Chapter 1

Milestone 4

1.1 Force-Derivation from the Lenard Jones Potential

$$\overrightarrow{f_k} = \sum_i \frac{\partial V}{\partial r_{ik}} \hat{r_{ik}} \tag{1.1}$$

$$V(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right] \tag{1.2}$$

$$\frac{\partial V}{\partial r_{ik}} = 4\epsilon \left(\frac{6\sigma^6}{r_{ik}^6} - \frac{12\sigma^{12}}{r_{ik}^{13}} \right) \tag{1.3}$$

[4]:
```
import sympy as sp
import warnings
warnings.filterwarnings('ignore')
sp.init_printing()
eps = sp.Symbol("e")
sig = sp.Symbol("s")
rad = sp.Symbol("r")
energyRad = 4 * eps * ((sig/rad)**12 - (sig/rad)**6)
energyRad.diff(rad)
```

[4]:
$$4e \left(\frac{6s^6}{r^7} - \frac{12s^{12}}{r^{13}} \right)$$

1.2 Different Time Steps

1.3 Simulation Snapshots

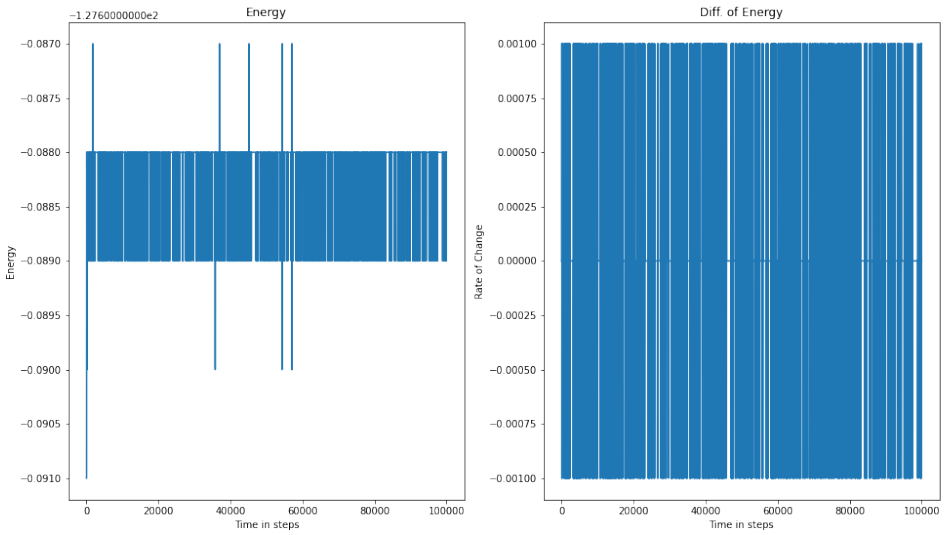

Figure 1.1: Simulation with a time step of 0.001

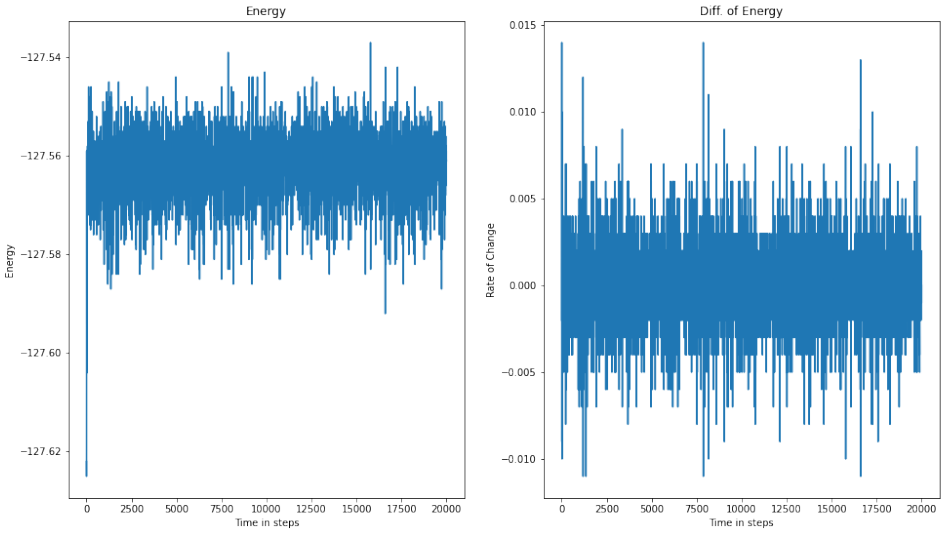

Figure 1.2: Simulation with a time step of 0.005

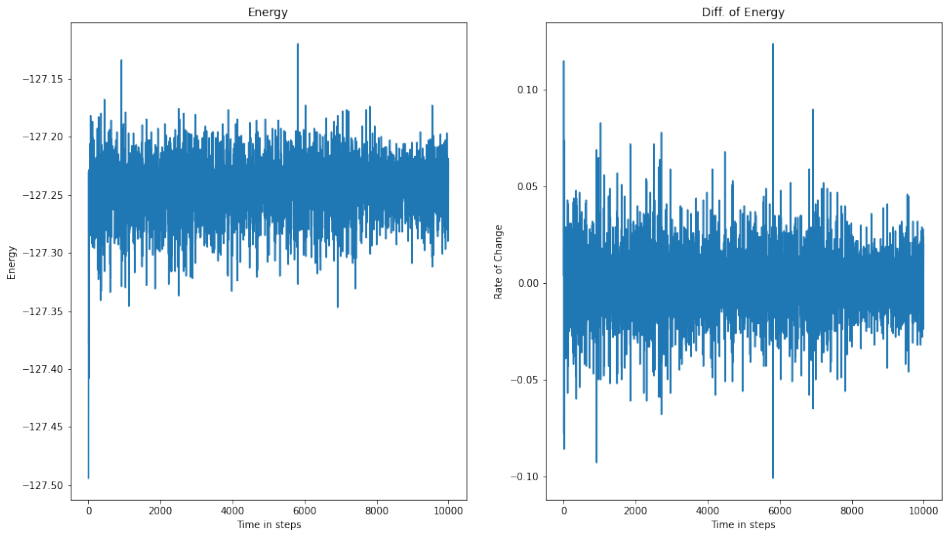

Figure 1.3: Simulation with a time step of 0.01

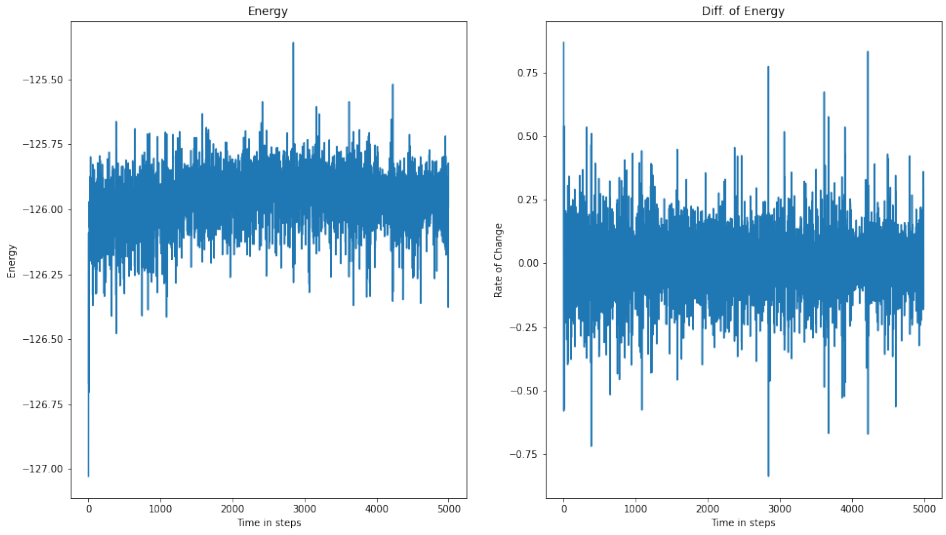

Figure 1.4: Simulation with a time step of 0.02

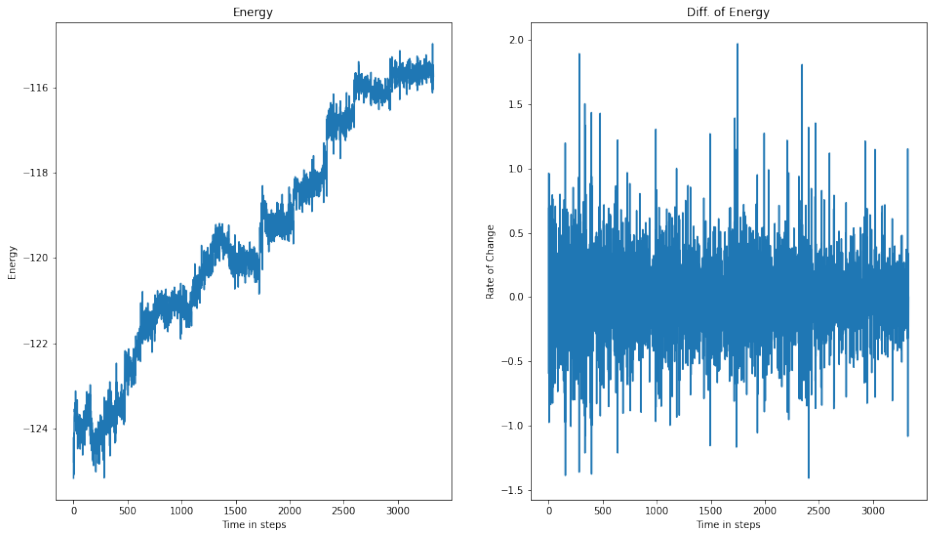

Figure 1.5: Simulation with a time step of 0.03

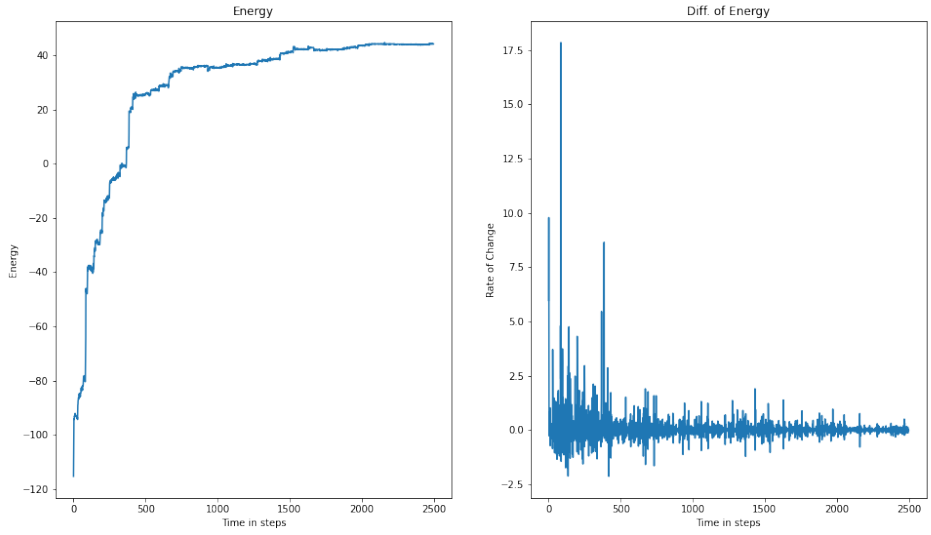

Figure 1.6: Simulation with a time step of 0.04

Figure 1.7: Simulation

Figure 1.8: Simulation

Figure 1.9: Simulation

Figure 1.10: Simulation

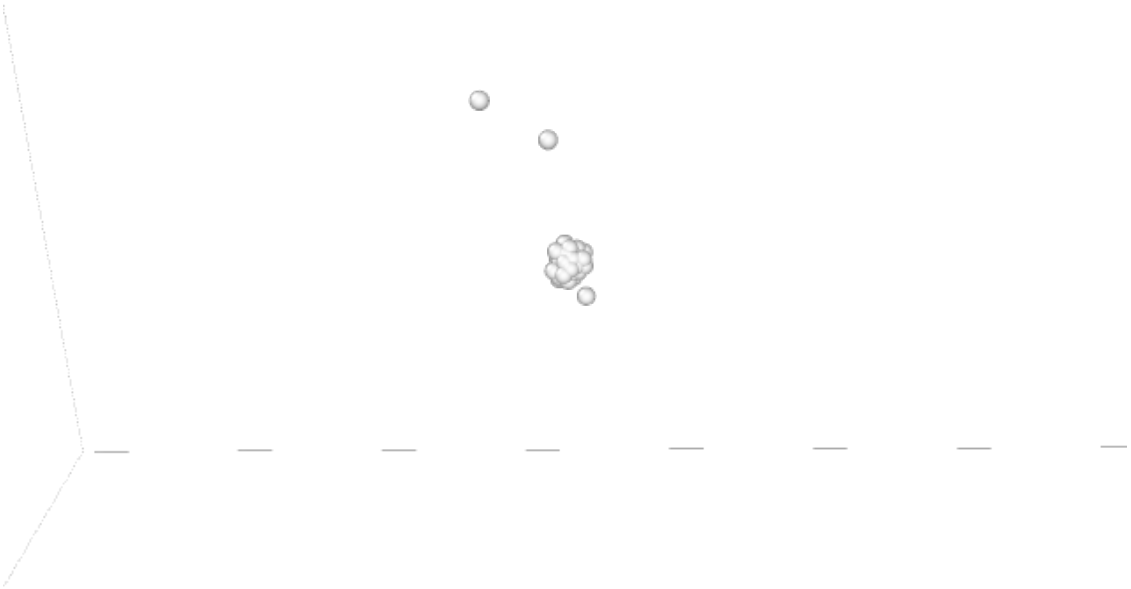

Figure 1.11: Simulation

Chapter 2

Milestone 5

2.1 Berendsen Thermostat Teststrategy

$$\lambda = \sqrt{1 + \left(\frac{T_0}{T} - 1\right)\frac{\Delta t}{\tau}} \tag{2.1}$$

$$T(t) = T_0 + (T_1 - T_0)e^{-t/\tau} \tag{2.2}$$

2.2 Simulation Time

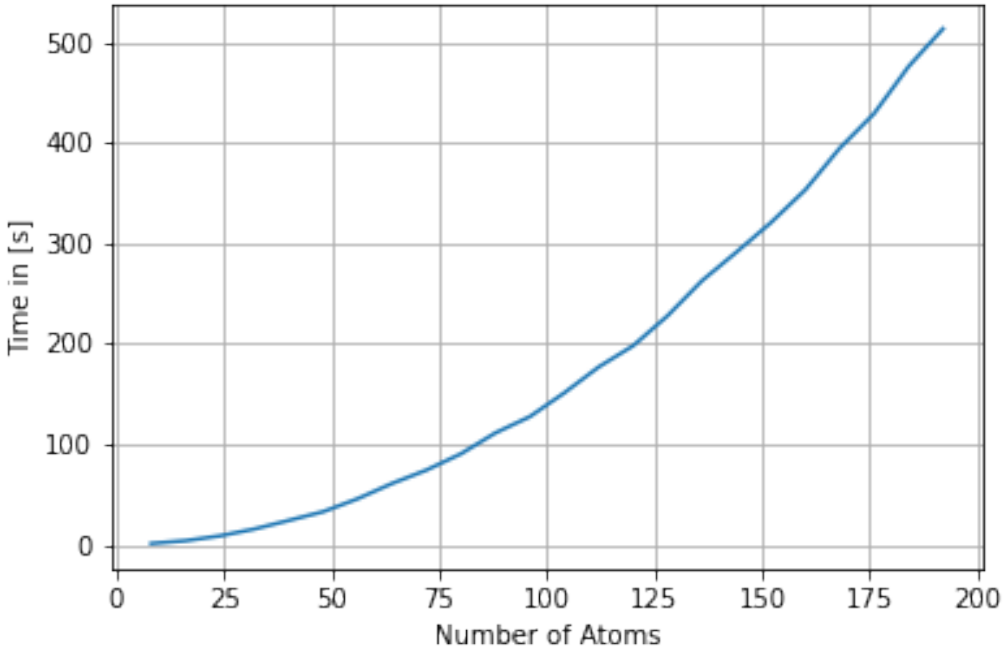

Figure 2.1: Simulationtime from 8 to 192 Atoms

Chapter 3

Milestone 6

The Images are outdated!!!!

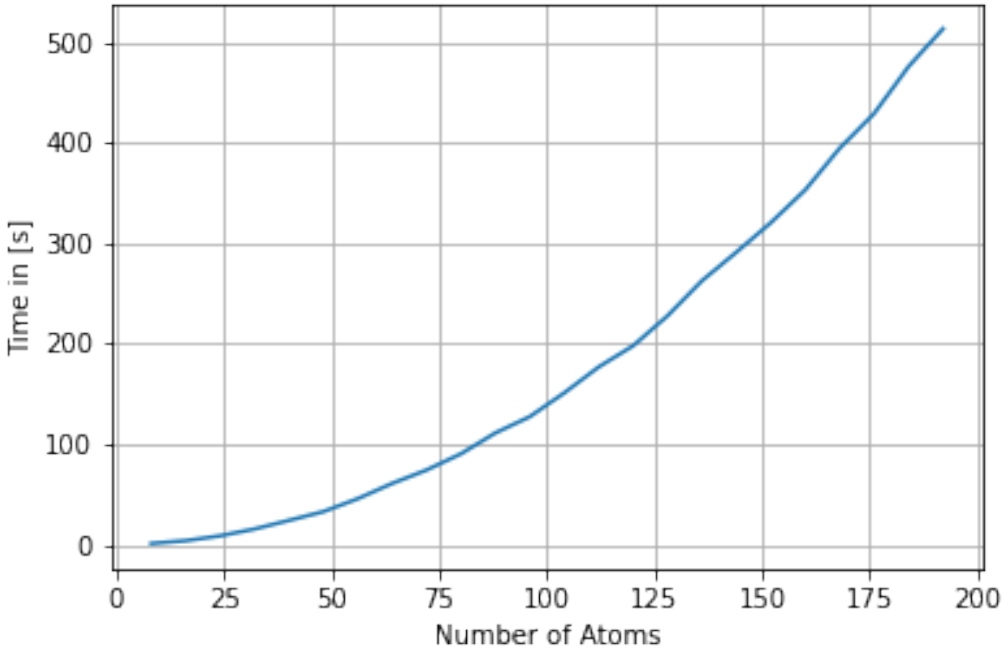

Figure 3.1: Simulationtime without Neighborlist

New images that are correct.

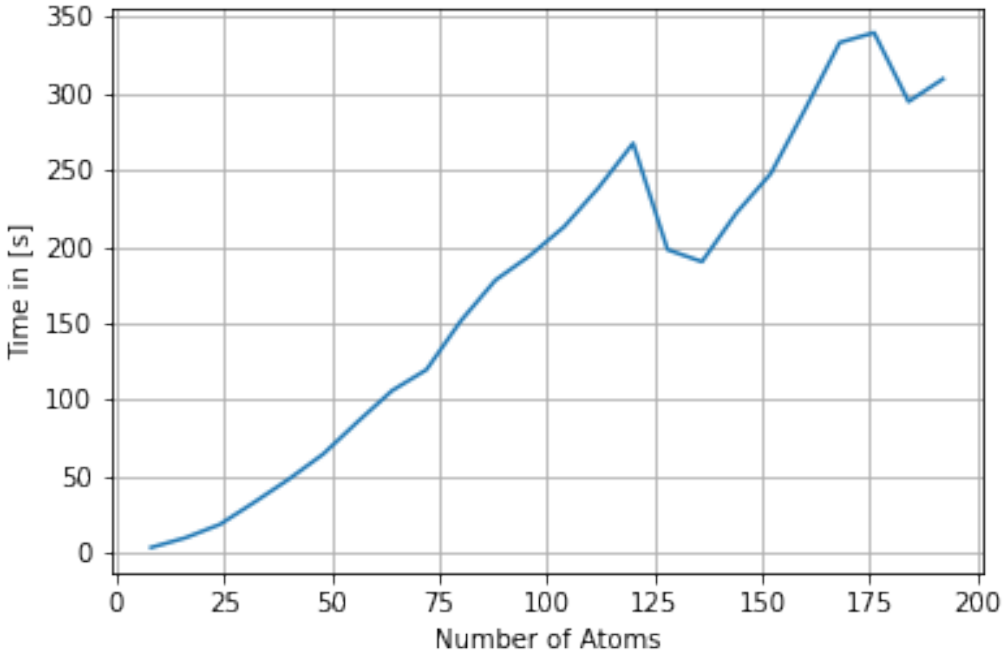

Figure 3.2: Simulationtime with the old Neighborlist

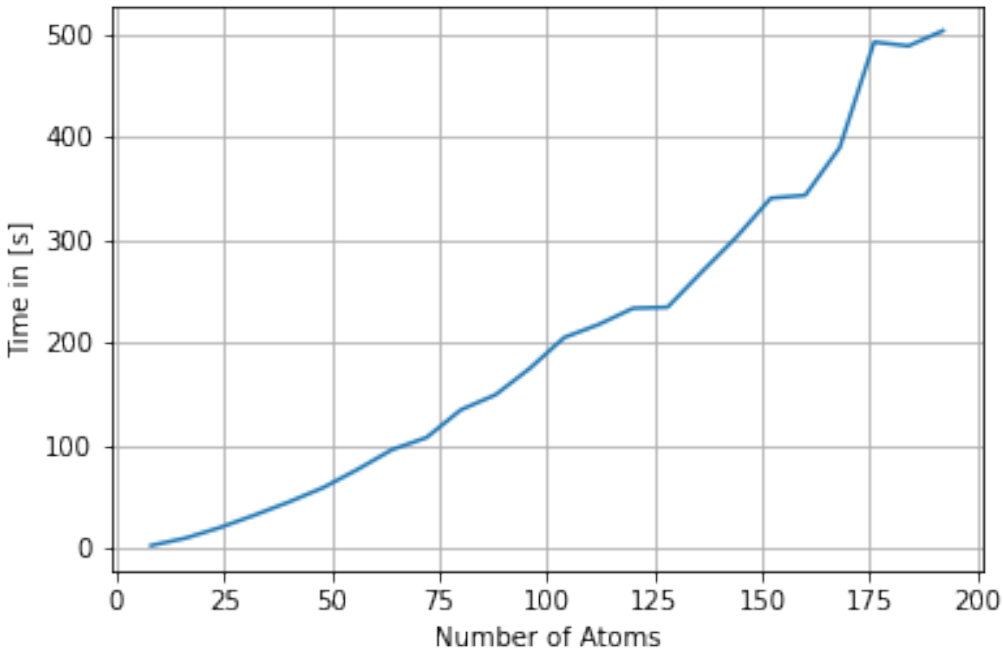

Figure 3.3: Simulationtime with the new Neighborlist

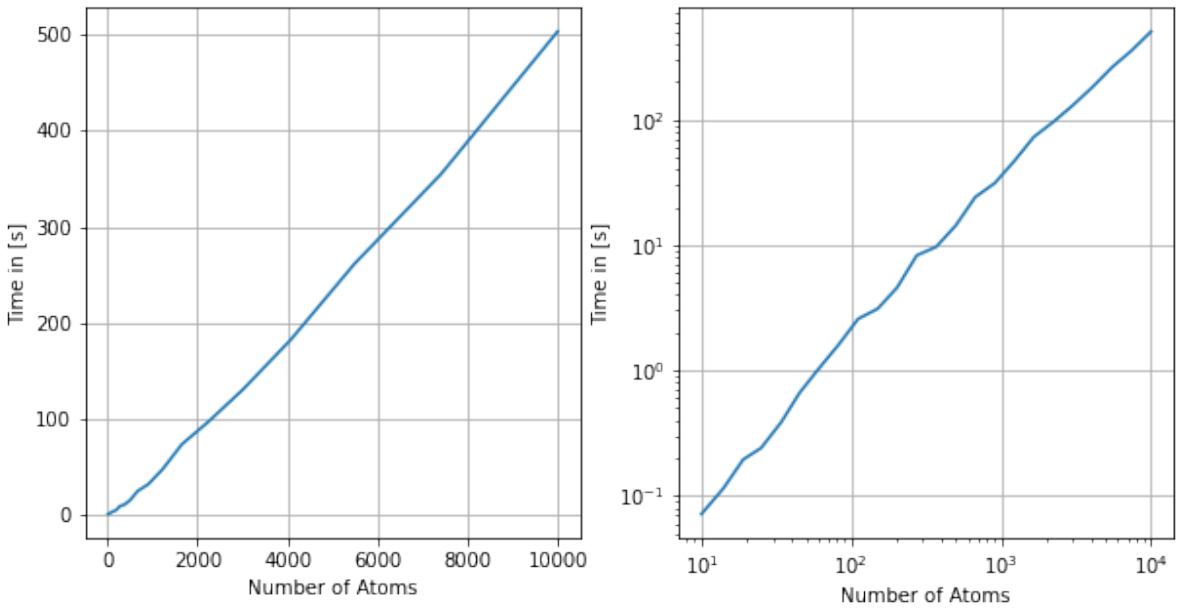

Figure 3.4: Simulationtime with the new Neighborlist

Contents

List of Figures

Chapter 4

Methods

4.1 Integration

4.2 Potentials

4.2.1 Lenard-Jones-Potential

4.2.2 Neighborhood-Search Algorithm

4.2.3 Embedded-Atom Method Potentials

Chapter 5

Implementation

The simulation code was written in C++, most of it just as functions, although the positions, velocities, etc. of the individual atoms where saved in a container-class. While writing the functions, these were also tested with unit-tests. Plots generation and automation for running the project were written in python.

The C++-code was developed in CLion, an IDE which bundles many useful features together (CMake, GDB and Git). The python-code was written in jupyter-notebook. Additional libaries used where: googletest for the unit-tests and eigen for the arrays used for data storage in the container-class.

Chapter 6

Results

6.1 Results from the Lenard-Jones Potential with direct Summation

Figure 6.1: Simulation

6.2 Result from the Simulation with the Berendsen Thermostat

6.3 Results from the Simulation with the Neighborhood-List

Figure 6.2: Simulation

6.4 Results from the Simulation with the Gupta-Potential

Random citation [1]

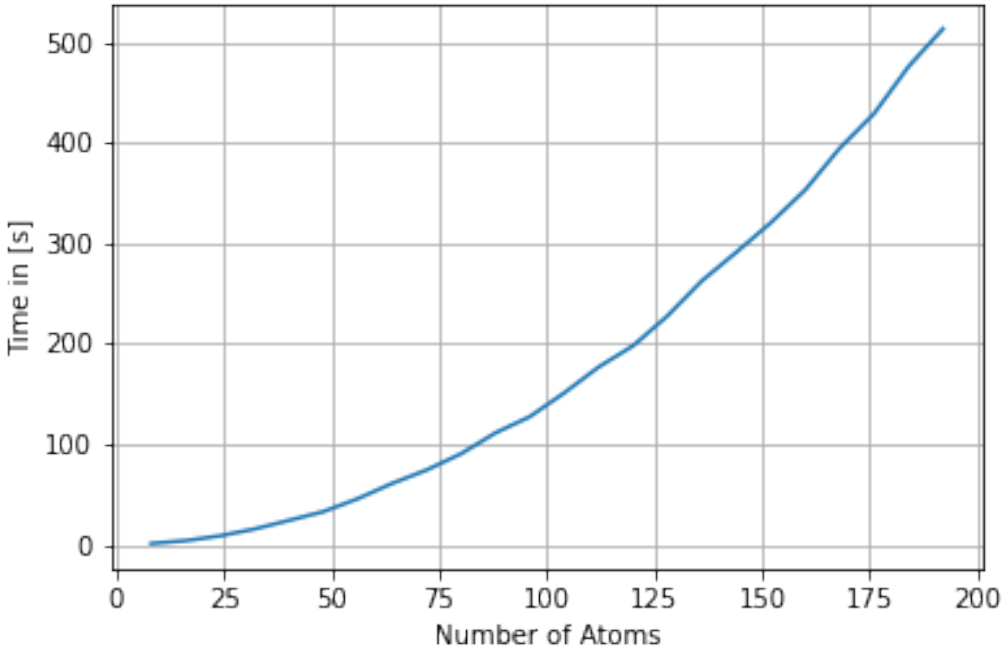

Figure 6.3: Simulation-time with the Berendsen Thermostat from 8 to 192 Atoms

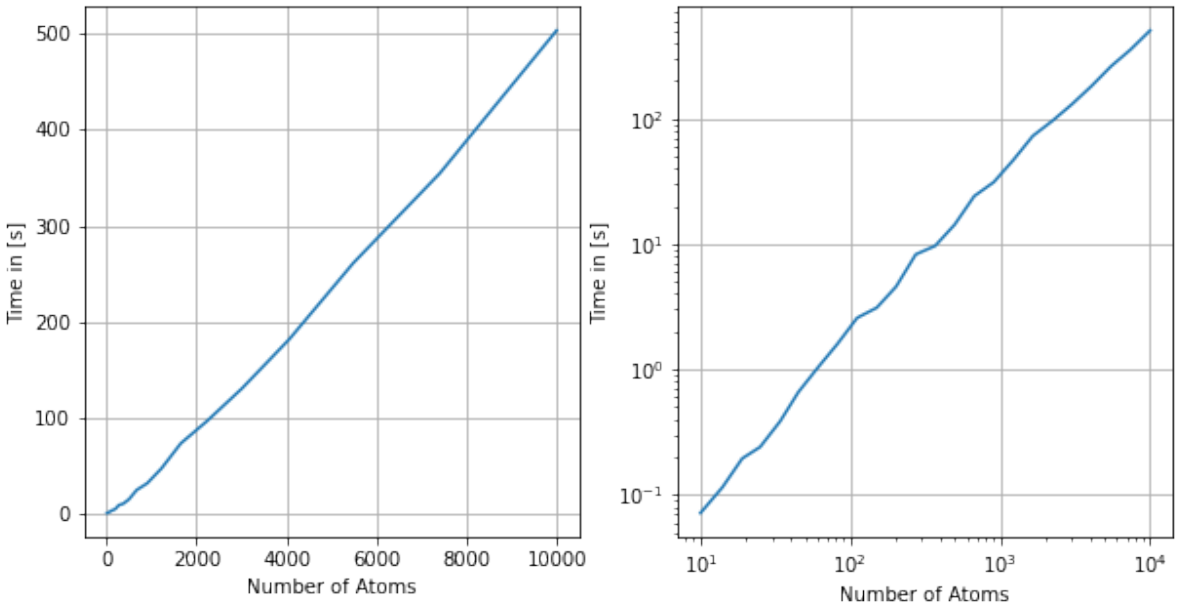

Figure 6.4: Simulation-time with the Neighbor list

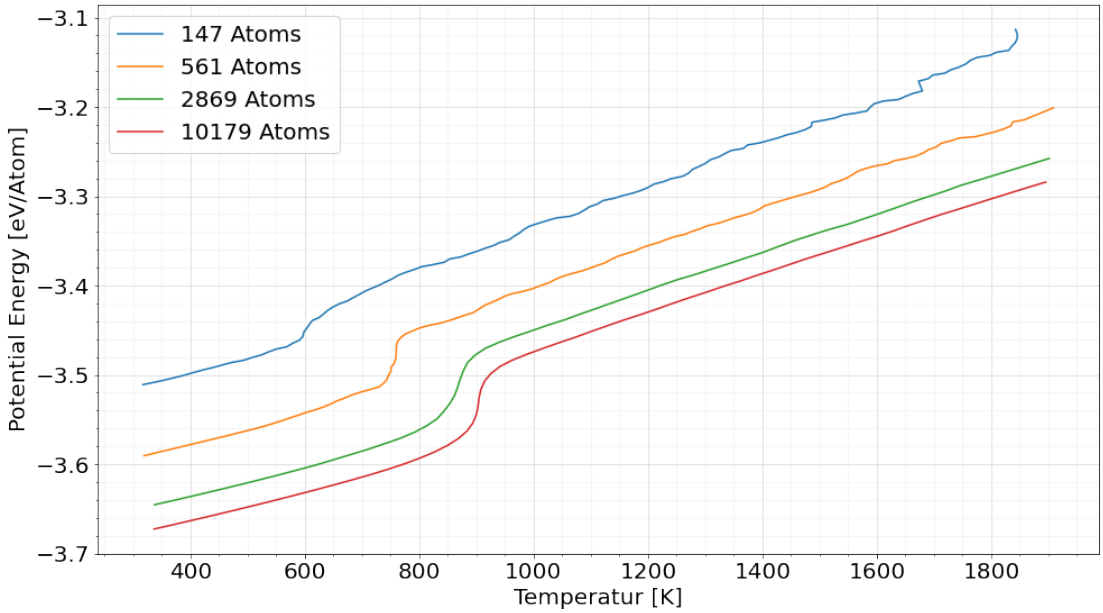

Figure 6.5: Gold Cluster Simulation

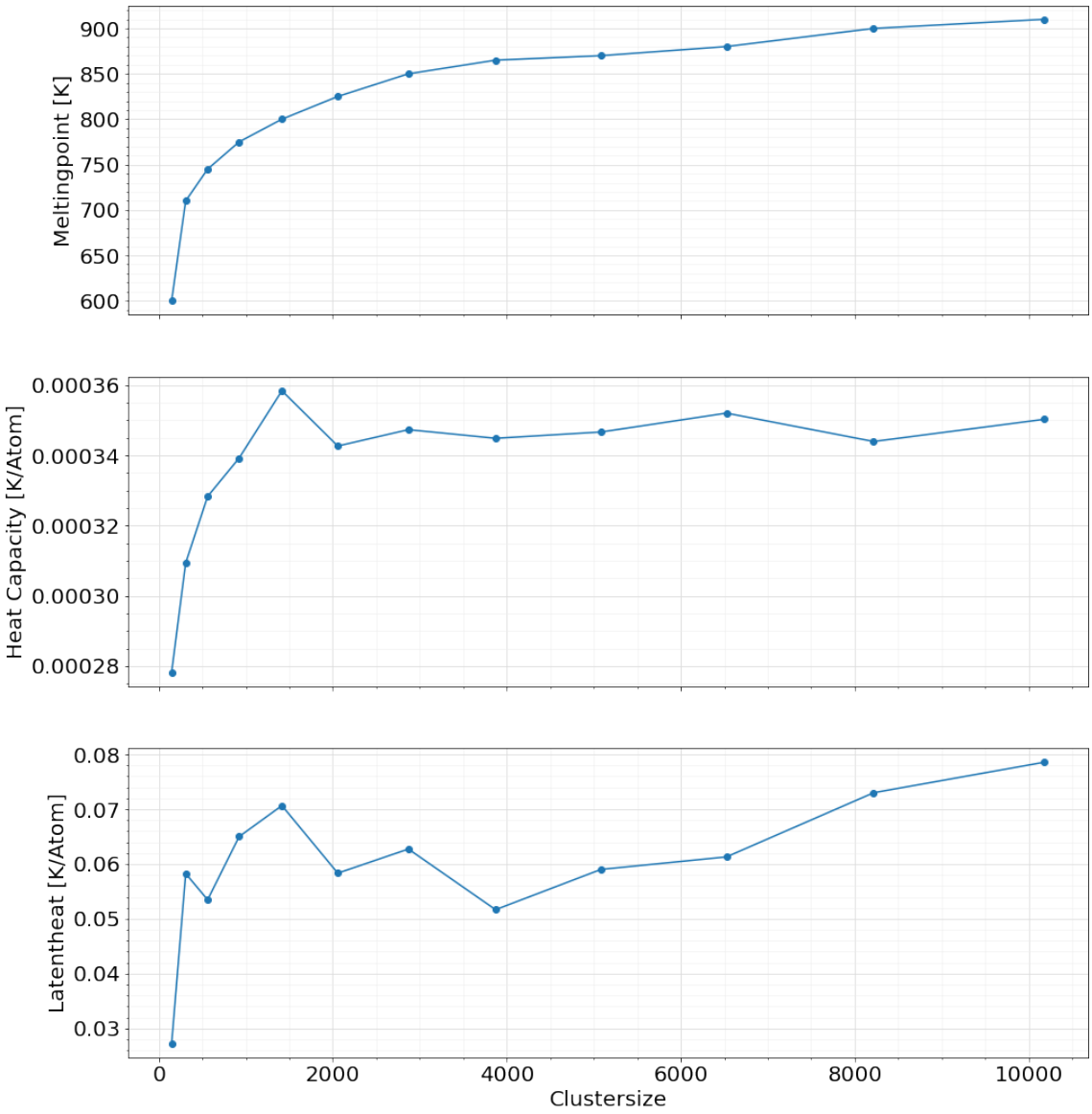

Figure 6.6: Melting Point, Heat Capacity and Latent Heat vs Clustersize

Bibliography

[1] MultiMedia LLC. *MS Windows NT Kernel Description*. 1999. URL: http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm (visited on 09/30/2010).