

# Studies of Temperature Effects on a Morse Lattice via Molecular Dynamics

Robert Hart<sup>#1</sup>, Gregory Tanner<sup>\*2</sup>, Shaoping Xiao<sup>#3</sup>

<sup>#</sup>Department of Mechanical and Industrial Engineering, The University of Iowa  
College of Engineering, The University of Iowa, Iowa City, IA, USA

<sup>1</sup>robert-hart@uiowa.edu

<sup>3</sup>shaoping-xiao@uiowa.edu

<sup>\*</sup>Program of Applied Mathematical and Computational Sciences, The University of Iowa  
College of Liberal and Arts, The University of Iowa, Iowa City, IA, USA

<sup>2</sup>gregory-tanner@uiowa.edu

**Abstract**— A "Morse" lattice, a square lattice bonded via the Morse potential, was studied in this paper via molecular dynamics to demonstrate the temperature effects on the stress and strain of the nanostructured materials. Only two-body potential was used here without considering angle bending or torsion of bonds. Velocity scaling technique was implemented in molecular dynamics to maintain the simulated system at a desired temperature. Thermal stresses were observed when volume expansion was not allowed. When a positive strain was applied to the lattice, stress relaxation was observed as the temperature increased. Uniaxial tension simulations were conducted at various temperatures to study stress-strain curves. It was found that the material failure stress and strain were smaller at high temperature. In addition, the histograms of atomistic velocities at the final equilibrium states were studied and compared to the theoretical Boltzmann distribution at various.

**Keywords**— molecular dynamics, Morse lattice, temperature, stress.

## I. INTRODUCTION

Recent developments in nanotechnology demand that nano/micro scale building blocks will complement and enhance new engineering techniques in the relevant technology areas, including materials science. It has been known that materials with specific structures at the nanoscale often have unique optical, electronic, or mechanical properties. For example, carbon nanotubes ([1], [2]), which have been known as unique-structured materials, have extraordinary material and electrical properties. They have been used to design novel nanoscale composites and devices. Therefore, studying mechanical behaviours of nano-structured materials become vital in the current revolution of material and machine design.

Numerical methods have been playing an important role in engineering design procedures. In specific, molecular dynamics (MD) is a powerful tool to clarify complex physical phenomena at the nanoscale. MD employs classical Lagrangian mechanics, and the motion of atoms follow Newton's second law. MD has been used to study mechanical behavior of nanocomposites ([3]-[5]) and nanodevices ([6]-[7]) as well as the material phase changing problems [8]. However, it has been known that MD has limitations in both length and time scales.

More and more research studies have been conducted to elucidate how materials behave differently at different scales.

For example, in ICE melting phenomena, the melting speeds are different at different scales [9], including both length and time scales. Therefore, the multiscale methods, which can handle modeling and simulation in multiple length and time scales, draw attentions from researchers and scientists. One type of the multiscale methods is called concurrent multiscale methods, which employ different methods at different scale simultaneously. Some concurrent multiscale methods couple the molecular domain with the continuum domain. The representative works include the MAAD (Macro-Atomistic-Ab initio-Dynamics) method [10], the bridging domain coupling method ([11]-[13]), and the bridging scale method ([14], [15]). The bridging domain coupling method has been used to study nanocomposites [16] and nanodevices ([17]-[19]). Another multiscale approach is using homogenization techniques ([20]-[22]) to link the continuum model with the molecular model.

Temperature effects on the material behaviors are always interesting topics in various communities. Temperature is known as a macroscale quantity representing the averaged atomistic kinetic energy at the nanoscale. Mathematical heat equations, e.g. the thermal diffusion equation, are only derived for the continuum model. In addition, the thermal diffusion equation becomes invalid when the scale moves down the microscale from the macroscale. At the microscale level, the thermal wave equation [23] shall be used. At the nanoscale, temperature regulation is always used to maintain the simulated system at a constant temperature when using MD. The temperature can be evaluated based on the averaged atomistic kinetic energy. It should be noted that a temperature related homogenization technique ([24]-[26]) can be employed in the nanoscale continuum approximation so that the temperature effects at the nanoscale can be studied via the continuum model. At the nanoscale, it has been investigated that the crack propagation speeds were different at different temperatures [25].

In this paper, MD was used to study temperature effects on the stress and strain of a Morse lattice. The thermal stress and the stress strain curves at various temperatures were studied. The distribution of atomistic velocities at different temperatures were plotted and compared to the theoretical Boltzmann distributions. The outline is described as below. After introduction, the methodology is described in section 2. The simulation procedure and the developed Matlab code (in

Appendix) are described in section 3. Results are discussed in section 4 followed by the conclusions.

## II. METHODOLOGY

### A. Molecular dynamics

In the MD simulation, the atoms or molecules in the simulated system follow the laws of classical mechanics. The motion of an atom, e.g. atom  $i$ , with mass  $m_i$ , is due to its interaction with other atoms in the system according to Newton's second law:

$$m_i \vec{a}_i = \vec{f}_i = -\nabla W = -\frac{\partial W}{\partial \vec{r}_i} \quad (1)$$

where  $\vec{a}_i$  is the acceleration of atom  $i$ , and the interatomic force,  $\vec{f}_i$ , applied on atom  $i$ , is derived from the total potential energy,

$$W(\vec{r}_i) = \frac{1}{2} \sum_{ij} W_{ij}(\vec{r}_i, \vec{r}_j) \quad (2)$$

where  $\vec{r}_i$  and  $\vec{r}_j$  are the atomic positions of atoms  $i$  and  $j$  respectively.  $W_{ij}$  is the potential function to describe the interaction between atoms  $i$  and  $j$ .

During the MD simulation, the accelerations are calculated based on the forces, i.e. Eq. (1). The velocities are then determined by integrating acceleration, and positions by integrating velocity. In this paper, the time integrations are performed using the velocity Verlet method given below.

$$u(t + \Delta t) = u(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 \quad (3)$$

$$a(t + \Delta t) = -\frac{1}{m}\nabla W(u(t + \Delta t)) \quad (4)$$

$$v(t + \Delta t) = v(t) + \frac{\Delta t}{2}(a(t) + a(t + \Delta t)) \quad (5)$$

One of key issues in MD simulation is the temperature regulation because the temperature of the simulated system would increase infinitely due to the work done by externally prescribed force or displacement if only modeling the simulated system as an isolated one. To model the heat dissipation from the simulated system to the surrounding media, a numerical heat bath is implemented in MD simulations. A classic approach to adding or subtracting kinetic energy to the system is multiplying the velocities of all particles with the same global factor, so that the temperature can be maintained. In this simplest version of velocity rescaling, the factor is chosen to keep the kinetic energy constant at each time step by correcting the velocities obtained above as

$$v^*(t + \Delta t) = \sqrt{\frac{T_o}{T_t}} v(t + \Delta t) \quad (6)$$

where  $T_o$  is the desired temperature and  $T_t$  is the temperature of the system evaluated after calculating new velocities via Eq. (5) at each time step. It shall be noted that the evaluation of temperature is based on the averaged atomistic kinetic energy of the system. After the velocities are corrected,  $v^*$  is the one used to update motion of the atoms in the next time step.

### B. Morse lattice

The purpose of this paper is to show the temperature effects on stress analysis of nanostructured materials. The material

used in the current model is a non-existing material. The lattice structure of the atoms is a square-shaped lattice with all bond lengths being the same, i.e. the equilibrium bond length of  $r_0 = 1.42 \text{ nm}$ . The atoms have the same mass of  $1.998467e-26 \text{ kg}$ . The potential function used here to describe the interaction between bonded atoms is the Morse potential function [5]. The potential function and its resulting bond force are functions of bond length  $r$  and given as

$$W(r) = D_e [e^{-2\beta(r-r_0)} - 2e^{-\beta(r-r_0)}] \quad (7)$$

$$F(r) = 2D_e\beta [e^{-2\beta(r-r_0)} - e^{-\beta(r-r_0)}] \quad (8)$$

where the parameters include  $D_e = 0.603105 \text{ nN} \cdot \text{nm}$  and  $\beta = 26.25 \text{ nm}^{-1}$ . It should be noted that we only consider two body potential in this paper.

In the initial state, the atoms were placed in a square lattice structure with initial bond length at or very near the equilibrium bond length. Each atom was given a small, random initial velocity. Fig. 1 shows a sample vector plot showing the direction and magnitude of the initial velocities to a group of atoms.

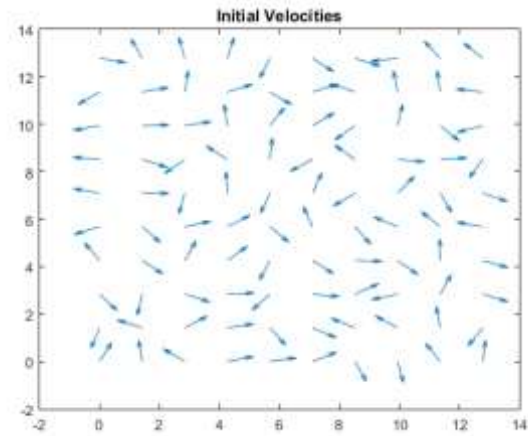


Fig.1 Vector plot of initial velocities applied to the atoms in the lattice structure

### C. Maxwell-Boltzmann distribution

In an equilibrium system at the nanoscale, the velocities (energies) of the atoms will vary greatly, with some atoms at very low velocity (low energy) and others with very high velocity (high energy). The standard Maxwell-Boltzmann distribution plots the probability of atoms at a given velocity in three-dimensional space. A two-dimensional Maxwell-Boltzmann can be derived from the assumption that the velocity in each direction follows an independent Boltzmann distribution given by

$$P(v_x)dv_x = \sqrt{\frac{m}{2\pi k_B T}} \exp\left(-\frac{mv_x^2}{2k_B T}\right) dv_x \quad (9)$$

The joint probability function is just the product of the probabilities (under the assumption of independence).

$$P(v_x, v_y)dv_x dv_y = \frac{m}{2\pi k_B T} \exp\left(-\frac{m(v_x^2 + v_y^2)}{2k_B T}\right) dv_x dv_y \quad (10)$$

In order to compute the probability that a particle has velocity  $v = \sqrt{v_x^2 + v_y^2}$ , the coordinate system is changed to

polar coordinates ( $dv_x dv_y = v dv d\theta$ ) and integrated with respect to the angle  $\theta$

$$\begin{aligned} P(v)dv &= \int_0^{2\pi} \frac{m}{2\pi k_B T} \exp\left(-\frac{mv^2}{2k_B T}\right) v dv d\theta \\ &= \frac{mv}{k_B T} \exp\left(-\frac{mv^2}{2k_B T}\right) dv \end{aligned} \quad (11)$$

### III. SIMULATION PROCEDURE

The simulation procedure is described as below with the interpretation of Matlab code framework. The Matlab code is provided in the appendix.

The first step in the MD Matlab code was to define the potential parameters discussed above in the previous section. Although discussion of the results focuses on the Morse lattice in the next section of this paper, the Lennard-Jones potential was added as an alternative potential function in the Matlab code. For all results presented in this paper, a 100 atom x 100 atom square domain was utilized, which resulted in a simulation of  $N = 10^4$  atoms. The atoms were first arranged in a square grid at the equilibrium bond length, and then each atom was given a small random perturbation. The code for initialization of the lattice structure is shown in the Appendix in lines 53 - 68. The next step was to assign random initial velocities. The equilibrium average velocity was calculated due to the Boltzmann distribution. The average velocity was decomposed into the x and y direction (horizontal and vertical direction) parts with a random variation in velocity, as shown in the Appendix in lines 71-75.

After the model was initiated, the interatomic potential and forces needed to be calculated for each pair of atoms. In the current model, only two-body interactions were considered between atoms directly to the right, left, up, and down in the lattice structure. The atomistic forces were then calculated on each atom for solving the equations of motion. After calculating forces, time integration was performed using the velocity Verlet method (see Appendix lines 119-168). Accelerations were calculated by dividing force by mass at the previous time step. The position was then updated using the acceleration and velocity at the previous time step. The positions of all atoms were adjusted based on the location of the atom at the lower left corner to prevent drifting of the lattice. After updating the positions, the forces were updated using the method described previously. Using the updated forces, stress was calculated at the final time step.

In order to ensure that the temperature remained constant during the MD simulation, at the end of each time step, the velocity needed to be scaled. First the test velocity and average kinetic energy were calculated. These were used to calculate the current temperature, given the kinetic energy. Finally, the velocity was scaled, and the loop was repeated for the next time step, as shown in the Appendix in lines 170 - 180. A flowchart of the general code method is provided below in Fig. 2.

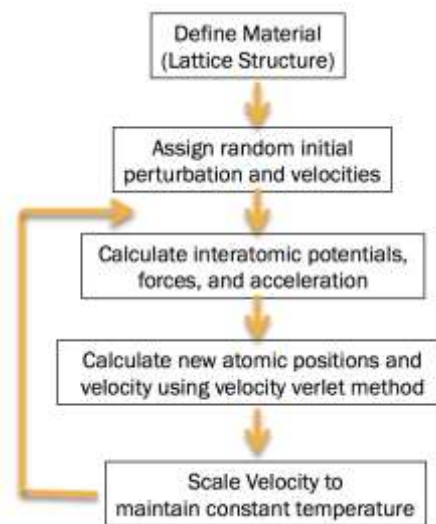


Fig. 2 General Flow Chart of Matlab Code

### IV. RESULTS AND DISCUSSIONS

Generally, when the temperature increases and a material is allowed to expand freely, no thermal stresses develop. In the case of this simulation, however, expansion of the domain was not allowed, thus thermal stresses developed. As shown in Fig. 3 below the temperature had a significant influence on the stress of the Morse lattice.

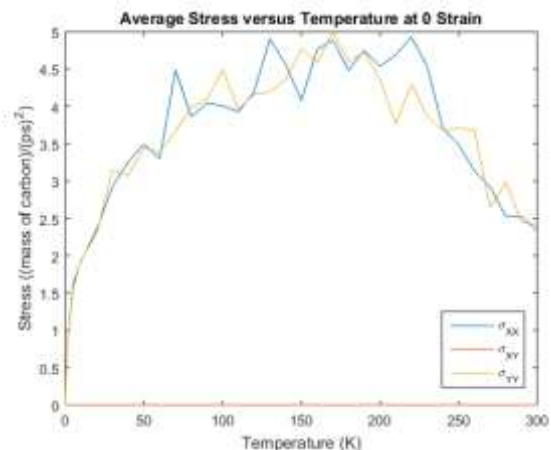


Fig. 3 Plot of Stress versus Temperature

As the temperature increased from 0K to 150 K, the stress in the material increased as expected. This increase in stress was due to the increase in the equilibrium bond length as temperature increases. Since the atoms were restricted within a constant volume, this led to an increase in the interatomic forces and thus an increase in stress. As temperature increased from 200K to 300K, however, the stress began to decrease, which was not expected. In order to determine the cause of this unexpected decrease in stress, further investigation was required. The primary explanation for the decrease in stress after 200K is that at this point, the bonds lengths have been stretched beyond the length at which the peak force is achieved. Beyond this length, the bond force begins to decrease, which leads to a decrease in the stress. A possible

alternative explanation was formed when observing the changes in bond angles, which is not accounted for in the two-body potential in the current model. When observing the localized displacements of the atoms, a possible cause of the decrease in stress was identified. The model only accounted for two-body potential and the potential did not account for any changes in angle between the bonds within the lattice structure. For the simulations between 200K and 300K, the lattice structure began to deform and some of the squares in the lattice structure began to collapse slightly and become parallelogram-shaped, as shown in Fig. 4. It shall be noted that no potentials due to angle bending or torsion were considered so that the Morse lattice can fail even at low temperatures compared to real materials

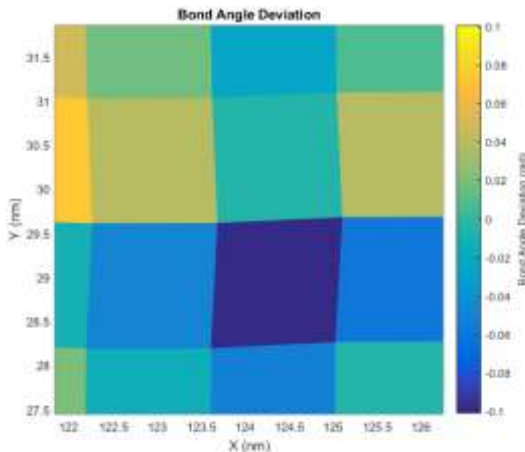


Fig. 4 Deformed lattice structure for 250K simulation

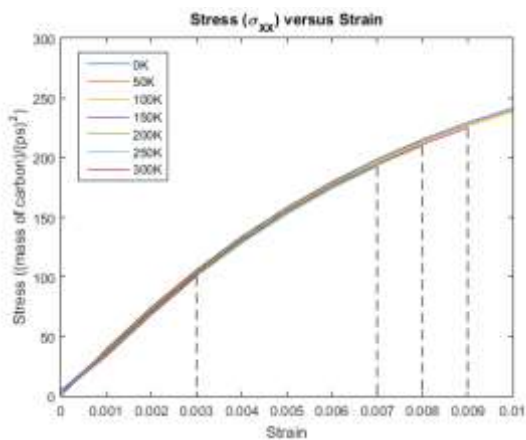


Fig. 5 Stress versus Strain Curves at the temperatures of 0K, 50K, 150K, 200K, 250K, 300K

In order to further study the temperature effect on stress, the domain was stretched in the x-direction from strains of 0 to 0.01 at various temperatures. It has been known that for metals as temperature increases the stress decreases, given constant strain. A similar result was found in the current study for the simulated Morse lattice. The stress versus strain curves are plotted to failure in Fig. 5. Failure was characterized by a sharp decrease in stress or complete failure of the lattice

structure. The first noteworthy trend from Fig. 5 was that for all tests, the stress was non-zero at “zero” strain, which represents the thermal stresses discussed previously. Next, as temperature increased from 0K to 300K, the failure stress decreased, which is a similar trend discussed previously for metals. Another noteworthy takeaway was that at higher temperatures (above 150K), the simulation failed prior to reaching strain of 0.01. This result may be attributed due to instability created by the higher energy state at elevated temperature. In order to better illustrate the influence of temperature on stress, the stress versus temperature curves were plotted at several constant strain values from 0 to 0.010 in increments of 0.001 in Fig. 6 and Fig. 7.

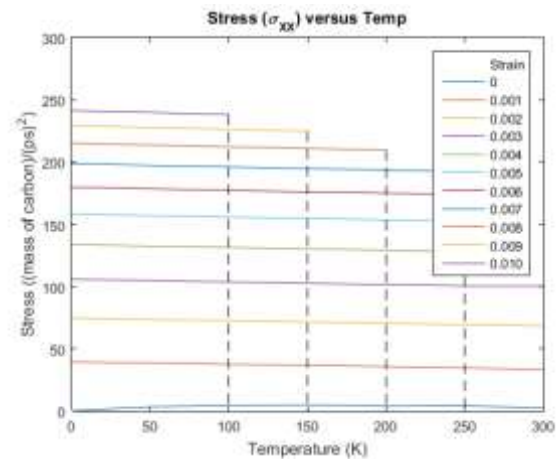


Fig. 6 Stress versus Temperature curves for different strain values from 0 to 0.010

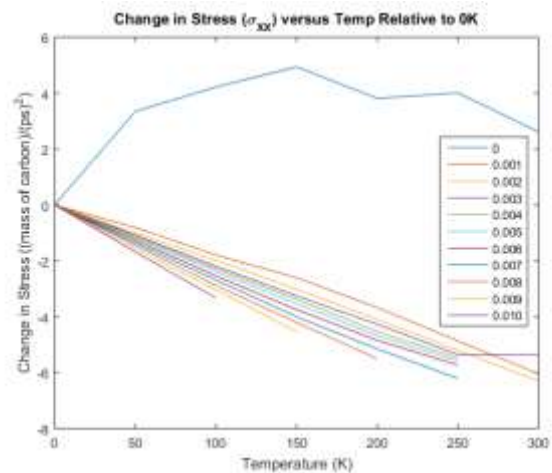


Fig. 7 Change in stress due to temperature relative to zero temperature at various strains

When observing the curves in Fig. 6 and Fig. 7 some interesting temperature effects become apparent. First, the stresses at zero strain (ie: thermal stresses discussed previously) were very small compared to the stresses induced even at the smallest strain studied at 0.001. The zero strain curve was actually the only curve that showed an increase in stress with temperature. For all other curves, the stress decreased as temperature increased. This result shows that the strain applied in the x-direction was sufficient such that the



thermal stresses were relaxed and thus the material exhibited the normal trend that temperature leads to a decrease in stress.

At last, the molecular dynamics simulation was employed to study the temperature effect on the Maxwell-Boltzmann distribution, which shows the probability of atoms at a given energy state. Indeed, the Maxwell-Boltzmann distribution shows how the velocities (energies) of a mixture of atoms vary at a particular temperature. The theoretical distribution, discussed in Section 2, gives the probability of an atom at a given temperature, and integrating the area under the curve gives a value of one. The peak of the Boltzmann distribution indicates the most probable kinetic energy, and the average kinetic energy is greater than the most probable. The theoretical and simulated velocity distributions are shown below in Fig. 8 and Fig. 9.

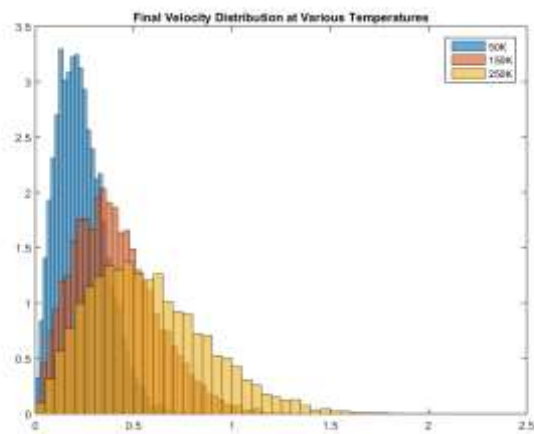


Fig. 8 Velocity distribution for several temperatures

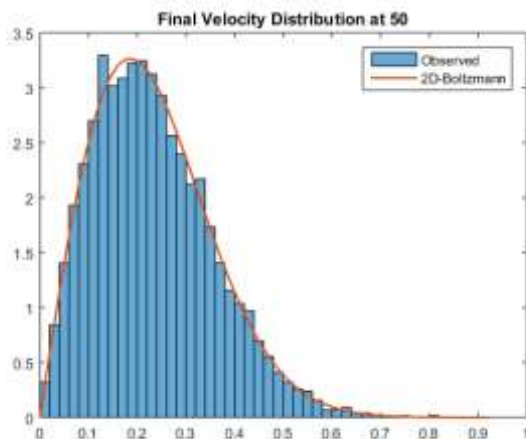


Fig. 9 Theoretical Boltzmann distribution overlaid with simulated velocity distribution at 50K.

Some interesting observations can be made from the data. First, there are no atoms at zero energy, meaning that all atoms have some associated velocity (energy). Another significant observation from Fig. 8 was that as the temperature in the simulation increased, the distributions shifted rightward, meaning that the most probable kinetic energy and average kinetic energy increased with increasing temperature. In addition, as the temperature increased, the distributions

became more flattened and the range of probable kinetic energies increased. Next, at all temperatures, there were very few atoms at high energy and low energy and the majority of all atoms in the simulation were at intermediate energies. These conclusions followed the theoretical distribution almost identically, as evidenced in Fig. 9.

## V. CONCLUSIONS

In this paper, a two-dimensional molecular dynamics (MD) code was developed in Matlab to study the temperature effects on stress of a nanostructured material. The simulation was performed on a fictitious material in a square Morse lattice of 100 x 100 atoms. The Morse potential was used as the two-body interatomic potentials and velocity scaling was employed to maintain the simulated system at any given constant temperature. When restricting expansion, it was found that the temperature caused the development of thermal stresses in the material, which increased as temperature increased. The stress-strain curves at various temperatures showed that material failure occurred at lower strain when temperature is higher. In addition, the velocity or energy distributions were plotted at various temperatures, compared well with the theoretical Maxwell-Boltzmann distributions. It was shown that as temperature increased, the probability of atoms at higher energy increased as well. Although the fictitious material with the Morse lattice is studied here, the provided Matlab code can be easily modified to study real nanostructured materials.

## REFERENCES

- [1] Iijima, S., "Helical microtubules of graphitic carbon", *Nature*, Vol. 354, pp. 56-58, 1991.
- [2] Popov, V. N., "Carbon nanotubes: properties and application", *Materials Science and Engineering R – Reports*, Vol. 43(3), pp. 61-102, 2004.
- [3] Xiao, S. P. and Hou, W. Y., "Fracture of vacancy-defected carbon nanotubes and their embedded nanocomposites", *Physical Review B*, Vol 73, pp. 115406, 2006.
- [4] Hou, W. Y. and Xiao, S. P., "Mechanical behaviors of carbon nanotubes with randomly located vacancy defects" *Journal of Nanoscience and Nanotechnology*, Vol. 7, pp. 4478-4485, 2007.
- [5] Xiao, S. P. and Hou, W. Y., "Studies of size effects on carbon nanotubes' mechanical properties by using different potential functions", *Fullerenes, Nanotubes, and Carbon Nanostructures*, Vol. 14, pp. 9-16, 2006.
- [6] Xiao, S. P., Han, R. and Hou, W. Y., "Spin in carbon nanotube-based oscillators", *International Journal of Nanoscience*, Vol. 5, pp. 47-55, 2006.
- [7] Xiao, S. P., Andersen, D. R., Han, R. and Hou, W. Y., "Studies of carbon nanotube-based oscillators using molecular dynamics", *International Journal of Computational and Theoretical Nanoscience*, Vol. 3, pp. 142-147, 2006.
- [8] Wei, X. P., Xiao, S. P., Ni, J., "Studies of ice melting via molecular dynamics", *Molecular Simulations*, Vol 36(11), 2010, 823-830
- [9] Wei, X. P., Xiao, S. P., Ni, J., "Ice melting speeds at various scales", *Advanced Science Letters*, Vol 17(1), pp. 172-178, 2012.
- [10] Abraham, F., Broughton, J., Bernstein, N. and Kaxiras, E., "Spanning the continuum to quantum length scales in a dynamic

simulation of brittle fracture", Europhysical Letter, Vol. 44, pp. 783-787, 1997.

[11] Belytschko, T. and Xiao, S. P. "Coupling methods for continuum model with molecular model". International Journal of Multiscale Computational Engineering, Vol. 1, pp. 115-126, 2003.

[12] Xiao, S. P. and Belytschko, T., "A bridging domain method for coupling continua with molecular dynamics", Computational Methods in Applied Mechanics and Engineering. Vol. 193, pp. 1645-1669, 2004.

[13] Xiao, S. P., Ni, J. and Wang, S. W., "The bridging domain multiscale method and its high performance computing implementation", Journal of Computational and Theoretical Nanoscience, Vol. 5, pp. 1220-1229, 2008.

[14] Liu, W. K., Park, H. S., Qian, D., Karpov, E. G., Kadowaki, H. and Wagne, G. J., "Bridging scale methods for nanomechanics and materials", Computational Methods in Applied Mechanics and Engineering, Vol. 195, pp. 1407-1421, 2003.

[15] Liu, W. K., Qian, D., Gonella, S., Li, S. F., Chen, W. and Chirputkar, C., "Multiscale methods for mechanical science of complex materials: Bridging from quantum to stochastic multisolution continuum", International Journal for Numerical Methods in Engineering. Vol. 83, pp. 1039-1080, 2010.

[16] Xiao, S. P. and Hou, W. Y., "Studies of nanotube-based aluminum composites using the bridging domain coupling method", International Journal of Mutliscale Computational Engineering, Vol. 5, pp. 447-459, 2007.

[17] Xiao, S. P. and Hou, W. Y. "Studies of nanotube-based resonant oscillators via multiscale modeling and simulation", Physical Review B, Vol. 75, pp. 125414, 2007.

[18] Xiao, S. P. and Hou, W. Y., "Multiscale modeling and simulation of nanotube-based torsional oscillators", Nanoscale Research Letters, Vol. 2, pp. 54-59, 2007.

[19] Xiao, S. P., Andersen, D. R. and Yang, W. X., "Design and analysis of nanotube-based memory cells", Nanoscale Research letters, Vol. 3, pp. 416-420, 2008.

[20] Tadmor, E. B., Ortiz, M. and Phillips, R., "Quasicontinuum analysis of defects in solids", Philosophy Magazine A, Vol. 73, pp. 1529-1563, 1997

[21] Chen, W. and Fish, J. "A mathematical homogenization perspective of virial stress", International Journal for Numerical methods in Engineering, Vol. 67, pp. 189-207, 2006.

[22] Fish, J., Chen, W. and Li, R. G., "Generalized mathematical homogenization of atomistic media at finite temperature in three dimensions. Computational Methods in Applied Mechanics and Engineering, Vol. 196, pp. 908-922, 2007.

[23] Tzou, D. Y., "Macro-to microscale heat transfer", Taylor & Francis Inc., Washington, DC, 1997

[24] Xiao, S. P. and Yang, W. X., "Temperature-related Cauchy-Born rule for multiscale modeling of crystalline solids", Computational Materials Science, Vol. 37, pp. 374-379, 2006.

[25] Xiao, S. P. and Yang, W. X., "A temperature-related homogenization technique and its implementation in the meshfree particle method for nanoscale simulations", International Journal for Numerical Methods in Engineering, Vol. 69, pp. 2099-2125, 2007

[26] Yang, W. X. and Xiao, S. P., "Extension of the TCB rule: material stability and thermal-mechanical coupling", Computational Materials Science, Vol. 41, pp. 431-439, 2008

#### APPENDIX: MATLAB CODE

%% 2-D Molecular Dynamics Simulation by Greg Tanner and Robert Hart

```
%% Parameters
% Mass of Carbon = 1.998467E-26 kg
% Boltzmann Constant: kb = 1.3806485E-23 J/K
% = 6.908537894295983E-04 (mass of carbon)(nm)^2/((ps)^2*K)
% Conversion Factor: J = 0.500383543986466E20 (mass of
carbon)(nm)^2/(ps)^2
kb = 6.908537894295983E-04; % Boltzmann constant
Temp_desired = 5; % Temperature in K
SF = 0.5; % safety factor in time-step

%% Lennard Jones Potential
% LJ parameters: r0 = 0.3834 nm
% eps = 3.8253E-22 J = 0.019141089 *(mass of
carbon)(nm)^2/(ps)^2
% Force Constant: k = d^2E/dr^2 = 72 eps/r0^2
% = 3.594570704225352 (mass of carbon)/(ps)^2
% Wave speed: c = sqrt(k*A/(N*m)) = sqrt(k*r0^2/(mass of carbon))
% = sqrt(72*eps/(mass of carbon)) = 1.173949917160013
nm/ps
% Time Step: dT = SF*r0/c = SF*0.326589741517688 ps
% We used a safety factor of SF = 0.5
eps = 0.019141089;
LJ_dT = 0.01*0.326589741517688;
LJ_r0 = 0.3834;
LJ_Potential = @(r)(eps*((LJ_r0./r).^12-2*(LJ_r0./r).^6)); % in units
of (mass of carbon)(nm)^2/(ps)^2
LJ_Force = @(r)(12*eps/LJ_r0*((LJ_r0./r).^13-(LJ_r0./r).^7)); % in
units of (mass of carbon)(nm)/(ps)^2

%% Morse Potential
% Morse parameters: beta = 26.25 nm^(-1)
% r0 = 1.42 nm
% D_e = 0.603105E-18 J = 30.178381729595756(mass of
carbon)(nm)^2/(ps)^2
% Force Constant: k = d^2E/dr^2 = 2*beta^2*D_e
% = 4.158958232109915e+04 (mass of carbon)/(ps)^2
% Wave speed: c = sqrt(k*A/(N*m)) = sqrt(k*r0^2/(mass of carbon))
% = 2.895880415215109e+02 nm/ps
% Time Step: dT = SF*r0/c = SF*0.004903517398506 ps
% We used a safety factor of SF = 0.5
D_e = 30.178381729595756;
beta = 26.25;
Morse_r0 = 1.42;
Morse_dT = 0.5*0.004903517398506;
Morse_Potential = @(r)(D_e*(exp(-2*beta*(r-Morse_r0))-2*exp(-
beta*(r-Morse_r0))));
Morse_Force = @(r)(2*beta*D_e*(exp(-2*beta*(r-Morse_r0))-exp(-
beta*(r-Morse_r0))));

%% Potential Used
Force = Morse_Force;
Potential = Morse_Potential;
dT = Morse_dT;
r0 = Morse_r0;

%% Set up Rectanglular Lattice
Nx = 100; % Number of molecules in x
Ny = 100; % Number of molecules in y
N = Nx*Ny; % Total number of molecules
A = N*r0^2; % Area of computational domain
strainX = 0; % Strain in the x-direction
widthX = (1.0+strainX)*Nx*r0; % Width in x-direction of
computational domain
```

```

widthY = Ny*r0; % % Width in y-direction of computational domain
rx_start = (1.0+strainX)*r0; % initial interatomic spacing in x-
direction

% Matrices with the initial x,y coordinates of each atom
[Y_init,X_init] = meshgrid(r0*(0:(Ny-1)),rx_start*(0:(Nx-1)));
delX = 0.00001*r0*randn(size(X_init));
delY = 0.00001*r0*randn(size(Y_init));
X = X_init + delX; % small random perturbations
Y = Y_init + delY; % small random perturbations

%% Randomly generate the initial velocity direction
v0 = sqrt(2*kb*Temp_desired); % Root-mean-square velocity in
nm^2/ps^2
theta = 2*pi*rand(Nx,Ny); % Uniform Random number in [0,2 pi]
Vx = v0*cos(theta);
Vy = v0*sin(theta);

figure(1);
quiver(X,Y,Vx,Vy,0.5);
title('Initial Velocities');

%% Initial Force Calculation
% Convention : Positive force is repulsive, negative is attractive.
Fx = zeros(Nx,Ny);
Fy = zeros(Nx,Ny);
plusx = [2:Nx,1]; % index of neighbor in +x-direction
minux = [Nx,1:Nx-1]; % index of neighbor in -x-direction
plusy = [2:Ny,1]; % index of neighbor in +y-direction
minusy = [Ny,1:Ny-1]; % index of neighbor in -y-direction

% Calculate force with neighbor in +x-direction
dX = [X(2:Nx,:);X(1,:)+widthX]-X;
dY = Y(plusx,:)-Y;
R = sqrt(dX.^2+dY.^2); % distance to neighbor in +x-direction
F = Force(R);
Fx = Fx - F.*dX./R + F(minusx,:).*dX(minusx,:)/R(minusx,:); %
subtract force from right and add force from left
Fy = Fy - F.*dY./R + F(minusx,:).*dY(minusx,:)/R(minusx,:); %
subtract force from right and add force from left

% Calculate force with neighbor in +y-direction
dX = X(:,plusy)-X;
dY = [Y(:,2:Ny),Y(:,1)+widthY]-Y;
R = sqrt(dX.^2+dY.^2); % distance to neighbor in +y-direction
F = Force(R);
Fx = Fx - F.*dX./R + F(:,minusy).*dX(:,minusy)/R(:,minusy); %
subtract force from above and add force from below
Fy = Fy - F.*dY./R + F(:,minusy).*dY(:,minusy)/R(:,minusy); %
subtract force from above and add force from below

% {
figure(2);
quiver(X,Y,Fx,Fy,0.8), hold on;
quiver(X,Y,delX,delY,0.5), hold off;
legend('Force','Displacement');
title('Initial States before First Time Step')
% }

%% Time Integration using the Velocity Verlet Method
tic;
N_steps = 20000;

p_out = zeros(N_steps,2); % position of center atom
E_out = zeros(N_steps,1); % potential
Stress_out = zeros(N_steps,4); % stress
for t = 1:N_steps
    % Update X and Y position at new time step using old velocity
    and
    % acceleration
    X = X + Vx*dT + 0.5*dT^2*Fx;
    Y = Y + Vy*dT + 0.5*dT^2*Fy;
    % Fix the first atom to prevent drift
    X = X - X(1,1);
    Y = Y - Y(1,1);
    p_out(t,1) = X(floor(Nx/2),floor(Ny/2))-
    X_init(floor(Nx/2),floor(Ny/2));
    p_out(t,2) = Y(floor(Nx/2),floor(Ny/2))-
    Y_init(floor(Nx/2),floor(Ny/2));

    % Calculate Forces at the updated positions
    Fx_new = zeros(Nx,Ny);
    Fy_new = zeros(Nx,Ny);

    % Calculate force with neighbor in +x-direction
    dX = [X(2:Nx,:);X(1,:)+widthX]-X;
    dY = Y(plusx,:)-Y;
    R = sqrt(dX.^2+dY.^2);
    F = Force(R);
    W = Potential(R);
    E_out(t) = E_out(t) + sum(sum(W));
    Fx_new = Fx_new - F.*dX./R +
    F(minusx,:).*dX(minusx,:)/R(minusx,:); % subtract force from right
    and add force from left
    Fy_new = Fy_new - F.*dY./R +
    F(minusx,:).*dY(minusx,:)/R(minusx,:); % subtract force from right
    and add force from left
    Stress_out(t,1) = Stress_out(t,1) + sum(sum(F.*dX.*dX./R)); %
    sigma_XX
    Stress_out(t,2) = Stress_out(t,2) + sum(sum(F.*dX.*dY./R)); %
    sigma_XY
    Stress_out(t,3) = Stress_out(t,3) + sum(sum(F.*dY.*dY./R)); %
    sigma_YY

    % Calculate force with neighbor in +y-direction
    dX = X(:,plusy)-X;
    dY = [Y(:,2:Ny),Y(:,1)+widthY]-Y;
    R = sqrt(dX.^2+dY.^2);
    F = Force(R);
    W = Potential(R);
    E_out(t) = E_out(t) + sum(sum(W));
    Fx_new = Fx_new - F.*dX./R +
    F(:,minusy).*dX(:,minusy)/R(:,minusy); % subtract force from
    above and add force from below
    Fy_new = Fy_new - F.*dY./R +
    F(:,minusy).*dY(:,minusy)/R(:,minusy); % subtract force from
    above and add force from below
    Stress_out(t,1) = Stress_out(t,1) + sum(sum(F.*dX.*dX./R)); %
    sigma_XX
    Stress_out(t,2) = Stress_out(t,2) + sum(sum(F.*dX.*dY./R)); %
    sigma_XY
    Stress_out(t,3) = Stress_out(t,3) + sum(sum(F.*dY.*dY./R)); %
    sigma_YY

    % Calculate test velocity (before scaling)
    Vx = Vx + 0.5*dT*(Fx+Fx_new);
    Vy = Vy + 0.5*dT*(Fy+Fy_new);

```

```

% Calculate current temp
KE_average = sum(sum(Vx.^2+Vy.^2))/(2*N);
Temp = KE_average/kb;

% Scale velocity
Vx = sqrt(Temp_desired/Temp)*Vx;
Vy = sqrt(Temp_desired/Temp)*Vy;

% Set Forces
Fx = Fx_new;
Fy = Fy_new;
end

%% Plot output
figure(3);
quiver(X,Y,Vx,Vy), hold on;
quiver(X,Y,Fx,Fy), hold on;
quiver(X,Y,Vx*dT + 0.5*dT^2*Fx, Vy*dT + 0.5*dT^2*Fy), hold off;
legend('Velocity','Force','Displacement');
title(['States for Final Time Step (' ,num2str(Temp_desired),')']);

figure(4);
quiver(X,Y,X-X_init,Y-Y_init);
title(['Total Displacement after Final Time Step (' ,num2str(Temp_desired),'K)']), hold on;
plot([0, widthX, widthX, 0, 0],[0, 0, widthY, widthY, 0]), hold off;

figure(5);
plot((1:N_steps)*dT,p_out/r0);
legend('X','Y');
ylabel('Displacement (Fraction of Equilibrium Bond Length)');
xlabel('Time (ps)');
title(['Displacement of Center Atom (' ,num2str(Temp_desired),'K)']);

figure(6);
Potential = E_out/N;
plot((1:N_steps)*dT,Potential,[0,N_steps*dT],[-2*eps,-2*eps]);
ylim([-2.1*eps, 0+0.1*eps]);
ylabel('Potential ((mass of carbon)(nm)^2/(ps)^2)');
xlabel('Time (ps)');
legend('Potential','Minimum Possible Potential');
title(['Potential versus Time (' ,num2str(Temp_desired),'K)']);

figure(7);
Stress = Stress_out/A;
plot((1:N_steps)*dT,Stress);
ylabel('Stress ((mass of carbon)/(ps)^2)');
xlabel('Time (ps)');
legend('\sigma_{XX}','\sigma_{XY}','\sigma_{YY}');
title(['Stress versus Time (' ,num2str(Temp_desired),'K)']

```