Molecular Dynamics Introduction

Tristan Johnston-Wood

May 28, 2020

Contents

1 Classical Mechanics

Before discussing the physics of molecular dynamics, it is worth spending some time revising the foundations that it lays upon, classical mechanics. Like quantum mechanics, classical mechanics describes the motion and interplay between particles. Here, however, the laws of physics are deterministic, meaning that if we know the initial conditions of all the particles in our system, we can apply the laws of classical physics and determine with complete certainty the state of the system after an arbitrary time. While we know this is not the correct description of reality, it offers a surprisingly good approximation for heavy atoms, even at the molecular level.

We start by first defining some way to describe the positions of the particles, $r(t)$, at time t in our system. For this, we use Cartesian coordinates, **Eqn. 1**.

$$r(t) = (x(t), y(t), z(t)) \tag{1}$$

But how do we evolve our system in time? For this, we turn to Newton's second law: the force exerted by a particle produces an acceleration equal to that force divided by its mass, **Eqn. 2**.

$$\boldsymbol{a} = \frac{\boldsymbol{F}}{m}, \qquad \boldsymbol{F} = m\boldsymbol{a} \tag{2}$$

For a collection of particles, such as the atoms in a protein, we ascribe each its own mass, acceleration and force, **Eqn. 3**. This set of equations is referred to as the equations of motion for the system.

$$\boldsymbol{F}_i = m_i\boldsymbol{a}_i \tag{3}$$

To build up a more complete view of the motion of particles, as well as the positions of the particles, we need the velocities (or more specifically the momenta, p, ($p = mv$)) of the particles. The set of positions and momenta for a system is called the phase space. Phase space is $6N$ dimensional as for each particle there are three Cartesian coordinates for the position and momentum. We can write down the phase space vector, x, as the set of positions and momenta, **Eqn. 4**.

$$\mathrm{x}_t = (r_1(t), ..., r_N(t), p_1(t), ..., p_N(t)) \tag{4}$$

The phase space of your system comprises the microscopic state of your system, and this is the what we desire when running numerical simulations, such as molecular dynamics.

At this point, you have the theoretical minimum required to progress ahead to the next chapter – there really isn't that much needed to understand the basics of classical mechanics. Once you know the initial conditions, you can calculate the forces and evolve your system in time using Newton's second law. For those interested in the basis of deriving the equations of motions, see 'Statistical Mechanics – Theory and Molecular Simulation (M. E. Tuckerman)' for details on Lagrangian and Hamiltonian mechanics.[1]

2 Statistical Mechanics and Expectation Values

Classical mechanics states that if we know the point in phase space at any time (i.e. the positions and momenta), then we know the past and future evolution of our system. Applying these laws would enable us to obtain thermodynamic properties, such as energy, temperature, pressure etc. However, while we can in principle solve Newton's equations for a classical system, real systems consist of $\sim 10^{23}$ particles and exhibit highly nontrivial interactions. Moreover, the memory requirement for a computer to compute just one phase space point for 10^{23} particles is excessive.

To the rescue comes statistical mechanics. This is a theory that relates the microscopic laws of motion with macroscopic properties by applying statistical averaging to a large number of particles. This is applied through the basis of an *ensemble*, which is a collections of systems that share common macroscopic properties. One example of an ensemble is the canonical ensemble, often referred to as the NVT ensemble in molecular dynamics. Statistical mechanics is a rich and extensive theory but we will take only the relevant results from it to explain molecular dynamics.

We therefore turn out attention to expectation values. Broadly speaking, these relate a system property, a, to its ensemble average, A. This can be expressed heuristically as follows, **Eqn. 5**.

$$A = \sum_i P_i a_i \equiv \langle a \rangle \tag{5}$$

Here, we weight the system property, a_i, by the probability, P_i, of it occurring at a particular point in phase space and sum them up. In the thermodynamic limit, the sum becomes an integral. Moreover, we can express the probability of a system property occurring as the Boltzmann distribution divided by the partition function, Z, **Eqn. 6**.

$$P_i = \frac{e^{-\epsilon_i / k_B T}}{Z} \tag{6}$$

The partition function is a measure of the number of microstates in the phase space accessible within a given ensemble. Therefore, for a canonical ensemble, NVT, we can expression an ensemble average as the Boltzmann weighted integral of the property, normalised by the partition function, **Eqn. 7**.

$$A = \frac{\int a e^{-\beta E(\mathrm{x})} d\tau}{Z} = \frac{\int a e^{-\beta E(\mathrm{x})} d\tau}{\int e^{-\beta E(\mathrm{x})} d\tau} \equiv \langle a \rangle \tag{7}$$

So we can see that by sufficiently sampling over phase space (using e.g. molecular dynamics) we can obtain macroscopic properties. An example of one property we could obtain is the pressure of our system, **Eqn. 8**.

$$P = \frac{\int p e^{-\beta E(\mathrm{x})} d\tau}{\int e^{-\beta E(\mathrm{x})} d\tau} \equiv \langle p \rangle \tag{8}$$

Note, we cannot directly compute Z but we can sample a distribution to obtain thermodynamic averages.

3 Molecular Dynamics

Armed with our toolkit of classical statistical mechanics, we know that if we can sample the phase space points in a system, we can compute expectation values and understand the behaviour of the system. One method to sample phase space is by applying Newton's equation to the atoms in a periodic box.

3.1 Verlet Algorithm

Suppose we wish to know the Cartesian coordinates of N particles in a system at time $t+\Delta t$. This expression can be written as a Taylor series truncated at third order, **Eqn. 9** (see Appendix for Taylor series summary). Note that the dot notation refers to the time derivative of the function.

$$r_i(t+\Delta t) \approx r_i(t) + \Delta t\ \dot{r}_i(t) + \frac{\Delta t^2}{2}\ddot{r}_i(t) \tag{9}$$

But first recall that velocity, $v_i(t)$, is the first time derivative of position. Secondly, recall Newton's second law of motion, **Eqn 10**.

$$\begin{aligned} v_i(t) &= \frac{dr_i(t)}{dt} \\ F_i(t) = m_i a_i(t) = m_i \frac{dv_i(t)}{dt} &= m_i \frac{d^2 r_i(t)}{dt^2} \end{aligned} \tag{10}$$

Therefore, we can substitute these expressions into our Taylor series, giving **Eqn. 11**.

$$r_i(t+\Delta t) \approx r_i(t) + \Delta t\ v_i(t) + \frac{\Delta t^2}{2m_i} F_i(t) \tag{11}$$

We now have an expression for computing the positions of a set of particles at time $t+\Delta t$ given their positions, velocities and forces at time t. However, we can't easily calculate the velocities given only the positions and force at one time. But we can obtain a velocity-independent scheme by considering $r_i(t-\Delta t)$, **Eqn. 12**.

$$r_i(t-\Delta t) \approx r_i(t) - \Delta t\ v_i(t) + \frac{\Delta t^2}{2m_i} F_i(t) \tag{12}$$

Note only the velocity changes sign as the time in the force term is quadratic. Adding **Eqn. 11** and **Eqn. 12**, we get **Eqn. 13**.

$$r_i(t+\Delta t) + r_i(t-\Delta t) \approx 2r_i(t) + \frac{\Delta t^2}{m_i}F_i(t) \tag{13}$$

And so rearranging for $r_i(t+\Delta t)$, gives **Eqn. 14**.

$$r_i(t+\Delta t) \approx 2r_i(t) - r_i(t-\Delta t) + \frac{\Delta t^2}{m_i}F_i(t) \tag{14}$$

Thus, given a set of initial conditions $\{r_n(0)\}$, $\{v_n(0)\}$, we can obtain coordinates at $\{r_n(\Delta t)\}$ and generate a trajectory of arbitrary length. Note that while we can generate velocities at any point, the algorithm does not evolve the velocities. We want a method that reconstructs the full phase space consists of positions and momenta.

3.2 Velocity Verlet Algorithm

Consider again $r_i(t+\Delta t)$, **Eqn. 11**. If we instead start from $r_i(t+\Delta t)$ and $v_i(t+\Delta t)$ and compute $F_i(t+\Delta t)$, we can evolve backwards in time to $r_i(t)$ using a time step of $-\Delta t$, **Eqn. 15**.

$$r_i(t) \approx r_i(t+\Delta t) - \Delta t\ v_i(t+\Delta t) + \frac{\Delta t^2}{2m_i}F_i(t) \tag{15}$$

Substituting **Eqn. 15** into **Eqn. 11**, we get an expression for the velocities at time $(t+\Delta t)$, **Eqn. 16**.

$$\begin{aligned} & r_i(t) \approx r_i(t) + \Delta t\ v_i(t) + \frac{\Delta t^2}{2m_i}F_i(t) - \Delta t\ v_i(t+\Delta t) + \frac{\Delta t^2}{2m_i}F_i(t) \\ \therefore\quad & v_i(t+\Delta t) \approx v_i(t) + \frac{\Delta t}{2m_i}\big[F_i(t) + F_i(t+\Delta t)\big] \\ \text{and}\quad & r_i(t+\Delta t) \approx r_i(t) + \Delta t\ v_i(t) + \frac{\Delta t^2}{2m_i}F_i(t) \end{aligned} \tag{16}$$

These velocity Verlet equations solve the positions and velocities (phase space) simultaneously.

Leapfrog Algorithm. The leapfrog algorithm is very similar to velocity-Verlet but computes position and velocity at differences of half a time-step (i.e. $r_i(t+\Delta t)$ and $v_i(t+\frac{1}{2}\Delta t)$. While velocity-Verlet is marginally more accurate (to reproduce the correct pressure ensemble) than leapfrog, leapfrog is quicker to compute due only half as many communication call being required compared to velocity-Verlet.

Other integrators. Although these equations are truncated Taylor series, and other methods may be more accurate e.g. predictor-corrector methods, they are neither time-reversible or symplectic (and so don't obey Hamilton's equations). Therefore, these alternative methods experience large drifts in the total energy.

Choosing initial conditions. If we wish to sample from a canonical distribution, we need to generate initial velocities according to the Maxwell–Boltzmann distribution, **Eqn. 17**, expressed in terms of speed.

$$f(v) = \left(\frac{2m^3}{\pi k_B^3 T^3}\right)^{\frac{1}{2}} v^2 e^{-mv^2/2k_BT} \tag{17}$$

A method called Box-Muller sampling allows for the generation of Gaussian random numbers using random numbers sampled from a uniform distribution.

4 Canonical Ensemble and Thermostats

While using Newton's equation to sample the phase space in the constant energy space (microcanonical ensemble, NVE) is perfectly reasonable, experiments are often performed under the constant temperature regime (canonical ensemble, NVT). Therefore, we need to modify our simulations such that the energy is allowed to fluctuate in order to keep the temperature constant. This is where the idea of a thermostat comes in. The role of a thermostat is to act as a thermal reservoir, ensuring that the average kinetic energy remains roughly constant. A number of thermostats have been developed, and we will go over a few of them here.

4.1 Berendsen temperature coupling

This thermostat controls the temperature of the system by rescaling the velocities of each particle every step by a time–dependent factor λ (usually $0.8 \leq \lambda \leq 1.25$). The deviation from the actual temperature, T, to the target temperature, T_0, follows an exponential decay, meaning the system approaches the target temperature exponentially quickly, as controlled by the time constant τ, **Eqn. 18**.

$$T - T_0 = Ae^{-t/\tau} \tag{18}$$

While this allows the target temperature to be reached quickly, the thermostat suppresses fluctuations in the kinetic energy, meaning the correct NVT ensemble is not obtained. Therefore, in practice this thermostat is not often used as the velocity rescaling thermostat is preferred.

4.2 Velocity-rescaling temperature coupling

The velocity–rescaling thermostat is identical to the Berendsen thermostat but includes an additional term, dK, which ensures the correct kinetic energy distribution is obtained. This is a choice thermostat to use as it retains the speed of Berendsen as well as the correct canonical ensemble.

4.3 Andersen thermostat

The Andersen thermostat works by modifying the velocities of every particle according to a Maxwell-Boltzmann distribution. Either all particles' velocities are randomised every $\tau_T/\Delta t$ timestep (Anderson–massive) or all particles' velocities are randomised every Δt timestep with probability $\Delta T/\tau_T$. In both cases, the shorter the coupling timescale, τ_T, the more often (or higher probability) the particles' velocities get re-scaled. While this avoids some ergodicity issues, the kinetics are slowed and so this is not suitable for kinetic studies or for transport properties.

4.4 Nosé–Hoover temperature coupling

As we stated, the Berendsen thermostat does not produce the correct canonical ensemble and so one solution is the Nosé–Hoover thermostat. This thermostat works by introducing friction and a thermal reservoir. The Hamiltonian for this new system is given in **Eqn. 19**.

$$H_N = \sum_{i=1}^{N} \frac{\boldsymbol{p}_i^2}{2m_i s^2} + U(\boldsymbol{r}_1, ..., \boldsymbol{r}_N) + \frac{p_s^2}{2Q} + gkT \ln s \tag{19}$$

Where the s and p_s terms act as the friction terms. It can be shown that sampling using this Hamiltonian will generate a canonical distribution. Using this thermostat produces oscillatory relaxation (c.f. Berendsen produces exponential relaxation). Moreover, this method may be non-ergodic and so Nosé–Hoover chains are used (allowing for a wider section of phase space to be sampled). The details can be realised using the Liouville operator, however this is beyond the scope of this introduction.

5 Isobaric – Isothermic Ensemble and Barostats

The final chapter we will discuss is the need for not only a constant temperature ensemble, but a constant temperature and constant pressure ensemble, the NPT or isobaric–isothermic ensemble (which will often represent actual experimental conditions). In addition to the thermal reservoir, there is a pressure reservoir. This can be viewed as coupling a second larger system to the original system via piston, which extends and contracts so as to keep the pressure in the original system constant (the volume may change). As with the thermostats, there are a number of barostats available.

5.1 Berendsen pressure coupling

The Berendsen barostat works similar to the Berendsen thermostat but it rescales the coordinates and the box by a scaling matrix μ. The pressure reaches the reference pressure according to **Eqn. 20**.

$$P - P_0 = Ae^{-t/\tau_p} \tag{20}$$

It is not surprisingly that while this method does produce the correct pressure, it does not yield the NPT ensemble. This is due to fluctuations in the pressure being suppressed.

5.2 Parrinello–Rahman Pressure Coupling

The Parrinello–Rahmann barostat correctly produces the NPT ensemble. It does this by modifying the equations of motion by a matrix $\boldsymbol{M}$, **Eqn. 21**, which is defined in terms of box vectors that themselves obey an equation of motion correcting the pressure.

$$\frac{d^2\boldsymbol{r}_i}{dt^2} = \frac{\boldsymbol{F}_i}{m_i} - \boldsymbol{M}\frac{d\boldsymbol{r}_i}{dt} \tag{21}$$

Although this method does produce the NPT ensemble, if your pressure is far from equilibrium, Parrinello–Rahman could cause large box oscillations. Therefore, it is best to equilibrate your system using an exponentially decaying barostat (Berendsen) before using Parrinello–Rahman.

Summary. This is a very brief overview of some of the aspects of molecular dynamics. For more detail, see the references at the end of this document.[1–3]

5.3 Appendix

Mathematics Summary. In obtaining expressions for the phase space points in molecular dynamics, the use of Taylor series are employed. For those that need a reminder of what a Taylor series is, the following may be useful. First consider a function, such as $y = \sin(x)$.

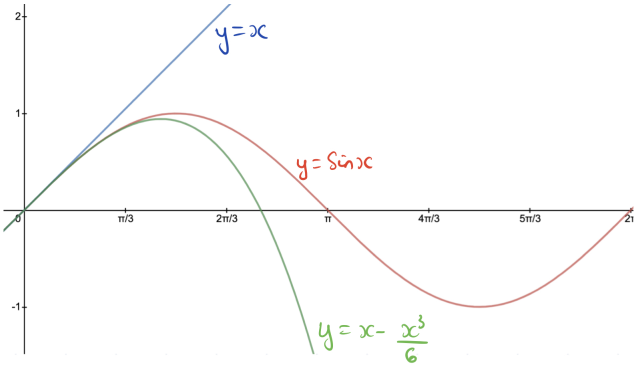

Figure 1: $y = \sin(x)$

This curve can be approximated around the point $x = 0$ using the function $y = x$. The further you strive from $x = 0$, the worse this approximation gets. Therefore, we can add another term, $y = x - x^3/6$. This further improves the approximation. In fact, the more polynomial terms you add (with the corrected coefficients and powers) the closer you get to the true function, eventually reproducing it exactly. There is a general formula that can be used to generate these polynomials.

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n \tag{22}$$

You can think of the higher order terms as capturing more of the curvature of the function. Moreover, we use derivatives of the function in higher order terms to describe how the function curves.

This approximation is incredibly useful in physics as we can approximate complex functions using simple polynomials. You can see that the approximation $y = x - x^3/6$ for $y = \sin(x)$ is very accurate up to even $\pi/3$.

References

[1] M. Tuckerman, *Statistical mechanics: theory and molecular simulation*, Oxford university press, 2010.

[2] M. Abraham, D. Van Der Spoel, E. Lindahl and B. Hess, *The GROMACS development team GROMACS user manual version 5.0. 4*, 2014.

[3] L. Susskind, *The Theoretical Minimum, Statistical Mechanics Lecture 1, YouTube*, 2013.