IMTEK – Department of Microsystems Engineering
University of Freiburg

High-Performance Computing: Molecular Dynamics with C++
11LE50MO-5288 ESE PO 2021

# High-Performance Computing: Molecular Dynamics with C++

**Final Report**

Reetika Rao (5577071)

September 13, 2024

universität freiburg

# Contents

# 1 Introduction

Broadly speaking, all matter that we see, feel or experience is built up of atoms and molecules. At a macroscopic level the interactions and evolution of the objects can be experienced, and these are governed by equations based in physics. However, to fully understand matter we need to experience matter at a microscopic level, this is made easier through molecular dynamics simulation.
Molecular dynamics (MD) simulation is defined as the science of simulating the time dependent behavior of a system of particles. This time evolution of atoms is obtained by integrating their equations of motion with appropriate boundary conditions[2].
The MD simulations have greatly influenced the advancements in domains such as material science, biophysics and pharmaceuticals to name a few.
In this project, an MD simulation is set up with various configurations of gold atoms to demonstrate propagation of atoms, energy conservation and to estimate material properties like melting point, heat capacity, latent heat, stress, strain etc.
This project was completed in the Summer Semester 2024, under the guidance of Professor Dr. Lars Pastewka and the simulation department at Albert-Ludwigs-Universität Freiburg.
The project's source code and related documentation are hosted on GitHub.

# 2 Velocity-Verlet Integration

As previously discussed, a Molecular Dynamics simulation involves simulating and studying at an atomic scale the behaviour and interactions of the involved atoms and molecules.
The atoms and/or molecules are constantly interacting with the neighbouring atoms, thus resulting in a force that acts on atoms and defines the evolution of their trajectory with respect to time. The trajectory of an atom from the current time-step ($t$) to the next time-step ($t + \Delta t$) is characterized by its position and velocity. Euler Integration, Leap-Frog integration, Velocity-Verlet integration, among others, are a few ways in which the velocity and positions can be calculated at the next time-step ($t + \Delta t$).[1]
In this molecular dynamics simulation, the Velocity-Verlet integration method is used.
The Velocity Verlet algorithm was developed in 1982 based on the Verlet integrator by Loup Verlet. It uses the Taylor expansion of position coordinates and gives an expression for r($t + \Delta t$) and v($t + \Delta t$) with an error term of order $\mathcal{O}(t^4)$[3][1]. The expressions are as follows:

$$\vec{r}(t + \Delta t) = \vec{r}(t) + \vec{v}(t)\Delta t + \frac{1}{2}\vec{a}(t)\Delta t^2 = \vec{r}(t) + \vec{v}(t)\Delta t + \frac{1}{2m}\vec{f}(t)\Delta t^2 \tag{1}$$

$$\vec{v}(t + \Delta t) = \vec{v}(t) + \frac{\vec{a}(t) + \vec{a}(t + \Delta t)}{2}\Delta t = \vec{v}(t) + \frac{\vec{f}(t) + \vec{f}(t + \Delta t)}{2m}\Delta t \tag{2}$$

The algorithm is divided into two steps, the *Predictor* step and the *Corrector* step. The Velocity Verlet algorithm's steps for a single particle are discussed below.

## 2.1 Verlet Predictor Step

In the Verlet Predictor step, velocity $\vec{v}(t + \frac{\Delta t}{2})$ is calculated using which the position $\vec{r}(t + \Delta t)$ is calculated.

[1] In the reference, $\vec{r}$ is denoted by $\vec{x}$.

**Algorithm 1** Verlet Predictor step

**procedure** VERLET PREDICTOR STEP($x, y, z, vx, vy, vz, fx, fy, fz, timestep, mass$)
  $vx \leftarrow vx + 0.5 \cdot \frac{fx \cdot timestep}{mass}$
  $vy \leftarrow vy + 0.5 \cdot \frac{fy \cdot timestep}{mass}$
  $vz \leftarrow vz + 0.5 \cdot \frac{fz \cdot timestep}{mass}$
  $x \leftarrow x + vx \cdot timestep$
  $y \leftarrow y + vy \cdot timestep$
  $z \leftarrow z + vz \cdot timestep$
**end procedure**

## 2.2 Verlet Corrector Step

In the Verlet Corrector step, velocity $\vec{v}(t + \Delta t)$ is calculated using the force $\vec{f}(t + \Delta t)$.

**Algorithm 2** Verlet Corrector Step

**procedure** VERLET CORRECTOR STEP($vx, vy, vz, fx, fy, fz, timestep, mass$)
  $vx \leftarrow vx + 0.5 \cdot \frac{fx \cdot timestep}{mass}$
  $vy \leftarrow vy + 0.5 \cdot \frac{fy \cdot timestep}{mass}$
  $vz \leftarrow vz + 0.5 \cdot \frac{fz \cdot timestep}{mass}$
**end procedure**

In this project, the Velocity Verlet's *Predictor* and *Corrector* steps are modified to be compatible with more than one particle.
For an accurate simulation the value of $\Delta t$ must be carefully selected and must be sufficiently small. To ensure that the $\Delta t$ is accurately chosen, one can ensure if the energy of the system is being conserved[4].

## 2.3 Test Strategy for Velocity Verlet Algorithm

The test cases for the Velocity Verlet algorithm (interchangeably referred to as Verlet algorithm) is divided into two parts: the test for a single atom system and the test for multi-atom system.
For a single atom system, the velocities, positions and forces are initialized with a random value. A value for mass, time-step and number of steps are chosen. Maintaining a constant force, the system is propagated with the Velocity Verlet algorithm.
Finally, the values of position and velocity obtained by the Velocity Verlet algorithm are compared to an analytical value obtained using Newton's equations of motion.

**Algorithm 3** Random Values Test for Single Atom Verlet Integration

**procedure** RANDOMVALTEST
  Initialize variables $x, y, z, vx, vy, vz, fx, fy, fz$ with random values
  Set $timestep \leftarrow 1e-15$, $mass \leftarrow 3.27$
  $nb\_steps \leftarrow 1000$
  Calculate analytical values $r\_analytical$, $v\_analytical$ using Newton's equations of motion
  Perform Verlet integration with `single_atom_verlet` for nb_steps number of steps
  Calculate $r\_verlet \leftarrow \sqrt{x^2 + y^2 + z^2}$
  Calculate $v\_verlet \leftarrow \sqrt{vx^2 + vy^2 + vz^2}$
  Compare $r\_verlet$ with $r\_analytical$ ▷ Expect near equality
  Compare $v\_verlet$ with $v\_analytical$ ▷ Expect near equality
**end procedure**

For the multi-atom system, all atoms are considered to be identical. A time-step, number of steps and mass values are chosen. An atom's position, velocity and force are randomly chosen, and these values are

then copied onto the multi-atom system. The single atom is propagated using the *single_atom_verlet* and the multi-atom system is propagated using the *multi_atom_verlet*.
The values of positions and velocities from the multi-atom system are compared with the single atom system.

**Algorithm 4** Multi-Atom Test for Verlet Integration

**procedure** MULTIATOMTEST
    Initialize $pos, vel, f$ arrays with random values for one atom
    Set $timestep \leftarrow 1e-15$, $mass \leftarrow 3.27$
    $nb\_steps \leftarrow 1000$, $nb\_atoms \leftarrow 4$
    Initialize an atoms object with $positions, velocities, forces$ arrays for $nb\_atoms$
    Assign single atom's $pos, vel, f$ to $atoms.positions, atoms.velocities, atoms.forces$
    Perform Verlet integration on single atom using `single_atom_verlet`
    Perform Verlet integration on multiple atoms using `multi_atom_verlet`
    Compare final *velocities* with propagated *vel* ▷ Expect near equality
    Compare final *positions* with propagated *pos* ▷ Expect near equality
**end procedure**

# 3 Time-step and Energy Conservation

A fundamental aspect of any closed system is the conservation of total energy. This holds true even for an MD simulation. That is, if no energy is externally supplied or removed, the total energy of the system must be conserved for the MD system. Conservation of total energy is thus one of the indications of a good or accurate simulation.

## 3.1 Results: Demonstration of Energy Conservation

The total energy conservation and the time-step chosen for the simulation are inter-related, as shown in Figure 1. Results were obtained by simulating 54 gold (Au) atoms for 100 LJ-time units with m = 1, $\sigma = 1$ and $\epsilon = 1$ in LJ-units.

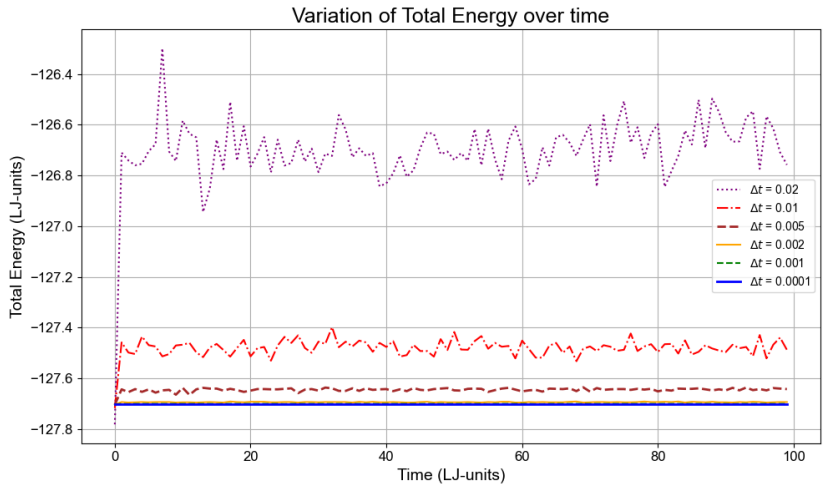


Figure 1: Total Energy vs Time for different timesteps.

However, the smaller the time-step, greater is the computational power required and therefore a balance

needs to be achieved. The time-step chosen must be lower than and close to the least value of the time-step at which the energy of the closed MD system is still conserved.
A good estimate is about $1/10^{th}$ the order of the characteristic period of the MD system[7], more specifically, it must accommodate the fastest movement in the system, i.e, vibrational frequency of the bond ($\nu$),

$$\nu = \frac{1}{2\pi}\sqrt{\frac{k}{\mu}} \tag{3}$$

where $k$ is the force constant and $\mu$ is the reduced mass. Then the time-step $\Delta t$,

$$\Delta t \leq \frac{1}{10}.\frac{2\pi}{\sqrt{\frac{k}{\mu}}} \tag{4}$$

As Lennard-Jones sytems use reduced units[2], a reasonable time-step to use is 0.001 $\tau_{LJ}$, given by,

$$\tau_{\mathrm{LJ}} = \sqrt{\frac{m\sigma^2}{\epsilon}} \tag{5}$$

where $m$ is the mass, $\sigma$ is the characteristic length and $\epsilon$ is the characteristic energy.
In the following Figures 2 and 3, the energy conservation is demonstrated for 54 gold atoms, closer to equilibrium and further away from equilibrium. In both cases, the energy is conserved and the time-step chosen is $\Delta t = 0.001$, with m = 1, $\sigma = 1$ and $\epsilon = 1$ in LJ-units.

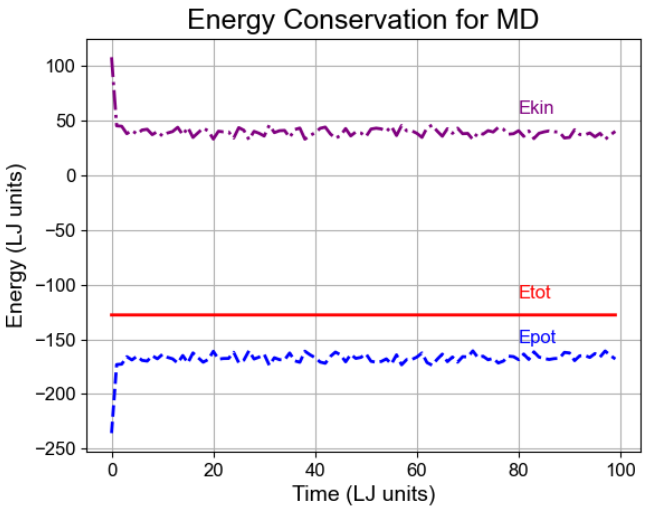


Figure 2: Energy conservation of Au atoms near equilibrium

[2]discussed in section4.6

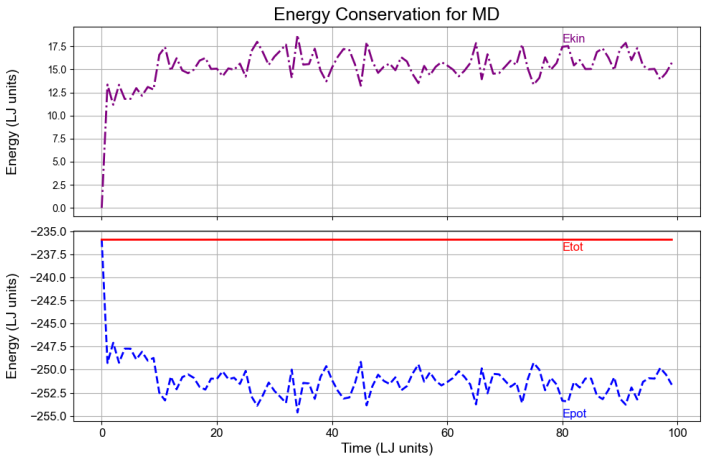


Figure 3: Energy conservation of Au atoms further from equilibrium

# 4 Lennard-Jones Potential and Pair Forces

## 4.1 Lennard-Jones Potential

Introduced in 1924 by John Lennard-Jones, the Lennard-Jones potential (LJ-Potential) mathematically models two important aspects of interactions between a pair of atoms: short-range repulsion and van der Waal's (long-range) attraction[5].
The short-ranged strong repulsion is attributed to the overlapping of electron orbitals described by the Pauli's exclusion principle. The repulsive forces are a function of the distance between the atoms and are considered to vary proportional to $(\frac{1}{r^{12}})$[2].
The weaker attractive van der Waal's forces are also a function of the distance between atoms and arise due to the induced dipole-dipole interactions. The attractive forces vary proportional to $(\frac{1}{r^6})$[2].

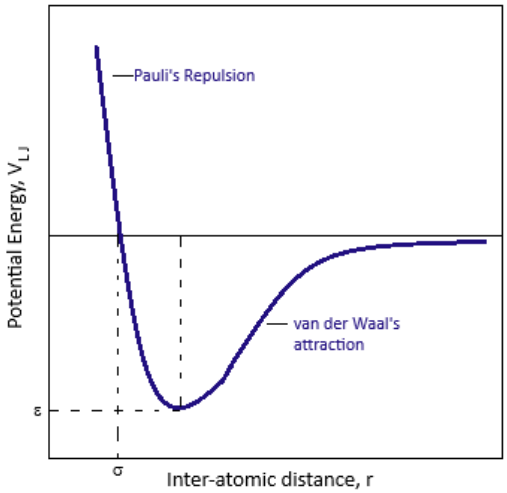


Figure 4: Expected curve of the Lennard-Jones potential as a function of the inter-atomic distance $r$.

The Lennard-Jones Potential is given by[4]:

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

The total potential energy of the system is given by the superposition of individual potential energies summed once over all pairs.
The Figure 4 illustrates the expected curve of the LJ-Potential between two atoms.
The potential energy $V_{LJ}$ varies with the inter-atomic distance $r$. As shown in the graph, the potential energy decreases sharply at shorter distances and approaches zero at larger distances, displaying a characteristic potential well.

## 4.2 LJ-Potential with cutoff distance

Since an MD simulation can quickly become computationally expensive, especially if more number of particles are taken into consideration, it is beneficial to limit the number of operations wherever possible. One of the ways of achieving this is by introducing a cutoff distance ($r_c$).
As at larger distances the potential energy $V_{LJ}$ approaches zero, the interaction between atoms becomes negligible and can be disregarded. The Lennard-Jones Potential[4] is then modified as:

$$V_{LJ}(r) = \begin{cases} 4\epsilon \left[ \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right] & \text{if } r \leq r_c \\ 0 & \text{if } r > r_c \end{cases} \tag{7}$$

## 4.3 Results: Execution time with and without $R_c$

The Figure 5 depicts the execution times for equilibrating different number of atoms implemented with and without cutoff distance.
These graphs were obtained for MD systems of varying number of atoms from 5 to 150, with $\sigma = 1$ and $\epsilon = 1$ in LJ-units. The MD systems were equilibrated at $temperature = 0.3$ for 100,000 steps and the energy conservation was confirmed for another 100,000 steps at $timestep = 0.001$. The cutoff radius of 5.0 was chosen.
With lesser number of atoms, it can be seen that the run-times with and without $r_c$ are almost similar, and the algorithm with $r_c$ performs better as the number of atoms increase.
Interestingly, for lesser number of atoms the algorithm with $r_c$ may also be slightly less efficient due to the overhead of computing the *Neighbour Lists*.

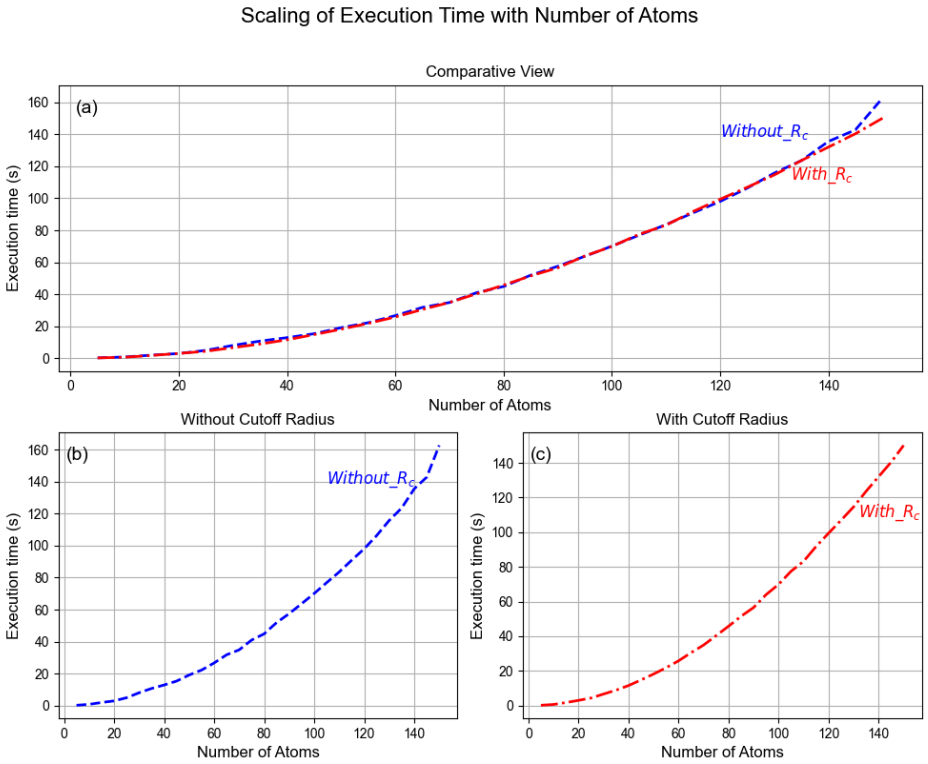


Figure 5: Execution time for various configurations of gold atoms. (a) Comparison of algorithm runtimes between with and without cutoff radius; (b) Execution times without $r_c$; (c) Impact of $r_c$ on execution time

## 4.4 Pair Force

The LJ-potential describes the potential energy between two atoms. To propagate an MD system, the force on each atom needs to be calculated.
The force acting between any two atoms, $i$ and $j$, where $i, j \in \mathrm{N_{atoms}}$, is described as follows[6][3].:

$$f_{ij} = -\nabla V_{LJ}(r_{ij}) \tag{8}$$

From equation 6,

$$f_{ij} = -\frac{d}{dr}\left(4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]\right) \tag{9}$$

$$= 24\epsilon\left(2\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right)\frac{r_{ij}}{r^2} \tag{10}$$

An important point to note here is that by Newton's third law we get $f_{ij}$ = - $f_{ji}$.
The total force acting on a single atom is given by the summation of all the forces acting on it[6]. That is,

$$f_i = \sum_{j=1, j\neq i}^{N_{\mathrm{atoms}}} f_{ij} \tag{11}$$

[3]Notation changed from reference for consistency

## 4.5 Algorithm for Total Potential Energy and Total Force on atom

The force on each atom at time $(t + \Delta t)$ is calculated before the Verlet Corrector Step as discussed in section 2.2, using one of the following algorithms.

### 4.5.1 Without cutoff radius

**Algorithm 5** Lennard-Jones Direct Summation

**procedure** LJ_DIRECT_SUMMATION(Atoms &atoms, double epsilon = 1.0, double sigma = 1.0)
  $E_{\text{pot}} \leftarrow 0$
  atoms.forces.setZero()
  **for** $i \leftarrow 0$ **to** atoms.nb_atoms() - 1 **do**
    $\mathbf{f_{ij}} \leftarrow \mathbf{0}$ ▷ Initialize force fij
    **for** $j \leftarrow$ atoms.nb_atoms() $- 1$ **to** $i + 1$ **do**
      $\mathbf{r_{ij}} \leftarrow$ atoms.positions.col($i$) $-$ atoms.positions.col($j$)
      $r \leftarrow \|\mathbf{r_{ij}}\|$ ▷ Magnitude of $\mathbf{r_{ij}}$
      $E_{\text{pot}} \leftarrow E_{\text{pot}} + \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}$
      $\mathbf{f_{ij}} \leftarrow 24 \cdot \epsilon \cdot \frac{\mathbf{r_{ij}}}{r^2} \cdot \left(2 \cdot \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right)$
      atoms.forces.col($i$) $\leftarrow$ atoms.forces.col($i$) + $\mathbf{f_{ij}}$
      atoms.forces.col($j$) $\leftarrow$ atoms.forces.col($j$) - $\mathbf{f_{ij}}$ ▷ Because $\mathbf{f_{ji}}$ is in the opposite direction
    **end for**
  **end for**
  $E_{\text{pot}} \leftarrow 4 \cdot \epsilon \cdot E_{\text{pot}}$
  **return** $E_{\text{pot}}$
**end procedure**

The algorithm was tested for correctness against the test provided by Professor Dr. Lars Pastewka, Albert-Ludwigs-Universität Freiburg.
The test compares the result of the force computed using the *lj_direct_summation* function to the forces calculated using finite differences by changing the positions of the atoms with $\delta = 0.0001$.

### 4.5.2 With cutoff radius

The function for evaluating LJ-potential and Pair force using a cutoff distance $r_c$ requires computation of neighbor lists.
The function and test for computing the neighbor list is also provided as a part of the course material. The computation of neighbor list involves:

- Identifying neighboring atoms based on cutoff distance.
- Calculating the bounding box and grid size.
- Sorting the atoms into the right boxes.

**Algorithm 6** Lennard-Jones Direct Summation with Cutoff Radius

**procedure** LJ_DIRECT_SUMMATION_RC(Atoms &atoms, double rc, double epsilon = 1.0, double sigma = 1.0)
  $E_{\text{pot}} \leftarrow 0$
  atoms.forces.setZero()
  **Initialize** NeighborList neighbor_list
  neighbor_list.update(atoms, rc)
  $E_{\text{rc}} \leftarrow \left(\frac{\sigma}{rc}\right)^{12} - \left(\frac{\sigma}{rc}\right)^{6}$ ▷ for Shifting the Potential due to cutoff
  **for** each $(i, j)$ in neighbor_list **do**
    **if** $i < j$ **then**
      $\mathbf{r_{ij}} \leftarrow$ atoms.positions.col($i$) − atoms.positions.col($j$)
      $r \leftarrow \|\mathbf{r_{ij}}\|$
      $E_{\text{pot}} \leftarrow E_{\text{pot}} + \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] - E_{\text{rc}}$
      $\mathbf{f_{ij}} \leftarrow 24 \cdot \epsilon \cdot \frac{\mathbf{r_{ij}}}{r^2} \cdot \left(2 \cdot \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right)$
      atoms.forces.col($i$) ← atoms.forces.col($i$) + $\mathbf{f_{ij}}$
      atoms.forces.col($j$) ← atoms.forces.col($j$) - $\mathbf{f_{ij}}$
    **end if**
  **end for**
  $E_{\text{pot}} \leftarrow 4 \cdot \epsilon \cdot E_{\text{pot}}$
  **return** $E_{\text{pot}}$
**end procedure**

## 4.6 Lennard-Jones Reduced Units

In MD systems with only one type of particle, such as this project with Au atoms, it is generally beneficial to set the mass of all particles as unity. Similarly, other fundamental parameters for Lennard-Jones potential such as the length parameter ($\sigma$) and energy parameter ($\epsilon$) can be set as unity. The units of all other quantities are derived from these three parameters[8]. Since we deal with dimensionless quantities, setting the mass, $\sigma$ and $\epsilon$ to unity increases comprehensibility, while maintaining and modeling all important physical properties in reduced units. Normalization of units also ensures scalability and to an extent generality.

Using reduced units can potentially be computationally better, especially because the resulting quantities are comparable to the order of 1, thereby increasing numeric stability and reducing computations with very large or very small values. As certain quantities are unity, reduced units mean simplified equations.

However, in mixed particle systems or complicated potentials with many adjustable parameters, the computational advantage of using reduced units is limited[8].

The Table 1 enables the conversion from reduced LJ-units to SI units or other preferred unit systems[8][4].

[4]In the reference conversion from physical units to reduced units is given.

| Quantity | Conversion Formula |
|---|---|
| Length (L) | $L = L_{\mathrm{LJ}} \times \sigma'$ |
| Energy (E) | $E = E_{\mathrm{LJ}} \times \epsilon'$ |
| Temperature (T) | $T = T_{\mathrm{LJ}} \times \frac{\epsilon'}{k_B}$ |
| Time (t) | $t = t_{\mathrm{LJ}} \times \sqrt{\frac{m'\sigma'^2}{\epsilon'}}$ |
| Velocity (v) | $v = v_{\mathrm{LJ}} \times \sqrt{\frac{\epsilon'}{m'\sigma'^2}}$ |
| Force (F) | $F = F_{\mathrm{LJ}} \times \frac{\epsilon'}{\sigma'}$ |

Table 1: Conversion from Lennard-Jones reduced units to physical units, where $\epsilon'$, $m'$ and $\sigma'$ are non normalized values of $\epsilon$, $m$ and $\sigma$

### 4.7 Results: MD simulation of 54 Au atoms

The Figure 6 shows snapshots from an initial Molecular Dynamics Simulation setup. The MD simulation is run for 100 units of time, using $\Delta t = 0.001, m = 1$, $\sigma = 1$ and $\epsilon = 1$ in LJ-units. The simulation was performed without a cut-off radius. The configuration file containing the positions and velocities for 54 equilibrated Au atoms was provided in the course material.

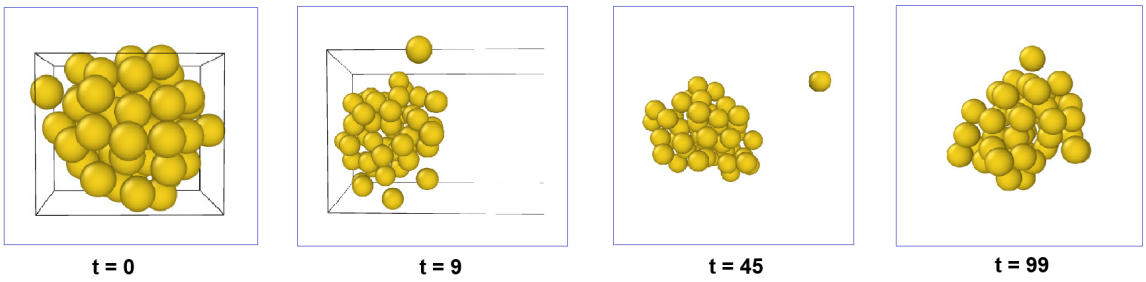


Figure 6: Trajectory of 54 gold atoms at different time stamps

It can be noticed that some of the atoms evaporate, this is due to the system attaining equilibrium at an average temperature of 0.5 in LJ units.

## 5 Thermostats and Thermal Equilibrium

Many a time, one would like to carry out MD simulations that require the system to be at a certain temperature, or one may need to adjust the temperature of the system from time to time. This can be achieved with the use of thermostats.
A thermostat models a large heat bath for the MD system that is being simulated[1]. Since the coupled heat bath is considered to be large enough that its temperature fluctuation is minimal, the heat bath acts as a heat source and heat sink for the MD system.
A thermostat can be local or global. A local thermostat uses stochastic methods to rescale temperatures of only particles that are *'too hot'* or *'too cold'*, while a global thermostat uses the global temperature feedback to control the temperatures of all particles uniformly[9].

## 5.1 Velocity Rescaling

Temperature (T) is directly related to the kinetic energy ($E_{kin}$) of the system by,

$$E_{kin} = \frac{3}{2} N k_B T \tag{12}$$

where, N is the number of particles and $k_B$ is the Boltzmann constant.
The kinetic energy of the system is also given by,

$$E_{kin} = \frac{1}{2} \sum_i m_i v_i^2 \tag{13}$$

It can therefore be concluded that the temperature (T) of the system is directly related to the velocities of particles in the system. Thus, scaling the velocities scales the temperature.
The velocity rescaling thermostat is a global thermostat that uses this principle, wherein all velocities in the system are scaled by a rescale factor ($\lambda$) to control the temperature at a time t. The rescale factor is given by[9][5],

$$\lambda = \sqrt{\frac{T_0}{T(t)}} \tag{14}$$

where, $T_0$ is the required temperature.
However, the velocity rescaling thermostat does not allow for the system to relax and the kinetic energy does not fluctuate[9]. As this is not a desired behaviour, this project uses the *Berendsen Thermostat*.

## 5.2 Berendsen Thermostat

The Berendsen thermostat is also a global thermostat that uses the same principle as velocity rescaling. However, the rescale factor takes into account the coupling between the heat bath and MD system. It also factors in the temperature difference between them.
The rescale factor for the Berendsen thermostat is given by[9][6],

$$\lambda = \sqrt{1 + \frac{\Delta t}{\tau} \left( \frac{T_0}{T(t)} - 1 \right)} \tag{15}$$

where, $T_0$ is the required temperature, $\Delta t$ is the time-step and $\tau$ is coupling factor known as relaxation time.
The relaxation time chosen must be large enough to allow the system to relax and must be small enough such that the required temperature is maintained. Generally, a good estimate for $\tau$ is 100 to 200 times the time-step $\Delta t$.

## 5.3 Equilibrating a MD system

This project uses a Berendsen thermostat to equilibrate the MD system, given below is the algorithm.

[5] $T_0$ is denoted by $T_{des}$ in reference
[6] $T_0$ is denoted by $T_{bath}$ in reference

**Algorithm 7** Equilibrating MD system with Berendsen Thermostat

**procedure** MD_SIMULATION_WITH_BERENDSEN(int nb_atoms, double target_temp)
    **Initialize** system with *nb_atoms* number of atoms ▷ system initialized as a simple cubic lattice
    Set simulation length $t_{\text{sim}}$ and time step $\Delta t$
    Set mass $m$, Boltzmann constant $k_B$
    Calculate initial potential energy $E_{\text{pot}}$ and atoms.forces
    **Part 1: Equilibration with Berendsen Thermostat**
    **for** $i = 1$ **to** $t_{sim}$ **do**
        Perform Verlet predictor step
        Calculate potential energy $E_{\text{pot}}$ and atoms.forces
        Perform Verlet corrector step
        Calculate kinetic energy $E_{\text{kin}}$
        Compute temperature $T_{\text{old}} = 2E_{\text{kin}}/3Nk_B$
        **Apply Berendsen Thermostat**
            **Arguments:** &atoms, $T_{\text{old}}$, $\Delta t$, $T_{target}$, $T_{relaxation}$
            Compute rescale factor: $\lambda = \sqrt{1 + \frac{\Delta t}{T_{relaxation}} \left( \frac{T_{target}}{T_{old}} - 1 \right)}$
            Scale velocities: **atoms.velocities** $= \lambda *$ **atoms.velocities**
        Calculate new kinetic energy $E_{\text{kin}}$
        Compute new temperature $T = 2E_{\text{kin}}/3Nk_B$
    **end for**
    **Part 2: Check Energy Conservation**
    Perform steps 7 through 11 and verify that the total energy is conserved and the temperature of the system is close to target temperature
**end procedure**

## 5.4 Results: Equilibration of MD System using Berendsen Thermostat

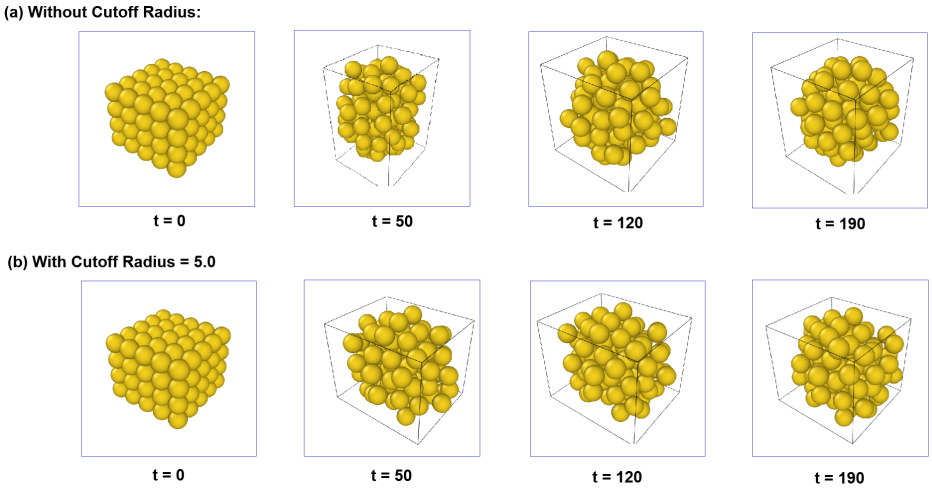


Figure 7: (a) Trajectory of 100 atoms using Berendsen Thermostat without $r_c$ ; (b) Trajectory of 100 atoms using Berendsen Thermostat with $r_c = 5.0$

Figure 7 shows snapshots from a Molecular Dynamics Simulation setup of 100 atoms. The MD simulation is equilibrated for 100 units of time and then run for another 100 units of time, using

$\Delta t = 0.001, m = 1$, $\sigma = 1$ and $\epsilon = 1$ in LJ-units. The MD system was equilibrated at 0.2 with a $\tau = 100.\Delta t$.
The simulation was performed twice: without a cut-off radius and with $r_c = 5.0$.

## 5.5 Test Strategy for Berendsen Thermostat

The test is executed in two parts. The first step involves testing the Berendsen thermostat's ability to immediately rescale the temperature to a target value when the relaxation time is set equal to the time-step. The new temperature should be nearly equal to the target temperature. This ensures that the Berendsen thermostat is able to rescale to desired temperature.

**Algorithm 8** Immediate Rescale Test

**procedure** IMMEDIATE_RESCALE_TEST
  **Initialize** $nb\_atoms = 10$, $target\_temp = 0.3$, $timestep = 0.001$, $k_b \leftarrow 1.0$, $mass \leftarrow 1.0$
  $relax\_time \leftarrow timestep$ ▷ For Immediate rescaling
  **Initialize** Atoms $atoms(nb\_atoms)$
  atoms.positions ← Random(), atoms.velocities ← Random()
  $E_{kin} \leftarrow$ **Calculate_Kinetic_Energy**(atoms, mass)
  $T_{old} \leftarrow 2 \cdot E_{kin}/(3 \cdot nb\_atoms \cdot k_b)$
  **Apply Berendsen Thermostat**(atoms, $T_{old}$, timestep, relax_time, target_temp)
  $E_{kin} \leftarrow$ **Calculate_Kinetic_Energy**(atoms, mass)
  $T_{new} \leftarrow 2 \cdot E_{kin}/(3 \cdot nb\_atoms \cdot k_b)$
  Compare $T_{new}$ with $target\_temp$ ▷ Expect Near Equality
**end procedure**

**Algorithm 9** Rescale Factor Test

**procedure** RESCALE_FACTOR_TEST
  **Initialize** $nb\_atoms = 10$, $target\_temp = 0.3$, $timestep = 0.001$, $kb \leftarrow 1.0$, $mass \leftarrow 1.0$
  $relax\_time \leftarrow 100 \times timestep$ ▷ Good Relaxation Time
  **Test Case 1:** $T_{old} > target\_temp$
  **Initialize** Atoms $atoms1(nb\_atoms)$
  atoms1.positions ← Random()
  atoms1.velocities ← **Constant**(0.8)
  $E_{kin} \leftarrow$ **Calculate_Kinetic_Energy**(atoms1, mass)
  $T_{old} \leftarrow 2 \cdot E_{kin}/(3 \cdot nb\_atoms \cdot k_b)$
  $rescale\_factor \leftarrow$ **Apply Berendsen Thermostat**(atoms1, $T_{old}$, timestep, relax_time, target_temp)
  **Expect:** $rescale\_factor < 1$
  **Test Case 2:** $T_{old} < target\_temp$
  **Initialize** Atoms $atoms2(nb\_atoms)$
  atoms2.positions ← Random()
  atoms2.velocities ← **Constant**(0.2)
  $E_{kin} \leftarrow$ **Calculate_Kinetic_Energy**(atoms2, mass)
  $T_{old} \leftarrow 2 \cdot E_{kin}/(3 \cdot nb\_atoms \cdot k_b)$
  $rescale\_factor \leftarrow$ **Apply Berendsen Thermostat**(atoms2, $T_{old}$, timestep, relax_time, target_temp)
  **Expect:** $rescale\_factor > 1$
**end procedure**

The second part involves testing the Berendsen thermostat's ability to adjust the rescale factor $\lambda$ based on the current temperature (T_old). The test ensures that the rescale factor $\lambda$ is less than 1 when

T_old is greater than the target temperature and $\lambda$ is greater than 1 when T_old is less than the target temperature.

# 6 Embedded Atom Method (EAM) Potential

Thus far, all simulations were carried out with the LJ-Potential. However, LJ-potential models only the pair potential between two atoms (usually a neutral atom) without considering the effects of other surrounding atoms. The simplicity of the LJ-potential makes it ideal for simulating the behaviour of non-bonded interactions, e.g inert gases or simple liquids, as the pair-potential based interactions are more prominent in these.
The Embedded Atom Potential is a *many-body potential* that was introduced by Daw and Baskes as an alternative to the pair potential[10]. The EAM potential considers complex interactions between an atom and the surrounding electron density created by its neighboring atoms, which reflects the collective behavior of electrons. It also separately models the pair-wise interactions.
The potential energy of an atom is given by,

$$E_{\text{pot}}(\vec{r}_i) = F_i(\bar{\rho}_i) + \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^{N} u(r_{ij}) \tag{16}$$

Then the total potential energy of the atom is given by[10],

$$E_{\text{pot}} = \sum_{i=1}^{N}\left[F_i(\bar{\rho}_i) + \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^{N} u(r_{ij})\right] \tag{17}$$

where $F_i(\bar{\rho}_i)$ is the energy required to locate atom $i$ into the background electron density $\bar{\rho}_i$ at position $i$; $u(r_{ij})$ is the pair interaction potential between atom $i$ and atom $j$ having the distance $r_{ij}$ [10].
The EAM potential is more suited for metals where the behavior of an atom is significantly influenced by the collective effect of surrounding atoms.

## 6.1 Gupta-Ducastelle Potential

The Gupta-Ducastelle potential is an example of the EAM potential and has been used in this project for estimating physical properties of gold.
The Gupta-Ducastelle potential models in detail the atomic interactions in metallic systems. The potential energy given by Gupta-Ducastelle includes two parts:

1. The repulsive interaction between atoms $i$ and $j$ described by,

$$V_{ij}^{\text{rep}} = 2A\exp\left(-p\left(\frac{r_{ij}}{r_e} - 1\right)\right) \tag{18}$$

   where $A$ is a parameter that determines the strength of the repulsion, $p$ is a parameter that controls the range of the repulsive interaction, $r_{ij}$ is the distance between atoms $i$ and $j$, $r_e$ is a reference distance.

2. The many-body embedding energy for atom $i$ which is calculated from its neighbors by,

$$\rho_i = \sum_{j\neq i}\xi^2\exp\left(-2q\left(\frac{r_{ij}}{r_e} - 1\right)\right) \tag{19}$$

   where $\xi$ is a parameter related to the strength of the many-body interaction, $q$ is a parameter controlling the range of the many-body interaction.

Then the total potential energy contribution for each atom $i$ is given by,

$$E_{pot_i} = \frac{1}{2} \sum_{j \neq i} V_{ij}^{\text{rep}} - \sqrt{\rho_i} \tag{20}$$

where values of all parameters are optimized for specific materials and are generally experimentally obtained. In this project, the parameters used are Cleri & Rosato's parameterization for gold[13].

## 6.2 Simulation of Au Clusters with EAM potential

This following MD simulation is setup to estimate thermodynamic properties of a gold cluster. The clusters used in this simulation are Mackay icosahedrons[11], chosen for their defect-free surfaces that consist of perfect atomic planes with a (111) crystal orientation. The simulation was repeated for varying cluster sizes and these clusters were generated using the *Mackay Icosahedron Structure Generator* by Yanting Wang of University of Rochester .

### 6.2.1 Fixing Simulation Units

As EAM potential uses actual physical units, these parameters need to be defined. In this project the units of energy, length and time are fixed as *[E] = eV, [l] = Å and [t] = fs*. Then the units for force and mass are derived as *[F] = eV/Å and [m] = 0.009649 g/mol*. The relation between quantities is as follows:

$$[F] = [E]/[l]$$
$$[t] = [l]\sqrt{[m]/[E]}$$

### 6.2.2 Fixing Time-step

As discussed in the equation 4, the time-step can be estimated as follows:

$$\Delta t \leq \frac{1}{10} . \frac{2\pi}{\sqrt{\frac{k}{\mu}}}$$

Here, $k \approx 20$Å and $\mu = 1.635 \times 10^{-25}$

$$\Delta t \leq \frac{1}{10} \cdot \frac{2\pi}{\sqrt{\frac{1.635 \times 10^{-25}\,\text{kg}}{20\,\text{N/m}}}}$$

therefore, $\Delta <= 50fs$. The time-step chosen is 5 fs for best results

### 6.2.3 Algorithm for estimating thermodynamic properties

The algorithm simulates the melting process of a gold atom cluster. The atom cluster is propagated using the Velocity Verlet method and the potential energy and forced are calculated using the Gupta-Ducastelle potential. During the simulation, velocities are rescaled and randomized slightly to input energy into the system. The average energy and temperature over fixed periods are recorded for analysis. The function and test for Gupta-Ducastelle potential was provided in the lecture material.

**Algorithm 10** Gold Melting Point Simulation

**procedure** GOLD_MELTING_POINT
  **Initialize** parameters $A$, $\xi$, $p$, $q$, $r_e$, $r_c$, timestep, $nb\_steps$, $k_B$, mass
  **Initialize** atom cluster
  Compute initial potential energy $E_{\text{pot}}$ and kinetic energy $E_{\text{kin}}$
  Calculate initial temperature $T$
  **for** $i = 1$ **to** $nb\_steps$ **do**
    Perform Verlet predictor step
    Update neighbor list and calculate $E_{\text{pot}}$
    Perform Verlet corrector step
    Calculate $E_{\text{kin}}$ and temperature $T$
    Compute total energy $E_{\text{tot}} = E_{\text{pot}} + E_{\text{kin}}$
    **if** relax_time == RELAX_TIME **then**
      input energy/temperature using using **velocity_rescale_energy**
    **end if**
    **if** relax_time $\leq$ AVERAGING_WINDOW **then**
      Accumulate $avg\_T$ and $avg\_E$
    **end if**
    Decrease relax_time by 1
    **if** relax_time == 0 **then**
      Save average energy and temperature
      Reset $avg\_T$ and $avg\_E$
      Reset relax_time to RELAX_TIME
    **end if**
  **end for**
**end procedure**

## 6.3 Results: Thermodynamic properties of Au

### 6.3.1 Melting Point

The melting point of a substance is defined as the temperature at which its state changes from solid to liquid, usually at atmospheric pressure.
The Figure 8, shows a cluster of gold with 923 atoms at 0K and at 800K. The simulation parameters used were : $A = 0.2061\,\text{eV}$, $\xi = 1.790\,\text{eV}$, $p = 10.229$, $q = 4.036$, $r_e = \frac{4.079}{\sqrt{2}}$ Å, $r_c = 5.0$ Å, time-step $= 5\,\text{fs}$, $k_B = 8.617 \times 10^{-5}\,\text{eV/K}$, mass $= 196.967 \times 103.6\,\text{g/mol}$.

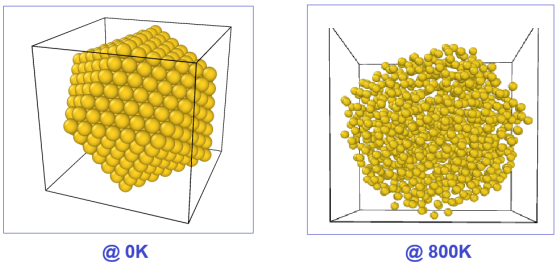


Figure 8: Gold cluster of 923 atoms at 0K (solid state) and at 800K (molten state)

The Figure 9 shows the variation in total energy of the cluster of 923 atoms with increase in temperature. On this graph, both the melting point and latent heat are also indicated.

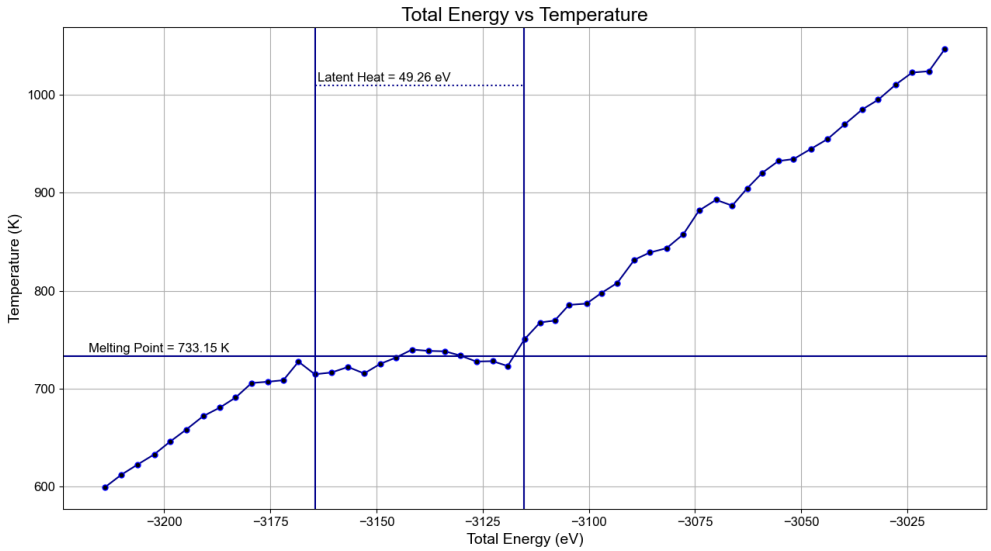


Figure 9: Variation of total energy of the cluster(923 atoms) with increase in temperature, with melting point and latent heat.

The variation of melting point with cluster size is depicted in Figure 10.

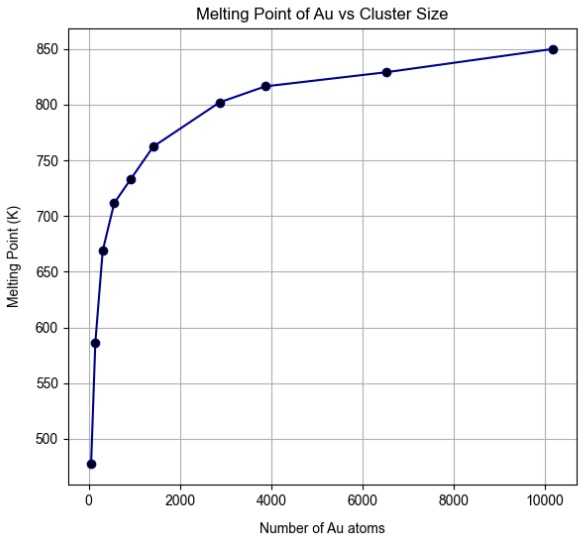


Figure 10: Dependence of cluster-size on melting point

Melting point is an *intensive* property, meaning that it is inherent to the material itself and does not depend on the quantity of the material. However, in this simulation, it is observed that the melting point increases with the cluster size. This is due to finite-size effects.
In smaller clusters, a significant proportion of atoms reside on the surface, where they are less tightly

bound compared to atoms in the bulk, leading to a lower melting point. As the cluster-size grows the effect of surface atoms reduces and melting point approaches the melting point of the bulk. A similar result can also be seen in the study by Gomez & Rincon[12].
It is also worth noting that the accuracy of the melting point depends on the interatomic potential used in the simulation. The parameters of the Gupta-Ducastelle potential are not optimized for thermodynamic properties and therefore we get a melting point of about 850 K instead of the typical value of 1337.33 K for gold.

### 6.3.2 Latent Heat

The latent heat is the amount of energy required to change the phase of a given amount of substance, in this case, from solid to liquid.
Unlike melting point, latent heat is an *extensive* property. An extensive property is dependent on the quantity of the material rather than just the material itself. That is, the value of an extensive property changes proportionally with the quantity of material in the system.
The Figure 11 shows the latent heat steadily increasing with respect to cluster size. In contrast, the intensive property *latent heat per atom* remains almost constant with cluster size.

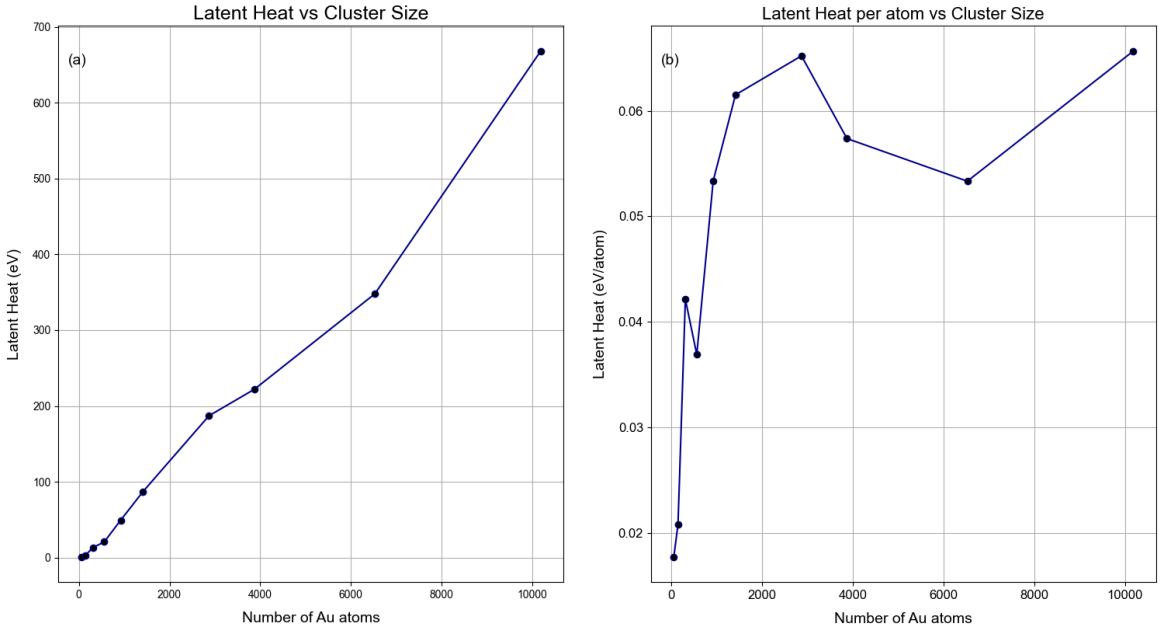


Figure 11: (a) Latent heat vs Cluster-size; (b) Latent heat per atom vs Cluster-size

### 6.3.3 Heat Capacity

The Heat capacity of a substance is the amount of heat energy required to raise the temperature by one degree. The Figure 12 shows the caloric curve and corresponding heat capacity curve with respect to temperature.

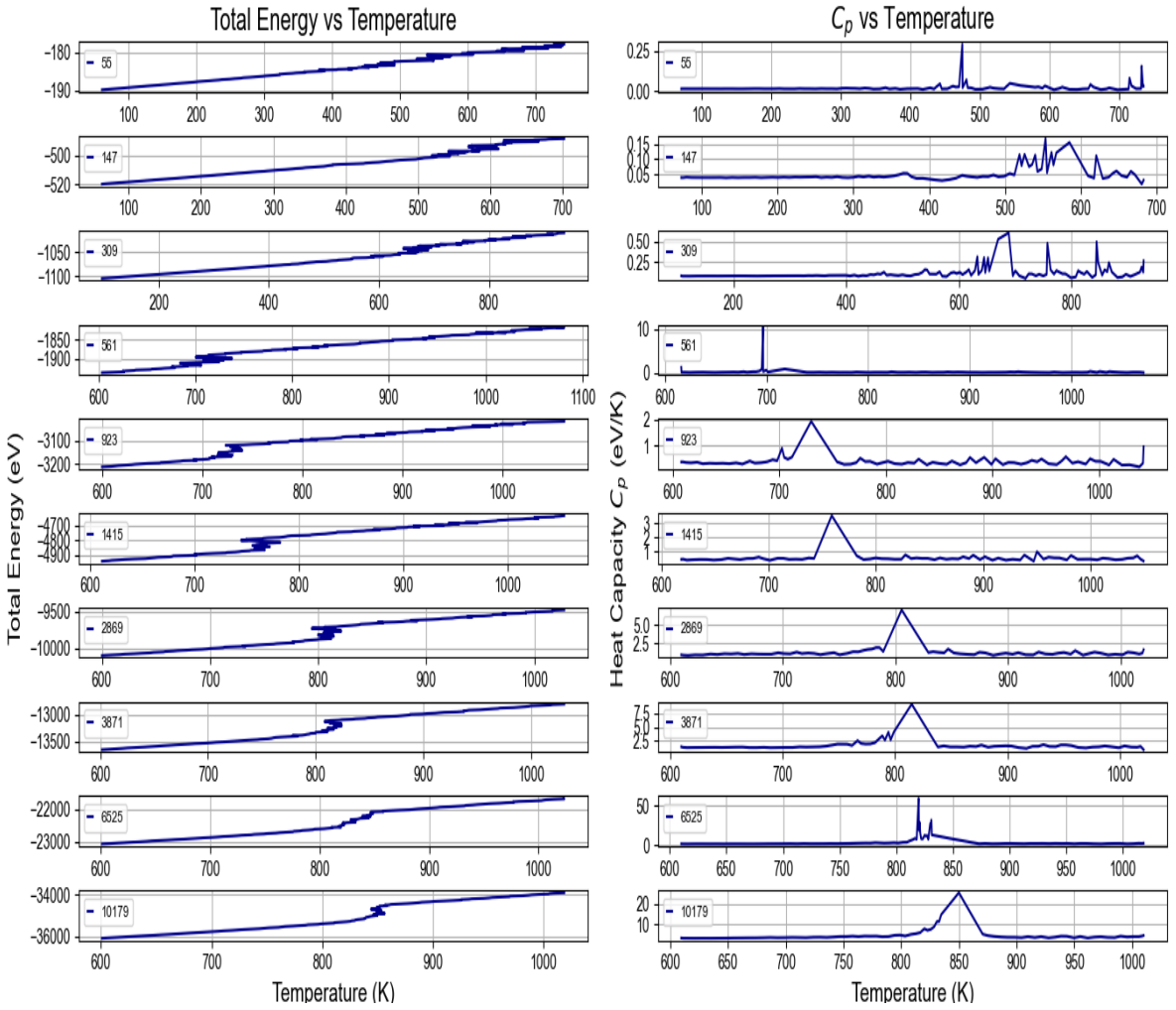


Figure 12: (a) Caloric Curve vs Temperature ; (b) Corresponding heat capacity curve

It can be observed that for smaller clusters, the melting transitions are not sharply defined[12], mainly due to finite-size effects. Around the melting temperature, a sharp increase in the heat capacity curves is also observed, which is attributed to the absorption of latent heat, i.e, more energy is required to increase the temperature of the cluster.
Heat capacity is an extensive property. The Figure 13 shows the increasing heat capacity measured at 600K with respect to cluster-size. In contrast, the specific heat, which is the amount of heat energy required to raise the temperature of a single atom in a substance by one degree, remains somewhat constant. The specific heat is an intensive property.

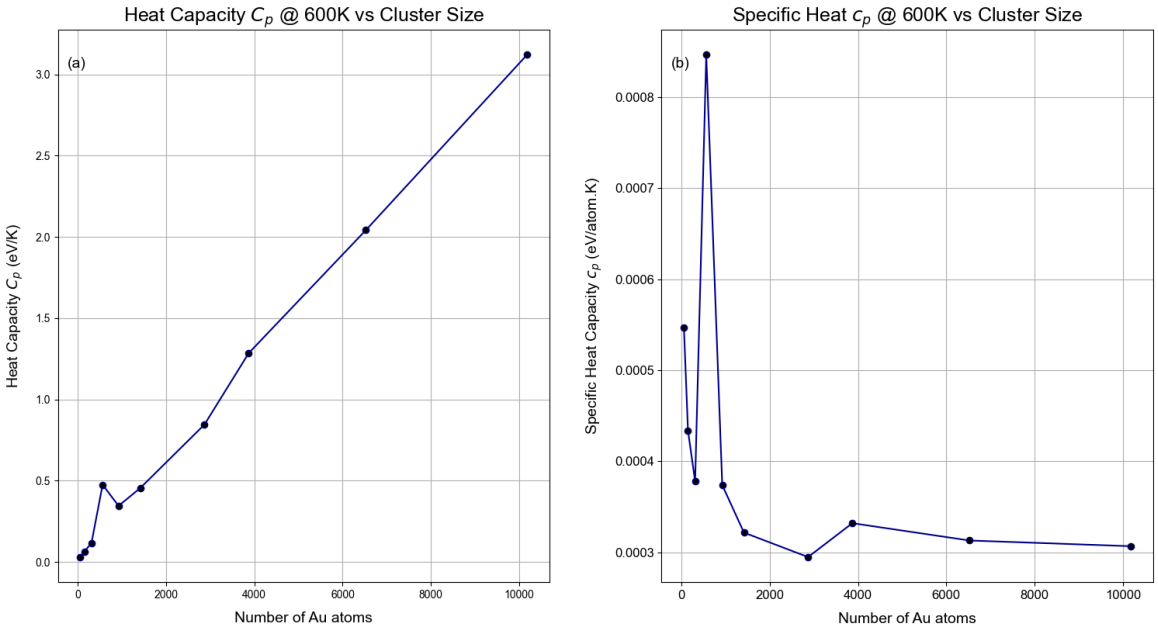


Figure 13: (a) Heat Capacity at 600K vs Cluster-size ; (b) Specific heat capacity at 600K vs Cluster-size

# 7 Parallel Computing and Message Passing Interface

As the cluster size increases, even with a cut off radius, time required to execute a simulation significantly increases. Thus, for large MD systems, executing the simulation program serially may not be sufficient or efficient. To address these challenges, parallel computing and the Message Passing Interface (MPI) become essential.

Parallel computing utilizes the multiple cores present in modern central processing units (CPUs)[1]. It chunks the problem into smaller, concurrent tasks that can be executed simultaneously across multiple processors or nodes. This approach is extremely beneficial for managing MD simulations where the computational workload increases exponentially with the MD system's size and requires a high volume of calculations. The parallel computing is facilitated by MPI.

Message Passing Interface (MPI) is a standardized framework that enables communication between processes in a parallel computing environment. MPI allows processes to send and receive messages, enabling them to coordinate and share data efficiently.

Thus far, all MD simulations carried out in this project existed in infinite vacuum. However, to divide the problem for parallel computing, the domain in which the MD system lies needs to be decomposed. For this purpose a simulation domain $\Omega$ is defined.

Consider three linear, independent vectors $\vec{a}_1, \vec{a}_2, \vec{a}_3$, such that we can express the position $\vec{r}_i$ of an atom $i$ as a function of these vectors and $s_i$, where $s_i$ is the scaled position of the atom. Then, $\vec{r}_i$ can be written as,

$$\vec{r}_i = s_{i,1}\vec{a}_1 + s_{i,2}\vec{a}_2 + s_{i,3}\vec{a}_3$$

For a rectilinear domain, we set $\vec{a}_1 = (L_x, 0, 0)$, $\vec{a}_2 = (0, L_y, 0)$, and $\vec{a}_3 = (0, 0, L_z)$, where $L_x$, $L_y$, and $L_z$ are the linear dimensions. We decompose this domain into smaller scales: $L_x/N_x$, $L_y/N_y$, and $L_z/N_z$. Each subdomain processes its portion independently, and the final results are collected and shared across all ranks.

## 7.1 Algorithm for basic MD simulation with Parallel Computing

**Algorithm 11** Energy Conservation with MPI

**procedure** ENERGY_CONSERVATION_MPI
  **Initialize** MPI environment
  **Define** Domain with periodicity {0, 0, 1} and dimensions {35, 35, 35} ▷ Depends on size of the cluster
  **Initialize** parameters $A$, $\xi$, $p$, $q$, $r_e$, $r_c$, timestep, $nb\_steps$, $k_B$, mass
  **Initialize** energy variables $E_{tot}$, $E_{pot\_loc}$, $E_{pot\_total}$, $E_{kin\_loc}$, $E_{kin\_total}$
  **Enable** domain for atoms
  **Initialize and update** neighbor list with cutoff $r_c$
  **Perform** Ducastelle potential calculation
  **for** $i = 0$ **to** $nb\_steps$ **do**
    $E_{pot\_total} \leftarrow 0$, $E_{pot\_loc} \leftarrow 0$
    **Perform** Verlet predictor step
    **Exchange** atoms across domains
    **Update** ghost atoms and neighbor list
    **Perform** Ducastelle potential calculation
    **Perform** Verlet corrector step
    $E_{pot\_loc} \leftarrow$ **sum** of local atom energies
    $E_{kin\_loc} \leftarrow$ **kinetic energy** from direct summation
    **MPI_Allreduce**($E_{pot\_loc}$, $E_{pot\_total}$)
    **MPI_Allreduce**($E_{kin\_loc}$, $E_{kin\_total}$) ▷ MPI_Allreduce shares final result across all ranks
    **if** $SAVE\_INTERVAL$ **then**
      **Disable** domain for atoms
      **if** domain rank == 0 **then** ▷ Write result only once
        $E_{tot} \leftarrow E_{pot\_total} + E_{kin\_total}$
        **Write** current positions to trajectory file
        **Write** $E_{pot\_total}$, $E_{kin\_total}$, $E_{tot}$ to energy file
      **end if**
      **Re-enable** domain for atoms
      **Update** ghost atoms and neighbor list
      **Perform** Ducastelle potential calculation
    **end if**
  **end for**
  **MPI_Finalize**()
**end procedure**

The required functions for the class 'Domain' and the 'MPI' were provided in the course.

## 7.2 Results: Energy conservation in Parallel processing

In section 3, it was discussed that a good way to check for the correctness of a simulation algorithm is to ensure that the energy is conserved. The same holds true for an MD simulation computed in parallel. The MD simulation for a cluster of 923 atoms is run for 10000 fs, across 4 cores. The Figure 14 shows the energy conserved for the simulation parameters: $A = 0.2061\,\text{eV}$, $\xi = 1.790\,\text{eV}$, $p = 10.229$, $q = 4.036$, $r_e = \frac{4.079}{\sqrt{2}}$ Å, $r_c = 5.0$ Å, time-step = 5 fs, $k_B = 8.617 \times 10^{-5}\,\text{eV/K}$, mass = $196.967 \times 103.6\,\text{g/mol}$.

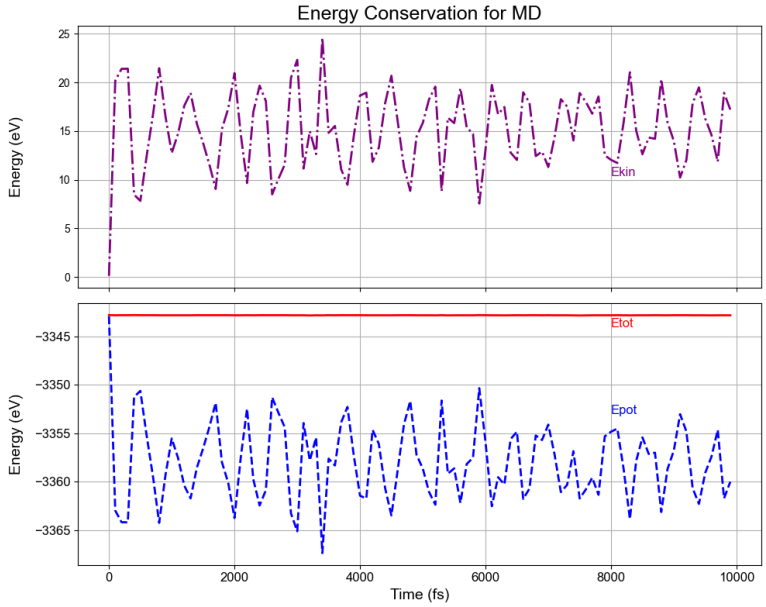


Figure 14: Energy conservation in MD simulation run on 4 cores

# 8 Tensile behaviour of Gold Nano-wires

Gold nano-wires are slender, elongated structures of gold with diameters and lengths on the nanometer scale. In section 8.1, an algorithm is discussed to apply tensile strain to Au nano-wires by defining a simulation domain. The internal force acting on the nano-wire is calculated upon application of strain. This internal force is further used to calculate stress. Understanding the stress-strain response of materials is important for assessing their performance, ensuring safety and reliability of the material in engineering applications. The Figure 15, shows a gold nano-wire under different strains.

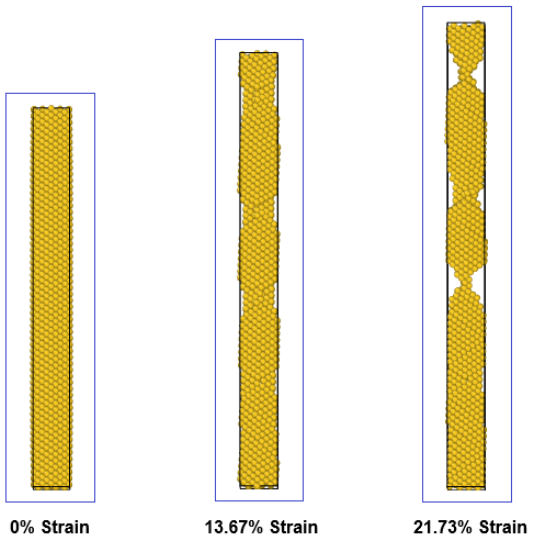


Figure 15: Gold Nano-wire of 3120 atoms under varying strains at 0K

## 8.1 Introduction of Strain on Au Nano-wire by defining Simulation Domain

**Algorithm 12** Apply Strain to Gold Nano-wires

**procedure** APPLY_STRAIN_NANOWIRE(filename, lx0, ly0, lz0, temp, del_l)
  **Initialize** MPI environment
  **Define** Domain with dimensions {lx0, ly0, lz0}, grid size, periodic boundaries in z-direction
  **Initialize** parameters and Atoms with positions, velocitites
  **Enable** domain for Atoms
  **Initialize and update** neighbor list with cutoff $r_c$
  **Perform** Ducastelle potential calculation
  **for** $i = 0$ **to** $nb\_steps - 1$ **do**
    **if** $i\%SCALE\_INTERVAL == 0$ **and** $i \neq 0$ **then** ▷ Does not scale at i=0
      `lz` $\leftarrow$ `lz` + `del_l`
      **Update** domain length and scale domain with Atoms
    **end if**
    **Initialize** force on ghost atoms $ghost\_force = 0$
    **Propagate** using Verlet integrator and Ducastelle potential. Keep ghost atoms and neighbour list updated
    **Compute** temperature $T$ and apply Berendsen thermostat
    **MPI_Allreduce**($ghost\_force$, $ghost\_force\_g$)
    **if** $RELAX\_INTERVAL$ **then**
      **Disable** domain for Atoms
      **if** MPI rank == 0 **then**
        $strain \leftarrow (lz - lz0)/lz0$
        $E_{tot} \leftarrow E_{pot\_total} + E_{kin\_total}$
        **Write** current positions, strain, ghost force, and area (lx * ly).
      **end if**
      **Re-enable** domain for Atoms
      **Update** ghost atoms and neighbor list
    **end if**
  **end for**
  **Finalize** MPI environment
**end procedure**

## 8.2 Results: Stress-strain Curve and Hooke's Law for Gold Nano-wire

Strain is the measure of how much a material is stretched or deformed. Depending on the axes on which force is applied to the material, the type of strain may vary. In this project, tensile strain ($\epsilon$) has been used. That is, strain along the length of the object, and is given by,

$$\epsilon = \frac{L - L_0}{L_0}$$

where, $L$ is the final length and $L_0$ is the initial length of the nano-wire.
The corresponding stress in the system is called tensile stress ($\sigma$). It is the measure of internal resistance of the material under strain, to elongate under the applied force. It is given by,

$$\sigma = \frac{F}{A}$$

where, $F$ is the Force applied along the length and $A$ is the cross-sectional area of the nano-wire.
The following Figure 16 , shows the stress-strain curve of gold nano-wires made of different number of atoms.

The gold nano-wires are elongated along z-axis upto 20% strain, at a strain rate of 0.005 Å/fs. The wire was maintained at 0 K temperature. The other parameters used for the simulation are $A = 0.2061\,\text{eV}$, $\xi = 1.790\,\text{eV}$, $p = 10.229$, $q = 4.036$, $r_e = \frac{4.079}{\sqrt{2}}$ Å, $r_c = 5.0\,\text{Å}$, time-step $= 5\,\text{fs}$, $k_B = 8.617 \times 10^{-5}\,\text{eV/K}$, mass $= 196.967 \times 103.6\,\text{g/mol}$.

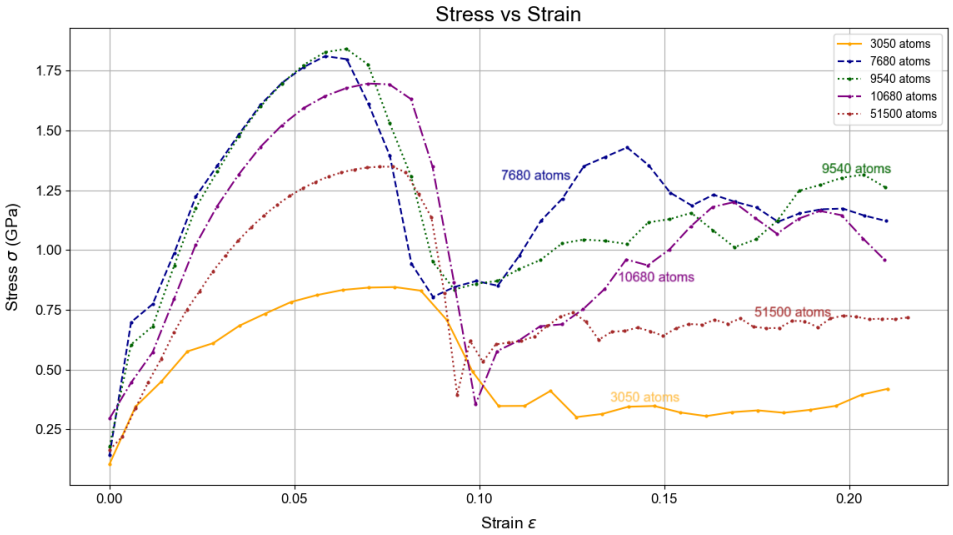


Figure 16: Stress-Strain Curve of Gold Nano-wires made of varying number of Au atoms

It can be observed that for all the gold nano-wires investigated, irrespective of the number of Au atoms, Hooke's law is followed in the elastic region. Hooke's law is applicable in the elastic region and states that the amount of deformation due to strain is directly proportional to the applied stress in the material. Although all gold nano-wires investigated fracture at about 10% strain, the maximum stress a gold nano-wire can endure is highly dependent on the number of atoms, atom configuration, diameter of the nano-wire and defects (if present).
The stress-strain curve was examined for a gold nano-wire of 3050 atoms by varying strain rate at 0 K, and then again at a strain rate of 0.005 Å/fs by varying the temperatures. All other simulation parameters were kept the same. The Figure 17 and Figure 18, illustrate the dependency of stress on strain-rate and temperatures respectively.

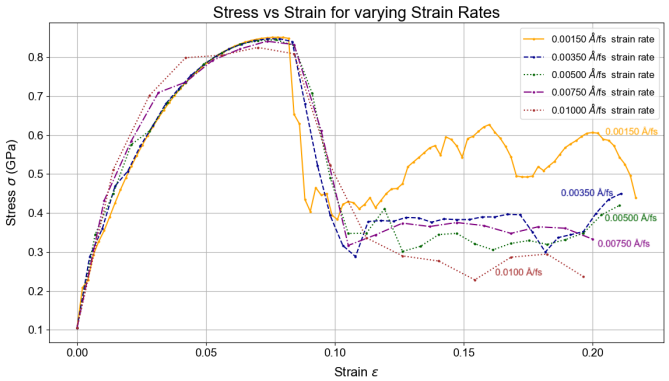


Figure 17: Stress-Strain Curve of Gold Nano-wire of 3050 atoms under varying strain rates

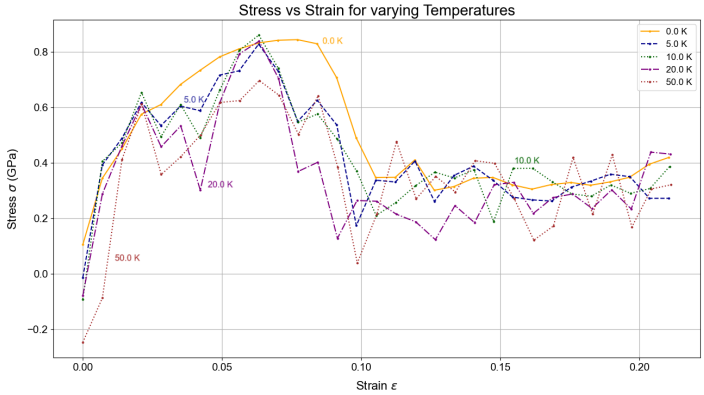


Figure 18: Stress-Strain Curve of Gold Nano-wire of 3050 atoms under varying temperatures

## 8.3 Results: Defects in the Gold Nano-wire with Strain

The Figure 19, shows the defect progression with increasing strain in the gold nano-wire made up of 51500 atoms at a strain rate of 0.005 Å/fs and 0 K, using the same simulation parameter as above. The green coloured atoms represent the Face-Centered Cubic (FCC) lattice structure, the red coloured atoms represent Hexagonal-Close-Packed (HCP) lattice structure, and the remaining atoms are shown in gold colour.

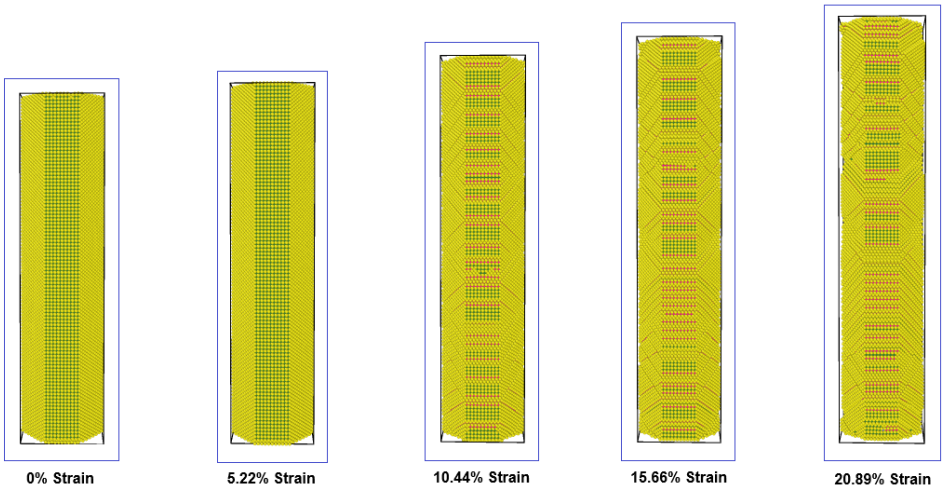


Figure 19: Defect progression in Gold Nano-wire of 51500 atoms under strain

# 9 Conclusion

In this project, various molecular dynamics (MD) simulations were conducted, using the Velocity Verlet algorithm to propagate the systems. The simulations made use of either the Lennard-Jones (LJ) potential or Gupta-Ducastelle potential for force and potential energy calculations, and the Berendsen thermostat was used for equilibration. These MD simulations were set up to explore different aspects of MD simulations and to estimate thermodynamic properties and tensile behaviour gold.

One of the key aspects demonstrated was the energy conservation and its dependence on the chosen time-step($\Delta t$). This property of energy conservation and its relation to $\Delta t$ was one of the methods used in verifying the accuracy of subsequent simulations. The importance of cutoff distance in the potential and force calculation was also emphasized. Although it was seen that its use is most beneficial when simulating a larger number of atoms, as in smaller clusters there is a bigger computational overhead.

In this project, it was shown that LJ potential uses arbitrary dimensionless units and is advantageous for simplistic simulations and MD systems where van der Waal's forces dominate. While the dimensionless units obtained in a Lennard-Jones system can be converted into physical units, the Gupta-Ducastelle potential was used as it better models the interactions in metals.

The melting point, latent heat and heat capacity were calculated for varying cluster sizes and their dependency on cluster size was demonstrated. For cluster sizes from 55 atoms to 10179 atoms, the melting point varied from 500 K to 850 K, this is mostly due to the effect of surface atoms. A similar trend was also observed for the latent heat and heat capacity, however, latent heat per atoms and the specific heat capacity remained more or less constant.

The stress induced due to strain on a gold nano-wire was studied by introducing and altering a simulation domain, thus effectively simulating mechanical deformations. The stress vs strain curves were examined for different lengths and diameters of the nano-wire with number of atoms in the nano-wire ranging from 3050 atoms to 51500 atoms. The stress-strain curves were also examined by varying the temperature and strain rates and subsequently the defect progression in the gold nano-wire made of 51500 atoms was observed at a strain rate of 0.005 Å/fs and 0 K. All gold nano-wires at 0 K fractured at 10% strain.

Finally, a parallelized MD simulation compatible with MPI processing was implemented, demonstrating the scalability of the simulation for more extensive, resource-intensive computations.

Overall, these simulations have provided a comprehensive understanding of the role of time-steps, cutoff distances, and force calculations in MD simulations, while also enabling a more detailed evaluation of gold's material properties.

# References


[1] Lars Pastewka, Wolfram Nöhring, Lucas Frérot. Molecular Dynamics.
https://pastewka.github.io/MolecularDynamics/_lecture/chapter02.html

[2] R. Schneider et al.: Introduction to Molecular Dynamics, Lect. Notes Phys. 739, 3–40.
https://doi.org/10.1007/978-3-540-74686-7

[3] C. Holm, Simulation Methods in Physics 1, Institute for Computational Physics. University of Stuttgart, Stuttgart, Germany (2013)
https://www2.icp.uni-stuttgart.de/ icp/mediawiki/images/5/54/Skript_sim_methods_I.pdf

[4] Matteo Rizzi, Markus Schmitt, Molecular Dynamics, Lennard Jones Fluid, Institute for Theoretical Physics, Cologne.
https://www.thp.uni-koeln.de/trebst/PracticalCourse/molecular_dynamics/molecular_dynamics.pdf

[5] Lennard-Jones, J. E. (1924). On the Determination of Molecular Fields. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 106(738), 463–477. doi:10.1098/rspa.1924.0082
https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.1924.0082

[6] Rapaport, D. C. The Art of Molecular Dynamics Simulation. 2nd ed., Cambridge University Press, 2004.
https://doi.org/10.1017/CBO9780511816581

[7] Lamberti, Vincent & Fosdick, Lloyd & Jessup, Elizabeth. (2002). A Hands-On Introduction to Molecular Dynamics. Journal of Chemical Education - J CHEM EDUC. 79. 10.1021/ed079p601.
DOI:10.1021/ed079p601

[8] M. P. Allen, and D. J. Tildesley. "Computer simulation of liquids." (1988).

[9] Yong and Zhang,Thermostats and thermostat strategies for molecular dynamics simulations of nanofluidics, The Journal of Chemical Physics 138, 084503 (2013); doi: 10.1063/1.4792202
http://dx.doi.org/10.1063/1.4792202

[9] https://www.compchems.com/thermostats-in-molecular-dynamics

[10] Schommers, W. (2019). Theoretical and Computational Methods. In Basic Physics of Nanoscience (pp. 83-142). Elsevier.
https://doi.org/10.1016/B978-0-12-813719-4.00004-3

[11] http://doye.chem.ox.ac.uk/jon/structures/Morse/paper/node6.html

[12] Gómez, JC Ruiz, and L. Rincon. "Melting of intermediate-sized gold nanoclusters." Revista Mexicana de Física 53.7 (2007): 208-211.
https://www.redalyc.org/pdf/570/57036163049.pdf

[13] F. Cleri and V. Rosato. Tight-binding potentials for transition metals and alloys. Phys. Rev. B, 48(1):22–33, 1993.
https://doi.org/10.1103/PhysRevB.48.22.