Beyond Local Optimization: Introduction to Molecular Dynamics (MD)

So far in this class and in labs 1-4 we have focused on local optimization of chemical structures with the hopes of finding stable, probable structures in a chemical system. You have learned both how local optimization works and how we can use it to predict the stability and catalytic ability of materials. Today we will start to learn about other tools that are commonly used in molecular simulations.

In Lab 5, you will utilize molecular dynamics simulations. Molecular Dynamics (MD) is a computer simulation used to mimic how atoms move over time in a physical system. The following outlines the content of the in class worksheet and how it relates to MD simulations:

1. Physics Review: Review of physics equations used in MD
2. Euler’s Method: Introduction to the mathematical approximations used in MD and how it can be used to derive the MD equations
3. Ensembles: What thermodynamic variables do we want to fix in our simulation?

Physics Review

We can think of the motion of nuclei in a chemical system the same way physicists think of classical mechanics. Below is a physics review. We will be using these equations and concepts throughout this worksheet:

2. Recall that given a position of an object at a time t, r(t),
   a. the velocity, v, of that object is \( v(t) = \frac{d}{dt} r(t) \)
   b. the acceleration, a, of the object is \( a(t) = \frac{d}{dt} v(t) \)
3. The momentum, p(t), is defined as follows: \( p = mv(t) \) where m is the mass and v is velocity.
4. The force, F, is also related to these equations and the PES in a few ways:
   a. \( F(r(t)) = ma(t) \)
   b. The force is also influenced by the configuration, r, on the potential energy surface by, \( F(r(t)) = -\nabla V(r(t)) \)

Euler’s Method

Euler’s method is a procedure that we can use to numerically simulate how objects move over time. This method comes from information you already know. Let’s start from the definition of a derivative (without taking a limit as \( dt \) goes to 0):

\[
 f'(t) \equiv \frac{f(t + dt) - f(t)}{dt}
\]

where \( dt \) is a sufficiently small number (we will explore what this value should be in lab 5!). Use algebra and the equation above to solve for \( f(t+dt) \)
\[ f(t + dt) = f(t) + f'(t)dt \]

This is Euler’s method! To find how a function changes over time we would need to start at \( t=0 \). Given a value of \( dt \) that is ‘small enough’, we would need to iterate through the above equation. Each time we iterate through this equation, we move forward in time by \( dt \). For example, to approximate what \( f \) is at \( t=4dt \) given \( f(0) = f_0 \) you would need to iterate through the above equations four times (See figure 1). In this way we can find how a function changes over time.

![Figure 1. Illustration of a surface, \( f(t) \), in blue and an approximated surface with Euler’s method in red. The notation above means \( f_i = f(n \times dt) \), (e.g. \( f_i = f(4 \times dt) \) ).](image)

In order to simulate how objects move over time, we will need to numerically integrate the equations of motion. To do this we can use Euler’s method and equations we know from physics.

First write down Euler’s method for position, \( r(t+dt) \), in terms of \( r \) and \( r' \). Use the physics review to substitute \( r' \). Note: we could write the equations in terms of \( p \) or \( v \), but to keep everyone consistent, write your equation in terms of \( p \).

\[
\begin{align*}
  r(t + dt) &= r(t) + r'(t)dt \\
  p &= mv \rightarrow r'(t) = \frac{p(t)}{m} \\
  r(t + dt) &= r(t) + \frac{p(t)}{m}dt
\end{align*}
\]

Then write down Euler’s method for momentum, \( p(t+dt) \), in terms of \( p \) and \( p' \). Use the physics review to substitute \( p' \).

\[
\begin{align*}
  p(t + dt) &= p(t) + p'(t)dt \\
  p &= mv \rightarrow p' = ma = F \\
  p(t + dt) &= p(t) + F(r(t))dt
\end{align*}
\]

You have just derived the MD equations using Euler’s method! Since the momentum, \( p(t) \), is required to simulate how the positions, \( r(t) \), change over time and vice versa, you will need to use Euler’s method on both the momentum and the position simultaneously in molecular dynamic simulations.
Next, let's look at an example. Using the equations for \( r(t+dt) \) and \( p(t+dt) \) you just found, let's see how the position and momentum change over time with the following potential energy surface and mass:

\[
V(r) = (r-2)^2 \\
\text{Mass} = 1.
\]

Fill out the chart below and draw the trajectory on the graph below. We call what you are solving molecular dynamics **trajectory**, how atoms move over time. Do not assume that your total energy will be fixed in this simulation!

**Potential Energy Surface and Mass**

\[
V(r) = (r-2)^2 \\
\text{Mass} = 1.
\]

<table>
<thead>
<tr>
<th>Time</th>
<th>Position ((r))</th>
<th>Momentum ((p))</th>
<th>Force (-2(r-2))</th>
<th>Potential Energy ((r-2)^2)</th>
<th>Kinetic Energy (0.5mv^2)</th>
<th>Total Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>(t=0)</td>
<td>3</td>
<td>1</td>
<td>-2</td>
<td>1</td>
<td>0.5</td>
<td>1.5</td>
</tr>
<tr>
<td>(t=0.1)</td>
<td>(r(0.1) = r(0) + \frac{p(0)}{m} \times 0.1)</td>
<td>(p(0.1) = p(0) + F(r(0)) \times 0.1)</td>
<td>-2.2</td>
<td>1.21</td>
<td>0.32</td>
<td>1.53</td>
</tr>
<tr>
<td></td>
<td>(r(0.1) = 3 + \frac{1}{1} \times 0.1 = 3.1)</td>
<td>(p(0.1) = 1 + \frac{-2}{0.1} = 0.8)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(t=0.2)</td>
<td>(r(0.2) = r(0.1) + \frac{p(0.1)}{m} \times 0.1)</td>
<td>(p(0.2) = p(0.1) + F(r(0.1)) \times 0.1)</td>
<td>-2.35</td>
<td>1.39</td>
<td>0.168</td>
<td>1.56</td>
</tr>
<tr>
<td></td>
<td>(r(0.2) = 3.1 + \frac{0.8}{1} \times 0.1)</td>
<td>(p(0.2) = 0.8 + \frac{-2.2}{0.1} = 0.58)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(t=0.3)</td>
<td>(r(0.3) = r(0.2) + \frac{p(0.2)}{m} \times 0.1)</td>
<td>(p(0.3) = p(0.2) + F(r(0.2)) \times 0.1)</td>
<td>-2.48</td>
<td>1.53</td>
<td>0.059</td>
<td>1.59</td>
</tr>
<tr>
<td></td>
<td>(r(0.3) = 3.18 + \frac{0.58}{1} \times 0.1)</td>
<td>(p(0.3) = 0.58 + \frac{-2.36}{0.1} = 0.344)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(0.344\)
In your own words, describe the initial conditions of this molecular dynamics trajectory.

The initial positions for the diatomic is that the bond is stretched passed the equilibrium bondlength. The initial velocities are pushing the atoms apart.

Does the trajectory do what you would expect in the short amount of time simulated?

Yes, the bond length initially increases due to the initial velocity. The velocity decreases due to the force acting on the atoms with the stretched bondlength.

Does the total energy remain fixed in the simulation? Would you expect it to?

No, the total energy does not remain fixed although I would expect it to. The time step and the first order numerical integrator contribute to the error.

What is the goal of performing molecular dynamics? If you were to continue the trajectory from the exercise above what would happen?

The goal of molecular dynamics is to solve how atoms in a molecular system move over time. If I were to continue the trajectory give above, I would expect that diatomic molecule to eventually get to 0 velocity and then, the direction of the velocity would change and the atoms would start to move closer together. Then the atoms would move past the ideal bond length (where the maximum velocity would occur) to compressed bond and so on. Overall, the atoms would oscillate back and forth around the ideal bond length of the diatomic molecule.

**How is molecular dynamics different from local optimization?**

To answer this question, let’s first revisit gradient descent. Assume that your initial r is 3 and alpha is equal to 0.25. What would the next two configuration in a gradient descent trajectory be?

\[
r_0 = 3 \\
\begin{align*}
  r_1 &= r_0 + \alpha F(r_0) = 3 + 0.25(-2) = 2.5 \\
  r_2 &= r_1 + \alpha F(r_1) = 2.5 + 0.25(-1) = 2.25
\end{align*}
\]
What is the goal of performing local optimization and how does it differ from molecular dynamics? If you were to continue the trajectory from the exercise above what would happen?

The goal of local optimization is to find a nearby local minimum as quickly as possible. In local optimization, each step should be taken to move the atomic system to a lower potential energy configuration. Once, the forces (or derivative) is close to zero, no more steps are taken.

The goal of molecular dynamics is to solve how atoms move over time. In a MD simulation, the atoms would naturally oscillate around the local minimum.

Velocity-Verlet Algorithm: A more accurate numerical integrator

Euler’s method is a first order (only uses first derivatives) numerical procedures for solving differential equations. However, it is rarely used since more accurate methods with improved numerical stability (does a better job keeping the total energy close to fixed) exist with no additional cost. A commonly used numerical integrator that will be used in this class is the velocity-Verlet algorithm. This method uses higher order approximations (or approximate second derivatives). I will not derive these equations in this class. Here is how the equations change:

\[
\begin{align*}
    p(t + 0.5\Delta t) &= p(t) + f(r(t))\Delta t/2 \\
r(t + \Delta t) &= r(t) + p(t + 0.5\Delta t)\Delta t/m \\
f &= F(r) \\
p(t + \Delta t) &= p(t + 0.5\Delta t) + f(r(t + \Delta t))\Delta t/2
\end{align*}
\]

In the equations above \(dt\) is replaced with \(\Delta t\), which is known as the timestep. In general the higher order approximations will be a more accurate numerical integrator.

To implement this procedure on a computer we would need to loop through the above equations for the specified time \(N\Delta t\). Below is pseudocode for microcanonical ensemble (big word that will be defined in the next section) molecular dynamics:

- \(r(0) = r_0\) \hspace{1cm} Initialize the positions
- \(p(0) = p_0\) \hspace{1cm} Initialize the momentum
- \(f = F(r)\) \hspace{1cm} Calculate the initial force

for i in the range N:  \hspace{1cm} Loop through the velocity verlet equations \(N\) times

Velocity Verlet Equations Below

- \(p(t + 0.5\Delta t) = p(t) + f(r(t))\Delta t/2\)
- \(r(t + \Delta t) = r(t) + p(t + 0.5\Delta t)\Delta t/m\)
- \(f = F(r)\)
- \(p(t + \Delta t) = p(t + 0.5\Delta t) + f(r(t + \Delta t))\Delta t/2\)
Ensembles

When running MD simulations, you will need to select an ensemble, or assume that certain thermodynamic variables are fixed in your simulation. Here are two commonly used ensembles:

1. Canonical ensemble: Number of atoms (N), Volume (V), and Temperature (T) are fixed.
2. Microcanonical ensemble: Number of atoms (N), Volume (V), and Total Energy (E) are fixed.

In both of these ensembles the number of atoms and volume is fixed. In the canonical ensemble, the temperature is fixed. In the microcanonical ensemble, the total energy is fixed.

We will primarily use the canonical ensemble where temperature is fixed in our chemical system. However, the microcanonical ensemble can be used to test if your MD simulation conserves energy for the selected finite difference time step (Δt). This test is critical for determining a timestep for your MD simulations. Lab 5 will teach you how to select this parameter based on your chemical system, the mass of the atoms, the amount of kinetic energy, etc.

For the canonical ensemble, we need a way for energy to enter and leave the system to know how to fix the temperature of the simulation.

Thermostats: How to Fix the Temperature in a Simulation?

*Andersen thermostat*

One commonly used method to fix temperature in a computer simulation is the Andersen thermostat. The idea of the Andersen thermostat is that your simulation (inner box in figure below) is inside of a large heat bath (outer box) which controls the temperature of your simulation. Occasionally instantaneous interaction between fictional particles and atoms within the simulation will occur where an exchange in kinetic energy will takes place.

*Figure.* Above is a pictorial representation of this thermostat. The blue atoms represent the atoms in your simulation, while the orange atoms are the fictional particles. The circled atoms represent an interaction between two particles where an exchange in kinetic energy would take place.
Two parameters control the thermostat: 1) \( \tau \) determines the time interval in which collisions take place; 2) \( \alpha \) determines the strength of the interaction between the two particles.

The collisions are described as a random process so that the probability of a collision is,

\[
P_{\text{col}} = 1 - e^{-\Delta t/\tau}
\]

The kinetic energy contained within the fictional particles is determined from the Boltzmann distribution at the given temperature,

\[
p_{\text{heat-bath}} \propto e^{-p^2/2mkT}
\]

The thermostat can be softened by taking a linear combination of the momentum of the fictional particle, \( p_{\text{new}} \), and the atom, \( p_{\text{old}} \).

\[
p = p_{\text{current}} + \alpha p_{\text{heat-bath}}
\]

The pseudo code for molecular dynamics using the NVT ensemble is as follows:

\[
\begin{align*}
    r(0) &= r_0 & \text{Initialize the positions} \\
    p(0) &= p_0 & \text{Initialize the momentum} \\
    f &= F(r) & \text{Calculate the initial force} \\
    \text{Initialize the Thermostat variables (}\alpha, \tau, kT, P_{\text{col}}\text{); Code not listed} \\
    \text{for i in the range N: Loop through the velocity verlet equations N times} \\
    \quad \text{Velocity Verlet Equations Below} \\
    & \quad p(t + 0.5\Delta t) = p(t) + f\Delta t/2 \\
    & \quad r(t + \Delta t) = r(t) + p(t + 0.5\Delta t)\Delta t/m \\
    & \quad f = F(r) \\
    & \quad p(t + \Delta t) - p + f\Delta t/2 \\
    \quad \text{Andersen Thermostat equations are below} \\
    \quad \text{For all atoms in the chemical system:} \\
    & \quad \text{If Random number between 0 and } 1 < P_{\text{col}}: \\
    & \quad \quad \text{Select a new momentum for from the Boltzmann distribution(} \text{Gaussian distribution) } \\
    & \quad \quad p_{\text{heat-bath}}(\text{atom } i) \propto e^{-p^2/2mkT} \\
    & \quad \quad \text{Reset the momentum of this atom at current time} \\
    & \quad \quad p(\text{atom } i) = \sqrt{1 - \alpha^2p_{\text{current}} + \alpha p_{\text{heat-bath}}}
\end{align*}
\]