1 Balls attached by springs (under tension)

We will begin by studying a system of balls and springs under tension. This discrete system will help us to understand the continuum wave equation. Consider an sequence of balls attached by springs, with the whole thing under tension T: Missing `/var/www/paradigms_media_2/media/activity_media/balls-and-springs.pdf` We will consider *transverse* oscillations, in which the balls move in the y direction, and their attachment is in the x direction. For small displacements, the equation of motion of each ball is very simple:

Missing `/var/www/paradigms_media_2/media/activity_media/ball-spring-force.pdf`

$$\begin{aligned}\vec{F}_i \cdot \hat{y} &= -T\sin(\theta_L) - T\sin(\theta_R) && (1)\\ &\approx -T\left(\frac{\Delta y_R}{\Delta x} + \frac{\Delta y_L}{\Delta x}\right) && (2)\\ &= \tfrac{T}{\Delta x}\left(y_{i+1} + y_{i-1} - 2y_i\right) && (3)\\ &= m\vec{a}_i \cdot \hat{y} && (4)\\ &= m\frac{d^2 y_i}{dt^2} && (5)\end{aligned}$$

or, solving for the second derivative, we have

$$\frac{d^2 y_i}{dt^2} = \tfrac{T}{m\Delta x}\left(y_{i+1} + y_{i-1} - 2y_i\right) \tag{6}$$

At this point, we can discretize the time dimension with time step Δt, and when we write the second derivative as a finite difference, we find

$$\frac{y_i(t+\Delta t) + y_i(t-\Delta t) - 2y_i(t)}{\Delta t^2} \approx \tfrac{T}{m\Delta x}\left(y_{i+1}(t) + y_{i-1}(t) - 2y_i(t)\right) \tag{7}$$

Now by solving for the "future" value of the position $y_i(t+\Delta t)$, we obtain the Verlet method for these coupled balls:

$$y_i(t+\Delta t) = 2y_i(t) - y_i(t-\Delta t) + \tfrac{T\Delta t^2}{m\Delta x}\left(y_{i+1}(t) + y_{i-1}(t) - 2y_i(t)\right) \tag{8}$$

This gives us a formula from which we can obtain the future position if we know the current position and previous position.

1. Create a 2D array for $y_i(t)$, where the two dimensions correspond to ball number (i) and the time t. Write code to find all the positions at all future times, given the positions at the first two time steps. The position at the first two time steps is the initial conditions. Later, you will want to change the number of balls and the time step Δt, so please plan ahead for that.

2. Create an animation (see below for help) of your simulation, and use it to convince yourself that it is behaving physically. You may freely experiment with initial conditions. Increase the number of balls sufficiently the system clearly looks like a wave system.

extra fun Play with different initial conditions, and try to get your balls to do interesting things!

3. Create a *static* visualization of the time-dependent data you are animating. (There are several effective ways to do this!) The idea here is to imagine that you have to convey the same information that is in your animation, but need to do it without animation (e.g. in a figure in a PDF or on paper). You need to convey both the time-dependence *and* the spatial dependence in a single plot.

1.0.1 Creating animations

Once you've written the above algorithm, you'll want to see what is going on. To do so, write code to animate the string's motion. You could google for *matplotlib animation*, but to save you some time (and because the documentation that is out there for creating animations is confusing), you can use the following example:

```
import numpy as np
import matplotlib.pyplot as plt

L = 1 # meter
k = 10 # per meter
dx = 0.01
x = np.arange(0, L, dx)
v = 3 # meters/second
tmax = 10*L/v
dt = 0.01

for t in np.arange(0, tmax, dt):
    plt.cla() # Clear off what was previously drawn
    plt.plot(x, np.sin(k*(x-v*t)))
    plt.pause(0.001) # Draw to the screen (and wait a moment)

plt.show() # (optional) show the plot after it is done animating
```

There are only two necessary lines here that are related to animation:

1. `plt.cla()`, which clears out anything that had previously been plotted.
2. `plt.pause(0.001)`, which says to draw the plot as it currently is and wait a millisecond. You can pause longer if you want to slow down your animation.

Note: the above animation is just the example code, which is ***not*** *how your system of balls and springs will behave.*

1.1 Normal modes

A "normal mode" is a set of initial conditions for which the *shape* of the set of balls does not change over time, but only its amplitude. In other words, the balls will all oscillate in a single repeating pattern.

1. Try to find a normal mode of the system (which has a repeating pattern). Do this with just two or three moving balls (not counting stationary balls at the borders).
2. What is the period (and frequency) of this normal mode? You do not need to write code to compute the freqency automatically, but could instead just run a simulation and watch it to see what the period is, e.g. by printing the time at each frame of the animation.
3. Try to find *all* the normal modes. How many do you expect? It will help to start with only a few balls.
4. Find the frequency of your normal modes.

Extra fun Plot the frequency versus $1/\lambda$ where λ is the wavelength. This is called the **dispersion relation**, and will be discussed in Periodic Systems.