Fourier Transform

The Fourier method was named after a French mathematician and physicist Joseph Fourier. The Fourier method is used to study heat transfer. The basic idea is to express some complicated functions as the infinite sum of sine and cosine waves. Similar to how we decompose a function using a Taylor series which expresses the function with an infinite sum of polynomials.

The Fourier method is used in signal processing, partial differential equations, image processing, etc. It was chosen as one of the 10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century.

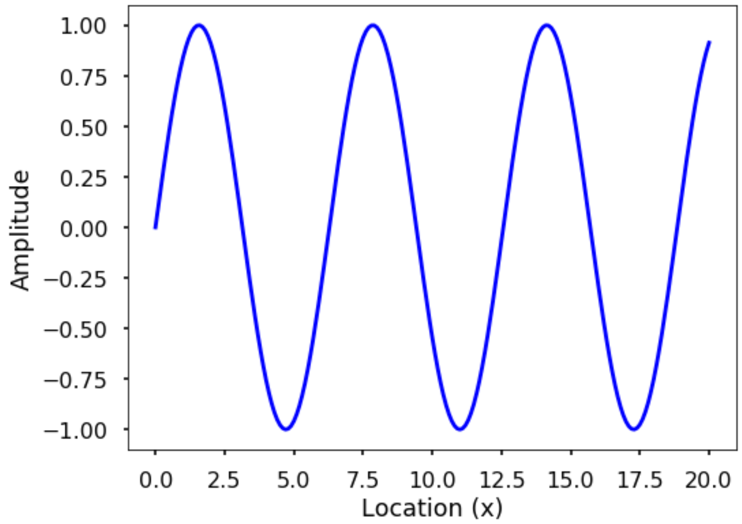
1. The basics of Waves

There are many types of waves in our life. If we throw a rock into a pond, we can see the waves form and travel in the water. Some other waves are sound, earthquake, and microwaves which are difficult to see. In physics, a wave is a disturbance that travels through space and matter with a transferring energy from one place to another.

Model a wave using mathematical tool

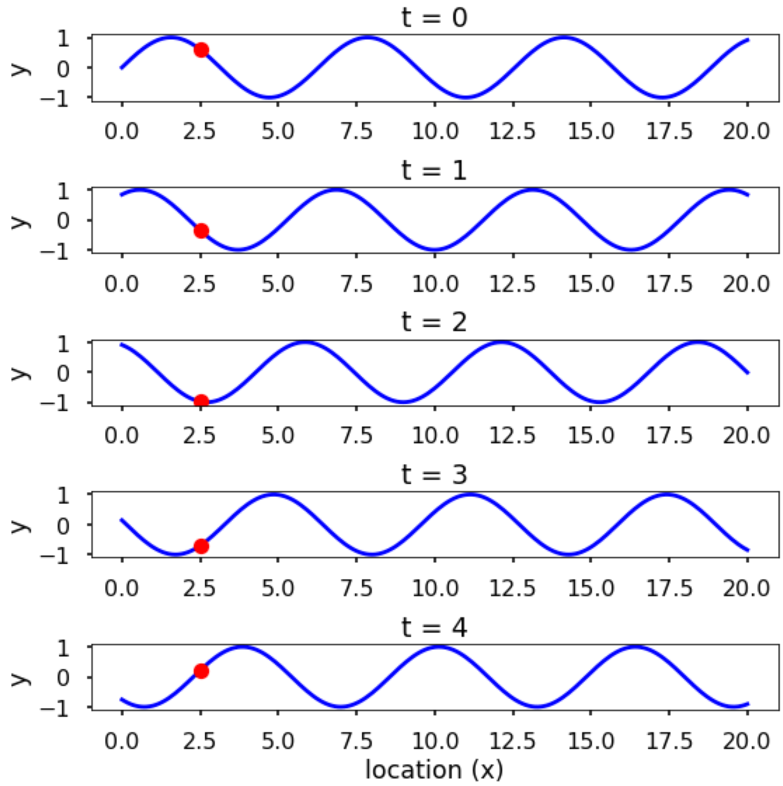
A single wave can be modeled as a field with a function F(x,t), where x is the location of a point in a space, and t is the time. A simple case can be seen in a sine wave change over x.

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

plt.style.use('seaborn-poster')
%matplotlib inline



x = np.linspace(0, 20, 201)
y = np.sin(x)

plt.figure(figsize = (8, 6))
plt.plot(x, y, 'b')
plt.ylabel('Amplitude')
plt.xlabel('Location (x)')
plt.show()
```



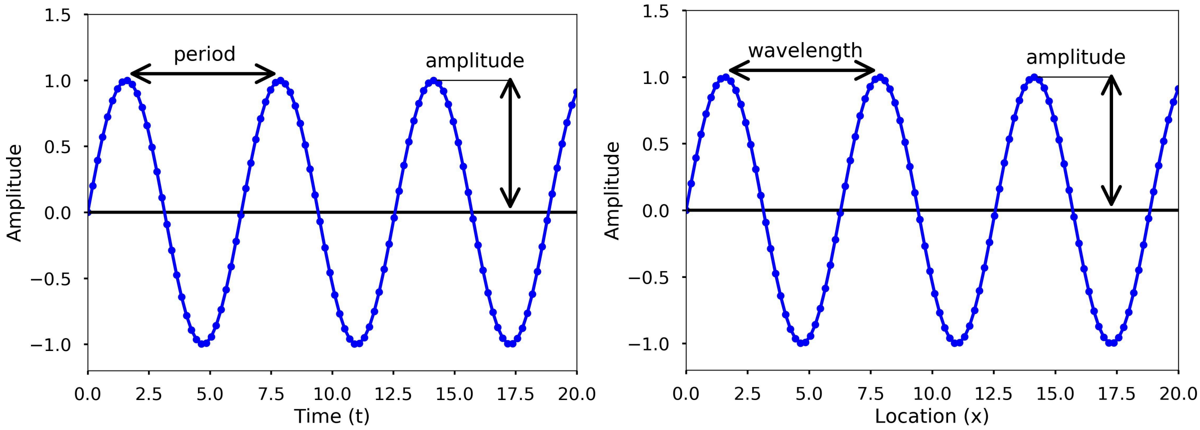
The sine wave can change both in time and space. If we plot changes at various locations, each time snapshot will be a sine wave changes with location. See the following figure with a fix point at x=2.5 showing as a red dot. Of course, you can see the changes over time at specific locations as well, you can plot this by yourself.

```
fig = plt.figure(figsize = (8,8))

times = np.arange(5)

n = len(times)

for t in times:
    plt.subplot(n, 1, t+1)
    y = np.sin(x + t)
    plt.plot(x, y, 'b')
    plt.plot(x[25], y [25], 'ro')
    plt.ylim(-1.1, 1.1)
    plt.ylabel('y')
    plt.title(f't = {t}')

plt.xlabel('location (x)')
plt.tight_layout()
plt.show()
```



Characteristics of a wave

Waves can be a continuous entity both in time and space. But in reality, many times we discrete time and space at various points. For example, we can use sensors such as accelerometers (to measure the acceleration of a movement) at different locations on the Earth to monitor the earthquakes, which is spatial discretization. The sensors usually record the data at a certain times which is a temporal discretization. For a single wave it has different characteristics.



Amplitude = the difference between the maximum values to the baseline value. A sine wave is a periodic signal, meaning that it repeats itself after a certain time, which can be measured by period.

Period = The time it takes to finish the complete cycle (space between two adjacent peaks)

Wavelength measures the distance between two successive crests or troughs of a wave.

Frequency describes the number of waves that pass a fixed place in a given amount of time. Frequency can be measured by how many cycles pass within 1 second. The unit of frequency is cycles/second, or more commonly used Hertz or Hz.

$$period = 1 \,/\, frequency$$

Looking at the two figures, the blue dots on the sine waves are discretization points we did both in time and space. We only sample the value of the wave on those specific points. When we record a wave, we need to specify how often we sample the wave in time, this is called **sampling**. This rate is called **Sampling rate**, with the unit Hz. For example, If we sample a wave at 2 Hz, it means that every second we have two data points.

$$y(t) = A\ sin(\omega t + \phi)$$

That is a representation of a sine wave.

- A is the amplitude of the wave.
- ω is the angular frequency, which specifies how many cycles occur in a second, in radians per second.
- ϕ is the phase of the signal.

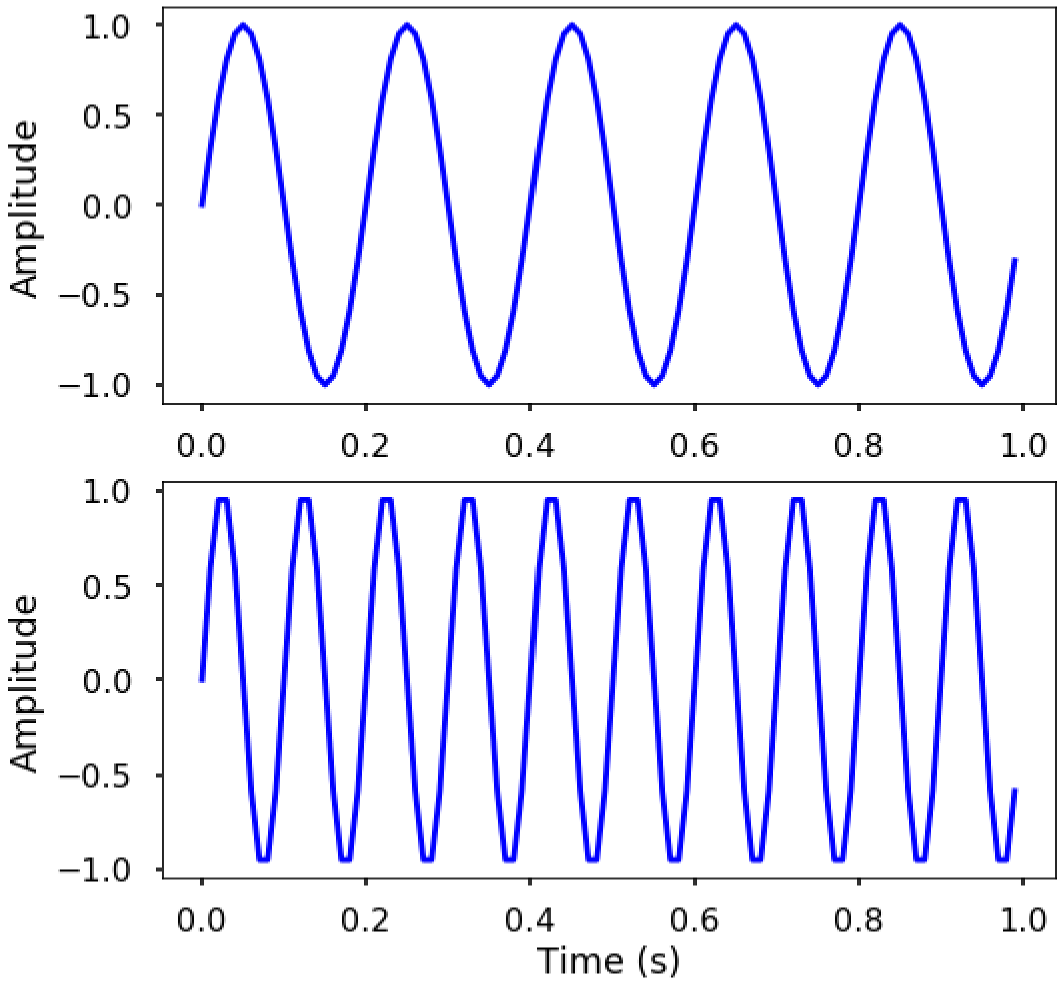
If T is the period of the wave, and f is the frequency of the wave, then w has the following relationship to them:

$$\omega = 2\pi \,/\, T = 2\pi f$$

TRY IT! Generate two sine waves with time between 0 and 1 seconds and frequency is 5 Hz and 10 Hz, all sampled at 100 Hz. Plot the two waves and see the difference. Count how many cycles in the 1 second.

```
# sampling rate
sr = 100.0
# sampling interval
ts = 1.0/sr
t = np.arange(0,1,ts)

# frequency of the signal
freq = 5
y = np.sin(2*np.pi*freq*t)

plt.figure(figsize = (8, 8))
plt.subplot(211)
plt.plot(t, y, 'b')
plt.ylabel('Amplitude')
```

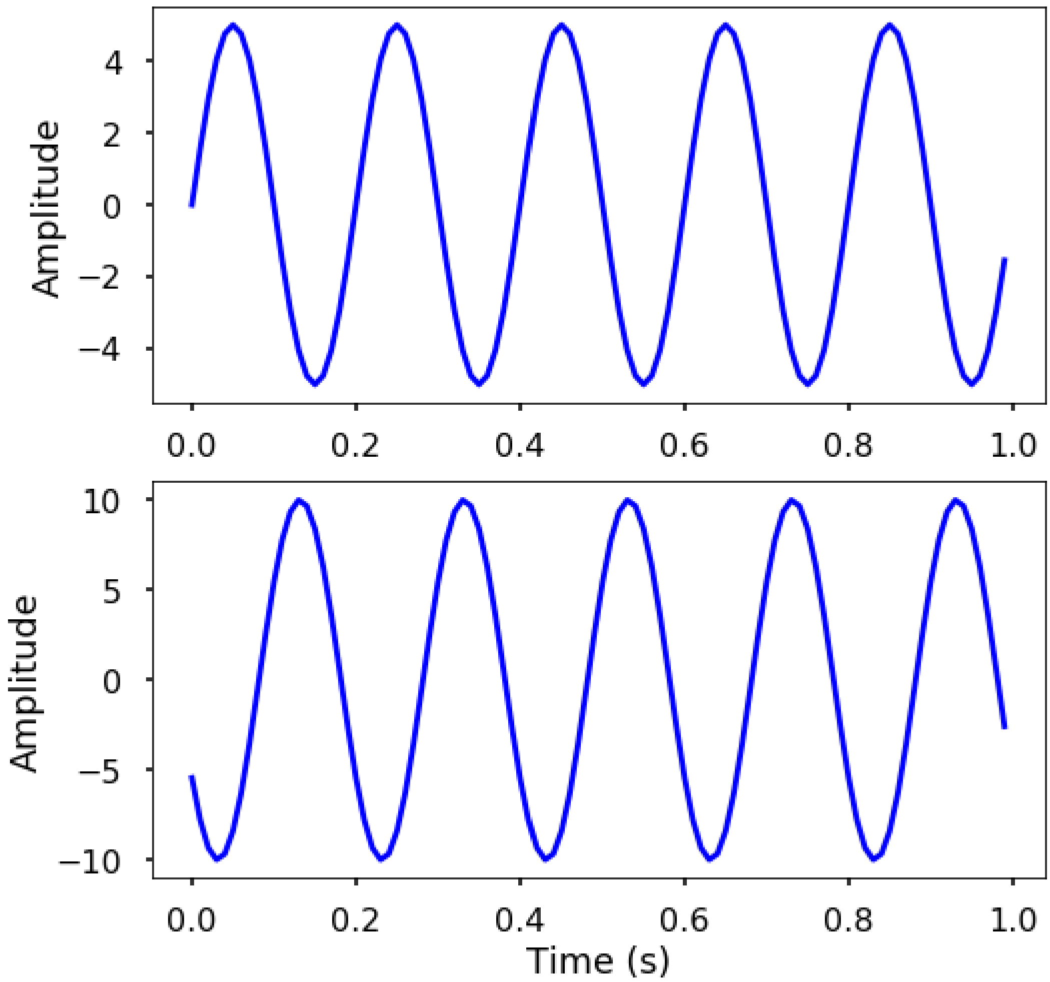
```
freq = 10
y = np.sin(2*np.pi*freq*t)

plt.subplot(212)
plt.plot(t, y, 'b')
plt.ylabel('Amplitude')

plt.xlabel('Time (s)')
plt.show()
```



TRY IT! Generate two sine waves with time between 0 and 1 seconds. Both waves have frequency 5 Hz and sampled at 100 Hz, but the phase at 0 and 10, respectively. Also the amplitude of the two waves are 5 and 10. Plot the two waves and see the difference.

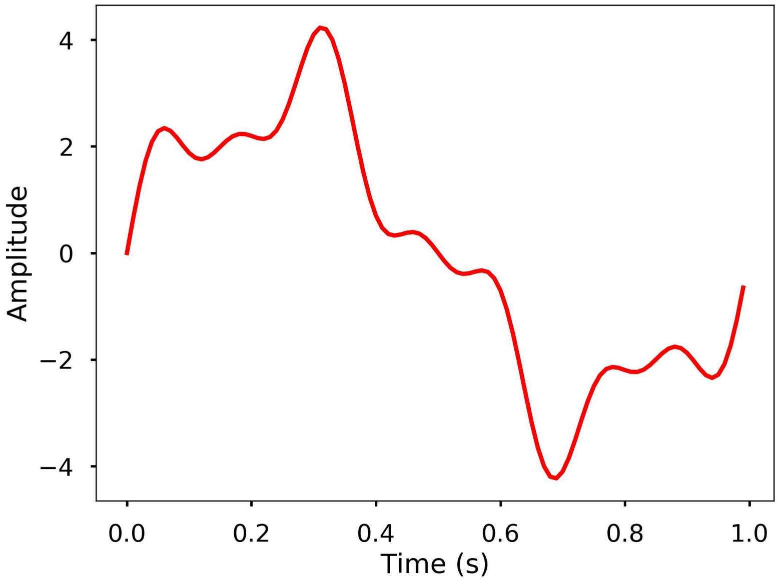
```
# sampling rate
sr = 100.0
# sampling interval
ts = 1.0/sr
t = np.arange(0,1,ts)

# frequency of the signal
```

```
freq = 5
y = 5*np.sin(2*np.pi*freq*t + 0)

plt.figure(figsize = (8, 8))
plt.subplot(211)
plt.plot(t, y, 'b')
plt.ylabel('Amplitude')

y = 10*np.sin(2*np.pi*freq*t + 10)

plt.subplot(212)
plt.plot(t, y, 'b')
plt.ylabel('Amplitude')

plt.xlabel('Time (s)')
plt.show()
```



2. Discrete Fourier Transform (DFT)

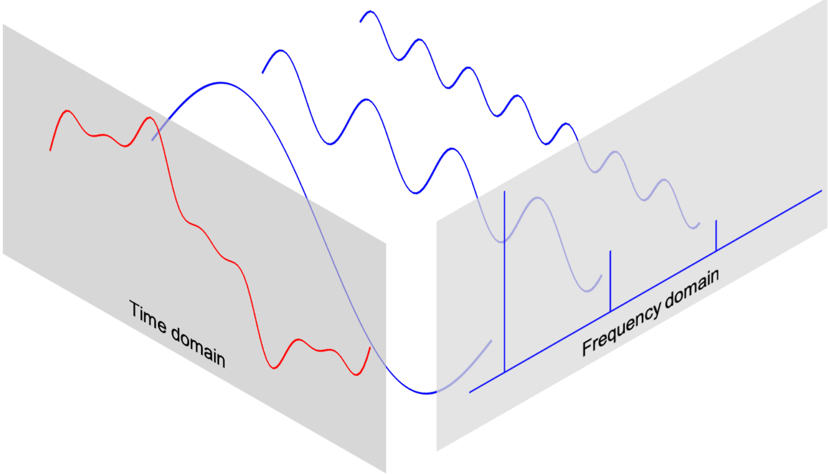
We learned how to characterize a wave with period/frequency, amplitude, phase. For simple periodic signal like sine and cosine waves, these are very easy to spot. For complicated waves, it is very hard to tell. Look at the following wave graph as an example.



In real life scenarios, there will be even more complicated graphs. Therefore, it would be great to have a method to analyze the characteristics of the wave. The **Fourier Transform** is one way of doing this.

The Fourier Transform decomposes any signal into a sum of simple sine and cosine waves that we can easily measure the frequency, amplitude, and phase. The Fourier transform can be applied to continuous or discrete waves.

Discrete Fourier Transform can decompose the above signal to a series of sinusoids and each of them will have a different frequency. The idea is illustrated by the following 3D figure.



The figure shows that the original wave is the result of the sums of 3 different sine waves.
And also, the original wave which is in time domain signal

TEStT