MODULE 2:
MOLECULAR DYNAMICS

LAMMPS Walkthrough
SOFTWARE

- We will be using LAMMPS to perform classical molecular simulations to predict materials properties and behavior.


- We will be visualizing MD trajectories using the OVITO visualization package.

- OVITO is freely available from [www.ovito.org](http://www.ovito.org)
LAMMPS + OVITO

EWS PRE-INSTALLATION

- A full installation of LAMMPS 29Sep2021 release is available on EWS Linux at: /class/mse404pla/lammps-29Sep2021/bin/lmp_serial

- The LAMMPS documentation is available in online at docs.lammps.org/Manual.html

- An installation of OVITO 2.6.1 is on EWS Linux at: module load ovito

- This may require you to correct your library path: export LD_LIBRARY_PATH=/lib64/:$LD_LIBRARY_PATH

- The Ovito documentation is available in online at http://www.ovito.org/manual/
Tutorial 1: Al cohesive energy
We will use LAMMPS to estimate the Al fcc cohesive energy, $E_{\text{cohe}}$, and lattice parameter, $a$.

$$E_{\text{cohe}} = E_{\text{solid}} - \sum_{\text{atoms}} E_{\text{isolated}}$$

Experimentally, $E_{\text{cohe}} = -3.39 \text{ eV/atom}^*$ and $a = 4.0495 \text{ Å}^*$.

**Strategy:** We shall use a modern EAM potential for Al and optimize $E_{\text{cohe}}$ as a function of $a$. 

Be careful! In this first tutorial we are NOT performing conventional molecular dynamics (i.e., integrating $F=ma$).

Rather we are performing a potential energy minimization, to find the lowest potential energy crystal coordinates.

We achieve this by performing conjugate gradient (or steepest descent) minimization of $PE$ wrt atomic coords.

Accordingly, there are no atom velocities or temperature!
Tutorial 1: Al cohesive energy

2. Copy LAMMPS input file
/class/mse404pla/LAMMPS/Al_fcc.in
Tutorial 1: Al cohesive energy

For style metal, these are the units:

- mass = grams/mole
- distance = Angstroms
- time = picoseconds
- energy = eV
- velocity = Angstroms/picosecond
- force = eV/Angstrom
- torque = eV
- temperature = Kelvin
- pressure = bars
- dynamic viscosity = Poise
- charge = multiple of electron charge (1.0 is a proton)
- dipole = charge*Angstroms
- electric field = volts/Angstrom
- density = gram/cm^3

- # specifies a comment
- x, y, z periodic boundaries
Tutorial 1: Al cohesive energy

- Specify **fcc lattice** with $a=4$ Å

- Define **cuboidal block** labeled **box** holding one **lattice cell**

- Create **box** with **1** atom type

```plaintext
# Initialize Simulation
units metal
dimension 3
boundary p p p
atom_style atomic

# Create Atoms
lattice fcc 4
region box block 0 1 0 1 0 1 units lattice
create_box 1 box

lattice fcc 4 orient x 1 0 0 orient y 0 1 0 orient z 0 0 1
create_atoms 1 box
replicate 2 2 2

# Define Interatomic Potential
pair_style eam/alloy
pair_coeff * * Al99.eam.alloy Al
neighbor 2.0 bin
neigh_modify delay 10 check yes

# Define Settings
compute eng all pe/atom
compute eatoms all reduce sum c_eng

# Dump Options
dump 1 all atom 1 dump.relax

# Run Minimization
reset_timestep 0
fix 1 all box/relax iso 0.0 v_max 0.001
thermo_style custom step pe lx ly lz press pxx pyy pzz c_eatoms
min_style cg
minimize 1e-25 1e-25 5000 10000

variable natoms equal "count(all)"
variable teng equal "c_eatoms"
variable a equal "lx/2"
variable ecoh equal "v_teng/v_natoms"

print "Total energy (eV) = ${teng};"
print "Number of atoms = ${natoms};"
print "Lattice constant (Angstroms) = ${a};"
print "Cohesive energy (eV/atom) = ${ecoh};"
print "All done!"
```
Tutorial 1: Al cohesive energy

- Specify fcc lattice orientation
- Create atoms of type 1 on lattice sites within box
- Replicate domain by 2x2x2 in x,y,z
  [replicate 1 1 1 would be more parsimonious for this trivially periodic system]
Tutorial 1: Al cohesive energy

- Define form of pairwise interaction potential as **eam/alloy**
  [misnomer; EAM is n-body]

- Use **Al** block of **Al99.eam.alloy** - specifies cutoff, F, ρ, and Φ - for all pairs
  [for one atom type, 1 1 fine]

- **2 Å skin thickness** for neighbor list binning

- Build neighbor list every **10 steps**, but **check** atom moved more than half skin thickness
Tutorial 1: Al cohesive energy

- Define **computes** - quantities recalculated every time step [cf. **variables**, which evaluate a formula when called]

- Reference computes as **c_<name>**

- **c_eng** defined over all atoms to compute potential energy per atom

- **c_eatoms** performs sum reduce of c_eng vector over all atoms [alternatively: compute eatoms all pe]
Tutorial 1: Al cohesive energy

• A **dump** specifies how to write output data

• Tag dump with id 1 to write to **dump.relax** every 1 steps the coords of all of the atoms

• Dump format:

```plaintext
ITEM: TIMESTEP
0
ITEM: NUMBER OF ATOMS
32
ITEM: BOX BOUNDS pp pp pp
0 8
0 8
0 8
ITEM: ATOMS id type xs ys zs
1 1 0 0 0
2 1 0.25 0.25 0
3 1 0.25 0 0.25
4 1 0 0.25 0.25
```

---

```plaintext
# Initialize Simulation
units metal
dimension 3
boundary p p p
atom_style atomic

# Create Atoms
lattice fcc 4
region box block 0 1 0 1 0 1 units lattice
create_box 1 box
lattice fcc 4 orient x 1 0 0 orient y 0 1 0 orient z 0 0 1
create_atoms 1 box
replicate 2 2 2

# Define Interatomic Potential
pair_style eam/alloy
pair_coeff * * Al99.eam.alloy Al
neighbor 2.0 bin
negh_modify delay 10 check yes

# Define Settings
compute eng all pe/atom
compute eatoms all reduce sum c_eng

# Dump Options
dump 1 all atom 1 dump.relax

# Run Minimization
reset_timestep 0
fix 1 all box/relax iso 0.0 v0max 0.001
thermo_style custom step pe lx ly lz press pxx ppy ppz ceatoms
min_style cg
minimize 1e-25 1e-25 5000 10000

variable natoms equal "count(all)"
variable teng equal "c_eatoms"
variable a equal "lx/2"
variable ecoh equal "v_teng/v_natoms"
print "Total energy (eV) = ${teng};"
print "Number of atoms = ${natoms};"
print "Lattice constant (Angstroms) = ${a};"
print "Cohesive energy (eV/atom) = ${ecoh};"
print "All done!"
```
Tutorial 1: Al cohesive energy

- Reset time steps to 0
- A **fix** is an operation applied at every time step
- Define fix 1 operating on **all** atoms **relaxes box** to an external **isotropic pressure** of 0.0 bar with a 0.1% maximum fractional volume change per step
Tutorial 1: Al cohesive energy

- Output **thermodynamic info** to screen every 10 steps [use `fix / dump` for file write]

- Customize thermo output

- Perform energy minimization by **conjugate gradient**

- **Minimize** $E = E_{FF} + E_{fix}$ with $\Delta E = 10^{-25}$ (i.e., 1 part in $10^{25}$) and $\Delta f = 10^{-25}$, and a maximum of 5000 iterations and 10000 energy evaluations
Tutorial 1: Al cohesive energy

- Define **variables** as formulas evaluated when called [cf. *computes*, simulation values recomputed each step]

- Reference variables as \( v_{<\text{name}>} \)

- \( \text{natoms} = \# \text{ atoms} \)
  \( \text{teng} = \text{total PE (c\_eatoms)} \)
  \( a = \text{lattice parameter} \)
  (box side in \( x \) divided by \# \times \text{replicas} = 2)
  \( \text{ecoh} = \text{cohesive energy /atom} \)
Tutorial 1: Al cohesive energy

- Print terminal output to screen
Tutorial 1: Al cohesive energy

3. Let’s run! \texttt{lmp\_serial < Al\_fcc.in}

```
tuckermuck:1_al_cohesive_energy alf $ ./lmp_mac < Al_fcc.in
LAMMPS (1 Feb 2014)
Lattice spacing in x,y,z = 4.44
Created orthogonal box = (0 0 0) to (4.44)
1 by 1 by 1 MPI processor grid
Lattice spacing in x,y,z = 4.44
Created 4 atoms
Replicating atoms ...
  orthogonal box = (0 0 0) to (8 8 8)
  1 by 1 by 1 MPI processor grid
32 atoms
WARNING: Resetting reneighboring criteria during minimization (.../min.cpp:173)
Setting up minimization ...
Memory usage per processor = 3.39896 Mbytes
Step PotEng Lx Ly Lz Press Pxx Pyy Pzz eatoms
0  -107.3423  0  0  0  29590.11  29590.11  29590.11  29590.11  29590.11
10  -107.5123  0.00  0.00  0.00  5053.953  5053.953  5053.953  5053.953  5053.953
14  -107.52  0.1  0.1  0.1  2.726913  2.726913  2.726913  2.726913  2.726913
Loop time of 0.00931406 on 1 procs for 14 steps with 32 atoms

Minimization stats:
Stopping criterion = linesearch alpha is zero
Energy initial, next-to-last, final = -107.3422985373
-107.51999962
-107.51999962
Force two-norm initial, final = 28.3679 0.02680085
Force max component initial, final = 28.3679 0.00268005
Final line search alpha, max atom move = 0.00145753 3.96625e-06
Iterations, force evaluations = 14 23

Pair time (%) = 0.00081649 (64.5950)
Neigh time (%) = 0 (0)
Comm time (%) = 0.00095582 (10.2621)
Outpt time (%) = 0.000859677 (9.13326)
Other time (%) = 0.00149107 (16.0088)
NLocal: 32 ave 32 max 32 min
Histogram: 1 0 0 0 0 0 0 0 0
Mhost: 1067 ave 1067 max 1067 min
Histogram: 1 0 0 0 0 0 0 0 0
Neighs: 2240 ave 2240 max 2240 min
Histogram: 1 0 0 0 0 0 0 0 0
Total # of neighbors = 2240
Ave neighs/atom = 70
Neighbor list builds = 0
Dangerous builds = 0
Total energy (eV) = -107.51999962832;
Number of atoms = 32;
Lattice constant (Angstroms) = 4.05;
Cohesive energy (eV/atom) = -3.35999998135;
All done!
```
4. Analysis

<table>
<thead>
<tr>
<th></th>
<th>LAMMPS</th>
<th>Expt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lattice constant / Å</td>
<td>4.05</td>
<td>4.0495</td>
</tr>
<tr>
<td>Cohesive energy / eV/atom</td>
<td>-3.36</td>
<td>-3.39</td>
</tr>
</tbody>
</table>

We should be shocked if these quantities did not agree — EAM FF parametrized wrt experimental data.

Q. What about if we were studying a new material with experimentally unknown $E_{\text{cohe}}$ and $a$?
Tutorial 1: Al cohesive energy

5. Visualization in OVITO
Tutorial II: Al crack propagation
OK, now to the real MD simulations!

An important mode of materials failure is propagation of exterior cracks.

The stress field at the crack tip in an amorphous material can be modeled by continuum equations or FEM.
Tutorial II: Al crack propagation

Using MD simulation, we can visualize the stress field near the crack tip with **atomistic resolution**

**Strategy:** Construct an exterior crack in a semi-periodic fcc Al xtal and measure atomic stress upon deformation

http://www.adglass.eu/at_mod_cohesion.html
1. Download **Al99.eam.alloy** EAM potential from NIST Interatomic Potentials Repository Project (http://www.ctcms.nist.gov/potentials)
2. Copy `Al_crack.in`, `Al_eq.m`, and `Al_crack.m` from `/class/mse404pla/LAMMPS/`
Tutorial II: Al crack propagation

- Initializing system
- Building geometry
- Defining force field
Tutorial II: AI crack propagation

- Defining groups: topWall, bottomWall, and mobile
- Eliminating a notch of atoms to form the crack
- Specifying computes (incl. per atom stress tensor)
Tutorial II: Al crack propagation

- topWall
- botWall
- periodic in y
Tutorial II: Al crack propagation

- NVT equilibration of mobile atoms, topWall & botWall frozen
- Instrumentation, thermo, and output
- MD simulation

```plaintext
# EQUILIBRATION

# reset timer
reset_timestep 0

# 2 fs time step
timestep 0.002

# initial velocities
velocity mobile create 300 12345 mom yes rot yes
velocity boundary set 0.0 0.0 0.0

# thermostat + barostat
fix 1 mobile npt temp 300 300 1 y 0 0 1 drag 1.0

# instrumentation and output
variable s1 equal "time"
variable s2 equal "lx"
variable s3 equal "ly"
variable s4 equal "lz"
variable s5 equal "vol"
variable s6 equal "press"
variable s7 equal "pe"
variable s8 equal "ke"
variable s9 equal "etotal"
variable s10 equal "temp"
fix writer all print 250 "${s1} ${s2} ${s3} ${s4} ${s5} ${s6} ${s7} ${s8} ${s9} ${s10}" file Al_eq.txt
screen no

# thermo
thermo 500
thermo_style custom step time cpu cpurenain lx ly lz press pe temp

# dumping trajectory
dump 1 all atom 250 dump.eq.lammpstrj

# MD simulation
run 15000

# clearing fixes and dumps
unfix 1
undump 1

# saving equilibrium length for strain calculation
variable tmp equal "lz"
variable lz0 equal ${tmp}
```
• NVT integration of mobile atoms, topWall pulled up, botWall frozen in place
• Instrumentation, thermo, and output (incl. custom strain calculation and atom stress cfg dump)
• MD simulation
3. Let's run!  `lmp_serial < Al_crack.in`

If execution is very slow on a serial machine, reduce system size:
```
region whole block 0 15 0 2 0 15
```
4. Analyze approach to equilibrium using Al_eq.m

>> Al_eq('Al_eq.txt')
Tutorial II: Al crack propagation

- Box side $y$ / Angstroms
  - Time vs. box side $y$ over 30 ps

- $P_z$ / bar
  - Time vs. $P_z$ over 30 ps

- Temperature $T$ / K
  - Time vs. temperature over 30 ps

- Energy / eV
  - Time vs. energy over 30 ps

Legends:
- PE
- KE
- E
5. Analyze crack formation using `Al_crack.m`

`>> Al_crack('Al_crack.txt')`
Tutorial II: Al crack propagation

![Graph showing stress-strain relationship with various stress components.](image)
Tutorial II: Al crack propagation

6. Visualization in OVITO
Tutorial II: Al crack propagation

\[ \sigma_{zz} \]

+200 GPa

-200 GPa
Tutorial II: Al crack propagation

Extension I
- Change crystal orientation to explore the effect of cracks in different crystallographic faces

Extension II
- {Reduce / remove / enlarge} initial crack to explore the impact of the initial imperfection size

Extension III
- Explore the effect of system size in x,y,z on atomic stresses