trajectories diverge from each other exponentially quickly, make this impossible to achieve. All these observations tend to favor the Verlet algorithm in one form or another, and we look closely at this in the following section.

1.3.1 The Verlet Algorithm

There are various, essentially equivalent, versions of the Verlet algorithm, including the original method, and a "leapfrog" form. Here we concentrate on the "velocity Verlet" algorithm, which may be written

$$\mathbf{p}_i\left(t+\tfrac{1}{2}\delta t\right) = \mathbf{p}_i\left(t\right) + \tfrac{1}{2}\delta t\mathbf{f}_i\left(t\right) \tag{1.11}$$

$$\mathbf{r}_i\left(t+\delta t\right) = \mathbf{r}_i\left(t\right) + \delta t\mathbf{p}_i\left(t+\tfrac{1}{2}\delta t\right)/m_i \tag{1.12}$$

$$\mathbf{p}_i\left(t+\delta t\right) = \mathbf{p}_i\left(t+\tfrac{1}{2}\delta t\right) + \tfrac{1}{2}\delta t\mathbf{f}_i\left(t+\delta t\right)\ . \tag{1.13}$$

After step 1.12, a force evaluation is carried out, to give $\mathbf{f}_i\left(t+\delta t\right)$ for step 1.13. This scheme advances the coordinates and momenta over a timestep δt.

Important features of the Verlet algorithm are: (a) it is exactly time reversible; (b) it is symplectic; (c) it is low order in time, hence permitting long timesteps; (d) it requires just one (expensive) force evaluation per step; (e) it is easy to program.

1.3.2 Constraints

It is quite common practice in classical computer simulations not to attempt to represent intramolecular bonds by terms in the potential energy function, because these bonds have very high vibration frequencies (and arguably should be treated in a quantum mechanical way rather than in the classical approximation). Instead, the bonds are treated as being constrained to have fixed length. In classical mechanics, constraints are introduced through the Lagrangian or Hamiltonian formalisms. Given an algebraic relation between two atomic coordinates, for example a fixed bond length b between atoms 1 and 2, one may write a constraint equation, plus an equation for the time derivative of the constraint

$$\chi\left(\mathbf{r}_1,\mathbf{r}_2\right) = \left(\mathbf{r}_1-\mathbf{r}_2\right)\cdot\left(\mathbf{r}_1-\mathbf{r}_2\right) - b^2 = 0 \tag{1.14}$$

$$\dot{\chi}\left(\mathbf{r}_1,\mathbf{r}_2\right) = 2\left(\mathbf{v}_1-\mathbf{v}_2\right)\cdot\left(\mathbf{r}_1-\mathbf{r}_2\right) = 0\,. \tag{1.15}$$

In the Lagrangian formulation, the constraint forces acting on the atoms will enter thus:

$$m_i\ddot{\mathbf{r}}_i = \mathbf{f}_i + \Lambda\mathbf{g}_i \tag{1.16}$$

where Λ is the undetermined multiplier and

$$\mathbf{g}_1 = -\frac{\partial\chi}{\partial\mathbf{r}_1} = -2\left(\mathbf{r}_1-\mathbf{r}_2\right) \quad \mathbf{g}_2 = -\frac{\partial\chi}{\mathbf{r}_2} = 2\left(\mathbf{r}_1-\mathbf{r}_2\right)\ . \tag{1.17}$$

It is easy to derive an exact expression for the multiplier Λ from the above equations; if several constraints are imposed, a system of equations (one per constraint) is obtained. However, this exact solution is not what we want: in practice, since the equations of motion are only solved approximately, in discrete timesteps, the constraints will be increasingly violated as the simulation proceeds. The breakthrough in this area came with the proposal to determine the constraint forces in such a way that the constraints are satisfied exactly at the end of each timestep. For the original Verlet algorithm, this scheme is called SHAKE. The appropriate version of this scheme for the velocity Verlet algorithm is called RATTLE.

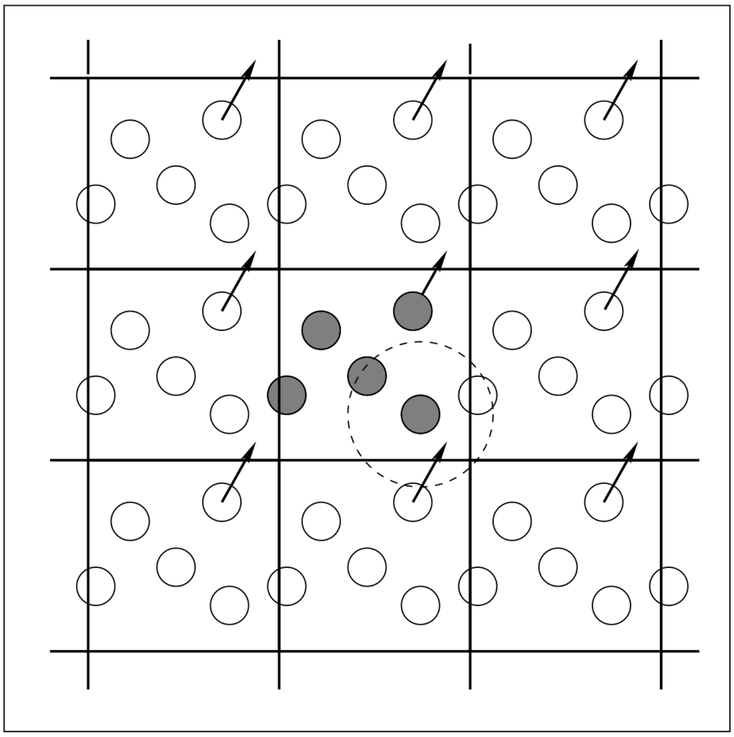


Figure 1.4: Periodic boundary conditions. As a particle moves out of the simulation box, an image particle moves in to replace it. In calculating particle interactions within the cutoff range, both real and image neighbors are included.

1.3.3 Periodic Boundary Conditions

Small sample size means that, unless surface effects are of particular interest, periodic boundary conditions need to be used. Consider 1000 atoms arranged in a $10 \times 10 \times 10$ cube. Nearly half the atoms are on the outer faces, and these will have a large effect on the measured properties. Even for $10^6 = 100^3$ atoms, the surface atoms amount to 6 % of the total, which is still nontrivial. Surrounding the cube with replicas of itself takes care of this problem. Provided the potential range is not too long, we can adopt the minimum image convention that each atom interacts with the nearest atom or image in the periodic array. In the course of the simulation, if an atom leaves the basic simulation box, attention can be switched to the incoming image. This is shown in Figure 1.4. Of course, it is important to bear in mind the imposed artificial periodicity when considering properties which are influenced by long-range correlations. Special attention must be paid to the case where the potential range is not short: for example for charged and dipolar systems.

1.4 Molecular Dynamics in Different Ensembles

In this section we briefly discuss molecular dynamics methods in the constant-NVT ensemble; the reader should be aware that analogous approaches exist for other ensembles, particularly to simulate at constant pressure or stress.

There are several approaches to conducting molecular dynamics at constant temperature rather than constant energy. One method, simple to implement and reliable, is to periodically reselect atomic velocities at random from the Maxwell-Boltzmann distribution. This is rather like an occasional random coupling with a thermal bath. The resampling may be done to individual atoms, or to the entire system.

A second approach is to introduce an extra "thermal reservoir" variable into the dynamical equations:

$$\dot{\mathbf{r}}_i = \mathbf{p}_i / m_i \tag{1.18}$$

$$\dot{\mathbf{p}}_i = \mathbf{f}_i - \zeta \mathbf{p}_i \tag{1.19}$$

$$\dot{\zeta} = \nu_T^2 \left(\frac{\mathcal{T}}{T} - 1 \right) . \tag{1.20}$$

Here ζ is a friction coefficient which is allowed to vary in time; ν_T is a relaxation rate for thermal fluctuations; and $\mathcal{T}$ stands for the instantaneous "mechanical" temperature. If $\mathcal{T} > T$, i.e. the system is too hot, then the "friction coefficient" ζ will tend to increase; when it is positive the system will begin to cool down. If the system is too cold, the reverse happens, and the friction coefficient may become negative, tending to heat the system up again.

2 Extracting Information from MD Simulations

The following is an adaptation of Martin Brehm's PhD thesis. He is the original author of TRAVIS, the Trajectory Visualizer and Analyzer, our in-house MD post-processing code. The section is designed to give you an overview of TRAVIS' capabilities. Together with a background knowledge of molecular dynamics simulations, you should be able to perform your own analyses.

2.1 Introduction

When a molecular simulation is performed, the primary result is always a trajectory, comprising the positions of all simulated particles in each simulated step. In the case of a molecular dynamics simulation, also the velocities are known. But within the assumption of sufficiently small trajectory step length, the velocities can be reconstructed from consecutive position vectors. Therefore, we will consider only the position vectors of the trajectory in the following; velocities are obtained from positions if required.

The trajectory contains all information that is available from the simulation; hence, all quantities of interest can be derived. However, mathematically speaking, a trajectory is a discrete path through a $3N$-dimensional space, where N is the number of particles. When several hundred or thousand particles have been simulated, this is a high-dimensional object, and therefore not suitable for direct evaluation (as an example, consider Fig. 2.1 on the following page, which depicts a snapshot of a simulation cell with approx. 1000 atoms, delivering a confusing picture).

Methods are required to reduce the dimensionality of the input data, enabling the visualization and interpretation. Within the scope of this exercise, these methods will be termed analyses.

Analyses can be divided into two basic types. The first type of analyses works on single simulation steps. If more than one step is analyzed, the result will simply be the arithmetic average over all single-step results. If this condition is fulfilled, the analysis will be called structural analysis, because it only analyzes the spatial structure within each simulation step, and does not take into account any notion of time or consecutive step order: If the simulated steps are mixed up in random order before performing a structural analysis, the result will not change at all.

All analyses which do not fulfill the criterion for structural analyses are called dynamical analyses. They take into account the information from the ordering of the steps (for molecular dynamics simulations, this is simply the time).

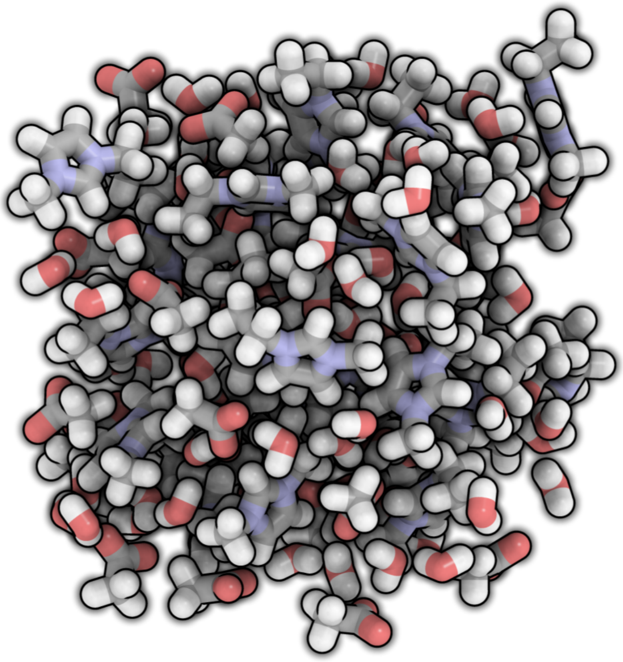


Figure 2.1: Snapshot of a simulation with approx. 1000 atoms, which cannot be evaluated directly.

2.2 Structural Analyses

In this section, many of the structural analyses implemented in TRAVIS will be discussed.

2.2.1 One-dimensional Histograms

TRAVIS allows the definition of several scalar quantities within the simulated system, over which histograms can be computed. Quantities that are relevant for this exercise are:

Distances between two atoms (or virtual atoms) can be defined. These atoms may be located within the same molecule or in different molecules, making both intra- and intermolecular analyses possible.

Angles can be defined by two vectors. Each vector can either be defined by two points in the system (atoms or virtual atoms), or as the normal vector of a plane defined by three points. These points may be located either within the same molecule (intramolecular), or within two different molecules (intermolecular). Angles between three different molecules can currently not be defined. Other vectors than position vectors can also be used for defining angles (e.g., dipole vectors or velocity vectors).

Dipole Moments of molecules can be defined if the trajectory contains either atomic partial charges or maximally localized Wannier centers for the dipole calculation.

Distances

The probably simplest analysis in TRAVIS is a distance histogram between two points in the system. A basic version of such a histogram is given in Equation 2.1. Due to the normalization factor, $\hat{g}_{ab}\left(r\right)$ depicts the average number of atoms of type b which are located at a distance of r from an atom of type a, where N_a and N_b depict the number of atoms from type a and b, respectively, $\mathbf{r}_i\left(t\right)$ represents the position vector of atom i in step t, T is the total number of trajectory steps, and δ stands for the Dirac delta distribution.

$$\hat{g}_{ab}\left(r\right) = \frac{1}{N_a N_b T}\sum_{t=1}^{T}\sum_{i=1}^{N_a}\sum_{j=1}^{N_b}\delta\left(r - \left|\mathbf{r}_i\left(t\right) - \mathbf{r}_j\left(t\right)\right|\right) \tag{2.1}$$

However, histograms of this kind will rise in the order of r^2 for larger values of r, as of course more and more neighbors can be found with increasing distance. This effect can be canceled by introducing an additional prefactor of $\frac{1}{r^2}$ into the equation. This is called "correction of the radial distribution". Further modification of the prefactor yields a histogram which is relative to the uniform density, where V is the simulation cell volume:

$$g_{ab}\left(r\right) = \frac{V}{\frac{4}{3}\pi r^2 N_a N_b T}\sum_{t=1}^{T}\sum_{i=1}^{N_a}\sum_{j=1}^{N_b}\delta\left(r - \left|\mathbf{r}_i\left(t\right) - \mathbf{r}_j\left(t\right)\right|\right) \tag{2.2}$$

This is the well-known radial pair distribution function (RDF), also termed as $g\left(r\right)$ by many authors. RDFs are important quantities for the comparison between simulation and experiment, because they can also be obtained experimentally (e.g., by neutron or X-ray scattering experiments).

Although the $\hat{g}_{ab}\left(r\right)$ function from above, which depicts the neighbor count up to a given distance, is mathematically not the integral of $g_{ab}\left(r\right)$, it is commonly referred to as "RDF integral".

As an example, the RDF (black) as well as the "RDF integral" (red) of a bulk phase water simulation are presented in Fig. 2.2 on the next page. The distance was measured between the centers of mass of the water molecules.

The RDF clearly shows the first three coordination shells of water. The first shell corresponds to the peak at around 275 pm, which is very sharp and distinct. According to the RDF integral, there are 4 to 5 water molecules within this first shell (measured up to 325 pm, where the RDF's first minimum is located, separating the first from the second coordination shell). The peaks in the RDF which correspond to the other shells are less distinct, because liquids do not possess a long-range order. For larger distances, the RDF converges to 1, which is (by definition) the uniform density of the system. The RDF can only be evaluated up to 780 pm, because in a system with periodic boundary conditions, distances larger than half of the simulation box edge length (which is 1567 pm here) cannot be taken into account.

Angles

Similarly to analyzing distances, TRAVIS can compute histograms over angles within the simulated system. This is expressed in Equation2.3, where $\sphericalangle\left(\mathbf{x},\mathbf{y}\right)$ depicts the angle between two vectors $\mathbf{x}$ and $\mathbf{y}$, N_a, N_b, and N_c depict the number of atoms from type a,

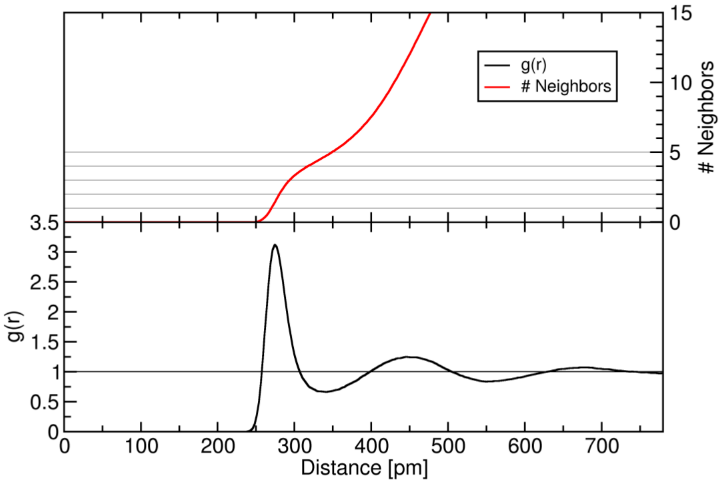


Figure 2.2: Radial distribution function (black) together with neighbor count (red) for bulk phase water simulation; distance measured between centers of mass.

b, and c, respectively, $\mathbf{r}_i)(t)$ represents the position vector of atom i in step t, T is the total number of trajectory steps, and δ stands for the Dirac delta distribution.

$$\hat{f}_{abc}(\alpha) = \frac{1}{N_a N_b N_c T} \sum_{t=1}^{T} \sum_{i=1}^{N_a} \sum_{j=1}^{N_b} \sum_{k=1}^{N_c} \delta\left(\alpha - \sphericalangle\left(\mathbf{r}_j(t) - \mathbf{r}_i(t), \mathbf{r}_j(t) - \mathbf{r}_k(t)\right)\right) \tag{2.3}$$

However, for angles defined in an intermolecular manner, Equation 2.3 will not give the expected results. To give an example, if a N_2 molecule within argon gas is simulated, and the angle $\sphericalangle(\mathrm{N-N}, \mathrm{N-Ar})$ is considered, a relatively uniform angular distribution is expected, but is not yielded from this equation. Angles around 90° are much more likely to be found than angles around 0° or 180°, distorting the picture. An additional correction prefactor is required, which is $\frac{1}{\sin\alpha}$. This correction is often called cone correction, and is required in most cases. With applied cone correction, the resulting equation reads

$$f_{abc}(\alpha) = \frac{1}{\sin\alpha N_a N_b N_c T} \sum_{t=1}^{T} \sum_{i=1}^{N_a} \sum_{j=1}^{N_b} \sum_{k=1}^{N_c} \delta\left(\alpha - \sphericalangle\left(\mathbf{r}_j(t) - \mathbf{r}_i(t), \mathbf{r}_j(t) - \mathbf{r}_k(t)\right)\right) . \tag{2.4}$$

To give an example, the angular distribution function between the dipole vectors of water molecules in a bulk phase water simulation is presented in Fig. 2.3 on the facing page. It can be seen that the angular distribution is almost uniform, not showing any distinct features. The jumps at 0° and 180° are due to poor sampling. Orientations with angles $< 90°$ are slightly preferred over such with angles $> 90°$, which is expected due to the dipole-dipole interaction between the molecules. But there is only a minor difference, which can be explained by the fact that all pairs of water molecules are taken into account for the angular distribution—and there are much more pairs at a large distance to each other than at a small distance. Therefore, pairs of neighboring water molecules (for which a strong correlation would be expected) have only a very small influence on the overall angular distribution. This problem can be solved by only taking into account pairs up to a certain cutoff distance.

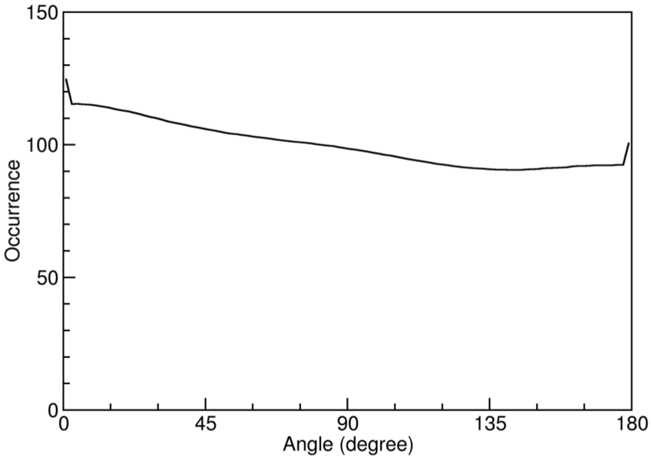


Figure 2.3: Angular distribution function between the dipole vectors of water molecules in bulk phase.

2.2.2 Combined Distribution Functions

Combined distribution functions (CDFs) are based on two or more one-dimensional histograms, which are evaluated at the same time. For a pair of molecules which is considered, all the one-dimensional functions yield a certain value. For each such pair, all values are put together into an n-tuple, and this n-tuple contributes to an n-dimensional histogram, which is the combined distribution function. Such a function contains more information than all the one-dimensional functions, because the correlation between the different quantities is preserved. The underlying one-dimensional functions are termed as channels of the CDF.

As a first example for a CDF, the hydrogen bond geometry in 1-Ethyl-3-methylimidazolium acetate is investigated. The corresponding CDF can be seen in Fig. 2.4 on the next page. The first CDF channel (x-axis) depicts the distance between the cation's H2 atom and the anion's oxygen atom (i.e., the hydrogen bond length), whereas the second channel (y-axis) measures the hydrogen bond angle. For an illustration of both CDF channels, see the sketch on the right hand side of the plot. On both plot axes, the one-dimensional histograms are depicted as axis projections.

The contour plot of the CDF clearly is much more informative than both one-dimensional axis projections. Each minimum and maximum of the distance distribution can be clearly associated to a distinct contribution in the contour plot, located at a certain angle. The axis projection of the angular axis seems almost uniformly distributed, however, the contour plot shows clear preferences for certain angles at certain distances.

The very intense signal at around 250 pm and angles between 120° and 180° corresponds to the strong hydrogen bond between cation and anion. The smaller peak at around 400 pm and angles of around 180° stems from the other oxygen atom of the acetate anion, which is not directly coordinating to the cation. Finally, the contribution at around 600 pm and around 30° is attributed to anions which coordinate to H4 and H5 of the cation (on the other side of the ring), leading to small angles and larger distances. The yellow rectangle on the upper left of the plot, comprising the interval of 0 pm to

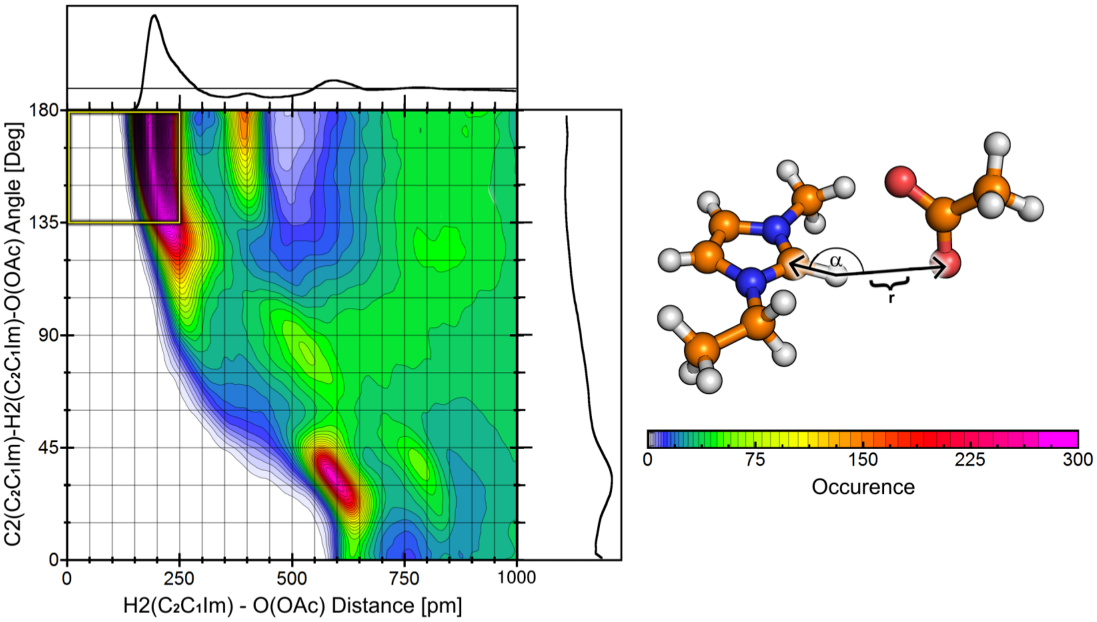


Figure 2.4: Combined distribution function depicting distance and angle of the hydrogen bond in [C_2C_1Im][OAc], together with axis projections of the one-dimensional histograms.

250 pm and 135° to 180°, is one possible criterion for a hydrogen bond.

Because the CDF shown above contains a distance axis and an angular axis, both radial correction and cone correction need to be applied to get an unbiased plot which actually depicts probability densities.

The channels of a CDF can be chosen from all available one-dimensional functions. This offers a large amount of flexibility, and there is a virtually infinite number of different CDFs which can be computed from one trajectory.

2.3 Dynamical Analyses

This chapter presents some of the dynamical analyses which have been implemented into TRAVIS.

2.3.1 Mean Square Displacement

When investigating the dynamics of a simulation, a very important tool is the *mean square displacement* (MSD) analysis. In simple words, this analysis depicts how far a selected particle on average moved away from its original position after a certain time τ. The exact definition is given by

$$MSD(\tau) = \frac{1}{N\,(T-\tau)} \sum_{i=1}^{N} \sum_{t=0}^{T-\tau} \left\| \mathbf{p}^i(t) - \mathbf{p}^i(t+\tau) \right\|^2 = \left\langle \left\| \mathbf{p}(t_0) - \mathbf{p}(t_0+\tau) \right\|^2 \right\rangle , \quad (2.5)$$

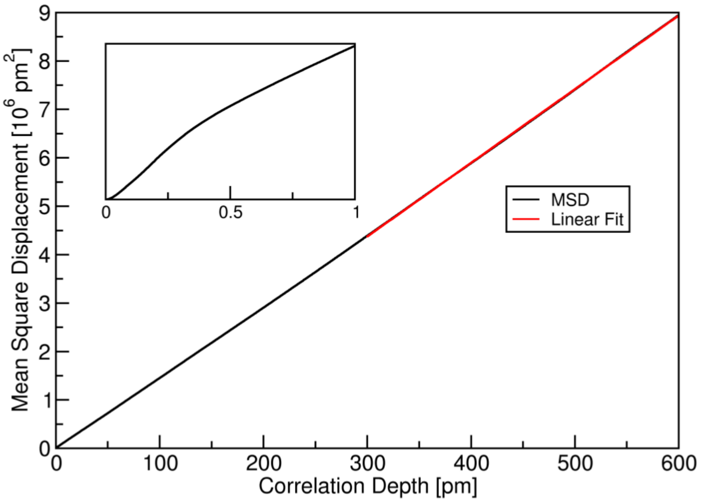


Figure 2.5: Mean square displacement of water molecules' centers of mass in classical water simulation *(black)*, together with linear fit *(red)*.

where N is the total number of observed particles, T is the number of simulated steps, and $\mathbf{p}^i(t)$ denotes the position vector of particle i in timestep t. The notation on the right-hand side is a short-hand expression, which indicates the averaging over all particles i and time origins t_0 by the angle brackets.

Based on the mean square displacement, the *self diffusion coefficient D* of the selected particles can be obtained by applying the so-called Einstein relation

$$2d\,D = \lim_{\tau\to\infty} \frac{\left\langle \left\| \mathbf{p}(t_0) - \mathbf{p}(t_0+\tau) \right\|^2 \right\rangle}{\tau}, \tag{2.6}$$

with d being the dimensionality of the system (usually 3). This equality states that the average slope of the mean square displacement becomes equal to $2d\,D$ in the limit of $\tau \to \infty$. The maximal possible value of τ is limited by the total simulation time. Therefore, it is necessary to perform rather long molecular dynamics simulations to obtain reliable self diffusion coefficients.

In the following example, the mean square displacement of water molecules' centers of mass in a classical molecular dynamics simulation of liquid water at 300 K will be determined. As this is a classical simulation with a large system (512 molecules), as well as a long simulation time (1 ns), a high linearity of the MSD curve is expected to be found. The result is depicted in Fig. 2.5 *(black curve)*, together with the linear fit *(red curve)*.

The MSD curve becomes linear after 1 ps (see inset). The linear fit shows a very good agreement with the original curve, indicating that Einstein's theory of diffusion holds for this system. The slope of the fit yields a self diffusion coefficient of $D = 2.54 \times 10^{-9}\,\mathrm{m}^2/\mathrm{s}$, which is in very good agreement with the experimental value of $D = 2.30\,\mathrm{m}^2/\mathrm{s}$ at 300 K.

2.3.2 Velocity Auto Correlation Function

Another interesting dynamical property of trajectories are velocity autocorrelation functions. These are defined as the temporal vector autocorrelation function of the velocity vectors of certain atoms from the simulation:

$$C(\tau) = \frac{1}{N\,(T-\tau)} \sum_{i=1}^{N} \sum_{t=0}^{T-\tau} \mathbf{v}^i(t) \cdot \mathbf{v}^i(t+\tau) = \Big\langle \mathbf{v}(t_0) \cdot \mathbf{v}(t_0+\tau) \Big\rangle , \tag{2.7}$$

where $\cdot$ denotes the dot product between the vectors, T is the total number of simulated timesteps, $\mathbf{v}^i(t)$ represents the velocity vector of atom i in timestep t, and N atoms are observed in total. The term on the right-hand side is a shorthand expression for the autocorrelation, which implicitly denotes the averaging over all time origins t_0 and all atoms i by the angle brackets. This function is not normalized to $C(\tau=0) = 1$, it rather starts at a value which is the mean square velocity of the observed atoms over the complete trajectory. For increasing correlation time τ, this function decays to smaller values, finally reaching zero at some point. Negative function values can be taken, because the vector autocorrelation function may yield negative values for antiparallel vectors. A function value of zero indicates that there is no correlation left within the velocities for the given correlation depth (except the case in which the vectors are correlated, but orthogonal to each other). One application of this function is the estimation of the *correlation time* of a system, after which no correlation in the velocity vectors remains. A second important application of velocity autocorrelation functions is the computation of power spectra, which are obtained by applying a Fourier transform to the autocorrelation function.

2.3.3 Vector Reorientation Dynamics

When the reorientation of certain vectors in the system is of interest, the *reorientation dynamics* analysis is very helpful. It is defined as the vector autocorrelation

$$C(\tau) = \frac{1}{T-\tau} \sum_{t=0}^{T-\tau} \frac{\mathbf{u}(t) \cdot \mathbf{u}(t+\tau)}{\|\mathbf{u}(t)\|\ \|\mathbf{u}(t+\tau)\|} , \tag{2.8}$$

where $\cdot$ denotes the vector dot product, T is the total number of simulated timesteps, and $\mathbf{u}(t)$ is a vector fixed on a molecule in step t. If more than one vector (or more than one molecule) shall be considered, the arithmetic average of all vector's autocorrelation functions will be taken. The use of normalized vectors for the autocorrelation function disregards the vector's lengths and ensures that $C(\tau=0) = 1$ holds.

For many analyses, it is important to calculate vector autocorrelation functions $C_n(\tau)$ of a certain order n (i.e., to apply the Legendre polynomial P_n) to the dot product of the vectors before autocorrelating:

$$C_n(\tau) = \frac{1}{T-\tau} \sum_{t=0}^{T-\tau} P_n\!\left(\frac{\mathbf{u}(t) \cdot \mathbf{u}(t+\tau)}{\|\mathbf{u}(t)\|\ \|\mathbf{u}(t+\tau)\|} \right) \tag{2.9}$$

As $P_1(x) \equiv x$, the "normal" autocorrelation given in Eq. 2.8 can be written as $C_1(\tau)$, which is often done to clarify that no Legendre polynomial was applied.

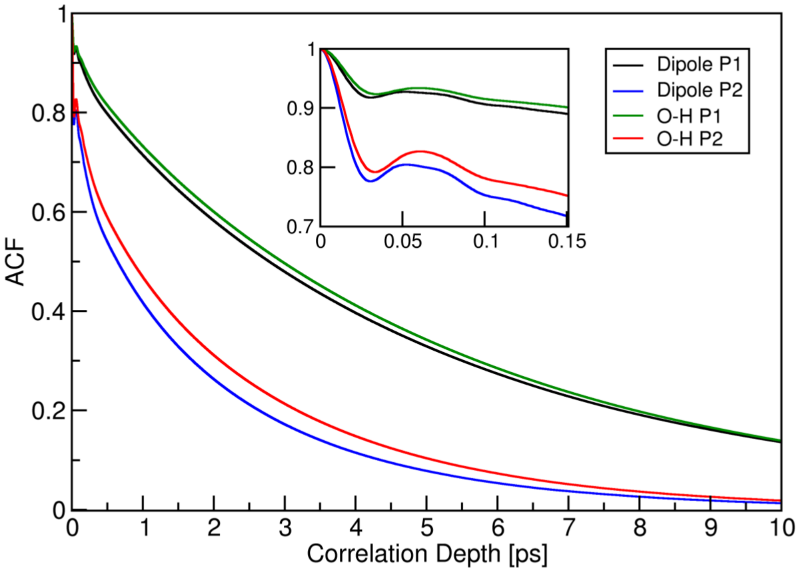


Figure 2.6: Vector reorientation dynamics of different vectors in a bulk phase water simulation.

Vector reorientation dynamics functions are a very important connection between simulation and experiment: First order relaxation times τ_1 can be inferred from the experimental Debye dielectric relaxation time τ_D. Second order relaxation times τ_2 can be experimentally determined by several methods: NMR, femtosecond infrared spectroscopy, and quasi-elastic neutron scattering (QENS). The time resolution of ultrafast infrared spectroscopy even allows the direct determination of $C_2(\tau)$. On the simulation side, the reorientation time can be obtained from a vector autocorrelation function by integrating the curve from $\tau = 0$ to $\tau = \infty$.

TRAVIS allows the definition of the vector for autocorrelation in different ways. It is either possible to define a vector by specifying two points, or by defining a plane based on three points, whose normal vector is used. Apart from position data, TRAVIS can also use velocity vectors or dipole vectors (e.g., from atomic partial charges or from Wannier centers) for the autocorrelation. This leads to an important application of the vector autocorrelation function: The Fourier transform of the dipole vector autocorrelation function is the infrared spectrum of a system, which is another very important quantity for comparison between theory and experiment.

To give an example for the methodology introduced above, a classical molecular dynamics simulation of liquid water (512 molecules), which has been performed at a temperature of 300 K was analyzed. The reorientation dynamics of the O−H vector and the dipole vector of water were investigated up to correlation depths of 50 ps. Both functions were computed with and without use of the second Legendre polynomial, denoted by P2 and P1, respectively. The four resulting curves are shown in Fig. 2.6.

The plot shows that the vector autocorrelation functions with second Legendre polynomial applied (P2) differ strongly from those of first order (P1), whereas the functions for the different vectors (Dipole and O−H) are similar. This leads to the conclusion that the water molecules in the simulation rotate in a manner such that both vectors reorientate similarly. The P2 functions decay to zero at a correlation depth of around $\tau = 20\,\text{ps}$, whereas the P1 functions reach this state at around 40 ps (not shown in the plot due to the axis interval). A value of zero means that at this correlation depth the vectors are completely uncorrelated. In principle, the functions might also take negative values, because the scalar product becomes negative for antiparallel vectors. However, for most systems, this can be regarded as artifact due to poor sampling, and will vanish if the simulation time is increased.

2.3.4 Lifetime of Configurations

When analyzing simulations, in many cases the average lifetime of a certain configuration (e.g., hydrogen bonds, ion pairs, rotamers) is of particular interest. Such configurations can be defined via geometrical criteria (i.e., distances and angles between atoms of the molecules). A pair of molecules is called a *dimer* in one given simulation step if the criteria are fulfilled within this step. The lifetime of such dimers can be analyzed using correlation functions, which shall be explained in the following.

For each pair of molecules *i* and *j* within the simulation, we define a simple function as follows:

$$\beta_{ij}\left(t\right) = \begin{cases} 1, & \text{if criteria are fulfilled between } i \text{ and } j \text{ in step } t \\ 0, & \text{else} \end{cases} \tag{2.10}$$

Then, the autocorrelation of each such function is determined, and the average over all pairs is taken:

$$C_\beta(\tau) = \frac{1}{N^2\,(T-\tau)} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=0}^{T-\tau} \beta_{ij}(t)\,\beta_{ij}(t+\tau) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \Big\langle \beta_{ij}(t_0)\,\beta_{ij}(t_0+\tau) \Big\rangle, \tag{2.11}$$

where N depicts the number of particles analyzed, and T is the number of simulation steps. The resulting function $C_\beta(\tau)$ starts at 1 for $\tau = 0$, and decays for increasing correlation depth. There is an intuitive interpretation for the values of such functions: $C_\beta(t+\tau)$ denotes the probability of the criteria being fulfilled at a given time $t+\tau$, when they were fulfilled at time t. The function does not decay to zero, but to the finite value $\left\langle \beta_{ij}(t) \right\rangle^2$, which is the square of the average probability for any two molecules to form a dimer at some time. Because it is more convenient to have a correlation function which decays to zero if no correlation exists anymore, the average value is subtracted. Autocorrelation functions of this type are termed as *intermittent* functions, because autocorrelations are calculated irrespective of possible dissociation events: The dimer may reform at any time, resulting in contributions at large correlation times, similar to as if the dimer would have lived without interruption.

There is also another way of computing dimer autocorrelation functions, which is opposite to the *intermittent* formalism from above: The *continuous* autocorrelation function. Each time a dimer dissociates, this dimer is considered as "definitely broken" for the rest of the simulation. If the dimer reforms, it is considered as a different dimer, though being formed from the same two molecules.

For a system in which each pair of molecules only forms a dimer for one continuous interval (e.g., a very large gas simulation at high temperature), the *continuous* and *intermittent* functions will be identical. The longer two molecules stay in close proximity, breaking and reforming dimers from the same pairs of molecules very often (e.g., in a simulation of a very viscous liquid), the larger the difference between both functions becomes. Because simulations are performed at a finite temperature, several types of intra- and intermolecular vibrations occur, which can cause a dimer to dissociate and reform many times (due to using a simple geometric condition). The *intermittent* autocorrelation function has the advantage of not overvaluing such events, because dissociated dimers are allowed to reform there.

The dimer autocorrelation functions for four different criteria for dimers of 1-Ethyl-3-methylimidazolium acetate (including three hydrogen bond criterion) are presented in Fig. 2.7 on the following page. The upper panel depicts the *intermittent* autocorrelation functions, whereas the lower panel contains the *continuous* functions.

It can be seen that the *intermittent* functions generally show a much slower decay, which is clear from the definition (reforming of dimers is allowed). Please note the different X axis scale of the plots. Interestingly, the four *intermittent* functions are very similar to each other, in spite of the strongly differing criteria which have been used. The *continuous* functions strongly depend on the exact properties of the criterion, because every short interruption of a dimer life span significantly influences the result.

2.3.5 Vibrational Spectra

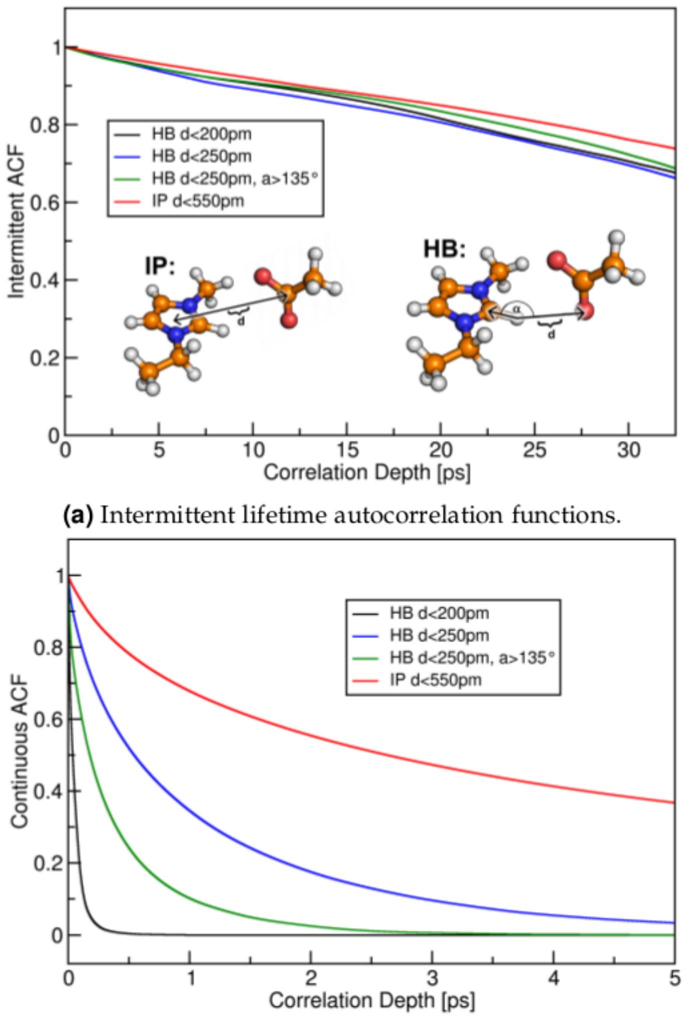
Computational spectra constitute a very important tool for the comparison between experiment and simulation. There exists a variety of methods to measure vibrational spectra (e.g., infrared and Raman spectra) from all different kinds of samples, and there also exist powerful methods for the calculation of such spectra from molecular dynamics trajectories, as discussed in the following.

If the velocity autocorrelation of all atoms within a molecular dynamics simulation is computed, and this autocorrelation function is then Fourier transformed, the *power spectrum* of the system is obtained. Because all atoms are considered, the power spectrum contains all vibrations which occur within the system (i.e., all normal modes). Unfortunately, there is no experimental method to measure power spectra. However, a power spectrum gives information on the wave number of the system's normal modes, and as some of these normal modes are probably also visible in an infrared spectrum, the wave numbers may be compared anyway.

The intensities of the peaks cannot be directly compared to the experiment, but they still carry useful information: In a power spectrum, the integral over each peak is proportional to the average energy of the corresponding normal mode of the system (i.e., the temperature of this degree of freedom). If the system is in thermal equilibrium, the *equipartition theorem* states that all degrees of freedom contain the same amount of energy on average. If this is the case, then all peaks in the spectrum will possess the same integral value. Therefore, it is possible to judge if a system is well equilibrated by checking the peak integrals in the power spectrum for equal value, which is an important application. However, as in larger systems the normal modes overlap strongly, this requires knowledge on how many normal modes are corresponding to each visible peak.

Fig. 2.8 on page 25 depicts the power spectrum of an *ab initio* molecular dynamics simulation of 1-Ethyl-3-methylimidazolium acetate in the liquid phase.

The power spectrum looks very similar to an infrared spectrum. The peaks at wave



(a) Intermittent lifetime autocorrelation functions.

(b) Continuous lifetime autocorrelation functions.

Figure 2.7: Intermittent *(upper panel)* and continuous *(lower panel)* lifetime autocorrelation functions for 3 different hydrogen bond criteria *(black, blue, green)* as well as an ion pair criterion between [EMIm]$^+$ and [OAc]$^-$ *(red)* in System A.

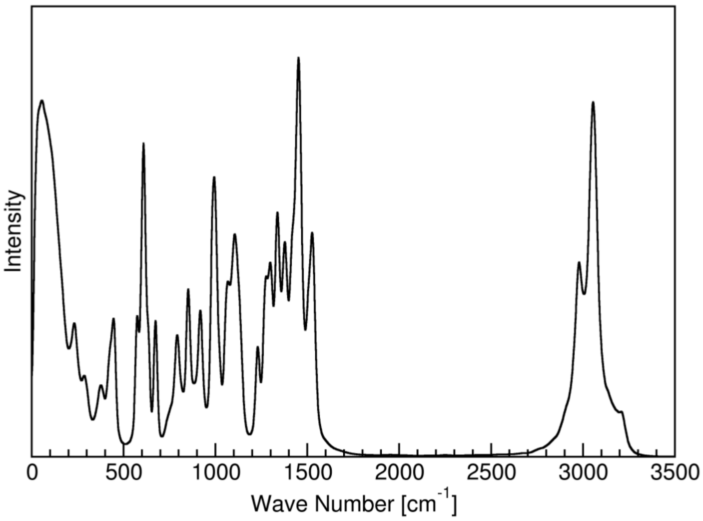


Figure 2.8: Power spectrum of an [EMIm][OAc] simulation.

numbers of around 3000 cm^{-1} correspond to C−H stretching vibrations, whereas the lower-frequency peaks up to 1000 cm^{-1} are mainly due to bending modes. Contributions at very low wave numbers of around 100 cm^{-1} to 200 cm^{-1} correspond to intermolecular motions.

One important advantage of a power spectrum is that it is computed on a per-atom basis, and does not "know" about the concept of molecules. Therefore, power spectra can not only be obtained for the complete simulation box, but also for selected molecules, and even for selected single atoms. In such a power spectrum, all normal modes which comprise the selected atoms will appear as peaks. This is of great help for assigning normal modes to certain functional groups within a molecule (e.g., in bulk phase).

3 Exercise

Perform three classical molecular dynamics simulations of liquid water with different force fields! These force fields are the rigid SPC/E model, the flexible SPC/Fw model, and the electronically polarizable SPC-FQ model. Use TRAVIS to perform the following analyses:

- radial distribution functions
- angular distribution functions
- combined distribution functions
- hydrogen bond dynamics
- mean square displacements
- velocity auto correlation functions
- power spectra
- dipole reorientation dynamics

Verify your results by comparing to published simulation data! If possible, evaluate the accuracy of the force fields by comparing to experimental data! Discuss the dynamical quantities (hydrogen bond life times, reorientation times, diffusion coefficient) in the context of the putative memory of water!

3.1 Working Environment

We assume that you are familiar with the Linux working environment. This includes

- navigating though the file system
- copying and moving files, creating directories, etc.
- viewing and editing text files
- plotting data

If this is not the case or if you need any help, contact your assistant.

You will be provided with a set of files and directories to work with. We assume that you have copied them to your working directory. Please perform all simulations, analyses, etc. in the temporary directory of your workstation /tmp1. To avoid confusion and conflicts with other users, create a unique working directory. For example:

```
> mkdir /tmp1/name
> cd /tmp1/name
> scp lincipb:/home/domaros/wp12.tar.gz .
> tar -xf wp12.tar.gz
```

The resulting directory contains the LAMMPS and TRAVIS executables, as well as one directory for each simulation that you are going to perform. Each of these directories contains a directory called `travis`, which we recommend to use for the analyses.

3.2 SPC/E Water

The extended simple point charge water model (SPC/E) is a rigid, three-site, effective pair potential.[3] It is composed of three interaction sites located on the oxygen and and both hydrogen atoms All interaction sites carry point charges. Effective pair potentials incorporate the average many-body interactions into the interaction between pairs.

Non-bonded interactions are described by a Lennard–Jones potential between oxygen–oxygen pairs. There are no non-bonded interactions between hydrogen–hydrogen and oxygen–hydrogen pairs.

The model is rigid, thus constraints must be applied during the simulation to keep the geometry intact. Rigid models allow larger time-steps, because the timestep in an MD simulation must reflect the fastest motion of the system. In the case of rigid water models, these are not the fast intramolecular vibrations, but the somewhat slower librational motions around hydrogen bonds. This makes rigid force fields computationally very efficient.

3.2.1 Preparations

Before starting the simulation, perform a literature research and look up the following force field parameters for the SPC/E water model! We are going to need them in a moment.

- oxygen and hydrogen charges (in multiples of the electron charge)
- Lennard–Jones well depth and diameter (in kcal mol^{-1} and deg)
- O−H bond length and H−O−H angle (in Å and degrees)
- density at ambient conditions ($T = 298.15\,\mathrm{K}, P = 1\,\mathrm{bar}$)

We start all water simulations by putting $8 \times 8 \times 8$ water molecules on a cubic lattice. Because a cubic lattice is an energetically unfavorable configuration, we start the simulation at reduced density of $\sim 0.7\,\mathrm{g\,cm^{-3}}$. Over the course of the simulation, we will slowly reduce the box size to its final value, which shall correspond to the density of liquid water at ambient conditions. Since our box is cubic, its volume will simply be $V = a^3$, where a is the extent of the box in any dimension. Given the density and the number of water molecules, you can calculate the desired box volume and thus the final value of a. Calculate this value (in Å), we are going to need it in a moment!

3.2.2 Simulation

Edit the LAMMPS data file for a single SPC/E water molecule `spce.lammpsdata` and substitute the place holders with the force field parameters that you have acquired. Edit the LAMMPS input file `lammps.in` and enter the correct value of the box size! Start your simulation!

```
> cd /tmp1/name/wp12/spce
> nano spce.lammpsdata
> nano lammps.in
> mpirun -np 4 /tmp1/name/wp12/lammps -in lammps.in
```

LAMMPS will start, showing you important information about the course of the simulation on the screen. This output is also written to a file called `log.lammps`, in case you would like to inspect it later.

The first part of the simulation will be introduced by the line » `STARTING MELTING AND COMPRESSION TO FINAL BOX SIZE!` «. Follow the screen output and verify that the box compression ends at the desired target density! The trajectory belonging to this part of the simulation will be written to a file called `melting.lmp`.

After the box compression, an equilibration is started, which is introduced by the line » `STARTING EQUILIBRATION AT TARGET BOX SIZE!` «. This equilibration ensures that the system loses all information about its initial configuration, so that the analyses that are going to be performed on subsequent simulation steps are not biased by the initial configuration. Notice, how the temperature adjusts to its final value of 298.15 K and starts fluctuating around this value.

The final step of the simulation is introduced by the line » `STARTING PRODUCTION RUN!` «. This is the part of the trajectory that we are going to use for analyses with TRAVIS. Two trajectory files are written in this step. The file `production100ps.lmp` contains the first 100 ps of the production run, with one configuration dump every 1 fs. This trajectory file will be used to perform analyses that require velocity information (acfs, power spectra). The second trajectory file is called `production1ns.lmp`. This file contains the complete trajectory of the production run, with configurations dumped every 5 fs. The production run lasts for 1 ns. Splitting trajectories in segments with high and low dumping frequencies is a common technique in classical molecular dynamics simulations to deal with the otherwise arising huge file sizes.

3.2.3 Visualization

Its of crucial importance to visually check MD trajectories before proceeding with their analysis. This may reveal errors in the simulation protocol, which can then be fixed. To visualize the trajectory of the melting process, open it in VMD.

```
> vmd -lammpstrj melting.lmp
```

VMD provides a movie-player-like interface that can be used to browse through the trajectory. Use the `Graphics` → `Representations` menu to switch between drawing methods! Try CPK, VDW, and HBonds. Multiple such representations can be created and overlap, by hitting the `Create Rep` button. Try overlapping CPK and HBonds!

To visualize the simulation box, open VMD's command line interface `Extension` → `TK Console`, and enter `pbc box`!

3.2.4 Analyses

Perform the following analyses with TRAVIS! Before you start, read the following notes that apply to each TRAVIS analysis performed in this exercise.

- Use a new directory for each TRAVIS analysis and for each simulation! For example for the calculation of the mean square displacement of SPC/E water, use `/tmp1/name/wp12/spce/travis/msd`. These directories have already been created for you.

- Start TRAVIS from the corresponding directory like this:
  ```
  > /tmp1/name/wp12/travis -p ../../production1ns.lmp
  ```
  Or, if you are asked to use the `production100ps.lmp` trajectory file:
  ```
  > /tmp1/name/wp12/travis -p ../../production100ps.lmp
  ```

- TRAVIS works interactively by asking questions that you must answer. For example, the first question that TRAVIS asks is:
  ```
  Use the advanced mode until the analysis selection menu (y/n)?  [no]
  ```
  Possible answers for these questions are given in parentheses. In this case, the answer can be yes or no. TRAVIS almost always gives reasonable default answers. These are given in brackets and can be chosen by simply hitting the `Enter` key. In this example, the default answer is reasonable. As rule of thumb: You will never need the advanced mode of TRAVIS in any of the analyses that follow.

- The next couple of questions are the same for all analyses. The cell geometry should be recognized correctly, so answer with yes. The cell geometry in our simulations is fixed, so answer with no. The molecule recognition should succeed; an H_2O molecule should be recognized. Answer with yes. You probably won't want images of the structures, so answer with no.

- The timestep in `production1ns.lmp` is 5 fs. The timestep in `production100ps.lmp` is 1 fs. Note that the actual timestep for most simulations is smaller, but coordinates will always be dumped in the given intervals.

- TRAVIS will inform you about the estimated remaining time of the analyses (ETA).

- At the end of an analysis, TRAVIS prints results to the screen and informs you, which files were created. If you lose the screen output, you can retrieve it by looking at the file `travis.log`. Files ending in .csv can be plotted with Xmgrace. Files ending in .agr can be opened with Xmgrace. Files ending in .gp can be visualized with gnuplot. Files ending in .xyz can be opened with VMD.

- TRAVIS has the concept of reference molecules and observed molecules. Since this a pure water simulation, both will be water. This concept will be explained in more detail in the ADF section.

Radial Distribution Functions – rdf

Calculate intermolecular radial distribution functions for the oxygen–oxygen and oxygen–hydrogen distances. To calculate the oxygen–oxygen distance, choose the oxygen atom from the reference molecule and the oxygen atom from the observed molecule. The default values for radius, resolution, correction, temporal development, and conditions are reasonable in this analysis. Keep them!

Since we would like to calculate two RDFs (a second one for the oxygen–hydrogen distance), answer the question about "adding another observation" with yes. This time, choose the oxygen atom from the reference molecule and the hydrogen atom from the observed molecule. Use only every 20th timestep to keep the time of the analysis reasonable. Answer all other questions with their default!

Angular Distribution Functions – adf

Calculate an intermolecular angular distribution function of the hydrogen bond angle. This angle is defined as the $\mathrm{O^A \cdots O^D - H}$ angle, where $\mathrm{O^D}$ denotes the oxygen atom of a donor molecule and $\mathrm{O^A}$ denotes the oxygen atom of an acceptor molecule. In TRAVIS, this angle can be calculated as the angle between two vectors which describe a position. One vector would be the $\mathrm{O^D - H}$ vector and the other one the $\mathrm{O^D \cdots O^A}$ vector. Thus both vectors connect two points (the respective atoms).

Let's set the donor molecule as reference molecule. In this case, the base point of the first vector ($\mathrm{O^D - H}$) will be the oxygen atom that belongs to the reference molecule. The tip of the vector will be the hydrogen atom that belongs to the reference molecule.

The base point of the second vector ($\mathrm{O^D \cdots O^A}$) will again be the oxygen atom of the reference molecule. Note that this is *not* the default. The tip will be the oxygen atom of the observed molecule.

This ADF is not supposed to be mirror-symmetric. All other questions can be answered with their defaults, except for the very last question. Once more, choose only every 20th step for analysis, to keep the computational time of the analysis reasonable.

Combined Distribution Functions – cdf

Calculate combined distribution functions with two channels: one for the oxygen–oxygen distance (rdf channel) and another for the $\mathrm{O^A \cdots O^D - H}$ angle between two neighboring water molecules (i.e, the hydrogen bond angle). The latter is an adf channel. This is no three-body analysis.

The two channels are equal to the two distribution functions that you have already calculated. You can use the answers to TRAVIS' question that you gave above. Once more, use only every 20th step.

Extract geometric hydrogen bond criteria form this plot. These shall be in the form: If the $\mathrm{O^A \cdots O^D}$ distance is less than r_{cut} and if the $\mathrm{O^A \cdots O^D - H}$ angle is less then θ_{cut}, then two water molecules are bound by a hydrogen bond. You will need these in the following analysis.

Hydrogen Bond Dynamics – aggr

In TRAVIS, the hydrogen bond time correlation function belongs to the category of aggregation functions `aggr`. It is a dimer existence auto correlation function (DACF). We do not want to calculate any other function in this menu.

The resolution of this analysis should be 250 ps, which corresponds to 50 000 steps at a timestep of 5 fs. Since we want to determine the hydrogen bond life time, which is related to the integral of this function, we would like to calculate the integral as well. Keep all settings to their default, until you are asked to define a distance criterion. Answer with yes. Define a distance criterion between two oxygen atoms, where the allowed distances are between 0 and the value that you have determined in the previous analysis.

Define an angular condition as well. Enter the vectors that describe the hydrogen bond angle $O^A \cdots O^D$ just as you have entered them in the two previous analyses.

Start by calculating an intermittent function and calculate a continuous function in a different run. Use every available timestep for this analysis! Keep all other options at their default.

Mean Square Displacement – msd

Calculate the mean square displacement of the oxygen atoms. Choose a resolution of 1 ns, which corresponds to 200 000 steps. Use every 100th step for the calculation of the mean square displacement. Keep all other options at their default.

Velocity Auto Correlation Function – acf

Use the `production100ps.lmp` trajectory file in this analysis.

Calculate the velocity auto correlation function of the oxygen atoms. Do not compute a power spectrum in this menu! Do not compute a global VACF! Keep all other options at their default. For visualization, choose the file where a windows function has been applied (ending in .w.csv).

Power Spectrum – power

Use the `production100ps.lmp` trajectory file in this analysis.

Calculate the power spectrum of the oxygen atoms and of the complete system. Thus choose the oxygen atom, when asked for an atom selection and answer the question about calculating the power spectrum of the complete system with yes. Keep all other options at their default.

Dipole Reorientation Dynamics – rdyn

Calculate the dipole reorientation dynamics of the system. Specify the dipole moment by providing fixed atomic charges to TRAVIS. Use the same charges that you have entered in the LAMMPS data file. Keep all other options at their default! Fit the resulting function to a sum of two exponentials:

$$c\left(t\right) = a_1 \exp\left(-\frac{t}{\tau_1}\right) + a_2 \exp\left(-\frac{t}{\tau_2}\right) \tag{3.1}$$

If you do so, the integral over the time correlation function will simply be

$$\tau_{\text{reor}} = \int_0^\infty a_1 \exp\left(-\frac{t}{\tau_1}\right) + a_2 \exp\left(-\frac{t}{\tau_2}\right) dt = a_1\tau_1 + a_2\tau_2\,. \tag{3.2}$$

This integral is the average time of water reorientation.

3.3 SPC/Fw Water

The flexible simple point charge water model (SPC/Fw) is a flexible, effective, three-site, pair potential.[6] It is composed of three interaction sites located on the oxygen and and both hydrogen atoms. All interaction sites carry point charges.

Non-bonded interactions are described by a Lennard–Jones potential between oxygen–oxygen pairs. There are no non-bonded interactions between hydrogen–hydrogen and oxygen–hydrogen pairs.

The model tries to improve the simple rigid version, by adding bond and angle flexibility. This limits the maximum timestep that can be used for such a simulation and makes it computationally more demanding to achieve the same total simulation length.

3.3.1 Preparations

Before starting the simulation, perform a literature research and look up the following force field parameters for the SPC/Fw water model! We are going to need them in a moment.

- oxygen and hydrogen charges (in multiples of the electron charge)
- Lennard–Jones well depth and diameter (in $\mathrm{kcal\,mol^{-1}}$ and Å)
- O−H bond length and H−O−H angle (in Å and deg)
- O−H stretching force constant (in $\mathrm{kcal\,mol^{-1}\,\AA^{-2}}$)
- H−O−H bending force constant (in $\mathrm{kcal\,mol^{-1}\,deg^{-2}}$)
- density at ambient conditions ($T = 298.15\,\mathrm{K}, P = 1\,\mathrm{bar}$)

Important note: In LAMMPS, harmonic bond and angle potentials are defined as

$$U_{\text{bond}}\left(r\right) = K_{\text{bond}}\left(r - r_0\right)^2 \tag{3.3}$$

$$U_{\text{angle}}\left(\theta\right) = K_{\text{angle}}\left(\theta - \theta_0\right)^2\ , \tag{3.4}$$

where K_{bond} and K_{angle} are the bond and angle force constants and r_0 and θ_0 are the equilibrium bond length and angle. In the literature, however, the force fields are often specified as

$$U_{\text{bond}}\left(r\right) = \frac{K'_{\text{bond}}}{2}\left(r - r_0\right)^2 \tag{3.5}$$

$$U_{\text{angle}}\left(\theta\right) = \frac{K'_{\text{angle}}}{2}\left(\theta - \theta_0\right)^2\ . \tag{3.6}$$

Thus depending on your source of these force constants, you may need to divide the force constants by two!

3.3.2 Simulation, Visualization and Analyses

Our simulation protocol is the same as for the SPC/E model, except for the constraints. Edit the LAMMPS data file for a single SPC/Fw water molecule `spcfw.lammpsdata` and substitute the place holders with the force field parameters that you have acquired. Edit the LAMMPS input file `lammps.in` and enter the correct value of the box size! Start your simulation!

```
> cd /tmp1/name/wp12/spcfw
> nano spcfw.lammpsdata
> nano lammps.in
> mpirun −np 4 /tmp1/name/wp12/lammps −in lammps.in
```

Check your simulation visually in VMD! Perform the same analyses that you have performed for SPC/E!

3.4 SPC-FQ Water

The SPC-FQ water model is a rigid, three-site, polarizable water model.[5] Unlike its non-polarizable counterpart, the SPC/E model, no fixed charged are used in this model. Instead, the charges on all atoms may change because of fluctuations in the local electric field.

The model tries to improve the simple SPC/E water model, by introducing many-body effects in the form of electronic polarization. Special algorithms are required to solve the electrostatic equations of a such a system, which make polarizable force fields very expensive. The obvious advantage is that polarizable models may react to changes in the environment of a water molecule, for example, introduction of an interface.

3.4.1 Preparations

Before starting the simulation, perform a literature research and look up the following force field parameters for the SPC-FQ water model! We are going to need them in a moment.

- Lennard–Jones well depth and diameter (in $\mathrm{kcal\,mol^{-1}}$ and Å)
- O−H bond length and H−O−H angle (in Å and deg)
- density at ambient conditions ($T = 298.15\,\mathrm{K}$, $P = 1\,\mathrm{bar}$)

3.4.2 Simulation, Visualization and Analyses

Our simulation protocol is the same as for the SPC/E model, except for special algorithms, which treat the electronic polarizability of SPC-FQ. Edit the LAMMPS data file for a single SPC-FQ water molecule `spc-fq.lammpsdata` and substitute the place holders with the force field parameters that you have acquired. Edit the LAMMPS input file `lammps.in` and enter the correct value of the box size! Start your simulation!

```
> cd /tmp1/name/wp12/spc−fq
> nano spc−fq.lammpsdata
> nano lammps.in
> mpirun −np 4 /tmp1/name/wp12/lammps −in lammps.in
```

Check your simulation visually in VMD! Perform the same analyses that you have performed for SPC/E! Note that although TRAVIS can deal with fluctuating charges, it complicates the calculation of the dipole vector. For this reason, you do not need to calculate the dipole reorientation dynamics for this water model.

3.5 Report and Questions

Write a report about your results!

Write down the details about the simulation protocol of the SPC/E simulation. For this purpose, find out the following pieces of information by reading this document, the LAMMPS input file, and the LAMMPS data file:

- number of molecules
- force field
- box size
- temperature
- thermostat type
- thermostat time constant
- constraints algorithm
- timestep
- Lennard–Jones cutoff
- treatment of long-range Coulomb forces
- initial configuration
- length of equilibration
- length of production

Giving this kind of information is important to ensure reproducibility by other groups. To get the information, open the LAMMPS input file and try to understand its meaning. Use the LAMMPS online manual http://lammps.sandia.gov/doc/Manual.html to understand the commands better. If in doubt, ask your assistant.

Compare the results of all models with each other and, if possible, with experimental data or published literature values from other simulations. Based on that comparison, judge the quality of each model and its computational cost. Note, that LAMMPS shows its total run time at the end of each simulation. Make a recommendation which water model performs best for the purpose of bulk water simulations at ambient temperatures.

You have calculated two values per model that are related to the average hydrogen bond life time: The integrals over the intermittent and continuous hydrogen bond time correlation functions. The integral over the continuous function represents the time until an initial hydrogen bond first breaks. It is a measure for the strength of a hydrogen bond. Use these times to judge the hydrogen bond strength of each water model.

The integral over the intermittent function represents the time until an initially bound pair of water molecules is fully separated. In the case of liquid water, this process is however coupled to diffusion, which happens on a much longer timescale than the hydrogen bond formation and breaking. The two integrals thus represent a lower and an upper bound for the average life time of a hydrogen bond. Extracting the true life time from these functions is somewhat involved.[4] For the purpose of this exercise the range given by both integrals will be sufficient.

Discuss the life times, the average time of dipole reorientation, and the self diffusion coefficient. Would you suspect that water has memory based on these values?

Bibliography

[1] M. P. Allen. *Introduction to Molecular Dynamics Simulation*. URL: http://udel.edu/~arthij/MD.pdf.

[2] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Oxford Science Publications. Clarendon Press, 1989. ISBN: 9780198556459.

[3] H. J. C. Berendsen, J. R. Grigera, and T. P. Straatsma. "The missing term in effective pair potentials." In: *J. Phys. Chem.* 91.24 (1987), pp. 6269–6271. DOI: 10.1021/j100308a038.

[4] A. Luzar. "Resolving the hydrogen bond dynamics conundrum." In: *J. Chem. Phys.* 113.23 (2000), pp. 10663–10675.

[5] S. W. Rick, S. J. Stuart, and B. J. Berne. "Dynamical fluctuating charge force fields: Application to liquid water." In: *J. Chem. Phys.* 101.7 (1994), pp. 6141–6156. DOI: http://dx.doi.org/10.1063/1.468398.

[6] Y. Wu, H. L. Tepper, and G. A. Voth. "Flexible simple point-charge water model with improved liquid-state properties." In: *J. Chem. Phys.* 124.2 (2006), p. 024503. DOI: http://dx.doi.org/10.1063/1.2136877.