

FREIE UNIVERSITÄT BERLIN

COURSE REPORT

**Soft Matter:
Mathematical Aspects, Physical Modelling
and Computer Simulation**

Author:
Benjamin KAHL

Supervisors:
Prof. Luigi DELLE SITE
Dr. Christian KREKELER

Winter Semester 2018/19

Theoretical and Mathematical Physics in Molecular Simulation

Department of Mathematics and Computer Science

February 6, 2019

Motivation

The field of molecular dynamics is a compound intersection point for a variety of long-standing disciplines. Advanced mathematical concepts are employed to model chemical phenomena on the basis of molecular mechanics.

When the complexity of such systems reaches a breaking point, the application of computer simulations can greatly ameliorate the endeavour of extrapolating useful data.

In order to enable sophisticated chemical compounds to be simulated on consumer grade hardware, a manifold of reductionist simplifications need to be employed. Among other things, subatomic particles are disregarded and the complex quantum model of atomic orbitals is replaced by rudimentary mechanical physics.

However, despite these apparent shortcomings, molecular dynamics simulations have found a plethora of successes in predicting both static and dynamic properties of chemical substances, even on ordinary hardware systems.

Given that no alternative approach exists capable of providing physically faithful results to a broad range of problems within a limited computation time, these methods have proven themselves indispensable in the fields of material sciences and biochemistry.

This report aims at relaying the knowledge acquired during the practical portion of the course *Soft Matter: Mathematical Aspects, Physical Modelling and Computer Simulation*, absolved in the 2018 winter semester at the Free University of Berlin.

Software and Dependencies

The modelling of molecular mechanics by the application of numerical methods is a task grappled by a manifold of diverse software packages. Applications such as *LAMMPS*, *GROMOS* and *GROMACS*, to name a few, each provide distinct advantages and drawbacks, including both proprietary and open-source license variants.

Given the immense possibilities of combining various molecular dynamics (MD) software, this chapter aims to provide a rough understanding of the applications used to conduct the simulations listed in chapter 6.

2.1 Operating Environment

All simulations were carried out on a *Linux* operating system, primarily via a *Unix command shell*.

The configuration and execution of each simulation was accomplished with the aid of a self-implemented *bash shell script*. A detailed description of this script's inner workings and its purveyed effects are provided in chapter 5.

The complete source code as well as the results and configurations of all executed simulations can be viewed in the repository linked in reference [6].

2.2 Used Software

2.2.1 GROMACS

The Groningen Machine for Chemical Simulations (henceforth called GROMACS) is a software package primarily purposed to simulate the molecular behavior of polymers, proteins as well as inorganic substances.

It is a free, open-source program distributed under a *LGPL* license and is largely maintained by contributors in research centers worldwide.

Based on a rough, Monte-Carlo approach, GROMACS allows for complex systems to be simulated even on consumer grade hardware. However, lengthy simulations can lead to inaccurate results due to the generation of cumulative errors in numerical integration.

These incongruities can be minimized by running each simulation on appropriately chosen parameters, but never eliminated completely.

The results provided in this report were obtained with a 2016.1-version of GROMACS running without GPU support on a 64-bit memory model.

GROMACS fulfills most of the necessities to run general MD simulations, although additional tools are required for the express purposes of visualization and composition.

2.2.2 VMD

Visual Molecular Dynamics (VMD) is an OpenGL-based rendering tool for the visualization and analysis of molecular dynamics simulations.

Rather than merely evaluating simulations by their statistical properties alone, the visualization of a simulated system can be greatly beneficial to extrapolate sought-after data and identify potential mistakes a priori.

In gratitude to its native compatibility with GROMACS, VMD is the de-facto recommended tool for the visualization of GROMACS-based simulations.[3]

2.2.3 Packmol

In order to receive useful data from a simulation, the participating molecules first need to be arranged in a physically feasible manner. Due to the Coulomb interaction between atoms, molecules will experience an exorbitant repulsive force when placed too close to one another, leading the system to blow up within the first few time-frames.

To rectify this issue, a starting point to the simulation needs to be established by evenly spreading out the molecules across a specified volume. Due to the potential number of molecules involved, this task is both tedious and ineffective to perform manually.

The open source program *Packmol* solves this problem by procedurally packing specified molecules into defined regions of space. The results are a set of homogeneously spread coordinates and orientations that correspond to the individual molecules' predicaments in space.

A variety of spatial constraints can be imposed upon the algorithm, but only Packmol's default settings will be utilized for the remainder of this report.

Formats and File extensions

The above-listed software handles and saves simulation data in a finite set of specified formats. Each of these is designated by a corresponding file extension.

First understanding the meaning and use of these file extensions and their respective formats can be of great benefit to smooth out the learning curve GROMACS would otherwise provide.

This chapter gives a general overview of the data-formats used within the bounds of this report.

3.1 Structure Files

Simply put, structure files describe the geometry of a molecular structure. The two prevalent formats are the *Gromos87* (file-extension '.gro') and the *brookhaven Protein DataBank* (file-extension '.pdb') formats.

Both of these are for the most part interchangeable as PDB-to-GRO conversion is easily possible with the GROMACS `gmx editconf` command.

In essence, the positional coordinates of every atom or atom-group are listed within these files. By concatenating a multitude together, the extrapolation of an atoms trajectory becomes possible, which constitutes a *trajectory file* (see 3.4).

The previously mentioned program *Packmol* will generate such PDB files.

3.1.1 PDB File

The above introduced PDB format follows a tabular structure with each row of the table corresponding to an atom or atom-group of the molecule.

The description of, for instance, a Propane molecule (C_3H_8) tends to fall within one of two approaches:

All-Atom Approach

The all-atom approach describes every atom individually. In the case of Propane, there would be a total of 11 rows in the PDB the table, three of which would correspond to the carbon atoms.

This technique will typically lead to a more realistic simulation outcome, but will dramatically affect performance and simulation time.

For the purposes of this report, the all-atom approach will be discarded in favour of the significantly faster united-atom approach.

United-Atom Approach

The united-atom approach simplifies the description of the molecule by combining each of its chemical functional groups into single entities. In the case of Propane, this would lead to 3 of such entities: two CH_3 groups and one CH_2 group.

The standardized format of a PDB table contains up to 16 rows. Only 11 of these will be contemplated here, leaving the remainder blank, as they are not relevant to the contents of this report.

- The first column contains a name for the given row itself. Common practice is to simply ascribe this field the value HETATM.
- The second and third columns give the atoms or atom groups at hand a unique serial number *anum* as well as a name *aname*. As can be seen in table 3.1 the CH_3 groups are labeled as 'C1' and 'C3' and the CH_2 group as 'C2', signifying the carbon-atom number across the molecule.
- The fourth and fifth columns ascribe a serial number *rnum* and name *rname* to the residue (molecule) the atoms are a part of. As the current context only demands the definition of a single molecule (Propane), these values will remain identical for all three rows. The chosen "PRO" label in table 3.1 corresponds to the molecule name declared in the corresponding ITP file (see 3.2).
- The following three values correspond to the atoms respective *x*, *y* and *z* coordinates in space. These can be vaguely approximated by trigonometric calculations on known molecule structures or precisely obtained through a variety of chemical and quantum measurement techniques.

As can be seen in 3.1, the *z* coordinates of a propane molecule are all equal, meaning that the molecules' atoms all lie within a single two-dimensional plane.

- The next two values describe the occupancy *o* and temperature factor *tf*. The occupancy can be used to describe the observed likelihood of different crystal conformations. The temperature factor can be used to adjust an atoms vibrations to match the experimentally observed electron density distribution. Given that these two variables have no relevance to the purposes of this report, they will always be ascribed their default values.
- The last value describes the atom groups' formal charge *ch*. In table 3.1, for instance, all groups are given the neutral charge of a hydro-carbon group by ascribing them the value 'C'.

<i>name</i>	<i>anum</i>	<i>aname</i>	<i>rname</i>	<i>rnum</i>	<i>x</i>	<i>y</i>	<i>z</i>	<i>o</i>	<i>tf</i>	<i>ch</i>
HETATM	1	C1	PRO	0	-1.278	0.269	-0.000	1.00	0.00	C
HETATM	2	C2	PRO	0	-0.000	-0.577	-0.000	1.00	0.00	C
HETATM	3	C3	PRO	0	1.278	0.269	0.000	1.00	0.00	C

TABLE 3.1: Value-table of a propane molecule according to the pdb format.

Each atoms individual properties and positions are now defined, but the bonds binding them into a molecule are not.

PDB files do indeed allow for bond definitions to be made, but here, the bond definitions will be defined exclusively in a so-called topology file.

3.2 Topology Files

3.2.1 ITP File

Topology files contain a description of the overall molecule structure. The commonly used ITP format contains a finite set of tables each of which describe a certain property of the molecules' geometry.

The primary of these is typically an atom table which declares additional properties of the given atoms such as their mass, charge and type. In many aspects, this is duplicate data from the PDB file described above, but lacks the atoms' starting positions.

Additional tables contained within this file provide data on the individual covalent bonds of the molecule such as their preferred angle.

The exact contents of each individual table is too lengthy to be reviewed here, but can be retrieved on the ITP format documentation. [2]

3.2.2 TOP File

Furthermore, ITP files can be referenced in a TOP file. These are read by GROMACS and compiled together with a parameter file (see 3.3) into an executable simulation.

A common approach to the initial configuration of a simulation, is to define the atom positions within a PDB file, the molecules' topology within an ITP file and then referencing said ITP file via a TOP file, which is handed to GROMACS.

The three described structure files can be combined to form a defined molecule.

In order to expand, for instance, the definition of a united-atom Propane molecule into a Hexane one, three additional rows would need to be added into the PDB file, each with their respective coordinates. The three carbon groups would then have to be referenced in the ITP file with the respective bonds and angles.

3.3 Parameter Files

MDP files are essentially a long list of parameters that is passed to GROMACS in order to specify the nature of a simulation. General settings like the amount of simulation steps, spatial constraints and time-intervals are included as well as more distinct values such as cutoff-schemes, force-radii and pressure couplings.

Any settings not specified in the file, will be assigned a default value. The simulation will then be run on an automatically generated MDP file that contains both the explicitly set as well as the default values. Worded differently, the MDP file used in a simulation will override the default simulation settings set by GROMACS.

In this view, an MDP file describes the procedure or actions that are performed on any given molecule setup when a simulation is executed.

The MDP files used in this report correspond to the three procedures *Energy Minimization*, *NPT* and *NVT*, further described in chapter 4.

3.4 Trajectory Files

As previously stated, the molecules' trajectories can be extrapolated by chaining together a series of structure files. The resulting chain of structure files is stored as a single trajectory file, which adopts one of two formats:

3.4.1 XTC and TRR Files

XTC and TRR files contain the simulations' trajectory including all atoms' coordinates, velocities, forces and energies. They constitute a record of all data acquired throughout all steps of a simulation.

In contrast, a corresponding GRO file would only contain the final state of the simulation.

3.5 Miscellaneous

3.5.1 TPR File

TPR files are compiled binaries which include all data required by GROMACS to run a simulation.

GROMACS combines a TOP file, including all its references, together with an MDP file into an executable TPR binary. When executed, the binary will run the configured simulation.

3.5.2 XVG and EDR Files

After a simulation has successfully concluded, GROMACS provides a number of tools that can extrapolate relevant data from the respective structure- and trajectory-files.

Data that concerns energy values are stored using the xdr protocol, which creates files of an EDR format.

On the other hand, XVG files can contain a variety of data, such as radii of gyration, molecule density and others.

The XVG format is compatible with a variety of third-party programs and is thus easily plotted into a graph.

General Simulation Procedure

With an understanding of the utilized software and file-formats in place, this chapter will illustrate the general simulation procedure carried out for a variety of molecules in chapter 6.

To exemplify this process the individual simulation-steps of 500 hexane molecules will be conducted in the following subsections.

4.1 Molecule Building

In order to set up a simulation, the participating molecules need first to be defined.

A complete description includes the individual atoms' starting positions as well as the default geometry of their respective covalent bonds. A molecule blueprint comprised of this data can easily be achieved with a PDB- and a corresponding ITP file.

As stated in 3.2.2, these files can be created and adjusted manually. This process is, however, both tedious and completely unnecessary, as a number of online repositories already provide libraries of MD files for a vast number of different molecules.

The ITP and PDB files used for the purposes of this report were all retrieved from the *Automated Topology Builder (ATB) and Repository*[1], which includes numerical descriptions of over 250.000 pre-calculated molecules.

With these files being set up for one individual molecule, in this case Hexane, the next step becomes the creation of the simulations' starting point.

4.2 Molecule Scattering

As previously stated in 2.2.3, placing molecules too close, or worse, directly on top of one another will lead to an inconclusive and physically inaccurate simulation.

Consequently, a rudimentary simulation environment must first be set up by spreading the desired amount of molecules across a given volume.

An effective program to achieve this purpose is the already introduced Packmol. The tool is executed with an input file (typically called `pure.inp`) which references the PDB file of the defined molecule. In addition, the file contains supplementary parameters such as the molecule amount and the volume space across which the molecules are meant to be scattered.

In the present example of Hexane, the respective PDB file is linked inside the created `pure.inp` and the parameters are set to a bounding box of $50\text{\AA} * 50\text{\AA} * 50\text{\AA}$ and a total amount of 500 molecules.

Subsequently, the scatter algorithm can be executed by passing this file as regular input to the Packmol executable:

```
/ $ packmol < pure.inp
```

The resulting setup of 500 Hexane molecules scattered across a 50\AA volumetric cube can be observed in Fig. 4.1.

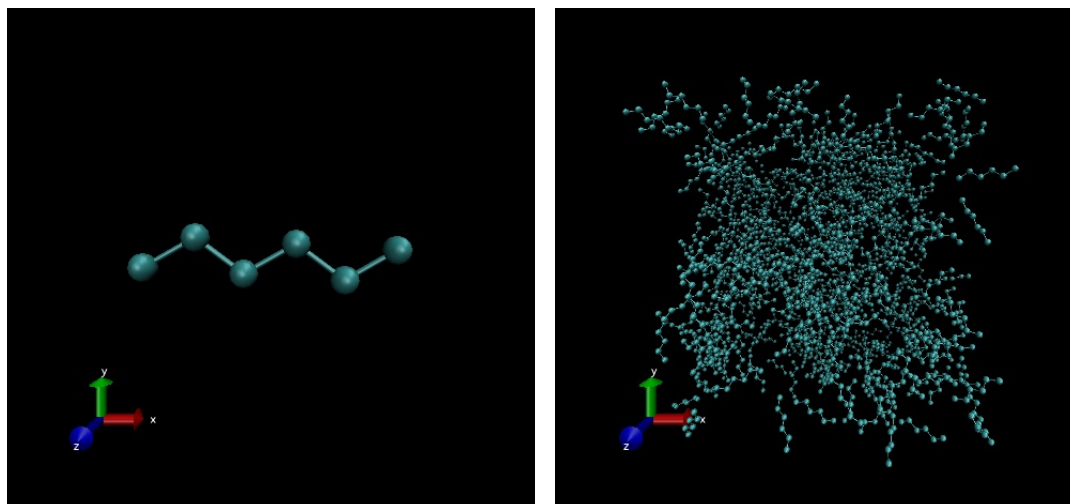


FIGURE 4.1: Single hexane molecule (left) and 500 hexane molecules randomly placed by Packmol into a 50Å cubic bounding box (right) (visualized with VMD)

4.3 Simulation State Creation

Unfortunately, the PDB files produced by Packmol cannot be read by GROMACS, and thus, must first be converted into GRO files. The `gmx editconf` command can perform this step automatically.

```
/ $ gmx editconf -f hexane_500.pdb -o hexane_500.gro
```

One persisting problem of this otherwise simple conversion is that Packmol deals in distance units of Å (Angstrom) while GROMACS exclusively uses *nm* (Nanometers).

For this reason, the measures of the bounding box defined in `pure.inp` must now explicitly be inserted in terms of nanometers into the last line of the current GRO file.

To conclude the creation of a runnable simulation state, a TOP file referencing the original ITP file needs to be created. Additionally to the ITP file, the TOP file also includes any other necessary dependencies. In this case, only the default library `gromos53a6.ff/forcefield.itp` is required.

The produced GRO and TOP file now constitute a simulation state that GROMACS can apply MDP procedures on.

A general overview of setting up and executing simulations including all files involved can be seen in Fig. 4.2.

4.4 Energy Minimization

As is visible in Fig. 4.1, the molecules have now been placed by Packmol, but still maintain the default pose initially defined to the molecule. The stiffness of this configuration does not correspond to the rather entropic stances molecules typically adopt in the real world.

Consequently, an algorithm which ‘relaxes’ the molecules and lets them adopt a more feasible pose is crucial in order to obtain trustworthy results. This procedure is usually termed *energy minimization* and it converts the system into one of less potential energy.

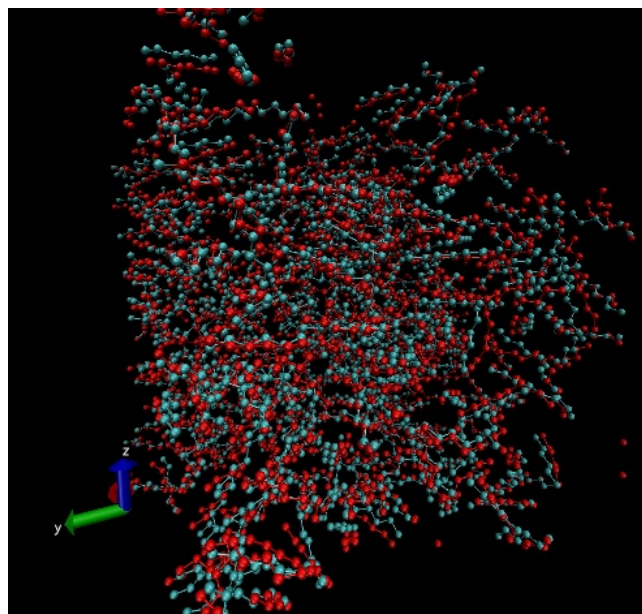


FIGURE 4.3: Hexane molecules before (in cyan) and after (in red) energy minimization

the desired state of matter, the bounding box will have to be removed and a specified temperature and pressure will have to be applied.

This process is called *NPT Equilibration* and typically applies a temperature of 298K (24.85°C) and a pressure of 1bar, as these are standard environment conditions.

The same process as in 4.4 is applied, just with a different MDP file and the simulation state that was obtained through the energy minimization run. The MDP used in this report runs the NPT procedure for 1million steps, each consisting of 2000ps.

The resulting simulation is visualized in Fig 4.4.

4.6 NVT

The NVT Equilibration procedure only differs from the NPT in that the pressure coupling is disabled. Instead, the system adopts a constant volume and temperature.

This crucial part of the simulation takes place after all environmental conditions have been normalized. This means that molecules' expected physical behaviour can best be studied by repeated applications of the NVT phase.

The MDP used in this report runs the NVT process for a total of 100.000 steps, each of 200ps.

4.7 Data Analysis

GROMACS provides a variety of commands for the extrapolation of relevant data from simulations:

```
/$ gmx energy -f [...].edr -o [...]_energy
/$ gmx desinty -f [...].xtc -s [...].tpr -o [...]_density
/$ gmx gyrate -f [...].xtc -s [...].tpr -o [...]_gyrate
/$ gmx polystat -f [...].xtc -s [...].tpr -o [...]_polystat
```

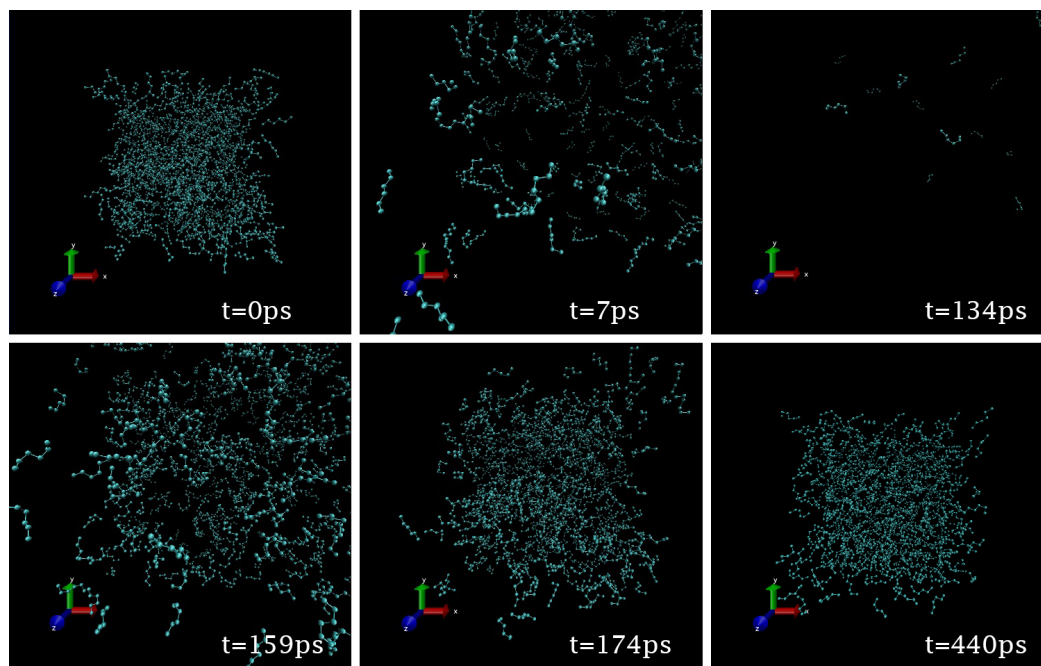


FIGURE 4.4: NPT Equilibration of Hexane. Vast initial expansion in volume as boundary restrictions are removed. Subsequent adoption of correct volume at standard conditions. The latter volume remains mostly constant for the remainder of the NPT Equilibration

The `gmx energy` command extracts the energy components from a file and lists them for every passed time-frame. A variety of energy terms can be selected, including density, potential, pressure and others.

The resulting XVG file can then be plotted using third-party programs such as `xmgrace`.

In Fig. 4.5 we have the potential energy during energy minimization plotted as a function of simulation steps. Even though the maximum amount of steps was set to 400,000, the graph reveals that a reasonable threshold was reached within only 212 steps. Note as well the early steep decline in energy as the algorithm has an easier time finding lower energy states than further into the execution time.

On the other hand, the graph in Fig 4.6 matches the exact development observed in Fig. 4.4.

The removal of boundary constraints leads to a steep decrease in density followed by a steady increase, as the 1bar of pressure starts to have an effect.

The density settles at around $620\text{kg}/\text{m}^3$ fairly rapidly and experiences only small oscillations near this value.

Somewhat surprisingly, this value differs substantially from the *actual* density of hexane of $655\text{kg}/\text{m}^3$.

Possible explanations for this discrepancy can include the inaccuracies of the united-atom approach as well as an inconsiderate simulation time.

The `gmx density` command differs from the density mentioned above in that it measures partial density values across the width of the box. These measurements are only conducted for the final state of the simulation and not over time.

This command will see use in chapter 6, where the partial densities of all simulated molecules will be compared to one another.

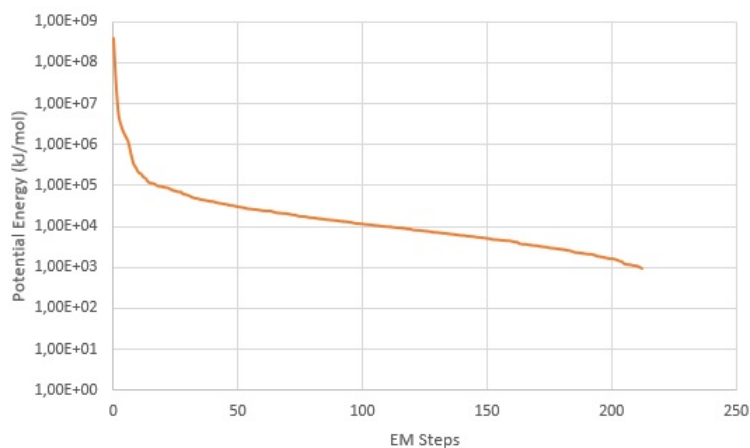


FIGURE 4.5: Energy minimization of hexane. Enabled logarithmic scaling and an added constant of 6000 kJ/mol for the sake of readability

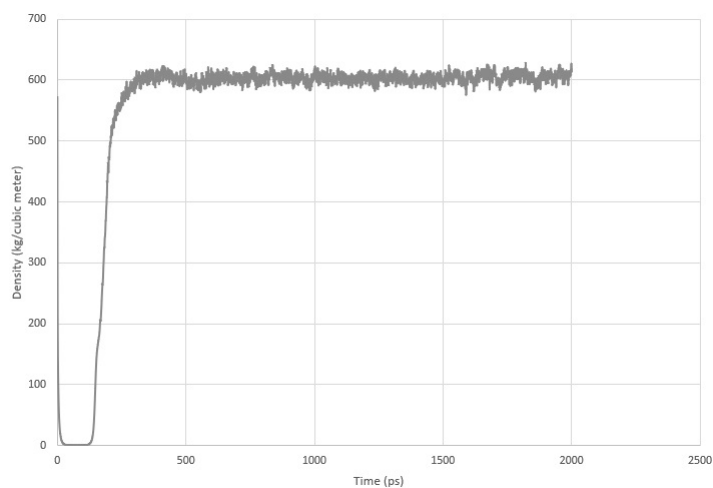


FIGURE 4.6: NPT Equilibration of hexane.

Furthermore, the commands `gmx gyrate` and `gmx polystat` provide the molecules' average radius of gyration and end-to-end distance respectively. Also provided as a measurement over time, these commands will see use in chapter 6 to study the effects of NPT Equilibration on large polymers.

Process Automation

The procedure of running simulations is a lengthy and tedious process. More importantly, GROMACS simply dumps simulation data in the directory from which a command is called. As a result, in order to keep simulation data sorted in an organized folder structure, file directories need to be included, leading to very lengthy shell commands.

An easy and user-friendly solution is to create a simple shell script that not only inserts all required commands automatically, but also keeps the data neatly organized.

```
/$ gmx gyrate
  -f base/states/hexane_500/afer_npt/hexane_500.xtc
  -s base/states/hexane_500/afer_npt/hexane_500.tpr
  -o base/states/hexane_500/afer_npt/data/hexane_500.xvg
```

FIGURE 5.1: Example of a lengthy command when using an organized directory structure

5.1 Shell scripts

GROMACS functions primarily through the interactions of a Unix shell. As a consequence, the process of simulation can be greatly simplified by an automated script that executes the respective commands on the shell for the user.

Shell scripts are available in a variety of scripting languages. The automation process in this report was accomplished with the common scripting language of *Bash*.

5.2 Folder Structure

The following section provides an overview of the proposed and subsequently used folder structure for the organization of this report's produced simulation data.

5.2.1 Working Directory

The overarching working directory includes three folders as well as the main executable file that launches the script itself.

- The "ps" directory includes subscripts that are run by the main shell script. These modules communicate with each other through a `coms.txt` file in the main directory. This is, in essence, the 'home' of the underlying program and contains no simulation data.

- The "tools" directory includes the Packmol executable and other conceivable software dependencies.
- The "base" directory is the actual area of interest from a MD point of view. All the data regarding simulations are stored here.

5.2.2 Base Directory

All simulation data is subdivided into 4 different folders.

- "mols": Exclusively includes the definitions of single molecules as described in [4.1](#). Each molecule is given its own folder which contains a respective ITP and PDB file.
- "procs": Includes MDP files corresponding to possible simulation procedures. `minim.nvt`, `npt.mdp` and `nvt.mdp` are included here.
- "setups": Includes the spread out molecule-configurations produced by Packmol. Individual setups will have their own sub-directory and follow a naming scheme equivalent to `hexane_500` for 500 molecules of hexane. Both the PDB file produced by Packmol and the corresponding `pure.inp` file used are included here.
- "states": Simulations get split up into multiple states. Each simulation will have its own overarching directory with sub-folders for the individual states. Each state-folder will include the produced `gro`, `trr` and `edr` files as well as the following two sub folders:
 - "bin": This is the folder that the simulations' binaries are executed in. It contains TPR, TOP and LOG files.
 - "data": Results obtained from data analysis commands, as outlined in [4.7](#), are included here.

An overview of the outlined structure can be seen in [Fig. 5.2](#).

5.3 Menu Structure

The goal of this shell script is to provide a more user friendly interface to the otherwise cyclic process of running simulations.

For this purpose, a menu-like interface was opted for. The available options of this menu are listed in [Fig. 5.3](#) and their functions are implied by their label.

Setting up new simulation will prompt all required data such as molecule type, amount, and volume to then create a new folder in the `base/setups` directory. All relevant files and commands will be executed by the script, leaving the building of the molecule as the only requirement to the user.

Similarly, running a simulation will execute a selected MDP procedure on an existing simulation state and subsequently create a new state that includes the results.

Analyzing and visualizing results will simply run the corresponding `vmd`, `xmgrace`, `gmx energy` etc. commands on a selected state.

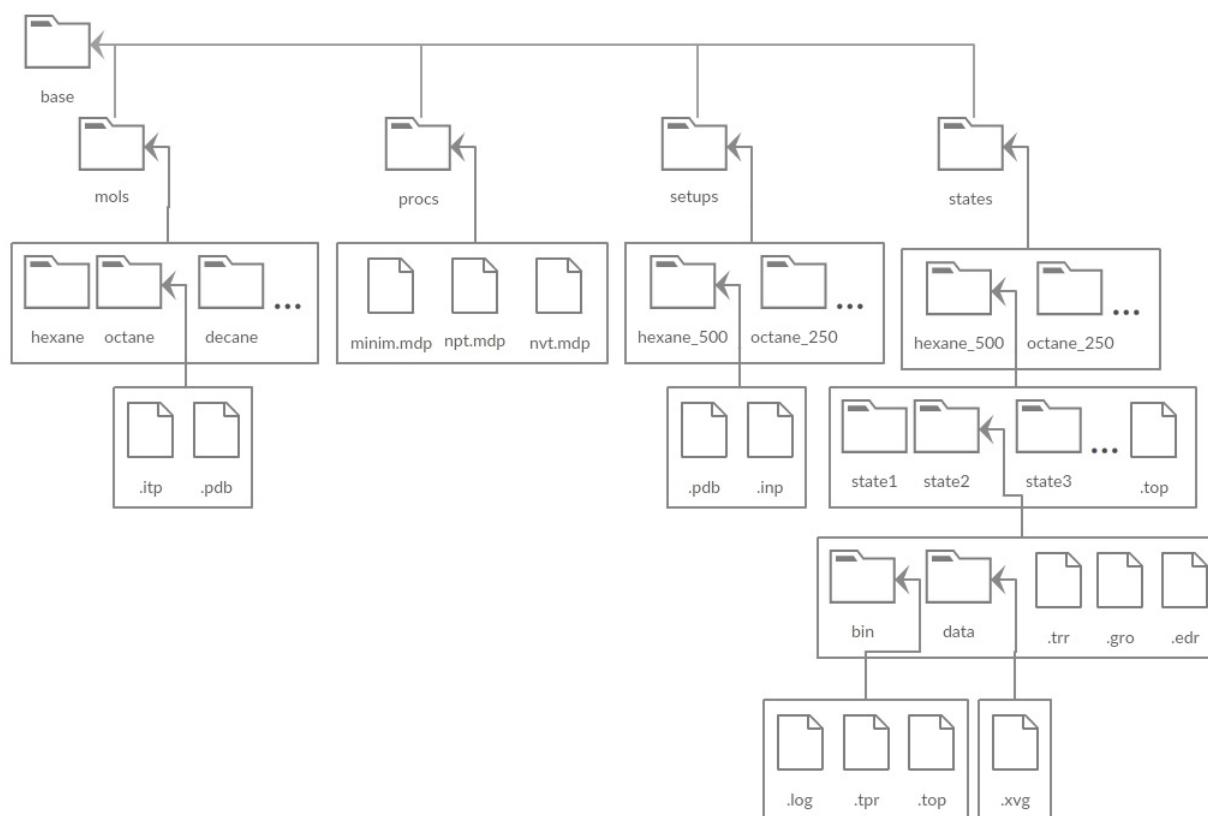


FIGURE 5.2: Folder structure of simulation data

```

/$ Select an option :
  1) Set up new simulation      3) Analyze results
  2) Run a simulation          4) Visualize results
>$ #

```

FIGURE 5.3: Shell view of the proposed and implemented script.

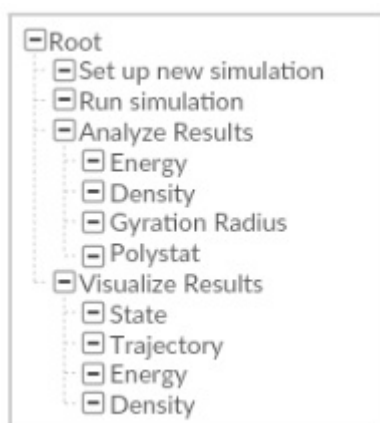


FIGURE 5.4: Menu structure of selectable options

Result Evaluation

The shell script described in the previous section greatly facilitates the process of running simulations while maintaining a structured organization to the received data.

With its aid, the same procedure described in chapter 4 was applied to a number of molecules. This chapter serves to evaluate the data obtained from these simulations.

6.1 Pure Alkanes

In terms of pure alkanes, four different molecule types were simulated. The initial one, Hexane (C_6H_{14}), is demonstrated in chapter 4. Afterwards, the same process was repeated for *Decane* ($C_{10}H_{22}$), *Hexadecane* (also known as *Cetane*) ($C_{16}H_{34}$) and *Dotriacontane* ($C_{32}H_{66}$).

The parameters for each simulation can be found in table 6.1.

<i>Name</i>	<i>Formula</i>	<i>MoleculeAmount</i>	<i>BoxDimensions</i> (Å)
Hexane	C_6H_{14}	500	50x50x50
Decane	$C_{10}H_{22}$	500	50x50x50
Cetane	$C_{16}H_{34}$	250	50x50x50
Dotriacontane	$C_{32}H_{66}$	500	100x100x100

TABLE 6.1: Used alkane simulation parameters

6.1.1 Partial Densities

Fig. 6.1 portrays the final partial densities of all alkanes after their NVT Equilibration.

The most noticeable aspect of the charted data is the fact that the densities are sorted relative to polymer length.

Since an increased polymer length strengthens the molecules' Van-Der-Waals interactions, this is precisely what one might expect on a chemical basis.

When comparing these values with the respective textbook-densities as seen in table 6.2, the largest discrepancies can be found for Hexane and Dotriacontane.

It is curious that the error margin is increased for very large and very small polymers, in particular if the fact is considered that the exact same MDP procedures were applied to all four simulations. While raising some questions about the inner workings of GROMACS, no clear conclusions can be drawn from a mere sample size of 4.

Another interesting observation is the fact that the partial densities of larger polymers oscillate to a higher degree than they do for smaller ones.

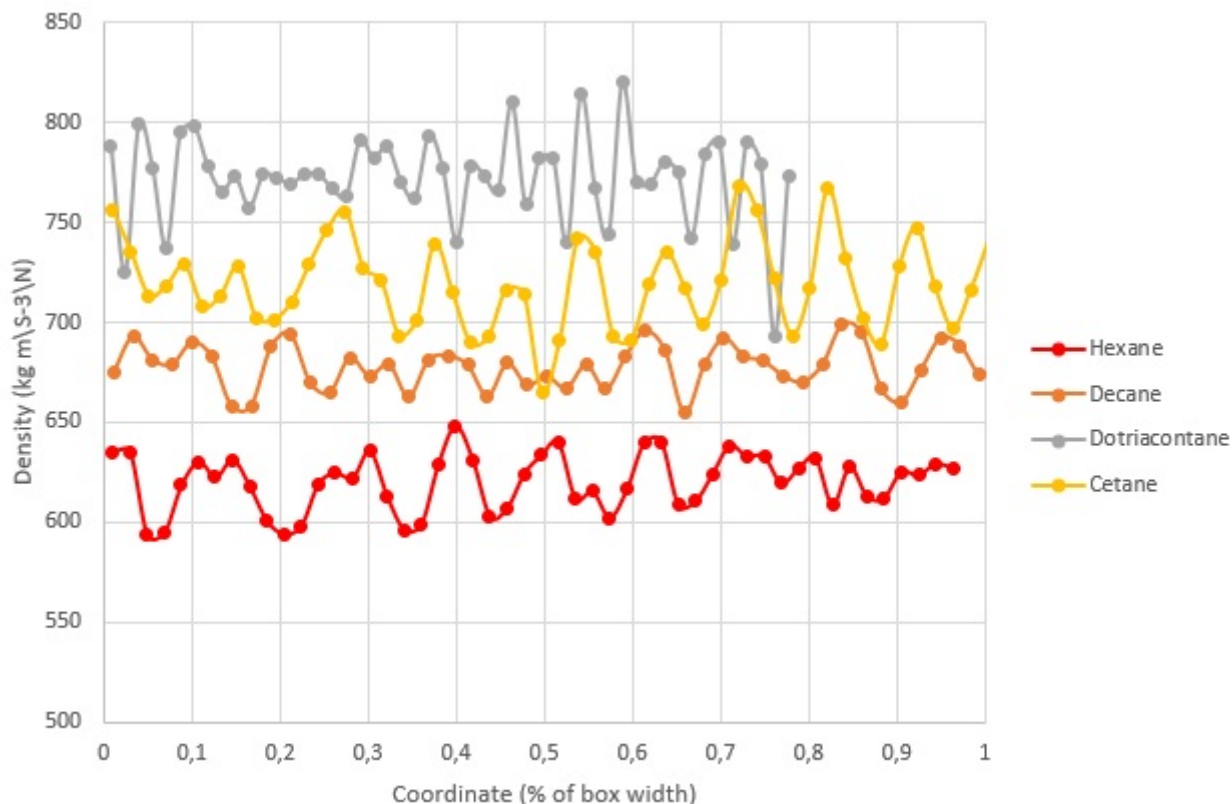


FIGURE 6.1: Partial densities of pure alkanes after NVT Equilibration

<i>Alkane</i>	<i>ExpectedDensity(kg/m³)</i>
Hexane	655
Decane	730
Cetane	770
Dotriacontane	800

TABLE 6.2: Alkane densities at standard conditions

6.1.2 Simulation Time

Running on the same CPU model, the wall clock times of all four simulations were recorded in table 6.3.

<i>Alkane</i>	<i>MoleculeAmount</i>	<i>AtomGroupAmount</i>	<i>SimulationTime</i>
Hexane	500	3000	67.558s
Decane	500	5000	70.275s
Cetane	250	4000	69.880s
Dotriacontane	500	16000	346.373s

TABLE 6.3: Simulation times required for NVT Equilibration

As one might reasonably expect, the time required by the simulation roughly correlates with the amount of atom groups that need to be tracked.

6.2 Radius of Gyration (Dotriacontane)

Lengthy polymers are more likely to curl and twist over the course of a simulation than shorter ones.

We can attempt to observe this phenomenon by examining the molecules' radius of gyration.

Plotting the average radius of gyration next to the systems density over the course of the NPT Equilibration, as done in Fig 6.2, reveals some curious behaviour:

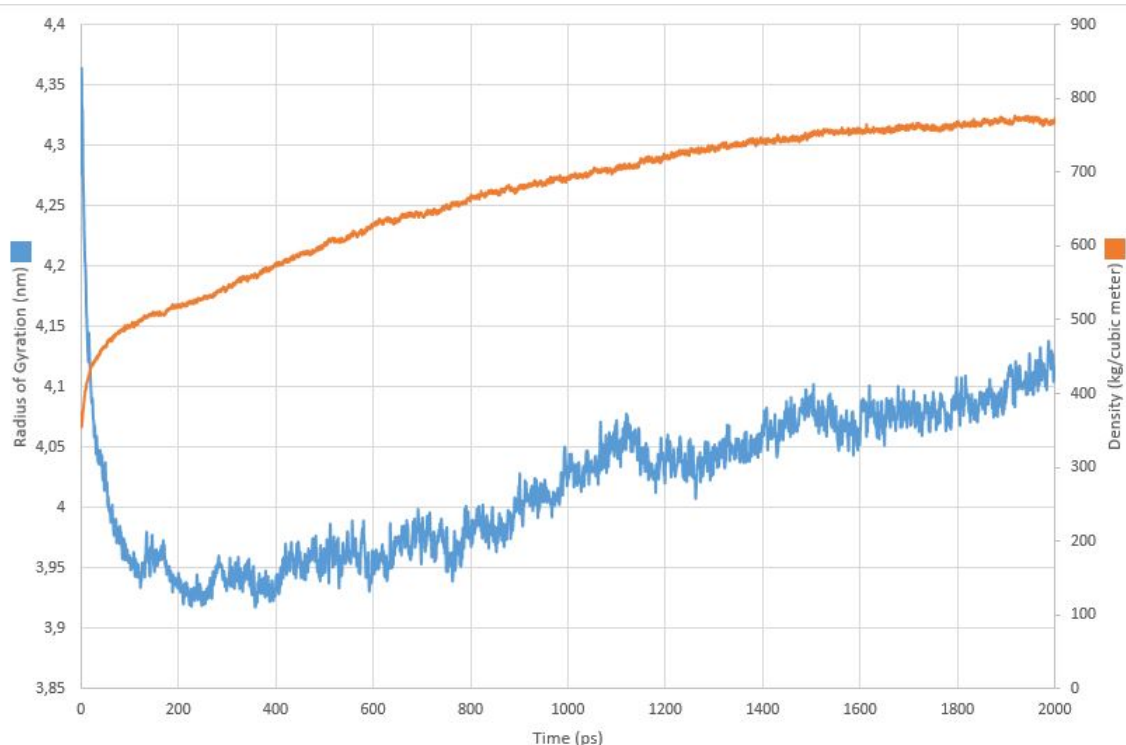


FIGURE 6.2: Dotriacontane radius of gyration and density during NPT Equilibration

The radius of gyration experiences a sharp drop within the first few pico-seconds of the simulation, but afterwards commences a steady *increase*, loosely correlated to the systems density.

At the beginning, the molecules have been given almost no time to move, thus, the radius of gyration is expected to start at a very high value. Once the polymers are given some room to maneuver, the molecules begin curling, dramatically decreasing the average radius of gyration.

With regard to the protracted, mild increase in the radius, one conceivable hypothesis is that the pressure and the radius of gyration are actually not correlated at all. Instead, the radius of gyration simply overshoots its' preferred conformation in the starting phase of the simulation and later normalizes into a more moderate value.

In order to investigate the causes of this observation, the following section will mix a far lesser quantity of dotriacontane with a large quantity of ethanol, and see if the pattern persists.

6.3 Complex Systems

Thus far, only pure alkane substances have been regarded.

To take the complexity a step further, this section will examine the results of a simulated molecule blend.

6.3.1 Setup

Arbitrarily chosen, the next simulation features 1000 Ethanol (C_2H_5OH) molecules mixed with 10 Dotriacontane ($C_{32}H_{66}$) molecules.

Since the default `forcefield.itp` library does not define oxygen atoms, an additional library, obtained through the ATB Repository [1], had to be linked in the topology file.

To accommodate for size of Dotriacontane molecules, the bounding box size was set to $80\text{\AA} * 80\text{\AA} * 80\text{\AA}$. Additionally, the simulation time of the NVT phase was doubled from the regular 100.000 steps.

Furthermore, changes to the shell script had to be done in order for it to support multi-molecule simulations.

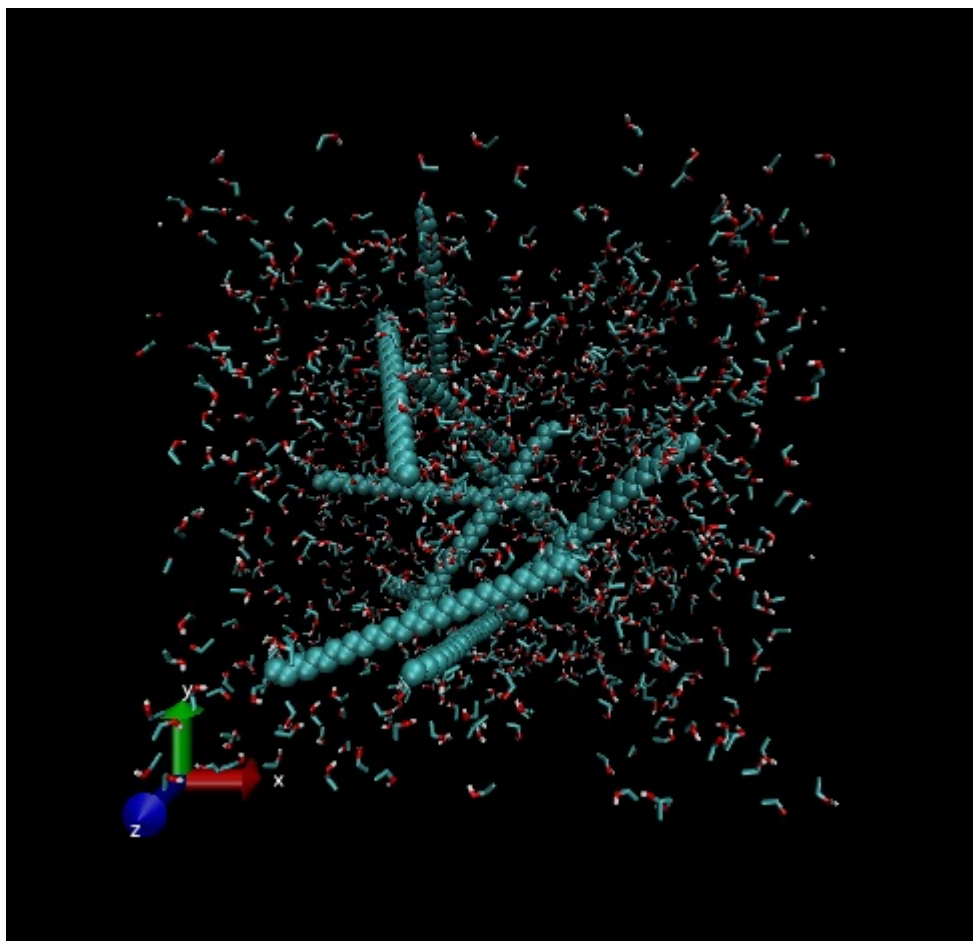


FIGURE 6.3: 1000 Ethanol and 10 Dotriacontane molecules after Energy Minimization

Despite these changes, the simulation process remained mostly the same as described in chapter 4.

6.3.2 End-to-End Distance

During the NPT Equilibration, the average end-to-end distance of the ten dotriacontane polymers, shown in Fig. 6.4, turns out to be exactly as one might expect.

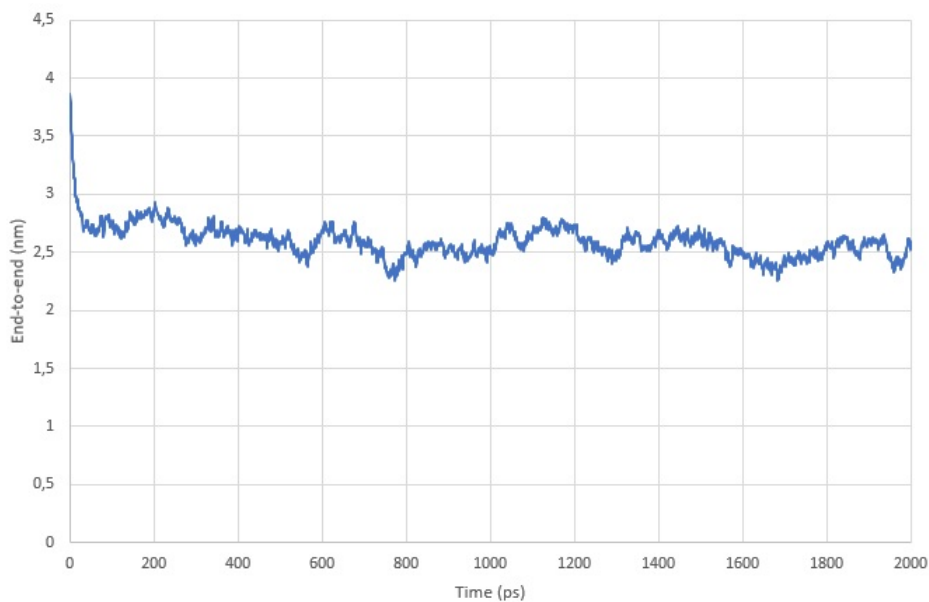


FIGURE 6.4: 1000 Ethanol and 10 Dotriacontane molecules after Energy Minimization

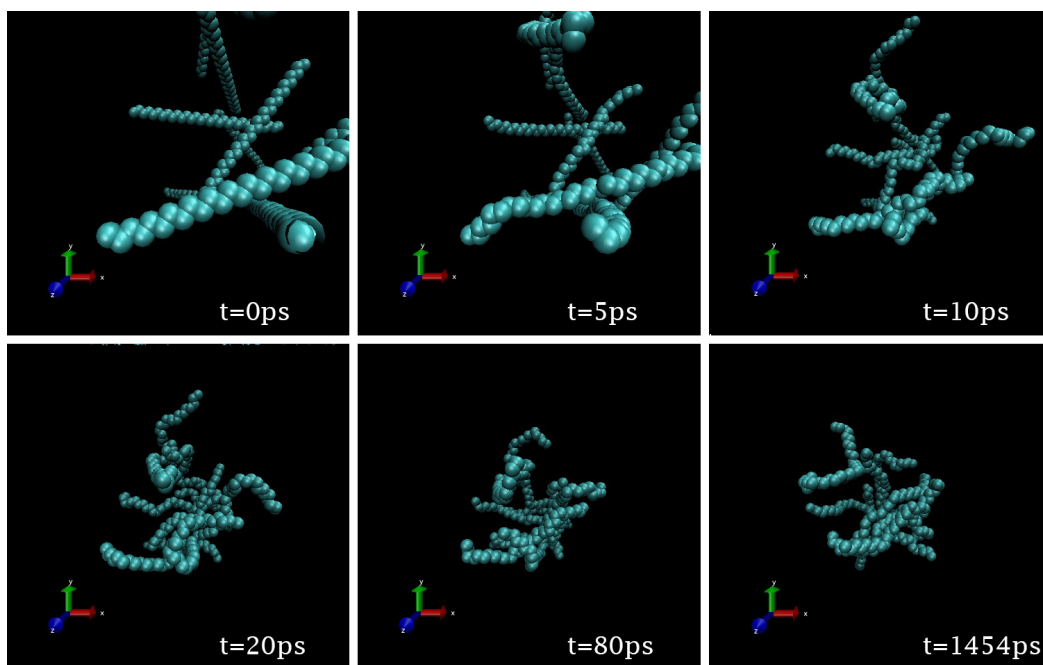


FIGURE 6.5: Stages of ethanol and dotriaconante NPT Equilibration. (Ethanol molecules hidden for the sake of visibilty)

Similarly to pure dotriacontane, the size of the polymer decreases dramatically in the starting phase of the simulation but normalizes to a constant value rather quickly and experiences no major fluctuations for the remainder of the simulation.

The NVT continues the same trend set by the latter part of the NPT: a steady polymer size with no major fluctuations.

6.4 Summary

To summarize, the essentials of GROMACS and additional tools were presented and subsequently applied in a series of simulations.

Four different alkanes were subject to a regular procedure carried out with a shell script written in bash.

This procedure consisted of setting up an initial starting point, minimizing the systems energy, normalizing pressure, temperature and subsequently keeping these values constant.

The resulting simulation data provided a reasonable approximation for each substances' density. Furthermore, an unexpected observation was made with regard to the radius of gyration of lengthy polymers. The same effect could not be observed for a small amount of polymers suspended in ethanol.

References

- [1] *Automated topology builder (atb) and repository, version 3.0*. URL: <https://atb.uq.edu.au/>.
- [2] *Gromacs documentation on file formats*. URL: <http://manual.gromacs.org/documentation/5.1/user-guide/file-formats.html>.
- [3] *Gromacs recommended visualization software*. URL: http://www.gromacs.org/Downloads/Related_Software/Visualization_Software.
- [4] *Packmol documentation*. URL: <http://m3g.iqm.unicamp.br/packmol/home.shtml>.
- [5] M. DOI, H. SEE, *Introduction to polymer physics*, 1996.
- [6] B. KAHL, *Shell script github repository*, 2019. URL: <https://github.com/Helliaca/gromacs-bash-script>.
- [7] U. OF ILLINOIS AT URBANA-CHAMPAIGN, *Visual molecular dynamics documentation*. URL: <https://www.ks.uiuc.edu/Research/vmd/current/docs.html>.