University of Vienna

Bachelor thesis

Molecular Dynamic Simulations

Mathias Fleisch

supervised by
Dr. Marcello Sega

August 25, 2016

Contents

1   Introduction

Computer simulations are often used when closed-form analytic solutions are not available for modelling the system of interest, especially for systems containing many interacting particles (many-body problem). Although the computational power of computers is continuously growing, even with the most recent machines it is not possible to take into account all chemical and mechanical processes at the atomic level (for example with so-called *ab-initio* methods) for systems larger than few hundreds of atoms, or for times longer than about 1 ns. Such a treatment would limit significantly the accessible time- and length-scale of a simulation. To overcome these gaps, a combination of different approaches can be used. One way to increase the accessible system sizes is to use effective potentials and coarse-grained models to reduce the number of degrees of freedom. For example, a simulation containing a large amount of molecules can be problematic if not only the interaction between molecules, but also the intra-molecular forces between the atoms are included. By treating the molecules as point-like particles and using an appropriate intermolecular potential (e.g. Lennard-Jones potential) the degrees of freedom are reduced and the system can be handled by classical molecular dynamics (MD) methods. MD simulations date back to the mid 50s, when Fermi, Pasta and Ulam first studied the effect on non-linearity in a vibrating string to test the ergodic hypothesis [1]. This early work was followed, for example, by MD simulations of liquid argon using the Lennard-Jones potential in 1964 by Rahman [2]. Nowadays, MD simulations are routinely used to investigate the properties systems in all branches of condensed-matter physics, chemistry, and also biology. MD simulations are based on the numerical integration of Newton's equation of motion. Widely used algorithms for integrating such differential equations are the Leapfrog or Velocity Verlet methods, because they are easy to implement, stable (symplectic) and add no significant computational cost to a simple Euler method. In its simplest form, MD samples the microcanonical ensemble, where the total energy of the system is conserved. By adding a stochastic and a friction term to Newton's equation of motion (the Langevin equation) one can sample instead the canonical ensemble at constant amount of particles (N), volume (V) and temperature (T). Constant temperature simulations are often easier to be compared to experimental data because the temperature of an experiment is often one of the control parameters, while the total energy is rarely accessible. One of the greatest challenges in MD simulations lies in the fact that for complex systems the potential energy surfaces have complicated features such as many local minima and high energy barriers. As a result, accurate sampling of the configuration space becomes a very difficult task. Different methods have been developed to accelerate the sampling. High temperatures are able to overcome energy barriers and can therefore be used to improve the conformational search ability. However, with a change in temperature, the ensemble distribution is also changed. This problem can be avoided by using an enhanced simulation method called parallel tempering, or replica exchange Langevin dynamics (RXLD) [4]. Replicas of the target system, with higher temperatures, are simulated in parallel and the configuration of these replicas (for example, the coordinates) are exchanged periodically between the replicas. To preserve the canonical nature of the ensemble, the exchange probability must be such that it preserves the detailed balance. This can be achieved by using, for example, the Metropolis criterion. Another method recently developed uses the fact that low frequency motions are important for conformational searching. By adding a guiding force that heats up low frequency motions and cools down high frequency motions, without changing the overall temperature, the sampling is increased significantly [5]. However, the guiding force also changes the target ensemble. By applying the same idea as in RXLD, but with different strengths of guiding forces, this behaviour can also be avoided. Such a simulation is called replica exchange self guided Langevin dynamics (RXSGLD) [6]. SGLD causes less perturbation on conformational distribution, therefore exchanges between different SGLD stages are more likely than for a RXLD simulation. The purpose of this bachelor thesis is to present the necessary theoretical background for the mentioned methods (chapter 2) and to introduce several techniques which eventually reduce the computational cost of a simulation (chapter 3). Parallel computing is of particular interest where different replicas have to be computed in parallel. OpenMP provides an easy-to-use API for parallel computing. The implementation of these methods was done from scratch using object-oriented C++. Several checks were performed

during the implementation to ensure a simulation produces correct values. Simple model systems are chosen to show the improvements introduced with the enhanced simulation methods. As test systems, both the Lennard-Jones fluid and coarse-grained models of polymers have been used, in combination with a double-well external potential, which models in a controlled way a free energy landscape with a single barrier. The combination of RXLD and SGLD turned out to outperform dramatically the RXLD for the polymer case, showing that the redistribution of thermal energy from fast to slow degrees of freedom is a powerful concept when applied to configurational searches.

2 Theoretical Background

2.1 Langevin Dynamics (LD)

The Langevin equation is an equation of motion with additional stochastic and friction terms. The equation for the i-th particle is

$$m_i \ddot{\vec{x}}_i = \vec{F}_i - \gamma_i m_i \dot{\vec{x}}_i + \vec{R}_i, \tag{1}$$

where $\vec{F}_i$ is the force acting on particle i,, γ_i is the collision frequency and $\vec{R}_i$ represents a random force which is related to the mass m_i, the temperature T and the collision frequency γ_i by

$$\langle \vec{R}_i(t) \vec{R}_j(t') \rangle = 2 m_l k_B T \gamma_l \delta_{i,j} \delta(t - t') \tag{2}$$

and

$$\langle \vec{R}_i(t) \rangle = 0. \tag{3}$$

This is equivalent to

$$\vec{R}_i(t) = \sqrt{2 k_B T \gamma_i m_i} \eta(t), \tag{4}$$

where $\eta(t)$ is a white-noise with unit variance. Without the friction term and the stochastic force, the Langevin equation is a regular Newton equation of motion and will therefore conserve energy. Such a simulation will sample the microcanonical ensemble. With the two additional terms, the system is coupled to a heat bath and the temperature stays, within fluctuations, constant. Such a simulation will sample the canonical ensemble.

2.2 Self-Guided Langevin Dynamics (SGLD)

A method for enhancing conformational searching is by using a self-guided Langevin equation [7]

$$m_i \ddot{\vec{x}}_i = \vec{F}_i - \gamma_i m_i \dot{\vec{x}}_i + \vec{R}_i + \vec{g}_i \tag{5}$$

This equation contains a guiding force $\vec{g}_i$ with the following expression

$$\vec{g}_i = \lambda_i \gamma_i (\tilde{\vec{p}}_i - \xi \vec{p}_i) \tag{6}$$

λ_i is the guiding factor and defines the strength of the guiding force. ˜ denotes a local average and can be calculated, for any property P, in the following way

$$\tilde{P}(t) = \left(1 - \frac{\Delta t}{t_L}\right) \tilde{P}(t - \Delta t) + \frac{\Delta t}{t_L} P(t) \tag{7}$$

t_L is the local average time. To avoid an energy flow into or out of the system, the constraint parameter ξ is chosen such that there is no net energy input from the guiding force. Therefore ξ is calculated at every timestep by

$$\xi = \frac{\sum_i \lambda_i \gamma_i \tilde{\vec{p}}_i \cdot \dot{\vec{x}}_i}{\sum_i \lambda_i \gamma_i \vec{p}_i \cdot \dot{\vec{x}}} \tag{8}$$

The guiding force enhances the low frequency motions like rotations and suppresses high frequency motions like bond vibrations. The low frequency motions will become hotter and the low frequency motions cooler, without changing the overall temperature. Because of the guiding force, SGLD has its own conformational distribution. The partition function of the SGLD-Ensemble has the following form [7]

$$\Theta_{SGLD} \approx \sum \exp\left(-\frac{\lambda_{lf}\chi_{lf}\tilde{E}_p}{k_BT} - \frac{\lambda_{hf}\chi_{hf}(E_p - \tilde{E}_p)}{k_BT}\right) \tag{9}$$

The summation runs over all microscopic states. λ_{lf} and λ_{hf} are called low frequency energy factor and high frequency energy factor, respectively. χ_{lf} and χ_{hf} are calculated according to projections of the guiding forces in the direction of the friction forces and are called low frequency collision factor and high frequency collision factor. These quantities are calculated as follows

$$\lambda_{lf} = \frac{\langle \sum_i (\tilde{\vec{F}}_i + \tilde{\vec{g}}_i - \gamma_i\tilde{\vec{p}}_i) \cdot \tilde{\vec{F}}_i \rangle}{\langle \sum_i (\vec{F}_i - \tilde{\vec{F}}_i) \cdot (\vec{F}_i - \tilde{\vec{F}}_i) \rangle} \tag{10}$$

$$\lambda_{hf} = \frac{\langle \sum_i (\vec{F}_i - \tilde{\vec{F}}_i + \vec{g}_i - \tilde{\vec{g}}_i - \gamma_i(\vec{p}_i - \tilde{\vec{p}}_i)) \cdot (\vec{F}_i - \tilde{\vec{F}}_i) \rangle}{\langle \sum_i (\vec{F}_i - \tilde{\vec{F}}_i) \cdot (\vec{F}_i - \tilde{\vec{F}}_i) \rangle} \tag{11}$$

$$\chi_{lf} = 1 - \frac{\langle \sum_i \tilde{\vec{g}}_i \gamma_i \tilde{\vec{p}}_i \rangle}{\langle \sum_i \gamma_i^2 \tilde{\vec{g}}_i \cdot \tilde{\vec{g}}_i \rangle} \tag{12}$$

$$\chi_{hf} = 1 - \frac{\langle \sum_i \gamma_i(\vec{g}_i - \tilde{\vec{g}}_i) \cdot (\vec{p}_i - \tilde{\vec{p}}_i) \rangle}{\langle \sum_i \gamma_i^2(\vec{p}_i - \tilde{\vec{p}}_i) \cdot (\vec{p}_i - \tilde{\vec{p}}_i) \rangle} \tag{13}$$

The partition function Θ_{LD} from the LD simulation is related with Θ_{SGLD} by the following equation

$$\begin{aligned}
\Theta_{LD} &= \sum \exp\left(\frac{E_p}{k_BT}\right) \\
&= \sum \exp\left(-\lambda_{lf}\chi_{lf}\frac{\tilde{E}_p}{k_BT} - \lambda_{hf}\chi_{hf}\frac{E_p - \tilde{E}_p}{k_BT}\right) \\
&\quad \times \exp\left((\lambda_{lf}\chi_{lf} - 1)\frac{\tilde{E}_p}{k_BT} + (\lambda_{hf}\chi_{hf} - 1)\frac{E_p - \tilde{E}_p}{k_BT}\right) \\
&= \Theta_{SGLD}\langle w_{SGLD}\rangle_{SGLD}
\end{aligned} \tag{14}$$

With the SGLD reweighting factor w_{SGLD}

$$w_{SGLD} = \exp\left((\lambda_{lf}\chi_{lf} - 1)\frac{\tilde{E}_p}{k_BT} + (\lambda_{hf}\chi_{hf} - 1)\frac{E_p - \tilde{E}_p}{k_BT}\right) \tag{15}$$

With the reweighting factor, any property P in a SGLD simulation can be calculated as

$$\langle P \rangle_{LD} = \frac{\langle Pw_{SGLD}\rangle_{SGLD}}{\langle w_{SGLD}\rangle_{SGLD}} \tag{16}$$

2.3 Reduced Units

As molecular dynamics actions take place on a small time- and space-scale, it's convenient to change the unit system to the system of reduced units (MD units) [8]. Following changes are made

$$r^* \rightarrow r\sigma \tag{17}$$

$$E^* \rightarrow E\epsilon \tag{18}$$

$$m^* \rightarrow mm_a \tag{19}$$

where σ and ϵ are parameter of the Lennard-Jones potential (ϵ governs the strength of the interaction and σ defines a length scale, see section 2.4.1) and m_a is the atomic mass. In order to derive the factor for the conversion factor of time, one can use the fact that the formula for the kinetic energy shouldn't change under unit system transformations. This leads to

$$t^* \rightarrow t\sqrt{\frac{m_a\sigma^2}{\epsilon}} \tag{20}$$

By setting $k_B = 1$ the MD unit of temperature is now also defined. The following table contains the most used quantities, as well as values for argon [8]

Physical quantity	Unit	Value for Ar
Length	σ	$3.4 \cdot 10^{-10}$ m
Energy	ϵ	$1.65 \cdot 10^{-21}$ J
Mass	m	$6.69 \cdot 10^{-26}$ kg
Time	$\sigma(m/\epsilon)^{1/2}$	$2.17 \cdot 10^{-12}$ s
Velocity	$(\epsilon/m)^{1/2}$	$1.57 \cdot 10^2$ m/s
Force	ϵ/σ	$4.85 \cdot 10^{-12}$ N
Pressure	ϵ/σ^3	$4.20 \cdot 10^7$ Nm^{-2}
Temperature	ϵ/k_B	120 K

Table 1: Conversation factors for MD units and specific values for argon

For the mass of argon the reduced timeunit corresponds to $2.161 \cdot 10^{-12}$ s. Thus a simulation time-step of $\Delta t = 0.005$ would correspond to approx. 10^{-14} s. For a liquid density of 0.942 g/cm^3 the reduced length of a box sized simulation region is $L = 1.218N^{1/3}$. Quantities with a star, e.g. ρ^*, are given in MD units.

2.4 Potentials

2.4.1 Lennard-Jones Potential

The Lennard-Jones potential approximates the interaction potential between two neutral atoms or molecules as

$$U = 4\epsilon\left(\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^6\right) \tag{21}$$

where $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$ and $r_{ij} = |\vec{r}_{ij}|$ is the distance between the particle i and j, σ is the finite distance at which the inter-particle potential is zero and ϵ is the depth of the potential well.
For simulations it is necessary to cut off the potential at a given radius r_c in order to reduce the simulation time. The potential then becomes

$$U_{LJ} = \begin{cases} 4\epsilon \left(\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right) & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c \end{cases} \tag{22}$$

To avoid a discontinuity at the cut-off it is common to use a truncated and shifted potential. The following potential satisfies this condition

$$U = \begin{cases} U_{LJ} - U_{LJ}(r_c) & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c \end{cases} \tag{23}$$

The force is

$$-\vec{\nabla} U = \vec{F}_{ij} = \begin{cases} \frac{48\epsilon}{\sigma^2} \left(\left(\frac{\sigma}{r_{ij}} \right)^{14} - \frac{1}{2} \left(\frac{\sigma}{r_{ij}} \right)^{8} \right) \vec{r}_{ij} & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c \end{cases} \tag{24}$$

When reduced units are used, the force reduces to the following form

$$-\vec{\nabla} U = \vec{F}_{ij} = \begin{cases} 48 \left(r_{ij}^{-14} - \frac{1}{2} r_{ij}^{-8} \right) \vec{r}_{ij} & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c \end{cases} \tag{25}$$

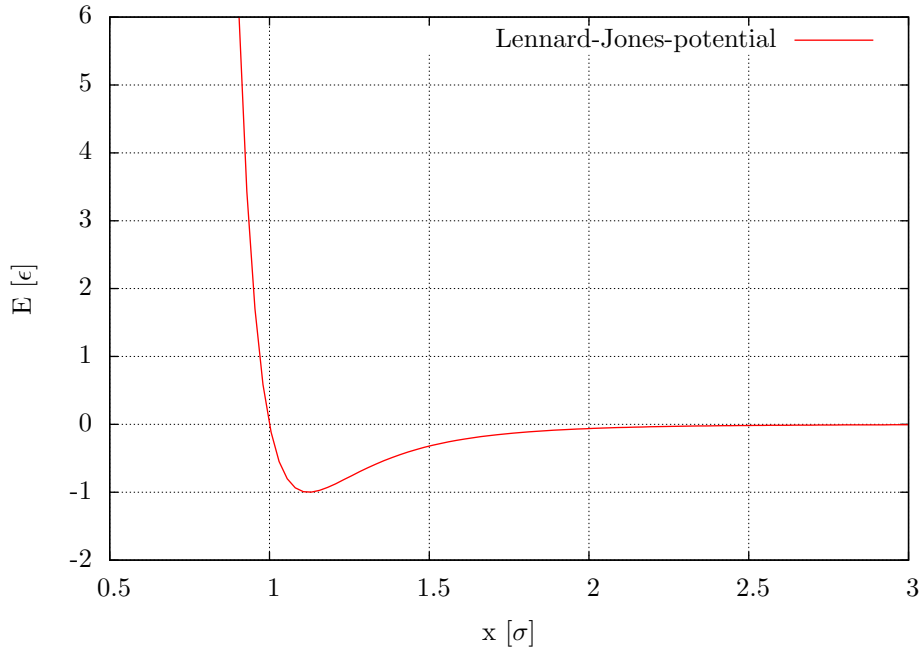A plot of the Lennard-Jones potential in reduced units can be seen in figure 1.

Figure 1: Lennard-Jones potential in reduced units

A simulation containing only the Lennard-Jones potential is referred to as a Lennard-Jones fluid.

2.4.2 External Double Well Potential

To demonstrate the conformational search ability a double well potential, similar to the study of Xiongwu Wu et al. [7], is used. It has the following form

$$U(x, y, z) = U(y) = \frac{b}{w^4} y^2 (y - w)^2 + \frac{s}{w} y \tag{26}$$

b defines the height of the potential between the two wells, w the location of the second well and s the energy difference between the two wells. Figure 2 shows the y-component of the double well potential for different values of s.
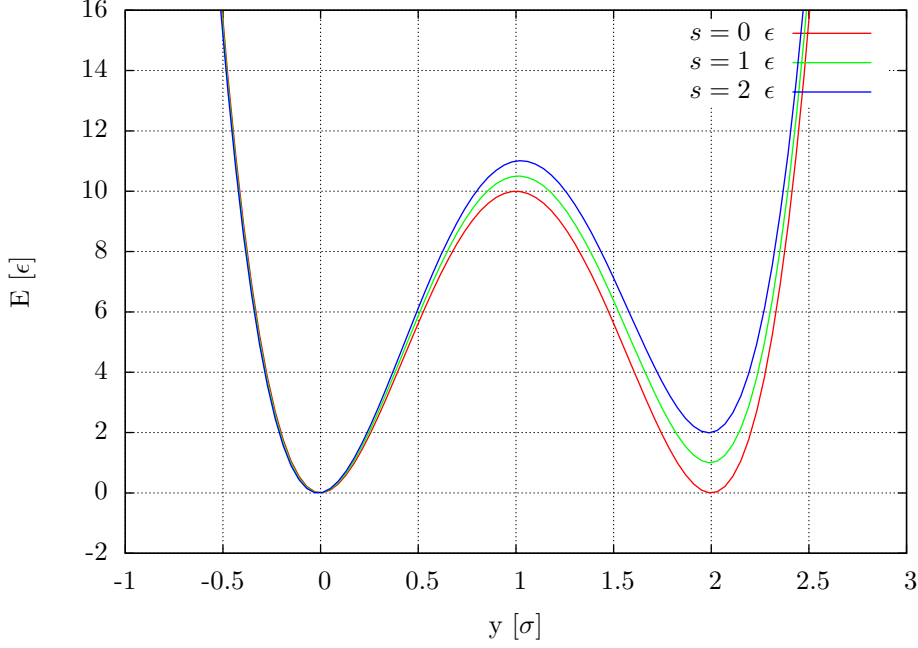

Figure 2: y-component of double well potential for three different parameters s, $b = 160\ \epsilon$, $w = 2$ [6]

The partition function can be separated for each degree of freedom [6],

$$Q_y = \int_{-\infty}^{\infty} dy \exp\left(-\frac{\frac{b}{w^4} y^2 (w - y)^2 + \frac{s}{w} y}{k_B T} \right) \tag{27}$$

And therefore

$$\rho_y(y) = \frac{1}{Q_y} \exp\left(-\frac{\frac{b}{w^4} y^2 (w - y)^2 + \frac{s}{w} y}{k_B T} \right) \tag{28}$$

This property can be used to verify the implementation and check the conformational search ability.

2.5 Replica Exchange

2.5.1 Parallel Tempering (RXLD)

Parallel tempering, also known as temperature replica exchange molecular dynamics (RXLD) [4], is an enhanced sampling method. It works by simulating different replicas, each in the canonical ensemble at different temperatures $T_1 < T_2 < ... < T_M$, where T_1 is the temperature of the system of interest. Periodically the temperatures are exchanged between the replicas, according to a transition probability. This condition is such that it preserves the statistical ensemble, in

particular the detailed balance. This means that the probability of exchanging a replica is the same as exchanging it back. This can be satisfied by the usual Metropolis criterion:

$$p\left(\vec{X}_i \to \vec{X}_j\right) = \min\left(1, \exp\left(\left(E_p^{(i)} - E_p^{(j)}\right)\left(\beta^{(i)} - \beta^{(j)}\right)\right)\right) \tag{29}$$

where $\beta^{(i)} = \frac{1}{k_B T_i}$ and $E_p^{(i)}$ is the potential energy of the i-th replica. To satisfy $\langle E_{kin} \rangle = \frac{3}{2}Nk_BT$ we need to rescale the momenta of all atoms in the replicas uniformly

$$p_i = p_i\sqrt{\frac{T_j}{T_i}} \qquad \text{and} \qquad p_j = p_j\sqrt{\frac{T_i}{T_j}} \tag{30}$$

Another method consists in exchanging coordinates instead of exchanging temperatures

$$\vec{X}_i(x_i, p_i, T_i) \xleftrightarrow{exchange} \vec{X}_j(x_j, p_j, T_j) \Rightarrow \vec{X}_i(x_j, p_i, T_i) \text{ and } \vec{X}_j(x_i, p_j, T_j) \tag{31}$$

The advantage of this method is that there is less post-processing needed to obtain values from the simulation. For the temperature exchange one needs to keep track of the exchanges and reconstruct the trajectories for the initial temperature afterwards.

2.5.2 Replica Exchanging Self-Guided Langevin Dynamics (RXSGLD)

RXSGLD relies on the same principle as RXLD. Instead of using different temperatures for the replicas, different guiding factors are used for each replica. Based on the partition function, equation 9, the distribution probability of a state $\vec{X}_m^{(i)}$ with temperature T_m is given by [7]

$$\begin{aligned}\rho_{SGLD}(X_m^{(i)}) &= \frac{1}{\Theta_{SGLD}^{(m)}} \exp\left(-\frac{\lambda_{lf}^{(m)}\chi_{lf}^{(m)}\tilde{E}_p^{(i)}}{k_BT_m} - \frac{\lambda_{hf}^{(m)}\chi_{hf}^{(m)}\left(E_p^{(i)} - \tilde{E}_p^{(i)}\right)}{k_BT_m}\right) \\ &= \frac{1}{\Theta_{SGLD}^{(m)}} \exp\left(-\tilde{\mu}_m\tilde{E}_p^{(i)} - \mu_m E_p^{(i)}\right)\end{aligned} \tag{32}$$

with parameters defined as

$$\tilde{\mu}_m = \frac{\lambda_{lf}^{(m)}\chi_{lf}^{(m)}\tilde{E}_p^{(i)}}{k_BT_m} \tag{33}$$

and

$$\mu_m = \frac{\lambda_{hf}^{(m)}\chi_{hf}^{(m)}\left(E_p^{(i)} - \tilde{E}_p^{(i)}\right)}{k_BT_m} \tag{34}$$

The exchange probability can then be expressed in the following way [7]

$$\begin{aligned}\pi_{RX}\left(\left\{\vec{X}_m^{(i)}, \vec{X}_n^{(j)}\right\} \to \left\{\vec{X}_m^{(j)}, \vec{X}_n^{(i)}\right\}\right) \approx \exp\Big(&-(\tilde{\mu}_m - \tilde{\mu}_n)\left(\tilde{E}_p\left(\vec{X}_n^{(j)}\right) - \tilde{E}_p\left(\vec{X}_m^{(i)}\right)\right) \\ &- (\mu_m - \mu_n)\Big)\left(E_p(\vec{X}_n^{(j)}) - E_p\left(\vec{X}_m^{(i)}\right)\right)\Big)\end{aligned} \tag{35}$$

The exchange criterion stays the same (see equation 29) and if $\lambda = 0$ the exchange probability is the same as in equation 29. SGLD causes less pertubation on conformational distribution and therefore the exchange probability is much higher than in a RXLD simulation. From equations 29, 35 and figure 3 it can be seen that an exchange is only possible if the potential energy distributions overlap. Especially for big systems the exchange probability is much higher for RXSGLD simulations than for RXLD simulations. This can be seen in figure 3.
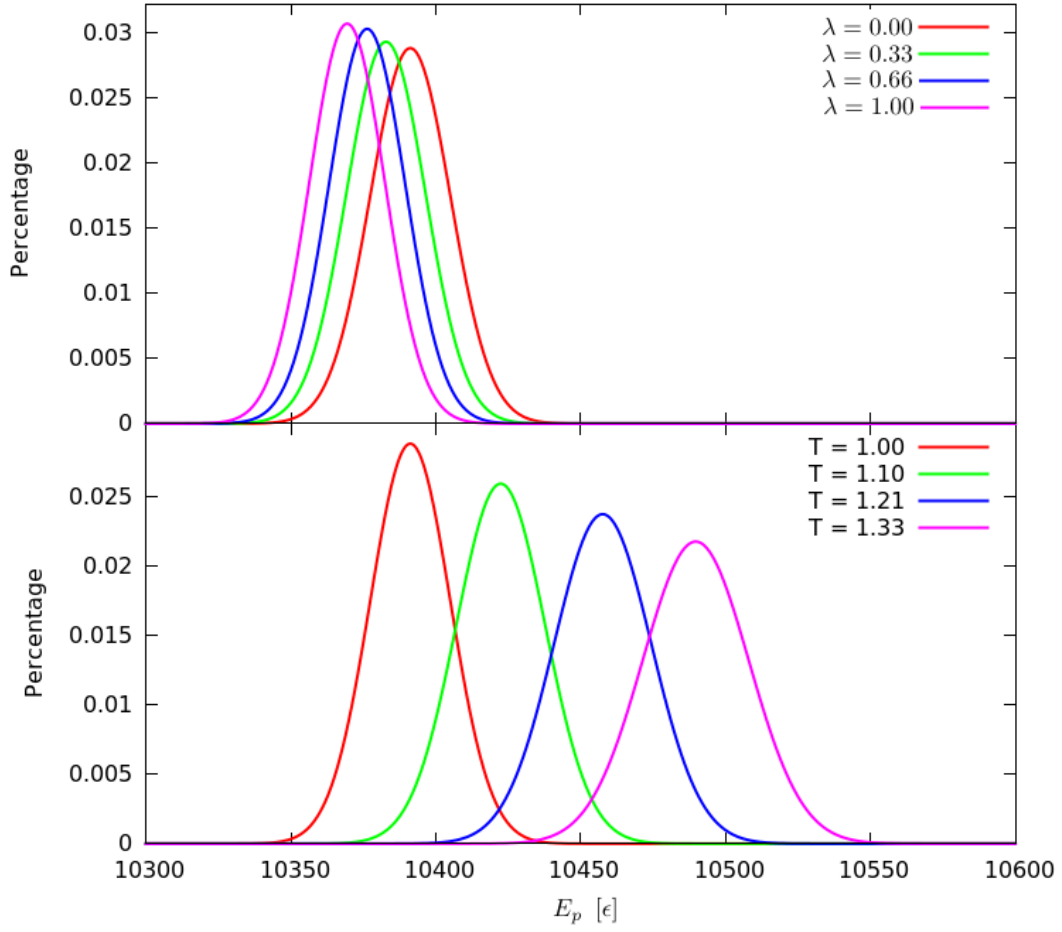
Figure 3: Potential energy distributions for a polymer ($N = 500$, FENE-potential: $R_0^* = 1.5\ \sigma$, $k^* = 30\ \epsilon/\sigma^2$, $\Delta t^* = 0.005\sigma(m/\epsilon)^{1/2}$, $\gamma^* = 1\ \sigma^{-1}(m/\epsilon)^{-1/2}$) for 4 different temperatures and 4 different guiding factors

2.6 Polymers

A simple way to represent a polymer is by using springs between point-like particles. This model is called the Rouse model [9]. A schematic representation of this model can be seen in figure 4.
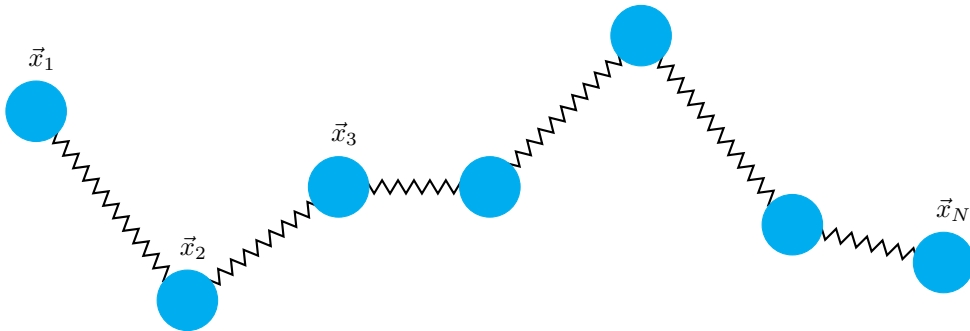

Figure 4: Rouse model

2.6.1 Gaussian Chain

A possible way of defining the potential due to the springs is

$$U_{ij} = \frac{1}{2}k\left(\vec{x}_i - \vec{x}_j\right)^2 \tag{36}$$

with spring constant $k = \frac{3k_B T}{b^2}$ where b is the statistical segment length and $b^2 = \frac{\langle R_{ee}^2 \rangle}{N-1}$ with $\langle R_{ee}^2 \rangle$ the second moment of end-to-end vector.

The force on the i-th bond is

$$\vec{F}_i = k\left(\vec{x}_{i+1} - 2\vec{x}_i + \vec{x}_{i-1}\right) \tag{37}$$

In the cases where $i = 1$ set $\vec{x}_0 = \vec{x}_1$ and where $i = N$ set $\vec{x}_{N+1} = \vec{x}_N$.

This model is called a Gaussian chain, because every bond length satisfies the distribution [10]

$$\psi(\vec{x}) = \left(\frac{3}{2\pi b^2}\right)^{\frac{3}{2}} \exp\left(-\frac{3\vec{x}^2}{2b^2}\right) \tag{38}$$

For given m,n the distribution of the vector $\vec{x}_n - \vec{x}_m = \sum_{i=m}^{n-1}(\vec{x}_{i+1} - \vec{x}_i)$ is given by

$$\phi(\vec{x}_n - \vec{x}_m, n - m) = \left(\frac{3}{2\pi b^2 |n-m|}\right)^{\frac{3}{2}} \exp\left(-\frac{3(\vec{x}_n - \vec{x}_m)^2}{2|n-m|b^2}\right) \tag{39}$$

2.6.2 Normal Coordinates

To provide meaningful results, physical quantities should be sampled only once the system has equilibrated. For a Rouse chain, it is possible to give an analytic estimate of this time by introducing normal coordinates [11]. The equation of motion is

$$m_i\ddot{\vec{x}}_i = k(\vec{x}_{i+1} - 2\vec{x}_i + \vec{x}_{i-1}) - \gamma m_i\dot{\vec{x}}_i + \vec{R}_i \tag{40}$$

The term $m_i\ddot{\vec{x}}_i$ is important for vibrational motion, but is negligible for long time scales (μs to s). The mass m_i is dropped for simplicity. The equation of motion becomes

$$\gamma\dot{\vec{x}}_i = k(\vec{x}_{i+1} - 2\vec{x}_i + \vec{x}_{i-1}) + \vec{R}_i \tag{41}$$

The i-th normal coordinate is defined by

$$\vec{q}_i = \frac{1}{N}\sum_{n=1}^{N}\cos\left(\frac{in\pi}{N}\right)\vec{x}_n \tag{42}$$

The 0-th normal coordinate describes the center of mass motion

$$\vec{q}_0 = \frac{1}{N}\sum_{n=1}^{N}\vec{x}_n \tag{43}$$

The first normal coordinate is a vector from the center of mass of the first half pointing to the center of mass of the second half. Therefore $\vec{q}_1$ describes the rotation of the chain.

The equation of motion in normal coordinates is

$$\xi_i\dot{\vec{q}}_i = -k_i\vec{q}_i + \vec{g}_i \tag{44}$$

with

$$\xi_i = \begin{cases} N\gamma & \text{if } i = 0 \\ 2N\gamma & \text{if } i \neq 0 \end{cases} \tag{45}$$

The force constant k_i of the i-th mode is given as

$$k_i = k \frac{\xi_i}{\gamma} \left(\frac{i\pi}{N}\right)^2 = \frac{6\pi^2 k_B T}{Nb^2} i^2 \tag{46}$$

The random force $\vec{g}_i$ is defined by

$$\vec{g}_i = \frac{\xi_i}{\gamma} \frac{1}{N} \sum_{n=1}^{N} \cos\left(\frac{in\pi}{N}\right) \vec{R}_n \tag{47}$$

The equation for $\vec{q}_i$ does not depend on other $\vec{q}_j (j \neq i)$ and is therefore decoupled. The relaxation time is defined as

$$\tau_i = \frac{\xi_i}{k_i} \tag{48}$$

Expressed with the relaxation time the equation of motion is

$$\dot{\vec{q}}_i = -\frac{1}{\tau_i} \vec{q}_i + \frac{1}{\xi_i} \vec{g}_i \tag{49}$$

For the Rouse-Model:

$$\tau_i = \frac{\gamma N^2 b^2}{3\pi^2 k_B T} \frac{1}{i^2} \tag{50}$$

As $\frac{1}{\tau_0} = 0$, τ_1 can be used as an estimate when equilibrium is achieved.

2.6.3 Excluded Volume

If only the harmonic potential is used, two particles can be at the same place. If the Lennard-Jones potential is included as well, two particles cannot occupy the same place and the equation of motion becomes

$$m_i \ddot{\vec{x}}_i = k(\vec{x}_{i+1} - 2\vec{x}_i + \vec{x}_{i-1}) + 48 \left(r_{ij}^{-14} - \frac{1}{2} r_{ij}^{-8} \right) \vec{r}_{ij} - \gamma_i m_i \dot{\vec{x}}_i + \vec{R}_i \tag{51}$$

$\langle R_{ee}^2 \rangle$ is now not longer proportional to $N - 1$, but

$$\langle R_{ee}^2 \rangle \propto (N-1)^{2\nu} \tag{52}$$

where ν is about 3/5 [10]. The Lennard-Jones potential is cut off at the minimum ($\approx 1.122 \ \sigma$) and only the repulsive part is used.

2.6.4 FENE-Potential

The springs in the Gaussian chain are infinitely expandable. This causes non-physical chain configurations, especially in a double well potential where it could happen that one particle is in one of the wells, where another particle, connected by a spring, is in the other. An alternative potential is the FENE-Potential [12]

$$U_{ij} = \begin{cases} -\frac{1}{2} k R_0^2 \ln \left(1 - \left(\frac{r_{ij}}{R_0}\right)^2\right) & r_{ij} \leq R_0 \\ 0 & r_{ij} > R_0 \end{cases} \tag{53}$$

The maximum bond length with the FENE-Potential is R_0. Common values for the constants are $k = 30 \ \epsilon/\sigma^2$ and $R_0 = 1.5 \ \sigma$.

3 Implementation Details

3.1 Boundary Conditions

For a simulation of a Lennard-Jones fluid, a large amount of particles would be necessary to avoid the influence of boundaries. To avoid this problem, periodic boundary conditions are often used. A small cube with side-length L is filled with particles for a given density. If a particle leaves the simulation region, the coordinate is updated, such that the particle enters the simulation region on the other side. The particle is wrapped around the simulation region. Such a system simulates an infinite, space-filling fluid. The implementation needs two steps:

Data: New position x_i of particle, simulation region size L
Result: If particle leaves simulation region, update position based on periodic boundary
conditions
initialization;
if $x_i > L$ **then**
  $\mid$ $x_i = x_i - L$;
end
if $x_i < 0$ **then**
  $\mid$ $x_i = x_i + L$;
end

Algorithm 1: Particle wraparound

This can be applied in any direction. For the double well potential, only the x and z boundaries have to be periodic. The y boundary is limited by the potential. For polymers another possibility is to reset the center of mass every n-th timestep to the origin. This can reduce the computational cost.

3.2 Neighborlist

For a system of N particles, most of the simulation time is spent in calculating the force between different particles. A way to reduce this calculation time is to cut off the potential after a specific radius r_c and only include particles within this radius. For the Lennard-Jones potential typical values are $r_c = 2.5\ \sigma - 3.5\ \sigma$. A way to improve this technique was introduced by Loup Verlet [13], the so-called neighbor list or Verlet list. The idea is to use a second radius $r_v > r_c$ and calculate the forces for all particles based on this list. The list is updated every n-th step. In order to estimate r_v one can use

$$r_v = r_c + \Delta r \tag{54}$$

with

$$\Delta r \leq n \tilde{v} \Delta t \tag{55}$$

where Δt is the time step, $\tilde{v}$ the root-mean-square. See figure 5 for a graphic representation of this method. This means that within the following $n - 1$ steps no particles, other the ones in the list, will move into the cut-off radius. A typical number for n is 10 - 20. The computation time can be reduced by a factor of 10.
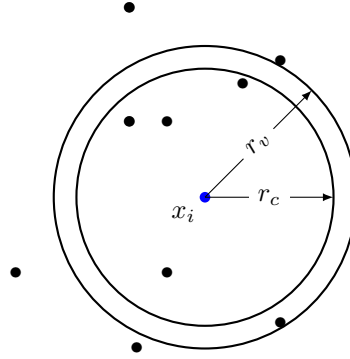
Figure 5: Neighbor list

3.3 OpenMP

OpenMP [14] is an application programming interface (API) which supports parallel programming in C/C++ and Fortran. The easiest way to use OpenMP is by splitting up indpendent loops in a given amount of threads, so that each thread can calculate parallel (see figure 6 for a schematic representation of this method).
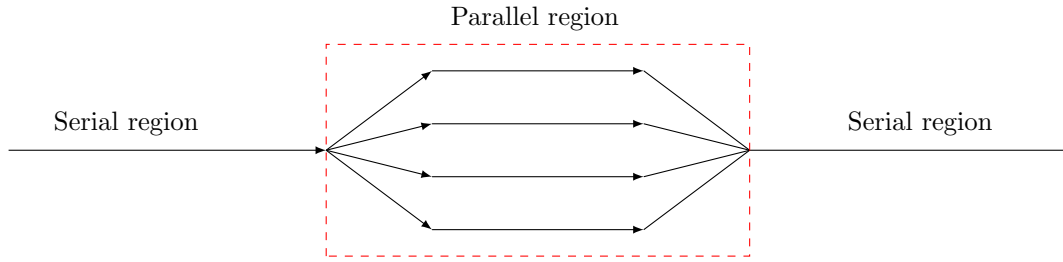

Figure 6: OpenMP

To use OpenMP one needs to import the header "omp.h". The following function call specifies the amount of threads which want to be used in the following code segment

omp_set_num_threads(NUM_THREADS);

It is not guaranteed that the specified amount of threads are created. A parallel region is introduced by

#pragma omp parallel for

and closed by

#pragma omp barrier

The last statement ensures that the computation continues only if every thread has finished its tasks. It is possible to keep variables private inside threads. The following example splits the loop over i in the specified amount of threads, but keeps j and k private for each thread to prevent unexpected changes of these variables. Another way of keeping variables private is to declare them inside the loops.

```
omp_set_num_threads(NUM_THREADS);
#pragma omp parallel for private(j,k)
  for (i=0; i<amount; i++){
```

```cpp
    int l;
    for (j=0; j<amount; j++){
      for (k=0; k<3; k++){
        // Computations
      }
    }
  }
#pragma omp barrier
```

Listing 1: Example: OpenMP

If two or more threads want to change a shared value, it causes problems if this happens at the same time. There are two ways to prevent this. First by using the *reduction* directive. This statement creates a copy of the argument for each thread and uses the unity element for the specified operation to combine them after the parallel region.

```
#pragma omp parallel for reduction(+:variable)
```

This method doesn't work for class members. In this case the *atomic* directive can be used.

```
#pragma omp atomic
```

The statement after this directive is protected and can only be written if no other thread is currently writing. If a compiler doesn't support OpenMP, the directives will cause errors. To avoid this, the following statement is used

```cpp
#ifdef _OPENMP
  #include <omp.h>
#endif
```

Listing 2: To avoid error messages if OpenMP is not installed

3.4 Integrators

3.4.1 Langevin Dynamics

In order to solve the differential equation one needs to integrate the equation of motion. A second-order integrator for Langevin equations was developed by Eric Vanden-Eijnden and Giovanni Ciccotti [3] and is a generalization of the BBK integrator [15]. The algorithm reduces to the velocity-Verlet algorithm when $\gamma_i = 0$. The Integrator found is

$$
\begin{cases}
\vec{v}^{(n+1/2)} = \vec{v}^{(n)} + \frac{1}{2}\Delta t \vec{f}(\vec{x}^{(n)}) - \frac{1}{2}\Delta t \gamma \vec{v}^{(n)} + \frac{1}{2}\sqrt{\Delta t}\sigma \vec{\xi}^{(n)} \\
\qquad - \frac{1}{8}\Delta t^2 \gamma(\vec{f}(\vec{x}^{(n)}) - \gamma\vec{v}^{(n)}) - \frac{1}{4}\Delta t^{3/2}\gamma\sigma \left(\frac{1}{2}\vec{\xi}^{(n)} + \frac{1}{\sqrt{3}}\vec{\eta}^{(n)}\right) \\
\vec{x}^{(n+1)} = \vec{x}^{(n)} + \Delta t \vec{v}^{(n+1/2)} + \Delta t^{3/2}\sigma\frac{1}{\sqrt{12}}\vec{\eta}^{(n)} \\
\vec{v}^{(n+1)} = \vec{v}^{(n+1/2)} + \frac{1}{2}\Delta t \vec{f}(\vec{x}^{(n+1)}) - \frac{1}{2}\Delta t \gamma \vec{v}^{(n+1/2)} + \frac{1}{2}\sqrt{\Delta t}\sigma \vec{\xi}^{(n)} \\
\qquad - \frac{1}{8}\Delta t^2 \gamma(\vec{f}(\vec{x}^{(n+1)}) - \gamma\vec{v}^{(n+1/2)}) - \frac{1}{4}\Delta t^{3/2}\gamma\sigma \left(\frac{1}{2}\vec{\xi}^{(n)} + \frac{1}{\sqrt{3}}\vec{\eta}^{(n)}\right)
\end{cases}
\tag{56}
$$

where $(\vec{\xi}^{(n)}, \vec{\eta}^{(n)})$ are independent Gaussian variables with mean zero and covariance

$$
\langle \xi_i^{(n)} \xi_j^{(n)} \rangle = \langle \eta_i^{(n)} \eta_j^{(n)} \rangle = \delta_{i,j} \qquad \langle \xi_i^{(n)} \eta_j^{(n)} \rangle = 0
\tag{57}
$$

3.4.2 Self-Guided Langevin Dynamics

Xiangwu Wu and Bernard R. Brooks describe a leap-frog Verlet SGLD simulation algorithm [7]. This integrator was used for the SGLD and RXSGLD simulations.

3.5 Initial Configuration

The initial configuration is chosen such that it is suitable for the specific problem. For a Lennard-Jones fluid the initial positions are such that the particles are equally spaced inside the box. They produce a crystal like grid in the beginning. This avoids problems with particles too close to each other in the beginning. The initial velocities are drawn from a Boltzmann distribution based on the temperature of the system. The initial positions for a Gaussian chain follow a Gaussian distribution (equation 39), whereas the positions for a polymer with the FENE potential are equally spaced. In both cases the initial velocities are also drawn from a Boltzmann distribution. Another possibility for the initial configuration is by using .xyz-files containing the initial positions and/or velocities. For simulations with a double well potential, the initial positions are such, that all particles start in the well placed at 0.

3.6 Random numbers

Simulations containing the random force require independent, standard, normally distributed random values. A method to create such values during the simulation is the Box-Muller method [16]. Starting from two uniformly distributed values u_1 and u_2 in the interval $(0, 1)$, then

$$z_1 = \sqrt{-2 \ln u_1} \cos(2\pi u_2) \tag{58}$$

and

$$z_2 = \sqrt{-2 \ln u_1} \sin(2\pi u_2) \tag{59}$$

are independent random variables with a standard normal distribution. u_1 and u_2 are created via the built-in C++ random number generator and scaled to the needed interval.

3.7 Code

All of the mentioned methods are implemented in C++ using object oriented programming. The code for this project is released under the GNU General Public License and can be found here: https://github.com/mr-georgebaker/bachelorthesis

4 Measurements

Several measurement can be taken in a simulation and compared to analytic values. The kinetic energy is

$$E_{Kin} = \frac{m}{2} \sum_{i=1}^{N} \vec{v}_i^2 \tag{60}$$

The temperature of the system is

$$T = \frac{1}{3N} \sum_{i=1}^{N} \vec{v}_i^2 = \frac{2E_{Kin}}{3N} \tag{61}$$

The pressure is defined in terms of the virial expression [10]

$$PV = NT + \frac{1}{3} \left\langle \sum_{i=1}^{N} \vec{x}_i \cdot \vec{F}_i \right\rangle \tag{62}$$

5 Results

5.1 Lennard-Jones Fluid

An important check is to show that the energy is conserved if $\gamma = 0$ within a LD simulation. Figure 7 shows the energy conservation for a Lennard-Jones fluid with 512 particles and a reduced density of $\rho^* = 0.6$.
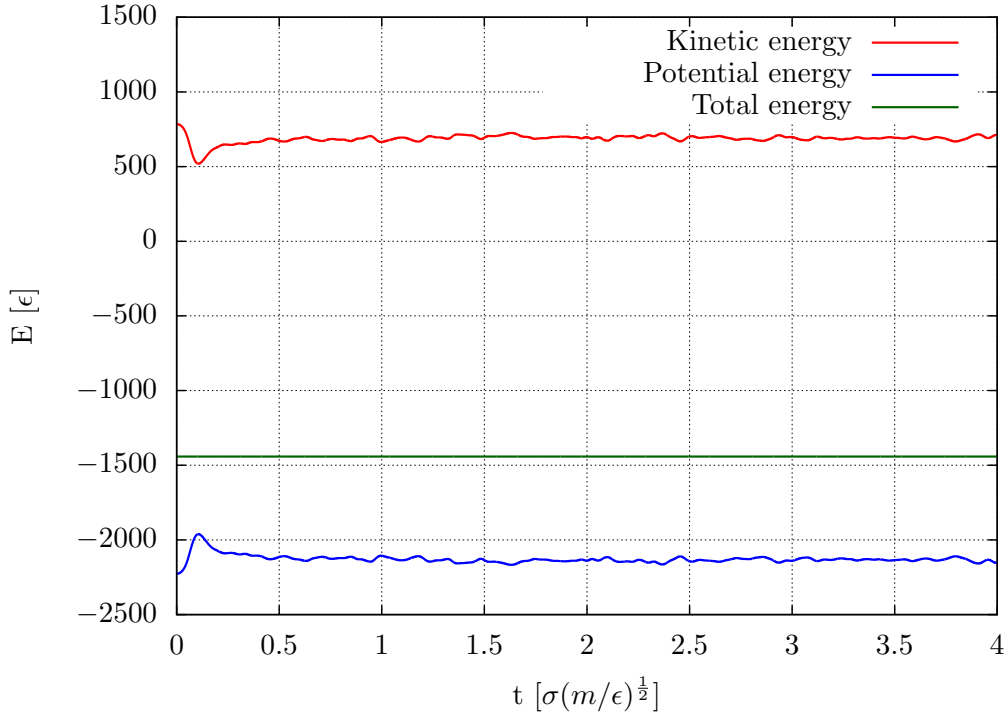

Figure 7: Energy evolution ($\gamma^* = 0$, $N = 512$, $\rho^* = 0.6$, $\Delta t^* = 0.001$, $T^* = 1$)

Another useful check is the value of the temperature if $\gamma \neq 0$. The thermostat should keep the temperature, and therefore the kinetic energy, within fluctuations, constant. Figure 8 shows the

temperature for two systems. One without a thermostat, the other one is coupled to a heat bath with the Langevin equation. One can clearly see the significant temperature drop if no thermostat is used.
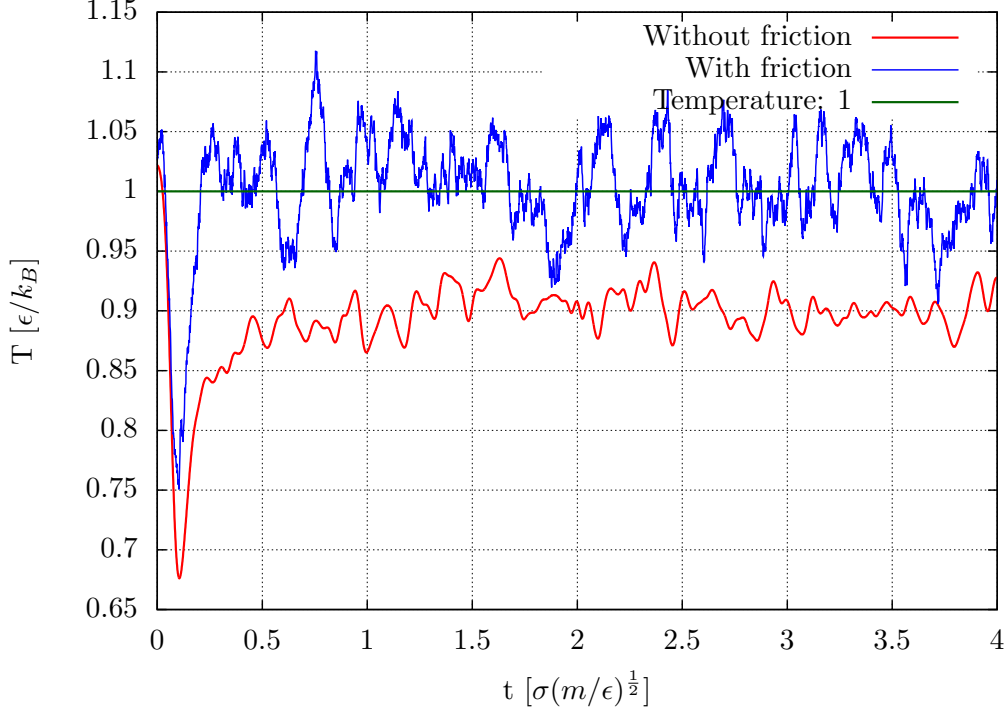

Figure 8: Temperature ($N = 512$, $\rho^* = 0.6$ $1/\sigma^3$, $\Delta t^* = 0.001$ $\sigma(m/\epsilon)^{1/2}$, $T^* = 1$ ϵ/k_B) Red: $\gamma^* = 0$, Blue: $\gamma^* = 5$ $\sigma^{-1}(m/\epsilon)^{-1/2}$, Green: Initial temperature 1 ϵ/k_B

5.2   External Double Well Potential

To verify if the simulation correctly reproduces ρ_y (equation 28) a single particle is placed in a double well potential with parameters $s^* = 0$, $w^* = 2$, $b^* = 16$ ϵ (figure 9) and $s^* = 1$ ϵ, $w^* = 2$, $b^* = 16$ ϵ (figure 10).
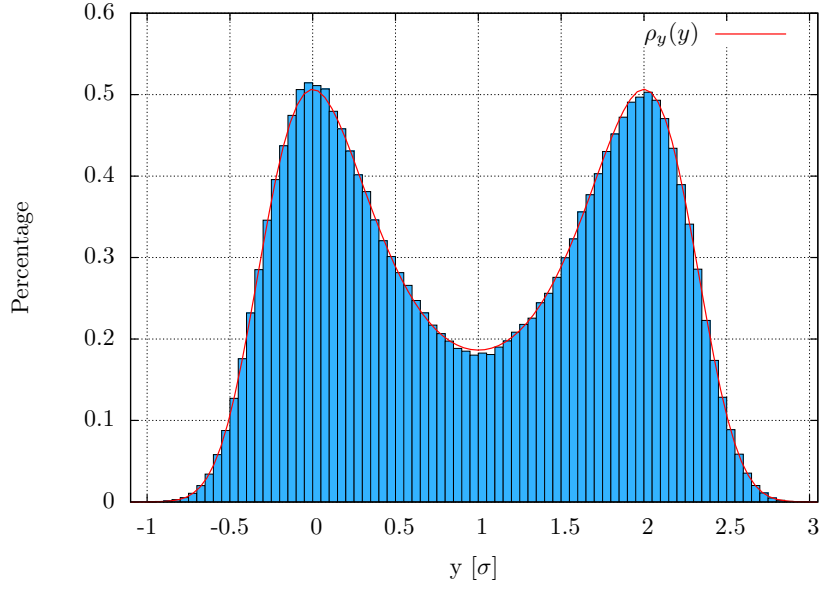
Figure 9: $\rho_y(y)$ for a single particle in double well potential ($\Delta t^* = 0.005 \ \sigma(m/\epsilon)^{1/2}$, $T^* = 1 \ \epsilon/k_B$, $\gamma^* = 1 \ \sigma^{-1}(m/\epsilon)^{-1/2}$, timesteps: 10^7)
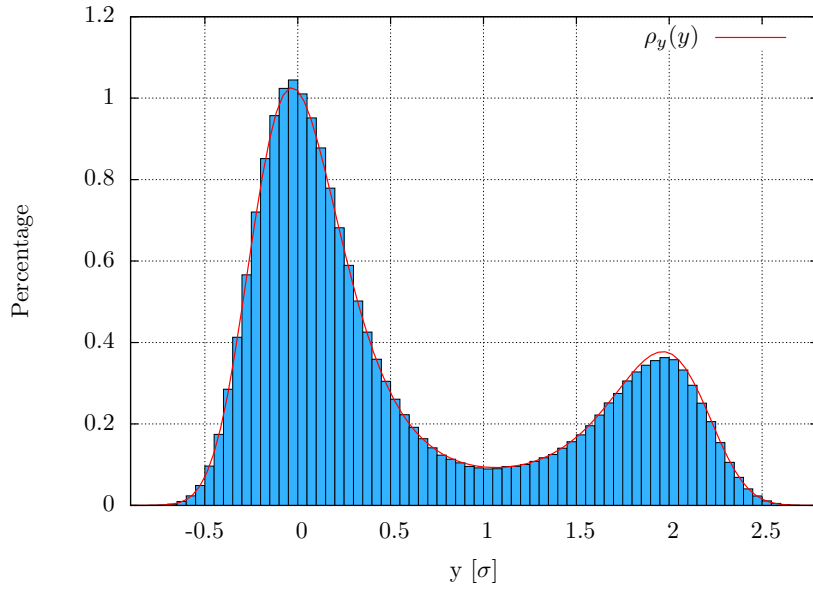

Figure 10: $\rho_y(y)$ for a single particle in double well potential ($s^* = 1 \ \epsilon$, $w^* = 2$, $b^* = 30 \ \epsilon$, $\Delta t^* = 0.005 \ \sigma(m/\epsilon)^{1/2}$, $T^* = 1 \ \epsilon/k_B$, $\gamma^* = 1 \ \sigma^{-1}(m/\epsilon)^{-1/2}$, timesteps: 10^7)

Both simulations show a very good accordance with the analytic expression.

17

5.3 Self-Guided Langevin Dynamics

To show the guiding effect, a single particle is placed inside a double well potential with parameters $s^* = 0$, $w^* = 2$ and $b^* = 160$ ϵ. The ability to overcome the energy barrier is significantly increased for a SGLD simulation. This also shows the higher conformational search ability of the SGLD method.
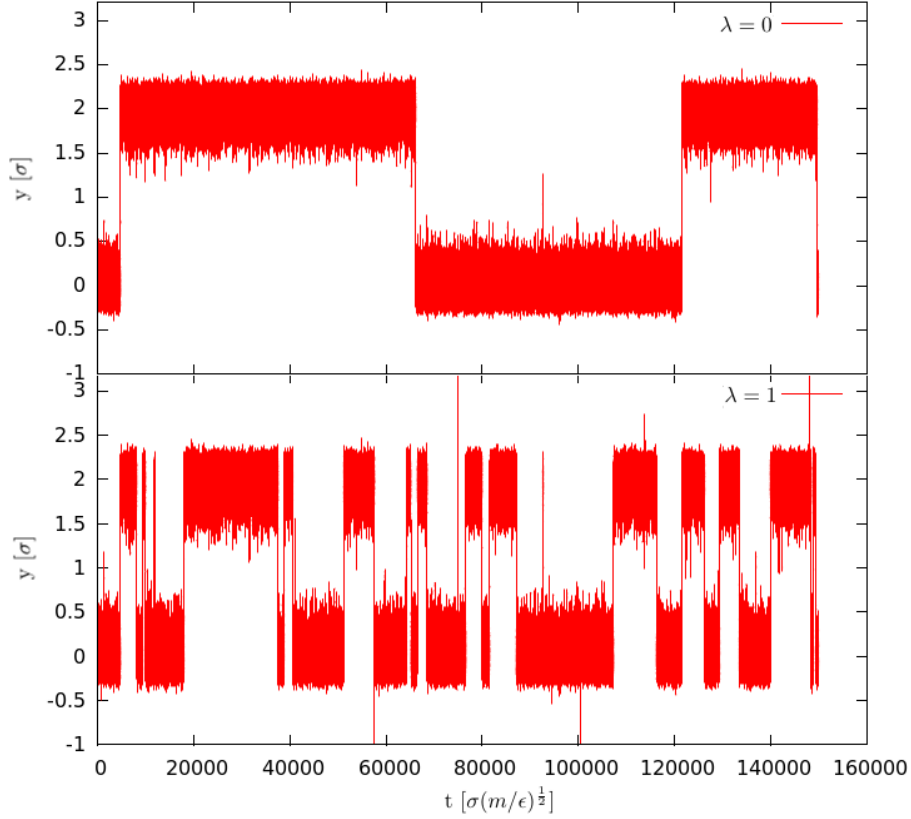

Figure 11: Comparison of a single Particle in a double well potential without SGLD ($\lambda = 0$) and with SGDL ($\lambda^* = 1$, $t_L^* = 0.05$ $\sigma(m/\epsilon)^{1/2}$), ($\gamma^* = 15$ $\sigma^{-1}(m/\epsilon)^{-1/2}$, $\Delta t^* = 0.005$ $\sigma(m/\epsilon)^{1/2}$, $T^* = 1$ ϵ/k_B, timesteps: $3 \cdot 10^7$)

5.4 Parallel Tempering

To show the increase of the conformational search ability for the RXLD method, a polymer ($N = 10$) is placed inside a double well potential with parameters $s^* = 0$, $w^* = 2$, $b^* = 200$ ϵ. Figure 12 shows a simulation without parallel tempering, figure 13 a simulation with 4 replicas and figure 14 a simulation with 8 replicas. With increasing amount of replicas, the conformational sampling is accelerated.
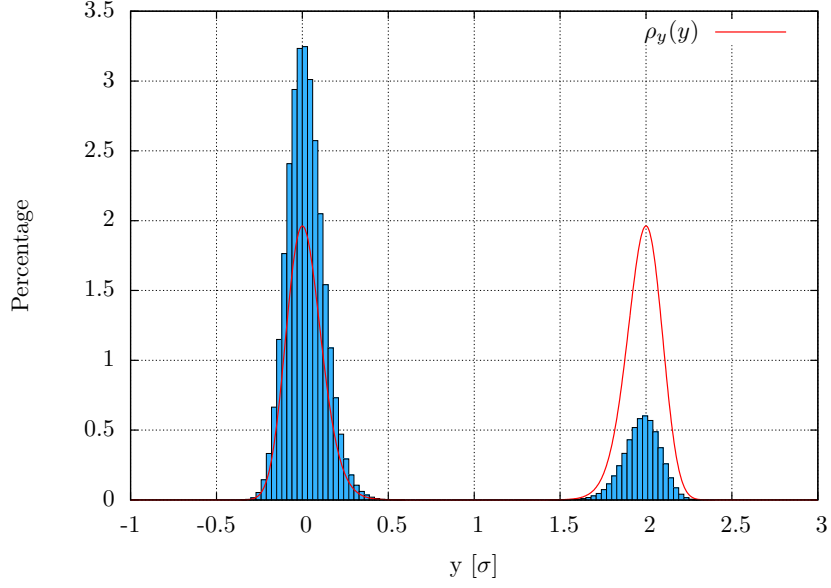
Figure 12: $\rho_y(y)$ for a polymer in a double well potential without parallel tempering ($N = 10$, $\Delta t^* = 0.005$ $\sigma(m/\epsilon)^{1/2}$, $T^* = 1$ ϵ/k_B, $\gamma^* = 1$ $\sigma^{-1}(m/\epsilon)^{-1/2}$, timesteps: $3 \cdot 10^6$)
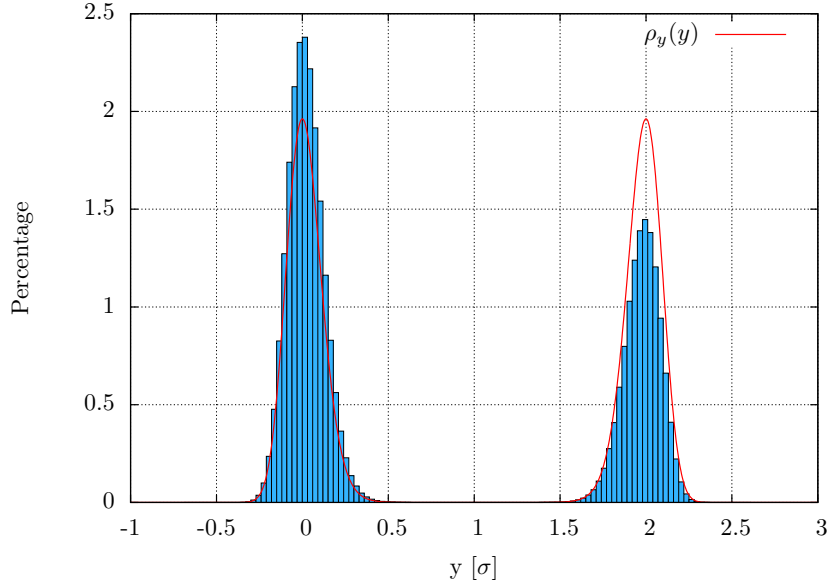

Figure 13: $\rho_y(y)$ for a polymer in a double well potential with parallel tempering (coordinate exchange) and 4 replica exchange molecular dynamics, or parallel tempering, uses replicas with higher temperatures than the target system and ex eplicas ($N = 10$, $\Delta t^* = 0.005$ $\sigma(m/\epsilon)^{1/2}$, $\gamma^* = 1$ $\sigma^{-1}(m/\epsilon)^{-1/2}$, $T_1^* = 1$ ϵ/k_B, $T_2^* = 1.1$ ϵ/k_B, $T_3^* = 1.21$ ϵ/k_B, $T_4^* = 1.33$ ϵ/k_B, timesteps: $3 \cdot 10^6$, exchanges were performed every 20000 steps $\sim$ 10 times relaxation time)
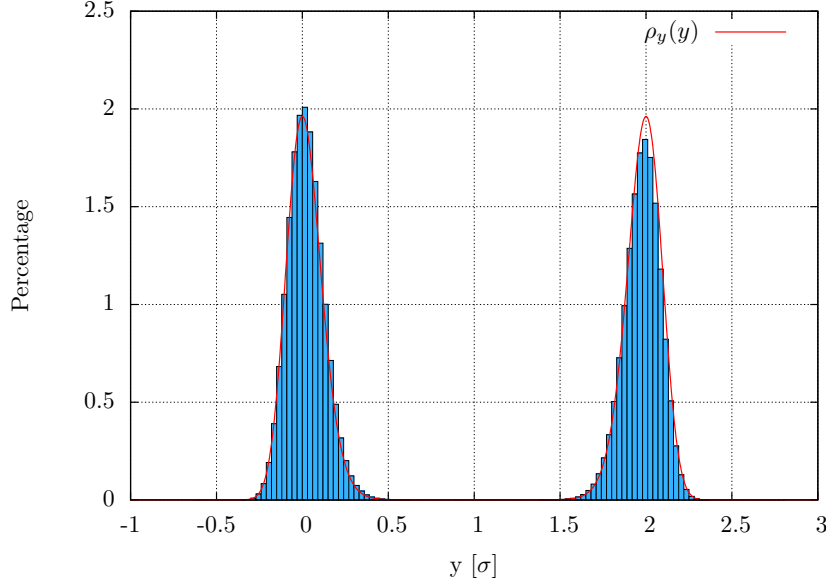
19

Figure 14: $\rho_y(y)$ for a polymer in a double well potential with parallel tempering (coordinate exchange) and 8 replicas ($N = 10$, $\Delta t^* = 0.005$ $\sigma(m/\epsilon)^{1/2}$, $\gamma^* = 1$ $\sigma^{-1}(m/\epsilon)^{-1/2}$, $T_1^* = 1$ ϵ/k_B, $T_2^* = 1.1$ ϵ/k_B, $T_3^* = 1.21$ ϵ/k_B, $T_4^* = 1.33$ ϵ/k_B, $T_5^* = 1.46$ ϵ/k_B, $T_6^* = 1.61$ ϵ/k_B, $T_7^* = 1.77$ ϵ/k_B, $T_8^* = 1.95$ ϵ/k_B, timesteps: $3 \cdot 10^6$, exchanges were performed every 20000 steps ~ 10 times relaxation time)

5.5 Replica Exchanging Self-Guided Langevin Dynamics

In some cases, a LD simulation is not capable of sampling the conformational space and gets stuck in a local minima. Such a simulation represents figure 15. A polymer ($N = 16$) is set in a double well potential with parameters $s^* = 0$, $w^* = 2$, $b^* = 120$ ϵ. The polymer in the LD simulation is not able to leave the first well within the simulation time. RXLD with 4 replicas ($T_1^* = 1$ ϵ/k_B, $T_2^* = 1.1$ ϵ/k_B, $T_3^* = 1.21$ ϵ/k_B, $T_4^* = 1.33$ ϵ/k_B) provides a way out of the well and increases the conformational search ability. A RXSGLD simulation with also 4 replicas ($\lambda_1 = 0$, $\lambda_2 = 0.3$, $\lambda_3 = 0.6$, $\lambda_4 = 1.2$) increases this ability even more. For this system, the sampling is increased significantly. The expected value for $\langle y \rangle$ is 1 as the double well potential is symmetric and it is equally likely for the polymer to be in either of the wells.
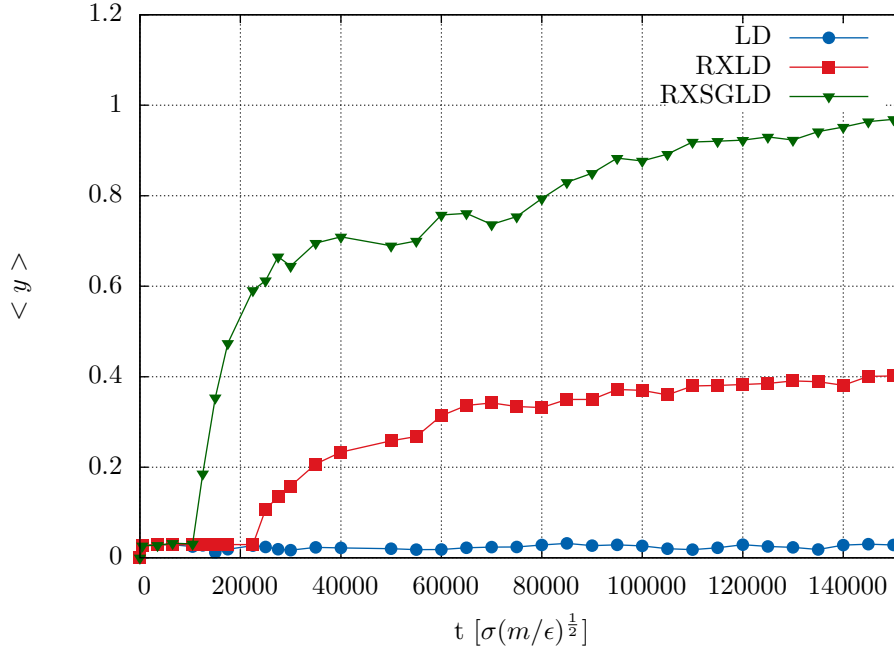
Figure 15: Comparison of conformational search ability for LD-, RXLD- and RXSGLD-simulations for a polymer ($N = 16$, FENE-potential: $R_0^* = 1.5\ \sigma$, $k^* = 30\ \epsilon/\sigma^2$, $\Delta t^* = 0.005\ \sigma(m/\epsilon)^{1/2}$, $\gamma^* = 15\ \sigma^{-1}(m/\epsilon)^{-1/2}$) in a double well potential

6   Conclusions

This thesis reports on the implementation, testing and application of the Replica exchange self guided Langevin dynamics (RXSGLD) algorithm to the problem of a polymer crossing a high free energy barrier, and shows how dramatic the effect of kinetic energy redistribution across modes can be, when sampling the configuration space. Various checks and measurements were performed to verify the correct implementation for different enhanced molecular dynamics simulation methods. A simulation without the stochastic term in the Langevin equation needs to conserve energy and by adding such a term, the values obtained from a Lennard-Jones fluid simulation have to match the analytic expressions from the ideal gas. Simulations containing polymers were tested by comparing average ensemble properties with their analytic expressions for simple model systems. Several techniques were presented, which eventually reduce the overall computational cost. By cutting off the potential and using a neighbor list, the simulation time is reduced significantly. OpenMP, an easy-to-use API for multi-threaded parallel computing has been used as the main parallelization paradigm. Without the need of major code changes, it splits up loops which then can be computed in parallel. This property is particularly useful for simulations with replicas. In a replica exchange Langevin dynamics simulation (RXLD) replicas of the target system with higher temperatures are simulated in parallel and their configurations are exchanged periodically according to an exchange probability based on the Metropolis criterion. This method makes configurations at high temperatures available for simulations performed at low temperatures. The conformational search in a system composed of a polymer in a double well potential is accelerated by increasing the amount of replicas. However, high temperatures cause perturbation on the conformational distribution. Especially for big systems, this leads to a reduced exchange probability and therefore a less effective method. Self guided Langevin dynamics (SGLD) is a method which can also increase the conformational search ability. An additional guiding force is added to the Langevin equation which heats up low frequency motions and cools down high frequency motions, without changing the overall temperature. Low frequency motions, like chain rotations of a poly-

21

mer, are important for conformational searching. For a simple system containing one particle, placed in a potential double well, the amount of energy barrier crossings between the two wells is increased significantly. By adding the guiding force, the partition function is changed and therefore all values in the simulation have to be re-weighted. RXSGLD combines instead the advantages of RXLD and SGLD. Replicas with different strengths of guiding forces are simulated parallel and also exchanged based on the Metropolis criterion. The base stage of such a simulation contains no guiding force and will therefore correctly produce the target ensemble. Initially placed at the origin, a polymer is stuck in the first well during the whole simulation when using simple LD. By using a RXLD scheme the sampling is clearly enhanced, as the polymer starts crossing the barrier, but an equiibrium sampling was not attained during the whole simulation. For the RXSGLD case the improvement is even more remarkable, allowing to reach the equilibrium distribution.

References

[1] Enrico Fermi, J Pasta, and S Ulam. Studies of nonlinear problems. *Los Alamos Report LA-1940*, 978, 1955.

[2] A Rahman. Correlations in the motion of atoms in liquid argon. *Physical Review*, 136(2A): A405, 1964.

[3] Eric Vanden-Eijnden and Giovanni Ciccotti. Second-order integrators for langevin equations with holonomic constraints. *Chemical physics letters*, 429(1):310–316, 2006.

[4] Yuji Sugita and Yuko Okamoto. Replica-exchange molecular dynamics method for protein folding. *Chemical physics letters*, 314(1):141–151, 1999.

[5] Xiongwu Wu and Bernard R Brooks. Self-guided langevin dynamics simulation method. *Chemical Physics Letters*, 381(3):512–518, 2003.

[6] Xiongwu Wu, Milan Hodoscek, and Bernard R Brooks. Replica exchanging self-guided langevin dynamics for efficient and accurate conformational sampling. *The Journal of chemical physics*, 137(4):044106, 2012.

[7] Xiongwu Wu and Bernard R Brooks. Toward canonical ensemble distribution from self-guided langevin dynamics simulation. *The Journal of chemical physics*, 134(13):134108, 2011.

[8] Dennis C Rapaport. *The art of molecular dynamics simulation*. Cambridge university press, 2004.

[9] Prince E Rouse Jr. A theory of the linear viscoelastic properties of dilute solutions of coiling polymers. *The Journal of Chemical Physics*, 21(7):1272–1280, 1953.

[10] Masao Doi and Sam F Edwards. *The theory of polymer dynamics*, volume 73. oxford university press, 1988.

[11] Iwao Teraoka. *Polymer Solutions: An Introduction to Physical Properties*. John Wiley & Sons, 2002.

[12] Gary S Grest and Kurt Kremer. Molecular dynamics simulation for polymers in the presence of a heat bath. *Physical Review A*, 33(5):3628, 1986.

[13] Loup Verlet. Computer" experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical review*, 159(1):98, 1967.

[14] OpenMP. OpenMP API for parallel programming, version 3.1.

[15] Axel Brünger, Charles L Brooks, and Martin Karplus. Stochastic boundary conditions for molecular dynamics simulations of st2 water. *Chemical physics letters*, 105(5):495–500, 1984.

[16] George EP Box, Mervin E Muller, et al. A note on the generation of random normal deviates. *The annals of mathematical statistics*, 29(2):610–611, 1958.