Response of copper nanowire under tensile loading

LAMMPS REPORT

Praveenkumar Hiremath

May 3, 2020

1 Aim and objectives

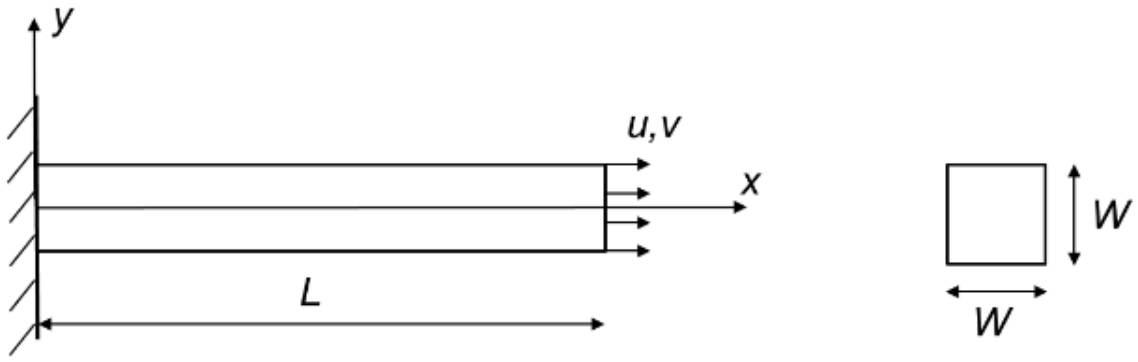
For the design and manufacturing of better quality materials, it has been witnessed to be beneficial to first investigate the materials for the better understanding of their thermal, mechanical, optical, magnetic properties and the mechanisms responsible for these properties. These properties in turn are the results of electronic configurations of materials. The techniques such as density functional theory (DFT), hartree fock (HF), dynamical mean field theory (DMFT), to name a few, can predict the electronic properties of materials to an acceptable degree of accuracy. Then the technique of molecular dynamics has enabled us to bridge the gap between electronic scale to nanoscale study of materials through the usage of interatomic potential functions like Lennard-Jones (LJ) potential, morse potential, embedded atom method (EAM), modified embedded atom method (MEAM) and so on.

In this work, EAM and 6-12 LJ potentials are used to study the response of single crystal copper nanobeam under tensile loading, using molecular dynamics technique. To this end, LAMMPS [1], the molecular dynamics package has been used.

2 Problem description and LAMMPS

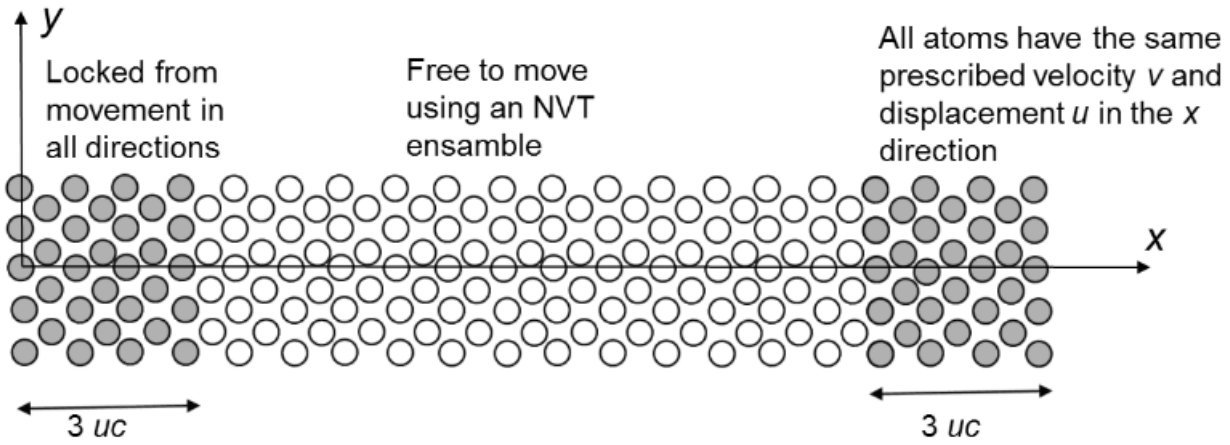
In nanoelectromechanical systems, nanowires i.e wires with diameter of the order of a few nm, are used. This is one of the reason for investigating nanowires for their mechanical properties. In this work, a tensile load is applied on a single crystal copper (fcc crystal structure) nanobeam of two different geometries, along the directions [100] and [110] (x-axis). For one geometry, the cross section (parallel to yz-plane) has an area = lattice parameter*(6 unit cells * 6 unit cells) $Å^2$. For the other geometry, has an area = lattice parameter*(30 unit cells * 30 unit cells) $Å^2$. The lattice parameter of fcc copper used here is 3.615 Å.

Figure 1: Schematics of the copper nanobeam considered here. (Geometry G1 → W = 6 unit cells, geometry G2 → W = 30 unit cells



These lammps simulations of tensile loading are performed with a timestep of $\Delta t = 0.005$ ps, relaxation time = 5000*Δt = 25 ps, temperature = 0.01 K, velocity = 0.03615 Å/ps. The boundary conditions applied here are as shown in Figure 2.

Figure 2: Boundary conditions on the nanobeam



To perform lammps simulations, it is necessary to define the interatomic potentials. The interatomic potentials used in this work are of type EAM and LJ potentials. The EAM potential is taken from the work of Foiles et al [2] and the LJ potential is of the form,

$$\phi_{LJ}(r_{ij}) = 4\epsilon[(\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^{6}] \tag{1}$$

Total energy in EAM formalism is given by

$$E = \frac{1}{2}\sum_{\alpha,\beta,\alpha\neq\beta}\phi_{\alpha\beta}(r^{\alpha\beta}) + \sum_{\alpha}U_{\alpha}(r^{\alpha}) \tag{2}$$

here, $U_\alpha(r^\alpha)$ is the embedding energy at the location of atom α and the first term is the contribution from pair interaction between pairs of atoms (α,β)

The following three tasks are performed in this work.

- Comparison of the shape of the cross section at the right end of the beam before and after relaxation for geometries G1 and G2 and, the crystallographic loading directions [100], [110].
- Comparison between EAM and LJ : Plotting of average stress vs. displacement and average stress vs. strain. Computation of Young's modulus E for the loading direction [100] in both G1 and G2 geometries. Comparison of Es calculated using EAM and LJ.
- Comparison of different crystallographic directions : Compute the average Young's modulus E using the EAM potential, for geometry G2, for both crystallographic orientations and compare it to tabular values.

2.1 Simulation setup in lammps

At first, the type of units to be used in the simulation is set up using "units metal". Then files for the atomic configurations of Cu fcc crystal oriented along 1. X = [100] Y = [010] Z = [001], 2. X = [110] Y = [1-10] Z = [001](represented as I-[110]), 3. X = [110] Y = [1-11] Z = [1-1-2] (represented as II-[110]) with two geometries G1 and G2 are created either using the "lattice" command in lammps or using the software ovito or by creating a script that can create the atomic configurations. The following labels are used for different orientations of the Cu crystal.

- X = [100] Y = [010] Z = [001] with G1 geometry → G1-[100]
- X = [100] Y = [010] Z = [001] with G2 geometry → G2-[100]
- X = [110] Y = [1-10] Z = [001] with G1 geometry → G1-I-[110]
- X = [110] Y = [1-10] Z = [001] with G2 geometry → G2-I-[110]
- X = [110] Y = [1-11] Z = [1-1-2] with G1 geometry → G1-II-[100]
- X = [110] Y = [1-11] Z = [1-1-2] with G2 geometry → G2-II-[110]

Figure 3: Orientation 1 of the Cu crystal : G1/G2-[100] with loading axis [100]

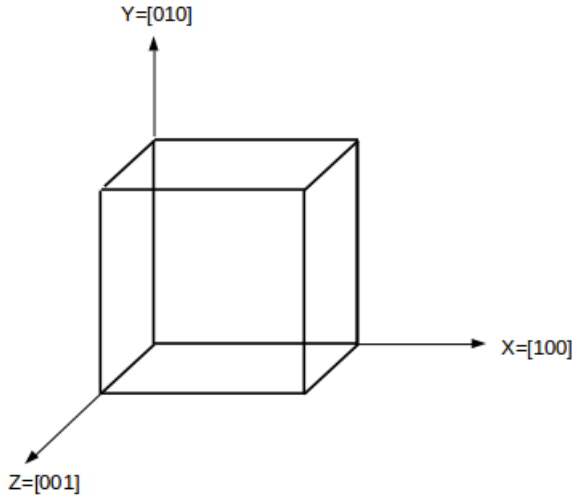


Figure 4: Orientation 2 of the Cu crystal : G1/G2-I-[110] with loading axis [110]

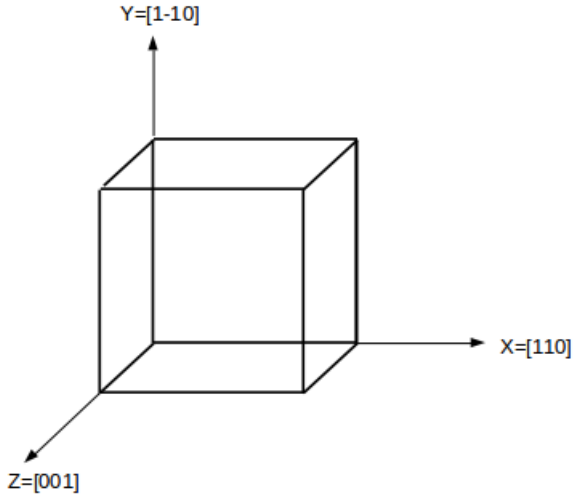
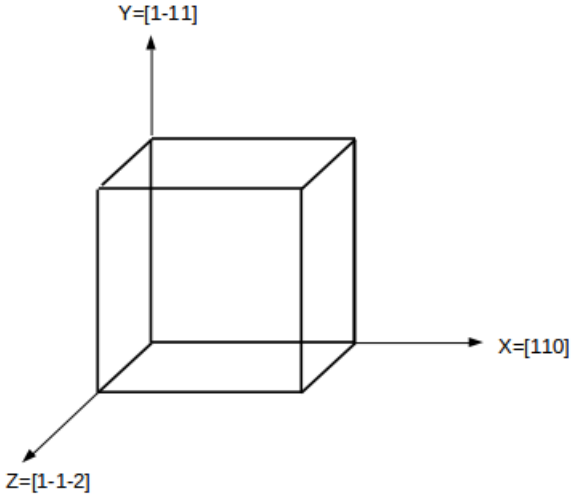


Figure 5: Orientation 3 of the Cu crystal : G1/G2-II-[110] with loading axis [110]



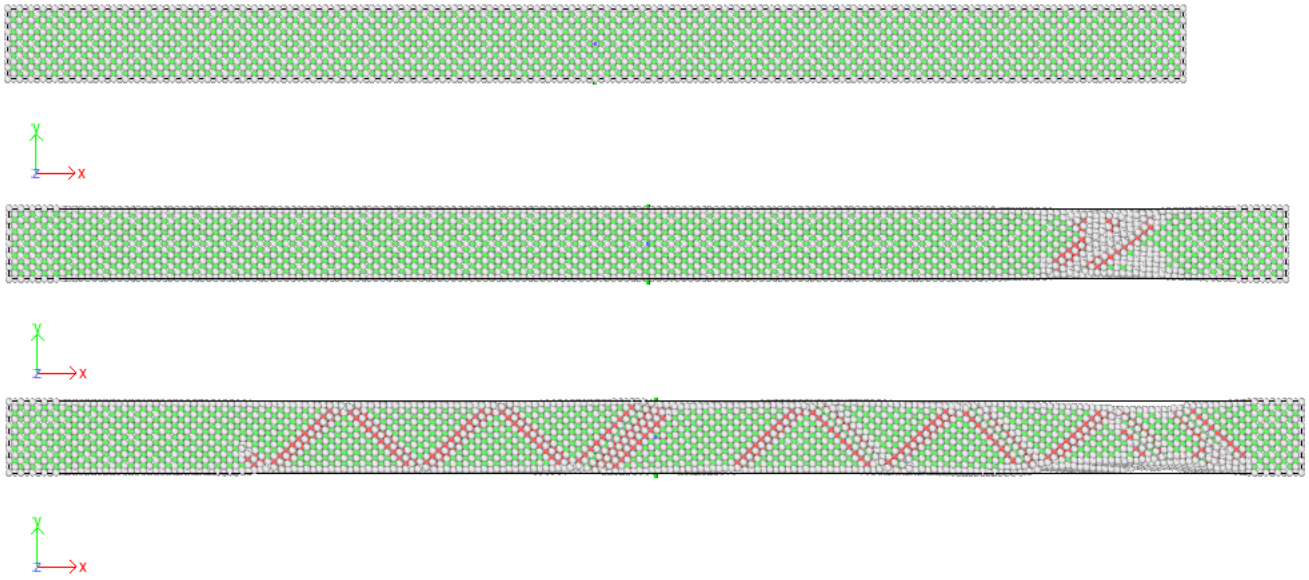
After the atomic configuration file is imported in lammps, the atoms are grouped into three different regions namely leftbc, middle_atoms, rightbc according to the atoms grouped in Figure 2. Then to achieve a temperature of 0.01 K in the system, velocities are assigned to the atoms by using "velocity" command of lammps. Then as described in Figure 2., the cantilever beam is realized by fixing the atoms in "leftbc" through setting the forces on these atoms to zeros at all times during the simulation. The atoms belonging to "rightbc" are displaced along the required X-directions with the velocity 0.03615 Å/ps, using "displace_atoms" command in lammps while the atoms belonging to "middle_atoms" are allowed to move according to NVT ensemble. Displacing of atoms is done in a loop such that the final displacement of the "rightbc" atoms is 10 unitcells in the X-direction.

The lammps input script for these tensile loading simulations is given in the appendix.

3 Results and discussion

The visualization of the tensile loading simulation of Cu single crystal nanobeam with geometry G1 and loading direction [100] is shown in Figure 6.

Figure 6:]
G1 tensile loading in [100] : a) nanobeam before loading, b) nanobeam when necking is initiated c) nanobeam with maximum displacement of 10 unit cells along [100]



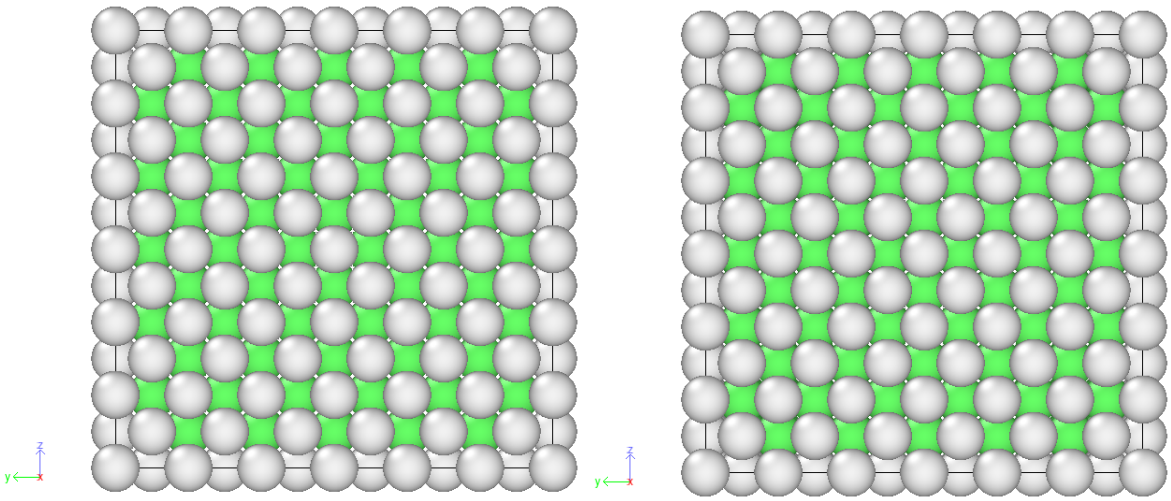
Here in Figure 6, we can see that the left end of the nanobeam is fixed and loading is done on the right end. We can clearly see the initiation of necking and the cross sectional area decreasing. If the simulation was performed for longer time, we would see the rupture of the beam at necking location.

3.1 Relaxation

Comparison of the shape of the cross section at the right end of the beam before and after relaxation for geometries G1 and G2 and, the crystallographic loading directions [100], [110].

The shape of cross section at the right end of the beam before and after the relaxation remained the same for both geometries G1 and G2 in all the three orientations of the crystal. This is due to the crystal structure already being at lower energy configuration (because 3.615 Åis used as the lattice parameter of fcc Cu which is exactly what has been predicted by the EAM and LJ potentials used in this work) and a very low temperature of 0.01 K.

Figure 7: Shape of cross section before (left) and after relaxation (right)



3.2 Comparison between EAM and LJ potentials

Calculation and plotting of the average stress in the loading direction Vs. displacement using both G1 and G2 geometries while the beam is loaded in the [100] direction. The results of both the LJ and EAM potentials are compared

The average stress Vs. displacement plots and average stress Vs. strain plots (using both EAM and LJ) for G1 and G2 geometries of orientation with X=[100] Y=[010] Z=[001] where X=[100] is the loading axis, are shown in Figures 8, 9, 14, 15. The computed Young's moduli are tabulated in Table 1. The Young's modulus is calculated by finding the slope of the linear part (for +ve stress values) of the stress-strain plot where strain is the engineering strain.

Table 1: Young's modulus using EAM and LJ potentials for loading in [100] direction.

Loading direction	E - units	E - EAM	E - LJ	Reference
G1-[100]	GPa	74.053	109.447	72[3], 80[4]
G1-I-[110]	GPa	139.634	191.689	141[3], 150[4]
G1-II-[110]	GPa	135.309	179.655	-

Figure 8: Stress Vs Displacement G1 geometry tensile loading axis [100] (X=[100] Y=[010] Z=[001])

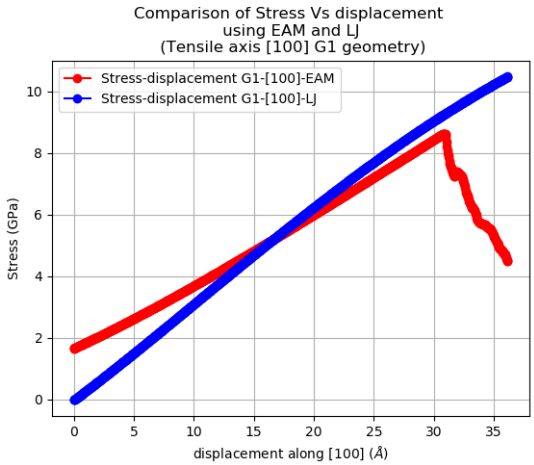


Figure 9: Stress Vs Strain G1 geometry tensile loading axis [100] (X=[100] Y=[010] Z=[001])

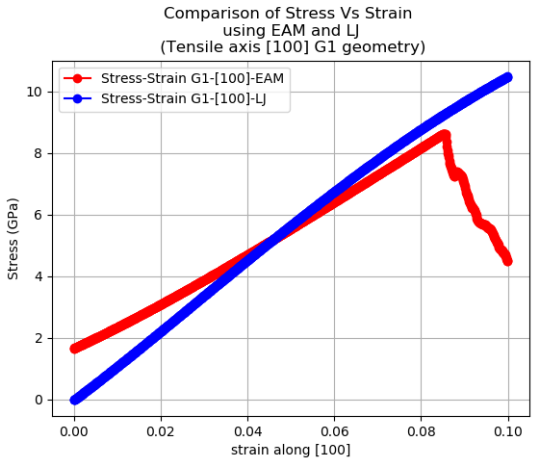


Figure 10: Stress Vs Displacement G1 geometry tensile loading axis [110] (X=[110] Y=[1-10] Z=[001])

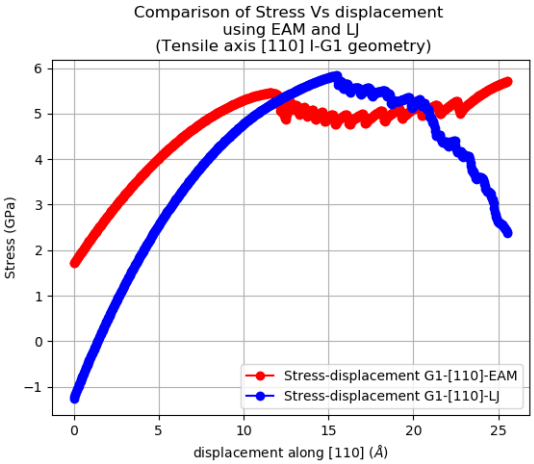


Figure 11: Stress Vs Strain G1 geometry tensile loading axis [110] (X=[110] Y=[1-10] Z=[001])

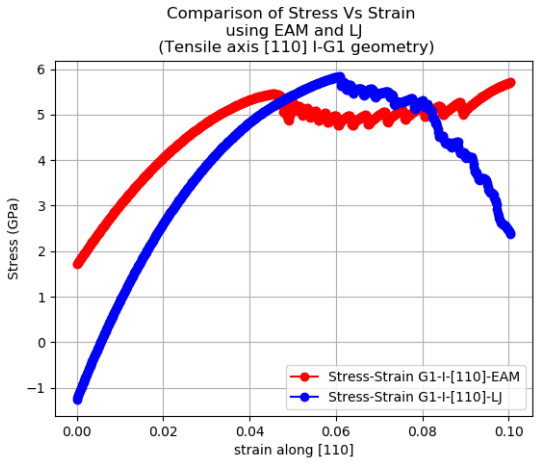


Figure 12: Stress Vs Displacement G1 geometry tensile loading axis [110] (X=[110] Y=[1-11] Z=[1-1-2])

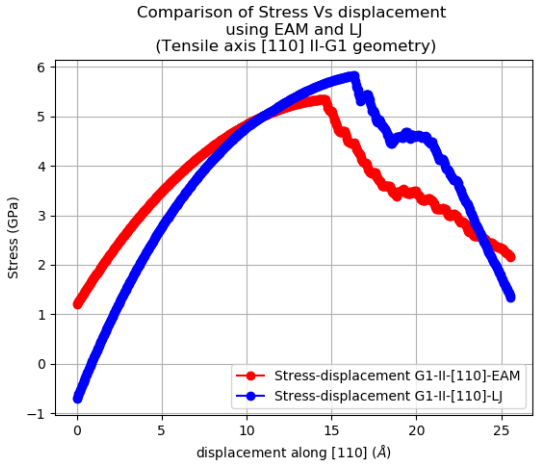
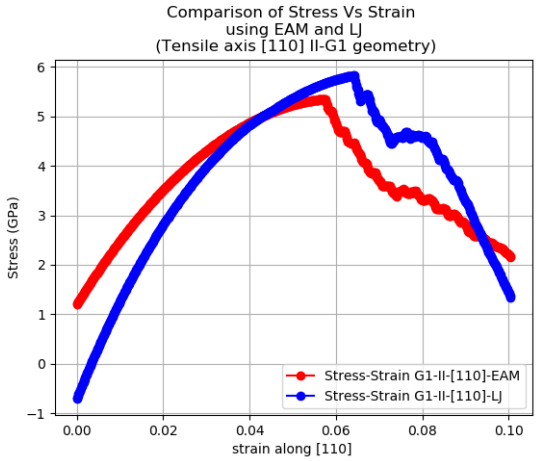


Figure 13: Stress Vs Strain G1 geometry tensile loading axis [110] (X=[110] Y=[1-11] Z=[1-1-2])



3.3 Comparison of different crystallographic directions

Compute the average Young's modulus E using the EAM potential, for geometry G2, for both crystallographic orientations and compare it to tabular values.

Table 2: Young's modulus in G1 and G2 geometries using EAM potential for loading in [100], [110] directions.

Loading direction	Geometry	E - units	E - EAM	Reference
G1-[100]	G1	GPa	74.053	72[3], 80[4]
G2-[100]	G2	GPa	74.35	-
G1-I-[110]	G1-I	GPa	138.51	141[3], 150[4]
G2-I-[110]	G2-I	GPa	132.81	-
G1-II-[110]	G1-II	GPa	135.308	-
G2-II-[110]	G2-II	GPa	138.369	-

Figure 14: Stress Vs Displacement G2 geometry tensile loading axis [100] (X=[100] Y=[010] Z=[001])

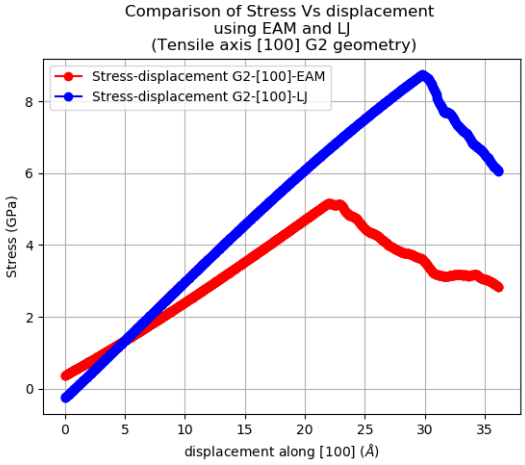


Figure 15: Stress Vs Strain G2 geometry tensile loading axis [100] (X=[100] Y=[010] Z=[001])

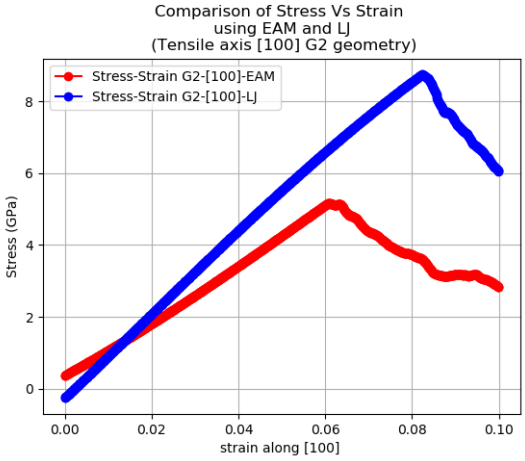


Figure 16: Stress Vs Displacement G2 geometry tensile loading axis [110] (X=[110] Y=[1-10] Z=[001])

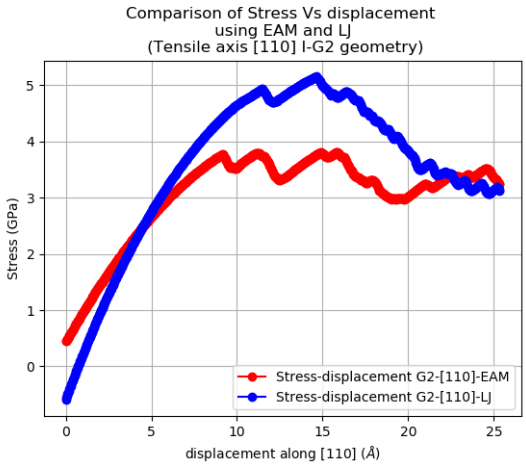


Figure 17: Stress Vs Strain G2 geometry tensile loading axis [110] (X=[110] Y=[1-10] Z=[001])

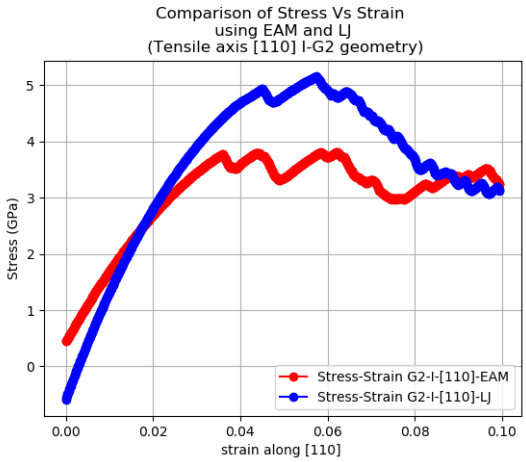


Figure 18: Stress Vs Displacement G2 geometry tensile loading axis [110] (X=[110] Y=[1-11] Z=[1-1-2])

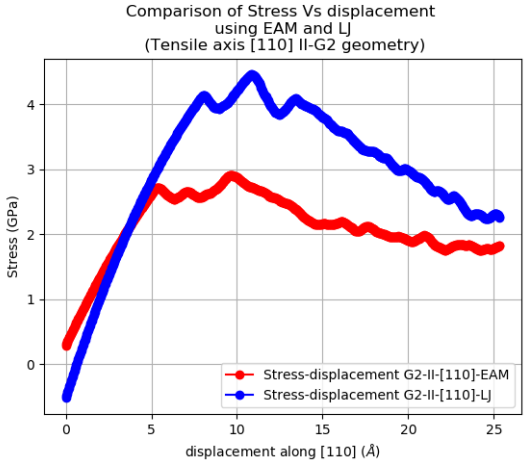
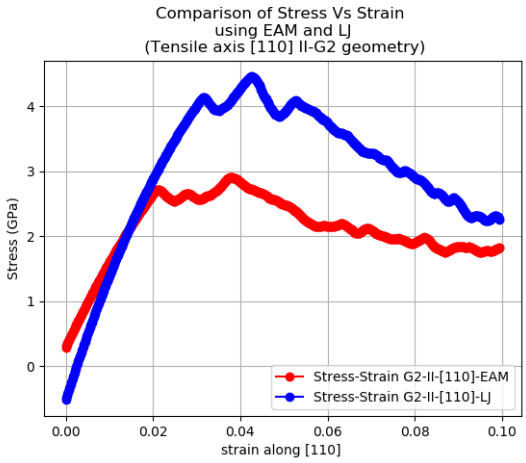


Figure 19: Stress Vs Strain G2 geometry tensile loading axis [110] (X=[110] Y=[1-11] Z=[1-1-2])



Comparing the Tables 1, 2 and stress Vs. strain plots above, it is clear that the EAM potential is well suited for the simulation of Cu nanowires as LJ potential over-estimates the Young's modulus. Also -ve stresses are observed for LJ potential in stress Vs. strain plots. This may be because of the facts that 1. LJ potential usually gives more realistic results in solidified ideal gases, 2. inaccurate repulsive part of the LJ potential in cases metallic materials.

4 Conclusion

Comparison of EAM and LJ
In this both EAM and LJ formalism are used to evaluate their performance in simulating metallic Cu system and, also investigate the geometry and tensile loading direction dependence of Young's modulus E. The EAM potential has predicted the Young's modulus (for both [100] and [110] loading directions) with $\approx$ 98.6% accuracy when compared to experimental values [3] and LJ potential has predicted the Young's modulus (for both [100] and [110] loading directions) with $\approx$ 93% accuracy. Also, negative stresses were observed using LJ potential in the stress Vs. strain plots. This observation is unphysical. Therefore, EAM potential describes the metallic Cu system with better accuracy than LJ potential does.

Comparison of different crystallographic orientations
Generally, it was observed that the Young's modulus value for loading direction [110] is much higher than the Young's modulus value for loading direction [100] (Table 1). This can be attributed to the critically resolved shear stress (CRSS) and active available slip systems in fcc crystal structure of Cu.

Comparison of Young's modulus in different geometries Further, for loading direction [100], G2 geometry shows slightly higher Young's modulus than G1 (Table 2). This is because of the increased cross-sectional area in G2, as expected. The same observation was made for the loading direction [110] in the Cu crystal with orientations X=[110] Y=[1-11] Z=[1-1-2] (Table 2). But for loading direction [110] in crystal with orientations X=[110] Y=[1-10] Z=[001], it was observed that G1 geometry has slightly higher Young's modulus than for G2.

5 Appendices

5.1 Configuration file created in lammps

```
##### TENSILE TEST Cu NANOWIRE

units metal  #metallic units used in the simulation
dimension 3  #3D simulation box
boundary s s s #non-periodic and shrink-wrapped BCs
atom_style atomic #style of atoms
variable latconst equal 3.615 #lattice parameter of Cu (FCC)

#creating the fcc crystal with desired orientation
lattice fcc ${latconst}
region nanowire block 0 100 0 6 0 6 #region where atoms will be created
create_box 3 nanowire #creating a box of dimensions defined through the region
lattice fcc ${latconst} orient x 1 0 0 orient y 0 1 0 orient z 0 0 1
create_atoms 1 region nanowire #creating atoms in the fcc crystal

#defining left, right and middle regions
region    leftbc block 0 4 INF INF INF INF
group        leftbc region leftbc
region    rightbc block 96 100 INF INF INF INF
group        rightbc region rightbc
group        boundary union leftbc rightbc
goup        middle_atoms subtract all boundary

#setting atomic masses. Must be set to atomic mass=63.55 for LJ potential
mass * 1.0
#LJ potential
#pair_style lj/cut 6.4
#pair_coeff * * 0.1515 2.338
#EAM potential
pair_style eam
pair_coeff * * Cu_u3.eam

#writing the trajectory files to an output file
dump 1 all custom 1000000 dump.initial id type x y z
minimize 1e-14 1.0e-14 1000000 1000000
undump 1

#setting temperature = 0.01 K by creating atomic velocities
velocity    middle_atoms create 0.01 511124 rot yes mom yes
fix 1 middle_atoms nvt temp 0.01 0.01 0.1

#Freezing the atoms belonging to leftmost region
fix 2 leftbc setforce 0.0 0.0 0.0
```

```
timestep   0.005

#setting for screen output
thermo     100
thermo_style custom step lx temp pxx pyy pzz

dump 1 all atom 500 dump.all.indent
dump_modify  1 append yes

run  5000 #achieving 0.01 K temperature in 5000 timesteps = 25 ps

# Storing final cell length along x-axis for strain calculations
variable tmp equal "lx"
variable L0 equal ${tmp}
print "Initial Length, L0: ${L0}"
dump 2 all atom 200 newdump.tension
reset_timestep 0
timestep 0.005
variable a loop 1000

label loop
#moving (i.e applying tensile load) only the atoms belonging extreme right region
displace_atoms rightbc move 0.03615 0.0 0.0 units box
#allowing the middle atoms to move according to NVT ensemble
fix 3 middle_atoms nvt temp 0.01 0.01 0.01
variable tempstrain equal "(lx - v_L0)/v_L0"
variable strain equal v_tempstrain
thermo  200
thermo_style custom step lx v_strain temp pxx pyy pzz
run 200
next a
jump in.tension loop

# SIMULATION DONE
print "All done"
```

5.2 Configuration file created outside lammps

```
##### TENSILE TEST Cu NANOWIRE

units metal
dimension 3
boundary s s s
atom_style atomic
#importing configuration from outside of lammps
read_data ../G1_2Cu_110.lmp
```

```
region    leftbc block 0 7.66857303 INF INF INF INF
group        leftbc region leftbc
region    rightbc block 247.950528 255.619101 INF INF INF INF
group        rightbc region rightbc
group        boundary union leftbc rightbc
group        middle_atoms subtract all boundary

mass * 63.55
#pair_style lj/cut 6.4
#pair_coeff * * 0.1515 2.338
pair_style eam
pair_coeff * * Cu_u3.eam

dump 1 all custom 1000000 dump.initial id type x y z
minimize 1e-14 1.0e-14 1000000 1000000
undump 1

velocity    middle_atoms create 0.01 511124 rot yes mom yes
fix 1 middle_atoms nvt temp 0.01 0.01 0.1
fix 2 leftbc setforce 0.0 0.0 0.0

timestep   0.005

thermo     100
thermo_style custom step lx temp pxx pyy pzz

dump 1 all atom 500 dump.all.indent
dump_modify  1 append yes

run  5000

# Store final cell length for strain calculations
variable tmp equal "lx"
variable L0 equal ${tmp}
print "Initial Length, L0: ${L0}"

dump 2 all atom 200 newdump.tension
#change_box all boundary p p p
reset_timestep 0
timestep 0.005
variable a loop 1000

label loop
displace_atoms rightbc move 0.03615 0.0 0.0 units box
fix 3 middle_atoms nvt temp 0.01 0.01 0.01
```

```
variable tempstrain equal "(lx - v_L0)/v_L0"
variable strain equal v_tempstrain
thermo  200
thermo_style custom step lx v_strain temp pxx pyy pzz
run 200
next a
jump in.tension loop

# SIMULATION DONE
print "All done"
```

References

[1] S. Plimpton. *Fast parallel algorithms for short-range molecular dynamics*. Tech. rep. Sandia National Labs., Albuquerque, NM (United States), 1993.

[2] S. Foiles, M. Baskes, and M. Daw. "Erratum: Embedded-atom-method functions for the fcc metals Cu, Ag, Au, Ni, Pd, Pt, and their alloys".
In: *Physical Review B* 37.17 (1988), p. 10378.

[3] C. Kittel, P. McEuen, and P. McEuen. *Introduction to solid state physics*. Vol. 8. Wiley New York, 1996.

[4] A. Ahadi, P. Hansson, and S. Melin.
"Tensile behavior of single-crystal nano-sized Cu beams–Geometric scaling effects".
In: *Computational Materials Science* 135 (2017), pp. 127–133.