Lab 7: Time Evolution of Coupled Oscillators

Objectives

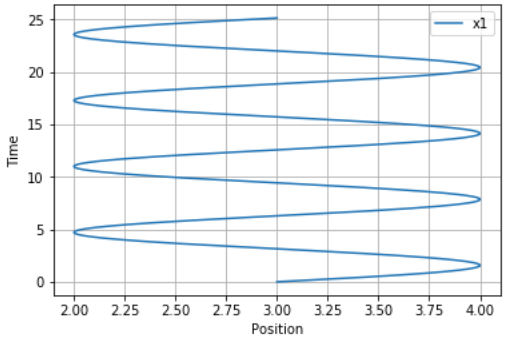
By the end of this lab you should:

- Know how to manipulate complex numbers and vectors.
- Be able to compute time evolution of a arbitrary initial positions.
- Construct the matrix for a different set of two coupled oscillators.

Prelab question:

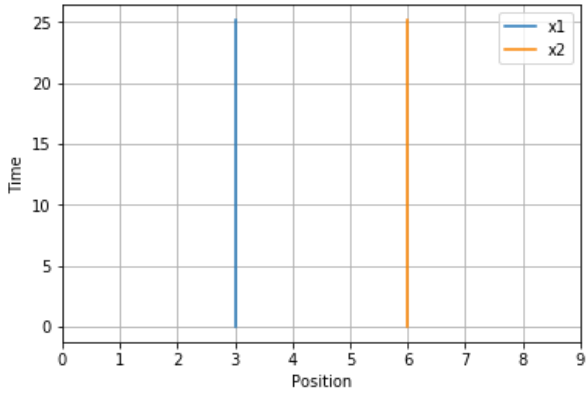
In this lab we will make graphs of the position of each mass as a function of time. To make it easier to see how those graphs correspond to the motion of the pair of masses we have been discussing, we will make these graphs with ***time on the vertical axis*** and ***position on the horizontal axis.***

For a single mass on a single spring the graph would like this:



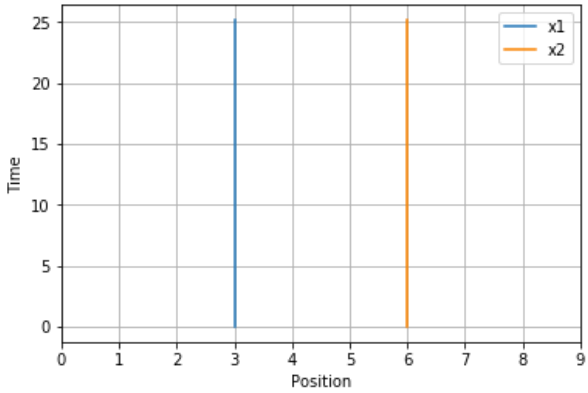
1. Where on the graph is the initial time?

   For the two mass case the graph for the position of each mass *if the masses are not moving* is



2. Draw the position as a function of time for both mass on the graph *above* if their initial state is the symmetric state, $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Draw them as a ***solid*** line.
3. Draw the position as a function of time for both mass on the graph *above* if their initial state is the antisymmetric state, $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$. Draw them as a ***dashed*** line.

4. Draw the position as a function of time for both mass on the graph ***below*** if their initial state is the state, $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$. (Sketch what you think this might look like — it is not intended that you do a calculation for this)



Background: Coupled oscillators are an example of an application of eigenvectors and eigenvalues in physics. As shown in lecture, the equations of motion for a two masses and three springs can be written

$$\frac{d^2\vec{v}}{dt^2} = -M\vec{v}$$

where M is the matrix

$$M = \frac{k}{m}\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}.$$

We also showed that the eigenvectors of M evolve in a straightforward way. If $\vec{v}_e$is an eigenvector with eigenvalue λ_e, i.e. if

$$M\vec{v}_e = \lambda_e\vec{v}_e,$$

then the equation of motion for $\vec{v}_e$ becomes

$$\frac{d^2\vec{v}_e}{dt^2} = -\lambda_e\vec{v}_e.$$

The solution to this equation is the real part of

$$\vec{v}_e(t) = Ce^{i\phi}e^{i\omega_e t}\vec{v}_e(0),$$

where $\omega_e = \sqrt{\lambda_e}$. For the special case when the initial velocities of the two masses is zero, which is the only case we will consider, ϕ=0, and the real part becomes

$$\vec{v}_e(t) = C\cos(\omega_e t)\vec{v}_e(0).$$

As we discussed in class, it is very convenient to use *normalized* eigenvectors; fortunately, that is what numpy's eigh produces already.

In that case, then for any initial conditions

$$\vec{x}_0 = \begin{pmatrix} x_{10} \\ x_{20} \end{pmatrix}$$

the time dependence is given by

$$\vec{x}(t) = a\cos(\omega_1 t)\hat{v}_{e1} + b\cos(\omega_2 t)\hat{v}_{e2}$$

where

$$a = \vec{x}\cdot\hat{v}_{e1} \quad \text{and} \quad b = \vec{x}\cdot\hat{v}_{e2};$$

in both of these the normalized eigenvectors are called $\hat{v}_{e1}$ and $\hat{v}_{e2}$.

Part 1: Write a function to calculate the frequencies and normalized eigenvectors of a matrix

- Your function should be have signature `calc_frequencies_and_modes(matrix, k_over_m)`
- It should return two things: frequencies (not eigenvalues) and normalized eignevectors.
- Check that it returns the frequencies and vectors you expect for the matrix above, assuming `k_over_m` is one.

Part 2: Calculate the coefficients a and b for arbitrary initial conditions

- Write a function with signature `calc_components_from_initial_conditions(x_init, modes)` that calculates the coefficients *a* and *b* defined above and returns *a* and *b*.
- Check that it produces the result you expect in three cases:
  - `x_init` is the vector $\hat{v}_{e1}$.
  - `x_init` is the vector $\hat{v}_{e2}$.
  - `x_init` is the vector $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

Part 3: Calculate $\overrightarrow{x}(t)$ for arbitrary initial conditions

- Write a function with signature that calculates the position of both masses at future times.
- For your function to work, you will need to define time the way it is in the starting code (with a specific shape).
- Your new function should call the other two functions you wrote.
- Check your that your function works by using it to find $\overrightarrow{x}(t)$ for each of the two eigenvectors.

Part 4: Other initial conditions

- Now try your code for the initial conditions $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and comment on the result. How does it differ from the normal modes?
- Try the code for the initial conditions $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and comment on the result. How does it compare to the previous case?
- Try one other set of initial conditions (your choice).

Part 5: Another mass-spring system

- Determine the appropriate matrix for the case when all three springs are the same, but $m_2 = 10m_1$.
- Describe the normal modes (i.e. eigenvectors) in words.
    - By this we mean describe them the way we described the two modes in class — in one case, both mass move with the same amplitude, in the other they move with equal and opposite displacements.
- Graph the motion for each of the eigenvectors and discuss.
- Graph the motion for each of these two initial conditions and explain why they are different:
    - $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$
    - $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$

What to turn in at the end of lab:

- Submit the Python code to github. USE THE NAMES OF THE FUNCTIONS/FILES WE HAVE ALREADY DEFINED.

You should save a copy of the code on a flash drive or email the program to yourself. Your work will be evaluated and returned by Friday. Revisions will be due the beginning of next lab.