1 Even and Odd Parts

We would like to express a discrete-time signal \( x[n] \) as the sum of an even part \( x_e[n] \) where

\[
x_e[-n] = x_e[n]
\]

and an odd part \( x_o[n] \) where

\[
x_o[-n] = -x_o[n].
\]

**Part 1.** Is such a decomposition possible for all possible signals \( x[n] \)? Prove your result.

When the decomposition is possible, is the answer always unique? Prove your result.

Assume that \( x[n] \) can be written as the sum of even and odd parts:

\[
x[n] = x_e[n] + x_o[n].
\]

Then

\[
x[-n] = x_e[-n] + x_o[-n] = x_e[n] - x_o[n].
\]

For every value of \( n \), the previous two equations provide two constraints on two unknowns: \( x_e[n] \) and \( x_o[n] \). Solving, we find that

\[
x_e[n] = \frac{x[n] + x[-n]}{2} \quad \text{and} \quad x_o[n] = \frac{x[n] - x[-n]}{2}.
\]

The case when \( n = 0 \) looks special, but is not. There are still two unknowns, \( x_e[0] \) and \( x_o[0] \), but now just one known quantity \( x[0] \), which appears in two different equations:

\[
x[0] = x_e[0] + x_o[0] = x[-0] = x_e[0] - x_o[0]
\]

so that there is still a single unique solution: \( x_e[0] = x[0] \) and \( x_o[0] = 0 \).

Since these solutions always exist (for well-behaved signals), the decomposition into even and odd parts is always possible.

Since the equations are linear and independent, there is a single solution. Therefore the decomposition into even and odd parts is always unique.
Part 2. Let \( x \) represent the signal whose samples are given by
\[
x[n] = \begin{cases} 
(\frac{1}{2})^n & \text{if } n \geq 0 \\
0 & \text{otherwise}
\end{cases}.
\]

Determine the even and odd parts of the signal \( x \) and plot your results.

Add these equations and divide by 2 to obtain
\[
x_e[n] = \frac{x[n] + x[-n]}{2} = \begin{cases} 
\frac{1}{2} (\frac{1}{2})^n & \text{if } n > 0 \\
\frac{1}{2} & \text{if } n = 0 \\
\frac{1}{2} (\frac{1}{2})^{-n} & \text{if } n < 0
\end{cases}
\]

then subtract these equations and divide by 2 to obtain
\[
x_o[n] = \frac{x[n] - x[-n]}{2} = \begin{cases} 
\frac{1}{2} (\frac{1}{2})^n & \text{if } n > 0 \\
0 & \text{if } n = 0 \\
-\frac{1}{2} (\frac{1}{2})^{-n} & \text{if } n < 0
\end{cases}
\]

\[ x_e[n] \quad n \]
\[ x_o[n] \quad n \]
2 CT transformations

Let \( x(t) \) represent the signal shown in the following plot.

![Signal Plot]

The signal is zero outside the range \(-2 < t < 2\).

**Part 1.** Sketch \( y_1(t) = x(t - 1) \).

Set \( t = 0 \). Then \( y_1(0) = x(-1) \). Set \( t = 1 \). Then \( y_1(1) = x(0) \). Set \( t = 2 \). Then \( y_1(2) = x(1) \). These examples make it clear that \( y_1(t) \) is a delayed version of \( x(t) \).

![Sketch y1(t)]

What is the range of \( t \) for which \( y_1(t) \neq 0 \)?

**range:** \((-1 < t < 0.5) \) or \((0.5 < t < 3)\)

**Part 2.** The following plot shows \( y_2(t) \), which is a signal that is derived from \( x(t) \).

![Signal Plot]

Determine an expression for \( y_2(t) \) in terms of \( x(\cdot) \).

\[
y_2(t) = x(1 - t)
\]
Part 3. Let $y_3(t) = x(2t + 3)$. Determine all values of $t$ for which $y_3(t) = 1$.

Range of $t$: $\left[-\frac{3}{2}, -\frac{1}{2}\right]$.

$x(t) = 1$ for $0 \leq t < 2$. Therefore $y_3(t) = 1$ if $0 < 2t + 3 < 2$, i.e., $-\frac{3}{2} \leq t < -\frac{1}{2}$.

Part 4. Let $y_4(t) = x(at + b)$. Determine all values of $a$ and $b$ for which $y_4(1) = 1$.

Values of $a, b$: $0 \leq a + b < 2$.

Sketch the region in the $a$-$b$ plane for which $y_4(1) = 1$. 
Part 5. Assume that $x(t)$ can be written as the sum of an even part
\[ x_e(t) = x_e(-t) \]
and an odd part
\[ x_o(t) = -x_o(-t). \]

For what values of $t$ is $x_e(t) = 0$?

Values of $t$: \[ |t| \geq 2 \text{ or } |t| = 1 \]

Let $x(t) = x_e(t) + x_o(t)$. Then $x(-t) = x_e(-t) + x_o(-t)$. By the definitions of even and odd, it follows that $x(-t) = x_e(t) - x_o(t)$. Add this to the first equation to get $x(t) + x(-t) = 2x_e(t)$. Thus
\[ x_e(t) = \frac{1}{2} (x(t) + x(-t)) \]
is uniquely determined by $x(t)$. The function $x_e(t)$ is plotted below.

From the plot, it is clear that $x_e(t) = 0$ if $|t| > 2$ or $|t| = 1$. Since $x(t)$ is defined to be zero at $t = \pm 2$, we should include those points as well, so $|t| \geq 2$ or $|t| = 1$. 
3 Complexity

Part 1. Use the series expansions below to verify Euler’s formula. Show your work.

\[ e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots \]

\[ \cos x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots \]

\[ \sin x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots \]

We want to show that \( \cos \theta + j \sin \theta = e^{j\theta} \). Starting with the expansion above, we have:

\[
e^{j\theta} = 1 + j\theta + \frac{j^2\theta^2}{2!} + \frac{j^3\theta^3}{3!} + \frac{j^4\theta^4}{4!} + \frac{j^5\theta^5}{5!} + \frac{j^6\theta^6}{6!} + \frac{j^7\theta^7}{7!} + \ldots
\]

\[= 1 + j\theta - \frac{\theta^2}{2!} - \frac{j\theta^3}{3!} + \frac{\theta^4}{4!} + \frac{j\theta^5}{5!} - \frac{\theta^6}{6!} - \frac{j\theta^7}{7!} + \ldots\]

Grouping together terms that have a \( j \) gives:

\[e^{j\theta} = \left(1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \ldots\right) + \left(j\theta - \frac{j\theta^3}{3!} + \frac{j\theta^5}{5!} - \frac{j\theta^7}{7!} + \ldots\right)\]

Pulling out a factor of \( j \):

\[e^{j\theta} = \left(1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \ldots\right) + j \left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \ldots\right)\]

The parenthesized series above are the expansions of \( \cos \theta \) and \( \sin \theta \), respectively, so we have:

\[e^{j\theta} = \cos \theta + j \sin \theta\]

Part 2. Use Euler’s formula to show that \( e^{j\theta} + e^{-j\theta} = 2 \cos \theta \).

By Euler’s formula,

\[e^{j\theta} + e^{-j\theta} = \cos \theta + j \sin \theta + \cos(-\theta) + j \sin(-\theta)\]

Since cosine is even, \( \cos(-\theta) = \cos \theta \). And since sine is odd, \( \sin(-\theta) = -\sin \theta \). Substituting:

\[e^{j\theta} + e^{-j\theta} = \cos \theta + j \sin \theta + \cos \theta - j \sin \theta = 2 \cos \theta\]
Part 3. Use Euler’s formula to show that $e^{j\theta} - e^{-j\theta} = 2j \sin \theta$. You may use your result from Part 2 if that is helpful.

By Euler’s formula,

$$e^{j\theta} - e^{-j\theta} = \cos \theta + j \sin \theta - \cos(-\theta) - j \sin(-\theta)$$

Since cosine is even, $\cos(-\theta) = \cos \theta$. And since sine is odd, $\sin(-\theta) = -\sin \theta$. Substituting:

$$e^{j\theta} + e^{-j\theta} = \cos \theta + j \sin \theta - \cos \theta + j \sin \theta = 2j \sin \theta$$
**Part 4.** Find the real and imaginary parts of the following, without using a calculator. Show your work.

1. $3e^{j\pi/3} + 4e^{-j\pi/6}$

Adding complex numbers tends to be easiest in rectangular form, so we will first convert the two numbers to rectangular form.

\[
3e^{j\pi/3} = 3 \cos(\pi/3) + 3j \sin(\pi/3) = \frac{3}{2} + \frac{3\sqrt{3}}{2}j
\]

\[
4e^{-j\pi/6} = 4 \cos(\pi/6) - 4j \sin(\pi/6) = \frac{4\sqrt{3}}{2} - 2j
\]

Their sum, therefore, is:

\[
3e^{j\pi/3} + 4e^{-j\pi/6} = \frac{4\sqrt{3} + 3}{2} + \frac{3\sqrt{3} - 4}{2}j
\]

2. $\left(\sqrt{3} + j\right)^{11}$

Exponentiation tends to be easiest in polar form. So we’ll start by converting this number to polar form. This is a number with magnitude 2, and its angle in the complex plane is $\tan^{-1}(1/\sqrt{3}) = \pi/6$.

Therefore, we have:

\[
\left(\sqrt{3} + j\right)^{11} = \left(2e^{j\pi/6}\right)^{11} = 2^{11}e^{j11\pi/6} = 2048e^{-j\pi/6}
\]

Converting back to rectangular form, we have:

\[
\left(\sqrt{3} + j\right)^{11} = 2048e^{-j\pi/6}
\]

\[
= 2048 \left(\cos(-\pi/6) + j \sin(-\pi/6)\right)
\]

\[
= 2048 \left(\cos(\pi/6) - j \sin(\pi/6)\right)
\]

\[
= 2048 \left(\frac{\sqrt{3}}{2} - j \frac{1}{2}\right)
\]

\[
= 1024\sqrt{3} - 1024j
\]

3. $j^3$

$j$ is a number in the complex plane with magnitude 1 and angle $\pi/2$. Thus, we can rewrite $j = e^{j\pi/2}$. Therefore:

\[
j^3 = \left(e^{j\pi/2}\right)^3 = e^{j^2\pi/2} = e^{-\pi/2}
\]

This is a purely real number.
Firstly, let’s figure out an equivalent expression for the value in the parentheses. The $\sqrt{2}e^{j\pi/4}$ and $\frac{2}{\sqrt{2}}e^{j5\pi/4}$ terms have equal magnitude and complementary angles, so they entirely cancel each other out when added together. After this cancellation, we are interested in:

$$\left(\sqrt{2}e^{j\pi/2} + \frac{1}{\sqrt{2}}e^{j\pi} - \frac{1}{\sqrt{2}}j\right)^{218}$$

Since $e^{j\pi} = -1$ and $e^{j\pi/2} = j$, we can write the above as:

$$\left(-\frac{1}{\sqrt{2}} + \sqrt{2}j - \frac{1}{\sqrt{2}}j\right)^{218} = \left(-\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}j\right)^{218}$$

The value in the parentheses is a complex number with a magnitude of 1 and an angle of $3\pi/4$. Thus, the expression about can be rewritten as:

$$\left(e^{j3\pi/4}\right)^{218}$$

This is a complex number whose magnitude is 1 and whose angle is $\frac{\pi \times 3 \times 218}{4}$. We can ignore full revolutions (multiples of $2\pi$ in the angle), so the angle can also be written as $\frac{\pi \times (3 \times 218 \mod 8)}{4} = \frac{\pi \times (654 \mod 8)}{4} = \frac{6\pi}{4} = \frac{3\pi}{2}$.

So ultimately, we have a number whose magnitude is 1 and whose angle is $\frac{3\pi}{2}$. In rectangular form, this number is $-j$. 
4 Mystery Signal

Download the zip archive at
https://sigproc.mit.edu/_static/fall18/pset/01/mystery.zip
and unzip it so you have a local copy of the files contained therein.

This zip archive contains a few files:

- **edp01.py**: a skeleton file for you to work in
- **wav_utils.py**: a file containing utility functions for loading and saving WAV files.
- **coeffs.pkl**: a “pickled” Python object containing some data (see below for details)

`edp01.py` contains code that loads the data from `coeffs.pkl` into two lists.

The first list, `sin_coeffs`, contains coefficients $A_0, A_1, \ldots$. Each represents the amplitude of a pure sine wave: $A_1$ represents the amplitude of a pure sine wave at the fundamental frequency of some signal, $A_2$ the amplitude of a pure sine wave at the second harmonic (double the fundamental frequency), and so on.

Similarly, the second list contains coefficients $B_0, B_1, B_2, \ldots$. Each represents the amplitude of a pure cosine wave: $B_1$ represents the amplitude of a pure cosine wave at the fundamental frequency, $B_2$ the amplitude of a pure cosine wave at the second harmonic, and so on.

Together, these coefficients comprise a representation for an audio signal. Your goal in this exercise is to reconstruct that audio signal, and explain some effects related to modifications we could make to the $A_i$ and $B_i$ coefficients.

Your submission for this problem should include answers to the following questions, as well as any Python code, plots, and/or WAV files that you used to convince yourself of the answers. In particular, answer the questions on the following page.
Questions:

• What sound is contained in the signal represented by these coefficients? Include a WAV file as well as an explanation, and describe the process you used to determine the answer. Note that you can use the \texttt{wav\_write} function to save a wave file from a list of samples.

The signal is a recording is of a person saying, “I paid 5 dollars to hear \textit{that}?"

We can recover the signal by summing several sines and cosines. One important question is how to determine the fundamental period (or fundamental frequency) of the signal. One way to approach this would be to guess-and-check until finding something that sounds recognizable.

Another approach would be to note that, since we have 11,704 terms in each list, we have a DC component, a fundamental, a second harmonic, etc, all the way up to (and including) the 11,703rd harmonic.

A reasonable guess would be that the 11,703rd harmonic is in fact the fastest frequency we can represent in a DT signal. Under this assumption, the 11,703rd harmonic would have a period of 2 samples. Using this information, we could backsolve for the period of the fundamental under this assumption, which would turn out to be $11,703 \times 2 = 23,406$ samples.

This gives us a reasonable guess to start with, and we can use this to construct the the discrete samples in the signal. Then there remains the question of choosing an appropriate sampling rate. This part is really about guess-and check: too fast of a sampling rate, and the recording will sound too fast and high-pitched; too slow, and it will sound slow and low-pitched.

• How is the output different if you use the exact same coefficients but construct an output consisting of too few samples, or more samples than necessary? Explain why this is the case.

Generating fewer than 23,406 samples but using the same coefficients simply results in a portion of the phrase being generated, since we are synthesizing the exact same waveform but generating less than one fundamental period (and one fundamental period contains the entire phrase).

Generating more than one period causes the signal to repeat itself. We should expect this since all of the components we used to construct the signal were periodic in the fundamental period (and so they should repeat).

• How is the output different if you negate all of the $A_i$ coefficients? Explain why this is the case.
Here, we are flipping the odd part of the signal about the horizontal axis. For odd signals, this is equivalent to flipping about the vertical axis. Since the even part is always symmetric over one period and we ultimately flipped the odd part horizontally, we end up with a signal that represents the original sound played backward (i.e., \( x_2(t) = x(-t) \)).

- **How is the output different if each of the \( A_i \) coefficients is set to 0?** Explain why this is the case.

  Setting all of the \( A_i \) coefficients to zero removes the entire odd component of the signal. This leaves us with only the even part of the signal. We should expect that one period, then, looks the same going left-to-right as it does from right-to-left. When we listen to the sound, we hear the original phrase going forward and also something similar to the result from the previous part played at the same time. It does turn out that playing this sound forward or backward sounds the same.

- **How is the output different if you only include the first half of the \( A_i \) and \( B_i \) coefficients (i.e., \( A_i \) and \( B_i \) s.t. \( i \leq L/2 \), where \( L \) is the number of coefficients in each list)?** What if you only include the second half? Explain.

  Including only the first half of the coefficients eliminates much of the high-frequency content. This makes the output sound "muffled," as though it is being heard through a wall or something similar.

  Including only the second half of the coefficients eliminates much of the low-frequency content. While you can still (barely) make out the words, the phrase sounds very "tinny" or "breathy."

- **Many of the coefficients are near 0, but one in particular (\( A_0 \)) is actually 0. How would the output be different if \( A_0 \) were 100 instead?** Why is this the case?

  \( A_0 \) corresponds to a sine wave with 0 frequency (i.e., infinite period). The contribution of this wave to the overall output at time \( n \) is \( A_0 \times \sin(2\pi 0n/(2L)) \), which is always 0 regardless of the value of \( A_0 \). So setting \( A_0 = 100 \) would not actually change anything about the signal.

- **Ben Bitdiddle suggests that, rather than using two coefficient lists, it is possible to use a single list of coefficients \( C_i \) (where \( C_i = A_i + B_i \)) and to reconstruct the original sound using the following sum:**

\[
x[n] = \sum_{k=0}^{L-1} C_k \times e^{j2\pi kn/L}
\]

Is Ben’s reasoning correct? Explain.
Ben is correct that we can represent this signal using complex exponentials, but there are a number of issues with his formulation.

For one thing, Ben needs his exponent to be \( j2\pi kn/(2L) \) instead of \( j2\pi kn/L \), since there are \( 2L \) samples in one fundamental period.

Perhaps more importantly, all of the \( C_i \) values he proposes to use are purely real. If this is the case, though, then at least some of the \( x[n] \) values would be complex-valued. So at the very least, we need different \( C_i \) values to reconstruct the (purely real-valued) signal.

In order to think about what those must be, we can think of this in terms of our original components. In order to make something like \( B_1 \cos(2\pi kn/(2L)) \), we actually need something in the sum that is equivalent to \( (B_1/2)e^{j2\pi n/(2L)} + (B_1/2)e^{-j2\pi n/(2L)} \).

Similarly, to make something like \( A_1 \sin(2\pi n/(2L)) \), we need something in the sum that is equivalent to \( (A_1/2j)e^{j2\pi n/(2L)} - (A_1/2j)e^{-j2\pi n/(2L)} \).

Notice that we can accomplish this by including negative \( k \) values in our sum:

\[
x[n] = \sum_{k=-L+1}^{L-1} C_i e^{j2\pi kn/(2L)}
\]

where

\[
C_i = \begin{cases} 
B_{|i|}/2 - A_{|i|}/2j, & \text{if } i < 0 \\
B_0, & \text{if } i = 0 \\
B_i/2 + A_i/2j & \text{if } i > 0 
\end{cases}
\]

**Debugging Hints:**

This program can take a long time to run! As such, you should avoid debugging using only the provided signal. Here are some tips for debugging:

- **try solving a simpler problem first.**
  - Reduce the number of time samples from thousands to tens.
  - Reduce the number of coefficients from thousands to one or two.
  - Choose small/simple test cases whose output you can easily verify.

- **look at intermediate results.**
  - Use print statements to check your assumptions.
  - Plot intermediate results to get graphical insights into the behavior of your program.