Computational Physics: Molecular Dynamics

Konstantinos Kilmetis (s3745597) & Diederick Vroom (s2277387)

Abstract

Through the usage of molecular dynamics simulations, emergent properties of Argon atoms in different states of matter is calculated. Assuming that the Argon atoms interact through a Lennard – Jones potential, and periodic boundary conditions, we are able to calculate the pressure and the pair correlation function (PCF). The parameter space of the phase diagram is explored and the triple point of Argon is calculated to be at (T,P) = (90 K , 10^{-4} GPa)

1 Introduction

The dynamics of individual atoms of any species are impossible to calculate analytically. Their emergent behaviour becomes chaotic, in the same fashion that the three-body problem becomes. Indeed, they are more complex than the three-body problem since gravity is a purely attractive force, while the electromagnetic interactions which govern molecular dynamics attract and repel.

Despite that, there is much to be learned by studying such systems. A lot of questions are open, from the behaviour of molecules in extreme conditions to the nature of protein structure. In this work we will be focusing in a much more constrained case; the phases of Argon.

Argon's inert[1] nature, makes it ideal for a proof of concept calculation. We assumed that each atom interacts with one solely through the Lennard – Jones potential, expressed in Eqn. 1,

$$V_{\mathrm{LJ}}(r) = 4\varepsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^{6} \right]. \tag{1}$$

It is described by two terms. The attractive *van der Waals* term, scaling with r^{-6}, and the repulsive *Pauli* term, scaling with r^{-12}. Intuitively, these two combine to form a mildly attractive central force, which becomes fiercely repulsive in very short scales.

Said scales are described by the σ and ε factor. Respectively they correspond to the size of each particle and the minimum energy. They identify the species of each particle, in our case they all take the Argon values of $\sigma = 3.405$ Å and $\varepsilon = 119.8\ k_{\mathrm{B}}$ K.

---

[1] one might say, noble

2 Methods

The Lennard – Jones assumption implicitly ignores any quantum dynamical effects. By making it we assume that any such effects have such a minute influence on the system, so much so that they may be safely ignored. Then we are left with Newtonian Dynamics. The equations which we need to solve are only the classical motion equation for each particle i, shown below:

$$m_{\mathrm{Ar}}\frac{d^2\vec{x}_i}{dt^2} = -\sum_j \nabla V_{\mathrm{LJ}}(\vec{x}_i - \vec{x}_j), \tag{2}$$

the fact that the Lennard – Jones potential is central works to our advantage here. The equation has only to be integrated in the radial directions, since the potential does not predict any force in non-radial directions. Then, can then be simplified as follows,

$$\vec{F}(\vec{x}_i) = -\sum_j \nabla V_{\mathrm{LJ}}\left(\left\|\|\vec{x}_i\| - \|\vec{x}_j\|\right\|\right). \tag{3}$$

Admittedly this is denser notation, put it is simpler to compute.

2.1 Integration Scheme

There is an abundance of ways to integrate Eqn. 3 but not all of them have desirable properties. What concerns us the most is the conservation of energy. By Noether's theorem we know that since the system is symmetric with regard to time -there is no explicit time dependence our potential- energy should be conserved. To achieve this we employ the *velocity Verlet* integrator.

Any numerical solution will diverge from the true solution due to truncation errors. This is a natural consequence of the limited amount of accuracy any computation on limited memory has. Verlet's algorithms also known as leapfrog take advantage of the time-symmetric nature of the problem to cancel these errors out.

This is achieved by taking an initial half step before starting the full calculation. By doing so and subtracting, we both get rid of $\mathcal{O}(h^2)$ terms and introduce time-symmetry. Our calculations still come with a significant $\mathcal{O}(h^3)$ error, but every step it oscillates between underestimating and overestimating the true energy. It doesn't diverge. During the half step, we could either check the new position or velocity during the half step, we chose velocity. Thus, this leaves us with the following equations to evolve:

$$\vec{v}(t + g) = \vec{v}(t) + \frac{h}{2m_{\mathrm{Ar}}}\left[\vec{F}(x(t+h)) + \vec{F}(x(t))\right] + \mathcal{O}(h^3) \tag{4}$$

$$\vec{x}(t + h) = \vec{x}(t) + h\vec{v}(t) + \frac{h^2}{2m_{\mathrm{Ar}}}\vec{F}(\vec{x}(t)) + \mathcal{O}(h^3) \tag{5}$$

2.2 Boundary Conditions

To approximate even a tiny quantity of Argon, we would need to simulate a number of atoms close to Avogadro's number. In the interest of computational expediency we only simulate 108 Argon atoms. In order to both circumvent the need to resolve collisions from firm boundaries and approximate an infinite quantity of Argon, we employ a periodic boundary condition. We seek to copy and paste our cube of 108 particles to all directions, so our original 108 particles are embedded in a wider sea of Argon. Again this conflicts with our desire for computational expediency.

Enter, minimal image convention. It stipulates that, when calculating the distance from particle A to particle B, only use the copy of particle B which lies closest to particle A. Since we know that the maximum distance between particles in one dimension is half the box size; if the copy within the box is less than half a box length away, then this is the closest one. If the copy within the box lies at a distance d more than half a box length away, the closest copy would be $L/2 - d$. We can implement it through the following one liner, which we repeat for every dimension.

```
1  diff = self.pos[j] - particle.pos[j]
2  diff -= boxL * round(diff * c.inv_boxL,0) # 0 in, 1 out
```

MIC.py

That way we can simulate 108 particles, while pretending that they are surrounded by a sea of particles. A corollary of the minimum image convention is the *pacman*-esque behaviour of our boundary. When a particle leaves from the left side, it reappears on the right one.

2.3 Initial Conditions

Our goal for this project is to simulate a collection of Argon atoms in the three phases of matter: Gas, liquid, and solid. Picking the right initial densities and temperatures should be all that is required to observe Argon in these phases. In the real world this might suffice, however, in simulation world this is an enormous bottleneck. Wanting to observe the phases of Argon in this manner requires letting our system settle until it naturally forms the structures corresponding to the different phases. A process which takes seconds in the real world, but can take billions of iterations in simulation world. Hence, we have to be clever about our initial conditions. Our initial conditions consist of two parts: the initial positions of the Argon atoms and their velocities. For both of these conditions, we let the real world nudge us in the right direction.

2.3.1 Position

At first thought, the initial positions of the Argon atoms should be totally random and uniformly distributed. But we know from lab experiments and theory that this is not the case. The atoms interact and therefore have a preferred configuration, which minimizes the total energy of the system. This configuration differs for each state of matter, but does follow a structure. For gas phase Argon, we would expect each atom to have high kinetic energy and thus our collection of atoms to be relatively homogeneously distributed in the container box. For solid phase Argon, the atoms form a crystal structure going by a Face-Centred Cubic (FCC) design, see Fig. 1.
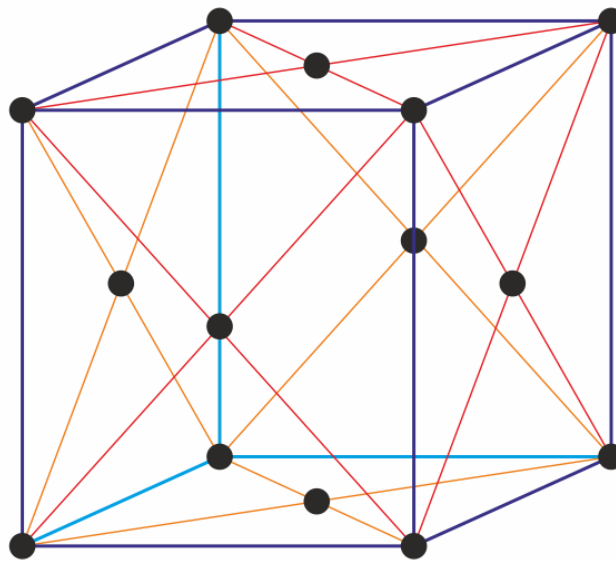

Figure 1: A schematic depiction of a face-centred cubic lattice portion. The gray dots are the desired locations of the Argon atoms in the solid state.

This is the most complex configuration of Argon. For the liquid phase, we expect a middle ground between the gas and solid phase configurations.

Since the solid phase configuration is the most complex, we expect it to also be the most difficult to form. Hence, we will use the FCC-lattice as the initial positions of our Argon atoms. Should we wish to simulate solid phase Argon, the atoms are already in the right positions, and should we wish to simulate the other phases, the atoms more easily relocate to the newly desired configuration.

2.3.2 Velocity

Initially, we allowed the initial velocities to be of some random magnitude and direction, with a homogeneous distribution. But using statistical physics we know that in the real world this not the case. As per the classical interpretation, velocities of particles are dependent on the temperature of the surrounding environment. More precisely, the magnitude of each velocity component v_x follows a Maxwellian distribution, a Gaussian,

$$p(v_x) \propto \exp\left(-\frac{m_{\mathrm{Ar}} v_x^2}{2 k_{\mathrm{B}} T}\right), \tag{6}$$

where $p(v_x)$ is the probability of finding the magnitude v_x of the x-component of the velocity. So we should expect our simulation to obey this distribution as well, and hence we draw random velocities according to the Maxwell distribution.

Unfortunately, there is one caveat. For Eqn. 6 to hold in our simulated system we require a large number of Argon atoms, far too many for us to simulate in any reasonable amount of time. Consequently, since Eqn. 6 holds for an environment in thermal equilibrium, our simulated system will most likely not be in thermal equilibrium. For it then to transform to being in equilibrium, the Argon atoms will need to interact and exchange energy, a process which, again, can take an enormous amount of time. We accelerate this process by the nudging the velocities of each Argon atom in the right direction.

One other result from statistical physics is that the total kinetic energy of a system is approximately given by,

$$E_{\mathrm{kin}} = (N-1)\frac{3}{2} k_{\mathrm{B}} T, \tag{7}$$

with N the number of particles in the system. Like before, this equation only holds for a system containing a large number of particles. However, we can use this theoretical kinetic energy as a target energy for our simulated system, such that it more closely follows reality. Should we find that our simulated system has a total kinetic energy close to the target energy, we can safely assume that our simulated system is (close to being) in equilibrium. On the contrary, should we find that the simulated total kinetic energy deviates substantially from the target kinetic energy, we change the velocities of the Argon atoms according to the scaling parameter λ,

$$\lambda = \sqrt{\frac{(N-1) 3 k_{\mathrm{B}} T}{\sum_i m_{\mathrm{Ar}} v_i^2}}, \tag{8}$$

with the $\sum_i m_{\mathrm{Ar}} v_i^2$ term corresponding to the total kinetic energy of our simulated system. Using λ, we scale all velocities of our Argon atoms equally $(\vec{v}_i \rightarrow \lambda \vec{v}_i)$.

The important part of this process is to do it after the simulation has already run for several integration steps, doing this allows the Argon atoms to already exchange some energy with each other.

This process of finding the correct initial velocities we called "equilibration", and can be summarized in 5 steps:

1. Draw a velocity for each Argon atom, following the Maxwellian distribution given in Eqn. 6.

2. Run the simulation for several integration steps, we chose to run it for 10 integration steps during the equilibration process.

3. Determine the total simulated kinetic energy.

4. If the total simulated kinetic energy deviates significantly from the target kinetic energy (see Eqn. 7), rescale the velocities of all Argon atoms by λ (see Eqn. 8). We used a tolerance of 5% to determine whether the simulated kinetic energy deviates significantly from the target kinetic energy.

5. Repeat steps 2-4 until the simulated total kinetic energy is within the desired tolerance of the target kinetic energy.

Using the initial FCC-positions and performing the equilibration process will result in a system of Argon atoms in equilibrium with the surrounding thermal bath, and now better represents Argon in the corresponding state of matter. From here onwards, the true simulation can begin.

2.4 Observables

We have already discussed that we are able to extract energies from our simulation, and thus can observe intrinsic properties of our simulated system. In this section, we describe two more properties we can observe from our simulated system: pair correlation and pressure.

2.4.1 Pair Correlation

Depending on the initial conditions, in order to minimize their energy, Argon atoms will arrange themselves into different configurations. These are the three familiar states of matter: solid, liquid and gas. In order to quantify the difference between these states we calculate the distribution of distances between pairs of Argon atoms. This is done via the pair correlation distribution $g(r)$,

$$g(r) = \frac{2L^3}{N(N-1)} \frac{\langle n(r) \rangle}{4\pi r^2 \Delta r},$$

(9)

where L is the box size, N the number of Argon atoms, Δr by how much we increase the distance r each step (the bin size), and $n(r)$ counts how many Argon atoms we encounter in the interval $[r, r + \Delta r]$. $\langle\,\rangle$ indicates an average over configurations. Where a configuration can either be a snapshot of the simulation at a different integration step, or a different simulation with the same initial density and temperature but different initial positions and velocities.

In our collection of Argon atoms, we pick a reference atom and from there we move radially outwards and count how many other atoms we encounter over distance. Doing this for all Argon atoms and summing the results gives the $n(r)$ for one integration step. Doing this for many integration steps during the simulation and then averaging the results gives $g(r)$. For this process, we expect a different result for each state of matter.

As we have established before, for the solid state of matter, from lab observations it is known that Argon atoms form a regular crystal structure following the FCC configuration. Due to this regular structure, we only expect to encounter other Argon atoms at very regular distances, hence $g(r)$ for solid state Argon would look like a collection of Dirac-delta functions ("Barcode").

We know the gas to be the most energetic state of matter, hence the average energy per Argon atom in our simulations should be very high for a gas. This high energy causes the Argon atoms to interact numerous times during the simulation, eventually settling down into a structure which distributes the atoms as homogeneously as possible. Hence, we expect $g(r)$ for a gas state simulation of Argon to be relatively flat over distance, since the likelihood of finding an Argon atom at any distance should be constant. In fact, the prefactors in Eqn. 9 have been tuned in such a manner that for a gas we expect $g(r) \approx 1$.

For the liquid state, we expect $g(r)$ to follow similar behaviour as Argon on a macro level, namely, that it inherits properties of both the solid and gas state. This gives that $g(r)$ is relatively constant; no sharp peaks, like a gas. But does rise and fall gradually over distance, like a solid.

2.4.2 Pressure

From a classical point of view, pressure would be the average force exerted on the boundary of our simulation box. However, since we are considering our box to be infinite by the minimal image convention, we need to use a different way of calculating the pressure of our simulated system, namely, an adapted version of the equation for pressure P given in Verlet 1967 [3],

$$\frac{P}{k_{\mathrm{B}}Tn} = 1 - \frac{1}{6Nk_{\mathrm{B}}T}\left\langle \sum_i \sum_{j>i} r_{ij}\frac{\partial V_{\mathrm{LJ}}(r_{ij})}{\partial r}\right\rangle, \tag{10}$$

where n is the number density of Argon atoms, and $\langle \ \rangle$ is again the average over configurations. Eqn. 10 consists of two parts: the 1 encodes the ideal gas law, as per the factors on the left-hand-side, and the sum-part encodes the change in pressure due to interactions occurring between Argon atoms, as per the Lennard – Jones potential V_{LJ}. Using Eqn. 10, we determined pressures for our simulations by calculating $\sum_i \sum_{j>i} r_{ij} \frac{\partial V_{\mathrm{LJ}}(r_{ij})}{\partial r}$ at each integration step corresponding to 10% of the total number of integration steps, and afterwards averaging each calculated sum-part to get the total pressure of our simulated system.

2.5 Triple Point

Whether or not matter exists in a certain phase depends on its pressure and temperature. However, the boundaries between two phases are not clear cut. Take water as an example. Its freezing point is at 0°C, but if we placed some of it in 0.5 °C -and atmospheric pressure- some of it would be a solid and some of it would be liquid. The two phases are allowed to co-exist.

Pushing this idea, we can imagine that there should be a point in the Temperature-Pressure parameter space where all three phases co-exist. In order to find this sweet-spot in the parameter space, we use our simulation code as an experimental bench. We calculated the pressure and deduced which phase of matter argon is in through the pair correlation function for multiple initial condition. We did not conduct an exhaustive search, but rather took advantage of the known shape (Υ) of the transition lines. Thus, we attempted to find where the transition lines are and follow them to the triple point.

3 Results

We ran 3 simulations, each corresponding to a different phase of matter for our Argon atoms. The initial conditions of these simulations are summarised in Tab. 1.

State	Temperature		Density	
	$[k_{\mathrm{B}}\varepsilon]$	$[\mathrm{K}]$	$[m_{\mathrm{Ar}}\ \sigma^{-3}]$	$[\mathrm{g\,cm}^{-3}]$
Solid	0.5	59.90	1.20	2.01
Liquid	1.0	119.8	0.80	1.34
Gas	3.0	359.4	0.30	0.50

Table 1: Initial conditions. Each simulation was also run using 108 Argon atoms, ran over $10\,\mathrm{ps}$, and with a time step of $0.001\,\mathrm{ps}$.

These initial conditions were confirmed to correspond to their associated state, as per Tab. 1. For each simulation, an energy-evolution figure was created as a check to see if the simulation behaved as expected. The energy-evolution figure for the Gas-state simulation is given in Fig. 2.
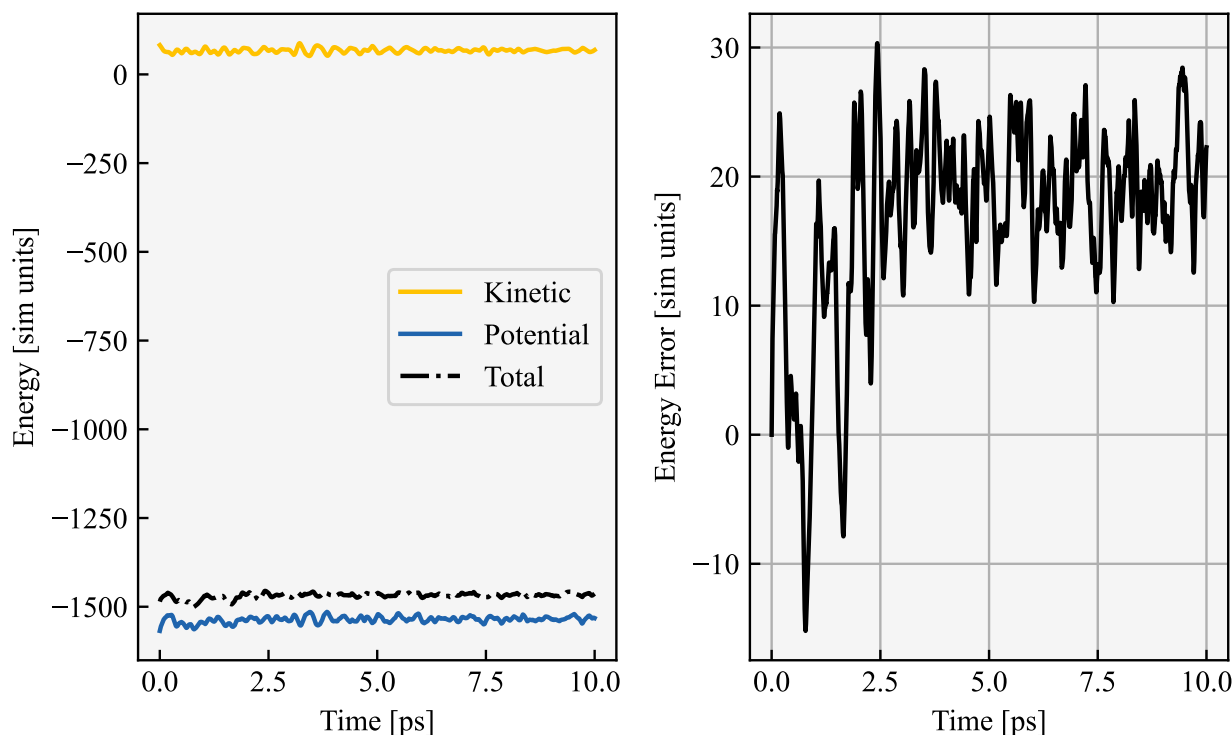
Solid

Figure 2: Total energies of the Argon atoms during the simulation for the Gas simulation given in Tab. 1.

The most important thing to note in Fig. 2 is that the total energy, the black dash-dotted line in the left panel, is relatively constant over the course of the simulation. This indicates that we have done the integration correctly, since the Verlet algorithm approximately conserves energy. All simulations with the initial conditions given in Tab. 1 produced similar energy-evolution figures.

For each simulation, we created an additional figure, the pair correlation distribution over distance, as can be seen in Fig. 3.
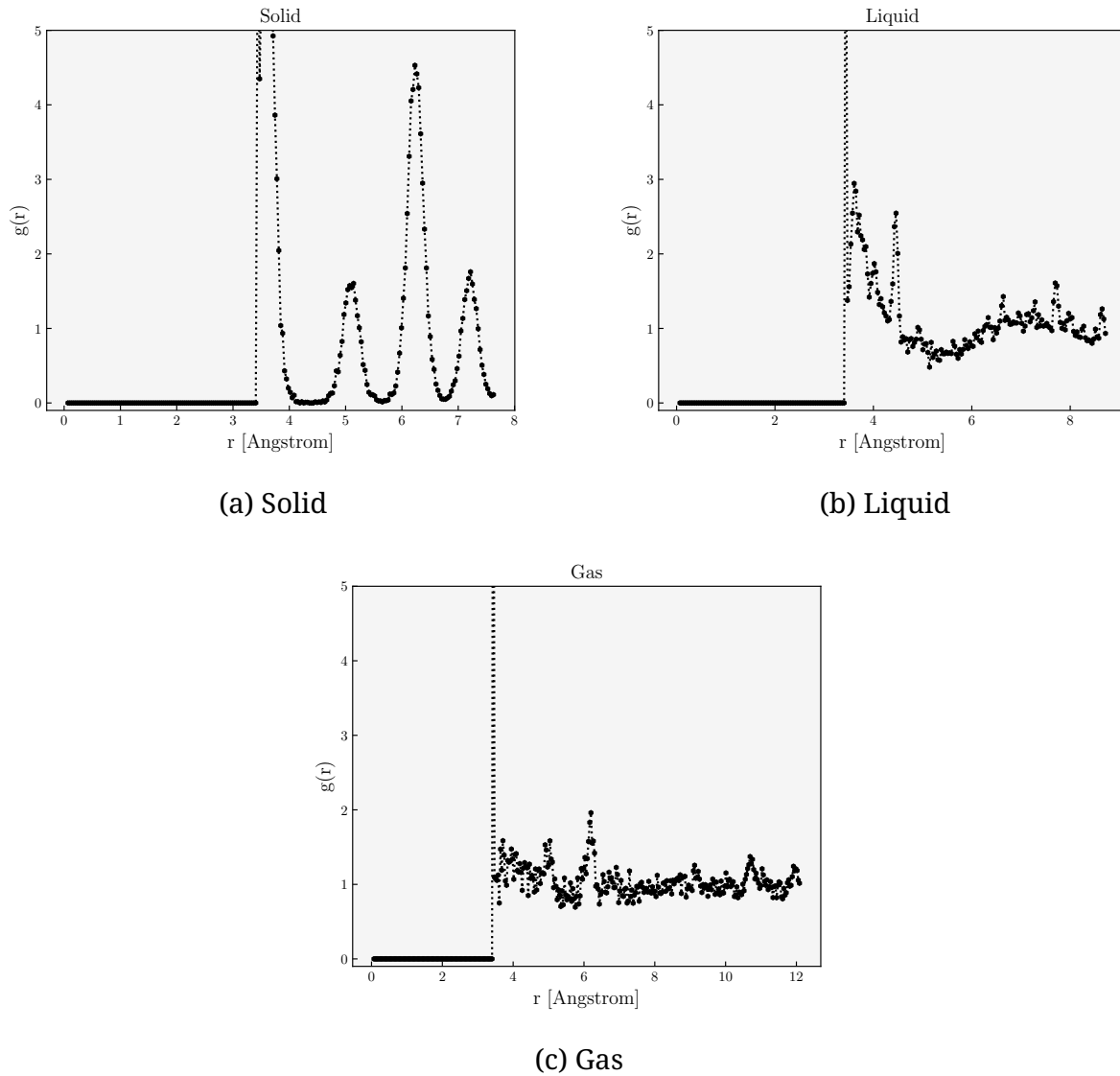
(a) Solid

(b) Liquid

(c) Gas

Figure 3: Pair Correlation distribution over distance for each simulated state.

The main things to note are the distinctive shapes of the pair correlation distributions of the solid and gas states (Figs. 3a and 3c respectively). As we have established in Sec. 2.4.1, for the solid state simulation we expect other Argon atoms to only appear at certain distances with respect to the reference Argon atom, giving rise to the barcode-like shape given in Fig. 3a.

As established in Sec. 2.4.1, in the gas phase, the Argon atoms are the most unbound from one another and thus distribute themselves nearly homogeneously, giving rise to a relatively constant probability of finding an Argon atom a certain distance from another Argon atom, hence the flat pair correlation distribution given in Fig. 3c.

As expected, the liquid pair correlation distribution (see Fig. 3b) shares aspects of both the solid and gas state pair correlation distribution.

In summary: Our generated pair correlation distributions match our intuition, indicating that the initial conditions given in Tab. 1 correspond to their respective states.

Lastly, we calculated the average pressure of each of our simulated systems, as per Eqn. 10, the mean values and standard deviations of which are given in Tab. 2.

State	Pressure	
	$[k_{\mathrm{B}}\varepsilon\,\sigma^{-3}]$	$[\times 10^9\,\mathrm{erg\,cm^{-3}}]$
Solid	8.0 ± 0.3	3.3 ± 0.1
Liquid	1.0 ± 0.3	0.4 ± 0.1
Gas	0.9 ± 0.1	0.38 ± 0.04

Table 2: Measured pressures and standard deviations for the 3 simulated collections of Argon atoms for different states, as per Tab. 1.

Next, we present the phase space of Argon, as we calculated it from our numerical experiments.
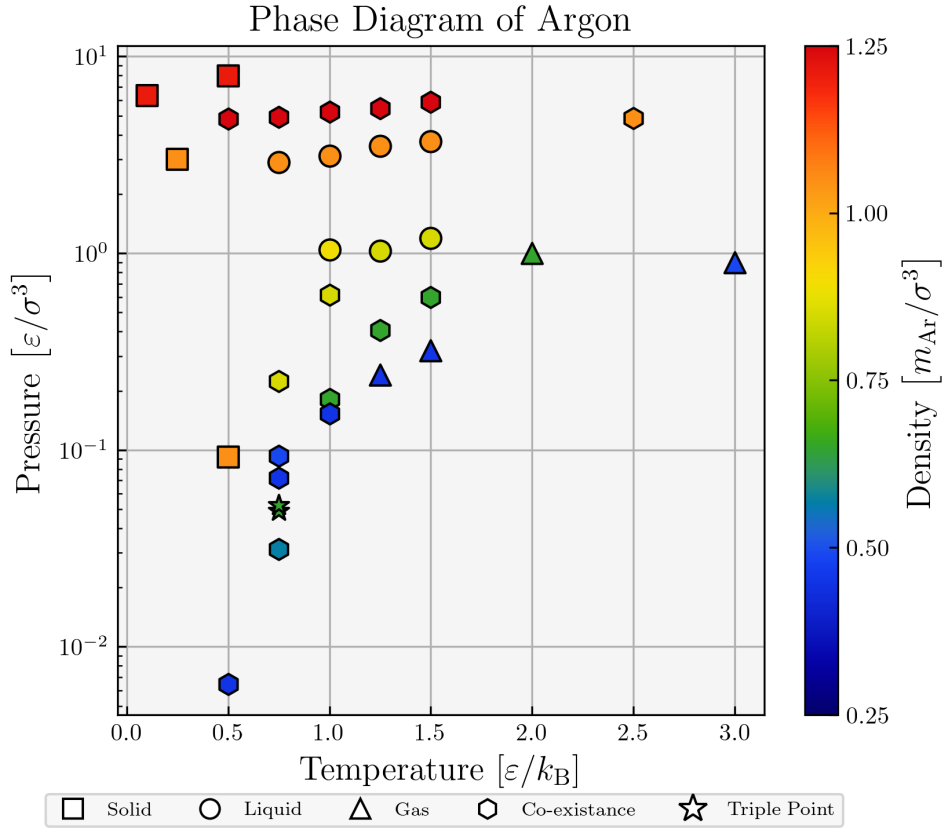

Figure 4: Phase space of Argon. X-axis is the temperature, y-axis is the pressure, color scale corresponds to density. Different markers signify different phases of matter, or coexistance. When three phases co-existed, we deduced it is the triple point.

Our deduced triple point (T,P) = (0.75 $k_{\mathrm{B}}\varepsilon$, 0.004 $k_{\mathrm{B}}\varepsilon\sigma^{-3}$) = (90 K , $10^{-4} \pm 9\times10^{-4}$ GPa) agrees with the literature value [2] of a Lennard – Jones Argon atom (T,P) = (0.69 $k_{\mathrm{B}}\varepsilon$, 0.001 $k_{\mathrm{B}}\varepsilon\sigma^{-3}$) and with the experimentally deduced value, as is evident in Fig. 4.
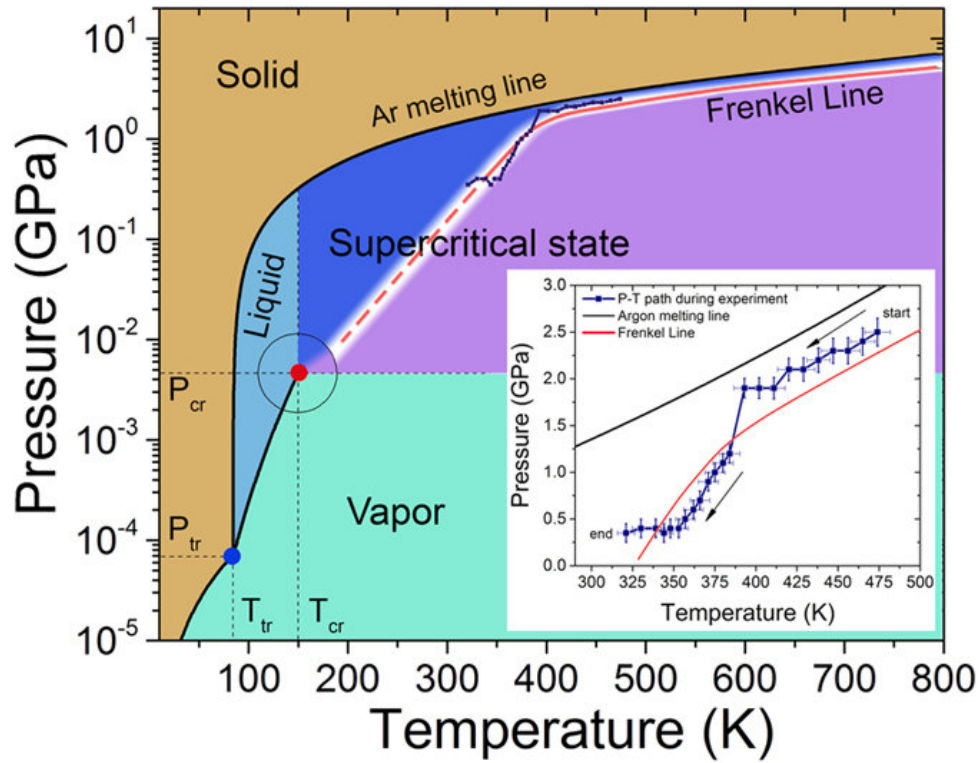
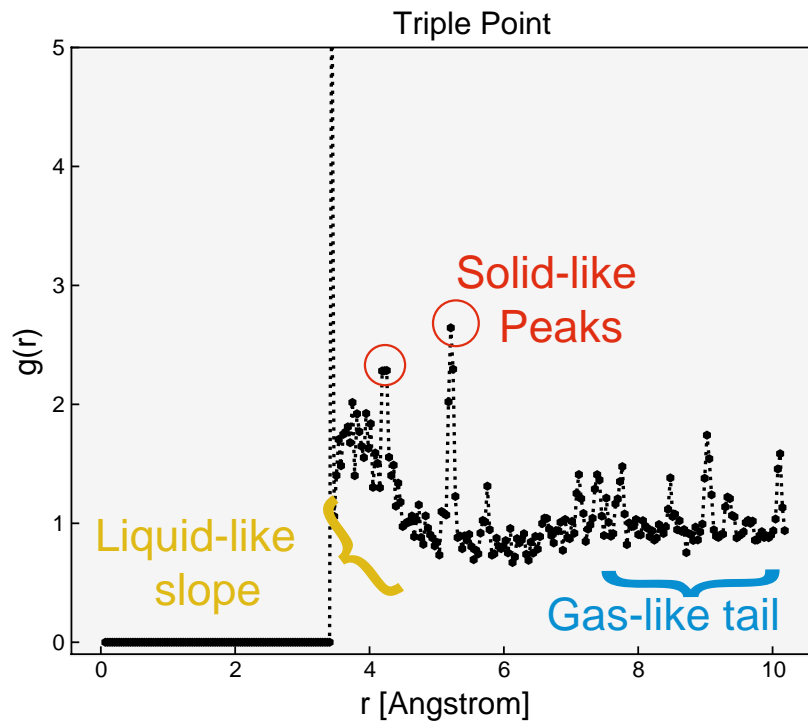Figure 5: Experimentally determined phase space of Argon, adapted from Bolmatov et al. (2015) [1].

Figure 6: The pair correlation function for our triple point, (T,ρ) = (0.75 $k_B\varepsilon$, 0.5 $m_{Ar}\sigma^{-3}$). It exhibits characteristics found in all three phases of matter.

4 Summary

We present here a tentative summary of how we executed Project 1: Molecular Dynamics of the MSc course Computational Physics given in the year 2024 at Leiden University.

1. We built a code collection capable of simulating Argon atoms contained inside a pseudo-infinite box, as by the minimal image convention.

2. Our basis for the simulation lies in the usage of the Lennard – Jones potential (see Eqn. 1) and implementing the Verlet integration algorithm (see Eqns. 4 and 5).

3. As to more accurately describe the macro observations of Argon in different states, we implemented the face-centred cubic lattice as the initial positions of our Argon atoms in the simulation.

4. For similar reasons, we implemented a setup-stage before the simulation (equilibration), so that the velocities of the Argon atoms more closely follow what we would expect to see for a large number of atoms.

5. Having run the simulations for each state of matter (see Tab. 1), we created two figures for each state: the energy-evolution and pair correlation distribution. Each figure matched what we intuitively would expect.

6. For each ran simulation, we calculated the mean pressure and its standard deviation via Eqn. 10, the results of which are showcased in Tab. 2.

7. By running large grids of simulations, we were able to create a phase diagram of Argon and were able to deduce that the triple point of Argon occurs at (T,P) = (90 K , $10^{-4} \pm 9 \times 10^{-4}$ GPa).

Bibliography

[1] Bolmatov, Dima and Zhernenkov, Mikhail and Zav'yalov, Dmitry and Tkachev, Sergey and Cunsolo, Alessandro and Cai, Yong, *The frenkel line: A direct experimental evidence for the new thermodynamic boundary*, Scientific Reports, (2015).

[2] A. J. Schultz and D. A. Kofke, *Comprehensive high-precision high-accuracy equation of state and coexistence properties for classical Lennard-Jones crystals and low-temperature fluid phases*, The Journal of Chemical Physics, 149 (2018).

[3] L. Verlet, *Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules*, Physical review, 159 (1967), p. 98.