Temperature Measurements in Optical Tweezer Experiments

Mathias Höld, BSc.

2016

1 Introduction

2 Motivation

3 The experiment

The starting point of this thesis is an experiment conducted by Gieseler et al [1]. It is an optical tweezer experiment, where the motion of a glass nanoparticle in a laser trap was used to investigate the fluctuation theorem[2].

3.1 Experimental setup

In the experiment, a silica nano particle with a radius of about 75 nm and mass of about 3×10^{-18}kg is trapped in a laser beam within a vacuum chamber. The trapping of the silica nano particle (which will be referred to as *glass particle*) is achieved by a gradient force of the laser beam acting on the particle. The experimental setup is depicted in fig. 1.
The particle fluctuates within the trap in all three spatial directions. These fluctuations can be approximated such that they are decoupled, which means that they can be described by a 1-dimensional Langevin equation:

$$\ddot{x} + \Gamma_0 \dot{x} + \Omega_0^2 x = \frac{1}{m} \left(F_{\text{fluct}} + F_{\text{ext}} \right) \tag{1}$$

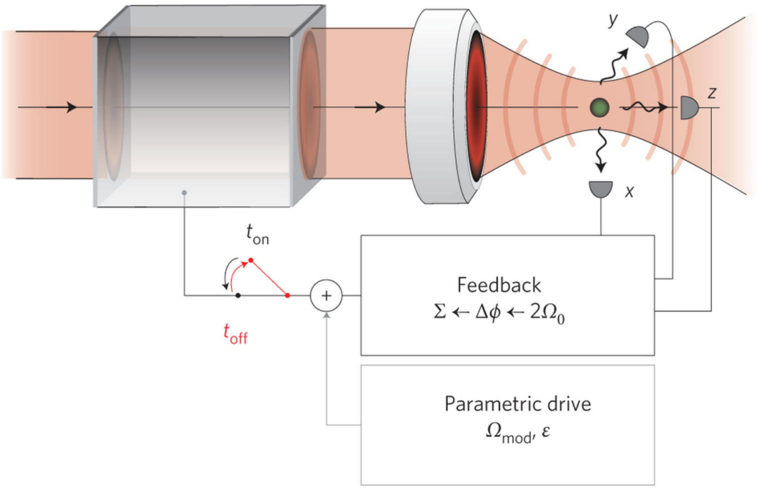

y
z
x
t_{on}
t_{off}
+
Feedback
$\Sigma \leftarrow \Delta\phi \leftarrow 2\Omega_0$
Parametric drive
$\Omega_{\text{mod}}, \varepsilon$

Figure 1: Experimental setup of the optical tweezer experiment. A silica nano particle is trapped in a laser beam via gradient force in a vacuum. The feedback is used to cool down the particle and create a non-equilibrium steady state. In the first part of the experiment, the feedback is turned off and the motion of the particle towards an equilibrated state is observed. In the second part of the experiment, the steady state of the particle is modified by a parametric drive. Both the parametric drive and the feedback are turned off and – as in the first part – the motion of the particle towards an equilibrated state is observerd.

On the left hand side we have the friction coefficient Γ_0 and the angular frequency that describes the fluctuation along the chosen axis. On the right hand side, there are two forces. The first one is F_{fluct}, which describes a stochastic force caused by interactions with the gas in the vacuum chamber. This force is given by

$$F_{\text{fluct}} = \sqrt{2m\Gamma_0 k_B T_0}\;\xi\left(t\right) \tag{2}$$

where T_0 is the temperature of the heat bath (i.e. the surrounding gas in the vacuum chamber), k_B is the Boltzmann constant and $\xi(t)$ is white noise, which obeys the equations $\langle\xi(t)\rangle = 0$ and $\langle\xi(t)\xi(t')\rangle = \delta(t-t')$, which means that it is a random force. The term Γ_0 appears in the formulat due to the fluctuation-dissipation theorem, which links the damping rate to the stochastic force.

The external force F_{ext} is part of the experimental setup, where the frequency of the fluctuation along an axis, Ω_0, is measured and used to suppress the motion along said axis. This causes a decrease in the particle fluctuations and thus acts as a cooling mechanism for the particle in the trap. This process creates a non-equilibrium steady state $\rho_{ss}(u,\alpha)$, which is not known analytically. This state is the starting point of this thesis.

4 Simulation

The problem at hand can be studied on an atomic level with the use of computer simulation. There is a variety of methods for computer simulations that are widely used, one of which being Molecular Dynamics (MD) simulations. The following section will give a brief overview of the concepts of this method, which is followed by the application to the simulation of the experiment.

4.1 Molecular Dynamics

Molecular Dynamics[3] simulations is a technique for simulating, as the name suggests, the dynamics of a classical many-body system. In this case, classical means, that the trajectories of the individual particles are calculated using classical mechanics rather then quantum mechanics. For relatively big atoms/molecules this is a very good approximation, whereas for systems consisting of hydrogen or helium the effects of quantum mechanics cannot be neglected and other methods (such as ab-initio simulation) has to be used.
The dynamics of the system are obtained by solving Newton's equations of motion for every particle.

4.2 The Glass Particle

The glass particle from the experiment will be modeled as a system of particles interacting via a Lennard-Jones pair potential,

$$U(r) = 4\varepsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right] \tag{3}$$

where ε is the depth of the potential well (and thus its unit is energy) and σ is the distance at which the potential is zero. The form of the potential and the relation to the parameters is depicted in Fig. 2. Since ε and σ are crucial parameters for the simulation and do (usually) not change over time, it is practical to use them to define the dimensions of the system. This means that the unit of distance is σ, the unit of energy is ε and the unit of mass is the mass of the simulated particle. The so called *reduced units* can be constructed from these three parameters and put into relation to the original units. Here are some examples:

- distance: $r^* = r/\sigma$
- potential energy: $U^* = U/\varepsilon$
- temperature: $T^* = k_B T/\varepsilon$

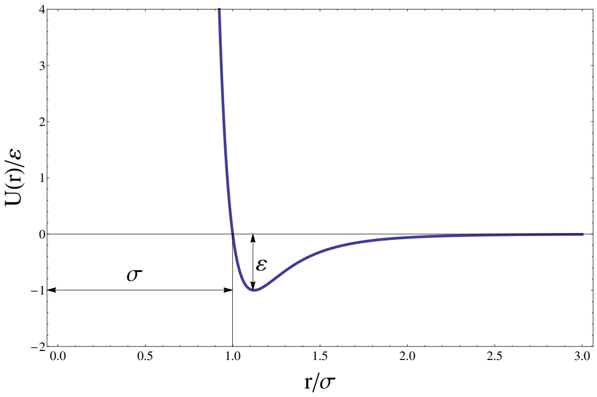


Figure 2: The Lennard-Jones 12-6 potential from (3). The x-axis is the particle distance divided by σ and the y-axis is the potential divided by the depth of the potential well.

- time: $t^* = t\sqrt{\varepsilon/(m\sigma^2)}$
- pressure: $P^* = P\sigma^3/\varepsilon$
- density: $\rho^* = \rho\sigma^3$

One very popular choice for the simulated atoms is Argon because it is an inert gas and the atoms behave approximately like hard spheres which attract each other with weak van der Waals forces, which justifies the use of the Lennard-Jones potential. Argon has a mass of m $= 6.69 \times 10^{-26}$ kg, $\sigma = 3.4 \times 10^{-10}$m and $\varepsilon = 1.65 \times 10^{-21}$J.

With the above introduced reduced units, the Lennard-Jones potential can be written as

$$U(r^*) = 4\left[r^{*-12} - r^{*-6}\right]. \tag{4}$$

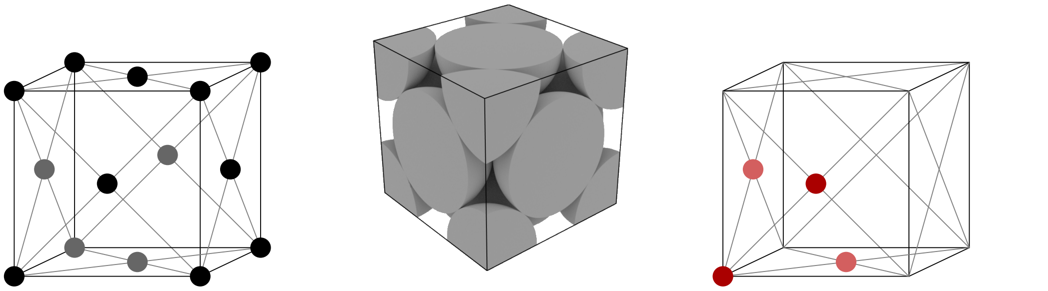
Since the reduced units will be used throughout the rest of this thesis, i will drop the asterisk henceforth.

From the Lennard-Jones potential the corresponding force can be calculated by taking the derivative with respect to the direction of interest:

$$\begin{aligned} F_x &= -\frac{\partial}{\partial x}U(r) \\ &= -\frac{\partial}{\partial x}4\left[r^{-12} - r^{-6}\right] \\ &= -4\left[(-12)r^{-13} - (-6)r^{-7}\right]\frac{\partial r}{\partial x} \\ &= 48\left[r^{-13} - 0.5\, r^{-7}\right]\frac{x}{r} \\ &= 48\left[r^{-14} - 0.5\, r^{-8}\right]x \end{aligned} \tag{5}$$

The force in the y and z direction can be calculated analogously.
The initial configuration of the particles is a face centered cubic (FCC) lattice. A schematic of the FCC lattice is depicted in fig. 3a. With the choice of FCC as initial configuration, there are optimal numbers for the numbers of the particles in the system. Since once FCC cell (as depicted) shares its atoms with its next neighbours, the number of atom per unit cell is 4, as depicted in fig. **??**. The whole system of atoms is then created by repeating this cell structure. One convenient way is to arrange the unit cells in a cubic system, so if there are M FCC unit cells on one edge, the whole system consists of M^3 cells. Since there are 4 particles per cell, there are ideal or so called *magic numbers* for atoms for which this setup works perfectly: $N = 4M^3 = 4, 32, 108, 256, 500, 864, \ldots$.

There are several ways to achieve this initial configuration and the one used in this thesis [5] was to create a kind of unit cell consisting of four atoms, as shown in fig. 3c, which can be described by a set of points



(a) Schematic figure of a face centered cubic (FCC) lattice

(b) Schematic of the number of particles contained in one FCC unit cell. The atoms on the edges are shared by 8 cells and the ones on the faces by 2 cells[4]

(c) Setup for successively building a FCC lattice

$$
\begin{aligned}
p_1 &= \{0, 0, 0\} \\
p_2 &= \{0.5, 0.5, 0\} \\
p_3 &= \{0.5, 0, 0.5\} \\
p_4 &= \{0, 0.5, 0.5\}
\end{aligned}
$$

From the particle number N and the number of FCC unit cells per edge M the lattice constant a can be calculated

$$a = \frac{L}{M} \tag{6}$$

where L is the side length of the cube that is the whole system and it is calculated via the density of the system

$$L = \sqrt[3]{\frac{N}{\rho}} \tag{7}$$

With the lattice constant and the 4 points of the FCC cell, all the particles can be put into place.

4.3 The Velocity-Verlet Algorithm

When we look at the system from a thermodynamical standpoint, we see that it follows some kind of path in the phase space as time progresses. Every point in this space corresponds to a set of positions and momenta and the connection between two points corresponds to the evolution of the system from one state to another. As mentioned above, this evolution (the dynamics of the system) is a crucial element to Molecular Dynamics. Since the equations of motion cannot be solved analytically in general, we need to approximate the solution.
The method used here is called finite difference approach. The trajectory of the system in the phase space is cut into finite pieces of length Δt and the equations of motion are solved for every segment separately (see fig. 4).

There are several ways to solve this kind of problem, but since we are interested in implementing it into a computer program the ideal solution should have some basic properties [6]:

- It should be fast and require little memory
- It should permit the use of a large time step Δt
- It should duplicate the classical trajectory as closely as possible
- It should satisfy known conservation laws
- It should be simple and easy to program

One algorithm that has all of the above mentioned features is the one proposed by Verlet [7]. In his paper, Verlet starts by Taylor expanding the coordinate mathbftor $\mathbf{r}_i$ for one particle after one time step Δt:

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \dot{\mathbf{r}}_i \Delta t + \frac{1}{2}\ddot{\mathbf{r}}_i \Delta t^2 + \frac{1}{3!}\dddot{\mathbf{r}}_i \Delta t^3 + \mathcal{O}(\Delta t^4) \tag{8}$$

Since the first derivative of the coordinate mathbftor is the velocity, $\dot{\mathbf{r}}_i(t) = \mathbf{v}_i(t)$, and the second derivative of the coordinate mathbftor is the acceleration, $\ddot{\mathbf{r}}_i(t) =$

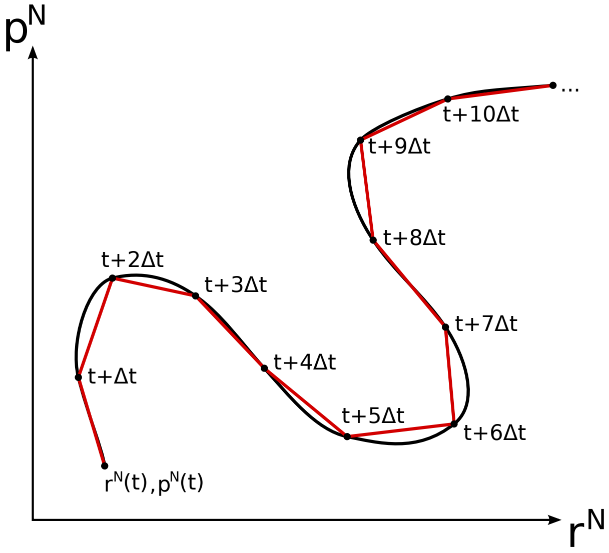


Figure 4: Simplified graphical schematic of the finite difference approach. The evolution of the system from a point $(r^N(t), p^N(t))$ in the phase space is approximated by slicing it up into pieces of length Δt. On every stop after the starting point $(t+\Delta t,\ t+2\Delta t,\ t+3\Delta t, \ldots)$ the equations of motion can be solved numerically.

$\mathbf{a}_i(t)$, using Newton's law $\mathbf{F}_i(t) = m_i\mathbf{a}_i(t)$ equation (8) can be written as:

$$\mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + \mathbf{v}_i(t)\Delta t + \frac{1}{2m}\mathbf{F}_i(t)\Delta t^2 + \frac{1}{3!}\dddot{\mathbf{r}}_i(t)\Delta t^3 + \mathcal{O}(\Delta t^4) \tag{9}$$

The same calculation can be carried out for one time step before t:

$$\mathbf{r}_i(t-\Delta t) = \mathbf{r}_i(t) - \mathbf{v}_i(t)\Delta t + \frac{1}{2m}\mathbf{F}_i(t)\Delta t^2 - \frac{1}{3!}\dddot{\mathbf{r}}_i(t)\Delta t^3 + \mathcal{O}(\Delta t^4) \tag{10}$$

The sum of those two equations yields

$$\mathbf{r}_i(t+\Delta t) + \mathbf{r}_i(t-\Delta t) = 2\mathbf{r}_i(t) + \frac{1}{2}\mathbf{F}_i(t)\Delta t^2 + \mathcal{O}(\Delta t^4) \tag{11}$$

and from this we get the final form for the new coordinates:

$$\mathbf{r}_i(t+\Delta t) = 2\mathbf{r}_i(t) + \mathbf{r}_i(t-\Delta t) + \frac{1}{2}\mathbf{F}_i(t)\Delta t^2 \tag{12}$$

This means that the new coordinates can be calculated using the current coordinates, the current forces and the coordinates from the past time step.
From equation (12) one thing becomes clear: the velocities are not necessary to calculate the new positions and they are not computed in the process, but for

the calculation of i.e. the kinetic energy the velocities are needed. Thus, the velocities have to be calculated with a combination of the Taylor expansions of the coordinate mathbftors at $t + \Delta t$ and $t - \Delta t$:

$$\mathbf{r}_i(t + \Delta t) - \mathbf{r}_i(t - \Delta t) = 2\mathbf{v}_i(t)\Delta t + \mathcal{O}(\Delta t^3) \tag{13}$$

which can be rewritten as

$$\mathbf{v}_i(t) = \frac{\mathbf{r}_i(t + \Delta t) - \mathbf{r}_i(t - \Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^2) \tag{14}$$

This approach has the advantage that it is fast, requires little memory and is reliable in the sense that there is no energy drift occuring during the simulation, which means that the energy is conserved. When we compare this to the list of desired properties this seems like a good algorithm.

This algorithm however has two significant disatvantages. The first one is the calculation of the velocities. As can be seen in equation (14), the accuracy of the calculation is only $\mathcal{O}(\Delta t^2)$ while the positions can be calculated with an error of order Δt^4. The other big disatvantage is the first step of the algorithm. Since the calculation of the new positions requires the current positions and the ones from one time step before, which technically do not exist.

The solutions to this problem is to include the stepwise calculation of the velocities [8]. For this we start again with the Taylor expansion of the coordinates mathbftor (9) and instead of Taylor expanding $\mathbf{r}_i(t - \Delta t)$, we write it as

$$\mathbf{r}_i(t) = \mathbf{r}_i(t + \Delta t) - \mathbf{v}_i(t + \Delta t)\Delta t + \frac{1}{2m}\mathbf{F}_i(t + \Delta t)\Delta t^2 \tag{15}$$

Using (9) in the abvove equation we get

$$\begin{aligned} \mathbf{r}_i(t) = \mathbf{r}_i(t) &+ \mathbf{v}_i(t)\Delta t + \frac{1}{2m}\mathbf{F}_i(t)\Delta t^2 \\ &- \mathbf{v}_i(t + \Delta t)\Delta t + \frac{1}{2m}\mathbf{F}_i(t + \Delta t)\Delta t^2 \end{aligned} \tag{16}$$

and thus

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \frac{1}{2m}\left(\mathbf{F}_i(t) + \mathbf{F}_i(t + \Delta t)\right)\Delta t \tag{17}$$

With the addition of this equation this algorithm is called the Velocity-Verlet algorithm, which is summarized in the following two equations:

$$\begin{aligned} \mathbf{r}_i(t + \Delta t) &= \mathbf{r}_i(t) + \mathbf{v}_i(t)\Delta t + \frac{1}{2m}\mathbf{F}_i(t)\Delta t^2 \\ \mathbf{v}_i(t + \Delta t) &= \mathbf{v}_i(t) + \frac{1}{2m}\left(\mathbf{F}_i(t) + \mathbf{F}_i(t + \Delta t)\right)\Delta t \end{aligned} \tag{18}$$

This algorithm is self starting, uses a small amount of memory, conserves the energy and gives a very good approximation to the originial trajectory in the phase space. To program this algorithm the following steps are needed:

1. Calculate all the forces between the particles (for the first run only)
2. Calculate the new positions with current velocity and forces
3. Calculate first half of the new velocities with current forces
4. Calculate all the forces for the new position
5. Use new forces to calculate second half of new velocities

The first point only has to be carried out for the first run, because the forces have not been calculated at this point. As the forces are calculated for the new positions in step 4, they can be used for the next time step. For the first time step the velocities have to be chosen randomly and are calculated with this algorithm from that point forward.

4.4 The Laser Beam - Energy influx

The glass sphere is trapped in the laser beam. While the motion of the center of mass is suppressed, which acts as a cooling mechanism, the individual atoms that make up the glass sphere absorb the energy from the laser which increases their velocity.

To simulate this kind behaviour, thermostat algorithms [9] such as the Nosé-Hoover [10, 11] or the Berendsen [12] algorithms.

The problem with these kind of algorithms is, that a target temperature has to be fixed which will be reached at some point. In order to simulate the influx of energy from the laser it would be better to have an algorithm that supplies the system with a certain amount of energy continuously. Fortunately, such an algorithm exists.

The algorithm is called Heat Exchange Algorithm (HEX) [13]. Its intended purpose is the use in non-equilibrium molecular dynamics (NEMD) to study transport phenomena and determine transport coefficients. The algorithm works by introducing two regions in the system, one serving as a heat source and the other as a heat sink. A specific amount of heat is then exchanged between those two reservoirs. As it turns out however, this algorithm introduces an energy shift for longer simulation times. This led Wirnsberger et al [14] to revisit the algorithm and identify the cause of this energy drift, to create a more suitable algorithm, which they called eHEX.

As in the HEX algorithm, regions are introduced to the system which act either

as heat sinks or heat sources. These are labelled with Γ_k, where $k > 0$, and have corresponding amount of exchanged heat ΔQ_{Γ_k}. If ΔQ_{Γ_k} is negative, heat is subtracted from the system and vice versa. Regions that neither act as heat source or heat sink are labelled with Γ_0, which are also called Hamiltonian regions (see fig. 5). The simulation box, denoted by Ω, and the regions Γ_k are assumed to be moving with velocities v_Ω and v_{Γ_k} respectively.
The change of the energy in a region Γ_k is achieved by rescaling the velocities by a factor ξ_k and shifted by the velocity of the corresponding region:

$$\mathbf{v}_i \to \bar{\mathbf{v}}_i = \xi_k \mathbf{v}_i + (1 - \xi_k)\mathbf{v}_{\Gamma_k} \tag{19}$$

The bar over a quantity denotes the value after the exchange of heat.
The factor ξ_k is given by

$$\xi_k = \sqrt{1 + \frac{\Delta Q_{\Gamma_k}}{\mathcal{K}_{\Gamma_k}}} \tag{20}$$

where ΔQ_{Γ_k} is the exchanged heat in the region Γ_k and $\mathcal{K}_{\Gamma_k}$ is the non-translational kinetic energy of the region Γ_k and is given by

$$\mathcal{K}_{\Gamma_k} = \sum_{i \in \gamma_k} \frac{m_i v_i^2}{2} - \frac{m_{\Gamma_k} v_{\Gamma_k}^2}{2} \tag{21}$$

The sum is taken over all indices in γ_k which is the set of indices of particles in the region Γ_k.
For the final version of the eHEX algorithm there are three more quantities needed. The first one is the heat flux per time step, denoted by $\mathcal{F}_{\Gamma_k}$:

$$\mathcal{F}_{\Gamma_k} = \frac{\Delta Q_{\Gamma_k}}{\Delta t} \tag{22}$$

The second one is the thermostatting force η_i, which is defined as

$$\eta_i = \begin{cases} m_i \, \dfrac{\mathcal{F}_{\Gamma_{k(\mathbf{r}_i)}}}{2\mathcal{K}_{\Gamma_{k(\mathbf{r}_i)}}} \left(\mathbf{v}_i - \mathbf{v}_{\Gamma_{k(\mathbf{r}_i)}}\right) & \text{if } k(\mathbf{r}_i) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

where $k(\mathbf{r}_i)$ is the index of the region in which particle i located, i.e. $k(\mathbf{r}_i) = 0$ means that the particle is in a Hamiltonian region and $k(\mathbf{r}_i) > 0$ denotes the heat sinks and sources.
The last quantity is the one that corrects the long term energy drift of the HEX algorithm, denoted by $\mathcal{E}r_{i,\alpha}$. The analysis and derivation of this term is given in

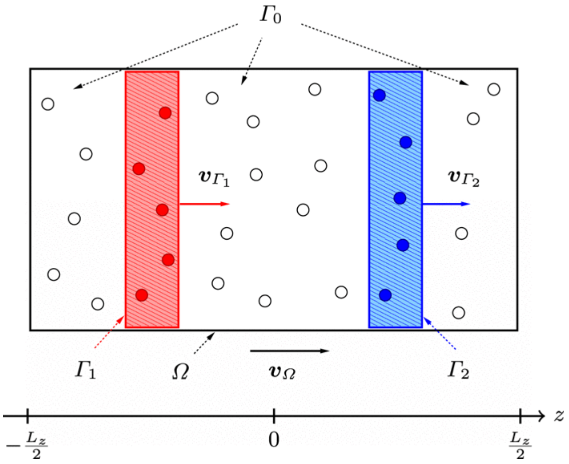


Figure 5: Setup for the use of the HEX and eHEX algorithms. The simulationbox Ω, which is moving with a velocity of v_Ω, contains a heat source (red), Γ_1, which is moving with velocity v_{Γ_1} and a heat sink (blue), Γ_2, which is moving with velocity v_{Γ_2}. The regions that are neither heat sources or heat sinks are denoted by Γ_0.

[14].

$$\begin{aligned}\mathcal{E}r_{i,\alpha} = \frac{\eta_{i,\alpha}}{m_i \mathcal{K}_{\Gamma_{k(\mathbf{r}_i)}}} \left[\frac{\mathcal{F}_{\Gamma_{k(\mathbf{r}_i)}}}{48} + \frac{1}{6} \sum_{j \in \gamma_{k(\mathbf{r}_i)}} \mathbf{f}_j \cdot \left(\mathbf{v}_j - \mathbf{v}_{\Gamma_{k(\mathbf{r}_i)}} \right) \right] \\ - \frac{\mathcal{F}_{\Gamma_{k(\mathbf{r}_i)}}}{12 \mathcal{K}_{\Gamma_{k(\mathbf{r}_i)}}} \left[\frac{f_{i,\alpha}}{m_i} - \frac{1}{m_{\Gamma_{k(\mathbf{r}_i)}}} \sum_{j \in \gamma_{k(\mathbf{r}_i)}} f_{j,\alpha} \right]\end{aligned} \tag{24}$$

The $\mathbf{f}$ in the above equation denotes the force corresponding to the chosen intermolecular potential $U(\mathbf{r})$,

$$\mathbf{f} = -\nabla_{\mathbf{r}_i} U(\mathbf{r}_i) \tag{25}$$

With all the necessary quantities established, the updating sequence of the eHEX algorithm can be written down:

$$\bar{\mathbf{v}}_i^n = \xi_{k(\mathbf{r}_i)}^n \mathbf{v}_i^n + \left(1 - \xi_{k(\mathbf{r}_i)}^n\right) \mathbf{v}_{\Gamma_{k(\mathbf{r}_i)}}^n \tag{26a}$$

$$\bar{\mathbf{v}}_i^{n+\frac{1}{2}} = \bar{\mathbf{v}}_i^n + \frac{\Delta t}{2m_i} \mathbf{f}_i^n \tag{26b}$$

$$\bar{\mathbf{r}}_i^{n+1} = \mathbf{r}_i^n + \Delta t \bar{\mathbf{v}}_i^{n+\frac{1}{2}} \tag{26c}$$

$$\mathbf{f}_i^{n+1} = -\nabla_{\mathbf{r}_i} U(\mathbf{r})|_{\mathbf{r}=\bar{\mathbf{r}}^{n+1}} \tag{26d}$$

$$\bar{\mathbf{v}}_i^{n+1} = \bar{\mathbf{v}}_i^{n+\frac{1}{2}} + \frac{\Delta t}{2m_i} \mathbf{f}_i^{n+1} \tag{26e}$$

$$\mathbf{v}_i^{n+1} = \bar{\xi}_{k(\bar{\mathbf{r}}_i)}^{n+1} \bar{\mathbf{v}}_i^{n+1} + \left(1 - \bar{\xi}_{k(\bar{\mathbf{r}}_i)}^{n+1}\right) \bar{\mathbf{v}}_{\Gamma_{k(\bar{\mathbf{r}}_i)}}^{n+1} \tag{26f}$$

$$\mathbf{r}_i^{n+1} = \bar{\mathbf{r}}^{n+1} - \Delta t^3 \mathcal{E} \bar{\mathbf{r}}_i^{n+1} \tag{26g}$$

5 Results

6 Conclusion

References

[1] Jan Gieseler, Romain Quidant, Christoph Dellago, and Lukas Novotny. Dynamic relaxation of a levitated nanoparticle from a non-equilibrium steady state. *Nat Nano*, 9(5):358–364, May 2014.

[2] Gavin E. Crooks. Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences. *Phys. Rev. E*, 60:2721–2726, Sep 1999.

[3] D. Frenkel and B. Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Computational science series. Elsevier Science, 2001.

[4] Image taken from `https://upload.wikimedia.org/wikipedia/commons/2/27/Elementarzelle_einer_kubisch_raumzentrierten_Elementarzelle.png`.

[5] Idea taken from the Molecular Dynamics part of this lecture: `http://www.physics.buffalo.edu/phy411-506/`.

[6] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Clarendon Press, New York, NY, USA, 1989.

[7] Loup Verlet. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Phys. Rev.*, 159:98–103, Jul 1967.

[8] William C. Swope, Hans C. Andersen, Peter H. Berens, and Kent R. Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *The Journal of Chemical Physics*, 76(1):637–649, 1982.

[9] Philippe H. Hünenberger. *Thermostat Algorithms for Molecular Dynamics Simulations*, pages 105–149. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[10] Shuichi Nosé. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of Chemical Physics*, 81(1):511–519, 1984.

[11] William G. Hoover. Canonical dynamics: Equilibrium phase-space distributions. *Phys. Rev. A*, 31:1695–1697, Mar 1985.

[12] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *The Journal of Chemical Physics*, 81(8):3684–3690, 1984.

[13] B. Hafskjold, T. Ikeshoji, and S. Kjelstrup Ratkje. On the molecular mechanism of thermal diffusion in liquids. *Molecular Physics*, 80:1389–1412, December 1993.

[14] P. Wirnsberger, D. Frenkel, and C. Dellago. An enhanced version of the heat exchange algorithm with excellent energy conservation properties. *The Journal of Chemical Physics*, 143(12), 2015.