# PS06

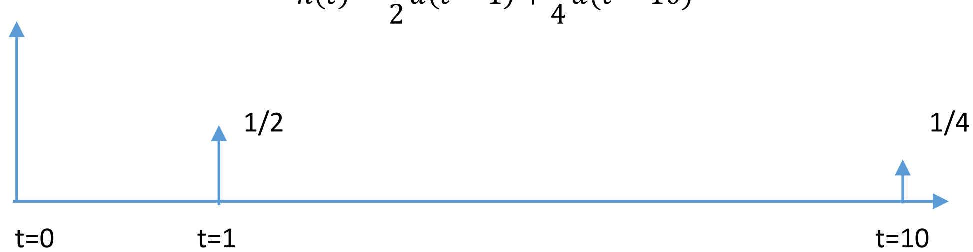
1) Earlier in the course, you saw how the recorded audio signal of a gun being fired in a shooting range can be convolved with a violin recording to approximate how the violin would sound if played in a shooting range. Please explain this using what you know about the impulse and impulse responses.

   For this convolution to produce the desired result, the gun shot has to produce an impulse response which fully classifies the system. As the shot is a singular, large scale impulse, the response will, in turn, fully classify the system. When this response is convolved with the violin recording, the resulting sound is a violin in the firing range where the shot was recorded. The response amplifies the parts of the violin at the same points as the reverb from the shot were recorded, producing the echo sound.

2) Consider a simple model of an echo channel. Suppose that the output of the echo channel is y(t) and the input is x(t), and the input and output are related as follows $y(t) = \frac{1}{2}x(t-1) + \frac{1}{4}x(t\ -\ 10)$. Explain why it is reasonable to call this an echo channel and find an expression for the impulse response of this system, and sketch it.

   This is an echo response because the impulse response of the impulse will be two separate peaks, one at half magnitude at t=1 and one at quarter magnitude at t=10.

$$h(t) = \frac{1}{2}d(t-1) + \frac{1}{4}d(t-10)$$




3)

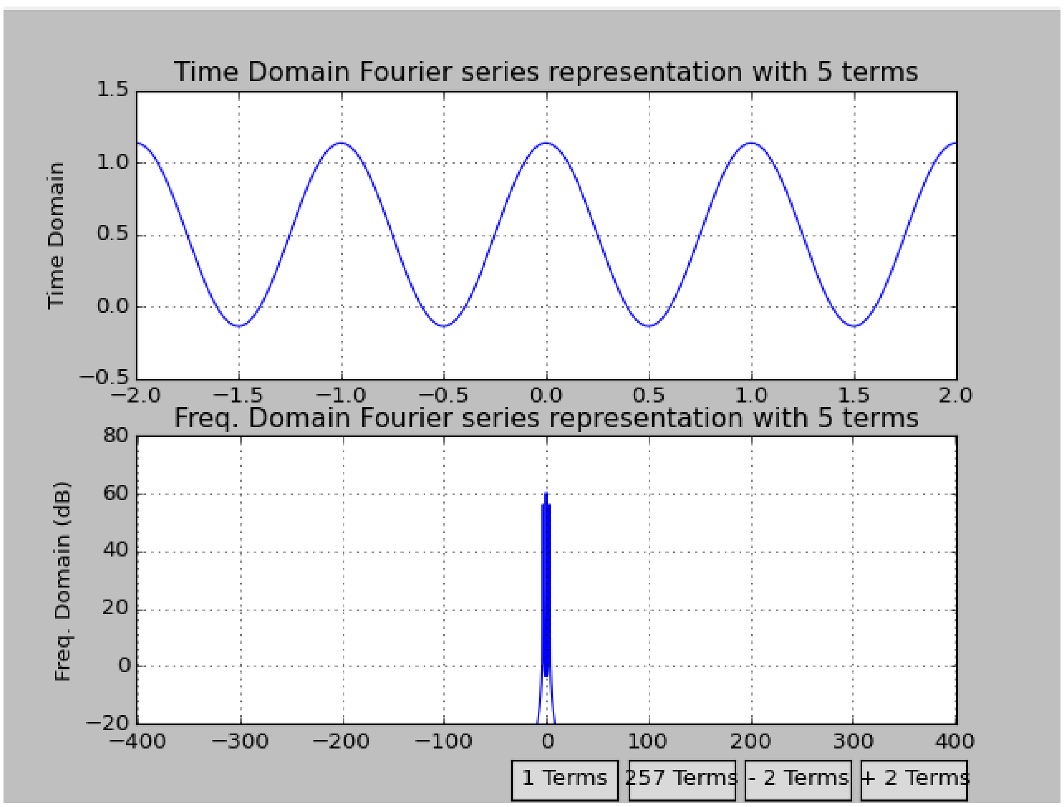
   a. Find the Fourier series representation for the square wave in Figure 1. You may find the identity $sin(\theta) = \frac{1}{2j}e^{j\theta}\ - \frac{1}{2j}e^{-j\theta}$ useful here.

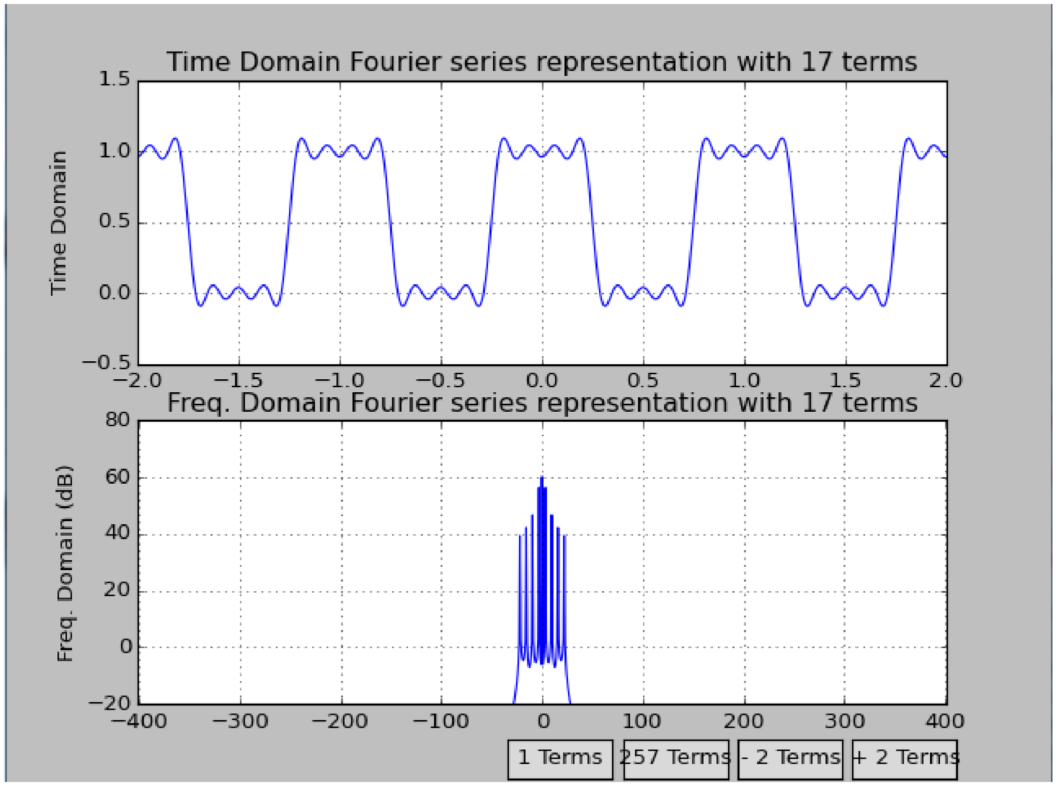
$$C_k = \frac{1}{T}\int_{-\frac{T}{4}}^{\frac{T}{4}} Ae^{(-jk\omega t)}dt$$

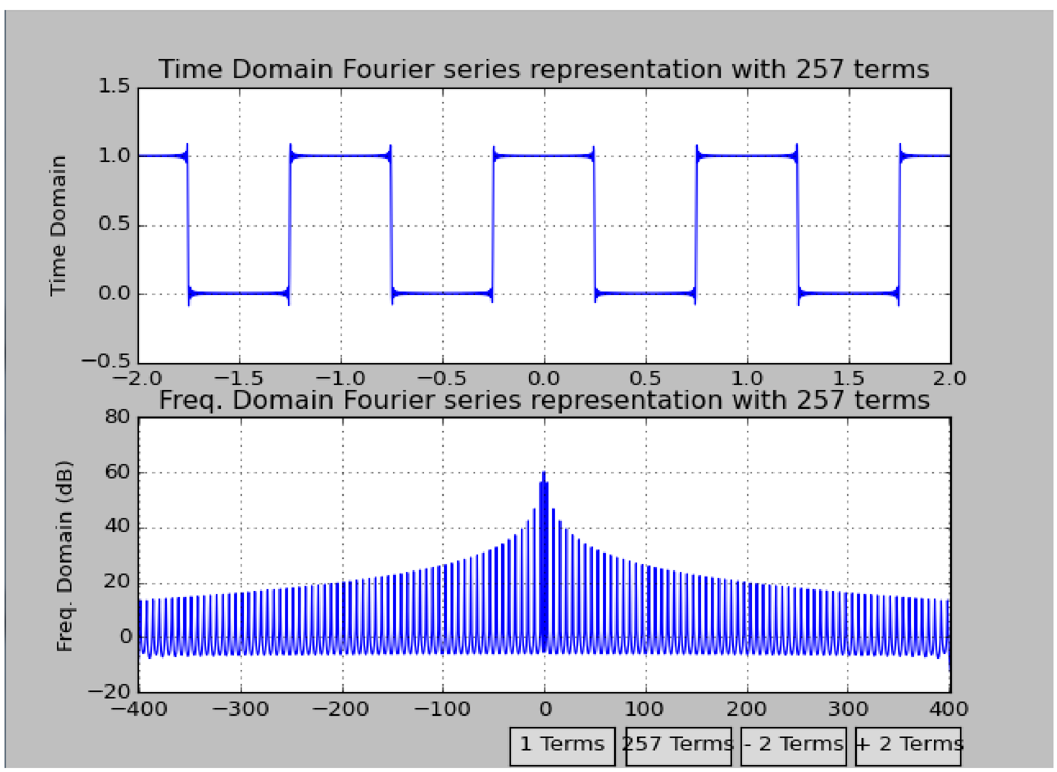
$$= \frac{A}{T}\frac{1}{-j\theta}e^{-jk\omega t}\ |_{-\frac{T}{4}}^{\frac{T}{4}}$$

$$= \frac{A}{-Tjk\omega}\left[e^{-\frac{jk\omega T}{4}} - e^{\frac{jk\omega T}{4}}\right]$$
$$\omega = \frac{2\pi}{T}$$
$$= \frac{A}{-Tjk\frac{2\pi}{T}}\left[e^{-\frac{jk\frac{2\pi}{T}T}{4}} - e^{\frac{jk\frac{2\pi}{T}T}{4}}\right] = \frac{A}{-2\pi jk}\left[e^{-\frac{jk\pi}{2}} - e^{\frac{jk\pi}{2}}\right]$$
$$\theta = k\pi$$
$$= \frac{A}{\theta}\frac{1}{2j}\left[e^{\frac{j\theta}{2}} - e^{-\frac{j\theta}{2}}\right]$$
$$\sin(\theta) = \frac{1}{2j}\left[e^{j\theta} - e^{-j\theta}\right]$$
$$= \frac{A}{\theta}\sin(\theta) = A\,sinc(k)$$

b. Using a computer, plot the Fourier series representation of the square wave in the previous part with fundamental period T = 4, and for 5, 17, and 257 terms in the Fourier series. For clarity, you should plot them in separate subplots/plots. Note that the function $\frac{sin(\pi x)}{\pi x}$ is called the *sinc* function and is available in the numpy package in python.




Time Domain Fourier series representation with 17 terms
Time Domain
Freq. Domain Fourier series representation with 17 terms
Freq. Domain (dB)
1 Terms
257 Terms
- 2 Terms
+ 2 Terms


Time Domain Fourier series representation with 257 terms
Time Domain
Freq. Domain Fourier series representation with 257 terms
Freq. Domain (dB)
1 Terms
257 Terms
- 2 Terms
+ 2 Terms

c. Describe what you see in the Fourier series representation, at the discontinuous points of the square wave, i.e. the points where the square wave goes from 1 to 0 and 0 to 1. How can you reconcile this with (10) in the Fourier series notes? Note: what you should observe is a manifestation of the Gibbs phenomenon, which is caused by the inability of the Fourier series to produce accurate representations of periodic signals at points of discontinuity.

At the discontinuous points, the signal appears to overshoot the correct amplitude and travel beyond 1 or 0 respectively. This is the Gibbs phenomenon.

4)

a. Suppose that x(t) is a periodic signal with fundamental period T, and has a Fourier series representation with coefficients Ck. Consider a new signal, y(t) = x(t−T1), where |T1| < T. Thus y(t) is a delayed version of x(t). Find the Fourier series coefficients for y(t) in terms of Ck. You will find the following property of Fourier series coefficients helpful here

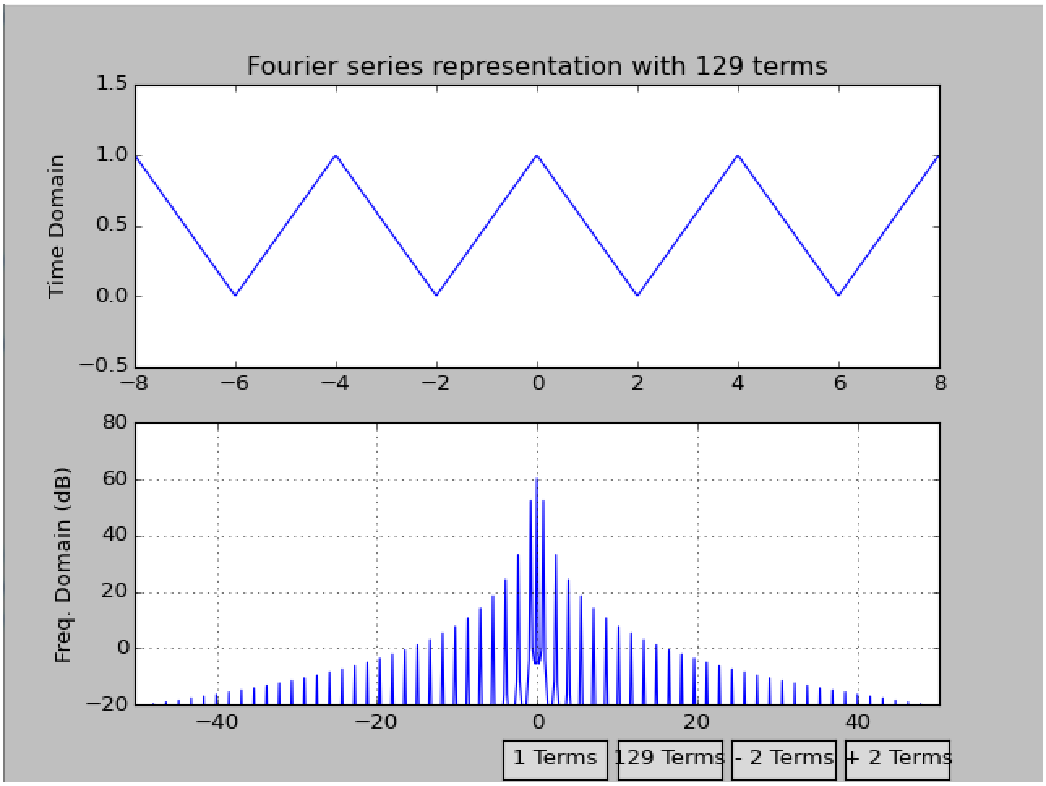
$$C_k = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)e^{\left(-\frac{j(2\pi)}{T}kt\right)}dt$$
$$= \frac{1}{T}\int_{-\frac{T}{2}-W}^{\frac{T}{2}-W} x(t)e^{\left(-\frac{j(2\pi)}{T}kt\right)}dt$$

where W is any number here. The previous equation says that the integral for the Fourier series coefficient can be done over any interval of duration T.

$$C_k = \frac{1}{T}\int_{-\frac{T}{2}-T_1}^{\frac{T}{2}-T_1} x(t-T_1)e^{\left(-\frac{j(2\pi)}{T}kt\right)}dt$$

$$let\ u = t - T_1$$
$$C_k = \frac{1}{T}\int_{-\frac{T}{2}-(u-t)}^{\frac{T}{2}-(u-t)} x(u)e^{\left(-\frac{j(2\pi)}{T}k(u+T_1)\right)}du$$
$$e^{-\frac{j2\pi KT_1}{T}}\frac{1}{T}\int_{-\frac{T}{2}-(u-t)}^{\frac{T}{2}-(u-t)} x(u)e^{\left(-\frac{j(2\pi)}{T}k(u)\right)}du$$

b. Using your answer above, find the Fourier series coefficients for the triangle wave in Figure 2. Verify that your answer is correct by modifying and running the code for the Fourier series of the triangle wave that you used in class. Please turn in a listing of your code and a plot of the triangle wave you generated.


Fourier series representation with 129 terms
Time Domain
Freq. Domain (dB)
1 Terms
129 Terms
- 2 Terms
+ 2 Terms

```
def fs_triangle(ts, M=3, T=4):
    # computes a fourier series representation of a triangle wave
    # with M terms in the Fourier series approximation
    # if M is odd, terms -(M-1)/2 -> (M-1)/2 are used
    # if M is even terms -M/2 -> M/2-1 are used

    # create an array to store the signal
    x = np.zeros(len(ts))

    # if M is even
    if np.mod(M,2) ==0:
        for k in range(-int(M/2), int(M/2)):
            # if n is odd compute the coefficients
            if np.mod(k, 2)==1:
                Coeff = -2/((np.pi)**2*(k**2))
            if np.mod(k,2)==0:
                Coeff = 0
            if n == 0:
                Coeff = 0.5
            x = x + Coeff*np.exp(1j*2*np.pi/T*k*ts)*np.exp(1j*np.pi*k)

    # if M is odd
    if np.mod(M,2) == 1:
        for k in range(-int((M-1)/2), int((M-1)/2)+1):
           # if n is odd compute the coefficients
            if np.mod(k, 2)==1:
                Coeff = -2/((np.pi)**2*(k**2))
            if np.mod(k,2)==0:
                Coeff = 0
            if k == 0:
                Coeff = 0.5
            x = x + Coeff*np.exp(1j*2*np.pi/T*k*ts)*np.exp(1j*np.pi*k)

    return x
```