Supporting Information

Discovery based approach to real gas properties using molecular dynamics simulation

Chelsea Sweet┼, Oyewumi Akinfenwa┼, Jonathan J. Foley, IV\*

Department of Chemistry, William Paterson University, 300 Pompton Road, Wayne, NJ, 07470, USA

Contents of this Supporting Information Document

1. Protocol for users with remote-access to a Linux workstation
2. Protocol for Mac OSX users running MD program natively
3. Instructions for Suggested Programming Exercises

In all protocols, commands entered into the terminal command line are given in **bold font**. Commands should be followed by pressing the ENTER key.

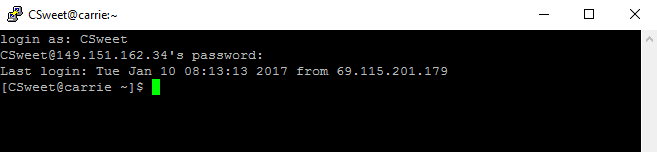
**I. Protocol for users with remote-access to a Linux workstation**

**Requirements**

These steps assume the MD program will run on a multi-user Linux workstation that the user will access remotely via ssh. Detailed instructions are provided for using the free ssh client PuTTY for Windows users. Additionally, we provide instructions for copying trajectory files created by the simulation from the MD program from the Linux workstation to the users Windows machine using the free software WinSCP, and for visualizing the trajectories using the free software VMD for windows. Mac OSX users with remote access to a Linux workstation can also follow these instructions with a few minor modifications noted. These steps also assume the GNU Compiler Collection (GCC), including the C/C++ compiler g++, and git installed on the users Linux workstation. If this is not the case, your Linux admin can install these for you.

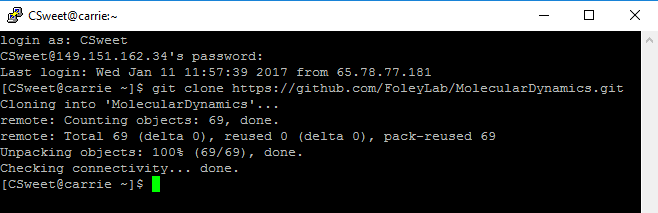
**Instructions**

1. Establish your remote connection using ssh via PuTTY on Windows or the Terminal program on Mac OSX.



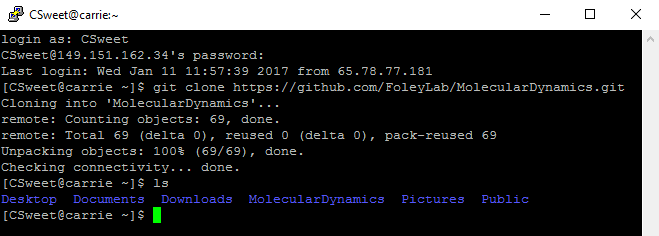
1. You will now download and install the Molecular Dynamics program from the repository hosting service GitHub. To clone the molecular dynamics folder, type the following and press ENTER:

**git clone** [**https://github.com/FoleyLab/MolecularDynamics.git**](https://github.com/FoleyLab/MolecularDynamics.git)



1. To view the available folders, use the “list command”

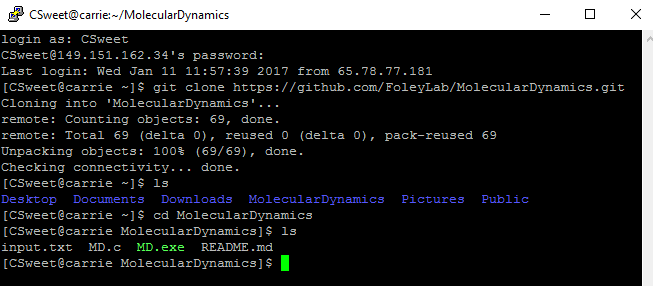
**ls**



1. Move into thefolder containing the molecular dynamics files by tying

**cd MolecularDynamics**

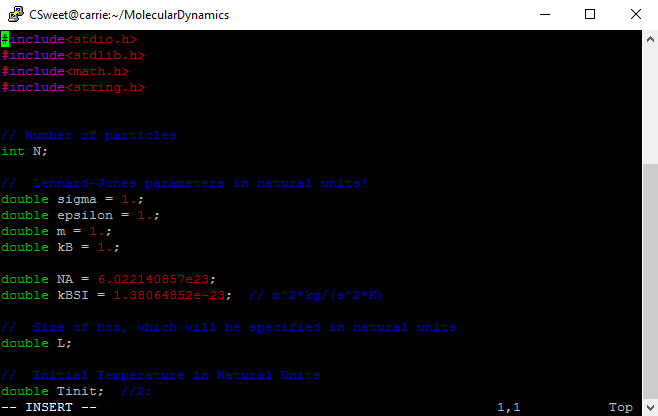
Note: folder and file names are case-sensitive. Use the **ls** command to view the contents



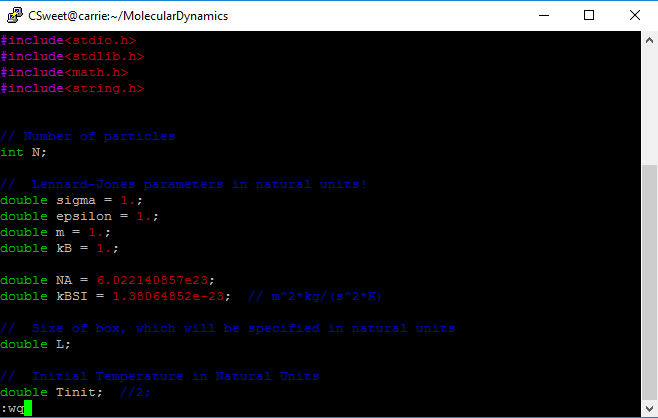
1. To open up the source code file for editing, use the Vi improved text editor:

**vim MD.c**

To engage in text editing mode, press **I** (the word **- - INSERT - -** will appear on the bottom); navigate the cursor (green rectangle) using the arrow keys; use BACKSPACE to delete characters to the left of the cursor; use DELETE to remove the highlighted character. For suggestions on parts of the code to edit for completion of programming exercises, see Section IV.

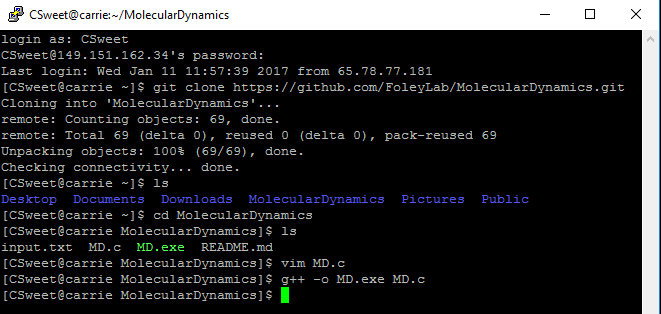


1. Press ESC to exit text editing mode, then type **:wq** ENTER to save changes and exit the file; this will return you to the folder view

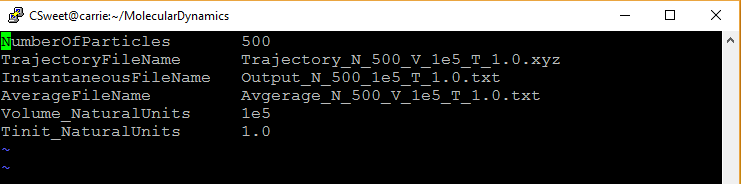


1. To compile the program using the g++ compiler and create an executable called ‘MD.exe’, type

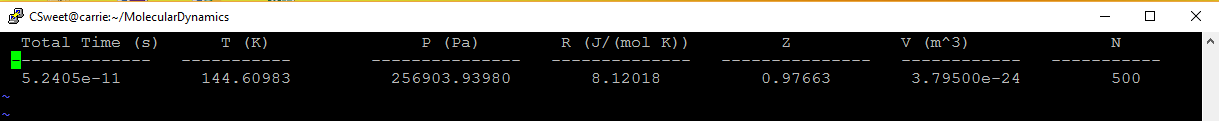
**g++ -o MD.exe MD.c**



1. To view/edit the input parameters of the simulation (number of particles, simulation volume, and initial temperature) type **vim input.txt** and press ENTER. **Note:** Volume and temperature are entered in natural units, for more details on converting natural units to SI units, see **Section IV**). Be sure to adjust the names of the output files to reflect the current input data. Three output files are generated:
   1. TrajectoryFileName – file to contain x, y, and z coordinates of all particles at all timesteps in a format that can be rendered by the software VMD
   2. AverageFileName – file to contain the simulated thermodynamic quantities including temperature, pressure, compressibility, and the gas constant in SI units
   3. InstantaneousFileName – file to contain instantaneous values of temperature, pressure, kinetic energy, and potential energy

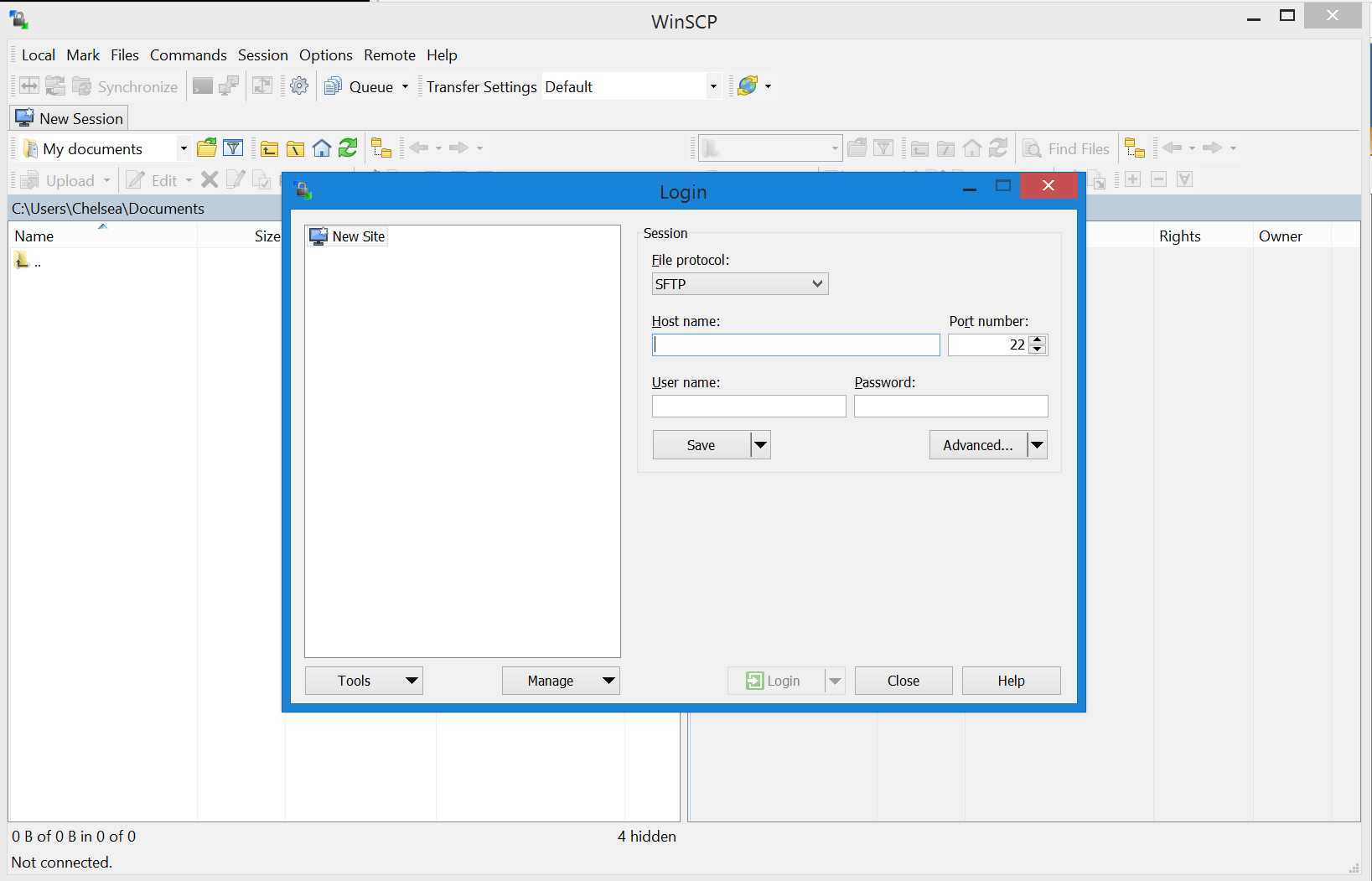


1. To edit input parameters, enter insert mode by pressing **I** and change the number of particles, initial temperature, and volume if desired.; press ESC and type **:wq** to save the changes and exit the file (note: we recommend keeping the number of particles to ~1000 or smaller to reduce waiting time.)
2. To run the program with the new data input, type **./MD.exe &** to run the program in the background, which allows commands to be entered while the program runs
3. To view the simulated thermodynamic quantities (see AverageFileName field in screenshot above), type **vim Average\_N\_500\_v\_1e5\_T\_1.0.txt**

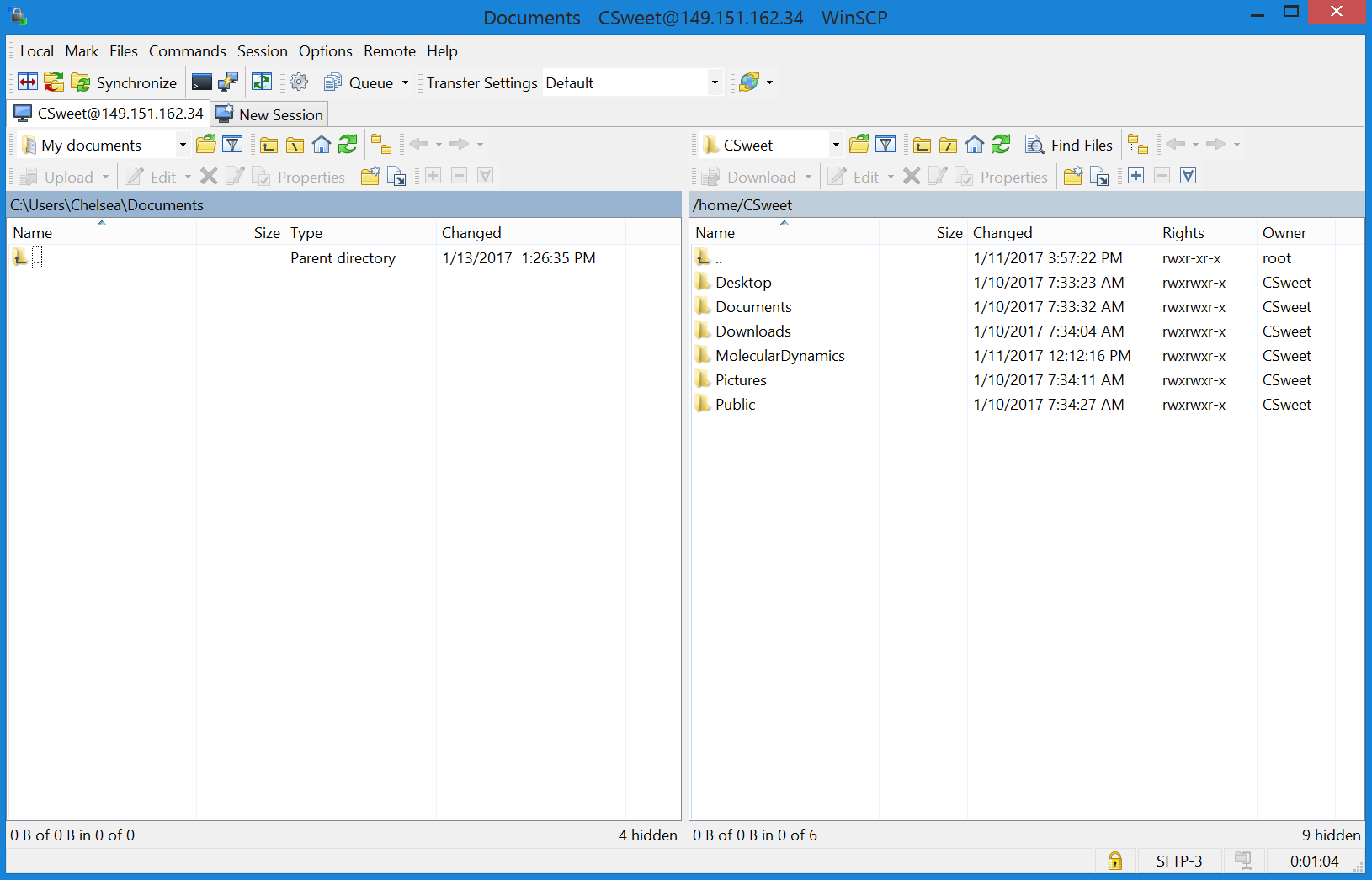


To animate your MD trajectory, first copy your file with the trajectory data (see TrajectoryFileName field in screenshot above) to your Windows machine using WinSCP. Instructions for installing WinSCP can be found in Section VI. Mac OSX users can transfer these files from the Linux workstation to their Mac directly from the terminal using sftp, see Section VII.

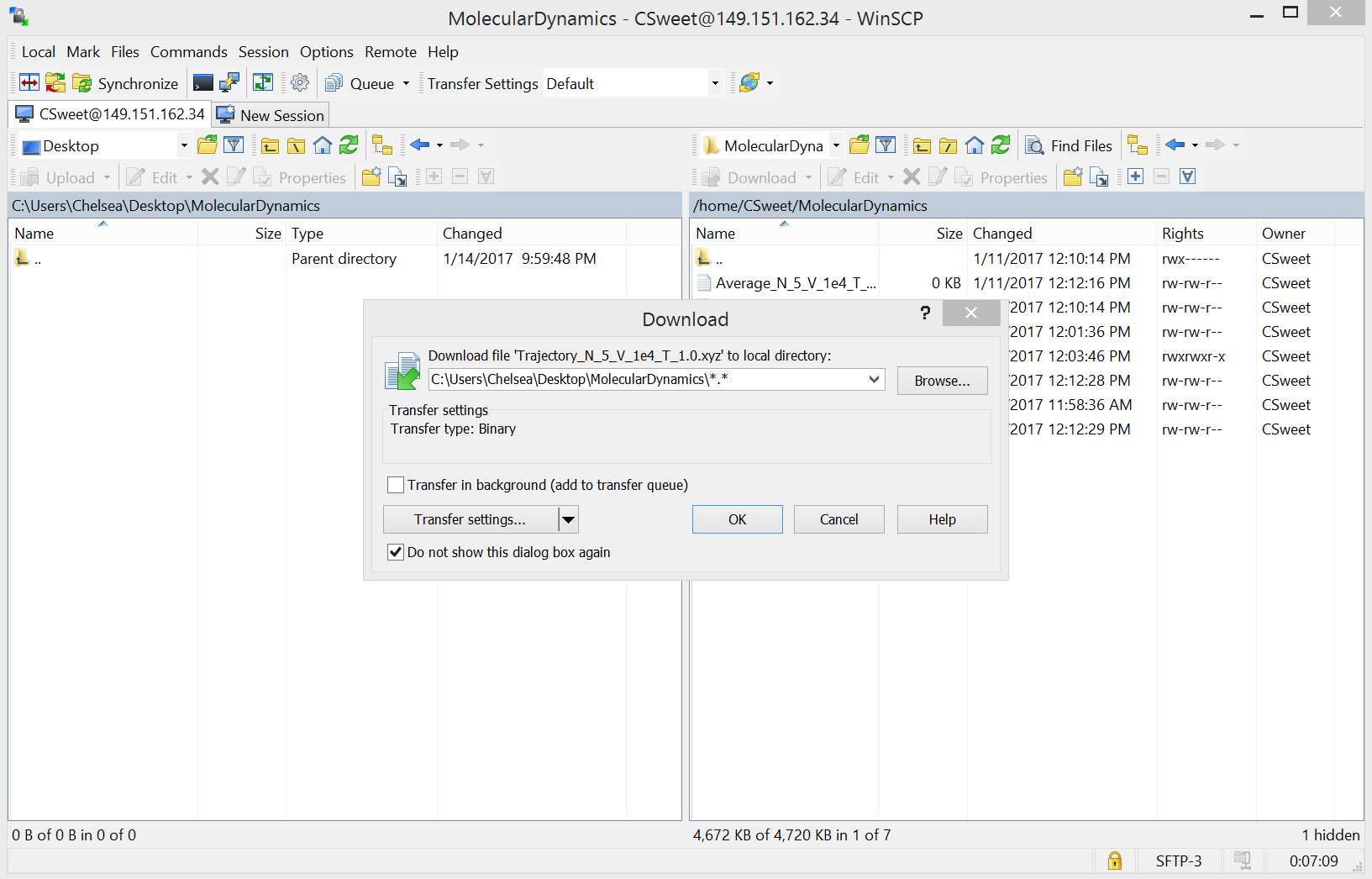
1. **Launch WinSCP** to open the prompt for the login screen; enter the same IP address (host name), username, and password you used to achieve remote access with PuTTY



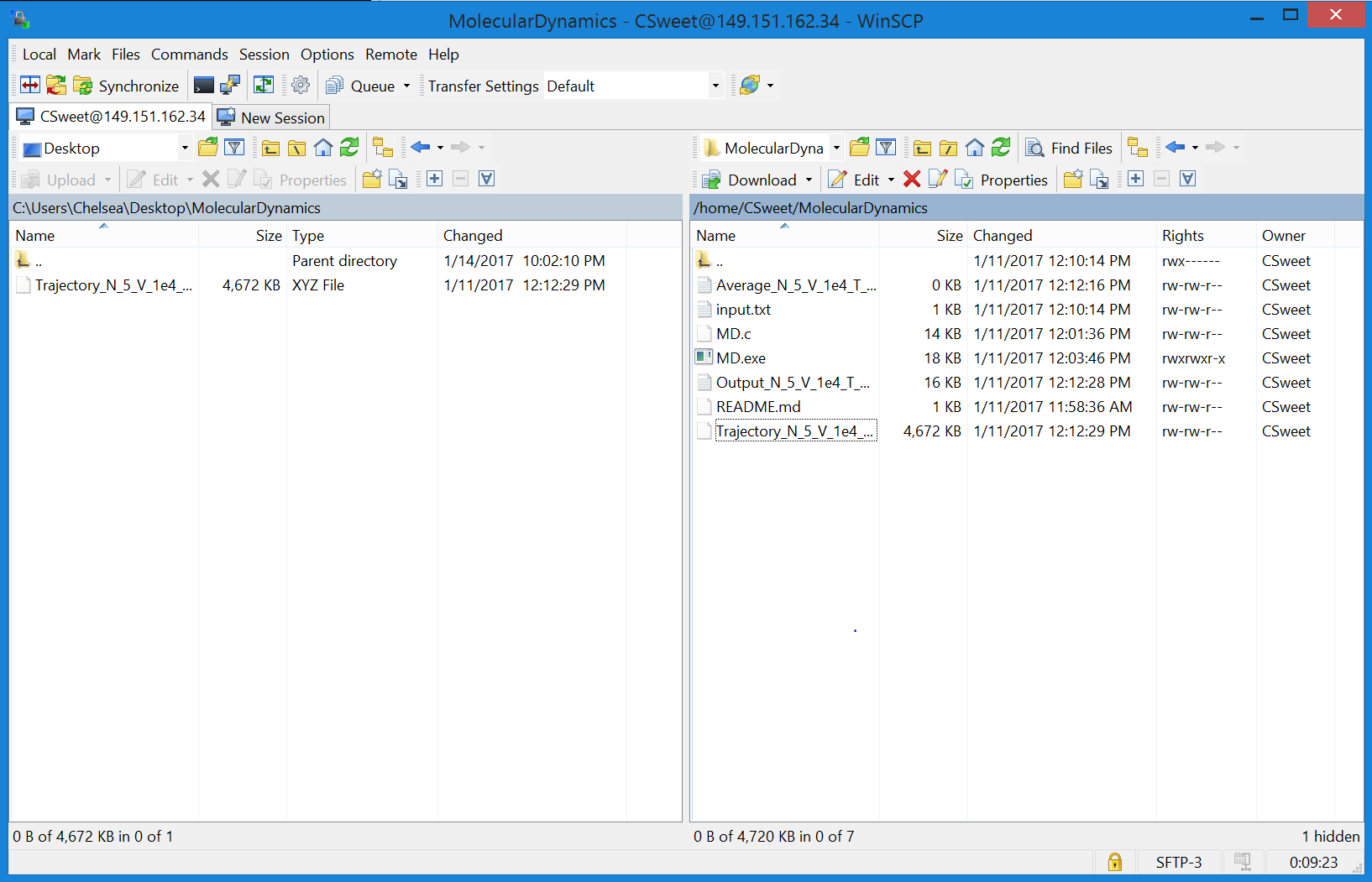
1. Click the **Login** button and wait for the server to connect (note: you may be prompted to allow permission to connect); the folder on the left shows your computer and the folder on the right shows the computer being remotely accessed



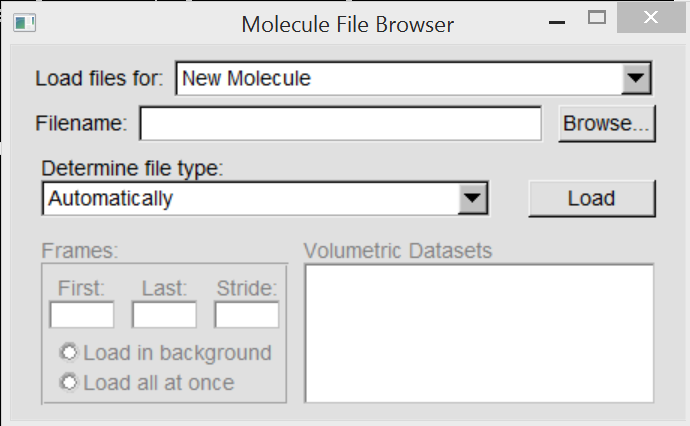
1. Double-click the **MolecularDynamics** folder on the right to open folder; transfer the **Trajectory\_** file from the folder on the right to a folder of your choice on the left (open **Parent directory** and browse for the location where you would like to store and access the trajectory files; this is the location where you will access the files from within VMD; to make a new folder, right-click, select **New Directory**, and name the folder **MolecularDynamics**); make sure the local directory you are transferring the file to has the correct folder path.



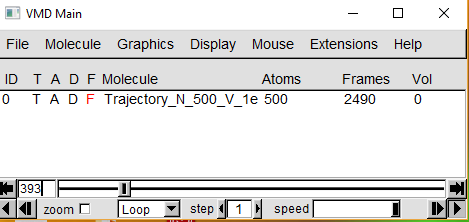
1. The trajectory file should now be visible in the folder on the left; it is ready to be accessed and opened by VMD



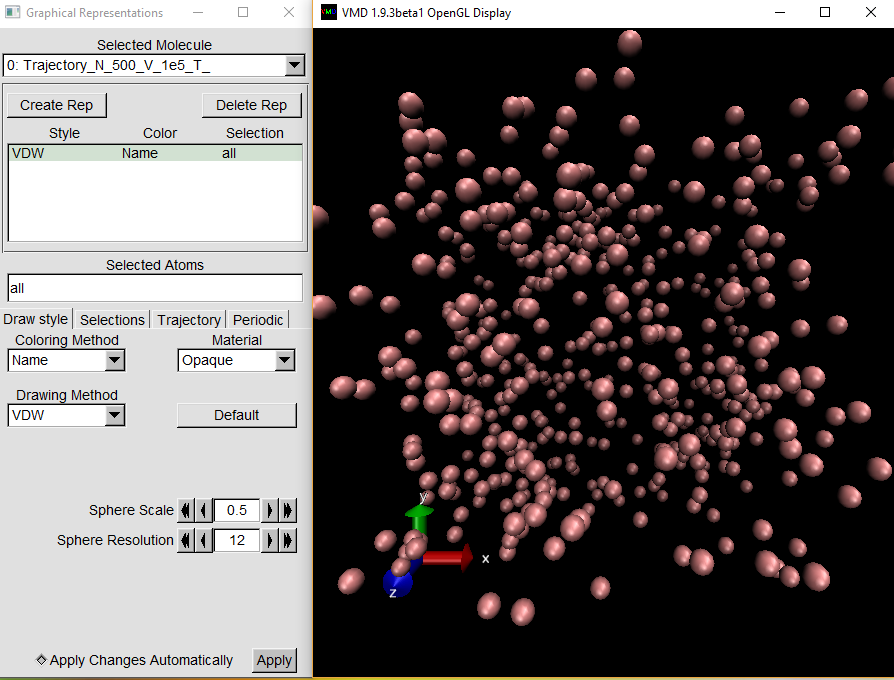
1. Launch the program VMD on your windows computer. In the **VMD Main** box, select **File > New Molecule** to open the **Molecule File Browser**; click **Browse** and open the folder containing the trajectory file; select the appropriate file and click **Load** (note: file type should have a **.xyz** extension)



1. The complete **Trajectory\_** file name should appear in the **VMD Main** window; highlight the filename by clicking on it, then click the menu for **Graphics > Representation**



1. In the **Graphical Representations** window, open the **Drawing Method** drop-down menu and select **VDW**; click **Apply**



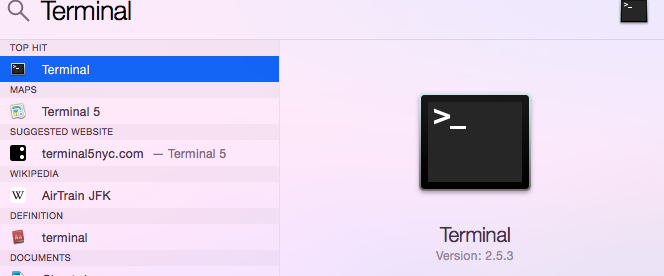
1. The simulation should now be running on the **VMD 1.9.3 OpenGL Display** window (note: zoom in and out using the mouse wheel; pause, fast forward, and rewind from the **VMD Main** window using the sliding bar and the arrow buttons in the bottom left and right corners of the window)

**II. Protocol for Mac OSX users running MD program natively**

**Requirements**

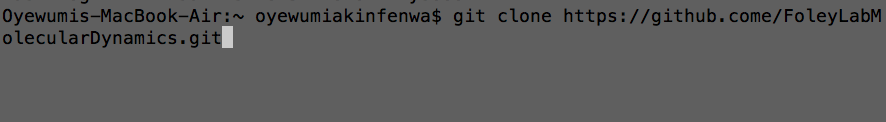
These steps assume the MD program will be run locally on a computer running Mac OSX; we have tested these steps on computers running Yosemeti and El Capitain. These steps also assume the GNU Compiler Collection (GCC), including the C/C++ compiler g++, and git installed on the Mac computer; these may be freely installed as part of the Apple XCode Developer package. **Note:** Windows users running the Linux emulator **Cygwin** may also follow the procedure below once the Cygwin terminal has been launched if the **vim**, **git**, and **gcc-g++-C++ Compiler** packages have been installed with Cygwin.

1. Launch a terminal session (Open Spotlight Search, search for and open “Terminal”)



1. When Terminal is open, type the following and press ENTER

**git clone** [**https://github.com/FoleyLab/MolecularDynamics.git**](https://github.com/FoleyLab/MolecularDynamics.git)

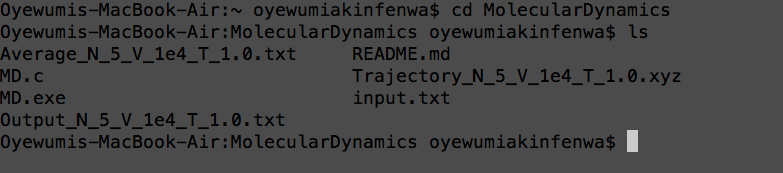


1. Move into the Molecular Dynamic folder by typing

**cd MolecularDynamics**

To view the files and folders within the “Molecular Dynamics” folder, type

**ls**



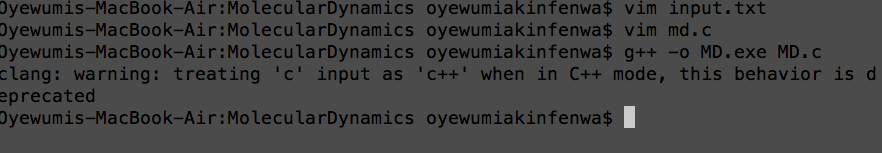
1. To open up the source code file for editing, use the Vi improved text editor:

**vim MD.c**

To engage in text editing mode, press **I** (the word **- - INSERT - -** will appear on the bottom); navigate the cursor (green rectangle) using the arrow keys; use BACKSPACE to delete characters to the left of the cursor; use DELETE to remove the highlighted character. For suggestions on parts of the code to edit for completion of programming exercises, see Section IV.

1. Press ESC to exit text editing mode, then type **:wq** ENTER to save changes and exit the file; this will return you to the folder view
2. To compile the program, type

**g++ -o MD.exe MD.c**



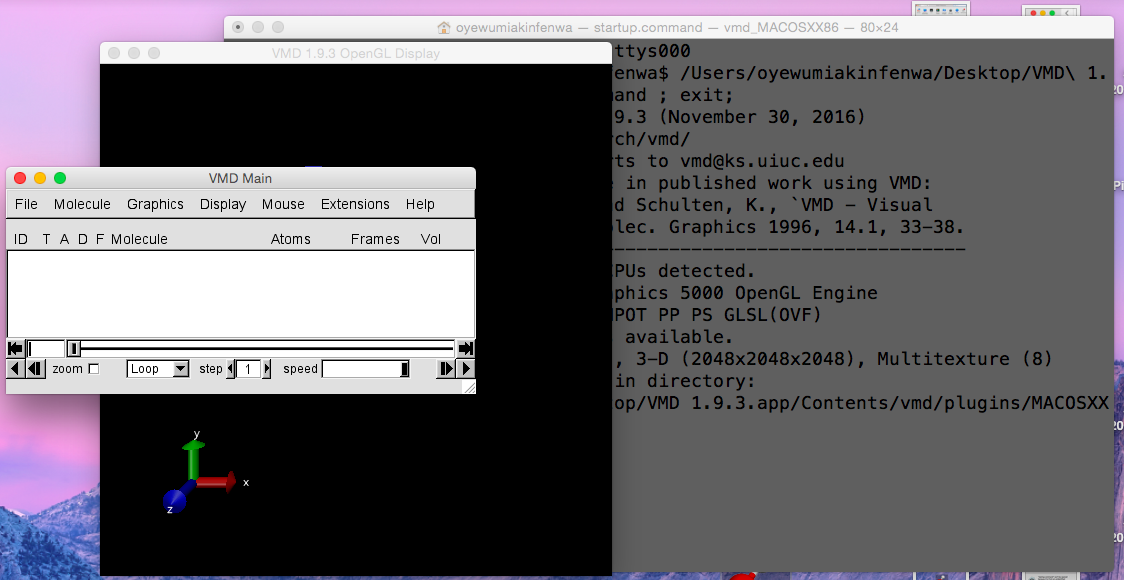
1. To view/edit the input parameters of the simulation (number of particles, simulation volume, and initial temperature) type **vim input.txt** and press ENTER. **Note:** Volume and temperature are entered in natural units, for more details on converting natural units to SI units, see **Section IV**). Be sure to adjust the names of the output files to reflect the current input data. Three output files are generated:
   1. TrajectoryFileName – file to contain x, y, and z coordinates of all particles at all timesteps in a format that can be rendered by the software VMD
   2. AverageFileName – file to contain the simulated thermodynamic quantities including temperature, pressure, compressibility, and the gas constant in SI units
   3. InstantaneousFileName – file to contain instantaneous values of temperature, pressure, kinetic energy, and potential energy



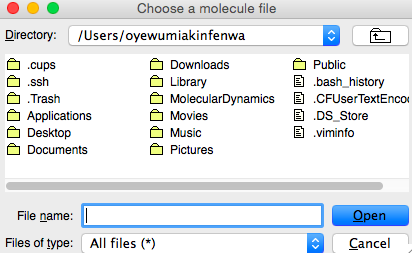
1. To edit input parameters, enter insert mode by pressing **I** and change the number of particles, initial temperature, and volume if desired.; press ESC and type **:wq** to save the changes and exit the file (note: we recommend keeping the number of particles to ~1000 or smaller to reduce waiting time.)
2. To run the program with the new data input, type **./MD.exe &** to run the program in the background, which allows commands to be entered while the program runs
3. To view the simulated thermodynamic quantities (see AverageFileName field in screenshot above), type **vim Average\_N\_1000\_v\_1e5\_T\_1.0.txt**



12. To animate the MD trajectory, launch VMD. In the main screen select File, New Molecule.

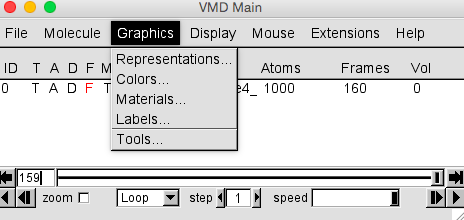


13. Select Browse. You will need to open the “MolecularDynamics” folder to open the trajectory file. Open the Trajectory file that you saved.





15. The simulation will begin to run. You will need to open Graphics, Representations to make adjustments of the view of the simulation.



16. Adjust the drawing method to VDW. You might also need to make the sphere scale smaller to view the simulation clearly.



**EXAMPLE OUTPUT**

****

Snapshots of animated trajectories corresponding to ideal (a), non-ideal attractive (b), and non-ideal repulsive (c) conditions. The particles have relatively large separation under ideal conditions, so the inter-particle forces make a negligible contribution to the gas properties on average. The average separation is much smaller in the non-ideal cases ( (b) and (c)), so the intermolecular forces make a much more substantial contribution to the observed behavior. The difference between attractive and repulsive behavior is more easily discerned in the dynamics of the particles than in a single snapshot. For example, clusters of particles (dimers, trimers, etc) can form and persist under non-ideal conditions where attractive forces are prominent (b), while close encounters between particles tend to be short lived under non-ideal conditions where repulsive forces are prominent (c).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Initial Temp (K)** | **Average T (K)** | **Average P (Pa)** | **V (m3)** | **Z** |
| **Ideal** | 141.9 | 141.9 | 5568 | 3.795e-22 | 1.077 |
| **Attractive** | 35.48 | 42.20 | 125,400 | 3.795e-25 | 0.8169 |
| **Repulsive** | 851.9 | 880.8 | 77,200,000 | 1.897e-25 | 1.204 |

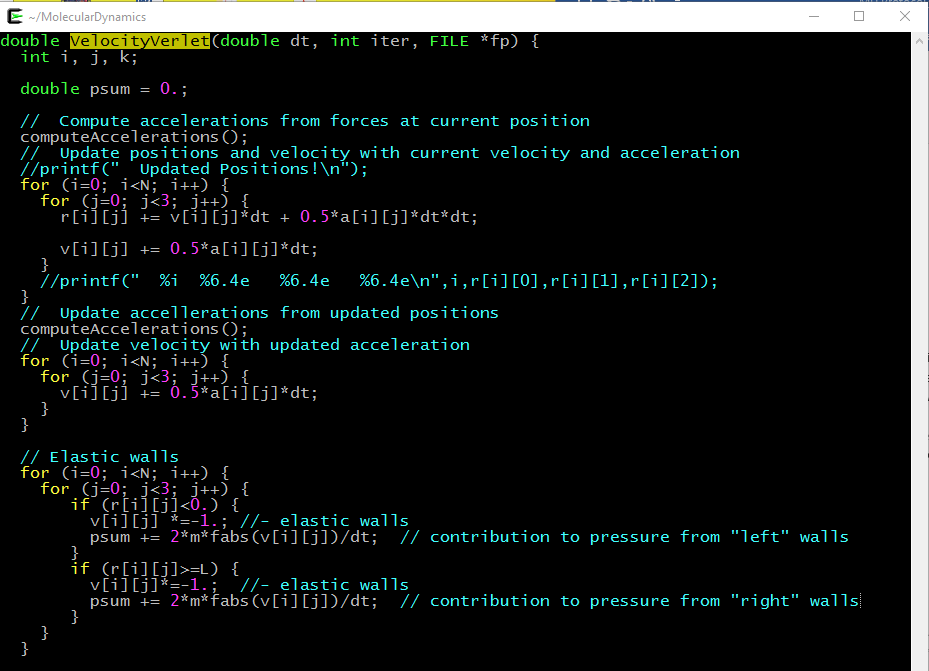
III. Instructions for Suggested Programming Exercises

The following steps provide further information to aid the programming exercises suggested in the manuscript. In particular, we highlight the parts of the code that perform the desired functions. Instructors who wish to provide these programming challenges may delete parts or all of the code shown below before introducing the program to their students, and work with the students to build the desired functionality back in by planning and writing similar code themselves.

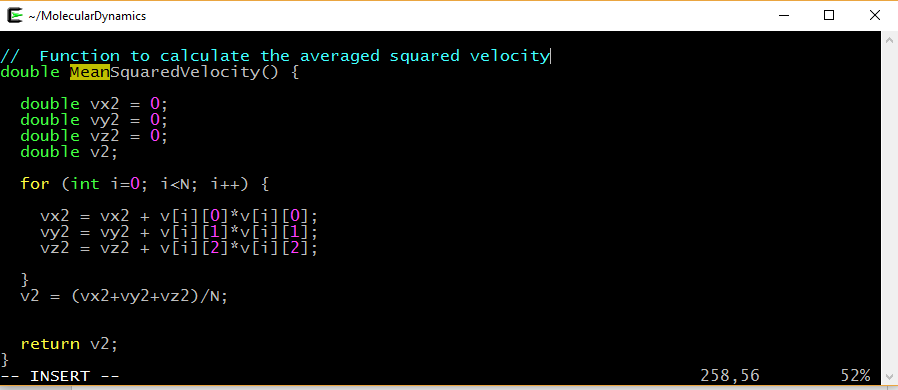
1. Programming Kinetic Theory Equations

The first exercise involves computing thermodynamic quantities (temperature and pressure) from the molecular dynamics trajectories by using kinetic theory expressions. Note that the trajectory information (position, velocity, and acceleration), as well as the forces, are stored as 2D arrays r[][], v[][], a[][], F[][], respectively. These arrays have global scope so their data can be accessed and manipulated to all functions in the program. In all of these arrays, the inner-index references the particle number and the outer-index references the Cartesian component. For example, the x-position of particle 1 is stored in r[0][0] and the z-component of velocity of particle 99 is stored in v[98][2] (array indices start from 0 in c).

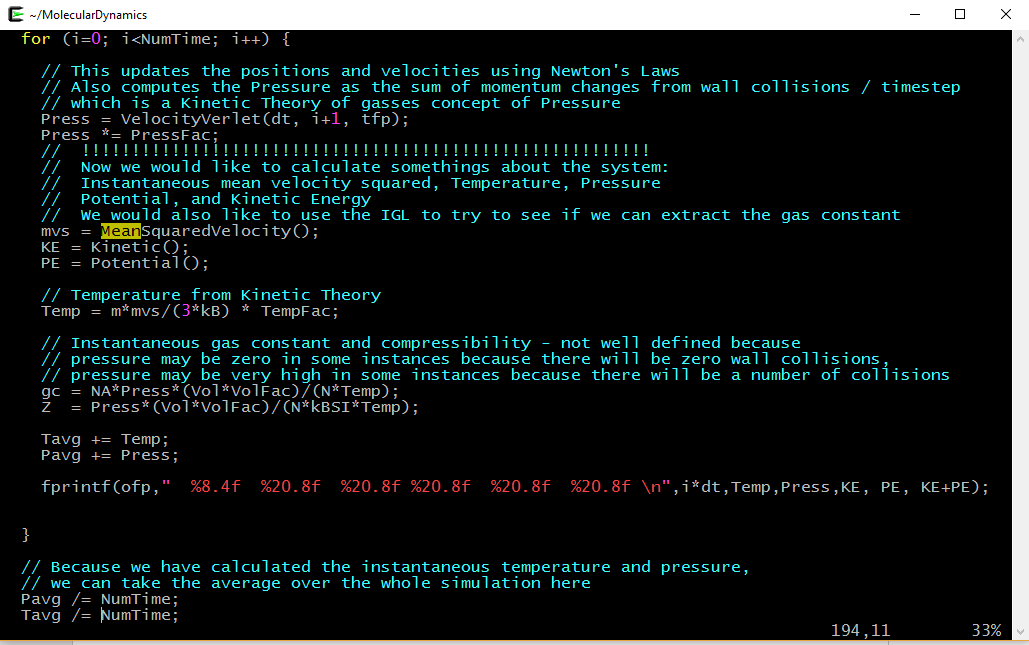
As discussed in the main text, instantaneous contributions to the pressure and temperature can be computed from trajectories using Kinetic Theory. In particular, the instantaneous pressure relies on summing up the momentum changes associated with wall collisions. The part of the code that updates the positions and velocities of the particles at each time-step must account for wall collisions anyway, so this is a natural place to compute these contributions. These updates occur in the function called ‘VelocityVerlet’, and the pressure contributions are accumulated in the variable ‘psum’ in the following code block:



The instantaneous contribution to the temperature depends upon the instantaneous mean squared velocity, which is computed in the following block of code in the function called ‘MeanSquaredVelocity’:



To estimate the thermodynamic temperature and pressure of the simulated system, we take the average of the instantaneous quantities over each time step. Hence, a sum of instantaneous temperatures and pressures is accumulated over each time-step in the simulation, and then divided by the total number of time steps at the end:



1. Simulating different gas types

The second programming exercise enables the simulation of different gas types. As described in the manuscript, our MD program uses a natural system of units where the particle mass , the Lennard-Jones length parameters , the Lennard-Jones energy parameters , and the Boltzmann constant are defined to have the value of 1. Therefore, each unique particle type has a unique set of factors to convert quantities in natural units to quantities in SI units, for example. Table 1 gives the mass and Lennard-Jones parameters for several noble gasses in SI units, and Table 2 gives natural-to-SI conversion factors for Volume, Temperature, and Pressure, and time.

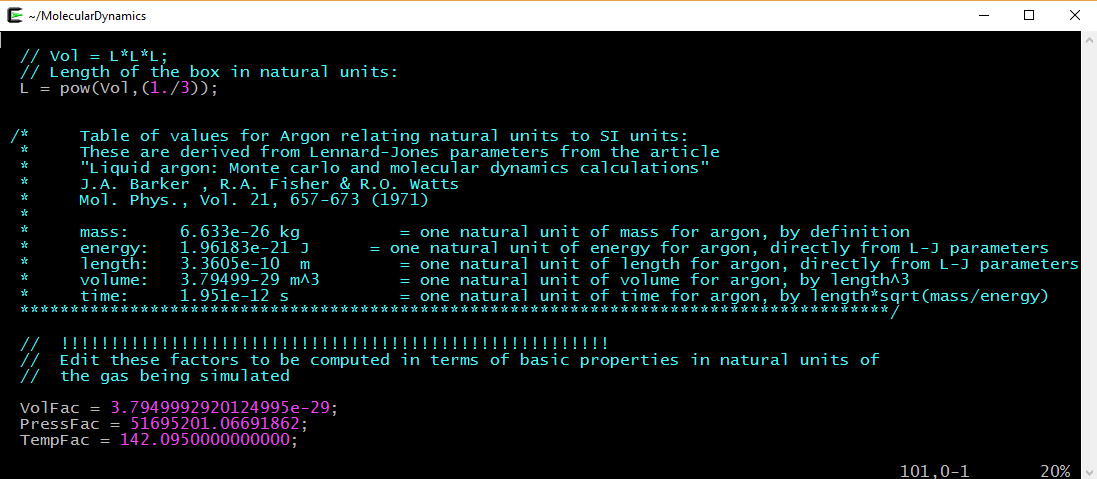
Table S1 Lennard-Jones parameters for several noble gasses along with particle mass. Lennard-Jones parameters for He, Ne, Kr, Xe are taken from Ref. 1, and parameters for Ar are taken from Ref. 2.

|  |  |  |  |
| --- | --- | --- | --- |
| **Particle** | **(kg)** | **(m)** | **(J)** |
| Helium | 6.646e-27 | 2.64e-10 | 1.5e-22 |
| Neon | 3.350e-26 | 2.74e-10 | 5.6e-22 |
| Argon | 6.633e-26 | 3.3605e-10 | 1.96183e-21 |
| Krypton | 1.391e-26 | 3.58e-10 | 2.75e-21 |
| Xenon | 2.180e-26 | 3.80e-10 | 3.87e-21 |

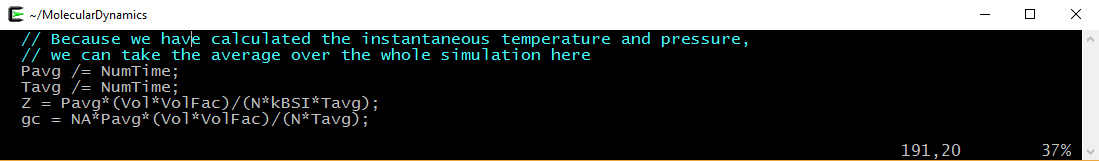
Table S2 Factors for converting several quantities from natural units to SI units. For example, one natural unit of volume for Helium is equal to 1.84e-29 .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Particle** | **Volume ()** | **Temperature (** | **Pressure (** | **Time** |
| Helium | 1.84e-29 | 10. | 8.1e+6 | 1.7e-12 |
| Neon | 2.06e-29 | 40. | 2.7e+7 | 2.1e-12 |
| Argon | 4.6850e-29 | 142.095 | 4.1874e+7 | 2.09618e-12 |
| Krypton | 4.58e-29 | 199. | 5.99e+7 | 8.05e-13 |
| Xenon | 5.48e-29 | 280. | 7.05e+7 | 9.01e-13 |

Students can simulate different gasses simply by modifying the conversion factors. In particular, for exploring the conditions for ideal and non-ideal behavior for different particle types, students can introduce these different conversion factors into the code and can subsequently compute the compressibility using the volume, the average temperature and pressure, and the gas constant. We show example code where these conversion factors are defined for a simulation of argon:



And subsequently used in computing the compressibility and the universal gas constant through the ideal gas law once the time-averaged temperature and pressure have been computed:



REFERENCES

1. Schroeder, D. V. Interactive molecular dynamics. *Am. J. Phys.* **2015,** *83*, 210-218.
2. Barker, J. A.; Fisher, R. A.; Watts, R. O. Liquid argon: Monte carlo and molecular dynamics calculations. *Mol. Phys.* **1971**, *21*, 657-673