Assignment 2
Due November 15, 2024, on learn dropbox
Show your work for each question

This a simulation assignment. The required Python scripts and pdb data files are available on the UW Learn page, in the same content section this Assignment PDF file was found.

Software requirements:
The openMM package (can be installed using anaconda or miniconda)
The openMM-Tools package (can be installed using anaconda or miniconda)
Plotting: you can use Matplotlib or MS Excel

Question 1: Use the python script `water2.py` to study the dynamics of two rigid water molecules under NVE (microcanonical) conditions. The parameters are at the beginning of the script. You will need to set the ensemble flag to NVE as shown below:

```
#####Parameters
steps = 400
skipSteps = 1
Temperature=75. # temperature in Kelvin
dt = 1. * unit.femtoseconds
ensemble='NVE' # NVE or NVT
#####
```

The `water2.py` script requires the `water2.pdb` file which contains the coordinates of the two water molecules.

1. Report all the parameters of your simulations with units.
2. Plot the kinetic, potential, and total energies (stored in files `KE.NVE`, `PE.NVE`, and `TE.NVE`). Note your observations.
3. Compute the average and variance of the kinetic, potential, and total energies. Vary the time step (dt) and the number of time steps and note your observations. Which quantity do you expect will be conserved?
4. Plot the distance between the two oxygen atoms of each water as a function of time (file `rOO.NVE`). Do you observe oscillations? Report the mean OO distance along with its variance.

Question 2: Use the python script `water2.py` to study the dynamics of two water molecules under NVT (canonical) conditions (change `ensemble` flag to `NVT`). Data files will now have the suffix NVT, ie: rOO.NVT, KE.NVT, PE.NVT, and TE.NVT.

1. Report all the parameters of your simulations
2. Repeat the steps of the NVE analysis above, but this time with in the NVT ensemble.

Question 3: Perform an NVT MD simulation of liquid argon using `arBox.py`. Report all parameters of the simulation (ie, the total number of atoms, the temperature, the density, the time step, the number of time steps, etc.). Use the suggested T and density values. Increase the number of time steps to improve the quality of your results.

1. Use the `Vanalysis.py` script to calculate the average potential energy. The script takes the output trajectory file name as a command line argument. The filename is `ar_liquid_traj200.h5` for N=200. Vary the number of particles in the simulation box (e.g. N= 200, 300, 400, 500, 600) and note the change in average potential energy per particle V/N. What happens as the system size increases? (Note that the script outputs the total potential energy in the file V). Estimate the standard error bars of the average potential energy per particle (hint, you need the variance of your data along with the total number of steps).
2. Use `genPairDistancesArgon.py` to calculate the $g(r)$ for the simulations above. The script takes the output trajectory file name as a command line argument. What happens to the $g(r)$ as the system size increases?
3. Calculate the number of neighbours in the first shell using the $g(r)$. Show all your work.

Question 4: Perform an NVT MD simulation of liquid water using the `waterBox.py`. Report all the parameters of the simulation (ie, the total number of atoms, the temperature, the density, the time step, the number of time steps, etc.). Use the suggested T and density values. Increase the number of time steps to improve the quality of your results. Also increase the size of the simulations box.

1. Use the `genPairDistancesWater.py` script to calculate the $g_{OO}(r)$ and $g_{OH}(r)$ for the simulations above. The script takes the output trajectory file name as a command line argument. Describe your results. What happens to the $g(r)$ as the system size increases?
2. Calculate the number of neighbours in the first shell using the $g_{OO}(r)$. Show all your work.