

Course Report:

Molecular Dynamics Simulation

High-Performance Computing: Molecular Dynamics using C++
University of Freiburg
Winter Semester 2023

Fabian Krause

Contents

1	Introduction	3
2	Methods	3
2.1	Potentials	3
2.2	Integration Methods	5
2.3	Thermodynamic Properties	6
2.4	Thermostats	7
2.5	Stress and Strain	7
2.6	Units	8
3	Implementation	8
3.1	Software and Environment	8
3.2	Algorithms	8
3.2.1	Simulation Loop	8
3.2.2	Neighbor and Cell Lists	9
3.2.3	Domain Decomposition	9
3.3	Validation	12
4	Results	13
4.1	Simple Lennard-Jones Simulation	13
4.2	Equilibration of Cubic Lattice Systems	14
4.3	Melting of Gold Clusters	15
4.4	Deformation of Gold Whiskers	16
5	Conclusions	20

List of Figures

1	Derivation of heat capacity, latent heat and melting point from temperature and total energy data. In this example, the heat capacity is $C = 50 \text{ eV}/191.28 \text{ K} = 0.26 \text{ eV/K}$, the latent heat is 41 eV and the melting point is 710 K. For more information on the experiment, see Chapter 4.3.	6
2	Example of a pitfall when calculating stress in a decomposed domain. The pair (a_1, a_2) contributes to overall stress twice: Once when calculating the stress for domain d_1 , and once when calculating the stress for domain d_2	10
3	Total energy for simulations of a 5083 atom Mackay gold cluster, running for $t = 100 \text{ ps}$ at $\Delta t = 10 \text{ fs}$, using domain decomposition with 8 processes. There were no differences in total energy when using 1, 2 or 4 processes.	11
4	Total energy for simulations of a 5083 atom Mackay gold cluster using Berendsen thermostat with target temperature $T_0 = 300 \text{ K}$, using $\tau = 1 \text{ ps}$, running for $t = 100 \text{ ps}$ at $\Delta t = 10 \text{ fs}$, using domain decomposition with 1, 2, 4 and 8 processes. Only the time range [50 ps, 100 ps] is shown here. Non-transparent lines have been smoothed using a generalized additive model, transparent lines represent true data.	12
5	Total energy of a 54 atom system using Lennard-Jones potentials with direct summation for different timesteps.	13
6	Render of snapshots of the simulation run of a 54 atom system using Lennard-Jones potentials with direct summation and $\Delta t = 0.02$	14
7	Relationship between simulation time and number of atoms in a simple cubic lattice, running equilibration calculations for $t = 500$ at $\Delta t = 0.001$ using Lennard-Jones with and without cutoff. The Berendsen thermostat with $\tau = 2$, doubling after every interval of 100 Lennard-Jones time units, was used.	14
8	Render of snapshots of equilibration of a 64 atom system in a simple cubic lattice, using Lennard-Jones potentials with direct summation and Berendsen thermostat with $\tau = 2$, doubling after every interval of 100 Lennard-Jones time units, for $t = 500$ at $\Delta t = 0.001$	15
9	Energy and temperature of a simulation of a Mackay icosahedron of 2057 Au atoms being heated, leading to melting of the system. The dashed red lines enclose the area of phase transition, where heat is absorbed by the system.	16
10	Heat capacity, melting point and latent heat of Mackay icosahedron of sizes 147, 309, 561, 923, 2057, 3871 and 5083 atoms.	16
11	Stress-strain curves for gold nanowhisker of size 3050 and 51500 atoms at a fixed strain rate for temperatures 0 K, 150 K, 300 K, 450 K, 600 K. Note that for $T \neq 0 \text{ K}$, lines were smoothed using a generalized additive model.	17
12	Stress-strain curves for gold nanowhisker of size 3050 and 51500 atoms at a fixed temperature of $T = 0 \text{ K}$ for different strain rates 10^8 s^{-1} , $5 \cdot 10^8 \text{ s}^{-1}$ and 10^9 s^{-1}	18
13	Zoom-in of stress-strain curve for 3050 atom whisker at 0 K and a strain rate of 10^9 s^{-1}	18
14	3050 atom whisker at different strains.	19

1 Introduction

Molecular Dynamics is a computational method enabling us to simulate atoms and molecules composed thereof. It allows us to study behavior and properties of materials in conditions that are impossible to replicate in reality, to verify that our theories are correct by comparing experimental results with computational results and to get data at a temporal and spatial scale that is yet impossible to obtain using experimental methods [1]. Its applications range from drug discovery [2] to astrophysics [3].

In this report for a lecture held at the University of Freiburg [4], the process and results of implementing a simple yet powerful MD program are described. Features include: A pair and an embedded potential, temperature control, calculation of thermodynamic properties and stress, and parallelization via domain decomposition. The code can be found at <https://github.com/kcaliban/moleculardynamics>.

2 Methods

2.1 Potentials

Potentials determine how potential energy of the system is modeled. Since the force on each atom i is derived from the overall potential energy E_{pot} ,

$$\vec{f}_i = -\frac{\partial}{\partial \vec{r}_i} E_{pot},$$

the choice of potential is important in getting accurate simulations.

Lennard-Jones The Lennard-Jones potential is a *pair potential*, i.e. a summation of energies between atom pairs,

$$E_{pot} = \sum_{i=1}^n \sum_{j=i+1}^n V(r_{ij}).$$

In the *direct summation* method, energies are calculated for each pair based on the (Euclidian) distance r_{ij} . A smarter approach is to use a *cutoff*, that is a certain distance between atoms for which energies are set to zero if atoms are farther away than that distance. This allows us to only calculate $V(r_{ij})$ for atoms that are close to each other.

The pair potentials for Lennard-Jones, given by

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

are a combination of the London dispersion force, approximated by σ^6/r^6 , and Pauli repulsion, approximated by σ^{12}/r^{12} . The London dispersion force is the weakest intermolecular force, caused by temporary dipoles created by the movement of electrons. Pauli repulsion is the reason atoms do not collapse into each other. It is an empirical approximation of quantum mechanical effects. The parameters σ and ϵ set the length and energy scale, respectively.

We now derive the forces acting upon atoms when using the Lennard-Jones potential. The force acting upon a single atom k is given by

$$\vec{f}_k = -\frac{\partial}{\partial \vec{r}_k} E_{pot} = -\left(\sum_{j>k} \frac{\partial}{\partial r_k} V(r_{kj}) + \sum_{j<k} \frac{\partial}{\partial r_k} V(r_{jk}) \right),$$

where r_{kj} is the Euclidian distance between \vec{r}_k and \vec{r}_j . Applying the chain rule, we get

$$\frac{\partial}{\partial r_k} V(r_{kj}) = \frac{\partial V(r_{kj})}{\partial r_{kj}} \cdot \frac{\partial}{\partial r_k} \|\vec{r}_k - \vec{r}_j\|_2.$$

For the second term,

$$\frac{\partial}{\partial r_k} \|\vec{r}_k - \vec{r}_j\|_2 = \frac{\vec{r}_k - \vec{r}_j}{\|\vec{r}_k - \vec{r}_j\|_2},$$

which we get by applying the chain rule after seeing $\|\cdot\|_2 = \sqrt{\|\cdot\|_2^2}$, since $\frac{d}{dx} \sqrt{x} = \frac{1}{2}x^{-1/2}$ and $\frac{d}{dx} x^2 = 2x$. For the first term,

$$\begin{aligned} \frac{\partial V(r)}{\partial r} &= \frac{\partial}{\partial r} 4\epsilon \left[\frac{\sigma^{12}}{r^{12}} - \frac{\sigma^6}{r^6} \right] \\ &= 4\epsilon \left[\frac{\partial}{\partial r} \frac{\sigma^{12}}{r^{12}} - \frac{\partial}{\partial r} \frac{\sigma^6}{r^6} \right] \\ &= 4\epsilon \left[12 \frac{\sigma^{12}}{r^{13}} - 6 \frac{\sigma^6}{r^7} \right]. \end{aligned}$$

We know that $\|\vec{r}_k - \vec{r}_j\|_2 = \|\vec{r}_j - \vec{r}_k\|_2$ and $\vec{r}_k - \vec{r}_j = -(\vec{r}_j - \vec{r}_k)$. The final force acting upon atom k is therefore

$$\begin{aligned} \vec{f}_k &= -\sum_{j>k} 4\epsilon \left[12 \frac{\sigma^{12}}{(\|\vec{r}_k - \vec{r}_j\|_2)^{13}} - 6 \frac{\sigma^6}{(\|\vec{r}_k - \vec{r}_j\|_2)^7} \right] \cdot \frac{\vec{r}_k - \vec{r}_j}{\|\vec{r}_k - \vec{r}_j\|_2} \\ &\quad + \sum_{j<k} 4\epsilon \left[12 \frac{\sigma^{12}}{(\|\vec{r}_k - \vec{r}_j\|_2)^{13}} - 6 \frac{\sigma^6}{(\|\vec{r}_k - \vec{r}_j\|_2)^7} \right] \cdot \frac{\vec{r}_k - \vec{r}_j}{\|\vec{r}_k - \vec{r}_j\|_2}. \end{aligned}$$

Ducastelle Ducastelle is an *embedded-atom potential* (EAM), i.e. the summation of a repulsion energy and an embedding energy,

$$E_{pot} = E_{rep} + E_{embedding}.$$

The embedding energy approximates the influence of surrounding electrons on an atom, and is given by the sum

$$E_{embedding} = \sum_i \mathcal{F}(\rho_i),$$

where \mathcal{F} is a function on the local density ρ_i of the system for each atom i . The local density is usually given by the sum over a function f on the distance to other atoms,

$$\rho_i = \sum_j f(r_{ij}).$$

A trivial way to calculate the local density for a certain cutoff r_c is to count the number of atoms within the sphere defined by said cutoff, and divide this number by the volume of the sphere. To achieve this, f is chosen as a step function that is $1/V$ if $r_{ij} \leq r_c$, and 0 otherwise. Since this function is not differentiable, and we would like to analytically derive forces, a smooth function with $f(r_{ij}) > 0$ for $r_{ij} \leq r_c$ and $f(r_{ij}) \approx 0$ for $r_{ij} > r_c$ is used instead. The EAM used in this report is based on the work of Ducastelle [5], Gupta [6] and Cleri and Rosato [7], its implementation was provided by Pastewka, Nöhring, and Frérot [4].

2.2 Integration Methods

To calculate the movements of atoms, we require their positions \vec{r}_i and velocities \vec{v}_i . Newton's Second Law of Motion gives us the following system of differential equations:

$$\begin{aligned}\dot{\vec{v}}_i(t) &= \vec{f}_i(t)/m, \\ \dot{\vec{r}}_i(t) &= \vec{v}_i(t),\end{aligned}$$

since $\vec{f}_i(t) = m\vec{a}_i(t) = m\dot{\vec{v}}_i(t) = m\ddot{\vec{r}}_i(t)$. To get the positions and velocities, we require numerical methods for integrating above equations.

Verlet The Verlet integration method is derived by addition and subtraction of the third order Taylor expansion of atomic positions at $t + \Delta t$ and $t - \Delta t$,

$$\begin{aligned}\vec{r}_i(t + \Delta t) &= \vec{r}_i(t) + \vec{v}_i(t)\Delta t + \frac{1}{2m}\vec{f}_i(t)\Delta t^2 + \frac{1}{6}\ddot{\vec{v}}_i(t)\Delta t^3 + \mathcal{O}(\Delta t^4), \\ \vec{r}_i(t - \Delta t) &= \vec{r}_i(t) - \vec{v}_i(t)\Delta t + \frac{1}{2m}\vec{f}_i(t)\Delta t^2 - \frac{1}{6}\ddot{\vec{v}}_i(t)\Delta t^3 + \mathcal{O}(\Delta t^4).\end{aligned}$$

Adding $\vec{r}_i(t + \Delta t)$ to $\vec{r}_i(t - \Delta t)$ yields

$$\vec{r}_i(t + \Delta t) = 2\vec{r}_i(t) + \frac{1}{m}\vec{f}_i(t)\Delta t^2 - \vec{r}_i(t - \Delta t) + \mathcal{O}(\Delta t^4),$$

and subtracting $\vec{r}_i(t - \Delta t)$ from $\vec{r}_i(t + \Delta t)$ yields

$$\vec{v}_i(t + \Delta t) = \frac{\vec{r}_i(t + \Delta t) - \vec{r}_i(t - \Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^3),$$

giving us both the positions and velocities of atoms. An issue with this algorithm is that at the beginning, $t = 0$, both the positions at $t = 0$ and $t = -\Delta t$ have to be known, although initial conditions are usually only given at $t = 0$.

Velocity-Verlet The Velocity-Verlet integration method, which can be shown to be equivalent to the Velocity integration method, does not have the same issue with initial conditions. It is given by the following equations,

$$\begin{aligned}\vec{r}_i(t + \Delta t) &= \vec{r}_i(t) + \vec{v}_i(t)\Delta t + \frac{1}{2m}\vec{f}_i(t)\Delta t^2 + \mathcal{O}(\Delta t^4), \\ \vec{v}_i(t + \Delta t) &= \vec{v}_i(t) + \frac{1}{2m}(\vec{f}_i(t) + \vec{f}_i(t + \Delta t))\Delta t + \mathcal{O}(\Delta t^3).\end{aligned}$$

At timestep t , only current positions and velocities have to be known. Forces at timestep $t + \Delta t$ have to be known as well, hence the algorithm is often split into a predictor-corrector scheme,

where velocities are first calculated for $t + \Delta t/2$, positions are updated, new forces calculated and finally velocities calculated for $t + \Delta t$:

$$\begin{aligned}\vec{v}_i(t + \Delta t/2) &= \vec{v}_i(t) + \frac{1}{2m} \vec{f}_i(t) \Delta t, \\ \vec{r}_i(t + \Delta t) &= \vec{r}_i(t) + \vec{v}_i(t + \Delta t/2) \Delta t, \\ \vec{v}_i(t + \Delta t) &= \vec{v}_i(t + \Delta t/2) + \frac{1}{2m} \vec{f}_i(t + \Delta t) \Delta t.\end{aligned}$$

2.3 Thermodynamic Properties

Temperature The temperature of a system is calculated using the kinetic theory of gases,

$$T = \frac{2}{3Nk_B} \sum_i \frac{1}{2} m \|\vec{v}_i\|_2^2,$$

where k_B is the Boltzmann constant, m is the mass of the atoms (in this report, all atoms of a system have the same mass) and N is the number of atoms.

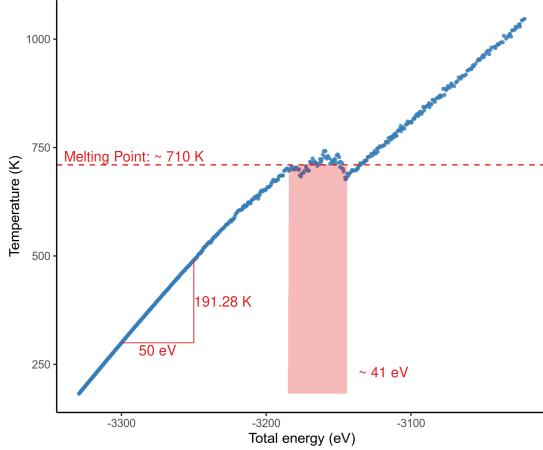


Figure 1: Derivation of heat capacity, latent heat and melting point from temperature and total energy data. In this example, the heat capacity is $C = 50 \text{ eV}/191.28 \text{ K} = 0.26 \text{ eV/K}$, the latent heat is 41 eV and the melting point is 710 K. For more information on the experiment, see Chapter 4.3.

Heat Capacity, Latent Heat, Melting Point Heat capacity is the amount of heat that has to be added to a system to raise its temperature,

$$C = \lim_{\Delta T \rightarrow 0} \frac{\Delta Q}{\Delta T}.$$

In isolated systems, we can write $C = dE/dT$. Latent heat is the amount of heat that gets absorbed by a system during a phase transition, i.e. the amount of heat required for the system to change phase. The melting point is simply the temperature at which the transition from solid to liquid occurs.

These properties relating to phase transitions can be approximated from measurements of total energy and temperature, see Figure 1, which shows the properties for experiments which will be described in detail in Chapter 4.3.

2.4 Thermostats

Since the system of equations is an *initial value problem*, we often need to guess the initial configuration, i.e. positions and velocities of atoms at $t = 0$. This guess is likely not close to the equilibrium, which can lead to a quick increase in temperature during simulation, and ultimately in the vaporization of the structure. To stop this from happening, thermostats can be used to equilibrate the system at a certain temperature by manipulating velocities.

Velocity Rescaling Velocity rescaling is the simplest form of a thermostat. The target temperature T_0 is held by directly setting the velocities after each time step:

$$\vec{v}_i := \vec{v}_i \sqrt{\frac{T_0}{T}}.$$

To see why this works, we take a look at the formula for temperature and substitute in the changed velocities:

$$\begin{aligned} T &= \frac{2}{3Nk_B} \sum_i \frac{1}{2} m \|\vec{v}_i\|_2^2 \\ &= \frac{2}{3Nk_B} \sum_i \frac{1}{2} m \|\sqrt{\frac{T_0}{T}} \vec{v}_i\|_2^2 \\ &= \frac{T_0}{T} \frac{2}{3Nk_B} \sum_i \frac{1}{2} m \|\vec{v}_i\|_2^2 = \frac{T_0}{T} T = T_0. \end{aligned}$$

Since this method is quite intrusive, it is not used in practical situations.

Berendsen The Berendsen thermostat is a more gentle method for temperature control. It works by exponentially relaxing the temperature from the initial value T towards the target value T_0 ,

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

The variable τ is the so-called relaxation time constant, which determines how strongly the heat bath is coupled to the atomic system. Berendsen is also implemented by scaling velocities, but not directly to the target temperature. During a single timestep Δt , the temperature should change from the current temperature T to $T_0 + (T - T_0)e^{-\Delta t/\tau}$, so the velocity is rescaled as follows:

$$\vec{v}_i := \vec{v}_i \sqrt{\frac{T_0}{T} + \left(1 - \frac{T_0}{T}\right) e^{-\Delta t/\tau}} \approx \vec{v}_i \sqrt{\frac{T_0}{T} + \left(1 - \frac{T_0}{T}\right) \frac{\Delta t}{\tau}}.$$

2.5 Stress and Strain

Cauchy stress is calculated using the virial stress theorem as follows:

$$\begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} = \underline{\sigma} = \frac{1}{V} \sum_{i < j} \vec{r}_{ij} \otimes \vec{f}_{ij}.$$

Note that at $T \neq 0$ K, velocities should be taken into account. This would be beyond the scope of this report, and was therefore skipped. Strain is simply the proportion of additional length of the system at a specific time $L(t)$ to the initial length $L(0)$ of the system,

$$\epsilon(t) = (L(t) - L(0))/L(0) = L(t)/L(0) - 1.$$

2.6 Units

When using Lennard-Jones potentials, our units are dimensionless. Using the Ducastelle potential, we fix the units as follows. We set $[E] = \text{eV}$, $[l] = \text{\AA}$ and we want the timestep to be $[t] = 1 \text{ fs}$. We need to get the unit for mass:

$$[m] = [E][t]^2/[l]^2 = 1.6 \cdot 10^{-19} \text{ J} \times 10^{-30} \text{ s}^2 / (10^{-20} \text{ m}) = 1.6 \cdot 10^{-29} \text{ kg}.$$

Since we want to express our mass in terms of g/mol, we calculate:

$$[m] = 1.6 \cdot 10^{-29} \text{ kg} \times \text{mol/mol} = 1.6 \cdot 10^{-29} \times 10^3 \times 6 \cdot 10^{23} \text{ g/mol} = 0.009649 \text{ g/mol}.$$

All masses in g/mol have to be multiplied by $1/0.009649 = 103.6$ to get a timestep of 1 fs. To set the unit for temperature, we need to use the Boltzmann constant in the right units. Kinetic energy will be given in eV,

$$[l]^2/[t]^2 \cdot [m]^2 = [l]^2/[t]^2 \cdot [E][t]^2/[l]^2 = [E] = \text{eV},$$

hence the Boltzmann constant must have unit eV/K to get Kelvin:

$$\begin{aligned} k_B &= 1.380 \times 10^{-23} \text{ J/K} \\ &= 1.380 \cdot 6.242 \times 10^{18} \times 10^{-23} \text{ eV/K} \\ &= 8.617 \cdot 10^{-5} \text{ eV/K}. \end{aligned}$$

3 Implementation

3.1 Software and Environment

If not specified otherwise, computations were conducted on a Lenovo laptop with an 8 core/16 thread AMD Ryzen 7 5800U CPU and 14 GB of RAM (2 GB of the total 16 GB are reserved for VRAM) with Manjaro Linux (5.15.133-1-MANJARO) installed.

Code was entirely written in C++. For parallel computations, Open MPI 4.1.5 [8] was used. For parsing command line arguments, the header-only library cxxopts (<https://github.com/jarro2783/cxxopts>) was used. Efficient vector and matrix storage and calculations relied on the Eigen [9] library. CMake (<https://cmake.org/>) was used as the build system. Most of the implementation was provided by Pastewka, Nöhring, and Frérot [4]. The full implementation can be found at <https://github.com/kcaliban/moleculardynamics>. For visualization, manipulation and analysis of molecular structures, the tool OVITO [10] was used.

3.2 Algorithms

3.2.1 Simulation Loop

The main simulation loop used for all experiments consists of the following steps:

1. Predictor-step of Velocity-Verlet. Updating positions and velocities using previous forces.
2. If using: Updating of neighbor lists, necessary after position updates.
3. Calculation of new forces and potential energy using the specified potential.
4. Corrector-step of Velocity-Verlet. Updating velocities using new forces.
5. If using: Application of Berendsen thermostat.
6. Calculation of kinetic energy and thermodynamic properties, if of interest.
7. Output of quantities and writing of trajectory at specified intervals.

3.2.2 Neighbor and Cell Lists

When using potentials with a cutoff, the simulation code must know for all atoms the atoms within the range of the cutoff. The data structures for this purpose are called neighbor lists. A trivial way to create neighbor lists is to calculate the distances between each atom pair and fill the lists accordingly. The time complexity of this naive approach is $\mathcal{O}(n^2)$ for n atoms.

A better solution works by dividing the simulation domain into a grid of cells. The underlying data structures are called cell lists. Atoms are assigned to a cell by simply dividing the coordinates by the cell size. To get the neighbors of an atom, only the distances to atoms in the same and any adjacent cells have to be checked. This algorithm has the improved time complexity of $\mathcal{O}(n)$ for n atoms.

Note that the implementation provided by Pastewka, Nöhring, and Frérot [4] already provides a neighbor list based on cell lists, i.e. already checks distances to atoms in the same and adjacent cells during creation of the neighbor list, so this does not have to happen in the implementation of the potentials.

3.2.3 Domain Decomposition

Domain decomposition essentially divides the simulation domain into cells which are simulated by separate processes in parallel. This requires the exchange of (ghost) atoms between cells and the incorporation of ghost atoms. The implementation provided by Pastewka, Nöhring, and Frérot [4] uses the C++ library MPI as the underlying inter-process communication mechanism.

When a simulation step leads to an atom leaving the current domain, the atom has to be transferred to the process of the target domain. This is done via the MPI function `MPI_Sendrecv`. Ghost atoms are atoms that do not live within a domain, but who influence the forces of the atoms in a domain. Since all potentials in this report use a cutoff, ghost atoms are simply all atoms in adjacent cells within the distance of the used cutoff r_c from the boundaries of the current cell. Note that EAMs require the calculation of the embedding energy $E_{embedding}$ also of ghost atoms, therefore the cutoff for ghost atoms has to be $2r_c$, as the embedding energy depends on atoms in a range of r_c of an atom. Ghost atoms also have to be updated after every simulation step. They also serve the purpose of representing atoms beyond periodic bounds.

Domain decomposition requires some modifications in the way energies and stress are calculated. For the calculation of potential and kinetic energy, each process calculates the energies of local atoms, and `MPI_Allreduce` is used to sum up the energies of all domains. Note that the

temperature used for the Berendsen thermostat must depend on the total kinetic energy.

For stress calculation, we could naively calculate the stress for each domain separately. But then we would miss the forces of atoms between domains. Another approach is to calculate the stress for all pairs of atoms in a domain d , and all pairs of atoms where one atom is in the domain, and the other atom is a ghost atom,

$$\sigma(d) = \frac{1}{2V} \sum_{i,j \in d} \vec{r}_{ij} \otimes \vec{f}_{ij} + \frac{1}{V} \sum_{\substack{i \in d, \\ j \in \text{ghosts}(d)}} \vec{r}_{ij} \otimes \vec{f}_{ij}.$$

The issue with this approach is that the stress contribution of pairs that lie within the domain cutoff will be counted twice, see Figure 2: The pair (a_1, a_2) contributes to overall stress once when calculating the stress for domain d_1 , and once when calculating the stress for domain d_2 .

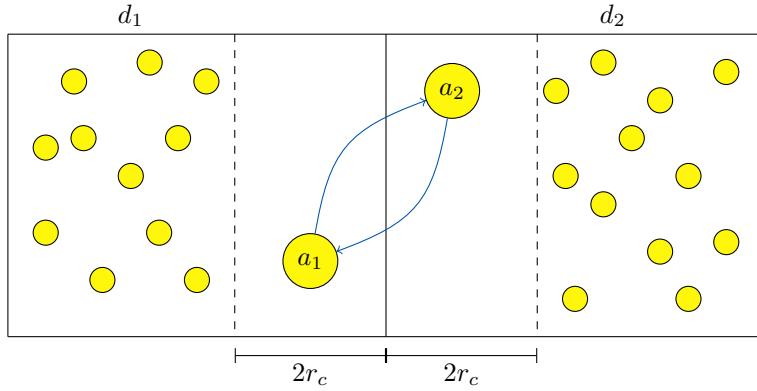


Figure 2: Example of a pitfall when calculating stress in a decomposed domain. The pair (a_1, a_2) contributes to overall stress twice: Once when calculating the stress for domain d_1 , and once when calculating the stress for domain d_2 .

The solution is simple:

1. For each domain d , separately calculate stress based on atoms local to the domain, $\sigma_{local}(d)$, and stress based on atom pairs where one atom is in the domain, and the other is a ghost atom, $\sigma_{ghost}(d)$
2. Sum up the stress as follows:

$$\sigma = \sum_d \sigma_{local}(d) + \frac{1}{2} \sum_d \sigma_{ghost}(d)$$

Since we are counting all interactions between domain and ghost atoms exactly twice, we can simply divide this by two. Note that we also only count stress arising from ghost atoms at periodic boundaries once.

Another pitfall when using domain decomposition is that in the case of the provided implementation, the origin of the domain had to be $(0, 0, 0)$. Mackay clusters generated with the

provided script had to be affinely transformed such that the atoms have strictly positive positions. Furthermore, sufficient space had to be added around the structures to allow atoms to move around. This was done using OVITO [10], with a space of 20 Å around structures.

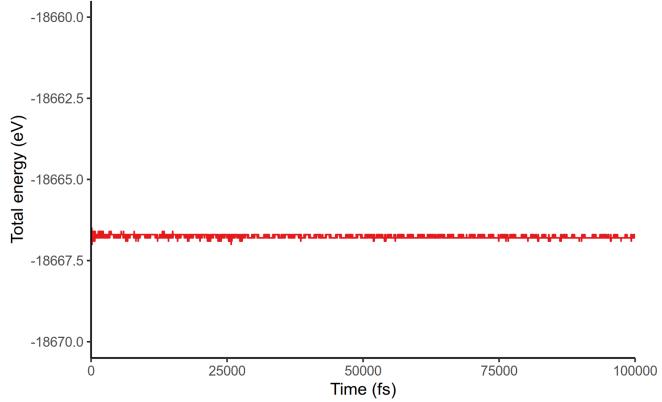


Figure 3: Total energy for simulations of a 5083 atom Mackay gold cluster, running for $t = 100$ ps at $\Delta t = 10$ fs, using domain decomposition with 8 processes. There were no differences in total energy when using 1, 2 or 4 processes.

Figure 3 shows the total energy and time for a simulation of a 5083 atom Mackay gold cluster using the Ducastelle potential, running for $t = 100$ ps at $\Delta t = 10$ fs on 8 parallel processes. Even though the trajectories and outputs are identical, there may be small differences in energies due to different orders of summation of floating point numbers. When using the Berendsen thermostat, these small differences lead to different trajectories, as velocities are adjusted based on temperature, which is based on kinetic energies. This in turn leads to bigger differences in the calculated energies, as demonstrated in Figure 4.

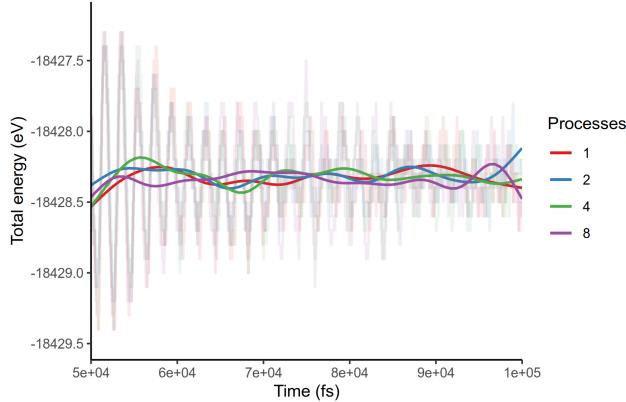


Figure 4: Total energy for simulations of a 5083 atom Mackay gold cluster using Berendsen thermostat with target temperature $T_0 = 300\text{ K}$, using $\tau = 1\text{ ps}$, running for $t = 100\text{ ps}$ at $\Delta t = 10\text{ fs}$, using domain decomposition with 1, 2, 4 and 8 processes. Only the time range $[50\text{ ps}, 100\text{ ps}]$ is shown here. Non-transparent lines have been smoothed using a generalized additive model, transparent lines represent true data.

3.3 Validation

Much of the implementation provided by Pastewka, Nöhring, and Frérot [4] already included tests. The following are additional implemented tests.

Velocity-Verlet To verify that the implementation of Velocity-Verlet yields correct results, two simple tests were created. In the first test, a single atom with initial velocities $\vec{v}(0)$, positions $\vec{r}(0)$ and constant forces \vec{f} acting on it was simulated for 10 steps with a timestep of $\Delta t = 0.1$, i.e. $t = 1$. Afterwards, the resulting positions and velocities were compared with the analytical results, obtained by

$$\begin{aligned}\vec{v}(t) &= \vec{v}(0) + \vec{f} \cdot t, \\ \vec{r}(t) &= \vec{r}(0) + \vec{v}(0) \cdot t + \frac{1}{2} \cdot \vec{f} \cdot t^2.\end{aligned}$$

The second test is identical, except that three atoms with different (constant) forces acting upon them are simulated.

Berendsen Two tests were implemented for the Berendsen thermostat. In the first test, we verify that the temperature reaches within 0.1 of the target temperature in a single step of Velocity-Verlet and Lennard-Jones followed by Berendsen given $\tau = 1$ and $\Delta t = 1$. For $\tau = 1 = \Delta t$, Berendsen corresponds to velocity rescaling. The test uses 10 atoms with randomly initialized velocities and positions in the range $[0, 1]$. The target temperature is set to $300K$.

In the second test, a timestep of $\Delta t = 1$ is chosen, but τ is set to 2. The test also uses 10 atoms with randomly initialized velocities and positions in the same range, and the same target temperature. This time, simulation using Velocity-Verlet and Lennard-Jones followed by Berendsen is run for 100 steps. After these steps, it is checked whether the temperature is within 0.1 of the target temperature.

Virial Stress The tests for virial stress include a simple calculation using three atoms and tests to make sure that the modified version used when domain decomposition is enabled returns the same results.

Kinetic Energy and Temperature For kinetic energy and temperature, simple regression tests are performed to ascertain that the underlying formulae are correct.

4 Results

4.1 Simple Lennard-Jones Simulation

As a simple first simulation, a system of 54 atoms was simulated for 100 Lennard-Jones time units using direct summation. Simulations were repeated with different timesteps to determine the optimal one. Results can be seen in Figure 5. The timestep $\Delta t = 0.04$ is too large and leads to the system evaporating. For $\Delta t = 0.03$, the system evaporates almost completely, and energy is not conserved. When using $\Delta t = 0.02$, $\Delta t = 0.01$ or $\Delta t = 0.001$, only a few atoms are lost, and the total energy is stable. Therefore, a timestep of $\Delta t = 0.02$ seems reasonable for this system. A series of snapshots from the simulation run using this timestep can be seen in Figure 6.

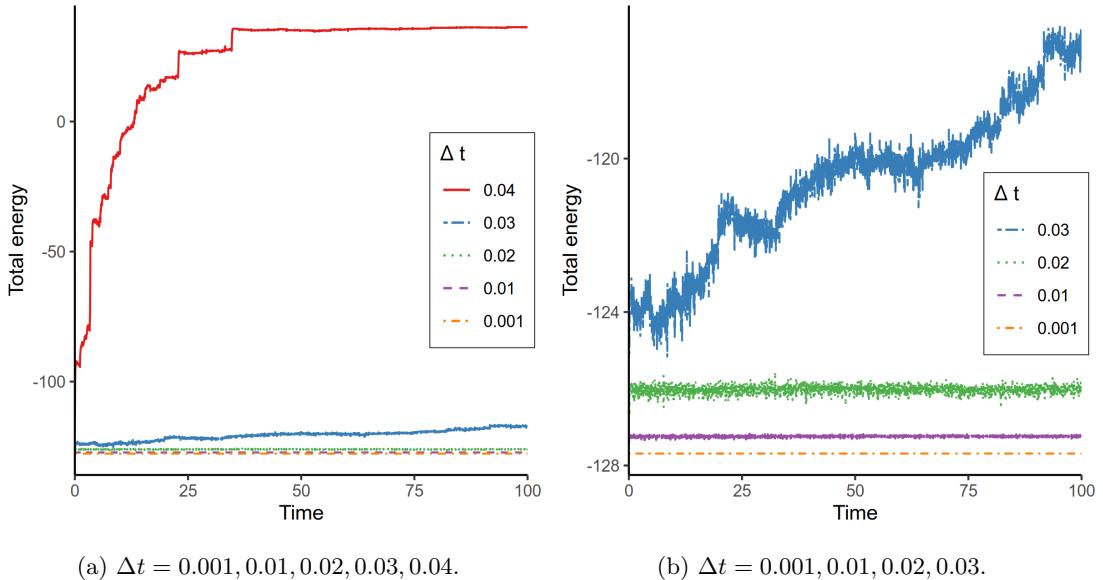


Figure 5: Total energy of a 54 atom system using Lennard-Jones potentials with direct summation for different timesteps.

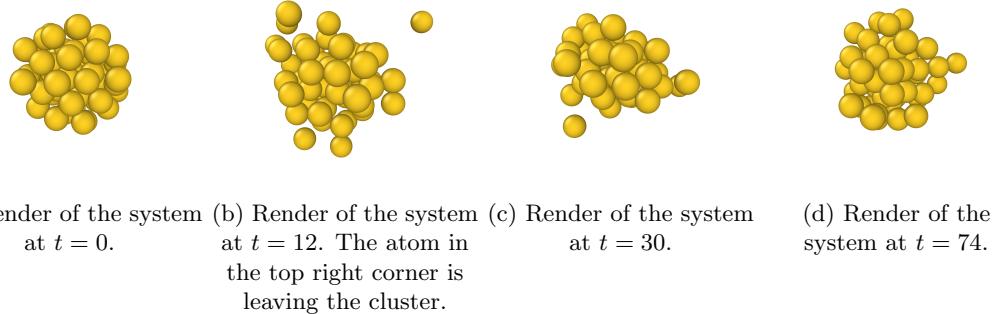


Figure 6: Render of snapshots of the simulation run of a 54 atom system using Lennard-Jones potentials with direct summation and $\Delta t = 0.02$.

4.2 Equilibration of Cubic Lattice Systems

As a next step, cubic lattice systems of various sizes were equilibrated for $t = 500$ at $\Delta t = 0.001$, once using Lennard-Jones potentials with direct summation and once using the usual cutoff of $r_c = 2.5\sigma$, see Toxvaerd and Dyre [11]. The Berendsen thermostat with $\tau = 2$ was used for thermal control. After every interval of 100 Lennard-Jones time units, τ was doubled to lessen the impact of the thermostat on the simulation.

Figure 7 shows the relationship between number of atoms and time in seconds for simulation runs with and without using a cutoff. When using no cutoff, each simulation step requires calculation of forces for all n^2 pairs, given n atoms, leading to a quadratic time complexity $\mathcal{O}(n^2)$. When using a cutoff and a cell list datastructure, this complexity reduces to $\mathcal{O}(n)$, as discussed in Chapter 3.2.2. Figure 8 shows the initial lattice configuration as well as the resulting equilibrium of a 64 atom system for the simulation run using the direct-summation Lennard-Jones potential.

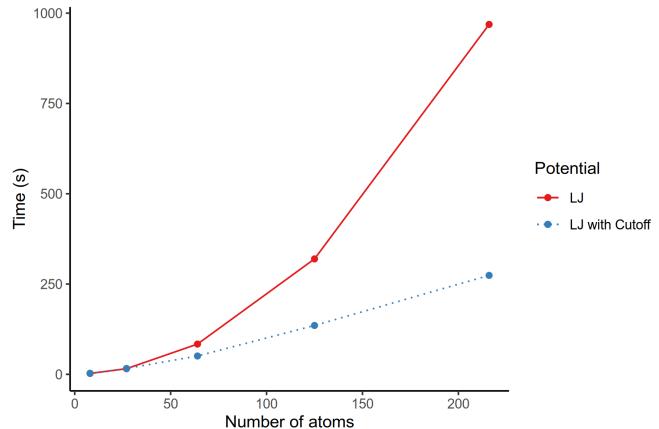
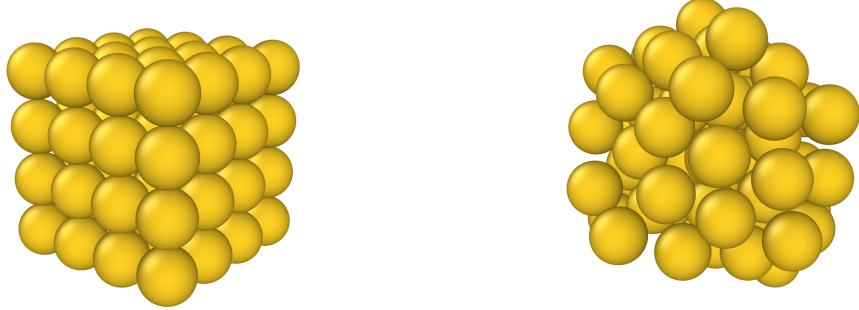


Figure 7: Relationship between simulation time and number of atoms in a simple cubic lattice, running equilibration calculations for $t = 500$ at $\Delta t = 0.001$ using Lennard-Jones with and without cutoff. The Berendsen thermostat with $\tau = 2$, doubling after every interval of 100 Lennard-Jones time units, was used.



(a) Render of the system at $t = 0$. Structure is a simple cubic lattice. (b) Render of the system at $t = 61$, after the system is fully equilibrated.

Figure 8: Render of snapshots of equilibration of a 64 atom system in a simple cubic lattice, using Lennard-Jones potentials with direct summation and Berendsen thermostat with $\tau = 2$, doubling after every interval of 100 Lennard-Jones time units, for $t = 500$ at $\Delta t = 0.001$.

4.3 Melting of Gold Clusters

Next, gold clusters of various sizes were melted in a NVE ensemble. The Ducastelle potential with a cutoff of $r_c = 10 \text{ \AA}$ and the parameters for gold atoms described in Cleri and Rosato [7] were used. The simulated systems were of sizes 147, 309, 561, 923, 2057, 3871 and 5083 atoms. First, the systems were simulated for 100 ps with an initial velocity of zero for all atoms. No thermostat was used. A timestep of $\Delta t = 20 \text{ fs}$ was chosen for this initial simulation, as it lead to stable results.

Next, heat was continuously added to the systems via velocity rescaling. The process went as follows:

1. Increase temperature by 1 K through velocity rescaling, adding heat to the system.
2. Run the simulation for $t = 10 \text{ ps}$ at $\Delta t = 10 \text{ fs}$.
3. Measure temperature and total energy for $t = 10 \text{ ps}$ at $\Delta t = 10 \text{ fs}$.
4. Using the output of this simulation as input, repeat step 1.

This process was repeated 600 times for each system, for a total simulation time of 12.1 ns per system. Temperatures and total energies were averaged for each measurement phase. The smaller timestep of $\Delta t = 10 \text{ fs}$ was chosen to allow for more accuracy in measurement. As an example, the measurements for the cluster of size 2057 can be seen in Figure 9. Heat capacity, latent heat and melting point can be directly derived from this figure. The dashed red lines enclose the area of phase transition, where heat is absorbed by the system.

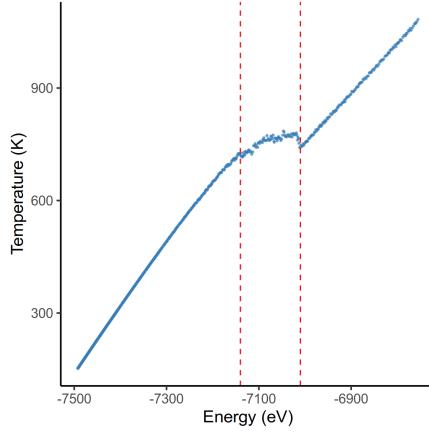
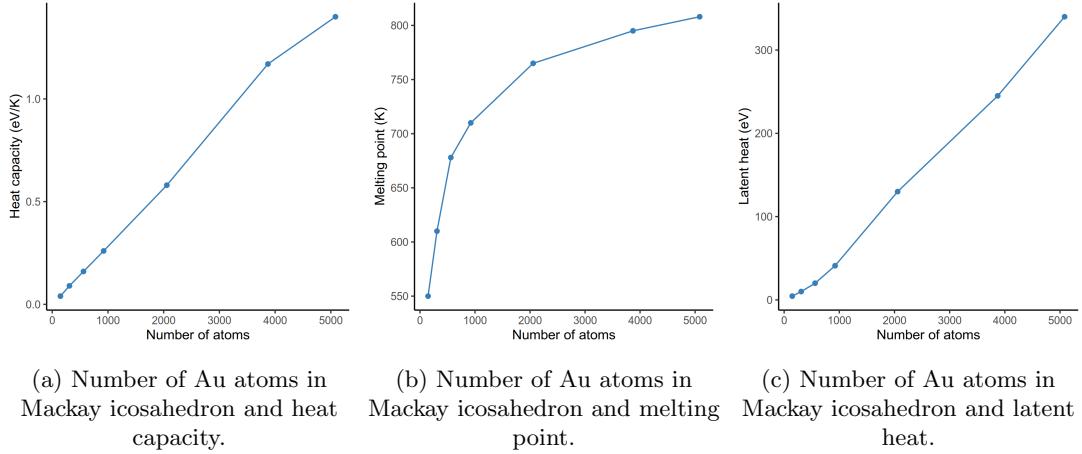


Figure 9: Energy and temperatue of a simulation of a Mackay icosahedron of 2057 Au atoms being heated, leading to melting of the system. The dashed red lines enclose the area of phase transition, where heat is absorbed by the system.

Derived thermodynamic properties of all systems can be seen in Figure 10. Relationship between system size and heat capacity as well as latent heat seem to be of a linear nature, whereas the melting point data suggest an underlying logarithmic relationship, perhaps approaching the melting point of gold. Note that the calculated heat capacity and latent heat are not *specific*, i.e. they do not depend on the mass of the system, explaining the linear relationships.



(a) Number of Au atoms in Mackay icosahedron and heat capacity.

(b) Number of Au atoms in Mackay icosahedron and melting point.

(c) Number of Au atoms in Mackay icosahedron and latent heat.

Figure 10: Heat capacity, melting point and latent heat of Mackay icosahedron of sizes 147, 309, 561, 923, 2057, 3871 and 5083 atoms.

4.4 Deformation of Gold Whiskers

In a final experiment, two nanowhiskers of 3050 and 51500 gold atoms, with diameters of 57.4 Å and 114.8 Å, respectively, were strained along the z axis in a NVT ensemble. The Ducastelle potential was used, with $r_c = 10$ Å and the standard gold parameters described in Cleri and Rosato [7]. The domain was decomposed along the z axis, using periodic conditions along

this axis and a ghost atom cutoff of $2r_c$. The effects of various strain rates and temperatures were examined. A Berendsen thermostat with $\tau = 500$ fs was used for to control temperature. The systems with zero velocities were first equilibrated at the target temperature for 10 ps at $\Delta t = 5$ fs, afterwards the strain was continuously increased at the given strain rate and stress measured. Computations were performed on the bwUniCluster2.0 (<https://wiki.bwhpc.de/e/BwUniCluster2.0>). Simulation time depended on strain rate and ranged from 0.5 ns to 3 ns. Figure 11 shows the stress-strain curves at a fixed strain rate for various temperatures, showing that higher temperature lead to earlier defects. At $T = 600$ K, there is small necking in the smaller nanowhisker almost immediately. The bigger nanowhisker fractures only at this temperature, given the 10^9 s^{-1} strain rate (this rate was chosen given the number of experiments and maximum job time of 72 hours on the bwUniCluster2.0). In Figure 12, stress-strain curves at a constant temperature are presented for different strain rates. It can be observed that higher strain rates result in delayed necking, a pattern that will eventually reverse at a certain $\Delta\epsilon_{\max}$.

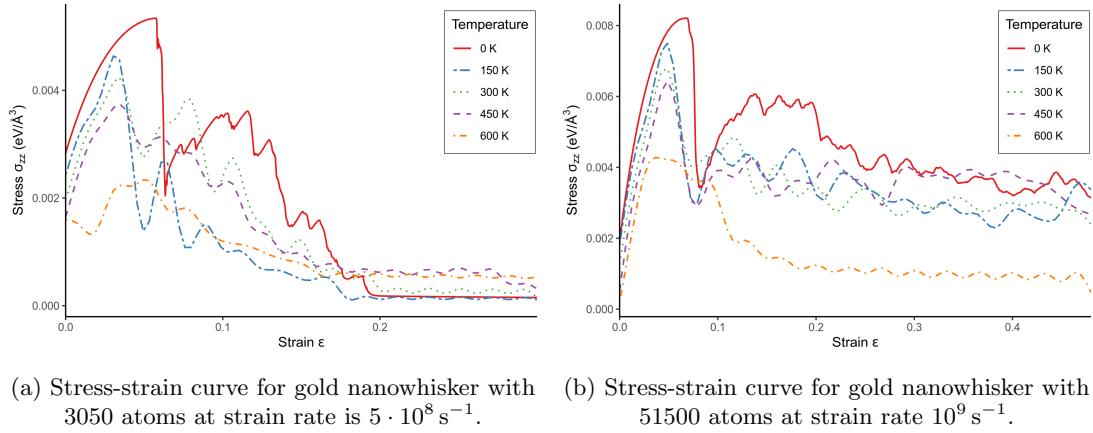
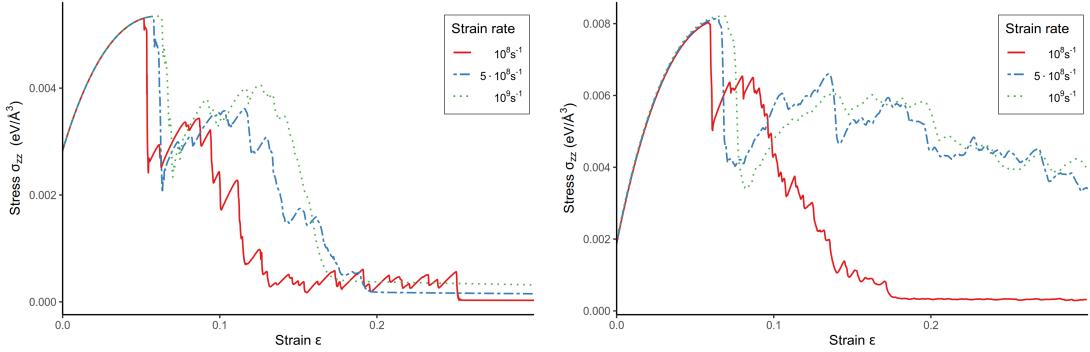


Figure 11: Stress-strain curves for gold nanowhisker of size 3050 and 51500 atoms at a fixed strain rate for temperatures 0 K, 150 K, 300 K, 450 K, 600 K. Note that for $T \neq 0$ K, lines were smoothed using a generalized additive model.



(a) Stress-strain curve for gold nanowhisker with 3050 atoms at various strain rates. (b) Stress-strain curve for gold nanowhisker with 51500 atoms at various temperatures.

Figure 12: Stress-strain curves for gold nanowhisker of size 3050 and 51500 atoms at a fixed temperature of $T = 0 \text{ K}$ for different strain rates 10^8 s^{-1} , $5 \cdot 10^8 \text{ s}^{-1}$ and 10^9 s^{-1} .

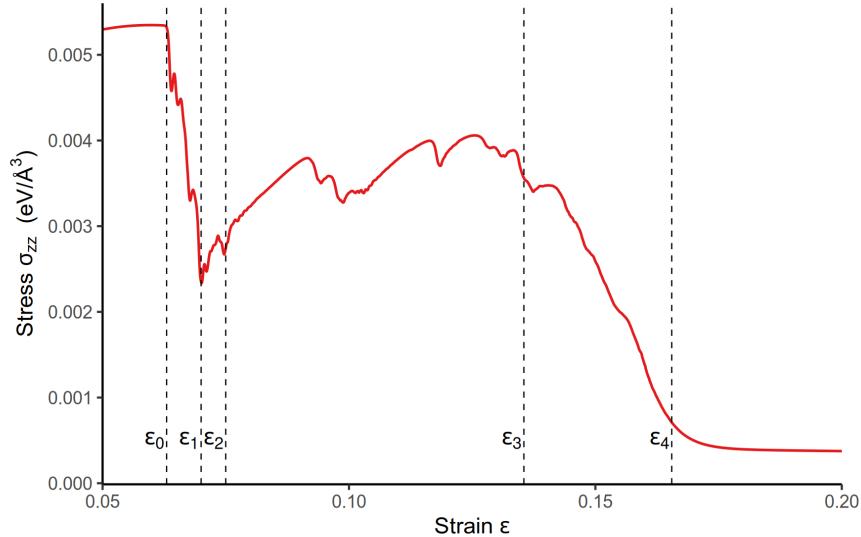


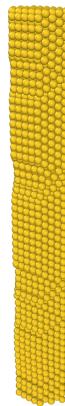
Figure 13: Zoom-in of stress-strain curve for 3050 atom whisker at 0 K and a strain rate of 10^9 s^{-1} .

We now take a closer look at the defects of the 3050 atom whisker at 0 K and a strain rate of 10^9 s^{-1} . A zoom-in of the corresponding stress-strain curve can be seen in Figure 13. Results of dislocation analysis (Dislocation Extraction Algorithm [12]) using OVITO [10] at the different ϵ_i can be seen in Figure 14. At ϵ_0 , the whisker has reached the point of ultimate strength. At ϵ_1 , necking is observable on the surface of the whisker. Dislocation analysis yielded only dislocations labeled as “other”. At ϵ_2 , Shockley dislocations, which are a partial dislocation characterized by a partial extra plane of atoms inserted within the crystal structure, can be observed. The same dislocations can be observed at ϵ_3 , right before fracturing of the whisker at

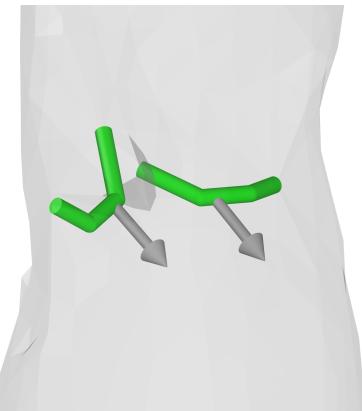
the location shown. The fractured whisker can be observed at ϵ_4 .



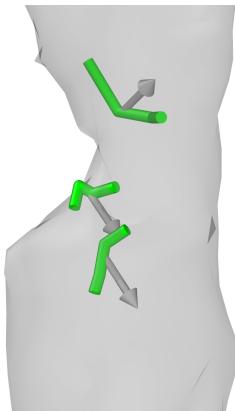
(a) Nanowhisker at ϵ_0 , at point of ultimate strength.



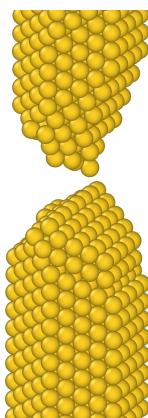
(b) Nanowhisker at ϵ_1 , showing necking.



(c) Shockley dislocations at ϵ_2 . Burgers vectors are shown by gray arrows.



(d) Shockley dislocations at ϵ_3 . Burgers vectors are shown by gray arrows.



(e) Fractured whisker at ϵ_4 .

Figure 14: 3050 atom whisker at different strains.

5 Conclusions

We have successfully implemented simulation methods for Molecular Dynamics simulations, improved these through clever algorithms such as cell lists and domain decomposition, and verified our implementation by performing experiments ranging from simple simulations of small systems to investigating the properties of a 51500-atom material under strain.

To widen the scope of application of this software, support for systems with differing atoms may be added. Furthermore, additional potentials and integration methods could cover more use-cases. Further validation through reproduction of well-established MD experiments can solidify the trust in this implementation.

References

- [1] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. USA: Clarendon Press, 1989. ISBN: 0198556454.
- [2] Helena Engel et al. “finDr: A web server for in silico D-peptide ligand identification”. In: *Synthetic and Systems Biotechnology* 6.4 (2021), pp. 402–413. ISSN: 2405-805X. DOI: <https://doi.org/10.1016/j.synbio.2021.11.004>. URL: <https://www.sciencedirect.com/science/article/pii/S2405805X21000727>.
- [3] N. Pineau et al. “Molecular dynamics simulations of shock compressed heterogeneous materials. II. The graphite/diamond transition case for astrophysics applications”. In: *Journal of Applied Physics* 117.11 (Mar. 2015), p. 115902. ISSN: 0021-8979. DOI: [10.1063/1.4914481](https://doi.org/10.1063/1.4914481). eprint: https://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/1.4914481/13294055/115902_1_online.pdf. URL: <https://doi.org/10.1063/1.4914481>.
- [4] Lars Pastewka, Wolfram Nöhring, and Lucas Frérot. *Molecular Dynamics Course Materials*. 2023. URL: <https://pastewka.github.io/MolecularDynamics/>.
- [5] F. Ducastelle. “Modules élastiques des métaux de transition”. In: *J. Phys. France* 31 (1970), pp. 1055–1062. DOI: [10.1051/jphys:019700031011-120105500](https://doi.org/10.1051/jphys:019700031011-120105500). URL: <https://doi.org/10.1051/jphys:019700031011-120105500>.
- [6] Raju P. Gupta. “Lattice relaxation at a metal surface”. In: *Phys. Rev. B* 23 (12 June 1981), pp. 6265–6270. DOI: [10.1103/PhysRevB.23.6265](https://doi.org/10.1103/PhysRevB.23.6265). URL: <https://link.aps.org/doi/10.1103/PhysRevB.23.6265>.
- [7] Fabrizio Cleri and Vittorio Rosato. “Tight-binding potentials for transition metals and alloys”. In: *Phys. Rev. B* 48 (1 July 1993), pp. 22–33. DOI: [10.1103/PhysRevB.48.22](https://doi.org/10.1103/PhysRevB.48.22). URL: <https://link.aps.org/doi/10.1103/PhysRevB.48.22>.
- [8] Edgar Gabriel et al. “Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation”. In: *Proceedings, 11th European PVM/MPI Users’ Group Meeting*. Budapest, Hungary, Sept. 2004, pp. 97–104.
- [9] Gaël Guennebaud, Benoît Jacob, et al. *Eigen v3*. <http://eigen.tuxfamily.org>. 2010.
- [10] Alexander Stukowski. “Visualization and analysis of atomistic simulation data with OVITO - the Open Visualization Tool”. In: *Modelling and Simulation in Materials Science and Engineering* 18.1 (2010), 015012 (7pp). URL: <http://stacks.iop.org/0965-0393/18/015012>.

- [11] Søren Toxvaerd and Jeppe C. Dyre. “Communication: Shifted forces in molecular dynamics”. In: *The Journal of Chemical Physics* 134.8 (Feb. 2011), p. 081102. ISSN: 0021-9606. DOI: [10.1063/1.3558787](https://doi.org/10.1063/1.3558787). eprint: https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/1.3558787/14903344/081102/_1/_online.pdf. URL: <https://doi.org/10.1063/1.3558787>.
- [12] Alexander Stukowski, Vasily V Bulatov, and Athanasios Arsenlis. “Automated identification and indexing of dislocations in crystal interfaces”. In: *Modelling and Simulation in Materials Science and Engineering* 20.8 (Oct. 2012), p. 085007. DOI: [10.1088/0965-0393/20/8/085007](https://doi.org/10.1088/0965-0393/20/8/085007). URL: <https://dx.doi.org/10.1088/0965-0393/20/8/085007>.