NTNU – Trondheim
Norwegian University of
Science and Technology

Molecular Dynamics Simulations in GROMACS

Project report

TKP 4555 Advanced Simulation (Fall 2017)

Prof. Heinz A. Preisig and Assoc. Prof. Brian A. Grimes

Written by: Jan C. Schulze (RWTH Aachen, Germany)

Contents

Nomenclature and Definitions

Symbol	***Description***	***Units***
$\mathbf{f}$	Force vector	$^{\mathrm{kJ}}/_{\mathrm{mol\cdot nm}}$
f	Electric conversion factor	$^{\mathrm{kJ\cdot nm}}/_{\mathrm{mol\cdot e^2}}$
k_b	Force constant (streching vibrations)	$^{\mathrm{kJ}}/_{\mathrm{mol\cdot nm^2}}$
k_θ	Force constant (bending vibrations)	$^{\mathrm{kJ}}/_{\mathrm{mol\cdot rad^2}}$
m	Atomic mass	u
p	Pressure	bar
q	Charge	e
$\mathbf{r}$	Position vector or distance	nm
t	Time	ps
T	Temperature	K
$\mathbf{u}$	Velocity vector	$^{\mathrm{nm}}/_{\mathrm{ps}}$
V	Potential	$^{\mathrm{kJ}}/_{\mathrm{mol}}$
ε	Energy minimum depth (Lennard-Jones)	$^{\mathrm{kJ}}/_{\mathrm{mol}}$
σ	Characteristic atomic diameter (Lennard-Jones)	nm
ϕ	Dihedral angle	rad
θ	Bending angle	rad
Definitions and Abbreviations		
AMBER	Assisted Model Building with Energy Refinement force field	
Dihedral	Geometric configuration of (four points lying in) two planes.	
GROMACS	Groningen Machine for Chemical Simulations	
LJ	Lennard-Jones	
MD	Molecular dynamics	
MDEA	Methyl diethanolamine	
MEG	Monoethylene glycol	
OPLS	Optimized Potential for Liquid Systems force field	
System	In molecular dynamics, a system is a collection of a specific number of atoms with a defined geometrical shape.	
TIP4P	Model for water and its interactions.	
vdW	Van-der-Waals	
VLE	Vapor-liquid equilibrium	
VMD	Visual Molecular Dynamics. Program for visualization of coordinate and trajectory files from MD simulations.	
File Types in GROMACS		
`.edr`	File with energies as calculated in the simulation.	
`.gro`	Atom coordinate (box) file. Generated during box-making process and at last instant of time in every simulation.	
`.itp`	Force field file. Contains information about functional forms and parameters for calculation of the potential. This information is split into multiple `.itp` files as discussed in section 5.2.	
`.mdp`	Simulation settings file. Value assignment for simulation parameters such as: time step size, number of time steps, data logging frequencies for different types of generated data files, cut-off radii, boundary conditions, pressure and temperature control.	
`.pdb`	Molecular geometry file with atom coordinates and connectivity.	
`.top`	Topology file for an individual box. Contains references to all force field files and listing of all components and their number of molecules.	
`.tpr`	“Big” file from merging all necessary files before starting the simulation.	
`.trr`	Trajectory file with atom coordinates, velocities and forces. The frequency of data logging is defined in the `.mdp` simulation settings.	
`.xtc`	Compressed trajectory file.	
`.xvg`	Result file from data analysis.	

1 ps = 10^{-12} s
1 nm = 10^{-9} m
1 u = $1.660539040 \cdot 10^{-27}$ kg (“unified atomic mass unit”)
1 e = $1.6021766208 \cdot 10^{-19}$ C (“elementary charge”)

1 Introduction

Molecular simulations offer a way to generate thermodynamic data sets without any experimental contribution, providing access to information that could otherwise not be obtained without high effort or risk. The *Molecular Dynamics* (MD) and the *Monte Carlo* (MC) techniques are the two main approaches to carry out such computer experiments.

In contrast to stochastic *Monte Carlo* equilibrium simulations, *Molecular Dynamics* are based on an explicit time-evaluation of the dynamical behavior of a molecular system. This gives a route to transport properties as well as time-dependent responses to perturbations [2]. In molecular simulations, a system is often called a “box”, since the preparation process includes the definition of a bounded volume and insertion of a certain amount of molecules into this volume. The subsequent simulation computes how these molecules behave in this box. Finally, desired properties such as density or viscosity can be derived from the simulated motion of the atoms. The aim of this report is to give a short introduction to MD simulations and present its application to a “simple” vapor-liquid equilibrium calculation using the software package GROMACS (Groningen Machine for Chemical Simulations).

2 Basic Principles of MD Simulations

Molecular Dynamic simulations are based on an evaluation of Newton’s equations of motion for every atom i in a molecular system:

$$\frac{d}{dt}\mathbf{r}_i = \mathbf{u}_i \qquad \text{and} \qquad m_i \cdot \frac{d}{dt}\mathbf{u}_i = \mathbf{f}_i \qquad\qquad ,\forall\ i = 1,...,N \tag{2.1}$$

with the atomic mass m_i, its position $\mathbf{r}_i$, velocity $\mathbf{u}_i$ and the force $\mathbf{f}_i$ that causes the atom to move in space. This force can be calculated from the potential energy $V(\mathbf{r}^N)$ which depends on all positions of the N atoms in the system:

$$\mathbf{f}_i = -\frac{\partial}{\partial \mathbf{r}_i} V(\mathbf{r}^N) \tag{2.2}$$

with:

$$\mathbf{r}^N = (\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N)$$

and has to be calculated individually for every atom i. It should be emphasized that the equations of motion are set up for atoms and not for molecules. However, molecular structures are introduced by the potential.

The potential energy function $V(\mathbf{r}^N)$ accounts for both intermolecular and intramolecular interactions. Intermolecular forces occur between non-bonded molecules that interact through the electrostatic field of their protons and electrons. Intramolecular forces, in contrast, occur within molecules where associated atoms cannot move independently due to molecular bonding. Within these two categories, different phenomena contribute to the potential energy function $V(\mathbf{r}^N)$:

$$V(\mathbf{r}^N) = V_{nb}(\mathbf{r}^N) + V_b(\mathbf{r}^N) \tag{2.3}$$

2.1 Non-bonded interactions

Even though physically not correct, interactions between non-bonded atoms are often assumed to be pair-additive. This means that contributions to V resulting from interaction between three or more atoms, e.g. induction, are neglected or accounted for by a "fudge factor". The "non-bonded" contribution to the potential energy function then becomes:

$$V_{nb}(\mathbf{r}^N) = \sum_{i=1}^{N} \sum_{j>i}^{N} V_{ij}(\mathbf{r}_i, \mathbf{r}_j) \quad . \tag{2.4}$$

Two common models for modeling intermolecular interactions are the *Coulomb potential* and the *Lennard-Jones potential* [11]. As shown in figure 2.1 (right), a typical non-bonded potential possesses a zero-gradient minimum, i.e. zero force. Two atoms in this configuration neither attract nor repel each other.

Coulomb interactions occur between two (partially) charged atoms, as can be found in ions or multipolar[1] molecules. The related potential function contribution has the form:

$$V_{ij}^{Coulomb} = \underbrace{\frac{1}{4\pi \cdot \varepsilon_0}}_{=f} \cdot \frac{q_i \cdot q_j}{r_{ij}} \quad . \tag{2.5}$$

The factor[2] f is referred to as *electric conversion factor*, q_i and q_j (unit: e) are the partial charges of the atoms i and j. For many atomic groups these partial charges can be found in literature. Otherwise, they can be determined by means of quantum mechanical calculations.

The **Lennard-Jones potential** accounts for two phenomena:

1. Van-der-Waals interactions, present at intermediate distances.
2. Repulsion, a phenomenon that occurs, if two atoms approach so closely that their electron shells overlap.

The (empirical) Lennard-Jones potential for these contributions can be stated as:

$$V_{ij}^{LJ} = 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right] \tag{2.6}$$

---

[1] Molecules constituted of atoms with different electronegativity.

[2] $f = 138.935458\,\mathrm{kJ\,mol^{-1}\,nm\,e^{-2}}$

and is visualized in figure 2.1 (left). As can be seen from the figure, parameter ε (kJ/mol) corresponds to the depth of the energy minimum, whereas the parameter σ (nm) can be thought of as atom specific diameter. The two quantities ε_{ij} and σ_{ij} are mean values and normally calculated as:

$$\varepsilon_{ij} = \sqrt{\varepsilon_i \cdot \varepsilon_j} \tag{2.7}$$

$$\sigma_{ij} = \sqrt{\sigma_i \cdot \sigma_j} \quad \text{(OPLS)} \qquad \text{or} \qquad \sigma_{ij} = \frac{1}{2}(\sigma_i + \sigma_j) \quad \text{(AMBER)}$$

for the interaction of atoms i and j. The quantities σ_i, σ_j and ε_i, ε_j are model parameters and have to be fitted for each atomic group i and j.

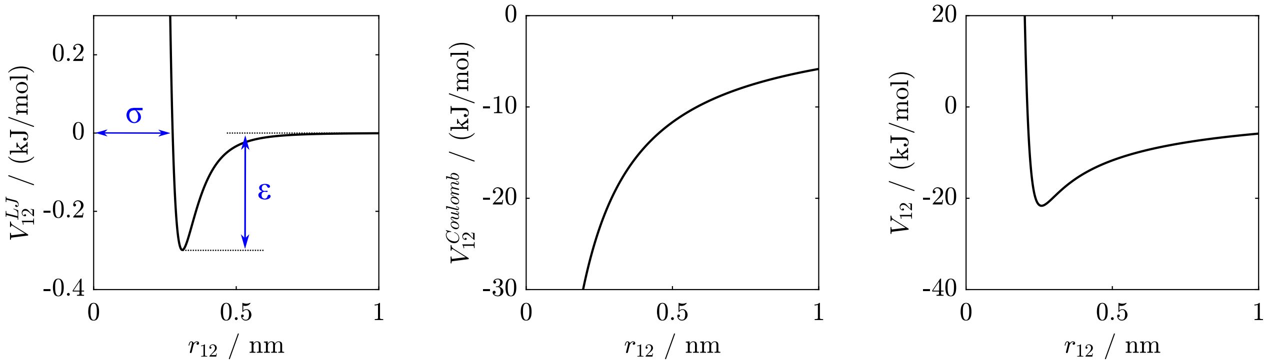


Figure 2.1: Lennard-Jones potential (left), Coulomb potential (middle) and sum of both potentials (right) for the interaction between O in ethylene glycol and H in methane. Parameters taken from OPLS force field. $\sigma_{12} = 0.277\,\text{nm}$ and $\varepsilon_{12} = 0.2988\,\text{kJ} \cdot \text{mol}^{-1}$ calculated as in eqn. 2.7; $q_O = -0.7\,\text{e}$, $q_H = 0.06\,\text{e}$.

2.2 Bonding Potentials

Atoms in molecules are linked by bonds and can thus not move independently. Moreover, atoms in the same molecules interact differently than the LJ and Coulomb approaches would suggest, leading to special bonding potentials. Different types of bonding potentials are:

- Stretching vibrations of bonds
- Bending vibrations of bonds (in molecules of 3 or more atoms)
- Internal rotation/torsion (in molecular chains of 4 or more atoms)

The potential function for internal rotation is formulated as the sum of 4-atom (1-2-3-4) chain rotations in a molecule. This means that rotations in longer chains are split into multiple 4-atom (i,j,k,l) chain interactions:

$$V_{torsion} = \sum_{\{i,j,k,l\}} V_{ijkl} \tag{2.8}$$

Since the 1-2-3 atom chain and 2-3-4 chain form two geometrical planes, the potential contribution for internal rotation is formulated as function of the angle between these two planes, the so-called “dihedral angle” ϕ, see[3] figure 2.2. Consequently, the geometrical configuration of the atoms in a 4-atom chain is termed “dihedral”.

As is known, rotation around multiple bonds, e.g. the double bond in ethene, is chemically not possible. To prevent rotation around such bonds in a simulation, “improper dihedrals” can be defined to force all four atoms to stay in one plane, i.e. in *cis* or *trans* configuration. The dihedral angle of improper dihedrals is then kept constant in the simulation. The introduced terms are essential to understand the force field files presented in section 5.

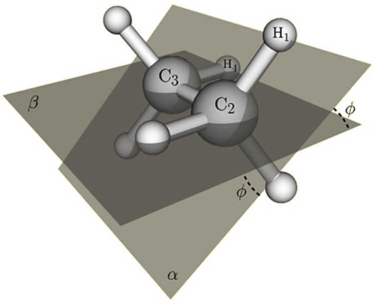


Figure 2.2: Dihedral angle[3] ϕ of H$_1$-C$_2$-C$_3$ and C$_2$-C$_3$-H$_4$ planes in ethane.

Typically, harmonic potentials are used for all types of intramolecular interactions listed above. These are formulated as functions of a displacement length or bending/torsion angles, cf. equation 2.9. For further discussion on bonding potentials see also appendix A.1 .

2.3 Force fields

Since there are lots of different approaches to modeling the contributions to the potential function V, one has to decide which models to apply in a simulation. For example, Lennard-Jones is not the only model that could be used to describe van-der-Waals interactions and repulsion. A **force field** can be understood as such a “decision”. In general, it consists of:

- a set of **functional terms** (Lennard-Jones, Coulomb, harmonic vibrations,...),
- **model parameters** for all these functional terms (σ_i, ε_i, q_i, ...).

A frequently applied force field is the *optimized potential for liquid systems* (OPLS) force field [3, 9] which was introduced to model liquid mixtures of organic compounds. The resulting potential function for the OPLS force field is exemplarily stated in equation 2.9. It will be applied in the simulations in section 5.

$$\begin{aligned} V(\mathbf{r}^N) &= \sum_{bonds} k_b(r-r_0)^2 + \sum_{angles} k_\theta(\theta-\theta_0)^2 \\ &+ \sum_{torsion} \left\{ \frac{V_1}{2}[1+\cos(\phi)] + \frac{V_2}{2}[1-\cos(2\phi)] + \frac{V_3}{2}[1+\cos(3\phi)] + \frac{V_4}{2}[1-\cos(4\phi)] \right\} \\ &+ \sum_i \sum_{j>i} \left\{ V_{ij}^{Coulomb}(r_{ij}) + V_{ij}^{LJ}(r_{ij}) \right\} \end{aligned} \tag{2.9}$$

---

[3] Source: `http://quantumwise.com/documents/manuals/VNL-2008.10_Manual/chap.molbuilder.html`

3 Simplifications and Computational Aspects

The equations of motion for every atom (eqn. 2.1) can be evaluated by insertion of several Taylor series approximations for velocity and position that lead to a discretization followed by step-by-step solution in time. This strategy allows for easy parlallelization of the computation process from one sampling point so the next. However, the updated positions of all atoms have to be merged after each step in order to calculate the new potential energy function. The described strategy is referred to as *leap-frog algorithm* or *Verlet algorithm*. A more detailed description of the algorithm can be found appendix A.3 and in [1, 2], but is not important for the subsequent chapters.

3.1 Periodic boundary condition

Since the duration of a simulation run scales with the number of molecules, it is desirable to simulate as few molecules as possible. If only bulk properties are of interest, walls and thereby wall effects should be excluded from the simulation domain. This is possible by introduction of the so-called *periodic boundary condition* or *periodic walls*. If an atom "hits" a periodic wall, it is removed and simply re-inserted at the opposite wall with the same velocity and direction as before. This leads to a bulk-like behavior even at the boundary of the simulation domain.

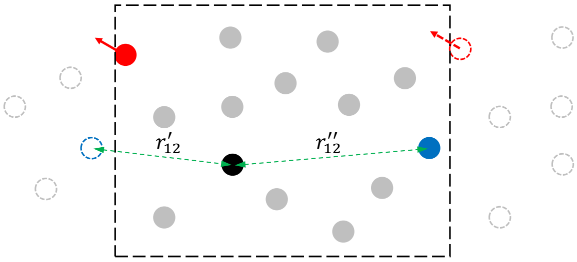


Figure 3.1: Periodic boundary conditions and minimum-image convention.

The application of periodic boundary conditions has some consequences that must be taken into account:

1. If atoms can move through periodic walls, they can also interact through them. This leads to the *minimum-image convention* which states that an atom always interacts with the closest "image". It is visualized in figure 3.1: $r'_{12} < r''_{12}$, i.e. V_{12} is calculated through the periodic boundary.
2. Application of periodic boundaries allows to reduce the number of molecules in the system (make a smaller box). However, this reduction is still not unrestricted. In a system with strong long-range interactions such as ionic systems, periodicity and a too small box introduce a big error, since the atoms would interact with lots of remote atoms but there are no such atoms present[4].

---

[4] Methods like *Ewald sum*, PME, PPPM can reduce this problem.

3.2 Cutoff radius and neighbor list

From figure 2.1 it can be found that the van-der-Waals and repulsion interactions (Lennard-Jones) are only of short-range significance. For high distances the potential stays constant and does therefore not contribute to the force $\mathbf{f}_{12}$. With this in mind, it seems undesirable to calculate V_{ij}^{LJ} for every pair of atoms, but rather intuitive to restrict the evaluation of V_{ij}^{LJ} to atoms j in immediate vicinity of i. This restriction is called *cutoff radius*, the resulting set of atoms j within the cutoff radius the *pair list*. In principle, the same applies to Coulomb interactions, even though these are of longer range.

$$V_{nb}^{LJ}(\mathbf{r}^N) = \sum_{i} \sum_{j>i \wedge r_{ij}<r_{cutoff}^{LJ}} V_{ij}^{LJ}(r_{ij}) \tag{3.1}$$

3.3 Simulation context (thermodynamic ensembles)

As known from classical thermodynamics, every system requires specification of its number of particles N (or chemical potential) as well as two more state variables in order to be definite. Molecular simulations are typically carried out for systems, in which those three quantities are kept constant during the entire simulation run. This is not surprising since we are usually interested in thermodynamic properties, e.g. density, at a given point in state space, e.g. given temperature and pressure. However, this means that we have to ensure these properties stay constant during the simulation.

Note that the number of molecules N is a “degree of freedom” of the modeling process. It must be chosen appropriately; big enough to reduce statistical errors, but at the same time small enough to keep the simulation time within acceptable limits. Two important types of such simulations contexts applied in this project are listed below:

NVT simulation: This corresponds to a simulation of a fixed number of molecules in a box with constant dimensions. As known from molecular thermodynamics, the macroscopic temperature of a fluid is related to its kinetic energy and thus to the velocity of all atoms. Therefore, to keep the macroscopic temperature of the system constant, the velocity of all atoms is rescaled in the simulation. A simulation environment that keeps the temperature constant is termed a *thermostat*.

NpT simulation: In such a simulation, the pressure is kept constant besides the temperature. Thus, the simulator is a combination of *thermostat* and so-called *barostat*. The pressure is manipulated by constantly rescaling one or multiple dimensions of the box, i.e. all atom coordinates, during the simulation.

Note also that “simulation contexts” are often referred to as “ensembles”, a term from statistical thermodynamics. An extensive discussion on the concept of ensembles is presented by Lucas [11], but is not important for this project.

3.4 GROMACS

GROMACS is a free available open-source software package designed for molecular dynamics simulations as well as subsequent analysis and evaluation of generated data. It was developed in the 90s at the university of Groningen (Sweden) and is today maintained by universities and research centers worldwide. It incorporates an implementation of all principles discussed in the sections above. A practical introduction to GROMACS is given in section 5. More detailed descriptions of the different methods and commands can be found in the GROMACS user manual [1] and on the GROMACS web page[5]. The Linux version of GROMACS will be used within this project. Since there is no graphical user interface, commands are inserted through the console only.

3.5 Length and time scales in molecular dynamics

As shown in figure 2.1, molecular phenomena occur on length scales of nanometers. By convention, this is the universal unit in which all coordinates and dimensions are stated in GROMACS. Likewise, the standard velocity scale is $\mathrm{nm\,ps^{-1}} = 10^3\,\mathrm{nm\,ns^{-1}}$. Typically, atom velocities in the range of $0.01 \dots 1.0\,\mathrm{nm\,ps^{-1}}$ can be observed in MD simulations, cf. fig 5.13. However, since liquid molecules are very inhibited in their motion, their effective velocity is up to factor thousand lower.

As discussed previously, the number of molecules is a degree of freedom in the modeling procedure. A typical number is several hundreds to thousands molecules, depending on the constitution of the system (homogeneous, multi-phase), its composition and the size of the molecules. Usually the number of small solvent molecules lies in the range of several thousands, whereas in biochemical simulations often only one macro-molecule like a protein is inserted into the box. In this project, the resulting total number of atoms was around 70.000 in a $5\,\mathrm{nm} \times 5\,\mathrm{nm} \times 82.2\,\mathrm{nm}$ two-phase box with a liquid phase of circa 15 nm thickness. In general, MD simulation times for full equilibration can range from only a few picoseconds up to multiple microseconds, depending on system size and constitution. For numerical stability and accuracy, the time step size must be chosen according to the smallest time scale, usually bond vibrations ($\approx 10^{-3}\,\mathrm{ps}$). A common time step size is thus $10^{-4} \dots 10^{-3}\,\mathrm{ps}$.

4 Introduction to the Case Study

The aim of this project is to apply the theory discussed in sections 2 and 3 to a problem that is of practical relevance for chemical engineering. The chosen problem is presented in figure 4.1. The purpose of the MD simulations is to generate a set of distribution coefficients of CH_4 and H_2S for different temperatures and pressures. MEG and MDEA are absorption agents that hold the water back in the liquid phase and enhance the absorption of H_2S, respectively.

---

[5] `http://www.gromacs.org/` and `http://manual.gromacs.org/programs/byname.html`

The H_2S is removed from the gas phase through physical and reactive absorption. However, the reactive contribution is not modeled in this MD project, see also appendix A.4.

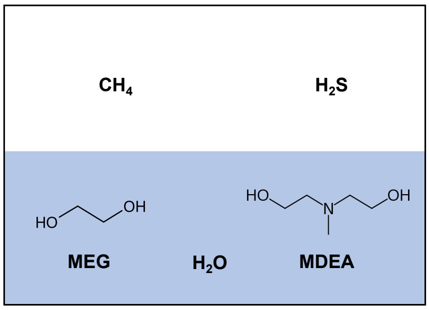


Figure 4.1: Two-phase system of methane (CH_4), hydrogen sulfide (H_2S), ethylene glycol (MEG), methyl diethanolamine (MDEA) and water. Objective: Absorption of hazardous component H_2S from a raw gas stream.

4.1 Simulation strategy

Since this project deals with a two-phase equilibrium, we first have to ensure that the molecules form a two-phase system during the simulation. An intuitive idea could be to define an overall composition as well as box volume (density) and temperature in the two-phase region, insert all molecules into one big box and simply run the simulation. Major drawback of this strategy is that the formation of a two phase system might take a while in a simulation. Furthermore, we cannot guarantee that one big liquid phase is obtained. However, since the aim of the simulation is to generate bulk properties, one big liquid phase is desired. For that reason, the (better) strategy followed in this project is to, first, generate a liquid phase box and a gas box and, second, connect them to a big box.

Boundary conditions will be applied for all three spatial directions to exclude wall effects and thereby reduce the size of the system. This requires a phase-symmetric box, i.e. an atom “leaving” the gas phase through a periodic wall must be inserted into the gas phase again. Therefore, we have to generate two gas boxes that are placed on both sites of the liquid box. This idea is visualized in figure 4.2.

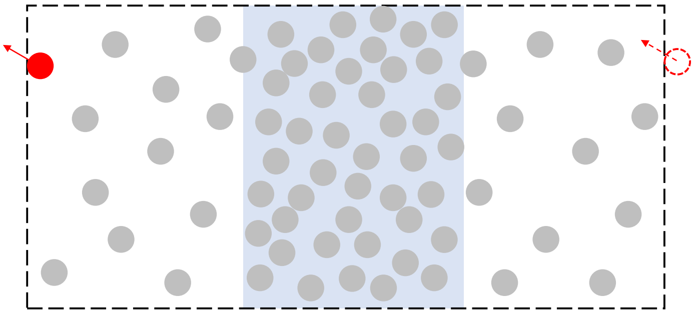


Figure 4.2: Simulation strategy for a two-phase system with periodic boundary conditions.

5 Molecular Simulations with GROMACS

The basic procedure for performing a molecular simulation in GROMACS is outlined in figure 5.1, together with the most important file types that are produced in these steps. Note that positions and velocities of all atoms are the main outcome of MD simulations as they result from solution of the equations of motion. Based on these trajectory data, all thermodynamic quantities are calculated in a subsequent analysis step. Molecular simulations are a very straightforward procedure; not complicated but with lots of details to be taken into account. The different steps and related commands are explained in the next sections.

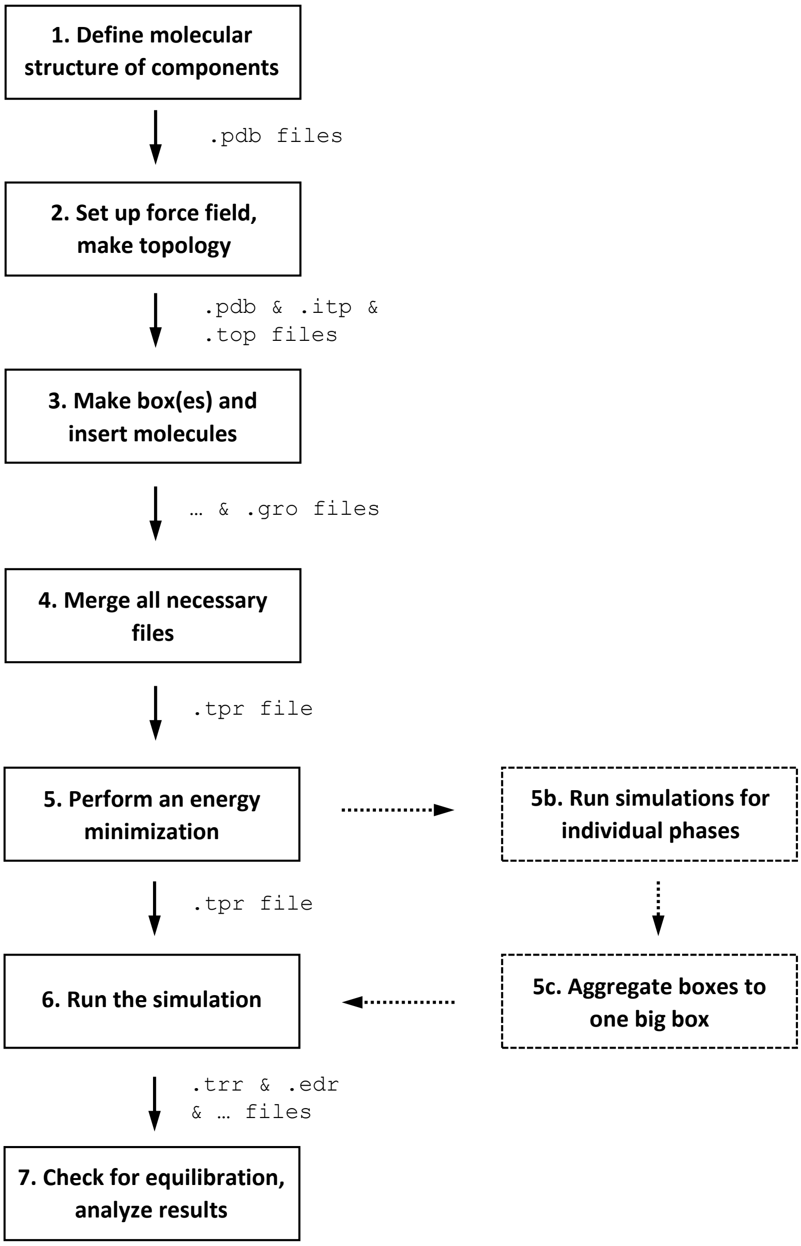


Figure 5.1: Basics steps of a GROMACS simulation and the most important file types that are passed along the steps.

5.1 Step 1 – Definition of molecular structures

The starting point of any molecular simulation is the definition of all chemical components that are part of the system. This can be easily done with a program like Avogadro[6].

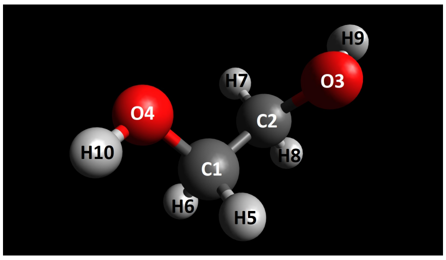
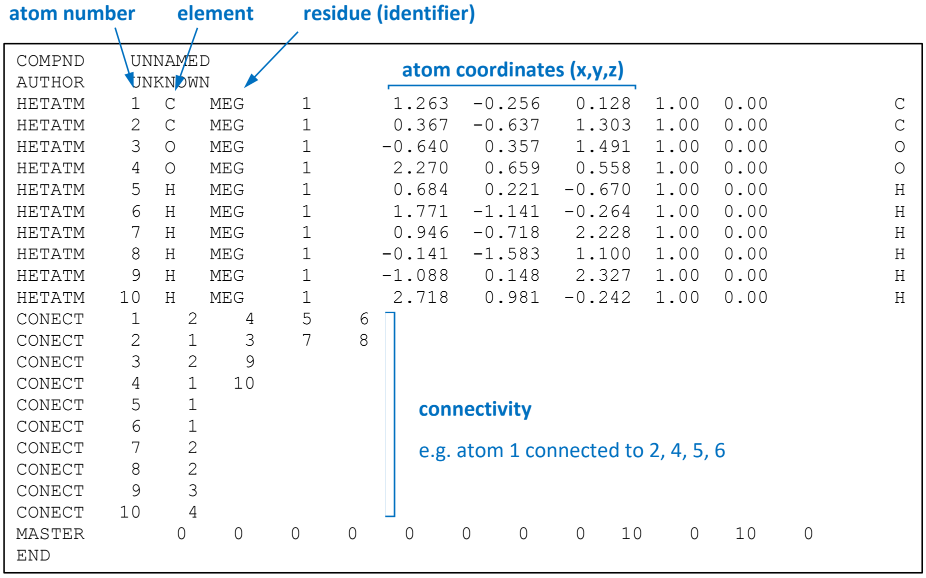


Figure 5.2: Illustration of an ethylene glycol (MEG) molecule in Avogadro after geometrical optimization. SMILES code: C(CO)O

Such programs usually allow for fast building of molecular structures through insertion of molecular codes such as the *Simplified molecular-input line-entry system* (SMILES). If necessary, the molecular structure of a component can be edited afterwards. In order to generate a reasonable initial arrangement of the atoms in a molecule, it is recommended to perform a simple geometry optimization as provided by Avogadro. Further, it is advantageous to rename the “residue” `RES` to a unique identifier, e.g. `MEG` for ethylene glycol. This enables an easy identification of the molecules in the evaluation process and for plotting. Finally, the structure is saved as `.pdb` file that contains information about the type of atoms in a molecule, their positions and connectivity. An exemplary `.pdb` file is given in figure 5.3.



```
COMPND    UNNAMED
AUTHOR    UNKNOWN
HETATM    1  C   MEG     1       1.263  -0.256   0.128  1.00  0.00           C
HETATM    2  C   MEG     1       0.367  -0.637   1.303  1.00  0.00           C
HETATM    3  O   MEG     1      -0.640   0.357   1.491  1.00  0.00           O
HETATM    4  O   MEG     1       2.270   0.659   0.558  1.00  0.00           O
HETATM    5  H   MEG     1       0.684   0.221  -0.670  1.00  0.00           H
HETATM    6  H   MEG     1       1.771  -1.141  -0.264  1.00  0.00           H
HETATM    7  H   MEG     1       0.946  -0.718   2.228  1.00  0.00           H
HETATM    8  H   MEG     1      -0.141  -1.583   1.100  1.00  0.00           H
HETATM    9  H   MEG     1      -1.088   0.148   2.327  1.00  0.00           H
HETATM   10  H   MEG     1       2.718   0.981  -0.242  1.00  0.00           H
CONECT    1    2    4    5    6
CONECT    2    1    3    7    8
CONECT    3    2    9
CONECT    4    1   10
CONECT    5    1
CONECT    6    1
CONECT    7    2
CONECT    8    2
CONECT    9    3
CONECT   10    4
MASTER        0    0    0    0    0    0    0    0   10    0   10    0
END
```

MEG.pdb

Figure 5.3: The `.pdb` file of the ethylene gylcol (MEG) molecule.

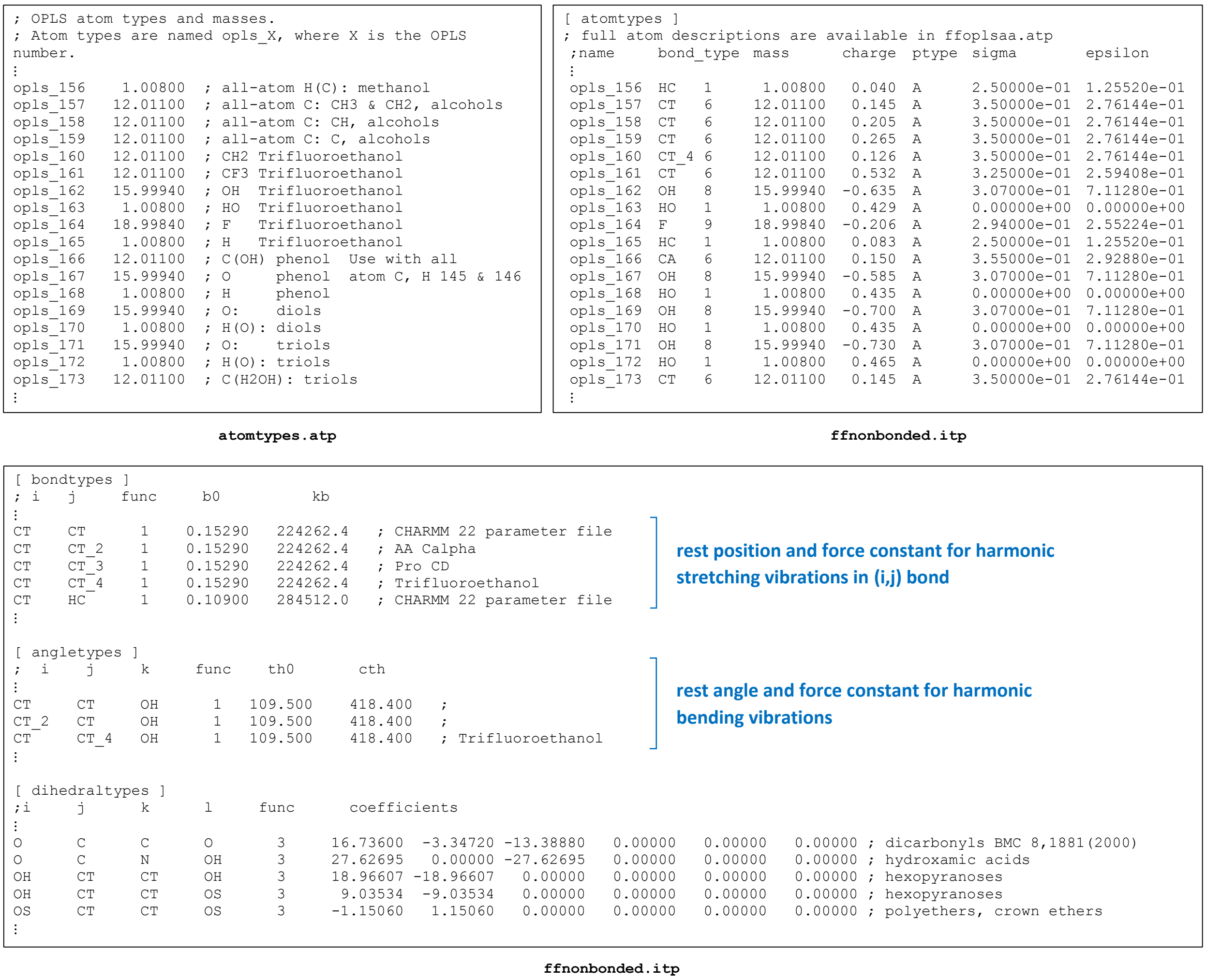
---

[6] Freeware, downloadable on: `https://avogadro.cc/`

5.2 Step 2 – Force field and topology

The subsequent step is concerned with the set-up of the force field and generation of the “topology” and can be considered the key step in preparing a simulation, because it strongly influences the quality of the results.

As mentioned previously, the OPLS force field will be applied within this project. It is valid for gas phase simulations as well as for the liquid phase. Set-up of the force field means that the parameters in the OPLS potential function $V(\mathbf{r}^N)$ (cf. eqn. 2.9) have to be defined for all “atom types”. Since, for example, a carbon atom in a methyl group interacts with other atoms in a different way than a carbon atom in CO_2, in the literature, parameters of force fields are stated for such “atom types” instead of elements. As a consequence, in this step we have to determine the correct type of every atom in all molecules. In recent years, parameter sets for a large number of atom types have been published in literature. At the same time, lots of authors have published insights about what atom types are appropriate for simulation of various molecules. This facilitates the set-up procedure considerably.



```
; OPLS atom types and masses.
; Atom types are named opls_X, where X is the OPLS
number.
⋮
opls_156    1.00800  ; all-atom H(C): methanol
opls_157   12.01100  ; all-atom C: CH3 & CH2, alcohols
opls_158   12.01100  ; all-atom C: CH, alcohols
opls_159   12.01100  ; all-atom C: C, alcohols
opls_160   12.01100  ; CH2 Trifluoroethanol
opls_161   12.01100  ; CF3 Trifluoroethanol
opls_162   15.99940  ; OH  Trifluoroethanol
opls_163    1.00800  ; HO  Trifluoroethanol
opls_164   18.99840  ; F   Trifluoroethanol
opls_165    1.00800  ; H   Trifluoroethanol
opls_166   12.01100  ; C(OH) phenol  Use with all
opls_167   15.99940  ; O     phenol  atom C, H 145 & 146
opls_168    1.00800  ; H     phenol
opls_169   15.99940  ; O:    diols
opls_170    1.00800  ; H(O): diols
opls_171   15.99940  ; O:    triols
opls_172    1.00800  ; H(O): triols
opls_173   12.01100  ; C(H2OH): triols
⋮
```

atomtypes.atp

```
[ atomtypes ]
; full atom descriptions are available in ffoplsaa.atp
 ;name     bond_type  mass      charge  ptype  sigma        epsilon
 ⋮
 opls_156  HC    1      1.00800   0.040  A      2.50000e-01  1.25520e-01
 opls_157  CT    6     12.01100   0.145  A      3.50000e-01  2.76144e-01
 opls_158  CT    6     12.01100   0.205  A      3.50000e-01  2.76144e-01
 opls_159  CT    6     12.01100   0.265  A      3.50000e-01  2.76144e-01
 opls_160  CT_4  6     12.01100   0.126  A      3.50000e-01  2.76144e-01
 opls_161  CT    6     12.01100   0.532  A      3.25000e-01  2.59408e-01
 opls_162  OH    8     15.99940  -0.635  A      3.07000e-01  7.11280e-01
 opls_163  HO    1      1.00800   0.429  A      0.00000e+00  0.00000e+00
 opls_164  F     9     18.99840  -0.206  A      2.94000e-01  2.55224e-01
 opls_165  HC    1      1.00800   0.083  A      2.50000e-01  1.25520e-01
 opls_166  CA    6     12.01100   0.150  A      3.55000e-01  2.92880e-01
 opls_167  OH    8     15.99940  -0.585  A      3.07000e-01  7.11280e-01
 opls_168  HO    1      1.00800   0.435  A      0.00000e+00  0.00000e+00
 opls_169  OH    8     15.99940  -0.700  A      3.07000e-01  7.11280e-01
 opls_170  HO    1      1.00800   0.435  A      0.00000e+00  0.00000e+00
 opls_171  OH    8     15.99940  -0.730  A      3.07000e-01  7.11280e-01
 opls_172  HO    1      1.00800   0.465  A      0.00000e+00  0.00000e+00
 opls_173  CT    6     12.01100   0.145  A      3.50000e-01  2.76144e-01
 ⋮
```

ffnonbonded.itp

```
[ bondtypes ]
; i   j     func     b0          kb
⋮
CT    CT      1    0.15290   224262.4   ; CHARMM 22 parameter file
CT    CT_2    1    0.15290   224262.4   ; AA Calpha
CT    CT_3    1    0.15290   224262.4   ; Pro CD
CT    CT_4    1    0.15290   224262.4   ; Trifluoroethanol
CT    HC      1    0.10900   284512.0   ; CHARMM 22 parameter file
⋮
```

rest position and force constant for harmonic stretching vibrations in (i,j) bond

```
[ angletypes ]
;  i    j     k     func    th0       cth
⋮
CT     CT     OH      1   109.500    418.400   ;
CT_2   CT     OH      1   109.500    418.400   ;
CT     CT_4   OH      1   109.500    418.400   ; Trifluoroethanol
⋮
```

rest angle and force constant for harmonic bending vibrations

```
[ dihedraltypes ]
;i      j      k      l      func      coefficients
⋮
O      C      C      O        3     16.73600  -3.34720 -13.38880   0.00000   0.00000   0.00000 ; dicarbonyls BMC 8,1881(2000)
O      C      N      OH       3     27.62695   0.00000 -27.62695   0.00000   0.00000   0.00000 ; hydroxamic acids
OH     CT     CT     OH       3     18.96607 -18.96607   0.00000   0.00000   0.00000   0.00000 ; hexopyranoses
OH     CT     CT     OS       3      9.03534  -9.03534   0.00000   0.00000   0.00000   0.00000 ; hexopyranoses
OS     CT     CT     OS       3     -1.15060   1.15060   0.00000   0.00000   0.00000   0.00000 ; polyethers, crown ethers
⋮
```

ffnonbonded.itp

Figure 5.4: Extract of parameter files from the OPLS folder in GROMACS.

An assembly of force field parameters from the literature can be found in the OPLS files `atomtypes.atp`, `ffnonbonded.itp` and `ffbonded.itp` that are part of the GROMACS library. While the former contains descriptions of the atom types, the latter two files list the parameters for the non-bonded and bonded potential functions, respectively. An extract of the files is presented in figure 5.4. The `opls_X` number denotes the atom type and is used to find the respective parameters in `ffnonbonded.itp`. Note also that the `bond_type` attribute in `ffnonbonded.itp` is used to find the correct parameter sets in `ffbonded.itp`.
The functional terms $V_i(\mathbf{r}^N)$ introduced in section 2 are implemented in the GROMACS standard library. For OPLS, the "decision" on which functional terms to use is listed in `forcefield.tip` in the `oplsaa.ff` folder in GROMACS' folder structure and does not need to be specified by the user.

Force field for particular system

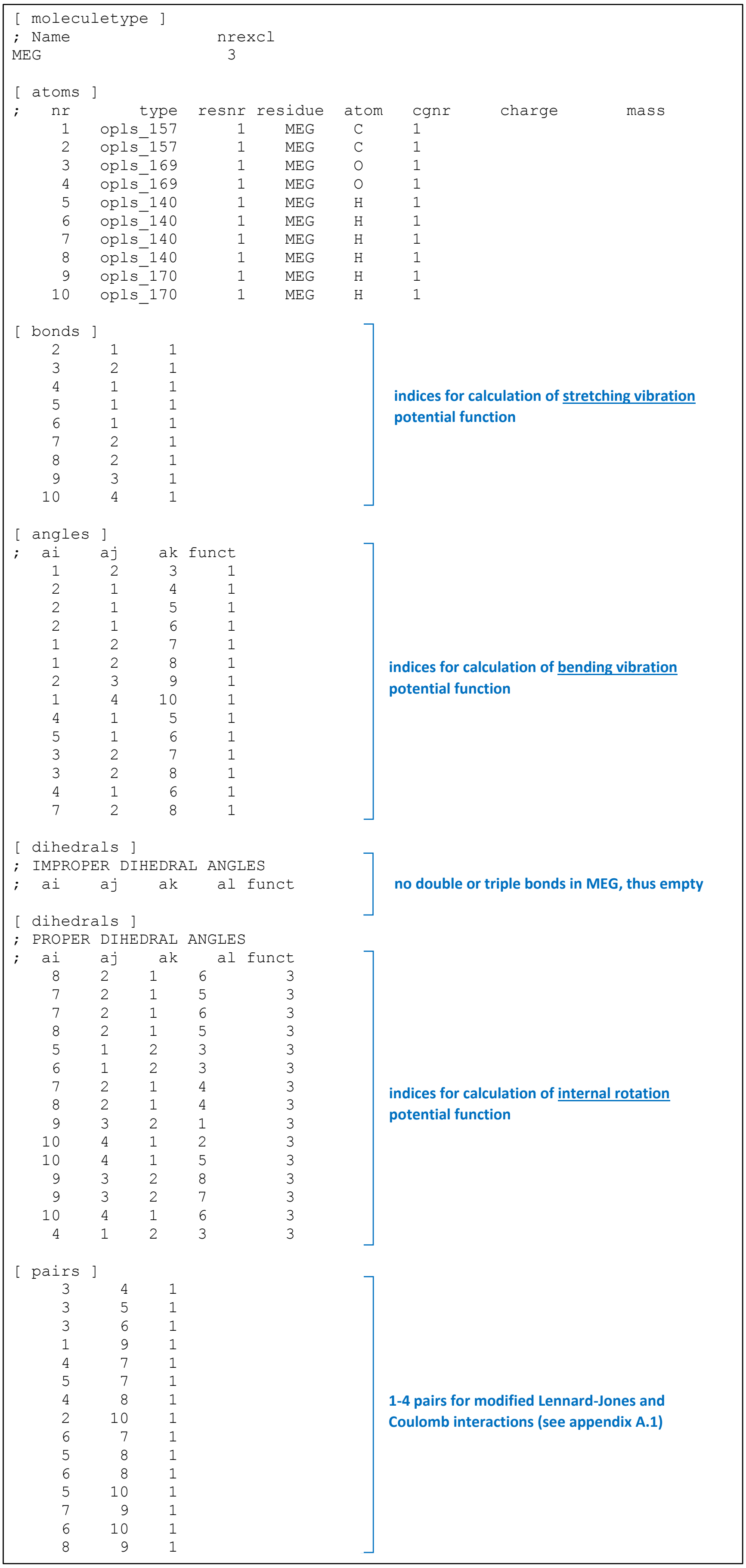
For every molecule, we have to create a `.itp` force field file that contains information about atom types and intramolecular interactions. A good starting point are scripts like LigParGen[7]. These scripts use the `.pdb` molecular geometry information from step 1 to generate a `.itp` file that has code structure as required by GROMACS. In addition, the scripts assume atom types and provide the related `opls_X` number (cf. fig. 5.4). This assumption, however, is often not correct and has to be checked and modified by the user.
Suitable atom types for CH_4 and H_2S were found in `atomtypes.atp` directly. For water, the TIP4P model [7] is used that is compatible with OPLS and already implemented in the OPLS folder. Atom types for MEG and MDEA were found in [4] and [5]. In a final step, the total charge of the molecule must be checked for the chosen atom types and their related partial charges in `ffnonbonded.itp`. If the total charge is nonphysical, partial charges must be recalculated applying quantum-mechanical methods.
The complete `.itp` force field file for MEG is stated in figure 5.5 with already modified `opls_X` references. As can be seen, it is not necessary to list the force field parameters explicitly in this file. However, if necessary, modified parameters can be included in additional columns, see [1]. Note also that the category `funct` enables the user to implement modification of the force field functional forms. The OPLS-specific default value is set by the above-mentioned scripts and should not be changed unless expressly intended.

Topology

The **topology** (`.top`) file can be regarded as data set that incorporates all *constant attributes* of a molecular system, i.e. references to all force field parameter files, the functional terms of the force field and the number of molecules of all species as will be inserted during step 3, figure 5.6. In contrast, *dynamic attributes*, such as positions and velocities that are generated

---

[7] LigParGen: `http://zarbi.chem.yale.edu/ligpargen/`



```
[ moleculetype ]
; Name                 nrexcl
MEG                     3

[ atoms ]
;   nr       type  resnr  residue  atom   cgnr     charge       mass
     1   opls_157      1     MEG     C      1
     2   opls_157      1     MEG     C      1
     3   opls_169      1     MEG     O      1
     4   opls_169      1     MEG     O      1
     5   opls_140      1     MEG     H      1
     6   opls_140      1     MEG     H      1
     7   opls_140      1     MEG     H      1
     8   opls_140      1     MEG     H      1
     9   opls_170      1     MEG     H      1
    10   opls_170      1     MEG     H      1

[ bonds ]
    2     1     1
    3     2     1
    4     1     1
    5     1     1
    6     1     1
    7     2     1
    8     2     1
    9     3     1
   10     4     1

[ angles ]
;  ai    aj    ak  funct
    1     2     3     1
    2     1     4     1
    2     1     5     1
    2     1     6     1
    1     2     7     1
    1     2     8     1
    2     3     9     1
    1     4    10     1
    4     1     5     1
    5     1     6     1
    3     2     7     1
    3     2     8     1
    4     1     6     1
    7     2     8     1

[ dihedrals ]
; IMPROPER DIHEDRAL ANGLES
;  ai    aj    ak    al  funct

[ dihedrals ]
; PROPER DIHEDRAL ANGLES
;  ai    aj    ak    al  funct
    8    2    1    6        3
    7    2    1    5        3
    7    2    1    6        3
    8    2    1    5        3
    5    1    2    3        3
    6    1    2    3        3
    7    2    1    4        3
    8    2    1    4        3
    9    3    2    1        3
   10    4    1    2        3
   10    4    1    5        3
    9    3    2    8        3
    9    3    2    7        3
   10    4    1    6        3
    4    1    2    3        3

[ pairs ]
     3     4     1
     3     5     1
     3     6     1
     1     9     1
     4     7     1
     5     7     1
     4     8     1
     2    10     1
     6     7     1
     5     8     1
     6     8     1
     5    10     1
     7     9     1
     6    10     1
     8     9     1
```

indices for calculation of stretching vibration potential function

indices for calculation of bending vibration potential function

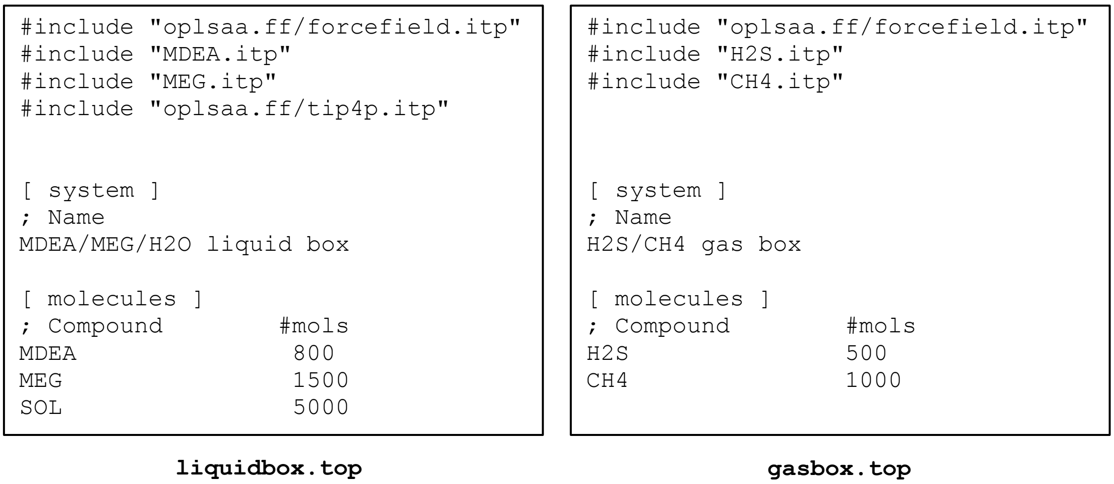
no double or triple bonds in MEG, thus empty

indices for calculation of internal rotation potential function

1-4 pairs for modified Lennard-Jones and Coulomb interactions (see appendix A.1)

Figure 5.5: The `.itp` force field file for MEG with correct `opls_X` number.

during the simulation run are stored in coordinate and trajectory files. Since the liquid and gas phase will be initialized separately, we have to generate two topologies. Note that the compound and their number of molecules must be stated in the order as they are inserted during the box-making process. This is why creating the topology might be done after step 3. For a better understanding of the topology file structure in GROMACS, the references in the topology file are visualized in figure 5.7.



```
#include "oplsaa.ff/forcefield.itp"
#include "MDEA.itp"
#include "MEG.itp"
#include "oplsaa.ff/tip4p.itp"


[ system ]
; Name
MDEA/MEG/H2O liquid box

[ molecules ]
; Compound        #mols
MDEA               800
MEG                1500
SOL                5000
```

liquidbox.top

```
#include "oplsaa.ff/forcefield.itp"
#include "H2S.itp"
#include "CH4.itp"




[ system ]
; Name
H2S/CH4 gas box

[ molecules ]
; Compound         #mols
H2S                500
CH4                1000
```

gasbox.top

Figure 5.6: Topology `.top` files (`liquidbox.top` and `gasbox.top`) for gas and liquid phase.

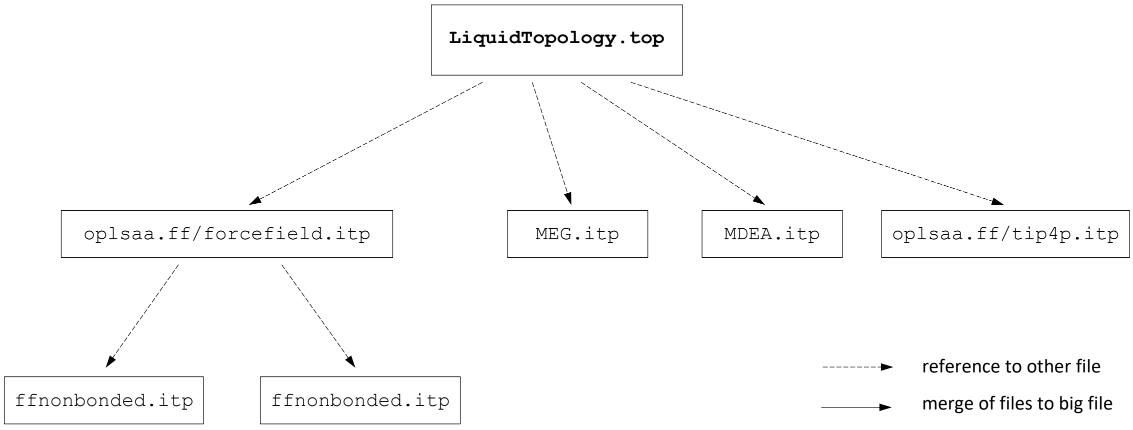


Figure 5.7: File structure of topology in GROMACS.

In this step we also have to decide on the number of molecules that will be simulated. When decreasing this number, the statistical influence on the results increases and reduces the quality of generated data. In contrast, a high number of molecules increases the computation time. This decision is thus always a trade-off and must be made with experience and refinement studies.

5.3 Step 3 – Make a box

Depending on the GROMACS version installed on the computer, GROMACS commands[8] in the console start with `gmx` or `gmx_mpi`. The command for making a box is then `gmx_mpi insert-molecules` followed by settings and specification of all input and output files which are included with code terms like `-f` or `-o`.

A box is initialized and filled step-wise for each component. The "box file" is a `.gro` position (and velocity) file that simply contains the positions of all atoms. The first chemical component is inserted with the command:

```
 gmx_mpi insert-molecules -ci MDEA.pdb -o liquid_box.gro -box 5 5 50
-nmol 800
```

This generates a 5 nm × 5 nm × 50 nm box (`liquid_box.gro`) containing 800 randomly distributed MDEA molecules with a geometrical structure as defined in `MDEA.pdb`. The insertion process is iterative and GROMACS inserts the molecules such that atom shells do not overlap. The other components are inserted into to the priorly defined box:

```
                           input  old box without MEG  molecular geometry  output
 gmx_mpi insert-molecules   -f   liquid_box.gro       -ci MEG.pdb          -o
liquid_box.gro -nmol 1500
updated box with MEG

gmx_mpi insert-molecules -f liquid_box.gro -ci tip4p.gro -o
liquid_box.gro -nmol 5000
```

Since a liquid system is characterized by high packing density, it is usually not possible to define total molecule number and box volume according to the desired density. The GROMACS procedure would then lead to a "never-ending" insertion run or abnormal termination due to massive overlap of atom shells. Instead, the recommended strategy for setting up a liquid box is to define a much bigger box (factor 10) and shrink it after the insertion procedure in a molecular simulation run (see step 5b).

As mentioned previously, the TIP4P model is applied to model water molecules. In this case, the molecular geometry of TIP4P water is defined in a `.gro` file. A brief introduction to the TIP4P model is given in Appendix A.2 together with `tip4p.gro`.

The resulting atom position file `liquid_box.gro` can be plotted to visualize the initial state, figure 5.8. The exact same procedure has to be conducted for the gas box as well, figure 5.9. However, due to the low density it is not necessary to shrink and equilibrate the gas box in an individual simulation before assembling the final two-phase box.

---

[8] A list of all GROMACS commands can be found on `http://manual.gromacs.org/programs/byname.html`

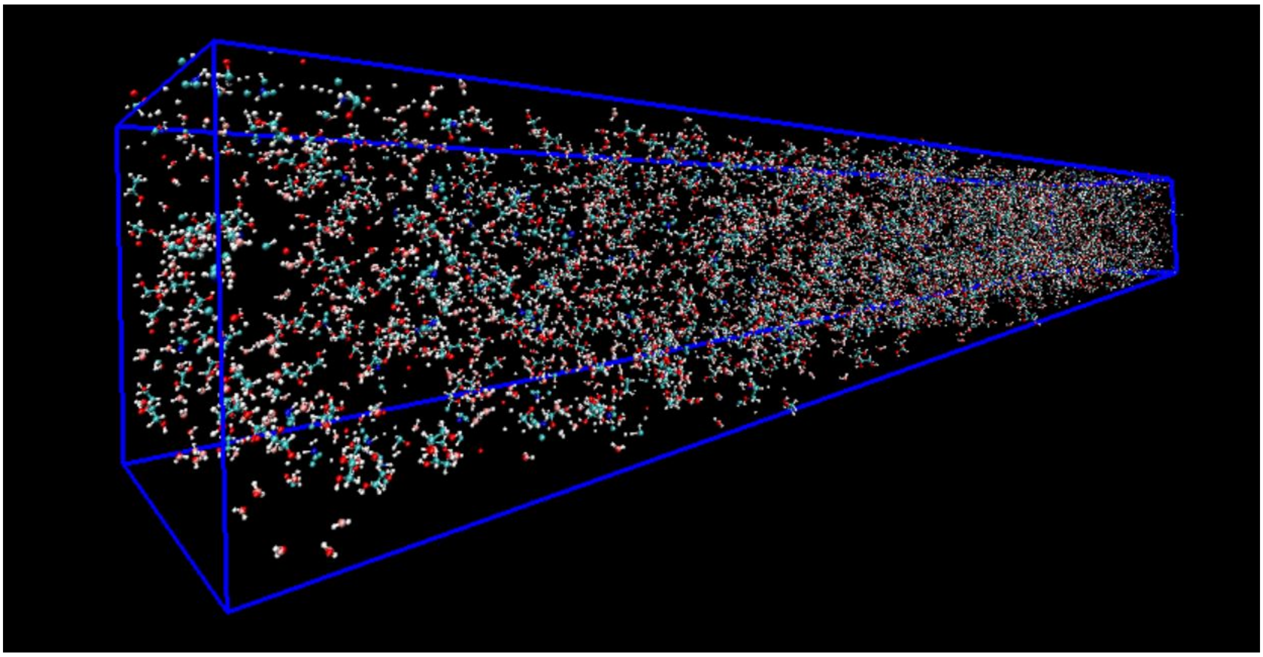


Figure 5.8: Liquid box of 5 nm × 5 nm × 50 nm after the box filling procedure. Visualization of `liquid_box.gro` with the program VMD: `http://www.ks.uiuc.edu/Research/vmd/`

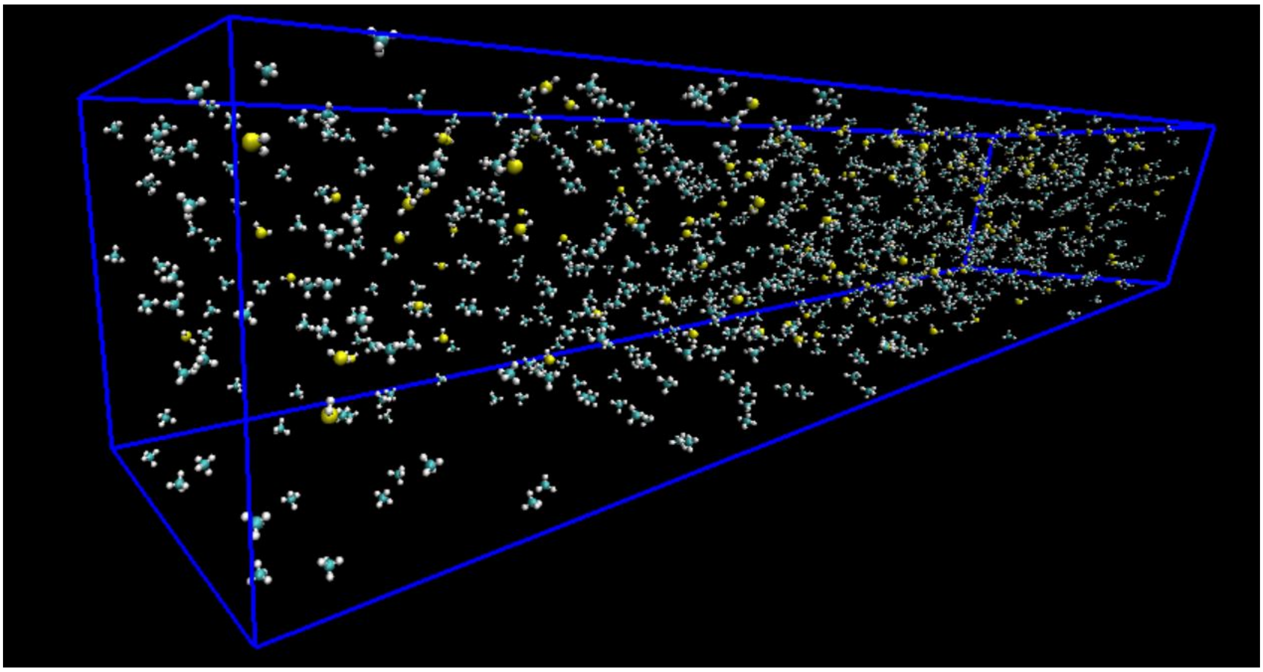


Figure 5.9: Gas box. Visualization of `gas_box.gro` in VMD.

5.4 Step 4 – Merging of files, initialization of simulation

The actual MD simulation start command in GROMACS requires a single input file that contains all necessary information. Therefore, in this step all data files are merged to one “big” `.tpr` file. Prior to this, we have to define all simulation settings. These are listed in a `.mdp` file. Two possible configurations, namely those for NpT and NVT simulations, are presented in figure A.4 and A.5 in the appendix. For equilibration of the liquid we will perform an NpT simulation, whereas the VLE final simulation is an NVT simulation. To keep this report within limits, the files are not discussed in detail. Instead, the parameters are commented to provide easy access for the interested reader.

Subsequent to setting up the simulation settings, all files are merged with the command `gmx_mpi grompp`. For example[9]:

```
                          simulation settings      initial positions of atoms
 gmx_mpi grompp -f     npt.mdp    -c   liquid_box.gro   -p
 liquidbox.top   -o liquidbox_simulation.tpr
topology (force field)           "big" output file
```

5.5 Step 5 – Energy minimization

Eventhough overlapping of atom shells is avoided, the insertion procedure can still result in problematic arrangement of molecules. Thus, an energy minimization, i.e. a special type of short and robust simulation, is helpful to find a good starting point for the actual simulation. Relevant `.mdp` simulation settings can be found in figure 5.10.

```
; minim.mdp - used as input into grompp to generate em.tpr
integrator = steep         ; Algorithm (steep = steepest descent minimization)
emtol      = 1000.0        ; Stop minimization when the maximum force < 1000.0 kJ/mol/nm
emstep     = 0.01          ; Energy step size
nsteps     = 50000         ; Maximum number of (minimization) steps to perform

; Parameters describing how to find the neighbors of each atom and how to calculate the interactions
nstlist         = 1        ; Frequency to update the neighbor list and long range forces
cutoff-scheme   = Verlet
ns_type         = grid     ; Method to determine neighbor list (simple, grid)
coulombtype     = PME      ; Treatment of long range electrostatic interactions
rcoulomb        = 1.0      ; Short-range electrostatic cut-off
rvdw            = 1.0      ; Short-range Van der Waals cut-off
pbc             = xyz      ; Periodic Boundary Conditions (yes/no)
```

`minim.mdp`

Figure 5.10: The `.mdp` file containing all simulation settings for an energy minimization simulation. Optimization: slow but robust steepest descent algorithm.

---

[9] Not part of the real simulation procedure since we perform an energy minimization prior to the actual NpT simulation (see next step).

As before, all necessary files are merged in a first step:

```
 gmx_mpi grompp -f minim.mdp -c liquid_box.gro -p liquidbox.top -o
liquidbox_minim.tpr

gmx_mpi mdrun -v -deffnm liquidbox_minim -ntomp   32   &
                         Name of .tpr file.      Number of
                        All result files will    CPU cores
                           have this name.       to be used
```

The obtained `liquidbox_minim.tpr` file is then used to start the minimization simulation with `gmx_mpi mdrun`, in this case on 32 CPU cores. The duration of energy minimization simulations is in the range of seconds or minutes rather than hours.

For example, the energy minimization before the box-shrinking simulations led to a reduction of the potential energy from 10^8 kJ/mol to $3 \cdot 10^4$ kJ/mol, i.e. a tremendous reduction of physically unfavorable molecular configurations and prevention of “kicks” at the beginning of the simulation.

5.5.1 General information on simulation

With energy minimization a typical simulation run has already been introduced. Such a simulation creates different types of files and saves them to the folder from where the simulation was started. Important result files are:

`*.gro`:	Coordinate file with atom positions and velocities at the end of the simulation.
`*.trr`:	Phase space trajectory file with atom positions and velocities (and forces) saved every `nstxout`th[10] and `nstvout`th time step, respectively.
`*.xtc`:	Compressed trajectory file, much smaller than `.trr` file. Useful for visualization.
`*.edr`:	Energy file. Important for analysis of simulation results.

The `.gro` file from a simulation can be used as initial positions for a subsequent run.

5.5.2 Step 5b – Simulation of individual phases

As described before, the liquid box was initialized to be oversized and must be shrinked to obtain a liquid phase with desired density. For that reason, this step is concerned with reduction of the box volume. Shrinking a box is realized with a barostatic simulation, where the pressure control in the MD algorithm reduces the box dimensions until the specified pressure is reached. The respective simulation settings are listed in `npt.mdp`. To accelerate the simulation, the pressure in the rescaled box dimension[11] can be defined quite high in a first simulation, e.g. 1000 bar. A second simulation is then performed for “fine-tuning” at the desired pressure.

---

[11] Dimension with non-zero compressibility, here z-direction.

Necessary commands to run the simulation are again:

```
                                     atom positions from
                                     energy minimization
 gmx_mpi grompp -f npt.mdp -c liquidbox_minim.gro -p liquidbox.top -o
liquidbox_simulation.tpr

gmx_mpi mdrun -v -deffnm liquidbox_simulation -ntomp 32 &
```

Note that box-shrinking requires small time steps (e.g. $\Delta t = 0.0001\,\mathrm{ps}$) because atoms move a lot in these simulations. A visualization of the shrinked box after a 1000 bar and subsequent 20 bar simulation is given in figure 5.11.

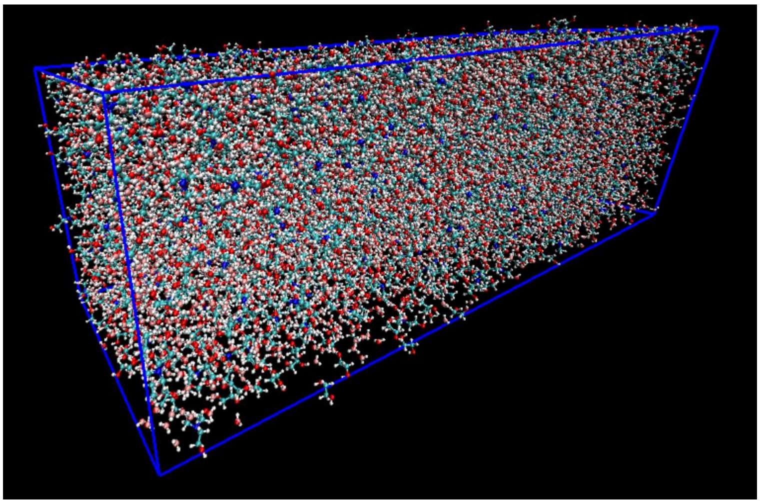


Figure 5.11: Equilibrated liquid box.

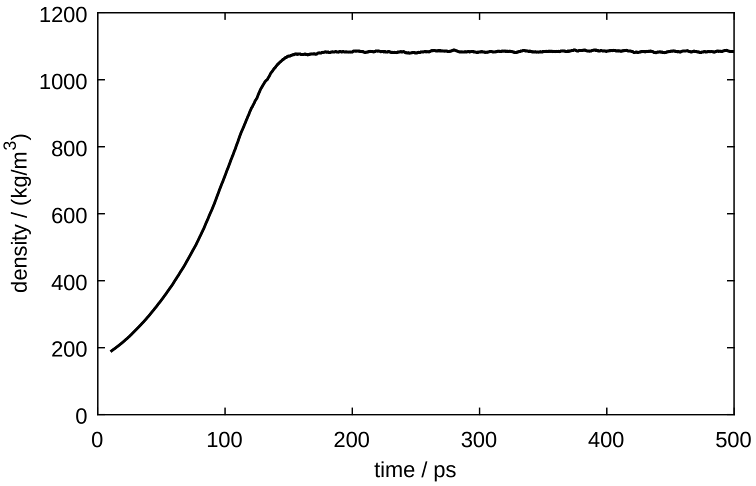
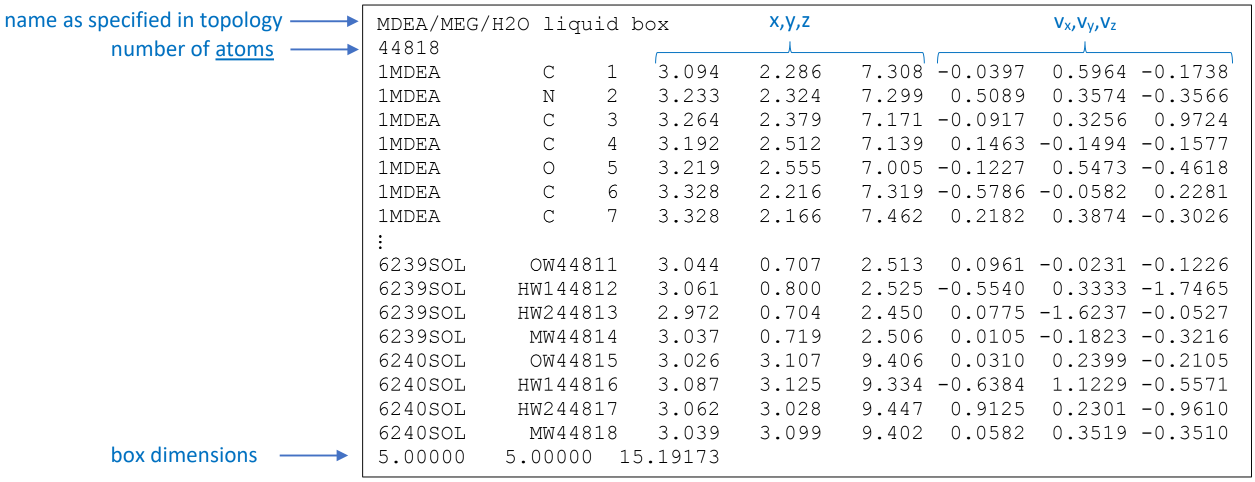


Figure 5.12: Equilibration process of liquid box, 1000 bar simulation.

The described simulation starts in a not equilibrated state and reaches equilibrium at a certain point during the simulation. This process is called *equilibration* and can be visualized with the analysis functions in GROMACS subsequent[12] to the simulation, figure 5.12. Apparently, 500 ps simulation is long enough to reach mechanical (density) equilibrium. Chemical equilibrium is not considered in this pre-simulation.
The `.gro` position file of the equilibrated liquid box is presented in figure 5.13.



```
MDEA/MEG/H2O liquid box
44818
1MDEA        C    1   3.094   2.286   7.308 -0.0397  0.5964 -0.1738
1MDEA        N    2   3.233   2.324   7.299  0.5089  0.3574 -0.3566
1MDEA        C    3   3.264   2.379   7.171 -0.0917  0.3256  0.9724
1MDEA        C    4   3.192   2.512   7.139  0.1463 -0.1494 -0.1577
1MDEA        O    5   3.219   2.555   7.005 -0.1227  0.5473 -0.4618
1MDEA        C    6   3.328   2.216   7.319 -0.5786 -0.0582  0.2281
1MDEA        C    7   3.328   2.166   7.462  0.2182  0.3874 -0.3026
⋮
6239SOL     OW44811   3.044   0.707   2.513  0.0961 -0.0231 -0.1226
6239SOL    HW144812   3.061   0.800   2.525 -0.5540  0.3333 -1.7465
6239SOL    HW244813   2.972   0.704   2.450  0.0775 -1.6237 -0.0527
6239SOL     MW44814   3.037   0.719   2.506  0.0105 -0.1823 -0.3216
6240SOL     OW44815   3.026   3.107   9.406  0.0310  0.2399 -0.2105
6240SOL    HW144816   3.087   3.125   9.334 -0.6384  1.1229 -0.5571
6240SOL    HW244817   3.062   3.028   9.447  0.9125  0.2301 -0.9610
6240SOL     MW44818   3.039   3.099   9.402  0.0582  0.3519 -0.3510
5.00000   5.00000  15.19173
```

liquidbox_simulation.gro

Figure 5.13: The `.gro` position file of the equilibrated liquid box.

5.5.3 Step 5c – Aggregation of boxes

In this step the created and equilibrated boxes are placed next to each other and connected to one big box. For this purpose, we must translate the coordinates of the boxes first. For this all atoms in the equilibrated liquid box (`liquidbox_simulation.gro`) are shifted by the z-length of the first gas box. Similarly, all atoms of the second gas box are shifted by the summarized length of first gas box and liquid box. The dimensions of the boxes are stated at the end of the respective `.gro` files, compare figure 5.13. The related command for shifting a box is `gmx_mpi editconf`.

```
 gmx_mpi editconf -f liquidbox_simulation.gro -o
liquidbox_eqil_shifted.gro -translate 0 0 33.5

gmx_mpi editconf -f gasbox.gro -o gasbox_shifted.gro -translate 0 0 48.7
```

---

[12] see section 5.7

The three boxes can then simply be combined with the command (not a GROMACS command):

```
 cat liquidbox_equil_shifted.gro gasbox.gro gasbox_shifted.gro >
VLEbox.gro
```

Finally, the user has to (manually) remove old headlines and bottom lines in the merged document and sum up atom numbers and dimensions to generate a correct new headline and bottom line. A visualization of the new `VLEbox.gro` position file is given in figures 5.15 and 5.16.

Since we have obtained a box with all components, a new topology file must be created. Again, in the topology file components must be stated in the order as they were inserted and merged. The topology file thus becomes:

```
#include "oplsaa.ff/forcefield.itp"
#include "MDEA.itp"
#include "MEG.itp"
#include "oplsaa.ff/tip4p.itp"
#include "H2S.itp"
#include "CH4.itp"

[ system ]
; Name
MDEA/MEG/H2O/H2S/CH4 VLE box

[ molecules ]
; Compound         #mols
H2S                 500
CH4                 1000
MDEA                800
MEG                 1500
SOL                 5000
H2S                 500
CH4                 1000
```

VLEbox.top

Figure 5.14: Topology of the two phase system.

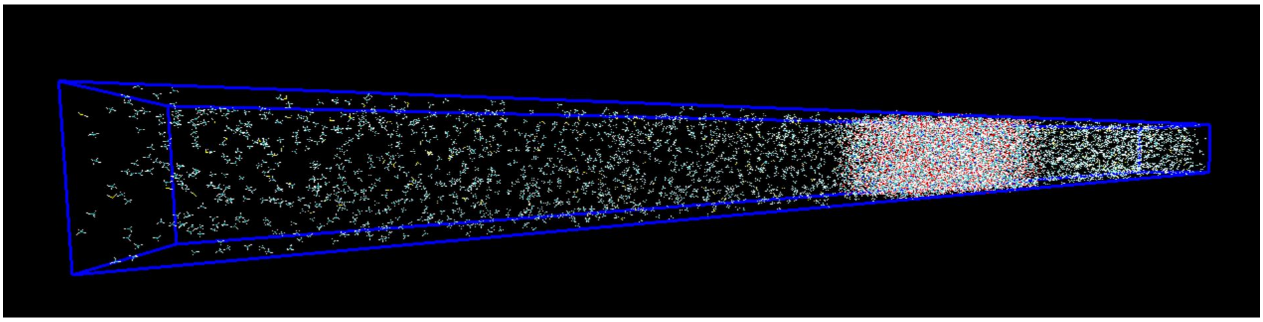


Figure 5.15: Visualization of `VLEbox.gro` in VMD before energy minimization and simulation.

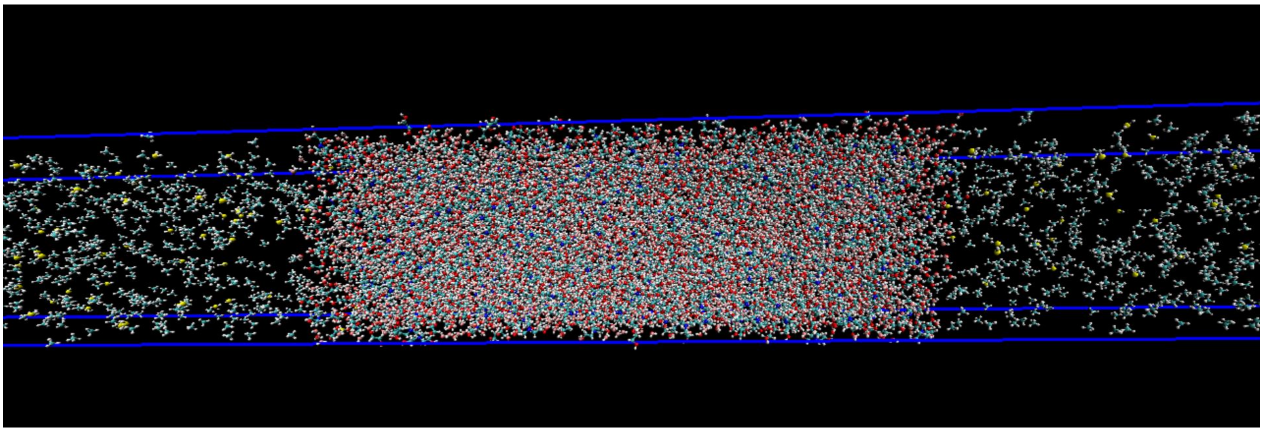


Figure 5.16: Visualization of liquid phase in `VLEbox.gro` in VMD before energy minimization and simulation

5.6 Step 6 – Simulation

The final simulation of the two-phase system follows the same procedure as before. Typical simulation times are (at least) $t_{end} - t_0 = 100\,\mathrm{ns}$ with time step size of $\Delta t = 0.001\,\mathrm{ps}$.
The idea of this simulation is to reproduce the conditions in the experiments. Since the experiments are carried out at constant volume and without any pressure control, the VLE simulation is a NVT simulation. Volume and molecule number of the gas boxes have previously been chosen according to the initial pressure[13] of the gas phase and gas-to-liquid ratio in the experiments. For sake of completeness, all necessary commands are stated below:

```
 gmx_mpi grompp -f minim.mdp -c VLEbox.gro -p VLEbox.top -o
VLEbox_minim.tpr

gmx_mpi mdrun -v -deffnm VLEbox_minim -ntomp 32 &

gmx_mpi grompp -f nvt.mdp -c VLEbox_minim.gro -p VLEbox.top -o
VLEbox_sim.tpr

gmx_mpi mdrun -v -deffnm VLEbox_sim -ntomp 32 &
```

Prior to the main simulation, an energy minimization is again performed to avoid adverse atom configurations in the not yet equilibrated gas phase and in regions where two boxes have been interconnected.

---

[13] The pressure in the resulting gas box can be determined with analysis methods described in section 5.7. The insertion process for the gas box is thus iterative (until the desired gas pressure is met).

5.7 Step 7 – Analysis of results

Besides performing molecular simulations, GROMACS enables the user to analyze generated trajectory and energy data. Some important functions are `gmx_mpi energy` and `gmx_mpi density`. Moreover, tools for "advanced analysis", e.g. pair correlation functions, are provided. `gmx_mpi energy` offers multiple quantities to be computed such as pressure, temperature, total density or energies, whereas `gmx_mpi density` calculates a time-averaged density distribution of one component in the box and thereby gives access to quantities like mole fractions or distributions coefficients. A comprehensive introduction to the analysis functions of GROMACS is given in chapter 8 of the GROMACS manual [1].
The data for the density graph (figure 5.12) was generated by means of `gmx_mpi energy`. The related command is:

```
 gmx_mpi energy -f liquidbox_simulation.edr -s liquidbox_simulation.tpr
-o liquidbox_simulation.xvg
```

As can be seen, the analysis function in GROMACS requires an energy file (`.edr`) as well as trajectory file (`.xtc`) and information about the simulation (`.tpr`) that was used to generate the data. Subsequently, "density" could be selected in the appearing menu in the console. The generated data is saved in table-form into a `.xvg` file and can be plotted in MATLAB.

Figure 5.17 displays the system pressure over time. Pressure is a typical system quantity that fluctuates strongly with time since its calculation is very sensitive to slight changes in atom positions. Even though the very noisy shape does not seem very informative, the time-average value is relatively exact.

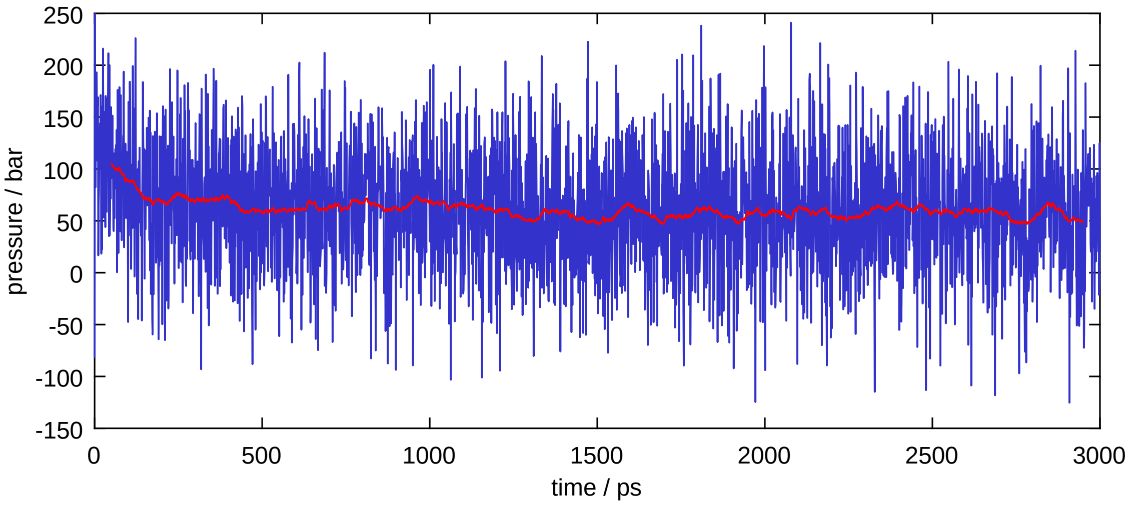


Figure 5.17: Pressure of VLE system during the first 3 ns of NVT simulation at approx. 60 bar. Averaged pressure (red) for intervals of 1 ps.

The chemical equilibration procedure is shown in figure 5.18 as the species density of H_2S at different instants of time. The generated `.xvg` file was plotted in MATLAB. The calculated species number density is averaged over a time integral of 200 ps and determined along z-direction for 100 Δz slices. In this analysis method the `.trr` file provides the actual information. Therefore, it is crucial that the `nstxout` log frequency in the simulation settings, cf. fig A.4, was set to a small enough value (here 5 ps). Otherwise the amount of available information in a time integral of 200 ps is too low to generate accurate data.

```
 gmx_mpi density -f VLEbox_100bar_120C.trr -s VLEbox_100bar_120C.tpr -o
Density_H2S.xvg -dens number    -b 30000      -e 30200       -sl 100
                number density  evaluation start  evaluation stop  number of slices
```

Apparently, the chemical component H_2S accumulates in the liquid phase, but is also present in the gas phase, figure 5.18. The relatively smooth and constant distribution in the liquid phase at $t = 100$ ns indicates approach to an equilibrium. However, since the density profile in bulk and boundary layer differs significantly between 60 ns, 80 ns (not shown for the sake of clarity) and 100 ns, the simulation should be extended until a fully equilibrated state is reached.

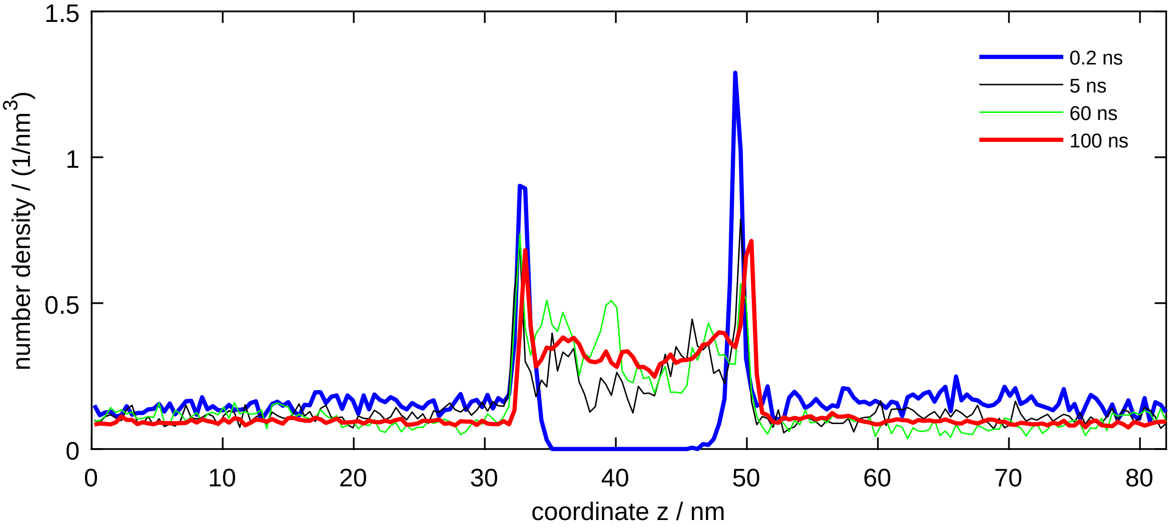


Figure 5.18: Species density of H_2S in VLE system at different instants of time. Simulation: 120 °C, approx. 20 bar, 100 ns.

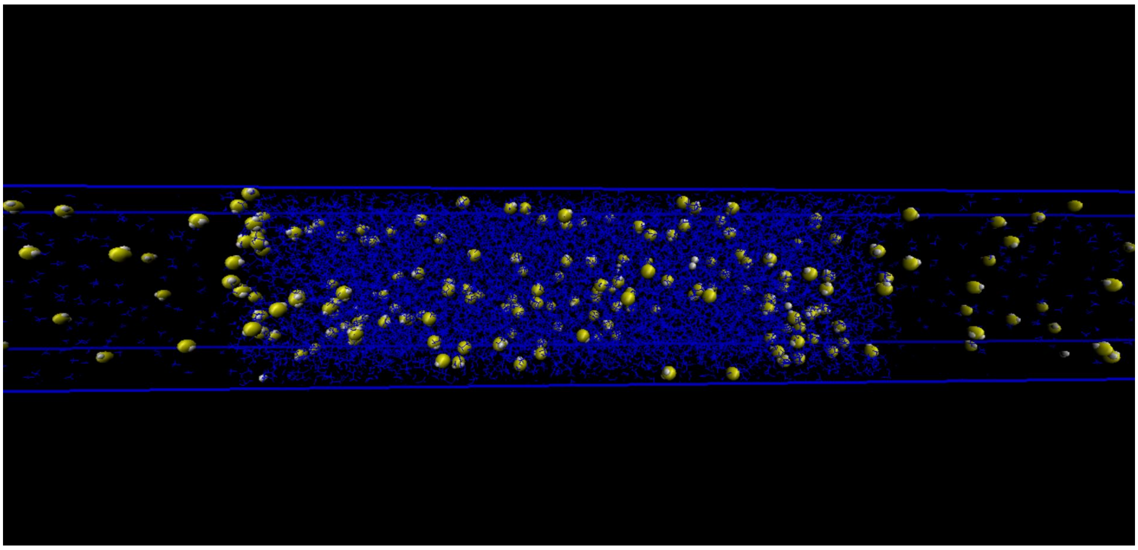


Figure 5.19: Visualization of the final system state in VMD. H_2S molecules as yellow van-der-Waals spheres, all other components in blue. Simulation: 120 °C, approx. 20 bar, 100 ns.

6 Conclusion

GROMACS is a versatile software package for simulation and analysis of molecular systems. The intent of this report is to provide the reader with a simple and vivid introduction to molecular simulations. However, many details have only been barely covered since MD simulations involve many different methods for simulation and analysis. Nevertheless, the reader of this report should be able to set up simple MD simulations and assess the most important simulation parameters displayed in figures A.4 and A.5. A well written and fine-grained description of all MD principles and their implementation in GROMACS can be found in the GROMACS manual [1], which is frequently refined and updated by the developers. Since GROMACS does not have a graphical user interface, the Linux console is used for insertion of all commands. GROMACS commands are, however, very straightforward, even for the Linux-unexperienced user, and should not be seen as obstacle for performing molecular simulations.

References

[1] Abraham, M., van der Spoel, D., Lindahl, E., Hess, B., and the GROMACS development team. *GROMACS User Manual version 2016.3*. www.gromacs.org, 2017.

[2] Allen, M. P., et al. Introduction to molecular dynamics simulation. *Computational soft matter: from synthetic polymers to proteins 23* (2004), 1–28.

[3] Damm, W., Frontera, A., Tirado-Rives, J., and Jorgensen, W. L. OPLS all-atom force field for carbohydrates. *Journal of Computational Chemistry 18*, 16 (1997), 1955–1970.

[4] de Oliveira, O. V., and Freitas, L. C. G. Molecular dynamics simulation of liquid ethylene glycol and its aqueous solution. *Journal of Molecular Structure: THEOCHEM 728*, 1 (2005), 179–187.

[5] Farmahini, A. H. *Molecular dynamics simulation studies of piperazine activated MDEA absorption solution with methane and carbon dioxide.* PhD thesis, The University of Bergen, 2010.

[6] Fogarty, J. C., Aktulga, H. M., Grama, A. Y., Van Duin, A. C., and Pandit, S. A. A reactive molecular dynamics simulation of the silica-water interface. *The Journal of chemical physics 132*, 17 (2010), 174704.

[7] Jorgensen, W. L., Chandrasekhar, J., Madura, J. D., Impey, R. W., and Klein, M. L. Comparison of simple potential functions for simulating liquid water. *The Journal of chemical physics 79*, 2 (1983), 926–935.

[8] Jorgensen, W. L., and Madura, J. D. Temperature and size dependence for monte carlo simulations of tip4p water. *Molecular Physics 56*, 6 (1985), 1381–1392.

[9] Jorgensen, W. L., Maxwell, D. S., and Tirado-Rives, J. Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *J. Am. Chem. Soc 118*, 45 (1996), 11225–11236.

[10] Lawrence, C., and Skinner, J. Flexible TIP4P model for molecular dynamics simulation of liquid water. *Chemical physics letters 372*, 5 (2003), 842–847.

[11] Lucas, K. *Molecular models for fluids.* Cambridge University Press, 2007.

[12] Van Duin, A. C., Dasgupta, S., Lorant, F., and Goddard, W. A. Reaxff: a reactive force field for hydrocarbons. *The Journal of Physical Chemistry A 105*, 41 (2001), 9396–9409.

A Appendix

A.1 Comment on bonding potentials

Interactions between bonded atoms have a special quantum mechanical character that is approximated by harmonic potentials. The Lennard-Jones (and Coulomb) pair interaction models are not valid for these types of atom pairs and must be excluded from the calculation of the potential. Otherwise the short distance of bonded atoms would lead to high repulsion in the Lennard-Jones potential and thereby cause the molecule to break apart. As explained, direct bonding of atoms is modeled by stretching potentials, whereas bending vibration (1-3 interaction in 1-2-3 chain) is accounted for by bending potentials. Even the "1-4 interaction" between the first and the last atom in a four-atom chain cannot be modeled with the standard Lennard-Jones and Coulomb potential. For these interaction an internal rotation potential is applied in combination with corrected intermolecular potentials (often reduced by factor[14] 0.5 or individual LJ parameterization for 1-4 interactions). Relevant 1-4 interaction pairs are listed in the category `[pairs]` in the `.itp` files. In contrast, atoms separated by more than two atoms, i.e. 1-5, 1-6 pairs and so on, are considered to interact with regular Lennard-Jones and Coulomb forces.

A.2 The TIP4P model

The behavior of water is not so simple to be accounted for in molecular simulations. Its interactions are strongly dominated by hydrogen bonds that lead to the characteristic abnormality in almost every respect. Various approaches to molecular modeling of water molecules have been published in the past, including the TIP4P model. The main idea in TIP4P is to add a non-physical mass-less but charged "atom" M to the molecule, whereas the oxygen atom has a mass but no partial charge. In this context, the four "atoms" of the TIP4P water construct are referred to as "sites" instead. The OPLS force field is then parameterized for all four sites (three atom types: H, O and M). The TIP4P water model is visualized in figure A.1. Further discussion on the TIP4P model can be found in [7, 8, 10].

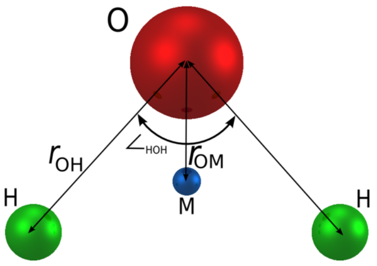


Figure A.1: The TIP4P molecular construct for modeling water. Source: `sklogwiki.org/SklogWiki/index.php/TIP4P_model_of_water`

---

[14] `fudgeLJ` and `fudgeQQ`

```
216 TIP4P Water Molecules Equilibrated for 20 ps at 300 K
  4
    1SOL     OW    1   1.736   0.839   0.257 -0.0525 -0.0128  0.1333
    1SOL    HW1    2   1.777   0.781   0.322  0.3406  0.5030  0.3534
    1SOL    HW2    3   1.643   0.831   0.274  0.0528  0.2742  0.9186
    1SOL     MW    4   1.730   0.831   0.267  0.3505  0.3106  0.0246
   1.86824   1.86824   1.86824
```

`tip4p.gro`

Figure A.2: Molecular coordinate file for TIP4P water.

A.3 The MD integrator

Starting from the equations of motion, eqn. 2.1, velocities and positions are time-discretized by means of Taylor expansion. Developers of MD algorithms have found a numerical benefit in shifting the sampling points for velocity and positions relative to each other by $\Delta t/2$:

$$\mathbf{u}_i(t + \frac{\Delta t}{2}) = \mathbf{u}_i(t - \frac{\Delta t}{2}) + \frac{\Delta t}{m_i} \cdot \mathbf{f}_i \tag{A.1}$$

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta t \cdot \mathbf{u}_i(t + \frac{\Delta t}{2})$$

the so-called "leap-frog" integrator[15]. Apparently, this algorithm needs initial information on both velocity and position. While positions are specified in terms of an initial box, initial velocities are often not defined during the box-making process. In such a case velocities are generated based on the relation between temperature and kinetic energy (from kinetic theory of gases) as Maxwell-Boltzmann probability distribution p:

$$p(v_i) = \sqrt{\frac{m_i}{2\pi k T_0}} \exp\left(-\frac{m_i v_i^2}{2kT_0}\right) \tag{A.2}$$

with Boltzmann constant k and temperature T_0 for generation of initial velocities, compare last category in `.mdp` files, figures A.4 and A.5.

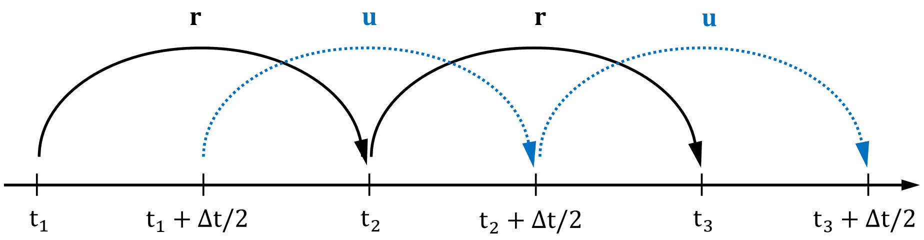


Figure A.3: The leap-frog integrator.

---

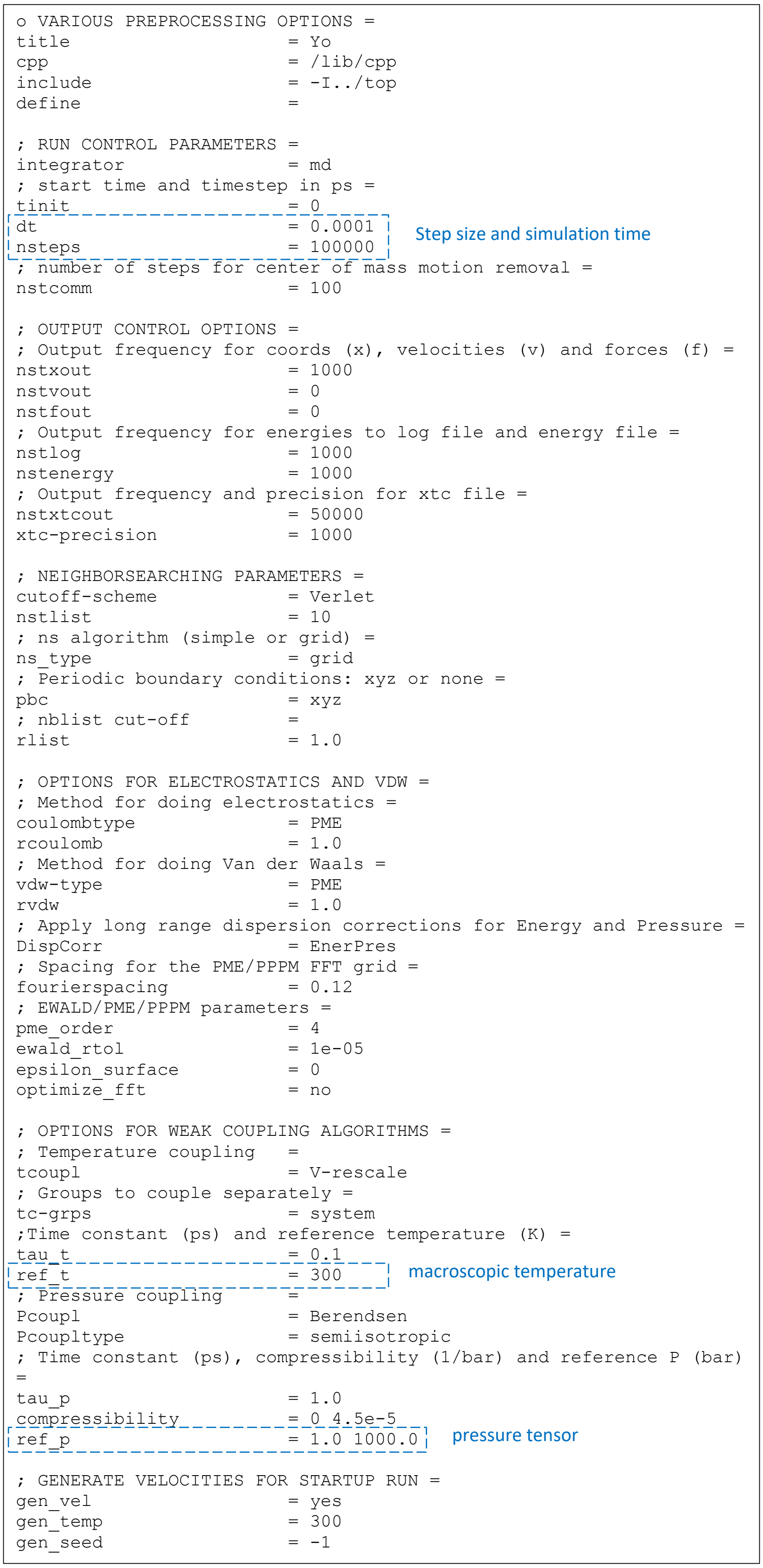
[15] Mental association, since velocities and positions are updated as frogs jump over each other.

A.4 Comment on chemical reactions

In molecular dynamics, atoms are modeled as single point mass with a certain partial charge rather than protons and electrons. Electrons are assumed to be in electronic ground state ("conservative force field"), an assumption that is usually valid for chemical engineering conditions. On the contrary, chemical reactions have a strong quantum mechanical (QM) character and their modeling requires different electronic states to be taken into account. This is why MD by itself is not able to simulate molecular reactions properly. However, there are methods to incorporate reaction mechanisms into MD simulations:

Reactions can be realized by coupling molecular dynamics and quantum mechanical methods. GROMACS provides interfaces to popular QM software such as Gaussian. For further discussion see [1].

In recent years, authors have tried to develop force fields that imitate the quantum mechanical character of reactions by introducing additional potential functions that allow for cleavage and formation of bonds [12, 6]. However, these relatively new force fields need a high amount of additional parameters that might not yet be available for many components. Furthermore, the results are not always in agreement with QM calculations.



```
o VARIOUS PREPROCESSING OPTIONS =
title                    = Yo
cpp                      = /lib/cpp
include                  = -I../top
define                   =

; RUN CONTROL PARAMETERS =
integrator               = md
; start time and timestep in ps =
tinit                    = 0
dt                       = 0.0001
nsteps                   = 100000
; number of steps for center of mass motion removal =
nstcomm                  = 100

; OUTPUT CONTROL OPTIONS =
; Output frequency for coords (x), velocities (v) and forces (f) =
nstxout                  = 1000
nstvout                  = 0
nstfout                  = 0
; Output frequency for energies to log file and energy file =
nstlog                   = 1000
nstenergy                = 1000
; Output frequency and precision for xtc file =
nstxtcout                = 50000
xtc-precision            = 1000

; NEIGHBORSEARCHING PARAMETERS =
cutoff-scheme            = Verlet
nstlist                  = 10
; ns algorithm (simple or grid) =
ns_type                  = grid
; Periodic boundary conditions: xyz or none =
pbc                      = xyz
; nblist cut-off         =
rlist                    = 1.0

; OPTIONS FOR ELECTROSTATICS AND VDW =
; Method for doing electrostatics =
coulombtype              = PME
rcoulomb                 = 1.0
; Method for doing Van der Waals =
vdw-type                 = PME
rvdw                     = 1.0
; Apply long range dispersion corrections for Energy and Pressure =
DispCorr                 = EnerPres
; Spacing for the PME/PPPM FFT grid =
fourierspacing           = 0.12
; EWALD/PME/PPPM parameters =
pme_order                = 4
ewald_rtol               = 1e-05
epsilon_surface          = 0
optimize_fft             = no

; OPTIONS FOR WEAK COUPLING ALGORITHMS =
; Temperature coupling   =
tcoupl                   = V-rescale
; Groups to couple separately =
tc-grps                  = system
;Time constant (ps) and reference temperature (K) =
tau_t                    = 0.1
ref_t                    = 300
; Pressure coupling      =
Pcoupl                   = Berendsen
Pcoupltype               = semiisotropic
; Time constant (ps), compressibility (1/bar) and reference P (bar)
=
tau_p                    = 1.0
compressibility          = 0 4.5e-5
ref_p                    = 1.0 1000.0

; GENERATE VELOCITIES FOR STARTUP RUN =
gen_vel                  = yes
gen_temp                 = 300
gen_seed                 = -1
```

Figure A.4: File structure of NpT simulation settings file. V-rescale for thermostat, Berendsen for barostat.

```
o VARIOUS PREPROCESSING OPTIONS =
title                    = VLE

; RUN CONTROL PARAMETERS =
integrator               = md
; start time and timestep in ps =
tinit                    = 0
dt                       = 0.001
nsteps                   = 1000000
; number of steps for center of mass motion removal =
nstcomm                  = 100

; OUTPUT CONTROL OPTIONS =
; Output frequency for coords (x), velocities (v) and forces (f) =
nstxout                  = 5000
nstvout                  = 0
nstfout                  = 0
; Output frequency for energies to log file and energy file =
nstlog                   = 1000
nstenergy                = 1000
; Output frequency and precision for xtc file =
nstxtcout                = 50000
xtc-precision            = 1000

; NEIGHBORSEARCHING PARAMETERS =
cutoff-scheme            = Verlet
nstlist                  = 10
; ns algorithm (simple or grid) =
ns_type                  = grid
; Periodic boundary conditions: xyz or none =
pbc                      = xyz
; nblist cut-off         =
rlist                    = 1.0

; OPTIONS FOR ELECTROSTATICS AND VDW =
; Method for doing electrostatics =
coulombtype              = PME
rcoulomb                 = 1.0
; Method for doing Van der Waals =
vdw-type                 = PME
rvdw                     = 1.0
; Apply long range dispersion corrections for Energy and Pressure =
DispCorr                 = EnerPres
; Spacing for the PME/PPPM FFT grid =
fourierspacing           = 0.12
; EWALD/PME/PPPM parameters =
pme_order                = 4
ewald_rtol               = 1e-05
epsilon_surface          = 0
optimize_fft             = no

; OPTIONS FOR WEAK COUPLING ALGORITHMS =
; Temperature coupling   =
tcoupl                   = V-rescale
; Groups to couple separately =
tc-grps                  = system
;Time constant (ps) and reference temperature (K) =
tau_t                    = 0.1
ref_t                    = 300

; GENERATE VELOCITIES FOR STARTUP RUN =
gen_vel                  = yes
gen_temp                 = 300
gen_seed                 = -1
```

Figure A.5: File structure of NVT simulation settings file.

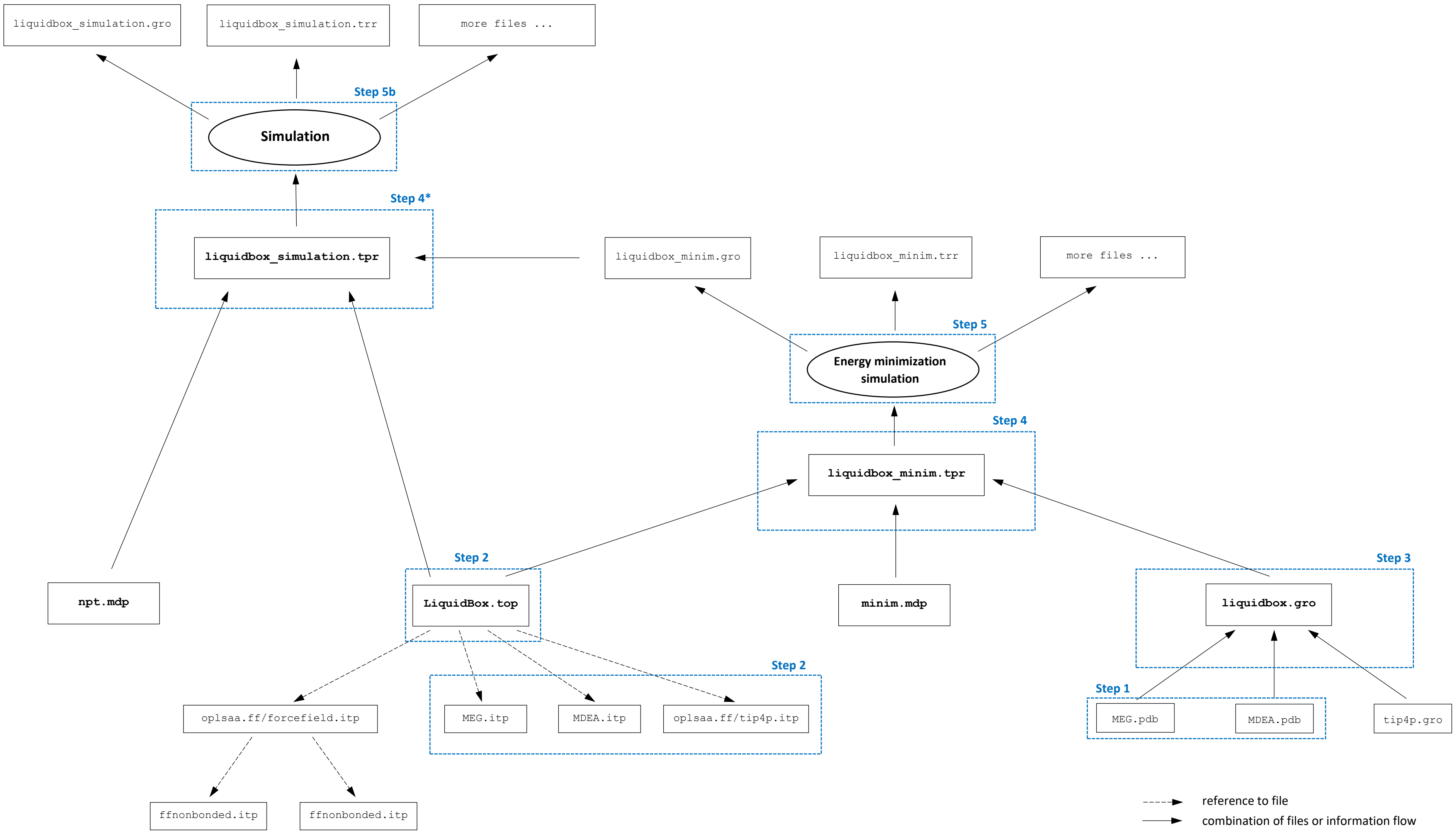


Figure A.6: Complete file structure and flow diagram for liquid box equilibration in GROMACS.