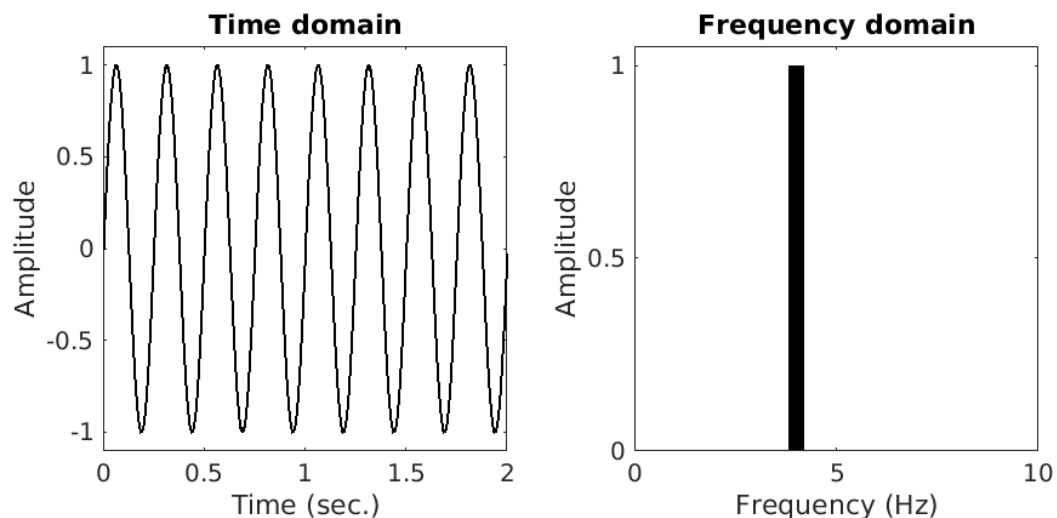
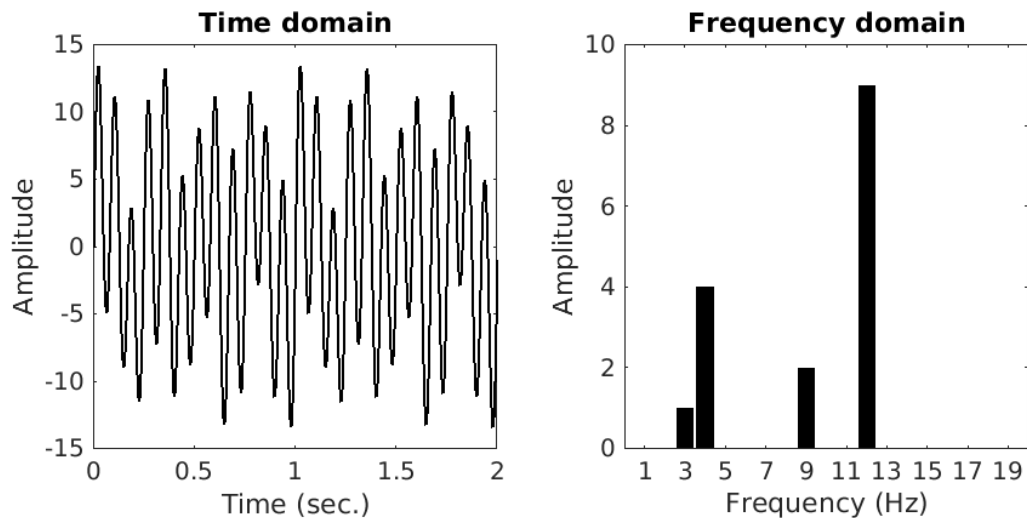


## Course reader: *Nontechnical introduction to the Fourier transform*

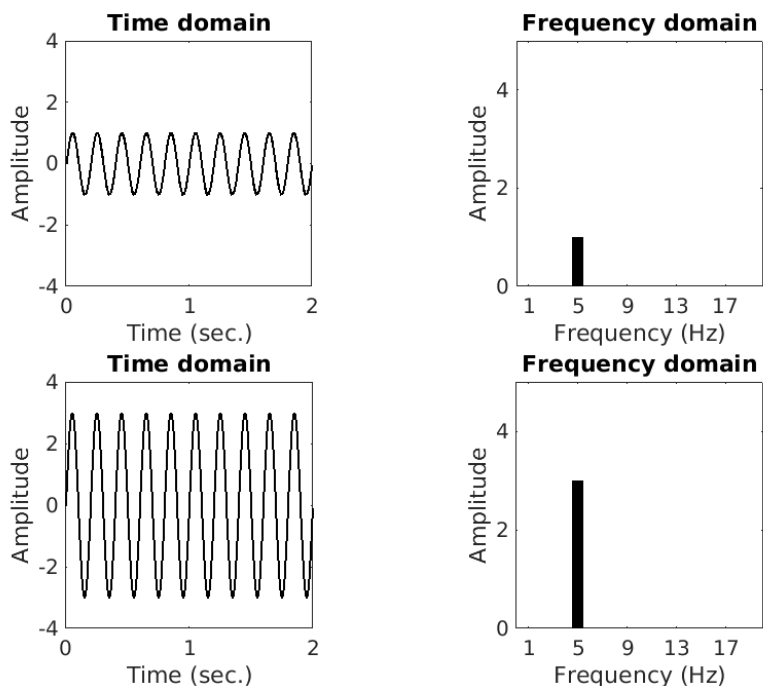
- The Fourier transform is one of the most important algorithms in all of science, mathematics, and communications technology.
- Joseph Fourier developed his theorem in the 19<sup>th</sup> century to characterize heat transfer in metals, but his ideas turned out to be applicable to many other applications.
- In a math class, you would learn about the *continuous* Fourier transform, which involves integrals and functions. In engineering, signal processing, and statistics, you would learn about the *discrete* Fourier transform, which involves sampled sine waves and data vectors on computers. The latter approach is stressed here because it is more amenable to learning by interacting with computers.
- The Fourier Theorem states that **any signal can be perfectly represented as a sum of sine waves, each sine wave having its own phase, frequency, and amplitude**. Basically, we use sine waves to represent a signal.
- This provides an efficient way to transform a signal (e.g., a radio wave) from the *time domain* to the *frequency domain*. The **time domain** is what you are used to seeing (e.g., a radio wave is a fluctuation of signal amplitude over time). The **frequency domain** shows how much energy is in the signal at each frequency (sine wave speed). This allows you to view the individual components of a signal, and therefore provides a better understanding of complicated signals.
- Below is a picture of a sine wave in the time domain (left) and in the frequency domain (right). There are 4 waves in a one-second period, and a corresponding peak at 4 Hz in the power spectrum ("Hz" stands for Hertz and is defined as 1/second).



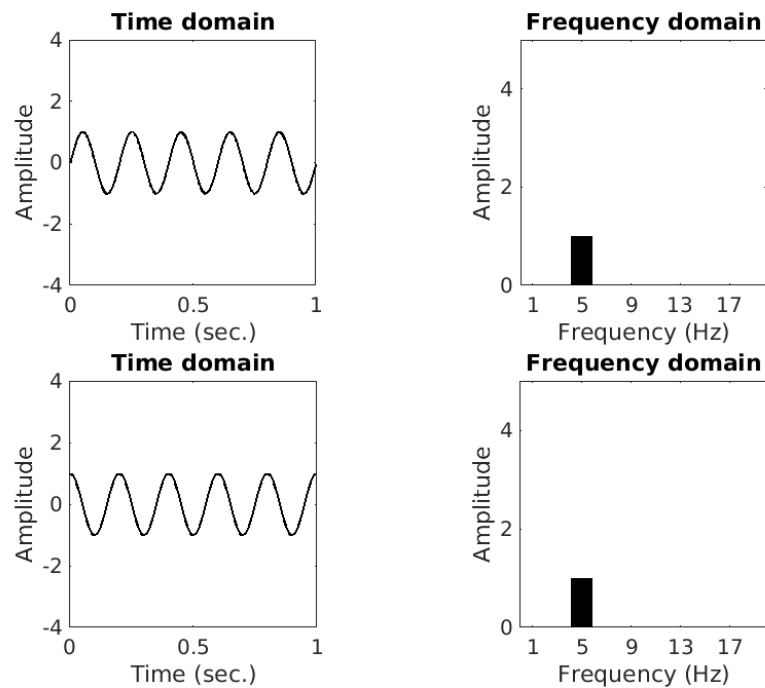
- The picture below shows the insight gained by looking at a signal in the frequency domain. The time-domain signal looks complicated. The frequency-domain representation, however, clearly shows that this signal contains a small number of well-defined rhythmic features.



- The Fourier transform provides two sets of information about the signal, per frequency:
  1. **Power**, a.k.a. amplitude, energy, magnitude: This refers to the size of the deflections of the waves away from the  $y=0$  line. When a system is more "active," the power is higher. Below are two examples.



2. **Phase**: This refers to the value of the sine wave when it crosses  $X=0$ . Phase is a circular property and is measured in radians ( $2\pi$  radians makes a full circle). Below is an example. Notice that *phase offsets have no effect on the amplitude spectrum* (the offset is reflected in the phase spectrum, which is not shown here).

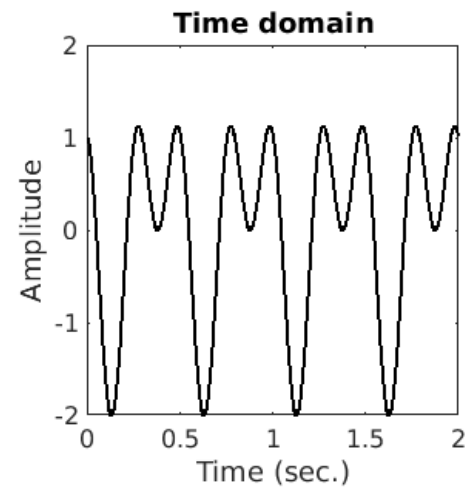
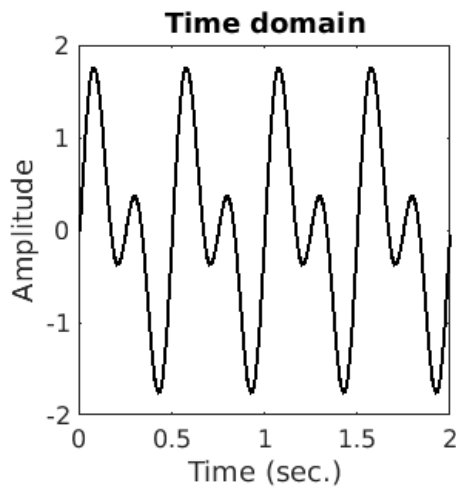
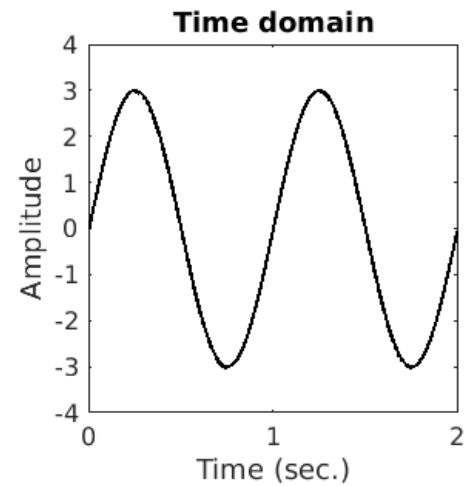
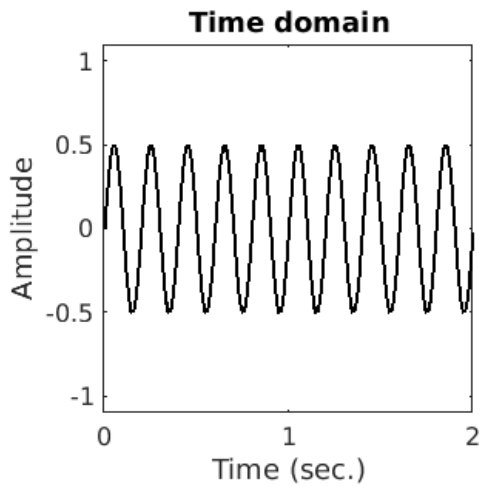


- Broadly speaking, the Fourier transform is used for two purposes:
  1. As a method to inspect frequency-specific energy and phase dynamics of a signal or image, like in the examples above.
  2. As an intermediary tool in an algorithm, such as filtering or convolution. In this case, the goal is not to interpret the results of the Fourier transform, but to use the fast Fourier transform (FFT) to move quickly between the time- and frequency-domains. This in turn is useful because of the convolution theorem, which allows many time-domain operations to be performed in the frequency domain. Frequency-domain operations are usually faster than the corresponding time-domain operation.
- The results of a Fourier transform are easily visually interpreted only for *stationary* signals that contain *rhythmic* components. Stationary means that the signal looks similar at different windows of time. Rhythmic means that the signal contains features that regularly repeat, like a sine wave. Still, the Fourier transform retains all the information of the signal (it is a lossless transform), regardless of the signal properties.
- The Fourier transform can be extended from 1D (such as a radio signal) to 2D (such as an image).
- This course comes with MATLAB code for each section, which can also be directly run in Octave, a free cross-platform environment that interprets MATLAB code. I also provide Python code that matches the MATLAB code (except for some plotting differences). The Python code is written for the Jupyter (formerly iPython) interactive environment, meaning you can run it in your web browser. During the lectures, I only discuss the MATLAB code, mainly in the interest of time.

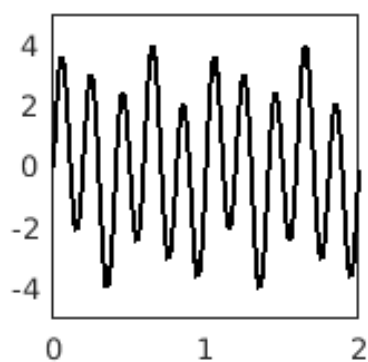
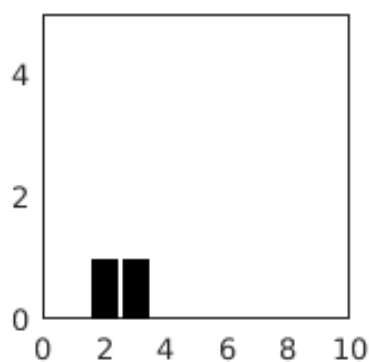
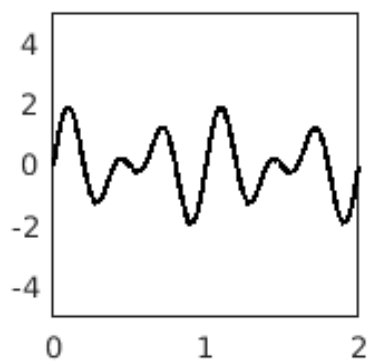
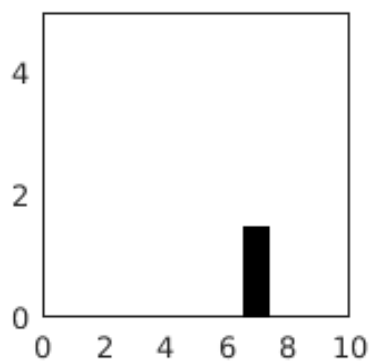
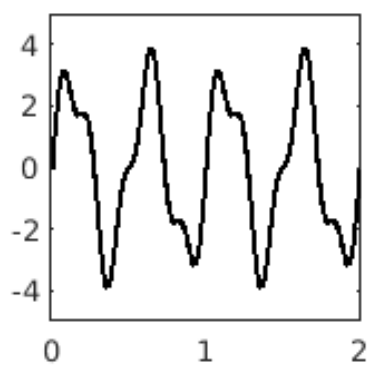
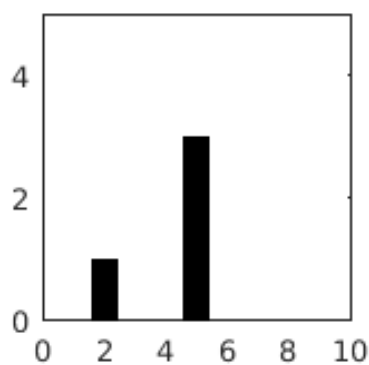
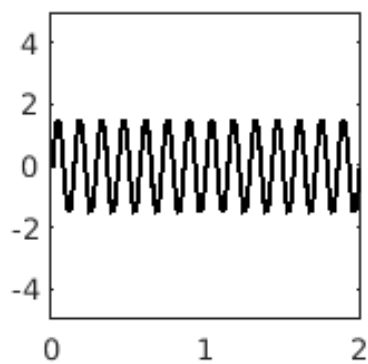
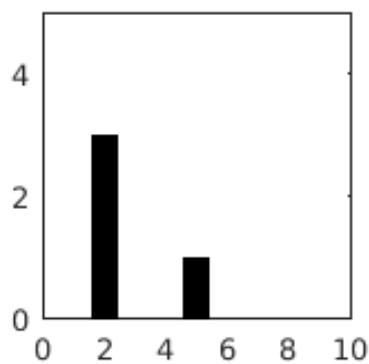
Programming is a useful and underutilized tool in understanding concepts in mathematics. I want you to be able to follow along in the course with either MATLAB or Python. Of course, if you prefer just to watch the videos without following the code, that's also fine.

## Exercises

1. The pictures below show sine waves at various frequencies. Draw their frequency-domain amplitude spectra.

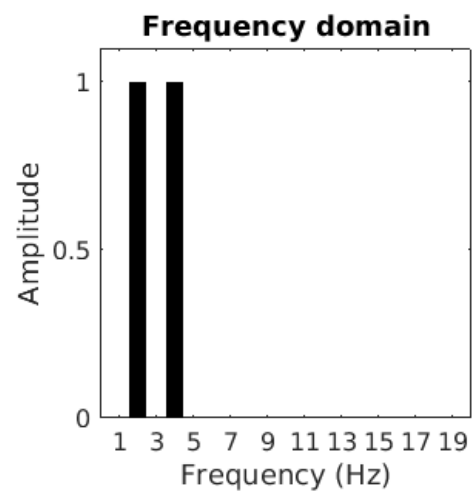
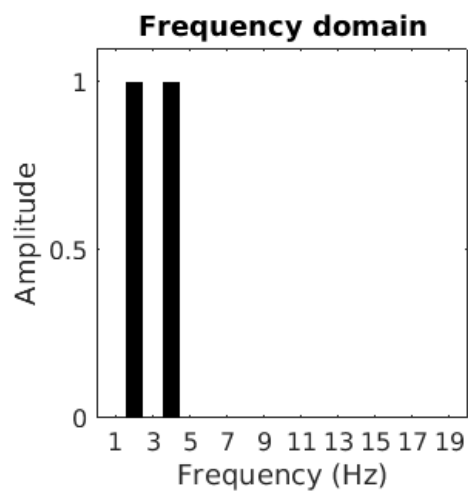
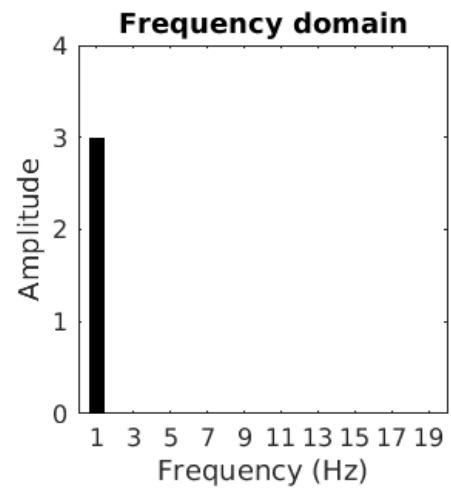
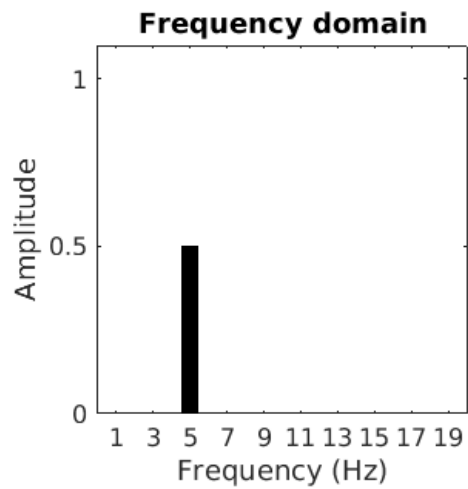


2. Match the amplitude spectrum in the left-hand column with its time-domain signal in the right-hand column. (This is not a trivial problem! Try to work by process-of-elimination.)



## Answers

1. Note that the lower two time series have the same amplitude spectra; they differ only by phase offsets.



2. Numbers indicate matches.

