Tune-Up Tuesday #1 for September 3, 2024

% Copy this file into a Matlab script window, add your code and answers to the
% questions as Matlab comments, hit "Publish", and upload the resulting PDF file
% to this page for the tune-up assignment. Please do not submit a link to a file
% but instead upload the file itself. *Late penalty*: 2 points per minute late.

% Please complete each section below.

% Please include your Matlab code as well as any answers to the questions as
% MATLAB comments in the full answer that you submit on Canvas.

% (a) Copy, paste and run the Matlab code from slide 1-16 to generate a cosine signal
% at frequency 440 Hz to play it as an audio signal at a sampling rate of 8000 Hz:

```
f0 =  440;              % 440 Hz (A in 4th octave, or A4)
fs = 8000;              % sampling rate in Hz (a.k.a. samples/s)
Ts = 1/fs;              % sampling time in s
t = 0 : Ts : 3;         % sample times from 0s to 3s
x = cos(2*pi*f0*t);
sound(x, fs);
```

% (b) Modify the code in (a) to change the cosine frequency to 880 Hz and run the code.
% Any difference in what you hear vs. a cosine frequency of 440 Hz?
% *Note: A note in the next higher octave is at twice the frequency. The 'A' note*
% *is 440 Hz in the fourth octave, 880 Hz in the fifth, 1760 Hz in the sixth, etc.*

```
pause(4);               % pause for 4s to prevent sounds from overlapping
f0 =  880;              % change from 440 Hz (A4) to 880 Hz (A5)
fs = 8000;              % sampling rate in samples/s
Ts = 1/fs;              % sampling time in s
t = 0 : Ts : 3;         % 3 seconds in duration
x = cos(2*pi*f0*t);
sound(x, fs);

% Changing f0 from 440 Hz to 880 Hz creates a sinusoidal tone
% at 880 Hz that sounds like a higher frequency tone ("pitch")
% than the cosine at 440 Hz. They are both 'A' notes on the Western
% scale, with 440 Hz in the fourth octave and 880 Hz in the fifth.
% The 880 Hz tone sounds louder than the 440 Hz tone, even though
% each is a sinusoidal signal with amplitude from -1 to 1. One reason
% is that the human auditory system has a stronger response to 880 Hz
% than 440 Hz. See https://en.wikipedia.org/wiki/A-weighting
```

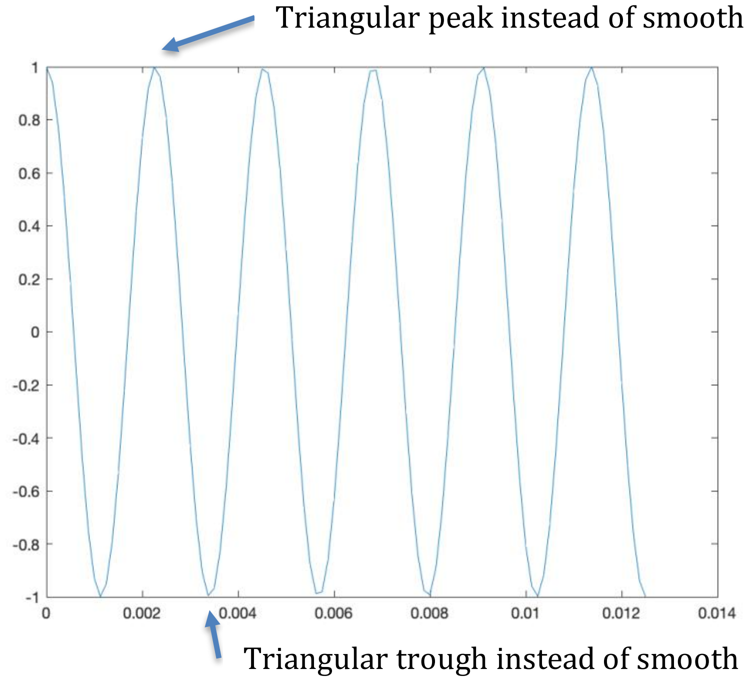
% (c) Using MATLAB, plot the first 12.5 ms of the signal in the time domain using the
% **plot** command.
% Note: The reason part (c) had asked to plot the signal over 12.5ms is so you could see the
% oscillation. Plotting the signal over 3s would have 24000 points, i.e. 3s x 8000 samples/s,
% and plotting 24000 points in a small plot window would blur together into a blue rectangle.
% This part could have been done with f0 = 440 Hz or f0 = 880 Hz.

```
f0 =  440;              % change from 440 Hz (A4) to 880 Hz (A5)
fs = 8000;              % sampling rate in samples/s
Ts = 1/fs;              % sampling time in s
t = 0 : Ts : 12.5*10^(-3);
x = cos(2*pi*f0*t);
plot(t, x);             % see the next page for plots for f0=440 and f0=880
```

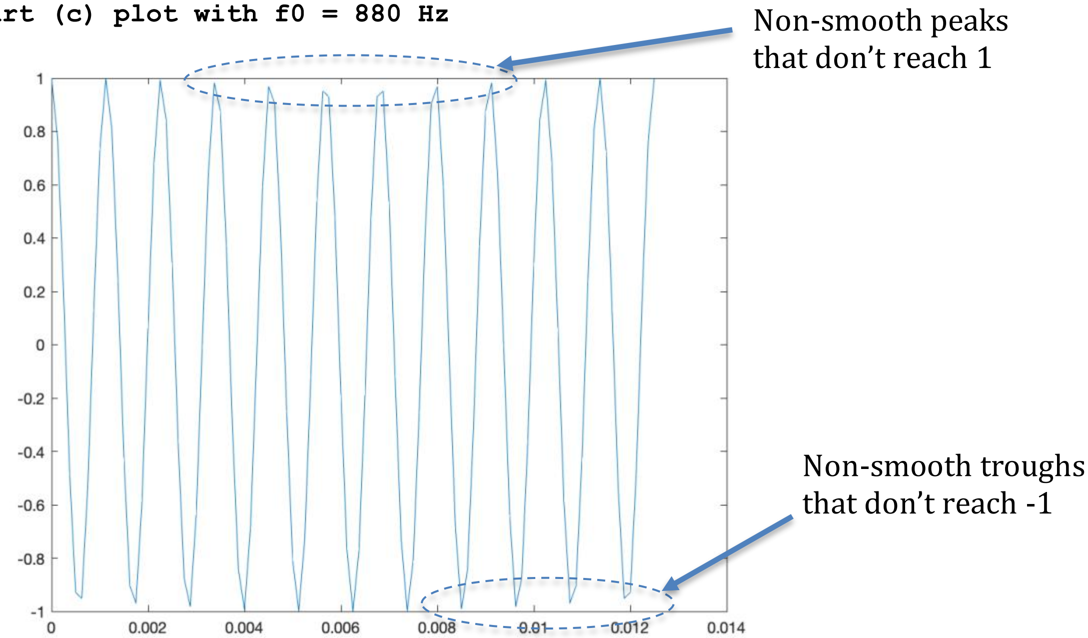
% (d) Describe your plot in (c).

```
% Plot has jagged peaks and troughs. Some peaks don't reach 1, and some
% troughs don't reach -1, because there aren't enough samples. Not periodic.
```

```
% part (c) plot with f0 = 440
```




```
% part (c) plot with f0 = 880 Hz
```



% **Please note:** the plot command "connects the dots" of the amplitude values by
% drawing straight lines whereas an audio playback system uses a special curve
% which gives a more accurate reconstruction of the sampled signal in continuous time