Chem9664B winter 2023

Assignment 3

- **All questions are of equal value.**
- **Important:** This assignment should be returned using GitHub. If you don't have an account, create a GitHub account, and in GitHub create a directory called Chem9664B. Make that directory private and share it with me. My username at GitHub is mejk (you need that for sharing). This assignment also assumes that you have now access to the command line terminal (Linux, WSL in Windows or Mac terminal), and have python and Jupyter Lab installed. Installation help is in the online course notes.
- **All of the tasks below should be in one single Jupyter notebook (ipynb file).**

Due: Friday Mar 3, 2023 at midnight.

---

1. Simulation of simple harmonic oscillator in one dimension. Set the equilibrium distance, mass and the spring constant equal to unity. Write an MD simulation code for the harmonic oscillator using the forward Euler integration, Verlet integration and velocity-Verlet integration. In the following, plot the results using the three different integrator in the same plot for easier comparison and interpret your results, and compare with the exact analytical results whenever possible. a) plot the amplitude as a function of time. b) plot pairs (x_i,p_i) in an x,p-coordinate system where the index i represents the time.

2. Work in dimensionless units (mass and the Boltzmann constant should be set to one) and in two dimensions, and use the Lennard-Jones potential. Write an MD simulation code using the same three integrators as above and below, plot the results from the three integrators in the same plot. You must interpret the results in all cases as well as provide explanation/derivation/formula for the quantities that are being measured. Use three different time steps for each integrator: dt=0.01, dt=0.001 and dt=0.02. a) Plot the total energy as a function of time. Whenever possible, give also a statistical measure for the fluctuations are. b) plot the temperature as a function time. c) Try to achieve the temperature of 1.2. As for the initial configuration, use the 2d lattice you created in the last exercise and remove two arbitrary particles from it. Apply periodic boundary conditions. Initialize the velocities using Gaussian distributed random numbers around zero. Remember to provide visualizations of the system, including a short movie (that should be uploaded to github separately – and of course the python files).