Molecular Dynamics Homework

Taavi Tammaru

November 28, 2023

1 Introduction

The assignment was to study the properties of single-walled carbon nanotubes with molecular dynamic simulations. More specifically using LAMMPS software. The goal was to get the CNT to the breaking point and analyze the conditions that led up to that point.

2 Methodology

The LAMMPS script was written based on the instructions given. I used the usual Notepad text editor on Windows. The files for the two potentials were identical apart from the potential models and data file.
The code for the Tersoff potential:

```
# General Parameters
units metal
dimension 3
boundary f f p
atom_style atomic

read_data tersoff.data

# Define Potential Models
pair_style tersoff
pair_coeff * * BNC.tersoff C

# Simulation Parameters
timestep 0.001
mass 1 12.011
velocity all create 300.0 4928459 dist gaussian

# Integrator with Thermostat and Barostat
fix npt all npt temp 300.0 300.0 0.1 z 0.001 1 0.1

# Output Options
thermo 100  # Adjust frequency as needed
thermo_style custom lx ly lz pzz temp

shell mkdir dump

# Dump Command for Snapshot
dump myDump all custom 100 ./dump/dump* id x y z

# Run Simulation
run 5000
```

```
write_data equilibrated_datafile.txt

# Step 1: Save initial box dimensions
variable l0 equal lz

variable pzz equal pzz

# Step 2: Define strain variable
variable strain equal "(lz - v_l0) / v_l0"

# Step 3: (Optional) Reset timestep
#reset_timestep 0

# Step 4: Unfix NPT fix
unfix npt

# Step 5: Define new NVT fix at 300 K
fix 2 all nvt temp 300.0 300.0 1.0

# Step 6: Deform the simulation box in the z-direction
fix 3 all deform 1 z scale 1.3 remap x

# Step 7: Output strain and stress every 200 timesteps
fix 4 all print 200 "${strain}" file strain.txt screen no
fix 5 all print 200 "${pzz}" file stress.txt screen no


shell mkdir dumpy

# Step 8: Adjust dump configuration
undump myDump
dump 2 all custom 200 ./dumpy/dump_strain*.txt id x y z

# Step 9: Run the simulation for 50 ps
run 50000
```

Then I used a python script to make arrays out of the pzz and strain output so I could use it later to make a graph:

```python
with open('strain.txt', 'r') as input_file:

    values = [line.strip() for line in input_file.readlines()]

formatted_values = ','.join(values)

# Print the formatted string
print(formatted_values)
```

3 AIREBO Results

3.1 Mean Atomic Distance

The mean atomic distance for the system at 300 K was 1.4

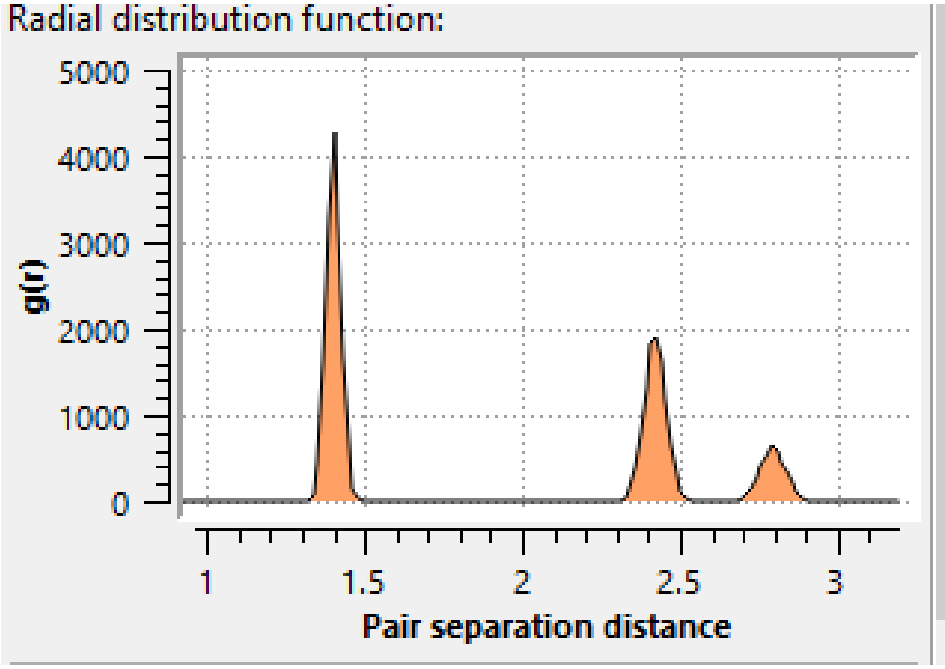

Figure 1: mean atomic distance

3.2 Stress-Strain Graph

The stress strain graph was found using the python code:

```python
import matplotlib.pyplot as plt
import math

# Chirality index
n = 20

# Mean atomic distance
a = 1.4 # Replace with your actual mean atomic distance value

h = a  # Assuming thickness is the same as the mean atomic distance

# Nanotube length in the z-direction
lz = 209 # Replace with your actual nanotube length in the z-direction

l = 197

def calculate_preal(pzz, l, a, n, lz):
    # Calculate diameter
    d = (math.sqrt(3) * n * a) / math.pi

    # Thickness of the nanotube
    h = a  # Assuming thickness is the same as the mean atomic distance
    # Calculate preal
    preal = (l**2 * pzz) / (math.pi * d * h)
```

```
    return preal

pzz_values = [-20.6259807630532,-68.3291941735122, ...]

# Your strain and stress data
strain_values = [1.22022771096115e-15,9.47928483234349e-16 ...]
stress_values = [calculate_preal(pzz, l, a, n, lz) for pzz in pzz_values]

# Convert stress to positive values (optional)
#stress_values = [-stress for stress in stress_values]

# Create the stress-strain graph
plt.plot(strain_values, stress_values, label='Stress-Strain Curve')

# Add labels and title
plt.xlabel('Strain')
plt.ylabel('Stress')
plt.title('Stress-Strain Graph')

# Add legend
plt.legend()

# Show the graph
plt.show()
```

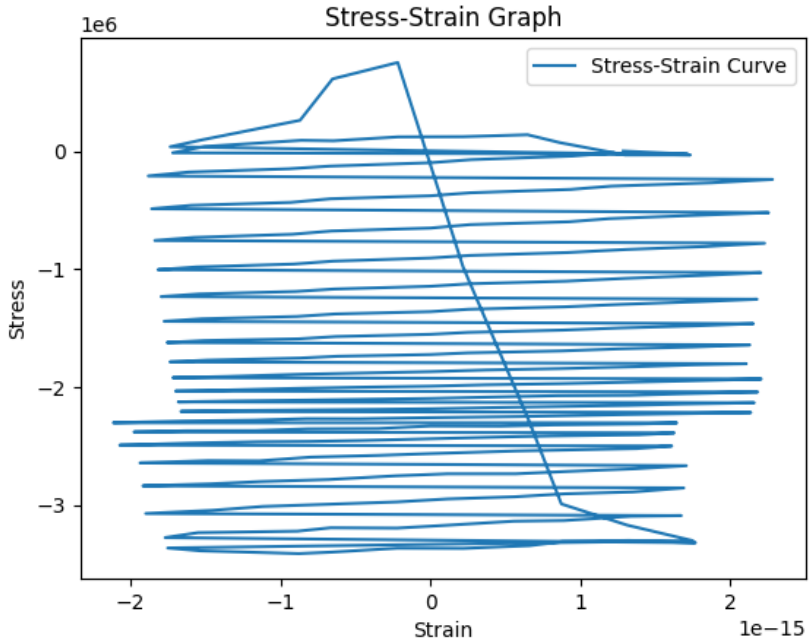

Figure 2: Stress-Strain Graph

3.3 Young Modulus

taking only the first few datapoints we get that Youngs Modulus is equal to:

$$E = 7.66330562997713e + 19 \tag{1}$$

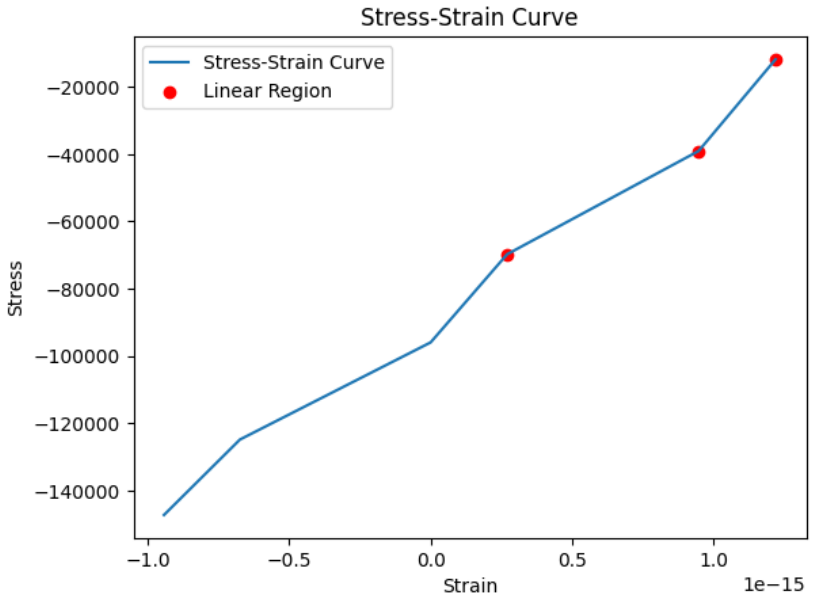

Figure 3: Stress-Strain Graph

3.4 Maximal Strain and Stress Before Breaking

Max stress = 755092,1084136028

Max strain = 2.28295816674807e-15

4 Tersoff Results

4.1 Mean Atomic Distance

The mean atomic distance for the system at 300 K was 1.5

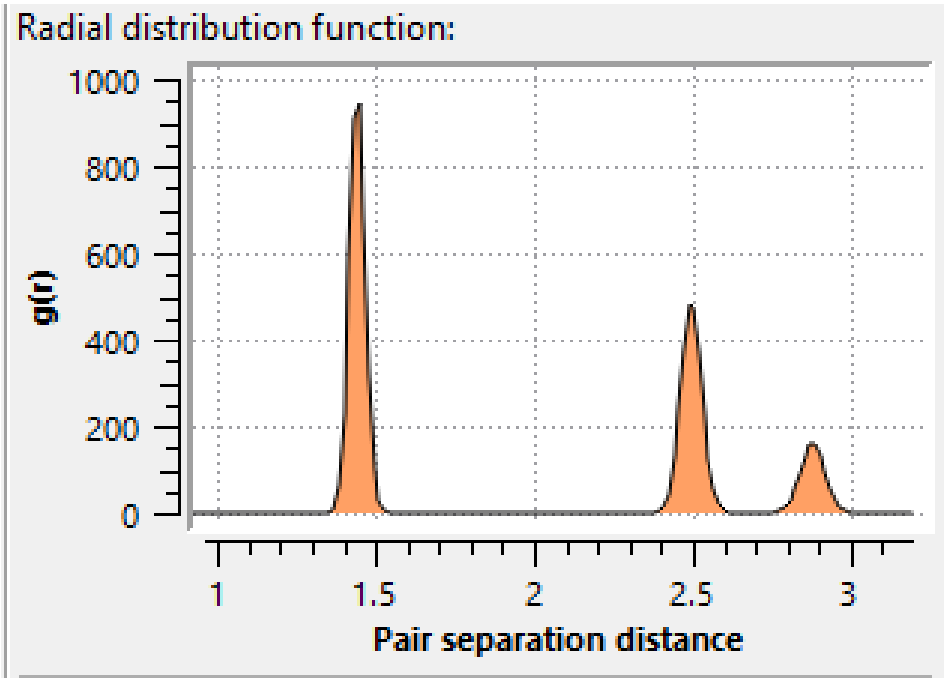

Figure 4: mean atomic distance

4.2 Stress-Strain Graph

The stress strain graph was found using the python code as with the other potential.

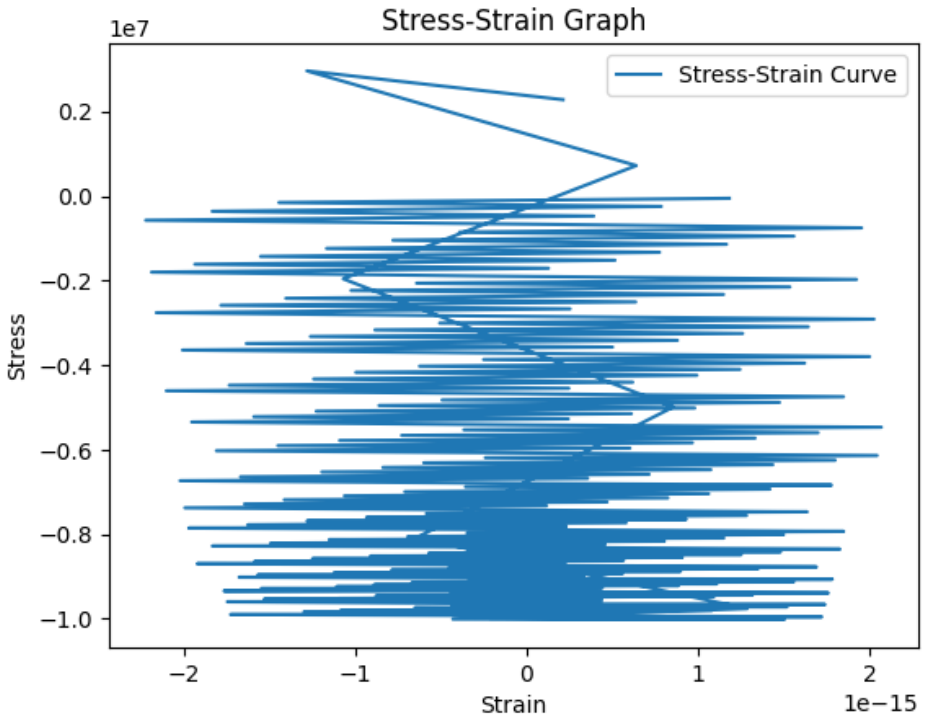

Figure 5: Stress-Strain Graph

4.3 Young Modulus

$$E = \frac{\text{stress}}{\text{strain}} \tag{2}$$

taking only the first few datapoints we get that Youngs Modulus is equal to:

$$E = 1.6368527297350733e + 20 \tag{3}$$

4.4 Maximal Strain and Stress Before Breaking

Obtained using python:

Max stress = 2958615,0907886303

Max strain = 2.07109620523779e-15

5 Comparison of Potentials

The results obtained with different potentials do not agree, the Young modulus is about a magnitude off. The maximum strain agrees while the maximum stress does not.

6 Snapshots from OVITO

A few snapshots of the breaking process from OVITO:

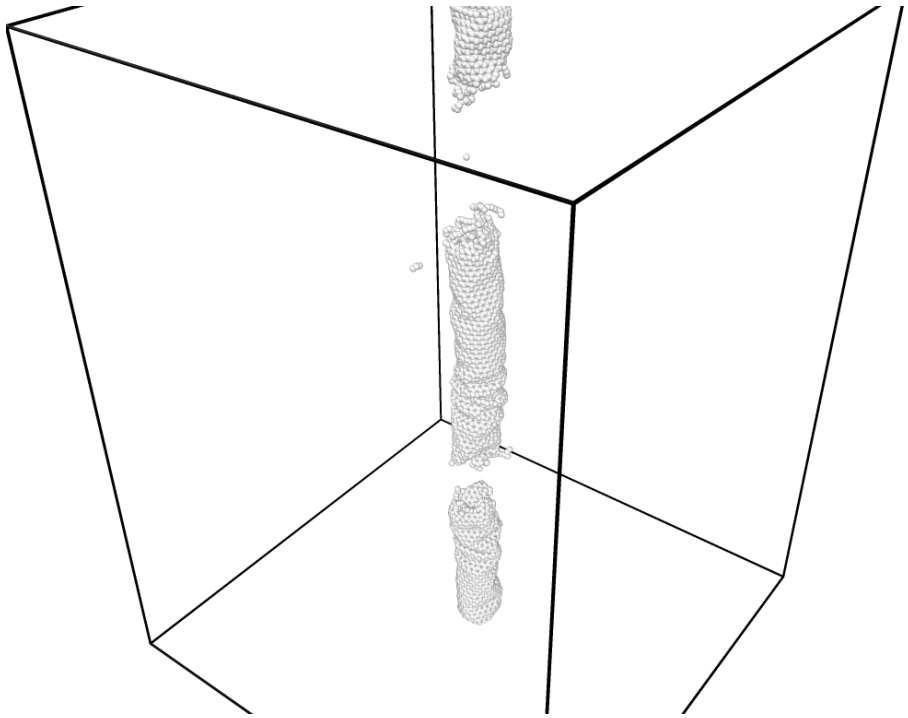

Figure 6: AIREBO potential breaking Process

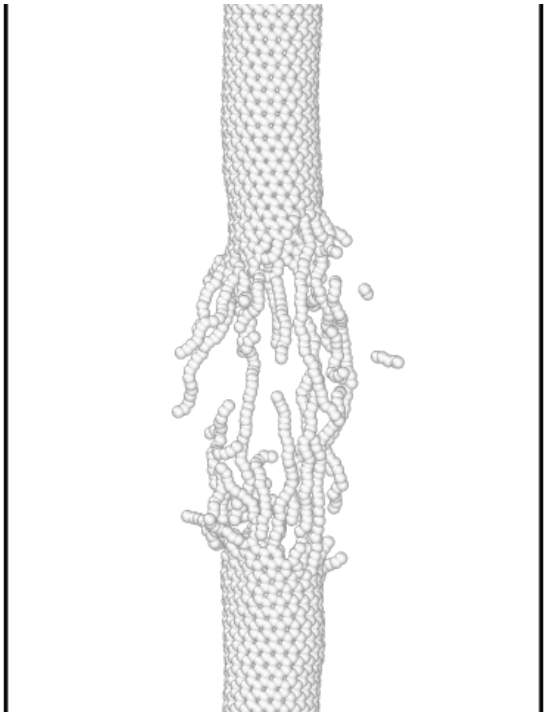

Figure 7: Tersoff potential breaking Process

7 Conclusion

The AIREBO potential simulation surprisingly took around 40 minutes to run while the Tersoff simulation only took around 15 minutes. Furthermore, I found interesting that while using the AIREBO potential the CNT broke from two places at simultaneously but while using the Tersoff potential it broke only from the center.