INDIAN INSTITUTE OF TECHNOLOGY INDORE

Department of Metallurgy Engineering and Materials Science (MEMS)

MM 309

Project Report

"Application of Molecular dynamics (MD) in Materials Science and Engineering".

Submitted By-

Nikhil Gupta	**(190005026)**
Ishan Bharti	**(190005019)**
Ankit Pandey	**(190005004)**
Piyush Nimbhore	**(190005028)**
Pradumn Patidar	**(190005030)**
Ayush Nath	**(190005009)**

- **Introduction**

Molecular Dynamics (MD) is a computer simulation method for analyzing the physical movement of atoms and molecules. The trajectories of atoms and molecules are determined by numerically solving Newton's equations of motion for a system of interacting particles. MD follows deterministic modeling which yields the exact same results for a particular set of inputs, no matter how many times one recalculates the model. MD connects the macroscopic world and the microscopic world provided by the theory of statistical mechanics and thus it is termed "statistical mechanics by numbers". MD has also been termed as "Laplace's vision of Newtonian Mechanics" of predicting the future by animating nature's forces and allowing insight into molecular motion on an atomic scale.

- **History of MD**

➢ Early 1950s (Metropolis, Rosenbluth & Teller) :
- ❖ MANIAC (one of the first computers) was operational at Los Alamos National Lab.
- ❖ Monte Carlo method to address the entropic problem.

➢ 1953 (Fermi) :
- ❖ Simulation of an harmonic 1D crystal with the help of MANIAC I.

➢ 1956 (Alder and Wainwright) :
- ❖ Used IBM 704 computer to simulate perfectly elastic collisions between hard spheres.

➢ 1960 (Vineyard group) :
- ❖ Simulation of damaged Cu crystal by using Born-Mayer type of repulsive interaction.

➢ 1964 (Rahman) :
- ❖ MD simulation of liquid Ar by using Lennard-Jones potential.

➢ 1967 (L. Verlet) :
- ❖ MD simulation on properties of Ar by using Lennard-Jones potential.
- ❖ Introduced the Verlet Integration Algorithm.

- 1971 (Rahman and Stillinger) :
  - MD simulation of water.

- **Molecular Dynamics Procedure**

  - Choose initial positions for the atoms.

  - Choose an initial set of atom velocities.
    These are usually chosen to obey a Boltzmann distribution for some temperature, then normalized so that the momentum for the entire system is zero.

  - Compute the momentum of each atom from its velocity and mass.

  - Compute the forces on each atom from the energy expression This is usually a molecular mechanics force field designed to be used in dynamical simulations.

  - Compute new positions for the atoms in a short time step (Δt).
    This is a numerical integration of Newton's equations of motion using the information obtained in the previous steps.

  - Compute new velocity and accelerations for the atoms.

  - Repeat steps (3) through (6).

  - Repeat this iteration long enough for the system to reach equilibrium.
    In this case, equilibrium is not the lowest energy configuration; it is configuration that is reasonable for the system with the given amount of energy.

  - Once the system has reached equilibrium, begin saving the atomic coordinates every iteration.
    This list of coordinates over time is called a trajectory.

  - Continue iterating and saving data until enough data have been collected to give results with the desired accuracy.

  - Analyze the trajectories to obtain information about the system.
    This might be determined by computing radial distribution functions, diffusion coefficients, vibrational motions, or any other property computable from this information.

- **Useful Algorithms Used In Molecular Dynamics**

➢ Verlet algorithm

Verlet integration is a numerical method used to integrate Newton's equations of motion. It is frequently used to calculate trajectories of particles in molecular dynamics simulations.

➢ Velocity Verlet algorithm

This Algorithm except that the velocity and position are calculated at the same value of the time variable.

➢ The Leap-Frog Algorithm

The velocities are first calculated at the time t+1/2 δt and using this velocity, position r is obtained at time t+δt.
The advantage of this algorithm is that the velocities are explicitly calculated.

➢ Gear's predictor-corrector Algorithms

It belongs to a class of algorithms designed to integrate ordinary differential equations – to find an unknown function that satisfies a given differential equation.

➢ Beeman's Algorithm

Beeman's algorithm uses positions, velocities, and accelerations from the previous time step. This algorithm is closely related to the Verlet algorithm.

- **INTRODUCTION TO LAMMPS**

➢ Classical Molecular Dynamics Program And Can Simulate For Atomic, Polyatomic, Biological, Metallic, Granular System From Few Particles Up To Billions Of Particles.

➢ Message Passing (Mpi) For Parallel Communication Which Ensures Multi-Core Devices Simulate With Full Efficiency Without Any Lag.

➢ We Provide The Software Some Initial Parameters Like Initial Positions, Initial Velocities And The Software Process These Values To Give Us The Results In The Form Of Trajectories, Forces, Velocities, etc.

➢ Applications Fields In Which LAMMPS Is Useful Are –

- ◆ Solid Mechanics
- ◆ Material Sciences
- ◆ Granular Flows
- ◆ Biophysics
- ◆ Quantum Chemistry.

- ## Molecular Dynamics Example-1

  Molecular simulation (MD) of uniaxial tensile loading of an aluminum single crystal oriented in the <100> direction.

❖ Entire Code

```
# Input file for uniaxial tensile loading of single-crystal aluminum

# ---------------------- INITIALIZATION ----------------------------
units           metal
dimension       3
boundary        p       p       p
atom_style      atomic
variable latparam equal 4.05

# ---------------------- ATOM DEFINITION ---------------------------
lattice         fcc ${latparam}
region          whole block 0 10 0 10 0 10
create_box      1 whole
lattice fcc ${latparam} orient x 1 0 0 orient y 0 1 0 orient z 0 0 1
create_atoms        1 region whole

# ---------------------- FORCE FIELDS ------------------------------
pair_style      eam/alloy
pair_coeff      * * Al99.eam.alloy Al

# ----------------------- SETTINGS ---------------------------------
compute csym all centro/atom fcc
compute peratom all pe/atom

##########################################
# EQUILIBRATION
reset_timestep      0
timestep 0.001
velocity all create 300 12345 mom yes rot no
fix 1 all npt temp 300 300 1 iso 0 0 1 drag 1

# Set thermo output
thermo 1000
thermo_style custom step lx ly lz press pxx pyy pzz pe temp

# Run for at least 10 picosecond (assuming 1 fs timestep)
```

```
run 20000
unfix 1

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

##########################################
# DEFORMATION
reset_timestep        0

fix                   1 all npt temp 300 300 1 y 0 0 1 z 0 0 1 drag 1
variable srate equal 1.0e10
variable srate1 equal "v_srate / 1.0e12"
fix                   2 all deform 1 x erate ${srate1} units box remap x

# Output strain and stress info to file
# for units metal, pressure is in [bars] = 100 [kPa] = 1/10000 [GPa]
# p2, p3, p4 are in GPa
variable strain equal "(lx - v_L0)/v_L0"
variable p1 equal "v_strain"
variable p2 equal "-pxx/10000"
variable p3 equal "-pyy/10000"
variable p4 equal "-pzz/10000"
fix def1 all print 100 "${p1} ${p2} ${p3} ${p4}" file Al_SC_100.def1.txt screen no

# Use cfg for AtomEye
dump          1 all cfg 250 dump.tensile_*.cfg mass type xs ys zs c_csym c_peratom
fx fy fz
dump_modify 1 element Al

# Display thermo
thermo        1000
thermo_style custom step v_strain temp v_p2 v_p3 v_p4 ke pe press

run           20000

##########################################
# SIMULATION DONE
print "All done"
```

Stepwise CODE with Explanation

```
# Input file for uniaxial tensile loading of single crystal aluminium
# ---------------------- INITIALIZATION ---------------------------
units           metal
dimension       3
boundary        p       p       p
atom_style      atomic
variable latparam equal 4.05
```

- This section initializes the simulation, the unit command specifies the unit that will be used for the remainder of the simulation, here 'metal' uses Å and eV values.
  The dimension 3 command specifies a 3- dimensional simulation cell.
  And the boundary p p p specifies periodic boundaries in x, y, and z-direction.

```
# ---------------------- ATOM DEFINITION ---------------------------
lattice         fcc ${latparam}
region          whole block 0 10 0 10 0 10
create_box      1 whole
lattice fcc ${latparam} orient x 1 0 0 orient y 0 1 0 orient z 0 0 1
create_atoms          1 region whole
```

- The lattice command defines the FCC lattice with lattice parameter=4.
  Region command creates a geometric region of size 10 lattice spacing in each direction.
  create_box command uses region command to create a box of the lattice.
  lattice command sets the orientation of FCC lattice in x, y, and z-direction.

```
# ---------------------- FORCE FIELDS ----------------------------
pair_style      eam/alloy
pair_coeff      * * Al99.eam.alloy Al
```

- The pair_style command specifies what kind of interatomic potential used ' Here we used EAM Al alloy potential'.
  & the pair_coeff uses the file that pair potential is stored, Al99.eam.alloy.

```
# ----------------------- SETTINGS -------------------------------
compute csym all centro/atom fcc
compute peratom all pe/atom
```

- Compute command calculates the centro-symmetry parameter for each atom in FCC lattice
  Another compute, assigns a variable named ‘peratom’ to store the potential energy of each atom.

```
# EQUILIBRATION
reset_timestep        0
timestep 0.001
velocity all create 300 12345 mom yes rot no
fix 1 all npt temp 300 300 1 iso 0 0 1 drag 1
```

- The reset_timestep command resets the timestep to 0.
  ‘timestep’, set the timestep size to 10^{-3} for the simulation.
  velocity all create , set the velocity of all atoms & create_style generates an ensemble of velocity at specified temperature.
  mom= yes, Linear momentum of ensemble velocity is ZERO.
  rot= no, angular momentum is not ZERO.
  These commands are used to generate position and velocities of a ‘NPT’ ensemble, subsequently at each timestep.

```
# Set thermo output
thermo 1000
thermo_style custom step lx ly lz press pxx pyy pzz pe temp
```

- The thermo_style is used to print the thermodynamic data to the screen & logfile data.
  Here, a custom step provides the user to define variables such as lx, ly, and lz for the logfile data.

```
# Run for at least 10 picosecond (assuming 1 fs timestep)
run 20000
unfix 1
```

- The run commands run simulation to 20000 timestep.

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

- This section defines some variables such as ‘tmp’ for cell length & it is initialized with L0 { initial length } which further will be used for straini calculation.

```
# DEFORMATION
reset_timestep	0

fix		1 all npt temp 300 300 1 y 0 0 1 z 0 0 1 drag 1
variable srate equal 1.0e10
variable srate1 equal "v_srate / 1.0e12"
fix		2 all deform 1 x erate ${srate1} units box remap x
```

- Again, the timestep resets to 0 and the position and velocities are updated. Variable ‘srate’ is created equals to 1×10^{10} and Variable ‘srate1’ equals srate/10^{12}.
  Then, the fixed deformation updates the change in volume and shape during simulation.

```
# Output strain and stress info to file
# for units metal, pressure is in [bars] = 100 [kPa] = 1/10000 [GPa]
# p2, p3, p4 are in GPa
variable strain equal "(lx - v_L0)/v_L0"
variable p1 equal "v_strain"
variable p2 equal "-pxx/10000"
variable p3 equal "-pyy/10000"
variable p4 equal "-pzz/10000"
fix def1 all print 100 "${p1} ${p2} ${p3} ${p4}" file Al_SC_100.def1.txt screen no
```

- Further, the set of variables are assigned for the output stress-strain data file, which is useful in Graph plotting. Here different variables are assigned

such as ‘strain = (L-L0)/L0 ‘ and p1,p2,p3 and p4 & the values are stored in file “ Al-SC-100.def1.txt “.

```
# Use cfg for AtomEye
dump            1 all cfg 250 dump.tensile_*.cfg mass type xs ys zs c_csym
c_peratom fx fy fz
dump_modify 1 element Al
```

- Dump command prints all the atomistic calculations to the file which is used in OUTPUT Visualization.

```
# Display thermo
thermo          1000
thermo_style custom step v_strain temp v_p2 v_p3 v_p4 ke pe press
```

- Now, all the thermodynamic data is displayed on the screen and stored in the form of a logfile.
  simulation will run for 20000 steps
  At the end of the simulation, the command prints **“ All Done “** .

❖ OUTPUT

➔**OVITO Video clip is Linked here**

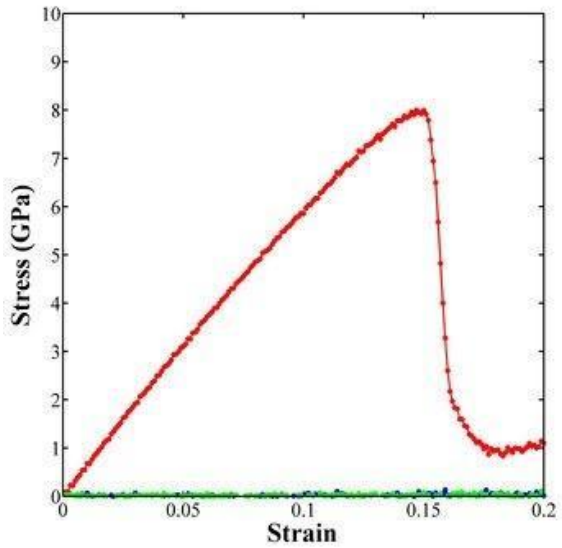


Fig.1- “File Al_SC_100.def1.txt “ is created after simulation. Using the MATLAB software,

The stress-strain curve is plotted.

- **Molecular Dynamics Example-2**

❖ Abstract

The objective of this tutorial is to generate a symmetric tilt grain boundary in LAMMPS [1]. This tutorial can serve as a precursor to more advanced techniques, whereby in-plane translations and atom deletion criteria are used to sample a large number of potential structures to find the global minimum energy grain boundary structure [2-3].

❖ Description of Simulation

This LAMMPS [1] molecular dynamics simulation first generates a simulation cell with fcc atoms with two blocks of different orientations to create the grain boundary: a $<031>$ direction normal to the interface plane (often called grain boundary normal), a $<013>$ direction in the grain boundary plane (y-direction), a shared $<100>$ tilt direction in the z-direction. This is a symmetric tilt grain boundary that is known as the $\Sigma5(310)$ symmetric tilt grain boundary since both lattices are rotated about a common tilt direction, the $<100>$ direction. The potential used is the Mishin et al. (1999) aluminum potential [4]. The cfg dump files include the x, y, and z coordinates, the centrosymmetry values, the potential energies, and forces for each atom. This can be directly visualized using OVITO [5] and AtomEye [6] visualization tools.

➔ Step 1: Create the LAMMPS input script

This input script was run using the January 2020 version of LAMMPS. The following input script shows how multiple translations and an atom deletion criteria are used to calculate the minimum energy structure. This input script for LAMMPS [1] can be called with a command of the form, lmp_exe < input.script. This script contains loops over x-translations, z-translations, and atom overlap distances (an atom is deleted when an atom pair with a nearest neighbor distance is less than this distance). The unique minimum energy structures are saved as a dump file with the energy appended to the filename in a new folder specified by the gbname variable. The dump files can then be easily scanned through for the global minimum energy structure.

➔ This is the entire code.

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

# ------ Create Atomistic Structure ----------
lattice fcc 4.05
region whole block 0.000000 12.807225 -64.0361225 64.0361225 0.000000 4.050000 units box
create_box 2 whole
region upper block INF INF 0.000000 64.0361225 INF INF units box
lattice fcc 4.05 orient x  0  3  1 orient y  0 -1  3 orient z  1  0  0
create_atoms 1 region upper
region lower block INF INF -64.0361225 0.000000 INF INF units box
lattice fcc 4.05 orient x  0  3 -1 orient y  0  1  3 orient z  1  0  0
create_atoms 2 region lower
group upper type 1
group lower type 2

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

# ------ Displace atoms and delete overlapping atoms ----------
displace_atoms upper move 0 0 0 units lattice
delete_atoms overlap 0.35 lower upper

# ------ Define Settings ----------
compute csym all centro/atom fcc
compute eng all pe/atom
compute eatoms all reduce sum c_eng

# ------ Run Minimization ----------
reset_timestep 0
thermo 10
thermo_style custom step pe lx ly lz press pxx pyy pzz c_eatoms
dump 1 all cfg 25 dump.sig5_minimization_*.cfg mass type xs ys zs c_csym c_eng fx fy fz
dump_modify 1 element Al Al
min_style cg
minimize 1e-15 1e-15 5000 5000
undump 1

# ------ Run Minimization 2----------
# Now allow the box to expand/contract perpendicular to the grain boundary
reset_timestep 0

thermo 10
thermo_style custom step pe lx ly lz press pxx pyy pzz c_eatoms
fix 1 all box/relax y 0 vmax 0.001
min_style cg
minimize 1e-15 1e-15 5000 5000

# ------ Calculate GB Energy ----------
variable minimumenergy equal -3.360000
variable esum equal "v_minimumenergy * count(all)"
variable xseng equal "c_eatoms - (v_minimumenergy * count(all))"
variable gbarea equal "lx * lz * 2"
variable gbe equal "(c_eatoms - (v_minimumenergy * count(all)))/v_gbarea"
```

```
variable gbemJm2 equal ${gbe}*16021.7733
variable gbernd equal round(${gbemJm2})
print "GB energy is ${gbemJm2} mJ/m^2"

# ————— Dump data into Data file ————————-
reset_timestep 0
dump 1 all cfg 10000 dump.al_sig5_310_*.cfg mass type xs ys zs c_csym c_eng fx fy fz
dump_modify 1 element Al Al
minimize 1e-15 1e-15 5000 5000
undump 1

write_restart restart.al_sig5_310_stgb

print "All done"
```

Following is the code Explanation segment-wise

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

Firstly we need to initialize the simulation. The unit's command helps us set the style of units to be used, which is metal in this particular simulation. It determines the units of all quantities to input script and data file. Next, we set the dimensionality of the simulation as 3dimensional. We use the boundary command to set the style of boundaries for the global simulation box as p which indicates the box is periodic. The particles can interact across the boundaries. They exit through one end and re-enter through another end. The atom-style command defines the style of atoms as atomistic.

```
# ---------- Create Atomistic Structure ---------------------
lattice fcc 4.05
region whole block 0.000000 12.807225 -64.0361225 64.0361225 0.000000 4.050000
units box
```

```
create_box 2 whole
region upper block INF INF 0.000000 64.0361225 INF INF units box
lattice fcc 4.05 orient x  0  3  1 orient y  0 -1  3 orient z  1  0  0
create_atoms 1 region upper
region lower block INF INF -64.0361225 0.000000 INF INF units box
lattice fcc 4.05 orient x  0  3 -1 orient y  0  1  3 orient z  1  0  0
create_atoms 2 region lower
group upper type 1
group lower type 2
```

Second, comes the creation of the atomistic structure. We define an fcc lattice with a lattice parameter of 4.05 A. Next region_command defines the geometric region of space, a block with minimum and maximum x values of 0 to 12.807 A and similarly in y and z directions, relative to the fcc lattice already created. create_box specifies the created region as the simulation cell. But there are no atoms yet so we populate the lattice with create_atoms. Next, we define the lattice again and specify the crystal orientation of the upper block of the structure. We populate the upper block of the simulation cell with atoms using create_atoms. We do the same with the lower block.

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

Here we define the interatomic potential. Interatomic potential tells us how the atoms should be interacting. Pair style is set to eam/alloy style(metal alloy). The pair_coeff specifies the pair_wise interactions for all types of atoms in the simulation. We get a forced cutoff distance from the eam-alloy file. The neighbor skin command basically gives us an extra distance between neighboring atoms for forced cutoff. neigh_delay gives the time limit for each neighboring atoms lists build.

```
# ---------- Displace atoms and delete overlapping atoms -------------------
displace_atoms upper move 0 0 0 units lattice
delete_atoms overlap 0.35 lower upper
```

Next, we start configuring atom positions and delete the overlapping atoms in 0.35 Å overlap between the upper and lower regions of the block. 2 atoms were deleted in total.

```
# ---------- Define Settings ---------------------
compute csym all centro/atom fcc
compute eng all pe/atom
compute eatoms all reduce sum c_eng
```

Fifthly, settings get defined. We compute the centro-symmetry parameter and per-atom potential energy for each atom in a group.
(pe= pair + bond + angle + dihedral + improper energy) The compute_reduce command gives us the sum of all values in the potential energy vector.

```
# ---------- Run Minimization ---------------------
reset_timestep 0
thermo 10
thermo_style custom step pe lx ly lz press pxx pyy pzz c_eatoms
dump 1 all cfg 25 dump.sig5_minimization_*.cfg mass type xs ys zs c_csym c_eng fx
fy fz
dump_modify 1 element Al Al
min_style cg
minimize 1e-15 1e-15 5000 5000
undump 1
```

We run minimization next. The time skip counter is set to zero. Using thermo, we compute and print ten parameters like potential energy, mean pressure, and the individual pressures in x, y, z directions. The snapshot of atom configurations are outputted to the file location every 25 timesteps with dump cfg command. dump_modify helps us set the color for the Aluminium atoms in the orthogonal box. The minimize command helps us to minimize the energy of the system by iteratively adjusting atom coordinates.

```
# ---------- Run Minimization 2--------------------
# Now allow the box to expand/contract perpendicular to the grain boundary
reset_timestep 0
thermo 10
thermo_style custom step pe lx ly lz press pxx pyy pzz c_eatoms
fix 1 all box/relax y 0 vmax 0.001
min_style cg
minimize 1e-15 1e-15 5000 5000
```

We run minimization for a second time and allow the box to expand/contract perpendicular to the grain boundary. Here we allow the box to relax by fixing some boundary conditions like reducing the pressure in the y-direction to zero and allowing the maximum volume change during iterations to be 0.001.

```
# ---------- Calculate GB Energy --------------------
variable minimumenergy equal -3.360000
variable esum equal "v_minimumenergy * count(all)"
variable xseng equal "c_eatoms - (v_minimumenergy * count(all))"
variable gbarea equal "lx * lz * 2"
variable gbe equal "(c_eatoms - (v_minimumenergy * count(all)))/v_gbarea"
variable gbemJm2 equal ${gbe}*16021.7733
variable gbernd equal round(${gbemJm2})
print "GB energy is ${gbemJm2} mJ/m^2"
```

Here the calculation of grain boundary energy takes place.

```
# ---------- Dump data into Data file -------------
reset_timestep 0
dump 1 all cfg 10000 dump.al_sig5_310_*.cfg mass type xs ys zs c_csym c_eng fx fy
fz
dump_modify 1 element Al Al
minimize 1e-15 1e-15 5000 5000
undump 1

write_restart restart.al_sig5_310_stgb

print "All done"
```

Here we reset the counter to zero. We dump the resultant computed configuration data into data files and run minimization again.

Step 2: Run the LAMMPS Simulation

❖ OUTPUT

1. OVITO video clip of Grain boundary Tilt is linked here.

2. After calculation, GB Energy was found to be the value 564.16610218266 mJ/m^2 in the log file.

INDIVIDUAL RESEARCH PAPER REPORTS

Name: Nikhil Gupta **Roll No.: 190005026**

MM309: Computational Methods for Materials

Research Paper Report

Molecular dynamics studies of CNT-reinforced aluminum composites under uniaxial tensile loading

Overview

This paper investigates the mechanical behavior of carbon nanotube (CNT)-reinforced aluminum composites (CNT-Al composites) under a tensile loading condition using molecular dynamics (MD) simulations. A review of current computational and experimental studies highlights the benefits of CNT-Al composites from a structural point of view. The component analysis shows that the mechanical properties such as Young's Moduli and toughness of CNT-Al composites improve significantly, even with a small amount of CNT, due to the considerable load-bearing capacity of CNT. Furthermore, the MD simulations provide insights into various fracture behaviors at the atomic scale, including lattice disordering, local changes in lattice structures due to stacking faults, and void nucleation/growth.

Approach Used

MD Simulations are performed by taking a small portion of a dog bone specimen of a CNT-Al composite specimen under tensile loading. The simulations are performed for three different models having the same aluminum matrix geometry with various CNT diameters. The armchair nanotubes of (4,4)CNT, (6,6)CNT, and (8,8)CNT are placed inside holes of 5.4 Å, 8.1 Å, and 10.8 Å diameter, respectively.

Before conducting the tensile simulations, the three systems are equilibrated over 1 ns with a time step of 1 fs under the NPT ensemble at 300 K and with a pressure of 0 bar to relax the internal stresses. During the tensile simulations, the simulation box is strained in the z-direction at a strain rate of 10^{-6} /ps, which is slower than a typical strain rate used in other MD simulations. This is carried out

to balance the computation time by considering the issues of undesirable amorphous-disorder deformation with a fast rate, versus calculation expense with a slow rate. The NVT ensemble is applied during the tensile simulation at 300 K. The time step is reduced to 0.5 fs to prevent a blowout of the CNT atoms during fracture.

- **Conclusions**

  In the present study, MD simulations were conducted in order to investigate the mechanical behavior of CNT-reinforced aluminum composite with various CNT diameters. The stress-strain curves were obtained under tensile loading, and the variations of the curves were analyzed through MD snapshots that enable detailed analyses of atomic behaviors. The overall mechanical properties, such as Young's modulus and toughness, increased considerably as the CNT reinforcement was increased. Furthermore, the component analysis revealed that even a small fraction of CNT can play a significant role in enhancing the mechanical properties of composites.

  In addition, the MD simulations offered a detailed fracture mechanism of the MMCs at the atomic scale. For example, the stress decreases were shown to be due to the combination of lattice structure change, dislocation, stacking faults, and micro-void nucleation. It is of considerable significance that these studies at the atomic scale reveal the fundamental factors that define the macroscale properties. A multiscale model can be developed to characterize the mechanical properties of MMCs by combining the MD model and a continuum model with desirable mechanical properties by tailoring the nanoscale variables.

- **References**

Molecular dynamics studies of CNT-reinforced aluminum composites under uniaxial tensile loading.
- Bong Kyu Choi, Gil Ho Yoon, Seungjun Lee
https://www.sciencedirect.com/science/article/pii/S1359836816000263

Name: Ishan Bharti
Roll No.: 190005019

MM309: Computational Methods for Materials

Research Paper Report

Molecular dynamics simulations of tensile deformation of gradient nano-grained copper film.

● Overview

Molecular dynamic simulations have been widely used to investigate the plastic deformation of nanocrystalline metals. This report performs molecular dynamics simulations to investigate the plastic deformation of a gradient nano-grained (GNG) copper film.

Gradient nano-grained (GNG) metals are a unique class of materials with spatial gradients in grain size, typically from the surface to the bulk. In which the grain size increases along with the depth from the nanoscale to microscale.

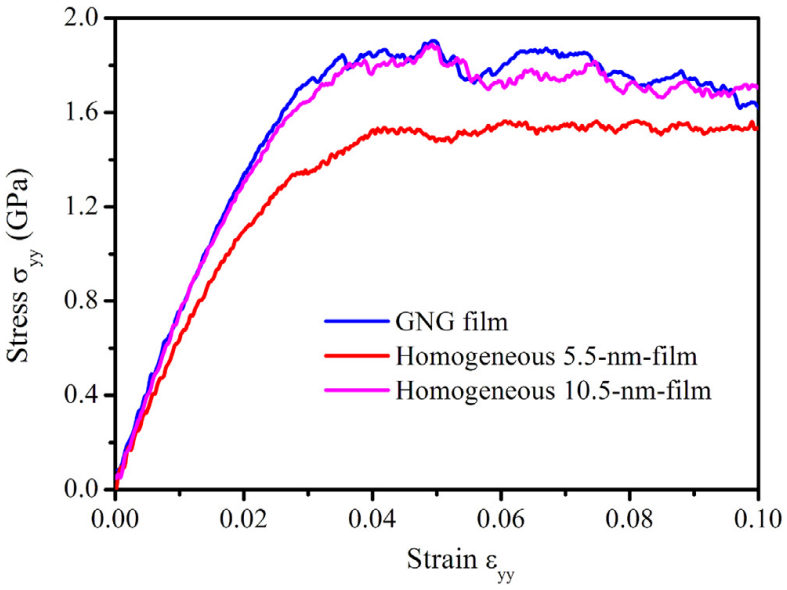
● Introduction

Nano-grained (NG) metals usually exhibit several times higher strength than their coarse-grained counterparts but a very low tensile ductility, so gradient microstructure which helps increase the ductility of nanostructured metals is used. The plastic deformation of a copper sample with a gradient microstructure was observed to occur orderly from coarse grains to small grains with increasing load. In this work, MD simulations are utilised to explore the atomistic deformation system of GNG copper film.

The simulation results reveal a different plastic deformation behaviour of the GNG film compared to the GNG layer adherent on a coarse-grained copper substrate. For the GNG film, the plastic deformation is observed gradually propagating from small grain-size layer to large grain-size layer with increasing strain. An extra strengthening induced by the GNG structure is also observed. Additionally, the influence of strain rates and temperature on the plastic deformation behaviour of the GNG copper film is investigated.

● Simulation details

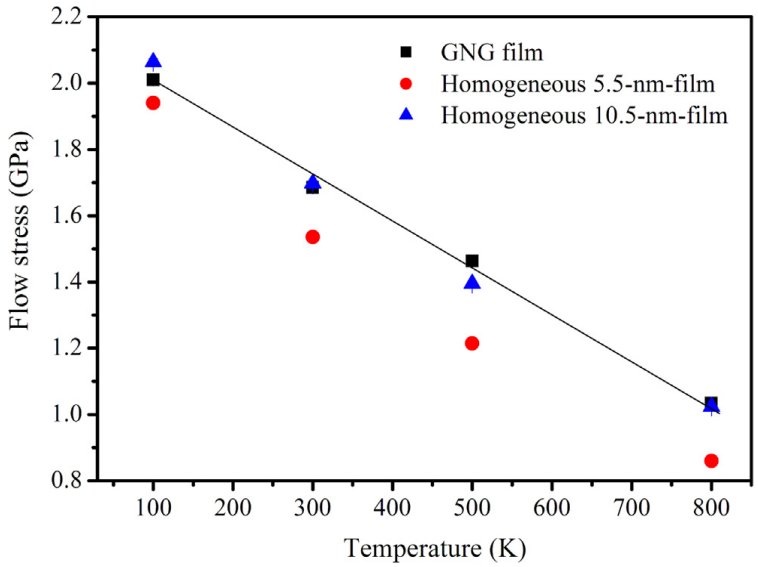
MD simulations were carried out using the LAMMPS code. An embedded-atom method (EAM) potential developed by Mishinet al. was used to describe the interaction between copper atoms. Three fully 3-dimensional models, i.e., one GNG model and two homogeneous NG models, were constructed using the Voronoi tessellation technique. To simulate the uniaxial tension of GNG or



NG films, a periodic boundary condition was applied along the y and z directions, and a non-periodic boundary condition was imposed in the x-direction during the MD simulations. Using the conjugate gradient method, energy minimisations were performed on three models first, and then they were relaxed at 300 K for at least 50 ps using the NPT ensemble. The graph represents the simulated stress-strain curves of the GNG film and the homogeneous NG films with mean grain sizes of 5.5nm and 10.5 nm. The simulation result shows the strength of the GNG film is nearly the same as that of the homogeneous 10.5-nm-film.

We can see that the plastic deformation initially occurs in small grains and propagates into larger grains with increasing strain at a strain rate of $1x10^{9}$ s^{-1}. Flow stress vs temperature graph indicates that the strain rate does not influence the orderly plastic deformation induced by the gradient structure.

The flow stress of the GNG film and the homogeneous NG film as a function of temperature is presented in the above figure. The flow stress decreases linearly with increasing temperature, indicating a softening of the GNG film and the homogeneous NG films.



• Conclusion

Uniaxial tensile deformation of the GNG film has been investigated by MD simulations. Extra strengthening is observed in the GNG film compared with the rule-of-mixture. Orderly plastic deformation is observed and attributed to the GNG structure. The onset of plastic deformation of the GNG film occurs first at small grains and then propagates into larger grains with increasing strain. Orderly plastic deformation is not influenced by strain rate and temperature. The stress distribution in the cross-section is inhomogeneous.

• References

https://www.sciencedirect.com/science/article/pii/S0927025617306079

Name - Ankit Pandey Roll No. 190005004

MM309 Computational Methods for Materials Research paper Report

Molecular dynamics study of the thermodynamic and kinetic properties of the solid-liquid interface in Fe Mn

● Abstract

In this Paper a new MD simulation method for computing solute trapping behavior that is able to access relatively low interface velocities. The segregation coefficient as a function of velocity for the Fe-Mn system was determined for three temperatures and the results are discussed within the framework of previously proposed theoretical models.

● Theory

Investigated the solidification cracking susceptibility in Fe-Mn-C and Fe-Mn-Al-C steels and found that cracking behavior depends on the segregation of Al and Mn in the dendritic microstructure.
The phase-field model has been shown to be a very effective tool for simulating the growth and solute segregation behavior of dendrites and dendrite arrays

● Simulation Procedures

To determine the Fe-Mn phase diagram the approximate liquidus points were computed for a range of Mn compositions (5%, 10%, 30%). We created a solid-liquid system where the composition was equal in both phases and estimated the temperature at which the phase fraction remains constant over the MD run. The next step in the phase diagram calculation

utilized the semi-grand canonical (SGMC) feature of lamps. The chemical potential difference between Mn and Fe(μΔ) was fixed and atom type changes were attempted and accepted or rejected using the Metropolis algorithm.
Interface position vs. time for two separate runs in the Fe-Mn system. In both cases the Mn concentration
The interfacial energy for the final composition is in reasonable agreement with the MD results, the other two experimental data points are substantially higher

Conclusion

From the Result Plot and discussion of the Research Paper, the conclusions are as follow

- The second nearest neighbor MEAM potential for the Fe-Mn system exhibits an equilibrium phase diagram in qualitative agreement.
- The capillary fluctuation method was used to compute the solid-liquid interfacial free energy.
- Solute trapping behavior at fairly low velocities was studied using a free solidification technique. The kV() function is characterized by the diffusive speed and we find VD varies from≈ 1.1 m/s at T = 1603 K to.45 m/s at T = 1460 K.

Reference

https://www.sciencedirect.com/science/article/pii/S0927025620302640

Name:- Piyush Nimbhore
Roll no:- 190005028

MM309: COMPUTATIONAL METHODS FOR MATERIALS

Research Paper Report

MOLECULAR DYNAMICS SIMULATIONS OF SUBSTITUTIONAL DIFFUSION

By - X.W. Zhou , R.E. Jones, J. Gruber
Sandia National Laboratories, USA

- **OBJECTIVE –**

  The Objective Of The Present Work Is To Provide Researchers With A Good Reference As How MD Methods Can Be Used To Calculate Highly Converged Arrhenius Plots For Substitutional Diffusion, And What Role Vacancies Play In Such Calculations.

- **APPROACH USED –**

  $In_xGa_{1-X}N$ Is Chosen Because It Is An Important Solid-State Lighting Material Whose Properties Can Be Improved Through Control Of Kinetics Of The Synthesis Processes.

  LAMMPS Is Used For All MD Simulations Under Periodic Boundary Conditions And A Zero Pressure NPT (Constant Number Of Atoms, Pressure, and Temperature) NOSÈ-HOOVER Thermostat/Barostat.Varying Time range [Between 0.5 and 1000.0 ns ($t = 0.004$ ns for t 20.0 ns, and $t = 0.1$ ns for $t > 20.0$ ns)] , temperature range ($T^*/ T^*_m = 0.87–0.98$) , and Vacancy Concentrations ($C_V = 1\% - 5\%$) are Taken into Considerations for Plotting the Graphs .

- **CONCLUSIONS -**

1. Highly Converged and Errorless Results are Obtained When Using An Elevated Temperature Range, An Extended Simulation Time And Various Vacancy Concentrations, therefore being really helpful for good results in research purposes.

2. The Observed Improvement Of Convergence Due To Increased Temperature, Time, And Vacancy Concentration Can All Be Attributed To Increased Sampling Of Diffusion Events During Simulations.

3. With The Help Of Vacancies, MD Simulations Can Be Used To Equilibrate Atomic Populations In Substitutional Lattices.

- **REFERENCES –**

1) Molecular dynamics simulations of substitutional diffusion
X.W. Zhou , R.E. Jones, J. Gruber
https://www.sciencedirect.com/science/article/pii/S0927025616306085

Name:- Pradumn Patidar
Roll No:- 190005030

Research Paper Report

Solute Partition at Solid-liquid Interface of Binary Alloy From Molecular Dynamics Simulation.

Overview

In this paper, molecular dynamics simulations are performed to estimate the partition-based properties such as solidification and interfacial dynamics. The main aim of the simulation is to control the solidification microstructure to obtain products with desired properties. At present, it is difficult to know solute partition-based properties from the MD simulation, therefore composition ratio of a binary alloy can be estimated using the semi-grand canonical Monte-Carlo (SGCMC) simulation. The main advantage of the semi-grand canonical Monte-Carlo simulation is that the relationship between the entire free energy and composition at any temperature is derived effectively, which is the driving force for solidification.

Approach Used

MD simulations are performed using the LAMMPS for the Fe-Cr binary alloy system.
For the sample preparation,

1. For alloys, concentration is annealed at 500 and 5000 K for 10 ps with the NVT ensemble.
2. Two bcc crystals for pure Fe and Cr. Then, these structures are connected with the (100) plane appearing on the interface.
3. Melting points of pure Fe and Cr are estimated by the convergence temperature technique, first these are used with NPT for 0-25 ps and for NPH for subsequent 25-1975 ps , the pressure is maintained at 0 Pa in all processes.

The EAM potential fitted by Bonny et al. The velocity-Verlet method is used. The Nose–Hoover thermostat and barostat are used to control temperature and pressure throughout the simulation. Finally , The OVITO (open visualization tool) is used for the visualization of simulation results.

As an example, the OVITO clip shows the motion of a solidification in Fe-Cr where the temperature of the system is carefully thermostatted so that the velocity of the interface can be accurately measured can be seen ----- HERE

Conclusions

From this simulation, I have observed that the partition-based properties including interfacial dynamics such as solidus-liquidus composition, $eq^{b}m$ temperature , phase diagram of Fe-Cr , propagation velocity and kinetic coefficient are estimated.

Fe-Cr composition & Equilibrium Temperature.
Simulation results are observed at different frame rates (3,000-15,000 ps) and the convergence compositions are estimated to be 7% and 14% for Fe-90%Cr at 2500 K and 12% and 23% for Fe-85%Cr at 2400 K, respectively. The interface propagates toward the liquid phase and solute partition happens at the solid–liquid interface during the solidification.
For the Fe-10%Cr during the NPH ensemble, the equilibrium temperature is 1773K

Fe-Cr Phase Diagram.
Phase diagram of iron carbon is plotted with the help of EAM potential near the equilibrium temperature, and the Miscibility gap (range of temperature and composition on the phase diagram where a phase that is stable at higher temperatures decomposes into two or more phases, example Spinodal Decomposition) are also revealed.
The phase diagram helps in further study of microstructure evolution.

Propagation velocity & Kinetic coefficient.
The driving force for interface propagation depends on both temperature and composition in a binary alloy. Propagation is a diffusion-controlled process during solid partition, the effect of diffusion is not dominant during the partitionless solidification. For the different compositions of binary alloy (Fe-83%Cr (solid), Fe-67%Cr (liquid) and Fe-19.1%Cr.) the propagation velocity of the interface in this system is examined between 2000 and 2550 K at 50 K interval. Therefore, kinetic coefficient is estimated which is useful in evaluating the crystal-melt growth interface velocity.

References

1. Solute partition at solid-liquid interface of binary alloy from molecular dynamics simulation.
Kensho Ueno, Yasushi Shibuta
https://www.sciencedirect.com/science/article/abs/pii/S2589152918301996

Name:- Ayush Nath

Roll No:- 190005009

Molecular dynamics simulations of wetting behavior of water droplets on polytetrafluorethylene surfaces

Introduction

Polytetrafluorethylene (PTFE) is a widely used engineering material in aviation, mechanics, and other fields, especially because of its hydrophobicity. The nanostructure of the solid surface has an important influence on its hydrophobicity. In addition, because of the interest in designing microfluidic systems, the nano-wetting properties of surfaces are attracting increasing attention.

Aim

- Study the effect of cutoff on the contact angle of the flat surface and the variation trend of the contact angles.
- Investigate the differences between the two equilibrium states and the reasons why cross contact exists. This research is significant for studying the characteristics of the water–solid interface, including the wettabilities of different nanostructured surfaces.

Simulation Model

For the simulations, a model composed of a cubic box of water molecules located on the solid surface was constructed.

The water model in this study was the rigid extended simple point charge potential (SPC/E). The charges on the oxygen and hydrogen atoms

were –0.8476e and +0.4238e. The O–H distance was 1 Å and the H–O–H angle was 109.47°.

MD Simulation using Lammps package

The mixed-atom interatomic potentials were obtained through the Lorentz–Berthelot mixing rules.

The electrostatic interaction was modeled using Coulomb's law, and the PPPM method was applied to minimize the errors in the long-range interactions. All simulations were carried out in the NVT (fixed number of particles, volume, and temperature) ensemble at T = 300 K for 1 ns with an integration time step of 1 fs.

Variation of contact angle with water droplet size for different cutoffs

The results in Figure 2 show that the sensitivity of the contact angle to the number of water molecules is low. The contact angle at a cutoff of 9 Å changed from 98° to 103°, which is a small variation. This phenomenon was similar for the other two cutoffs. Increasing the cutoff increases the simulation accuracy, as well as increasing the simulation time.

Selecting the cutoff for simulation

The deviation of the potential energy at a cutoff of 9 Å was as high as 9.93%, which also corresponds to the deviation of the contact angles (10%). Therefore, it can be concluded that the effect of cutoff on the simulated results could be indirectly reflected by the deviation of the potential energy. Therefore, to obtain an appropriate cutoff for the simulations, calculations of potential energy can be used. It also showed that the deviations decreased to ~2% when the cutoff was increased to 15 Å, suggesting that a cutoff of 15 Å is acceptable for this system. Considering the deviation and simulation time, 15 Å was selected as the cutoff for the simulations.

Variation of contact angle for droplet size for flat, P1, P2 surfaces of PTFE.
The value of the contact angle is dependent on the size of the droplet for the wetted contact. Also shows that the contact angle increases with increasing droplet size. It also revealed that the contact angle of the P2 surface was always larger than that of the P1 surface for the same-sized droplet. For a same-sized water droplet, the droplet left a larger volume of liquid between pillars of the P1 surface because the volume of the hollows on the P1 surface was larger than that on the P2 surface. In this case, the water formed a smaller droplet above the P1 surface, which resulted in a smaller contact angle. In summary, the contact angles of wetted contact were affected by the proportion of the volume of the water droplet confined between pillars to the volume of the water droplet above the surface. For the P1 and P2 surfaces, because of the same value of the whole area per unit (Eq. (9)), it can be concluded that the contact angle increases with increasing droplet size and pillar area ratio f.

Variation of contact angle for droplet size for flat, P3, P4 surfaces of PTFE.
Contact angle of different droplet sizes remains stable. Contact angles of cross-contact had weak sensitivity to the droplet sizes. For the same-sized droplet, the contact angle of the P3 surface is larger than that of the flat surface and smaller than that of the P4 surface. The increase of contact angles from P3 surface to P4 surface could be seen to follow the decrease in the contact area of the liquid with the solid.

Conclusion

This paper provided a molecular-level description of the wetting behaviour of nanosized droplets on the solid surface of PTFE.

Some of the key conclusions drawn are :-

- The effect of the cutoff of the interatomic potential on the simulation accuracy.
- On flat surfaces, contact angles of the PTFE surface were near 90° and independent of droplet size.
- On pillar surfaces, wetted contact and cross-contact appeared.
- For wetted contact, contact were dependent on droplet size and pillar dimensions within a certain range.
- For cross contact, contact angle values remained stable.
- Additional simulations indicate the final state depends on the initial geometry and cross-contact is a metastable wetting state.

References

Molecular dynamics simulations of wetting behavior of water droplets on polytetrafluorethylene surfaces.
Chen S, Wang J, Ma T, Chen D. J Chem Phys. 2014 Mar 21;140(11):114704. doi: 10.1063/1.4868641. PMID: 24655195.
https://www.researchgate.net/publication/261031116_Molecular_dynamics_simulations_of_wetting_behavior_of_water_droplets_on_polytetrafluorethylene_surfaces