

Introduction to LAMMPS

LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) is an open-source molecular dynamics (MD) software package designed to simulate particles at atomic, mesoscopic, and continuum scales. Developed by Sandia National Laboratories, it is optimized for high-performance parallel computing (HPC) and is widely used in physics, chemistry, materials science, and engineering.

LAMMPS is particularly effective in simulating large systems containing millions of atoms due to its domain decomposition method, which efficiently distributes computational work across multiple processors. It is highly modular, allowing users to integrate different force fields, computational models, and external packages. The software supports classical molecular dynamics, as well as hybrid models that incorporate quantum effects via external coupling.

Key Features

LAMMPS offers a wide range of functionalities, making it a versatile tool for molecular dynamics simulations:

- **Interatomic Potential Support:** LAMMPS includes a variety of force fields such as Lennard-Jones, EAM (Embedded Atom Method), Tersoff, ReaxFF (reactive force fields) etc., allowing for the simulation of different materials and molecular systems.
- **Parallel Computation:** Designed for high-performance computing, LAMMPS efficiently scales across CPUs and GPUs.
- **Flexibility in Simulation Types:** Users can perform energy minimization, equilibrium and non-equilibrium MD, thermostating, barostatting, and shockwave simulations.
- **Highly Customizable:** It allows users to define complex simulation environments, manipulate atomic structures, and apply external forces, making it useful for a wide range of scientific applications.

Basic Structure of a LAMMPS Input Script

A typical LAMMPS input script consists of several key components. Below is an overview of the most important sections, along with an example command for each. Furthermore, for each presented command you can find a short explanation. If you need more information, you can click on them to access the official LAMMPS documentation. Alternatively, you can click [here](#) to view the LAMMPS commands documentation.

1. Initialization

The initialization section defines the units of measurement, the system's dimensionality, and the boundary conditions. It is important to specify these settings before defining atomic structures.

```
units      metal
dimension  3
boundary  p p p
```

- The `units` command sets the energy, length, time, and mass units (e.g., `metal`, `lj`, `real`).
- `dimension` defines the simulation dimensionality (2D or 3D).
- `boundary` defines the boundary conditions (periodic, fixed, or shrink-wrapped).

2. System Definition

This section specifies the atomic lattice, defines the simulation box, and places atoms inside it.

```
lattice    bcc 4.0
region    box block 0 10 0 10 0 10
create_box 1 box
create_atoms 1 box
```

- `lattice` sets the crystal structure and lattice constant.
- `region` defines the spatial region of the simulation box.
- `create_box` initializes the simulation box.
- `create_atoms` populates the box with atoms.

3. Defining Atomic Interactions

LAMMPS provides various interatomic potentials to model atomic interactions.

```
pair_style eam/alloy
pair_coeff * * Al99.eam.alloy Al
```

- `pair_style` selects the interatomic potential (e.g., `lj/cut`, `eam/alloy`, `reax/c`).
- `pair_coeff` assigns potential parameters for atomic interactions.

4. Simulation Settings

Before running a simulation, neighbor list settings and velocity initialization are configured.

```
neighbor 0.3 bin
neigh_modify delay 5 every 1
```

- `neighbor` sets the neighbor list cutoff for force calculations.
- `neigh_modify` refines how often the neighbor list updates.

5. Running the Simulation

Once the system is set up, integrators are defined, and the simulation is executed. Here, we run an evaluation of the energy.

```
run 0
```

- `run` starts the simulation for a specified number of steps.

6. Output and Data Collection

LAMMPS provides various ways to collect simulation data, such as thermodynamic output and trajectory dumps.

```
write_data output.data
```

- `write_data` Write a data file in text format of the current state of the simulation.

Applications of the software

LAMMPS is extensively used in computational research for modeling the physical properties of materials. Some of its key applications include:

- **Material Science:** Simulating the mechanical, thermal, and electrical properties of metals, ceramics, and semiconductors.
- **Biomolecular Simulations:** Modeling proteins, lipid membranes, and DNA with force fields like CHARMM and AMBER.
- **Surface Interactions and Catalysis:** Studying adsorption, diffusion, and chemical reactions using reactive force fields.
- **Extreme Environments:** Simulating shock waves, radiation damage, and high-pressure conditions in planetary science.

Now, that we saw the basics of how LAMMPS works, we are ready to put everything to practice during the next exercises. By calculating interesting properties for different materials.

In the rest of this document, red shaded boxes highlight the tasks and problems that you need to solve (and for which you need to write a report afterwards). After each box, various useful comments and tips will clarify some points and provide additional information.

Excercise 1

For this exercise, one has to calculate the minimum energy structure of silver (Ag). Silver is an element that crystallizes in a face-centered cubic (FCC) structure. To achieve this, you will use the **LAMMPS** simulation package and apply two different approaches.

Problem 1 : Relaxation of the FCC-Ag structure.

From a computational point of view, this involves providing a code (LAMMPS for this lab) with an initial atomic configuration—specifically, a unit cell and a set of atomic positions. These positions do not have to be perfectly accurate but should serve as a reasonable initial guess for the compound of choice. The goal is to determine the equilibrium configuration, which corresponds to the state of lowest energy.

The software employs a minimization algorithm to adjust the atomic positions, reducing the potential energy, interatomic forces, and stresses. The configuration that results in the lowest energy is known as a local minimum.

In general, both atomic positions and the cell size (lattice vectors) can be relaxed. However, in some cases, it may be necessary to relax only one of the two. For this exercise, we will relax both.

Before you start your calculations, make a plot of the unit cell of FCC-Ag. What is the packing fraction for it?

A. (5 points) Calculate the lattice constant (in Å) and the total energy (in eV) for bulk silver (Ag) in the FCC structure using LAMMPS. In this first problem, to model the interaction between atoms, use the supplied 12-6 Lennard-Jones (LJ) potential (see input file below).

- 1 By using the minimizer in LAMMPS, which performs atomic relaxation, one can optimize the structure. To optimize both the unit cell and atomic positions, the *fix box/relax* command must be used before calling *minimize*. Record the total energy and the lattice constant after this full optimization
- 2 By selecting different values for the lattice parameter near its equilibrium value and performing an energy calculation for each, one can determine the optimal lattice constant. To do this, manually set the unit cell parameters using the *lattice* command, then use *run 0* to perform a single-point energy evaluation in LAMMPS. Record the total energy obtained for each lattice constant and plot total energy vs lattice constant. What type of curve do you obtain?

For the problem A1, you will use the input file provided below. To keep your directories organized, consider structuring your files properly so they are easy to locate later. To begin this exercise, follow these steps:

1. Create a directory using the `mkdir` command.
2. Open a new file with the editor of your choice.
3. Name the file with a `.in` extension.
4. Copy and paste the provided script into the newly created file. Sometimes when you copy-paste it adds blank spaces, make sure your file looks exactly as the one in the handout.

For the script to run, you need to replace the placeholders for dimensionality, atomic mass, and lattice constant with the correct values.

```

# This section defines the units that you are using. The dimensions of your
# simulation and the periodic boundary conditions.
units      metal
dimension  <dimension_number>
boundary   p p p

# Define the initial lattice structure and size.
atom_style atomic
lattice    fcc <lattice_constant> # Initial lattice constant
region     box block 0 1 0 1 0 1 # Define box with 1 unit cell per side

# Create the simulation box and populate it with atoms.
create_box 1 box
create_atoms 1 box

# Define atomic mass for the element being simulated.
mass        1 <atomic_mass_of_Ag>

# Define interatomic potential using Lennard-Jones potential.
pair_style lj/cut 20.0
pair_coeff 1 1 0.341 2.648 # LJ parameters for Ag ( $\epsilon$  in eV,  $\sigma$  in Å)

# Define neighbor list settings to improve computational efficiency.
neighbor    0.3 bin
neigh_modify delay 5 every 1

# Allow isotropic box relaxation and atomic position relaxation.
fix 1 all box/relax iso 0.0 vmax 0.001

# Monitor thermodynamic properties with energy output at every step.
thermo 1
thermo_style custom step temp pe lx ly lz press

# Minimize energy to optimize geometry. Cut-offs and steps from energy and forces.
minimize 1.0e-6 1.0e-8 1000 10000

# Save the final structure and trajectory.
write_data optimized_structure.data

```

Read the script before you run the simulation and understand what does each keyword do. Make sure you know what are the units that are specified here with `units metal` command. To run this script you can do.

```
lmp_mpi -in filename.in
```

For the problem A2, the main part of the input script remains the same. The key adjustment needed is the lattice constant, which should be modified each time you execute the script. To facilitate analysis, you can define a variable named `pot_e` (potential energy) and store the `pe` (LAMMPS keyword for potential energy) of the system in it. At the end of the script, you can append this value to a file using this in the end of the file.

```
print "{$pot_e}" append energy_volume.dat
```

B. (5 points) Repeat these calculations using the supplied Embedded Atom Model (EAM) potential (the potential is provided in the input file `Ag_u3.eam`).

1. How do the EAM and Lennard-Jones (LJ) lattice constants compare with experimental values? (The experimental lattice parameter for Ag is 4.09 Å.) Is this expected? Why?
2. The energies of the two calculations (A and B) will be different. Is this a problem? Justify your answer.

The procedure is the same. However, you are going to use the `ag.eam.minimize.in` input file. Make sure to create a new directory and copy there all the necessary files (lammps input and potential). To do this you can either use the command line (remember the commands `mkdir`, `cp` and `cd`) or the file explorer.

Some words regarding the Lennard-Jones potential.

The Lennard-Jones (LJ) potential is a widely used mathematical model that describes the interaction between a pair of neutral atoms or molecules. It is expressed as:

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

where ϵ represents the depth of the potential well (measuring the strength of the interaction), σ is the finite distance at which the interparticle potential is zero, and r is the interatomic distance. The repulsive term $(\frac{\sigma}{r})^{12}$ accounts for the Pauli exclusion principle, while the attractive term $(\frac{\sigma}{r})^6$ models van der Waals interactions.

Another common form of the LJ potential uses A and B parameters:

$$V(r) = Ar^{-12} - Br^{-6}$$

where A and B are alternative parameters related to ϵ and σ by:

$$A = 4\epsilon\sigma^{12}, \quad B = 4\epsilon\sigma^6$$

These two forms are mathematically equivalent, and conversion between them is straightforward.

Why do E(V) curves matter?

You might wonder why one should use the second approach based on the energy-volume curve, instead of just running the energy minimization leveraging common minimization routines.

The energy-volume curve gives you access not only to the equilibrium lattice constant but also to the bulk modulus. The bulk modulus measures the resistance of a material to compression. Hence, it can be of interest to design a material with a desired bulk modulus. In the last part of exercise 1, you will perform a so-called Birch-Murnaghan fit (see details below) to obtain the bulk modulus of FCC-Ag.

C. Bulk modulus (5 points):

- 1 Calculate the bulk modulus of FCC-Ag using the LJ potential (the same that you have already used in part 2. of this exercise). Explain how you performed the fit, and visualize your energy vs. volume datapoints together with the fit.
- 2 Compare your results with the literature (experimental and/or other theoretical values) and comment on your findings.

The bulk modulus can be related to a number of thermodynamic quantities, such as the volume V or the pressure P , through a so-called equation of state. A very well-known one is the third-order Birch-Murnaghan isothermal equation of state:

$$P(V) = \frac{3B_0}{2} \left[\left(\frac{V_0}{V} \right)^{\frac{7}{3}} - \left(\frac{V_0}{V} \right)^{\frac{5}{3}} \right] \left\{ 1 + \frac{3}{4}(B_0' - 4) \left[\left(\frac{V_0}{V} \right)^{\frac{2}{3}} - 1 \right] \right\},$$

where B_0 is the bulk modulus, B_0' its derivative, V_0 the equilibrium volume and E_0 the equilibrium energy. Integration of the above expression yields:

$$E = E_0 + \left(\frac{9}{8} \right) B_0 V_0 \left[\left(\frac{V_0}{V} \right)^{2/3} - 1 \right]^2 + \left(\frac{9}{16} \right) B_0 V_0 (B_0' - 4) \left[\left(\frac{V_0}{V} \right)^{2/3} - 1 \right]^3.$$

To find B_0 , this is the equation you will need to fit your numerical results to.

Note: You may reuse the calculations that you performed in the 2. part of this exercise in order to extract a set of values of the lattice parameters and cell volumes.

Hint for the fitting: Be careful with the fitting tools you choose. Moreover, remember that the

most important fitting region is the one around the equilibrium lattice constant. Therefore, the volume range should be chosen with care: a too small volume range can give inaccurate fit results because of numerical noise, while a too large volume range might not be well described by the Birch-Murnaghan equation of state.

Exercise 2

This problem deals with the *vacancy formation energy*, that can be obtained from total energy calculations, defining the required energy to form a vacancy in a material. In order to estimate the vacancy formation energy of FCC-Ag, we will study the effects of supercell convergence, relaxation and shortcomings of different empirical potentials.

Note: Before starting this exercise, please run the following command:

```
pip install --user ase
```

Problem 2 (25+10 optional points): Calculation of the vacancy formation energy.

The vacancy formation energy in a crystal is defined as the cost of removing an atom and bringing it at infinity, where it no longer interacts with the other atoms. Using equations:

$$E_{f,v} = E_{i,N-1} - \left(\frac{N-1}{N} \right) E_{i,N} \quad (1)$$

where $E_{f,v}$ is the the vacancy formation energy, N the number of sites in the cell and $E_{i,N-1}$ and $E_{i,N}$ the energy of the crystal with $N-1$ and N atoms respectively.

A. (12.5 points) Compute the vacancy formation energy for FCC-Ag using the LJ potential. In particular, compute the vacancy formation energy as a function of different supercell sizes. Use the provided `create_supercell.py` script to construct the supercell, see further hints below.

Important: For this part, *do not relax* the supercell after removing the atom.

- What is the size (in the x,y,z directions) of the supercell that you used to achieve convergence? **Note: Remember to state the threshold that you used to define convergence.**
- The cohesive energy is defined as the difference between the bulk energy per atom and the energy of one isolated atom. For the potentials used, the energy of an isolated atom is simply zero. What is the ratio of vacancy formation energy to cohesive energy? Is this what you would expect? Justify your answer.

B. (12.5 points) Repeat the above calculations, but this time, relax the supercells after removing one atom.

- Again, what is the size (in the x,y,z directions) of the supercell that you used to achieve convergence? Is it the same as part in A.? **Note: Remember to state the threshold that you used to define convergence.**

- How does the vacancy formation energy change when compared to the one obtained without relaxation in part A.?
- The experimental vacancy formation energy for FCC-Ag is 1.11 eV. How do the values obtained from parts A. and B. compare to the experimental one? Comment on your results.

C. (optional 10 points) Compute the vacancy formation energy using the provided EAM potential. Compare the results with those obtained for the LJ potential and also with the experimental value. Comment on the accuracy of the different potentials used in your simulations.

Note: To save time, you may run calculations in part C. either by relaxing the supercells, or by fixing the supercell geometry. In either case, justify your choice of calculation. Remember that you still need to converge the size of the supercell.

You will need to copy over the EAM potential file and also take the parameters for the LJ potential from the previous exercise.

Note: As mentioned in the beginning, LAMMPS allows to create supercells directly in the input file, by setting the region parameter accordingly. Moreover, one could also delete an atom afterwards, using the `delete_atoms` command (https://docs.lammps.org/delete_atoms.html). However, in this tutorial, we provide a script to create the supercells and to manually remove an atom from the created file, as described in the following paragraph.

How to run the `create_supercell.py` script

To build the supercells, which you will need to calculate the vacancy formation energies, you will use the simple `create_supercell.py` script that we provide. The script relies on the **Atomic Simulation Environment (ASE)** (<https://wiki.fysik.dtu.dk/ase/>) library, which is a very useful toolkit to create and manipulate crystal structures.

Use the following command to run the script:

```
python3 create_supercell.py <element> <lattice_constant> <nx> <ny> <nz>
```

You need to specify the chemical element, the lattice constant (use the optimized one from the previous exercise), and the number replicates in each direction, specified by nx , ny , nz .

Either you simply create the supercells by running this command multiple times and change the size accordingly, or you wrap this in a `bash` or `Python` script (you can of course import the underlying function into your own scripts).

The script directly stores the generated supercell in a `.lmp` file: `element_supercell_<nx>x<ny>x<nz>.lmp`.

Hint: Since you are dealing with a cubic system, it is sufficient to only consider the cases $nx = ny = nz$.

Next, you can simply remove one atom from each supercell, to simulate the vacancy.

How to read the supercell files and run the calculations

You can use the `read_data` command in your LAMMPS input file, to read the structural information when running your calculations. This avoids the necessity of manually specifying the supercells. Remove the following lines that you copied from exercise 1:

```
# Define the initial lattice structure and size.
atom_style atomic
lattice fcc 4.45 # Initial lattice constant
region box block 0 5 0 5 0 5 # Define box with 5 unit cells per side

# Create the simulation box and populate it with atoms.
create_box 1 box
create_atoms 1 box

# Define atomic mass for the element being simulated.
mass 1 107.8682
```

and replace them with the following line:

```
# Read the data file
read_data <file_name_of_your_supercell>
```

Exercise 3

In general, studying surfaces is important for many applications, such as catalysis or adhesion. The goal of this exercise is to demonstrate the calculation of the so-called surface energy (see the definition below). In reality, many different surface terminations do exist (depending on how one cuts the bulk material). To guide the analysis and focus on the most likely ones, the estimate of the surface energy is a relevant quantity.

Note: Before starting this exercise, please run the following command:

```
pip install --user pymatgen
```

Problem 3 (20 points): Calculation of the surface energy.

Compute the surface energy of a solid (100) surface of FCC-Ag using the LJ potential. The surface energy is defined as the energy cost of creating a termination of the bulk crystal along a crystallographic plane. To compute it, you will need to build a supercell containing a slab (cut) of the bulk and some vacuum, see Figure 1 for an exemplary visualization. To create these slabs, we provide a Python script `create_slab.py` (the usage of the script is described below).

The surface energy can be computed according to the formula:

$$\gamma_{\text{slab}} = \frac{E_{\text{slab}} - N E_{\text{bulk}}}{2 A_{\text{slab}}} \quad (2)$$

E_{slab} is the energy of the slab, E_{bulk} the bulk energy *per atom*, N number of atoms in the surface slab and A_{slab} is the surface area. The factor 2 comes from the fact that 2 surfaces are created due to the periodic boundary conditions.

- A. (3 points) Briefly comment why you need to converge the thickness of the vacuum and the slab.
- B. (10 points) Comment on your results studying the convergence of both the thickness of the slab and the thickness of the vacuum. **Note: Remember to state the threshold that you used to define convergence.**
- C. (4 points) Why are we doing single point calculations? What do you think will happen if we relax the entire slab (as done for the bulk before)?
- D. (3 point) What do you notice? Is the surface energy positive or negative? What does this mean and how could you explain it?

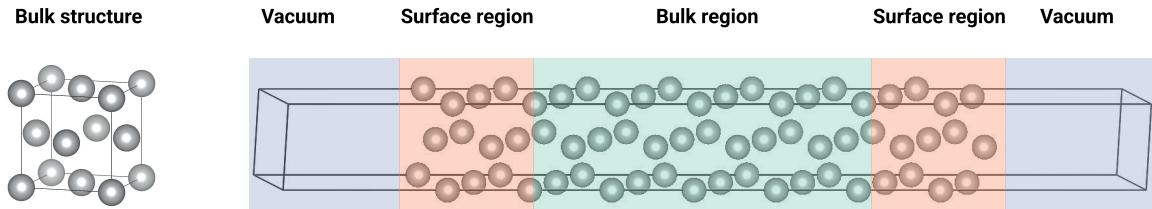


Figure 1: Visualization of the FCC bulk structure and a surface slab (with seven layers and two layers of vacuum).

How to run the `create_slab.py` script

To run the script, you need to provide the chemical element, the lattice constant (the one you obtained in exercise), the number of layers of your material (i.e., repetitions of the unit cell) and the number of vacuum layers:

```
python3 script.py <element> <lattice_constant> <num_layers> <num_vacuum_layers>
```

The number of vacuum layers is internally expressed in terms of the lattice constant, and the slab will be centered in the resulting supercell, see again Figure 1 for reference. The script uses the `pymatgen` (<https://pymatgen.org>) library, which is another very useful and commonly adopted library to work with structures and simplifies many analyses.

The script will produce a data file for the resulting slab, which can be directly read by LAMMPS, as you have already seen in the context of exercise 2.

Exercise 4

Problems 4 and 5 require just a little calculation and are more focused on the theory. Try to be concise and give your answer as clearly as possible. There is no need to write too much, but it is important to be exhaustive.

Problem 4 (5 points): Choice of empirical potentials.

The choice of the empirical potential critically affects the results. In these exercises, we have used the Lennard-Jones and EAM potentials.

1. For what other types of problems would you use Lennard-Jones potentials? When would you use EAM instead? When would neither be appropriate?
2. Can you give an example of yet another empirical potential and explain when you would expect it to perform well?

You can refer to the LAMMPS documentation for a list of existing potentials and after you pick one, you can do some literature search.

Exercise 5

Problem 5 (5 points): Minimum of the Lennard-Jones potential.

Given the simple form of the Lennard-Jones potential, one might wonder why we use computers to solve what appears to be a pen-and-paper problem. The following exercise will help to clarify this issue.

1. Calculate the distance where the $12 - 6$ form of the Lennard-Jones potential reaches the minimum value, i.e., express r_0 in terms of A and B.

$$U_{\text{LJ}}(r) = \frac{A}{r^{12}} - \frac{B}{r^6} \quad (3)$$

2. Insert the values of A and B appearing in the LAMMPS input file for the Ag Lennard-Jones potential and compare the result with your answers of Problem 1. Explain the discrepancies.

Exercise 6

In the final exercise of this lab, we will focus on calculating the structural properties of a more complex binary alloy. We have previously discussed the importance of these calculations, as they serve as the initial step in a series of workflows aimed at determining more interesting properties.

Problem 6 (30 + 5 points): Structural properties of the $L1_0$ CuAu binary phase.

In the previous exercises, you worked with single-element structures, which are arguably the simplest cases. In the rest of this exercise, you will study a more complex material, namely the $L1_0$ CuAu binary phase.

To do this, you will need to construct the empirical potential that you will use. This is typically done by selecting a mathematical form based on physical and chemical insights (e.g., two-body versus many-body interactions, covalent vs. ionic bonds) and then determining the parameters to be used in these expressions. These parameters are usually obtained by fitting them to reproduce known properties, such as lattice constants.

In this exercise, you will set up from scratch a calculation for a gold-copper alloy (CuAu) in the $L1_0$ structure, by building the input file and the appropriate empirical potentials with the help of the literature, and the previously created files.

- A. (10 points) Firstly, look up in the literature for bulk CuAu in the $L1_0$ phase. Then, look for the Lennard-Jones parameters for the elemental Au and Cu, useful resources for this can be [1, 3, 4, 6, 7, 8], of course not every set of Lennard-Jones parameters will give the same results, so try to choose wisely. As you can see, the parameters that you find correspond to elemental materials, for this exercise you need to use a *mixing rule*[2] to calculate the average Lennard-Jones parameters. For this exercise you need to use the standard geometric. However, if you have time, feel free to try any of the rest to experience for yourself if they give different results.
- B. (10 points) Compute the equilibrium lattice parameter (using the LAMMPS optimizer) for CuAu. What are the initial lattice vectors and cell volume for this compound? What are the final values after the optimization? How can you explain the large discrepancies from the experimental/theoretical value (the one you found in the literature)? Also, a pre-fitted EAM potential for the CuAu alloy is provided to you. Modify your LAMMPS script to use this potential and recalculate the lattice parameters. How do they compare to those obtained using the Lennard-Jones potential?
- C. (10 points) Consider, the FCC-Au crystal. In the paper [5] an EAM potential particularly suited to describe simple metals and their alloys has been derived. After reading the referenced paper, answer the following questions about EAM potentials.
 1. Is F a pair-functional or a cluster one or some other type of many-body potential? Briefly justify your answer. Is it taking into account many-body effects? If yes, explain how.
 2. Give an expression for ρ_i assuming that only nearest neighbors contribute to its value (think how many neighbors there are in an FCC lattice).
- D. (optional 5 points) Show that in a monoatomic crystal with N atoms, $E_{\text{tot}} = NE$, where

$$E = F(\rho) + \frac{1}{2} \sum_m \phi(r_m), \quad (4)$$

based on the expressions in the paper.

In this exercise, you will set up *from scratch* a calculation for binary Cu-Au alloy, by building the input file and the appropriate Lennard-Jones potential with the help of the literature.

In problem A, you have to set up a simulation for CuAu in the so-called $L1_0$ phase. Begin by finding out (either through a literature search or using a crystallographic database) what is the structure of this compound (basis vectors, primitive vectors). Having done that you can start editing your LAMMPS input file.

Having set up the first part of your simulation input script, where you define the units that you are going to use throughout your simulation (**ALWAYS** careful with the units), the periodic boundary

conditions, etc. Now it is time to set up the simulation box and position the atoms inside it. Your script should look like this:

```
region      box block xlo xhi ylo yhi ylo yhi
create_box  <num_types> box

# Define atomic positions for L1_0 CuAu
create_atoms 1 single X Y Z
create_atoms 2 single X Y Z
```

You can obtain all this information from the literature on the specific material phase you are trying to simulate. Next, you need to edit the empirical potential section of the file. If you are using the 12-6 LJ potential, your input file will contain the following lines:

```
pair_style <style_of_the_potential> <cutoff_value> -> For metals a value of 8-10 Å is
      recommended.
pair_coeff <element 1> <element 1> <epsilon> <sigma> -> Make sure they are in the correct
      units.
pair_coeff <element 2> <element 2> <epsilon> <sigma>
pair_coeff <element 1> <element 2> <epsilon> <sigma>
```

The rest of the file can be created using the one from Exercise 1 as a prototype. When replacing the potential, you have to adapt the values in the section above accordingly.

In Problem C, consider the FCC-Au crystal. Read the following paper[5], especially Sections 2 and 3 (the paper can be downloaded from Moodle). In the embedded-atom method (EAM), the total energy is expressed as:

$$E_t = \sum_i F(\rho_i) + \sum_{i>j} \phi(r_{ij}) \quad (5)$$

where $\phi(r_{ij})$ is a two-body potential between atoms i and j , r_{ij} is the distance between atoms i and j , and $F(\rho_i)$ is the embedding energy that depends on the quantity:

$$\rho_i = \sum_{j \neq i} f(r_{ij}) \quad (6)$$

where $f(r_{ij})$ represents the electron density at atom i due to all other atoms j . (This is where the term "embedding" in the embedded-atom method comes from.) In the paper, you will find an analytic formula for the density function $f(r)$ (Eq. 4), as well as for the embedding function F (Eq. 6).

References

- [1] Mansoor H Alshehri. Investigation of interaction of noble metals (cu, ag, au, pt and ir) with nanosheets. *Micromachines*, 12(8):906, 2021.
- [2] DL-SDG. Force field ff10 documentation, 2025. https://dl-sdg.github.io/RESOURCES/FORCE_FIELD/ff10.html.
- [3] NIST Center for Theoretical and Computational Materials Science (CTCMS). Nist interatomic potentials repository, 2025. <https://www.ctcms.nist.gov/potentials/>.
- [4] Hendrik Heinz, RA Vaia, BL Farmer, and RR Naik. Accurate simulation of surfaces and interfaces of face-centered cubic metals using 12- 6 and 9- 6 lennard-jones potentials. *The Journal of Physical Chemistry C*, 112(44):17281–17290, 2008.
- [5] RA Johnson. Alloy models with the embedded-atom method. *Physical Review B*, 39(17):12554, 1989.
- [6] Peter Schwerdtfeger, Antony Burrows, and Odile R Smits. The lennard-jones potential revisited: Analytical expressions for vibrational effects in cubic and hexagonal close-packed lattices. *The Journal of Physical Chemistry A*, 125(14):3037–3057, 2021.
- [7] SimonEnsemble. Porousmaterials.jl - force field documentation, 2025. https://github.com/SimonEnsemble/PorousMaterials.jl/blob/master/docs/src/force_field.md.

[8] Pham Duy Tan, Pham Dinh Tam, and Bui Duc Tinh. Thermal expansion of metals cu, au and ag at higher temperature and under pressure. In *Journal of Physics: Conference Series*, volume 1506, page 012016. IOP Publishing, 2020.