Contents

1 Langevin Equation

1.1 Langevin Dynamics (LD)

The Langevin Equation for the i-th particle is:

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

where $\vec{F}(\vec{x}_i(t)) = \vec{F}_i$ is the interaction force, 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:

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

and

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

This is equivalent to

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

where $\eta(t) = \dot{W}(t)$ is a white-noise ($W(t)$ being a Wiener process).

1.2 Self Guided Langevin Dynamics (SGLD)

A method for enhancing conformational searching is by using a self-guided Langevin equation.

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

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

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

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 [9]

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

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

$$\begin{aligned} \Theta_{LD} &= \sum \exp\left(\frac{E_p}{k_B T}\right) \\ &= \sum \exp\left(-\lambda_{lf}\chi_{lf}\frac{\tilde{E}_p}{k_B T0} - \lambda_{hf}\chi_{hf}\frac{E_p - \tilde{E}_p}{k_B T}\right) \\ &\quad \times \exp\left((\lambda_{lf}\chi_{lf} - 1)\frac{\tilde{E}_p}{k_B T} + (\lambda_{hf}\chi_{hf} - 1)\frac{E_p - \tilde{E}_p}{k_B T}\right) \\ &= \Theta_{SGLD} \langle w_{SGLD} \rangle_{SGLD} \end{aligned} \tag{14}$$

With the SGLD reweighting factor w_{SGLD}

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

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

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

2 Integrators

2.1 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 found by Eric Vanden-Eijnden and Giovanni Ciccotti [1] and is a generalization of the BBK integrator. The algorithm reduces to the velocity-Verlet algorithm when $\gamma_i = 0$. The Integrator found is

$$\begin{cases} \vec{v}^{(n+1/2)} = \vec{v}^{(n)} + \frac{1}{2}\Delta t \vec{f}(\vec{x}^{(n)}) - \frac{1}{2}\Delta t \gamma \vec{v}^{(n)} + \frac{1}{2}\sqrt{\Delta t}\sigma \vec{\xi}^{(n)} \\ \qquad - \frac{1}{8}\Delta t^2 \gamma (\vec{f}(\vec{x}^{(n)}) - \gamma \vec{v}^{(n)}) - \frac{1}{4}\Delta t^{3/2}\gamma\sigma \left(\frac{1}{2}\vec{\xi}^{(n)} + \frac{1}{\sqrt{3}}\vec{\eta}^{(n)}\right) \\ \vec{x}^{(n+1)} = \vec{x}^{(n)} + \Delta t \vec{v}^{(n+1/2)} + \Delta t^{3/2}\sigma \frac{1}{\sqrt{12}}\vec{\eta}^{(n)} \\ \vec{v}^{(n+1)} = \vec{v}^{(n+1/2)} + \frac{1}{2}\Delta t \vec{f}(\vec{x}^{(n+1)}) - \frac{1}{2}\Delta t \gamma \vec{v}^{(n+1/2)} + \frac{1}{2}\sqrt{\Delta t}\sigma \vec{\xi}^{(n)} \\ \qquad - \frac{1}{8}\Delta t^2 \gamma (\vec{f}(\vec{x}^{(n+1)}) - \gamma \vec{v}^{(n+1/2)}) - \frac{1}{4}\Delta t^{3/2}\gamma\sigma \left(\frac{1}{2}\vec{\xi}^{(n)} + \frac{1}{\sqrt{3}}\vec{\eta}^{(n)}\right) \end{cases} \tag{17}$$

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

$$\langle \xi_i^{(n)} \xi_j^{(n)} \rangle = \langle \eta_i^{(n)} \eta_j^{(n)} \rangle = \delta_{i,j} \qquad \langle \xi_i^{(n)} \eta_j^{(n)} \rangle = 0 \tag{18}$$

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) [5]. Following changes are made

$$\tilde{r} \rightarrow r\sigma \tag{19}$$

$$\tilde{E} \rightarrow E\epsilon \tag{20}$$

$$\tilde{m} \rightarrow m m_a \tag{21}$$

where σ and ϵ are parameter of the Lennard-Jones potential (ϵ governs the strength of the interaction and σ defines a length scale, both just numbers) 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 for example the formula for the kinetic energy shouldn't change under unit system transformations. This leads to

$$\tilde{t} \rightarrow t\sqrt{\frac{m_a \sigma^2}{\epsilon}} \tag{22}$$

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:

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 timesunit 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 $L = 1.218 N^{1/3}$.

4 Potentials

4.1 Lennard-Jones Potential

The Lennard-Jones Potential approximates the interaction between neutral atoms or molecules. A common expression is

$$U = 4\epsilon \left(\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right) \tag{23}$$

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, and the potential becomes

$$U_{LJ} = \begin{cases} 4\epsilon\left(\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right) & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c \end{cases} \tag{24}$$

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

$$U = \begin{cases} U_{LJ} - U_{LJ}(r_c) & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c \end{cases} \tag{25}$$

The force is

$$-\vec{\nabla} U = \vec{F}_{ij} = \begin{cases} \frac{48\epsilon}{\sigma^2}\left(\left(\frac{\sigma}{r_{ij}}\right)^{14} - \frac{1}{2}\left(\frac{\sigma}{r_{ij}}\right)^{8}\right)\vec{r}_{ij} & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c \end{cases} \tag{26}$$

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

$$-\vec{\nabla} U = \vec{F}_{ij} = \begin{cases} 48\left(r_{ij}^{-14} - \frac{1}{2}r_{ij}^{-8}\right)\vec{r}_{ij} & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c \end{cases} \tag{27}$$

A plot of the Lennard-Jones-Potential can be seen in figure 1.

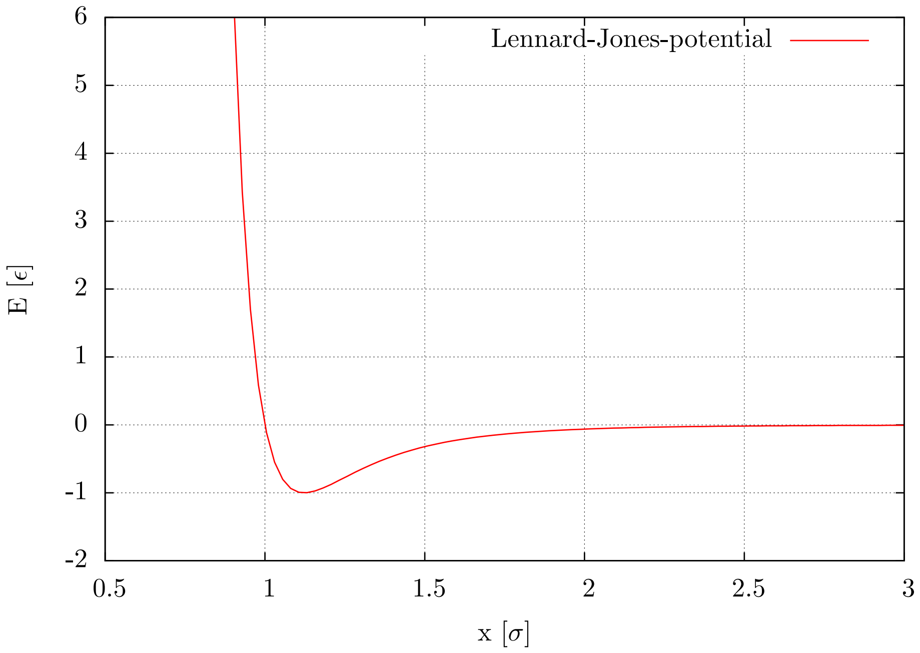

Figure 1: Lennard-Jones-potential in reduced units

4.2 External Double Well Potential

For the simulations with a double well potential, the following potential is used

$$U(x,y,z) = \frac{b}{w^4} y^2 (y-w)^2 + \frac{s}{w} y \tag{28}$$

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.

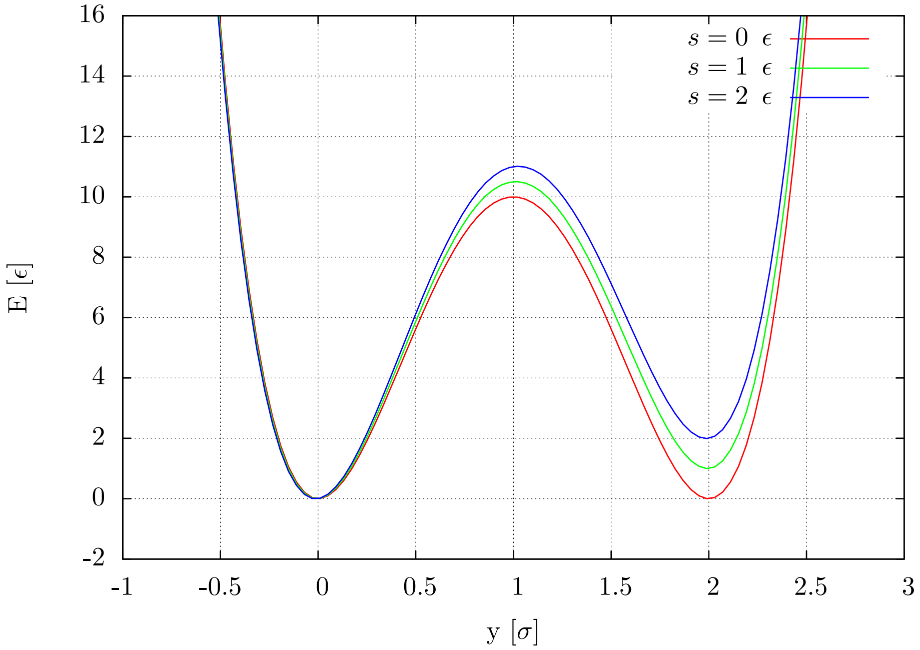

Figure 2: y-component of double well potential for three different parameters s and $b = 160\ \epsilon$ [10]

The partition function can be separated for each degree of freedom [10],

$$Q_y = \int_{-\infty}^{\infty} dy \exp\left(-\frac{\frac{b}{w^4} y^2 (w-y)^2 + \frac{s}{w} y}{k_B T}\right) \tag{29}$$

And therefore

$$\rho_y(y) = \frac{1}{Q_y} \exp\left(-\frac{\frac{b}{w^4} y^2 (w-y)^2 + \frac{s}{w} y}{k_B T}\right) \tag{30}$$

5 Replica Exchange

5.1 Parallel Tempering

Parallel tempering, also known as temperature replica exchange molecular dynamics (RXLD), 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 the probability of exchanging a replica is the same as exchanging it back. This can be satisfied by the usual Metropolis criterion [11]:

$$p(X_i \rightarrow X_j) = \min\left(1, \exp((E_p^{(i)} - E_p^{(j)})(\beta^{(i)} - \beta^{(j)}))\right) \tag{31}$$

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

$$p_i = p_i \sqrt{\frac{T_j}{T_i}} \qquad \text{and} \qquad p_j = p_j \sqrt{\frac{T_i}{T_j}} \tag{32}$$

Another method is by exchanging coordinates instead of exchanging temperatures

$$\vec{X}_i(x_i, p_i, T_i) \xleftrightarrow{exchange} \vec{X}_j(x_j, p_j, T_j) \Rightarrow \vec{X}_i(x_j, p_i, T_i) \text{ and } \vec{X}_j(x_i, p_j, T_j) \tag{33}$$

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.

5.2 Replica Exchanging Self-Guided Langevin Dynamics

Based on the partition function, equation 9, the distribution probability of a state $\vec{X}_m^{(i)}$ with temperature T_m is given by [9]

$$\begin{aligned} \rho_{SGLD}(X_m^{(i)}) &= \frac{1}{\Theta_{SGLD}^{(m)}} \exp\left(-\frac{\lambda_{lf}^{(m)} \chi_{lf}^{(m)} \tilde{E}_p^{(i)}}{k_B T_m} - \frac{\lambda_{hf}^{(m)} \chi_{hf}^{(m)} \left(E_p^{(i)} - \tilde{E}_p^{(i)} \right)}{k_B T_m} \right) \\ &= \frac{1}{\Theta_{SGLD}^{(m)}} \exp\left(-\tilde{\mu}_m \tilde{E}_p^{(i)} - \mu_m E_p^{(i)} \right) \end{aligned} \tag{34}$$

with parameters defined as

$$\tilde{\mu}_m = \frac{\lambda_{lf}^{(m)} \chi_{lf}^{(m)} \tilde{E}_p^{(i)}}{k_B T_m} \tag{35}$$

and

$$\mu_m = \frac{\lambda_{hf}^{(m)} \chi_{hf}^{(m)} \left(E_p^{(i)} - \tilde{E}_p^{(i)} \right)}{k_B T_m} \tag{36}$$

The exchange probability can then be expressed in the following way [9]

$$\begin{aligned} \pi_{RX}\left(\left\{ \vec{X}_m^{(i)}, \vec{X}_n^{(j)} \right\} \rightarrow \left\{ \vec{X}_m^{(j)}, \vec{X}_n^{(i)} \right\} \right) \approx \exp\Big(-(\tilde{\mu}_m - \tilde{\mu}_n) \left(\tilde{E}_p\left(\vec{X}_n^{(j)} \right) - \tilde{E}_p\left(\vec{X}_m^{(i)} \right) \right) \\ - (\mu_m - \mu_n)\Big) \left(E_p(\vec{X}_n^{(j)} \right) - E_p\left(\vec{X}_m^{(i)} \right) \Big) \end{aligned} \tag{37}$$

The exchange criterion stays the same (see equation 31) and if $\lambda = 0$ the exchange probability is the same as in equation 31. SGLD causes less pertubation on conformational distribution and therefore the exchange probability is much higher than in a RXLD-simulation. From equations 31, 37 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.

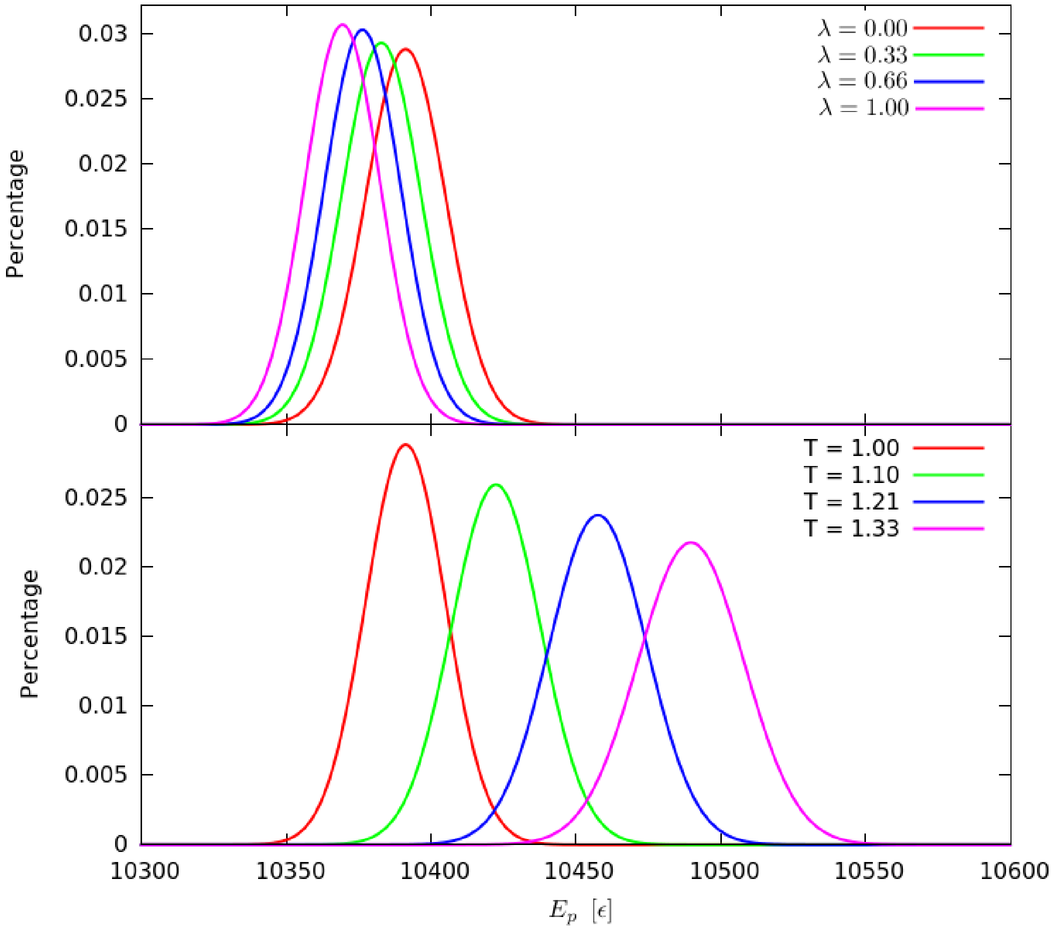

Figure 3: Potential energy distributions for a polymer ($N = 500$, FENE-potential: $R_0 = 1.5\ \sigma$, $k = 30\epsilon/\sigma^2$, $\Delta t = 0.005$, $\gamma = 1$) for 4 different temperatures and 4 different guiding factors

6 Polymer Physics

A simple way to represent a polymer is by using springs between the atoms. This model is called the Rouse model [6]. A schematic representation of this model can be seen in figure 6.

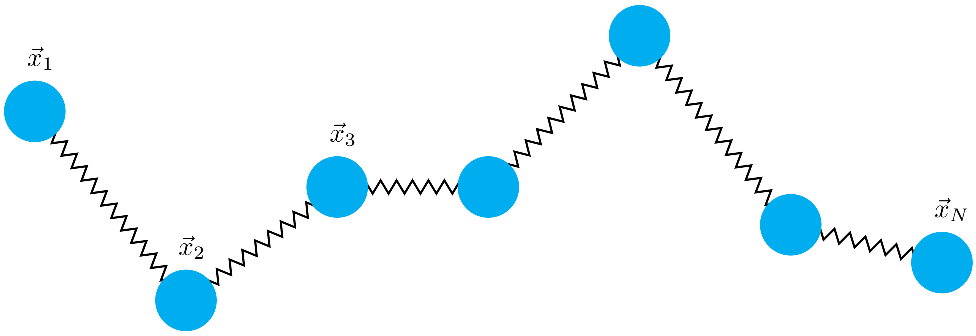

Figure 4: Rouse model

The potential energy of one bond is

$$U_{ij} = \frac{1}{2}k(\vec{x}_i - \vec{x}_j)^2 \tag{38}$$

with spring constant $k = \frac{3k_BT}{b^2}$ 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

$$\vec{F}_i = k(\vec{x}_{i+1} - 2\vec{x}_i + \vec{x}_{i-1}) \tag{39}$$

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 also called the Gaussian chain, because every bond length satisfies the distribution [3]

$$\psi(\vec{x}) = \left(\frac{3}{2\pi b^2}\right)^{\frac{3}{2}} \exp\left(-\frac{3\vec{x}^2}{2b^2}\right) \tag{40}$$

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

$$\phi(\vec{x}_n - \vec{x}_m, n-m) = \left(\frac{3}{2\pi b^2|n-m|}\right)^{\frac{3}{2}} \exp\left(-\frac{3(\vec{x}_n - \vec{x}_m)^2}{2|n-m|b^2}\right) \tag{41}$$

A plot of the distribution due to the simulation and the analytic one can be found in figure 5.

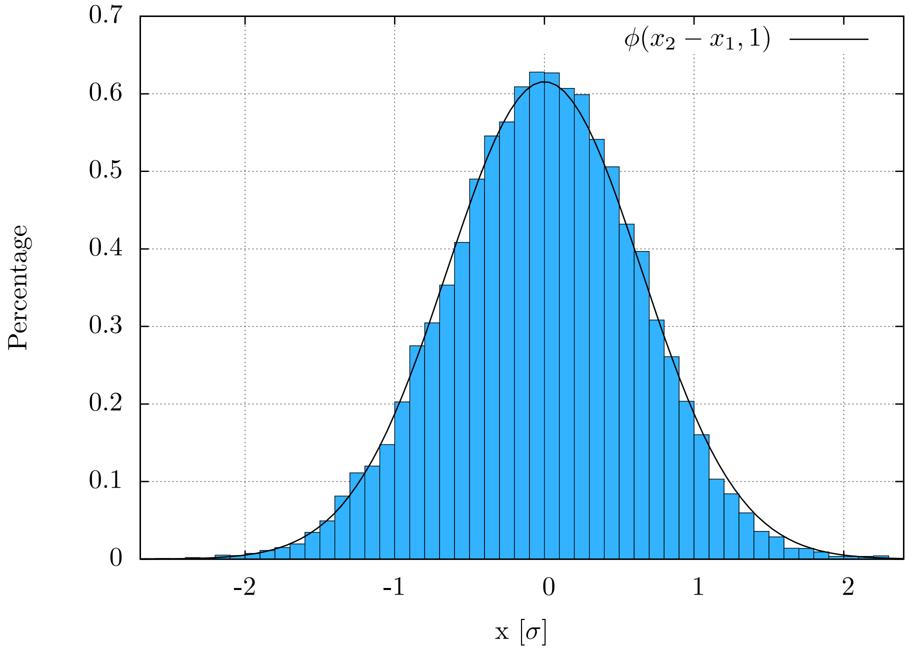

Figure 5: End-to-end distribution between the first and second bead ($\gamma = 1$, $N = 10$, $\Delta t = 0.001$, $T = 1$)

6.1 Normal Coordinates

Measurements in a molecular dynamics simulation can only be performed when the system is equilibrated. A way to give an estimate of the relaxation time needed can be done by introducing normal coordinates [7]. The equation of motion is

$$m_i\ddot{\vec{x}}_i = k(\vec{x}_{i+1} - 2\vec{x}_i + \vec{x}_{i-1}) - \gamma m_i\dot{\vec{x}}_i + \vec{R}_i \tag{42}$$

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

$$\gamma\dot{\vec{x}}_i = k(\vec{x}_{i+1} - 2\vec{x}_i + \vec{x}_{i-1}) + \vec{R}_i \tag{43}$$

The i-th normal coordinate is defined by

$$\vec{q}_i = \frac{1}{N}\sum_{n=1}^{N} \cos\left(\frac{in\pi}{N}\right)\vec{x}_n \tag{44}$$

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

$$\vec{q}_0 = \frac{1}{N}\sum_{n=1}^{N}\vec{x}_n \tag{45}$$

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

$$\xi_i\dot{\vec{q}}_i = -k_i\vec{q}_i + \vec{g}_i \tag{46}$$

with

$$\xi_i = \begin{cases} N\gamma & \text{if } i = 0 \\ 2N\gamma & \text{if } i \neq 0 \end{cases} \tag{47}$$

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

$$k_i = k\frac{\xi_i}{\gamma}\left(\frac{i\pi}{N}\right)^2 = \frac{6\pi^2 k_B T}{Nb^2}i^2 \tag{48}$$

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

$$\vec{g}_i = \frac{\xi_i}{\gamma}\frac{1}{N}\sum_{n=1}^{N}\cos\left(\frac{in\pi}{N}\right)\vec{R}_n \tag{49}$$

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

$$\tau_i = \frac{\xi_i}{k_i} \tag{50}$$

Expressed with the relaxation time the equation of motion is

$$\dot{\vec{q}}_i = -\frac{1}{\tau_i}\vec{q}_i + \frac{1}{\xi_i}\vec{g}_i \tag{51}$$

For the Rouse-Model:

$$\tau_i = \frac{\gamma N^2 b^2}{3\pi^2 k_B T}\frac{1}{i^2} \tag{52}$$

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

6.2 FENE-Potential

Another potential for connecting the monomers is the FENE-Potential [2]

$$U_{ij} = \begin{cases} -\frac{1}{2}kR_0^2 \ln\left(1 - \left(\frac{r_{ij}}{R_0}\right)^2\right) & r_{ij} \leq R_0 \\ 0 & r_{ij} > R_0 \end{cases} \tag{53}$$

This potential has an advantage over the harmonic one. The springs of the harmonic potential are infinitely expandable, whereas the maximum bond length with the FENE-Potential is R_0. Common values for the constants are $k = 30\ \epsilon/\sigma^2$ and $R_0 = 1.5\ \sigma$.

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

$$m_i\ddot{\vec{x}}_i = k(\vec{x}_{i+1} - 2\vec{x}_i + \vec{x}_{i-1}) + 48\left(r_{ij}^{-14} - \frac{1}{2}r_{ij}^{-8}\right)\vec{r}_{ij} - \gamma_i m_i \dot{\vec{x}}_i + \vec{R}_i \tag{54}$$

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

$$\langle R_{ee}^2 \rangle \propto (N-1)^{2\nu} \tag{55}$$

where ν is about 3/5 [3].

7 Implementation details

7.1 General algorithm

In the case where the force depends on all other particles (e.g. Lennard-Jones-Potential) one is forced to split the algorithm in two different parts. One for the calculation of the half-step for the velocity and the new position. And the other part for the calculation of the new velocity. A pseudo-code for the implementation could be like Algorithm 1 and a expiration chart of this algorithm may look like Figure 6.

```
Data: Initial positions and velocities of $m$ particles $(x_1^1, x_2^1, ..., x_m^1)$, $(v_1^1, v_2^1, ..., v_m^1)$
Result: Final positions $(x_1^n, x_2^n, ..., x_m^n)$ and velocities $(v_1^n, v_2^n, ..., v_m^n)$
initialization;
for $k \leftarrow 1$ to $n$ do
    for $i \leftarrow 1$ to $m$ do
        calculate force for all $m$ particles;
        update positions of all particles;
        calculate velocity half step;
        calculate force for new positions;
        update velocity;
    end
end
```

Algorithm 1: Main algorithm

7.2 Periodic Boundary Conditions

When simulating liquids the problem is the huge amount of particles. To reduce this problem one can introduce periodic boundary conditions, where the simulation takes place in a container of some kind and considering an infinite, space-filling array of identical copies of the simulation

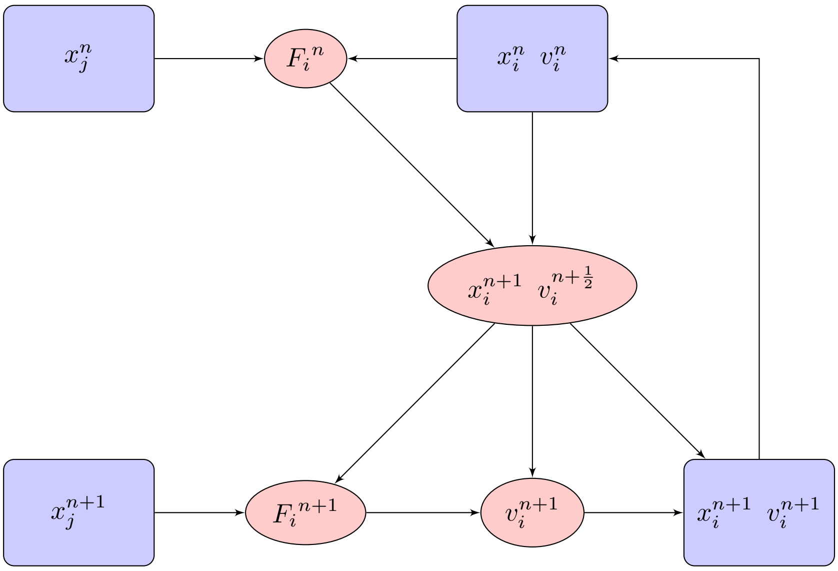

Figure 6: Expiration chart for the implementation. Red: calculation, Blue: storage

region. This means, the particles near the boundaries effect the particles on the other side of the simulation space. This leads to a wraparound effect. If a particle moves outside the region, it has to be set on the other side. The following pseudo-code takes care of this behaviour

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

Algorithm 2: Particle wraparound

7.3 Force calculation

For a system of N particles, most of the simulation time is spend in calculating the force between different particles. A way to reduce this calculation time is to cut off after a specific radius r_c and only include particle 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 [8], the so-called neighbor list or Verlet list. The idea is to use a second radius $r_v > r_c$ and calculate the force for all particles within this radius, but only at every n-th step. In order to estimate r_v one can use

$$r_v = r_c + \Delta r \tag{56}$$

with

$$\Delta r \leq n\tilde{v}\Delta t \tag{57}$$

where Δt is the time step, $\tilde{v}$ the root-mean-square. See 7.3 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 is reduced by a factor of 10.

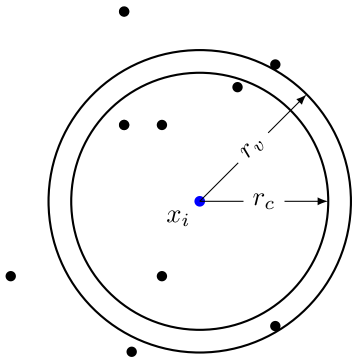

Figure 7: Neighbor list

For periodic boundary conditions the distance between to particles depends on the simulation region, as two particles near the boundaries can effect each other. A pseudo-code for the calculation of the force F_i for the i-th particle is listed below (see algorithm 3).

```
Data: Position of all $N$ particles at timestep $m$ $(x_1^m, x_2^m, ..., x_N^m)$, boolean periodic
Result: Force $F_i^m$ for i-th particle
initialization;
if $m \mod n == 0$ then
    for $j \leftarrow 1$ to $N$ do
        calculate distance $d$ from particle $i$ to other particles;
        if $d \leq r_v$ then
            add particle position to neighbor list;
        end
    end
end
calculate force on i-th particle based on neighbor list;
```

Algorithm 3: Neighbor list with boundary conditions

7.4 OpenMP

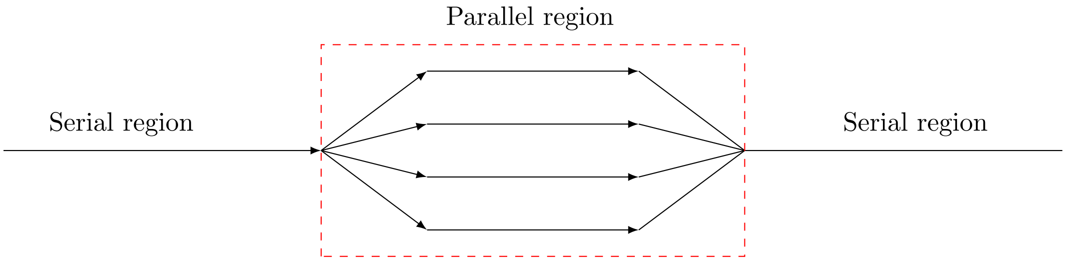

Figure 8: OpenMP

OpenMP [4] is 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. 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 it's task. It is possible to keep variables private inside of threads. The following example splits the loop over i in a 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++){
    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 cause errors on this systems. To avoid this, the following statement is used

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

Listing 2: To avoid error messages where OpenMp is not installed

8 Measurements

The kinetic and potential energies are

$$E_K = \frac{m}{2} \sum_{i=1}^{N} \vec{v}_i^2 \tag{58}$$

$$E_U = 4 \sum_{i \leq i < j \leq N} (r_{ij}^{-12} - r_{ij}^{-6}) \tag{59}$$

The temperature of the system is

$$T = \frac{1}{3N} \sum_{i=1}^{N} \vec{v}_i^2 = \frac{2E_K}{3N} \tag{60}$$

The pressure is defined in terms of the virial expression

$$PV = NT + \frac{1}{3} \Big\langle \sum_{i=1}^{N} \vec{x}_i \cdot \vec{F}_i \Big\rangle \tag{61}$$

9 Results

9.1 Lennard-Jones-Fluid

If the friction constant γ is set to zero, the integrator reduces to the velocity verlet integrator and therefore energy should be conserved. To qualify this, the shift with respect to the initial value is calculated. This is done in the following way

$$\Delta E = \frac{E(t) - E(0)}{E(t)} \tag{62}$$

The energy evolution over time is ploted in figure 9.

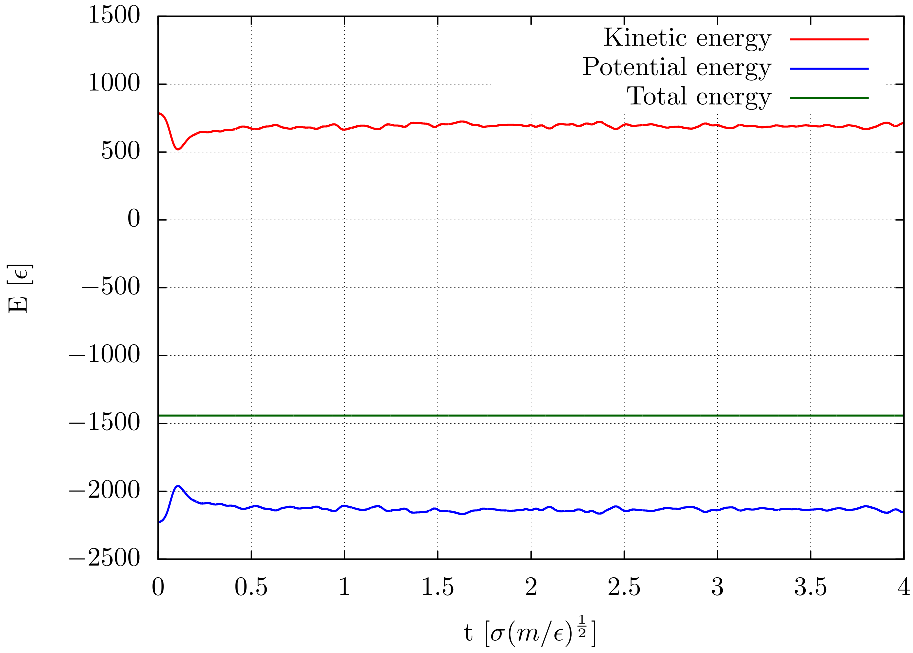

Figure 9: Energy evolution ($\gamma = 0$, $N = 512$, $\rho = 0.6$, $\Delta t = 0.001$, $T = 1$)

The energyshift can be seen in figure 10.

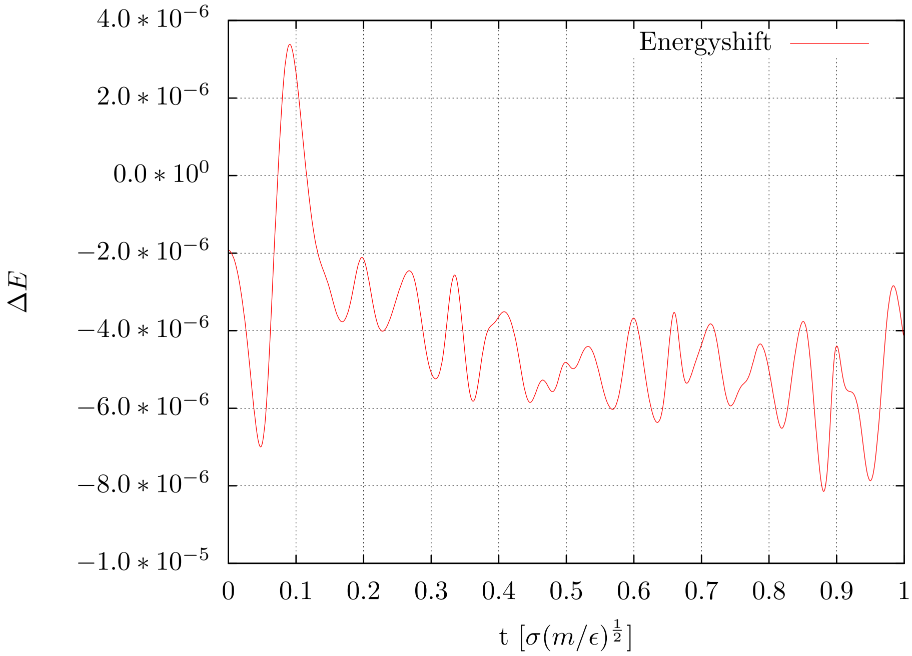

Figure 10: Energyshift ($\gamma = 0$, $N = 512$, $\rho = 0.6$, $\Delta t = 0.001$, $T = 1$)

The Langevin-Integrator works as a thermostat, i.e. the system is coupled to a heat bath. Like the kinetic energy and the pressure, the temperature fluctuates. The difference between a system with and without a heat bath can be seen in figure 11.

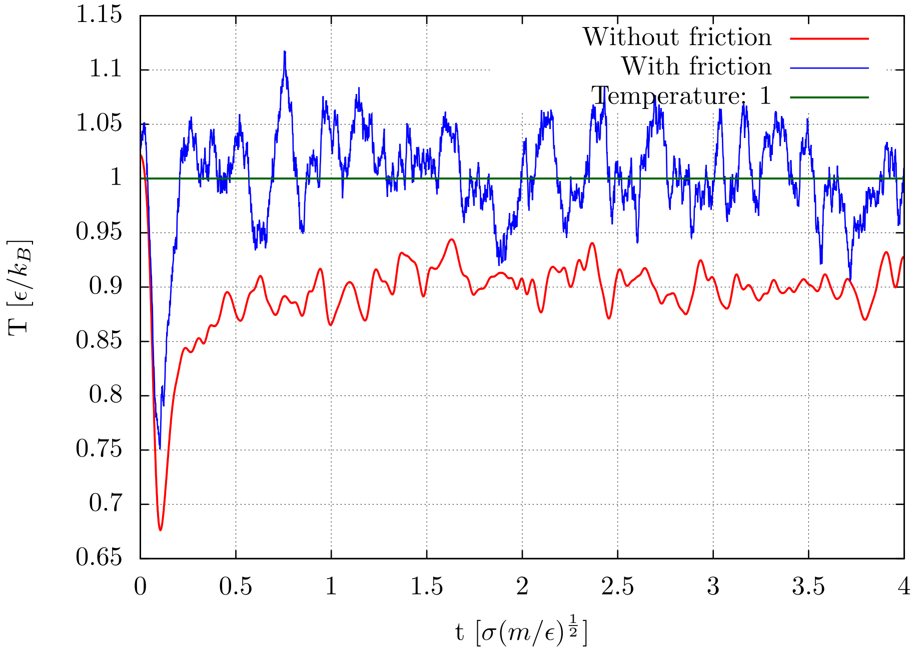

Figure 11: Temperature ($N = 512$, $\rho = 0.6$, $\Delta t = 0.001$, $T = 1$) Red: $\gamma = 0$, Blue: $\gamma = 5$, Green: Initial temperature 1

The pressure can be compared to the value for the pressure of the ideal gas.

$$PV = Nk_BT \tag{63}$$

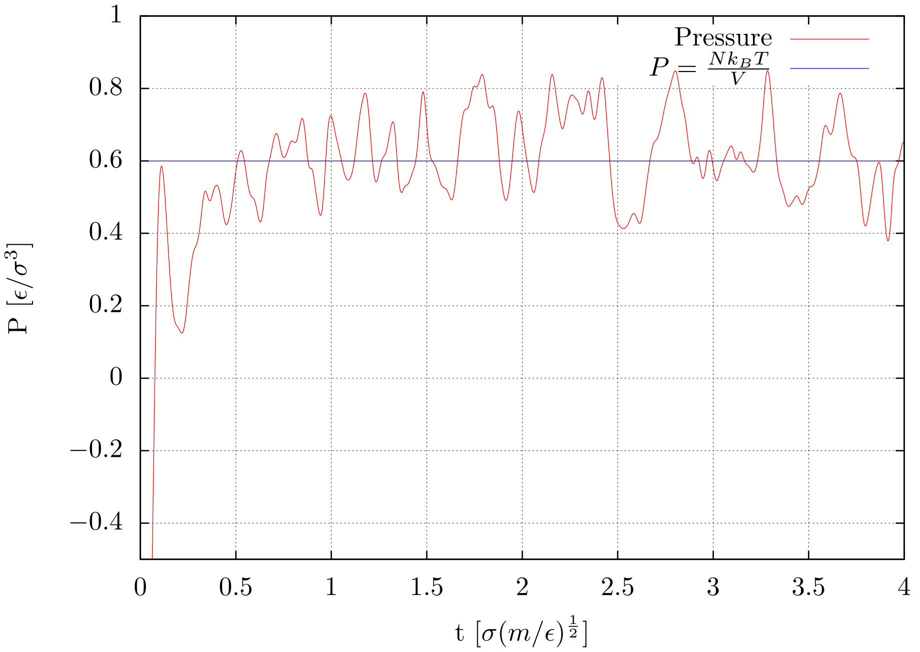

Figure 12: Pressure ($\gamma = 0$, $N = 512$, $\rho = 0.6$, $\Delta t = 0.001$, $T = 1$)

9.2 External Double Well Potential

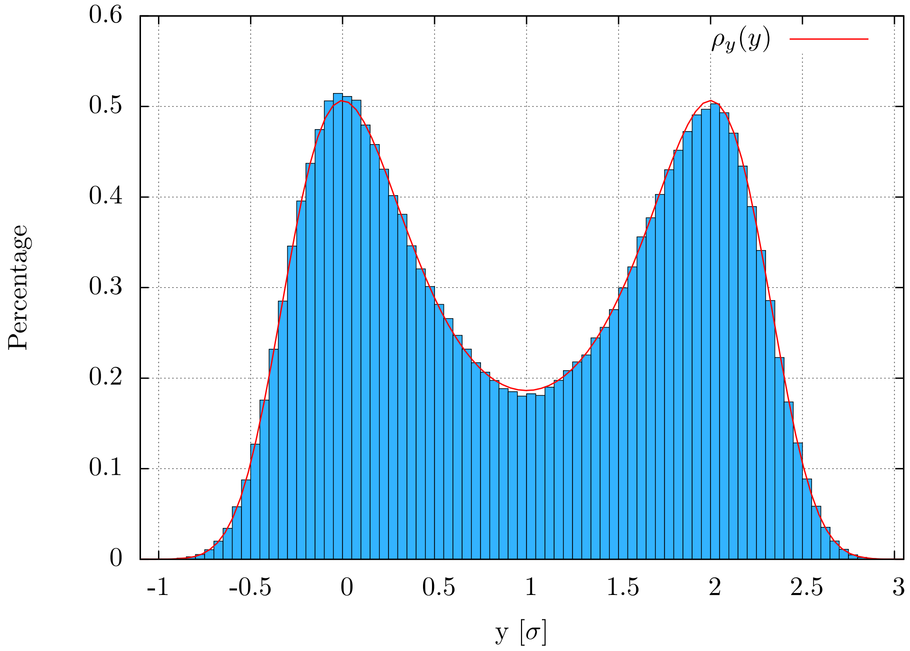

Figure 13: $\rho_y(y)$ for a single particle in double well potential ($s = 0$, $w = 2$, $b = 16$, $dt = 0.005$, $T = 1$, $\gamma = 1$, timesteps: 10^7)

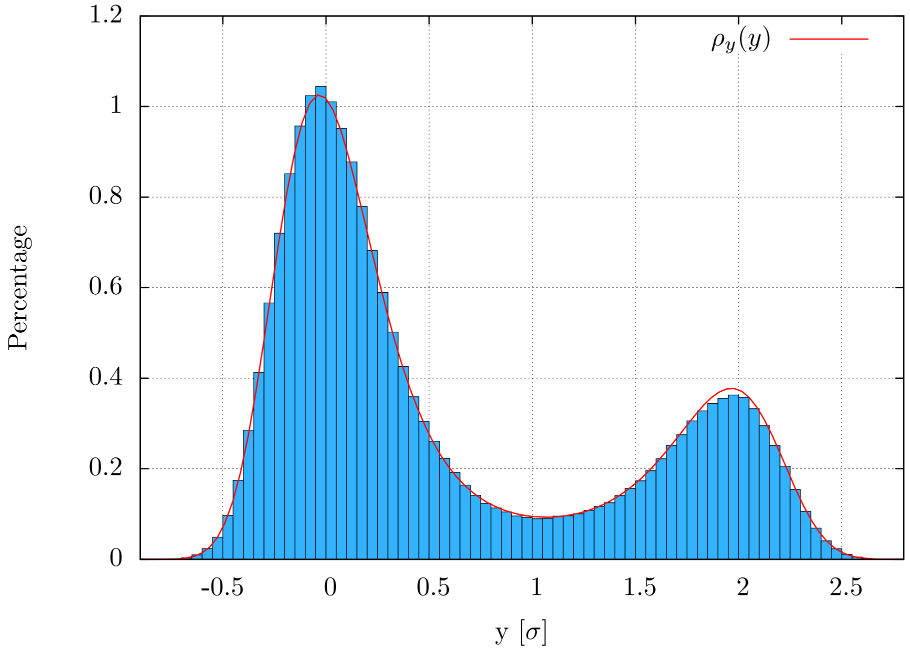

Figure 14: $\rho_y(y)$ for a single particle in double well potential ($s = 1$, $w = 2$, $b = 30$, $dt = 0.005$, $T = 1$, $\gamma = 1$, timesteps: 10^7)

9.3 Parallel Tempering

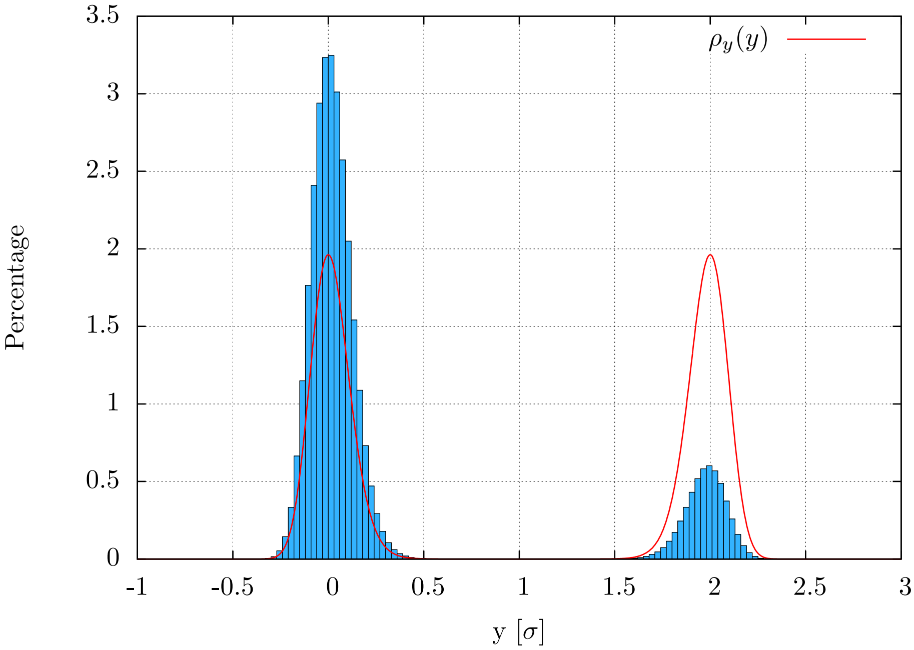

Figure 15: $\rho_y(y)$ for a polymer in a double well potential without parallel tempering ($N = 10$, $s = 0$, $w = 2$, $b = 200$, $dt = 0.005$, $T = 1$, $\gamma = 1$, timesteps: $3 \cdot 10^6$)

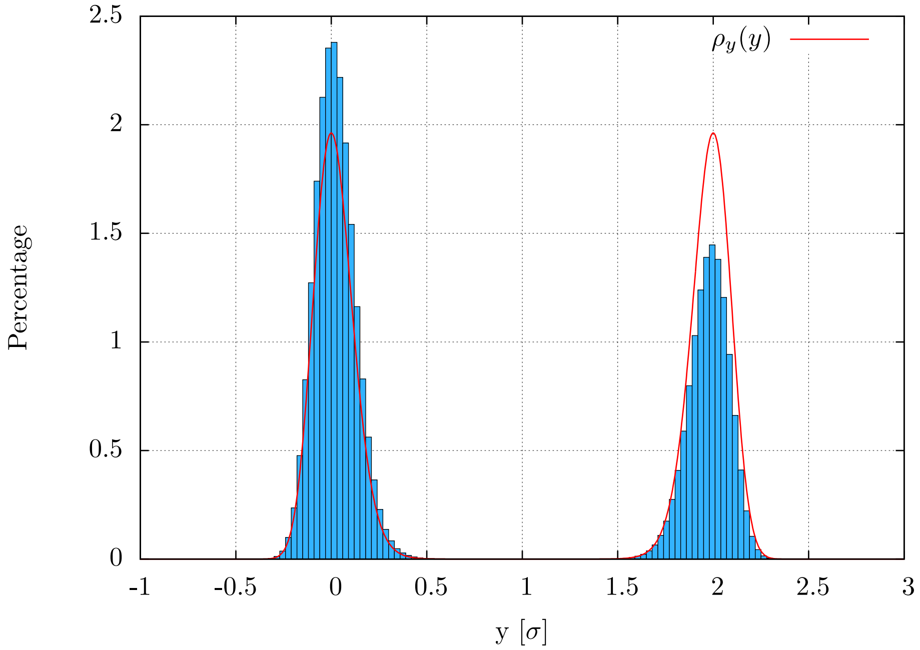

Figure 16: $\rho_y(y)$ for a polymer in a double well potential with parallel tempering (coordinate exchange) and 4 replicas ($N = 10$, $s = 0$, $w = 2$, $b = 200$, $dt = 0.005$, $T_1 = 1$, $T_2 = 1.1$, $T_3 = 1.21$, $T_4 = 1.33$, $\gamma = 1$, timesteps: $3 \cdot 10^6$, exchanges were performed every 20000 steps $\sim$ 10 times relaxation time)

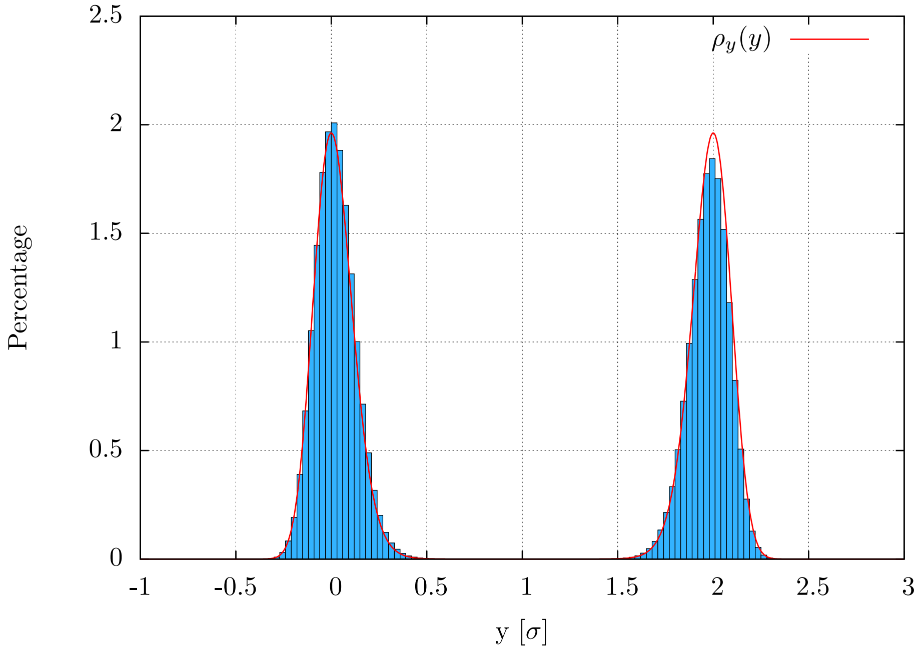

Figure 17: $\rho_y(y)$ for a polymer in a double well potential with parallel tempering (coordinate exchange) and 8 replicas ($N = 10$, $s = 0$, $w = 2$, $b = 200$, $dt = 0.005$, $T_1 = 1$, $T_2 = 1.1$, $T_3 = 1.21$, $T_4 = 1.33$, $T_5 = 1.46$, $T_6 = 1.61$, $T_7 = 1.77$, $T_8 = 1.95$, $\gamma = 1$, timesteps: $3 \cdot 10^6$, exchanges were performed every 20000 steps $\sim$ 10 times relaxation time)

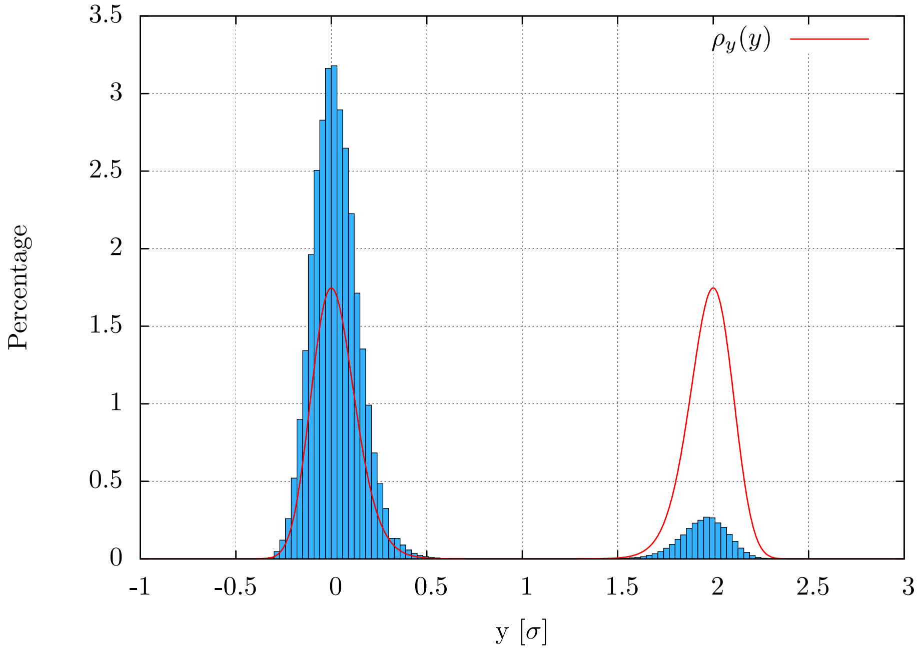

Figure 18: $\rho_y(y)$ for a polymer in a double well potential without parallel tempering ($N = 30$, $s = 0$, $w = 2$, $b = 160$, $dt = 0.005$, $T = 1$, $\gamma = 1$, timesteps: $2 \cdot 10^6$)

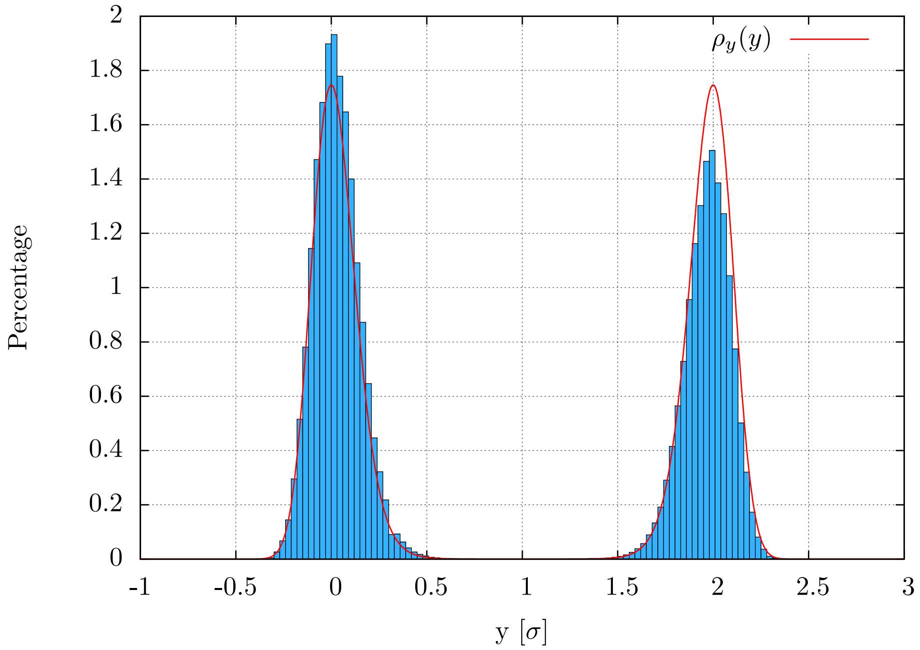

Figure 19: $\rho_y(y)$ for a polymer in a double well potential with parallel tempering (coordinate exchange) and 12 replicas ($N = 30$, $s = 0$, $w = 2$, $b = 200$, $dt = 0.005$, temperature from $1 - 2.1$, each 7 % bigger than the one before, $\gamma = 1$, timesteps: $2 \cdot 10^6$, exchanges were performed every 200000 steps $\sim$ 10 times relaxation time)

9.4 Self Guided Langevin Dynamics

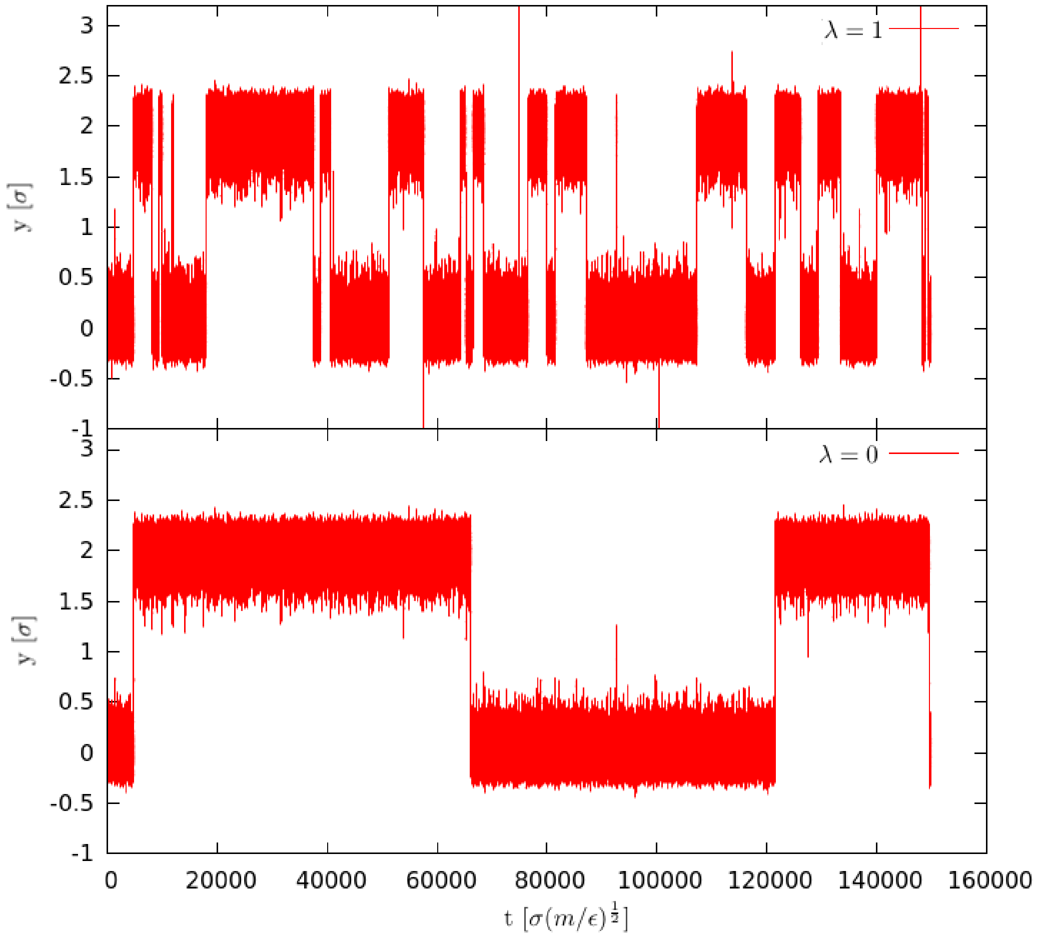

Figure 20: Comparison of a single Particle in a double well potential ($s = 0$, $w = 2$, $b = 160$) without SGLD ($\lambda = 0$) and with SGDL ($\lambda = 1$, $t_L = 0.05$), ($\gamma = 15$, $\Delta t = 0.005$, timesteps: $3 \cdot 10^7$)

9.5 RXLD vs. RXSGLD

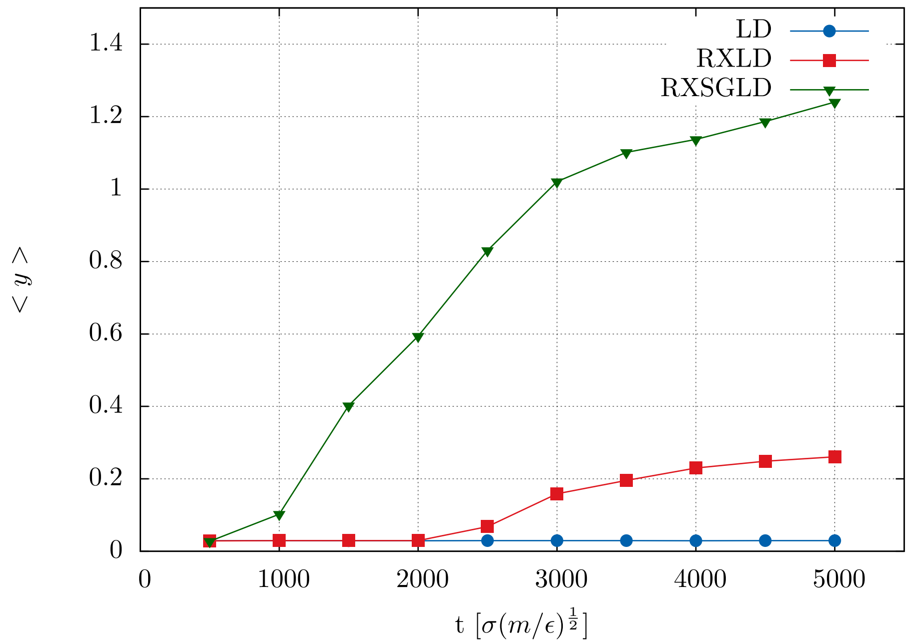

Figure 21: Comparison of conformational search ability for LD-, RXLD- and RXSGLD-simulations for a polymer ($N = 20$, FENE-potential: $R_0 = 1.5\ \ \sigma$, $k = 30\epsilon/\sigma^2$, $\Delta t = 0.005$, $\gamma = 15$) in a double well potential ($s = 0$, $w = 2$, $b = 120$)

References

[1] Giovanni Ciccotti Eric Vanden-Eijnden. Second-order integrators for langevin equations with holonomic constraints. *Chemical Physics Letters*, 429, 2006.

[2] Kurt Kremer Gary S. Grest. Molecular dynamics simulation for polymers in the presence of a heat bath. *Physical Review A*, 1986.

[3] S. F. Ewards M. Doi. *The Theory of Polymer Dynamics.* Clarendon Press, 1988.

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

[5] D. C. Rapaport. *The Art of Molecular Dynamics Simulation.* Cambridge, 2004.

[6] Prince E. Rouse. A theory of the linear viscoelastic properties of dilute solutions of coiling polymers. *The Journal of Chemical Physics*, 1953.

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

[8] Loup Verlet. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical Review*, 159, 1967.

[9] Bernard R. Brooks Xiongwu Wu. Toward canonical ensemble distribution from self-guided langevin dynamics simulation. *The Journal of Chemical Physics*, 134, 2011.

[10] Bernard R. Brooks Xiongwu Wu, Milan Hodoscek. Replica exchanging self-guided langevin dynamics for efficient and accurate conformational sampling. *The Journal of Chemical Physics*, 2012.

[11] Yuko Okamoto Yuji Sugita. Replica-exchange molecular dynamics method for protein folding. *Chemical Physics Letters*, 314, 1999.