**Tune-Up Tuesday for September 14, 2021**

% This problem uses Fourier series to synthesize a periodic waveform

% one term at a time. The problem revisits homework problem 2.4(a).

% (a) Use the Matlab code below that generates a cosine signal

% x0(t) = 0.5\*cos(2 pi f0 t) with f0 = 440 Hz for 3 seconds at

% a sampling rate of fs = 48000 Hz and play it as a note.

% The gain of 0.5 is to prevent clipping when using the sound command.

fs = 48000; % sampling rate

Ts = 1/fs; % sampling time

tmax = 3; % 3 sec

t = 0 : Ts : tmax;

f0 = 440;

x0 = 0.5\*cos(2\*pi\*f0\*t);

sound(x0, fs);

pause(tmax+1);

% (b) create y(t) = | x0(t) | and play y(t) as sound.

% Describe the difference in the sound in (b) vs. the sound in (a).

*% Answer: The sound in (b) has a higher pitch (frequency) than the*

*% sound in (a). Per part (c), the principal frequency in the sound in (b)*

*% is twice the principal frequency of the sound in (a), i.e one octave higher.*

y = abs(x0);

sound(y, fs);

pause(tmax+1);

% (c) plot y(t) for 5 periods of x0(t).

% How many periods of y(t) are there?

*% Answer: 10 periods of y(t) in the same duration of 5 periods of x0(t).*

*% The fundamental period of y(t) is half that of x0(t) and hence the*

*% the fundamental frequency of y(t) is twice that of x0(t).*

t5periods = 5/f0;

n5periods = round(t5periods/Ts);

figure;

plot( t(1:n5periods), y(1:n5periods) );

% (d) using Fourier series synthesis, use an increasing number

% of terms N = 1, 2, 3, 4, 5, and play each synthesized sound.

% The Fourier series coefficient formulas from homework

% problem 2.4(a) are the following after accounting for the

% gain of 0.5: A0 = 1 / pi and Ak = cos(pi k) / ( pi (1 - 4\*k^2) ).
*% Comment: The values of Ak decay in absolute value at a*

*% rate of 1/k^2. We won’t need many terms for the synthesized*

*% sound to match the sound in (b).*

% Create an array to hold the synthesized sound for efficiency

numSamples = length(t);

synthSound = zeros(1, numSamples);

% Add the first term

A0 = 1/pi;

synthSound = synthSound + A0\*ones(1, numSamples);

f0y = 2\*f0;

for k = 1:5

 % Add in terms for +k and -k

 Ak = cos(pi\*k) / ( pi\*(1 - 4\*k^2) );

 fk = k \* f0y;

 synthSound = synthSound + Ak\*exp(j\*2\*pi\*fk\*t);

 kneg = -k;

 Akneg = cos(pi\*kneg) / ( pi\*(1 - 4\*kneg^2) );

 fkneg = kneg \* f0y;

 synthSound = synthSound + Akneg\*exp(j\*2\*pi\*fkneg\*t);

 sound(synthSound, fs);

 pause(tmax+1);

end

Although not asked, here’s a plot of a cosine at 440 Hz and its absolute value to show that the fundamental period for the absolute value is half of the period for the cosine at 440 Hz; in other words, the fundamental frequency doubles.

*t*

cos(2p (*440Hz*) *t*)

| cos(2p (*440Hz*) *t*) |

Although not asked, here’s a plot of the Ak coefficients as a function of k. In this case, the values of Ak are real-valued and even symmetric.

k = -10 : 10;

Ak = cos(pi\*k)./ ( pi\*(1 - 4\*k.^2) );

stem(k, Ak);



*k*