

H1b MD simulation - dynamic properties

In this homework problem you are asked to use the molecular-dynamics (MD) simulation technique. You will study dynamic properties of aluminum, both in the solid and liquid phase, using a Neuroevolution potential (NEP) [1].

In the first part, task 1-4, you will develop a basic MD program in C. **Keep in mind that the runtime of your program will be rather long. Plan accordingly!** For your convenience a few C routines are provided. In the second part, task 5-7, you will analyze the output data and determine the mean squared displacement $\Delta_{MSD}(t)$, the velocity correlation function $\Phi(t)$ and its spectrum $\hat{\Phi}(\omega)$. Here you are asked to develop your own analysis programs.

It is highly recommended to run your MD program and store the results for the trajectories. You can then start to analyse your results based on the stored trajectories. You should use the "Atomic simulation units", introduced in appendix A in exercise E1.

Task 1-4 should be done in C. For the analyzes in task 5-7 you are allowed to use Python.

You should present your results in a written report. **For instructions how to write the report, read the information on the homepage.**

Task

1. The course Gitlab repo contains code that you may use as a starting point. In the `run.c` file provided to you (under `src/`), create a `n_atoms` by 3 matrix of coordinates representing atomic positions in an fcc lattice using the `init_fcc` function in `lattice.c`. The syntax for this is

```
init_fcc(positions, N, lattice_param);
```

where `positions` is the matrix in which the atomic positions are stored, `N` is the number of unit cells in each direction and `lattice_param` is the lattice constant. Use a $4 \times 4 \times 4$ supercell, i.e. so that the number of atoms `n_atoms` is 256. We encourage you to reuse your library from C4 for this and later tasks (put it in `tools.c` and `tools.h`). Run `make` to compile the program.

Create fcc structures of different volume by varying the lattice constant and calculate the potential energy of each generated structure using the `get_energy_AL` function in `potential.c`, the syntax of which is

```
E_pot = get_energy_AL(positions, N * a0, natoms);
```

where `E_pot` is the variable in which the energy is stored. Write the lattice constant and the corresponding energies to file and determine the equilibrium lattice constant by doing a quadratic fit in `Python`. Make sure that you can reproduce the data in Fig. 1. What is the theoretical lattice parameter at 0 K? (1p)

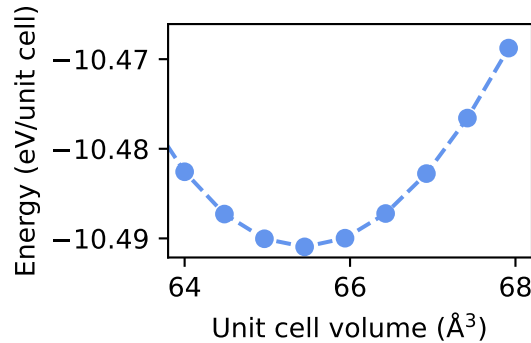


Figure 1: Energy-Volume diagram of aluminium.

2. Introduce small random deviations from the initial lattice positions. The displacements should be uniform in the interval $\pm 6.5\%$ of the lattice spacing in each direction. Solve the equation of motion using the velocity Verlet algorithm with periodic boundary conditions. Compute forces (and in later tasks virials) by calling `get_forces_AL` and `get_virial_AL` in `potential.c`. Alternatively, you can call `calculate`, which gives you all quantities at once, saving precious simulation time. You should not fold the atomic positions back into the simulation box, as the potential provided in `potential.c` correctly applies the boundary conditions.

Set the initial velocities to zero and study the time-evolution of the potential, kinetic and total energies. Compute the average temperature and check that it is around 600-850 K. What is a suitable time step? Vary the time step and investigate the effect on energy conservation. How does the system behave with timesteps that are far too large, just a bit too large, and small enough? (You are, of course, expected to explicitly show this with plots and discuss them.) **Hint:** Read section 4.3 in the lecture notes "Molecular dynamics". The effects of a timestep that is barely too large can be very subtle. Carefully consider your plot limits! (3p)

3. Implement the equilibration routines described in appendix E in the lecture notes "Molecular dynamics". Note that the isothermal compressibility can be treated as a constant scaling of the time constant.

Further, the isothermal compressibility is the inverse of the bulk modulus. You may use a tabulated value for the isothermal compressibility and the same value can be for both the solid and the liquid state. The time constants should be on the order of at least a few hundred time steps.

Equilibrate the system in the solid state, at $T=500^\circ\text{C}$ and $P=1$ bar. Notice that for the small system sizes used in the simulation study, the magnitude of the fluctuations is large. When equilibrating at low pressure the instantaneous pressure can become negative (see Fig. 2). What is the lattice parameter of the equilibrated system? Is it what you expect? **Hint:** It is useful to study the time-evolution of the simulation cell size for the pressure equilibration. **It is also a good idea to dump the simulation data to file, so that you can develop your analysis scripts separately from your simulation code.**

Determine the temperature T and pressure P using constant energy and volume simulation by making a time-average along the MD trajectory. Convince yourself that the system is in a solid state by plotting the components of the position as a function of time for a few different particles. Verify that the particles stay close to their initial positions also for long times. (4p)

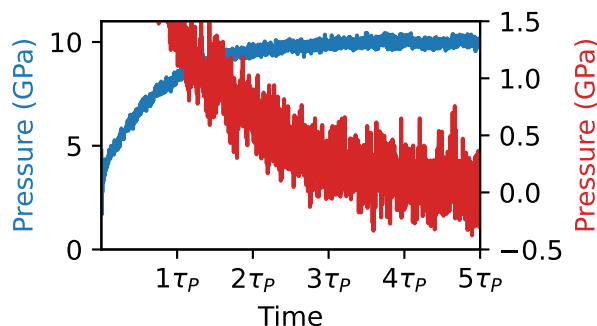


Figure 2: Example of fluctuations during pressure equilibration for high pressure equilibration (10 GPa) and low pressure equilibration (0.1 MPa).

4. Repeat the previous task for the liquid state, at $T=700^\circ\text{C}$ and $P=1$ bar. Convince yourself that the system is in a liquid state by plotting the components of the position as a function of time for a few different particles. Verify that the particle trajectories deviate substantially from their initial positions for long times. What is the lattice parameter of the equilibrated system? Is it in line with what you would expect in comparison to that in Task 3? **Hint:** To melt the system it can

be useful to first increase the temperature above $T=700^\circ\text{C}$ and then decrease it to $T=700^\circ\text{C}$. (2p)

5. Determine the mean squared displacement $\Delta_{MSD}(t)$, defined in Eq.(85) in the lecture notes "Molecular dynamics", both in the solid and the liquid state. Use the same temperatures and pressures as in the two previous tasks and for the evaluation of $\Delta_{MSD}(t)$ use the algorithm described in section 8.1 in the lecture notes "Molecular dynamics". Determine also the self-diffusion coefficient D_s in the liquid state (see Eq.(89)). In the solid state the mean squared displacement approaches a constant value, which? (3p)
6. Consider next the velocity correlation function $\Phi(t)$, introduced in Eq.(75). Use the same algorithm as for the mean squared displacement and determine the velocity correlation function $\Phi(t)$ perform for the liquid state.

The velocity correlation function can also be obtained by first determine the power spectrum and then making an inverse Fourier transform. However, some care has to be taken due to that the FFT assumes a periodic function. To circumvent this one can add a buffer zone with zeroes and normalise the correlation function correctly. This is sometimes denoted the "Fast Correlation Algorithm" and it is described in appendix G in the lecture notes "Molecular dynamics". Perform such a calculation for the velocity correlation function $\Phi(t)$ in the liquid state. To get better statistics and less noise you can divide your data into chunks of say about 2000 steps. Perform the evaluation for each chunk of data and then take a mean of the different data sets.

Compare your results using the two different methods. (4p)

7. When using the Fast Correlation Algorithm you also get access to the power spectrum or spectral function

$$\begin{aligned}\hat{\Phi}(\omega) &= \int_{-\infty}^{\infty} \Phi(t)e^{i\omega t}dt \\ &= 2 \int_0^{\infty} \Phi(t)\cos(\omega t)dt\end{aligned}$$

Plot the result you obtained for $\hat{\Phi}(\omega)$ as function of ω . The self-diffusion coefficient is given by the zero frequency component of the spectral function (see Eq. (92) in the lecture notes "Molecular dynamics"). What value do you obtain for the self-diffusion coefficient? Is this consistent with the value you obtained from the mean squared displacement? (3p)

References

1. Fan, Z. *et al.* GPUMD: A package for constructing accurate machine-learned potentials and performing highly efficient atomistic simulations. *The Journal of Chemical Physics* **157**, 114801. <https://doi.org/10.1063/5.0106617> (2022).