Week 2 Problem Set

For this homework set, you will be writing a class to represent a harmonic oscillator. A harmonic oscillator is important in both classical and quantum physics. In molecular mechanics, it can be used to represent bonds in molecules.

The first part of this section gives some background and equations. Then, you should follow the specifications outlined in the **Specifications** section to write your class.

Background: Diatomic Molecules - The Harmonic Oscillator

The behavior of bonds is often approximated by a harmonic oscillator.

The classical harmonic oscillator is also called a "mass on a spring", and is described using Hooke's Law. Hooke's Law states the force needed to compress or stretch a spring is negatively proportional to the spring constant, k and the displacement (amount spring is stretched or compressed from equilibrium, x)

$$F(x) = -kx$$

The potential energy of the harmonic oscillator is described by

$$U(x) = \frac{1}{2}kx^2$$

The diatomic (consisting of two bonded atoms) is the simplest type of molecule that has a bond. For our purposes, we will consider a diatomic molecule to be two masses connected by a spring with the behavior described above.

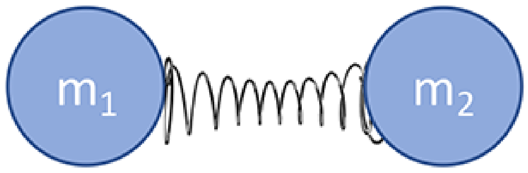

Figure 1: The bond as two masses on a spring.

We will be writing a class to represent a diatomic molecule modeled by a harmonic oscillator.

To simplify calculation, we will consider the system as a reduced mass, μ, connected by a spring to an infinite mass. The reduced mass is calculated using

$$\mu = \frac{m_1 m_2}{m_1 + m_2}$$

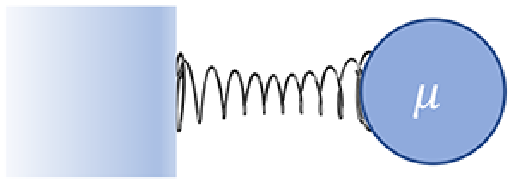

Figure 2: Reduced mass diatomic.

For this task, you will write a class called `Diatomic` to represent a diatomic molecule. In addition to calculating the force and potential energy, we will also want to calculate the kinetic energy, which will require the mass to have a velocity.

$$K(v) = \frac{1}{2}\mu v^2$$

The total energy of the oscillator is

$$Total\ Energy = Kinetic\ Energy + Potential\ Energy$$

The total energy of our classical harmonic oscillator will be **constant**.

The time dependent behavior (position and velocity) of the oscillator can be solved analytically. The position and velocity at a certain time, t can be calculated with the following equations.

$$x(t) = A\cos(\omega t + \phi)$$

$$v(t) = \frac{dx(t)}{dt} = -A\omega\sin(\omega t + \phi)$$

where is the maximum amplitude (maximum distance bond can stretch according to this model) $A = \sqrt{2\frac{Total\ Energy}{k}}$, $\omega = \sqrt{\frac{k}{\mu}}$, and ϕ depends on the initial separation distance since $x(0) = A\cos(\phi)$.

Specifications

Object Oriented Programming

Because we don't yet know all of the strategies for writing Python classes, some aspects of this class may seem clunky. The parameters in this system are highly coupled (updating the position should update the velocity, the kinetic energy, and the potential energy). We will see later in the course how we can use other Python language features to make it simpler.

For now, create a class called `Diatomic` to follow the following specifications:

You may use anything in the Python Standard Library, NumPy, and Matplotlib for this assignment.

1. The constructor should take a reduced mass, a force constant, an initial separation, and an initial velocity. These should all be stored as instance attributes.
2. The class should have instance methods `potential_energy` , `kinetic_energy` . These methods should not take any parameters besides `self` . They should use

the data associated with the instance. The calculated values for these should be **returned**.

3. Use your methods from step 2 to add calculation of ω, A, and ϕ to the constructor. Store these as instance attributes according to the equations given above.
4. Add an instance method for computing the position at a certain time and an instance method for computing the velocity at a certain time using the analytical solution. Name these methods `analytical_position` and `analytical_velocity`. Note that these values only depend on model constants (`mass`, `k`, `amplitude`, etc), and not on separation distance or velocity.
5. Add a module `test_diatomic.py`. Create tests to
   - Check that your object is propery constructed.
   - Check the `kinetic_energy` and the `potential_energy` methods.
   - Check the `analytical_position` and `analytical velocity` method.

Exceptions and Inheritance

1. Add one custom Exception type. Your Exception should inherit from an appropriate `Exception` class. Add appropriate error checking and raising of this error to your class.

Formatting and Makefiles

Create a Makefile which has the following targets:

1. `lint` - runs `black` and `flake8` on your homework module. You should not see any errors from either one after using this target.
2. `test` - uses pytest to run your test cases.
3. `plot` - uses Python (matplotlib) to create a plot of analytical position (y axis) vs time (x axis) and analytical velocity (y axis) vs time (x axis). Plot at least 2 periods of movement (suggest using time intervals of 0.1).

Documentation and Reflection

Add a `README.md` to your repo. Explain what this project is, and how to use the `Makefile`. Pretend you are writing for an audience on GitHub who is not familiar with the assignment or what your repository does.

Answer the following questions and add them to your README.

1. What kind of custom error did you make and why?
2. Are there any limitations to your class, or do you think we should have made any different design choices? If so, what are they?