

# PHYS2020 Week 8 Project Milestone

Frey Boulton

April 30, 2023

*Note that this milestone builds on the previous one. A lot of information has not been included here because it was in the previous one.*

The goal of this project is to simulate particles on a microscopic scale and match their microscopic properties to more macroscopic properties like temperature and pressure, making use of phase diagrams.

The simulation will have a large number of particles, each interacting via the Lennard-Jones potential.

## 1 Lennard-Jones Potential

The Lennard-Jones (LJ) potential describes potentials between a pair of particles, including repulsive and attractive forces. As a function of distance  $r$  between particle centres, it is

$$V_{LJ}(r) = 4\epsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right]$$

$\epsilon$  controls the strength of the potentials, and  $\sigma$  is the distance between particles for which  $V_{LJ}(\sigma) = 0$ , that is, there is no potential energy. Treating particles as spheres,  $\sigma$  is like the diameter of the sphere.

The LJ potential is split into two components, each describing a different interaction. The  $\left(\frac{\sigma}{r}\right)^{12}$  term is for Pauli repulsion (at very close distances, electron orbitals start to overlap, but this is not possible due to Pauli exclusion, so they repel each other). The  $\left(\frac{\sigma}{r}\right)^6$  term is for attractive dispersion (a subtype of van der Waals) forces. The exponents thus account for the range of the interactions.

Force due to L-J in a given direction is the negative partial derivative in that direction.

## 2 Hamiltonian

Putting the L-J potential into the Hamiltonian gives

$$H = V_{LJ} + T = 4\epsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right] + T$$

Using Hamiltonian dynamics, we put the above into the following pair of differential equations:

$$\frac{d\mathbf{p}}{dt} = \frac{\partial H}{\partial \mathbf{q}} \quad \text{and} \quad \frac{d\mathbf{q}}{dt} = -\frac{\partial H}{\partial \mathbf{p}}$$

### 3 Notes for computational implementation

With the L-J potential, it's faster to set  $A = 4\varepsilon\sigma^{12}$  and  $B = 4\varepsilon\sigma^6$  and use

$$V_{LJ} = \frac{A}{r^{12}} - \frac{B}{r^6}$$

It's worth noting that the exponents of the Lennard-Jones potential were chosen to balance physical accuracy and computational efficiency.

The 6 exponent for dispersion forces is very physically accurate, and comes from the following approximation (the Unsoeld approximation) in quantum mechanics:

$$E_{AB}^{\text{disp}} \approx -\frac{3}{2} \frac{I_A I_B}{I_A + I_B} \frac{\alpha'_A \alpha'_B}{R^6}$$

The above relates the energy of dispersion forces between particles  $A$  and  $B$  to the energies of first ionisation  $I_A$  and  $I_B$  and the polarisability volumes  $\alpha'_A$  and  $\alpha'_B$ .

The 12 exponent for Pauli repulsion is a somewhat good approximation to reality, but it was mostly chosen because it makes the repulsion term the square of the attraction term. This allows for many computational shortcuts.

Some other things to keep in mind include:

- Division of floats is computationally expensive, it is better to divide doubles, or not divide at all
- Square rooting floats and doubles is even more expensive, so better to start with the root and square it (and even more so for higher roots)
- Being implemented in Python (as opposed to a lower level language like C) means it will run slower overall, so a compiler should be used to optimise. Numba seems like the best option.
- As in the last milestone, there are some compromises (limited number of particles, larger time steps, smaller simulation volume, 2 dimensions instead of 3+) that will probably need to be made. I will choose them in a way that detracts accuracy the least (in particular, larger time steps makes accuracy quite bad).

### **3.1 Good resources**

A few helpful resources exist, which I haven't yet used properly but want to record here for my own notekeeping and also as a reference.

#### **3.1.1 Molecular Dynamics Simple Simulations (YouTube)**

<https://www.youtube.com/watch?v=CIbKCTfpG8>